

Master of Science HES-SO in
Av. de Provence 6
CH-1007 Lausanne

Master of Science HES-SO in Engineering

Orientation : Technologies de l'information et de la communication
(TIC)

Plateforme de distribution MPEG-DASH

Projet « Open-Source Cloud Infrastructure for Encoding and Distribution »

Fait par

David Fischer

Sous la direction de
Prof. Andrés, Revuelta
Institut inIT, hepia

Expert externe : Bram Tullemans, Project Manager Broadband Technology and Online Services of the
Technical Department at EBU/UEB

Genève, HES-SO//Master, 2013

CONTENTS

| | | |
|----------|--|-----------|
| 1 | Introduction | 7 |
| 1.1 | Context | 7 |
| 1.2 | Use Cases and Needs | 8 |
| 1.3 | Proposed Solution | 10 |
| 1.4 | Structure of the Report | 11 |
| 2 | State of the Art | 13 |
| 2.1 | Contents | 13 |
| 2.2 | Overview of Cloud Based Multimedia Platforms | 13 |
| 2.3 | Overview of Compatible Cloud Providers | 14 |
| 2.4 | Overview of OSS Hypervisor Technologies | 15 |
| 2.5 | Overview of OSS Private Cluster/Cloud IaaS | 16 |
| 2.6 | Overview of OSS Cloud Orchestration Tools | 18 |
| 2.7 | Overview of OSS Storage Technologies | 21 |
| 2.8 | Conclusion | 26 |
| 3 | Project Specifications | 27 |
| 3.1 | Contents | 27 |
| 3.2 | Motivation | 27 |
| 3.3 | Work to be done | 27 |
| 3.4 | Development cycles | 28 |
| 3.5 | Time span first cycle | 28 |
| 3.6 | Development phases | 28 |
| 3.7 | Set up | 28 |
| 3.8 | Documentation | 29 |
| 3.9 | Organizational consequences | 29 |
| 3.10 | Future cycles | 29 |
| 3.11 | Future development | 29 |
| 3.12 | Planning | 30 |
| 3.13 | Management | 31 |
| 3.14 | Conclusion | 33 |
| 4 | OSCIED Demonstrator | 34 |
| 4.1 | Contents | 34 |
| 4.2 | Introduction to Layers | 34 |
| 4.3 | Server Layer | 36 |
| 4.4 | Cloud Layer | 37 |
| 4.5 | Clouds Orchestration Layer | 47 |
| 4.6 | Application Layer | 51 |
| 4.7 | Demonstrator FAQ | 83 |
| 4.8 | Deployment Scenarios | 91 |
| 4.9 | Tests and Results | 98 |
| 4.10 | Future Extensions | 103 |

| | |
|--|------------|
| 5 Conclusion | 105 |
| 6 Appendix | 106 |
| 6.1 Abbreviations | 106 |
| 6.2 References | 108 |
| 6.3 TRAC - An Overview | 113 |
| 6.4 OSCIED - Source Code Licensing | 121 |
| 6.5 OSCIED - Orchestra RESTful API | 121 |
| 6.6 OSCIED - Main Scripts | 148 |
| 6.7 OSCIED - JuJu Hacks Source | 172 |
| 6.8 OSCIED - Components Common Library | 182 |
| 6.9 OSCIED - Orchestra Source Code | 204 |
| 6.10 OSCIED - Publisher Source Code | 213 |
| 6.11 OSCIED - Transform Source Code | 221 |
| 6.12 OSCIED - Storage Source Code | 228 |
| 6.13 OSCIED - WebUI Source Code | 230 |
| 6.14 OSCIED - FIMS Scripts | 273 |
| 6.15 OSCIED - OpenStack Nodes Scripts | 276 |
| 6.16 FFmpeg Documentation | 311 |
| HTTP Routing Table | 321 |
| Python Module Index | 322 |
| Index | 323 |

ABOUT

The Author

author David Fischer
address Chemin de Saule 57, 1233 Bernex
contact david.fischer.ch@gmail.com
motto It's not a bug - it's an undocumented feature.

The Thesis

title MPEG-DASH Distribution Platform
school hepia : Haute école du Paysage d'Ingénierie et d'Architecture
organization HES-SO Master of Science in Engineering
domain Multimedia & Informations Technologies
responsible Prof. Andrés Revuelta
date 8th February 2012

The Project

name Open-Source Cloud Infrastructure for Encoding and Distribution
organization European Broadcasting Union (EBU/UER)
project leader Bram Tullemans
main developer David Fischer
license GNU GPLv3
release Basic V1.0
revision 1330

ACKNOWLEDGMENTS

I would like to gracefully thanks Mr. Bram Tullemans for his guidance and support during this work.

I would also thanks the EBU Technical staff and my colleagues at hepia for their interest and support.

Thanks to my brother Michaël, who put his experience in web technologies to help me developing the nice Web User Interface.

Thanks to Dimitri Racordon, an invaluable colleague, experienced in cloud technologies for his help with Open Stack.

Thanks to my PA follower, prof. Andrés Revuelta, for his good mood and for all the professional and personal experience we shared in common with our team on the telecommunications laboratory of hepia.

Thanks to my family, especially to my wife, Claire, for the personal support they provided to my during this work and also during the whole time I followed the Master courses.

... And sorry if I forgot anyone !

Kind Regards,

David Fischer



EXECUTIVE SUMMARY

French

Here is the original executive summary.

| | |
|---------------------------------------|---|
| Titre de Thèse | Plateforme de distribution MPEG-DASH |
| Projet à l'UER | Infrastructure Cloud Open-Source pour la Distribution et l'Encodage |
| Responsable | Revuelta Andrés |
| MRU | TIC / hepia |
| Orientation | TIC |
| Axes technologiques concernées | TIC / Systèmes d'information et multimédia |
| Entreprise | EBU / UER |

Résumé

Le [Cloud computing](#) est un élément clé permettant de rendre une application capable de s'adapter (*élastique*) à la charge en allouant à la volée de nouvelles ressources informatique. Pour cela, il est nécessaire de disposer d'une plateforme [IaaS](#) au sein de l'entreprise. Le projet libre [OpenStack](#) propose ce genre de service. Le projet, développé en collaboration avec l'[EBU](#) consiste en la création d'un outil [Open-Source](#) dédié aux tâches multimédia (transcodage, publication, ...) permettant non seulement d'ajuster l'utilisation des ressources internes à l'entreprise mais aussi de pouvoir monter en charge en utilisant les offres d'[IaaS](#) telles qu' [Amazon AWS](#). L'avantage principal de ce genre d'approche et de rendre l'application *élastique* tout en optimisant les coûts d'utilisation de resources informatiques louées (offre [IaaS](#)) en temps-réel (coût actuel de l'offre, charge des serveurs privés, situation géographique des clients finaux ...).

D'autres extensions sont envisagées comme la possibilité de générer du contenu au format de streaming adaptatif [MPEG-DASH](#).

Cahier des charges

See Also:

Veuillez lire la rubrique [Project Specifications](#) pour de plus amples informations.

- Réaliser un cahier des charges détaillé
- Installer la plateforme de cloud privé [OpenStack](#) sur 4 serveurs Dell (à commander)
- Définir l'architecture de la plateforme [Open-Source](#) de démonstration
- Implémenter la version de base de la plateforme [Open-Source](#) de démonstration

Connaissances préalables

- Très bonnes connaissances du monde [GNU/Linux](#) et capacité à administrer un système [Ubuntu](#)
- Capacité à définir l'architecture logicielle d'un système distribué
- Éléments de transmission et codage numérique multimédia
- Problématique de la transmission en temps réel sur IP

English

| | |
|------------------------------|--|
| Thesis Title | MPEG-DASH Distribution Platform |
| EBU Project | Open-Source Cloud Infrastructure for Encoding and Distribution |
| Responsible | Revuelta Andrés |
| MRU | TIC / hepia |
| Orientation | TIC |
| Technological domains | TIC / Multimedia & informations technologies |
| Company | EBU / UER |

Resume

Cloud computing is one of the key to create scalable applications or services able to scale-up scale-down on demand. In such model, the computing resources are abstracted and the application will consume them as such. In one of the most known scenarios the computing resources are provided by a cloud provider (e.g. [Amazon AWS](#), [HP Cloud](#), ...) as a service and typically delivered over a network such as Internet. The end-user will consume the resources on a remote fashion. One of the most promising [Open-Source](#) project called [OpenStack](#) is another key for any enterprise to convert they internal IT infrastructure to a private cloud in the form of an Infrastructure as a Service ([IaaS](#)). Thus enable the possibility to uncouple the application of the computing resources and at the same time uses the internal resources of the enterprise. Any application designed to run on top of a cloud should be able to virtually run everywhere a cloud is available. If such application is split into components, they can potentially run on multiple clouds in parallel. The main advantage of such approach is that the service can scale-up scale-down based on the real-time conditions and business rules (actual pricing, load of the private servers, geographical location of end-users, ...).

This project, developed in collaboration with the [European Broadcasting Union \(EBU/UER\)](#) will consist of an [Open-Source demonstrator](#) in the form of :

1. A minimum setup of 4 machines running [OpenStack](#) to provide a private [IaaS](#) to the [application](#)
2. A scalable [application](#) able to run on top of the private cloud and able to scale-up to the public clouds ([Amazon AWS](#), [HP Cloud](#), [Rackspace](#))
3. A set of nice to have [future extensions](#), one of them is adding [MPEG-DASH](#) encoding capabilities to the platform

[MPEG-DASH](#) ('Dynamic Adaptive Streaming over HTTP') is the upcoming standard for online video deliverance to multiple devices. In this upcoming standard different profiles are described and it seems that the live profile has the biggest potential to be picked up in the market. The goal of the corresponding extension is the addition of [MPEG-DASH](#) encoding capabilities to the [demonstrator](#) in order to popularize [MPEG-DASH](#).

Specifications

See Also:

Please see [Project Specifications](#) for further details.

- Project specifications refinement
- Setup the private cloud with [OpenStack](#)
- Design & implement the [Open-Source demonstrator](#)

Prerequisites

- Strong knowledges of [GNU/Linux](#) & [Ubuntu](#) system administration
- Excellent software architecture skills are required
- Good knowledges of the multimedia transmission standards & codecs
- Understand the constraints of the real-time streaming over IP

Keywords

TIC IPTV; TIC Multimedia; TIC Video Streaming; TIC Digital TV

DEFINITIONS

Multimedia

Adaptive streaming over HTTP HTTP-based adaptive bit-rate streaming technologies. Such technologies are specifically designed in order to provide to client a way to handle network conditions variations ¹ by continuously selecting an optimized bit-rate representation of the multimedia content. The delivery server must provides multimedia content encoded on multiple representations with specific bitrate and resolution.

Broadband streaming Multimedia content delivery through a broadband network, most (if not all) of the streaming technologies are based on the IP stack of the Internet protocols.

Broadcast streaming Multimedia content delivery through a broadcast network, such networks are designed to provide an unidirectional way to transmit productions from an unique source to the mass. Classically, multimedia content is encapsulated in MPEG-2 TS packets delivered by Digital Video Broadcasting systems.

Linear multimedia content Multimedia content intended to be viewed in real-time, the audience will consume this type of content linearly from the beginning to the end. Typical applications : TV channels, books.

Non-linear multimedia content Multimedia content that can be consumed on a non-linear way. The client can seek freely on such type of content. Typical applications : Video on demand, video games, ...

Media transcoding Transcoding is the direct digital-to-digital data conversion of one encoding to another.

Representation A representation is a specific content encoded on a specific parameter set (quality, geometry, bitrate, ...).

Operational & Cloud

DevOps DevOps is a new term describing what has also been called “agile system administration” or “agile operations” joined together with the values of agile collaboration between development and operations staff.

Hypervisor In computing, a hypervisor or virtual machine manager (VMM) is a piece of computer software, firmware or hardware that creates and runs virtual machines.

Elasticity ” Elasticity applied to computing can be thought as the amount of strain an application or infrastructure can withstand while either expanding or contracting to meet the demands place on it. ” ²

Web Services Web services are typically application programming interfaces (API) or Web APIs that are accessed via Hypertext Transfer Protocol (HTTP).

¹ Others metrics can be user at client side : Screen resolution, computational power ...

² Defining Elastic Computing - <http://www.elasticvapor.com/2009/09/defining-elastic-computing.html>

JuJu from Canonical Ltd.

Bootstrap To bootstrap an environment means initializing it so that Services may be deployed on it.

Endpoint The combination of a service name and a relation name.

Environment An Environment is a configured location where Services can be deployed onto.

Charm A Charm provides the definition of the service, including its metadata, dependencies to other services, packages necessary, as well as the logic for management of the application.

Repository A location where multiple charms are stored. Repositories may be as simple as a directory structure on a local disk, or as complex as a rich smart server supporting remote searching and so on.

Relation Relations are the way in which juju enables Services to communicate to each other, and the way in which the topology of Services is assembled. The Charm defines which Relations a given Service may establish, and what kind of interface these Relations require.

Service juju operates in terms of services. A service is any application (or set of applications) that is integrated into the framework as an individual component which should generally be joined with other components to perform a more complex goal.

Service Unit A running instance of a given juju Service. Simple Services may be deployed with a single Service Unit, but it is possible for an individual Service to have multiple Service Units running in independent machines. All Service Units for a given Service will share the same Charm, the same relations, and the same user-provided configuration.

Service Configuration There are many different settings in a juju deployment, but the term Service Configuration refers to the settings which a user can define to customize the behavior of a Service. The behavior of a Service when its Service Configuration changes is entirely defined by its Charm.

Provisioning Agent Software responsible for automatically allocating and terminating machines in an Environment, as necessary for the requested configuration.

Machine Agent Software which runs inside each machine that is part of an Environment, and is able to handle the needs of deploying and managing Service Units in this machine.

Service Unit Agent Software which manages all the life-cycle of a single Service Unit.

INTRODUCTION

1.1 Context

The Internet is growing in importance for broadcasters as delivery system for video and video related services to their audience. It allows broadcasters to deliver content directly to end-users and interact with them via interfaces. A downside to this story is the fact that distributing content over the Internet is very expensive for broadcasters.

One of the challenges they face is the transcoding of their on demand content libraries to new file formats that are optimized for multi-screen consumption. Fragmentation of media devices in technical capability (which codec settings they can display) or screen size (aspect ratio and pixel size of the video). Normally they need to encode to at least 8 file representations of the same video. This process is a constant factor when the daily production of content is contributed but also knows peaks in cases when libraries need to be transcoded.

Another challenge is the scaling of distribution servers. This process is dominated by changing demands because traffic peaks during the day, is at a minimum at night and sometimes when a video becomes a hit it will peak also.

Both these challenges are met by an encoding (or transcoding) and distribution environment that can up- or down-scale capacity easily. Ideally a distribution environment is downscaled at night and only upscaled at peak events. A transcoding environment needs to be upscaled when more transcoding jobs are waiting.

The OSCIED (Open-Source Cloud Infrastructure for Encoding and Distribution) project that is described in this paper addresses exactly these use cases.

I build a cloud-aware platform that can up-or down-scale transcoding or distribution nodes in a private (local servers) or in a public cloud (like [Amazon AWS](#)). Made possible because platform's functionalities are split into components and therefore can be deployed on multiple clouds in parallel and even more !

This environment will allow all kinds of other interesting functionalities for content providers as broadcasters. The encoding/transcoding can easily make new codecs available, you can optimize costs by up scaling in the night when the cloud computing resources are cheaper, encoding of live-content can be added etc. etc. From the distribution side the media gateways of different cloud providers can be used to cache content closer to the end user and with that optimize the data flows, or add different types of streaming, or define edges in different [CDN](#)'s and use OSCIED as a [CDN](#) overlay. The system as a whole can grow into a full fledged publication platform with professional management layers (which I started with already) that can publish, revoke but perhaps in the future also use cloud computing resources to deliver personalized transformations of the content deep into the network close to the end user.

[Open-Source](#) software has been chosen because anyone have access to the level of the source code and it allows other developers to build upon the work done. From the broadcast community and beyond already interest is shown to invest in my approach. Furthermore if the main components of the virtualized services for transcoding ([FFmpeg](#)) of distribution ([Apache 2](#)) have a new LibDASH or version that is made available by their specific [Open-Source](#) development community it can be integrated easily in the system.

1.2 Use Cases and Needs

Here will be explained the use cases this project focused-on such as the transcoding and online delivery of medias.

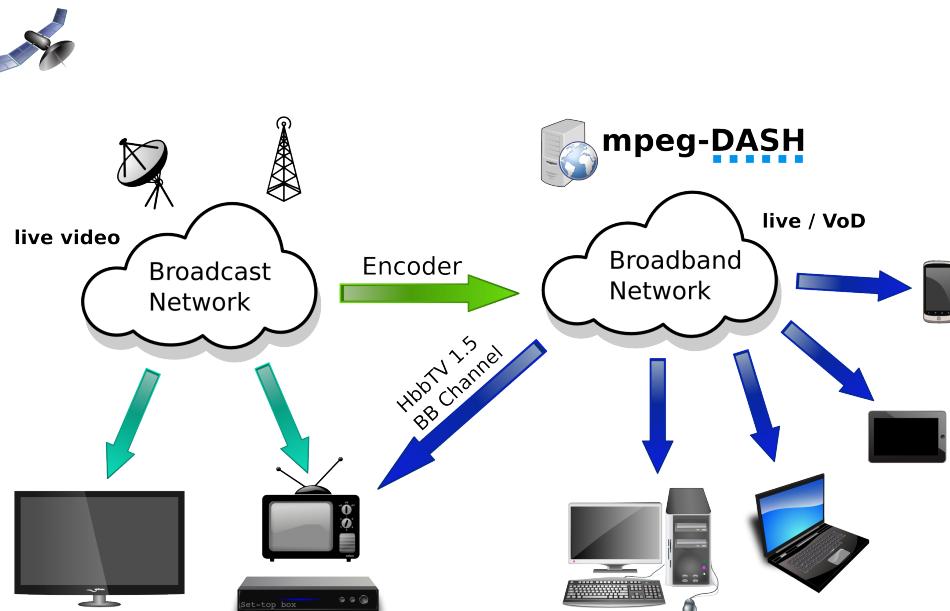


Figure 1.1: Multimedia content delivery through broadcast and broadband networks

1.2.1 Transcoding

" ... Transcoding is the direct digital-to-digital data conversion of one encoding to another, such as for movie data files or audio files. This is usually done in cases where a target device (or workflow) does not support the format or has limited storage capacity that mandates a reduced file size, or to convert incompatible or obsolete data to a better-supported or modern format. ..." source Wikipedia ([Transcoding](#))

A problematic use case of transcoding is when you need to transform a large collection of medias from a format to another. As this process is heavily demanding in computing resources and this requirement will increases with the every increasing quality of the content produced by cameras.

Transcoding is a never ending process as the media formats and AV codecs are evolving rapidly. Luckily the storage needs are partially balanced by the increasing compression efficiency of newer codecs.¹ For example, first version of [High Efficiency Video Coding](#) has just been released this January 2013.²

Another typical use case is the encoding (aka. digitalization) of archival materials to file based formats. This application needs specific equipments that are out of the preliminary demonstrator's specifications.

Note: At [European Broadcasting Union \(EBU/UER\)](#) it is actually a dedicated digitalization project called Transition to File where broadcasters are helped to migrate from tape based to file based environments.

1.2.2 Online Delivery

" ... Digital distribution ... describes the delivery of media content such as audio, video, software and video games, without the use of physical media usually over online delivery mediums, such as the Internet. ... With the advancement of network bandwidth capabilities, digital distribution became

¹ This sentence is partially true as the older versions may need to be keep in storage.

² Article H.265 standard finalized - <http://www.extremetech.com/extreme/147000-h-265-standard-finalized-could-finally-replace-mpeg-2-and-usher-in-uhdtv>

prominent in the 2000s. Content distributed online may be streamed or downloaded. Streaming involves downloading and using content “on-demand” as it is needed. ... Specialist networks known as content delivery networks help distribute digital content over the Internet by ensuring both high availability and high performance. ... ” source Wikipedia ([Distribution](#))

A problematic use case of online delivery is the publication of media related to high-audience events such as online TV journal with online *breaking* news. The other problematic use case is the every-growing demand of **VoD** content, as the computing resources can't be as easy scheduled as in any classical linear program.

Next generation connected televisions pushes the usage of broadband content delivery and new services related to that are emerging. Specifically the added possibility for HbbTV users to consume new programs using their Internet connection in parallel to the main broadcast.

TV to Internet : During latest Olympic Games the broadcasted *main* program showed the competitions split in small parts of few minutes. The audience could choose to continue watching one specific discipline by connecting the the Official Website and clicking on the media to play it.

HbbTV BC to BB : When a program (eg. Tennis) is longer than expected, an advertisement is overlayed to explain that the broadcasting of this program will be stopped and next scheduled TV program will be launched soon. The user can choose to continue the *standard* program or he can continue watching the end of the competition (e.g. Tennis) by clicking on the red button. This later choice means that the TV will display content delivered through the Internet.

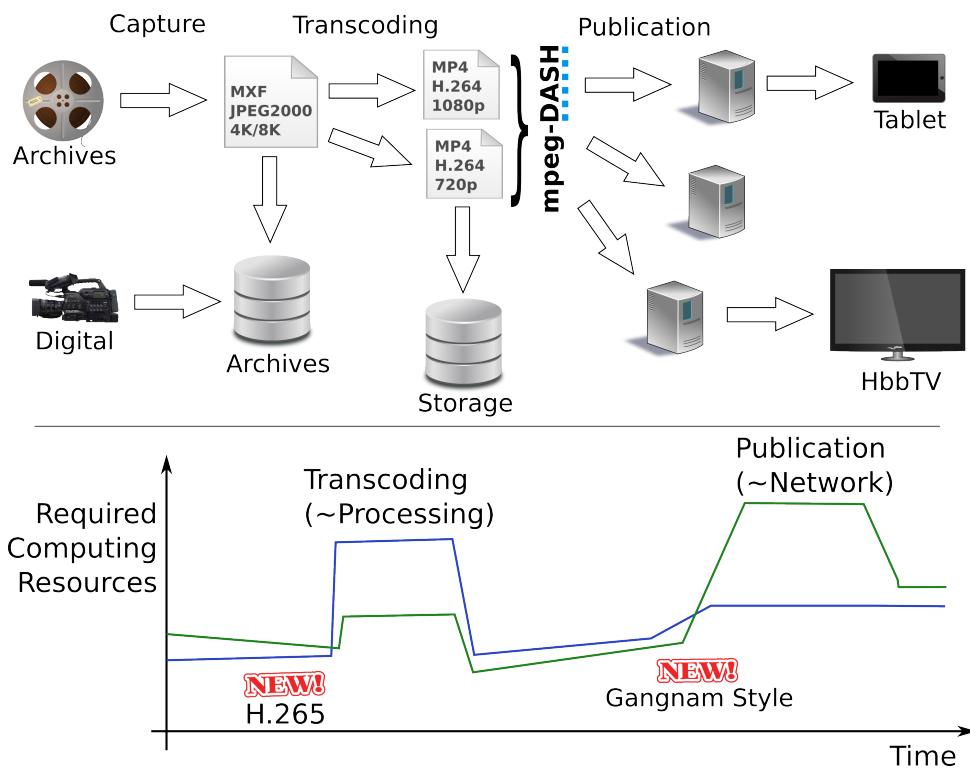


Figure 1.2: Simplified overview of the workflow the project focused-on.

1.2.3 Needs

These use cases are large libraries that needs to be encoded and transcoded all the time (due to new file formats and services). Therefore an adaptive elastic environment is needed that can migrate files to new file formats at certain moments. Virtualization makes it possible to add computing resources when needed and change codecs when needed.

For example when broadcasters want to migrate their library to **MPEG-DASH** they normally will buy new encoders that support the new file format. However it is much more efficient to rent virtual servers in the cloud during night blocks for the period of the job.

For distribution a flexible environment is great to upscale [Apache 2](#) servers when a lot of traffic is expected instead of having to support a full park that is scaled for a peak event.

... Typically, the computing scalability issues can be solved :

1. By upgrading enterprise's IT infrastructure (the servers room) :
 - Adding new servers to racks ;
 - Replacing servers with more powerful models ;
2. By using appropriate third party services :
 - Content delivery networks ([CDN](#)) e.g. [Akamai](#) ;
 - Cloud video transcoding platform ([SaaS](#)) e.g. [Zencoder](#) ;
3. By using cloud provider's resources as a service ([IaaS](#)) :
 - Consuming computing resources, e.g. [Amazon Elastic Clod Compute \(EC2\)](#) ;
 - Consuming delivery resources, e.g. [Amazon Cloud Front](#) ;
 - Consuming storage resources, e.g. [Amazon Simple Storage Service \(S3\)](#) ;

We can separate any varying workload in two main parts, the *constant* and the *varying* parts.

The *constant* part is related to the solution 1. The most common approach used in any enterprise requiring computing resources. This approach is interesting if the enterprise's workload is quite stable.

The *varying* part is related to any of the remaining solutions, 2 or 3. One typical approach adopted by Broadcasters is to uses services offered by [CDN](#) to improve online delivery capacity. The content will be cached by [CDN](#) servers closer to the end-user, decreasing networks load and increasing audience [QoE](#).

The third approach is quite promising but not as used as the others (at least by Broadcasters).

This is more difficult to consume these type of services as they are designed to be as generic as possible ³. The resources are provided through programmatic APIs (Web Services) and the computing resources needs to be *configured* before any *real* usage.

1.3 Proposed Solution

1.3.1 Foreword

The 20-21 November 2012 I attended to a workshop organized by the European Broadcasting Union (EBU/UER) called [EBU Cloud Workshop 2012](#).

From this two days interactive workshop we gather the needs of the broadcasters regarding the cloud. To sum-up what we learned, here is a modified extract of the *Broadcasters Wishlist* for the cloud(s), thanks to Félix Poulin from [EBU](#).

1. Easy to bring back & leave (right to be forgotten)
2. High availability (sometimes 100%)
3. QoS guarantees (depending on the needs)
4. Cost effective (as cheap as in-house)
5. Measurable performances (SLAs)
6. Fast uploads & downloads
7. Customizable & extensible
8. Access on any device
9. Media file aware

³ I speak about [IaaS](#) where computing resources are provided in the form of virtual machines instances.

10. Open Standards

Here are the wishes, regrouped by topic :

1. 1-5 are related to the usage of public-cloud only multimedia platforms
2. 6 is related to the performance of the networks involved for transfer of files
3. 7-10 are related to the multimedia platform itself (the software)

Proposed solution can potentially solve :

- Topic 1 by using private and public clouds at the same time (called hybrid cloud)
- Topic 2 by using in-house private-cloud and scaling to public-cloud when necessary
- Topic 3 by developing an application based on the needs of the broadcasters

1.3.2 The Idea

”Open-Source Cloud Infrastructure for Encoding and Distribution où le Cloud Maîtrisé ! “

The following project, based on cloud-era [Open-Source](#) technologies, can potentially fix this scalability issue by providing a rather simple but yet powerful way to consume already existing enterprise's IT resources mixed with necessary amount of public cloud resources !

In proposed solution, a set of the enterprise servers are dedicated to run the [Open-Source IaaS](#) called [OpenStack](#). The proposed [Open-Source](#) application is then deployed on this setup running the enterprise's private cloud. The application can be scaled-up to any compatible cloud provider in case of workload increasing⁴. Of course the scale-down of the application can be achieved as easily as the scale-up of it !

The main advantage of this solution :

- Is (itself) and is based on [Open-Source](#) technologies, no vendor lock-in, community driven developments ;
- Is fast and easy to scale, necessary setup/configuration is handled by the application⁵ ;
- Is divided in scalable and (potentially) interchangeable components⁶ ;
- Is not only compatible with cloud layers, components can run on standalone hardware⁷ ;
- The private cloud can run enterprise's services in parallel to proposed application ;
- Future extensions are already imagined, it is just matter of time and users requirements⁸ ;

1.4 Structure of the Report

The report is organized as follow :

- **State of the Art** this chapter gives a short overview of :
 - Cloud-based solutions from dedicated multimedia platforms (SaaS) to more general cloud-based services (IaaS).
 - Most interesting [Open-Source](#) technologies that makes the building of proposed solution a reality.
- **Project Specifications** this chapter details the specifications of the project. This chapter begins by exposing the goal of the project and the work to do during the thesis. It also describes the organization of the project in high-level, long-term developments cycles. Then more details are given about the first cycle, this thesis, in the form of development phases describing the high-level tasks to complete during the cycle. Finally, this chapter ends with details about the planning and the management tools used during this thesis.

⁴ This scenario describes one possible workflow, please see [Deployment Scenarios](#).

⁵ Thanks to Canonical's JuJu, please see [Clouds Orchestration Layer](#) for further details.

⁶ Thanks to application's design, please see [Application Layer](#) for further details.

⁷ Thanks to application's charms setup hooks and dev scripted tricks.

⁸ Please see [Future Extensions](#) for an overview of future extensions.

- **OSCIED Demonstrator** this chapter introduce you with the demonstrator in the form of logical layers, from physical servers to developed application. Each of those layers is described in dedicated sections of the chapter. The section about the application shows and details the main components of developed application. A FAQ about the demonstrator is also available in this chapter. Finally, the chapter ends with few example deployment scenarios (including example configuration files) and the tests and results section.
- **Conclusion** this chapter summarize features of OSCIED that are already available for broadcasters and what would be the most interesting features to integrate to developed platform.

STATE OF THE ART

2.1 Contents

This chapter gives a short overview of :

- Cloud-based solutions from dedicated multimedia platforms (SaaS) to more general cloud-based services (IaaS).
- Most interesting Open-Source technologies that makes the building of proposed solution a reality.

2.2 Overview of Cloud Based Multimedia Platforms

2.2.1 Quantel QTube

Official Description

” QTube is game-changing software that enables content creators, administrators and managers to interact with their content wherever they are and wherever their content is located. QTube is already in daily use transforming content creation in the same way, globalizing workflows for media organizations of all sizes around the world. ” ¹

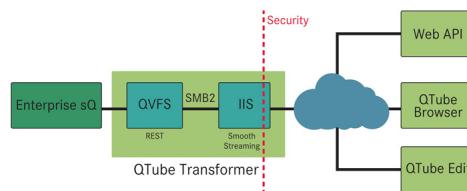


Figure 2.1: QTube workflow over IP, Copyright : Quantel

2.2.2 Zencoder

Official Description

” Zencoder is cloud-based video and audio encoding software as a service (SaaS). We have a wide range of customers, from individuals to Fortune 500 enterprise corporations, who all need to automate the encoding process through our encoding API. Because we’re based in the cloud, it means you have access to unlimited video encoding power, without having to pay for, manage, and scale expensive hardware and/or software. If you have any more questions feel free to contact us. ” ²

¹ Quantel About QTube – <http://uk1.quantel.co.uk/page.php?u=fc919cb52f36247c19a4fdade742b8ce>

² Zencoder Home Page – <http://zencoder.com/en/>



Figure 2.2: Zencoder live transcoding (beta), Copyright : Zencoder Inc.

2.3 Overview of Compatible Cloud Providers

2.3.1 Amazon Web Services

This is the public cloud provider I used during this project.

Official Description

” Amazon Web Services offers a complete set of infrastructure and application services that enable you to run virtually everything in the cloud: from enterprise applications and big data projects to social games and mobile apps. One of the key benefits of cloud computing is the opportunity to replace up-front capital infrastructure expenses with low variable costs that scale with your business. ”³



Figure 2.3: Amazon Web Services Logo, Copyright : Amazon

2.3.2 HP Cloud

Official Description

” HP Cloud Services is committed to delivering leading edge public cloud infrastructure, platform services, and cloud solutions for developers, ISVs, partners, service providers, and enterprises. HP Cloud Compute and HP Cloud Object Storage are built on HP’s world class hardware and software, with key elements of HP Converged Infrastructure and a developer-friendly integration of OpenStack technology. HP’s use of OpenStack technology and participation in its Open-Source project means that you get innovative, Open-Source-based cloud technology. This also means HP will be at the forefront of public cloud development and advancement as the OpenStack project evolves to deliver massively scalable applications. ”⁴



Figure 2.4: HP Cloud Services Logo, Copyright : HP

2.3.3 Rackspace Cloud

Official Description

” When you sign up for the Rackspace Cloud, you get access to all the tools you need to make your website or application a reality. Plus, enjoy convenient monthly pricing, and only pay for what you use

³ Amazon Web Services Home Page – <http://aws.amazon.com/>

⁴ HP Cloud Home Page – <https://www.hpcloud.com/>

(plus a monthly fee for managed cloud accounts). For example, Cloud Servers provides compute for your sites and apps. You get the persistence of a traditional server, plus the on-demand flexibility of the cloud. Because it delivers the computing power for your cloud configuration, your Cloud Servers are the heart of your Rackspace Cloud environment. Connect them to your data stored on Cloud Files or Cloud Block Storage, your Cloud Databases, and more. Use them with Cloud Load Balancers to deliver high availability. Plus, all Cloud Servers customers get free access to Cloud DNS, for easy management of your DNS records.”⁵



Figure 2.5: Rackspace Cloud Logo, Copyright : Rackspace

2.4 Overview of OSS Hypervisor Technologies

2.4.1 Linux KVM

This is the hypervisor I have chosen for running virtual machines on [OpenStack](#).

Official Description

” KVM (for Kernel-based Virtual Machine) is a full virtualization solution for Linux on x86 hardware containing virtualization extensions (Intel VT or AMD-V). It consists of a loadable kernel module, kvm.ko, that provides the core virtualization infrastructure and a processor specific module, kvm-intel.ko or kvm-amd.ko. KVM also requires a modified QEMU although work is underway to get the required changes upstream.

Using KVM, one can run multiple virtual machines running unmodified Linux or Windows images. Each virtual machine has private virtualized hardware: a network card, disk, graphics adapter, etc.

The kernel component of KVM is included in mainline Linux, as of 2.6.20.

KVM is [Open-Source](#) software.”⁶



Figure 2.6: KVM Logo, Copyright : Open Virtualization Alliance

2.4.2 LXC

This is the OS-level virtualization technology used by [JuJu](#) for running charms locally.

Official Description

” LXC is the userspace control package for Linux Containers, a lightweight virtual system mechanism sometimes described as “chroot on steroids”.

LXC builds up from chroot to implement complete virtual systems, adding resource management and isolation mechanisms to Linux’s existing process management infrastructure.

Linux Containers (lxc) implement:

⁵ Rackspace Cloud Products Page – <http://www.rackspace.com/cloud/products/>

⁶ KVM Main Page – http://www.linux-kvm.org/page/Main_Page

Resource management via “process control groups” (implemented via the cgroup filesystem) Resource isolation via new flags to the clone(2) system call (capable of creating several types of new namespaces for things like PIDs and network routing) Several additional isolation mechanisms (such as the “-o newinstance” flag to the devpts filesystem).

The LXC package combines these Linux kernel mechanisms to provide a userspace container object, a lightweight virtual system with full resource isolation and resource control for an application or a system.

Linux Containers take a completely different approach than system virtualization technologies such as KVM and Xen, which started by booting separate virtual systems on emulated hardware and then attempted to lower their overhead via paravirtualization and related mechanisms. Instead of retrofitting efficiency onto full isolation, LXC started out with an efficient mechanism (existing Linux process management) and added isolation, resulting in a system virtualization mechanism as scalable and portable as chroot, capable of simultaneously supporting thousands of emulated systems on a single server while also providing lightweight virtualization options to routers and smart phones.

The first objective of this project is to make the life easier for the kernel developers involved in the containers project and especially to continue working on the Checkpoint/Restart new features. The lxc is small enough to easily manage a container with simple command lines and complete enough to be used for other purposes.”⁷

2.4.3 OpenVZ

This is the virtualization technology I chose years ago for running servers of telecommunications laboratory at heapia.

Official Description

” OpenVZ is container-based virtualization for Linux. OpenVZ creates multiple secure, isolated Linux containers (otherwise known as VEs or VPSs) on a single physical server enabling better server utilization and ensuring that applications do not conflict. Each container performs and executes exactly like a stand-alone server; a container can be rebooted independently and have root access, users, IP addresses, memory, processes, files, applications, system libraries and configuration files. For more information about the technology and how it differs from the others like Xen, VMware etc.

OpenVZ software consists of an optional custom Linux kernel and command-line tools (mainly vzctl). Our kernel developers work hard to merge containers functionality into the Linux kernel, making OpenVZ team the biggest contributor to Linux Containers (LXC) kernel, with features such as PID and network namespaces, memory controller, checkpoint-restore etc. While OpenVZ can be used with recent upstream kernel, we recommend using OpenVZ kernel for security, stability and features.

OpenVZ is free Open-Source software, available under GNU GPL.”⁸



Figure 2.7: OpenVZ Logo, Copyright : Parallels

2.5 Overview of OSS Private Cluster/Cloud IaaS

2.5.1 Proxmox VE

This is the platform I choose years ago for running servers of telecommunications laboratory at heapia.

⁷ LXC Main Page – <http://lxc.sourceforge.net/>

⁸ OpenVZ Main Page – http://openvz.org/Main_Page

Official Description

“Proxmox VE is a complete virtualization management solution for servers. You can virtualize even the most demanding application workloads running on Linux and Windows Servers. It is based on the leading Kernel-based Virtual Machine (KVM) hypervisor and OpenVZ, the number one solution for container based virtualization. The best alternative to organizations looking for better total cost of ownership (TCO) and no vendor lock-in.”⁹

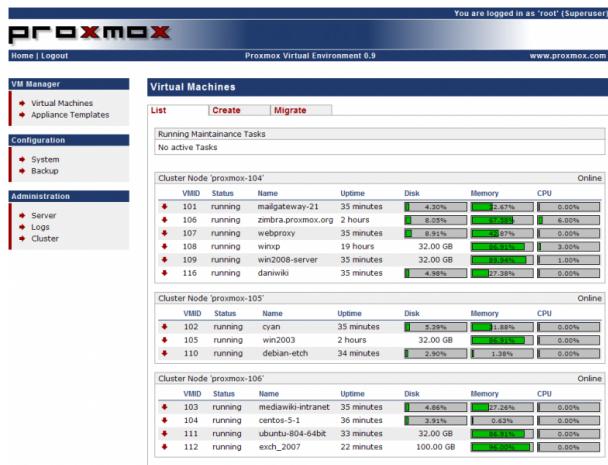


Figure 2.8: Integrated console view to the Virtual Machines, Copyright : YahyaNursalim

2.5.2 CloudStack

Official Description

“Apache CloudStack (Incubating) is Open-Source software designed to deploy and manage large networks of virtual machines, as a highly available, highly scalable Infrastructure as a Service (IaaS) cloud computing platform. CloudStack is used by a number of service providers to offer public cloud services, and by many companies to provide an on-premises (private) cloud offering, or as part of a hybrid cloud solution.

CloudStack is a turnkey solution that includes the entire “stack” of features most organizations want with an IaaS cloud: compute orchestration, Network-as-a-Service, user and account management, a full and open native API, resource accounting, and a first-class User Interface (UI).

CloudStack currently supports the most popular hypervisors: VMware, KVM, XenServer and Xen Cloud Platform (XCP).

Users can manage their cloud with an easy to use Web interface, command line tools, and/or a full-featured RESTful API. In addition, CloudStack provides an API that’s compatible with AWS EC2 and S3 for organizations that wish to deploy hybrid clouds.”¹⁰

⁹ ProxmoxVE Main Page – <http://www.proxmox.com/products/proxmox-ve>

¹⁰ CloudStack Main Page – <http://incubator.apache.org/cloudstack/>

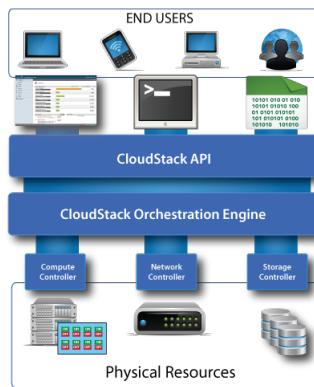


Figure 2.9: CloudStack conceptual infrastructure, Copyright : Apache Foundation

2.5.3 OpenStack

This is the [Infrastructure as a Service](#) I have chosen for this project.

Official Description

” OpenStack is an Infrastructure as a Service (IaaS) cloud computing project that is free [Open-Source](#) software released under the terms of the Apache License. The project is managed by the OpenStack Foundation, a non-profit corporate entity established in September 2012 to promote, protect and empower OpenStack software and its community.

More than 150 companies have joined the project among which are AMD, Intel, [Canonical](#), SUSE Linux, Red Hat, Cisco, Dell, HP, IBM, NEC, VMware and Yahoo!. It is portable software, but is mostly developed and used on the Linux operating system.

The technology consists of a series of interrelated projects that control large pools of processing, storage, and networking resources throughout a datacenter, all managed through a dashboard that gives administrators control while empowering their users to provision resources through a web interface.

OpenStack is committed to an open design and development process. The community operates around a six-month, time-based release cycle with frequent development milestones. During the planning phase of each release, the community gathers for the OpenStack Design Summit to facilitate live developer working sessions and assemble the roadmap. ” source Wikipedia ([OpenStack_IaaS](#))



Figure 2.10: OpenStack Logo, Copyright : OpenStack Foundation

2.6 Overview of OSS Cloud Orchestration Tools

2.6.1 HP CloudSystem

” Everyone has their own vision of how cloud computing will solve business problems. Why should you choose ours? HP delivers the most complete integrated system for enterprise and service providers to build and manage services across private, public and hybrid cloud environments.

- Unmatched automation & and orchestration
- The broadest support of applications, leading hypervisors, and operating systems
- Unified services management across cloud & and traditional IT
- Advanced application deployment, intelligent resource, & advanced configuration management
- Secure, scalable and extensible solutions built on proven and market leading Converged Infrastructure and Cloud Service Automation
- Integrated and automated application to infrastructure management for the cloud

HP CloudSystem is built on proven HP Cloud Service Automaton and Converged Infrastructure technologies. With support for the broadest set of applications, CloudSystem provides IT with a unified way to offer, provision and manage services across private clouds, public cloud providers, and traditional IT. It enables the flexibility to scale capacity within and outside your data center. And it's extensible to your existing IT infrastructure and can support heterogeneous environments.”¹¹

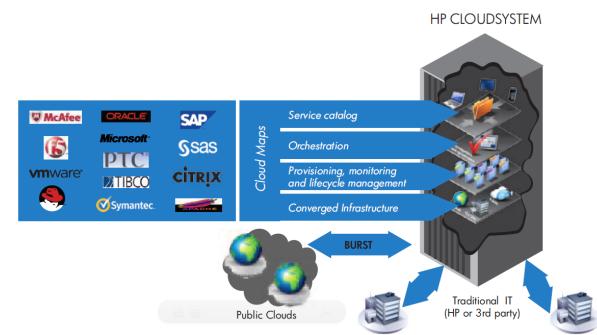


Figure 2.11: HP CloudSystem automation, Copyright : En Pointe

2.6.2 SlipStream

Official Description

” Automated provisioning and creation of cloud resources. SlipStream™ provides a simpler access to clouds, yet lets users do much more. For example automated, on-demand, creation of multi-machine runtime environments and version control the creation of custom machine images based on certified base images. SlipStream™ can also be used as your software engineering PaaS solution, as product or service. SlipStream will soon be released under an Open-Source license. This means that all SixSq cloud solutions will be available under a coherent Open-Source license.

Our customers use SlipStream™ to:

- Provision pre-certified production systems, as part of an overall vertical solution
- Cluster provisioning in the cloud - e.g. pre-configured clusters of user defined sizes
- Version control the creation of reference images, on which to base virtual machine deployments
- Software engineering Platform as a Service to speed-up project inception with provisioning of development tools (e.g. Jenkins/Hudson, Yum repository, Maven, Nexus)
- Single access to federated cloud, where users can switch between clouds yielding identical results on each

We are constantly amazed by new ways our customers come-up with using SlipStream™. If this is your case, please share them with us. SlipStream™ currently supports Amazon EC2 and StratusLab. We are actively adding several more, which will be announced soon. If you would like your cloud to be supported, please drop us a line.”¹²

¹¹ En Pointe HP CloudSystem – <http://www.enpointe.com/hp/cloud>

¹² SixSq. SlipStream Page – <http://sixsq.com/products/slipstream.html>

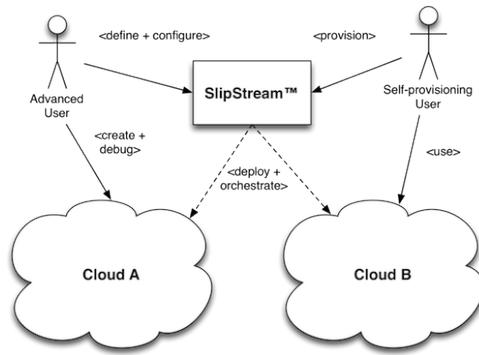


Figure 2.12: Shows how different user profiles can interact with SlipStream, Copyright : SiqSq.

2.6.3 JuJu

This is the cloud orchestration technology I have chosen for this project.

Official Description

” JuJu (formerly Ensemble) is a service orchestration management tool developed by Canonical Ltd.. It is an Open-Source project hosted on Launchpad released under the GNU Affero GPLv3. JuJu concentrates on the notion of service, abstracting the notion of machine or server, and defines relations between those services that are automatically updated when two linked services observe a notable modification. This allows for services to very easily be up and down scaled through the call of a single command. For example, a web service described as a JuJu charm that has an established relation with a load balancer can be scaled horizontally with a single `juju add-unit` call, without having to worry about re-configuring the load-balancer to declare the new instances: the charm’s event based relations will take care of that. JuJu’s charms can be written in any executable language.

What is JuJu ?

- Is DevOps Distilled. Through the use of charms, JuJu provides you with shareable, re-usable, and repeatable expressions of DevOps best practices. You can use them unmodified, or easily change and connect them to fit your needs. Deploying a charm is similar to installing a package on Ubuntu: ask for it and it’s there, remove it and it’s completely gone. With over 100 services ready to deploy, JuJu enables you to build entire environments in the cloud with only a few commands on public clouds like Amazon AWS, HP Cloud and Rackspace, to private clouds built on OpenStack, or raw bare metal via MAAS.
- Is a community of DevOps expertise. Most of the applications you want will be available in JuJu thus provides direct and free access to a DevOps community-contributed collection of charms.
- Provides service orchestration. JuJu focuses on managing the service units you need to deliver a single solution, above simply configuring the machines or cloud instances needed to run them. Charms developed, tested, and deployed on your own hardware will operate the same in an EC2 API compatible cloud.
- Is intelligent. JuJu exposes re-usable service units and well-defined interfaces that allow you to quickly and organically adjust and scale solutions without repeating yourself.
- Is easy. There’s no need to learn a domain specific language (DSL) to use JuJu or create charms. You can be up and running with your own charm in minutes.

JuJu GUI Live Demo Available here : <http://uistage.jujucharms.com:8080/> ” source Wikipedia (JuJu_software) + ¹³

¹³ JuJu FAQ Page – <https://juju.ubuntu.com/docs/faq.html#why-is-juju-useful>

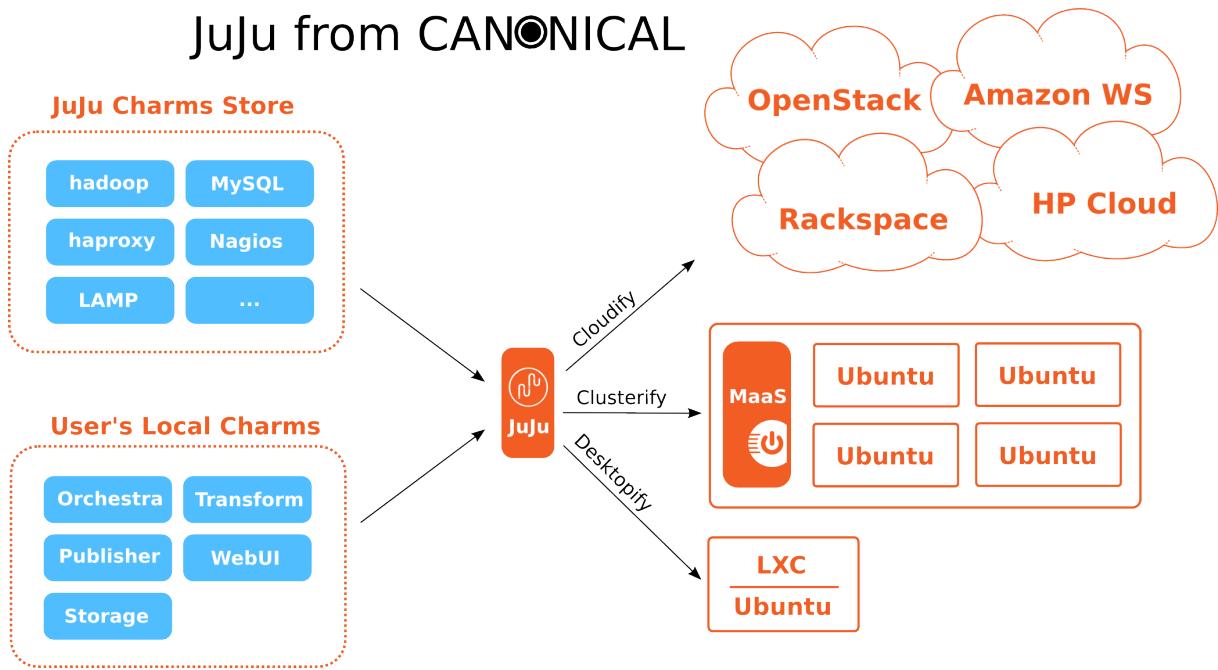


Figure 2.13: JuJu can deploy charms to a wide variety of environments, why not implementing your own provider ? Copyright : David Fischer (License : CC)

2.7 Overview of OSS Storage Technologies

See Also:

Please see [Ticket 122](#) for further interesting links.

2.7.1 Swift

Official Description

” The OpenStack Object Store project, known as [Swift](#), offers cloud storage software so that you can store and retrieve lots of data in virtual containers. It’s based on the Cloud Files offering from [Rackspace](#).

When you install [Swift](#), you can install multiple copies services that will track and retrieve the objects you want to store. Here’s a description of what you get with OpenStack Object Store:

- object server that stores objects (files less than 5 GB currently, support for large objects is in the works)
- a container server that keeps track of the objects
- a proxy server that handles all requests from the other server
- an authorization server so that your cloud storage is contained and authorized
- an account server that keeps track of all the containers
- Since [Rackspace](#) already has this system in production, we share our configuration but you can determine your own best performance and availability based on your hardware and networking capabilities. ”¹⁴

¹⁴ Swift Wiki Page – <http://wiki.openstack.org/Swift>

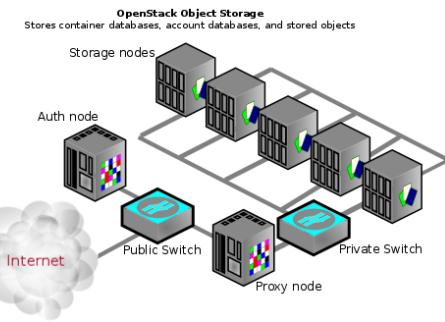


Figure 2.14: Example Swift installation architecture, Copyright : OpenStack Foundation

2.7.2 Ceph

Official Description

” The power of Ceph can transform your organization’s IT infrastructure and your ability to manage vast amounts of data. If your organization runs applications with different storage interface needs, Ceph is for you! Ceph’s foundation is the Reliable Autonomic Distributed Object Store (RADOS), which provides your applications with object, block, and file system storage in a single unified storage cluster—making Ceph flexible, highly reliable and easy for you to manage. Ceph’s RADOS provides you with extraordinary data storage scalability—thousands of client hosts or KVMs accessing petabytes to exabytes of data. Each one of your applications can use the object, block or file system interfaces to the same RADOS cluster simultaneously, which means your Ceph storage system serves as a flexible foundation for all of your data storage needs. You can use Ceph for free, and deploy it on economical commodity hardware. Ceph is a better way to store data. ”¹⁵



Figure 2.15: Ceph Logo, Copyright : Inktank Storage Inc.

2.7.3 GlusterFS

This is the storage technology I have chosen based on my *Decision Matrix*.

Official Description

” GlusterFS is an Open-Source, distributed file system capable of scaling to several petabytes (actually, 72 brontobytes!) and handling thousands of clients. GlusterFS clusters together storage building blocks over Infiniband RDMA or TCP/IP interconnect, aggregating disk and memory resources and managing data in a single global namespace. GlusterFS is based on a stackable user space design and can deliver exceptional performance for diverse workloads.

GlusterFS supports standard clients running standard applications over any standard IP network. Figure 1, above, illustrates how users can access application data and files in a Global namespace using a variety of standard protocols.

No longer are users locked into costly, monolithic, legacy storage platforms. GlusterFS gives users the ability to deploy scale-out, virtualized storage – scaling from terabytes to petabytes in a centrally managed and commoditized pool of storage.

¹⁵ Ceph Storage Home Page – <http://ceph.com/ceph-storage/>

Attributes of GlusterFS include:

- No limit on files sizes as compared to 5GB object size limit of OpenStack Swift¹⁶
- A unified view of data across NAS and Object Storage technologies
- Scalability and Performance
- High Availability
- Global Namespace
- Elastic Hash Algorithm
- Elastic Volume Manager
- Gluster Console Manager
- Standards-based
- Geo Replication ”¹⁷ ¹⁸



Figure 2.16: GlusterFS Logo, Copyright : Gluster

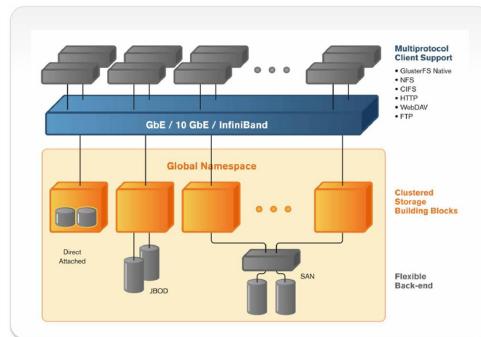


Figure 2.17: Gluster Open-Source Scalable NAS Implementation, Copyright : Gluster

2.7.4 LS4

Official Description

” LS4 is an Open-Source distributed storage system designed to store objects like photos, musics or movies.

- LS4 stores a set of objects identified by a key. Each object consists of data and attributes where data is a raw bytes and attributes are associative pairs. Objects are distributed to servers for scalability and copied on multiple servers for availability
- Each object can have multiple versions
- Replica set is a set of data servers that stores same objects
- Fail-back can be done without stopping the cluster

¹⁶ GlusterFS + OpenStack – <http://www.gluster.org/wp-content/uploads/2011/07/Gluster-Openstack-VM-storage-v1-shehjar.pdf>

¹⁷ GlusterFS About Page – <http://www.gluster.org/about/>

¹⁸ GlusterFS Admin Guide – http://www.gluster.org/wp-content/uploads/2012/05/Gluster_File_System-3.3.0-Administration_Guide-en-US.pdf

- With LS4 and nginx, contents can be transferred without passing through application servers while the application server proceeds HTTP requests. Thus you can reduce CPU load and network traffics. It's implemented using nginx's X-Accel-Redirect feature. See the HowTo to configure the bypass. Additionally, LVS's Direct Routing may be useful on the proxy
- You can configure LS4 to replicate data over remote datacenters while applications get data from the local datacenter ”¹⁹



Figure 2.18: LS4 Logo, Copyright : FURUHASHI Sadayuki

2.7.5 NFS

Official Description

” Network File System ([NFS](#)) is a distributed file system protocol originally developed by Sun Microsystems in 1984, allowing a user on a client computer to access files over a network in a manner similar to how local storage is accessed. [NFS](#), like many other protocols, builds on the Open Network Computing Remote Procedure Call (ONC RPC) system. The Network File System is an open standard defined in RFCs, allowing anyone to implement the protocol. ” Source Wikipedia ([NFS](#))

2.7.6 MongoDB

Official Description

” MongoDB (from “hu**mongo**us”) is a scalable, high-performance, Open-Source NoSQL database. Written in C++, MongoDB features:

- Document-Oriented Storage – [JSON](#)-style ([BSON](#)) documents with dynamic schemas offer simplicity and power
- Full Index Support – Index on any attribute, just like you’re used to
- Replication & High Availability – Mirror across LANs and WANs for scale and peace of mind
- Auto-Sharding – Scale horizontally without compromising functionality
- Querying – Rich, document-based queries
- Fast In-Place Updates – Atomic modifiers for contention-free performance
- Map/Reduce – Flexible aggregation and data processing
- GridFS – Store files of any size without complicating your stack
- Commercial Support – Enterprise class support, training, and consulting available ”²⁰



Figure 2.19: MongoDB Logo, Copyright : 10gen, Inc.

¹⁹ LS4 Home Page – <http://ls4.sourceforge.net/>

²⁰ MongoDB Home Page – <http://www.mongodb.org/>

2.7.7 Decision Matrix

This is a table comparing main features of the set of Open-Source storage or database technologies I introduced earlier.

As you can see, MongoDB is actually the only database technology I selected. This is mainly due to the fact that MongoDB handles json natively and it was also easy for me to integrate it to my Python application with pyMongo !

Using GlusterFS was another great choice, thanks to the excellent Gluster File System 3.3.0 Administration Guide.

Note:

- DB = Database, FS = Filesystem, BS = Block storage, OS = Object storage
 - Compl(exity) = Is the design complex ?
 - Scal(ability) = Is it scalable ?
 - Manag(ement) = Ease of management
 - Sec(urity) = Secured access ?
-

Database technologies

| Name | Description | DB Access | Complexity | Stability | Management | Scalability | Security | Charm |
|---------|----------------|----------------|------------|-----------|------------|-------------|----------|-------|
| MongoDB | NoSQL database | JSON documents | Low | Yes | Easy | High | Yes | Yes |

Storage technologies

| Name | Description | FS Access | BS Access | OS Access | Compl. | Regions | Stability | Manag. | Scal. | Sec. | Miscellaneous | Charm |
|-----------|---------------|-------------|-------------|-----------|--------|---------|-----------|--------|-------|------|---------------|-------|
| Swift | S3 storage | – | – | S3 API | High | (?) | (?) | (?) | High | (?) | Object < 5GB | ~Yes |
| Ceph | All in one | Linux mount | Linux mount | S3 API | Medium | (?) | (?) | (?) | High | (?) | – | ~Yes |
| LS4 | Media storage | – | – | REST API | High | Yes | Unit test | (?) | High | (?) | No updates | No |
| NFS | Filesystem | Linux mount | – | – | Low | No | Stable | Simple | Low | Yes | – | ~Yes |
| GlusterFS | Two in one | Linux mount | – | Yes | Low | Yes | Stable | Simple | High | Yes | – | ~Yes |

2.8 Conclusion

Choosing tools and technologies in order to build an [Open-Source](#) application is one of the most interesting and important preliminary step. In fact, most of the required functionnalities of OSCIED can be implemented by developing code using or gluing together [Open-Source](#) softwares or libraries. The design of OSCIED implies to select the technologies that provides required features (e.g. scalable storage, RESTful API) and at the same time fit the technical criterias (e.g. strong community, scalability, [KISS](#), ...).

This preliminary step was guided by the [*Project Specifications*](#), the chapter that follows.

PROJECT SPECIFICATIONS

3.1 Contents

This chapter details the specifications of the project. This chapter begins by exposing the goal of the project and the work to do during the thesis. It also describes the organization of the project in high-level, long-term developments cycles. Then more details are given about the first cycle, this thesis, in the form of development phases describing the high-level tasks to complete during the cycle. Finally, this chapter ends with details about the planning and the management tools used during this thesis.

EBU Technology & Innovation

Project EBU OSCIED part 1: The basic set up

Title Open-Source Cloud Infrastructure for Encoding and Distribution

Main Developer *David Fischer*

Project Leader Bram Tullemans

3.2 Motivation

This project is aimed by the goal of providing a scalable media [Open-Source](#) platform to members of the [European Broadcasting Union \(EBU/UER\)](#).

This platform, based on cloud-era [Open-Source](#) technologies, would be dedicated and designed based on broadcasters specific needs such as transcoding of a wide collection of medias or online publication of popular medias, two basic use cases of this preliminary demonstrator.

This [Open-Source](#) platform would be freely available for broadcasters to promote interchange of knowledge and drive the project's developments by a wider community of experts.

3.3 Work to be done

The focus will be to develop an [Open-Source](#) scalable cloud infrastructure for encoding and distribution of on demand video using H.264 video codec.

A professional management layer for the system as a whole will be at the core of this system.

Nice to haves are in order of importance the support of [MPEG-DASH](#), both for Live and [VoD](#) and play out to different devices (laptop, [HbbTV](#), Tablet and Smartphone both for Android and iOS).

3.4 Development cycles

The EBU OSCIED project consist out of different development cycles. Within the *first cycle* the basic setup of the project will be realized.

All the core functionalities will be ready and can be upgraded either by adding them in the available code and / or by adding or exchanging modules.

Furthermore OSCIED will consist out of three separate scalable environments for development, test and production allowing different developers to work on new cycles simultaneously.

3.5 Time span first cycle

The project is the Master Thesis of *David Fischer* who will spend 0.4 fte for 6 months on the project as a minimum. The first development cycle starts on the 25 September 2012 and ends on 25 March 2013. The Master Thesis itself starts on the 20 September 2012 and ends on 8 February 2013 (cf. [Planning](#)).

3.6 Development phases

1. **investigation** will consist of defining which Open-Source tools should be used in order to make this project successful. The main decision criteria are the size of the community behind the tool, the release of fixes & features, the quality of the documentation, the licensing, ...
2. **architecture** will consist of dividing the platform's architecture in specialized components (orchestrator, encoding and publishing units, ...) able to work together and to scale-up/down as easier as possible. During this key process, it is necessary to think about the media asset management and the future extensions
3. **servers** will consist of choosing appropriate server's configuration based on constraints such a price tag for a setup of 4 servers
4. **private cloud** will consist of installing Open-Source private cloud environment using [OpenStack](#). This long-run task should be as automated as possible and the output will consist of setup scripts in parallel to the setup itself
5. **environment** will consist of defining what kind of professional management is needed (performance servers or virtual machines, reboot, input selector, feed check, encoding settings, triggering of processes, metadata, xml-in feed from master control room, xml-output for MaM systems, scheduler, expand encoding or distribution parts to cloud) and develop basic interface
6. **application** will consist of implementing application-level components of the platform¹ :
 - **storage** : To store medias in a scalable way
 - **transform** : To transcode medias in a scalable way
 - **publisher** : To made medias available to end-users
 - **webui** : To provide a user interface for broadcasters
 - **orchestra** : To provide a RESTful API² and to orchestrate the other components

3.7 Set up

The development and distribution environment runs on the Open-Source cloud software [OpenStack](#). The project involves 4 servers consisting of a Controller and Nodes (Computer/Storage) machines. On this private cloud run virtual machines that enable separate controllable environments for development cycle :

¹ This listing wasn't complete at the beginning of the project as it was part of *phase 2*.

² This API will be used by MAM to integrate the platform in their (automated) workflow !

1. **Development** : Programmers access their own playground
2. **Test** : New code is tested in relation to the rest of the software
3. **Production** : Approved code is implemented in the live services

These different environments can be accessed from the outside allowing external approved developers, for example from **EBU** members, to work on new code, to test the code so it can be added to the production environment.

Up scaling of capacity can be done by adding servers in the private (local) cloud or by using the public cloud. For example within a separate DTP-environment one can define virtual machines allowing for example for adding encoding or distribution entities. Within the management interface one should be able to identify if hardware needs to be added in the private cloud or if one needs to define the extra virtual machines in a public cloud.

The total environment is flexible and redundant ³:

- Easily expandable by adding servers. Via the management layer one can add the server to the cluster ideally with a self install procedure
- Expandable via external Public Cloud Solutions (Parts can be up scaled via the public cloud, for example the encoding is scaled in the private cloud and the distribution is up scaled in the public cloud)
- Redundancy is accomplished by the fact that if one server is taken out, for example when it is broke, the rest will take over

3.8 Documentation

During the development process the **Open-Source** integrated ticket and project management tool **TRAC** (**OSCIED Project's TRAC Environment**) will be used to gather descriptions of the code and functionalities. The goal is to gather all necessary descriptions organically. This should reduce the effort to generate documentation after the project.

3.9 Organizational consequences

First of all we will have to decide which **Open-Source** legal infrastructure is going to be adopted (**MIT**, **Creative Commons**, **GNU GPLv3**, **GNU LGPLv3** or **LGPL Europe**). Together with the legal department we should also investigate responsibilities for the **EBU** when this software is distributed. Furthermore a organizational / legal order like a Swiss association would perhaps be appropriate to attract external developers and funding.

3.10 Future cycles

Nice to haves are in order of importance the support of **MPEG-DASH**, both for Live and **VoD** and play out to different devices (laptop, **HbbTV**, Tablet and Smartphone both for Android and iOS). If this is not realized in *cycle 1* it will be the core of cycle 2.

3.11 Future development

Here are some of the preliminary ideas ⁴ :

- **Open Broadcast Encoder** for delivering MPEG2TS feed
- Adding profiles for public clouds allowing to use different clouds at the same time and automatic scalability functions
- Adding native imports of distribution files to reduce latency

³ Some of the features will be implemented after the end of my Master Thesis (cf. *Planning*)

- ... A lot of features we will think of during the process (cf. [Future Extensions](#))

3.12 Planning

To be honest with you, I don't planned the whole project on the pit-start (as soon as the project started) !

In fact, I spend the first days to install my work place and to setup the [OSCIED Project's TRAC Environment](#).

We meet at least once a month with Mr. Bram Tullemans and Prof. Andrés Revuelta to manage this project. We discussed about the project to keep in sync what I have done and what I should do in order to be successful.

During my investigations I filled the [OSCIED Project's TRAC Environment](#) with links to the most interesting resources I founded on Internet. I also created tickets related to the development tasks not only to see what should be implemented but also to backup some of the future features I thought of. Of course, there were bugs and I noticed them as *defect* tickets and then I fixed them (as soon as possible).

All of these tickets are explicitly linked to a *component* of the application (e.g. Orchestra, Master Thesis Report, ...) and most of the time I set the *priority* field.

In order to reduce the entropy of this growing set of tickets I created *milestones* and grouped tickets.

This project started with my Master Thesis and hopefully it will not end with it ...

I planned my project as such :

- **25 September 2012 - 25 March 2012 :**

- Investigations helped with necessary resources such as Internet, books, ...
- Developments of automation scripts for easiness of tests, setup, ...
- Large amount of bugs fixed mainly of my own creation but not only ...
- Project's management with team members and with dedicated tools ...

- **25 September 2012 - 10 November 2012 :**

- Setup of development and management tools at (EBU, hepia, home)
- Project's specifications refinement with Mr. B. Tullemans and Prof. A. Revuelta
- Preliminary design decisions based on early investigations
- Bill of material of the servers (based on project's CAPEX)

- **25 September 2012 - 31 November 2012 :**

- Intensive scripting and readings of [OpenStack](#) documentation
- Initial [OpenStack](#) setup of the server based on automation scripts

- **1 December - 20 January 2012 :**

- Intensive development of the application, release of the first demonstrator
- Various deployments scenarios tested, application successfully deploy on [Amazon AWS](#)

- **21 January 2012 - 8 February 2012 :**

- Selection of best tools to increase speed and easiness of writing
- Cleanup and reordering of the project's tickets under [TRAC](#)
- Writing of the following report with required content ⁵

⁵ This report is actually not the only source of documentation for this project, see [OSCIED Project's TRAC Environment](#).

3.13 Management

Here will be introduced the tools used to manage code and tasks such as [Subversion](#) and [TRAC](#).

3.13.1 Source Code Versioning : SVN (and GIT)

What is SVN & GIT ?

[Subversion](#) is a versioning and revision control software based on a client/server architecture. Developers use this kind of tool to maintain current and historical versions (called revisions) of their work such as source code and documentation of their project(s).

[Subversion](#) is one of the most widely used versioning system on the [Open-Source](#) community.

The SVN server hosts the projects repositories and the developers will use a SVN client to get a local copy of the project. The developer will work locally on his copy (add, remove, rename, modify files and directories) and then propagate (commit) these modifications to the repository.

[GIT](#) is a versioning and revision control software based on a distributed architecture. [GIT](#) was initially designed and developed by Linus Torvalds for the development of the Linux kernel, the biggest [Open-Source](#) project ever made. Unlike [Subversion](#), every [GIT](#) local copy is a full-fledged repository with complete history. This design permit to maintain large distributed projects in a efficient manner.

Why using SVN & GIT ?

[MPEG-DASH](#) is a cutting-edge standard and the [Open-Source](#) community is actively implementing [MPEG-DASH](#) in their favorite [Open-Source](#) piece of software. This is why it is necessary to access to the latest revision of the source code of these softwares to understand what is implemented and what is not (profiles, ...) !

It is also necessary to manage backups of this project and [Subversion](#) is the best tool for that. Every backup (*commit*) is done manually and every single revision has a purpose, such a new functionality (code), a bug fix (code) or some files to backup (documents). The revision history permit revert changes if necessary and it also permit to get a statistical overview of the work (with [StatSVN](#) for example).

This tool is also a must have to synchronize the contributions (work units) of the team members !

We can also uses any [Version Control System](#) branching capabilities to manage release policy of the project. For example, we can create branches like :

- **trunk** related to latest features, here we can found the cutting-edge / development version of the software
- **testing** related to latest version to test before releasing it in production
- **production <version>** related to stable releases ⁶

And then *checkout* any of the following branches anywhere you want, the upgrade of running *local copy* will be simply done by calling the *update* method of the [Version Control System](#) !

Note: Most of the [Open-Source](#) repositories are hosted on [SourceForge](#) (SVN) or [GitHub](#) (GIT).

⁶ With a version number, I actually like the version numbering of [Ubuntu](#)'s releases.

3.13.2 Project Management : TRAC

What is TRAC ?

TRAC is a simple tickets tracking system aimed to provide a management and documentation layer to any project based on software development. This tool is accessed by users through a web user interface.

TRAC can be interfaced with some of the most used Version Control System and for this project, we thanks the Open-Source community to provide an interface for Subversion !

Here are the main features :

- **Wiki** to add collaborative documentation to the project
- **Timeline** to see what happens to the source-code or to TRAC itself
- **Roadmap** to manage Milestones (e.g. grouping of tickets into higher-level features with a delivery deadline)
- **Source Browser** to browse the source code of the project hosted by the VCS
- **Tickets** to filter tickets by clicking on specific Reports, e.g. *Active Tickets by Milestone ...*
- **Search** to search something into the Wiki, the Tickets, the Milestones, ...
- **Admin** to configure the tool, add/edit Users, Components, Milestones, ...

... And a lot of **plugins** you can add to TRAC !

Tickets in a nutshell

They are fields (in **bold** the fields I really take care about)⁷ :

- **Summary**
- Reporter
- **Description**
- Keywords
- Owner
- Cc
- **Type** (defect, enhancement, reference, task)
- **Priority** (none, trivial, minor, major, critical, blocker)
- **Component** (... , Cloud, Development, FIMS, MPEG-DASH, Master Thesis report, Orchestra, ...)
- **Milestone** (Demonstrator ready, Master Thesis Report, ...)
- **Status** (new, assigned, accepted, duplicate, fixed, invalid, wontfix, worksome, reopened)
- Version (1.0, 2.0)

They are three built-in type of tickets :

- **defect** typically used to report a bug or a missing feature
- **enhancement** typically used to describe interesting new features
- **task** typically used to specify what should be implemented and when

Why using TRAC ?

Mostly TRAC was used as a smart notepad (tasks, bookmarks, bugs) for my ideas and I actually have a lot of ideas for this project ;-)

For example, I created the ticket type called *reference* with related *None* priority to save (grouped by topic) bookmarks of the most interesting resources I founded on the Internet.

This tool is also useful to create *task* tickets reflecting any non-trivial unit of work. For example, add a feature (code), add a chapter to documentation (report), ...

⁷ Reference Type, Component, Milestone and Owner fields values are not built-in but created in the Admin tab

This ticket will be accepted by *someone* and when the work is done, the person who has done the work will update the ticket's *status* (e.g. close the ticket with status *fixed*).

3.14 Conclusion

This project started with a motivating, well defined set of uses-cases based on realistic challenges the broadcasters face to provide new *connected* services to their audience. The detailed specification of the project OSCIED is then defined to fit the uses-cases that motivated the development of OSCIED.

The set of cycles, phases and tasks permit to create a roadmap for the project, at a macro- (cycles) and micro-level (phases). First cycle specified the work to be done during this thesis.

Another key element is the decision to use specialized tools to manage the project ([TRAC](#)) and to use a Version Control System like [Subversion](#).

Chapter that follows will give more details about the builded demonstrator – a preliminary version of OSCIED.

OSCIED DEMONSTRATOR

4.1 Contents

This chapter introduce you with the demonstrator in the form of logical layers, from physical servers to developed application. Each of those layers is described in dedicated sections of the chapter. The section about the application shows and details the main components of developed application. A FAQ about the demonstrator is also available in this chapter. Finally, the chapter ends with few example deployment scenarios (including example configuration files) and the tests and results section.

4.2 Introduction to Layers

The following chapter will introduce you with the demonstrator in the form of logical layers.

Conceptually, you can picture the demonstrator as such :

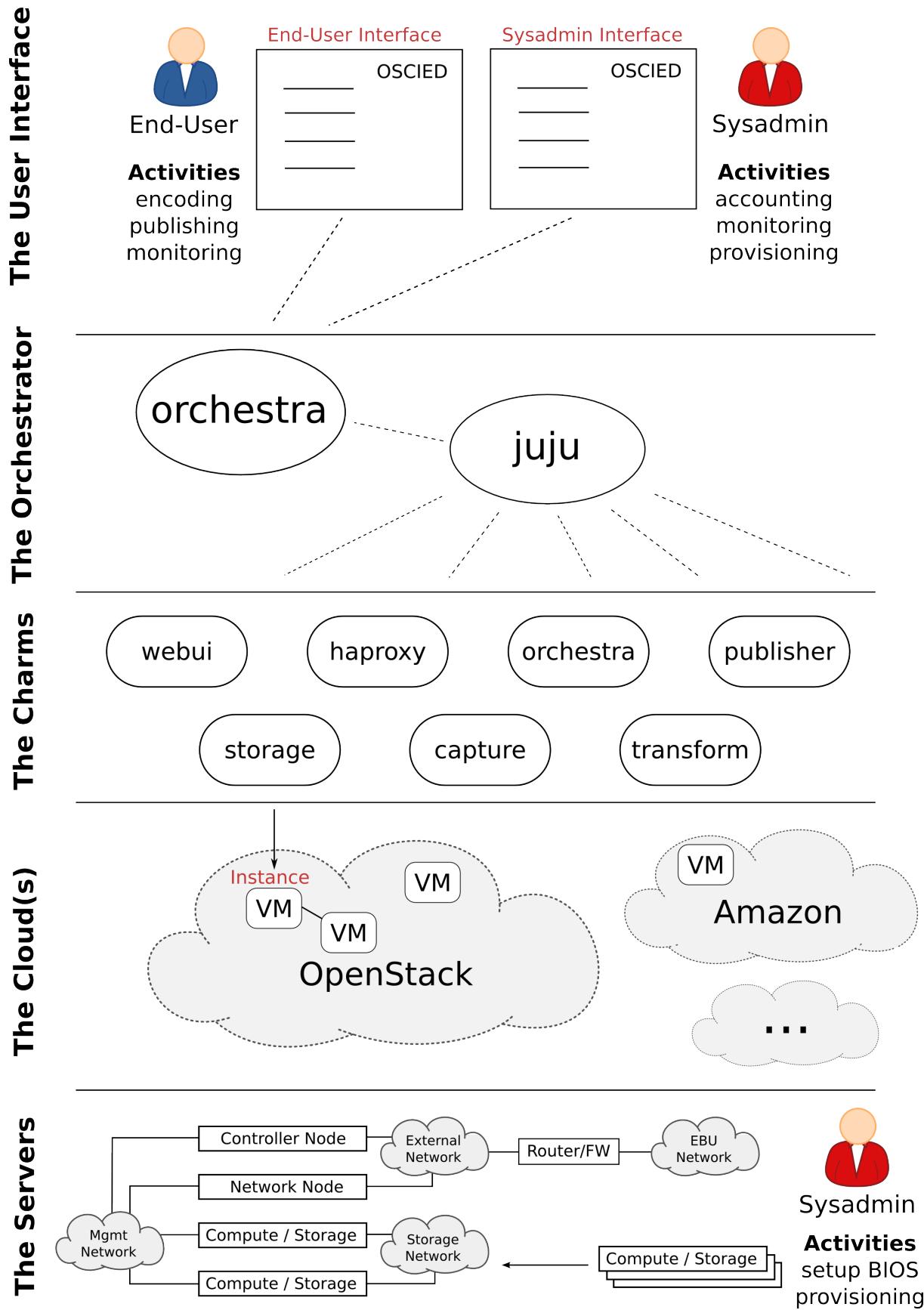


Figure 4.1: The project separated in logical layers, from the servers to the end-user

4.3 Server Layer

4.3.1 Specifications

In order to run the demonstrator's private cloud based on [OpenStack](#) we need servers.

A setup of four 1U servers were chosen and configured based on the following :

- The price-tag specified in the project's CAPEX
- The optimization of the configuration based on project's requirements
- The chosen OS [Ubuntu Quantal Server](#) -> [Ubuntu Server certified hardware \(Dell\)](#)

Here is the technical specifications of the servers.

” Get an energy-efficient, dense 1U server for your applications with the PowerEdge™ R420, featuring next-generation processing and flexible I/O options. ” source [Dell](#)

| Characteristic | Description |
|--------------------------------|--|
| Model | PowerEdge R420 |
| Processors | 2x Intel® Xeon® E5-2430 2.20GHz, 15M Cache, 7.2GT/s QPI, Turbo, 6C, 95W, Max Mem 1333MHz |
| Memory Configuration Type | Performance Optimized |
| Memory DIMM Type and Speed | 1333 MHz RDIMMs |
| Memory Capacity | 4GB RDIMM, 1333 MT/s, Low Volt, Dual Rank, x8 Data Width |
| Operating System | No Operating System |
| OS Media kits | No Operating System Media Kit |
| Chassis Configuration | 3.5 Chassis with up to 4 Cabled Hard Drives and Embedded SATA” |
| RAID Configuration | HW RAID 0 |
| RAID Controller | Dell PERC H310 Mini |
| Hard Drives | 2x 1TB 7.2K RPM SATA 3.5in Cabled Hard Drive |
| Power Supply | Single Cabled Power Supply 550W |
| Power Cords | NEMA 5-15P to C13 Wall Plug, 125 Volt, 15 AMP, 10 Feet (3m), Power Cord |
| Power Management BIOS Settings | Performance BIOS Setting |
| Embedded Systems Management | Basic Management |
| Add-in Network Adapter | On-Board Dual Gigabit Network Adapter |
| Rack Rails | No Rack Rails or Cable Management Arm |
| Bezel | No Bezel |
| Internal Optical Drive | No Internal Optical Drive for 4HD Chassis |
| System Documentation | Electronic System Documentation and OpenManage DVD Kit for R420 |
| PCIe Riser | PCIE Riser for Chassis with 2 Proc |
| Shipping | Shipping Material,PowerEdge R420 |
| Hardware Support Services | 3Yr Basic Hardware Warranty Repair: 5x10 (HW-Only, 5x10 NBD Onsite) |
| Installation Services | No Installation |
| Proactive Maintenance | Maintenance Declined |

4.4 Cloud Layer

4.4.1 OpenStack in a Nutshell

OpenStack is an Open-Source Infrastructure as a Service project initially launched by NASA () and Rackspace () in July 2010 and released under the Apache 2 license and now managed by the non-profit OpenStack Foundation. OpenStack is also the name of the initiative related to this project, aimed by the goal of providing to any organization a way to deploy a private cloud (IaaS) on top of their (commodity) IT infrastructure.

This cloud technology consist of a bunch of inter-related Open-Source components (each of them running a specific service) working together to abstract and control hardware computing resources such as processing, storage and networking.

The cloud infrastructure's users will consume computing resources in their abstracted form such as virtual machines, virtual networks, ...

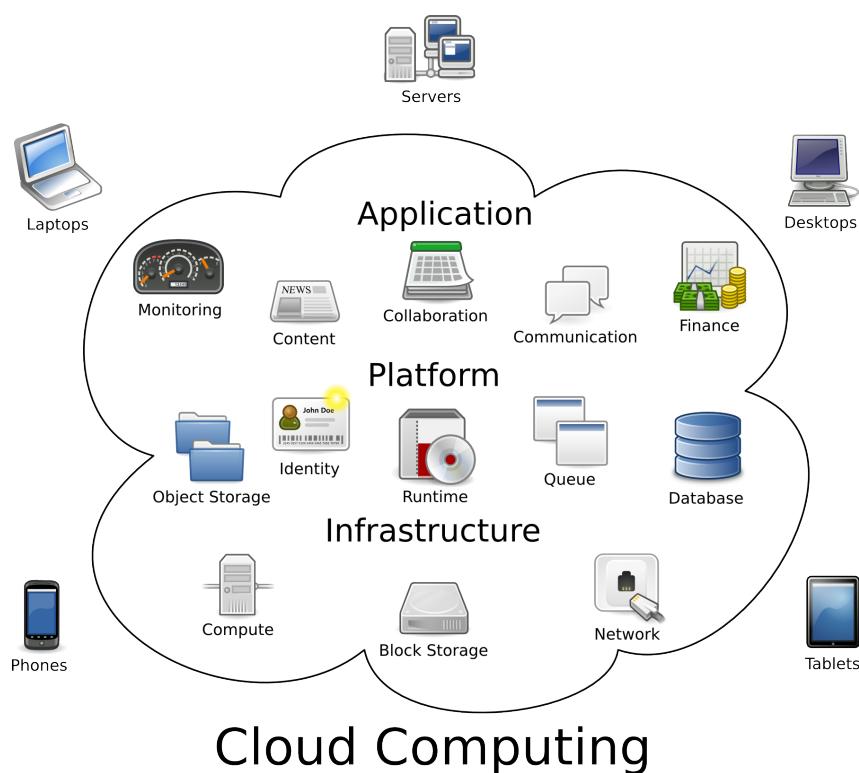


Figure 4.2: Source : Wikipedia - [Cloud computing](#) (Created by Sam Johnston)

Cloud Advantages

The main advantage of such approach is that the enterprise services are uncoupled from underlying hardware. The cloud infrastructure can improve the availability of services by enabling new way of managing IT resources.

One of them is called the **live-migration**, with it you can :

- Ensure the failover of services in case of hardware failure
- Replace computing resources without any service interruption
- Schedule the load of servers by specifying operational rules

Another advantage is that is easier to manage the usage of your IT infrastructure, here are some examples of what you can do :

- Easier scale up/down by adding/removing computing resources to the platform
- Specify platform's resources usage quotas, this is related to users
- Optimize hardware usage thus decrease the OPEX of your IT infrastructure

Nowadays, cloud-era tools (eg. [JuJu](#)) let you deploying and managing your services in any compatible cloud as easy as `juju deploy mysql` ! So why not the same for your own application on your own infrastructure ?

Then if you need more computing resources, you may decide to *seamlessly* use public cloud provider's resources complementary to your own cloud infrastructure.

And OpenStack

The strengths and weaknesses of the project called [OpenStack](#) are :

- **The modular architecture add flexibility to any deployment :**
 - (+) The components can be deployed in standalone (testing purposes) or in any form you want
 - (+) The underlying technologies are interchangeable, you may decide to use [LXC](#) instead of [KVM](#)
 - (+) Not all components are necessary (e.g. [Swift](#)), you may decide to do not use it or replace it by [GlusterFS](#)
- (+) The services collaborate by exchanging asynchronous messages, there is no locking synchronous call
- **This IaaS is designed to being deployed on commodity hardware :**
 - (+) You don't need to buy costly highly-available hardware such as RAID-based SAN or high performance fiber-channel
 - (+) Any node of any services can fail, the others will continue handling requests, so no single point of failure here
- (+) The [Open-Source](#) licensing means this project is community driven, no more vendor lock-in
- (+) The community of 150+ companies that have joined the project (Intel, [Canonical](#), Cisco, Red Hat, ...)
- **The high rate of releases means :**
 - (+) The project gain rapidly in maturity and features
 - (-) You may need to upgrade your infrastructure as fast as the releases are
 - (-) The documentation needs to be updated according to release (seriously !)
 - (-) **Configuration files are some kind of chaotic :** Not based on the same template, please why *-thing* called flags & why files like api-patch.ini ? This fact will not remain in future releases as there is some encouraging re-ordering
- **OpenStack is not a single apt-get'able package but a complex system based on services working together :**
 - (-) The learning curve is high, especially if you are not an expert on this domain
 - (-) You need to understand and configure technologies involved in this IaaS
- (+) The services are configured via the config files and/or the API's calls, the setup can potentially be automated

For all that reasons, I chose [OpenStack](#) !

4.4.2 The Main Components

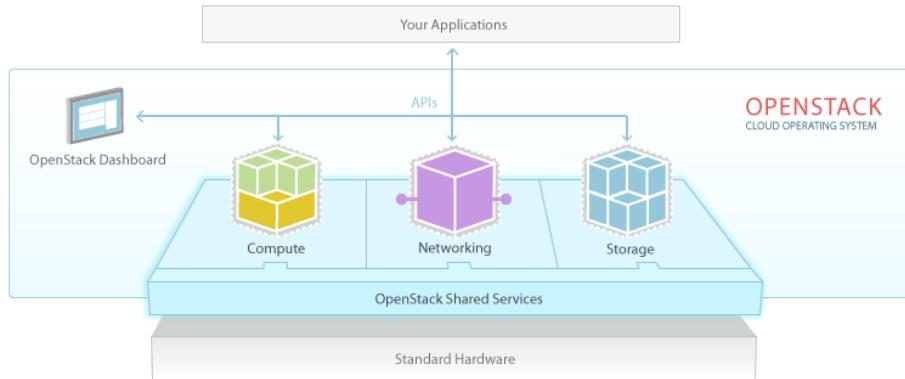


Figure 4.3: Original OpenStack's services-level diagram

- Identity Service (codename [Keystone](#)). This component provides a *centralized* Identity, Token, Catalog and Policy services intensively used by other components of [OpenStack](#). One can access to [Keystone](#) functionalities through the [keystone](#) command-line client or by using [Keystone](#) RESTful API directly. This component provides a way for users and services to authenticate using token based OAuth mechanisms. This feature is especially useful to add a security layer to any RESTful API.



- Images Service (codename [Glance](#)). This component provides a way to register, discover and retrieve virtual machine images and images metadata. One can access to [Glance](#) functionalities through the [glance](#) command-line client or by using the RESTful API directly. [Glance](#) can use various storage technologies to store the images, from simple filesystem to object-based storages like [Swift](#) or [AmazonS3](#).

- Block Storage (codename [Cinder](#)) originally developed by [NASA](#) and known as [nova-volume](#) in previous [OpenStack](#) releases. This component provides persistent, reusable storage volumes available for virtual machine instances. The volumes can be attached and detached to instances in the form of iSCSI mounted block storages. One can access to [Cinder](#) functionalities through the [cinder](#) command-line client or by using the RESTful API directly. This component is ideal for performance-critical storage mounted by virtual machines running low latency I/O applications such as databases. [Cinder](#) has a snapshot management capabilities to volumes, this powerful functionality is available thanks to the underlying technology used by [Cinder](#), called [LVM](#).

- Object Storage (codename [Swift](#)) originally developed by Rackspace. This component allows users to store, update and retrieve files in the form of objects (e.g. hash table) stored in a highly available, distributed, eventually consistent object/blob store. One can access to [Swift](#) functionalities through a simple RESTful API or a [AmazonS3](#) compatibility layer. This is not a traditional filesystem with folders and files but rather a sort of key-value store ideal for static data such as pictures, audio files, ... This store can be installed on commodity hardware in order to create a storage cluster based on [Swift](#).



- Network Service (codename [Quantum](#)) originally developed by [NASA](#) and known as [nova-network](#) in previous [OpenStack](#) releases. This component provides flexible, virtual/physical networks connectivity to virtual machine instances. [Quantum](#) is a tool aimed by the goal of providing network administrators with a simple but yet powerful approach to manage next-generation networks. One can access to [Quantum](#) functionalities through the [quantum](#) command-line client or by using the RESTful API directly. This component is ideal for managing highly-complex networking models mixing physical and virtual network and equipments. The pluggable design of [Quantum](#) (and of [OpenStack](#) in general) allow administrators to choose tools around [Quantum](#) such as the underlying network virtualization technology like [Open-vSwitch](#) or [Bridge-Utils](#).



- Compute Service (codename [Nova](#)) originally developed by [NASA](#). This component provides on-demand computing resources in the form of virtual machines instances managed by this cloud computing fabric controller (the main part an [IaaS](#)). One can access to [Nova](#) functionalities through the [nova/nova-manage](#) command-line clients or by using the RESTful API directly. In previous release, this component was also responsible of the network and volume services, each of these two services are now the responsibility of [Quantum](#) and [Cinder](#) projects. The flexible design of [Nova](#) let you choose tools and hardware around [Nova](#) such as the underlying hypervisor and the kind of computer's configuration (e.g. bare metal / HPC ...). The hypervisor choice is really a good thing, you may choose to use the widely used full-(para)virtualization hypervisor called [KVM](#) or to switch to a low overhead, high-density container-based isolation called [LXC](#).

| Instance Name | IP Address | Size | Status | Task | Power State | Actions |
|-------------------|-------------|--------------------------------|--------|----------|-------------|--------------------------------|
| test-www.demo.com | 10.4.128.20 | 4GB RAM 2 VCPU 10.0GB Disk | Active | None | Running | <button>Edit Instance</button> |
| test-www.demo.com | 10.4.128.19 | 4GB RAM 2 VCPU 10.0GB Disk | Build | Spawning | No State | <button>Edit Instance</button> |
| myserve | 10.4.128.18 | 2GB RAM 1 VCPU 10.0GB Disk | Active | None | Running | <button>Edit Instance</button> |
| myserver | 10.4.128.16 | 2GB RAM 1 VCPU 10.0GB Disk | Active | None | Running | <button>Edit Instance</button> |

Figure 4.4: Launching an instance with OpenStack Horizon

- Web User Interface (codename [Horizon](#)). This component provides a web user interface to manage and access to other [OpenStack](#) components functionalities by mapping interface interaction to [OpenStack](#) component's APIs calls. Third-party extensions can be plugged-in to this interface to extend and add features and services such as real-time monitoring, resources consumption billing

4.4.3 The Conceptual Architecture

The services briefly introduced in previous chapter needs to collaborate in order to build a complete cloud infrastructure, an [IaaS](#). For this integration to be successful, [OpenStack](#) is designed as such :

- **The services works together by calling other services :**
 - Each service functionalities are callable through corresponding RESTful API
 - Each service act as an user, it means that [Keystone](#) authentication mechanisms applies for users & services
- **The API's calls are handled in the form of messages handled by a message broker such as [RabbitMQ](#), ... :**
 - Each request is queued into the message's brokers queues and asynchronously consumed by [OpenStack](#) service's nodes

- Any available node of any service is able to handle service's requests, there is no single point of failure here

So, conceptually, you can picture the relationships between the services as such :

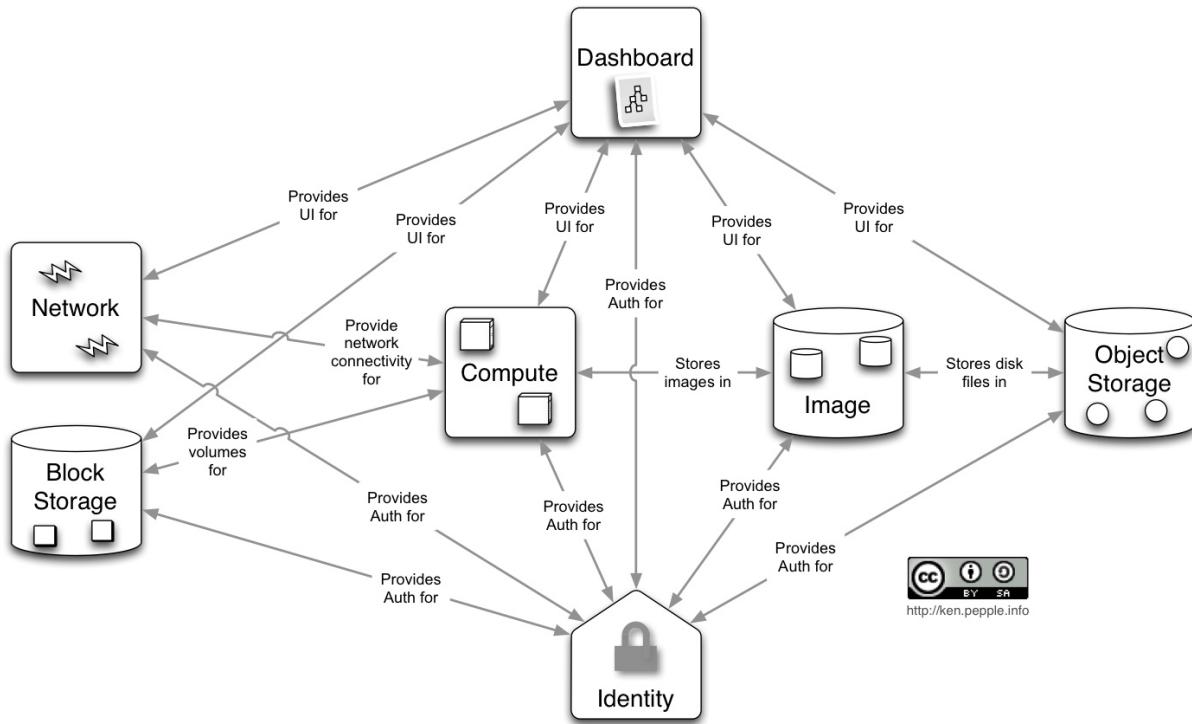


Figure 4.5: Nice diagram showing the main components of OpenStack

” This is a stylized and simplified view of the architecture, assuming that the implementer is using all of the services together in the most common configuration. It also only shows the *operator* side of the cloud – it does not picture how consumers of the cloud may actually use it. For example, many users will access object storage heavily (and directly). “

4.4.4 Logical Architecture

” As you can imagine, the logical architecture is far more complicated than the conceptual architecture shown above. As with any service-oriented architecture, diagrams quickly become “messy” trying to illustrate all the possible combinations of service communications. The diagram below, illustrates the most common architecture of an OpenStack-based cloud. However, as OpenStack supports a wide variety of technologies, it does not represent the only architecture possible. “

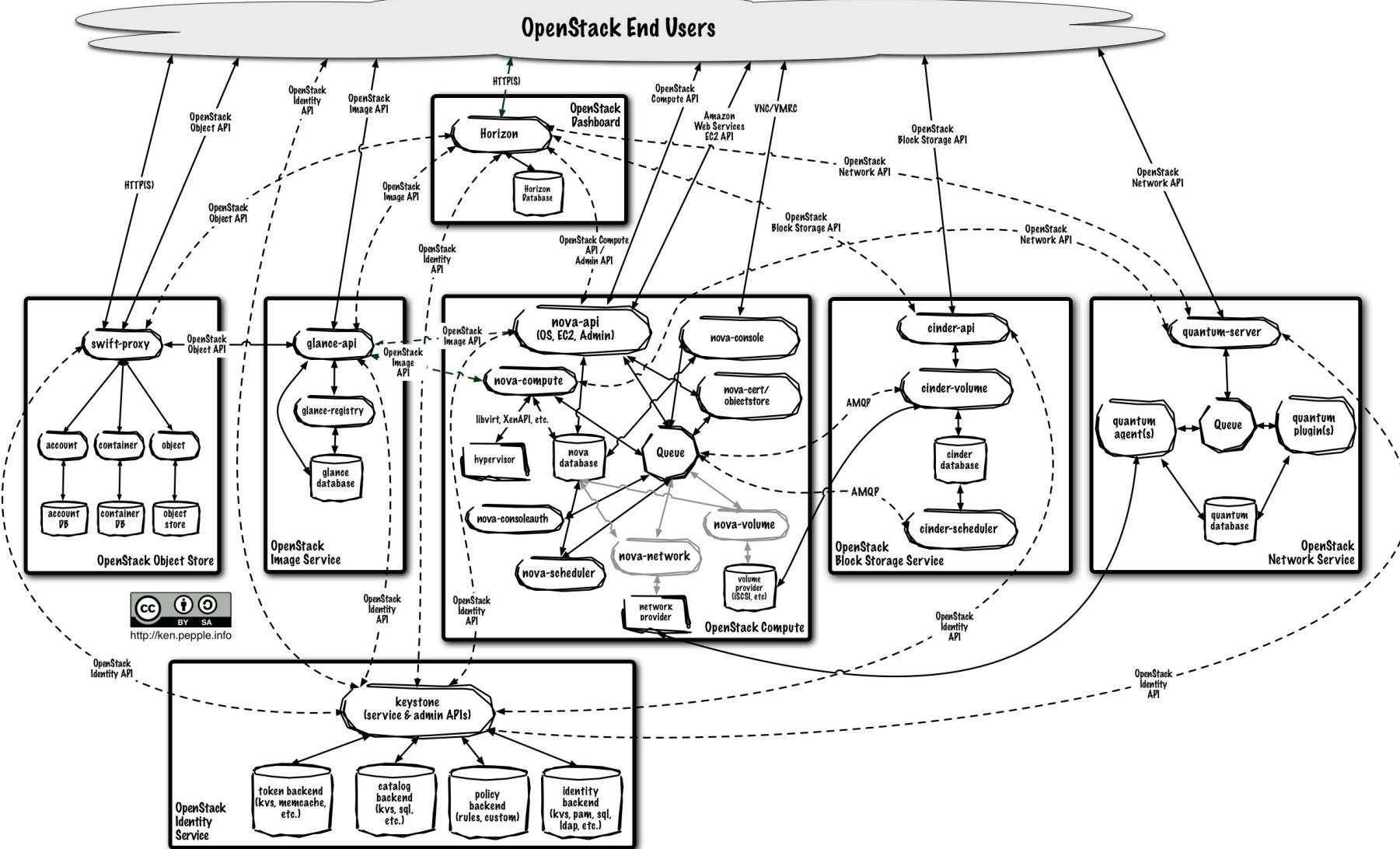


Figure 4.6: Nice diagram showing underlying softwares involved in an OpenStack setup

4.4.5 A Standalone Setup

Difficulty o . . .

Use case Development

Hardware 1 Desktop Computer

Software Ubuntu

The easiest way to test OpenStack is to install all components in one desktop computer running Ubuntu !

The fastest way to deploy this setup is to use an already available installation script e.g. DevStack.

Remark: I really like the idea behind this script, as installing OpenStack is a really complex task requiring a lot of trials and debugs (thanks to documentation ...). You actually can explain how to setup something or you can write scripts and add some documentation around it : Documentations become tools !

4.4.6 A Typical Setup

Difficulty o o o .

Use case Development - Production

Hardware 2+ Desktop/Server Computers

Software Ubuntu

The following multi-node deployment is designed to separate the critical services (called controller) of the computing services (called compute nodes). This setup was pretty well documented and it seem that the more scalable setup is getting more popular now.

4.4.7 A More Scalable Setup

Difficulty o o o o

Use case Production

Hardware 3+ Desktop/Server Computers

Software Ubuntu

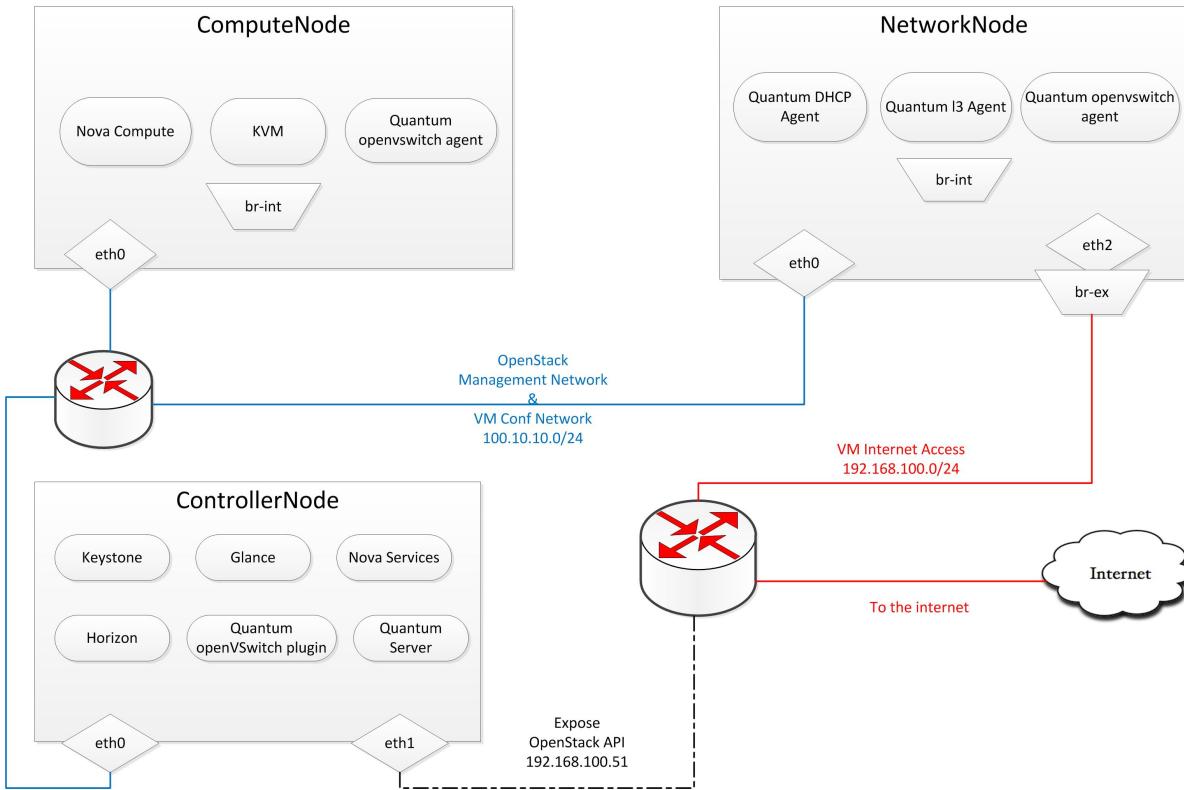


Figure 4.7: This multi-node setup separate the critical services, the compute service and the network GW

The setup of the guide from where this diagram comes from, specify that the controller node will host the following services :

- The shared database MySQL
- The message broker RabbitMQ
- The identity service Keystone
- The images service Glance
- The volumes service Cinder
- The compute service Nova (api, scheduler, ...)
- The network service Quantum (server, ...)
- The web user interface Horizon

The network node will host the following services :

- The network service Quantum (dhcp & layer3 agents, ...)
- The virtual switching technology Open-vSwitch

The compute nodes will host the following services :

- The network service Quantum (open-vswitch agent, ...)
- The virtual switching technology Open-vSwitch
- The compute service : Nova (compute-kvm, ...)
- The hypervisor technology KVM

4.4.8 A MAAS/Juju Powered Setup

Difficulty o o . .

Use case Production

Hardware 6+ Desktop/Server Computers

Software Ubuntu

See Also:

Please see [Clouds Orchestration Layer](#) for further details about JuJu.

If you plan to deploy OpenStack on a larger scale, you may be interested by using the metal-automation tool called MAAS from Canonical. When you install the first server with Ubuntu Quantal Server you specify to setup this servers as the MAAS master.

As an administrator, you will get access to a simple but rather efficient web user interface in order to *plug-in* new servers to the setup. Typically you only need to configure servers BIOS in order to enable Wake-on-LAN, enable remote-boot via PXE and take note of the network interfaces MAC addresses. Then, you will only specify servers MAC addresses to the MAAS master. The nodes will automatically be handled by the master and configured with Ubuntu.

Finally, you will use JuJu cloud orchestrator in order to deploy OpenStack components on your setup !

At time of writing this report, JuJu cannot merge charms¹, it means that for any service (~charm) you want to deploy, a separated instance is required (e.g. a VM for each service). The nodes handled by the MAAS provider maps the deployment charms to the server itself, without encapsulating the instance into a virtual machine. This is why at least 6 of them are required for OpenStack to be installed !

4.4.9 The setup at EBU

The demonstrator will be deployed on a small setup of 4 servers. This is the main reason why MAAS + JuJu were not used in order to deploy OpenStack.

I started my work by reading (a lot) of documentation about the topic and followed some of the best tutorials I founded. The resulting scripts are available as appendices at the end of this report, the link: [OSCIED - OpenStack Nodes Scripts](#). I developed the scripts in order to make my work repeatable, as the setup will not work at the first try for sure !

During my Thesis, I faced a lot of problems during my trials with OpenStack, partially due to the quality of the documentation (not the quantity). Trying to install latest release of OpenStack, Folsom, is not as easy as expected. Unfortunately I stopped my work on the setup to concentrate on the application when it was scheduled to do so.

Here is the setup I suggest to deploy at European Broadcasting Union (EBU/UER), strongly inspired by A full guide for OpenStack Folsom with Quantum (GRE 2NICs).

¹ Charms are DevOps distilled. In brief, they are encapsulated services to be connected and scaled on demand.

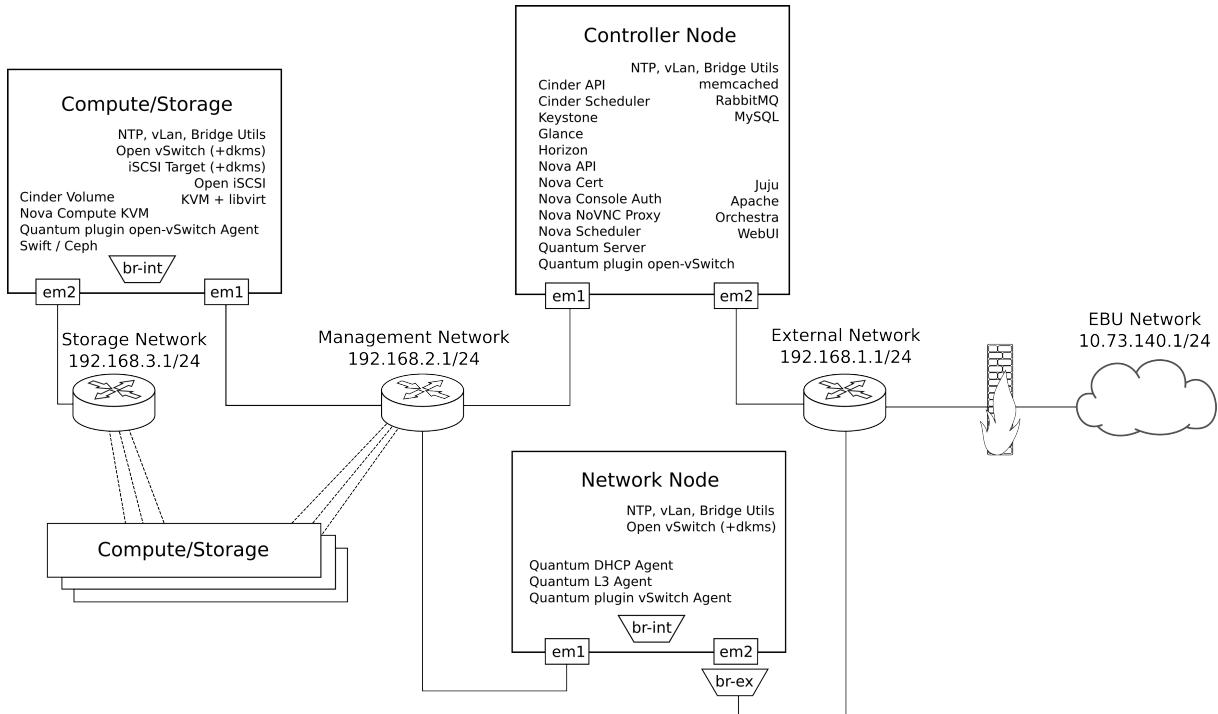


Figure 4.8: The proposed OpenStack setup

4.4.10 Documentation

Note: Please see [Ticket 37](#) for further documentations (30+ links).

- Official Documentation : [OpenStack Documentation](#)
- A full installation guide for [OpenStack Folsom with Quantum](#) : This step-by-step guide is one of the most interesting guide mostly explained with **real** code snippets
- A full guide for [OpenStack Folsom with Quantum \(GRE 2NICs\)](#) : This is the guide that inspired my for the [OpenStack](#) production setup
- [DevStack](#) : This is a documented shell script to setup a development [OpenStack](#) environment
- [TryStack](#) : This is a cluster setup running [OpenStack](#) and available online for free to try

4.5 Clouds Orchestration Layer

4.5.1 Introduction

This project cannot be successful without automation !

We choose to use cloud technologies in order to automate the usage of computing resources. Next step is to automate usage of cloud(s) (**IaaS**) in order to deploy services of the demonstrator. Moreover it is not only necessary to use clouds in an automated manner, but also to automate OSCIED itself.

Remaining the fact that the project's application is split into components to be scalable ...

So, each of these components must be able to automatically :

- Install and configure any required service (e.g. FFmpeg)
- Manage the internal service's daemons (e.g. start, stop)
- Handle the relation with other components of the application

This is a rather complex task that requires a lot of work ... We need **JuJu** !

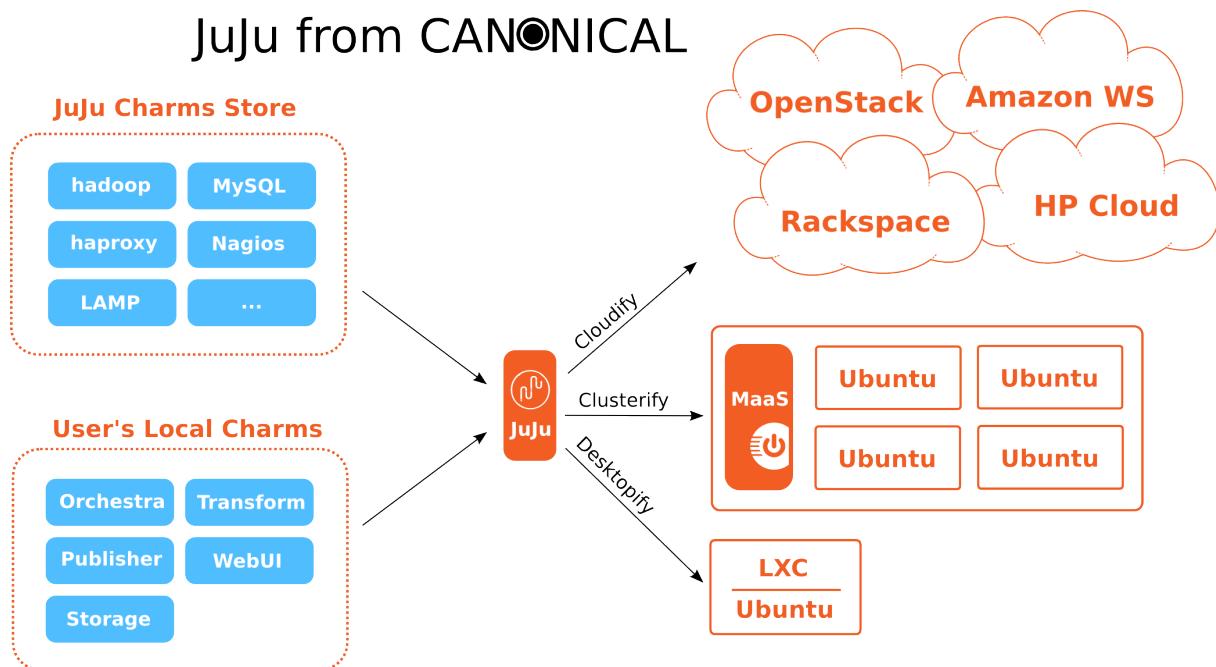


Figure 4.9: JuJu can deploy charms to a wide variety of environments, why not implementing your own provider ?

The goal of the following section is not to duplicate the official documentation from [Canonical](#). I prefer to introduce you with the **how** and **why** OSCIED actually integrates **JuJu**.

See Also:

Please see [JuJu](#) has some of the notions related to **JuJu** are necessary.

4.5.2 OSCIED + JuJu = ?

Integrating **JuJu** to OSCIED improved the automation of the project, by :

- **Packaging each of the application's component in a charm, requiring implementation of charm's :**
 - Automation scripts, called hooks (install, config-changed, start, stop, ...-relation-...)

- Metadata (`metadata.yaml`, e.g. name, description, ...)
- Configuration (`config.yaml`, e.g. mysql root password ...)
- The service source code itself (e.g. `orchestra.py`)
- Creating scripts to improve easiness of use of the preliminary demonstrator
- Future : Integrating **JuJu** to the app. itself, thus permit features like auto-scale, ...

All that work to finally, helped with **JuJu**, make OSCIED able to be :

- **Deployed on a wide variety of environments, thanks to **JuJu**'s providers :**
 - Clouds : [OpenStack](#), [Amazon AWS](#), [HP Cloud](#), [Rackspace](#)
 - Clusters : [MAAS](#)
 - Computers : [LXC](#)
- **Linked to 100+ already available charms, thanks to **JuJu Charms Store** :**
 - Databases : [CouchDB](#), [MongoDB](#), [MySQL](#)
 - Message brokers : [RabbitMQ](#), ...
 - Monitoring : [Ganglia](#), [Nagios](#), ...
 - Proxies : [HAProxy](#), [Nginx](#), ...
 - Storages : [Cassandra](#), [Ceph](#), [GlusterFS](#), [Hadoop](#), ...
 - Websites : [Apache 2](#), [LAMP](#), [Django](#), [Drupal6](#), ...
 - [OpenStack](#) (!) : [Cinder](#), [Glance](#), [Keystone](#), [Nova](#), ...
- **Managed easily (Of course this is **JuJu** that triggers the scripts) :**
 - Life-cycle of components are handled by automated charm's hooks
 - Relations between components are handled by automated charm's hooks
- **Future-proof, based on continuously improved tools with strong community support :** Ubuntu founder Mark Shuttleworth and former CEO of Canonical Ltd.² revealed that 2013 will be the year of the cloud in the sense that the enterprise will focus on making the cloud easier to use. Concretely, they will improve the already innovative projects called **JuJu** and **MAAS**. Canonical also contributes to [OpenStack](#) project by distributing and supporting [OpenStack](#) and are a Platinum Member of the [OpenStack Foundation](#) board.³

4.5.3 JuJu Tips & Facts

Here are some of the tricks & facts I collected by using **JuJu**, maybe helpful ...

- `$HOME/.juju/environments.yaml` : Here are configured the hosting environments for your services
- The local provider is quite useful to develop and test charms
- When you `juju bootstrap` an environment, a dedicated unit (`juju`) is deployed in order to manage the services
- When you use the `juju` command-line tool to manage an environment (status of running instances, ...), you interact with the orchestration unit running on this environment
- The communications (`ssh`) are secured with your private certificate `$HOME/.ssh/id_rsa`
- The services can be connected (e.g. *lamp* -> *mysql*) this is also handled by the hooks of the charms
- Each of the charms are sort of packaged services, the setup, start/stop of the service is handled by the hooks of the charm.

² In December 2009, he stepped down as the CEO of Canonical, Ltd. to focus energy on product design, partnership and customers.

³ Ubuntu UDS R – Mark Shuttleworth Keynote [Ubuntu 13.04] - <http://www.youtube.com/watch?NR=1&feature=endscreen&v=0voGsibCjHE>

- You can choose the type of instance, e.g. `juju deploy --constraints instance-type=c1.medium mysql`
- You can use your own configuration, e.g. `juju deploy --config your_config.yaml mysql`
- You can get a graphical representation of an environment `juju status --format svg --output status.svg`
- When only one service is erroneous (e.g. *mysql* install failed) you do not need to destroy the whole environment `juju destroy-service mysql`
- Do not hesitate to open another terminal and `juju debug-log`, if something fail, you have the log !
- You can `ssh` any running unit with `juju ssh <unit_name>` or `ssh ubuntu@<unit_public_ip>`
- Do not forget to `juju expose` service you want to expose !
- Just try to `ssh` to any running unit and go to path `/var/lib/juju/units/<unit_name>/`, interesting right ?
- You can access to the environments you deployed from any computer, you only need your rsa key and your environments file

4.5.4 Charm's Life-cycle

Here is shown in a state machine the life-cycle of a charm (relation's triggers are not represented) :

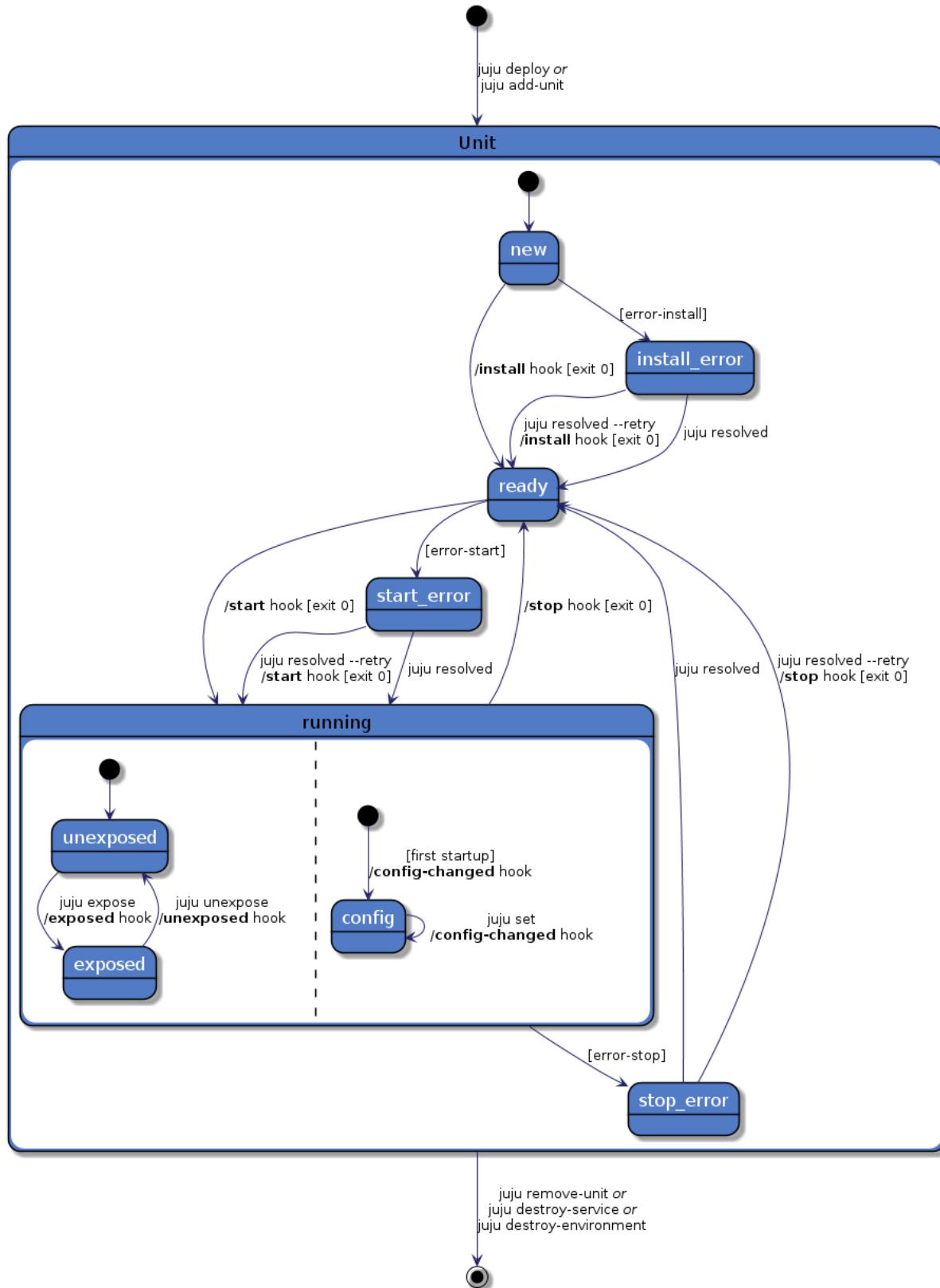


Figure 4.10: State machine representing the life-cycle of an unit (instance of a charm)

4.6 Application Layer

4.6.1 Structure of the Application

The application is composed of charms deployable with **JuJu** the cloud orchestrator.

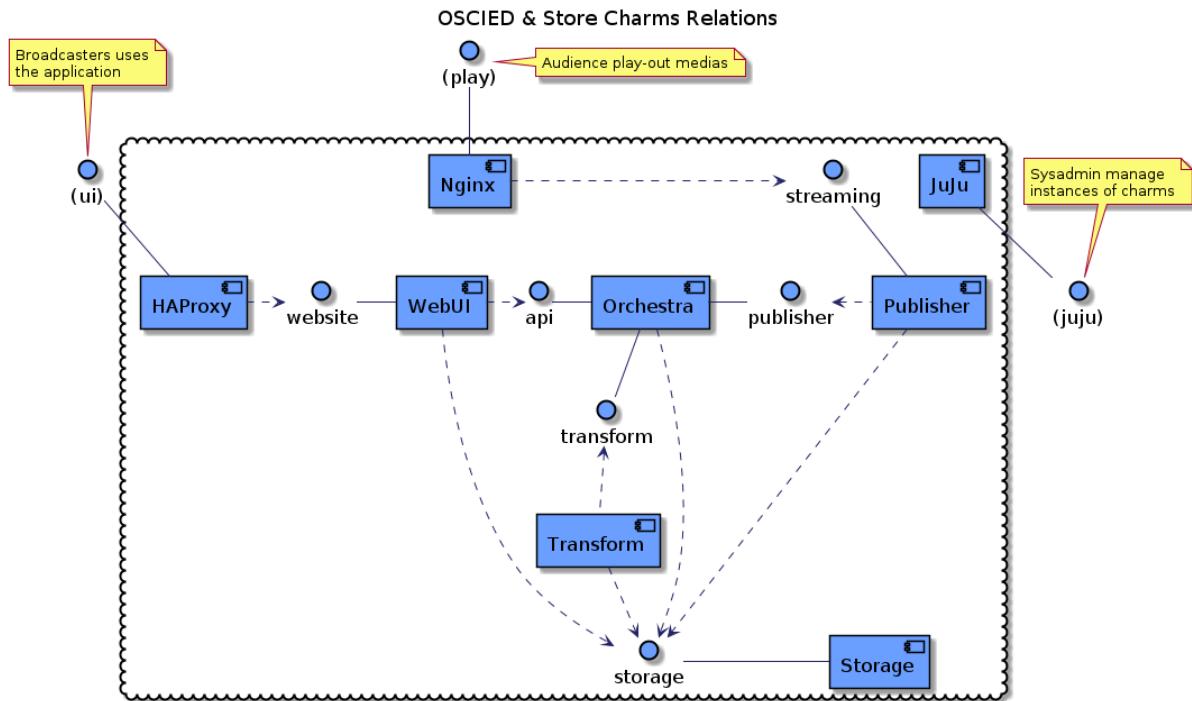


Figure 4.11: Charms of the project including charms from the store

This is an UML components diagram showing the architecture of the application deployed into a cloud (Cookie icon).

Here are the charms developed for this application :

| Charm | Short description (services and goals) | Provides | Requires |
|-----------|--|-------------------------|-------------------|
| WebUI | Provides a web based interface for the users of the platform (e.g. broadcasters) | website | storage api |
| Orchestra | Provides the RESTful API and handles the DB & jobs scheduling (the brain) | api transform publisher | storage |
| Transform | Handles media encoding jobs to transform medias from/to various formats | (nothing) | storage transform |
| Publisher | Handles media publication jobs to make medias available for the audience | (nothing) | storage publisher |
| Storage | Provides a shared medias storage mounted by other components of the application | storage | (nothing) |

The *provides* and *requires* columns are the name of the relations required or provided by the charm.

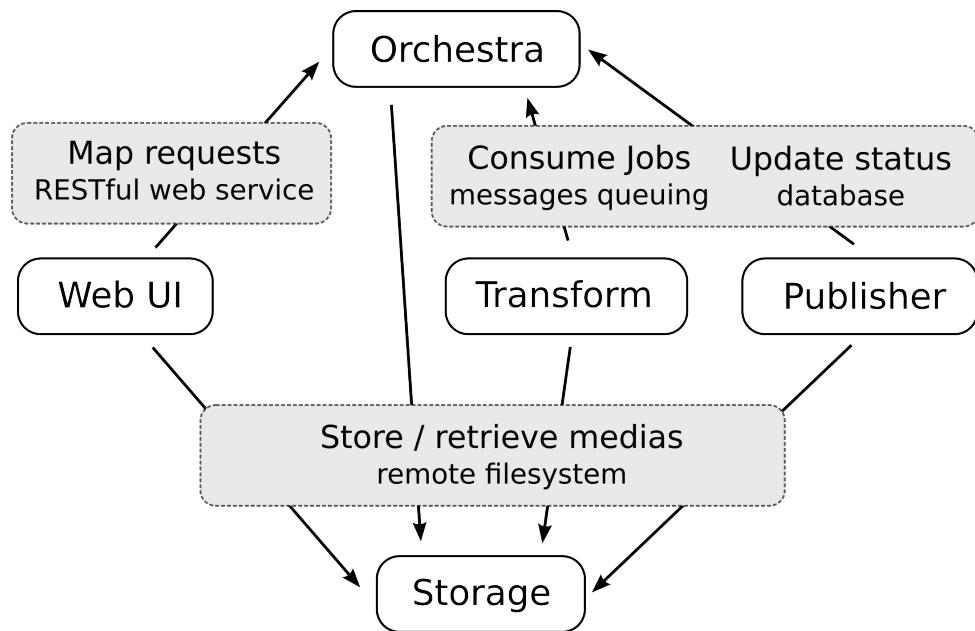


Figure 4.12: Purpose of the relations between components of OSCIED

Here are the relations :

| Relation | Interface | Provider side | Requirer side |
|-----------|-------------|---|-------------------------------------|
| api | orchestra | Send RESTful API url | Update config. with RESTful API url |
| website | http | Update config. & add proxy to white list | Enable HTTP redirection |
| transform | subordinate | Send database & message broker connections | Update config. & connect to broker |
| publisher | subordinate | Send database & message broker connections | Update config. & connect to broker |
| storage | mount | Send parameters required to mount the storage | Update config. & mount the storage |

Remark: Don't panic, they are only the required credentials to access to jobs related database !

The application's charms can be connected to charms of the [JuJu Charms Store](#) thanks to the nice contributors behind every of the charms. For example, one can imagine to plug [Nginx charm](#) in front of (e.g. 5x) publication points and closer to the user to reduce load and network traffic of backend publication points !

As you can imagine, next diagram will be a little bit more complicated, as it enter into charms to show you the Open-Source tools used internally by them.

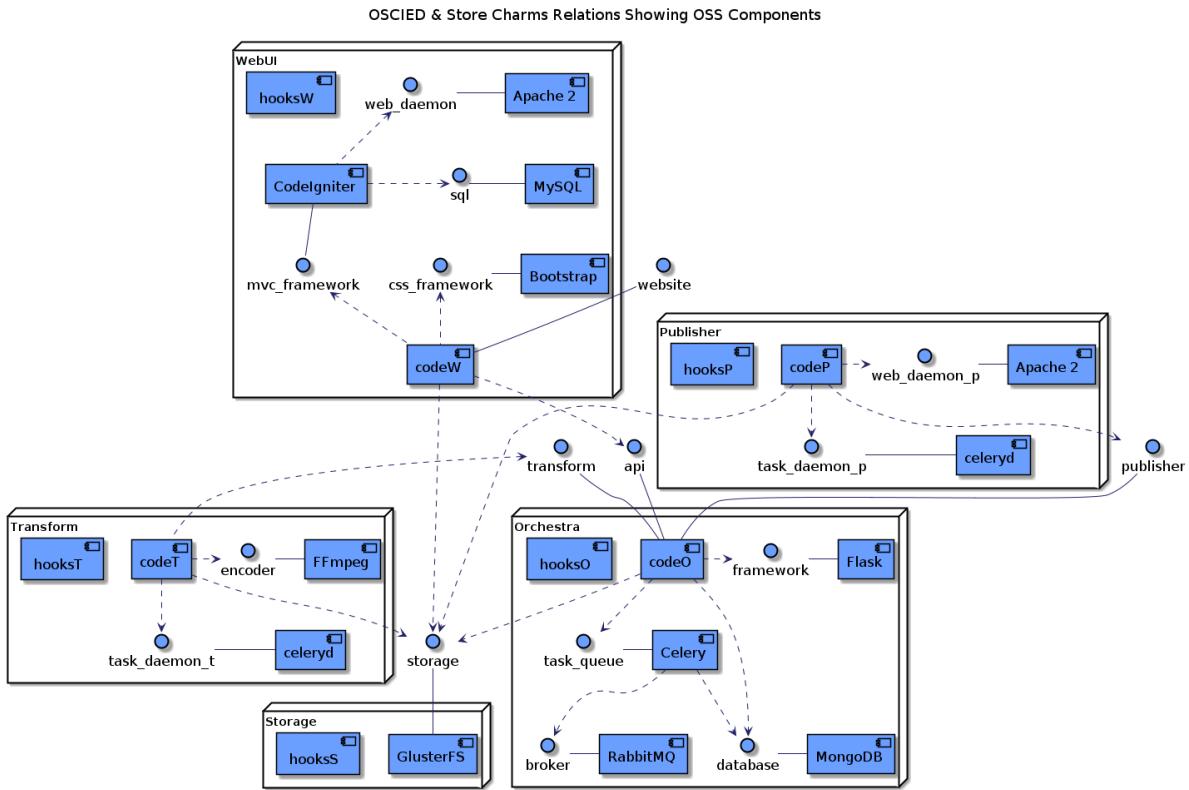


Figure 4.13: Charms of the project including involved sub-components

The charm's hooks (bash scripts) are represented by `hooks<X>` and the service's implementation (python source-code) is represented by `code<X>`.

4.6.2 OSCIED Advantages

Pluggable Design

The components of the application are defined in the form of pluggable charms and the relation between them are defined in the form of interfaces. One can potentially implement a compatible charm (eg. *StorageV2*) with the required interface and behavior and plug-in this new charm to any of the components requiring the implemented service.

For example, the *Storage* charm implements the storage service and provides the *storage* relation based on the *mount* interface. The *Orchestra* charm requires the *storage* relation based on the *mount* interface and you can plug the *Storage* units to the *Orchestra* as they are compatible !

Note: Please see [Application Layer](#) for further details.

So, for the needs of this preliminary demonstrator, the simplest form of a *GlusterFS* server is encapsulated into the *Storage* charm. This charm actually isn't capable of handling the scale-up/down of the service (adding or removing of instances, e.g. `juju add-unit`).

However, thanks to [JuJu](#) and to the pluggable design of this application, it is actually possible to go beyond this limiting factor by using your own network storage (see examples) !

Here are some example of what one can use for the storage service :

Implementations of the Storage Service

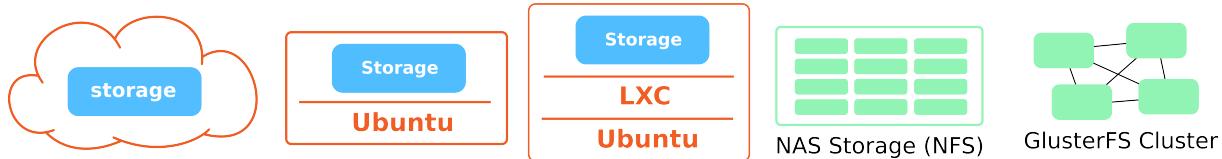


Figure 4.14: The project's storage charm can be replaced by any compatible physical or virtual storage

Note: If you prefer to use any external storage in place of the proposed charm, you need to specify the options related to storage in the configuration files used by juju to deploy the other components !

Strengths & Weaknesses

The strengths and weaknesses of this project are :

- (+) The project is based on [Open-Source](#) cloud-era tools in order to be elastic and scalable !
- (+) The [Open-Source](#) licensing means this project is community driven, no more vendor lock-in
- **The project is handled in a professional manner :**
 - (+) It is under revision control called [subversion](#)
 - (+) It is managed with a ticket system called [TRAC](#)
 - (+) The RESTful API methods responses are tested with [JSONLint](#)
 - (+) The RESTful API methods are tested with scripted unit-tests
- **The modular architecture add flexibility to any deployment :**
 - (+) The components can be deployed in standalone (testing purposes) or in any form you want
 - (+) The underlying technologies are interchangeable, you may decide to use [NFS](#) instead of [GlusterFS](#)
- (+) The services collaborate by exchanging asynchronous messages, there is no locking synchronous call
- **The application is designed to being deployed on commodity hardware :**
 - (+) You don't need to buy costly highly-available hardware such as RAID-based SAN or high performance fiber-channel
 - (+) Units of the services can fail, the others will continue handling requests, so no single point of failure here ⁴
- **The cloud orchestrator [JuJu](#) add some kind of magic to the project, the application :**
 - (+) Is easy to deploy thanks to automation handled by application's charms hooks scripts
 - (+) Is scalable as it is easy to adapt services to load by adding or removing units to services
 - (+) Is flexible and deployable to a wide variety of targets, such as clouds, clusters, servers ...
 - (+) Is pluggable to charms developed by the community such as haproxy, nginx, nagios, ...
- (+) The orchestrator provides a RESTful API, one can implement a higher-level tools based on OSCIED !
- **The distributed tasks queue [Celery](#) add some kind of magic to the orchestrator :**

⁴ This is true for the services that actually can scale-up/down such as the transform, publisher and webui charms.

- (+) The enterprise business rules can be implemented by connecting the workers⁵ to the right tasks queues and sending jobs to the right queues.
- The preliminary demonstrator is not perfect, some work is required to make it better, actually :
 - (-) The storage charms doesn't handle clustering (not scalable)
 - (-) The orchestrator charm cannot be highly-available (not scalable)
 - (-) The orchestrator charm cannot auto-scale the workers
 - (-) The orchestrator only uses the basic features of Celery !

4.6.3 Various OSS Tools Involved

Operating system : Ubuntu Quantal Server from Canonical

“Ubuntu is a computer operating system based on the Debian Linux distribution and distributed as free and Open-Source software, using its own desktop environment. ... Ubuntu is sponsored by the UK-based company Canonical Ltd., owned by South African entrepreneur Mark Shuttleworth. Canonical generates revenue by selling technical support and services related to Ubuntu, while the operating system itself is entirely free of charge. The Ubuntu project is committed to the principles of free software development; people are encouraged to use free software, improve it, and pass it on. “ source Wikipedia ([UbuntuOS](#))

4.6.4 Media's State Machine

It may seem obvious but the application not only stores media files into the shared storage but it also stores informations about it into database such as metadata (title, add_date, ... whatever you need), the id of the user who registered the media, ... And the actual state of the media :

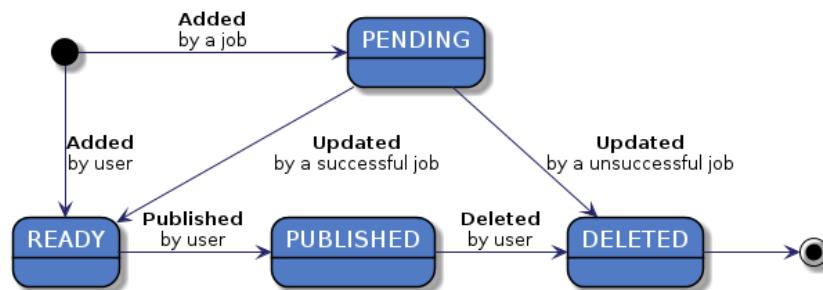


Figure 4.15: State machine of a media from registration to deletion

⁵ They are actually two kind of workers, the transform (encoding jobs) and publisher (publication jobs).

4.6.5 OSCIED-Orchestra : The Orchestrator

See Also:

You can browse the source of the Orchestrator or browse [OSCIED - Orchestra RESTful API](#) for further details.

OSS Tools

- [Flask](#) Python Micro Web Framework
- [PyMongo](#) Python module for working with [MongoDB](#)
- [MongoDB](#) Scalable, High Performance NoSQL Database from 10gen
- [RabbitMQ](#) AMQP Message Broker from vmware
- [Celery](#) Distributed Task Queue
- [JuJu](#) Cloud Orchestrator from Canonical

Introduction

This component is the brain of the application, responsible of :

- the RESTful API, to expose application's functionalities to user
- the database, to store application's data (users, profiles, jobs, ...)
- the message broker, to communicate with workers (transform & publisher)
- the cloud orchestrator, to manage other components⁶

The main advantage of providing an API *and* a separated web user interface is that one can use functionalities of the application by programming an higher-level tool using the API directly. It was also really useful for me to develop & test the API by scripting uses cases helped with [cURL](#).

The orchestrator is developed in [Python](#) and I actually choose this programming language for the following reasons:

- The tools used for this project, [OpenStack](#), [JuJu](#), ... are developed in [Python](#)
- The most interesting [Open-Source](#) tools involved in this charm are also developed in [Python](#)
- I like writing lesser code and I actually really want to practice this language !

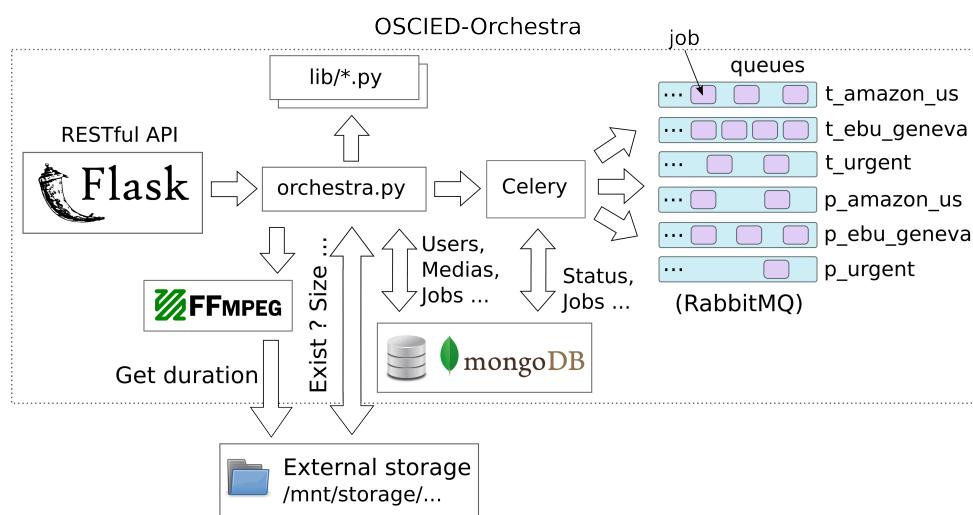


Figure 4.16: Architecture of the Orchestrator

⁶ This feature will be available on a future release.

Charm's Configuration

You can start the charm without specifying any configuration (default values will be used, see [OSCIED - Orchestra Source Code](#)) but I strongly recommend to specify your own values in production !

- **verbose** Set verbose logging
- **root_secret** Secret key used by API clients to manage users
- **nodes_secret** Secret key used by workers/nodes to callback API when they finish their job
- **repositories_user** OSCIED charms repositories client username
- **repositories_pass** OSCIED charms repositories client password
- **webui_repository** OSCIED Web UI charm will be checked out locally under ~/charms/(release)/oscied-webui
- **transform_repository** OSCIED Transform charm will be checked out locally under ~/charms/(release)/oscied-transform
- **publisher_repository** OSCIED Publisher charm will be checked out locally under ~/charms/(release)/oscied-publisher
- **mongo_admin_password** Database administrator password
- **mongo_nodes_password** Database nodes password ⁷
- **rabbit_password** Messaging queue user's password ²
- **storage_ip** Shared storage hostname / IP address (see interface mount of NFS charm) ⁸
- **storage_fstype** Shared storage filesystem type (e.g. NFS) ³
- **storage_mountpoint** Shared storage mount point (e.g. for NFS - /srv/data) ³
- **storage_options** Shared storage options (e.g. for NFS - rw,sync,no_subtree_check)

⁷ This secret is forwarded by the coordinator to managed units (transform, publish)

⁸ If all options are set this will override and disable storage relation

Charm's Hooks Activity Diagrams

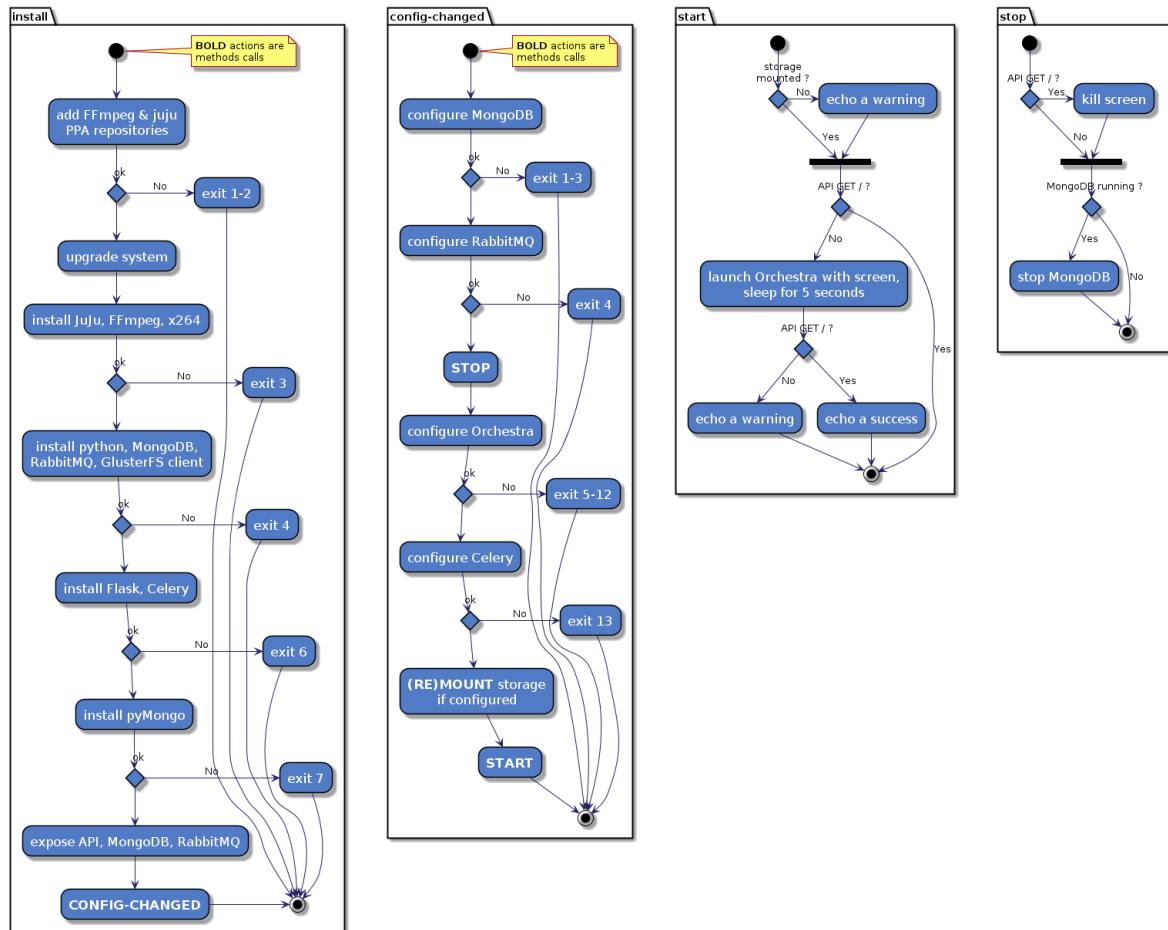


Figure 4.17: Activity diagram of Orchestra unit life-cycle hooks

Charm's Relations

- Provides : API [Orchestra], Transform [Subordinate], Publisher [Subordinate]
- Requires : Storage [Mount]

Warning: The unit's daemons will not start until a shared storage is mounted (via the storage relation or by specifying it into configuration).

4.6.6 OSCIED-Transform : The Transcoder

See Also:

You can [browse the source of the Transcoder](#)

OSS Tools

- Celery Distributed Task Queue
- FFmpeg Complete Multimedia Framework from the FFmpeg Foundation

Introduction

This component is the worker specialized in handling transformation jobs. In fact this is *celeryd* daemon that handles the requests and maps jobs to transform functions calls. This charm's start hook will launch and connect the daemon to the message broker's queue(s) specified in configuration⁹.

For example, one can choose that the workers running on his private & high priority transformation requests by setting worker's *rabbit_queues* option to "t_priv,t_high". Then one only need to launch jobs of such kind in one of the defined queues (*t_priv*, *t_high*) and that's it !

Moreover, one can choose to explicitly target a unique worker (e.g. *myWorker1*) by sending jobs to the queue *myWorker1*, this is another interesting feature offered by the application.

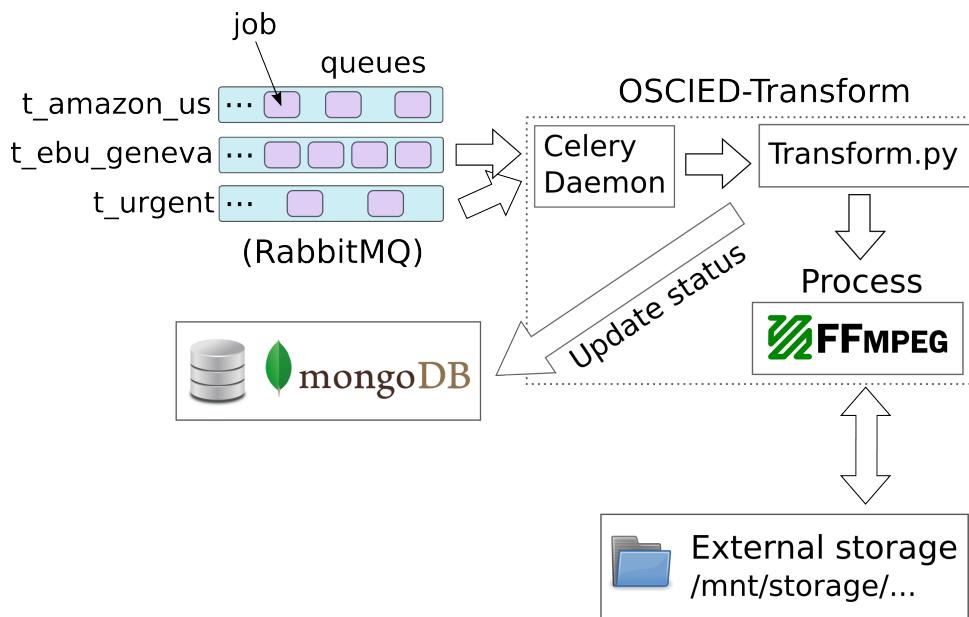


Figure 4.18: Architecture of the Transform Unit

Charm's Configuration

You can start the charm without specifying any configuration (default values will be used, see [OSCIED - Transform Source Code](#)) but I strongly recommend to specify your own values in production !

- **verbose** Set verbose logging
- **concurrency** Amount of tasks the worker can handle simultaneously
- **rabbit_queues** Worker connect to queues to receive jobs

⁹ Add worker's name to queues list, this make possible to launch jobs to this specific worker

- **mongo_connection** Orchestrator database connection ¹⁰
- **rabbit_connection** Orchestrator message broker connection ²
- **storage_ip** Shared storage hostname / IP address (see interface mount of NFS charm) ¹¹
- **storage_fstype** Shared storage filesystem type (e.g. NFS) ³
- **storage_mountpoint** Shared storage mount point (e.g. for NFS - /srv/data) ³
- **storage_options** Shared storage options (e.g. for NFS - rw, sync, no_subtree_check)

Charm's Hooks Activity Diagrams

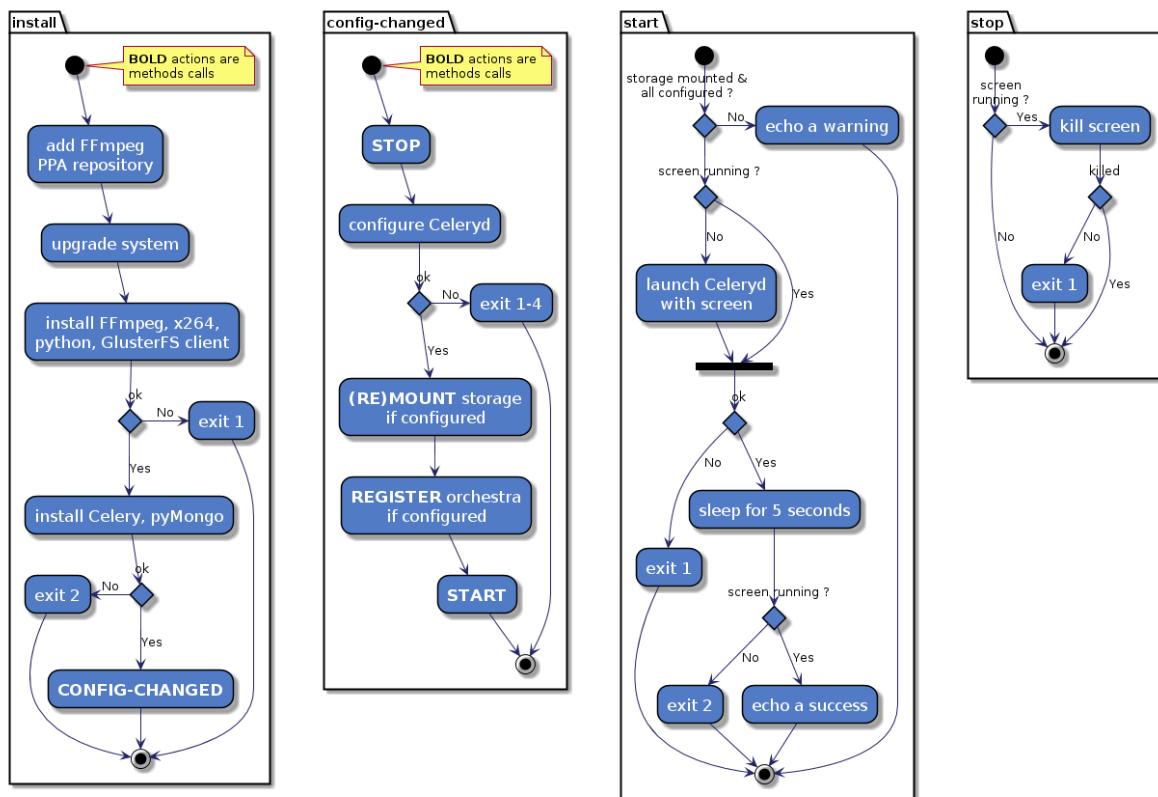


Figure 4.19: Activity diagram of Transform unit life-cycle hooks

Charm's Relations

- Provides : (nothing)
- Requires : Storage [Mount], Transform [Subordinate]

Warning: The unit's daemon will not start until both conditions are fulfilled :

- A shared storage is mounted (via the storage relation or by specifying it into configuration)
- An orchestrator is registered (via the transform relation or by specifying it into configuration)

¹⁰ If all options are set this will override and disable transform relation

¹¹ If all options are set this will override and disable storage relation

Job State Machine

The orchestrator stores informations about the transformation jobs into database in parallel to the informations that Celery also stores in. It may seem as duplicate however I choose to do as such for good reasons :

- Additional informations about the jobs can be stored.
- One can choose to replace Celery to use any other task queuing technology.
- Listing of jobs is easy to implement, no needs of Celery's inspect & filtering.

Celery's keep track of jobs and stores informations about jobs into a *backend*, the orchestrator's database (MongoDB).

” During its lifetime a task will transition through several possible states, and each state may have arbitrary metadata attached to it. When a task moves into a new state the previous state is forgotten about, but some transitions can be deducted, (e.g. a task now in the FAILED state, is implied to have been in the STARTED state at some point). ”¹²

So, the transformation jobs stored in database has a *statistic* field that is filled with values mainly generated by the orchestrator such as *add_date*. The state machine diagram shows what is store in this field plus the values that are appended to job's state metadata.

Remark: The orchestrator RESTful API transformation methods responses contains job's metadata, appended into *statistic* field.

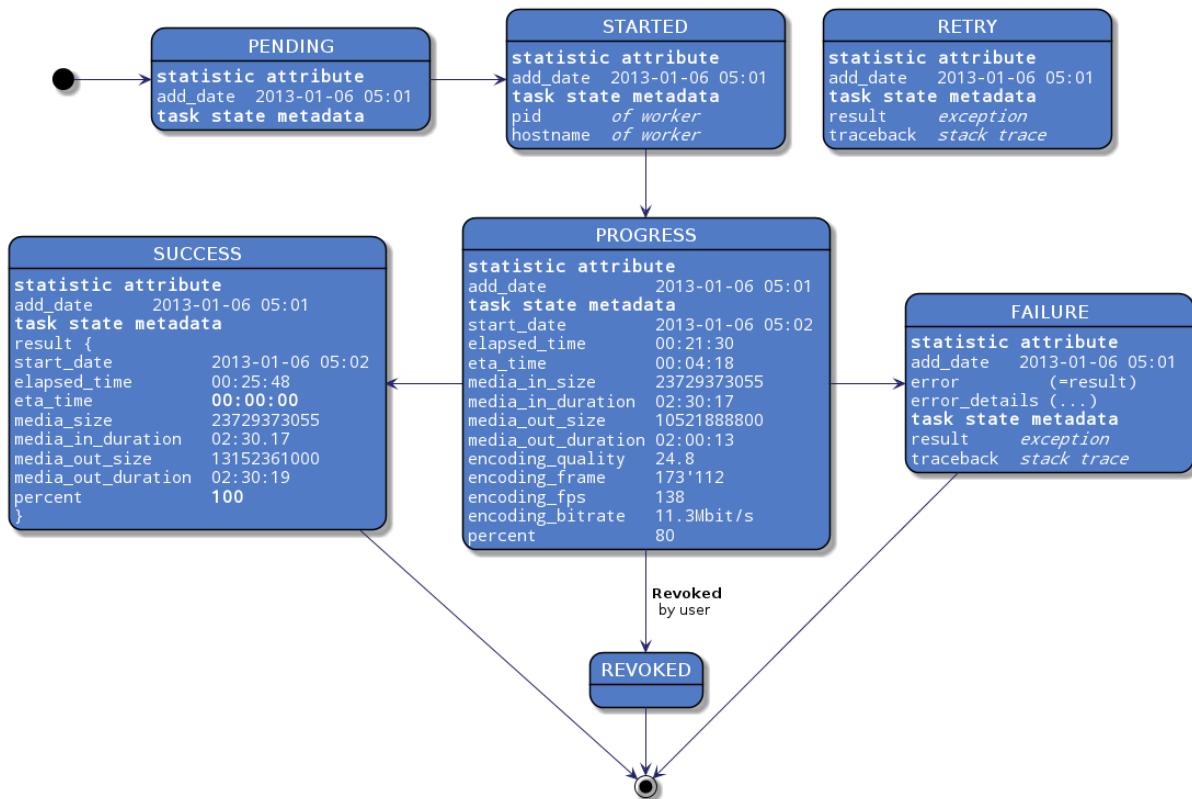


Figure 4.20: State machine of an encoding job (transform -> transform)

¹² Celery Tasks Page – <http://docs.celeryproject.org/en/latest/userguide/tasks.html>

Job Sequence Diagrams

A Successful Job

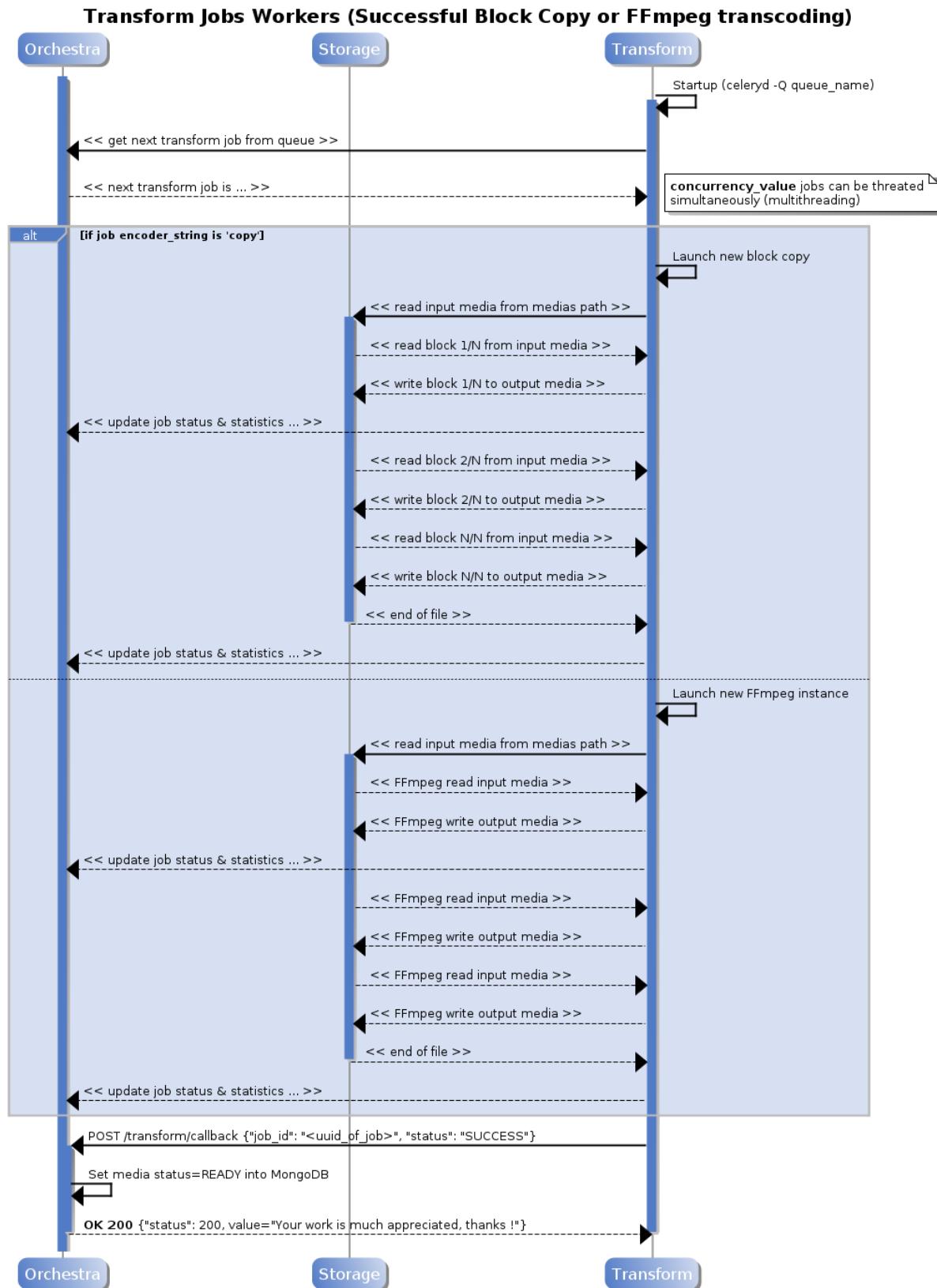


Figure 4.21: Sequence diagram of a successful encoding job (transform -> transform)

4.6.7 OSCIED-Publisher : The Publication Point

See Also:

You can [browse the source of the Publication Point](#)

OSS Tools

- Celery Distributed Task Queue
- Apache 2 HTTP Server from the Apache Software Foundation
- H264 Streaming Module from CodeShop

Introduction

This component is the worker specialized in handling publication jobs. In fact this is *celeryd* daemon that handles the requests and maps jobs to publisher functions calls. This charm's start hook will launch and connect the daemon to the message broker's queue(s) specified in configuration ¹³.

For example, one can choose that the workers running on [Amazon AWS](#) cloud will handle *public & low priority* publication requests by setting worker's *rabbit_queues* option to "t_pub,t_low". Then one only need to launch jobs of such kind in one of the defined queues (*t_pub*, *t_low*) and that's it !

Moreover, one can choose to explicitly target a unique worker (e.g. *myWorker2*) by sending jobs to the queue *myWorker2*, this is another interesting feature offered by the application.

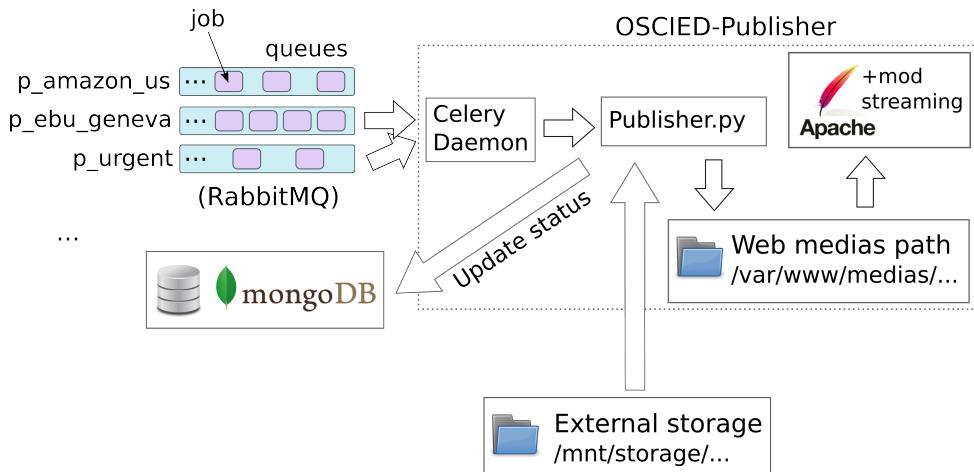


Figure 4.22: Architecture of the Publication Point

Charm's Configuration

You can start the charm without specifying any configuration (default values will be used, see [OSCIED - Publisher Source Code](#)) but I strongly recommend to specify your own values in production !

- **verbose** Set verbose logging
- **concurrency** Amount of tasks the worker can handle simultaneously
- **rabbit_queues** Worker connect to queues to receive jobs
- **max_upload_size** Maximum size for file uploads
- **max_execution_time** Maximum time for PHP scripts

¹³ Add worker's name to queues list, this make possible to launch jobs to this specific worker

- **max_input_time** Maximum time for HTTP post
- **mongo_connection** Orchestrator database connection ¹⁴
- **rabbit_connection** Orchestrator message broker connection ²
- **storage_ip** Shared storage hostname / IP address (see interface mount of NFS charm) ¹⁵
- **storage_fstype** Shared storage filesystem type (e.g. NFS) ³
- **storage_mountpoint** Shared storage mount point (e.g. for NFS - /srv/data) ³
- **storage_options** Shared storage options (e.g. for NFS - rw, sync, no_subtree_check)

Charm's Hooks Activity Diagrams

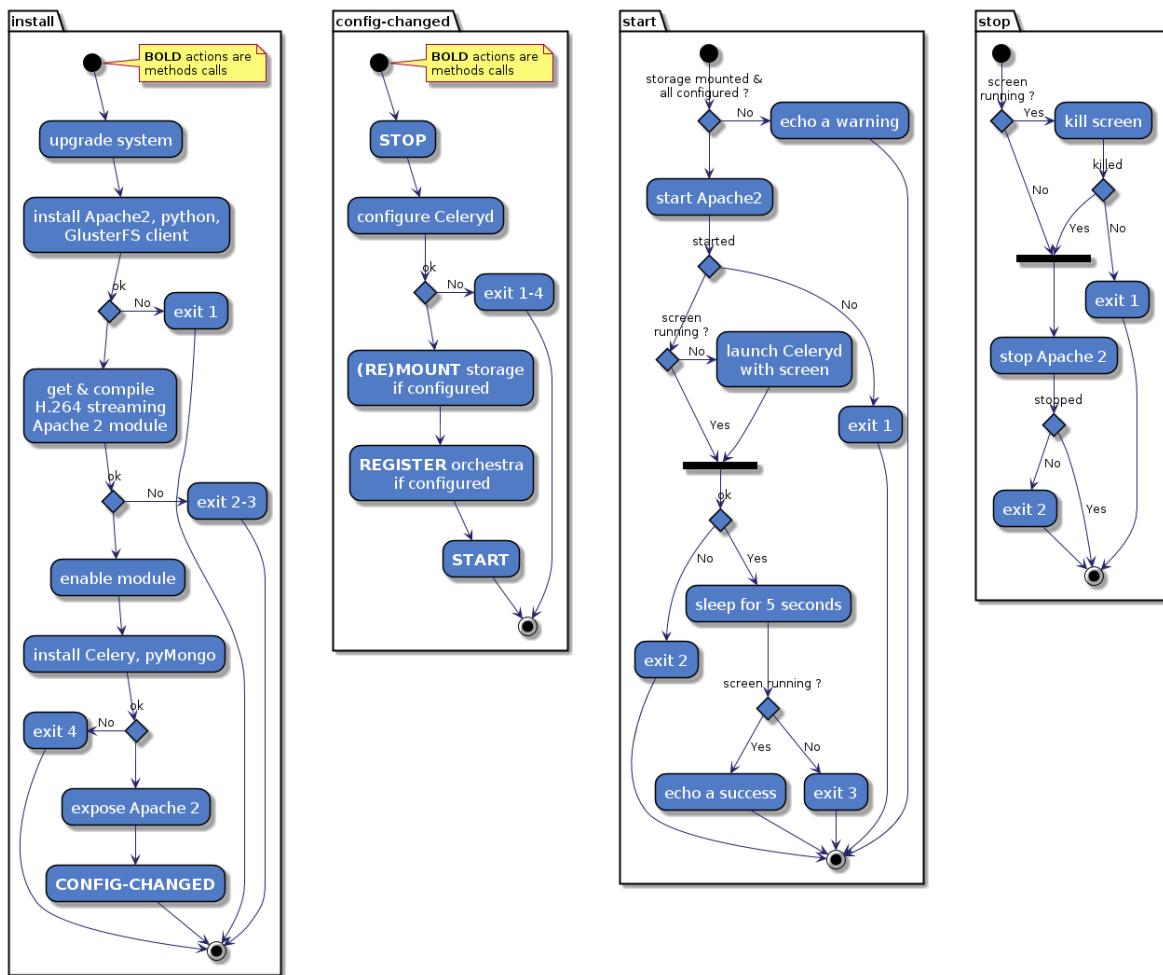


Figure 4.23: Activity diagram of Publisher unit life-cycle hooks

Charm's Relations

- Provides : (nothing)
- Requires : Storage [Mount], Publisher [Subordinate]

¹⁴ If all options are set this will override and disable publisher relation

¹⁵ If all options are set this will override and disable storage relation

Warning: The unit's daemon will not start until both conditions are fulfilled :

- A shared storage is mounted (via the storage relation or by specifying it into configuration)
- An orchestrator is registered (via the publisher relation or by specifying it into configuration)

Job State Machine

The orchestrator stores informations about the publication jobs into database in parallel to the informations that Celery also stores in. It may seem as duplicate however I choose to do as such for good reasons :

- Additional informations about the jobs can be stored.
- One can choose to replace Celery to use any other task queuing technology.
- Listing of jobs is easy to implement, no needs of Celery's inspect & filtering.

Celery's keep track of jobs and stores informations about jobs into a *backend*, the orchestrator's database (MongoDB).

” During its lifetime a task will transition through several possible states, and each state may have arbitrary metadata attached to it. When a task moves into a new state the previous state is forgotten about, but some transitions can be deducted, (e.g. a task now in the FAILED state, is implied to have been in the STARTED state at some point). ”¹⁶

So, the publication jobs stored in database has a *statistic* field that is filled with values mainly generated by the orchestrator such as *add_date*. The state machine diagram shows what is store in this field plus the values that are appended to job's state metadata.

Remark: The orchestrator RESTful API publication methods responses contains job's metadata, appended into *statistic* field.

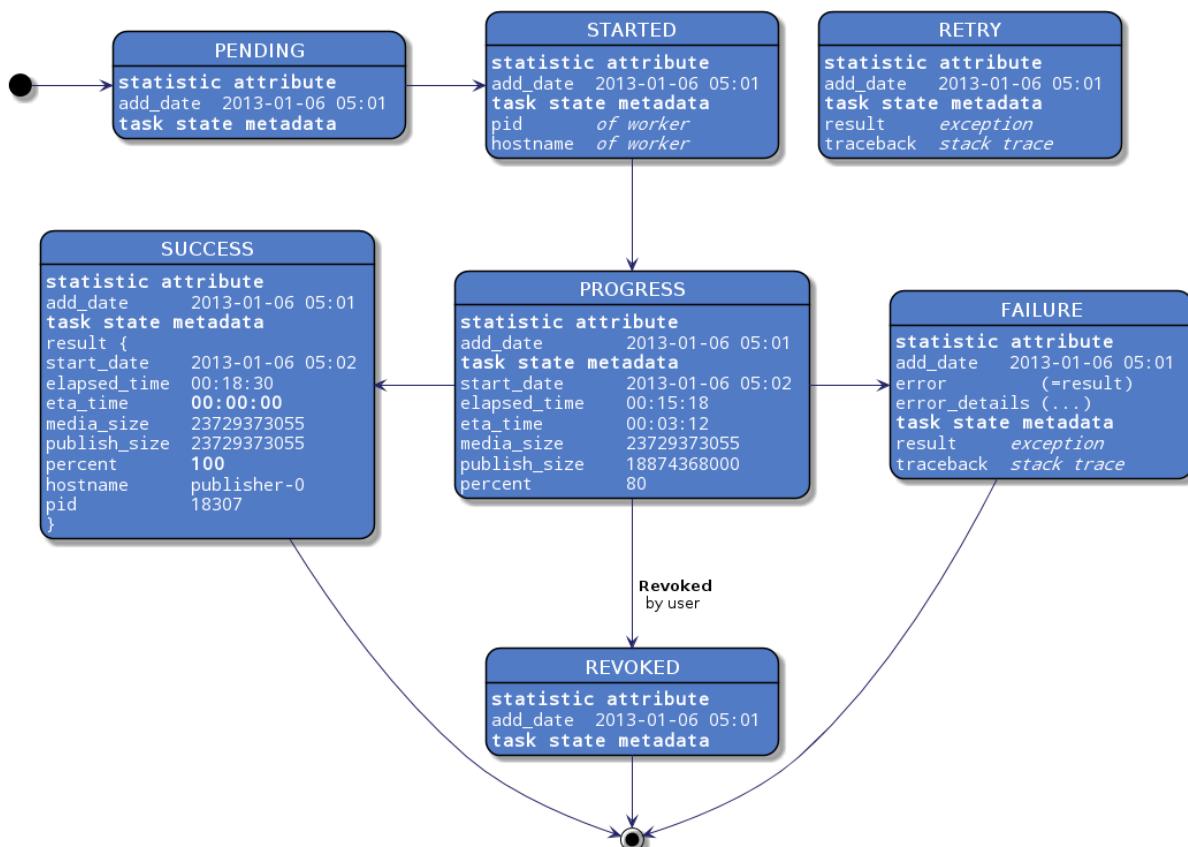


Figure 4.24: State machine of a publication job (publisher -> publish)

¹⁶ Celery Tasks Page – <http://docs.celeryproject.org/en/latest/userguide/tasks.html>

Job Sequence Diagrams

A Successful Job

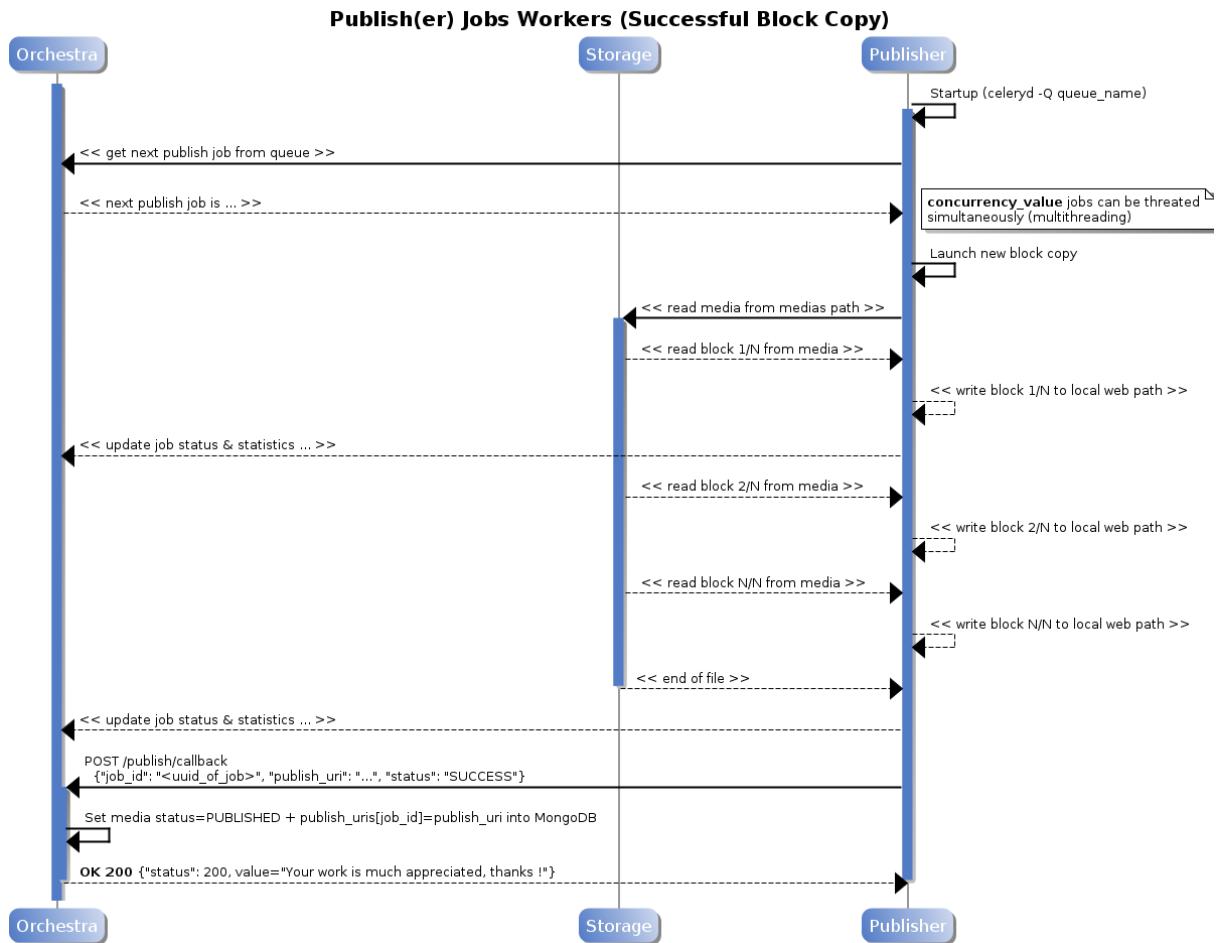


Figure 4.25: Sequence diagram of a successful publication job (publisher -> publish)

4.6.8 OSCIED-Storage : The Media Storage

See Also:

You can [browse](#) the source of the Medias Storage

OSS Tools

- [GlusterFS](#) Highly Scalable Distributed Filesystem

Introduction

This component is the application's scale-out network attached storage responsible of the medias collection.

Actual version of the charm is keep simple ([KISS](#)) for the purposes of the demonstrator. However the charm encapsulate a [GlusterFS](#) server and they are numerous advantages of using this technology, as listed in [GlusterFS](#).

Charm's Configuration

You can start the charm without specifying any configuration (default values will be used, see [OSCIED - Storage Source Code](#)) but I strongly recommend to specify your own values in production !

- **verbose** Set verbose logging
- **concurrency** Amount of tasks the worker can handle simultaneously
- **rabbit_queues** Worker connect to queues to receive jobs
- **max_upload_size** Maximum size for file uploads
- **max_execution_time** Maximum time for PHP scripts
- **max_input_time** Maximum time for HTTP post
- **mongo_connection** Orchestrator database connection ^{[17](#)}
- **rabbit_connection** Orchestrator message broker connection ^{[1](#)}
- **storage_ip** Shared storage hostname / IP address (see interface mount of NFS charm) ^{[18](#)}
- **storage_fstype** Shared storage filesystem type (e.g. NFS) ^{[2](#)}
- **storage_mountpoint** Shared storage mount point (e.g. for NFS - /srv/data) ^{[2](#)}
- **storage_options** Shared storage options (e.g. for NFS - rw,sync,no_subtree_check)

^{[17](#)} If all options are set this will override and disable publisher relation

^{[18](#)} If all options are set this will override and disable storage relation

Charm's Hooks Activity Diagrams

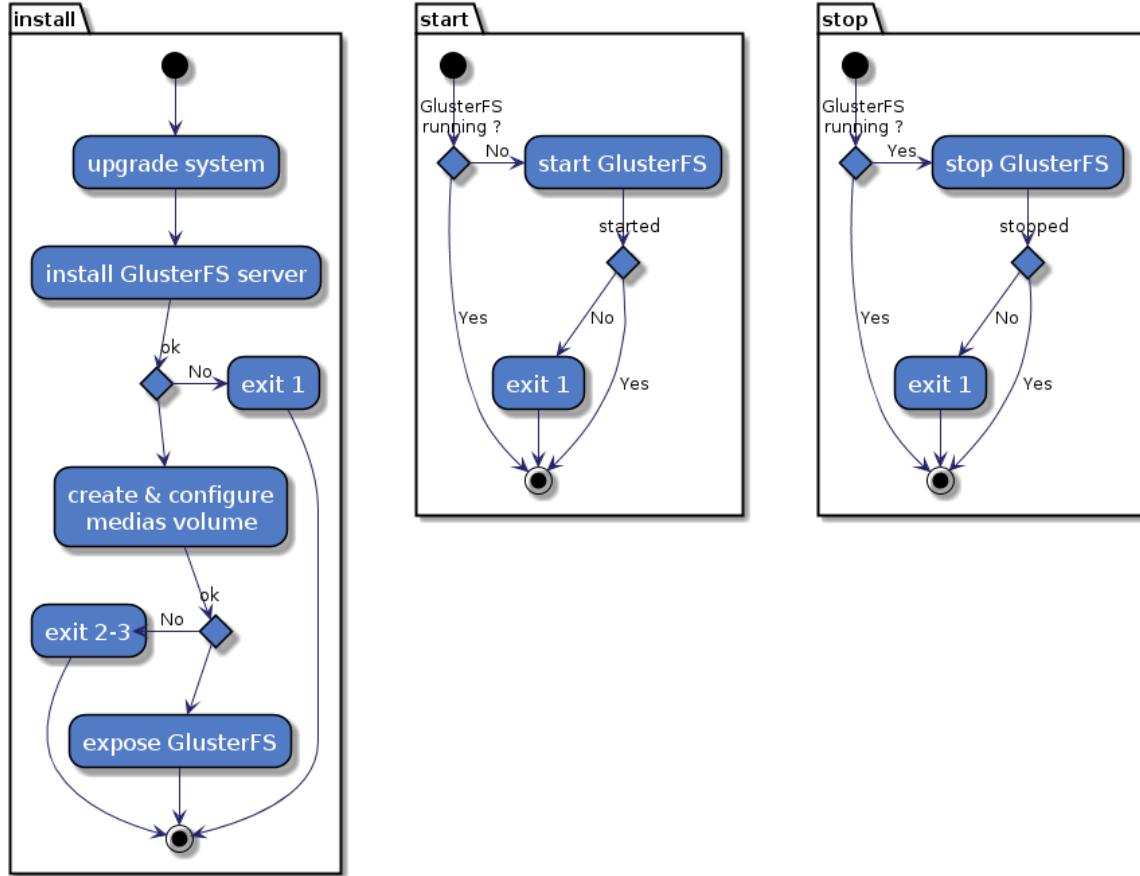


Figure 4.26: Activity diagram of Storage unit life-cycle hooks

Charm's Relations

- Provides : Storage [Mount]
- Requires : (nothing)

4.6.9 OSCIED-WebUI : The Web User Interface

See Also:

You can [browse the source of the Web User Interface](#)

OSS Tools

- Apache 2 HTTP Server from the Apache Software Foundation
- CodeIgniter Powerful PHP MVC Framework from EllisLab [Ticket 117](#)
- CSS Bootstrap Front-end Framework from Twitter [Ticket 36](#)

Introduction

This component is the user interface of the application providing an uncluttered, user-friendly web interface for using the functionalities of the application. The user's actions are mapped to orchestrator's RESTful API calls.

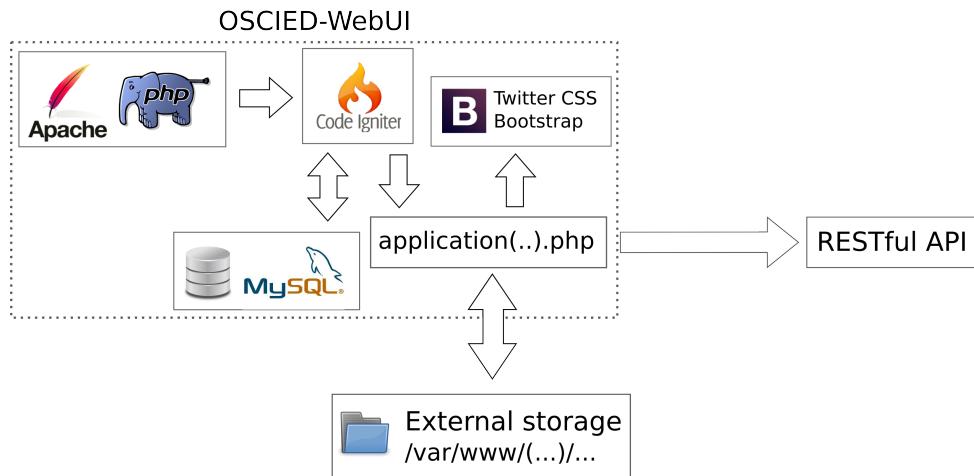


Figure 4.27: Architecture of the Web User Interface

Charm's Configuration

You can start the charm without specifying any configuration (default values will be used, see [OSCIED - WebUI Source Code](#)) but I strongly recommend to specify your own values in production !

- **verbose** Set verbose logging
- **max_upload_size** Maximum size for file uploads
- **max_execution_time** Maximum time for PHP scripts
- **max_input_time** Maximum time for HTTP post
- **mysql_my_password** Password for phpmyadmin
- **mysql_root_password** Password of MySQL root user
- **mysql_user_password** Password of MySQL webui user
- **api_url** Orchestrator REST API address ¹⁹
- **storage_ip** Shared storage hostname / IP address (see interface mount of NFS charm) ²⁰

¹⁹ If all options are set this will override and disable api relation

²⁰ If all options are set this will override and disable storage relation

- **storage fstype** Shared storage filesystem type (e.g. NFS)²
- **storage_mountpoint** Shared storage mount point (e.g. for NFS - /srv/data)²
- **storage_options** Shared storage options (e.g. for NFS - rw, sync, no_subtree_check)

Charm's Hooks Activity Diagrams

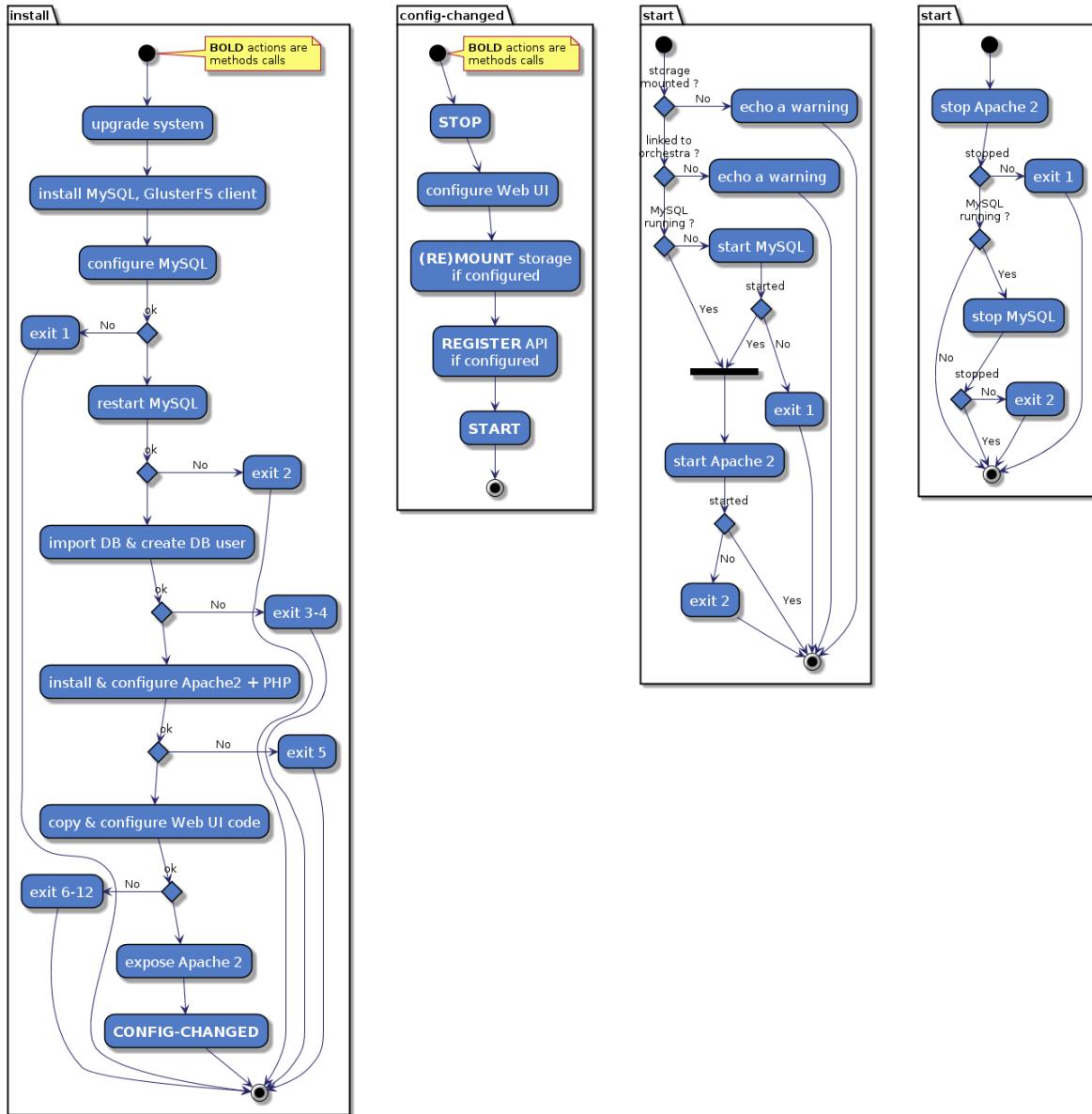


Figure 4.28: Activity diagram of WebUI unit life-cycle hooks

Charm's Relations

- Provides : Website [HTTP]
- Requires : Storage [Mount], API [Orchestra]

Warning: The unit's daemon will not start until both conditions are fulfilled :

- A shared storage is mounted (via the storage relation or by specifying it into configuration)
- An orchestrator is registered (via the api relation or by specifying it into configuration)

Users Tab

The user "Peter MacAvock" has been added.

Edit my account

| Id | First name | Last name | Email | Secret | |
|--------------------------------------|------------|-----------|---------|--------|---|
| 129ce728-cddf-4422-8b95-9c951cb3f11b | David | Fischer | d@f.com | | Edit Delete |

Edit other users

| Id | First name | Last name | Email | Secret | Admin platform | |
|--------------------------------------|------------|-----------|---------|--------|-------------------------------------|---|
| 695a8ed0-bbd6-4c29-b01c-4a0d52c04e08 | Loïc | Fischer | l@f.com | | <input type="checkbox"/> | Edit Delete |
| 911d2e61-fa20-44a8-85ff-da956a8f143a | Andrés | Revuelta | a@r.com | | <input type="checkbox"/> | Edit Delete |
| dc05cca9-9877-4148-afa7-f3fe480b45d0 | Michaël | Fischer | m@f.com | | <input checked="" type="checkbox"/> | Edit Delete |
| d2f07e46-cff0-4b54-8810-58e7243caf99 | Bram | Tullemans | b@t.com | | <input checked="" type="checkbox"/> | Edit Delete |
| 0ecbdfa6-71fc-43e7-9d6f-6aa97cb2d858 | Peter | MacAvock | p@a.com | | <input type="checkbox"/> | Edit Delete |

Add an user

| First name | Last name | Email | Secret | Admin platform |
|----------------------|----------------------|----------------------|----------------------|--------------------------|
| <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="checkbox"/> |

[Add user](#)

Figure 4.29: Adding an user with the users add form (WebUI -> Users)



Edit my account

User : secret is not safe (8+ characters, upper/lower + numbers eg. StrongP6s)

| Id | First name | Last name | Email | Secret | |
|--------------------------------------|-------------------|------------------|--------------|---------------|---|
| 129ce728-cddf-4422-8b95-9c951cb3f11b | David | Fischer | d@f.com | * | Edit Delete |

Edit other users

| Id | First name | Last name | Email | Secret | Admin platform | |
|--------------------------------------|-------------------|------------------|--------------|---------------|-------------------------------------|---|
| 695a8ed0-bbd6-4c29-b01c-4a0d52c04e08 | Loïc | Fischer | l@f.com | | <input type="checkbox"/> | Edit Delete |
| 911d2e61-fa20-44a8-85ff-da956a8f143a | Andrés | Revuelta | a@r.com | | <input type="checkbox"/> | Edit Delete |
| dc05cca9-9877-4148-afat-13fe480b45d0 | Michaël | Fischer | m@f.com | | <input checked="" type="checkbox"/> | Edit Delete |
| d2f07e46-cff0-4b54-8810-58e7243caf99 | Bram | Tullemans | b@t.com | | <input checked="" type="checkbox"/> | Edit Delete |

Add an user

User : mail is not a valid email address

| First name | Last name | Email | Secret | Admin platform |
|-------------------|------------------|--------------|---------------|-------------------------------------|
| Peter | MacAvock | p | ***** | <input checked="" type="checkbox"/> |

[Add user](#)

Figure 4.30: The API check validity of all inputs, e.g. weak secret + bad email format (WebUI -> Users)

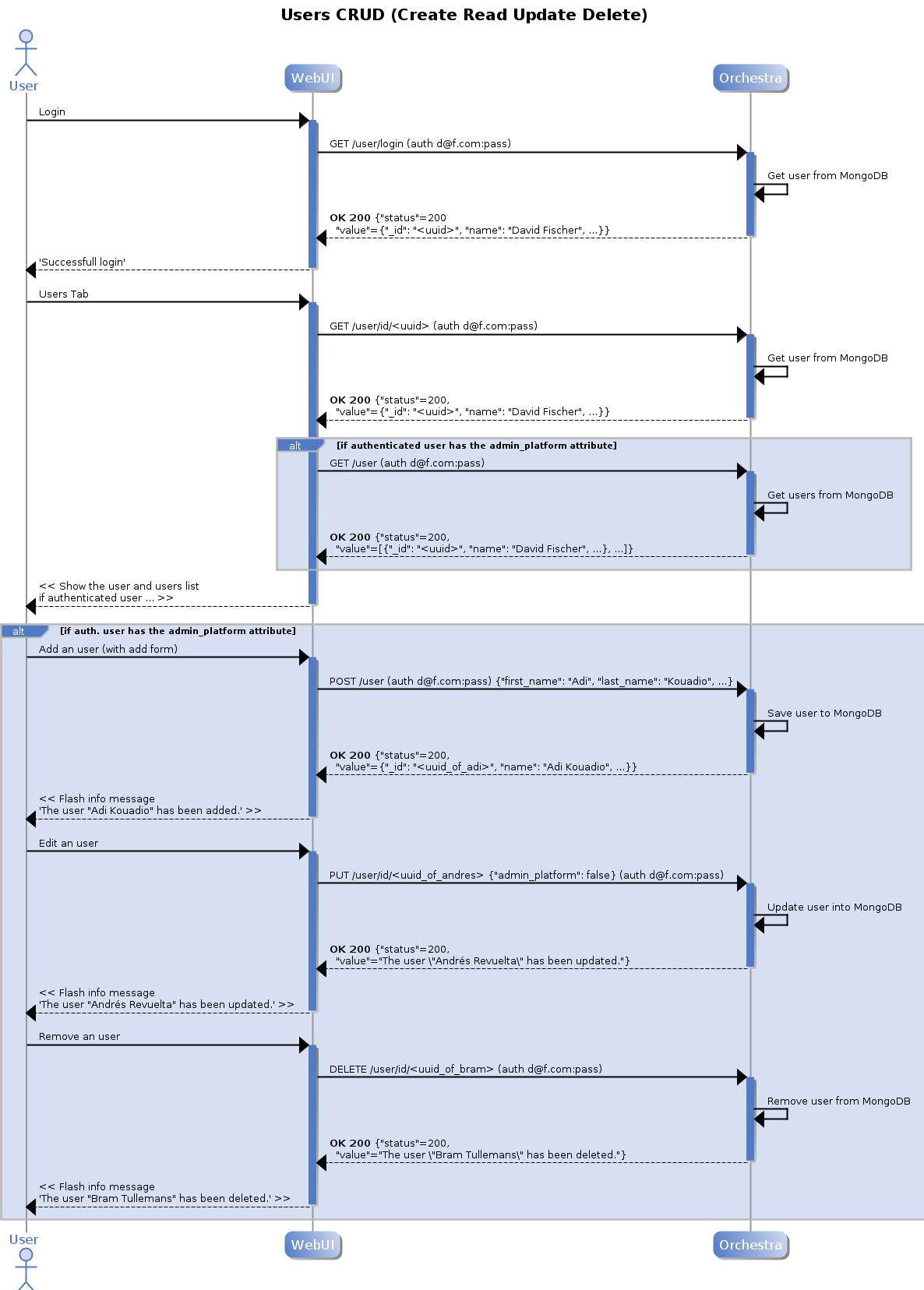


Figure 4.31: Sequence diagram of WebUI Users Tab CRUD (Create Read Update Delete)

Medias Tab

The media "Test" has been added.

Available medias

| Title | Virtual Filename | File size | Duration | Added on | Added by | Status | Delete |
|--|---------------------------------|-----------|-------------|------------------|---------------|--------|--------|
| Project London - Official Trailer (2009) | Project_London_trailer_2009.mp4 | 52.3 MB | 00:02:44.88 | 2013-02-02 14:05 | David Fischer | READY | Delete |
| Psy - Gangnam Style | Psy_gangnam_style.flv | 174.7 MB | 00:04:12.16 | 2013-02-02 14:05 | David Fischer | READY | Delete |
| Test | test.mp4 | 81.4 MB | 00:04:17.99 | 2013-02-02 14:56 | David Fischer | READY | Delete |

Add a media

Title

Virtual filename

+ Add **Cancel all**

You can drap and drop your files here

Add media

Figure 4.32: Adding a media with the medias upload form (WebUI -> Media)



Available medias

| Title | Virtual Filename | File size | Duration | Added on | Added by | Status | |
|--|---|-----------|-------------|------------------|---------------|-----------|------------------------|
| Psy - Gangnam Style 720p | Psy_gangnam_style_720p.mp4 | 174.8 MB | 00:04:12.16 | 2013-02-02 15:39 | David Fischer | PUBLISHED | Delete |
| Project London MP2 | Project_London.mpg | 24.4 MB | 00:00:01.95 | 2013-02-02 15:40 | David Fischer | READY | Delete |
| s | s.mp4 | 0 Bytes | | 2013-02-02 15:40 | David Fischer | DELETED | |
| Project London - Official Trailer (2009) | Project_London_trailer_2009.mp4 | 52.3 MB | 00:02:44.88 | 2013-02-02 15:38 | David Fischer | PUBLISHED | Delete |
| Psy - Gangnam Style | Psy_gangnam_style.flv | 174.7 MB | 00:04:12.16 | 2013-02-02 15:38 | David Fischer | PUBLISHED | Delete |
| Gaga | gaga.mp2 | 0 Bytes | | 2013-02-02 15:46 | David Fischer | PENDING | |
| PSY MP2 | PSY.mp2 | 0 Bytes | | 2013-02-02 15:47 | David Fischer | DELETED | |
| Project London MP2 Bis | Project_London.mpeg | 24.4 MB | 00:00:01.95 | 2013-02-02 15:48 | David Fischer | READY | Delete |

Add a media

Title

Virtual filename

[+ Add](#) [Cancel all](#)

You can drag and drop your files here

[Add media](#)

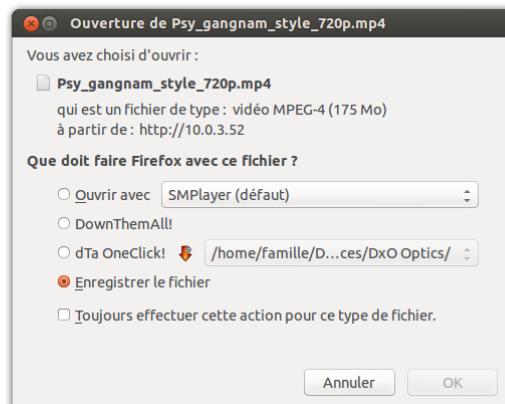


Figure 4.33: Downloading a media by clicking on the hyperlink (WebUI -> Media)

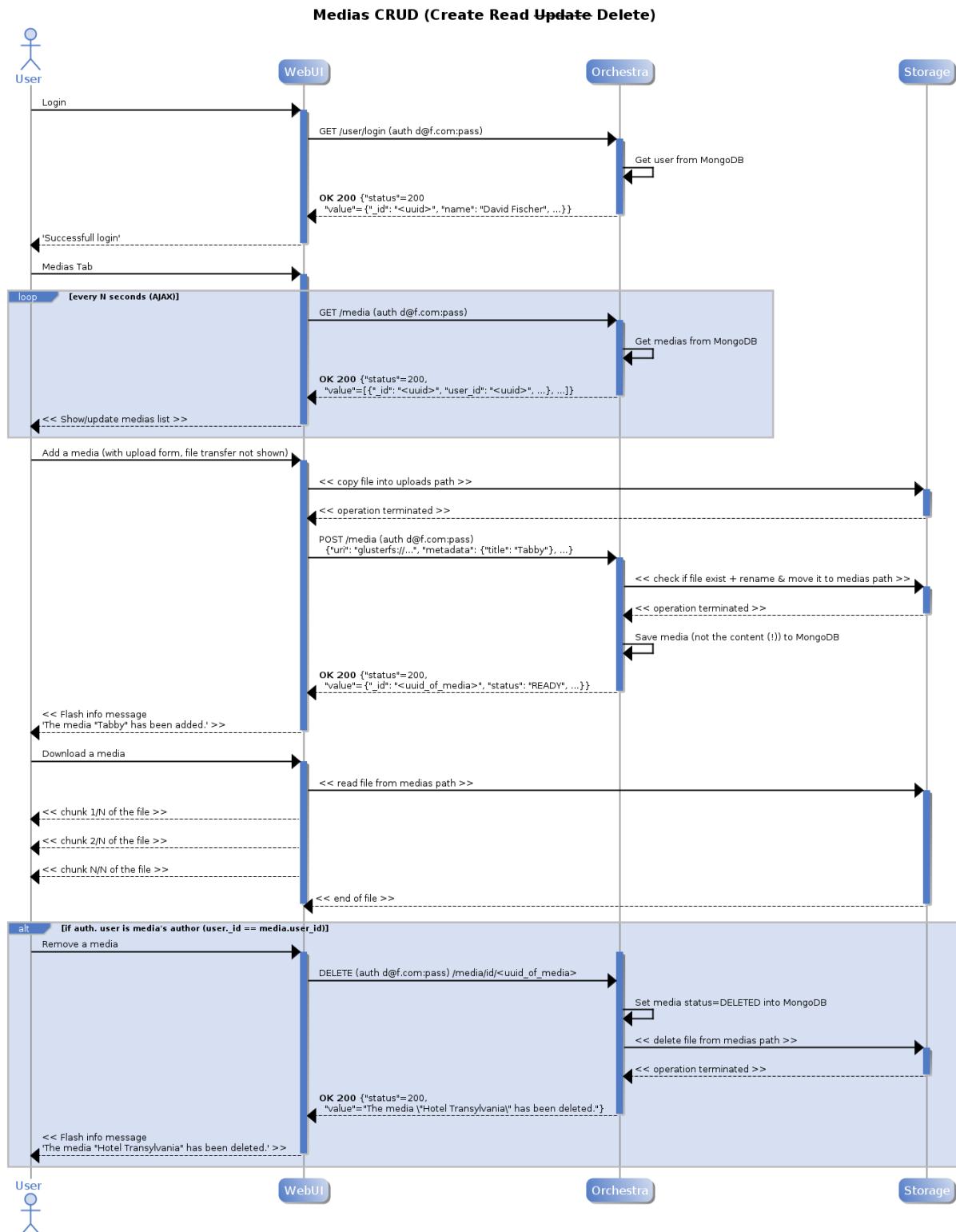


Figure 4.34: Sequence diagram WebUI Medias Tab CRUD (Create Read (update) Delete)

Transform Profiles Tab

The screenshot shows the OSCIED web interface with a dark header bar. On the left is the logo 'OSCIED'. To its right are links for 'Home', 'Links', 'Contact Us', and several icons for 'Users', 'Medias', 'Profiles' (which is highlighted in blue), 'Transform', and 'Publisher'. On the far right, it says 'Logged as David Fischer' with a dropdown arrow.

Available transform profiles

| Title | Description | Encoder string | |
|-----------|--|---|-------------------------|
| File Copy | A simple block file copy | copy | <button>Delete</button> |
| To MP4 | FFmpeg container -> MP4 | -acodec copy -vcodec copy -f mp4 | <button>Delete</button> |
| To MP2 | Convert video track to MPEG-2 format, copy audio track | -acodec copy -vcodec mpeg2video -f mpeg2video | <button>Delete</button> |
| To 720p | Force aspect to 16:9 and resolution to 720p | -aspect 16:9 -s 1280x720 -swsflags lanczos | <button>Delete</button> |

Add a transform profile

| Title | Description | Encoder string |
|----------------------|----------------------|----------------------|
| <input type="text"/> | <input type="text"/> | <input type="text"/> |

Add profile

Figure 4.35: List of available profiles that transform jobs can pick from (WebUI -> Profile)

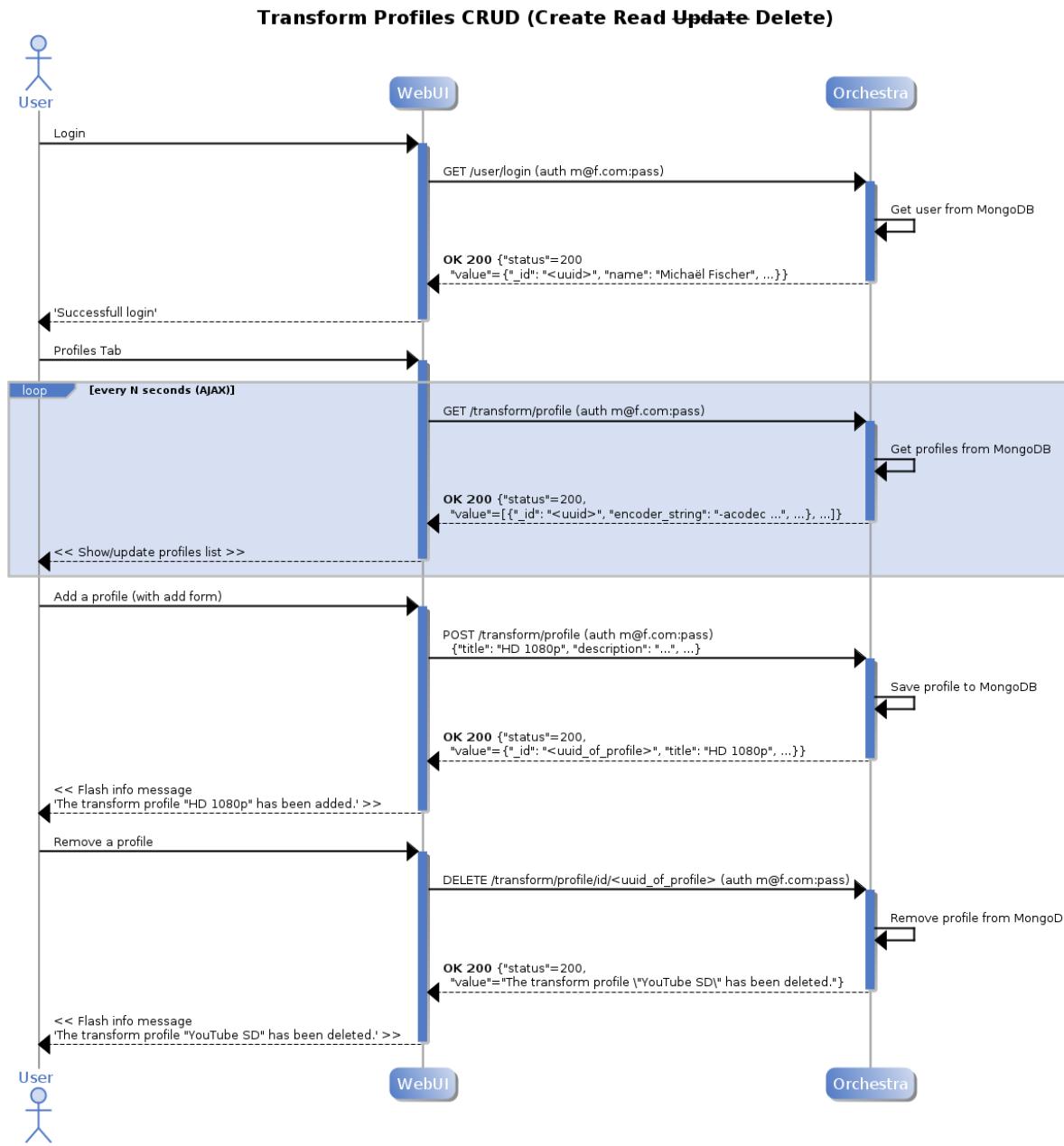


Figure 4.36: Sequence diagram of WebUI Transform Profiles Tab CRUD (Create Read (update) Delete)

Transform Jobs Tab



Transform jobs

| Input media | Output media | Profile | Added by | Added on | Started on | Elapsed | Progress | Error | Status |
|---------------------------------|----------------------------|---------|---------------|------------------|------------------|----------------------|---|--|----------|
| Psy_gangnam_style.flv | Psy_gangnam_style_720p.mp4 | To MP4 | David Fischer | 2013-02-02 15:10 | 2013-02-02 15:39 | 00:00:05 00:00:00 | <div style="width: 100%; background-color: #2e6b2e; height: 10px;"></div> | | SUCCESS |
| Project_London_trailer_2009.mp4 | Project_London.mpg | To MP2 | David Fischer | 2013-02-02 15:10 | 2013-02-02 15:40 | 00:00:21 00:00:00 | <div style="width: 100%; background-color: #2e6b2e; height: 10px;"></div> | | SUCCESS |
| Psy_gangnam_style_720p.mp4 | s.mp4 | To 720p | David Fischer | 2013-02-02 15:10 | | 00:00:00 00:00:00 | <div style="width: 0%; background-color: #cccccc; height: 10px;"></div> | Unable to parse FFmpeg output, encoding probably failed1 | FAILURE |
| Psy_gangnam_style_720p.mp4 | gaga.mp2 | To MP2 | David Fischer | 2013-02-02 15:10 | | 00:00:00 00:00:00 | <div style="width: 0%; background-color: #cccccc; height: 10px;"></div> | None1 | PENDING |
| Psy_gangnam_style_720p.mp4 | PSY.mp2 | To MP2 | David Fischer | 2013-02-02 15:10 | 2013-02-02 15:47 | 00:00:05 00:00:52 | <div style="width: 10%; background-color: #007bff; height: 10px;"></div> | terminated1 | REVOKED |
| Project_London_trailer_2009.mp4 | Project_London.mpeg | To MP2 | David Fischer | 2013-02-02 15:10 | 2013-02-02 15:48 | 00:00:16 00:00:08 | <div style="width: 50%; background-color: #007bff; height: 10px;"></div> | | PROGRESS |

Launch a transform job

| Input Media | Profile | Virtual Filename | Media Title | Queue |
|------------------------------|-----------|------------------|-------------|-------------------|
| Psy - Gangnam Style 720p - P | File Copy | | | transform_private |

Launch job

Figure 4.37: List of transform jobs (encoding) with various status (WebUI -> Transform)

Note: As you can see, here the components needs NTP time synchronization !

| | | | | | | | | | | | | | |
|------------------|---------------|------------------|--|----------------------|---|-----------------|----------------|---|--|------------------|-------------|-----|--|
| To 720p | David Fischer | 2013-02-02 15:10 | | 00:00:00 00:00:00 | <div style="width: 0%; background-color: #cccccc; height: 10px;"></div> | Unable to parse | FAILURE | <pre>ERROR\nUnable to parse FFmpeg output, encoding probably failed\n\nOUTPUT\n\\nffmpeg version 0.10.6-0.10.6-0ubuntu0on1 Copyright (c) 2000-2012 the FFmpeg\ndevelopers\\n built on Nov 12 2012 12:53:40 with gcc 4.7.2\\n configuration:\n-arch=amd64 -enable-pthreads -enable-runtime-cpudetect -extra-\nversion='0.10.6-0ubuntu0on1' -libdir=/usr/lib/x86_64-linux-gnu -disable-stripping\n-prefix=/usr -enable-bzlib -enable-libdc1394 -enable-libfreetype -enable-frei0r\n-enable-gnutls -enable-libgsm -enable-libmp3lame -enable-librtmp -enable-\nlibopenvc -enable-libopenjpeg -enable-libpulse -enable-libschoedinger -enable-\nlbspeex -enable-litheora -enable-vaapi -enable-vdpau -enable-libvorbis -enable-\nlibvpx -enable-zlib -enable-gpl -enable-postproc -enable-libcdio -enable-x11grab\n-enable-libz64 -shlibdir=/usr/lib/x86_64-linux-gnu -enable-shared -disable-static\nlibavutil 51.35.100 / 51.35.100\\n libavcodec 53.61.100 / 53.61.100\\n\nlibavformat 53.32.100 / 53.32.100\\n libavdevice 53.4.100 / 53.4.100\\n\nlibavfilter 2.61.100 / 2.61.100\\n libpostproc 52.0.100 / 52.0.100\\n\ninput #0,\nmov,mp4,m4a,3gp,3g2,mj2, from '/mnt/storage/medias/75fa4847-6ab3-4c90-\nafef62b4c70a1bad/35cef9ac-11a7-4358-8674-32d4f2d00d0f': Metadata:\\n\nmajor_brand : isom\\n minor_version : 512\\n compatible_brands:\nisomiso2avc1mp4\\n creation_time : 2012-07-14 07:48:50\\n encoder :\nLavf53.32.100\\n Duration: 00:04:12.16, start: 0.000000, bitrate: 5813 kb/s\\n\nStream #0:0(und): Video: h264 (High) (avc1/0x31637661), yuv420p, 1920x1080, 5617 kb/s,\n23.98 fps, 23.98 tbr, 48k tbn, 47.95 tbc\\n\nMetadata:\\n creation_time : 2012-07-14\n07:48:50\\n handler_name :VideoHandler\\n\nStream #0:1(und): Audio: aac (mp4a/\n0x6134706D), 44100 Hz, stereo, s16, 192 kb/s\\n\nMetadata:\\n creation_time :\n2012-07-14 07:48:50\\n handler_name :\\nUnrecognized option 'swsflags'\\nFailed to\nset value ' lanczos' for option 'swsflags'\\n</pre> | | | | | |
| | | | | | | | | <table border="1"> <tr> <td>Virtual Filename</td> <td>Media Title</td> </tr> <tr> <td>[]</td> <td></td> </tr> </table> | | Virtual Filename | Media Title | [] | |
| Virtual Filename | Media Title | | | | | | | | | | | | |
| [] | | | | | | | | | | | | | |

Figure 4.38: Details of the erroneous transform job (WebUI -> Transform)

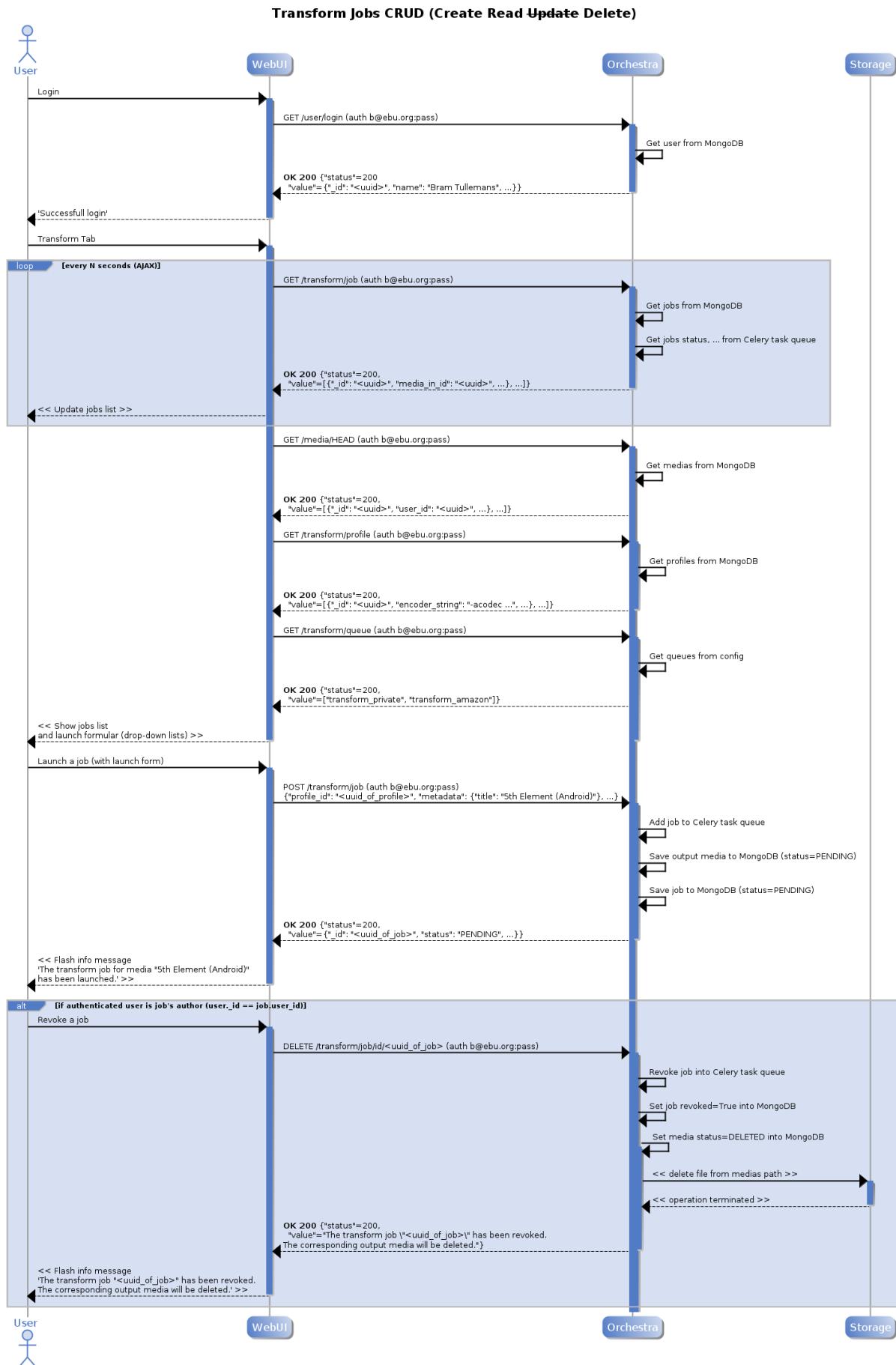


Figure 4.39: Sequence diagram of WebUI Transform Jobs Tab CRUD (Create Read (update) Delete)
4.6. Application Layer

Publish Jobs Tab



Publish(er) jobs

| Media | Added by | Added on | Started on | Elapsed | Progress | Error | Status | |
|---|---------------|------------------|------------------|----------------------|---|-------|---------|------------------------|
| Project_London_trailer_2009.mp4 | David Fischer | 2013-02-02 15:41 | 2013-02-02 15:41 | 00:00:00 00:00:00 | <div style="width: 100%; background-color: #2e6b2e; height: 10px;"></div> | | SUCCESS | |
| Psy_gangnam_style_720p.mp4 | David Fischer | 2013-02-02 15:41 | 2013-02-02 15:41 | 00:00:02 00:00:00 | <div style="width: 100%; background-color: #2e6b2e; height: 10px;"></div> | | SUCCESS | |
| Psy_gangnam_style.flv | David Fischer | 2013-02-02 15:41 | | 00:00:00 00:00:00 | <div style="width: 0%; background-color: #d9e1f2; height: 10px;"></div> | None1 | PENDING | Revoke |
| Psy_gangnam_style.flv | David Fischer | 2013-02-02 15:41 | 2013-02-02 15:42 | 00:00:02 00:00:00 | <div style="width: 100%; background-color: #2e6b2e; height: 10px;"></div> | | SUCCESS | |

Launch a publish job

Cannot launch the job, input media status is PUBLISHED.

| Media | Queue |
|--|-------------------|
| Psy - Gangnam Style - Psy_g... | publisher_private |

[Launch job](#)

Figure 4.40: List of publish jobs (publication) with API input validity error shown (WebUI -> Publisher)

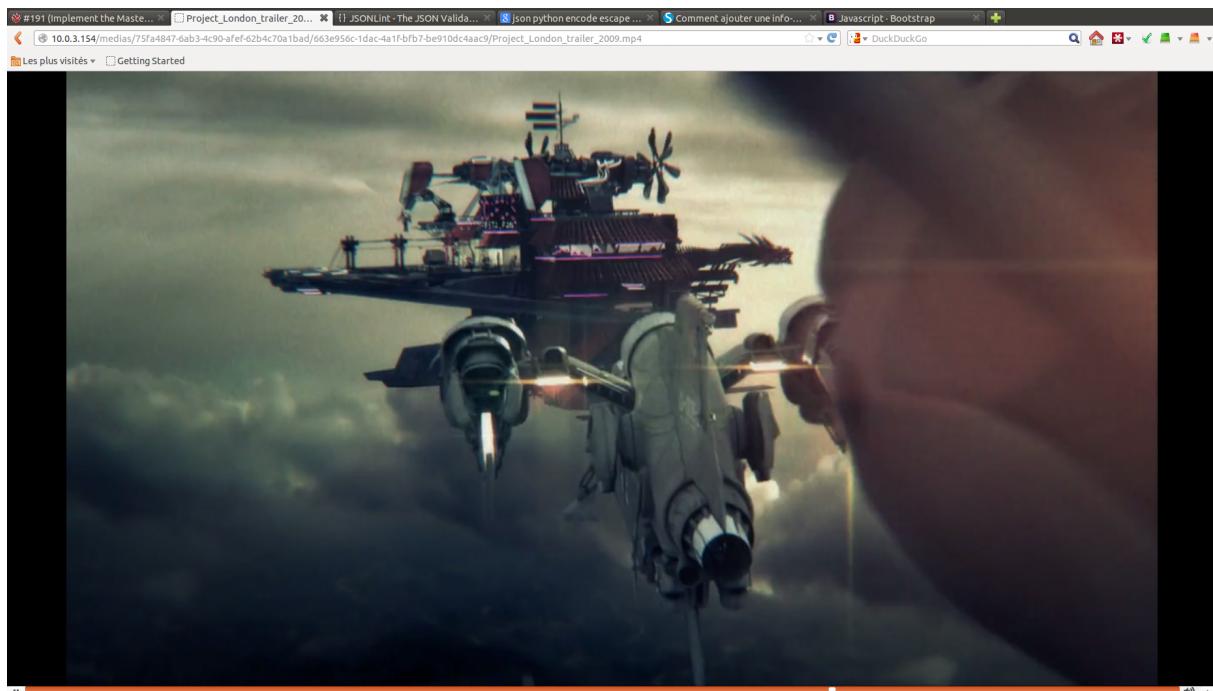


Figure 4.41: Play-out of a published media thanks to H.264 Streaming mod (WebUI -> Publisher)

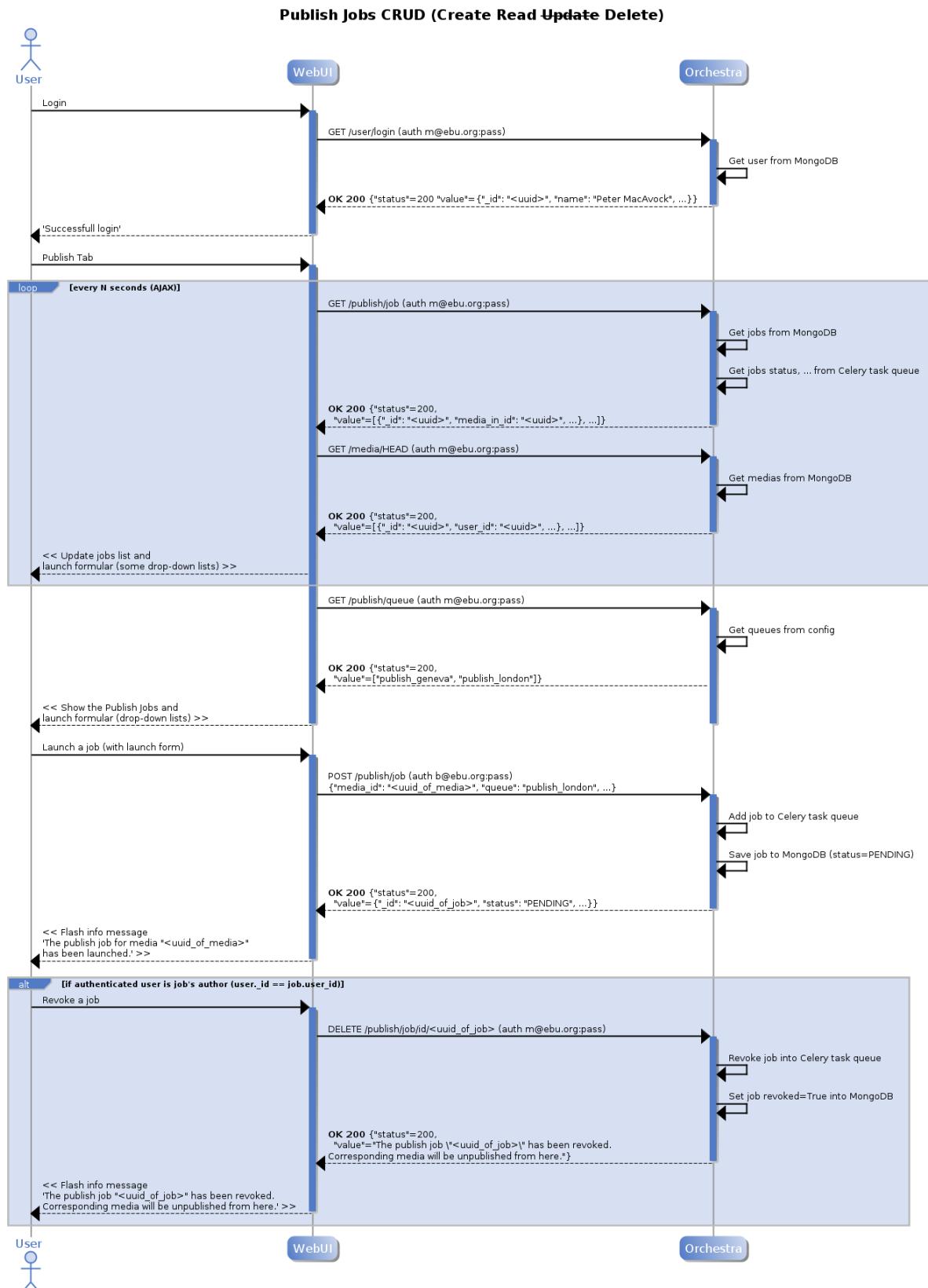


Figure 4.42: Sequence diagram WebUI Publish(er) Jobs Tab CRUD (Create Read (update) Delete)

4.7 Demonstrator FAQ

Note: This documentation is intended to be read by anyone with sufficient GNU/Linux / Ubuntu skills to understand what happens when executing the example of code snippets !

4.7.1 How to get a copy of the project ?

At time of writing this documentation (26 January 2013) the project is hosted at home and keep safe on my personal NAS. I installed a Subversion server, an Apache 2 module for SVN and a ticket management system called TRAC to host my projects.

So, to get a development copy of the project, you only need to open a terminal and run the following:

```
>>> ~$ cd
>>> ~$ sudo apt-get install subversion
>>> ~$ svn co https://claire-et-david.dyndns.org/prog/OSCIED
```

Then, I invite you to open a terminal and run the nice old-fashioned project's main menu and select **install**:

```
>>> ~$ cd $HOME/OSCIED/scripts/
>>> ~/OSCIED/scripts$ sh menu.sh
```

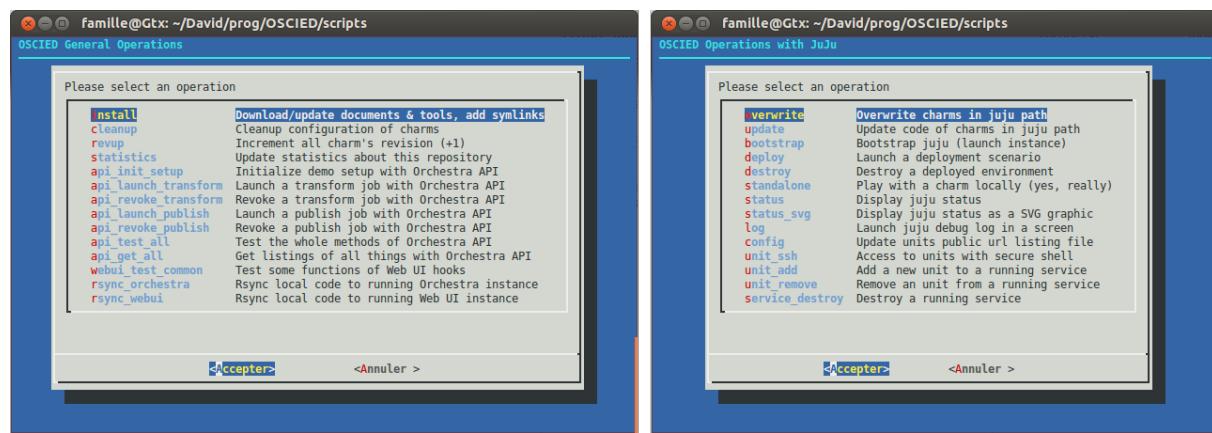


Figure 4.43: On the left : OSCIED Main Menu // On the right : OSCIED JuJu Menu

This will install required packages, update some reference documents, populate tools / paths with Open-Source tools source-code (e.g. Celery's one), ...

4.7.2 What are the functionalities of project's scripts ?

I prefer to explain you the functionalities offered by the script by describing some of the typical uses cases.

So, please browse this FAQ to find the right answer to your case. If you don't find the right answer, don't hesitate to contact me *About* !

4.7.3 How to configure the demonstrator ?

It is important to understand that are four main layers involved in this project :

1. The host computing resources :

- 1.1) Any (desktop) computer running [Ubuntu](#) ;
- 1.2) A bare-metal set of servers running [Ubuntu Quantal Server](#) ;
- 1.3) A bare-metal set of servers running [Ubuntu Quantal Server](#) with [MAAS](#) ;
- 1.4) ... The cloud providers computing resources running an [IaaS](#) ;
2. The host cloud or virtualization technology :
 - 2.1) A private cloud running on top of (1.1, 1.2, 1.3) eg. [OpenStack](#) ;
 - 2.2) A public cloud running on top of (1.4), eg. [Amazon AWS](#), [HP Cloud](#) ;
 - 2.3) An OS Level Virtualization technology running on top of (1.1, 1.2, 1.3), eg. [LXC](#) ;
 - 2.4) ... or even no virtualization at all (see [Deployment Scenarios](#)) ;
3. The clouds orchestrator called [JuJu](#) ;
4. The application itself called [OSCIED](#) ;

And all of them must be configured according to your needs.

So, here will be only introduced what to configure for 2 out of the 4 layers.

Note: In a future release [JuJu](#) will be embedded into the *Orchestra* charm to add auto-scaling features and improve easiness of deployment !

See Also:

Please see [Ticket 131](#) for quick start with [Amazon AWS](#) and [Ticket 40](#) for further details about [JuJu](#) and [MAAS](#).

Configuration files of JuJu

Please read [Deployment Scenarios](#) to gather the configuration files corresponding to the scenario you want to deploy. Then save your configuration files into config/juju/ path of the demonstrator :

```
>>> ~$ cd $HOME/OSCIED/config/juju/
>>> ~/OSCIED/config/juju$ ls -1
environments.yaml
id_rsa
id_rsa.pub
orchestra.yaml
publisher.yaml
storage.yaml
transform.yaml
webui.yaml
```

Note: You can generate the certificate used to connect to [JuJu](#)'s remote unit with ssh-keygen -t rsa -f id_rsa.

Configuration files of the demo

When launched, the demonstrator database is empty : The application should be initialized !

A utility script is available in order to allow you specifying the configuration into simple CSV files. You only need to fill these files with required content before launching the demonstrator itself.

Here are some example of API configuration files. Copy, paste, update and save them into config/api/ path of the demonstrator :

```
>>> ~$ cd $HOME/OSCIED/config/
>>> ~/OSCIED/config$ ls juju -1 | grep 'yaml' | grep -v 'environments.yaml'
orchestra.yaml
publisher.yaml
```

```
storage.yaml
transform.yaml
webui.yaml
>>> ~/OSCIED/config$ ls api -1
medias.csv
tprofiles.csv
users.csv
```

api/medias.csv

```
1 Project London - Official Trailer [2009].mp4;Project_London_trailer_2009.mp4;Project London - Official Trailer
2 Psy - Gangnam Style.flv;Psy_gangnam_style.flv;Psy - Gangnam Style
3 big_buck_bunny_720p_h264.mov;big_buck_bunny_720p_h264.mov;Big Buck Bunny 720p H.264 MOV
4 big_buck_bunny_720p_h264.mp4;big_buck_bunny_720p_h264.mp4;Big Buck Bunny 720p H.264 MP4
5 tears_of_steel_720p.mkv;tears_of_steel_720p.mkv;Tears of Steel 720p MKV
6 tears_of_steel_720p.mov;tears_of_steel_720p.mov;Tears of Steel 720p MOV
7 tears_of_steel_720p.mp4;tears_of_steel_720p.mp4;Tears of Steel 720p MP4
8 Agronomie.flv;hepia_Agronomie.flv;hepia - Agronomie
9 Architecture.flv;hepia_Architecture.flv;hepia - Architecture
10 Architecture du paysage.flv;hepia_Architecture_du_paysage.flv;hepia - Architecture du paysage
11 Génie civil.flv;hepia_Genie_civil.flv;hepia - Génie civil
12 Gestion de la nature.flv;hepia_Gestion_de_la_nature.flv;hepia - Gestion de la nature
13 Informatique.mp4;hepia_Informatique.mp4;hepia - Informatique
14 ITI.mov;hepia_ITI.mov;hepia - ITI
15 microtechniques.flv;hepia_Microtechniques.flv;hepia - Microtechniques
16 Télécommunications.avi;hepia_Telecommunications.avi;hepia - Télécommunications
17 MusiqueChinoise.mp4;MusiqueChinoise.mp4;Sounds of Shanghai (2013) by Andrés Revuelta
18 deploy_lxc_bug-cannonical_path.mp4;deploy_lxc_bug-cannonical_path.mp4;JuJu deploy LXC bug 'Cannonical path'
19 deploy_lxc_bug-memtest86+.mp4;deploy_lxc_bug-memtest86+.mp4;JuJu deploy LXC bug 'memtest86'
20 init_and_jobs.mp4;init_and_jobs.mp4;David deploying OSCIED r1110 -> LXC
```

api/medias-plus.csv (not used)

```
1 big_buck_bunny_720p_h264.mp4;big_buck_bunny_720p_h264.mp4;Big Buck Bunny 720p H.264 MP4
2 tears_of_steel_720p.mkv;tears_of_steel_720p.mkv;Tears of Steel 720p MKV
3 tears_of_steel_720p.mov;tears_of_steel_720p.mov;Tears of Steel 720p MOV
4 tears_of_steel_720p.mp4;tears_of_steel_720p.mp4;Tears of Steel 720p MP4
```

api/tprofiles.csv

```
1 File Copy;A simple block file copy;copy
2 To MP4;Change container -> MP4;-acodec copy -vcodec copy -f mp4
3 To MP2 (buggy);Convert video track to MPEG-2 format, copy audio track;-acodec copy -vcodec mpeg2video -sameq -
4 Force 720p MP4;Force aspect to 16/9 and resolution to 720p;-aspect 16:9 -s 1280x720 -sameq -f mp4
5 To 480p MP4;Resize (only if height > 480) to 480p -> MP4;-strict experimental -vf 'scale=trunc(oh*a/2)*2:min(480,oh)*2:ih*2' -f mp4
6 To H.264 MP4;Transcode video track to H.264 -> MP4;-acodec copy -vcodec libx264 -crf 10 -f mp4
7 Deinterlace MP4;Deinterlace video track -> MP4;-strict experimental -vf 'yadif=0.-1:0, scale=trunc(iw/2)*2:trunc(ih/2)*2'
8 Speed-up 2x MP4;Speed-up video track by a factor of 2, remove audio -> MP4;-strict experimental -vf 'setpts=0.5*PTS'
9 Negate MP4;Negate video track -> MP4;-strict experimental -vf 'negate' -sameq -f mp4
```

api/users.csv

```
1 David;Fischer;d@f.com;detgdm3D1;true
2 Loïc;Fischer;l@f.com;detgdm3D3;false
3 Andrés;Revuelta;a@r.com;detgdm3D4;false
4 Michaël;Fischer;m@f.com;detgdm3D2;false
5 Bram;Tullemans;b@t.com;detgdm3D5;true
6 Nabil;Abdennadher;n@a.com;detgdm3D6;false
7 Thomas;Kernen;t@k.com;detgdm3D7;false
8 Peter;Mac Avock;p@ma.com;detgdm3D8;true
9 Cédric;De Carvalho;c@c.com;detgdm3D9;false
10 Daniel;Maechler;d@m.com;detgdm3D10;false
```

Finally, the input media files should be put into medias/Test/ path of the demonstrator. The filename must match the first column of *medias.csv* (eg. *Project London - Official Trailer [2009].mp4*).

4.7.4 How to launch the demonstrator ?

Note: Do not forget to update the configuration **before** any deployment !

In order to deploy the application (a scripted scenario), do the following:

```
>>> ~$ cd $HOME/OSCIED/scripts/
>>> ~/OSCIED/scripts$ sh juju-menu.sh
```

- Select **overwrite** to copy application's charms to the deployment path of **JuJu** ;
- Select **bootstrap** to bootstrap the environment (launch the **JuJu**'s unit) ;
- Select **status** to ensure that the service is running ;
- (optional) Open a new tab to give an eye to the debug log of the deployment:

```
>>> ~$ juju debug-log
```

- Select **deploy** then launch one of the deployment scenarios ;
- Answer to questions of the scenarios, most of the time you only need to answer y for yes ;

During the deployment, you may want to check the status, so:

- (optional) See what happens in the debug log (enter the matrix ;-)) ;
- Select **status** or even **status_svg** to get latest deployment status ;

If the deployment is unsuccessful, you can recover the situation by more than two different ways:

- Destroying the whole environment with **destroy** ;
- Destroying the faulty service (e.g. *oscied-webui*) with **service_destroy** ;
- Fix the charm's source code and upgrade charms ... NOT YET IMPLEMENTED ;

Then, you need to fix the charm's source code (see *How to update the code ?*) ... and retry your deployment:

- ... Do all steps if you destroyed the whole environment ;
- ... Only **update**, **deploy** and choose to launch only the faulty service again ;

Note: You can check status faster by filtering the output of **JuJu**'s status:

```
>>> ~$ juju status | grep error
(nothing)
```

When all services are hopefully up and running you can setup the Orchestrator by doing the following:

- Select **config** to update generated configuration files used by scripts ;

Warning: The actual Orchestrator can't understand that the storage's public/private IP are linked to the same host. This is a known limitation that will be fixed in a future release.

It means that the private IP of the storage unit is hardcoded in `common.sh`. A dedicated utility script help you to update it easily.

```
>>> ~$ cd $HOME/OSCIED/scripts/
>>> ~/OSCIED/scripts$ sh autodetect-storage-ip.sh
```

This utility will connect to the orchestrator, grep the configuration file `/var/lib/.../charm/config.json` and extract `storage_ip` value. Last but not least, `common.sh` is then updated with auto-detected value.

- Start another script ;-)

```
>>> ~$ cd $HOME/OSCIED/scripts/
>>> ~/OSCIED/scripts$ sh menu.sh
```

- Select **api_init_setup** this will (most of the time by calling the Orchestra's RESTful API) :
 - Flush Orchestrator's database ;
 - Add the users of config/api/users.csv ;
 - **Send and add the medias of config/api/medias.csv :**
 - * Send the media file from medias/Test/ to storage's unit home path ²¹ ;
 - * Copy the media from storage's home path to storage's upload path ²² ;
 - * Add the media ;
 - Add the transform profiles of config/api/tprofiles.csv ;

Voila, your demonstrator is ready !

4.7.5 How to scale up/down a service of the demonstrator ?

Note: Do not forget to update the configuration **before** any deployment !

In order to add/remove an unit to/from a service do the following:

```
>>> ~$ cd ${HOME}/OSCIED/scripts/
>>> ~/OSCIED/scripts$ sh juju-menu.sh
```

- Select **unit_add** to add an unit to any service able to scale up/down ;
- Select **unit_remove** to remove an unit from any service able to scale up/down ;

Warning: At time of writing this report, only transform and publisher services can scale up/down !

Note: In some of the deployment scenarios, you may need to bypass this helper script to directly uses juju in order to specify the environment.

4.7.6 How to stop the demonstrator ?

Note: Do not forget to update the configuration **before** any deployment !

In order to stop the application do the following:

```
>>> ~$ cd ${HOME}/OSCIED/scripts/
>>> ~/OSCIED/scripts$ sh juju-menu.sh
```

- Select **destroy** to destroy the whole environment (files in *storage* charm are **lost**) ;

4.7.7 How to update the code ?

- Modify the source-code of charms (e.g. for *oscied-orchestra*) :

```
>>> ~$ cd ${HOME}/OSCIED/components/orchestra/charm/
>>> ~/OSCIED/components/orchestra/charm/$ nano orchestra.py
>>> ~/OSCIED/components/orchestra/charm/$ nano hooks_lib/common.sh.lu-dep
>>> ~/OSCIED/components/orchestra/charm/$ lu-importUtils . no
>>> ~/OSCIED/components/orchestra/charm/$ echo $((($cat revision)+1)) > revision
```

²¹ I known that is not realistic, HTTP media inject is currently not implemented.

²² It is done this way to avoid sending medias files every time the Orchestrator is (re)initialized !

Note: If you'll have a lot of components to update, you may find useful to skip the revision's increment step and do the following:

```
>>> ~$ cd $HOME/OSCIED/scripts/
>>> ~/OSCIED/scripts/$ sh menu.sh
```

- Select **revup** to increment all charm's revision otherwise **JuJu** will not deploy latest version of the code !
-

4.7.8 How to efficiently add features ?

Note: You need at least a running deployment to test the real behavior of components as described here [How to launch the demonstrator ?](#).

At the early stage of the development, it was easy to test the behavior of the Orchestrator without too much effort. In fact, it was easy because on previous versions it wasn't necessary to deploy the shared storage nor the transform/publisher units.

Now, this is not as easy as before as they are really nice features to test !

Here is explained the steps I followed during the developments (be warned, they are a lot):

1. Deploy the demonstrator on a local environment for speed or on a real cloud, as you prefer ;

2. The Orchestrator

- Edit the source code :

```
>>> ~$ cd $HOME/OSCIED/components/orchestra/charm
>>> ~/OSCIED/components/orchestra/charm$ nano ...
...
```

- Update the running unit :

```
>>> ~$ cd $HOME/OSCIED/scripts
>>> ~/OSCIED/scripts$ sh menu.sh
```

– Select **rsync_orchestra** to update running unit source code ;

- Commit any relevant modification to code :

```
>>> ~$ ~/OSCIED/components/orchestra/charm$ svn st
M orchestra.py
M lib/Orchestra.py
>>> ~/OSCIED/components/orchestra/charm$ svn commit -m 'I implemented something cool'
```

3. The Transform

- Add subversion to the running unit :

```
>>> ~$ ssh oscied-transform/0
>>> (transform) ~$ sudo su
>>> (transform) ~# ln -s /var/lib/juju/units/oscied-transform-0/charm
>>> (transform) ~# ln -s /var/lib/juju/units/oscied-transform-0/charm.log
>>> (transform) ~# apt-get install subversion
>>> (transform) ~# cd charm
>>> (transform) ~/charm# svn co https://claire-et-david.dyndns.org/prog/OSCIED/
components/transform/charm/
>>> (transform) ~/charm# mv charm/.svn .
>>> (transform) ~/charm# rm -rf charm
>>> (transform) ~/charm# svn st
? celeryconfig.py
```

- Attach to screen and edit code :

```
>>> (transform) ~/charm# screen -r
(you will see celeryd output 'log')
(CTRL+A C to create a new tab into screen)
>>> (transform) ~/charm# nano lib/Transform.py
(CTRL+A N to see celeryd output 'log')
(CTRL+C to stop celeryd)
>>> (transform) ~/charm# celeryd -Q transform_private
...

```

- Commit any relevant modification to code :

```
>>> (transform) ~/charm# svn st
? celeryconfig.py
M lib/Transform.py
M lib/Medias.py
>>> (transform) ~/charm# svn commit lib/ -m 'I implemented something cool'
```

4. The Web User Interface

- Edit the source code :

```
>>> ~$ cd ${HOME}/OSCIED/components/webui/charm/www
>>> ~/OSCIED/components/webui/charm/www$ nano ...
...

```

- Update the running unit :

```
>>> ~$ cd ${HOME}/OSCIED/scripts
>>> ~/OSCIED/scripts$ sh menu.sh
```

- Select **rsync_webui** to update running unit source code ;

- Commit any relevant modification to code :

```
>>> ~$ ~/OSCIED/components/webui/charm/www$ svn st
M application/controllers/medias.php
>>> ~/OSCIED/components/webui/charm/www$ svn commit -m 'I implemented something cool'
```

5. Add automated test calls to Orchestra's RESTful API in order to *unit test* the Orchestrator :

- Edit the script :

```
>>> ~$ cd ${HOME}/OSCIED/scripts
>>> ~/OSCIED/scripts$ nano menu.sh
...
(update api_test_all method) ...
```

- Launch the test :

```
>>> ~/OSCIED/scripts$ sh menu.sh
```

- Select **api_test_all** to test features of the API ;

| |
|--|
| Warning: This method will flush the Orchestrator's database ! |
|--|

6. Update OSCIED Project's TRAC Environment by creating / solving tickets and update **this documentation** too !

4.7.9 How to update documentation ?

Has you already noticed, in the `report/david/MA/` path of the project sit a lot of things and you will find in `source/` path some `.rst` files. If you wish to contribute to this the project documentation I must introduce you with `reStructuredText`.

`reStructuredText` is an easy-to-read plaintext markup syntax parseable by specific tools to produce documentation in a wide variety of formats. This markup syntax is really useful for developers to add *smart* in-line documentation to their code (such as `Python docstrings`).

For this project I wrote my documentation (my report) in .rst files and I used the powerful `Sphinx` parser to create a makefile for the documentation to be generated !

As always, a bash script is also provided to generate the documentation `generate-report.sh`, so you only need to run it to produce corresponding html + pdf²³:

```
>>> ~/OSCIED$ cd scripts/  
>>> ~/OSCIED/scripts$ sh generate-report.sh
```

This script will install required packages, cleanup output paths and generate the documentation into OSCIED/report/david/MA

See Also:

reStructuredText Rocks ! Please see [Ticket 141](#) to be convinced.

²³ Be aware the warnings & errors of the command-line, this is a compilation process, it may fail !

4.8 Deployment Scenarios

4.8.1 JuJu++

At time of writing this report, one cannot uses JuJu to :

1. Deploy charms locally without using the providers such as MAAS or LXC.
2. Add relations between services running on different environments.

... Challenge accepted, let's make it possible !

First Challenge

I partially hacked this by bypassing JuJu and creating my own set of tools to install charms locally, here are the key elements of this hack :

Problem

Charms hooks requires JuJu's callable methods like juju-log, config-get, unit-get, ...

Solution

Home-made implementation of the missing methods :

- **juju-log** is mapped to a bash script with colorful echoes
- ***-get** are mapped to a bash script having as input the options *name* and echoing corresponding *value* based on a predefined *name = value* file

Problem

Charms hooks must be called in order to deploy the application.

Solution

A simple command-line dialog menu allowing the user to call charms hooks directly. This utility copy the home-made implementation of the missing JuJu's methods to local binaries path /usr/local/bin/ before calling the hook.

See Also:

Hacks are available here : *JuJu Menu Templates and Utilities*.

Second Challenge

Problem

To connect services together (application's charms instances) one need to use `juju add-relation` however JuJu only allow you to connect services from the same environment.

Solution

In order to allow such kind of complex deployment the following have been added into charm's configuration file `config.yaml` :

- *storage* related options for charms requiring the *storage* relation.
- *publisher* related options for *Publisher* charm requiring the *publisher* relation.
- *transform* related options for *Transform* charm requiring the *transform* relation.

It may sound as redundant at first sight however it unlock the elasticity of the application !

For example, you only need to specify the *storage* related options into `orchestra.yaml` and then deploy the *Orchestra* charm with `juju deploy --config orchestra.yaml (...)` `oscied-orchestra` to make the orchestrator instance using the storage you specified. This will also disable the orchestrator instance's hooks related to *storage* relation.

4.8.2 Local Deployment

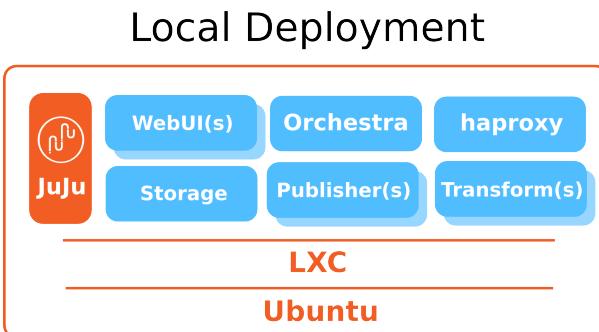


Figure 4.44: Local deployment is really useful for development & testing purposes

See Also:

Please see official documentation for JuJu [Local provider](#).

environments.yaml

```

default: local
environments:
  local:
    type: local
    control-bucket: juju-a14dfaef3830142d9ac23c499395c2785999
    admin-secret: 6608267bb6b447b8c90934167b2a294999
    data-dir: /home/<username>/juju/storage
    default-series: quantal
    juju-origin: ppa
  
```

Procedure

```

juju bootstrap
juju deploy --repository=charms/ local:quantal/oscied-orchestra
juju deploy --repository=charms/ local:quantal/oscied-webui
juju deploy --repository=charms/ local:quantal/oscied-storage
juju deploy --repository=charms/ local:quantal/oscied-transform -n 3
juju deploy --repository=charms/ local:quantal/oscied-publisher -n 2
juju deploy cs:precise/haproxy
juju expose oscied-storage
juju expose oscied-orchestra
juju expose oscied-publisher
juju expose haproxy
juju add-relation oscied-storage          oscied-transform
juju add-relation oscied-storage          oscied-publisher
juju add-relation oscied-storage          oscied-orchestra
juju add-relation oscied-storage          oscied-webui
juju add-relation oscied-orchestra:transform oscied-transform:transform
juju add-relation oscied-orchestra:publisher oscied-publisher:publisher
juju add-relation oscied-orchestra:api      oscied-webui:api
juju add-relation haproxy                 oscied-webui
  
```

4.8.3 Cloud Deployment

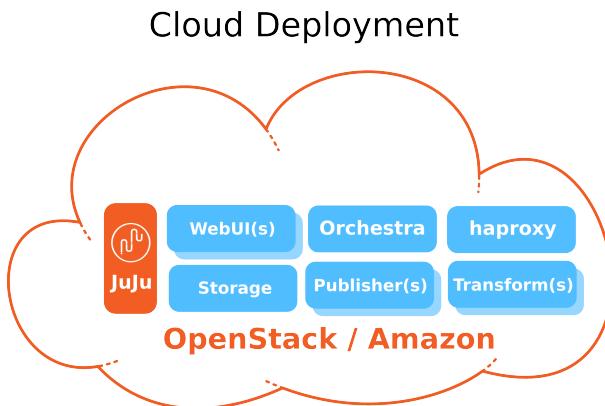


Figure 4.45: Cloud deployment is really interesting for his elasticity

See Also:

Please see official documentation for JuJu [OpenStack](#), [Amazon AWS](#), [HP Cloud](#), [Rackspace](#) providers.

environments.yaml

```
default: amazon
environments:
  amazon:
    type: ec2
    access-key: AKI*****  

    secret-key: Vl5g/QL10*****  

    control-bucket: juju-24etR*****.s3-website-us-east-1.amazonaws.com
    admin-secret: 81ale7429e6847*****  

    default-series: quantal
    juju-origin: ppa
```

Procedure

```
tmicro='instance-type=t1.micro'
mmedium='instance-type=m1.medium'
cmedium='instance-type=c1.medium'
juju bootstrap
juju deploy --constraints "$tmicro" --repository=charms/ local:quantal/oscied-orchestra
juju deploy --constraints "$tmicro" --repository=charms/ local:quantal/oscied-webui
juju deploy --constraints "$mmedium" --repository=charms/ local:quantal/oscied-storage
juju deploy --constraints "$cmedium" --repository=charms/ local:quantal/oscied-transform -n 3
juju deploy --constraints "$mmedium" --repository=charms/ local:quantal/oscied-publisher -n 2
juju expose oscied-storage
juju expose oscied-orchestra
juju expose oscied-publisher
juju expose oscied-webui
juju add-relation oscied-storage          oscied-transform
juju add-relation oscied-storage          oscied-publisher
juju add-relation oscied-storage          oscied-orchestra
juju add-relation oscied-storage          oscied-webui
juju add-relation oscied-orchestra:transform oscied-transform:transform
juju add-relation oscied-orchestra:publisher oscied-publisher:publisher
juju add-relation oscied-orchestra:api      oscied-webui:api
```

4.8.4 Multi-Environment Deployment

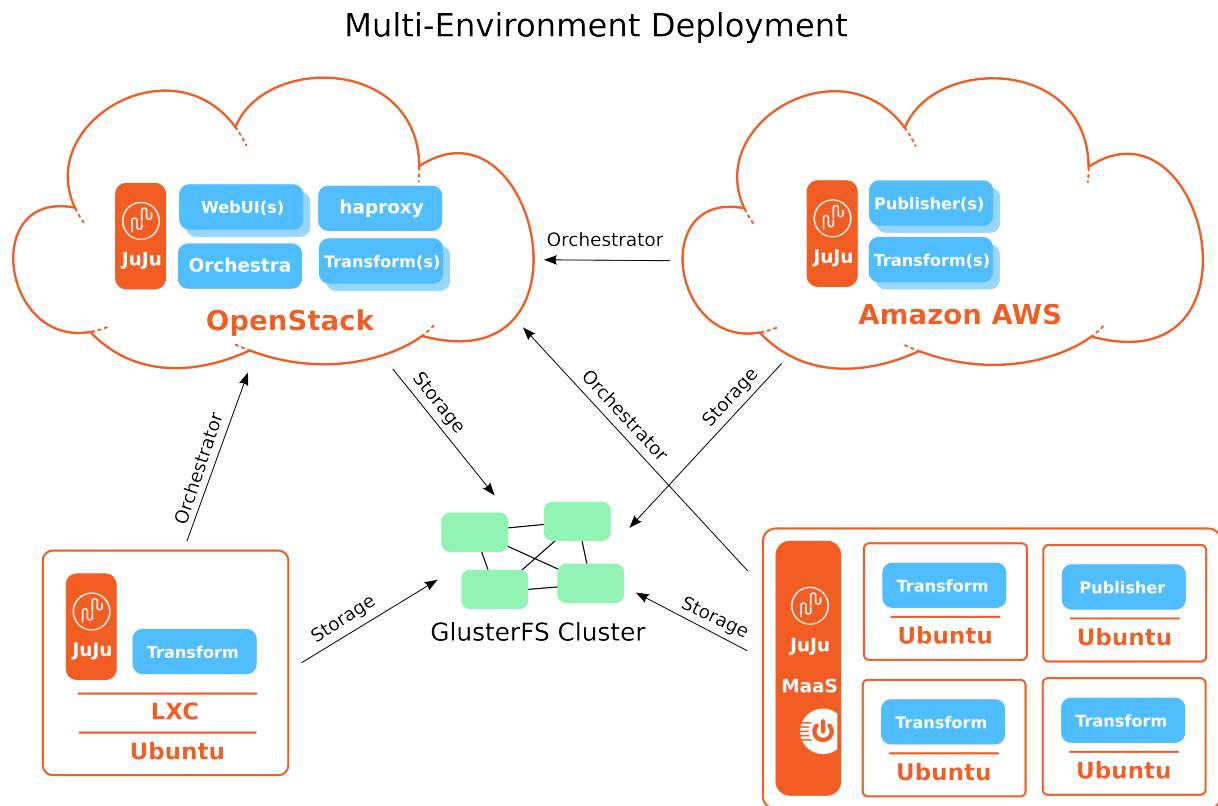


Figure 4.46: The application components can be deployed in parallel to any compatible environment !

See Also:

Please see official documentation for JuJu Local, MAAS, OpenStack, Amazon AWS, HP Cloud, Rackspace providers.

environments.yaml

```
default: amazon
environments:
  local:
    type: local
    control-bucket: juju-a14dfaef3830142d9ac23c499395c2785999
    admin-secret: 6608267bb6b447b8c90934167b2a294999
    data-dir: /home/<username>/juju/storage
    default-series: quantal
    juju-origin: ppa
  maas:
    type: maas
    maas-server: 'http://<maas_host>:5240'
    maas-oauth: '${maas-api-key}'
    admin-secret: 'nothing'
    juju-origin: ppa
  openstack:
    type: openstack_s3
    control-bucket: juju-05n2105318671zvmr9388d6db2725871636
    admin-secret: 63419585698545811584r7691832885p714
    auth-url: https://<yourkeystoneurl>:443/v2.0/
    default-series: quantal
    juju-origin: ppa
    ssl-hostname-verification: True
    default-image-id: bb636e4f-79d7-4d6b-b13b-c7d53419fd5a
```

```
default-instance-type: m1.small
amazon:
  type: ec2
  access-key: AKI*****
  secret-key: Vl5g/QLi0*****
  control-bucket: juju-24etR*****.s3-website-us-east-1.amazonaws.com
  admin-secret: 81a1e7429e6847*****
  default-series: quantal
  juju-origin: ppa
```

publisher-amazon.yaml

```
oscied-publisher:
  verbose: "true"
  concurrency: 4
  rabbit_queues: "publisher_amazon"
  max_upload_size: 4294967296
  max_execution_time: 180
  max_input_time: 600
  mongo_connection: "mongodb://nodes:M2qlif8rdtKtBYil@<orchestra_ip>:27017/celery"
  rabbit_connection: "amqp://nodes:OZy23iO0D4UpYS2k@<orchestra_ip>:5672/celery"
  storage_ip: "<storage_ip>"
  storage_fstype: "glusterfs"
  storage_mountpoint: "medias_volume"
  storage_options: ""
```

publisher-maas.yaml

```
oscied-publisher:
  verbose: "true"
  concurrency: 6
  rabbit_queues: "publisher_maas"
  max_upload_size: 4294967296
  max_execution_time: 180
  max_input_time: 600
  mongo_connection: "mongodb://nodes:M2qlif8rdtKtBYil@<orchestra_ip>:27017/celery"
  rabbit_connection: "amqp://nodes:OZy23iO0D4UpYS2k@<orchestra_ip>:5672/celery"
  storage_ip: "<storage_ip>"
  storage_fstype: "glusterfs"
  storage_mountpoint: "medias_volume"
  storage_options: ""
```

orchestra-openstack.yaml

```
oscied-orchestra:
  verbose: "true"
  root_secret: "bXzZ6SFmh0Z5a8PQ"
  nodes_secret: "Y3v8rXTyPjTAQgI0"
  repositories_user: ""
  repositories_pass: ""
  webui_repository: ""
  transform_repository: ""
  publisher_repository: ""
  mongo_admin_password: "Iz85QVjdCJugEosz"
  mongo_nodes_password: "M2qlif8rdtKtBYil"
  rabbit_password: "IwmZk3F3suCH8rvC"
  juju_environment: ""
  storage_ip: "<storage_ip>"
  storage_fstype: "glusterfs"
  storage_mountpoint: "medias_volume"
  storage_options: ""
```

transform-local.yaml

```
oscied-transform:  
  verbose: "true"  
  concurrency: 2  
  rabbit_queues: "transform_local"  
  mongo_connection: "mongodb://nodes:M2qlif8rdtKtBYil@<orchestra_ip>:27017/celery"  
  rabbit_connection: "amqp://nodes:OZy23iO0D4UpYS2k@<orchestra_ip>:5672/celery"  
  storage_ip: "<storage_ip>"  
  storage_fstype: "glusterfs"  
  storage_mountpoint: "medias_volume"  
  storage_options: ""
```

transform-maas.yaml

```
oscied-transform:  
  verbose: "true"  
  concurrency: 12  
  rabbit_queues: "transform_maas, transform_openstack"  
  mongo_connection: "mongodb://nodes:M2qlif8rdtKtBYil@<orchestra_ip>:27017/celery"  
  rabbit_connection: "amqp://nodes:OZy23iO0D4UpYS2k@<orchestra_ip>:5672/celery"  
  storage_ip: "<storage_ip>"  
  storage_fstype: "glusterfs"  
  storage_mountpoint: "medias_volume"  
  storage_options: ""
```

transform-openstack.yaml

```
oscied-transform:  
  verbose: "true"  
  concurrency: 2  
  rabbit_queues: "transform_openstack"  
  mongo_connection: ""  
  rabbit_connection: ""  
  storage_ip: "<storage_ip>"  
  storage_fstype: "glusterfs"  
  storage_mountpoint: "medias_volume"  
  storage_options: ""
```

transform-amazon.yaml

```
oscied-transform:  
  verbose: "true"  
  concurrency: 2  
  rabbit_queues: "transform_amazon"  
  mongo_connection: "mongodb://nodes:M2qlif8rdtKtBYil@<orchestra_ip>:27017/celery"  
  rabbit_connection: "amqp://nodes:OZy23iO0D4UpYS2k@<orchestra_ip>:5672/celery"  
  storage_ip: "<storage_ip>"  
  storage_fstype: "glusterfs"  
  storage_mountpoint: "medias_volume"  
  storage_options: ""
```

webui-openstack.yaml

```
oscied-webui:  
  verbose: "true"  
  max_upload_size: 4294967296  
  max_execution_time: 180  
  max_input_time: 600  
  mysql_my_password: "mUzf4JUwTIa3AIXj"  
  mysql_root_password: "mUzf4JUwTIa3AIXj"  
  mysql_user_password: "SD1MwuxjMzck2ZCs"  
  storage_ip: "<storage_ip>"  
  storage_fstype: "glusterfs"  
  storage_mountpoint: "medias_volume"  
  storage_options: ""
```

Procedure

```
tmicro='instance-type=t1.micro'
mmedium='instance-type=m1.medium'
cmedium='instance-type=c1.medium'
repo='--repository=charms/'
path='local:quantal/oscied'
c='--constraints'
cfg='--config'
stack='openstack'

juju bootstrap -e openstack
juju deploy -e $stack $c "$tmicro" $cfg orchestra-openstack.yaml $repo $oscied-orchestra
juju deploy -e $stack $c "$tmicro" $cfg webui-openstack.yaml $repo $oscied-webui
juju deploy -e $stack $c "$cmedium" $cfg transform-openstack.yaml $repo $oscied-transform -n 2
juju expose -e $stack oscied-orchestra
juju expose -e $stack oscied-webui
juju add-relation -e $stack oscied-orchestra:transform oscied-transform:transform
juju add-relation -e $stack oscied-orchestra:api oscied-webui:api

juju bootstrap -e amazon
juju deploy -e amazon $c "$cmedium" $cfg transform-amazon.yaml $repo $oscied-transform -n 2
juju deploy -e amazon $c "$mmedium" $cfg publisher-amazon.yaml $repo $oscied-publisher -n 2
juju expose -e amazon oscied-publisher

juju bootstrap -e maas
juju deploy -e maas $cfg transform-maas.yaml $repo $oscied-transform -n 3
juju deploy -e maas $cfg publisher-maas.yaml $repo $oscied-publisher -n 1
juju expose -e maas oscied-publisher

juju bootstrap -e local
juju deploy -e local $cfg transform-local.yaml $repo $oscied-transform
```

4.9 Tests and Results

4.9.1 Foreword

You may have noticed they are missing *numbers* here. The reason is the only conclusions one would gather from *performance benchmarks* would be the following :

- Actual storage charm cannot scale, this is the bottleneck of any deployment using it
- Celery does work as expected, workers keep connection to queues and jobs are handled rapidly
- Encoding : Sure, a Pentium IV is slower than a i7
- ...

This is out of the scope of the project.

4.9.2 Tests of Orchestra API

The Orchestrator's API is my first RESTful API !

So I decided to create various scripts using [cURL](#) in to help me testing the functionalities of the API to test various inputs and corresponding outputs. Outputs of the API were also validated with a [json](#) string validator called [JSONLint](#). The scripts are also useful to ensure security policies by testing responses (code + value) to scripted requests. One can call that unit testing ... OK it is.

4.9.3 Charms are Reliable

The application's components are charms, charms means hooks and hooks means code.

At the beginning of the charm's developments, it was an intensive period of deployment, debugging ... The charm's hooks where developed with the following algorithm :

```
counter = 0
while time < 03:00 AM:
    juju deploy charms
    juju debug-log
    if any error:
        reason = decipher (error)
        solution = find solution (reason)
    elif sufficient_features:
        say "Ouais !"
        break
    juju destroy
    counter++
    if counter mod 10 == 0:
        drink (water)
    foreach charm of the application:
        hooks += feature
    foreach faulty charm of the application:
        hooks += solution
    sleep (8 hours)
```

The charms are now robust as they are developed, tested and debugged. ... And also successfully deployed for weeks on [Amazon AWS](#) and [LXC](#).

4.9.4 Tested Deployment Scenarios

What was the procedure ?

- Deployment of the application's charms to the chosen environment
- Live checking of the juju's debug-log (are there any error ?)

- Remote access to unit's in order to update code in case of error
- Scaling of transform & publisher units by adding/removing units
- Usage of the orchestrator's API helped with scripts
- Usage of the platform with the web user interface

What was the results ?

- The scaling of transform and publisher actually works as expected
- The more you add transform units, the more you will encode in parallel
- The more you add publisher units, the more you will handle streaming sessions
- The API security policies are applied, e.g. one cannot revoke tasks of other users
- **The (error) messages & HTTP codes are accurate :**
 - 200 : User “Mathias Coinchon” successfully updated
 - 403 : Authentication failed
 - 404 : No media with id ...
 - ...

Nothing is perfect, they are room for improvements !

Here are the most interesting of the numerous live-tests (at least the one I keep records).

Amazon with the minimal setup

This scenario represent a typical *public cloud only* deployment of the platform.

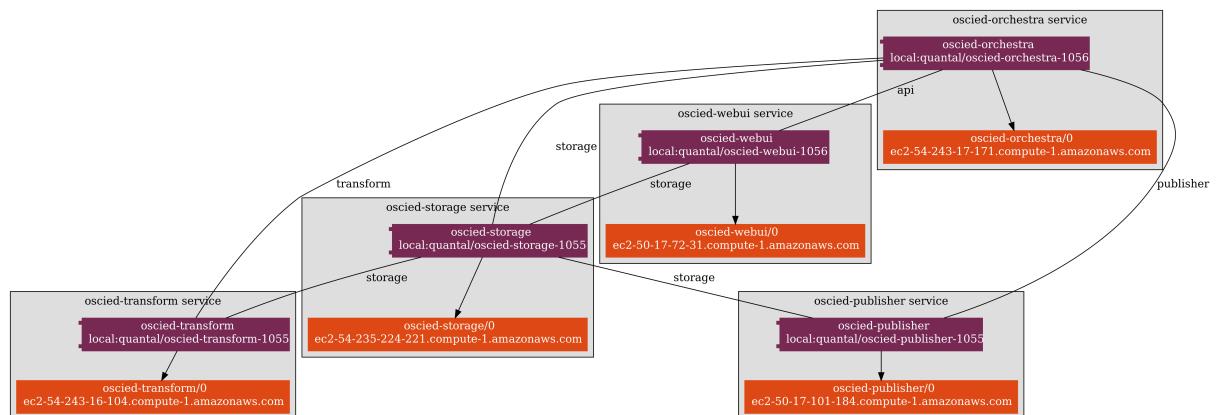


Figure 4.47: Status of the deployment by JuJu

The screenshot shows the EC2 Management Console interface. On the left, there's a sidebar with navigation links for EC2 Dashboard, Events, Instances (selected), Spot Requests, Reserved Instances, Images, AMIs, Bundle Tasks, Elastic Block Store, Volumes, Snapshots, Network & Security, Security Groups, Elastic IPs, Placement Groups, Load Balancers, Key Pairs, and Network Interfaces.

The main area displays a table of instances:

| Name | Instance | AMI ID | Root Device | Type | State | Status Checks | Alarm Status | Monitoring | Security Group |
|-----------|------------|--------------|-------------|----------|---------|-------------------|--------------|------------|-------------------|
| juju | i-a831fed8 | ami-7539b41c | ebs | m1.small | running | 2/2 checks passed | none | basic | juju-amazon-0, ju |
| orchestra | i-ee4b849e | ami-7539b41c | ebs | t1.micro | running | 2/2 checks passed | none | basic | juju-amazon-1, ju |
| webui | i-544a8524 | ami-7539b41c | ebs | t1.micro | running | 2/2 checks passed | none | basic | juju-amazon-2, ju |
| storage | i-804a85f0 | ami-7539b41c | ebs | t1.micro | running | 2/2 checks passed | none | basic | juju-amazon-3, ju |
| transform | i-16498666 | ami-7539b41c | ebs | t1.micro | running | 2/2 checks passed | none | basic | juju-amazon-4, ju |
| publisher | i-8a4986fa | ami-7539b41c | ebs | t1.micro | running | 2/2 checks passed | none | basic | juju-amazon-5, ju |

A modal window titled "1 EC2 Instance selected." is open for the instance "orchestra". It shows the following details:

- EC2 Instance: orchestra (i-ee4b849e)** (green status)
- Description:** ec2-54-243-17-171.compute-1.amazonaws.com
- AMI:** ubuntu/images/ebs/ubuntu-quantal-12.10-amd64-server-20121218 (ami-7539b41c)
- Zone:** us-east-1b
- Type:** t1.micro
- Scheduled Events:** No scheduled events
- Alarm Status:** none
- Security Groups:** juju-amazon-1, juju-amazon. [view rules](#)
- State:** running
- Owner:** 781188736812

Figure 4.48: Instances running on Amazon EC2

```
date 2013-01-26 19:13:24,322
```

```
revision 1056
```

```
juju debug.log
```

```
menu menu.log
```

Local with 3 Transform and 2 Publisher

This scenario represent a typical *development purposes only* local deployment of the platform.

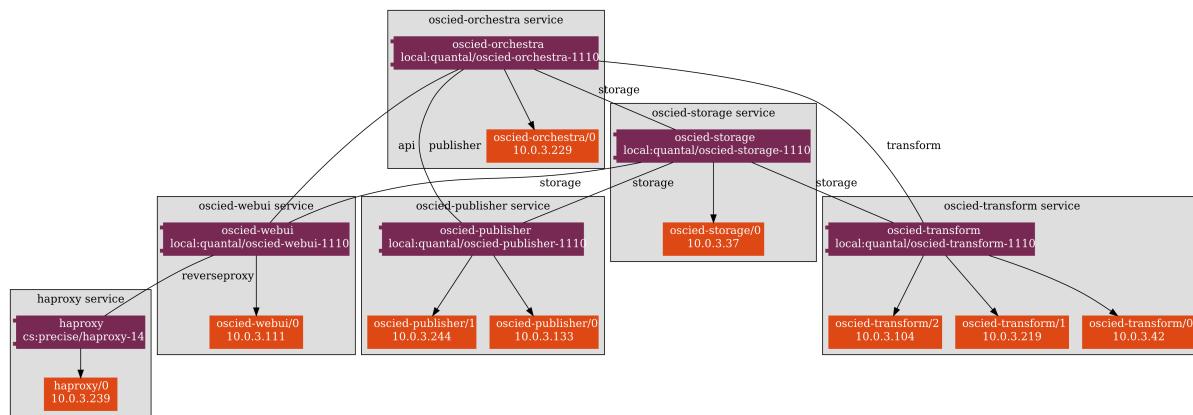


Figure 4.49: Status of the deployment by JuJu

```
date 2013-01-27 18:57:01,994
```

```
revision 1110
```

```
juju debug.log
```

screen init_and_jobs.mp4

Parallel deployment MaaS, Desktop, Amazon

This scenario represent a more powerful and realistic multi-environment deployment of the platform. In parallel are running two completely separated OSCIED : One of them is fully running on [Amazon AWS](#), the other is the one detailed here.

date 2013-03-08

revision 1278

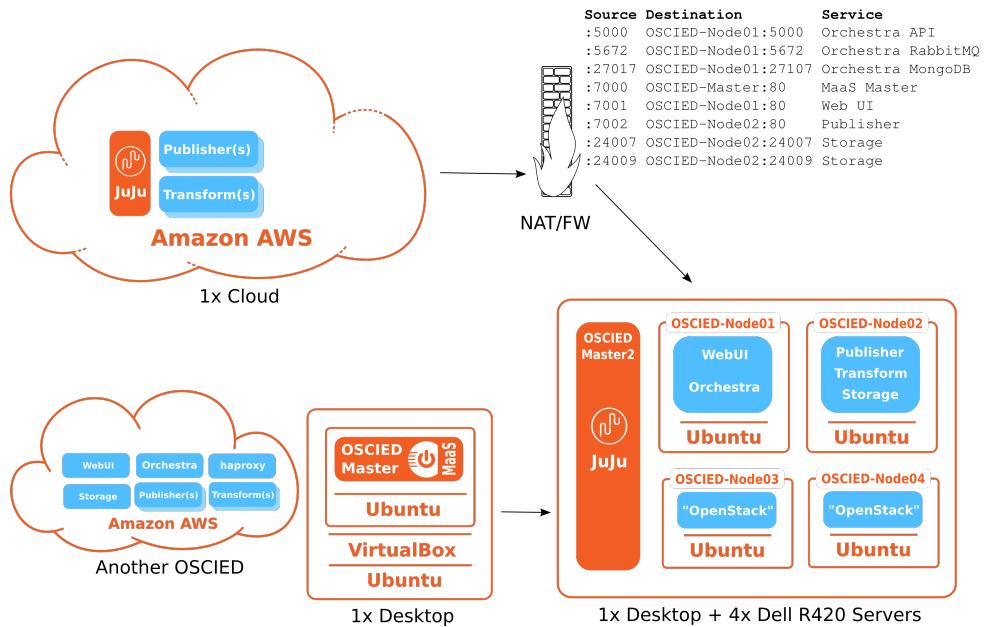


Figure 4.50: OSCIED made available as two completely separated platforms running in parallel.

| | | | | | | | |
|-------------------|----------------|------|-----|------------|-----|-------|-------|
| OSCIED-Master | Virtual Server | wan1 | any | 10.10.10.1 | any | 7000 | 80 |
| OSCIED-Website | Virtual Server | wan1 | any | 10.10.10.2 | any | 7001 | 80 |
| OSCIED-Publisher1 | Virtual Server | wan1 | any | 10.10.10.3 | any | 7002 | 80 |
| OSCIED-Publisher2 | Virtual Server | wan1 | any | 10.0.3.40 | any | 7003 | 80 |
| OSCIED-Storage | Virtual Server | wan1 | any | 10.10.10.3 | any | 24007 | 24007 |
| OSCIED-Storage2 | Virtual Server | wan1 | any | 10.10.10.3 | any | 24009 | 24009 |
| OSCIED-MongoDB | Virtual Server | wan1 | any | 10.10.10.2 | any | 27017 | 27017 |
| OSCIED-RabbitMQ | Virtual Server | wan1 | any | 10.10.10.2 | any | 5672 | 5672 |
| OSCIED-API | Virtual Server | wan1 | any | 10.10.10.2 | any | 5000 | 5000 |

Figure 4.51: Configuration of our FW

MaaS

Main components of OSCIED were installed on EBU's setup of servers. The cluster of servers were provisioned and made available by Canonical's [MaaS](#) for [JuJu](#) to deploy OSCIED. [OpenStack](#) is not required in such scenario as Canonical's [MaaS](#) provisioning help us to deploy services quite easily.

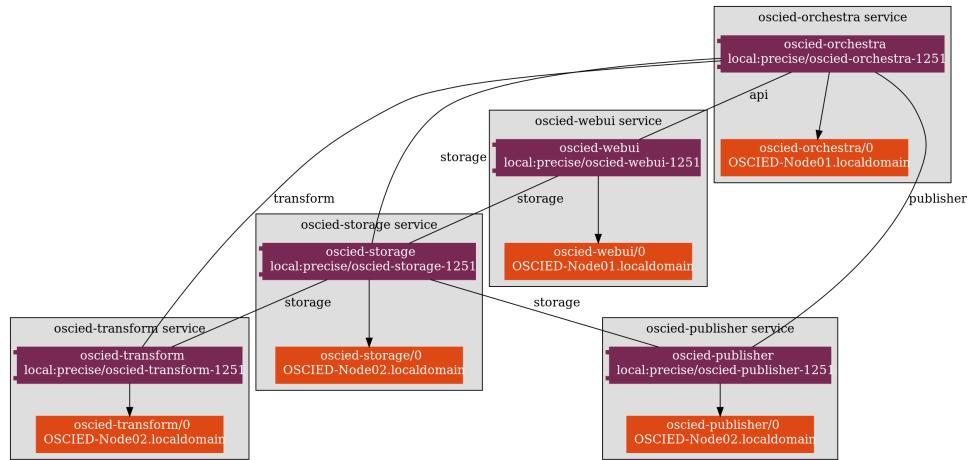


Figure 4.52: Status of the deployment by JuJu

```
menu maas maas_menu.log
juju maas maas_juju.log
```

Local

Some transform units were started on my workstation at hepia.

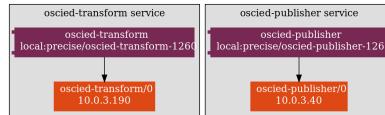


Figure 4.53: Status of the deployment by JuJu

```
menu local local_menu.log
juju local local_juju.log
```

Bernex

Some transform units were started on my personnal desktop computer at home.

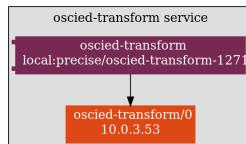


Figure 4.54: Status of the deployment by JuJu

```
menu bernex bernex_menu.log
juju bernex bernex_juju.log
```

Amazon

This public-cloud provider has been used to deploy transcoder and publication points.



Figure 4.55: Status of the deployment by JuJu

```
menu amazon amazon_menu.log
juju amazon amazon_juju.log
```

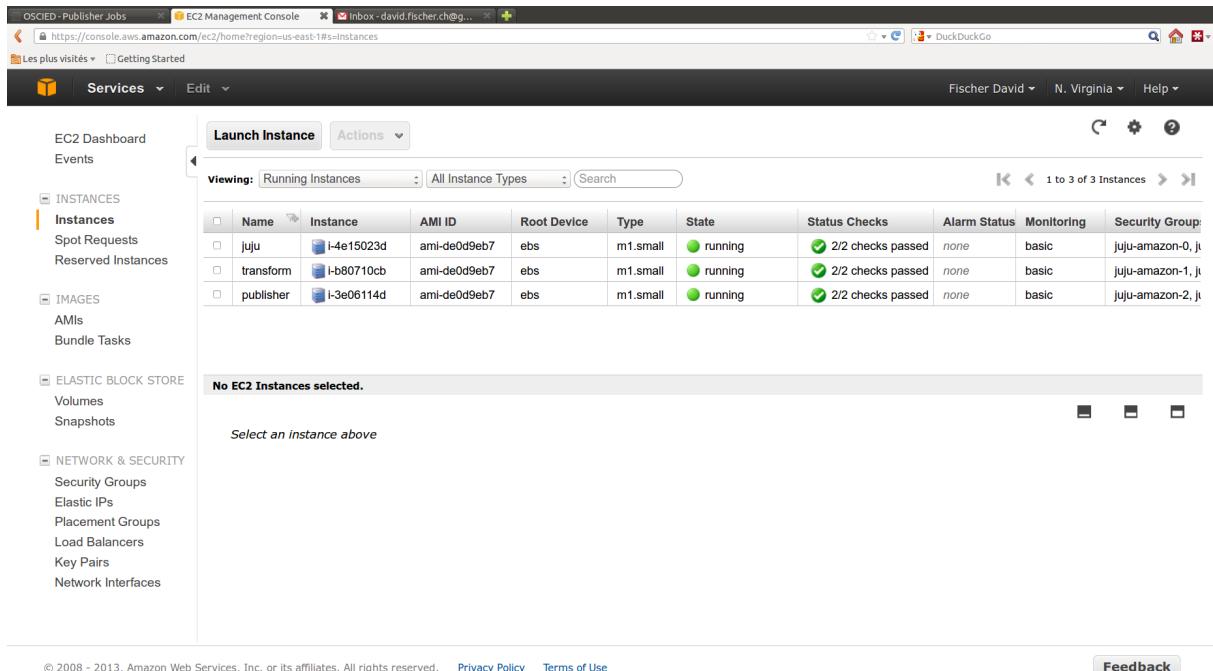


Figure 4.56: Amazon Web Services Console

4.10 Future Extensions

This section list the most interesting features to integrate to developed platform.

- See OSCIED Project's Roadmap

4.10.1 Demonstrator being Production Grade Tool

Collected ideas of improvement, for example :

- Storage : Enhance security + scalability
- Storage : Implement scalability
- Web UI : Improve tabs (add filters, ...)

4.10.2 Demonstrator with DASH Encoder

Adding MPEG-DASH encoding capabilities to OSCIED

4.10.3 Demonstrator with JuJu in Orchestra

- Improving ease of use of OSCIED
- Adding auto-scaling capabilities

4.10.4 Demonstrator with Easy Private-Cloud Provisioning

- Automating setup of new servers (maybe with [MAAS](#))

4.10.5 Demonstrator with FIMS Interface

- Being compatible with FIMS MaM's

4.10.6 Demonstrator with Codem Interface

- Integrating Codem encoder to OSCIED

CONCLUSION

With OSCIED I proved that building a platform based on cloud-era [Open-Source](#) technologies can fix the scalability issue by providing a rather simple but yet powerful way to consume already existing enterprise's IT resources mixed with necessary amount of public cloud resources.

The hybrid-cloud model is the perfect approach to combine the highly available, low-cost, in-house IT infrastructure with scalable, on-demand public cloud infrastructures : You decide, OSCIED do !

Developed application made available the following to broadcasters :

- On-demand, scalable transcoding services by running virtualized transformation nodes
- On-demand, scalable distribution services by running virtualized publisher nodes

The platform can be used through an uncluttered, user-friendly web interface that actually maps the call of orchestrator's RESTful API.

This API is a key feature allowing broadcasters to automate usage of the platform and make possible the integration of OSCIED in broadcasters automated workflows.

Implemented features are tested and works well, the multi-cloud deployment, mixed with bare-metal storage works even better. I deployed the platform for weeks on my desktop computer, on [Amazon AWS](#) and on any server I was authorized to use.

The demonstrator is the proof of concept of something bigger, something not expected, something called OSCIED!

Personal

During the development of the application I not only learned [Python](#) but also the tools I never used before like [Celery](#), [MongoDB](#), ... It was a quite interesting challenge to start my work at the servers room level and finally designing the interface that makes the platform easy to use !

The only regrets I have, is that most of the improvements I think-ed-of are easy to implement, the only concern is the lack of time, as right after this thesis I will work at full time for another nice project called GaVi (guide audiovisuel interactif).

... I will be back !



APPENDIX

6.1 Abbreviations

6.1.1 Entreprises & Institutions

| | |
|-------|---|
| EBU | European Broadcasting Union (EBU/UER) |
| HbbTV | Hybrid Broadcast Broadband Television |
| hepia | hepia : Haute école du Paysage d'Ingénierie et d'Architecture |
| IEEE | Institute of Electronical and Electronics Engineers |
| IETF | Internet Engineering Task Force |
| MPEG | Motion Picture Experts Group |
| MSE | HES-SO Master of Science in Engineering |
| OIPF | Open IPTV Forum |
| VCEG | Video Coding Experts Group |
| W3C | World Wide Web Consortium |

6.1.2 Open-Source & OSS License

| | |
|------|------------------|
| AGPL | GNU Affero GPLv3 |
| CC | Creative Commons |
| GPL | GNU GPLv3 |
| LGPL | GNU LGPLv3 |
| MIT | MIT License |
| OSS | Open-Source |

6.1.3 Networking

| | |
|-----|---------------------------|
| CDN | Content Delivery Network |
| DNS | Domain Name System |
| ISP | Internet Service Provider |
| LAN | Local Area Network |

6.1.4 Programming

| | |
|-----|-------------------------------------|
| API | Application Programming Interface |
| GIT | Git (version control system) |
| IDE | Integrated Development Environment |
| SVN | Subversion (version control system) |

6.1.5 Operational & Cloud

| | |
|------|-------------------------------|
| PXE | Preboot Execution Environment |
| SLA | Service-Level Agreement |
| MAAS | Metal as a Service |
| IaaS | Infrastructure as a Service |
| PaaS | Platform as a Service |
| SaaS | Software as a Service |

6.1.6 Multimedia

| | |
|-------|---------------------------------------|
| HbbTV | Hybrid Broadcast Broadband Television |
| QoE | Quality of Experience |
| QoS | Quality of Service |
| SDI | Serial Digital Interface |
| STB | Set-Top Box |
| TV | Television |
| VoD | Video On Demand |

6.1.7 Formats & Protocols

| | | |
|------|--------------|--------------------------------------|
| DASH | MPEG | Dynamic Adaptive Streaming over HTTP |
| FLV | Adobe | Flash Video |
| H264 | VCEG/MPEG | Advanced Video Coding |
| H265 | VCEG/MPEG | High Efficiency Video Coding |
| HLS | Apple Inc. | HTTP Live Streaming |
| HTML | W3C | Hypertext Mark-up Language |
| HTTP | IETF | Hypertext Transfer Protocol |
| IP | IETF | Internet Protocol |
| MMS | Microsoft | Microsoft Media Services |
| MP4 | MPEG | MPEG-4 File Format |
| MSS | Microsoft | Microsoft Smooth Streaming |
| RDT | RealNetworks | Real Data Transport |
| rst | Python | reStructuredText |
| RTCP | IETF | Real-Time Control Protocol |
| RTMP | Adobe | Real-Time Messaging Protocol |
| RTP | IETF | Real-Time Transport Protocol |
| RTSP | IETF | Real-Time Streaming Protocol |
| TCP | IETF | Transmission Control Protocol |
| TS | MPEG | Transport Stream (e.g. MPEG-2 TS) |
| UDP | IETF | User Datagram Protocol |
| XML | W3C | Extensible Markup Language |

6.2 References

6.2.1 OSCIED TRAC Management

- [trac_home] <http://claire-et-david.dyndns.org/OSCIED/>
- [trac_roadmap] <http://claire-et-david.dyndns.org/OSCIED/milestone>
- [trac_source] <http://claire-et-david.dyndns.org/OSCIED/browser>
- [trac_tickets_active] <http://claire-et-david.dyndns.org/OSCIED/report/3>
- [trac_tickets_all] <http://claire-et-david.dyndns.org/OSCIED/report/6>
- [trac_tickets_references] <http://claire-et-david.dyndns.org/OSCIED/report/10>

6.2.2 OSCIED TRAC Source-Code Browser

- [browse_orchestra] <http://claire-et-david.dyndns.org/OSCIED/browser/components/orchestra>
- [browse_publisher] <http://claire-et-david.dyndns.org/OSCIED/browser/components/publisher>
- [browse_storage] <http://claire-et-david.dyndns.org/OSCIED/browser/components/storage>
- [browse_transform] <http://claire-et-david.dyndns.org/OSCIED/browser/components/transform>
- [browse_webui] <http://claire-et-david.dyndns.org/OSCIED/browser/components/webui>

6.2.3 OSCIED TRAC Tickets

- [Ticket 25] <http://claire-et-david.dyndns.org/OSCIED/ticket/25>
- [Ticket 36] <http://claire-et-david.dyndns.org/OSCIED/ticket/36>
- [Ticket 37] <http://claire-et-david.dyndns.org/OSCIED/ticket/37>
- [Ticket 40] <http://claire-et-david.dyndns.org/OSCIED/ticket/40>
- [Ticket 117] <http://claire-et-david.dyndns.org/OSCIED/ticket/117>
- [Ticket 120] <http://claire-et-david.dyndns.org/OSCIED/ticket/120>
- [Ticket 122] <http://claire-et-david.dyndns.org/OSCIED/ticket/122>
- [Ticket 131] <http://claire-et-david.dyndns.org/OSCIED/ticket/131>
- [Ticket 141] <http://claire-et-david.dyndns.org/OSCIED/ticket/141>

6.2.4 Entreprises & Institutions

- [Akamai] <http://www.akamai.com/>
- [Amazon AWS] <http://aws.amazon.com/>
- [AmazonCDN] <http://aws.amazon.com/fr/cloudfront/>
- [AmazonEC2] <http://aws.amazon.com/fr/ec2/>
- [AmazonS3] <http://aws.amazon.com/fr/s3/>
- [Canonical] <http://www.canonical.com/>
- [EBU] <http://www.ebu.ch/fr/>
- [EBU_TECH] <http://tech.ebu.ch/>
- [EBU_Cloud] <http://tech.ebu.ch/cloudworkshop>

- [HbbTV] <http://www.hbbtv.org/>
- [hepia] <http://hepia.hesge.ch/>
- [HP Cloud] <https://www.hpccloud.com/>
- [IEEE] <http://www.ieee.org/index.html>
- [IETF] <http://www.ietf.org/>
- [MPEG] <http://mpeg.chiariglione.org/>
- [MSE] <http://www.hes-so.ch/en/master-engineering.html>
- [NASA] <http://www.nasa.gov/>
- [OIPF] <http://www.oipf.tv/>
- [Rackspace] <http://www.rackspace.com/>
- [W3C] <http://www.w3.org/>
- [Zencoder] <http://zencoder.com/en/>

6.2.5 Open-Source Licenses

- [AGPL] <http://www.gnu.org/licenses/agpl-3.0.html>
- [CC] <http://creativecommons.org/>
- [GPL] <http://www.gnu.org/licenses/gpl.html>
- [LGPL] <http://www.gnu.org/copyleft/lesser.html>
- [MIT] http://en.wikipedia.org/wiki/MIT_License
- [OSS] http://en.wikipedia.org/wiki/Open_source

6.2.6 Codecs, Formats & Protocols

- [BSON] <http://bsonspec.org/>
- [DASH] <http://dashif.org/mpeg-dash/>
- [FLV] http://en.wikipedia.org/wiki/Flash_Video
- [H264] <http://fr.wikipedia.org/wiki/H.264>
- [H265] http://en.wikipedia.org/wiki/High_Efficiency_Video_Coding
- [HLS] http://en.wikipedia.org/wiki/HTTP_Live_Streaming
- [HTML] <http://en.wikipedia.org/wiki/HTML>
- [HTTP] http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol
- [IP] http://en.wikipedia.org/wiki/Internet_Protocol
- [JSON] <http://www.json.org/>
- [JSONLint] <http://jsonlint.com/>
- [MMS] http://fr.wikipedia.org/wiki/Microsoft_Media_Services
- [MP4] http://en.wikipedia.org/wiki/MPEG-4_Part_14
- [MSS] <http://www.iis.net/downloads/microsoft/smooth-streaming>
- [RDT] http://en.wikipedia.org/wiki/Real_Data_Transport
- [rst] <http://docs.python.org/3.0/documenting/rest.html>
- [RTCP] http://en.wikipedia.org/wiki/RTP_Control_Protocol

- [RTMP] http://en.wikipedia.org/wiki/Real_Time_Messaging_Protocol
- [RTP] http://en.wikipedia.org/wiki/Real-time_Transport_Protocol
- [RTSP] <http://en.wikipedia.org/wiki/Rtsp>
- [TCP] http://en.wikipedia.org/wiki/Transmission_Control_Protocol
- [TS] http://en.wikipedia.org/wiki/Transport_stream
- [UDP] http://en.wikipedia.org/wiki/User_Datagram_Protocol
- [XML] <http://en.wikipedia.org/wiki/XML>

6.2.7 Database & Storage

- [Ceph] <http://ceph.com/ceph-storage/>
- [GlusterFS] <http://www.gluster.org/>
- [GlusterFS_admin_guide] http://www.gluster.org/wp-content/uploads/2012/05/Gluster_File_System-3.3.0-Administration_Guide-en-US.pdf
- [LS4] <http://ls4.sourceforge.net/>
- [LVM] http://en.wikipedia.org/wiki/Logical_Volume_Manager_%28Linux%29
- [MongoDB] <http://www.mongodb.org/>
- [MySQL] <http://www.mysql.com/>
- [NFS] http://en.wikipedia.org/wiki/Network_File_System

6.2.8 Multimedia

- [Distribution] http://en.wikipedia.org/wiki/Digital_distribution
- [FFmpeg] <http://ffmpeg.org/>
- [OBE] <http://www.ob-encoder.com/index.html>
- [QoE] http://en.wikipedia.org/wiki/Quality_of_experience
- [QoS] http://en.wikipedia.org/wiki/Quality_of_service
- [SDI] http://en.wikipedia.org/wiki/Serial_digital_interface
- [STB] http://en.wikipedia.org/wiki/Set-top_box
- [Transcoding] <http://en.wikipedia.org/wiki/Transcoding>
- [TV] <http://en.wikipedia.org/wiki/Television>
- [VoD] http://en.wikipedia.org/wiki/Video_on_demand

6.2.9 Networking

- [CDN] http://en.wikipedia.org/wiki/Content_delivery_network
- [DNS] http://en.wikipedia.org/wiki/Domain_Name_System
- [ISP] http://en.wikipedia.org/wiki/Internet_service_provider
- [LAN] http://en.wikipedia.org/wiki/Local_area_network

6.2.10 Operational & Cloud

- [Cloud computing] http://en.wikipedia.org/wiki/Cloud_computing
- [PXE] http://fr.wikipedia.org/wiki/Preboot_Execution_Environment
- [SLA] http://en.wikipedia.org/wiki/Service-level_agreement
- [MAAS] <http://www.ubuntu.com/cloud/orchestration/deployment>
- [IaaS] http://en.wikipedia.org/wiki/Infrastructure_as_a_service
- [PaaS] http://en.wikipedia.org/wiki/Platform_as_a_service
- [SaaS] http://en.wikipedia.org/wiki/Software_as_a_service

6.2.11 JuJu

- [JuJu] <https://juju.ubuntu.com/>
- [JuJu_software] http://en.wikipedia.org/wiki/Juju_%28software%29
- [juju_charms_store] <http://jujucharms.com/charms>
- [juju_unit_startup] <https://juju.ubuntu.com/docs/internals/unit-agent-startup.html>
- [juju_local_provider] <https://juju.ubuntu.com/get-started/local/>
- [juju_maas_provider] <https://juju.ubuntu.com/get-started/maas/>
- [juju_openstack_provider] <https://juju.ubuntu.com/get-started/openstack/>
- [juju_amazon_provider] <https://juju.ubuntu.com/get-started/amazon/>
- [juju_hpcloud_provider] <https://juju.ubuntu.com/get-started/hp-cloud/>
- [juju_rackspace_provider] <https://juju.ubuntu.com/get-started/rackspace/>
- [Nginx Charm] <http://jujucharms.com/~imbrandon/precise/nginx>

6.2.12 OpenStack

- [OpenStack] <http://www.openstack.org/>
- [OpenStack Foundation] <https://www.openstack.org/join>
- [OpenStack Documentation] <http://docs.openstack.org/>
- [OpenStack_IaaS] <http://en.wikipedia.org/wiki/OpenStack>
- [Cinder] <http://docs.openstack.org/developer/cinder/>
- [Glance] <http://docs.openstack.org/developer/glance/>
- [Horizon] <http://docs.openstack.org/developer/horizon/>
- [Keystone] <http://docs.openstack.org/developer/keystone/>
- [Nova] <http://docs.openstack.org/developer/nova/>
- [Quantum] <http://docs.openstack.org/developer/quantum/>
- [Swift] <http://docs.openstack.org/developer/swift/>
- [openstack_folsom_gre_2nic] <https://github.com/mseknibilel/OpenStack-Folsom-Install-guide>
- [OS_folsom_install_guide] <https://github.com/mseknibilel/OpenStack-Folsom-Install-guide>
- [OS_folsom_gre_2nics] https://github.com/mseknibilel/OpenStack-Folsom-Install-guide/blob/GRE/2NICs/OpenStack_Folsom
- [DevStack] <http://devstack.org/>

- [TryStack] <http://trystack.org/>

6.2.13 Programming

- [API] http://en.wikipedia.org/wiki/Application_programming_interface
- [GIT] <http://git-scm.com/>
- [GitHub] <https://github.com/>
- [IDE] http://en.wikipedia.org/wiki/Integrated_development_environment
- [Launchpad] <https://launchpad.net/>
- [snippets] [http://en.wikipedia.org/wiki/Snippet_\(programming\)](http://en.wikipedia.org/wiki/Snippet_(programming))
- [SourceForge] <http://sourceforge.net/>
- [Subversion] <http://subversion.apache.org/>
- [SVN] <http://subversion.apache.org/>
- [StatSVN] <http://www.statsvn.org/>
- [TRAC] <http://trac.edgewall.org/>
- [TRACP] <http://trac.edgewall.org/wiki/TracPlugins>
- [VCS] http://en.wikipedia.org/wiki/Revision_control

6.2.14 Python and Sphinx

- [docstrings] <http://www.python.org/dev/peps/pep-0257/>
- [PyMongo] <http://api.mongodb.org/python/current/>
- [Python] <http://www.python.org/>
- [Sphinx_autodoc] <http://sphinx-doc.org/ext/autodoc.html>
- [Sphinx_autoflask] <http://packages.python.org/sphinxcontrib-httpdomain/>
- [Sphinx] <http://sphinx-doc.org/>

6.2.15 Virtualization

- [BridgeUtils] <http://www.linuxfromscratch.org/blfs/view/svn/basicnet/bridge-utils.html>
- [KVM] http://www.linux-kvm.org/page/Main_Page
- [LXC] <http://lxc.sourceforge.net/>
- [OpenVZ] http://openvz.org/Main_Page
- [vSwitch] <http://openvswitch.org/>
- [YahyaNursalim] <http://www.yahyanursalim.com/linux/cloud-computing-with-proxmox-ve.html>

6.2.16 Web Development

- [Apache 2] <http://httpd.apache.org/>
- [Celery] <http://celeryproject.org/>
- [Celery_Tasks] <http://docs.celeryproject.org/en/latest/userguide/tasks.html>
- [CodeIgniter] <http://ellislab.com/codeigniter>

- [CSS Bootstrap] <http://twitter.github.com/bootstrap/>
- [cURL] <http://en.wikipedia.org/wiki/CURL>
- [Flask] <http://flask.pocoo.org/>
- [H264 Streaming Module] <http://h264.code-shop.com/trac/wiki/Mod-H264-Streaming-Apache-Version2>
- [RabbitMQ] <http://www.rabbitmq.com/>

6.2.17 Miscellaneous

- [KISS] http://en.wikipedia.org/wiki/KISS_principle
- [Linux] <http://www.gnu.org/gnu/linux-and-gnu.html>
- [SSH] http://en.wikipedia.org/wiki/Secure_Shell
- [Ubuntu] <http://www.ubuntu.com/>
- [UbuntuOS] http://en.wikipedia.org/wiki/Ubuntu_%28operating_system%29
- [Ubuntu Quantal Server] <http://www.canonical.com/content/ubuntu-server-1210-all-you-need-cloud>
- [Wiki] <http://en.wikipedia.org/wiki/Wiki>

6.3 TRAC - An Overview

Next pages are screenshots of OSCIED Project's TRAC Environment to give you an overview of the functionalities that helped me during this Thesis.

Note: The screenshots are clickable (sorry for the readers who printed my report) !

6.3.1 Roadmap

Here you can see the milestones with the *estimated* progress (as every ticket has the same *weight* and they cannot be *partially* implemented). I noticed that the milestone *Some preliminary documentation* wasn't closed and I updated the status of tickets *5 weeks too late* ;-)

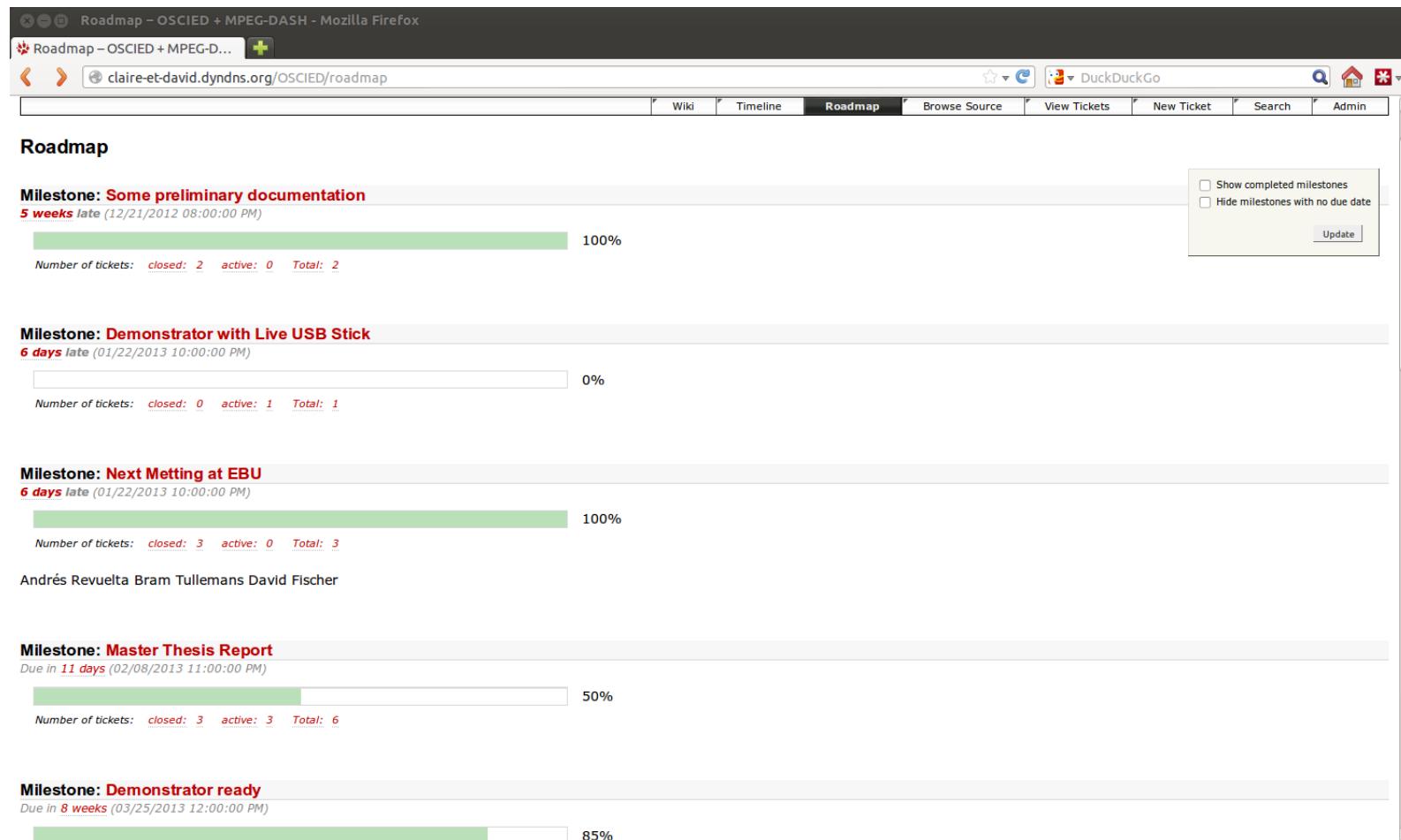


Figure 6.1: OSCIED Project's Roadmap

6.3.2 Source Browser

Here you can browse the Subversion repository of the project and you can travel back in time by viewing any of the thousand of revisions (not this is not a binary string 0b1110 but really the number 1'110) !

The screenshot shows a web browser window displaying the source code repository for the OSCIED project. The URL is <http://claire-et-david.dyndns.org/OSCIED/browser#components/orchestra/charm>. The page title is "Browse Source". The main content area displays a table of files with columns for Name, Size, Rev, Age, Author, and Last Change. The table is sorted by Name. A red banner at the top of the table header says "source: @ 1110". The table includes rows for various files and sub-directories under the "orchestra/charm" path, such as config.json, config.yaml, copyright, metadata.yaml, orchestra.py, revision, charm_juju, references, tools, publisher, and storage. The "orchestra.py" file is highlighted with a light green background.

| Name | Size | Rev | Age | Author | Last Change |
|---|-----------|-----------|----------|---------------|--|
| ↳ administration | | 263 | 5 months | david.fischer | some Ideas added |
| ↳ components | | 1108 | 17 hours | david.fischer | header updated |
| ↳ cloud | | 1108 | 17 hours | david.fischer | header updated |
| ↳ dash | | 965 | 11 days | david.fischer | latest Import of logiclibs ubuntu utils |
| ↳ FIMS | | 1108 | 17 hours | david.fischer | header updated |
| ↳ juju | | 1087 | 3 days | david.fischer | avoid embedding real password into public documentation ! |
| ↳ orchestra | | 1108 | 17 hours | david.fischer | header updated |
| ↳ charm | | 1108 | 17 hours | david.fischer | header updated |
| ↳ hooks | | 812 | 3 weeks | david.fischer | source path -> charm + reordering of orchestra source code |
| ↳ hooks_lib | | 1108 | 17 hours | david.fischer | header updated |
| ↳ lib | | 1101 | 40 hours | david.fischer | maybe the initial demo version ! |
| ↳ templates | | 879 | 2 weeks | david.fischer | disabled router added |
| config.json | 231 bytes | 896 | 2 weeks | david.fischer | hooks + config : nodes secret added Media.py : status field added ... |
| config.yaml | 2.4 KB | 1087 | 3 days | david.fischer | avoid embedding real password into public documentation ! |
| copyright | | 752 bytes | 621 | david.fischer | install Juju |
| metadata.yaml | | 359 bytes | 746 | david.fischer | all hooks -> common.sh, config -> config.json, gluster relation -> storage ... |
| orchestra.py | 30.0 KB | 1101 | 40 hours | david.fischer | maybe the initial demo version ! |
| revision | 5 bytes | 1056 | 7 days | david.fischer | Fix JuJu? LXC bug with grub-pc and memtest86+ |
| ↳ charm_juju | | 1108 | 17 hours | david.fischer | header updated |
| ↳ references | | 811 | 3 weeks | david.fischer | interesting article |
| ↳ tools | | 810 | 3 weeks | david.fischer | celery ! |
| ↳ publisher | | 1108 | 17 hours | david.fischer | header updated |
| ↳ storage | | 1108 | 17 hours | david.fischer | header updated |
| claire-et-david.dyndns.org/OSCIED/browser/components/orchestra/charm/orchestra.py | | 1108 | 17 hours | david.fischer | header updated |

Figure 6.2: OSCIED Project's Source Browser

6.3.3 Source Diff Tool

Here you can compare any versions of any files under [Version Control System](#). This feature is linked to one of the most interesting and intensively used feature of [Subversion](#) always used before *committing* anything. Useful for any developer to recall itself what he does in order to write the message of the *commit*.

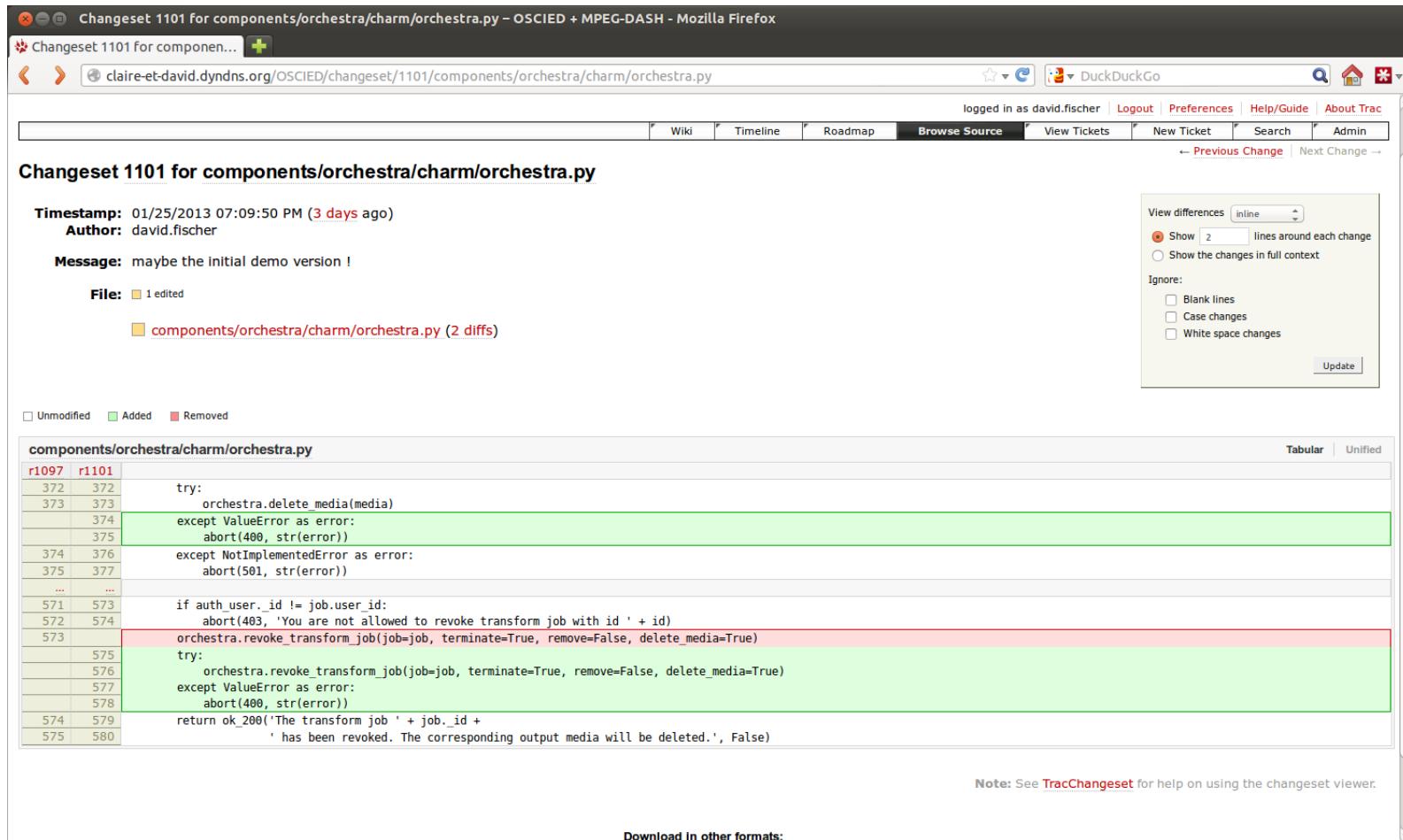


Figure 6.3: OSCIED Project's Source Diff Tool

6.3.4 Report : Active Tickets by Milestone (w/o reference)

Here you can browse through the pending tickets in order to select the next task you will work on or to show to your boss that the demonstrator is on the way to be ready on time.

Milestone Demonstrator ready (9 matches)

| Ticket | Summary | Component | Version | Type | Owner | Status | Created |
|--------|---|--------------------|---------|--------|-----------------|----------|------------|
| #181 | Web UI : Virtual Filenames with Whitespaces = Unable to download | Web User Interface | | defect | michael.fischer | assigned | 01/21/2013 |
| #147 | Publisher : Implement publisher API -> celery | Orchestra | | task | david.fischer | assigned | 01/13/2013 |
| #187 | Orchestra + Web UI : Show workers name (+queue) in jobs views | Orchestra | | task | david.fischer | new | 01/22/2013 |
| #171 | Storage : Mounting of running storage (amazon) from local client (home) failed | Storage | | defect | david.fischer | new | 01/20/2013 |
| #137 | Testing : Implement automation of tests (part 1) | Development | | task | david.fischer | assigned | 01/01/2013 |
| #140 | Scalability : Verify that increasing/decreasing units works | Development | | task | david.fischer | new | 01/04/2013 |
| #142 | Security : Minimum for a demo useable over a public Cloud | Development | | task | david.fischer | assigned | 01/08/2013 |
| #146 | Jobs hooks : send a mail / something else | Orchestra | | task | david.fischer | assigned | 01/13/2013 |
| #126 | WebUI : Apache could not reliably determin the server's fully qualified domain name | Web User Interface | | defect | david.fischer | new | 12/28/2012 |

Milestone Demonstrator with Codem Interface (2 matches)

| Ticket | Summary | Component | Version | Type | Owner | Status | Created |
|--------|--|-----------|---------|--------|-------|--------|------------|
| #11 | Exception thrown by codem-transcode | Encoder | | defect | david | new | 09/25/2012 |
| #8 | Check if codem can be useful in this project | Encoder | | task | david | new | 09/25/2012 |

Milestone Demonstrator with DASH Encoder (3 matches)

| Ticket | Summary | Component | Version | Type | Owner | Status | Created |
|--------|---|-----------|---------|-------------|-------|----------|------------|
| #1 | Exception thrown by dashRun -> dashEncoder | Encoder | | defect | david | assigned | 09/15/2012 |
| #7 | Read DASHEncoder source code | Encoder | | task | david | new | 09/19/2012 |
| #5 | dashCompile : avoid asking for DASHEncoder installation | MPEG-DASH | | enhancement | david | new | 09/19/2012 |

Milestone Demonstrator with Easy Private-Cloud Provisionning in Orchestra (3 matches)

| Ticket | Summary | Component | Version | Type | Owner | Status | Created |
|--------|--|-----------|---------|------|-------|--------|------------|
| #62 | Provisioning : Implement RabbitMQ -> PXE/REST provisioning | Agent | | task | david | new | 11/22/2012 |
| #98 | Provisioning : Implement provision API -> RabbitMQ | Orchestra | | task | david | new | 12/19/2012 |

Figure 6.4: OSCIED Project's Active Tasks Tickets

6.3.5 Report : All Tickets by Milestone (Including closed, w/o reference)

Here you can browse through all tickets, including the ones which are specified as *closed*. A not so useful report except for this Thesis.

The screenshot shows a Mozilla Firefox browser window with two tables of tickets from the OSCIED project:

Demonstrator with Codem Interface (2 matches)

| Ticket | Summary | Component | Status | Resolution | Version | Type | Priority | Owner | Modified |
|--------|--|-----------|--------|------------|---------|--------|----------|-------|------------|
| #11 | Exception thrown by codem-transcode | Encoder | new | | | defect | blocker | david | 01/21/2013 |
| #8 | Check if codem can be useful in this project | Encoder | new | | | task | minor | david | 01/21/2013 |

Demonstrator ready (37 matches)

| Ticket | Summary | Component | Status | Resolution | Version | Type | Priority | Owner | Modified |
|--------|---|---------------------|----------|------------|---------|--------|----------|-----------------|----------|
| #147 | Publisher : Implement publisher API -> celery | Orchestra | assigned | | | task | critical | david.fischer | 0 |
| #181 | Web UI : Virtual Filenames with Whitespaces = Unable to download | Web User Interface | assigned | | | defect | critical | michael.fischer | 0 |
| #187 | Orchestra + Web UI : Show workers name (+queue) in jobs views | Orchestra | new | | | task | critical | david.fischer | 0 |
| #137 | Testing : Implement automation of tests (part 1) | Development | assigned | | | task | major | david.fischer | 0 |
| #140 | Scalability : Verify that increasing/decreasing units works | Development | new | | | task | major | david.fischer | 0 |
| #142 | Security : Minimum for a demo useable over a public Cloud | Development | assigned | | | task | major | david.fischer | 0 |
| #146 | Jobs hooks : send a mail / something else | Orchestra | assigned | | | task | major | david.fischer | 0 |
| #171 | Storage : Mounting of running storage (amazon) from local client (home) failed | Storage | new | | | defect | major | david.fischer | 0 |
| #126 | WEBUI : Apache could not reliably determin the server's fully qualified domain name | Web User Interface | new | | | defect | trivial | david.fischer | 0 |
| #107 | Medias : Implement medias form -> Orchestra API | Web User Interface | closed | fixed | | task | major | michael.fischer | 0 |
| #134 | Profiles : Implement profile form -> Orchestra API | Web User Interface | closed | fixed | | task | major | michael.fischer | 0 |
| #148 | Publisher : Implement publisher form -> Orchestra API | Web User Interface | closed | fixed | | task | major | michael.fischer | 0 |
| #170 | Orchestra : Reject PUBLISHED medias as input of publish jobs | Orchestra | closed | fixed | | task | blocker | david.fischer | 0 |
| #188 | Orchestra + Web UI : Set medias status = DELETED Instead of removing it from database | Orchestra | closed | fixed | | task | critical | david.fischer | 0 |
| #189 | Orchestra + Web UI : Disable revoke if a job is finished | Orchestra | closed | fixed | | task | critical | david.fischer | 0 |
| #112 | Transform : Implement transform form -> Orchestra API | Web User Interface | closed | fixed | | task | major | michael.fischer | 0 |
| #165 | Web UI : Uses GET /transform,publisher)/queue | Web User Interface | closed | fixed | | task | blocker | david.fischer | 0 |
| #46 | Change TRAC service NAT port (change from 444 -> 80,443,8080 or ...) to bypass FWs | Project Admin / EBU | closed | fixed | | defect | critical | david | 0 |
| #13 | Development machine setup | Project Admin / EBU | closed | fixed | | task | major | david | 0 |
| #168 | Orchestra : Send 415 if ffmpeg is unable to detect media duration | Orchestra | closed | duplicate | | task | critical | david.fischer | 0 |
| #89 | Transform : Implement transform RabbitMQ -> system call to ffmpeg | Transform | closed | fixed | | task | major | david.fischer | 0 |
| #164 | Orchestra : Add GET /(transform,publish,unpublish)/queue as API methods | Orchestra | closed | fixed | | task | blocker | david.fischer | 0 |
| #166 | Orchestra : Reject PUBLISHED medias as input of publish jobs | Orchestra | closed | fixed | | task | critical | david.fischer | 0 |
| #163 | Orchestra : Reject PENDING medias as input of jobs | Orchestra | closed | fixed | | task | critical | david.fischer | 0 |
| #105 | Publisher : Implement publisher RabbitMQ -> Storage In to /www/D_Apache2 | Publisher | closed | fixed | | task | major | david.fischer | 0 |

Figure 6.5: OSCIED Project's All Tasks Tickets

6.3.6 Report : Reference Tickets by Component

Here you can browse through my *online* bibliography. This is actually a set of the most interesting resources I founded during my investigations, grouped by topic.

The screenshot shows a Mozilla Firefox browser window displaying a Trac report titled '{10} Reference Tickets by Component'. The URL is 'claire-et-david.dyndns.org/OSCIED/report/10'. The top navigation bar includes links for Wiki, Timeline, Roadmap, Browse Source, View Tickets (selected), New Ticket, Search, and Admin. Below the navigation is a search bar with 'Available Reports' and 'Custom Query' options, and a 'Max items per page' dropdown set to 100 with an 'Update' button.

{10} Reference Tickets by Component (36 matches)

- List all reference tickets.
- Sort reference tickets by component.

Component Cloud (12 matches)

| Ticket | Summary | Owner | Created |
|--------|---|---------------|------------|
| #14 | Virtualization technologies (KVM, LXC) | david | 09/27/2012 |
| #17 | Ubuntu Server certified hardware : Dell | david | 09/27/2012 |
| #20 | Multiple CPU Systems Benchmark | david | 09/27/2012 |
| #34 | Cloud libraries (libcloud, jClouds) | david | 10/03/2012 |
| #37 | OpenStack (API) Documentation & Tutorials | david | 10/03/2012 |
| #53 | Tutorial : How-to create a Live Clonezilla USB Stick | david | 11/09/2012 |
| #54 | Tutorial : How-to create a NFS share to store Clonezilla's Images | david | 11/09/2012 |
| #55 | Tutorial : How-to use Clonezilla | david | 11/09/2012 |
| #64 | OSCIED appendix : EBU's OpenStack setup | david | 11/28/2012 |
| #130 | Network install of Linux | david.fischer | 12/30/2012 |
| #131 | Amazon Web Services | david.fischer | 12/30/2012 |
| #190 | OpenStack : First SUCCESS ! | david.fischer | 01/22/2013 |

Component Development (2 matches)

| Ticket | Summary | Owner | Created |
|--------|--|-------|------------|
| #24 | Boost C++ libraries tutorial | david | 10/02/2012 |
| #83 | Developing Web Services (SOAP/RESTful) | david | 12/13/2012 |

Component FIMS (1 match)

| Ticket | Summary | Owner | Created |
|--------|---|-------|------------|
| #43 | EBU - FIMS Framework for Interoperable Media Services | david | 10/04/2012 |

Figure 6.6: OSCIED Project's All Reference Tickets

6.3.7 Reference Ticket #120 : Developing with Python

Here you can see the content of one of the reference tickets. As you can see most of them will take at least one A4 page if embedded in my Thesis ...

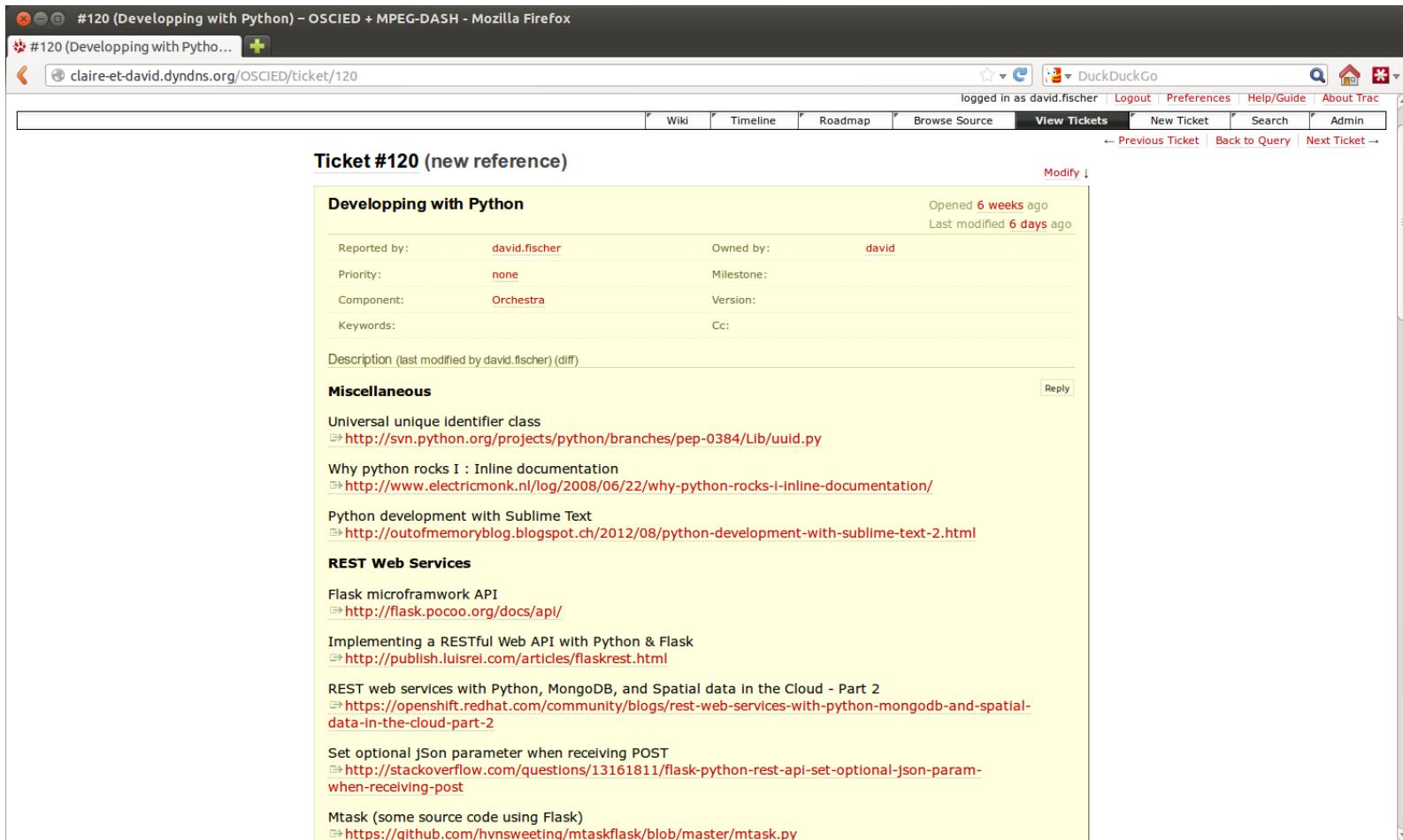


Figure 6.7: OSCIED Project's Reference Ticket #120

6.4 OSCIED - Source Code Licensing

6.4.1 Common HEADER (License)

```

1  #!/usr/bin/env bash
2
3  #####*
4  #          OPEN-SOURCE CLOUD INFRASTRUCTURE FOR ENCODING AND DISTRIBUTION : SCRIPTS
5  #
6  # Authors   : David Fischer
7  # Contact   : david.fischer.ch@gmail.com / david.fischer@hesge.ch
8  # Project   : OSCIED (OS Cloud Infrastructure for Encoding and Distribution)
9  # Copyright : 2012 OSCIED Team. All rights reserved.
10 #####
11 #
12 # This file is part of EBU/UER OSCIED Project.
13 #
14 # This project is free software: you can redistribute it and/or modify it under the terms of the
15 # GNU General Public License as published by the Free Software Foundation, either version 3 of the
16 # License, or (at your option) any later version.
17 #
18 # This project is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without
19 # even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
20 # See the GNU General Public License for more details.
21 #
22 # You should have received a copy of the GNU General Public License along with this project.
23 # If not, see <http://www.gnu.org/licenses/>
24 #
25 # Retrieved from:
26 #   svn co https://claire-et-david.dyndns.org/prog/OSCIED

```

6.5 OSCIED - Orchestra RESTful API

See Also:

Automated documentation. Please see [Sphinx_autodoc](#) and [Sphinx_autoflask](#) !

6.5.1 Utility Methods

```
orchestra.requires_auth(request, allow_root=False, allow_node=False, allow_any=False,
                        role=None, id=None, mail=None)
```

This method implements Orchestra's RESTful API authentication logic. Here is ensured that an access to a method of the API is filtered based on rules (this method's parameters). HTTP user agent must authenticate through HTTP basic access authentication. The username must be user's email address and password must be user's secret. This not apply for system-users like root or node as they do not have any e-mail address.

Warning: Username and password are passed as plaintext, SSL/TLS is one of the way to improve security although this was not tested during my thesis.

This method will abort request with HTTP 401 error if HTTP user agent doesn't authenticate.

Parameters

- **request** – the request itself, credentials are retrieved from request authorization header
- **allow_root** – if set to *True* root system-user will be allowed
- **allow_node** – if set to *True* node system-user will be allowed
- **allow_any** – if set to *True* any authenticated user will be allowed
- **role** – if set to <name>, any user will “name” role set to *True* will be allowed
- **id** – if set to <uuid>, any user with _id equal to “uuid” will be allowed

- **mail** – if set to <mail>, any user with mail equal to “mail” will be allowed

This method will abort request with HTTP 403 error if none of the following conditions are met.

Example:

```
# Allow any authenticated user
@app.route('/my/example/route', methods=['GET'])
def api_my_example_route():
    if request.method == 'GET':
        auth_user = requires_auth(request=request, allow_any=True)
        ...
    return ok_200('my return value', True)

# Allow root system-user or any user with admin attribute set
@app.route('/my/restricted/route', methods=['GET'])
def api_my_restricted_route():
    if request.method == 'GET':
        auth_user = requires_auth(request=request, allow_root=True, allow_role='admin')
        ...
    return ok_200('my return value', True)
```

6.5.2 The RESTful API

GET /transform/profile/count

Return profiles count.

Example request:

```
GET /transform/profile/count HTTP/1.1
Host: somewhere.com
Header: nabil@oscied.org:oscied
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json
```

```
{"status": 200, "value": 100}
```

Allowed Any user

Status Codes

- 200 – OK
- 401 – Authenticate.
- 403 – Authentication Failed.

GET /transform/job/count

Return transform jobs count.

Example request:

```
GET /transform/job/count HTTP/1.1
Host: somewhere.com
Header: marylene@oscied.org:oscied
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json
```

```
{"status": 200, "value": 15260}
```

Allowed Any user

Status Codes

- 200 – OK
- 401 – Authenticate.
- 403 – Authentication Failed.

GET /transform/job/HEAD

Return an array containing the transform jobs serialized as JSON.

The transform jobs attributes are appended with the Celery's `async_result` of the jobs.

Example request:

```
GET /transform/job/HEAD HTTP/1.1
Host: somewhere.com
Header: thomas@oscied.org:oscied
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "status": 200,
  "value": [{"_id": "...", "...": "..."}, {"_id": "..."}]
```

Allowed Any user

Status Codes

- 200 – OK
- 401 – Authenticate.
- 403 – Authentication Failed.

GET /publish/job/count

Return publish jobs count.

Example request:

```
GET /publish/job/count HTTP/1.1
Host: somewhere.com
Header: sophie@oscied.org:oscied
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json
```

```
{"status": 200, "value": 3904}
```

Allowed Any user

Status Codes

- 200 – OK
- 401 – Authenticate.
- 403 – Authentication Failed.

GET /publish/job/HEAD

Return an array containing the publish jobs serialized as JSON.
The publish jobs attributes are appended with the Celery's `async` result of the jobs.

Example request:

```
GET /publish/job/HEAD HTTP/1.1
Host: somewhere.com
Header: antonin@oscied.org:oscied
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json
```

```
{
  "status": 200,
  "value": [{"_id": "...", "...": "..."}, {"_id": "..."}]
}
```

Allowed Any user

Status Codes

- 200 – OK
- 401 – Authenticate.
- 403 – Authentication Failed.

POST /transform/callback

This method is called by transform workers when they finish their work.
If job is successful, the orchestrator will set media's status to READY. Else, the orchestrator will append `error_details` to `statistic` attribute of job.
The media will be deleted if job failed (even the worker already take care of that).

Example request:

```
POST /transform/callback HTTP/1.1
Host: somewhere.com
Header: node:abcdef
Accept: application/json
Content-Type: application/json

{
  "job_id": "1b96dcd6-7460-11e2-a06d-3085a9accc47",
  "status": "SUCCESS"
}
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{"status": 200, "value": "Your work is much appreciated, thanks !!"}
```

Allowed Node

Query Parameters

- `job_id` – Job's id (required)
- `status` – Job's status (SUCCESS) or error's details (required)

Status Codes

- 200 – OK
- 400 – Key key not found. or on type or value error
- 401 – Authenticate.
- 403 – Authentication Failed.
- 404 – No transform job with id `id`.
- 404 – Unable to find output media with id `id`.
- 415 – Requires (valid) json content-type.

GET /transform/profile

Return an array containing the transform profiles serialized to JSON.

Example request:

```
GET /transform/profile HTTP/1.1
Host: somewhere.com
Header: michel@oscied.org:oscied
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "status": 200,
  "value": [{"_id": "...", "...": "..."}, {"_id": "..."}]}
```

Allowed Any user

Status Codes

- 200 – OK
- 401 – Authenticate.
- 403 – Authentication Failed.

POST /transform/profile

Add a transform profile.

The transform profile's `encoder_string` attribute can be a keyword like :

- `copy` to bypass FFmpeg and do a simple file block copy ;
- ... or it can be a valid string containing FFmpeg options (see example) ;

Example request:

```
POST /transform/profile HTTP/1.1
Host: somewhere.com
Header: daniel@oscied.org:oscied
Accept: application/json
Content-Type: application/json

{
  "title": "To MP4",
  "description": "Convert to MP4 (container)",
  "encoder_string": "-acodec copy -vcodec copy -f mp4"
}
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "status": 200,
  "value": {
    "_id": "c316ff1a-74f8-11e2-82d4-3085a9acd33",
    "title": "To MP4",
    "description": "Convert to MP4 (container)",
    "encoder_string": "-acodec copy -vcodec copy -f mp4"
  }
}
```

Allowed Any user

Query Parameters

- `title` – New profile's title (required)
- `description` – New profile's description (required)
- `encoder_string` – New profile's (FFmpeg) encoder string (required)

Status Codes

- **200** – OK
- **400** – Key key not found. *or* on type or value error
- **400** – Duplicate transform profile title profile.
- **401** – Authenticate.
- **403** – Authentication Failed.
- **415** – Requires (valid) json content-type.

GET /transform/queue

Return an array containing the transform queues serialized to JSON.

Example request:

```
GET /transform/queue HTTP/1.1
Host: somewhere.com
Header: marco@oscied.org:oscied
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "status": 200,
  "value": ["transform_amazon", "transform_ebu_geneva"]
```

Allowed Any user

Status Codes

- **200** – OK
- **401** – Authenticate.
- **403** – Authentication Failed.

GET /unpublish/queue

Return an array containing the publish queues.

Example request:

```
GET /publish/queue HTTP/1.1
Host: somewhere.com
Header: jean-claude@oscied.org:oscied
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json
```

```
{
  "status": 200,
  "value": ["publisher_london", "publisher_ebu_geneva"]}
```

Allowed Any user

Status Codes

- **200** – OK

- 401 – Authenticate.
- 403 – Authentication Failed.

GET /publisher/queue

Return an array containing the publish queues.

Example request:

```
GET /publisher/queue HTTP/1.1
Host: somewhere.com
Header: jean-claude@oscied.org:oscied
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "status": 200,
  "value": ["publisher_london", "publisher_ebu_geneva"]
}
```

Allowed

Any user

Status Codes

- 200 – OK
- 401 – Authenticate.
- 403 – Authentication Failed.

GET /transform/job

Return an array containing the transform jobs serialized to JSON.

The transform jobs attributes are appended with the Celery's `async` result of the jobs.

All `thing_id` fields are replaced by corresponding `thing`. For example `user_id` is replaced by `user`'s data.

Example request:

```
GET /transform/job HTTP/1.1
Host: somewhere.com
Header: antoinette@oscied.org:oscied
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "status": 200,
  "value": [{"_id": "...", "...": "..."}, {"_id": "..."}]
}
```

Allowed

Any user

Status Codes

- 200 – OK
- 401 – Authenticate.
- 403 – Authentication Failed.

POST /transform/job

Launch a transform job.

Any user can launch a transform job using any media as input and any transform profile. This is linked to media and transform profile API methods access policies.

The output media is registered to the database with the PENDING status and the `parent_id` field is set to input media's `id`. This permit to know relation between medias !

The orchestrator will automatically add `add_date` to statistic.

Note: Interesting enhancement would be to :

- Schedule obs by specifying start time (...);
- Handle the registration of jobs related to PENDING medias ;

Example request:

```
POST /transform/job HTTP/1.1
Host: somewhere.com
Header: tabby@bernex.ch:miaow
Accept: application/json
Content-Type: application/json

{
  "media_in_id": "a396fe66-74ee-11e2-89ad-3085a9accbb8",
  "profile_id": "c316ff1a-74f8-11e2-82d4-3085a9accd33",
  "virtual_filename": "avatar.mp4",
  "metadata": {"title": "Avatar (1080p)" },
  "queue": "transform_ebu-geneva"
}
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{"status": 200, "value": "ea9088f0-74f8-11e2-b780-3085a9accb2a"}
```

Allowed Any user**Query Parameters**

- **media_in_id** – New job input media's id (required)
- **profile_id** – New job profile's id (required)
- **virtual_filename** – New job output media's virtual_filename (required)
- **metadata** – New job output media's metadata (required)
- **queue** – The transform queue used to route the new job (required)

Status Codes

- **200** – OK
- **400** – Key key not found. or on type or value error
- **400** – Unable to transmit job to workers of queue queue.
- **401** – Authenticate.
- **403** – Authentication Failed.
- **404** – No user with id id.
- **404** – No media with id media_in_id.
- **404** – No profile with id profile_id.
- **404** – No transform queue with name queue.
- **415** – Required (valid) json content-type.
- **501** – Cannot launch the job, input media status is status.

POST /publish/callback

This method is called by publisher workers when they finish their work.

If job is successful, the orchestrator will update `publish_uri` attribute of job, set media's status to SUCCESS and update `public_uris` attribute. Else, the orchestrator will append `error_details` to `statistic` attribute of job.

Example request:

```
POST /publish/callback HTTP/1.1
Host: somewhere.com
Header: node:abcdef
Accept: application/json
Content-Type: application/json

{
  "job_id": "1b96dc6-7460-11e2-a06d-3085a9accb47",
  "publish_uri": "http://<address>/medias/<user_id>/<media_id>/Project_London_trailer_2009.mp4",
  "status": "SUCCESS"
}
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{"status": 200, "value": "Your work is much appreciated, thanks !"}
```

Allowed Node**Query Parameters**

- **job_id** – Job's id (required)
- **publish_uri** – Publication URI of the media (required)
- **status** – Job's status (SUCCESS) or error's details (required)

Status Codes

- **200** – OK
- **400** – Key key not found. or on type or value error
- **401** – Authenticate.
- **403** – Authentication Failed.
- **404** – No publish job with id id.
- **404** – Unable to find media with id id.
- **415** – Requires (valid) json content-type.

GET /publish/queue

Return an array containing the publish queues.

Example request:

```
GET /publish/queue HTTP/1.1
Host: somewhere.com
Header: Jean-Claude@oscied.org:oscied
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
    "status": 200,
    "value": ["publisher_london", "publisher_ebu_geneva"]
```

Allowed Any user

Status Codes

- 200 – OK
- 401 – Authenticate.
- 403 – Authentication Failed.

GET /publish/job

Return an array containing the publish jobs serialized to JSON.

The publish jobs attributes are appended with the Celery's `async` result of the jobs.

All `thing_id` fields are replaced by corresponding `thing`. For example `user_id` is replaced by `user`'s data.

Example request:

```
GET /publish/job HTTP/1.1
Host: somewhere.com
Header: melanie@oscied.org:oscied
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
    "status": 200,
    "value": [{"_id": "...", "...": "..."}, {"_id": "..."}]
```

Allowed Any user

Status Codes

- 200 – OK
- 401 – Authenticate.
- 403 – Authentication Failed.

POST /publish/job

Launch a publish job.

Any user can launch a publish job using any media as input. This is linked to media API methods access policy.
The orchestrator will automatically add `add_date` to statistic.

Note: Interesting enhancements would be to :

- Schedule jobs by specifying start time (...)
- Handle the registration of jobs related to PENDING medias
- Permit to publish a media on more than one (1) publication queue
- Permit to unpublish a media via a unpublish (broadcast) message

Example request:

```
POST /publish/job HTTP/1.1
Host: somewhere.com
Header: tabby@bernex.ch:miaow
Accept: application/json
Content-Type: application/json

{
    "media_id": "a396fe66-74ee-11e2-89ad-3085a9accbb8",
    "queue": "publish_london"
}
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
    "status": 200,
    "value": "73abcf7e-74ef-11e2-9322-3085a9accc9b9"
```

Allowed Any user

Query Parameters

- `media_id` – New job input media's id (required)
- `queue` – The publish queue used to route job (required)

Status Codes

- 200 – OK
- 400 – Key key not found. or on type or value error
- 400 – Unable to transmit job to workers of queue `queue`.
- 401 – Authenticate.
- 403 – Authentication Failed.
- 404 – No user with id `id`.

- **404** – No media with id media_id.
- **404** – No publish queue with name queue.
- **415** – Required (valid) json content-type.
- **501** – Cannot launch the job, input media status is status.
- **501** – Cannot launch the job, input media will be published by another job with id id.

GET /media/count

Return medias count.

Example request:

```
GET /media/count HTTP/1.1
Host: somewhere.com
Header: tewfiq@oscied.org:oscied
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{"status": 200, "value": 8000}
```

Allowed Any user

Status Codes

- **200** – OK
- **401** – Authenticate.
- **403** – Authentication Failed.

GET /media/HEAD

Return an array containing the medias serialized to JSON.

Example request:

```
GET /media/HEAD HTTP/1.1
Host: somewhere.com
Header: andres@oscied.org:oscied
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "status": 200,
  "value": [{"_id": "...", "...": "..."}, {"_id": "..."}]
```

Allowed Any user

Status Codes

- **200** – OK
- **401** – Authenticate.
- **403** – Authentication Failed.

GET /user/login

Return authenticated user serialized to JSON if authentication passed (without secret field).

This method is useful for WebUI to simulate stateful login scheme and get informations about the user.

Note: This is kind of duplicate with API's GET /user/id/id method ...

Example request:

```
GET /user/login HTTP/1.1
Host: somewhere.com
Header: tabby@bernex.ch:miaow
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json
```

```
{
  "status": 200,
  "value": {
    "_id": "c4daa8a6-6be4-11e2-ae91-3085a9accb47",
    "first_name": "Tabby",
    "last_name": "Fischer",
    "name": "Tabby Fischer",
    "mail": "tabby@bernex.ch",
    "admin_platform": false
  }
}
```

Allowed Any user

Status Codes

- **200** – OK
- **401** – Authenticate.
- **403** – Authentication Failed.

GET /user/count

Return users count.

Example request:

```
GET /user/count HTTP/1.1
Host: somewhere.com
Header: bram@oscied.org:oscied
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{"status": 200, "value": 5000}
```

Allowed Root and any user**Status Codes**

- 200 – OK
- 401 – Authenticate.
- 403 – Authentication Failed.

GET /index

Return an about string.

This method is actually used by Orchestra charm's hooks to check API's status.

Example request:

```
GET / HTTP/1.1
Host: example.com
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json
```

```
{
  "status": 200, "value":
    "Orchestra : EBU's OSCIED Orchestrator by David Fischer 2012\n"
```

Allowed Any user (including unauthenticated)**Status Codes**

- 200 – OK

POST /flush

Flush Orchestrator's database.

This method is useful for test/development purposes.

Example request:

```
POST /flush HTTP/1.1
Host: example.com
Header: root:password
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json
```

```
{"status": 200, "value": "Orchestra database flushed !"}  

```

Allowed Only root**Status Codes**

- 200 – OK
- 401 – Authenticate.
- 403 – Authentication Failed.

GET /media

Return an array containing the medias serialized to JSON.

All thing_id fields are replaced by corresponding thing. For example user_id is replaced by user's data.

Example request:

```
GET /media HTTP/1.1
Host: somewhere.com
Header: nabil@oscied.org:oscied
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json
```

```
{
  "status": 200,
  "value": [{"_id": "...", "...": "..."}, {"_id": "..."}]
```

Allowed Any user**Status Codes**

- 200 – OK
- 401 – Authenticate.
- 403 – Authentication Failed.

POST /media

Add a media.

This method handle registration of already uploaded media to the shared storage. For example, the WebUI will upload a media to uploads path **before** registering it with this method.

Medias in the shared storage are renamed with the following convention:

```
storage_root/medias/user_id/media_id
```

When published or downloaded, media will be renamed to `virtual_filename`. Spaces () are not allowed and they will be converted to underscores (_).

Media's metadata must contain any valid JSON string. Only the `title` key is required. The orchestrator will automatically add `add_date` and `duration` to metadata.

Note: Registration of external media (aka. `http://`) will be an interesting improvement.

Example request:

```
POST /media HTTP/1.1
Host: somewhere.com
Header: d@f.com:oscied
Accept: application/json
Content-Type: application/json

{
  "uri": "glusterfs://<address>/medias_volume/uploads/
          Project London - Official Trailer [2009].mp4",
  "virtual_filename": "Project_London_trailer_2009.mp4",
  "metadata": {
    "title": "Project London - Official Trailer (2009)"
  }
}
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "status": 200,
  "value": {
    "_id": "96590fdc-74f8-11e2-8c58-3085a9acc651",
    "user_id": "1298f206-74f8-11e2-9b82-3085a9acc11b",
    "parent_id": null,
    "uri": "glusterfs://<address>/medias_volume/medias/
           <user_id>/<media_id>",
    "public_uris": null,
    "virtual_filename": "Project_London_trailer_2009.mp4",
    "metadata": {
      "add_date": "2013-02-02 14:05",
      "duration": "00:02:44.88", "size": 54871886,
      "title": "Project London - Official Trailer (2009)"
    },
    "status": "READY"
  }
}
```

Allowed Any user can do that

Query Parameters

- `uri` – Media's source URI, actually only shared storage's URI are handled (required)
- `virtual_filename` – Media's filename when downloaded or published (required)
- `metadata` – JSON string containing metadatas about the media (required)

Status Codes

- **200** – OK
- **400** – on type or value error
- **400** – Key key not found.
- **400** – The media uri `uri` is already used by another media.
- **400** – Title key is required in media metadata.
- **401** – Authenticate.
- **403** – Authentication Failed.
- **404** – An error occurred : OSerror
- **415** – Requires (valid) json content-type.
- **501** – FIXME Add of external uri not implemented.

GET /user

Return an array containing the users serialized to JSON (without `secret` fields).
Example request:

```
GET /user HTTP/1.1
Host: somewhere.com
Header: peter@oscied.org:oscied
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "status": 200,
```

```
"value": [{"_id": "...", "...": "..."}, {"_id": "..."}]}
```

Allowed Root and user with admin_platform attribute set

Status Codes

- **200** – OK
- **401** – Authenticate.
- **403** – Authentication Failed.

POST /user

Add an user.

Example request:

```
POST /user HTTP/1.1
Host: somewhere.com
Header: kouadi@oscied.org:oscied
Accept: application/json
Content-Type: application/json

{
  "first_name": "Laurent",
  "last_name": "Nicolet",
  "mail": "laurent@comique.ch",
  "secret": "genevois_style",
  "admin_platform": false
}
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "status": 200,
  "value": {
    "_id": "8bda488c-6be8-11e2-89b7-3085a9accb47",
    "first_name": "Laurent",
    "last_name": "Nicolet",
    "name": "Laurent Nicolet",
    "mail": "laurent@comique.ch",
    "admin_platform": false
  }
}
```

Allowed Root and user with admin_platform attribute set

Query Parameters

- **first_name** – New user's first name (required)

- **last_name** – New user's last name (required)
- **mail** – New user's email address (required)
- **secret** – New user's secret (required)
- **admin_platform** – New user's admin_platform (required)

Status Codes

- **200** – OK
- **400** – on type or value error
- **400** – Key key not found.
- **400** – The email address mail is already used by another user.
- **401** – Authenticate.
- **403** – Authentication Failed.
- **415** – Requires (valid) json content-type.

GET /

Return an about string.

This method is actually used by Orchestra charm's hooks to check API's status.

Example request:

```
GET / HTTP/1.1
Host: example.com
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "status": 200,
  "value":
    "Orchestra : EBU's OSCIED Orchestrator by David Fischer 2012\n"
```

Allowed Any user (including unauthenticated)

Status Codes

- **200** – OK

GET /transform/job/id/(id) /HEAD

Return a transform job serialized to JSON.

The transform job attributes are appended with the Celery's `async_result` of the job.

Example request:

```
GET /transform/job/id/48c111c8-74f8-11e2-a7a8-3085a9acc6c4/HEAD HTTP/1.1
Host: somewhere.com
Header: edoardo@oscied.org:oscied
Accept: application/json
Content-Type: application/json
```

Example response:

Floating numbers are here with “” for autoflask to work !!

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
    "status": 200,
    "value": {
        "_id": "48c111c8-74f8-11e2-a7a8-3085a9acc6c4",
        "user_id": "4e8albce-74f3-11e2-9660-3085a9acce0b",
        "media_in_id": "a8a165b8-74f7-11e2-a59e-3085a9acc049",
        "media_out_id": "52ea73ac-74f3-11e2-afdb-3085a9acc5ff",
        "profile_id": "55da66d6-74f3-11e2-9dff-3085a9acce4e",
        "statistic": {
            "add_date": "2013-02-11 22:44",
            "start_date": "2013-02-11 22:44",
            "elapsed_time": "19.241864919662476",
            "eta_time": 0, "percent": 100,
            "media_in_size": 54871886, "media_in_duration": "00:02:44.88",
            "media_out_size": 25601528, "media_out_duration": "00:00:01.95"
        },
        "revoked": false,
        "status": "SUCCESS"
    }
}
```

Allowed Any user

Parameters

- **id** – id of job to get

Status Codes

- **200** – OK
- **401** – Authenticate.
- **403** – Authentication Failed.
- **404** – No transform job with id **id**.
- **415** – Wrong id format **id**.

GET /publish/job/**id**/(**id**)/**HEAD**

Return a publish job serialized to JSON.

The publish job attributes are appended with the Celery’s `async_result` of

the job.

Example request:

```
GET /publish/job/id/c697f528-74f7-11e2-96a3-3085a9accc5d/HEAD HTTP/1.1
Host: somewhere.com
Header: tabby@bernex.ch:miaow
Accept: application/json
```

Example response:

Floating numbers are here with “” for autoflask to work !!

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
    "status": 200,
    "value": {
        "_id": "c697f528-74f7-11e2-96a3-3085a9accc5d",
        "user_id": "4e8ec55e-74f7-11e2-8451-3085a9acc8e0b",
        "media_id": "a8a165b8-74f7-11e2-a59e-3085a9acc049",
        "publish_uri": "http://<publish_uri>/medias/<user_id>/<media_id>/Project_London_trailer_2009.mp4",
        "statistic": {
            "add_date": "2013-02-11 22:38",
            "start_date": "2013-02-11 22:38",
            "elapsed_time": "0.5068690776824951",
            "eta_time": 0, "percent": 100,
            "media_size": 54871886, "publish_size": 54871886,
            "pid": 18307, "hostname": "famille-local-oscied-publisher-0"
        },
        "revoked": false,
        "status": "SUCCESS"
    }
}
```

Allowed Any user

Parameters

- **id** – id of job to get

Status Codes

- **200** – OK
- **401** – Authenticate.
- **403** – Authentication Failed.
- **404** – No publish job with id **id**.

GET /transform/profile/**id**/(**id**)

Return a transform profile serialized to JSON.

Example request:

```
GET /transform/profile/id/c316ff1a-74f8-11e2-82d4-3085a9accd33 HTTP/1
Host: somewhere.com
Header: francois@oscied.org:oscied
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "status": 200,
  "value": {
    "_id": "c316ff1a-74f8-11e2-82d4-3085a9accd33",
    "title": "To MP4",
    "description": "Convert to MP4 (container)",
    "encoder_string": "-acodec copy -vcodec copy -f mp4"
  }
}
```

Allowed Any user**Parameters**

- **id** – id of profile to get

Status Codes

- 200 – OK
- 401 – Authenticate.
- 403 – Authentication Failed.
- 404 – No transform profile with id **id**.
- 415 – Wrong id format **id**.

PUT /transform/profile/id/ (*id*)

Update a transform profile.

Warning: All fields can be updated, maybe not a good idea (??)

Example request:

```
PUT /transform/profile/id/c316ff1a-74f8-11e2-82d4-3085a9accd33 HTTP/1
Host: somewhere.com
Header: dimitri@oscied.org:oscied
Accept: application/json
Content-Type: application/json

{"description": "Convert any container to MP4"}
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "status": 200,
  "value": "The transform profile \"To MP4\" has been updated."
}
```

Allowed Any user**param id** id of profile to get**query title** Transform profile's title (optional)**query description** Transform profile's description (optional)**query encoder_string** Transform profile's (FFmpeg) encoder string (optional)**statuscode 200** OK**statuscode 400** Key key not found. or on type or value error**statuscode 400** Duplicate transform profile title **profile**.**statuscode 401** Authenticate.**statuscode 403** Authentication Failed.**statuscode 404** No transform profile with id **id**.**statuscode 415** Wrong id format **id**.**statuscode 415** Requires (valid) json content-type.**PATCH /transform/profile/id/ (*id*)**

Update a transform profile.

Warning: All fields can be updated, maybe not a good idea (??)

Example request:

```
PUT /transform/profile/id/c316ff1a-74f8-11e2-82d4-3085a9accd33 HTTP/1
Host: somewhere.com
Header: dimitri@oscied.org:oscied
Accept: application/json
Content-Type: application/json

{"description": "Convert any container to MP4"}
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "status": 200,
  "value": "The transform profile \"To MP4\" has been updated."
```

```

}

Allowed Any user
param id id of profile to get
query title Transform profile's title (optional)
query description Transform profile's description (optional)
query encoder_string Transform profile's (FFmpeg) encoder string (optional)
statuscode 200 OK
statuscode 400 Key key not found. or on type or value error
statuscode 400 Duplicate transform profile title profile.
statuscode 401 Authenticate.
statuscode 403 Authentication Failed.
statuscode 404 No transform profile with id id.
statuscode 415 Wrong id format id.
statuscode 415 Requires (valid) json content-type.

DELETE /transform/profile/id/ (id)
Delete a transform profile.

```

Example request:

```
DELETE /transform/profile/id/c316ff1a-(...)-3085a9accd33 HTTP/1.1
Host: somewhere.com
Header: dimitri@oscied.org:oscied
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "status": 200,
  "value": "The transform profile \"To MP4\" has been deleted."
}
```

Allowed Any user**Parameters**

- **id** – id of profile to delete

Status Codes

- **200** – OK
- **401** – Authenticate.
- **403** – Authentication Failed.

- **404** – No transform profile with id id.

- **415** – Wrong id format id.

GET /transform/job/id/ (id)

Return a transform job serialized to JSON.

The transform job attributes are appended with the Celery's `async_result` of the job.

All `thing_id` fields are replaced by corresponding `thing`. For example `user_id` is replaced by user's data.

Example request:

```
GET /transform/job/id/ea9088f0-74f8-11e2-b780-3085a9accb2a HTTP/1.1
Host: somewhere.com
Header: claire@oscied.org:oscied
Accept: application/json
Content-Type: application/json
```

Example response:

Floating numbers are here with “” for autoflask to work !!

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "status": 200,
  "value": {
    "_id": "48c111c8-74f8-11e2-a7a8-3085a9acc6c4",
    "user": {
      "_id": "4e8ec55e-74f7-11e2-8451-3085a9acc8e0b",
      "first_name": "David",
      "last_name": "Fischer",
      "name": "David Fischer",
      "mail": "d@f.com",
      "admin_platform": true
    },
    "media_in": {
      "_id": "a8a165b8-74f7-11e2-a59e-3085a9acc049",
      "user_id": "4e8a1bce-74f3-11e2-9660-3085a9acce0b",
      "parent_id": null,
      "uri": "glusterfs://<address>/medias_volume/medias/
<user_id>/<media_id>",
      "public_uris": {
        "c697f528-74f7-11e2-96a3-3085a9accc5d":
        "http://10.0.3.254/medias/<user_id>/<media_id>/
          Project_London_trailer_2009.mp4"
      },
      "virtual_filename": "Project_London_trailer_2009.mp4",
      "metadata": {
        "add_date": "2013-02-11 22:37",
        "content_type": "video/mp4",
        "duration": 1234567890,
        "height": 1080,
        "width": 1920
      }
    }
  }
}
```

```

    "duration": "00:02:44.88", "size": 54871886,
    "title": "Project London - Official Trailer (2009)"
  },
  "status": "PUBLISHED"
},
"media_out": {
  "_id": "52ea73ac-74f3-11e2-afdb-3085a9acc5ff",
  "user_id": "4e8ec5e-74f7-11e2-8451-3085a9acc8e0b",
  "parent_id": "a8a165b8-74f7-11e2-a59e-3085a9acc049",
  "uri": "glusterfs://<address>/medias_volume/medias/
    <user_id>/<media_id>",
  "public_uris": null,
  "virtual_filename": "project_london.mp2",
  "metadata": {
    "add_date": "2013-02-11 22:44",
    "duration": "00:00:01.95", "size": 25601528,
    "title": "Project London MP2"
  }
  "status": "READY"
},
"profile": {
  "_id": "55da66d6-74f3-11e2-9dff-3085a9acce4e",
  "title": "To MP2",
  "description":
    "Convert video track to MPEG-2 format, copy audio track",
  "encoder_string":
    "-acodec copy -vcodec mpeg2video -f mpeg2video"
},
"statistic": {
  "add_date": "2013-02-11 22:44",
  "start_date": "2013-02-11 22:44",
  "elapsed_time": "19.241864919662476",
  "eta_time": 0, "percent": 100,
  "media_in_size": 54871886, "media_in_duration": "00:02:44.88",
  "media_out_size": 25601528, "media_out_duration": "00:00:01.95"
},
"revoked": false,
"status": "SUCCESS"
}
}

```

Allowed Any user**Parameters**

- **id** – id of job to get

Status Codes

- **200** – OK
- **401** – Authenticate.

- **403** – Authentication Failed.
- **404** – No transform job with id **id**.
- **415** – Wrong id format **id**.

DELETE /transform/job/id/(id)

Revoke a transform job.

This method do not delete jobs from jobs database but set `revoked` attribute in jobs database and broadcast revoke request to transform units with Celery. If the job is actually running it will be canceled. The output media will be deleted.

Example request:

```
DELETE /transform/job/id/ea9088f0-74f8-11e2-b780-3085a9acccb2a HTTP/1.1
Host: somewhere.com
Header: tabby@bernex.ch:miaow
Accept: application/json
Content-Type: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "status": 200,
  "value": "The transform job \"<job_id>\" has been revoked.
Corresponding output media will be deleted."
}
```

Allowed Only author of the job**Parameters**

- **id** – id of job to delete

Status Codes

- **200** – OK
- **400** – on value error
- **400** – Transform job **id** is already revoked !
- **400** – Cannot revoke a transform job with status `status`.
- **401** – Authenticate.
- **403** – Authentication Failed.
- **403** – You are not allowed to revoke transform job with id **id**.
- **404** – No transform job with id **id**.
- **415** – Wrong id format **id**.

GET /publish/job/id/(id)

Return a publish job serialized to JSON.

The publish job attributes are appended with the Celery's `async` result of the job.

All `thing_id` fields are replaced by corresponding `thing`. For example `user_id` is replaced by `user`'s data.

Example request:

```
GET /publish/job/id/c697f528-74f7-11e2-96a3-3085a9accc5d HTTP/1.1
Host: somewhere.com
Header: tabby@bernex.ch:miaow
Accept: application/json
```

Example response:

Floating numbers are here with “” for autoflask to work !!

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
    "status": 200,
    "value": {
        "_id": "c697f528-74f7-11e2-96a3-3085a9accc5d",
        "publish_uri": "http://<address>/medias/<user_id>/<media_id>/Project_London_trailer_2009.mp4",
        "media": {
            "_id": "a8a165b8-74f7-11e2-a59e-3085a9acc049",
            "user_id": "4e8ec55e-74f7-11e2-8451-3085a9acc8e0b",
            "parent_id": null,
            "uri": "glusterfs://<address>/medias_volume/medias/<user_id>/<media_id>",
            "public_uris": {
                "c697f528-74f7-11e2-96a3-3085a9accc5d": "http://<address>/medias/<user_id>/<media_id>/Project_London_trailer_2009.mp4"
            },
            "virtual_filename": "Project_London_trailer_2009.mp4",
            "metadata": {
                "duration": "00:02:44.88", "add_date": "2013-02-11 22:37",
                "size": 54871886,
                "title": "Project London - Official Trailer (2009)"
            },
            "status": "PUBLISHED"
        },
        "user": { "name": "David Fischer", "...": "..." },
        "statistic": {
            "add_date": "2013-02-11 22:38", "start_date": "2013-02-11 22:38",
            "media_size": 54871886, "publish_size": 54871886,
            "elapsed_time": "0.5068690776824951",
            "eta_time": 0, "percent": 100,
            "pid": 18307, "hostname": "famille-local-oscied-publisher-0"
        },
        "revoked": false,
        "status": "SUCCESS"
    }
}
```

Allowed Any user

Parameters

- `id` – id of job to get

Status Codes

- **200** – OK

- **401** – Authenticate.
- **403** – Authentication Failed.
- **404** – No publish job with id `id`.

DELETE /publish/job/`id`/(`id`)

Revoke a publish job.

This method do not delete jobs from jobs database but set `revoked` attribute in jobs database and broadcast revoke request to publisher units with Celery. If the job is actually running it will be canceled. The output publication media will be deleted.

Example request:

```
DELETE /publish/job/id/c697f528-74f7-11e2-96a3-3085a9accc5d HTTP/1.1
Host: somewhere.com
Header: tabby@bernex.ch:miaow
Accept: application/json
Content-Type: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json
```

```
{
  "status": 200,
  "value": "The publish job \"<job_id>\" has been revoked.
            Corresponding media will be unpublished from here."
}
```

Allowed Only author of the job

Parameters

- `id` – id of job to delete

Status Codes

- **200** – OK
- **401** – Authenticate.
- **403** – Authentication Failed.
- **403** – You are not allowed to revoke publish job with id `id`.
- **404** – No publish job with id `id`.
- **415** – Wrong id format `id`.

GET /media/`id`/(`id`) /HEAD

Return a media serialized to JSON.

Example request:

```
GET /media/id/96590fdc-74f8-11e2-8c58-3085a9acc651/HEAD HTTP/1.1
Host: somewhere.com
Header: monique@oscied.org:oscied
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json
```

```
{
  "status": 200,
  "value": {
    "_id": "b8b9ae78-74f8-11e2-8dae-3085a9accb1",
    "user_id": "4e8ec55e-74f7-11e2-8451-3085a9acc8e0b",
    "parent_id": null,
    "uri": "glusterfs://<address>/medias_volume/medias/
           <user_id>/<media_id>",
    "public_uris": null,
    "virtual_filename": "Psy_gangnam_style.flv",
    "metadata": {
      "duration": "00:04:12.16",
      "add_date": "2013-02-11 22:37",
      "title": "Psy - Gangnam Style",
      "size": 183190475
    },
    "status": "READY"
  }
}
```

Allowed Any user

Parameters

- `id` – id of media to get

Status Codes

- **200** – OK
- **401** – Authenticate.
- **403** – Authentication Failed.
- **404** – No media with id `id`.
- **415** – Wrong id format `id`.

GET /media/`id`/(`id`)

Return a media serialized to JSON.

All `thing_id` fields are replaced by corresponding `thing`. For example `user_id` is replaced by `user`'s data.

Example request:

```
GET /media/id/96590fdc-74f8-11e2-8c58-3085a9acc651 HTTP/1.1
Host: somewhere.com
Header: estelle@oscied.org:oscied
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "status": 200,
  "value": {
    "_id": "b8b9ae78-74f8-11e2-8dae-3085a9accbc1",
    "user": {
      "_id": "4e8ec55e-74f7-11e2-8451-3085a9acc8e0b",
      "first_name": "David",
      "last_name": "Fischer",
      "name": "David Fischer",
      "mail": "d@f.com",
      "admin_platform": true
    },
    "parent": null,
    "uri": "glusterfs://<address>/medias_volume/medias/<user_id>/<media_id>",
    "public_uris": null,
    "virtual_filename": "Psy_gangnam_style.flv",
    "metadata": {
      "duration": "00:04:12.16",
      "add_date": "2013-02-11 22:37",
      "title": "Psy - Gangnam Style",
      "size": 183190475
    },
    "status": "READY"
  }
}
```

Allowed Any user

Parameters

- **id** – id of media to get

Status Codes

- **200** – OK
- **401** – Authenticate.
- **403** – Authentication Failed.
- **404** – No media with id **id**.
- **415** – Wrong id format **id**.

- **415** – Requires json content-type.

PUT /media/id/ (*id*)

Update a media (only virtual_filename and metadata field can be updated).

Example request:

```
PUT /media/id/a396fe66-74ee-11e2-89ad-3085a9accbb8 HTTP/1.1
Host: somewhere.com
Header: anthony@oscied.org:oscied
Accept: application/json
Content-Type: application/json
```

```
{"virtual_filename": "the_fifth_element.mp4"}
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "status": 200,
  "value": "The media \"fifth_element.mp4\" has been updated."
}
```

Allowed Only the author of the media

param id media's id

query virtual_filename Media's filename when downloaded or published (optional)

query metadata JSON string containing metadatas about the media (optional)

statuscode 200 OK

statuscode 400 Key key not found. *or* on type or value error

statuscode 401 Authenticate.

statuscode 403 Authentication Failed.

statuscode 403 You are not allowed to modify media with id **id**.

statuscode 404 No media with id **id**.

statuscode 415 Wrong id format **id**.

statuscode 415 Requires (valid) json content-type.

PATCH /media/id/ (*id*)

Update a media (only virtual_filename and metadata field can be updated).

Example request:

```
PUT /media/id/a396fe66-74ee-11e2-89ad-3085a9accbb8 HTTP/1.1
Host: somewhere.com
Header: anthony@oscied.org:oscied
Accept: application/json
Content-Type: application/json

{"virtual_filename": "the_fifth_element.mp4"}
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "status": 200,
  "value": "The media \"fifth_element.mp4\" has been updated."
}
```

Allowed Only the author of the media**param id** media's id**query virtual_filename** Media's filename when downloaded or published (optional)**query metadata** JSON string containing metadatas about the media (optional)**statuscode 200** OK**statuscode 400** Key key not found. *or* on type or value error**statuscode 401** Authenticate.**statuscode 403** Authentication Failed.**statuscode 403** You are not allowed to modify media with id *id*.**statuscode 404** No media with id *id*.**statuscode 415** Wrong id format *id*.**statuscode 415** Requires (valid) json content-type.**DELETE /media/id/**(*id*)

Delete a media.

The media file is removed from the shared storage and media's status is set to DELETED.

Example request:

```
DELETE /media/id/a396fe66-74ee-11e2-89ad-3085a9accbb8 HTTP/1.1
Host: somewhere.com
Header: sandro@oscied.org:oscied
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "status": 200,
  "value": "The media \"fifth_element.mp4\" has been deleted."
}
```

Allowed Only the author of the media**param id** id of media to delete**statuscode 200** OK**statuscode 400** Cannot delete the media, it is actually in use by transform job with id *id* and status *status*.**statuscode 400** Cannot delete the media, it is actually in use by publish job with id *id* and status *status*.**statuscode 401** Authenticate.**statuscode 403** Authentication Failed.**statuscode 403** You are not allowed to delete media with id *id*.**statuscode 404** No media with id *id*.**statuscode 415** Wrong id format *id*.**statuscode 501** FIXME Delete of external uri not implemented.**GET /user/id/**(*id*)

Return an user serialized to JSON (without secret field).

Example request:

```
GET /user/id/c4daa8a6-6be4-11e2-ae91-3085a9accb47 HTTP/1.1
Host: somewhere.com
Header: michael@oscied.org:oscied
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "status": 200,
  "value": {
    "_id": "c4daa8a6-6be4-11e2-ae91-3085a9accb47",
    "first_name": "Tabby",
```

```

    "last_name": "Fischer",
    "name": "Tabby Fischer",
    "mail": "tabby@bernex.ch",
    "admin_platform": false
}
}

```

Allowed Root and any user

Parameters

- **id** – id of user to get

Status Codes

- **200** – OK
- **401** – Authenticate.
- **403** – Authentication Failed.
- **404** – No user with id **id**.
- **415** – Wrong id format **id**.

PUT /user/**id**/*(id)*

Update an user.

User's admin_platform attribute can only be modified by root or any authenticated user with admin_platform attribute set.

Example request:

```

PUT /user/id/8bda488c-6be8-11e2-89b7-3085a9accb47 HTTP/1.1
Host: somewhere.com
Header: loic@oscied.org:oscied
Accept: application/json
Content-Type: application/json

{
  "mail": "laurent.nicolet@comiques.ch",
  "secret": "gnevois_style",
  "admin_platform": true
}

```

Example response:

```

HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "status": 200,
  "value": "The user \"Laurent Nicolet\" has been updated."
}

```

Allowed Root, user with admin_platform attribute set or the user it-

self

Parameters

- **id** – id of user to get

Query Parameters

- **first_name** – User's first name (optional)
- **last_name** – User's last name (optional)
- **mail** – User's email address (optional)
- **secret** – User's secret (optional)
- **admin_platform** – User's admin_platform (optional)

Status Codes

- **200** – OK
- **400** – on type or value error
- **400** – Key key not found.
- **400** – The email address **mail** is already used by another user.
- **401** – Authenticate.
- **403** – Authentication Failed.
- **404** – No user with id **id**
- **415** – Wrong id format **id**.
- **415** – Requires (valid) json content-type.

PATCH /user/**id**/*(id)*

Update an user.

User's admin_platform attribute can only be modified by root or any authenticated user with admin_platform attribute set.

Example request:

```

PUT /user/id/8bda488c-6be8-11e2-89b7-3085a9accb47 HTTP/1.1
Host: somewhere.com
Header: loic@oscied.org:oscied
Accept: application/json
Content-Type: application/json

{
  "mail": "laurent.nicolet@comiques.ch",
  "secret": "gnevois_style",
  "admin_platform": true
}

```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{  
    "status": 200,
```

```

    "value": "The user \"Laurent Nicolet\" has been updated."
}

```

Allowed Root, user with admin_platform attribute set or the user itself

Parameters

- **id** – id of user to get

Query Parameters

- **first_name** – User's first name (optional)
- **last_name** – User's last name (optional)
- **mail** – User's email address (optional)
- **secret** – User's secret (optional)
- **admin_platform** – User's admin_platform (optional)

Status Codes

- **200** – OK
- **400** – on type or value error
- **400** – Key key not found.
- **400** – The email address mail is already used by another user.
- **401** – Authenticate.
- **403** – Authentication Failed.
- **404** – No user with id id
- **415** – Wrong id format id.
- **415** – Requires (valid) json content-type.

DELETE /user/id/(*id*)

Delete an user.

Example request:

```

DELETE /user/id/8bda488c-6be8-11e2-89b7-3085a9accc47 HTTP/1.1
Host: somewhere.com
Header: laurent.nicolet@comiques.ch:gnevois_style
Accept: application/json

```

Example response:

```

HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "status": 200,
  "value": "The user \"Laurent Nicolet\" has been deleted."
}

```

Allowed Root, user with admin_platform attribute set or the user itself

Parameters

- **id** – id of user to delete

Status Codes

- **200** – OK
- **401** – Authenticate.
- **403** – Authentication Failed.
- **404** – No user with id id.
- **415** – Wrong id format id.

6.6 OSCIED - Main Scripts

6.6.1 autodetect-storage-ip.sh

```

1 . ./common.sh
2
3 get_unit_config 'oscied-orchestra' '0' 'storage_ip'
4 if [ ! "$REPLY" ]; then
5     xecho 'Unable to detect storage internal IP address'
6 else
7     mecho "Updating common.sh with detected storage internal IP = $REPLY"
8     sed -i "s#${STORAGE_PRIVATE_IP}.*#${STORAGE_PRIVATE_IP}=$REPLY#" common.sh
9 fi

```

6.6.2 celery-source.sh

```

1 #!/usr/bin/env bash
2 echo '/usr/local/lib/python2.7/dist-packages/celery'

```

6.6.3 common.sh.lu-dep

```

1 set -o nounset # will exit if an uninitialized variable is used
2
3 # Prevent importing N times the following (like C++ .h : #ifndef ... #endif)
4 if ! osciedCommonImported 2>/dev/null; then
5
6 # Constants -----
7
8 # FIXME the actual orchestrator doesn't accept external IP, the private is hardcoded here
9 STORAGE_PRIVATE_IP=''
10
11 # Common
12 RELEASE='precise'
13 NETWORK_IFACE='eth0' # FIXME this is hardcoded
14
15 SCRIPTS_PATH=$(pwd)
16 BASE_PATH=$(dirname "$SCRIPTS_PATH")
17 CONFIG_PATH="$BASE_PATH/config"
18
19 # Reports related configuration (e.g. listing of components)
20 REPORT_TOOLS_PATH="$BASE_PATH/report/tools"
21 REPORT_TOOLS_PLANTUML_BINARY="$REPORT_TOOLS_PATH/plantuml.jar"
22 DAVID_MA_REPORT_PATH="$BASE_PATH/report/david/MA"
23 DAVID_MA_REPORT_BUILD_PATH="$DAVID_MA_REPORT_PATH/build"
24 DAVID_MA_REPORT_RELEASES_PATH="$DAVID_MA_REPORT_PATH/releases"
25 DAVID_MA_REPORT_UML_PATH="$DAVID_MA_REPORT_PATH/uml"
26 DAVID_MA_REPORT_COMPONENTS_FILE="$DAVID_MA_REPORT_PATH/source/components.csv"
27
28 # Generated configuration
29 CONFIG_GEN_PATH="$CONFIG_PATH/generated"
30 CONFIG_GEN_AUTHS_FILE="$CONFIG_GEN_PATH/auths.list"
31 CONFIG_GEN_IDS_FILE="$CONFIG_GEN_PATH/ids.list"
32 CONFIG_GEN_JSON_FILE="$CONFIG_GEN_PATH/json.list"
33 CONFIG_GEN_UNITS_FILE="$CONFIG_GEN_PATH/units.list"
34 CONFIG_GEN_CONFIG_FILE="$CONFIG_GEN_PATH/config.yaml"
35
36 # Orchestra related configuration (e.g. initial setup)
37 CONFIG_API_PATH="$CONFIG_PATH/api"
38 CONFIG_API_USERS_FILE="$CONFIG_API_PATH/users.csv"
39 CONFIG_API_MEDIAS_FILE="$CONFIG_API_PATH/medias.csv"
40 CONFIG_API_TPROFILES_FILE="$CONFIG_API_PATH/tprofiles.csv"
41
42 # JuJu related configuration (e.g. environments)
43 CONFIG_JUJU_PATH="$CONFIG_PATH/juju"
44 CONFIG_JUJU_ID_RSA="$CONFIG_JUJU_PATH/id_rsa"
45 CONFIG_JUJU_ID_RSA_PUB="$CONFIG_JUJU_PATH/id_rsa.pub"
46 CONFIG_JUJU_ENVS_FILE="$CONFIG_JUJU_PATH/environments.yaml"
47 CONFIG_JUJU_FILES_PATH="$CONFIG_PATH/juju_files"

```

```

48 CONFIG_JUJU_TEMPL_FILE="$CONFIG_JUJU_FILES_PATH/environments.yaml.template"
49
50 CONFIG_SCENARIOS_PATH="$CONFIG_PATH/scenarios"
51
52 # TODO
53 MEDIAS_PATH="$BASE_PATH/medias"
54 MEDIAS_TEST_PATH="$MEDIAS_PATH/Test"
55 SUBLIME_PATH="$BASE_PATH/sublime"
56 COMPONENTS_PATH="$BASE_PATH/components"
57
58 CLOUD_REFERENCES_PATH="$COMPONENTS_PATH/cloud/references"
59 CLOUD_TOOLS_PATH="$COMPONENTS_PATH/cloud/tools"
60 JUJU_CHARMS_PATH="$COMPONENTS_PATH/juju/charms"
61 JUJU_TOOLS_PATH="$COMPONENTS_PATH/juju/tools"
62
63 JUJU_PATH="$HOME/.juju"
64 JUJU_STORAGE_PATH="$JUJU_PATH/storage/"
65 JUJU_ENVS_FILE="$JUJU_PATH/environments.yaml"
66
67 ORCHESTRA_CHARM_PATH="$COMPONENTS_PATH/orchestra/charm"
68 ORCHESTRA_LIB_PATH="$COMPONENTS_PATH/orchestra/charm/lib"
69 ORCHESTRA_TOOLS_PATH="$COMPONENTS_PATH/orchestra/tools"
70 PUBLISHER_CHARM_PATH="$COMPONENTS_PATH/publisher/charm"
71 TRANSFORM_CHARM_PATH="$COMPONENTS_PATH/transform/charm"
72 WEBUI_CHARM_HOOKS_LIB_PATH="$COMPONENTS_PATH/webui/charm/hooks_lib"
73 WEBUI_CHARM_WWW_PATH="$COMPONENTS_PATH/webui/charm/www"
74
75 BAD_AUTH='charlie@hacker.com:challenge_accepted'
76
77 # Utilities =====
78
79 # Checkout or update a clone of a git repository -----
80
81 git_co()
82 {
83     if [ $# -ne 2 ]; then
84         xecho "Usage: $(basename $0).git_co dir repository"
85     fi
86     dir=$1
87     repository=$2
88     if [ -d "$dir" ]
89     then cd "$dir" && git reset --hard && git pull && cd ..
90     else cd "$(dirname "$dir")" && git clone "$repository"
91     fi
92 }
93
94 # Parse config.json of a actually running charm instance ! -----
95
96 get_unit_config()
97 {
98     if [ $# -ne 3 ]; then
99         xecho "Usage: $(basename $0).get_config_unit name number option"
100    fi
101    name=$1
102    number=$2
103    option=$3
104    val=$(juju ssh $name/$number "cat /var/lib/juju/units/$name-*/*/charm/config.json")
105    REPLY=$(expr match "$val" ".*\\"$option\": \"\\([^\"]*\")\",.*")
106 }
107
108 # Parse orchestra.yaml configuration file to get options value -----
109
110 get_root_secret()
111 {
112     if [ -f "$CONFIG_GEN_CONFIG_FILE" ]; then
113         line=$(cat "$CONFIG_GEN_CONFIG_FILE" | grep root_secret)
114         root=$(expr match "$line" '.*\"(.*)\".*')
115     else
116         root='toto'
117     fi
118     [ ! "$root" ] && xecho 'Unable to detect root secret !'
119     REPLY="$root"
120 }

```

```

121
122 get_node_secret()
123 {
124     if [ -f "$CONFIG_GEN_CONFIG_FILE" ]; then
125         line=$(cat "$CONFIG_GEN_CONFIG_FILE" | grep nodes_secret)
126         node=$(expr match "$line" '.*"\(.*\)".*')
127     else
128         node='abcd'
129     fi
130     [ ! "$node" ] && xecho 'Unable to detect nodes secret !'
131     REPLY="$node"
132 }
133
134 # Parse charm's units URLs listing file to get specific URLs -----
135
136 get_units_dialog_listing()
137 {
138     REPLY=$(cat "$CONFIG_GEN_UNITS_FILE" | sort | sed 's==::g;s:\n: :g')
139     [ ! $REPLY ] && xecho 'Unable to generate units listing for dialog'
140 }
141
142 get_services_dialog_listing()
143 {
144     REPLY=$(cat "$CONFIG_GEN_UNITS_FILE" | sort | sed 's/[0-9]*=: :g;s:\n: :g' | uniq)
145     [ ! $REPLY ] && xecho 'Unable to generate services listing for dialog'
146 }
147
148 get_unit_public_url()
149 {
150     if [ $# -gt 2 ]; then
151         xecho "Usage: $(basename $0).get_unit_public_url name (number)"
152     fi
153     name=$1
154     [ $# -eq 2 ] && number=$2 || number='.*'
155     if [ -f "$CONFIG_GEN_UNITS_FILE" ]; then
156         url=$(cat "$CONFIG_GEN_UNITS_FILE" | grep -m 1 "^$name/$number=" | cut -d '=' -f2)
157     else
158         url='127.0.0.1'
159     fi
160     [ ! "$url" ] && xecho "Unable to detect unit $1 public URL !"
161     REPLY="$url"
162 }
163
164 get_orchestra_url()
165 {
166     if [ $# -eq 0 ]; then
167         get_unit_public_url 'oscied-orchestra'
168     elif [ $# -eq 1 ]; then
169         get_unit_public_url 'oscied-orchestra' "$1"
170     else
171         xecho "Usage: $(basename $0).get_orchestra_url (number)"
172     fi
173     REPLY="http://$REPLY:5000"
174 }
175
176 get_storage_uploads_url()
177 {
178     REPLY="glusterfs://$STORAGE_PRIVATE_IP/medias_volume/uploads"
179 }
180
181
182 get_storage_medias_url()
183 {
184     REPLY="glusterfs://$STORAGE_PRIVATE_IP/medias_volume/medias"
185 }
186
187 storage_upload_media()
188 {
189     if [ $# -ne 1 ]; then
190         xecho "Usage: $(basename $0).storage_upload_media filename"
191     fi
192     get_unit_public_url 'oscied-storage'

```

```

194 host="ubuntu@$REPLY"
195 bkp_path='/home/ubuntu/uploads'
196 dst_path="/expl/uploads"
197 certif="$CONFIG_JUJU_ID_RSA"
198 chmod 600 "$certif" || xecho 'Unable to find id_rsa certificate'
199 rsync -ah --progress --rsync-path='sudo rsync' -e "ssh -i '$certif'" "$1" "$host:$bkp_path/" || \
200 xecho "Unable to copy media file to $bkp_path path in storage"
201 ssh -i "$certif" "$host" -n "sudo rsync -ah --progress $bkp_path/ $dst_path/" || \
202 xecho "Unable to synchronize ($dst_path->$dst_path) paths in storage"
203 ssh -i "$certif" "$host" -n "sudo chown www-data:www-data $dst_path/ -R" || \
204 xecho "Unable to set owner www-data for $dst_path path in storage"
205 get_storage_uploads_url
206 REPLY="$REPLY/$(basename $1)"
207 }
208
209 # Wrapper to juju -----
210
211 juju_unit_add()
212 {
213     if [ $# -ne 1 ]; then
214         xecho "Usage: $(basename $0).juju_unit_add service"
215     fi
216     juju add-unit "$1"
217 }
218
219 juju_unit_remove()
220 {
221     if [ $# -ne 1 ]; then
222         xecho "Usage: $(basename $0).juju_unit_remove unit"
223     fi
224     if juju remove-unit "$1"; then
225         cat "$CONFIG_GEN_UNITS_FILE" | grep -v "$1=.*" > $tmpfile
226         mv $tmpfile "$CONFIG_GEN_UNITS_FILE"
227     fi
228 }
229
230 # Save and get configuration from corresponding generated files -----
231
232 save_auth()
233 {
234     cat "$CONFIG_GEN_AUTHS_FILE" 2>/dev/null | grep -v "^\$1=" > /tmp/$$
235     echo "\$1=\$2" >> /tmp/$$
236     mv /tmp/$$ "$CONFIG_GEN_AUTHS_FILE"
237 }
238
239 get_auth()
240 {
241     REPLY=$(cat "$CONFIG_GEN_AUTHS_FILE" 2>/dev/null | grep "^\$1=" | cut -d '=' -f2)
242     [ ! "$REPLY" ] && xecho "Unable to detect $1 authentication"
243 }
244
245 save_id()
246 {
247     cat "$CONFIG_GEN_IDS_FILE" 2>/dev/null | grep -v "^\$1=" > /tmp/$$
248     echo "\$1=\$2" >> /tmp/$$
249     mv /tmp/$$ "$CONFIG_GEN_IDS_FILE"
250 }
251
252 get_id()
253 {
254     REPLY=$(cat "$CONFIG_GEN_IDS_FILE" 2>/dev/null | grep "^\$1=" | cut -d '=' -f2)
255     [ ! "$REPLY" ] && xecho "Unable to detect $1 ID"
256 }
257
258 save_json()
259 {
260     cat "$CONFIG_GEN_JSON_FILE" 2>/dev/null | grep -v "^\$1=" > /tmp/$$
261     echo "\$1=\$2" >> /tmp/$$
262     mv /tmp/$$ "$CONFIG_GEN_JSON_FILE"
263 }
264
265 get_json()
266 {

```

```

267 REPLY=$(cat "$CONFIG_GEN_JSON_FILE" 2>/dev/null | grep "^\$1=" | cut -d '=' -f2)
268 [ ! "$REPLY" ] && xecho "Unable to detect $1 json"
269 }
270
271 # Generate valid json strings of Orchestra API's objects -----
272
273 json_user()
274 {
275     if [ $# -ne 5 ]; then
276         xecho "Usage: $(basename $0).json_user fname lname mail secret aplaftorm"
277     fi
278
279     a='admin_platform'
280     JSON="{\"first_name\": \"$1\", \"last_name\": \"$2\", \"mail\": \"$3\", \"secret\": \"$4\", \"$a\": \"$5\"}"
281 }
282
283 json_media()
284 {
285     if [ $# -ne 3 ]; then
286         xecho "Usage: $(basename $0).json_media uri vfilename title"
287     fi
288
289     JSON="{\"uri\": \"$1\", \"virtual_filename\": \"$2\", \"metadata\": {\"title\": \"$3\"}}"
290 }
291
292 json_tprofile()
293 {
294     if [ $# -ne 3 ]; then
295         xecho "Usage: $(basename $0).json_tprofile title description encoder_string"
296     fi
297
298     JSON="{\"title\": \"$1\", \"description\": \"$2\", \"encoder_string\": \"$3\"}"
299 }
300
301 json_tjob()
302 {
303     d='metadata'
304     m='media_in_id'
305     p='profile_id'
306     q='queue'
307     t='title'
308     v='virtual_filename'
309     y='priority'
310     if [ $# -ne 6 ]; then
311         xecho "Usage: $(basename $0).json_tjob $m $p $v $t $q $y"
312     fi
313
314     JSON="{\"$m\": \"$1\", \"$p\": \"$2\", \"$v\": \"$3\", \"$d\": {\"$t\": \"$4\"}, \"$q\": \"$5\", \"$y\": \"$6\"}"
315 }
316
317 json_pjob()
318 {
319     if [ $# -ne 3 ]; then
320         xecho "Usage: $(basename $0).json_tjob media_id queue priority"
321     fi
322
323     JSON="{\"media_id\": \"$1\", \"queue\": \"$2\", \"priority\": \"$3\"}"
324 }
325
326 # Used to test project's Python source code or to call/test Orchestra REST API -----
327
328 test_code()
329 {
330     if [ $# -ne 1 ]; then
331         xecho "Usage: $(basename $0).test_code python_filename"
332     fi
333
334     echo "[RUN] $1"
335     python $1
336     if [ $? -ne 0 ]; then
337         echo "[ERROR] $1" >&2
338         exit 1
339     fi

```

```

340     echo
341 }
342
343 test_api()
344 {
345     if [ $# -ne 5 ]; then
346         xecho "Usage: $(basename $0).test_api code method call user data"
347     fi
348
349     code=$1; m=$2; c=$3; u=$4; d=$5
350     aa='Accept: application/json'
351     ct='Content-type: application/json'
352     if [ "$u" -a "$d" ]; then
353         mecho "\nTest $code : $m $c auth: $u data: $d"
354         result=$(curl -H "$aa" -H "$ct" -u "$u" -d "$d" -X "$m" "$c" --write-out %{http_code})
355     elif [ "$u" ]; then
356         mecho "\nTest $code : $m $c auth: $u"
357         result=$(curl -H "$aa" -H "$ct" -u "$u" -X "$m" "$c" --write-out %{http_code})
358     elif [ "$d" ]; then
359         mecho "\nTest $code : $m $c data: $d"
360         result=$(curl -H "$aa" -H "$ct" -d "$d" -X "$m" "$c" --write-out %{http_code})
361     else
362         mecho "\nTest $code : $m $c"
363         result=$(curl -H "$aa" -H "$ct" -X "$m" "$c" --write-out %{http_code})
364     fi
365     echo $result
366     if ! echo "$result" | grep -q "$code\$"; then
367         xecho "Test $m $c failed with code : $result"
368     fi
369     echo
370     anum='0-9a-zA-Z'
371     regex=".*\\"([anum]{8})-([anum]{4})-([anum]{4})-([anum]{4})-([anum]{12})\".*"
372     ID=$(expr match "$result" "$regex")
373 }
374
375 osciedCommonImported()
376 {
377     echo > /dev/null
378 }
379 fi

```

6.6.4 generate-report.sh

```

1  ./common.sh
2
3 main()
4 {
5     revision=$(svn info | grep -m 1 '.*vision.*: *[0-9]*$')
6     revision=$(expr match "$revision" '[^0-9]*\([0-9]*\)[^0-9]*')
7     if [ ! "$revision" ]; then
8         xecho 'Unable to detect local copy revision number !'
9     fi
10
11 # Generate images from textual UMLs
12 java -jar "$REPORT_TOOLS_PLANTUML_BINARY" "$DAVID_MA_REPORT_UML_PATH" \
13     -failonerror || xecho 'Unable to generate images from UML diagrams'
14
15 cd "$DAVID_MA_REPORT_UML_PATH" || xecho "Unable to find path $DAVID_MA_REPORT_UML_PATH"
16
17 # Append hooks UMLs images together !
18 for name in 'orchestra' 'webui' 'storage' 'transform' 'publisher'
19 do
20     a="activity-$name-install.png"
21     b="activity-$name-config-changed.png"
22     c="activity-$name-start.png"
23     d="activity-$name-stop.png"
24     e="activity-$name-hooks.png"
25     convert $a $b $c $d +append $e || convert $a $c $d +append $e || \
26         xecho "Unable to append $name's hooks UMLs images"
27     rm $a $b $c $d 2>/dev/null
28 done

```

```

29
30 cd "$DAVID_MA_REPORT_PATH" || xecho "Unable to find path $DAVID_MA_REPORT_PATH"
31
32 listing=/tmp/$$.list
33 tmpfile=/tmp/$$
34 trap "rm -f '$listing' '$tmpfile' 2>/dev/null" INT TERM EXIT
35 find . -type f -name "*.rst.template" | sort > $listing
36 while read template
37 do
38     rest=$(dirname "$template")/$(basename "$template" .template)
39     sed "s:SVN_REVISION:$revision:g" "$template" > "$rest"
40 done < $listing
41
42 common=''
43 references=''
44 savedIFS=$IFS
45 IFS=';'
46 while read name replace url
47 do
48     if [ ! "$replace" -a ! "$url" ]; then
49         references="$references$name\n"
50     fi
51     if [ "$replace" ]; then
52         common="$common.. |$name| replace:: $replace\n"
53     fi
54     if [ "$url" ]; then
55         common="$common.. _$name: $url\n"
56         common="$common.. |${name}_link| replace:: [$name] $url\n"
57         references="$references* |${name}_link|\n"
58     fi
59 done < 'source/common.rst.links'
60 IFS=$savedIFS
61 echo $e_ "$common" > 'source/common.rst'
62 echo $e_ "$references" > 'source/appendices-references.rst'
63
64 find . -type f -name "*.rst.header" | sort > $listing
65 while read header
66 do
67     rest=$(dirname "$header")/$(basename "$header" .header)
68     cat "$header" "$rest" > "$tmpfile"
69     mv "$tmpfile" "$rest"
70 done < $listing
71
72 # FIXME echo about.rst -> index.rst for html version at least
73 $sudo rm -rf build/* 2>/dev/null
74 make html || xecho 'Unable to generate HTML version of the report'
75 make latexpdf || xecho 'Unable to generate PDF version of the report'
76
77 find "$DAVID_MA_REPORT_BUILD_PATH" -type f -name "*.pdf" -exec mv {} "$DAVID_MA_REPORT_RELEASES_PATH" \;
78 cd "$DAVID_MA_REPORT_RELEASES_PATH" || xecho "Unable to find path $DAVID_MA_REPORT_RELEASES_PATH"
79
80 gs -sDEVICE=pdfwrite -dCompatibilityLevel=1.4 -dPDFSETTINGS=/ebook -dNOPAUSE -dQUIET -dBATCH \
81 -sOutputFile=MA_DavidFischer_OSCIEd_compressed.pdf MA_DavidFischer_OSCIEd.pdf
82 }
83
84 old()
85 {
86     notes='--inline-footnotes'
87     report='OSCIEd-Report_draft'
88     rst2html "$report.rst" > "$report.html"
89     rst2pdf "$report.rst" -s stylesheet.txt -b1 $notes
90     evince "$report.pdf"
91 }
92
93 svn_dates()
94 {
95     svn log "$1" | grep david.fischer | cut -d' ' -f5 | uniq
96 }
97
98 svn_filter()
99 {
100    echo "$1" | sed 's: :\n:g' | grep "$2" | wc -l
101 }

```

```

102
103 svn_months()
104 {
105     dates=$(svn_dates "$1")
106     d1209=$(svn_filter "$dates" '2012-09')
107     d1210=$(svn_filter "$dates" '2012-10')
108     d1211=$(svn_filter "$dates" '2012-11')
109     d1212=$(svn_filter "$dates" '2012-12')
110     d1301=$(svn_filter "$dates" '2013-01')
111     d1302=$(svn_filter "$dates" '2013-02')
112     echo "$d1209, $d1210, $d1211, $d1212, $d1301, $d1302"
113 }
114
115 svn_components()
116 {
117     content=''
118     savedIFS=$IFS
119     IFS=';'
120     while read name path
121     do
122         mecho "Gathering days per month for $name ..."
123         [ ! \"$content\" ] && content="$name.axis: y\n"
124         content="$content      $name: $(svn_months \"$path\")\n"
125     done < "$DAVID_MA_REPORT_COMPONENTS_FILE"
126
127     content=".. vertical_barchart::\n\n$content"
128     echo $e_ $content
129 }
130
131 main "$@"

```

6.6.5 juju-gui.sh

6.6.6 juju-menu.sh

```

1 . ./common.sh
2
3 main()
4 {
5     if [ $# -gt 0 ]; then
6         operation_auto="$1"
7         shift
8     elif [ $# -eq 0 ]; then
9         operation_auto=''
10    else
11        xecho "Usage: $(basename $0) (nothing) -or- operation [parameters]"
12    fi
13
14    if ! which juju > /dev/null; then
15        xecho 'JuJu must be installed'
16    fi
17
18    autoInstall dialog dialog
19
20    listing=/tmp/$$.list
21    tmpfile=/tmp/$$
22    jujulog='/usr/local/bin/juju-log'
23    openport='/usr/local/bin/open-port'
24    cget='/usr/local/bin/config-get'
25    rget='/usr/local/bin/relation-get'
26    uget='/usr/local/bin/unit-get'
27    trap "rm -f '$listing' '$tmpfile' '$jujulog' '$openport' '$cget' '$rget' '$uget' 2>/dev/null" \
28          INT TERM EXIT
29
30    if [ "$operation_auto" ]; then
31        ok=$false
32        techo 'OSCIED Operations with JuJu [AUTO]'
33        mecho "Operation is $operation_auto"
34        eval $operation_auto "$@"
35        if [ $ok -eq $false ]; then
36            xecho "Unknown operation : $operation_auto"

```

```

37     fi
38   else
39     # Initialize main menu
40     while true
41     do
42       $DIALOG --backtitle 'OSCIED Operations with JuJu' \
43         --menu 'Please select an operation' 0 0 0 \
44         overwrite      'Overwrite charms in juju path' \
45         update        'Update code of charms in juju path' \
46         deploy        'Launch a deployment scenario' \
47         destroy       'Destroy a deployed environment' \
48         standalone    'Play with a charm locally (yes, really)' \
49         status        'Display juju status' \
50         status_svg    'Display juju status as a SVG graphic' \
51         log          'Launch juju debug log in a screen' \
52         config        'Update units public url listing file' \
53         unit_ssh     'Access to units with secure shell' \
54         unit_add     'Add a new unit to a running service' \
55         unit_remove   'Remove an unit from a running service' \
56         service_destroy 'Destroy a running service' 2> $tmpfile
57
58       retval=$?
59       operation=$(cat $tmpfile)
60       [ $retval -ne 0 -o ! "$operation" ] && break
61       echo "Execute operation $operation"
62       eval $operation
63       [ $retval -eq 0 ] && pause
64     done
65   fi
66 }
67
68 overwrite()
69 {
70   if [ $# -ne 0 ]; then
71     xecho "Usage: $(basename $0) overwrite"
72   fi
73   ok=$true
74
75   overwrite_helper 'orchestra' || xecho 'Unable to overwrite Orchestra charm'
76   overwrite_helper 'publisher' || xecho 'Unable to overwrite Publisher charm'
77   overwrite_helper 'storage' || xecho 'Unable to overwrite Storage charm'
78   overwrite_helper 'transform' || xecho 'Unable to overwrite Transform charm'
79   overwrite_helper 'webui' || xecho 'Unable to overwrite Web UI charm'
80 }
81
82 overwrite_helper()
83 {
84   [ $# -ne 1 ] && xecho 'OUPS !'
85   rm -rf "$JUJU_CHARMS_PATH/$RELEASE/oscied-$1" 2>/dev/null
86   mkdir -p "$JUJU_CHARMS_PATH/$RELEASE" 2>/dev/null
87   rsync -rtvh -LH --delete --progress --exclude='*.svn' --exclude='*.log' --exclude='*.pyc' \
88     --exclude='celeryconfig.py' "$COMPONENTS_PATH/$1/charm/" "$JUJU_CHARMS_PATH/$RELEASE/oscied-$1/"
89 }
90
91 update()
92 {
93   if [ $# -ne 0 ]; then
94     xecho "Usage: $(basename $0) update"
95   fi
96   ok=$true
97
98   update_helper 'orchestra' || xecho 'Unable to overwrite Orchestra charm'
99   update_helper 'publisher' || xecho 'Unable to overwrite Publisher charm'
100  update_helper 'storage' || xecho 'Unable to overwrite Storage charm'
101  update_helper 'transform' || xecho 'Unable to overwrite Transform charm'
102  update_helper 'webui' || xecho 'Unable to overwrite Web UI charm'
103 }
104
105 update_helper()
106 {
107   [ $# -ne 1 ] && xecho 'OUPS !'
108   rsync -rtvh -LH --progress --exclude='*.svn' --exclude='*.log' --exclude='*.pyc' \
109     --exclude='config.*' "$COMPONENTS_PATH/$1/charm/" "$JUJU_CHARMS_PATH/$RELEASE/oscied-$1/"

```

```

110 }
111
112 deploy()
113 {
114     if [ $# -eq 1 ]; then
115         scenario_auto=$1
116     elif [ $# -eq 0 ]; then
117         scenario_auto=''
118     else
119         xecho "Usage: $(basename $0) deploy [scenario]"
120     fi
121     ok=$true
122
123     pecho 'Initialize JuJu orchestrator configuration'
124     if [ ! -f ${HOME}/.ssh/id_rsa ]; then ssh-keygen -t rsa; fi # FIXME better trick
125     ssh-add # Fix ERROR SSH forwarding error: Agent admitted failure to sign using the key.
126     mkdir -p "$JUJU_STORAGE_PATH" 2>/dev/null
127     if [ -f "$CONFIG_JUJU_ENVS_FILE" ]; then
128         mecho "Using user defined environment : $CONFIG_JUJU_ENVS_FILE"
129         cp "$CONFIG_JUJU_ENVS_FILE" "$JUJU_ENVS_FILE" || \
130             xecho "Unable to copy juju's configuration file"
131     else
132         mecho 'Using default template to generate environment'
133         sed "s:STORAGE_PATH:$JUJU_STORAGE_PATH:g" "$CONFIG_JUJU_TEMPL_FILE" > "$JUJU_ENVS_FILE" || \
134             xecho "Unable to generate juju's configuration file"
135     fi
136     sudo ufw disable # Fix Ticket #80 - Juju stuck in pending when using LXC
137
138     cd "$CONFIG_SCENARIOS_PATH" || xecho "Unable to find path $CONFIG_SCENARIOS_PATH"
139
140     # Initialize scenarios menu
141     find . -type f -name '*.sh' | sort > $listing
142     scenariosList=''
143     while read scenario
144     do
145         . "$scenario" # Include scenario source
146         name=$(basename "$scenario" .sh)
147         description=$(eval "echo \$${name}Description" | sed 's: :_g')
148         scenariosList="$scenariosList$name $description"
149     done < $listing
150
151     if [ "$scenario_auto" ]; then
152         techo 'OSCIED Operations with JuJu > Deployment Scenarios [AUTO]'
153         mecho "Scenario is $scenario_auto"
154         eval "${scenario_auto}Scenario"
155     else
156         # Scenarios menu
157         while true
158         do
159             $DIALOG --backtitle 'OSCIED Operations with JuJu > Deployment Scenarios' \
160                 --menu 'Please select a deployment scenario' 0 0 0 \
161                 $scenariosList 2> $tmpfile
162
163             retval=$?
164             scenario=$(cat $tmpfile)
165             [ $retval -ne 0 -o ! "$scenario" ] && break
166             eval "${scenario}Scenario"
167             [ $retval -eq 0 ] && pause
168         done
169     fi
170 }
171
172 destroy()
173 {
174     if [ $# -ne 0 ]; then
175         xecho "Usage: $(basename $0) destroy"
176     fi
177     ok=$true
178
179     xecho 'FIXME TODO MENU'
180     juju destroy-environment
181 }
182

```

```

183 standalone()
184 {
185     if [ $# -eq 2 ]; then
186         charm_auto=$1
187         hook_auto=$2
188     elif [ $# -eq 0 ]; then
189         charm_auto=''
190         hook_auto=''
191     else
192         xecho "Usage: $(basename $0) standalone [charm hook]"
193     fi
194     ok=$true
195
196     cd "$JUJU_CHARMS_PATH" || xecho "Unable to find path $JUJU_CHARMS_PATH"
197
198     find .
199
200     if [ "$charm_auto" -a "$hook_auto" ]; then
201         techo 'OSCIED Operations with JuJu > Charms Standalone [AUTO]'
202         mecho "Charm is $charm_auto, hook is $hook_auto"
203         standalone_execute_hook "$JUJU_CHARMS_PATH/$charm_auto" "$hook_auto"
204     else
205         # Initialize charms menu
206         find . -mindepth 2 -maxdepth 2 -type d | sort > $listing
207         charmsList=''
208         while read charm
209         do
210             charmsList="$charmsList$charm - "
211         done < $listing
212
213         # Charms menu
214         while true
215         do
216             $DIALOG --backtitle 'OSCIED Operations with JuJu > Charms Standalone' \
217                 --menu 'Please select a charm' 0 0 0 \
218                 $charmsList 2> $tmpfile
219
220             retval=$?
221             charm="$JUJU_CHARMS_PATH/$(cat $tmpfile)"
222             [ $retval -ne 0 -o ! "$charm" ] && break
223             cd "$charm" || xecho "Unable to find path $charm"
224
225             # Initialize hooks menu
226             find 'hooks' -mindepth 1 -maxdepth 1 -type f | sort > $listing
227             hooksList=''
228             while read hook
229             do
230                 hooksList="$hooksList$hook - "
231             done < $listing
232
233             # Hooks menu
234             while true
235             do
236                 name=$(basename $charm)
237                 $DIALOG --backtitle "OSCIED Operations with JuJu > Charms Standalone > Charm $name" \
238                     --menu 'Please select a hook' 0 0 0 \
239                     $hooksList 2> $tmpfile
240
241             retval=$?
242             hook=$(cat $tmpfile)
243             [ $retval -ne 0 -o ! "$hook" ] && break
244             standalone_execute_hook "$charm" "$hook"
245             [ $retval -eq 0 ] && pause
246         done
247
248         # Charms menu pause
249         [ $retval -eq 0 ] && pause
250     done
251 fi
252 }
253
254 standalone_execute_hook()
255 {

```

```

256 [ $# -ne 2 ] && xecho 'OUPS !'
257
258 pecho 'Install juju-log & open-port tricks'
259 if ! getInterfaceIPv4 "$NETWORK_IFACE" '4'; then
260     xecho "Unable to detect network interface $NETWORK_IFACE IP address"
261 fi
262 ip=$REPLY
263 $sudo sh -c "cp -f $CONFIG_JUJU_FILES_PATH/juju-log      $jujulog; chmod 777 $jujulog"
264 $sudo sh -c "cp -f $CONFIG_JUJU_FILES_PATH/open-port    $openport; chmod 777 $openport"
265 $sudo sh -c "cp -f $CONFIG_JUJU_FILES_PATH/something-get $cget;   chmod 777 $cget"
266 $sudo sh -c "cp -f $CONFIG_JUJU_FILES_PATH/something-get $rget;   chmod 777 $rget"
267 $sudo sh -c "cp -f $CONFIG_JUJU_FILES_PATH/something-get $uget;   chmod 777 $uget"
268 $sudo sh -c "cp -f $CONFIG_JUJU_FILES_PATH/something-get.list /tmp/;""
269 $sudo sh -c "sed -i 's:127.0.0.1:$ip:g' /tmp/something-get.list"
270 pecho "Execute hook script $2"
271 cd "$1" || xecho "Unable to find path $1"
272 $sudo $2 || xecho 'Hook is unsucessful'
273 recho 'Hook successful'
274 }
275
276 status()
277 {
278     if [ $# -ne 0 ]; then
279         xecho "Usage: $(basename $0) status"
280     fi
281     ok=$true
282
283     juju status
284 }
285
286 status_svg()
287 {
288     if [ $# -ne 0 ]; then
289         xecho "Usage: $(basename $0) status_svg"
290     fi
291     ok=$true
292
293     juju status --format svg --output status.svg
294     eog status.svg
295 }
296
297 log()
298 {
299     if [ $# -ne 0 ]; then
300         xecho "Usage: $(basename $0) log"
301     fi
302     ok=$true
303
304     screen -dmS juju-log juju debug-log
305 }
306
307 config()
308 {
309     if [ $# -ne 0 ]; then
310         xecho "Usage: $(basename $0) config"
311     fi
312     ok=$true
313
314     content=''
315     count=1
316     last_name=''
317     juju status 2>/dev/null | \
318     {
319         while read line
320         do
321             name=$(expr match "$line" '.*\(\oscied-.*/[0-9]*\):.*')
322             address=$(expr match "$line" '.*public-address: *\([^\ ]*\) *')
323             [ "$name" ] && last_name=$name
324             if [ "$address" -a "$last_name" ]; then
325                 mecho "$last_name -> $address"
326                 [ "$content" ] && content="$content\n"
327                 content="$content$last_name=$address"
328                 last_name=''
```

```

329     fi
330     count=$((count+1))
331 done
332
333 if [ "$content" ]; then
334     echo $e_ "$content" > "$CONFIG_GEN_UNITS_FILE"
335     recho "Charms's units public URLs listing file updated"
336 else
337     xecho "Unable to detect charms's units public URLs"
338 fi
339 }
340 }
341
342 unit_ssh()
343 {
344     if [ $# -ne 0 ]; then
345         xecho "Usage: $(basename $0) unit_ssh"
346     fi
347     ok=$true
348
349     yesOrNo $true 'Update units listing'
350     [ $REPLY -eq $true ] && config
351
352     # Initialize remote menu
353     unitsList=$(cat "$CONFIG_GEN_UNITS_FILE" | sort | sed 's=: :g;s:\n: :g')
354
355     # Remote menu
356     while true
357     do
358         $DIALOG --backtitle 'OSCIED Operations with JuJu > Remote Access to Units' \
359             --menu 'Please select a unit' 0 0 0 \
360             $unitsList 2> $tmpfile
361
362         retval=$?
363         unit=$(cat $tmpfile)
364         [ $retval -ne 0 -o ! "$unit" ] && break
365         juju ssh "$unit"
366         [ $retval -eq 0 ] && pause
367     done
368 }
369
370 unit_add()
371 {
372     if [ $# -ne 0 ]; then
373         xecho "Usage: $(basename $0) unit_add"
374     fi
375     ok=$true
376
377     yesOrNo $true 'Update units listing'
378     [ $REPLY -eq $true ] && config
379
380     # Initialize add unit menu
381     get_services_dialog_listing
382     $DIALOG --backtitle 'OSCIED Operations with JuJu > Scale-up a Service' \
383             --menu 'Please select a service' 0 0 0 $REPLY 2> $tmpfile
384
385     retval=$?
386     service=$(cat $tmpfile)
387     if [ $retval -ne 0 -o ! "$service" ]; then
388         recho 'Operation aborted by user'
389     else
390         juju_unit_add "$service"
391     fi
392 }
393
394 unit_remove()
395 {
396     if [ $# -ne 0 ]; then
397         xecho "Usage: $(basename $0) unit_remove"
398     fi
399     ok=$true
400
401     yesOrNo $true 'Update units listing'

```

```

402 [ $REPLY -eq $true ] && config
403
404 # Initialize remove unit menu
405 get_units_dialog_listing
406 $DIALOG --backtitle 'OSCIED Operations with JuJu > Scale-down a Service' \
407     --menu 'Please select an unit' 0 0 0 $REPLY 2> $tmpfile
408
409 retval=$?
410 unit=$(cat $tmpfile)
411 if [ $retval -ne 0 -o ! "$unit" ]; then
412     recho 'Operation aborted by user'
413 else
414     juju_unit_remove "$unit"
415 fi
416 }
417
418 main "$@"

```

6.6.7 meld.sh

```

1 . ./common.sh
2
3 techo 'Compare hook scripts of two charms'
4
5 if [ $# -ne 2 ]; then
6     xecho "Usage: $(basename $0) charm1_name charm2_name"
7 fi
8
9 cd "$COMPONENTS_PATH" || xecho "Unable to find path $COMPONENTS_PATH"
10
11 meld "$1/charm/hooks_lib/common.sh.lu-dep" "$2/charm/hooks_lib/common.sh.lu-dep"

```

6.6.8 menu.sh

```

1 . ./common.sh
2
3 main()
{
4     if [ $# -gt 0 ]; then
5         operation_auto="$1"
6         shift
7     elif [ $# -eq 0 ]; then
8         operation_auto=''
9     else
10        xecho "Usage: $(basename $0) (nothing) -or- operation [parameters]"
11    fi
12
13    autoInstall dialog dialog
14
15    get_root_secret;   ROOT_AUTH="root:$REPLY"
16    get_node_secret;   NODE_AUTH="node:$REPLY"
17    get_orchestra_url; ORCHESTRA_URL=$REPLY
18
19    mkdir -p "$MEDIAS_TEST_PATH" 2>/dev/null
20    get_storage_uploads_url; UPLOADS_URL=$REPLY
21    get_storage_mediias_url; MEDIAS_URL=$REPLY
22
23    listing=/tmp/$$.list
24    tmpfile=/tmp/$$
25    trap "rm -f '$listing' '$tmpfile' 2>/dev/null" INT TERM EXIT
26
27    if [ "$operation_auto" ]; then
28        ok=$false
29        techo 'OSCIED General Operations [AUTO]'
30        mecho "Operation is $operation_auto"
31        eval $operation_auto "$@"
32        if [ $ok -eq $false ]; then
33            xecho "Unknown operation : $operation_auto"
34        fi
35    fi

```

```

36     else
37         # Initialize main menu
38         while true
39             do
40                 $DIALOG --backtitle 'OSCIED General Operations' \
41                         --menu 'Please select an operation' 0 0 0 \
42                         install          'Download/update documents & tools, add symlinks' \
43                         cleanup          'Cleanup configuration of charms' \
44                         revup            'Increment all charm's revision (+1)' \
45                         statistics       'Update statistics about this repository' \
46                         api_init_setup   'Initialize demo setup with Orchestra API' \
47                         api_launch_transform 'Launch a transform job with Orchestra API' \
48                         api_revoke_transform 'Revoke a transform job with Orchestra API' \
49                         api_launch_publish 'Launch a publish job with Orchestra API' \
50                         api_revoke_publish 'Revoke a publish job with Orchestra API' \
51                         api_test_all      'Test the whole methods of Orchestra API' \
52                         api_get_all       'Get listings of all things with Orchestra API' \
53                         webui_test_common 'Test some functions of Web UI hooks' \
54                         rsync_orchestra   'Rsync local code to running Orchestra instance' \
55                         rsync_publisher   'Rsync local code to running Publisher instance' \
56                         rsync_transform    'Rsync local code to running Transform instance' \
57                         rsync_webui       'Rsync local code to running Web UI instance' 2> $tmpfile
58
59             retval=$?
60             operation=$(cat $tmpfile)
61
62             # Main menu exit door
63             [ $retval -ne 0 -o ! "$operation" ] && break
64
65             techo "Execute operation $operation"
66             eval $operation
67
68             # Main menu pause
69             [ $retval -eq 0 ] && pause
70         done
71     fi
72 }
73
74 install()
75 {
76     if [ $# -ne 0 ]; then
77         xecho "Usage: $(basename $0) install"
78     fi
79     ok=$true
80
81     autoInstall git-core git
82     autoInstall bzr bzr
83
84     ln -s "$COMPONENTS_PATH/orchestra/charm/lib" 2>/dev/null
85     cp -f "$JUJU_CHARMS_PATH/$RELEASE/orchestra/celeryconfig.py" . 2>/dev/null
86
87     pecho 'Update Sublime Text configuration'
88     find "$SUBLIME_PATH" -type f -exec sed -i "s:$BASE_PATH:$BASE_PATH:g" {} \
89
90     pecho 'Component Cloud'
91
92     cd "$CLOUD_REFERENCES_PATH"|| xecho "Unable to find path $CLOUD_REFERENCES_PATH"
93     git_co 'openstack-folsom-guide' 'git://github.com/EmilienM/openstack-folsom-guide.git'
94     openstack='http://docs.openstack.org'
95     wget -N $openstack/trunk/openstack-compute/install/apt/openstack-install-guide-apt-trunk.pdf
96     wget -N $openstack/cli/quick-start/content/cli-guide.pdf
97     wget -N $openstack/api/openstack-compute/programmer/openstackapi-programming.pdf
98     wget -N $openstack/folsom/openstack-compute/admin/bk-compute-adminguide-folsom.pdf
99     wget -N $openstack/folsom/openstack-network/admin/bk-quantum-admin-guide-folsom.pdf
100    wget -N $openstack/folsom/openstack-object-storage/admin/os-objectstorage-adminguide-folsom.pdf
101
102    cd "$CLOUD_TOOLS_PATH"|| xecho "Unable to find path $CLOUD_TOOLS_PATH"
103    clonezilla='http://switch.dl.sourceforge.net/project/clonezilla'
104    wget -N $clonezilla/clonezilla_live_stable/1.2.12-67/clonezilla-live-1.2.12-67-amd64.zip
105    git_co 'openstack-scripts' 'https://github.com/neophilo/openstack-scripts/'
106
107    pecho 'Component JuJu'
108

```

```

109 cd "$JUJU_TOOLS_PATH" || xecho "Unable to find path $JUJU_TOOLS_PATH"
110 if [ -d 'juju' ]; then cd 'juju' && bzr merge && cd ..
111 else bzr branch lp:juju 'juju'
112 fi
113
114 addAptPpaRepo ppa:juju/pkgs juju || xecho 'Unable to add juju ppa repository'
115 eval $install lxc apt-cacher-ng libzookeeper-java zookeeper juju juju-jitsu charm-tools || \
116   xecho 'Unable to install juju orchestrator'
117
118 #cat \
119 # /var/lib/apt/lists/ppa.launchpad.net_juju_pkgs_ubuntu_dists_quantal_main_binary-amd64_Packages \
120 # / grep Package
121 #Package: python-txzookeeper
122 #Package: juju
123 #Package: charm-tools
124 #Package: charm-helper-sh
125 #Package: juju-jitsu
126 #Package: python-charmhelpers
127
128 pecho 'Component Orchestra'
129
130 cd "$ORCHESTRA_TOOLS_PATH" || xecho "Unable to find path $ORCHESTRA_TOOLS_PATH"
131 git_co 'celery'          'git://github.com/celery/celery.git'
132 git_co 'celery-examples' 'git://github.com/larsbutler/celery-examples.git'
133 git_co 'flask'            'git://github.com/mitsuhiko/flask.git'
134 git_co 'rabbitmq-tutorials' 'git://github.com/rabbitmq/rabbitmq-tutorials.git'
135
136 cd 'rabbitmq-tutorials' || xecho 'Unable to find RabbitMQ tutorials path'
137 readLine 'Please enter local RabbitMQ guest user password'
138 pass=$CHOICE
139 a='amqp://.*localhost/'
140 b="amqp://guest:$pass@localhost/"
141 find . -type f -exec sed -i "s#${a##$b#g}" {} \;
142 git status
143
144 pecho 'Component Report'
145
146 cd "$REPORT_TOOLS_PATH" || xecho "Unable to find path $REPORT_TOOLS_PATH"
147
148 plantuml=http://downloads.sourceforge.net/project/plantuml
149 wget -N $plantuml/plantuml.jar
150 wget -N $plantuml/PlantUML%20Language%20Reference%20Guide.pdf
151
152 #addAptPpaRepo ppa:phobie/ppa phobie || xecho 'Unable to add phobie ppa repository'
153 #autoInstall openproj openproj
154 eval $install texlive-latex-recommended texlive-latex-extra texlive-fonts-recommended
155 autoInstall python-pip pip
156 autoInstall rst2pdf rst2pdf
157
158 # FIX https://github.com/martinkou/bson/issues/19, bson not installed
159 $sudo pip install --upgrade docutils pygments sphinx sphinxcontrib-email \
160   sphinxcontrib-googlechart sphinxcontrib-httplib flask celery requests
161 $sudo easy_install pymongo
162
163 #Fix https://bitbucket.org/birkenfeld/sphinx/pull-request/98/fixes-typeerror-raised-from/diff
164 $sudo find /usr/local/lib/ -type f -name latex.py -path "*/sphinx/writers/*" -exec \
165   sed -i 's:letter.translate(tex_escape_map)):unicode(letter).translate(tex_escape_map)):g' {} \
166 }
167
168 cleanup()
169 {
170   if [ $# -ne 0 ]; then
171     xecho "Usage: $(basename $0) cleanup"
172   fi
173   ok=$true
174
175   find "$COMPONENTS_PATH" -maxdepth 3 -type d -name 'charm' | while read path
176   do
177     rm -f "$path/celeryconfig.py" 2>/dev/null
178     [ -f "$path/config.json" ] && svn revert "$path/config.json"
179   done
180 }
181

```

```

revup()
{
    if [ $# -ne 0 ]; then
        xecho "Usage: $(basename $0) revup"
    fi
    ok=$true

    cd "$COMPONENTS_PATH" || xecho "Unable to find path $COMPONENTS_PATH"
    find . -type f -not -path "*oscied-*" -name revision | while read revision
    do
        value=$(cat "$revision")
        echo $((value+1)) > "$revision"
        mecho "$(basename $(dirname $(dirname $revision))) is not at revision $(cat $revision)"
    done
}

statistics()
{
    if [ $# -ne 0 ]; then
        xecho "Usage: $(basename $0) statistics"
    fi
    ok=$true

    inc='components/**:scripts/**'
    exc='**/*.ova:**/*.pdf:**/*.jpg:**/*.png:**/*.mpd:**/*.svg:**/*.od*'
    #exc='administration/**:*/build/**:*/medias/**:*/references/**:*/statistics/**:*/todo/**:*/*.ova:**/*.pdf:**/*.jpg:**/*.png:**/*.mpd:**/*.svg:**/*.od*'

    cd "$BASE_PATH" || xecho "Unable to find path $BASE_PATH"

    yesOrNo $false 'update OSCIED project statistics'
    if [ $REPLY -eq $true ]; then
        autoInstall statsvn statsvn

        tmpFilename=/tmp/$$
        trap "rm -f '$tmpFilename' 2>/dev/null" INT TERM EXIT

        pecho '1/2 getting repository verbose log ...'
        url=`svn status | grep ^URL | sed 's/URL.*: //`'
        svn log -v --xml > $tmpFilename

        pecho '2/2 using statsvn to generate statistics ...'
        rm -f "statistics/*" 2>/dev/null
        statsvn -threads 10 $tmpFilename -include $inc -exclude $exc . -output-dir statistics/
    fi
}

api_init_setup()
{
    if [ $# -ne 0 ]; then
        xecho "Usage: $(basename $0) api_init_setup"
    fi
    ok=$true

    pecho 'Flush database'
    yesOrNo $false 'do you really want to flush orchestrator $ORCHESTRA_URL'
    if [ $REPLY -eq $false ]; then
        recho 'operation aborted by user'
        exit 0
    fi
    test_api 200 POST $ORCHESTRA_URL/flush "$$ROOT_AUTH" ''

    pecho 'Add users'
    count=1
    savedIFS=$IFS
    IFS=';'
    while read fname lname mail secret applatform
    do
        if [ ! "$fname" -o ! "$lname" -o ! "$mail" -o ! "$secret" -o ! "$applatform" ]; then
            xecho "Line $count : Bad line format !"
        fi
        json_user "$fname" "$lname" "$mail" "$secret" "$applatform"
    done
}

```

```

255     echo "$JSON"
256     test_api 200 POST $ORCHESTRA_URL/user "$ROOT_AUTH" "$JSON"
257     save_auth "user$count" "$mail:$secret"
258     save_json "user$count" "$JSON"
259     save_id   "user$count" "$ID"
260     count=$((count+1))
261 done < "$CONFIG_API_USERS_FILE"
262 IFS=$savedIFS
263
264 get_auth 'user1'; user1_auth=$REPLY
265
266 pecho 'Add medias'
267 count=1
268 savedIFS=$IFS
269 IFS=';'
270 while read uri vfilename title
271 do
272     if [ ! "$uri" -o ! "$vfilename" -o ! "$title" ]; then
273         xecho "Line $count : Bad line format !"
274     fi
275     media="$MEDIAS_TEST_PATH/$uri"
276     if [ ! -f "$media" ]; then
277         xecho "Unable to find media file $media"
278     fi
279     mecho "Uploading media $uri to uploads into shared storage unit ..."
280     storage_upload_media "$media" || xecho "Unable to upload media"
281     json_media "$REPLY" "$vfilename" "$title"
282     echo "$JSON"
283     test_api 200 POST $ORCHESTRA_URL/media "$user1_auth" "$JSON"
284     save_json "media$count" "$JSON"
285     save_id   "media$count" "$ID"
286     count=$((count+1))
287 done < "$CONFIG_API_MEDIAS_FILE"
288 IFS=$savedIFS
289
290 pecho 'Add transform profiles'
291 count=1
292 savedIFS=$IFS
293 IFS=';'
294 while read title description encoder_string
295 do
296     if [ ! "$title" -o ! "$description" -o ! "$encoder_string" ]; then
297         xecho "Line $count : Bad line format !"
298     fi
299     json_tprofile "$title" "$description" "$encoder_string"
300     echo "$JSON"
301     test_api 200 POST $ORCHESTRA_URL/transform/profile "$user1_auth" "$JSON"
302     save_json "tprofile$count" "$JSON"
303     save_id   "tprofile$count" "$ID"
304     count=$((count+1))
305 done < "$CONFIG_API_TPROFILES_FILE"
306 IFS=$savedIFS
307
308 #pecho 'Add medias'
309 #$sudo mkdir -p /mnt/storage/medias /mnt/storage/uploads
310 #$sudo cp "$SCRIPTS_PATH/common.sh" /mnt/storage/uploads/tabby.mpg
311 #test_api 200 POST $ORCHESTRA_URL/media "$admin_auth" "$medial_json"; save_id 'medial' "$ID"
312
313 }
314
315 api_launch_transform()
316 {
317     if [ $# -ne 0 ]; then
318         xecho "Usage: $(basename $0) api_launch_transform"
319     fi
320     ok=$true
321
322     pecho 'Gather required authorizations and IDs'
323     get_auth 'user1'; user1_auth=$REPLY
324     get_id   'medial'; medial_id=$REPLY
325     get_id   'tprofile2'; tprofile1_id=$REPLY
326
327     pecho 'Launch a transform job'

```

```

328 json_tjob "$medial_id" "$tprofile1_id" "tabby2.mpg" 'transcoded medial' 'transform_private' 'high'
329 echo "$JSON"
330 test_api 200 POST $ORCHESTRA_URL/transform/job "$user1_auth" "$JSON"
331 save_id 'tjob1' "$ID"
332 get_id 'tjob1'
333 echo $REPLY
334 }
335
336 api_revoke_transform()
337 {
338 if [ $# -ne 0 ]; then
339 xecho "Usage: $(basename $0) api_revoke_transform"
340 fi
341 ok=$true
342
343 pecho 'Gather required authorizations and IDs'
344 get_auth 'user1'; user1_auth=$REPLY
345 get_id 'tjob1'; tjob1_id=$REPLY
346
347 pecho 'Revoke a transform job'
348 test_api 200 DELETE $ORCHESTRA_URL/transform/job/id/$tjob1_id "$user1_auth" ''
349 }
350
351 api_launch_publish()
352 {
353 if [ $# -ne 0 ]; then
354 xecho "Usage: $(basename $0) api_launch_publish"
355 fi
356 ok=$true
357
358 pecho 'Gather required authorizations and IDs'
359 get_auth 'user1'; user1_auth=$REPLY
360 get_id 'medial'; medial_id=$REPLY
361
362 pecho 'Launch a publish job'
363 json_pjob "$medial_id" 'publisher_private' 'high'
364 echo "$JSON"
365 test_api 200 POST $ORCHESTRA_URL/publish/job "$user1_auth" "$JSON"
366 save_id 'pjob1' "$ID"
367 get_id 'pjob1'
368 echo $REPLY
369 }
370
371 api_revoke_publish()
372 {
373 if [ $# -ne 0 ]; then
374 xecho "Usage: $(basename $0) api_revoke_publish"
375 fi
376 ok=$true
377
378 pecho 'Gather required authorizations and IDs'
379 get_auth 'user1'; user1_auth=$REPLY
380 get_id 'pjob1'; pjob1_id=$REPLY
381
382 pecho 'Revoke a publish job'
383 test_api 200 DELETE $ORCHESTRA_URL/publish/job/id/$pjob1_id "$user1_auth" ''
384 }
385
386 api_test_all()
387 {
388 if [ $# -ne 0 ]; then
389 xecho "Usage: $(basename $0) api_test_all"
390 fi
391 ok=$true
392
393 cd "$ORCHESTRA_LIB_PATH" || xecho "Unable to find path $ORCHESTRA_LIB_PATH"
394
395 techo '1/4 Test source code'
396
397 test_code Callback.py
398 test_code Media.py
399 test_code OrchestraConfig.py
400 test_code PublisherConfig.py

```

```

401 test_code TransformConfig.py
402 test_code TransformProfile.py
403 test_code User.py
404 test_code Utilities.py
405
406 techo '2/4 Initialize Orchestra database'
407
408 api_init_setup
409
410 techo '3/4 Gather required authorizations and IDs'
411
412 get_auth 'user1'; user1_auth=$REPLY
413 get_id 'user1'; user1_id=$REPLY
414 get_json 'user1'; user1_json=$REPLY
415 get_auth 'user2'; user2_auth=$REPLY
416 get_id 'user2'; user2_id=$REPLY
417 get_json 'user2'; user2_json=$REPLY
418 get_auth 'user3'; user3_auth=$REPLY
419 get_id 'user3'; user3_id=$REPLY
420 get_json 'user3'; user3_json=$REPLY
421 #get_id 'media1'; medial_id=$REPLY
422 #get_json 'media1'; medial_json=$REPLY
423 get_id 'tprofile1'; tprofile1_id=$REPLY
424 get_json 'tprofile1'; tprofile1_json=$REPLY
425
426 techo '4/4 Test Orchestra API'
427 api_test_main
428 api_test_user
429 #api_test_media
430 #api_test_tprofile
431 #api_test_tjob
432 mecho 'Unit test passed (Orchestrator API OK)'
433 }
434
435 api_test_main()
436 {
437 pecho 'Test main API'
438 test_api 401 POST $ORCHESTRA_URL/flush ''
439 test_api 401 POST $ORCHESTRA_URL/flush "$bad_auth" ''
440 test_api 403 POST $ORCHESTRA_URL/flush "$user2_auth" ''
441 test_api 200 GET $ORCHESTRA_URL ''
442 test_api 200 GET $ORCHESTRA_URL/index ''
443 test_api 200 GET $ORCHESTRA_URL "$bad_auth" ''
444 test_api 200 GET $ORCHESTRA_URL/index "$bad_auth" ''
445 test_api 200 GET $ORCHESTRA_URL "$ROOT_AUTH" ''
446 test_api 200 GET $ORCHESTRA_URL/index "$ROOT_AUTH" ''
447 test_api 200 GET $ORCHESTRA_URL "$NODE_AUTH" ''
448 test_api 200 GET $ORCHESTRA_URL/index "$NODE_AUTH" ''
449 test_api 200 GET $ORCHESTRA_URL "$user1_auth" ''
450 test_api 200 GET $ORCHESTRA_URL/index "$user1_auth" ''
451 test_api 200 GET $ORCHESTRA_URL "$user2_auth" ''
452 test_api 200 GET $ORCHESTRA_URL/index "$user2_auth" ''
453 }
454
455 api_test_user()
456 {
457 pecho 'Test user API'
458 mecho 'We cannot double post an user'
459 test_api 400 POST $ORCHESTRA_URL/user "$ROOT_AUTH" "$user1_json"
460 mecho 'Anonymous can get nothing except one'
461 test_api 200 GET $ORCHESTRA_URL '' ''
462 test_api 200 GET $ORCHESTRA_URL/index '' ''
463 test_api 401 GET $ORCHESTRA_URL/user/login '' ''
464 test_api 401 GET $ORCHESTRA_URL/user/count '' ''
465 test_api 401 GET $ORCHESTRA_URL/user '' ''
466 test_api 401 GET $ORCHESTRA_URL/user/id/$user1_id '' ''
467 mecho 'Charlie can get nothing except one'
468 test_api 200 GET $ORCHESTRA_URL "$bad_auth" ''
469 test_api 200 GET $ORCHESTRA_URL/index "$bad_auth" ''
470 test_api 401 GET $ORCHESTRA_URL/user/login "$bad_auth" ''
471 test_api 401 GET $ORCHESTRA_URL/user/count "$bad_auth" ''
472 test_api 401 GET $ORCHESTRA_URL/user "$bad_auth" ''
473 test_api 401 GET $ORCHESTRA_URL/user/id/$user1_id "$bad_auth" ''

```

```

474  mecho 'Root can get a lot of things'
475  test_api 200 GET $ORCHESTRA_URL           "$ROOT_AUTH" ''
476  test_api 200 GET $ORCHESTRA_URL/index      "$ROOT_AUTH" ''
477  test_api 403 GET $ORCHESTRA_URL/user/login   "$ROOT_AUTH" ''
478  test_api 200 GET $ORCHESTRA_URL/user/count    "$ROOT_AUTH" ''
479  test_api 200 GET $ORCHESTRA_URL/user          "$ROOT_AUTH" ''
480  test_api 200 GET $ORCHESTRA_URL/user/id/$user1_id "$ROOT_AUTH" ''
481  test_api 415 GET $ORCHESTRA_URL/user/id/salut   "$ROOT_AUTH" ''
482  mecho 'Node can get nothing except one'
483  test_api 200 GET $ORCHESTRA_URL           "$NODE_AUTH" ''
484  test_api 200 GET $ORCHESTRA_URL/index      "$NODE_AUTH" ''
485  test_api 403 GET $ORCHESTRA_URL/user/login   "$NODE_AUTH" ''
486  test_api 403 GET $ORCHESTRA_URL/user/count    "$NODE_AUTH" ''
487  test_api 403 GET $ORCHESTRA_URL/user          "$NODE_AUTH" ''
488  test_api 403 GET $ORCHESTRA_URL/user/id/$user1_id "$NODE_AUTH" ''
489  test_api 415 GET $ORCHESTRA_URL/user/id/salut   "$NODE_AUTH" ''
490  mecho 'Admin can get a lot of things'
491  test_api 200 GET $ORCHESTRA_URL           "$user1_auth" ''
492  test_api 200 GET $ORCHESTRA_URL/index      "$user1_auth" ''
493  test_api 200 GET $ORCHESTRA_URL/user/login   "$user1_auth" ''
494  test_api 200 GET $ORCHESTRA_URL/user/count    "$user1_auth" ''
495  test_api 200 GET $ORCHESTRA_URL/user          "$user1_auth" ''
496  test_api 200 GET $ORCHESTRA_URL/user/id/$user1_id "$user1_auth" ''
497  test_api 415 GET $ORCHESTRA_URL/user/id/salut   "$user1_auth" ''
498  mecho 'Simple user can get lesser things'
499  test_api 200 GET $ORCHESTRA_URL           "$user2_auth" ''
500  test_api 200 GET $ORCHESTRA_URL/index      "$user2_auth" ''
501  test_api 200 GET $ORCHESTRA_URL/user/login   "$user2_auth" ''
502  test_api 200 GET $ORCHESTRA_URL/user/count    "$user2_auth" ''
503  test_api 403 GET $ORCHESTRA_URL/user          "$user2_auth" ''
504  test_api 200 GET $ORCHESTRA_URL/user/id/$user1_id "$user2_auth" ''
505  test_api 415 GET $ORCHESTRA_URL/user/id/salut   "$user2_auth" ''
506  mecho 'Admin can modify users (and it should work as expected)'
507  test_api 200 PATCH $ORCHESTRA_URL/user/id/$user3_id "$user1_auth" \
508    '{"first_name":"the", "last_name":"test"}'
509  test_api 200 PATCH $ORCHESTRA_URL/user/id/$user3_id "$user1_auth" \
510    '{"mail":"t@t.com", "secret":"salutMe3"}'
511  test_api 200 GET $ORCHESTRA_URL/user/login     't@t.com:salutMec3' ''
512  test_api 200 PATCH $ORCHESTRA_URL/user/id/$user3_id "$user1_auth" "$user3_json"
513  test_api 200 GET $ORCHESTRA_URL/user/login     "$user3_auth" ''
514  mecho 'Admin can add/remove admin_platform role to users'
515  test_api 200 PATCH $ORCHESTRA_URL/user/id/$user2_id "$user1_auth" '{"admin_platform":true}'
516  test_api 200 GET $ORCHESTRA_URL/user          "$user2_auth" ''
517  test_api 200 PATCH $ORCHESTRA_URL/user/id/$user2_id "$user1_auth" '{"admin_platform":false}'
518  test_api 403 GET $ORCHESTRA_URL/user          "$user2_auth" ''
519  mecho 'Simple user cannot kill another, itself yes, but not 2x (Chuck Norris can)'
520  test_api 403 DELETE $ORCHESTRA_URL/user/id/$user1_id "$user3_auth" ''
521  test_api 200 DELETE $ORCHESTRA_URL/user/id/$user2_id "$user2_auth" ''
522  test_api 401 DELETE $ORCHESTRA_URL/user/id/$user2_id "$user2_auth" ''
523  test_api 401 GET $ORCHESTRA_URL/user/login     "$user2_auth" ''
524  test_api 200 POST $ORCHESTRA_URL/user          "$user1_auth" "$user2_json"
525  save_id 'user2' "$ID"
526  test_api 200 GET $ORCHESTRA_URL/user/login     "$user2_auth" ''
527 }
528
529 api_test_media()
530 {
531  pecho 'Test media API'
532  test_api 200 POST $ORCHESTRA_URL/media        "$user1_auth" "$medial_json"; medial_id=$ID
533  test_api 400 POST $ORCHESTRA_URL/media        "$user1_auth" "$medial_json"
534  test_api 200 GET $ORCHESTRA_URL/media/count    "$user2_auth" ''
535  test_api 200 GET $ORCHESTRA_URL/media          "$user2_auth" ''
536  test_api 403 PATCH $ORCHESTRA_URL/media/id/$medial_id "$user2_auth" "$medialb_json"
537  test_api 200 PATCH $ORCHESTRA_URL/media/id/$medial_id "$user1_auth" "$medialb_json"
538  test_api 200 GET $ORCHESTRA_URL/media/id/$medial_id "$user2_auth" ''
539  test_api 403 DELETE $ORCHESTRA_URL/media/id/$medial_id "$user2_auth" ''
540  test_api 200 DELETE $ORCHESTRA_URL/media/id/$medial_id "$user1_auth" ''
541  test_api 200 POST $ORCHESTRA_URL/media        "$user1_auth" "$medial_json"; medial_id=$ID
542 }
543
544 api_test_tprofile()
545 {
546  pecho 'Test transform profile API'

```

```

547     mecho 'We cannot double post a profile'
548     test_api 400 POST $ORCHESTRA_URL/transform/profile "$user1_auth" "$tprofile1_json"
549     mecho 'Anonymous can get nothing'
550     test_api 401 GET $ORCHESTRA_URL/transform/profile/count      ''
551     test_api 401 GET $ORCHESTRA_URL/transform/profile          ''
552     test_api 401 GET $ORCHESTRA_URL/transform/profile/id/$tprofile1_id ''
553     mecho 'Charlie can get nothing'
554     test_api 401 GET $ORCHESTRA_URL/transform/profile/count      "$bad_auth" ''
555     test_api 401 GET $ORCHESTRA_URL/transform/profile          "$bad_auth" ''
556     test_api 401 GET $ORCHESTRA_URL/transform/profile/id/$tprofile1_id "$bad_auth" ''
557     mecho 'Root can get nothing'
558     test_api 403 GET $ORCHESTRA_URL/transform/profile/count      "$ROOT_AUTH" ''
559     test_api 403 GET $ORCHESTRA_URL/transform/profile          "$ROOT_AUTH" ''
560     test_api 403 GET $ORCHESTRA_URL/transform/profile/id/$tprofile1_id "$ROOT_AUTH" ''
561     test_api 415 GET $ORCHESTRA_URL/transform/profile/id/salut   "$ROOT_AUTH" ''
562     mecho 'Node can get nothing'
563     test_api 403 GET $ORCHESTRA_URL/transform/profile/count      "$NODE_AUTH" ''
564     test_api 403 GET $ORCHESTRA_URL/transform/profile          "$NODE_AUTH" ''
565     test_api 403 GET $ORCHESTRA_URL/transform/profile/id/$tprofile1_id "$NODE_AUTH" ''
566     test_api 415 GET $ORCHESTRA_URL/transform/profile/id/salut   "$NODE_AUTH" ''
567     mecho 'Admin can get a lot of things'
568     test_api 200 GET $ORCHESTRA_URL/transform/profile/count      "$user1_auth" ''
569     test_api 200 GET $ORCHESTRA_URL/transform/profile          "$user1_auth" ''
570     test_api 200 GET $ORCHESTRA_URL/transform/profile/id/$tprofile1_id "$user1_auth" ''
571     test_api 415 GET $ORCHESTRA_URL/transform/profile/id/salut   "$user1_auth" ''
572     mecho 'Simple user can a lot of things'
573     test_api 200 GET $ORCHESTRA_URL/transform/profile/count      "$user2_auth" ''
574     test_api 200 GET $ORCHESTRA_URL/transform/profile          "$user2_auth" ''
575     test_api 200 GET $ORCHESTRA_URL/transform/profile/id/$tprofile1_id "$user2_auth" ''
576     test_api 415 GET $ORCHESTRA_URL/transform/profile/id/salut   "$user2_auth" ''
577 }
578
579 api_test_tjob()
580 {
581     pecho 'Test transform job API'
582     #test_api 200 GET $ORCHESTRA_URL/transform/job/count "$user2_auth" ''
583     #test_api 200 GET $ORCHESTRA_URL/transform/job          "$user2_auth" ''
584     #test_api 404 POST $ORCHESTRA_URL/transform/job        "$user2_auth" "$tjob0_json"
585     #test_api 404 POST $ORCHESTRA_URL/transform/job        "$user2_auth" "$tjob0b_json"
586     #test_api 404 POST $ORCHESTRA_URL/transform/job        "$user2_auth" "$tjob1_json"
587     #test_api 200 POST $ORCHESTRA_URL/transform/job        "$user2_auth" "$tjob2_json"; tjob2_id=$ID
588     #echo $tjob2_id
589     #test_api 415 GET $ORCHESTRA_URL/job/transform/1      "$admin" ''
590     #test_api 200 GET $ORCHESTRA_URL/job/transform/$id      "$admin" ''
591     #test_api 400 POST $ORCHESTRA_URL/transform/job        "$user2_auth" "$tprofile1_json"
592
593     #while read post_job
594     #do
595     # [ "$post_job" ] && test_api 0 POST $ORCHESTRA_URL/job/transform "$admin" "$post_job"
596     #done < "$ORCHESTRA_SCRIPTS_PATH/tests.jobs"
597
598     #test_api 501 PATCH $ORCHESTRA_URL/job/transform/$id "$admin" ''
599     #test_api 200 DELETE $ORCHESTRA_URL/job/transform/$id "$admin" ''
600 }
601
602 api_get_all()
603 {
604     if [ $# -ne 0 ]; then
605         xecho "Usage: $(basename $0) api_get_all"
606     fi
607     ok=$true
608
609     get_auth 'user1'; user1_auth=$REPLY
610     test_api 200 GET $ORCHESTRA_URL           ''
611     test_api 200 GET $ORCHESTRA_URL/user/count "$user1_auth" ''
612     test_api 200 GET $ORCHESTRA_URL/user       "$user1_auth" ''
613     test_api 200 GET $ORCHESTRA_URL/media/count "$user1_auth" ''
614     test_api 200 GET $ORCHESTRA_URL/media/HEAD  "$user1_auth" ''
615     test_api 200 GET $ORCHESTRA_URL/media       "$user1_auth" ''
616     test_api 200 GET $ORCHESTRA_URL/transform/profile/count "$user1_auth" ''
617     test_api 200 GET $ORCHESTRA_URL/transform/profile      "$user1_auth" ''
618     test_api 200 GET $ORCHESTRA_URL/transform/queue       "$user1_auth" ''
619     test_api 200 GET $ORCHESTRA_URL/transform/job/count   "$user1_auth" ''

```

```

620 test_api 200 GET $ORCHESTRA_URL/transform/job/HEAD           "$user1_auth" ''
621 test_api 200 GET $ORCHESTRA_URL/transform/job                 "$user1_auth" ''
622 test_api 200 GET $ORCHESTRA_URL/publish/queue                "$user1_auth" ''
623 test_api 200 GET $ORCHESTRA_URL/publisher/queue              "$user1_auth" ''
624 test_api 200 GET $ORCHESTRA_URL/unpublish/queue              "$user1_auth" ''
625 test_api 200 GET $ORCHESTRA_URL/publish/job/count            "$user1_auth" ''
626 test_api 200 GET $ORCHESTRA_URL/publish/job/HEAD             "$user1_auth" ''
627 test_api 200 GET $ORCHESTRA_URL/publish/job                 "$user1_auth" ''
628 }
629
630 webui_test_common()
{
631     if [ $# -ne 0 ]; then
632         xecho "Usage: $(basename $0) webui_test_common"
633     fi
634     ok=$true
635
636     . "$WEBUI_CHARM_HOOKS_LIB_PATH/common.sh"
637
638     ip1='10.10.4.3'
639     ip2='10.10.0.7'
640     ip3='1.1.1.1'
641     ip4='2.2.2.2'
642
643     rm proxy_ips 2>/dev/null
644     update_proxies add    $ip1; [ "$(cat proxy_ips)" != "$ip1" ] && xecho '1'
645     update_proxies add    $ip2; [ "$(cat proxy_ips)" != "$ip1,$ip2" ] && xecho '2'
646     update_proxies add    $ip3; [ "$(cat proxy_ips)" != "$ip1,$ip2,$ip3" ] && xecho '3'
647     update_proxies remove $ip3; [ "$(cat proxy_ips)" != "$ip1,$ip2" ] && xecho '4'
648     update_proxies remove $ip3; [ "$(cat proxy_ips)" != "$ip1,$ip2" ] && xecho '5'
649     update_proxies add    $ip4; [ "$(cat proxy_ips)" != "$ip1,$ip2,$ip4" ] && xecho '6'
650     update_proxies remove $ip2; [ "$(cat proxy_ips)" != "$ip1,$ip4" ] && xecho '7'
651     update_proxies remove $ip1; [ "$(cat proxy_ips)" != "$ip4" ] && xecho '8'
652     rm proxy_ips
653     mecho 'Unit test passed (update_proxies OK)'
654 }
655
656 rsync_orchestra()
{
657     if [ $# -ne 0 ]; then
658         xecho "Usage: $(basename $0) rsync_orchestra"
659     fi
660     ok=$true
661
662     certif="$CONFIG_JUJU_ID_RSA"
663     chmod 600 "$certif" || xecho 'Unable to find id_rsa certificate'
664
665     get_unit_public_url 'oscied-orchestra' '0'
666     host="ubuntu@$REPLY"
667     dest='/var/lib/juju/units/oscied-orchestra-0/charm'
668     ssh -i "$certif" "$host" -n "sudo chown 1000:1000 $dest -R"
669     rsync -avh --progress --delete -e "ssh -i '$certif'" --exclude=.svn --exclude=config.json \
670           --exclude=celeryconfig.py --exclude=*.pyc "$ORCHESTRA_CHARM_PATH/" "$host:$dest/"
671     ssh -i "$certif" "$host" -n "sudo chown root:root $dest -R"
672 }
673
674 rsync_publisher()
{
675     if [ $# -ne 0 ]; then
676         xecho "Usage: $(basename $0) rsync_publisher"
677     fi
678     ok=$true
679
680     certif="$CONFIG_JUJU_ID_RSA"
681     chmod 600 "$certif" || xecho 'Unable to find id_rsa certificate'
682
683     get_unit_public_url 'oscied-publisher' '0'
684     host="ubuntu@$REPLY"
685     dest='/var/lib/juju/units/oscied-publisher-0/charm/lib/'
686     ssh -i "$certif" "$host" -n "sudo chown 1000:1000 $dest -R"
687     rsync -avh --progress --delete -e "ssh -i '$certif'" --exclude=.svn \
688           --exclude=*.pyc "$PUBLISHER_CHARM_PATH/lib/" "$host:$dest/"
689     ssh -i "$certif" "$host" -n "sudo chown root:root $dest -R"
690 }
691

```

```

693
694 rsync_transform()
695 {
696     if [ $# -ne 0 ]; then
697         xecho "Usage: $(basename $0) rsync_transform"
698     fi
699     ok=$true
700
701     certif="$CONFIG_JUJU_ID_RSA"
702     chmod 600 "$certif" || xecho 'Unable to find id_rsa certificate'
703
704     get_unit_public_url 'oscied-transform' '0'
705     host="ubuntu@$REPLY"
706     dest='/var/lib/juju/units/oscied-transform-0/charm/lib/'
707     ssh -i "$certif" "$host" -n "sudo chown 1000:1000 $dest -R"
708     rsync -avh --progress --delete -e "ssh -i '$certif'" --exclude=.svn \
709         --exclude=*.pyc "$TRANSFORM_CHARM_PATH/lib/" "$host:$dest/"
710     ssh -i "$certif" "$host" -n "sudo chown root:root $dest -R"
711 }
712
713 rsync_webui()
714 {
715     if [ $# -ne 0 ]; then
716         xecho "Usage: $(basename $0) rsync_webui"
717     fi
718     ok=$true
719
720     certif="$CONFIG_JUJU_ID_RSA"
721     chmod 600 "$certif" || xecho 'Unable to find id_rsa certificate'
722
723     get_unit_public_url 'oscied-webui'
724     host="ubuntu@$REPLY"
725     dest='/var/www'
726     ssh -i "$certif" "$host" -n "sudo chown 1000:1000 $dest -R"
727     rsync -avh --progress -e "ssh -i '$certif'" --exclude=.svn --exclude=.htaccess \
728         --exclude=application/config/config.php --exclude=application/config/database.php \
729         --exclude=medias --exclude=uploads --exclude=orchestra_relation_ok --delete \
730         "$WEBUI_CHARM_WWW_PATH/" "$host:$dest/"
731     ssh -i "$certif" "$host" -n "sudo chown www-data:www-data $dest -R"
732 }
733
734 main "$@"

```

6.6.9 transform-test.py

```

1 import sys
2 from celery.task.control import inspect
3 import lib.Transform
4 #from celery.task.sets import TaskSet
5 from lib.Media import MEDIA_TEST
6 from lib.TransformProfile import TRANSFORM_PROFILE_TEST
7 from lib.User import USER_TEST
8
9 #def make_pi_tasks():
10 #    taskset = TaskSet(lib.Publisher.make_pi.subtask((x, )) for x in NUM_CALCS)
11 #    print "Dispatching tasks"
12 #    taskset_result = taskset.apply_async()
13 #    print "Waiting for results"
14 #    results = taskset_result.join_native()
15 #    print "Results:"
16 #    for i in results:
17 #        print i
18
19 if __name__ == '__main__':
20     user = USER_TEST
21     media_in = MEDIA_TEST
22     media_out = MEDIA_TEST
23     profile = TRANSFORM_PROFILE_TEST
24     print 'Launch the transform job'
25     result = lib.Transform.transform_job.apply_async(
26         args=(user.json(), media_in.json(), media_out.json(), profile.json(),),
27         queue='transform private')

```

```

28     print result.state
29     print result
30     i = inspect()
31     dump = i.active()
32     print dump
33     if dump:
34         for worker, tasks in dump.iteritems():
35             print worker
36             for task in tasks:
37                 if task['name'] == 'Transform.transform_job':
38                     for key, value in task.iteritems():
39                         print 'key ' + key + ' = ' + str(value)
40                     #print 'THE Task ' + str(task)
41                     #print 'Arguments ' + str(task['args'])
42                     print 'salut'
43
44     sys.exit(0)
45 #     make_pi_tasks()

```

6.7 OSCIED - JuJu Hacks Source

6.7.1 JuJu Menu Templates and Utilities

juju_files/environments.yaml.template

```

1 default: local
2 environments:
3     local:
4         type: local
5         control-bucket: juju-a14dfa3830142d9ac23c499395c2785999
6         admin-secret: 6608267bb6b447b8c90934167b2a294999
7         data-dir: STORAGE_PATH
8         default-series: quantal
9     amazon:
10        type: ec2
11        access-key: AKIAetc
12        secret-key: Vl5g/etc
13        control-bucket: juju-4560.s3-website-eu-west-1.amazonaws.com
14        admin-secret: 81ale7429e6847c4941fda7591246594
15        default-series: precise
16        juju-origin: ppa
17        ssl-hostname-verification: true

```

juju_files/juju-log

- This script will replace juju-log when installing charms without JuJu (aka standalone).

```

1 #!/bin/bash
2
3 TXT_BLD=$(tput bold)
4 TXT_GREEN=$(tput setaf 2)
5 TXT_RESET=$(tput sgr0)
6
7 if echo "\n" | grep -q '\n'
8 then e_='-'e'
9 else e_=''
10 fi
11
12 techo() { echo $e_ "$TXT_GREEN$txt$TXT_BLD$1$txt$TXT_RESET"; } # script title
13
14 techo "[ JUJU LOG ] $@"

```

juju_files/open-port

- This script will replace open-port when installing charms without JuJu (aka standalone).

```

1  #!/bin/bash
2
3  TXT_BLD=$(tput bold)
4  TXT_GREEN=$(tput setaf 2)
5  TXT_RESET=$(tput sgr0)
6
7  if echo "\n" | grep -q '\n'
8  then e_='`-e'
9  else e_=''
10 fi
11
12 techo() { echo $e_ "$TXT_GREEN$txt$TXT_BLD$1$txt$TXT_RESET"; } # script title
13
14 techo "[OPEN PORT] $@"

```

juju_files/something-get

- This script will replace config-get and relation-get when installing charms without JuJu (aka standalone).

```

1  #!/bin/bash
2
3  TXT_RED=$(tput setaf 1)
4  TXT_RESET=$(tput sgr0)
5
6  if echo "\n" | grep -q '\n'
7  then e_='`-e'
8  else e_=''
9  fi
10
11 xecho() # message (error)
12 {
13     echo $e_ "${TXT_RED} [ERROR] $1$txt$TXT_RESET" >&2
14     exit 1
15 }
16
17 values='/tmp/something-get.list'
18 line=$(grep "^\$1=\"$values")
19 if [ ! \"$line\" ]; then
20     xecho "Unable to read value $1 from $values"
21 fi
22 echo "$line" | cut -d'=' -f2

```

juju_files/something-get.list

Note: You need to update this file according to charm's configuration default values.

```

concurrency=5
max_execution_time=180
max_input_time=600
max_upload_size=4294967296
mongo_admin_password=Mongo_admin_1234
mongo_connection=mongodb://nodes:Mongo_user_1234@127.0.0.1:27017/celery
mongo_nodes_password=Mongo_user_1234
nodes_secret=abcd
private_address=127.0.0.1
publisher_repository=https://claire-et-david.dyndns.org/prog/OSCIED/components/publisher/charm
rabbit_connection=amqp://nodes:Alice_in_wonderland@127.0.0.1:5672/celery
rabbit_password=Alice_in_wonderland
rabbit_queues=transform_private
repositories_pass=
repositories_user=oscied
root_secret=toto
storage_fstype=glusterfs
storage_ip=127.0.0.1
storage_mountpoint=medias_volume

```

```
storage_options=
transform_repository=https://claire-et-david.dyndns.org/prog/OSCIED/components/transform/charm
verbose=false
webui_repository=https://claire-et-david.dyndns.org/prog/OSCIED/components/webui/charm
```

6.7.2 Scripted Deployment Scenarios

scenarios/osciedAmazon.sh

```

1 osciedAmazonDescription='Launch oscied (minimal setup) on Amazon'
2 osciedAmazonScenario()
3 {
4     cd "$COMPONENTS_PATH/juju" || xecho "Unable to find path $COMPONENTS_PATH/juju"
5     cfg="$CONFIG_JUJU_PATH/osciedAmazon.yaml"
6     cp -f "$cfg" "$CONFIG_GEN_CONFIG_FILE"
7     tm='instance-type=t1.micro'
8
9     techo '1/5 Cleanup and bootstrap juju environment'
10
11    juju destroy-environment --environment 'amazon'
12    juju bootstrap --environment 'amazon'
13
14    techo '2/5 Deploy services on Amazon'
15
16    pecho 'Deploy Orchestra (1 instance)'
17    yesOrNo $true 'deploy it now'
18    if [ $REPLY -eq $true ]; then
19        if [ -f "$cfg" ]; then
20            mecho "Using user define Orchestra configuration : $cfg"
21            juju deploy --environment 'amazon' --constraints "$tm" --config "$cfg" \
22                --repository=charms/ local:$RELEASE/oscied-orchestra || xecho '1'
23        else
24            mecho 'Using default Orchestra configuration'
25            juju deploy --environment 'amazon' --constraints "$tm" \
26                --repository=charms/ local:$RELEASE/oscied-orchestra || xecho '1'
27        fi
28        juju expose --environment 'amazon' oscied-orchestra || xecho '2'
29    fi
30
31    pecho 'Deploy Web UI (1 instance)'
32    yesOrNo $true 'deploy it now'
33    if [ $REPLY -eq $true ]; then
34        if [ -f "$cfg" ]; then
35            mecho "Using user define Web UI configuration : $cfg"
36            juju deploy --environment 'amazon' --constraints "$tm" --config "$cfg" \
37                --repository=charms/ local:$RELEASE/oscied-webui || xecho '1'
38        else
39            mecho 'Using default Web UI configuration'
40            juju deploy --environment 'amazon' --constraints "$tm" \
41                --repository=charms/ local:$RELEASE/oscied-webui || xecho '1'
42        fi
43        juju expose --environment 'amazon' oscied-webui || xecho '2'
44    fi
45
46    pecho 'Deploy Storage (1 instance)'
47    yesOrNo $true 'deploy it now'
48    if [ $REPLY -eq $true ]; then
49        if [ -f "$cfg" ]; then
50            mecho "Using user define Storage configuration : $cfg"
51            juju deploy --environment 'amazon' --constraints "$tm" --config "$cfg" \
52                --repository=charms/ local:$RELEASE/oscied-storage || xecho '1'
53        else
54            mecho 'Using default Storage configuration'
55            juju deploy --environment 'amazon' --constraints "$tm" \
56                --repository=charms/ local:$RELEASE/oscied-storage || xecho '1'
57        fi
58        juju expose --environment 'amazon' oscied-storage || xecho '2'
59    fi
60
61    pecho 'Deploy Transform (1 instance)'
```

```

62 yesOrNo $true 'deploy it now'
63 if [ $REPLY -eq $true ]; then
64   if [ -f "$cfg" ]; then
65     mecho "Using user define Transform configuration : $cfg"
66     juju deploy --environment 'amazon' --constraints "$tm" --config "$cfg" \
67       --repository=charms/ local:$RELEASE/oscied-transform || xecho '1'
68   else
69     mecho 'Using default Transform configuration'
70     juju deploy --environment 'amazon' --constraints "$tm" \
71       --repository=charms/ local:$RELEASE/oscied-transform || xecho '1'
72   fi
73 fi
74
75 pecho 'Deploy Publisher (1 instance)'
76 yesOrNo $true 'deploy it now'
77 if [ $REPLY -eq $true ]; then
78   if [ -f "$cfg" ]; then
79     mecho "Using user define Publisher configuration : $cfg"
80     juju deploy --environment 'amazon' --constraints "$tm" --config "$cfg" \
81       --repository=charms/ local:$RELEASE/oscied-publisher || xecho '1'
82   else
83     mecho 'Using default Publisher configuration'
84     juju deploy --environment 'amazon' --constraints "$tm" \
85       --repository=charms/ local:$RELEASE/oscied-publisher || xecho '1'
86   fi
87   juju expose --environment 'amazon' oscied-publisher || xecho '2'
88 fi
89
90 pecho 'Deploy haproxy (1 instance)'
91 yesOrNo $false 'deploy it now'
92 if [ $REPLY -eq $true ]; then
93   juju deploy --environment 'amazon' --constraints "$tm" cs:precise/haproxy || xecho '2'
94   juju expose --environment 'amazon' haproxy || xecho '3'
95 fi
96
97 techo "3/5 Add relation between Storage and other services"
98
99 pecho 'Add-relation Storage <-> Transform'
100 yesOrNo $true 'add it now'
101 if [ $REPLY -eq $true ]; then
102   juju add-relation --environment 'amazon' oscied-storage oscied-transform
103 fi
104
105 pecho 'Add-relation Storage <-> Publisher'
106 yesOrNo $true 'add it now'
107 if [ $REPLY -eq $true ]; then
108   juju add-relation --environment 'amazon' oscied-storage oscied-publisher
109 fi
110
111 pecho 'Add-relation Storage <-> Orchestra'
112 yesOrNo $true 'add it now'
113 if [ $REPLY -eq $true ]; then
114   juju add-relation --environment 'amazon' oscied-storage oscied-orchestra
115 fi
116
117 pecho 'Add-relation Storage <-> Web UI'
118 yesOrNo $true 'add it now'
119 if [ $REPLY -eq $true ]; then
120   juju add-relation --environment 'amazon' oscied-storage oscied-webui
121 fi
122
123 techo "4/5 Add relation between Orchestra and other services"
124
125 pecho 'Add-relation Orchestra <-> Transform'
126 yesOrNo $true 'add it now'
127 if [ $REPLY -eq $true ]; then
128   juju add-relation --environment 'amazon' oscied-orchestra:transform oscied-transform:transform
129 fi
130
131 pecho 'Add-relation Orchestra <-> Publisher'
132 yesOrNo $true 'add it now'
133 if [ $REPLY -eq $true ]; then
134   juju add-relation --environment 'amazon' oscied-orchestra:publisher oscied-publisher:publisher

```

```

135   fi
136
137   pecho 'Add-relation Orchestra <-> Web UI'
138   yesOrNo $true 'add it now'
139   if [ $REPLY -eq $true ]; then
140     juju add-relation --environment 'amazon' oscied-orchestra:api oscied-webui:api
141   fi
142
143   techo '5/5 Add relation between Web UI and HA Proxy'
144
145   pecho 'Add-relation haproxy <-> Web UI'
146   yesOrNo $false 'add it now'
147   if [ $REPLY -eq $true ]; then
148     juju unexpose --environment 'amazon' oscied-webui
149     juju add-relation --environment 'amazon' haproxy oscied-webui
150   fi
151 }

```

scenarios/osciedDemo.sh

```

1 osciedDemoDescription='Launch oscied demo setup (MaaS Cluster, Local, Amazon)'
2 osciedDemoScenario()
3 {
4   cd "$COMPONENTS_PATH/juju" || xecho "Unable to find path $COMPONENTS_PATH/juju"
5   cfg_maas="$CONFIG_JUJU_PATH/osciedDemo_maas.yaml"
6   cfg_local="$CONFIG_JUJU_PATH/osciedDemo_local.yaml"
7   cfg_bernex="$CONFIG_JUJU_PATH/osciedDemo_bernex.yaml"
8   cfg_amazon="$CONFIG_JUJU_PATH/osciedDemo_amazon.yaml"
9
10  cp -f "$cfg_maas" "$CONFIG_GEN_CONFIG_FILE"
11
12  yesOrNo $false "Deploy on MaaS Cluster"
13  if [ $REPLY -eq $true ]; then
14    osciedDemoScenario_maas
15  fi
16  yesOrNo $false "Deploy on Local Computer"
17  if [ $REPLY -eq $true ]; then
18    osciedDemoScenario_local
19  fi
20  yesOrNo $false "Deploy on Bernex Computer"
21  if [ $REPLY -eq $true ]; then
22    osciedDemoScenario_bernex
23  fi
24  yesOrNo $false "Deploy on Amazon"
25  if [ $REPLY -eq $true ]; then
26    osciedDemoScenario_amazon
27  fi
28}
29
30 osciedDemoScenario_maas()
31 {
32   techo '1/4 Deploy services on private MaaS Cluster'
33
34   pecho 'Cleanup and bootstrap juju maas environment'
35   yesOrNo $false 'do it now'
36   if [ $REPLY -eq $true ]; then
37     juju destroy-environment --environment 'maas'
38     juju bootstrap --environment 'maas'
39   fi
40
41   pecho 'Deploy Orchestra (1 instance)'
42   mecho "Using user define Orchestra configuration : $cfg_maas"
43   yesOrNo $false 'do it now'
44   if [ $REPLY -eq $true ]; then
45     juju deploy --environment 'maas' --config "$cfg_maas" \
46       --repository=charms/ local:$RELEASE/oscied-orchestra || xecho '1'
47     juju expose --environment 'maas' oscied-orchestra || xecho '2'
48   fi
49
50   idl=$(juju status --environment 'maas' oscied-orchestra | grep 'machine:' | cut -d':' -f2)
51   if ! validateNumber "$idl"; then
52     xecho "Unable to detect id of machine that runs Orchestra"

```

```

53   fi
54
55   pecho 'Deploy Web UI (1 instance)'
56   mecho "Using user define Web UI configuration : $cfg_maas"
57   yesOrNo $false 'do it now'
58   if [ $REPLY -eq $true ]; then
59     jitsu deploy-to "$id1" --environment 'maas' --config "$cfg_maas" \
60       --repository=charms/ local:$RELEASE/oscied-webui || xecho '1'
61     juju expose --environment 'maas' oscied-webui || xecho '2'
62   fi
63
64   pecho 'Deploy Storage (1 instance)'
65   mecho "Using user define Storage configuration : $cfg_maas"
66   yesOrNo $false 'do it now'
67   if [ $REPLY -eq $true ]; then
68     juju deploy --environment 'maas' --config "$cfg_maas" \
69       --repository=charms/ local:$RELEASE/oscied-storage || xecho '1'
70     juju expose --environment 'maas' oscied-storage || xecho '2'
71   fi
72
73   id2=$(juju status --environment 'maas' oscied-storage | grep 'machine:' | cut -d':' -f2)
74   if ! validateNumber "$id2"; then
75     xecho "Unable to detect id of machine that runs Storage"
76   fi
77
78   pecho 'Deploy Transform (1 instance)'
79   mecho "Using user define Transform configuration : $cfg_maas"
80   yesOrNo $false 'do it now'
81   if [ $REPLY -eq $true ]; then
82     jitsu deploy-to "$id2" --environment 'maas' --config "$cfg_maas" \
83       --repository=charms/ local:$RELEASE/oscied-transform || xecho '1'
84   fi
85
86   pecho 'Deploy Publisher (1 instance)'
87   mecho "Using user define Publisher configuration : $cfg_maas"
88   yesOrNo $false 'do it now'
89   if [ $REPLY -eq $true ]; then
90     jitsu deploy-to "$id2" --environment 'maas' --config "$cfg_maas" \
91       --repository=charms/ local:$RELEASE/oscied-publisher || xecho '1'
92     juju expose --environment 'maas' oscied-publisher || xecho '2'
93   fi
94
95   pecho 'Add-relation Storage <-> Transform'
96   yesOrNo $false 'do it now'
97   if [ $REPLY -eq $true ]; then
98     juju add-relation --environment 'maas' oscied-storage oscied-transform
99   fi
100
101  pecho 'Add-relation Storage <-> Publisher'
102  yesOrNo $false 'do it now'
103  if [ $REPLY -eq $true ]; then
104    juju add-relation --environment 'maas' oscied-storage oscied-publisher
105  fi
106
107  pecho 'Add-relation Storage <-> Orchestra'
108  yesOrNo $false 'do it now'
109  if [ $REPLY -eq $true ]; then
110    juju add-relation --environment 'maas' oscied-storage oscied-orchestra
111  fi
112
113  pecho 'Add-relation Storage <-> Web UI'
114  yesOrNo $false 'do it now'
115  if [ $REPLY -eq $true ]; then
116    juju add-relation --environment 'maas' oscied-storage oscied-webui
117  fi
118
119  pecho 'Add-relation Orchestra <-> Transform'
120  yesOrNo $false 'do it now'
121  if [ $REPLY -eq $true ]; then
122    juju add-relation --environment 'maas' oscied-orchestra:transform oscied-transform:transform
123  fi
124
125  pecho 'Add-relation Orchestra <-> Publisher'

```

```

126 yesOrNo $false 'do it now'
127 if [ $REPLY -eq $true ]; then
128     juju add-relation --environment 'maas' oscied-orchestra:publisher oscied-publisher:publisher
129 fi
130
131 pecho 'Add-relation Orchestra <-> Web UI'
132 yesOrNo $false 'do it now'
133 if [ $REPLY -eq $true ]; then
134     juju add-relation --environment 'maas' oscied-orchestra:api oscied-webui:api
135 fi
136 }
137
138 osciedDemoScenario_local()
139 {
140     techo "2/4 Deploy services (LXC containers) into David's Workstation at hepia"
141
142     pecho 'Cleanup and bootstrap juju local environment'
143     juju destroy-environment --environment 'local'
144     juju bootstrap --environment 'local'
145
146     mecho '[WARNING] Continue only when services deployed on the MaaS cluster are up and ready !'
147     pause
148
149     pecho 'Generate services configuration for local deployment'
150
151     get_unit_config 'oscied-transform' '1' 'storage_ip'
152     if [ ! "$REPLY" ]; then xecho 'Unable to detect storage internal IP address'; fi
153     storage="s<STORAGE_IP<$REPLY<g"
154
155     get_unit_config 'oscied-transform' '1' 'mongo_connection'
156     if [ ! "$REPLY" ]; then xecho 'Unable to detect MongoDB connection'; fi
157     mongo="s<MONGO<$REPLY<g"
158
159     get_unit_config 'oscied-transform' '1' 'rabbit_connection'
160     if [ ! "$REPLY" ]; then xecho 'Unable to detect RabbitMQ connection'; fi
161     rabbit="s<RABBIT<$REPLY<g"
162
163     sed "$storage;$mongo;$rabbit" < "$cfg_local.template" > "$cfg_local"
164
165     pecho 'Deploy Transform (1 instance)'
166     mecho "Using user define Transform configuration : $cfg_local"
167     juju deploy --environment 'local' --config "$cfg_local" \
168         --repository=charms/ local:$RELEASE/oscied-transform || xecho '1'
169
170     pecho 'Deploy Publisher (1 instance)'
171     mecho "Using user define Publisher configuration : $cfg_local"
172     juju deploy --environment 'local' --config "$cfg_local" \
173         --repository=charms/ local:$RELEASE/oscied-publisher || xecho '1'
174     juju expose --environment 'local' oscied-publisher || xecho '2'
175 }

176 osciedDemoScenario_bernex()
177 {
178     techo "3/4 Deploy services (LXC containers) into David's Home Computer at Bernex"
179
180     pecho 'Cleanup and bootstrap juju bernex environment'
181     juju destroy-environment --environment 'bernex'
182     juju bootstrap --environment 'bernex'
183
184     mecho '[WARNING] Continue only when services deployed on the MaaS cluster are up and ready !'
185     pause
186
187     pecho 'Gather services configuration for bernex deployment'
188
189     default='OSCIED-Node02.localdomain'
190     readLine "Please enter storage internal IP address [$default]"
191     if [ ! "$CHOICE" ]; then CHOICE=$default; fi
192     storage="s<STORAGE_IP<$CHOICE<g"
193
194     default='129.194.185.47:5000'
195     readLine "Please enter orchestra api public (NATed) socket [$default]"
196     if [ ! "$CHOICE" ]; then CHOICE=$default; fi
197     api_nat="s<API_NAT_SOCKET<$CHOICE<g"
198

```

```

199
200     default='129.194.185.47'
201     readLine "Please enter storage public (NATed) IP address [$default]"
202     if [ ! "$CHOICE" ]; then CHOICE=$default; fi
203     storage_nat="s<STORAGE_NAT_IP<$CHOICE<g"
204
205     default='mongodb://nodes:M2qlif8rdtKtBYil@129.194.185.47:27017/celery'
206     readLine "Please enter MongoDB connection [$default]"
207     if [ ! "$CHOICE" ]; then CHOICE=$default; fi
208     mongo="s<MONGO<$CHOICE<g"
209
210     default='amqp://nodes:OZy23i00D4UpYS2k@129.194.185.47:5672/celery'
211     readLine "Please enter RabbitMQ connection [$default]"
212     if [ ! "$CHOICE" ]; then CHOICE=$default; fi
213     rabbit="s<RABBIT<$CHOICE<g"
214
215     sed "$api_nat;$storage;$storage_nat;$mongo;$rabbit" < "$cfg_bernex.template" > "$cfg_bernex"
216
217     pecho 'Deploy Transform (1 instance)'
218     mecho "Using user define Transform configuration : $cfg_bernex"
219     juju deploy --environment 'bernex' --config "$cfg_bernex" \
220         --repository=charms/ local:$RELEASE/oscied-transform || xecho '1'
221 }
222
223 osciedDemoScenario_amazon()
224 {
225     techo "4/4 Deploy services on Amazon"
226
227     pecho 'Cleanup and bootstrap juju amazon environment'
228     yesOrNo $false 'do it now'
229     if [ $REPLY -eq $true ]; then
230         juju destroy-environment --environment 'amazon'
231         juju bootstrap --environment 'amazon'
232     fi
233
234     mecho '[WARNING] Continue only when services deployed on the MaaS cluster are up and ready !'
235     pause
236
237     pecho 'Gather services configuration for amazon deployment'
238
239     yesOrNo $true 'Is Orchestra behind a NAT/FW'
240     if [ $REPLY -eq $true ]; then
241         default='129.194.185.47:5000'
242         readLine "Please enter orchestra api public (NATed) socket [$default]"
243         if [ ! "$CHOICE" ]; then CHOICE=$default; fi
244     else
245         CHOICE=''
246     fi
247     api_socket=$CHOICE
248     api_nat="s<API_NAT_SOCKET<$CHOICE<g"
249
250     yesOrNo $true 'Is Storage behind a NAT/FW'
251     if [ $REPLY -eq $true ]; then
252         default='129.194.185.47'
253         readLine "Please enter storage public (NATED) IP address [$default]"
254         if [ ! "$CHOICE" ]; then CHOICE=$default; fi
255     else
256         CHOICE=''
257     fi
258     storage_nat="s<STORAGE_NAT_IP<$CHOICE<g"
259
260     pecho 'Generate services configuration for amazon deployment'
261
262     get_unit_config 'oscied-transform' '1' 'storage_ip'
263     if [ ! "$REPLY" ]; then xecho 'Unable to detect storage internal IP address'; fi
264     storage="s<STORAGE_IP<$REPLY<g"
265
266     get_unit_config 'oscied-transform' '1' 'mongo_connection'
267     if [ ! "$REPLY" ]; then xecho 'Unable to detect MongoDB connection'; fi
268     mongo="s<MONGO<$REPLY<g"
269
270     get_unit_config 'oscied-transform' '1' 'rabbit_connection'
271     if [ ! "$REPLY" ]; then xecho 'Unable to detect RabbitMQ connection'; fi

```

```

272 rabbit="s<RABBIT<$REPLY<g"
273
274 if [ "$api_nat" ]; then
275     api_ip=$(echo $api_socket | cut -d':' -f1)
276     mongo=$(echo $mongo | sed "s<@[^:]*:<@$api_ip:<")
277     rabbit=$(echo $rabbit | sed "s<@[^:]*:<@$api_ip:<")
278 fi
279
280 sed "$api_nat;$storage;$storage_nat;$mongo;$rabbit" < "$cfg_amazon.template" > "$cfg_amazon"
281
282 cat "$cfg_amazon"
283 yesOrNo $true 'deploy with this configuration'
284 if [ $REPLY -eq $true ]; then
285     pecho 'Deploy Transform (1 instance)'
286     mecho "Using user define Transform configuration : $cfg_amazon"
287     juju deploy --environment 'amazon' --config "$cfg_amazon" \
288         --repository=charms/ local:$RELEASE/oscied-transform oscied-transform-demo || xecho '1'
289
290     pecho 'Deploy Publisher (1 instance)'
291     mecho "Using user define Publisher configuration : $cfg_amazon"
292     juju deploy --environment 'amazon' --config "$cfg_amazon" \
293         --repository=charms/ local:$RELEASE/oscied-publisher oscied-publisher-demo || xecho '1'
294     juju expose --environment 'amazon' oscied-publisher-demo || xecho '2'
295 fi
296 }

```

scenarios/osciedLocal.sh

```

1 osciedLocalDescription='Launch oscied (minimal setup) locally (LXC provider)'
2 osciedLocalScenario()
3 {
4     cd "$COMPONENTS_PATH/juju" || xecho "Unable to find path $COMPONENTS_PATH/juju"
5     cfg="$CONFIG_JUJU_PATH/osciedLocal.yaml"
6     cp -f "$cfg" "$CONFIG_GEN_CONFIG_FILE"
7
8     techo '1/5 Cleanup and bootstrap juju environment'
9
10    juju destroy-environment --environment 'local'
11    juju bootstrap --environment 'local'
12
13    techo '2/5 Deploy services on this computer'
14
15    pecho 'Deploy Orchestra (1 instance)'
16    yesOrNo $true 'deploy it now'
17    if [ $REPLY -eq $true ]; then
18        if [ -f "$cfg" ]; then
19            mecho "Using user define Orchestra configuration : $cfg"
20            juju deploy --environment 'local' --config "$cfg" \
21                --repository=charms/ local:$RELEASE/oscied-orchestra || xecho '1'
22    else
23        mecho 'Using default Orchestra configuration'
24        juju deploy --environment 'local' \
25            --repository=charms/ local:$RELEASE/oscied-orchestra || xecho '1'
26    fi
27    juju expose --environment 'local' oscied-orchestra || xecho '2'
28 fi
29
30    pecho 'Deploy Web UI (1 instance)'
31    yesOrNo $true 'deploy it now'
32    if [ $REPLY -eq $true ]; then
33        if [ -f "$cfg" ]; then
34            mecho "Using user define Web UI configuration : $cfg"
35            juju deploy --environment 'local' --config "$cfg" \
36                --repository=charms/ local:$RELEASE/oscied-webui || xecho '1'
37    else
38        mecho 'Using default Web UI configuration'
39        juju deploy --environment 'local' \
40            --repository=charms/ local:$RELEASE/oscied-webui || xecho '1'
41    fi
42    juju expose --environment 'local' oscied-webui || xecho '2'
43 fi
44

```

```

45  pecho 'Deploy Storage (1 instance)'
46  yesOrNo $true 'deploy it now'
47  if [ $REPLY -eq $true ]; then
48    if [ -f "$cfg" ]; then
49      mecho "Using user define Storage configuration : $cfg"
50      juju deploy --environment 'local' --config "$cfg" \
51        --repository=charms/ local:$RELEASE/oscied-storage || xecho '1'
52  else
53    mecho 'Using default Storage configuration'
54    juju deploy --environment 'local' \
55      --repository=charms/ local:$RELEASE/oscied-storage || xecho '1'
56  fi
57  juju expose --environment 'local' oscied-storage || xecho '2'
58 fi
59
60 pecho 'Deploy Transform (1 instance)'
61 yesOrNo $true 'deploy it now'
62 if [ $REPLY -eq $true ]; then
63  if [ -f "$cfg" ]; then
64    mecho "Using user define Transform configuration : $cfg"
65    juju deploy --environment 'local' --config "$cfg" \
66      --repository=charms/ local:$RELEASE/oscied-transform || xecho '1'
67  else
68    mecho 'Using default Transform configuration'
69    juju deploy --environment 'local' \
70      --repository=charms/ local:$RELEASE/oscied-transform || xecho '1'
71  fi
72 fi
73
74 pecho 'Deploy Publisher (1 instance)'
75 yesOrNo $true 'deploy it now'
76 if [ $REPLY -eq $true ]; then
77  if [ -f "$cfg" ]; then
78    mecho "Using user define Publisher configuration : $cfg"
79    juju deploy --environment 'local' --config "$cfg" \
80      --repository=charms/ local:$RELEASE/oscied-publisher || xecho '1'
81  else
82    mecho 'Using default Publisher configuration'
83    juju deploy --environment 'local' \
84      --repository=charms/ local:$RELEASE/oscied-publisher || xecho '1'
85  fi
86  juju expose --environment 'local' oscied-publisher || xecho '2'
87 fi
88
89 pecho 'Deploy haproxy (1 instance)'
90 yesOrNo $false 'deploy it now'
91 if [ $REPLY -eq $true ]; then
92  juju deploy --environment 'local' cs:precise/haproxy || xecho '2'
93  juju expose --environment 'local' haproxy || xecho '3'
94 fi
95
96 techo "3/5 Add relation between Storage and other services"
97
98 pecho 'Add-relation Storage <-> Transform'
99 yesOrNo $true 'add it now'
100 if [ $REPLY -eq $true ]; then
101  juju add-relation --environment 'local' oscied-storage oscied-transform
102 fi
103
104 pecho 'Add-relation Storage <-> Publisher'
105 yesOrNo $true 'add it now'
106 if [ $REPLY -eq $true ]; then
107  juju add-relation --environment 'local' oscied-storage oscied-publisher
108 fi
109
110 pecho 'Add-relation Storage <-> Orchestra'
111 yesOrNo $true 'add it now'
112 if [ $REPLY -eq $true ]; then
113  juju add-relation --environment 'local' oscied-storage oscied-orchestra
114 fi
115
116 pecho 'Add-relation Storage <-> Web UI'
117 yesOrNo $true 'add it now'

```

```

118 if [ $REPLY -eq $true ]; then
119     juju add-relation --environment 'local' oscied-storage oscied-webui
120 fi
121
122 techo "4/5 Add relation between Orchestra and other services"
123
124 pecho 'Add-relation Orchestra <-> Transform'
125 yesOrNo $true 'add it now'
126 if [ $REPLY -eq $true ]; then
127     juju add-relation --environment 'local' oscied-orchestra:transform oscied-transform:transform
128 fi
129
130 pecho 'Add-relation Orchestra <-> Publisher'
131 yesOrNo $true 'add it now'
132 if [ $REPLY -eq $true ]; then
133     juju add-relation --environment 'local' oscied-orchestra:publisher oscied-publisher:publisher
134 fi
135
136 pecho 'Add-relation Orchestra <-> Web UI'
137 yesOrNo $true 'add it now'
138 if [ $REPLY -eq $true ]; then
139     juju add-relation --environment 'local' oscied-orchestra:api oscied-webui:api
140 fi
141
142 techo '5/5 Add relation between Web UI and HA Proxy'
143
144 pecho 'Add-relation haproxy <-> Web UI'
145 yesOrNo $false 'add it now'
146 if [ $REPLY -eq $true ]; then
147     juju unexpose --environment 'local' oscied-webui
148     juju add-relation --environment 'local' haproxy oscied-webui
149 fi
150 }

```

6.8 OSCIED - Components Common Library

6.8.1 lib/Callback.py

```

1 import requests
2 from urlparse import urlparse, ParseResult
3 from Utilities import json2object, object2json
4
5
6 class Callback(object):
7
8     def __init__(self, url, username, password):
9         self.url = url
10        self.username = username
11        self.password = password
12
13    def is_valid(self, raise_exception):
14        # FIXME check fields
15        return True
16
17    def replace_netloc(self, netloc):
18        url = urlparse(self.url)
19        url = ParseResult(url.scheme, netloc, url.path, url.params, url.query, url.fragment)
20        self.url = url.geturl()
21
22    def post(self, data_json):
23        # return requests.post(self.url, data_json, auth=(self.username, self.password))
24        headers = {'Content-type': 'application/json', 'Accept': 'text/plain'}
25        return requests.post(self.url, headers=headers, data=data_json,
26                            auth=(self.username, self.password))
27
28    @staticmethod
29    def load(json):
30        callback = Callback(None, None, None)
31        json2object(json, callback)

```

```

32         return callback
33
34 CALLBACK_TEST = Callback('http://127.0.0.1:5000/media', 'toto', '1234')
35
36 # -----
37
38 if __name__ == '__main__':
39     print object2json(CALLBACK_TEST, True)
40     CALLBACK_TEST.is_valid(True)
41     print CALLBACK_TEST.url
42     CALLBACK_TEST.replace_netloc('129.194.185.47:5003')
43     assert CALLBACK_TEST.url == 'http://129.194.185.47:5003/media'
44     print str(Callback.load(object2json(CALLBACK_TEST, False)))

```

6.8.2 lib/FFmpeg.py

```

1 import fcntl
2 import os
3 import re
4 import select
5 import shlex
6 import subprocess
7 import time
8
9
10 def get_media_duration(filename):
11     cmd = 'ffmpeg -i "%s"' % (filename)
12     pipe = subprocess.Popen(shlex.split(cmd), stderr=subprocess.PIPE, close_fds=True)
13     duration = re.search('Duration: (?P<duration>\S+)', pipe.stderr.read())
14     return None if not duration else duration.group('duration')
15
16
17 def encode(in_filename, out_filename, encoder_string, overwrite, sleep_time=1, callback=None):
18     if os.path.exists(out_filename):
19         if not overwrite:
20             return False
21         os.unlink(out_filename)
22     cmd = 'ffmpeg -i "%s" ' + encoder_string + ' "%s"' % (in_filename, out_filename)
23     pipe = subprocess.Popen(shlex.split(cmd), stderr=subprocess.PIPE, close_fds=True)
24
25     # http://stackoverflow.com/questions/1388753/how-to-get-output-from-subprocess-popen
26     fcntl.fcntl(pipe.stderr.fileno(), fcntl.F_SETFL,
27                 fcntl.fcntl(pipe.stderr.fileno(), fcntl.F_GETFL) | os.O_NONBLOCK)
28
29     # frame= 2071 fps= 0 q=-1.0 size= 34623kB time=00:01:25.89 bitrate=3302.3kbits/s
30     regex = re.compile("frame=\s*(?P<frame>\d+)" +
31                         "\s+fps=\s*(?P<fps>\d+)" +
32                         "\s+q=\s*(?P<q>\S+)" +
33                         "\s+\$size=\s*(?P<size>\S+)" +
34                         "\s+time=\s*(?P<time>\S+)" +
35                         "\s+bitrate=\s*(?P<bitrate>\S+)")
36
37     while True:
38         readx = select.select([pipe.stderr.fileno()], [], [])[0]
39         if readx:
40             chunk = pipe.stderr.read()
41             if chunk == '':
42                 break
43             match = regex.match(chunk)
44             if match and callback:
45                 callback(match.groupdict())
46             time.sleep(sleep_time)
47     return True
48
49 #def test_callback(dict):
50 #    print dict
51 #
52 #if __name__ == '__main__':
53 #    print FFmpeg.duration(movie)
54 #    FFmpeg.encode(movie, movie_out, '-acodec copy -vcodec copy', True, test_callback)

```

6.8.3 lib/Media.py

```

1 import uuid
2 from OrchestraConfig import ORCHESTRA_CONFIG_TEST
3 from Utilities import json2object, object2json, valid_filename, valid_uuid
4
5
6 class Media(object):
7
8     def __init__(self, _id, user_id, parent_id, uri, public_uris, virtual_filename, metadata,
9                  status):
10        if not _id:
11            _id = str(uuid.uuid4())
12        self._id = _id
13        self.user_id = user_id
14        self.parent_id = parent_id
15        self.uri = uri
16        self.public_uris = public_uris
17        try:
18            self.virtual_filename = str(virtual_filename).replace(' ', '_')
19        except:
20            self.virtual_filename = None
21        self.metadata = metadata
22        self.status = status
23
24    def is_valid(self, raise_exception):
25        if not valid_uuid(self._id, False):
26            if raise_exception:
27                raise TypeError(self.__class__.__name__ + ' : _id is not a valid uuid string')
28            return False
29        if hasattr(self, 'user_id') and not valid_uuid(self.user_id, False):
30            if raise_exception:
31                raise TypeError(self.__class__.__name__ + ' : user_id is not a valid uuid string')
32            return False
33        #FIXME check use if loaded
34        if hasattr(self, 'parent_id') and not valid_uuid(self.parent_id, True):
35            if raise_exception:
36                raise TypeError(self.__class__.__name__ + ' : parent_id is not a valid uuid string')
37            return False
38        #FIXME check parent if loaded
39        #FIXME check uri
40        #FIXME check public_uris
41        if not valid_filename(self.virtual_filename):
42            if raise_exception:
43                raise TypeError(self.__class__.__name__ + ' : virtual_filename is not a valid filename')
44            return False
45        #FIXME check metadata
46        if not self.status in ('PENDING', 'READY', 'PUBLISHED', 'DELETED'):
47            if raise_exception:
48                raise TypeError(self.__class__.__name__ + ' : status is not a valid status string')
49            return False
50        return True
51
52    def add_metadata(self, key, value, overwrite):
53        if overwrite or not key in self.metadata:
54            self.metadata[key] = value
55
56    def get_metadata(self, key):
57        return self.metadata[key] if key in self.metadata else None
58
59    #def detect_codecs(self, storage_path):
60    #''' Update media's metadata based on file's attribute '''
61
62    def load_fields(self, user, parent):
63        self.user = user
64        self.parent = parent
65        delattr(self, 'user_id')
66        delattr(self, 'parent_id')
67
68    @staticmethod
69    def load(json):
70        media = Media(None, None, None, None, None, None, None, None)
71        json2object(json, media)

```

```

72         return media
73
74 MEDIA_TEST = Media(None, str(uuid.uuid4()), str(uuid.uuid4()), None, None, 'tabby.mpg',
75                     {'title': "Tabby's adventures $1", 'description': 'My cat drinking water'},
76                     'PENDING')
77 MEDIA_TEST.uri = ORCHESTRA_CONFIG_TEST.storage_uri + '/medias/' + MEDIA_TEST.user_id + '/' + MEDIA_TEST._id
78 MEDIA_TEST.add_metadata('title', 'not authorized overwrite', False)
79 MEDIA_TEST.add_metadata('size', 4096, True)
80
81 # -----
82
83 if __name__ == '__main__':
84     print object2json(MEDIA_TEST, True)
85     MEDIA_TEST.is_valid(True)
86     print str(Media.load(object2json(MEDIA_TEST, False)))

```

6.8.4 lib/Orchestra.py

```

1 import logging
2 import pymongo
3 import Publisher
4 import Transform
5 from Storage import Storage
6 from celery import states
7 #from celery import current_app
8 #from celery.task.control import inspect
9 from celery.task.control import revoke
10 #from celery.events.state import state
11 from Callback import Callback
12 from Media import Media
13 from PublishJob import PublishJob
14 from TransformProfile import TransformProfile
15 from TransformJob import TransformJob
16 from User import User
17 from Utilities import object2json, datetime_now, UUID_ZERO, valid_uuid
18
19
20 class Orchestra(object):
21
22     def __init__(self, config):
23         self.config = config
24         self._db = pymongo.Connection(config.mongo_connection)['orchestra']
25         self.root_user = User(UUID_ZERO, 'root', 'oscied', 'root@oscied.org',
26                               self.config.root_secret, True)
27         self.nodes_user = User(UUID_ZERO, 'nodes', 'oscied', 'nodes@oscied.org',
28                               self.config.nodes_secret, False)
29
30     # -----
31
32     def flush_db(self):
33         self._db.drop_collection('users')
34         self._db.drop_collection('medias')
35         self._db.drop_collection('transform_profiles')
36         self._db.drop_collection('transform_jobs')
37         self._db.drop_collection('publish_jobs')
38         self._db.drop_collection('unpublish_jobs')
39         logging.info("Orchestra database's collections dropped !")
40
41     # -----
42
43     def save_user(self, user):
44         user.is_valid(True)
45         if self.get_user({'mail': user.mail, '_id': {'$ne': user._id}}, {'_id': 1}):
46             raise ValueError('The email address ' + user.mail + ' is already used by another user.')
47         self._db.users.save(user.__dict__)
48
49     def get_user(self, specs, fields=None):
50         entity = self._db.users.find_one(specs, fields)
51         if not entity:
52             return None
53         return User.load(object2json(entity, False))

```

```

54
55     def delete_user(self, user):
56         if valid_uuid(user, False):
57             user = self.get_user({'_id': user}, {'secret': 0})
58             user.is_valid(True)
59             self._db.users.remove({'_id': user._id})
60
61     def get_users(self, specs=None, fields=None):
62         users = []
63         for entity in list(self._db.users.find(specs, fields)):
64             users.append(User.load(object2json(entity, False)))
65     return users
66
67     def get_users_count(self, specs=None):
68         return self._db.users.find(specs, {'_id': 1}).count()
69
70 # -----
71
72     def save_media(self, media):
73         media.is_valid(True)
74         if self.get_media({'uri': media.uri, '_id': {'$ne': media._id}}, {'_id': 1}):
75             raise ValueError('The media uri ' + media.uri + ' is already used by another media.')
76         if not media.get_metadata('title'):
77             raise ValueError('Title key is required in media metadata.')
78         if media.status not in ('DELETED'):
79             size, duration = Storage.add_media(self.config, media)
80         else:
81             size, duration = (0, 0)
82         media.add_metadata('size', size, True)
83         if duration:
84             media.add_metadata('duration', duration, True)
85             media.add_metadata('add_date', datetime.now(), True)
86         self._db.medias.save(media.__dict__)
87
88     def get_media(self, specs, fields=None, load_fields=False):
89         entity = self._db.medias.find_one(specs, fields)
90         if not entity:
91             return None
92         media = Media.load(object2json(entity, False))
93         if load_fields:
94             media.load_fields(self.get_user({'_id': media.user_id}, {'secret': 0}),
95                               self.get_media({'_id': media.parent_id}))
96     return media
97
98     def delete_media(self, media):
99         if valid_uuid(media, False):
100            media = self.get_media({'_id': media})
101            media.is_valid(True)
102            job = self.get_transform_job({'media_in_id': media._id}, append_result=True)
103            if job and (job.status in states.UNREADY_STATES or job.status == 'PROGRESS'):
104                raise ValueError('Cannot delete the media, it is actually in use by transform ' +
105                                'job with id ' + job._id + ' and status ' + job.status + '.')
106            job = self.get_publish_job({'media_id': media._id}, append_result=True)
107            if job and (job.status in states.UNREADY_STATES or job.status == 'PROGRESS'):
108                raise ValueError('Cannot delete the media, it is actually in use by publish ' +
109                                'job with id ' + job._id + ' and status ' + job.status + '.')
110            media.status = 'DELETED'
111            self.save_media(media)
112            #self._db.medias.remove({'_id': media._id})
113            Storage.delete_media(self.config, media)
114
115     def get_medias(self, specs=None, fields=None, load_fields=False):
116         medias = []
117         for entity in list(self._db.medias.find(specs, fields)):
118             media = Media.load(object2json(entity, False))
119             if load_fields:
120                 media.load_fields(self.get_user({'_id': media.user_id}, {'secret': 0}),
121                                   self.get_media({'_id': media.parent_id}))
122             medias.append(media)
123     return medias
124
125     def get_medias_count(self, specs=None):
126         return self._db.medias.find(specs, {'_id': 1}).count()

```

```

127
128 # -----
129
130     def save_transform_profile(self, profile):
131         profile.is_valid(True)
132         if self.get_transform_profile(
133             {'title': profile.title, '_id': {'$ne': profile._id}}, {'_id': 1}):
134             raise ValueError('Duplicate transform profile title ' + profile.title + '.')
135         self._db.transform_profiles.save(profile.__dict__)
136
137     def get_transform_profile(self, specs, fields=None):
138         entity = self._db.transform_profiles.find_one(specs, fields)
139         if not entity:
140             return None
141         return TransformProfile.load(object2json(entity, False))
142
143     def delete_transform_profile(self, profile):
144         if valid_uuid(profile, False):
145             profile = self.get_profile({'_id': profile})
146             profile.is_valid(True)
147             self._db.transform_profiles.remove({'_id': profile._id})
148
149     def get_transform_profiles(self, specs=None, fields=None):
150         profiles = []
151         for entity in list(self._db.transform_profiles.find(specs, fields)):
152             profiles.append(TransformProfile.load(object2json(entity, False)))
153         return profiles
154
155     def get_transform_profiles_count(self, specs=None):
156         return self._db.transform_profiles.find(specs, {'_id': 1}).count()
157
158 # -----
159
160     def get_transform_queues(self):
161         return self.config.transform_queues
162
163     def launch_transform_job(self, user_id, media_in_id, profile_id, virtual_filename, metadata,
164                             queue, callback_url):
165         user = self.get_user({'_id': user_id}, {'secret': 0})
166         if not user:
167             raise IndexError('No user with id ' + user_id + '.')
168         media_in = self.get_media({'_id': media_in_id})
169         if not media_in: # FIXME maybe a media access control here
170             raise IndexError('No media with id ' + media_in_id + '.')
171         profile = self.get_transform_profile({'_id': profile_id})
172         if not profile: # FIXME maybe a profile access control here
173             raise IndexError('No profile with id ' + profile_id + '.')
174         if not queue in self.config.transform_queues:
175             raise IndexError('No transform queue with name ' + queue + '.')
176         if not media_in.status in('READY', 'PUBLISHED'):
177             raise NotImplementedError('Cannot launch the job, input media status is ' +
178                                     media_in.status + '.')
179         media_out = Media(None, user_id, media_in_id, None, None, virtual_filename, metadata,
180                           'PENDING')
181         media_out.uri = Storage.media_uri(self.config, media_out, True)
182         self.save_media(media_out) # Save pending output media
183         # FIXME create a one-time password to avoid fixed secret authentication ...
184         callback = Callback(self.config.api_url + callback_url, 'node', self.config.nodes_secret)
185         result = Transform.transform_job.apply_async(
186             args=(
187                 object2json(user, False), object2json(media_in, False),
188                 object2json(media_out, False), object2json(profile, False),
189                 object2json(callback, False)),
190             queue=queue)
191         if not result.id:
192             raise ValueError('Unable to transmit job to workers of queue ' + queue + '.')
193         logging.info('New transform job ' + result.id + ' launched.')
194         job = TransformJob(result.id, user._id, media_in._id, media_out._id, profile._id)
195         job.add_statistic('add_date', datetime.now(), True)
196         self._db.transform_jobs.save(job.__dict__)
197         return result.id
198
199     def get_transform_job(self, specs, fields=None, load_fields=False, append_result=True):

```

```

200     entity = self._db.transform_jobs.find_one(specs, fields)
201     if not entity:
202         return None
203     job = TransformJob.load(object2json(entity, False))
204     if load_fields:
205         job.load_fields(self.get_user({'_id': job.user_id}, {'secret': 0}),
206                         self.get_media({'_id': job.media_in_id}),
207                         self.get_media({'_id': job.media_out_id}),
208                         self.get_transform_profile({'_id': job.profile_id}))
209     if append_result:
210         job.append_async_result()
211     return job
212
213 def update_transform_job(self, job):
214     raise NotImplementedError('maybe in a near future.')
215
216 def revoke_transform_job(self, job, terminate=False, remove=False, delete_media=False):
217     """ This do not delete jobs from jobs database (if remove=False) but set revoked attribute
218     in jobs database and broadcast revoke request to transform units with Celery. If the job is
219     actually running it will be cancelled if terminated = True. The output media will be deleted
220     if corresponding argument, delete_media = True. """
221     # FIXME verify that no pending jobs needs the media that will be created by the job !
222     if valid_uuid(job, False):
223         job = self.get_transform_job({'_id': job})
224     job.is_valid(True)
225     if job.revoked:
226         raise ValueError('Transform job ' + job._id + ' is already revoked !')
227     if job.status in states.READY_STATES:
228         raise ValueError('Cannot revoke a transform job with status ' + job.status + '.')
229     job.revoked = True
230     revoke(job._id, terminate=terminate)
231     self._db.transform_jobs.save(job.__dict__)
232     if delete_media and valid_uuid(job.media_out_id, False):
233         self.delete_media(job.media_out_id)
234     if remove:
235         self._db.transform_jobs.remove({'_id': job._id})
236
237 def get_transform_jobs(self, specs=None, fields=None, load_fields=False, append_result=True):
238     jobs = []
239     for entity in list(self._db.transform_jobs.find(specs, fields)):
240         job = TransformJob.load(object2json(entity, False))
241         if load_fields:
242             job.load_fields(self.get_user({'_id': job.user_id}, {'secret': 0}),
243                             self.get_media({'_id': job.media_in_id}),
244                             self.get_media({'_id': job.media_out_id}),
245                             self.get_transform_profile({'_id': job.profile_id}))
246         if append_result:
247             job.append_async_result()
248         jobs.append(job)
249     return jobs
250     # FIXME this is celery's way to do that:
251     #for task in state.itertasks():
252     #    print task
253     #for entity in entities:
254     #    job = get_transform_job_helper(entity._id)
255
256 def get_transform_jobs_count(self, specs=None):
257     return self._db.transform_jobs.find(specs, {'_id': 1}).count()
258
259 # -----
260
261 def get_publisher_queues(self):
262     return self.config.publisher_queues
263
264 def launch_publish_job(self, user_id, media_id, queue, callback_url):
265     user = self.get_user({'_id': user_id}, {'secret': 0})
266     if not user:
267         raise IndexError('No user with id ' + user_id + '.')
268     media = self.get_media({'_id': media_id})
269     if not media: # FIXME maybe a media access control here
270         raise IndexError('No media with id ' + media_id + '.')
271     if not queue in self.config.publisher_queues:
272         raise IndexError('No publisher queue with name ' + queue + '.')

```

```

273     if not media.status in ('READY',):
274         raise NotImplementedError('Cannot launch the job, input media status is ' +
275                                     media.status + '.')
276     other = self.get_publish_job({'media_id': media._id})
277     if other and other.status not in states.READY_STATES and not other.revoked:
278         raise NotImplementedError('Cannot launch the job, input media will be published by ' +
279                                     + 'another job with id ' + other._id + '.')
280     # FIXME create a one-time password to avoid fixed secret authentication ...
281     callback = Callback(self.config.api_url + callback_url, 'node', self.config.nodes_secret)
282     result = Publisher.publish_job.apply_async(
283         args=(
284             object2json(user, False), object2json(media, False), object2json(callback, False)),
285             queue=queue)
286     if not result.id:
287         raise ValueError('Unable to transmit job to workers of queue ' + queue + '.')
288     logging.info('New publish job ' + result.id + ' launched.')
289     job = PublishJob(result.id, user._id, media._id, None)
290     job.add_statistic('add_date', datetime.now(), True)
291     self._db.publish_jobs.save(job.__dict__)
292     return result.id
293
294 def get_publish_job(self, specs, fields=None, load_fields=False, append_result=True):
295     entity = self._db.publish_jobs.find_one(specs, fields)
296     if not entity:
297         return None
298     job = PublishJob.load(object2json(entity, False))
299     if load_fields:
300         job.load_fields(self.get_user({'_id': job.user_id}, {'secret': 0}),
301                         self.get_media({'_id': job.media_id}))
302     if append_result:
303         job.append_async_result()
304     return job
305
306 def update_publish_job(self, job):
307     raise NotImplementedError('maybe in a near future.')
308
309 def revoke_publish_job(self, job, terminate=False, remove=False):
310     """ This do not delete jobs from jobs database (if remove=False) but set revoked attribute
311     in tasks database and broadcast revoke request to publisher units with celery. If the job is
312     actually running it will be cancelled if terminated = True. The output media will be deleted
313     """
314     if valid_uuid(job, False):
315         job = self.get_publish_job({'_id': job})
316     job.is_valid(True)
317     if job.revoked:
318         raise ValueError('Publish job ' + job._id + ' is already revoked !')
319     if job.status in states.READY_STATES:
320         raise ValueError('Cannot revoke a publish job with status ' + job.status + '.')
321     job.revoked = True
322     revoke(job._id, terminate=terminate)
323     self._db.publish_jobs.save(job.__dict__)
324     if remove:
325         self._db.publish_jobs.remove({'_id': job._id})
326
327 def get_publish_jobs(self, specs=None, fields=None, load_fields=False, append_result=True):
328     jobs = []
329     for entity in list(self._db.publish_jobs.find(specs, fields)):
330         job = PublishJob.load(object2json(entity, False))
331         if load_fields:
332             job.load_fields(self.get_user({'_id': job.user_id}, {'secret': 0}),
333                             self.get_media({'_id': job.media_id}))
334         if append_result:
335             job.append_async_result()
336         jobs.append(job)
337     return jobs
338     # FIXME this is celery's way to do that:
339     #for task in state.itertasks():
340     #    print task
341     #for entity in entities:
342     #    job = get_publish_job_helper(entity._id)
343
344 def get_publish_jobs_count(self, specs=None):
345     return self._db.publish_jobs.find(specs, {'_id': 1}).count()

```

```

346
347 # -----
348
349 def transform_callback(self, job_id, status):
350     job = self.get_transform_job({'_id': job_id})
351     if not job:
352         raise IndexError('No transform job with id ' + job_id + '.')
353     media_out = self.get_media({'_id': job.media_out_id})
354     if not media_out:
355         raise IndexError('Unable to find output media with id ' + job.media_out_id + '.')
356     if status == 'SUCCESS':
357         media_out.status = 'READY'
358         self.save_media(media_out)
359         logging.info('%s Media %s is now READY' % (job_id, media_out.virtual_filename))
360     else:
361         self.delete_media(media_out)
362         job.add_statistic('error_details', status.replace('\n', '\\n'), True)
363         self._db.transform_jobs.save(job.__dict__)
364         logging.info('%s Error: %s' % (job_id, status))
365         logging.info('%s Media %s is now deleted' % (job_id, media_out.virtual_filename))
366
367 def publish_callback(self, job_id, publish_uri, status):
368     job = self.get_publish_job({'_id': job_id})
369     if not job:
370         raise IndexError('No publish job with id ' + job_id + '.')
371     media = self.get_media({'_id': job.media_id})
372     if not media:
373         raise IndexError('Unable to find media with id ' + job.media_id + '.')
374     if status == 'SUCCESS':
375         media.status = 'PUBLISHED'
376         if not media.public_uris:
377             media.public_uris = {}
378         job.publish_uri = publish_uri
379         media.public_uris[job_id] = publish_uri
380         self._db.publish_jobs.save(job.__dict__)
381         self.save_media(media)
382         logging.info('%s Media %s is now PUBLISHED' % (job_id, media.virtual_filename))
383     else:
384         job.add_statistic('error_details', status.replace('\n', '\\n'), True)
385         self._db.publish_jobs.save(job.__dict__)
386         logging.info('%s Error: %s' % (job_id, status))
387         logging.info('%s Media %s is not modified' % (job_id, media.virtual_filename))

```

6.8.5 lib/OrchestraConfig.py

```

1 import logging
2 from Utilities import json2object, jsonfile2object, object2json
3
4
5 class OrchestraConfig(object):
6
7     def __init__(self, verbose, api_url, root_secret, nodes_secret, mongo_connection,
8                  rabbit_connection, storage_ip, storage_fstype, storage_mountpoint,
9                  storage_options, storage_path):
10        self.verbose = verbose
11        self.api_url = api_url
12        self.root_secret = root_secret
13        self.nodes_secret = nodes_secret
14        self.mongo_connection = mongo_connection
15        self.rabbit_connection = rabbit_connection
16        self.storage_ip = storage_ip
17        self.storage_fstype = storage_fstype
18        self.storage_mountpoint = storage_mountpoint
19        self.storage_options = storage_options
20        self.storage_path = storage_path
21
22    @property
23    def log_level(self):
24        return logging.DEBUG if self.verbose else logging.INFO
25
26    @property
27    def storage_uri(self):

```

```

28     if self.storage_fstype and self.storage_ip and self.storage_mountpoint:
29         return self.storage_fstype + '://' + self.storage_ip + '/' + self.storage_mountpoint
30     return None
31
32     @property
33     def transform_queues(self):
34         return ('transform_private', 'transform_amazon',)
35
36     @property
37     def publisher_queues(self):
38         return ('publisher_private', 'publisher_amazon',)
39
40     #def write(self, filename):
41     #    with io.open(filename, 'w', encoding='utf-8') as outfile:
42     #        json.dump(self, outfile)
43
44     @staticmethod
45     def read(filename):
46         config = OrchestraConfig(None, None, None)
47         jsonfile2object(filename, config)
48         return config
49
50     @staticmethod
51     def load(json):
52         config = OrchestraConfig(None, None, None)
53         json2object(json, config)
54         return config
55
56 ORCHESTRA_CONFIG_TEST = OrchestraConfig(True, 'http://127.0.0.1:5000', 'toto', 'abcd', '...', '...',
57                                         '10.1.1.2', 'glusterfs', 'medias_volume', '', '/mnt/medias')
58
59 # Main -----
60
61 if __name__ == '__main__':
62
63     print object2json(ORCHESTRA_CONFIG_TEST, True)
64     print ORCHESTRA_CONFIG_TEST.storage_uri
65     print str(OrchestraConfig.load(object2json(ORCHESTRA_CONFIG_TEST, False)))

```

6.8.6 lib/Publisher.py

```

1  import time
2  import os
3  from celery import current_task
4  from celery.decorators import task
5  from Callback import Callback
6  from Media import Media
7  from Storage import Storage
8  from PublisherConfig import PublisherConfig
9  from User import User
10 from Utilities import object2json, datetime_now
11
12
13 @task(name='Publisher.publish_job')
14 def publish_job(user_json, media_json, callback_json):
15
16     RATIO_DELTA = 0.05 # Update status if at least 5% of progress
17     TIME_DELTA = 1      # Update status if at least 1 second(s) elapsed
18
19     try:
20         # Avoid 'referenced before assignment'
21         media_path = None
22         publish_path = None
23         request = current_task.request
24
25         # Let's the task begin !
26         start_date = datetime_now()
27         start_time = time.time()
28         print('%s Publish job started' % (request.id))
29
30         # Load and check task parameters

```

```

31     user = User.load(user_json)
32     media = Media.load(media_json)
33     callback = Callback.load(callback_json)
34     user.is_valid(True)
35     media.is_valid(True)
36     callback.is_valid(True)
37
38     # Read current configuration to translate files uri to local paths
39     config = PublisherConfig.read('config.json')
40     print object2json(config, True)
41
42     # Update callback socket according to configuration
43     if config.api_nat_socket and len(config.api_nat_socket) > 0:
44         callback.replace_netloc(config.api_nat_socket)
45
46     # Verify that media file can be accessed and create output path
47     media_path = Storage.media_path(config, media, False)
48     if not media_path:
49         raise NotImplementedError('Media will not be readed from shared storage : %s' %
50                               media.uri)
51     (publish_path, publish_uri) = Storage.publish_point(config, media)
52     Storage.create_file_directory(publish_path)
53
54     # Initialize block-based copy
55     block_size = 1024 * 1024
56     media_file = open(media_path, "rb")
57     publish_file = open(publish_path, "wb")
58     media_size = os.stat(media_path).st_size
59
60     # Block-based copy loop
61     block_pos = 0
62     prev_ratio = 0
63     prev_time = 0
64     while True:
65         block = media_file.read(block_size)
66         ratio = float(block_pos) / media_size
67         elapsed_time = time.time() - start_time
68         if ratio - prev_ratio > RATIO_DELTA and elapsed_time - prev_time > TIME_DELTA:
69             prev_ratio = ratio
70             prev_time = elapsed_time
71             eta_time = int(elapsed_time * (1 - ratio) / ratio) if ratio > 0 else 0
72             publish_job.update_state(
73                 state="PROGRESS",
74                 meta={'hostname': request.hostname,
75                       'start_date': start_date,
76                       'elapsed_time': elapsed_time,
77                       'eta_time': eta_time,
78                       'media_size': media_size,
79                       'publish_size': block_pos,
80                       'percent': int(100 * ratio)})
81         block_pos += len(block)
82         if not block:
83             break # End of input media reached
84         publish_file.write(block)
85     media_file.close()
86     publish_file.close() # FIXME maybe a finally block for that
87
88     # Output media file sanity check
89     publish_size = os.stat(publish_path).st_size
90     if publish_size != media_size:
91         raise IOError(
92             "Output media size does not match input (%s vs %s)" %
93             (media_size, publish_size))
94
95     # Here all seem okay
96     elapsed_time = time.time() - start_time
97     print('%s Publish job successful' % (request.id))
98     print('%s Callback : Media published as %s' % (request.id, publish_uri))
99     data_json = object2json(
100         {'job_id': request.id, 'publish_uri': publish_uri, 'status': 'SUCCESS'}, False)
101     result = callback.post(data_json)
102     print('%s Code %s %s : %s' % (request.id, result.status_code, result.reason, result._content))
103     return {'hostname': request.hostname,

```

```

104         'start_date': start_date,
105         'elapsed_time': elapsed_time,
106         'eta_time': 0,
107         'media_size': media_size,
108         'publish_size': publish_size,
109         'percent': 100}
110
111     except Exception as error:
112
113         # Here something went wrong
114         if publish_path:
115             os.remove(publish_path)
116         print(' %s Publish job failed ' % (request.id))
117         print(' %s Callback : Something went wrong' % (request.id))
118         data_json = object2json({'job_id': request.id, 'status': str(error)}, False)
119         result = callback.post(data_json)
120         print(' %s Code %s %s : %s' % (request.id, result.status_code, result.reason, result._content))
121         raise error

```

6.8.7 lib/PublisherConfig.py

```

1 import logging
2 from Utilities import json2object, jsonfile2object, object2json
3
4
5 class PublisherConfig(object):
6
7     def __init__(self, verbose, public_address, mongo_connection, rabbit_connection, rabbit_queues,
8                  api_nat_socket, storage_ip, storage_fstype, storage_mountpoint, storage_options,
9                  storage_path):
10        self.verbose = verbose
11        self.public_address = public_address
12        self.mongo_connection = mongo_connection
13        self.rabbit_connection = rabbit_connection
14        self.rabbit_queues = rabbit_queues
15        self.api_nat_socket = api_nat_socket
16        self.storage_ip = storage_ip
17        self.storage_fstype = storage_fstype
18        self.storage_mountpoint = storage_mountpoint
19        self.storage_options = storage_options
20        self.storage_path = storage_path
21
22    @property
23    def log_level(self):
24        return logging.DEBUG if self.verbose else logging.INFO
25
26    @property
27    def publish_uri(self):
28        return 'http://' + self.public_address
29
30    @property
31    def publish_path(self):
32        return '/var/www'
33
34    @property
35    def storage_uri(self):
36        if self.storage_fstype and self.storage_ip and self.storage_mountpoint:
37            return self.storage_fstype + '://' + self.storage_ip + '/' + self.storage_mountpoint
38        return None
39
40    @staticmethod
41    def read(filename):
42        config = PublisherConfig(None, None, None, None, None, None, None, None, None, None, None)
43        jsonfile2object(filename, config)
44        return config
45
46    @staticmethod
47    def load(json):
48        config = PublisherConfig(None, None, None, None, None, None, None, None, None, None, None)
49        json2object(json, config)
50        return config

```

```

51
52 PUBLISHER_CONFIG_TEST = PublisherConfig(True, 'amazon.blabla.com',
53                                         'mongodb://guest:Mongo@10.1.1.3:27017',
54                                         'amqp://guest:Alice@10.1.1.3//', 'publisher_private',
55                                         '129.194.185.47:5000', '10.1.1.2', 'glusterfs',
56                                         'medias_volume', '', '/mnt/storage')
57
58 # Main -----
59
60 if __name__ == '__main__':
61
62     print object2json(PUBLISHER_CONFIG_TEST, True)
63     assert PUBLISHER_CONFIG_TEST.storage_uri == 'glusterfs://10.1.1.2/medias_volume'
64     print str(PublisherConfig.load(object2json(PUBLISHER_CONFIG_TEST, False)))

```

6.8.8 lib/PublishJob.py

```

1 import uuid
2 from celery.result import AsyncResult
3 from Media import MEDIA_TEST
4 from User import USER_TEST
5 from Utilities import json2object, object2json, valid_uuid
6
7
8 class PublishJob(object):
9
10    def __init__(self, _id, user_id, media_id, publish_uri, statistic={}, revoked=False):
11        if not _id:
12            _id = str(uuid.uuid4())
13        self._id = _id
14        self.user_id = user_id
15        self.media_id = media_id
16        self.publish_uri = publish_uri
17        self.statistic = statistic
18        self.revoked = revoked
19
20
21    def is_valid(self, raise_exception):
22        if not valid_uuid(self._id, False):
23            if raise_exception:
24                raise TypeError(self.__class__.__name__ + ' : _id is not a valid uuid string')
25            return False
26        if hasattr(self, 'user_id') and not valid_uuid(self.user_id, False):
27            if raise_exception:
28                raise TypeError(self.__class__.__name__ + ' : user_id is not a valid uuid string')
29            return False
30        # FIXME check user if loaded
31        if hasattr(self, 'media_id') and not valid_uuid(self.media_id, False):
32            if raise_exception:
33                raise TypeError(self.__class__.__name__ + ' : media_id is not a valid uuid string')
34            return False
35        # FIXME check media if loaded
36        # FIXME check publish_uri
37        # FIXME check statistic
38        # FIXME check revoked
39        return True
40
41    def add_statistic(self, key, value, overwrite):
42        if overwrite or not key in self.statistic:
43            self.statistic[key] = value
44
45    def get_statistic(self, key):
46        return self.statistic[key] if key in self.statistic else None
47
48    def append_async_result(self):
49        async_result = AsyncResult(self._id)
50        if async_result:
51            self.status = async_result.status
52            try:
53                self.statistic.update(async_result.result)
54            except:
55                self.statistic['error'] = str(async_result.result)

```

```

56     else:
57         self.status = 'UNDEF'
58
59     def load_fields(self, user, media):
60         self.user = user
61         self.media = media
62         delattr(self, 'user_id')
63         delattr(self, 'media_id')
64
65     @staticmethod
66     def load(json):
67         job = PublishJob(None, None, None, None)
68         json2object(json, job)
69         return job
70
71 PUBLISH_JOB_TEST = PublishJob(None, USER_TEST._id, MEDIA_TEST._id, 'http://amazon.com/salut.mpg')
72
73 # -----
74
75 if __name__ == '__main__':
76     print object2json(PUBLISH_JOB_TEST, True)
77     PUBLISH_JOB_TEST.is_valid(True)
78     print str(PublishJob.load(object2json(PUBLISH_JOB_TEST, False)))

```

6.8.9 lib/Storage.py

```

1  import os
2  import time
3  from FFmpeg import get_media_duration
4  from OrchestraConfig import ORCHESTRA_CONFIG_TEST
5
6
7  class Storage(object):
8
9      @staticmethod
10     def is_mounted(path):
11         return os.path.ismount(path)
12
13     @staticmethod
14     def create_file_directory(file_path):
15         if not os.path.exists(os.path.dirname(file_path)):
16             os.makedirs(os.path.dirname(file_path))
17
18     @staticmethod
19     def media_path(config, media, generate):
20         uri = config.storage_uri
21         if not uri:
22             return None
23         if generate:
24             return config.storage_path + '/medias/' + media.user_id + '/' + media._id
25         if not media.uri or not media.uri.startswith(uri):
26             return None
27         return config.storage_path + media.uri.replace(uri, '', 1)
28
29     @staticmethod
30     def media_uri(config, media, generate):
31         if not config.storage_uri:
32             return None
33         if generate:
34             return config.storage_uri + '/medias/' + media.user_id + '/' + media._id
35         return media.uri
36
37     @staticmethod
38     def publish_point(config, media):
39         common = '/medias/' + media.user_id + '/' + media._id + '/' + media.virtual_filename
40         return (config.publish_path + common, config.publish_uri + common)
41
42     @staticmethod
43     def add_media(config, media):
44         if not media.status in ('PENDING',):
45             media_src_path = Storage.media_path(config, media, False) # get actual media path

```

```

46     if media_src_path:
47         media_dst_path = Storage.media_path(config, media, True) # generate media storage path
48         if media_dst_path != media_src_path:
49             # Generate media storage uri and move it to media storage path + set permissions
50             media.uri = Storage.media_uri(config, media, True)
51             Storage.create_file_directory(media_dst_path)
52             the_error = None
53             for i in range(1,5):
54                 try:
55                     os.rename(media_src_path, media_dst_path)
56                     # FIXME chown chmod
57                     the_error = None
58                     break
59                 except OSError as error:
60                     the_error = error
61                     time.sleep(1)
62             if the_error:
63                 raise IndexError('An error occurred : ' + the_error + '.')
64             return (os.stat(media_dst_path).st_size, get_media_duration(media_dst_path))
65         else:
66             raise NotImplementedError('FIXME Add of external uri not implemented.')
67     return (0, None)

68 @staticmethod
69 def delete_media(config, media):
70     media_path = Storage.media_path(config, media, False)
71     if media_path:
72         try:
73             os.remove(media_path)
74         except:
75             pass
76     else:
77         raise NotImplementedError('FIXME Delete of external uri not implemented.')
78
# Main -----
79
80 if __name__ == '__main__':
81
82     #print ORCHESTRA_CONFIG_TEST.storage_mounted
83     assert ORCHESTRA_CONFIG_TEST.storage_uri == 'glusterfs://10.1.1.2/medias_volume'
84     #assert ORCHESTRA_CONFIG_TEST.media_path('glusterfs://10.1.1.2/medias_volume/medias/test.mpg')
85     #assert not ORCHESTRA_CONFIG_TEST.media_path('http://10.1.1.2/medias/test.mpg')
86
87

```

6.8.10 lib/Transform.py

```

1 import fcntl
2 import os
3 import re
4 import select
5 import shlex
6 import subprocess
7 import time
8 from celery import current_task
9 from celery.decorators import task
10 from Callback import Callback
11 from FFmpeg import get_media_duration
12 from Media import Media
13 from Storage import Storage
14 from TransformConfig import TransformConfig
15 from TransformProfile import TransformProfile
16 from User import User
17 from Utilities import object2json, datetime_now, duration2secs
18
19
20 @task(name='Transform.transform_job')
21 def transform_job(user_json, media_in_json, media_out_json, profile_json, callback_json):
22
23     try:
24         # Avoid 'referenced before assignment'
25         media_out = None
26         encoder_out = ''

```

```

27     request = current_task.request
28
29     # Let's the task begin !
30     start_date = datetime.now()
31     start_time = time.time()
32     print('%s Transform job started' % (request.id))
33
34     # Load and check task parameters
35     user = User.load(user_json)
36     media_in = Media.load(media_in_json)
37     media_out = Media.load(media_out_json)
38     profile = TransformProfile.load(profile_json)
39     callback = Callback.load(callback_json)
40     user.is_valid(True)
41     media_in.is_valid(True)
42     media_out.is_valid(True)
43     profile.is_valid(True)
44     callback.is_valid(True)
45
46     # Read current configuration to translate files uri to local paths
47     config = TransformConfig.read('config.json')
48     print object2json(config, True)
49
50     # Update callback socket according to configuration
51     if config.api_nat_socket and len(config.api_nat_socket) > 0:
52         callback.replace_netloc(config.api_nat_socket)
53
54     # Verify that media file can be accessed and create output path
55     media_in_path = Storage.media_path(config, media_in, False)
56     if not media_in_path:
57         raise NotImplementedError('Input media will not be readed from shared storage : %s' %
58                                     media_in.uri)
59     media_out_path = Storage.media_path(config, media_out, True)
60     if not media_out_path:
61         raise NotImplementedError('Output media will not be written to shared storage : %s' %
62                                     media_out.uri)
63     Storage.create_file_directory(media_out_path)
64
65     # Get input media duration to be able to estimate ETA
66     media_in_size = os.stat(media_in_path).st_size
67     media_in_duration = get_media_duration(media_in_path)
68
69     # NOT A REAL TRANSFORM : FILE COPY -----
70     if profile.encoder_string == 'copy':
71         block_size = 1024 * 1024
72         media_in_file = open(media_in_path, "rb")
73         media_out_file = open(media_out_path, "wb")
74
75         # Block-based media copy loop
76         block_pos = 0
77         while True:
78             block = media_in_file.read(block_size)
79             ratio = float(block_pos) / media_in_size
80             elapsed_time = time.time() - start_time
81             eta_time = int(elapsed_time * (1 - ratio) / ratio) if ratio > 0 else 0
82             transform_job.update_state(
83                 state="PROGRESS",
84                 meta={'hostname': request.hostname,
85                       'start_date': start_date,
86                       'elapsed_time': elapsed_time,
87                       'eta_time': eta_time,
88                       'media_in_size': media_in_size,
89                       'media_in_duration': media_in_duration,
90                       'media_out_size': block_pos,
91                       'percent': int(100 * ratio)})
92             block_pos += block_size
93             if block:
94                 media_out_file.write(block)
95             else: # End of input media reached
96                 break
97             media_in_file.close()
98             media_out_file.close() # FIXME maybe a finally block for that
99

```

```

100     # Output media file sanity check
101    media_out_size = os.stat(media_out_path).st_size
102    if media_out_size != media_in_size:
103        raise IOError("Output media size does not match input (%s vs %s)" %
104                      (media_in_size, media_out_size))
105
106    # A REAL TRANSFORM : TRANSCODE WITH FFmpeg -----
107
108    else:
109        # Create FFmpeg subprocess
110        cmd = 'ffmpeg -y -i "%s" %s "%s"' % (
111            media_in_path, profile.encoder_string, media_out_path)
112        ffmpeg = subprocess.Popen(shlex.split(cmd), stderr=subprocess.PIPE, close_fds=True)
113
114        # http://stackoverflow.com/questions/1388753/how-to-get-output-from-subprocess-popen
115        fcntl.fcntl(ffmpeg.stderr.fileno(), fcntl.F_SETFL,
116                    fcntl.fcntl(ffmpeg.stderr.fileno(), fcntl.F_GETFL) | os.O_NONBLOCK,)
117
118        # frame= 2071 fps= 0 q=-1.0 size= 34623kB time=00:01:25.89 bitrate=3302.3kbits/s
119        regex = re.compile("frame=\s*(?P<frame>\d+) " +
120                           "\s+fps=\s*(?P<fps>\d+)" +
121                           "\s+q=\s*(?P<q>\S+)" +
122                           "\s+\S+size=\s*(?P<size>\S+)" +
123                           "\s+time=\s*(?P<time>\S+)" +
124                           "\s+bitrate=\s*(?P<bitrate>\S+)")
125
126    while True:
127        readable = select.select([ffmpeg.stderr.fileno(), [], []])[0]
128        if readable:
129            chunk = ffmpeg.stderr.read()
130            if chunk == '':
131                break # End of encoding reached
132            encoder_out += chunk
133            match = regex.match(chunk)
134            if match:
135                stats = match.groupdict()
136                media_out_duration = stats['time']
137                ratio = durationsecs(media_out_duration) / durationsecs(media_in_duration)
138                elapsed_time = time.time() - start_time
139                eta_time = int(elapsed_time * (1 - ratio) / ratio) if ratio > 0 else 0
140                transform_job.update_state(
141                    state="PROGRESS",
142                    meta={'hostname': request.hostname,
143                          'start_date': start_date,
144                          'elapsed_time': elapsed_time,
145                          'eta_time': eta_time,
146                          'media_in_size': media_in_size,
147                          'media_in_duration': media_in_duration,
148                          'media_out_size': os.stat(media_out_path).st_size,
149                          'media_out_duration': media_out_duration,
150                          'percent': int(100 * ratio),
151                          'encoding_frame': stats['frame'],
152                          'encoding_fps': stats['fps'],
153                          'encoding_bitrate': stats['bitrate'],
154                          'encoding_quality': stats['q']})
155                time.sleep(1) # FIXME put this number as parameter of the Transform charm (?)
156
157    # FFmpeg output sanity check (yes, this is parsing)
158    # video:78315kB audio:4876kB global headers:0kB muxing overhead 0.214781%
159    match = re.search("\s+muxing overhead\s*(?P<overhead>\S+)", encoder_out)
160    if not match:
161        raise OSError('Unable to parse FFmpeg output, encoding probably failed')
162
163    # Output media file sanity check
164    media_out_duration = get_media_duration(media_out_path)
165    if durationsecs(media_out_duration) / durationsecs(media_in_duration) > 1.5 or < 0.8:
166        salut
167
168    # Here all seem okay -----
169    media_out_size = os.stat(media_out_path).st_size
170    elapsed_time = time.time() - start_time
171    print('%s Transform job successful' % (request.id))
172    print('%s Callback: Ask to update output media %s' % (request.id, media_out.virtual_filename))
173    data_json = object2json({'job_id': request.id, 'status': 'SUCCESS'}, False)

```

```

173     result = callback.post(data_json)
174     print(
175         '%s Code %s %s : %s' % (request.id, result.status_code, result.reason, result._content))
176     return {'hostname': request.hostname,
177             'start_date': start_date,
178             'elapsed_time': elapsed_time,
179             'eta_time': 0,
180             'media_in_size': media_in_size,
181             'media_in_duration': media_in_duration,
182             'media_out_size': media_out_size,
183             'media_out_duration': get_media_duration(media_out_path),
184             'percent': 100}
185
186 except Exception as error:
187
188     # Here something went wrong
189     if media_out:
190         name = media_out.virtual_filename
191         try:
192             Storage.delete_media(config, media_out)
193         except:
194             pass # FIXME a good idea ?
195     else:
196         name = 'None'
197     print('%s Transform job failed ' % (request.id))
198     print('%s Callback : Ask to delete output media %s' % (request.id, name))
199     data_json = object2json({
200         'job_id': request.id,
201         'status': 'ERROR\n%s\n\nOUTPUT\n%s' % (str(error), encoder_out,), False)
202     result = callback.post(data_json)
203     print('%s Code %s %s : %s' % (request.id, result.status_code, result.reason, result._content))
204     raise error

```

6.8.11 lib/TransformConfig.py

```

1 import logging
2 from Utilities import json2object, jsonfile2object, object2json
3
4
5 class TransformConfig(object):
6
7     def __init__(self, verbose, public_address, mongo_connection, rabbit_connection, rabbit_queues,
8                  api_nat_socket, storage_ip, storage_fstype, storage_mountpoint, storage_options,
9                  storage_path):
10        self.verbose = verbose
11        self.public_address = public_address
12        self.mongo_connection = mongo_connection
13        self.rabbit_connection = rabbit_connection
14        self.rabbit_queues = rabbit_queues
15        self.api_nat_socket = api_nat_socket
16        self.storage_ip = storage_ip
17        self.storage_fstype = storage_fstype
18        self.storage_mountpoint = storage_mountpoint
19        self.storage_options = storage_options
20        self.storage_path = storage_path
21
22    @property
23    def log_level(self):
24        return logging.DEBUG if self.verbose else logging.INFO
25
26    @property
27    def storage_uri(self):
28        if self.storage_fstype and self.storage_ip and self.storage_mountpoint:
29            return self.storage_fstype + '://' + self.storage_ip + '/' + self.storage_mountpoint
30        return None
31
32    @staticmethod
33    def read(filename):
34        config = TransformConfig(None, None, None, None, None, None, None, None, None, None, None)
35        jsonfile2object(filename, config)
36        return config

```

```

37
38     @staticmethod
39     def load(json):
40         config = TransformConfig(None, None, None)
41         json2object(json, config)
42         return config
43
44 TRANSFORM_CONFIG_TEST = TransformConfig(True, 'amazon.blabla.com',
45                                         'mongodb://guest:Mongo@10.1.1.3:27017',
46                                         'amqp://guest:Alice@10.1.1.3//', 'transform_private',
47                                         '129.194.185.47:5000', '10.1.1.2', 'glusterfs',
48                                         'medias_volume', '', '/mnt/storage')
49
50 # Main -----
51
52 if __name__ == '__main__':
53
54     print object2json(TRANSFORM_CONFIG_TEST, True)
55     assert TRANSFORM_CONFIG_TEST.storage_uri == 'glusterfs://10.1.1.2/medias_volume'
56     print TransformConfig.load(object2json(TRANSFORM_CONFIG_TEST, False))

```

6.8.12 lib/TransformJob.py

```

1 import uuid
2 from celery.result import AsyncResult
3 from Media import MEDIA_TEST
4 from TransformProfile import TRANSFORM_PROFILE_TEST
5 from User import USER_TEST
6 from Utilities import json2object, object2json, valid_uuid
7
8
9 class TransformJob(object):
10
11     def __init__(self, _id, user_id, media_in_id, media_out_id, profile_id, statistic={}, revoked=False):
12         if not _id:
13             _id = str(uuid.uuid4())
14         self._id = _id
15         self.user_id = user_id
16         self.media_in_id = media_in_id
17         self.media_out_id = media_out_id
18         self.profile_id = profile_id
19         self.statistic = statistic
20         self.revoked = revoked
21
22
23
24     def is_valid(self, raise_exception):
25         if not valid_uuid(self._id, False):
26             if raise_exception:
27                 raise TypeError(self.__class__.__name__ + ' : _id is not a valid uuid string')
28             return False
29         if hasattr(self, 'user_id') and not valid_uuid(self.user_id, False):
30             if raise_exception:
31                 raise TypeError(self.__class__.__name__ + ' : user_id is not a valid uuid string')
32             return False
33         # FIXME check user if loaded
34         if hasattr(self, 'media_in_id') and not valid_uuid(self.media_in_id, False):
35             if raise_exception:
36                 raise TypeError(self.__class__.__name__ + ' : media_in_id is not a valid uuid string')
37             return False
38         # FIXME check media_in if loaded
39         if hasattr(self, 'media_out_id') and not valid_uuid(self.media_out_id, False):
40             if raise_exception:
41                 raise TypeError(self.__class__.__name__ + ' : media_out_id is not a valid uuid string')
42             return False
43         # FIXME check media_out if loaded
44         if hasattr(self, 'profile_id') and not valid_uuid(self.profile_id, False):
45             if raise_exception:
46                 raise TypeError(self.__class__.__name__ + ' : profile_id is not a valid uuid string')
47             return False
48         # FIXME check profile if loaded
49         # FIXME check statistic

```

```

50         # FIXME check revoked
51     return True
52
53     def add_statistic(self, key, value, overwrite):
54         if overwrite or not key in self.statistic:
55             self.statistic[key] = value
56
57     def get_statistic(self, key):
58         return self.statistic[key] if key in self.statistic else None
59
60     def append_async_result(self):
61         async_result = AsyncResult(self._id)
62         if async_result:
63             self.status = async_result.status
64             try:
65                 self.statistic.update(async_result.result)
66             except:
67                 self.statistic['error'] = str(async_result.result)
68         else:
69             self.status = 'UNDEF'
70
71     def load_fields(self, user, media_in, media_out, profile):
72         self.user = user
73         self.media_in = media_in
74         self.media_out = media_out
75         self.profile = profile
76         delattr(self, 'user_id')
77         delattr(self, 'media_in_id')
78         delattr(self, 'media_out_id')
79         delattr(self, 'profile_id')
80
81     @staticmethod
82     def load(json):
83         job = TransformJob(None, None, None, None, None)
84         json2object(json, job)
85         return job
86
87 TRANSFORM_JOB_TEST = TransformJob(None, USER_TEST._id, MEDIA_TEST._id, MEDIA_TEST._id,
88                                     TRANSFORM_PROFILE_TEST._id)
89
90 # -----
91
92 if __name__ == '__main__':
93     print object2json(TRANSFORM_JOB_TEST, True)
94     TRANSFORM_JOB_TEST.is_valid(True)
95     print str(TransformJob.load(object2json(TRANSFORM_JOB_TEST, False)))

```

6.8.13 lib/TransformProfile.py

```

1  import uuid
2  from Utilities import json2object, object2json, valid_uuid
3
4
5  class TransformProfile(object):
6
7      def __init__(self, _id, title, description, encoder_string):
8          if not _id:
9              _id = str(uuid.uuid4())
10         self._id = _id
11         self.title = title
12         self.description = description
13         self.encoder_string = encoder_string
14
15     # FIXME test other fields
16     def is_valid(self, raise_exception):
17         if not valid_uuid(self._id, False):
18             if raise_exception:
19                 raise TypeError(self.__class__.__name__ + ' : _id is not a valid uuid string')
20             return False
21         return True
22

```

```

23     @staticmethod
24     def load(json):
25         profile = TransformProfile(None, None, None, None)
26         json2object(json, profile)
27         return profile
28
29 TRANSFORM_PROFILE_TEST = TransformProfile(None, 'HD 1080p', 'MP4 H.264 1080p, audio copy',
30                                         'ffmpeg -c:a copy ...')
31
32 # -----
33
34 if __name__ == '__main__':
35     print object2json(TRANSFORM_PROFILE_TEST, True)
36     TRANSFORM_PROFILE_TEST.is_valid(True)
37     print str(TransformProfile.load(object2json(TRANSFORM_PROFILE_TEST, False)))

```

6.8.14 lib/User.py

```

1 import uuid
2 from Utilities import json2object, object2json, valid_mail, valid_secret, valid_uuid
3
4
5 class User(object):
6
7     def __init__(self, _id, first_name, last_name, mail, secret, admin_platform=False):
8         if not _id:
9             _id = str(uuid.uuid4())
10        self._id = _id
11        self.first_name = first_name
12        self.last_name = last_name
13        self.mail = mail
14        self.secret = secret
15        self.admin_platform = True if str(admin_platform).lower() == 'true' else False
16
17    @property
18    def name(self):
19        if self.first_name and self.last_name:
20            return self.first_name + ' ' + self.last_name
21        return 'anonymous'
22
23    # FIXME test other fields
24    def is_valid(self, raise_exception):
25        if not valid_uuid(self._id, False):
26            if raise_exception:
27                raise TypeError(self.__class__.__name__ + ' : _id is not a valid uuid string')
28            return False
29        if not valid_mail(self.mail):
30            if raise_exception:
31                raise TypeError(self.__class__.__name__ + ' : mail is not a valid email address')
32            return False
33        if hasattr(self, 'secret') and not valid_secret(self.secret):
34            if raise_exception:
35                raise TypeError(self.__class__.__name__ +
36                                ' : secret is not safe (8+ characters, upper/lower + numbers eg. StrongP6s)')
37            return False
38        return True
39
40    @staticmethod
41    def load(json):
42        user = User(None, None, None, None, None)
43        json2object(json, user)
44        return user
45
46 USER_TEST = User(None, 'David', 'Fischer', 'david.fischer.ch@gmail.com', 'secret', True)
47
48 # -----
49
50 if __name__ == '__main__':
51     print object2json(USER_TEST, True)
52     USER_TEST.is_valid(True)
53     print str(User.load(object2json(USER_TEST, False)))

```

6.8.15 lib/Utilities.py

```

1 import datetime
2 import inspect
3 from ipaddr import IPAddress
4 import json
5 import logging
6 import logging.handlers
7 import re
8 import uuid
9 from bson.json_util import dumps, loads
10
11
12 class ForbiddenError(Exception):
13     pass
14
15
16 def datetime_now():
17     return datetime.datetime.utcnow().strftime("%Y-%m-%d %H:%M:%S")
18
19
20 ## http://stackoverflow.com/questions/6255387/mongodb-object-serialized-as-json
21 class SmartJSONEncoderV1(json.JSONEncoder):
22     def default(self, obj):
23         if hasattr(obj, '__dict__'):
24             return obj.__dict__
25         return super(SmartJSONEncoderV1, self).default(obj)
26
27
28 class SmartJSONEncoderV2(json.JSONEncoder):
29     def default(self, obj):
30         attributes = {}
31         for a in inspect.getmembers(obj):
32             if inspect.isroutine(a[1]) or inspect.isbuiltin(a[1]) or a[0].startswith('__'):
33                 continue
34             attributes[a[0]] = a[1]
35         return attributes
36
37
38 def json2object(json, something):
39     something.__dict__ = loads(json)
40
41
42 def jsonfile2object(filename, something):
43     something.__dict__ = json.load(open(filename))
44
45
46 def object2json(something, include_properties):
47     if not include_properties:
48         return dumps(something, cls=SmartJSONEncoderV1)
49     else:
50         return dumps(something, cls=SmartJSONEncoderV2)
51
52
53 def duration2secs(duration):
54     hours, minutes, seconds = duration.split(':')
55     return int(hours) * 3600 + int(minutes) * 60 + float(seconds)
56
57
58 def valid_mail(mail):
59     try:
60         return re.match(r'^[\w\.-]+@[^\w\.-]+\.[^\w\.-]+\w$', mail)
61     except:
62         return False
63
64
65 def valid_filename(filename):
66     try:
67         return re.match(r'^[\w\.-]+\w\.[^\w\.-]+\w$', filename)
68     except:
69         return False
70
71

```

```

72 def valid_secret(secret):
73     try:
74         return re.match(r' [A-Za-z0-9@#$%^&+=] {8,}', secret)
75     except:
76         return False
77
78
79 def valid_ip(ip):
80     try:
81         IPAddress(ip)
82         return True
83     except:
84         return False
85
86
87 def valid_port(port):
88     if not port:
89         return False
90     return True # FIXME TODO
91
92
93 def valid_uuid(id, none_allowed):
94     if not id and none_allowed:
95         return True
96     try:
97         uuid.UUID('' + str(id) + '')
98     except ValueError:
99         return False
100    return True
101
102
103 def setup_logging(filename, level, format='%(asctime)s %(levelname)-8s - %(message)s',
104                  datefmt='%d/%m/%Y %H:%M:%S'):
105     logging.basicConfig(filename=filename, level=level, format=format, datefmt=datefmt)
106     console = logging.StreamHandler()
107     console.setLevel(level)
108     console.setFormatter(logging.Formatter(format))
109     logging.getLogger('').addHandler(console)
110
111 UUID_ZERO = str(uuid.UUID('{00000000-0000-0000-0000-000000000000}'))
```

6.9 OSCIED - Orchestra Source Code

Note: config.json, copyright and revision are skipped as they are not so relevant here.

6.9.1 config.yaml

```

1 options:
2     verbose:
3         type: string
4         default: "false"
5         description: Set verbose logging. Can be set to true or false
6     root_secret:
7         type: string
8         default: "toto"
9         description: Secret key used by API clients to manage users. Please change it in production.
10    nodes_secret:
11        type: string
12        default: "abcd"
13        description: Secret key used by workers/nodes to callback API when they finish their job. Please change it in production.
14    repositories_user:
15        type: string
16        default: "oscied"
17        description: OSCIED charms repositories client username.
18    repositories_pass:
19        type: string
```

```

20     default: ""
21     description: OSCIED charms repositories client password.
22   webui_repository:
23     type: string
24     default: "https://claire-et-david.dyndns.org/prog/OSCIED/components/webui/charm"
25     description: OSCIED Web UI charm will be checked out locally under ~/charms/(release)/oscied-webui.
26   transform_repository:
27     type: string
28     default: "https://claire-et-david.dyndns.org/prog/OSCIED/components/transform/charm"
29     description: OSCIED Transform charm will be checked out locally under ~/charms/(release)/oscied-transform.
30   publisher_repository:
31     type: string
32     default: "https://claire-et-david.dyndns.org/prog/OSCIED/components/publisher/charm"
33     description: OSCIED Publisher charm will be checked out locally under ~/charms/(release)/oscied-publisher.
34   mongo_admin_password:
35     type: string
36     default: "Mongo_admin_1234"
37     description: Database administrator password.
38   mongo_nodes_password:
39     type: string
40     default: "Mongo_user_1234"
41     description: Database nodes password.
42   rabbit_password:
43     type: string
44     default: "Alice_in_wonderland"
45     description: Messaging queue user's password. This secret is then forwarded by the coordinator to managed
46   storage_ip:
47     type: string
48     default: ""
49     description: Shared storage hostname / IP address (see interface mount of nfs charm). The 'storage' option
50   storage_nat_ip:
51     type: string
52     default: ""
53     description: Shared storage reacheable (public) address e.g. when storage is behind a NAT/FW.
54   storage_fstype:
55     type: string
56     default: ""
57     description: Shared storage filesystem type (e.g. nfs)
58   storage_mountpoint:
59     type: string
60     default: ""
61     description: Shared storage mountpoint (e.g. for nfs - /srv/data)
62   storage_options:
63     type: string
64     default: ""
65     description: Shared storage options (e.g. for nfs - rw,sync,no_subtree_check)

```

6.9.2 metadata.yaml

```

1 ensemble: oscied
2 name: oscied-orchestra
3 summary: "EBU's OSCIED Orchestra"
4 maintainer: OSCIED Main Developper <david.fischer.ch@gmail.com>
5 description: |
6   Installs the EBU's OSCIED orchestrator.
7 provides:
8   api:
9     interface: orchestra
10    transform:
11      interface: subordinate
12    publisher:
13      interface: subordinate
14 requires:
15   storage:
16     interface: mount

```

6.9.3 hooks_lib/common.sh.lu-dep

```

1 set -o nounset # will exit if an uninitialized variable is used
2
3 # Constants =====
4
5 ECHO='juju-log' # Used by logicielsUbuntuUtils
6 RELEASE=$(lsb_release -cs)
7 JUJU_CHARMS SVN='https://claire-et-david.dyndns.org/prog/OSCIED/components'
8
9 # Charms paths
10 BASE_PATH=$(pwd)
11
12 # Charms files
13 CONFIG_FILE="$BASE_PATH/config.json"
14 CELERY_TEMPFILE="$BASE_PATH/templates/celeryconfig.py.template"
15 CELERY_CONFIG_FILE="$BASE_PATH/celeryconfig.py"
16 JUJU_TEMPFILE="$BASE_PATH/templates/environments.yaml.template"
17 REPOS_CERTIF_FILE="$BASE_PATH/templates/99a9179b9106d19d4e1cca7a720b079a"
18
19 # Shared storage paths
20 STORAGE_ROOT_PATH='/mnt/storage'
21 STORAGE_MEDIAS_PATH="$STORAGE_ROOT_PATH/medias"
22 STORAGE_UPLOADS_PATH="$STORAGE_ROOT_PATH/uploads"
23
24 # JuJu, ssh & subversion paths
25 HOME_PATH='/root'
26 CHARMS_PATH="$HOME_PATH/charms"
27 PUBLISHER_PATH="$CHARMS_PATH/$RELEASE/oscied-publisher"
28 TRANSFORM_PATH="$CHARMS_PATH/$RELEASE/oscied-transform"
29 WEBUI_PATH="$CHARMS_PATH/$RELEASE/oscied-webui"
30 JUJU_STORAGE_PATH="$HOME_PATH/.juju/storage"
31 SVN_CERTIFS_PATH="$HOME_PATH/.subversion/auth/svn.ssl.server"
32
33 # JuJu, ssh & subversion files
34 CERTIF_FILE="$HOME_PATH/.ssh/id_rsa"
35 JUJU_ENVS_FILE="$HOME_PATH/.juju/environments.yaml"
36 SVN_SERVERS_FILE='/etc/subversion/servers'
37
38 # MongoDB configuration files
39 MONGO_CONFIG_FILE='/etc/mongodb.conf'
40
41 # Configuration =====
42
43 if [ "$(config-get verbose)" = 'true' ] ; then
44     VERBOSE=0      # true
45     set -o xtrace # for verbose logging to juju debug-log
46 else
47     VERBOSE=1 # false
48 fi
49
50 OWN_IP=$(unit-get private-address)
51 API_URL="http://$OWN_IP:5000"
52 API_LOCAL_URL="http://127.0.0.1:5000"
53 ROOT_SECRET=$(config-get root_secret)
54 NODES_SECRET=$(config-get nodes_secret)
55 REPOS_USER=$(config-get repositories_user)
56 REPOS_PASS=$(config-get repositories_pass)
57
58 WEBUI_REPO=$(config-get webui_repository)
59 TRANSFORM_REPO=$(config-get transform_repository)
60 PUBLISHER_REPO=$(config-get publisher_repository)
61
62 MONGO_ADMIN_PASSWORD=$(config-get mongo_admin_password)
63 MONGO_NODES_PASSWORD=$(config-get mongo_nodes_password)
64 RABBIT_PASSWORD=$(config-get rabbit_password)
65
66 STORAGE_IP=$(config-get storage_ip)
67 STORAGE_NAT_IP=$(config-get storage_nat_ip)
68 STORAGE_FSTYPE=$(config-get storage_fstype)
69 STORAGE_MOUNTPOINT=$(config-get storage_mountpoint)
70 STORAGE_OPTIONS=$(config-get storage_options)
71

```

```

72 mongo_admin_connection()
73 {
74     if [ $# -ne 1 ]; then
75         xecho "Usage: $(basename $0).mongo_admin_connection IP"
76     fi
77     echo "mongodb://admin:$MONGO_ADMIN_PASSWORD@$1:27017/orchestra"
78 }
79
80 mongo_nodes_connection()
81 {
82     if [ $# -ne 1 ]; then
83         xecho "Usage: $(basename $0).mongo_nodes_connection IP"
84     fi
85     echo "mongodb://nodes:$MONGO_NODES_PASSWORD@$1:27017/celery"
86 }
87
88 rabbit_connection()
89 {
90     if [ $# -ne 1 ]; then
91         xecho "Usage: $(basename $0).rabbit_connection IP"
92     fi
93     echo "amqp://nodes:$RABBIT_PASSWORD@$1:5672/celery"
94 }
95
96 # Utilities =====
97
98 storage_config_is_enabled()
99 {
100    [ "$STORAGE_IP" -a "$STORAGE_FSTYPE" -a "$STORAGE_MOUNTPOINT" ]
101 }
102
103 storage_is_mounted()
104 {
105     mount | grep -q "$STORAGE_ROOT_PATH"
106 }
107
108 storage_remount()
109 {
110     # Overrides storage parameters with charm configuration
111     if storage_config_is_enabled; then # if storage options are set
112         ip=$STORAGE_IP
113         nat_ip=$STORAGE_NAT_IP
114         fstype=$STORAGE_FSTYPE
115         mountpoint=$STORAGE_MOUNTPOINT
116         options=$STORAGE_OPTIONS
117     # Or uses storage parameters from charm storage relation
118     elif [ $# -eq 4 ]; then # if function parameters are set
119         ip=$1
120         nat_ip=''
121         fstype=$2
122         mountpoint=$3
123         options=$4
124     elif [ $# -eq 0 ]; then
125         return
126     else
127         xecho "Usage: $(basename $0).storage_remount ip fstype mountpoint options"
128     fi
129
130     if [ "$nat_ip" ]; then
131         pecho "Update hosts file to map storage internal address $ip to $nat_ip"
132         if grep -q "$ip" /etc/hosts; then
133             sed -i "s<$nat_ip .*<$nat_ip $ip<" /etc/hosts
134         else
135             echo "$nat_ip $ip" >> /etc/hosts
136         fi
137     else
138         nat_ip=$ip
139     fi
140
141     storage_umount
142
143     r=$STORAGE_ROOT_PATH
144     pecho "Mount shared storage [$nat_ip] $ip:$mountpoint type $fstype options '$options' -> $r"

```

```

145 if [ ! -d "$STORAGE_ROOT_PATH" ]; then
146     mkdir "$STORAGE_ROOT_PATH" || xecho "Unable to create shared storage path $STORAGE_ROOT_PATH" 1
147 fi
148
149 # FIXME try 5 times, a better way to handle failure
150 for i in $(seq 1 5)
151 do
152     if storage_is_mounted; then
153         break
154     else
155         if [ "$options" ]
156             then mount -t "$fstype" -o "$options" "$nat_ip:$mountpoint" "$STORAGE_ROOT_PATH"
157         else mount -t "$fstype" "$nat_ip:$mountpoint" "$STORAGE_ROOT_PATH"
158         fi
159     fi
160     sleep 5
161 done
162
163 if storage_is_mounted; then
164     # FIXME update /etc/fstab (?)
165     pecho 'Configure Orchestra : Register shared storage'
166     setSettingJSON_STRING "$CONFIG_FILE" 'storage_ip'           "$ip"          || xecho 'Config' 2
167     setSettingJSON_STRING "$CONFIG_FILE" 'storage_fstype'      "$fstype"      || xecho 'Config' 3
168     setSettingJSON_STRING "$CONFIG_FILE" 'storage_mountpoint' "$mountpoint" || xecho 'Config' 4
169     setSettingJSON_STRING "$CONFIG_FILE" 'storage_options'    "$options"    || xecho 'Config' 5
170 else
171     xecho 'Unable to mount shared storage' 6
172 fi
173 }
174
175 storage_umount()
176 {
177     pecho 'Configure Orchestra : Unregister shared storage'
178     setSettingJSON_STRING "$CONFIG_FILE" 'storage_ip'           '' || xecho 'Config' 1
179     setSettingJSON_STRING "$CONFIG_FILE" 'storage_fstype'      '' || xecho 'Config' 2
180     setSettingJSON_STRING "$CONFIG_FILE" 'storage_mountpoint' '' || xecho 'Config' 3
181     setSettingJSON_STRING "$CONFIG_FILE" 'storage_options'    '' || xecho 'Config' 4
182
183 if storage_is_mounted; then
184     # FIXME update /etc/fstab (?)
185     pecho 'Unmount shared storage (is actually mounted)'
186     umount "$STORAGE_ROOT_PATH" || xecho 'Unable to unmount shared storage' 5
187     recho 'Shared storage successfully unmounted'
188 else
189     recho 'Shared storage already unmounted'
190 fi
191 }
192
193 storage_hook_bypass()
194 {
195     if storage_config_is_enabled; then
196         xecho 'Shared storage is set in config, storage relation is disabled' 1
197     fi
198 }
199
200 config_rabbitmq()
201 {
202     pecho 'Configure RabbitMQ Message Broker'
203     rabbitmqctl delete_user guest
204     rabbitmqctl delete_vhost /
205     rabbitmqctl add_user nodes "$RABBIT_PASSWORD"
206     rabbitmqctl add_vhost celery
207     rabbitmqctl set_permissions -p celery nodes ".*" ".*" ".*"
208     if ! rabbitmqctl list_users | grep -q 'nodes'; then
209         xecho 'Unable to add RabbitMQ user' 4
210     fi
211     if ! rabbitmqctl list_vhosts | grep -q 'celery'; then
212         xecho 'Unable to add RabbitMQ vhost' 5
213     fi
214 }
215
216 # HOOKS : Charm Setup =====

```

```

218 hook_install()
219 {
220     techo 'Orchestrator - install'
221
222     # Fix memtest86+ : https://bugs.launchpad.net/ubuntu/+source/grub2/+bug/1069856
223     #eval $purge grub-pc grub-common
224     #eval $install grub-common grub-pc
225
226     # I decided to use the real ffmpeg, not the libav version :
227     # http://blog.pkh.me/p/13-the-ffmpeg-libav-situation.html
228     # http://doc.ubuntu-fr.org/ffmpeg
229     apt-add-repository -y ppa:jon-severinsson/ffmpeg || xecho 'Unable to add FFmpeg repository' 1
230     apt-add-repository -y ppa:juju/pkgs || xecho 'Unable to add JuJu PPA repository' 2
231     eval $update
232     eval $upgrade
233
234     pecho 'Install and configure Network Time Protocol'
235     eval $install ntp || xecho 'Unable to install ntp' 3
236     eval $service ntp restart || xecho 'Unable to restart ntp service' 4
237
238     #pecho 'Checkout OSCIED charms locally'
239     #eval $install subversion || xecho 'Unable to install packages' 5
240     #setSettingBASH "$SVN_SERVERS_FILE" $true 'store-passwords'                                'yes' || exit 6
241     #setSettingBASH "$SVN_SERVERS_FILE" $true 'store-plaintext-passwords'                      'yes' || exit 7
242     #setSettingBASH "$SVN_SERVERS_FILE" $true 'store-ssl-client-cert-pp'                      'yes' || exit 8
243     #setSettingBASH "$SVN_SERVERS_FILE" $true 'store-ssl-client-cert-pp-plaintext'          'yes' || exit 9
244     #mkdir -p "$SVN_CERTIFS_PATH"; cp -f "$REPOS_CERTIF_FILE" "$SVN_CERTIFS_PATH/"
245     #checkout "$WEBUI_REPO"      "$WEBUI_PATH"      "$REPOS_USER" "$REPOS_PASS" || exit 10
246     #checkout "$TRANSFORM_REPO" "$TRANSFORM_PATH" "$REPOS_USER" "$REPOS_PASS" || exit 11
247     #checkout "$PUBLISHER_REPO" "$PUBLISHER_PATH" "$REPOS_USER" "$REPOS_PASS" || exit 12
248
249     pecho 'Install JuJu Cloud Orchestrator, (the real) FFmpeg and x264'
250     eval $install apt-cacher-ng charm-tools juju libzookeeper-java zookeeper ffmpeg x264 || \
251         xecho 'Unable to install packages' 5
252
253     pecho 'Install python, MongoDB Scalable NoSQL DB, RabbitMQ Message Broker and Gluster Filesystem'
254     eval $install build-essential python-dev python python-pip git-core mongodb rabbitmq-server \
255         glusterfs-client nfs-common || xecho 'Unable to install packages' 6
256
257     pecho 'Install BSON, Celery Distrib. Task Queue, Flask Python Web Framework, PyMongo MongoDB API'
258     pip install --upgrade bson celery flask ipaddr pymongo requests || \
259         xecho 'Unable to install packages' 7
260
261     pecho 'Expose RESTful API, MongoDB & RabbitMQ service'
262     open-port 5000/tcp    # Orchestra RESTful API
263     open-port 27017/tcp   # MongoDB portmongod and mongos instances
264     #open-port 27018/tcp # MongoDB port when running with shardsvr setting
265     #open-port 27019/tcp # MongoDB port when running with configsvr setting
266     open-port 28017/tcp   # MongoDB port for the web status page. This is always +1000
267     open-port 5672/tcp    # RabbitMQ service
268
269     # FIXME this call is not necessary, but config-changed create an infinite loop, so WE call it
270     hook_config_changed
271 }
272
273 hook_uninstall()
274 {
275     techo 'Orchestrator - uninstall'
276
277     hook_stop
278
279     # fix rabbitmq-server package uninstall error
280     mkdir /var/log/rabbitmq 2>/dev/null
281
282     juju destroy-environment
283     eval $purge apt-cacher-ng charm-tools juju libzookeeper-java lxc zookeeper
284     eval $purge mongodb rabbitmq-server glusterfs-client nfs-common
285     pip install uninstall bson flask celery
286     eval $autoremove
287     rm -rf $HOME/.juju $HOME/.ssh/id_rsa*
288     rm -rf /etc/rabbitmq/ /var/log/rabbitmq/
289 }
290

```

```

291 hook_config_changed()
292 {
293     techo 'Orchestrator - config changed'
294
295     pecho 'Configure MongoDB Scalable NoSQL DB'
296
297     echo "db.addUser('admin', '$MONGO_ADMIN_PASSWORD', false);" > f.js
298     mongo f.js; mongo 'orchestra' f.js
299     rm -f f.js
300
301     echo "db.addUser('nodes', '$MONGO_NODES_PASSWORD', false);" > g.js
302     mongo 'celery' g.js
303     rm -f g.js
304
305     setSettingBASH "$MONGO_CONFIG_FILE" $true 'bind_ip' '0.0.0.0' || xecho 'Config' 1
306     setSettingBASH "$MONGO_CONFIG_FILE" $true 'noauth' 'false' || xecho 'Config' 2
307     setSettingBASH "$MONGO_CONFIG_FILE" $true 'auth' 'true' || xecho 'Config' 3
308
309 config_rabbitmq
310
311 hook_stop
312
313     pecho 'Configure Orchestra the Orchestrator'
314     mongo=$(mongo_admin_connection 'localhost')
315     rabbit=$(rabbit_connection 'localhost')
316     mecho "MongoDB connection is $mongo, RabbitMQ connection is $rabbit"
317     if [ ! "$mongo" -o ! "$rabbit" ]; then
318         xecho 'Unable to detect MongoDB or RabbitMQ connection' 5
319     fi
320     setSettingJSON_BOOLEAN "$CONFIG_FILE" 'verbose' "$VERBOSE" || xecho 'Config' 6
321     setSettingJSON_STRING "$CONFIG_FILE" 'api_url' "$API_URL" || xecho 'Config' 7
322     setSettingJSON_STRING "$CONFIG_FILE" 'root_secret' "$ROOT_SECRET" || xecho 'Config' 8
323     setSettingJSON_STRING "$CONFIG_FILE" 'nodes_secret' "$NODES_SECRET" || xecho 'Config' 9
324     setSettingJSON_STRING "$CONFIG_FILE" 'mongo_connection' "$mongo" || xecho 'Config' 10
325     setSettingJSON_STRING "$CONFIG_FILE" 'rabbit_connection' "$rabbit" || xecho 'Config' 11
326     setSettingJSON_STRING "$CONFIG_FILE" 'storage_path' "$STORAGE_ROOT_PATH" || xecho 'Config' 12
327     sed "s<RABBIT_CONNECTION<$rabbit<g;s<MONGO_PASSWORD<$MONGO_NODES_PASSWORD<g" \
328         < "$CELERY_TEMPFILE" > "$CELERY_CONFIG_FILE" || xecho 'Config' 13
329
330 storage_remount
331
332 hook_start
333 # FIXME infinite loop is used as config-changed hook !
334 }
335
336 # HOOKS : Charm Service =====
337
338 hook_start()
339 {
340     techo 'Orchestrator - start'
341
342     if ! storage_is_mounted; then
343         recho 'WARNING: No shared storage, Orchestra may not start'
344     fi
345
346     # do not check status after all, orchestra can do it for us !
347     service mongodb start
348     service rabbitmq-server start
349
350     # FIXME this is not a good idea, but I have some trouble with precise release (see ticket #205)
351 config_rabbitmq
352
353     if ! curl -s "$API_LOCAL_URL" > /dev/null; then
354         screen -d -mS 'Orchestra' python orchestra.py
355         sleep 5
356     if ! curl -s "$API_LOCAL_URL" > /dev/null; then
357         # FIXME mail to very important people
358         recho 'Orchestra is not ready'
359     else
360         recho 'Orchestra successfully started'
361     fi
362 fi
363 }
```

```

364
365 hook_stop()
366 {
367     techo 'Orchestrator - stop'
368
369     if curl -s "$API_LOCAL_URL" > /dev/null; then
370         screen -X -S 'Orchestra' kill || xecho 'Unable to stop Orchestra daemon' 1
371     fi
372     service rabbitmq-server stop || xecho 'Unable to stop RabbitMQ' 2
373     if ! service mongodb status | grep -q 'stop'; then
374         service mongodb stop || xecho 'Unable to stop MongoDB' 3
375     fi
376 }
377
378 # HOOKS : Requires Storage =====
379
380 hook_storage_relation_joined()
381 {
382     techo 'Orchestra - storage relation joined'
383
384     storage_hook_bypass
385 }
386
387 hook_storage_relation_changed()
388 {
389     techo 'Orchestra - storage relation changed'
390
391     storage_hook_bypass
392
393     # Get configuration from the relation
394     ip=$(relation-get private-address)
395     fstype=$(relation-get fstype)
396     mountpoint=$(relation-get mountpoint)
397     options=$(relation-get options)
398
399     mecho "Storage IP is $ip, fstype: $fstype, mountpoint: $mountpoint, options: $options"
400     if [ ! "$ip" -o ! "$fstype" -o ! "$mountpoint" ]; then
401         recho 'Waiting for complete setup'
402         exit 0
403     fi
404
405     hook_stop
406     storage_remount "$ip" "$fstype" "$mountpoint" "$options"
407     hook_start
408 }
409
410 hook_storage_relation_broken()
411 {
412     techo 'Orchestra - storage relation broken'
413
414     storage_hook_bypass
415
416     hook_stop
417     storage_umount
418 }
419
420 # HOOKS : Provides API =====
421
422 hook_api_relation_joined()
423 {
424     techo 'Orchestrator - api relation joined'
425
426     # Send Orchestra API URL
427     relation-set "api_url=$API_URL"
428 }
429
430 hook_api_relation_changed()
431 {
432     techo 'Orchestrator - api relation changed'
433
434     # Get configuration from the relation
435     webui_ip=$(relation-get private-address)
436

```

```

437     mecho "Web UI IP is $webui_ip"
438     if [ ! "$webui_ip" ]; then
439         recho 'Waiting for complete setup'
440         exit 0
441     fi
442
443     # FIXME something to do (register unit ?)
444 }
445
446 # HOOKS : Provides Publisher =====
447
448 hook_publisher_relation_joined()
449 {
450     techo 'Orchestrator - publisher relation joined'
451
452     # Send MongoDB & RabbitMQ connections
453     mongo=$(mongo_nodes_connection "$OWN_IP")
454     rabbit=$(rabbit_connection "$OWN_IP")
455     mecho "MongoDB connection is $mongo, RabbitMQ connection is $rabbit"
456     if [ ! "$mongo" -o ! "$rabbit" ]; then
457         xecho 'Unable to detect MongoDB or RabbitMQ connection'
458     fi
459     relation-set "mongo_connection=$mongo" "rabbit_connection=$rabbit"
460 }
461
462 hook_publisher_relation_changed()
463 {
464     techo 'Orchestrator - publisher relation changed'
465
466     # Get configuration from the relation
467     publisher_ip=$(relation-get private-address)
468
469     mecho "Publisher IP is $publisher_ip"
470     if [ ! "$publisher_ip" ]; then
471         recho 'Waiting for complete setup'
472         exit 0
473     fi
474
475     # FIXME something to do (register unit ?)
476 }
477
478 # HOOKS : Provides Transform Relation =====
479
480 hook_transform_relation_joined()
481 {
482     techo 'Orchestrator - transform relation joined'
483
484     # Send MongoDB & RabbitMQ connections
485     mongo=$(mongo_nodes_connection "$OWN_IP")
486     rabbit=$(rabbit_connection "$OWN_IP")
487     mecho "MongoDB connection is $mongo, RabbitMQ connection is $rabbit"
488     if [ ! "$mongo" -o ! "$rabbit" ]; then
489         xecho 'Unable to detect MongoDB or RabbitMQ connection'
490     fi
491     relation-set "mongo_connection=$mongo" "rabbit_connection=$rabbit"
492 }
493
494 hook_transform_relation_changed()
495 {
496     techo 'Orchestrator - transform relation changed'
497
498     # Get configuration from the relation
499     transform_ip=$(relation-get private-address)
500
501     mecho "Transform IP is $transform_ip"
502     if [ ! "$transform_ip" ]; then
503         recho 'Waiting for complete setup'
504         exit 0
505     fi
506
507     # FIXME something to do (register unit ?)
508 }

```

6.9.4 templates/celeryconfig.py.template

```

1 import sys
2 from kombu import Queue
3
4 sys.path.append('.')
5
6 BROKER_URL = 'RABBIT_CONNECTION'
7 CELERY_RESULT_BACKEND = 'mongodb'
8 CELERY_MONGODB_BACKEND_SETTINGS = {
9     'host': '127.0.0.1',
10    'port': 27017,
11    'user': 'nodes',
12    'password': 'MONGO_PASSWORD',
13    'database': 'celery',
14    'taskmeta_collection': 'taskmeta',
15 }
16
17 #class OrchestraRouter(object):
18 #
19 #     def route_for_task(self, task, args=None, kwargs=None):
20 #         return {'exchange': '/',
21 #                 'exchange_type': 'topic',
22 #                 'queue': 'transform_new',
23 #                 'routing_key': 'transform.new'}
24
25 #CELERY_ROUTES = (OrchestraRouter(), )
26 CELERY_QUEUES = (
27     Queue('black_hole', routing_key='blackhole'),
28     Queue('transform_private', routing_key='transform.private'),
29     Queue('transform_amazon', routing_key='transform.amazon'),
30     Queue('publisher_private', routing_key='publisher.private'),
31     Queue('publisher_amazon', routing_key='publisher.amazon'),
32 )
33 CELERY_CREATE_MISSING_QUEUES = True
34 CELERY_DEFAULT_QUEUE = 'black_hole'
35 CELERY_DEFAULT_EXCHANGE = '/'
36 CELERY_DEFAULT_EXCHANGE_TYPE = 'topic'
37
38 CELERY_IMPORTS = ('lib.Transform', 'lib.Publisher',)
39 CELERY_IGNORE_RESULT = False
40 CELERY_SEND_EVENTS = True
41 CELERY_TASK_SERIALIZER = 'json'
42 CELERY_TASK_COMPRESSION = 'zlib'
43 CELERY_TRACK_STARTED = True

```

6.9.5 orchestra.py

See Also:

The source-code is not shown here, (2500+ LoC), please see [OSCIED - Orchestra RESTful API](#) for the RESTful API methods documentation.

6.10 OSCIED - Publisher Source Code

Note: config.json, copyright and revision are skipped as they are not so relevant here.

6.10.1 config.yaml

```

1 options:
2     verbose:
3         type: string
4         default: "false"
5         description: Set verbose logging. Can be set to true or false

```

```

6   concurrency:
7     type: int
8     default: 1
9     description: Amount of tasks the worker can handle simultaneously. Default value is conservative
10    rabbit_queues:
11      type: string
12      default: "publisher_private"
13      description: Worker connect to queues to receive jobs. (e.g. "publisher_amazon" for amazon, "publisher_pr...
14    max_upload_size:
15      type: int
16      default: 4294967296
17      description: Maximum size for file uploads. Default value means 4GB
18    max_execution_time:
19      type: int
20      default: 180
21      description: Maximum time for php scripts. Default value means 3 minutes
22    max_input_time:
23      type: int
24      default: 600
25      description: Maximum time for http post. Default value means 10 minutes
26  mongo_connection:
27    type: string
28    default: ""
29    description: Orchestrator database connection. The 'connection' options override and disable transform re...
30  rabbit_connection:
31    type: string
32    default: ""
33    description: Orchestrator message broker connection
34  api_nat_socket:
35    type: string
36    default: ""
37    description: Orchestrator API reachable (public) address (ip:port) e.g. when orchestra is behind a NAT/FW
38  storage_ip:
39    type: string
40    default: ""
41    description: Shared storage hostname / IP address (see interface mount of nfs charm). The 'storage' optio...
42  storage_nat_ip:
43    type: string
44    default: ""
45    description: Shared storage reachable (public) address e.g. when storage is behind a NAT/FW.
46  storage_fstype:
47    type: string
48    default: ""
49    description: Shared storage filesystem type (e.g. nfs)
50  storage_mountpoint:
51    type: string
52    default: ""
53    description: Shared storage mountpoint (e.g. for nfs - /srv/data)
54  storage_options:
55    type: string
56    default: ""
57    description: Shared storage options (e.g. for nfs - rw,sync,no_subtree_check)

```

6.10.2 metadata.yaml

```

1 ensemble: oscied
2 name: oscied-publisher
3 summary: "EBU's OSCIED Publisher"
4 maintainer: OSCIED Main Developer <david.fischer.ch@gmail.com>
5 description: |
6   Installs the EBU's OSCIED media publisher. Contains apache & mod streaming.
7   Can be scaled to multiple instances.
8 provides:
9   streaming:
10     interface: http
11 requires:
12   publisher:
13     interface: subordinate
14   storage:
15     interface: mount

```

6.10.3 get-dependencies.sh

```

1 #!/usr/bin/env bash
2 lib='mod_h264_streaming-2.2.7'
3 tar="apache_Slib.tar.gz"
4 wget -N http://h264.code-shop.com/download/$tar

```

6.10.4 hooks_lib/common.sh.lu-dep

```

1 set -o nounset # will exit if an uninitialized variable is used
2
3 # Constants =====
4
5 ECHO='juju-log' # Used by logicielsUbuntuUtils
6
7 # Charms paths
8 BASE_PATH=$(pwd)
9
10 # Charms files
11 CONFIG_FILE="$BASE_PATH/config.json"
12 CELERY_TEMPIL_FILE="$BASE_PATH/templates/celeryconfig.py.template"
13 CELERY_CONFIG_FILE="$BASE_PATH/celeryconfig.py"
14
15 # Shared storage paths
16 STORAGE_ROOT_PATH='/mnt/storage'
17 STORAGE_MEDIAS_PATH="$STORAGE_ROOT_PATH/medias"
18 STORAGE_UPLOADS_PATH="$STORAGE_ROOT_PATH/uploads"
19
20 # Apache 2 paths
21 WWW_ROOT_PATH='/var/www'
22
23 # Apache 2 configuration files
24 HTACCESS_FILE="$WWW_ROOT_PATH/.htaccess"
25
26 # Configuration =====
27
28 if [ "$(config-get verbose)" = 'true' ] ; then
29     VERBOSE=0      # true
30     set -o xtrace # for verbose logging to juju debug-log
31 else
32     VERBOSE=1 # false
33 fi
34
35 PUBLIC_ADDRESS=$(unit-get public-address)
36 THE_CONCURRENCY=$(config-get concurrency)
37 RABBIT_QUEUES=$(config-get rabbit_queues),$PUBLIC_ADDRESS"
38
39 PROXY_IPS=$(cat proxy_ips 2>/dev/null)
40 MAX_UPLOAD_SIZE=$(config-get max_upload_size)
41 MAX_EXECUTION_TIME=$(config-get max_execution_time)
42 MAX_INPUT_TIME=$(config-get max_input_time)
43
44 MONGO_CONNECTION=$(config-get mongo_connection)
45 RABBIT_CONNECTION=$(config-get rabbit_connection)
46 API_NAT_SOCKET=$(config-get api_nat_socket)
47
48 STORAGE_IP=$(config-get storage_ip)
49 STORAGE_NAT_IP=$(config-get storage_nat_ip)
50 STORAGE_FSTYPE=$(config-get storage_fstype)
51 STORAGE_MOUNTPOINT=$(config-get storage_mountpoint)
52 STORAGE_OPTIONS=$(config-get storage_options)
53
54 # Utilities =====
55
56 storage_config_is_enabled()
57 {
58     [ "$STORAGE_IP" -a "$STORAGE_FSTYPE" -a "$STORAGE_MOUNTPOINT" ]
59 }
60
61 storage_is_mounted()
62 {

```

```

63     mount | grep -q "$STORAGE_ROOT_PATH"
64 }
65
66 storage_remount()
67 {
68     # Overrides storage parameters with charm configuration
69     if storage_config_is_enabled; then # if storage options are set
70         ip=$STORAGE_IP
71         nat_ip=$STORAGE_NAT_IP
72         fstype=$STORAGE_FSTYPE
73         mountpoint=$STORAGE_MOUNTPOINT
74         options=$STORAGE_OPTIONS
75     # Or uses storage parameters from charm storage relation
76     elif [ $# -eq 4 ]; then # if function parameters are set
77         ip=$1
78         nat_ip=''
79         fstype=$2
80         mountpoint=$3
81         options=$4
82     elif [ $# -eq 0 ]; then
83         return
84     else
85         xecho "Usage: $(basename $0).storage_remount ip fstype mountpoint options"
86     fi
87
88     if [ "$nat_ip" ]; then
89         pecho "Update hosts file to map storage internal address $ip to $nat_ip"
90         if grep -q "$ip" /etc/hosts; then
91             sed -i "s<$nat_ip .*<$nat_ip $ip<" /etc/hosts
92         else
93             echo "$nat_ip $ip" >> /etc/hosts
94         fi
95     else
96         nat_ip=$ip
97     fi
98
99     storage_umount
100
101 r=$STORAGE_ROOT_PATH
102 pecho "Mount shared storage [$nat_ip] $ip:$mountpoint type $fstype options '$options' -> $r"
103 if [ ! -d "$STORAGE_ROOT_PATH" ]; then
104     mkdir "$STORAGE_ROOT_PATH" || xecho "Unable to create shared storage path $STORAGE_ROOT_PATH" 1
105 fi
106
107 # FIXME try 5 times, a better way to handle failure
108 for i in $(seq 1 5)
109 do
110     if storage_is_mounted; then
111         break
112     else
113         if [ "$options" ]
114             then mount -t "$fstype" -o "$options" "$nat_ip:$mountpoint" "$STORAGE_ROOT_PATH"
115         else mount -t "$fstype" "$nat_ip:$mountpoint" "$STORAGE_ROOT_PATH"
116         fi
117     fi
118     sleep 5
119 done
120
121 if storage_is_mounted; then
122     # FIXME update /etc/fstab (?)
123     pecho 'Configure Publisher : Register shared storage'
124     setSettingJSON_STRING "$CONFIG_FILE" 'storage_ip'           "$ip"          || xecho 'Config' 2
125     setSettingJSON_STRING "$CONFIG_FILE" 'storage_fstype'      "$fstype"       || xecho 'Config' 3
126     setSettingJSON_STRING "$CONFIG_FILE" 'storage_mountpoint' "$mountpoint" || xecho 'Config' 4
127     setSettingJSON_STRING "$CONFIG_FILE" 'storage_options'     "$options"      || xecho 'Config' 5
128 else
129     xecho 'Unable to mount shared storage' 6
130 fi
131 }
132
133 storage_umount()
134 {
135     pecho 'Configure Publisher : Unregister shared storage'

```

```

136 setSettingJSON_STRING "$CONFIG_FILE" 'storage_ip'      '' || xecho 'Config' 1
137 setSettingJSON_STRING "$CONFIG_FILE" 'storage_fstype'   '' || xecho 'Config' 2
138 setSettingJSON_STRING "$CONFIG_FILE" 'storage_mountpoint' '' || xecho 'Config' 3
139 setSettingJSON_STRING "$CONFIG_FILE" 'storage_options'  '' || xecho 'Config' 4
140
141 if storage_is_mounted; then
142     # FIXME update /etc/fstab (?)
143     pecho 'Unmount shared storage (is actually mounted)'
144     umount "$STORAGE_ROOT_PATH" || xecho 'Unable to unmount shared storage' 5
145     recho 'Shared storage successfully unmounted'
146 else
147     recho 'Shared storage already unmounted'
148 fi
149 }
150
151 storage_hook_bypass()
152 {
153     if storage_config_is_enabled; then
154         xecho 'Shared storage is set in config, storage relation is disabled' 1
155     fi
156 }
157
158 publisher_config_is_enabled()
159 {
160     [ "$MONGO_CONNECTION" -a "$RABBIT_CONNECTION" ]
161 }
162
163 publisher_hook_bypass()
164 {
165     if publisher_config_is_enabled; then
166         xecho 'Orchestrator is set in config, publisher relation is disabled' 1
167     fi
168 }
169
170 publisher_register()
171 {
172     # Overrides publisher parameters with charm configuration
173     if publisher_config_is_enabled; then # if publisher options are set
174         mongo=$MONGO_CONNECTION
175         rabbit=$RABBIT_CONNECTION
176         socket=$API_NAT_SOCKET
177         # Or uses publisher parameters from charm publisher relation
178     elif [ $# -eq 2 ]; then # if function parameters are set
179         mongo=$1
180         rabbit=$2
181         socket=''
182     elif [ $# -eq 0 ]; then
183         return
184     else
185         xecho "Usage: $(basename $0).publisher_register mongo rabbit"
186     fi
187
188     pecho 'Configure Publisher : Register the Orchestrator'
189     setSettingJSON_STRING "$CONFIG_FILE" 'mongo_connection' "$mongo" || xecho 'Config' 1
190     setSettingJSON_STRING "$CONFIG_FILE" 'rabbit_connection' "$rabbit" || xecho 'Config' 2
191     setSettingJSON_STRING "$CONFIG_FILE" 'api_nat_socket'    "$socket" || xecho 'Config' 3
192
193     host=$(expr match "$mongo" '.*mongodb://[^:]*:[^@]*@\([^\:]*\):[0-9]*/[a-z]*.*')
194     port=$(expr match "$mongo" '.*mongodb://[^:]*:[^@]*@[^\:]*:\([0-9]*\)/[a-z]*.*')
195     user=$(expr match "$mongo" '.*mongodb://\([^\:]*\):[^@]*@[^\:]*:[0-9]*/[a-z]*.*')
196     password=$(expr match "$mongo" '.*mongodb://[^:]*:\([^\:]*\)[^@]*:[0-9]*/[a-z]*.*')
197     database=$(expr match "$mongo" '.*mongodb://[^:]*:[^@]*@[^\:]*:[0-9]*/\([a-z]*\).*')
198     mecho "MongoDB host=$host, port=$port, user=$user, password=$password, database=$database"
199     if [ ! "$host" -o ! "$port" -o ! "$user" -o ! "$password" -o ! "$database" ]; then
200         xecho 'Unable to parse MongoDB connection' 3
201     fi
202
203     a="s<RABBIT_CONNECTION<$rabbit<g"
204     b="s<MONGO_HOST<$host<g"
205     c="s<MONGO_PORT<$port<g"
206     d="s<MONGO_USER<$user<g"
207     e="s<MONGO_PASSWORD<$password<g"
208     f="s<MONGO_DATABASE<$database<g"

```

```

209 g="s<THE_CONCURRENCY<$THE_CONCURRENCY<g"
210 sed "$a;$b;$c;$d;$e;$f;$g" "$CELERY_TEMPL_FILE" > "$CELERY_CONFIG_FILE" || xecho 'Config' 4
211 recho "Orchestrator successfully registered, it's time to wake-up"
212 }
213
214 publisher_unregister()
215 {
216     pecho 'Configure Publisher : Unregister the Orchestrator'
217     setSettingJSON_STRING "$CONFIG_FILE" 'mongodb_connection' '' || xecho 'Config' 1
218     setSettingJSON_STRING "$CONFIG_FILE" 'rabbitmq_connection' '' || xecho 'Config' 2
219     setSettingJSON_STRING "$CONFIG_FILE" 'api_nat_socket' '' || xecho 'Config' 3
220     rm -f "$CELERY_CONFIG_FILE"
221     recho 'Orchestrator successfully unregistered'
222 }
223
224 # HOOKS : Charm Setup =====
225
226 hook_install()
227 {
228     techo 'Publisher - install'
229
230     # Remark : You must get dependencies into this charm by executing get-dependencies.sh
231     # FIXME share variables with get-dependencies.sh
232
233     # Fix memtest86+ : https://bugs.launchpad.net/ubuntu/+source/grub2/+bug/1069856
234     #eval $purge grub-pc grub-common
235     #eval $install grub-common grub-pc
236
237     eval $update
238     eval $upgrade
239
240     pecho 'Install and configure Network Time Protocol'
241     eval $install ntp || xecho 'Unable to install ntp' 1
242     eval $service ntp restart || xecho 'Unable to restart ntp service' 2
243
244     pecho 'Install python, Apache 2 Web Server and Gluster Filesystem'
245     eval $install apache2 apache2-threaded-dev make python python-dev python-pip glusterfs-client \
246         nfs-common || xecho 'Unable to install packages' 3
247     #rm -rf /var/www/*
248
249     pecho 'Install Apache 2 H.264 streaming module'
250     lib='mod_h264_streaming-2.2.7'
251     tar="apache_${lib}.tar.gz"
252     tar -zvxf $tar
253     cd $lib || xecho "Unable to find directory $lib" 4
254     ./configure --with-apxs=$(which apxs2)
255     make && make install || xecho 'Unable to compile module' 5
256     cd ..
257     rm -rf $lib
258     #rm -f $zip
259
260     config='/etc/apache2/apache2.conf'
261     module='/usr/lib/apache2/modules/mod_h264_streaming.so'
262     if ! grep -q $module $config; then
263         echo "LoadModule h264_streaming_module $module" >> $config
264         echo 'AddHandler h264-streaming.extensions .mp4' >> $config
265     fi
266
267     pecho 'Install BSON Binary JSON, Celery Distributed Task Queue, MongoDB API and Requests'
268     pip install --upgrade bson celery ipaddr pymongo requests || xecho 'Unable to install packages' 6
269
270     pecho 'Expose Apache 2 service'
271     open-port 80/tcp
272
273     # FIXME this call is not necessary, but config-changed create an infinite loop, so WE call it
274     hook_config_changed
275 }
276
277 hook_uninstall()
278 {
279     techo 'Publisher - uninstall'
280
281     hook_stop

```

```

282 eval $purge apache2 apache2.2-common glusterfs-client nfs-common
283 pip install uninstall bson celery
284 eval $autoremove
285 rm -rf /etc/apache2/ /var/www/ /var/log/apache2/
286 mkdir /var/www/
287 }
288
289 hook_config_changed()
290 {
291     techo 'Publisher - config changed'
292
293     hook_stop
294     # FIXME FIXME pecho 'Configure Apache : Set limits (upload size, execution time, ...)'
295     #c='Config'
296     #upload='upload_max_filesize'
297     #execution='max_execution_time'
298     #setSettingHTA $HTACCESS_FILE $true "php_value $upload"      "$MAX_UPLOAD_SIZE"    || xecho $c 13
299     #setSettingHTA $HTACCESS_FILE $true 'php_value post_max_size' "$MAX_UPLOAD_SIZE"    || xecho $c 14
300     #setSettingHTA $HTACCESS_FILE $true "php_value $execution"    "$MAX_EXECUTION_TIME" || xecho $c 15
301     #setSettingHTA $HTACCESS_FILE $true 'php_value max_input_time' "$MAX_INPUT_TIME"   || xecho $c 16
302
303     pecho 'Configure Publisher : Set verbose, messaging queues and storage path'
304     setSettingJSON_STRING "$CONFIG_FILE" 'public_address' "$PUBLIC_ADDRESS"    || xecho 'Config' 1
305     setSettingJSON_BOOLEAN "$CONFIG_FILE" 'verbose'      "$VERBOSE"           || xecho 'Config' 2
306     setSettingJSON_STRING "$CONFIG_FILE" 'rabbit_queues' "$RABBIT_QUEUES"    || xecho 'Config' 3
307     setSettingJSON_STRING "$CONFIG_FILE" 'storage_path'  "$STORAGE_ROOT_PATH" || xecho 'Config' 4
308     storage_remount
309     publisher_register
310     hook_start
311     # FIXME infinite loop is used as config-changed hook !
312 }
313
314 # HOOKS : Charm Service =====
315
316 hook_start()
317 {
318     techo 'Publisher - start'
319
320     if ! storage_is_mounted; then
321         recho 'WARNING Do not start Publisher daemon : No shared storage'
322     elif [ ! -f "$CELERY_CONFIG_FILE" ]; then
323         recho 'WARNING Do not start Publisher daemon : No Celery configuration file'
324     elif [ ! "$RABBIT_QUEUES" ]; then
325         recho 'WARNING Do not start Publisher daemon : No RabbitMQ queue(s) declared'
326     else
327         service apache2 start || xecho 'Unable to start Apache 2' 1
328         if ! screenRunning 'Publisher'; then
329             cd "$BASE_PATH" || xecho "Unable to find path $BASE_PATH"
330             screenLaunch 'Publisher' celeryd --config 'celeryconfig' -Q "$RABBIT_QUEUES" || \
331                 xecho 'Unable to start Publisher daemon' 2
332         fi
333         sleep 5
334         if ! screenRunning 'Publisher'; then
335             xecho 'Publisher is not ready' 3
336         else
337             recho 'Publisher successfully started'
338         fi
339     fi
340 }
341
342 hook_stop()
343 {
344     techo 'Publisher - stop'
345
346     if screenRunning 'Publisher'; then
347         screenKill 'Publisher' || xecho 'Unable to stop Publisher daemon' 1
348     fi
349     service apache2 stop || xecho 'Unable to stop Apache 2' 2
350 }
351
352 # HOOKS : Requires Storage =====
353
354 hook_storage_relation_joined()

```

```

355  {
356      techo 'Publisher - storage relation joined'
357      storage_hook_bypass
358  }
359
360 hook_storage_relation_changed()
361 {
362     techo 'Publisher - storage relation changed'
363     storage_hook_bypass
364
365     # Get configuration from the relation
366     ip=$(relation-get private-address)
367     fstype=$(relation-get fstype)
368     mountpoint=$(relation-get mountpoint)
369     options=$(relation-get options)
370
371     mecho "Storage IP is $ip, fstype: $fstype, mountpoint: $mountpoint, options: $options"
372     if [ ! "$ip" -o ! "$fstype" -o ! "$mountpoint" ]; then
373         recho 'Waiting for complete setup'
374         exit 0
375     fi
376
377     hook_stop
378     storage_remount "$ip" "$fstype" "$mountpoint" "$options"
379     hook_start
380 }
381
382 hook_storage_relation_broken()
383 {
384     techo 'Publisher - storage relation broken'
385     storage_hook_bypass
386
387     hook_stop
388     storage_umount
389 }
390
391 # HOOKS : Requires Publisher =====
392
393 hook_publisher_relation_joined()
394 {
395     techo 'Publisher - publisher relation joined'
396     publisher_hook_bypass
397 }
398
399 hook_publisher_relation_changed()
400 {
401     techo 'Publisher - publisher relation changed'
402     publisher_hook_bypass
403
404     # Get configuration from the relation
405     orchestra_ip=$(relation-get private-address)
406     mongo=$(relation-get mongo_connection)
407     rabbit=$(relation-get rabbit_connection)
408     mecho "Orchestra IP is $orchestra_ip, MongoDB is $mongo, RabbitMQ is $rabbit"
409     if [ ! "$orchestra_ip" -o ! "$mongo" -o ! "$rabbit" ]; then
410         recho 'Waiting for complete setup'
411         exit 0
412     fi
413     hook_stop
414     publisher_register "$mongo" "$rabbit"
415     hook_start
416 }
417
418 hook_publisher_relation_broken()
419 {
420     techo 'Publisher - publisher relation broken'
421     publisher_hook_bypass
422
423     hook_stop
424     publisher_unregister
425 }
426
427 # HOOKS : Provides Website =====

```

```

428
429 hook_website_relation_joined()
430 {
431     techo 'Web UI - website relation joined'
432
433     # Send port & hostname
434     relation-set port=80 hostname=$(hostname -f)
435 }
```

6.10.5 templates/celeryconfig.py.template

```

1 import sys
2
3 sys.path.append('..')
4
5 BROKER_URL = 'RABBIT_CONNECTION'
6 CELERY_RESULT_BACKEND = 'mongodb'
7 CELERY_MONGODB_BACKEND_SETTINGS = {
8     'host': 'MONGO_HOST',
9     'port': 'MONGO_PORT',
10    'user': 'MONGO_USER',
11    'password': 'MONGO_PASSWORD',
12    'database': 'MONGO_DATABASE',
13    'taskmeta_collection': 'taskmeta',
14 }
15
16 CELERY_IMPORTS = ('lib.Publisher',)
17 CELERYD_CONCURRENCY = THE_CONCURRENCY
18 CELERY_IGNORE_RESULT = False
19 CELERY_SEND_EVENTS = True
20 CELERY_TASK_SERIALIZER = 'json'
21 CELERY_TASK_COMPRESSION = 'zlib'
22 CELERY_TRACK_STARTED = True
```

6.11 OSCIED - Transform Source Code

Note: config.json, copyright and revision are skipped as they are not so relevant here.

6.11.1 config.yaml

```

1 options:
2     verbose:
3         type: string
4         default: "false"
5         description: Set verbose logging. Can be set to true or false
6     concurrency:
7         type: int
8         default: 1
9         description: Amount of tasks the worker can handle simultaneously. Default value is conservative
10    rabbit_queues:
11        type: string
12        default: "transform_private"
13        description: Worker connect to queues to receive jobs. (e.g. "transform_amazon" for amazon, "transform_pr...
14    mongo_connection:
15        type: string
16        default: ""
17        description: Orchestrator database connection. The 'connection' options override and disable transform re...
18    rabbit_connection:
19        type: string
20        default: ""
21        description: Orchestrator message broker connection
22    api_nat_socket:
23        type: string
24        default: ""
```

```

25     description: Orchestrator API reacheable (public) address (ip:port) e.g. when orchestra is behind a NAT/FW
26 storage_ip:
27     type: string
28     default: ""
29     description: Shared storage hostname / IP address (see interface mount of nfs charm). The 'storage' option
30 storage_nat_ip:
31     type: string
32     default: ""
33     description: Shared storage reacheable (public) address e.g. when storage is behind a NAT/FW.
34 storage_fstype:
35     type: string
36     default: ""
37     description: Shared storage filesystem type (e.g. nfs)
38 storage_mountpoint:
39     type: string
40     default: ""
41     description: Shared storage mountpoint (e.g. for nfs - /srv/data)
42 storage_options:
43     type: string
44     default: ""
45     description: Shared storage options (e.g. for nfs - rw, sync,no_subtree_check)

```

6.11.2 metadata.yaml

```

1 ensemble: oscied
2 name: oscied-transform
3 summary: "EBU's OSCIED Encoder"
4 maintainer: OSCIED Main Developper <david.fischer.ch@gmail.com>
5 description: |
6     Installs the EBU's OSCIED media transform. Contains ffmpeg & x264.
7     Can be scaled to multiple instances.
8 requires:
9     transform:
10        interface: subordinate
11     storage:
12        interface: mount

```

6.11.3 hooks_lib/common.sh.lu-dep

```

1 set -o nounset # will exit if an uninitialized variable is used
2
3 # Constants -----
4
5 ECHO='juju-log' # Used by logicielsUbuntuUtils
6
7 # Charms paths
8 BASE_PATH=$(pwd)
9
10 # Charms files
11 CONFIG_FILE="$BASE_PATH/config.json"
12 CELERY_TEMPFILE="$BASE_PATH/templates/celeryconfig.py.template"
13 CELERY_CONFIG_FILE="$BASE_PATH/celeryconfig.py"
14
15 # Shared storage paths
16 STORAGE_ROOT_PATH='/mnt/storage'
17 STORAGE_MEDIAS_PATH="$STORAGE_ROOT_PATH/medias"
18 STORAGE_UPLOADS_PATH="$STORAGE_ROOT_PATH/uploads"
19
20 # Configuration -----
21
22 if [ "$(config-get verbose)" = 'true' ] ; then
23     VERBOSE=0      # true
24     set -o xtrace # for verbose logging to juju debug-log
25 else
26     VERBOSE=1 # false
27 fi
28
29 PUBLIC_ADDRESS=$(unit-get public-address)
30 THE_CONCURRENCY=$(config-get concurrency)

```

```

31 RABBIT_QUEUES=$(config-get rabbit_queues),$PUBLIC_ADDRESS"
32
33 MONGO_CONNECTION=$(config-get mongo_connection)
34 RABBIT_CONNECTION=$(config-get rabbit_connection)
35 API_NAT_SOCKET=$(config-get api_nat_socket)
36
37 STORAGE_IP=$(config-get storage_ip)
38 STORAGE_NAT_IP=$(config-get storage_nat_ip)
39 STORAGE_FSTYPE=$(config-get storage_fstype)
40 STORAGE_MOUNTPOINT=$(config-get storage_mountpoint)
41 STORAGE_OPTIONS=$(config-get storage_options)
42
43 # Utilities =====
44
45 storage_config_is_enabled()
46 {
47     [ "$STORAGE_IP" -a "$STORAGE_FSTYPE" -a "$STORAGE_MOUNTPOINT" ]
48 }
49
50 storage_is_mounted()
51 {
52     mount | grep -q "$STORAGE_ROOT_PATH"
53 }
54
55 storage_remount()
56 {
57     # Overrides storage parameters with charm configuration
58     if storage_config_is_enabled; then # if storage options are set
59         ip=$STORAGE_IP
60         nat_ip=$STORAGE_NAT_IP
61         fstype=$STORAGE_FSTYPE
62         mountpoint=$STORAGE_MOUNTPOINT
63         options=$STORAGE_OPTIONS
64     # Or uses storage parameters from charm storage relation
65     elif [ $# -eq 4 ]; then # if function parameters are set
66         ip=$1
67         nat_ip=''
68         fstype=$2
69         mountpoint=$3
70         options=$4
71     elif [ $# -eq 0 ]; then
72         return
73     else
74         xecho "Usage: $(basename $0).storage_remount ip fstype mountpoint options"
75     fi
76
77     if [ "$nat_ip" ]; then
78         pecho "Update hosts file to map storage internal address $ip to $nat_ip"
79         if grep -q "$ip" /etc/hosts; then
80             sed -i "s<$nat_ip .*<$nat_ip $ip<" /etc/hosts
81         else
82             echo "$nat_ip $ip" >> /etc/hosts
83         fi
84     else
85         nat_ip=$ip
86     fi
87
88     storage_umount
89
90     r=$STORAGE_ROOT_PATH
91     pecho "Mount shared storage [$nat_ip] $ip:$mountpoint type $fstype options '$options' -> $r"
92     if [ ! -d "$STORAGE_ROOT_PATH" ]; then
93         mkdir "$STORAGE_ROOT_PATH" || xecho "Unable to create shared storage path $STORAGE_ROOT_PATH"
94     fi
95
96     # FIXME try 5 times, a better way to handle failure
97     for i in $(seq 1 5)
98     do
99         if storage_is_mounted; then
100             break
101         else
102             if [ "$options" ]
103                 then mount -t "$fstype" -o "$options" "$nat_ip:$mountpoint" "$STORAGE_ROOT_PATH"

```

```

104     else mount -t "$fstype"           "$nat_ip:$mountpoint" "$STORAGE_ROOT_PATH"
105     fi
106   fi
107   sleep 5
108 done
109
110 if storage_is_mounted; then
111   # FIXME update /etc/fstab (?)
112   pecho 'Configure Transform : Register shared storage'
113   setSettingJSON_STRING "$CONFIG_FILE" 'storage_ip'      "$ip"          || xecho 'Config' 2
114   setSettingJSON_STRING "$CONFIG_FILE" 'storage_fstype'   "$fstype"      || xecho 'Config' 3
115   setSettingJSON_STRING "$CONFIG_FILE" 'storage_mountpoint' "$mountpoint" || xecho 'Config' 4
116   setSettingJSON_STRING "$CONFIG_FILE" 'storage_options'   "$options"     || xecho 'Config' 5
117 else
118   xecho 'Unable to mount shared storage' 6
119 fi
120 }
121
122 storage_umount()
123 {
124   pecho 'Configure Transform : Unregister shared storage'
125   setSettingJSON_STRING "$CONFIG_FILE" 'storage_ip'      '' || xecho 'Config' 1
126   setSettingJSON_STRING "$CONFIG_FILE" 'storage_fstype'   '' || xecho 'Config' 2
127   setSettingJSON_STRING "$CONFIG_FILE" 'storage_mountpoint' '' || xecho 'Config' 3
128   setSettingJSON_STRING "$CONFIG_FILE" 'storage_options'   '' || xecho 'Config' 4
129
130 if storage_is_mounted; then
131   # FIXME update /etc/fstab (?)
132   pecho 'Unmount shared storage (is actually mounted)'
133   umount "$STORAGE_ROOT_PATH" || xecho 'Unable to unmount shared storage' 5
134   recho 'Shared storage successfully unmounted'
135 else
136   recho 'Shared storage already unmounted'
137 fi
138 }
139
140 storage_hook_bypass()
141 {
142   if storage_config_is_enabled; then
143     xecho 'Shared storage is set in config, storage relation is disabled' 1
144   fi
145 }
146
147 transform_config_is_enabled()
148 {
149   [ "$MONGO_CONNECTION" -a "$RABBIT_CONNECTION" ]
150 }
151
152 transform_hook_bypass()
153 {
154   if transform_config_is_enabled; then
155     xecho 'Orchestrator is set in config, transform relation is disabled' 1
156   fi
157 }
158
159 transform_register()
160 {
161   # Overrides transform parameters with charm configuration
162   if transform_config_is_enabled; then # if transform options are set
163     mongo=$MONGO_CONNECTION
164     rabbit=$RABBIT_CONNECTION
165     socket=$API_NAT_SOCKET
166   # Or uses transform parameters from charm transform relation
167   elif [ $# -eq 2 ]; then # if function parameters are set
168     mongo=$1
169     rabbit=$2
170     socket=''
171   elif [ $# -eq 0 ]; then
172     return
173   else
174     xecho "Usage: $(basename $0).transform_register mongo rabbit"
175   fi
176 }
```

```

177 pecho 'Configure Transform : Register the Orchestrator'
178 setSettingJSON_STRING "$CONFIG_FILE" 'mongo_connection' "$mongo" || xecho 'Config' 1
179 setSettingJSON_STRING "$CONFIG_FILE" 'rabbit_connection' "$rabbit" || xecho 'Config' 2
180 setSettingJSON_STRING "$CONFIG_FILE" 'api_nat_socket' "$socket" || xecho 'Config' 3
181
182 host=$(expr match "$mongo" '.*mongodb://[^:]*:[^@]*@([:^]*\:[0-9]*/[a-z]*.*')
183 port=$(expr match "$mongo" '.*mongodb://[^:]*:[^@]*@[^\:\*:]\([0-9]*\)/[a-z]*.*')
184 user=$(expr match "$mongo" '.*mongodb://\([^\:\*:]\:[^@]*@[^\:\*:][0-9]*/[a-z]*.*')
185 password=$(expr match "$mongo" '.*mongodb://[^:]*:\([^\@\:\*\@]\)[^\:\*:][0-9]*/[a-z]*.*')
186 database=$(expr match "$mongo" '.*mongodb://[^:]*:[^@]*@[^\:\*:][0-9]*\([a-z]*\).*')
187 mecho "MongoDB host=$host, port=$port, user=$user, password=$password, database=$database"
188 if [ ! "$host" -o ! "$port" -o ! "$user" -o ! "$password" -o ! "$database" ]; then
189   xecho 'Unable to parse MongoDB connection' 3
190 fi
191
192 a="s<RABBIT_CONNECTION<$rabbit<g"
193 b="s<MONGO_HOST<$host<g"
194 c="s<MONGO_PORT<$port<g"
195 d="s<MONGO_USER<$user<g"
196 e="s<MONGO_PASSWORD<$password<g"
197 f="s<MONGO_DATABASE<$database<g"
198 g="s<THE_CONCURRENCY<$THE_CONCURRENCY<g"
199 sed "$a;$b;$c;$d;$e;$f;$g" "$CELERY_TEMPL_FILE" > "$CELERY_CONFIG_FILE" || xecho 'Config' 4
200 recho "Orchestrator successfully registered, it's time to wake-up"
201 }
202
203 transform_unregister()
204 {
205   pecho 'Configure Transform : Unregister the Orchestrator'
206   setSettingJSON_STRING "$CONFIG_FILE" 'mongodb_connection' '' || xecho 'Config' 1
207   setSettingJSON_STRING "$CONFIG_FILE" 'rabbitmq_connection' '' || xecho 'Config' 2
208   setSettingJSON_STRING "$CONFIG_FILE" 'api_nat_socket' '' || xecho 'Config' 3
209   rm -f "$CELERY_CONFIG_FILE"
210   recho 'Orchestrator successfully unregistered'
211 }
212
213 # HOOKS : Charm Setup =====
214
215 hook_install()
216 {
217   techo 'Transform - install'
218
219   # Fix memtest86+ : https://bugs.launchpad.net/ubuntu/+source/grub2/+bug/1069856
220   #eval $purge grub-pc grub-common
221   #eval $install grub-common grub-pc
222
223   # I decided to use the real ffmpeg, not the libav version :
224   # http://blog.pkh.me/p/13-the-ffmpeg-libav-situation.html
225   # http://doc.ubuntu-fr.org/ffmpeg
226   apt-add-repository -y ppa:jon-severinsson/ffmpeg
227
228   eval $update
229   eval $upgrade
230
231   pecho 'Install and configure Network Time Protocol'
232   eval $install ntp || xecho 'Unable to install ntp' 1
233   eval $service ntp restart || xecho 'Unable to restart ntp service' 2
234
235   pecho 'Install python, (the real) FFmpeg, x264 and Gluster Filesystem'
236   eval $install ffmpeg x264 python python-dev python-pip glusterfs-client nfs-common || \
237     xecho 'Unable to install packages' 3
238
239   pecho 'Install BSON Binary JSON, Celery Distributed Task Queue, MongoDB API and Requests'
240   pip install --upgrade bson celery ipaddr pymongo requests || xecho 'Unable to install packages' 4
241
242   # FIXME this call is not necessary, but config-changed may create an infinite loop, so WE call it
243   hook_config_changed
244 }
245
246 hook_uninstall()
247 {
248   techo 'Transform - uninstall'
249

```

```

250     hook_stop
251     eval $purge ffmpeg x264 glusterfs-client nfs-common
252     pip install uninstall bson celery
253     eval $autoremove
254 }
255
256 hook_config_changed()
257 {
258     techo 'Transform - config changed'
259
260     hook_stop
261     pecho 'Configure Transform : Set verbose, messaging queues and storage path'
262     setSettingJSON_STRING "$CONFIG_FILE" 'public_address' "$PUBLIC_ADDRESS" || xecho 'Config' 1
263     setSettingJSON_BOOLEAN "$CONFIG_FILE" 'verbose' "$VERBOSE" || xecho 'Config' 2
264     setSettingJSON_STRING "$CONFIG_FILE" 'rabbit_queues' "$RABBIT_QUEUES" || xecho 'Config' 3
265     setSettingJSON_STRING "$CONFIG_FILE" 'storage_path' "$STORAGE_ROOT_PATH" || xecho 'Config' 4
266     storage_remount
267     transform_register
268     hook_start
269     # FIXME infinite loop is used as config-changed hook !
270 }
271
272 # HOOKS : Charm Service =====
273
274 hook_start()
275 {
276     techo 'Transform - start'
277
278     if ! storage_is_mounted; then
279         recho 'WARNING Do not start Transform daemon : No shared storage'
280     elif [ ! -f "$CELERY_CONFIG_FILE" ]; then
281         recho 'WARNING Do not start Transform daemon : No Celery configuration file'
282     elif [ ! "$RABBIT_QUEUES" ]; then
283         recho 'WARNING Do not start Transform daemon : No RabbitMQ queue(s) declared'
284     else
285         if ! screenRunning 'Transform'; then
286             cd "$BASE_PATH" || xecho "Unable to find path $BASE_PATH"
287             screenLaunch 'Transform' celeryd --config 'celeryconfig' -Q "$RABBIT_QUEUES" || \
288             xecho 'Unable to start Transform daemon' 1
289     fi
290     sleep 5
291     if ! screenRunning 'Transform'; then
292         xecho 'Transform is not ready' 2
293     else
294         recho 'Transform successfully started'
295     fi
296 fi
297 }
298
299 hook_stop()
300 {
301     techo 'Transform - stop'
302
303     if screenRunning 'Transform'; then
304         screenKill 'Transform' || xecho 'Unable to stop Transform daemon'
305     fi
306 }
307
308 # HOOKS : Requires Storage =====
309
310 hook_storage_relation_joined()
311 {
312     techo 'Transform - storage relation joined'
313     storage_hook_bypass
314 }
315
316 hook_storage_relation_changed()
317 {
318     techo 'Transform - storage relation changed'
319     storage_hook_bypass
320
321     # Get configuration from the relation
322     ip=$(relation-get private-address)

```

```

323  fstype=$(relation-get fstype)
324  mountpoint=$(relation-get mountpoint)
325  options=$(relation-get options)
326
327  mecho "Storage IP is $ip, fstype: $fstype, mountpoint: $mountpoint, options: $options"
328  if [ ! "$ip" -o ! "$fstype" -o ! "$mountpoint" ]; then
329    recho 'Waiting for complete setup'
330    exit 0
331  fi
332
333  hook_stop
334  storage_remount "$ip" "$fstype" "$mountpoint" "$options"
335  hook_start
336 }
337
338 hook_storage_relation_broken()
{
339   techo 'Transform - storage relation broken'
340   storage_hook_bypass
341
342   hook_stop
343   storage_umount
344 }
345
346
347 # HOOKS : Requires Transform =====
348
349 hook_transform_relation_joined()
{
350   techo 'Transform - transform relation joined'
351   transform_hook_bypass
352 }
353
354 hook_transform_relation_changed()
{
355   techo 'Transform - transform relation changed'
356   transform_hook_bypass
357
358   # Get configuration from the relation
359   orchestra_ip=$(relation-get private-address)
360   mongo=$(relation-get mongo_connection)
361   rabbit=$(relation-get rabbit_connection)
362   mecho "Orchestra IP is $orchestra_ip, MongoDB is $mongo, RabbitMQ is $rabbit"
363   if [ ! "$orchestra_ip" -o ! "$mongo" -o ! "$rabbit" ]; then
364     recho 'Waiting for complete setup'
365     exit 0
366   fi
367   hook_stop
368   transform_register "$mongo" "$rabbit"
369   hook_start
370 }
371
372
373 hook_transform_relation_broken()
{
374   techo 'Transform - transform relation broken'
375   transform_hook_bypass
376
377   hook_stop
378   transform_unregister
379 }
380
381 }
```

6.11.4 templates/celeryconfig.py.template

```

1 import sys
2
3 sys.path.append('..')
4
5 BROKER_URL = 'RABBIT_CONNECTION'
6 CELERY_RESULT_BACKEND = 'mongodb'
7 CELERY_MONGODB_BACKEND_SETTINGS = {
8   'host': 'MONGO_HOST',
9   'port': 'MONGO_PORT',
```

```

10     'user': 'MONGO_USER',
11     'password': 'MONGO_PASSWORD',
12     'database': 'MONGO_DATABASE',
13     'taskmeta_collection': 'taskmeta',
14 }
15
16 CELERY_IMPORTS = ('lib.Transform',)
17 CELERYD_CONCURRENCY = THE_CONCURRENCY
18 CELERY_IGNORE_RESULT = False
19 CELERY_SEND_EVENTS = True
20 CELERY_TASK_SERIALIZER = 'json'
21 CELERY_TASK_COMPRESSION = 'zlib'
22 CELERY_TRACK_STARTED = True

```

6.12 OSCIED - Storage Source Code

Note: copyright and revision are skipped as they are not so relevant here.

6.12.1 config.yaml

```

1 options:
2   verbose:
3     type: string
4     default: "false"
5     description: Set verbose logging. Can be set to true or false

```

6.12.2 metadata.yaml

```

1 ensemble: oscied
2 name: oscied-storage
3 summary: "EBU's OSCIED Storage"
4 maintainer: OSCIED Main Developper <david.fischer.ch@gmail.com>
5 description: |
6   Installs the EBU's OSCIED shared storage based on glusterfs.
7 provides:
8   storage:
9     interface: mount

```

6.12.3 hooks_lib/common.sh.lu-dep

```

1 set -o nounset # will exit if an uninitialized variable is used
2
3 # Constants =====
4
5 ECHO='juju-log' # Used by logicielsUbuntuUtils
6 MEDIAS_VOLUME='medias_volume'
7 MEDIAS_BRICK='exp1'
8
9 # Configuration =====
10
11 if [ "$(config-get verbose)" = 'true' ] ; then
12   VERBOSE=0      # true
13   set -o xtrace # for verbose logging to juju debug-log
14 else
15   VERBOSE=1 # false
16 fi
17
18 # Utilities =====
19
20 # HOOKS : Charm Setup =====
21
22 hook_install()

```

```

23  {
24      techo ' Storage - install'
25
26      # Fix memtest86+ : https://bugs.launchpad.net/ubuntu/+source/grub2/+bug/1069856
27      #eval $purge grub-pc grub-common
28      #eval $install grub-common grub-pc
29
30      eval $update
31      eval $upgrade
32
33      pecho 'Install and configure Network Time Protocol'
34      eval $install ntp || xecho 'Unable to install ntp' 1
35      eval $service ntp restart || xecho 'Unable to restart ntp service' 2
36
37      pecho 'Install GlusterFS Server'
38      eval $install glusterfs-server nfs-common || xecho 'Unable to install packages' 3
39
40      pecho "Create medias volume $MEDIAS_VOLUME"
41      ip=$(unit-get private-address)
42      gluster volume create "$MEDIAS_VOLUME" "$ip:$MEDIAS_BRICK" || \
43          xecho "Unable to create medias volume $MEDIAS_VOLUME on brick $ip:$MEDIAS_BRICK" 4
44
45      gluster volume start "$MEDIAS_VOLUME" || xecho "Unable to start volume $MEDIAS_VOLUME" 5
46      gluster volume info
47
48      juju-log 'Expose GlusterFS Server service'
49      open-port 111/tcp    # is used for portmapper, and should have both TCP and UDP open
50      open-port 24007/tcp  # for the Gluster Daemon
51      #open-port 24008/tcp # Infiniband management (optional unless you are using IB)
52      open-port 24009/tcp # We have only 1 storage brick (24009-24009)
53      # For NFS (not used)
54      #open-port 38465/tcp
55      #open-port 38466/tcp
56      #open-port 38467/tcp
57  }
58
59  hook_uninstall()
60  {
61      techo 'Storage - uninstall'
62
63      hook_stop
64
65      eval $purge glusterfs-server nfs-common
66      eval $autoremove
67  }
68
69  # HOOKS : Charm Service =====
70
71  hook_start()
72  {
73      techo 'Storage - start'
74
75      if ! service glusterfs-server status | grep -q 'running'; then
76          service glusterfs-server start || xecho 'Unable to start GlusterFS Server' 1
77      fi
78  }
79
80  hook_stop()
81  {
82      techo 'Storage - stop'
83
84      if service glusterfs-server status | grep -q 'running'; then
85          service glusterfs-server stop || xecho 'Unable to stop GlusterFS Server' 1
86      fi
87  }
88
89  # HOOKS : Provides Storage =====
90
91  hook_storage_relation_joined()
92  {
93      techo 'Storage - storage relation joined'
94
95      # Send filesystem type, mount point & options

```

```

96     relation-set fstype='glusterfs' mountpoint="$MEDIAS_VOLUME" options=''
97 }
```

6.13 OSCIED - WebUI Source Code

Note: 000-default, apache2.conf, copyright, revision and webui-db.sql are skipped as they are not so relevant here.

I gracefully thanks my brother for his expertise in web development, he helped my by starting the web part of the source code. He also followed my work and fixed the bugs I added by coding features to this user interface ;-)

6.13.1 AUTHORS

David Fischer
Michaël Fischer

6.13.2 config.yaml

```

1  options:
2      verbose:
3          type: string
4          default: "false"
5          description: Set verbose logging. Can be set to true or false
6      max_upload_size:
7          type: int
8          default: 4294967296
9          description: Maximum size for file uploads. Default value means 4GB
10     max_execution_time:
11         type: int
12         default: 180
13         description: Maximum time for php scripts. Default value means 3 minutes
14     max_input_time:
15         type: int
16         default: 600
17         description: Maximum time for http post. Default value means 10 minutes
18     mysql_my_password:
19         type: string
20         default: "root"
21         description: Password for phpmyadmin. Please change it in production
22     mysql_root_password:
23         type: string
24         default: "root"
25         description: Password of MySQL root user. Please change it in production
26     mysql_user_password:
27         type: string
28         default: "webui"
29         description: Password of MySQL webui user. Please change it in production
30     api_url:
31         type: string
32         default: ""
33         description: Orchestrator REST API address. This option override and disable api relation.
34     storage_ip:
35         type: string
36         default: ""
37         description: Shared storage hostname / IP address (see interface mount of nfs charm). The 'storage' option
38     storage_nat_ip:
39         type: string
40         default: ""
41         description: Shared storage reacheable (public) address e.g. when storage is behind a NAT/FW.
42     storage_fstype:
43         type: string
44         default: ""
45         description: Shared storage filesystem type (e.g. nfs)
46     storage_mountpoint:
```

```

47     type: string
48     default: ""
49     description: Shared storage mountpoint (e.g. for nfs - /srv/data)
50   storage_options:
51     type: string
52     default: ""
53     description: Shared storage options (e.g. for nfs - rw, sync,no_subtree_check)

```

6.13.3 metadata.yaml

```

1 ensemble: oscied
2 name: oscied-webui
3 summary: "EBU's OSCIED Web UI"
4 maintainer: OSCIED Main Developper <david.fischer.ch@gmail.com>
5 description: |
6   Installs the EBU's OSCIED web user interface.
7   Can be scaled to multiple instances.
8 provides:
9   website:
10  interface: http
11 requires:
12   api:
13     interface: orchestra
14   storage:
15     interface: mount

```

6.13.4 hooks_lib/common.sh.lu-dep

```

1 set -o nounset # will exit if an uninitialized variable is used
2
3 # Constants =====
4
5 ECHO='juju-log' # Used by logicielsUbuntuUtils
6
7 # Charms paths
8 BASE_PATH=$(pwd)
9
10 # Charms files
11 WEBUI_DB_FILE="$BASE_PATH/webui-db.sql"
12 SITE_TEMPL_FILE="$BASE_PATH/000-default"
13
14 # Shared storage paths
15 STORAGE_ROOT_PATH='/mnt/storage'
16 STORAGE_TEMP_PATH="$STORAGE_ROOT_PATH/tmp"
17 STORAGE_MEDIAS_PATH="$STORAGE_ROOT_PATH/medias"
18 STORAGE_UPLOADS_PATH="$STORAGE_ROOT_PATH/uploads"
19
20 # MySQL configuration files & paths
21 MYSQL_CONFIG_FILE='/etc/mysql/my.cnf'
22 MYSQL_TEMP_PATH='/var/lib/mysql/tmp'
23
24 # Web user interface paths
25 WWW_ROOT_PATH='/var/www'
26 WWW_MEDIAS_PATH="$WWW_ROOT_PATH/medias"
27 WWW_UPLOADS_PATH="$WWW_ROOT_PATH/uploads"
28
29 # Web user interface configuration files
30 SITES_ENABLED_PATH='/etc/apache2/sites-enabled'
31 GENERAL_CONFIG_FILE="$WWW_ROOT_PATH/application/config/config.php"
32 DATABASE_CONFIG_FILE="$WWW_ROOT_PATH/application/config/database.php"
33 HTACCESS_FILE="$WWW_ROOT_PATH/.htaccess"
34 ORCHESTRA_FLAG="$WWW_ROOT_PATH/orchestra_relation_ok"
35
36 # Configuration =====
37
38 if [ "$(config-get verbose)" = 'true' ] ; then
39   VERBOSE=0      # true
40   set -o xtrace # for verbose logging to juju debug-log
41 else

```

```

42     VERBOSE=1 # false
43 fi
44
45 PROXY_IPS=$(cat proxy_ips 2>/dev/null)
46 MAX_UPLOAD_SIZE=$(config-get max_upload_size)
47 MAX_EXECUTION_TIME=$(config-get max_execution_time)
48 MAX_INPUT_TIME=$(config-get max_input_time)
49
50 API_URL=$(config-get api_url)
51
52 STORAGE_IP=$(config-get storage_ip)
53 STORAGE_NAT_IP=$(config-get storage_nat_ip)
54 STORAGE_FSTYPE=$(config-get storage_fstype)
55 STORAGE_MOUNTPOINT=$(config-get storage_mountpoint)
56 STORAGE_OPTIONS=$(config-get storage_options)
57
58 MYSQL_MY_PASS=$(config-get mysql_my_password)
59 MYSQL_ROOT_PASS=$(config-get mysql_root_password)
60 MYSQL_USER_PASS=$(config-get mysql_user_password)
61
62 # Utilities =====
63
64 api_config_is_enabled()
65 {
66     [ "$API_URL" ]
67 }
68
69 api_hook_bypass()
70 {
71     if api_config_is_enabled; then
72         xecho 'Orchestrator is set in config, api relation is disabled' 1
73     fi
74 }
75
76 api_register()
77 {
78     # Overrides api parameters with charm configuration
79     if api_config_is_enabled; then # if api options are set
80         api_url=$API_URL
81     # Or uses api parameters from charm api relation
82     elif [ $# -eq 1 ]; then # if function parameters are set
83         api_url=$1
84     elif [ $# -eq 0 ]; then
85         return
86     else
87         xecho "Usage: $(basename $0).api_register api_url"
88     fi
89
90     pecho 'Configure Web UI : Register the Orchestrator'
91     setSettingPHP $GENERAL_CONFIG_FILE 'config' 'orchestra_api_url' "$api_url" || xecho 'Config'
92     touch "$ORCHESTRA_FLAG" || xecho 'Unable to create flag'
93 }
94
95 api_unregister()
96 {
97     pecho 'Configure Web UI : Unregister the Orchestrator'
98     setSettingPHP $GENERAL_CONFIG_FILE 'config' 'orchestra_api_url' '' || xecho 'Config'
99     rm -f "$ORCHESTRA_FLAG" 2>/dev/null
100 }
101
102 update_proxies()
103 {
104     if [ $# -ne 2 ]; then
105         xecho "Usage: $(basename $0).update_proxies action ip"
106     fi
107     action=$1
108     ip=$2
109
110     PROXY_IPS=$(cat proxy_ips 2>/dev/null)
111     case "$action" in
112     'add' )
113         if ! echo $PROXY_IPS | grep -q "$ip"; then
114             [ "$PROXY_IPS" ] && PROXY_IPS="$PROXY_IPS,"

```

```

115     PROXY_IPS="$PROXY_IPS$ip"
116     echo $PROXY_IPS > proxy_ips
117     setSettingPHP $GENERAL_CONFIG_FILE 'config' 'proxy_ips' "$PROXY_IPS" || return $false
118   fi ;;
119   'remove' )
120   if echo $PROXY_IPS | grep -q "$ip"; then
121     sed -i "s<$ip,<>g;s<,$ip<>g;s<$ip<>g" proxy_ips
122     PROXY_IPS=$(cat proxy_ips)
123     setSettingPHP $GENERAL_CONFIG_FILE 'config' 'proxy_ips' "$PROXY_IPS" || return $false
124   fi ;;
125   'cleanup' )
126   if "$PROXY_IPS"; then
127     PROXY_IPS=''
128     echo '' > proxy_ips
129     setSettingPHP $GENERAL_CONFIG_FILE 'config' 'proxy_ips' "$PROXY_IPS" || return $false
130   fi ;;
131   * ) xecho "Unknown action : $action" ;;
132 esac
133
134 return $true
135 }
136
137 storage_config_is_enabled()
138 {
139   [ "$STORAGE_IP" -a "$STORAGE_FSTYPE" -a "$STORAGE_MOUNTPOINT" ]
140 }
141
142 storage_is_mounted()
143 {
144   mount | grep -q "$STORAGE_ROOT_PATH"
145 }
146
147 storage_remount()
148 {
149   # Overrides storage parameters with charm configuration
150   if storage_config_is_enabled; then # if storage options are set
151     ip=$STORAGE_IP
152     nat_ip=$STORAGE_NAT_IP
153     fstype=$STORAGE_FSTYPE
154     mountpoint=$STORAGE_MOUNTPOINT
155     options=$STORAGE_OPTIONS
156   # Or uses storage parameters from charm storage relation
157   elif [ $# -eq 4 ]; then # if function parameters are set
158     ip=$1
159     nat_ip=''
160     fstype=$2
161     mountpoint=$3
162     options=$4
163   elif [ $# -eq 0 ]; then
164     return
165   else
166     xecho "Usage: $(basename $0).storage_remount ip fstype mountpoint options"
167   fi
168
169   if [ "$nat_ip" ]; then
170     pecho "Update hosts file to map storage internal address $ip to $nat_ip"
171     if grep -q "$ip" /etc/hosts; then
172       sed -i "s<$nat_ip .*<$nat_ip $ip<" /etc/hosts
173     else
174       echo "$nat_ip $ip" >> /etc/hosts
175     fi
176   else
177     nat_ip=$ip
178   fi
179
180   storage_umount
181
182   r=$STORAGE_ROOT_PATH
183   pecho "Mount shared storage [$nat_ip] $ip:$mountpoint type $fstype options '$options' -> $r"
184   if [ ! -d "$STORAGE_ROOT_PATH" ]; then
185     mkdir "$STORAGE_ROOT_PATH" || xecho "Unable to create shared storage path $STORAGE_ROOT_PATH" 1
186   fi
187

```

```

188 # FIXME try 5 times, a better way to handle failure
189 for i in $(seq 1 5)
190 do
191     if storage_is_mounted; then
192         break
193     else
194         if [ "$options" ]
195         then mount -t "$fstype" -o "$options" "$nat_ip:$mountpoint" "$STORAGE_ROOT_PATH"
196         else mount -t "$fstype" "$nat_ip:$mountpoint" "$STORAGE_ROOT_PATH"
197         fi
198     fi
199     sleep 5
200 done
201
202 if storage_is_mounted; then
203     storage_migrate_path 'medias'    "$STORAGE_MEDIAS_PATH"    "$WWW_MEDIAS_PATH"    'root'      755 644
204     storage_migrate_path 'uploads'   "$STORAGE_UPLOADS_PATH"   "$WWW_UPLOADS_PATH"   'www-data'  755 644
205     # FIXME update /etc/fstab (?)
206     pecho 'Configure Web UI : Register shared storage'
207     # FIXME this is a little bit cheating with paths ;)
208     storage_uri="$fstype://$ip/$mountpoint"
209     uploads_uri="$storage_uri/uploads/"
210     medias_uri="$storage_uri/medias/"
211     setSettingPHP $GENERAL_CONFIG_FILE 'config' 'uploads_uri' "$uploads_uri" || xecho 'Config' 2
212     setSettingPHP $GENERAL_CONFIG_FILE 'config' 'medias_uri'  "$medias_uri"  || xecho 'Config' 3
213 else
214     xecho 'Unable to mount shared storage' 4
215 fi
216 }
217
218 storage_umount()
219 {
220     pecho 'Configure Web UI : Unregister shared storage'
221     setSettingPHP $GENERAL_CONFIG_FILE 'config' 'uploads_uri' '' || xecho 'Config' 1
222     setSettingPHP $GENERAL_CONFIG_FILE 'config' 'medias_uri'  '' || xecho 'Config' 2
223
224 if storage_is_mounted; then
225     # FIXME update /etc/fstab (?)
226     pecho 'Unmount shared storage (is actually mounted)'
227     umount "$STORAGE_ROOT_PATH" || xecho 'Unable to unmount shared storage' 3
228 else
229     recho 'Shared storage already unmounted'
230 fi
231 }
232
233 storage_hook_bypass()
234 {
235 if storage_config_is_enabled; then
236     xecho 'Shared storage is set in config, storage relation is disabled' 1
237 fi
238 }
239
240 # Migrate a local Web UI path to shared storage only if necessary -----
241 storage_migrate_path()
242 {
243     if [ $# -ne 6 ]; then
244         xecho "Usage: $(basename $0).storage_migrate_path name storage local owner dmod fmod"
245     fi
246
247     name=$1
248     storage=$2
249     local=$3
250     owner=$4
251     dmod=$5
252     fmod=$6
253
254     if [ ! -d "$storage" ]; then
255         pecho "Create $name path in storage"
256         mkdir -p "$storage" || xecho "Unable to create $name path" 1
257     else
258         recho "Storage $name path already created"
259     fi
260 }
```

```

261 if [ -d "$local" ]; then
262     mecho "Migrating files from Web UI $name path to $name path in storage ..."
263     rsync -a "$local/" "$storage/" || xecho "Unable to migrate $name files" 2
264     rm -rf "$local"
265 fi
266
267 if [ ! -h "$local" ]; then
268     pecho "Link Web UI $name path to $name path in storage"
269     ln -s "$storage" "$local" || xecho "Unable to create $name link" 3
270 fi
271
272 pecho "Ensure POSIX rights (owner=$owner:$owner mod=(d=$dmod,f=$fmod)) of $name path in storage"
273 chown "$owner:$owner" "$storage" -R || xecho "Unable to chown $storage" 4
274 find "$storage" -type d -exec chmod "$dmod" "$storage" \;
275 find "$storage" -type f -exec chmod "$fmod" "$storage" \;
276 }
277
278 # HOOKS : Charm Setup =====
279
280 hook_install()
281 {
282     techo 'Web UI - install'
283
284     # Fix memtest86+ : https://bugs.launchpad.net/ubuntu/+source/grub2/+bug/1069856
285     #eval $purge grub-pc grub-common
286     #eval $install grub-common grub-pc
287
288     eval $update
289     eval $upgrade
290
291     pecho 'Install and configure Network Time Protocol'
292     eval $install ntp || xecho 'Unable to install ntp' 1
293     eval $service ntp restart || xecho 'Unable to restart ntp service' 2
294
295     techo 'Install and configure MySQL'
296     sql='mysql-server mysql-server' # Tip : http://ubuntuforums.org/showthread.php?t=981801
297     echo "$sql/root_password select $MYSQL_ROOT_PASS" | debconf-set-selections
298     echo "$sql/root_password_again select $MYSQL_ROOT_PASS" | debconf-set-selections
299     mkdir /etc/mysql 2>/dev/null
300     eval $install mysql-server glusterfs-client nfs-common || xecho 'Unable to install packages' 3
301
302     # Now MySQL will listen to incoming request of any source
303     #sed -i 's/127.0.0.1/0.0.0.0/g' /etc/mysql/my.cnf
304
305     root=$MYSQL_ROOT_PASS
306
307     # Fix ticket #57 : Keystone + MySQL = problems
308     mysql -uroot -p"$root" -e "DROP USER ''@'localhost'; DROP USER ''@'$hostname');"
309     mysql -uroot -p"$root" -e "GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' WITH GRANT OPTION;"
310     mysql -uroot -p"$root" -e "SET PASSWORD FOR 'root'@'%' = PASSWORD('$MYSQL_ROOT_PASS');"
311
312     service mysql restart || xecho 'Unable to restart mysql' 4
313
314     pecho 'Import Web User Interface database'
315     mysql -u root -p"$MYSQL_ROOT_PASS" < "$WEBUI_DB_FILE" || xecho 'Unable to import database' 5
316
317     pecho 'Create Web User Interface user'
318     user='webui'
319     pass="$MYSQL_USER_PASS"
320     mysql -u root -p"$MYSQL_ROOT_PASS" \
321         -e "GRANT ALL ON webui.* TO '$user'@'%' IDENTIFIED BY '$pass';" || \
322     xecho 'Unable to create user' 6
323
324     pecho 'Install and configure Apache2 + PHP'
325     my='phpmyadmin phpmyadmin' # Tip : http://gercogandia.blogspot.ch/2012/11/automatic-unattended-install-of.ht
326     echo "$my/app-password password $MYSQL_MY_PASS" | debconf-set-selections
327     echo "$my/app-password-confirm password $MYSQL_MY_PASS" | debconf-set-selections
328     echo "$my/mysql/admin-pass password $MYSQL_ROOT_PASS" | debconf-set-selections
329     echo "$my/mysql/app-pass password $MYSQL_ROOT_PASS" | debconf-set-selections
330     echo "$my/reconfigure-webserver multiselect apache2" | debconf-set-selections
331     eval $install apache2 php5 php5-cli php5-curl php5-gd php5-mysql libapache2-mod-auth-mysql \
332         phpmyadmin || xecho 'Unable to install packages' 7
333

```

```

334 a2enmod rewrite
335
336 pecho 'Copy and configure Web User Interface'
337 cp -f "$SITE_TEMPL_FILE" "$SITES_ENABLED_PATH"
338 rsync -rtvh --progress --delete --exclude=.svn "www/" "/var/www/"
339 key=$(randpass 32 $false $false $false)
340 C='Config'
341 setSettingBASH $MYSQL_CONFIG_FILE $true 'tmpdir' "$MYSQL_TEMP_PATH" || xecho $c 8
342 setSettingPHP $GENERAL_CONFIG_FILE 'config' 'encryption_key' "$key" || xecho $c 9
343 setSettingPHP $GENERAL_CONFIG_FILE 'config' 'proxy_ips' "$PROXY_IPS" || xecho $c 10
344 setSettingPHP $DATABASE_CONFIG_FILE 'db' 'default' 'hostname' 'localhost' || xecho $c 11
345 setSettingPHP $DATABASE_CONFIG_FILE 'db' 'default' 'username' 'webui' || xecho $c 12
346 setSettingPHP $DATABASE_CONFIG_FILE 'db' 'default' 'password' "$MYSQL_USER_PASS" || xecho $c 13
347 setSettingPHP $DATABASE_CONFIG_FILE 'db' 'default' 'database' 'webui' || xecho $c 14
348 mkdir -p "$MYSQL_TEMP_PATH"
349 chown mysql:mysql "$MYSQL_TEMP_PATH"
350 chown www-data:www-data "$WWW_ROOT_PATH" -R
351
352 # config php, mettre short opentags à "on"
353 # lire les logs, problème MY_ my_ nom fichier
354
355 pecho 'Expose Apache 2 service'
356 open-port 80/tcp
357
358 # FIXME this call is not necessary, but config-changed create an infinite loop, so WE call it
359 hook_config_changed
360 }
361
362 hook_uninstall()
363 {
364     techo 'Web UI - uninstall'
365
366     hook_stop
367     eval $purge apache2 php5 php5-cli php5-gd php5-mysql libapache2-mod-auth-mysql phpmyadmin \
368         apache2.2-common mysql-client-5.5 mysql-client-core-5.5 mysql-common mysql-server \
369         mysql-server-5.1 mysql-server-5.5 mysql-server-core-5.5 glusterfs-client nfs-common
370     eval $autoremove
371     rm -rf /etc/apache2/ /var/www/ /var/log/apache2/ /etc/mysql/ /var/lib/mysql/ /var/log/mysql/
372     mkdir /var/www/
373 }
374
375 hook_config_changed()
376 {
377     techo 'Web UI - config changed'
378
379     hook_stop
380     pecho 'Configure Web UI : Set limits (upload size, execution time, ...)'
381     C='Config'
382     upload='upload_max_filesize'
383     execution='max_execution_time'
384     setSettingPHP $GENERAL_CONFIG_FILE 'config' 'max_upload_size' "$MAX_UPLOAD_SIZE" || xecho $c 1
385     setSettingPHP $GENERAL_CONFIG_FILE 'config' 'uploads_path' "$WWW_UPLOADS_PATH/" || xecho $c 2
386     setSettingPHP $GENERAL_CONFIG_FILE 'config' 'medias_path' "$WWW_MEDIAS_PATH/" || xecho $c 3
387     setSettingHTA $HTACCESS_FILE $true "php_value $upload" "$MAX_UPLOAD_SIZE" || xecho $c 4
388     setSettingHTA $HTACCESS_FILE $true "php_value post_max_size" "$MAX_UPLOAD_SIZE" || xecho $c 5
389     setSettingHTA $HTACCESS_FILE $true "php_value $execution" "$MAX_EXECUTION_TIME" || xecho $c 6
390     setSettingHTA $HTACCESS_FILE $true "php_value max_input_time" "$MAX_INPUT_TIME" || xecho $c 7
391     storage_remount
392     api_register
393     hook_start
394     # FIXME infinite loop is used as config-changed hook !
395 }
396
397 # HOOKS : Charm Service =====
398
399 hook_start()
400 {
401     techo 'Web UI - start'
402
403     if ! storage_is_mounted; then
404         recho 'WARNING Do not start Web UI : No shared storage'
405     elif [ ! -f "$ORCHESTRA_FLAG" ]; then
406         recho 'WARNING Do not start Web UI : No Orchestrator API'

```

```

407     else
408         if ! service mysql status | grep -q 'running'; then
409             service mysql start || xecho 'Unable to start MySQL' 1
410         fi
411         service apache2 start || xecho 'Unable to start Apache 2' 2
412     fi
413 }
414
415 hook_stop()
416 {
417     techo 'Web UI - stop'
418
419     service apache2 stop || xecho 'Unable to stop Apache 2' 1
420     if service mysql status | grep -q 'running'; then
421         service mysql stop || xecho 'Unable to stop MySQL' 2
422     fi
423 }
424
425 # HOOKS : Requires API =====
426
427 hook_api_relation_joined()
428 {
429     techo 'Web UI - api relation joined'
430     api_hook_bypass
431 }
432
433 hook_api_relation_changed()
434 {
435     techo 'Web UI - api relation changed'
436     api_hook_bypass
437
438     # Get configuration from the relation
439     ip=$(relation-get private-address)
440     api_url=$(relation-get api_url)
441
442     mecho "Orchestrator IP is $ip, API URL is $api_url"
443     if [ ! "$ip" -o ! "$api_url" ]; then
444         recho 'Waiting for complete setup'
445         exit 0
446     fi
447
448     hook_stop
449     api_register "$api_url"
450     hook_start
451 }
452
453 hook_api_relation_broken()
454 {
455     techo 'Web UI - api relation broken'
456     api_hook_bypass
457
458     hook_stop
459     api_unregister
460 }
461
462 # HOOKS : Requires Storage =====
463
464 hook_storage_relation_joined()
465 {
466     techo 'Web UI - storage relation joined'
467     storage_hook_bypass
468 }
469
470 hook_storage_relation_changed()
471 {
472     techo 'Web UI - storage relation changed'
473     storage_hook_bypass
474
475     # Get configuration from the relation
476     ip=$(relation-get private-address)
477     fstype=$(relation-get fstype)
478     mountpoint=$(relation-get mountpoint)
479     options=$(relation-get options)

```

```

480
481     mecho "Storage IP is $ip, fstype: $fstype, mountpoint: $mountpoint, options: $options"
482     if [ ! "$ip" -o ! "$fstype" -o ! "$mountpoint" ]; then
483         recho 'Waiting for complete setup'
484         exit 0
485     fi
486
487     hook_stop
488     storage_remount "$ip" "$fstype" "$mountpoint" "$options"
489     hook_start
490 }
491
492 hook_storage_relation_broken()
493 {
494     techo 'Web UI - storage relation broken'
495     storage_hook_bypass
496
497     hook_stop
498     storage_umount
499 }
500
501 # HOOKS : Provides Website =====
502
503 hook_website_relation_joined()
504 {
505     techo 'Web UI - website relation joined'
506
507     # Send port & hostname
508     relation-set port=80 hostname=$(hostname -f)
509 }
510
511 hook_website_relation_changed()
512 {
513     techo 'Web UI - website relation changed'
514
515     # Get configuration from the relation
516     proxy_ip=$(relation-get private-address)
517
518     mecho "Proxy IP is $proxy_ip"
519     if [ ! "$proxy_ip" ]; then
520         recho 'Waiting for complete setup'
521         exit 0
522     fi
523
524     hook_stop
525     pecho "Configure Web UI : Add $proxy_ip to allowed proxy IPs"
526     update_proxies add "$proxy_ip" || xecho 'Unable to add proxy'
527     hook_start
528 }
529
530 hook_website_relation_departed()
531 {
532     techo 'Web UI - website relation departed'
533
534     # Get configuration from the relation
535     proxy_ip=$(relation-get private-address)
536
537     mecho "Proxy IP is $proxy_ip"
538     if [ ! "$proxy_ip" ]; then
539         recho 'Waiting for complete setup'
540         exit 0
541     fi
542
543     hook_stop
544     pecho "Configure Web UI : Remove $proxy_ip from allowed proxy IPs"
545     update_proxies remove "$proxy_ip" || xecho 'Unable to remove proxy'
546     hook_start
547 }
548
549 hook_website_relation_broken()
550 {
551     techo 'Web UI - website relation broken'
552 }
```

```

553 # Get configuration from the relation
554 proxy_ip=$(relation-get private-address)
555
556 mecho "Proxy IP is $proxy_ip"
557 if [ ! "$proxy_ip" ]; then
558     recho 'Waiting for complete setup'
559     exit 0
560 fi
561
562 hook_stop
563 pecho "Configure Web UI : Remove $proxy_ip from allowed proxy IPs"
564 update_proxies remove "$proxy_ip" || xecho 'Unable to remove proxy'
565 # FIXME does relation broken means that no more proxies are linked to us ? if yes :
566 #pecho 'Configure Web UI : Cleanup allowed proxy IPs'
567 #update_proxies cleanup || xecho 'Unable to cleanup proxies'
568 hook_start
569 }

```

6.13.5 www/.../config

config.php (extract)

```

1  /*
2  | -----
3  | Reverse Proxy IPs
4  | -----
5  |
6  | If your server is behind a reverse proxy, you must whitelist the proxy IP
7  | addresses from which CodeIgniter should trust the HTTP_X_FORWARDED_FOR
8  | header in order to properly identify the visitor's IP address.
9  | Comma-delimited, e.g. '10.0.1.200,10.0.1.201'
10 |
11 */
12 $config['proxy_ips'] = '';
13
14
15 /**
16 | -----
17 | CUSTOM Variables
18 | -----
19 |
20
21 /**
22 | -----
23 | Max video file size authorized
24 | -----
25 |
26 */
27 $config['max_upload_size'] = '4294967296';
28
29 /**
30 | -----
31 | Orchestrator RESTful API URL
32 | -----
33 |
34 */
35 $config['orchestra_api_url'] = '';
36
37 /**
38 | -----
39 | Shared storage URI + local mount path for both uploads and stored medias
40 | -----
41 |
42 */
43 $config['uploads_uri'] = '';
44 $config['uploads_path'] = '';
45 $config['medias_uri'] = '';
46 $config['medias_path'] = '';
47
48 /**
49 | -----

```

```

50 | Refresh rates for each view (eg. Transform
51 | -----
52 |
53 */
54 $config['users_refresh'] = '10';
55 $config['medias_refresh'] = '5';
56 $config['profiles_refresh'] = '10';
57 $config['transform_refresh'] = '2';
58 $config['publisher_refresh'] = '2';
59
60 /* End of file config.php */
61 /* Location: ./application/config/config.php */

```

media.php

```

1 <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2
3 class Media extends MY_Controller
4 {
5
6     protected $page_name = 'media';
7
8     public function __construct() {
9         parent::__construct();
10    }
11
12 /**
13 * TODO : comments
14 */
15 public function index() {
16     $this->load->helper('number');
17     $this->load->spark('restclient/2.1.0');
18     $this->load->library('rest');
19     $this->rest->initialize(
20         array(
21             'server' => $this->config->item('orchestra_api_url'), 'http_auth' => 'basic',
22             'http_user' => $this->user->mail(), 'http_pass' => $this->user->secret()
23         )
24     );
25     $response = $this->rest->get('media');
26     if ($response->status != 200) {
27         print_r($response->value);
28         exit;
29     }
30     $data['medias'] = $response->value;
31     $this->add_content('page_title', 'OSCIED - Media');
32     $this->add_view('main', 'media/show', $data);
33
34     $header_data['page'] = 'media';
35     $this->render($header_data);
36 }
37
38 /**
39 * TODO : comments
40 */
41 public function refresh() {
42     $this->load->helper('number');
43     $this->load->spark('restclient/2.1.0');
44     $this->load->library('rest');
45     $this->rest->initialize(
46         array(
47             'server' => $this->config->item('orchestra_api_url'), 'http_auth' => 'basic',
48             'http_user' => $this->user->mail(), 'http_pass' => $this->user->secret()
49         )
50     );
51     $response = $this->rest->get('media');
52     if ($response->status != 200) {
53         print_r($response->value);
54         exit;
55     }
56     $data['medias'] = $response->value;

```

```

57     $this->load->view('media/show_medias', $data);
58 }
59
60 /**
61 * TODO : comments
62 */
63 public function delete($id) {
64     $this->load->spark('restclient/2.1.0');
65     $this->load->library('rest');
66     $this->rest->initialize(
67         array(
68             'server' => $this->config->item('orchestra_api_url'), 'http_auth' => 'basic',
69             'http_user' => $this->user->mail(), 'http_pass' => $this->user->secret()
70         )
71     );
72     $response = $this->rest->delete('media/id/' . $id);
73     // Set error or information message
74     if ($response->status == 200) {
75         $this->session->set_flashdata('infos', $response->value);
76         echo json_encode(array('redirect' => 'media'));
77     }
78     else {
79         $this->session->set_flashdata('errors', $response->value->description);
80         echo json_encode(array('redirect' => 'media'));
81     }
82 }
83
84 /**
85 * TODO : comments
86 */
87 public function force_download($id) {
88     $this->load->helper('download');
89     $this->load->spark('restclient/2.1.0');
90     $this->load->library('rest');
91     $this->rest->initialize(
92         array(
93             'server' => $this->config->item('orchestra_api_url'), 'http_auth' => 'basic',
94             'http_user' => $this->user->mail(), 'http_pass' => $this->user->secret()
95         )
96     );
97     $response = $this->rest->get('media/id/' . $id);
98     $name = isset($response->value->virtual_filename)?$response->value->virtual_filename:'Untitled';
99     $path = str_replace(
100         $this->config->item('medias_uri'),
101         $this->config->item('medias_path'),
102         $response->value->uri
103     );
104     force_download($name, $path);
105 }
106
107 /**
108 * TODO : comments
109 */
110 public function add_media() {
111     $this->load->model('tmp_files_model');
112     $this->load->library('form_validation');
113
114     $form_id = $this->input->post('form_id');
115
116     $this->form_validation->set_rules('title', 'Title', 'required');
117     $this->form_validation->set_rules('virtual_filename', 'Virtual filename', 'required');
118     $this->form_validation->set_rules('form_id', 'File', 'callback__file_check');
119
120     if ($this->form_validation->run() === FALSE) {
121         $errors = validation_errors();
122         echo json_encode(array('errors' => $errors));
123     }
124     else {
125         // Get the file infos
126         $old_dir = 'tmp/uploads/' . $form_id . '/';
127         $this->load->model('tmp_files_model');
128         // Get the unique file
129         $files = $this->tmp_files_model->get_files($form_id);

```

```

130     $file = $files[0];
131     // Add the media
132     $new_dir = $this->config->item('uploads_path');
133     $file['dir'] = $new_dir;
134     rename($old_dir.$file['name'], $new_dir.$file['name']);
135     // Remove files on the "tmp_files" table
136     $this->tmp_files_model->delete_files($form_id);
137
138     $this->load->spark('restclient/2.1.0');
139     $this->load->library('rest');
140     $this->rest->initialize(
141         array(
142             'server' => $this->config->item('orchestra_api_url'), 'http_auth' => 'basic',
143             'http_user' => $this->user->mail(), 'http_pass' => $this->user->secret()
144         )
145     );
146     $params = json_encode(
147         array(
148             'uri' => str_replace(
149                 $this->config->item('uploads_path'), $this->config->item('uploads_uri'), $new_dir
150             ).$file['name'],
151             'virtual_filename' => $this->input->post('virtual_filename'), // $file['name'],
152             'metadata' => array('title' => $this->input->post('title'))
153         )
154     );
155     $response = $this->rest->post('media', $params, 'json');
156     // Remove the temporary dir (tmp/uploads/form_id/)
157     $this->_delete_directory($old_dir);
158     // Set the flash message
159     $this->session->set_flashdata(
160         'infos', 'The media "'.$this->input->post('title').'" has been added.'
161     );
162     echo json_encode(array('redirect' => site_url('media')));
163 }
164 }
165
166 /**
167 * Validates the presence of a file
168 */
169 public function _file_check($form_id) {
170     if (!$this->tmp_files_model->a_file_exist($form_id)) {
171         $this->form_validation->set_message('_file_check', "The %s field is required");
172         return FALSE;
173     }
174     return TRUE;
175 }
176
177 // TODO : EXPORT IT INTO A HELPER
178 protected function _delete_directory($dir) {
179     if (!file_exists($dir)) return true;
180     if (!is_dir($dir) || is_link($dir)) return unlink($dir);
181     foreach (scandir($dir) as $item) {
182         if ($item == '.' || $item == '..') continue;
183         if (!$this->_delete_directory($dir.'/'.$item)) {
184             chmod($dir.'/'.$item, 0777);
185             if (!$this->_delete_directory($dir.'/'.$item)) return false;
186         };
187     }
188     return rmdir($dir);
189 }
190
191 /**
192 * TODO : comments
193 */
194 public function get_files($id) {
195     $this->load->model('files_model');
196     $dir = 'uploads/medias/'.$id '/';
197     $files = $this->files_model->get_files_in_dir($dir);
198     $info = array();
199     foreach ($files as $file) {
200         $info[] = $this->get_file_object($dir, $file);
201     }
202     header('Content-type: application/json');

```

```

203     echo json_encode($info);
204 }
205
206 /**
207 * TODO : comments
208 */
209 protected function get_file_object($dir, $file, $tmp=false) {
210     $file_path = $dir.$file->name;
211     $file_infos = new stdClass();
212     $file_infos->id = $file->id;
213     $file_infos->title = $file->title;
214     $file_infos->name = $file->name;
215     $file_infos->size = filesize($file_path);
216     // $file_infos->url = base_url() . $dir.rawurlencode($file->name);
217     $file_infos->tmp = $tmp;
218     return $file_infos;
219 }
220
221 /**
222 * TODO : EXPORT IT INTO A HELPER
223 */
224 public function delete_file($file_id, $tmp_file=false) {
225     // Is it a tmp file (/tmp/uploads/...) ?
226     if ($tmp_file) {
227         $this->load->model('tmp_files_model');
228         $file = $this->tmp_files_model->get_file($file_id);
229         if ($file !== NULL) {
230             // Remove the file in table and physically
231             $this->tmp_files_model->delete_file($file_id);
232             if (file_exists($file->dir.$file->name)) {
233                 unlink($file->dir.$file->name);
234             }
235             return true;
236         }
237     }
238     else {
239         $this->load->model('files_model');
240         $file = $this->files_model->get_file($file_id);
241         if ($file !== NULL) {
242             // Remove the file in table and physically
243             $this->files_model->delete_file($file_id);
244             if (file_exists($file->dir.$file->name)) {
245                 unlink($file->dir.$file->name);
246             }
247             return true;
248         }
249     }
250     return false;
251 }
252
253 }
254
255 /* End of file media.php */
256 /* Location: ./application/controllers/media.php */

```

6.13.6 www/.../controllers

misc.php

```

1 <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2
3 class Misc extends MY_Controller
4 {
5     public function index() {
6         $this->add_content('page_title', 'OSCIED - Home');
7         $this->add_view('main', 'homepage');
8
9         $header_data['page'] = 'home';
10        $this->render($header_data);
11    }
12}

```

```

13  public function contact() {
14      $this->add_content('page_title', 'OSCIED - Contact Us');
15      $this->add_view('main', 'contact');
16
17      $header_data['page'] = 'contact';
18      $this->render($header_data);
19  }
20
21 }
22
23 /* End of file misc.php */
24 /* Location: ./application/controllers/misc.php */

```

profile.php

```

1  <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2
3 class Profile extends MY_Controller
4 {
5
6     protected $page_name = 'profile';
7
8     public function __construct() {
9         parent::__construct();
10    }
11
12 /**
13 * TODO : comments
14 */
15 public function index() {
16     $this->load->helper('number');
17     $this->load->spark('restclient/2.1.0');
18     $this->load->library('rest');
19     $this->rest->initialize(
20         array(
21             'server' => $this->config->item('orchestra_api_url'), 'http_auth' => 'basic',
22             'http_user' => $this->user->mail(), 'http_pass' => $this->user->secret()
23         )
24     );
25     $response = $this->rest->get('transform/profile');
26     if ($response->status != 200) {
27         print_r($response->value);
28         exit;
29     }
30     $data['profiles'] = $response->value;
31
32     $this->add_content('page_title', 'OSCIED - Transform Profiles');
33     $this->add_view('main', 'profile/show', $data);
34
35     $header_data['page'] = 'profile';
36     $this->render($header_data);
37 }
38
39 /**
40 * TODO : comments
41 */
42 public function refresh() {
43     $this->load->helper('number');
44     $this->load->spark('restclient/2.1.0');
45     $this->load->library('rest');
46     $this->rest->initialize(
47         array(
48             'server' => $this->config->item('orchestra_api_url'), 'http_auth' => 'basic',
49             'http_user' => $this->user->mail(), 'http_pass' => $this->user->secret()
50         )
51     );
52     $response = $this->rest->get('transform/profile');
53     if ($response->status != 200) {
54         print_r($response->value);
55         exit;
56     }
57     $data['profiles'] = $response->value;

```

```

58     $this->load->view('profile/show_profiles', $data);
59 }
60
61 /**
62 * TODO : comments
63 */
64 public function delete($id) {
65     $this->load->spark('restclient/2.1.0');
66     $this->load->library('rest');
67     $this->rest->initialize(
68         array(
69             'server' => $this->config->item('orchestra_api_url'), 'http_auth' => 'basic',
70             'http_user' => $this->user->mail(), 'http_pass' => $this->user->secret()
71         )
72     );
73     $response = $this->rest->delete('transform/profile/id/'. $id);
74     // Set error or information message
75     if ($response->status == 200) {
76         $this->session->set_flashdata('infos', $response->value);
77         echo json_encode(array('redirect' => 'profile'));
78     }
79     else {
80         $this->session->set_flashdata('errors', $response->value->description);
81         echo json_encode(array('redirect' => 'profile'));
82     }
83 }
84
85 /**
86 * TODO : comments
87 */
88 public function add_profile() {
89     $this->load->library('form_validation');
90
91     $form_id = $this->input->post('form_id');
92
93     $this->form_validation->set_rules('title', 'Title', 'required');
94     $this->form_validation->set_rules('description', 'Description', 'required');
95     $this->form_validation->set_rules('encoder_string', 'Encoder string', 'required');
96
97     if ($this->form_validation->run() === FALSE) {
98         $errors = validation_errors();
99         echo json_encode(array('errors' => $errors));
100    }
101    else {
102        $this->load->spark('restclient/2.1.0');
103        $this->load->library('rest');
104        $this->rest->initialize(
105            array(
106                'server' => $this->config->item('orchestra_api_url'), 'http_auth' => 'basic',
107                'http_user' => $this->user->mail(), 'http_pass' => $this->user->secret()
108            )
109        );
110        $params = json_encode(
111            array(
112                'title' => $this->input->post('title'),
113                'description' => $this->input->post('description'),
114                'encoder_string' => $this->input->post('encoder_string')
115            )
116        );
117        $response = $this->rest->post('transform/profile', $params, 'json');
118        if ($response->status == 200) {
119            // Set the flash message
120            $this->session->set_flashdata(
121                'infos', 'The transform profile "'.$this->input->post('title').'" has been added.'
122            );
123            echo json_encode(array('redirect' => site_url('profile')));
124        }
125        else {
126            echo json_encode(array('errors' => $response->value->description));
127        }
128    }
129 }
130

```

```

131     /**
132      * TODO : comments
133     */
134     /*public function edit_profile($id) {
135         $this->load->library('form_validation');
136
137         $this->form_validation->set_rules('title', 'Title', 'required');
138
139         if ($this->form_validation->run() === FALSE) {
140             $errors = validation_errors();
141             echo json_encode(array('errors' => $errors));
142         }
143         else {
144             $old_profile = $this->profiles_model->get_profile($id);
145             $this->profiles_model->update_title($id, $this->input->post('title'));
146             $this->session->set_flashdata(
147                 'infos', 'The profile "'.$old_profile->title.'" has been modified.');
148         }
149         echo json_encode(array('redirect' => site_url('profile')));
150     }
151 } */
152
153 }
154
155 /* End of file profile.php */
156 /* Location: ./application/controllers/profile.php */

```

publisher.php

```

1 <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2
3 class Publisher extends MY_Controller
4 {
5
6     protected $page_name = 'publisher';
7
8     public function __construct() {
9         parent::__construct();
10    }
11
12    /**
13     * TODO : comments
14     */
15    public function index() {
16        $this->load->helper('number');
17        $this->load->spark('restclient/2.1.0');
18        $this->load->library('rest');
19        $this->rest->initialize(
20            array(
21                'server' => $this->config->item('orchestra_api_url'), 'http_auth' => 'basic',
22                'http_user' => $this->user->mail(), 'http_pass' => $this->user->secret()
23            )
24        );
25        $response = $this->rest->get('publish/job');
26        if ($response->status != 200) {
27            print_r($response->value);
28            exit;
29        }
30        $data['jobs'] = $response->value;
31        // Get the medias for the dropdown
32        $response = $this->rest->get('media/HEAD');
33        if ($response->status != 200) {
34            print_r($response->value);
35            exit;
36        }
37        $data['medias'] = array();
38        foreach ($response->value as $media) {
39            $data['medias'][$media->_id] = $media->metadata->title.' - '.$media->virtual_filename;
40        }
41        // Get the queues for the dropdown
42        $response = $this->rest->get('publish/queue');
43        if ($response->status != 200) {

```

```

44     print_r($response->value);
45     exit;
46 }
47 $data['queues'] = array();
48 foreach ($response->value as $queue) {
49     $data['queues'][$queue] = $queue;
50 }
51
52     $this->add_content('page_title', 'OSCIED - Publisher Jobs');
53 $this->add_view('main', 'publisher/show', $data);
54
55 $header_data['page'] = 'publisher';
56 $this->render($header_data);
57 }
58
59 /**
60 * TODO : comments
61 */
62 public function refresh() {
63     $this->load->helper('number');
64     $this->load->spark('restclient/2.1.0');
65     $this->load->library('rest');
66     $this->rest->initialize(
67         array(
68             'server' => $this->config->item('orchestra_api_url'), 'http_auth' => 'basic',
69             'http_user' => $this->user->mail(), 'http_pass' => $this->user->secret()
70         )
71     );
72     $response = $this->rest->get('publish/job');
73     if ($response->status != 200) {
74         print_r($response->value);
75         exit;
76     }
77     $data['jobs'] = $response->value;
78     // Get the medias for the dropdown
79     $response = $this->rest->get('media/HEAD');
80     if ($response->status != 200) {
81         print_r($response->value);
82         exit;
83     }
84     $data['medias'] = array();
85     $this->load->view('publisher/show_jobs', $data);
86 }
87
88 /**
89 * TODO : comments
90 */
91 public function revoke($id) {
92     $this->load->spark('restclient/2.1.0');
93     $this->load->library('rest');
94     $this->rest->initialize(
95         array(
96             'server' => $this->config->item('orchestra_api_url'), 'http_auth' => 'basic',
97             'http_user' => $this->user->mail(), 'http_pass' => $this->user->secret()
98         )
99     );
100    $response = $this->rest->delete('publish/job/id/' . $id);
101    // Set error or information message
102    if ($response->status == 200) {
103        $this->session->set_flashdata('infos', $response->value);
104        echo json_encode(array('redirect' => 'publisher'));
105    }
106    else {
107        $this->session->set_flashdata('errors', $response->value->description);
108        echo json_encode(array('redirect' => 'publisher'));
109    }
110 }
111
112 /**
113 * TODO : comments
114 */
115 public function launch_publish() {
116     $this->load->helper('number');

```

```

117     $this->load->library('form_validation');
118
119     $form_id = $this->input->post('form_id');
120
121     $this->form_validation->set_rules('media_id', 'Media', 'required');
122     $this->form_validation->set_rules('queue', 'Queue', 'required');
123
124     if ($this->form_validation->run() === FALSE) {
125         $errors = validation_errors();
126         echo json_encode(array('errors' => $errors));
127     }
128     else {
129         $this->load->spark('restclient/2.1.0');
130         $this->load->library('rest');
131         $this->rest->initialize(
132             array(
133                 'server' => $this->config->item('orchestra_api_url'), 'http_auth' => 'basic',
134                 'http_user' => $this->user->mail(), 'http_pass' => $this->user->secret()
135             )
136         );
137         $params = json_encode(
138             array(
139                 'media_id' => $this->input->post('media_id'),
140                 'queue' => $this->input->post('queue')
141             )
142         );
143         $response = $this->rest->post('publish/job', $params, 'json');
144         if ($response->status == 200) {
145             // Set the flash message
146             $this->session->set_flashdata(
147                 'infos', 'The publish job for media "'.$this->input->post('media_in_id').'" has been launched.');
148         }
149         echo json_encode(array('redirect' => site_url('publisher')));
150     }
151     else {
152         echo json_encode(array('errors' => $response->value->description));
153     }
154 }
155 }
156 }
157 }
158
159 /* End of file publisher.php */
160 /* Location: ./application/controllers/publisher.php */

```

transform.php

```

1  <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2
3  class Transform extends MY_Controller
4  {
5
6      protected $page_name = 'transform';
7
8      public function __construct() {
9          parent::__construct();
10     }
11
12     /**
13      * TODO : comments
14      */
15     public function index() {
16         $this->load->helper('number');
17         $this->load->spark('restclient/2.1.0');
18         $this->load->library('rest');
19         $this->rest->initialize(
20             array(
21                 'server' => $this->config->item('orchestra_api_url'), 'http_auth' => 'basic',
22                 'http_user' => $this->user->mail(), 'http_pass' => $this->user->secret()
23             )
24         );
25         $response = $this->rest->get('transform/job');

```

```

26     if ($response->status != 200) {
27         print_r($response->value);
28         exit;
29     }
30     $data['jobs'] = $response->value;
31     // Get the medias for the dropdown
32     $response = $this->rest->get('media/HEAD');
33     if ($response->status != 200) {
34         print_r($response->value);
35         exit;
36     }
37     $data['medias'] = array();
38     foreach ($response->value as $media) {
39         $data['medias'][$media->_id] = $media->metadata->title.' - '.$media->virtual_filename;
40     }
41     // Get the profiles for the dropdown
42     $response = $this->rest->get('transform/profile');
43     if ($response->status != 200) {
44         print_r($response->value);
45         exit;
46     }
47     $data['profiles'] = array();
48     foreach ($response->value as $profile) {
49         $data['profiles'][$profile->_id] = $profile->title;
50     }
51     // Get the queues for the dropdown
52     $response = $this->rest->get('transform/queue');
53     if ($response->status != 200) {
54         print_r($response->value);
55         exit;
56     }
57     $data['queues'] = array();
58     foreach ($response->value as $queue) {
59         $data['queues'][$queue] = $queue;
60     }
61
62     $this->add_content('page_title', 'OSCIED - Transform Jobs');
63     $this->add_view('main', 'transform/show', $data);
64
65     $header_data['page'] = 'transform';
66     $this->render($header_data);
67 }
68
69 /**
70 * TODO : comments
71 */
72 public function refresh() {
73     $this->load->helper('number');
74     $this->load->spark('restclient/2.1.0');
75     $this->load->library('rest');
76     $this->rest->initialize(
77         array(
78             'server' => $this->config->item('orchestra_api_url'), 'http_auth' => 'basic',
79             'http_user' => $this->user->mail(), 'http_pass' => $this->user->secret()
80         )
81     );
82     $response = $this->rest->get('transform/job');
83     if ($response->status != 200) {
84         print_r($response->value);
85         exit;
86     }
87     $data['jobs'] = $response->value;
88     $this->load->view('transform/show_jobs', $data);
89 }
90
91 /**
92 * TODO : comments
93 */
94 public function revoke($id) {
95     $this->load->spark('restclient/2.1.0');
96     $this->load->library('rest');
97     $this->rest->initialize(
98         array(

```

```

99         'server' => $this->config->item('orchestra_api_url'), 'http_auth' => 'basic',
100        'http_user' => $this->user->mail(), 'http_pass' => $this->user->secret()
101    )
102  );
103 $response = $this->rest->delete('transform/job/id/' . $id);
104 // Set error or information message
105 if ($response->status == 200) {
106     $this->session->set_flashdata('infos', $response->value);
107     echo json_encode(array('redirect' => 'transform'));
108 }
109 else {
110     $this->session->set_flashdata('errors', $response->value->description);
111     echo json_encode(array('redirect' => 'transform'));
112 }
113 }
114
115 /**
116 * TODO : comments
117 */
118 public function launch_transform() {
119     $this->load->helper('number');
120     $this->load->library('form_validation');
121
122     $form_id = $this->input->post('form_id');
123
124     $this->form_validation->set_rules('media_in_id', 'Input Media', 'required');
125     $this->form_validation->set_rules('profile_id', 'Profile', 'required');
126     $this->form_validation->set_rules('virtual_filename', 'Virtual filename', 'required');
127     $this->form_validation->set_rules('title', 'Media title', 'required');
128     $this->form_validation->set_rules('queue', 'Queue', 'required');
129
130     if ($this->form_validation->run() === FALSE) {
131         $errors = validation_errors();
132         echo json_encode(array('errors' => $errors));
133     }
134     else {
135         $this->load->spark('restclient/2.1.0');
136         $this->load->library('rest');
137         $this->rest->initialize(
138             array(
139                 'server' => $this->config->item('orchestra_api_url'), 'http_auth' => 'basic',
140                 'http_user' => $this->user->mail(), 'http_pass' => $this->user->secret()
141             )
142         );
143         $params = json_encode(
144             array(
145                 'media_in_id' => $this->input->post('media_in_id'),
146                 'profile_id' => $this->input->post('profile_id'),
147                 'virtual_filename' => $this->input->post('virtual_filename'),
148                 'metadata' => array('title' => $this->input->post('title')),
149                 'queue' => $this->input->post('queue')
150             )
151         );
152         $response = $this->rest->post('transform/job', $params, 'json');
153         if ($response->status == 200) {
154             // Set the flash message
155             $this->session->set_flashdata(
156                 'infos', 'The transform job for media "' . $this->input->post('title') . ' - ' .
157                 $this->input->post('virtual_filename') . '" has been launched.'
158             );
159             echo json_encode(array('redirect' => site_url('transform')));
160         }
161         else {
162             echo json_encode(array('errors' => $response->value->description));
163         }
164     }
165 }
166
167 }
168
169 /* End of file transform.php */
170 /* Location: ./application/controllers/transform.php */

```

upload_files.php

```

1  <?php if (!defined('BASEPATH')) exit('No direct script access allowed');
2
3  class Upload_files extends MY_Controller {
4
5      function __construct() {
6          parent::__construct();
7      }
8
9      /**
10      * TODO : comments
11      */
12      public function upload_video() {
13          // Create the upload folder if necessary
14          $form_id = $this->input->post('form_id');
15          $upload_folder = 'tmp/uploads/'.$form_id '/';
16          if (!is_dir($upload_folder)) {
17              mkdir($upload_folder);
18          }
19          $old_name = $_FILES['userfile']['name'];
20          $name = md5(uniqid(rand(), true)).$this->_ext($old_name);
21          /*$name = strtr($name, 'ÀÁÃÃÃÃÇÈÉÈÉííïòòòÙÙÙÝàáâââââçééééííïòòòòòùúûüýÿ', 'AAAAAAACEEEEIIIIOOOOOUUUU');
22          $name = preg_replace('/([^.a-z0-9]+)/i', '_', $name);*/
23          // $name = str_replace(' ', '_', $name);
24          $config['upload_path'] = $upload_folder;
25          $config['allowed_types'] = 'rv|3gp|asf|asx|avi|axv|dif|dl|dv|fli|flv|gl|lsf|lsx|m3u|m3u8|m4a|m4v|mng|movie|mov|mp4|mp3';
26          $config['max_size'] = $this->config->item('max_upload_size') / 1024; // In KB
27          $config['file_name'] = $name;
28          // $config['remove_spaces'] = false;
29          $this->load->library('upload', $config);
30          if ($this->upload->do_upload()) {
31              // Add file infos to the "tmp_file" table
32              $this->load->model('tmp_files_model');
33              $id = $this->tmp_files_model->add_file(
34                  $form_id, $upload_folder, $name, $old_name
35              );
36              // Set file infos
37              $data = $this->upload->data();
38              $info = new stdClass();
39              $info->id = $id;
40              $info->title = $old_name;
41              $info->name = $name;
42              $info->size = $data['file_size']*1024;
43              $info->tmp = true;
44              // Return the file infos
45              if (IS_AJAX) {
46                  echo json_encode(array($info));
47              }
48              else {
49                  $file_data['upload_data'] = $this->upload->data();
50                  echo json_encode(array($info));
51              }
52          }
53          else {
54              $info = new stdClass();
55              $info->title = $old_name;
56              $info->name = $name;
57              $info->error = $this->upload->display_errors('', '');
58              echo json_encode(array($info));
59          }
60      }
61
62      /**
63      * TODO : comments
64      */
65      protected function _ext($filename) {
66          $x = explode('.', $filename);
67          return '.' . end($x);
68      }
69
70  }
71

```

```

72  /* End of file upload_files.php */
73  /* Location: ./application/controllers/upload_files.php */

```

users.php

```

1  <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2
3  class Users extends MY_Controller
4  {
5
6      protected $page_name = 'users';
7
8      public function __construct() {
9          parent::__construct();
10     }
11
12     /**
13      * TODO : comments
14     */
15     public function index() {
16         $this->load->spark('restclient/2.1.0');
17         $this->load->library('rest');
18         $this->rest->initialize(
19             array(
20                 'server' => $this->config->item('orchestra_api_url'), 'http_auth' => 'basic',
21                 'http_user' => $this->user->mail(), 'http_pass' => $this->user->secret()
22             )
23         );
24         $response = $this->rest->get('user/id/'. $this->user->id());
25         if ($response->status != 200) {
26             print_r($response->value);
27             exit;
28         }
29         $data['user'] = $response->value;
30         if ($this->user->is_admin_platform()) {
31             $response = $this->rest->get('user');
32             if ($response->status != 200) {
33                 print_r($response->value);
34                 exit;
35             }
36             $data['users'] = $response->value;
37             foreach ($data['users'] as $k => $v) {
38                 if ($v->_id == $this->user->id()) {
39                     unset($data['users'][$k]);
40                 }
41             }
42         }
43         // Construct the page
44         $this->add_content('page_title', 'OSCIED - Users');
45         $this->add_view('main', 'users/show', $data);
46         $header_data['page'] = 'users';
47         $this->render($header_data);
48     }
49
50     /**
51      * TODO : comments
52     */
53     public function refresh() {
54         $this->load->spark('restclient/2.1.0');
55         $this->load->library('rest');
56         $this->rest->initialize(
57             array(
58                 'server' => $this->config->item('orchestra_api_url'), 'http_auth' => 'basic',
59                 'http_user' => $this->user->mail(), 'http_pass' => $this->user->secret()
60             )
61         );
62         $response = $this->rest->get('user/id/'. $this->user->id());
63         if ($response->status != 200) {
64             print_r($response->value);
65             exit;
66         }
67         $data['user'] = $response->value;

```

```

68     if ($this->user->is_admin_platform()) {
69         $response = $this->rest->get('user');
70         if ($response->status != 200) {
71             print_r($response->value);
72             exit;
73         }
74         $data['users'] = $response->value;
75         foreach ($data['users'] as $k => $v) {
76             if ($v->_id == $this->user->id()) {
77                 unset($data['users'][$k]);
78             }
79         }
80     }
81     $this->load->view('users/show_users', $data);
82 }
83
84 /**
85 * TODO : comments
86 */
87 public function edit($id) {
88     $this->load->library('form_validation');
89
90     $this->form_validation->set_rules('first_name', 'First name', 'required');
91     $this->form_validation->set_rules('last_name', 'Last name', 'required');
92     $this->form_validation->set_rules('mail', 'Email', 'required');
93
94     if ($this->form_validation->run() === FALSE) {
95         $errors = validation_errors();
96         echo json_encode(array('errors' => $errors));
97     }
98     else {
99         $this->load->spark('restclient/2.1.0');
100        $this->load->library('rest');
101        $this->rest->initialize(
102            array(
103                'server' => $this->config->item('orchestra_api_url'), 'http_auth' => 'basic',
104                'http_user' => $this->user->mail(), 'http_pass' => $this->user->secret()
105            )
106        );
107        $params = array(
108            'first_name' => $this->input->post('first_name'),
109            'last_name' => $this->input->post('last_name'),
110            'mail' => $this->input->post('mail')
111        );
112        $secret = $this->input->post('secret');
113        if (!empty($secret)) {
114            $params['secret'] = $this->input->post('secret');
115            $data = array('user_secret' => $this->input->post('secret'));
116        }
117        else {
118            $data = array();
119        }
120        if ($this->input->post('admin_platform') !== FALSE) {
121            $params['admin_platform'] = $this->input->post('admin_platform');
122        }
123        $response = $this->rest->put('user/id/'. $id, json_encode($params), 'json');
124        if ($response->status == 200) {
125            if ($id == $this->user->id()) {
126                $data = array_merge(
127                    $data,
128                    array(
129                        'user_mail' => $this->input->post('mail'),
130                        'user_name' => $this->input->post('first_name'). ' ' . $this->input->post('last_name'),
131                        'user_logged' => true
132                    )
133                );
134                $this->session->set_userdata($data);
135            }
136            // Set the flash message
137            $this->session->set_flashdata('infos', $response->value);
138            echo json_encode(array('redirect' => site_url('users')));
139        }
140        else {

```

```

141         echo json_encode(array('errors' => $response->value->description));
142     }
143 }
144 }
145 /**
146 * TODO : comments
147 */
148
149 public function delete($id) {
150     $this->load->spark('restclient/2.1.0');
151     $this->load->library('rest');
152     $this->rest->initialize(
153         array(
154             'server' => $this->config->item('orchestra_api_url'), 'http_auth' => 'basic',
155             'http_user' => $this->user->mail(), 'http_pass' => $this->user->secret()
156         )
157     );
158     $response = $this->rest->delete('user/id/'. $id);
159     // Set error or information message
160     if ($response->status == 200) {
161         if ($id == $this->user->id()) {
162             $this->session->sess_destroy();
163             $this->session->sess_create();
164             // Set the flash message and redirect the user
165             $infos[] = 'Now you are logged out.';
166             $this->session->set_flashdata('infos', $infos);
167             echo json_encode(array('redirect' => site_url()));
168         }
169         else {
170             // Set the flash message and redirect the user
171             $this->session->set_flashdata('infos', $response->value);
172             echo json_encode(array('redirect' => 'users'));
173         }
174     }
175     else {
176         $this->session->set_flashdata('errors', $response->value->description);
177         echo json_encode(array('redirect' => 'users'));
178     }
179 }
180 /**
181 * TODO : comments
182 */
183
184 public function add_user() {
185     $this->load->library('form_validation');
186
187     $form_id = $this->input->post('form_id');
188
189     $this->form_validation->set_rules('first_name', 'First name', 'required');
190     $this->form_validation->set_rules('last_name', 'Last name', 'required');
191     $this->form_validation->set_rules('mail', 'Email', 'required');
192     $this->form_validation->set_rules('secret', 'Secret', 'required');
193
194     if ($this->form_validation->run() === FALSE) {
195         $errors = validation_errors();
196         echo json_encode(array('errors' => $errors));
197     }
198     else {
199         $this->load->spark('restclient/2.1.0');
200         $this->load->library('rest');
201         $this->rest->initialize(
202             array(
203                 'server' => $this->config->item('orchestra_api_url'), 'http_auth' => 'basic',
204                 'http_user' => $this->user->mail(), 'http_pass' => $this->user->secret()
205             )
206         );
207         $params = json_encode(
208             array(
209                 'first_name' => $this->input->post('first_name'),
210                 'last_name' => $this->input->post('last_name'),
211                 'mail' => $this->input->post('mail'),
212                 'secret' => $this->input->post('secret'),
213                 'admin_platform' => $this->input->post('admin_platform')

```

```

214         )
215     );
216     $response = $this->rest->post('user', $params, 'json');
217     if ($response->status == 200) {
218         // Set the flash message
219         $this->session->set_flashdata(
220             'infos', 'The user "'.$this->input->post('first_name').'" .
221             '.$this->input->post('last_name')." has been added."
222         );
223         echo json_encode(array('redirect' => site_url('users')));
224     }
225     else {
226         echo json_encode(array('errors' => $response->value->description));
227     }
228 }
229 }
230
231 /**
232 * TODO : comments
233 */
234 public function login() {
235     // $this->load->model('users_model');
236     $this->load->library('form_validation');
237
238     $this->form_validation->set_rules('mail', "Email", 'required');
239     $this->form_validation->set_rules('secret', 'Secret', 'required');
240
241     if ($this->form_validation->run() === FALSE) {
242         $errors = validation_errors();
243         echo json_encode(array('errors' => $errors));
244     }
245     else {
246         // Try to connect the user
247         $this->load->spark('restclient/2.1.0');
248         $this->load->library('rest');
249         $this->rest->initialize(
250             array(
251                 'server' => $this->config->item('orchestra_api_url'),
252                 'http_auth' => 'basic',
253                 'http_user' => $this->input->post('mail'),
254                 'http_pass' => $this->input->post('secret')
255             )
256         );
257         $response = $this->rest->get('user/login');
258         if ($response->status == 200) {
259             // Set the session variables
260             $this->load->library('session');
261             $user = $response->value;
262             $data = array(
263                 'user_id' => $user->_id,
264                 'user_mail' => $user->mail,
265                 'user_secret' => $this->input->post('secret'),
266                 'user_name' => $user->first_name.' '.$user->last_name,
267                 'user_admin_platform' => $user->admin_platform,
268                 'user_logged' => true
269             );
270             $this->session->set_userdata($data);
271             // Set the flash message
272             $this->session->set_flashdata(
273                 'infos', 'Now you are logged as "'.$data['user_name'].'".'
274             );
275             echo json_encode(array('redirect' => site_url()));
276         }
277         else {
278             $error = "Email or secret error";
279             echo json_encode(array('errors' => $error));
280         }
281     }
282 }
283 /**
284 * TODO : comments
285 */
286 public function logout() {
287     $this->session->sess_destroy();

```

```

287     $this->session->sess_create();
288     $this->session->set_flashdata('infos', 'Now you are logged out.');
289     redirect(site_url());
290 }
291
292 }
293
294 /* End of file users.php */
295 /* Location: ./application/controllers/users.php */

```

6.13.7 www/.../helpers

flash_message_helper.php

```

1  <?php  if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2 /**
3 * Force Download
4 *
5 * Generates headers that force a download to happen
6 *
7 * @access public
8 * @param string filename
9 * @param mixed the data to be downloaded
10 * @return void
11 */
12 if ( ! function_exists('force_download'))
13 {
14     function force_download($filename = '', $file = '')
15     {
16         if ($filename == '' OR $file == '')
17         {
18             return FALSE;
19         }
20
21         // Try to determine if the filename includes a file extension.
22         // We need it in order to set the MIME type
23         if (FALSE === strpos($filename, '.'))
24         {
25             return FALSE;
26         }
27
28         // Grab the file extension
29         $x = explode('.', $filename);
30         $extension = end($x);
31
32         // Load the mime types
33         @include(APPPATH.'config/mimes'.EXT);
34
35         // Set a default mime if we can't find it
36         if ( ! isset($mimes[$extension]))
37         {
38             $mime = 'application/octet-stream';
39         }
40         else
41         {
42             $mime = (is_array($mimes[$extension])) ? $mimes[$extension][0] : $mimes[$extension];
43         }
44
45         // Generate the server headers
46         if (strpos($_SERVER['HTTP_USER_AGENT'], "MSIE") !== FALSE)
47         {
48             header('Content-Type: "'.$mime.'"');
49             header('Content-Disposition: attachment; filename="'.$filename.'"');
50             header('Expires: 0');
51             header('Cache-Control: must-revalidate, post-check=0, pre-check=0');
52             header("Content-Transfer-Encoding: binary");
53             header('Pragma: public');
54             header("Content-Length: ".filesize($file));
55         }
56         else
57         {

```

```

58         header('Content-Type: "'.$mime.'"');
59         header('Content-Disposition: attachment; filename="'.$filename.''");
60         header("Content-Transfer-Encoding: binary");
61         header('Expires: 0');
62         header('Pragma: no-cache');
63         header("Content-Length: ".$filesize[$file]);
64     }
65
66     readfile_chunked($file);
67     die;
68 }
69
70 /**
71 * readfile_chunked
72 *
73 * Reads file in chunks so big downloads are possible without changing PHP.INI
74 *
75 * @access public
76 * @param string file
77 * @param boolean return bytes of file
78 * @return void
79 */
80
81 if ( ! function_exists('readfile_chunked'))
82 {
83     function readfile_chunked($file, $retbytes=TRUE)
84     {
85         $chunksize = 1 * (1024 * 1024);
86         $buffer = '';
87         $cnt = 0;
88
89         $handle = fopen($file, 'r');
90         if ($handle === FALSE)
91         {
92             return FALSE;
93         }
94
95         while (!feof($handle))
96         {
97             $buffer = fread($handle, $chunksize);
98             echo $buffer;
99             ob_flush();
100            flush();
101
102            if ($retbytes)
103            {
104                $cnt += strlen($buffer);
105            }
106        }
107
108        $status = fclose($handle);
109
110        if ($retbytes AND $status)
111        {
112            return $cnt;
113        }
114
115        return $status;
116    }
117 }
118
119 /* End of file MY_download_helper.php */
120 /* Location: ./application/helpers/MY_download_helper.php */

```

MY_download_helper.php

Note: We gracefully thanks Derek Jones for his download helper that permit to the website to download large media files without using GB of RAM ! The link to the code : [Download-helper-for-large-files](#)

6.13.8 www/.../models

files_model.php

```

1  <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2
3  class Files_model extends CI_Model {
4
5      protected $table = 'files';
6
7      public function __construct() {
8          $this->load->database();
9      }
10
11
12      public function add_files($files) {
13          $this->db->insert_batch($this->table, $files);
14          // Return the inserted ids
15          $ids = array();
16          for ($i=0; $i < count($files); $i++) {
17              $ids[] = $this->db->insert_id() + $i;
18          }
19          return $ids;
20      }
21
22
23      public function get_name($file_id) {
24          $query = $this->db
25              ->select('name')
26              ->from($this->table)
27              ->where('id', $file_id)
28              ->get();
29          return ($query->num_rows() > 0)?$query->row()->name:NULL;
30      }
31
32
33      public function get_file($file_id) {
34          $query = $this->db
35              ->select('dir, name, title')
36              ->from($this->table)
37              ->where('id', $file_id)
38              ->get();
39          return ($query->num_rows() > 0)?$query->row():NULL;
40      }
41
42
43      public function get_files_in($ids) {
44          return $this->db
45              ->select('id, name, title')
46              ->from($this->table)
47              ->where_in('id', $ids)
48              ->order_by('id', 'ASC')
49              ->get()->result();
50      }
51
52
53      public function get_files_in_dir($dir) {
54          return $this->db
55              ->select('id, name, title')
56              ->from($this->table)
57              ->where('dir', $dir)
58              ->order_by('id', 'ASC')
59              ->get()->result();
60      }
61
62      public function delete_file($file_id) {
63          $this->db
64              ->where('id', $file_id)
65              ->delete($this->table);
66      }
67
68  }

```

tmp_files_model.php

```

1  <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2
3  class Tmp_Files_Model extends CI_Model {
4
5      protected $table = 'tmp_files';
6
7      public function __construct() {
8          $this->load->database();
9      }
10
11
12      public function add_file($form_id, $dir, $name, $title) {
13          $this->db
14              ->set('form_id', $form_id)
15              ->set('dir', $dir)
16              ->set('name', $name)
17              ->set('title', $title);
18          $this->db->insert($this->table);
19          return $this->db->insert_id();
20      }
21
22
23      public function get_file($file_id) {
24          $query = $this->db
25              ->select('dir, name, title')
26              ->from($this->table)
27              ->where('id', $file_id)
28              ->get();
29          return ($query->num_rows() > 0)?$query->row():NULL;
30      }
31
32
33      public function get_files($form_id) {
34          return $this->db
35              ->select('dir, name, title')
36              ->from($this->table)
37              ->where('form_id', $form_id)
38              ->get()->result_array();
39      }
40
41
42      public function delete_file($file_id) {
43          $this->db
44              ->where('id', $file_id)
45              ->delete($this->table);
46      }
47
48
49      public function delete_files($form_id) {
50          $this->db
51              ->where('form_id', $form_id)
52              ->delete($this->table);
53      }
54
55
56      public function a_file_exist($form_id) {
57          return $this->db
58              ->where('form_id', $form_id)
59              ->count_all_results($this->table);
60      }
61
62  }

```

6.13.9 www/.../views

contact.php

```
1 TODO
2 <br />
```

homepage.php

```
1 This is the home page :P
```

media/add_media_form.php

```
1 <!--<link rel="stylesheet" href="php echo base_url(); ?&gt;assets/css/fileupload/jquery-ui.css" id="theme"&gt;
2 &lt;link rel="stylesheet" href="<?= base_url(); ?&gt;assets/css/fileupload/jquery.fileupload-ui.css"&gt;--&gt;
3 &lt;?= $this-&gt;css_js-&gt;load_css('assets/css/fileupload/jquery-ui.css', 'theme'); ?&gt;
4 &lt;?= $this-&gt;css_js-&gt;load_css('assets/css/fileupload/jquery.fileupload-ui.css'); ?&gt;
5
6 &lt;h3&gt;Add a media&lt;/h3&gt;
7
8 &lt;div id="add_media_errors" class="alert alert-error hidden"&gt;&lt;/div&gt;
9 &lt;?= form_open_multipart('upload_files/upload_video', array('id' =&gt; 'fileupload', 'name' =&gt; 'fileupload', 'data'
10   &lt;?= form_hidden('form_id', md5(uniqid(rand(), true))); ?&gt;
11   &lt;label for="title"&gt;Title&lt;/label&gt;
12   &lt;?= form_input(array('name' =&gt; 'title', 'id' =&gt; 'title', 'class' =&gt; 'input-large')) ; ?&gt;
13   &lt;label for="virtual_filename"&gt;Virtual filename&lt;/label&gt;
14   &lt;?= form_input(array('name' =&gt; 'virtual_filename', 'id' =&gt; 'virtual_filename', 'class' =&gt; 'input-large')) ;
15   &lt;div id="attachments_div"&gt;
16     &lt;?= $this-&gt;load-&gt;view('fileupload/upload', array('table_id' =&gt; 'files')) ; ?&gt;
17   &lt;/div&gt;
18   &lt;br id="space" class="hide" /&gt;
19   &lt;?= form_submit('submit', 'Add media', 'id=add_media_submit' class="btn btn-primary"); ?&gt;
20 &lt;?= form_close("\n") ?&gt;
21
22 &lt;?= $this-&gt;load-&gt;view('fileupload/scripts', array('upload_id' =&gt; 'template-upload', 'download_id' =&gt; 'template
23
24 &lt;script type="text/javascript"&gt;
25 /*&lt;!-- Post data with AJAX --&gt;*/
26 $('#add_media_submit').click(function() {
27   $.post(
28     "&lt;?= site_url('media/add_media') ?&gt;",
29     $('#fileupload').serialize(),
30     function(data) {
31       if (data.errors) {
32         $('#add_media_errors').empty();
33         $('#add_media_errors').removeClass('hidden');
34         $('#add_media_errors').append(data.errors);
35       }
36       else {
37         window.location = data.redirect;
38       }
39     },
40     'json'
41   );
42   return false;
43 });
44 /*&lt;!-- Enable/disable submit button during upload --&gt;*/
45 $('#fileupload')
46   .bind('fileuploadstart', function (e, data) {
47     $('#add_media_submit').attr('disabled', true);
48   })
49   .bind('fileuploadstop', function (e) {
50     $('#add_media_submit').attr('disabled', false);
51   })
52 ;
53 /*&lt;!-- Set the action for file deletion --&gt;*/
54 $('body').on('click', '.template_download_button', function () {
55   if ($(this).is('.btn-warning')) {
56     $(this).closest('tr').remove();
57   }
58 });</pre

```

```

57     }
58     else {
59         if (confirm('Do you really want to delete the file "' + $(this).attr('title') + '"?')) {
60             var row = $(this).closest('tr');
61             $.get(
62                 '<?= base_url() ?>media/delete_file/' + $(this).attr('file_id') + '/tmp',
63                 function (data) {
64                     row.remove();
65                 });
66             $('#fileupload').fileupload(
67                 'option', 'maxNumberOfFiles', 1
68             );
69         );
70     }
71 }
72 return false;
73 });
74 /*!-- Initialize the jQuery File Upload widget -->*/
75 $('#fileupload').fileupload();
76 /*!-- Set the accepted file types and other options -->*/
77 $('#fileupload').fileupload('option', {
78     /*uploadTemplateId: 'template-upload',
79     downloadTemplateId: 'template_download',
80     filesContainer: widgetContainer.find('#files'),*/
81     filesContainer: '#files',
82     autoUpload: true,
83     maxNumberOfFiles: 1,
84     /*maxFileSize: <?= $this->config->item('max_upload_size') ?>,*/
85     acceptFileTypes: /(\.|\/)(rv|3gp|asf|asx|avi|axv|dif|dl|dv|fli|flv|gl|lsf|lsx|mkv|mng|movie|mov|mp4|mpeg|mp
86 });
87 </script>

```

media/show.php

```

1 <h1 class="page-header">Available media assets</h1>
2
3 <div id="medias">
4     <?= $this->load->view('media/show_medias') ?>
5 </div>
6
7 <script type="text/javascript">
8 /*!-- Set the action for file deletion -->*
9 $('#body').on('click', '.delete', function () {
10     if (confirm('Do you really want to delete the media "' + $(this).attr('title') + '"?')) {
11         $.get(
12             $(this).attr('href'),
13             function (data) {
14                 window.location = data.redirect;
15             },
16             'json'
17         );
18     }
19     return false;
20 });
21 </script>
22
23 <?php if ($this->user->is_logged()): ?>
24 <?= $this->load->view('media/add_media_form'); ?>
25 <?php endif; ?>

```

media/show_medias.php

```

1 <?php if (empty($medias)): ?>
2
3 <p>There is currently no media available.</p>
4
5 <?php else: ?>
6
7 <table class="table table-bordered table-condensed table-striped">
8     <thead>

```

```

9      <tr>
10     <th>Title</th>
11     <th>Virtual Filename</th>
12     <th>File size</th>
13     <th>Duration</th>
14     <th>Added on</th>
15     <th>Added by</th>
16     <th>Status</th>
17     <th></th>
18   </tr>
19 </thead>
20 <tbody>
21   <?php foreach ($medias as $k => $media): ?>
22   <?php $media = $media; ?>
23   <tr>
24     <td><?= (isset($media->metadata->title) ?$media->metadata->title:'') ?></td>
25     <td>
26       <?php
27         $status = isset($media->status)?strtoupper($media->status):'UNKNOWN';
28         if ($status == 'READY' or $status == 'PUBLISHED'): ?>
29         <a href="= site_url('media/force_download/'.$media-&gt;id) ?&gt;"&gt;&lt;?= isset($media-&gt;virtual_filename)?
30           &lt;?php else: ?&gt;
31           &lt;?= isset($media-&gt;virtual_filename)?$media-&gt;virtual_filename:'Untitled' ?&gt;
32           &lt;?php endif; ?&gt;
33       &lt;/td&gt;
34     &lt;td&gt;&lt;?= (isset($media-&gt;metadata-&gt;size)?byte_format($media-&gt;metadata-&gt;size):'') ?&gt;&lt;/td&gt;
35     &lt;td&gt;&lt;?= (isset($media-&gt;metadata-&gt;duration)?$media-&gt;metadata-&gt;duration:'') ?&gt;&lt;/td&gt;
36     &lt;td&gt;&lt;?= (isset($media-&gt;metadata-&gt;add_date)?$media-&gt;metadata-&gt;add_date:'') ?&gt;&lt;/td&gt;
37     &lt;td&gt;&lt;?= (isset($media-&gt;user-&gt;name)?$media-&gt;user-&gt;name:'') ?&gt;&lt;/td&gt;
38     &lt;td&gt;
39       &lt;?php
40         switch ($status) {
41           case 'PENDING': $class = 'label label-warning'; break;
42           case 'READY': $class = 'label label-success'; break;
43           case 'UNKNOWN': $class = 'label'; break;
44           default: $class = 'label label-inverse'; break;
45         }
46       &lt;?
47       &lt;span class="<?= $class ?&gt;"= $status ?&gt;&lt;/span&gt;
48     &lt;/td&gt;
49     &lt;td&gt;
50       &lt;?php if (($media-&gt;user-&gt;_id == $this-&gt;user-&gt;id()) and ($status == 'READY' or $status == 'PUBLISHED')) {
51         &lt;a class="delete" title="<?= $media-&gt;metadata-&gt;title ?&gt;" href="<?= site_url('media/delete/'.$media-&gt;id) ?&gt;"
52           <?php endif; ?>
53       </td>
54     </tr>
55   <?php endforeach; ?>
56 </tbody>
57 </table>
58
59 <?php endif; ?>
60
61 <script type="text/javascript">
62 $(document).ready(function() {
63   setTimeout(
64     function() {
65       $.get(
66         "<?= site_url('media/refresh') ?>",
67         function(data) {
68           $('#medias').html(data);
69         },
70         'html'
71       );
72     },
73     <?= $this->config->item('medias_refresh') * 1000 ?>
74   );
75 });
76 </script>

```

profile/add_profile_form.php

```

1 <h3>Add a transform profile</h3>
2
3 <div id="add_profile_errors" class="alert alert-error hidden"></div>
4 <?= form_open('profile/add_profile', array('id' => 'form_add_profile')) ; ?>
5   <?= form_hidden('form_id', md5(uniqid(rand()), true)); ?>
6   <table class="table table-bordered table-condensed">
7     <thead>
8       <tr>
9         <th>Title</th>
10        <th>Description</th>
11        <th>Encoder string</th>
12      </tr>
13    </thead>
14    <tbody>
15      <tr>
16        <td><?= form_input(array('name' => 'title', 'class' => 'title input-medium')) ?></td>
17        <td><?= form_input(array('name' => 'description', 'class' => 'description input-medium')) ?></td>
18        <td><?= form_input(array('name' => 'encoder_string', 'class' => 'encoder_string input-large')) ?></td>
19      </tr>
20    </tbody>
21  </table>
22  <?= form_submit('submit', 'Add profile', 'id="add_profile_submit" class="btn btn-primary"'); ?>
23 <?= form_close("\n") ?>
24
25 <script type="text/javascript">
26 /*<!-- Post data with AJAX --&gt;*/ 
27 $('#add_profile_submit').click(function() {
28   $.post(
29     "&lt;?= site_url('profile/add_profile') ?&gt;",
30     $('#form_add_profile').serialize(),
31     function(data) {
32       if (data.errors) {
33         $('#add_profile_errors').empty();
34         $('#add_profile_errors').removeClass('hidden');
35         $('#add_profile_errors').append(data.errors);
36       }
37       else {
38         window.location = data.redirect;
39       }
40     },
41     'json'
42   );
43   return false;
44 });
45 &lt;/script&gt;
</pre>

```

profile/show.php

```

1 <h1 class="page-header">Available transform profiles</h1>
2
3 <div id="profiles">
4   <?= $this->load->view('profile/show_profiles') ?>
5 </div>
6
7 <script type="text/javascript">
8 /*<!-- Set the action for file deletion --&gt;*/ 
9 $('body').on('click', '.delete', function () {
10   if (confirm('Do you really want to delete the profile "' + $(this).attr('title') + '"?')) {
11     $.get(
12       $(this).attr('href'),
13       function (data) {
14         window.location = data.redirect;
15       },
16       'json'
17     );
18   }
19   return false;
20 });
21 &lt;/script&gt;
</pre>

```

```

22
23 <?php if ($this->user->is_logged()): ?>
24 <?= $this->load->view('profile/add_profile_form'); ?>
25 <?php endif; ?>
```

profile/show_profiles.php

```

1  <?php if (empty($profiles)): ?>
2
3 <p>There is currently no profile available.</p>
4
5 <?php else: ?>
6
7 <table class="table table-bordered table-condensed table-striped">
8   <thead>
9     <tr>
10       <th>Title</th>
11       <th>Description</th>
12       <th>Encoder string</th>
13       <th></th>
14     </tr>
15   </thead>
16   <tbody>
17     <?php foreach ($profiles as $k => $profile): ?>
18     <tr>
19       <td><?= ($profile->title)?$profile->title:'></td>
20       <td><?= ($profile->description)?$profile->description:'></td>
21       <td><?= ($profile->encoder_string)?$profile->encoder_string:'></td>
22       <td>
23         <a class="delete" title="<?= $profile->title ?>" href="<?= site_url('profile/delete/'.$profile->i
24       </td>
25     </tr>
26     <?php endforeach; ?>
27   </tbody>
28 </table>
29
30 <?php endif; ?>
31
32 <script type="text/javascript">
33 $(document).ready(function() {
34   setTimeout(
35     function() {
36       $.get(
37         "<?= site_url('profile/refresh') ?>",
38         function(data) {
39           $('#profiles').html(data);
40         },
41         'html'
42       );
43     },
44     <?= $this->config->item('profiles_refresh') * 1000 ?>
45   );
46 });
47 </script>
```

publisher/launch_publish_form.php

```

1 <h3>Launch a publish job</h3>
2
3 <div id="launch_job_errors" class="alert alert-error hidden"></div>
4 <?= form_open('publisher/launch_publish', array('id' => 'form_launch_job')); ?>
5   <?= form_hidden('form_id', md5(uniqid(rand(), true))); ?>
6   <table class="table table-bordered table-condensed">
7     <thead>
8       <tr>
9         <th>Media</th>
10        <th>Queue</th>
11      </tr>
12    </thead>
13    <tbody>
```

```

14         <tr>
15             <td><?= form_dropdown('media_id', $medias, $this->input->post('media_id')) ?></td>
16             <td><?= form_dropdown('queue', $queues, $this->input->post('queue')) ?></td>
17         </tr>
18     </tbody>
19   </table>
20   <?= form_submit('submit', 'Launch job', 'id="launch_job_submit" class="btn btn-primary"'); ?>
21 <?= form_close("\n") ?>
22
23 <script type="text/javascript">
24 /*!-- Post data with AJAX --*/
25 $('#launch_job_submit').click(function() {
26     $.post(
27         "<?= site_url('publisher/launch_publish') ?>",
28         $('#form_launch_job').serialize(),
29         function(data) {
30             if (data.errors) {
31                 $('#launch_job_errors').empty();
32                 $('#launch_job_errors').removeClass('hidden');
33                 $('#launch_job_errors').append(data.errors);
34             }
35             else {
36                 window.location = data.redirect;
37             }
38         },
39         'json'
40     );
41     return false;
42 });
43 </script>

```

publisher/show.php

```

1 <h1 class="page-header">Publish(er) jobs</h1>
2
3 <div id="publisher_jobs">
4     <?= $this->load->view('publisher/show_jobs'); ?>
5 </div>
6
7 <script type="text/javascript">
8 /*!-- Set the action for file deletion --*/
9 $('body').on('click', '.revoke', function () {
10     if (confirm('Do you really want to revoke the job "' + $(this).attr('title') + '"?')) {
11         $.get(
12             $(this).attr('href'),
13             function (data) {
14                 window.location = data.redirect;
15             },
16             'json'
17         );
18     }
19     return false;
20 });
21 </script>
22
23 <?php if ($this->user->is_logged()): ?>
24 <?= $this->load->view('publisher/launch_publish_form'); ?>
25 <?php endif; ?>

```

publisher/show_jobs.php

```

1 <?php if (empty($jobs)): ?>
2
3 <p>Be the first to publish a media !</p>
4
5 <?php else: ?>
6
7 <table class="table table-bordered table-condensed table-striped">
8     <thead>
9         <tr>

```

```

10      <th class="span2">Media</th>
11      <th>Added by</th>
12      <th class="span2">Added / Started on</th>
13      <th>Publication point</th>
14      <th>Elapsed</th>
15      <th class="span3">Progress</th>
16      <th>Error</th>
17      <th>Status</th>
18      <th></th>
19      </tr>
20  </thead>
21  <tbody>
22  <?php foreach ($jobs as $k => $job): ?>
23  <?php $job = $job; ?>
24  <tr>
25      <td>
26          <?php $media = isset($job->media->virtual_filename)?$job->media->virtual_filename:''; ?>
27          <?php if (isset($job->publish_uri)): ?>
28              <a href="= $job-&gt;publish_uri ?&gt;"&gt;<?= $media ?&gt;&lt;/a&gt;
29          &lt;?php else: ?&gt;
30              &lt;?= $media ?&gt;
31          &lt;?php endif; ?&gt;
32      &lt;/td&gt;
33      &lt;td&gt;&lt;?= (isset($job-&gt;user-&gt;name)?$job-&gt;user-&gt;name:'') ?&gt;&lt;/td&gt;
34      &lt;td&gt;
35          &lt;?php
36              $add_date = isset($job-&gt;statistic-&gt;add_date)?$job-&gt;statistic-&gt;add_date('');
37              $start_date = isset($job-&gt;statistic-&gt;start_date)?$job-&gt;statistic-&gt;start_date('');
38              echo $add_date.'&lt;br/&gt;'.$start_date
39          ?&gt;
40      &lt;/td&gt;
41      &lt;td&gt;&lt;?= (isset($job-&gt;statistic-&gt;hostname)?$job-&gt;statistic-&gt;hostname:'') ?&gt;&lt;/td&gt;
42      &lt;td&gt;
43          &lt;?php
44              $elapsed = intval(isset($job-&gt;statistic-&gt;elapsed_time)?$job-&gt;statistic-&gt;elapsed_time:0);
45              $eta = intval(isset($job-&gt;statistic-&gt;eta_time)?$job-&gt;statistic-&gt;eta_time:0);
46              echo gmdate('H:i:s',$elapsed).'&lt;br/&gt;'.gmdate('H:i:s',$eta);
47          ?&gt;
48      &lt;/td&gt;
49      &lt;td&gt;
50          &lt;?php
51              $status = isset($job-&gt;status)?strtoupper($job-&gt;status):'UNKNOWN';
52              switch ($status) {
53                  case 'PENDING': $class = 'progress progress-striped progress-info'; break;
54                  case 'PROGRESS': $class = 'progress progress-striped progress-info active'; break;
55                  case 'RETRY': $class = 'progress progress-striped progress-warning'; break;
56                  case 'SUCCESS': $class = 'progress progress-striped progress-success'; break;
57                  case 'FAILURE': $class = 'progress progress-striped progress-danger'; break;
58                  case 'REVOKED': $class = 'progress progress-striped progress-info'; break;
59                  case 'UNKNOWN': $class = 'progress progress-striped progress-danger'; break;
60                  default: $class = 'progress progress-striped progress-info'; break;
61              }
62          ?&gt;
63          &lt;div class="<?= $class ?&gt;"&gt;
64              &lt;div class="bar" style="width: &lt;?= isset($job-&gt;statistic-&gt;percent)?$job-&gt;statistic-&gt;percent:0 ?&gt;"&gt;
65              &lt;/div&gt;
66          &lt;/div&gt;
67          &lt;td&gt;&lt;?= (isset($job-&gt;statistic-&gt;error)?print_r($job-&gt;statistic-&gt;error):'') ?&gt;&lt;/td&gt;
68          &lt;td&gt;
69              &lt;?php
70                  switch ($status) {
71                      case 'PENDING': $class = 'label label-warning'; break;
72                      case 'PROGRESS': $class = 'label label-info'; break;
73                      case 'RETRY': $class = 'label label-warning'; break;
74                      case 'SUCCESS': $class = 'label label-success'; break;
75                      case 'FAILURE': $class = 'label label-important'; break;
76                      case 'REVOKED': $class = 'label label-inverse'; break;
77                      case 'UNKNOWN': $class = 'label'; break;
78                      default: $class = 'label label-inverse'; break;
79                  }
80          ?&gt;
81          &lt;span class="<?= $class ?&gt;"= $status ?&gt;&lt;/span&gt;
82      &lt;/td&gt;
</pre

```

```

83         <td>
84             <?php if (($status != 'SUCCESS') && ($status != 'FAILURE') && ($status != 'REVOKE') && ($job->use
85                 <a class="revoke" title="<?= $job->_id ?>" href="<?= site_url('publisher/revoke/'.$job->_id) ?>"><
86                     <?php endif; ?>
87             </td>
88         </tr>
89         <?php endforeach; ?>
90     </tbody>
91 </table>
92
93 <?php endif; ?>
94
95 <script type="text/javascript">
96 $(document).ready(function() {
97     setTimeout(
98         function() {
99             $.get(
100                 "<?= site_url('publisher/refresh') ?>",
101                 function(data) {
102                     $('#publisher_jobs').html(data);
103                 },
104                 'html'
105             );
106         },
107         <?= $this->config->item('publisher_refresh') * 1000 ?>
108     );
109 });
110 </script>

```

transform/launch_transform_form.php

```

1 <h3>Launch a transform job</h3>
2
3 <div id="launch_job_errors" class="alert alert-error hidden"></div>
4 <?= form_open('transform/launch_transform', array('id' => 'form_launch_job')) ; ?>
5     <?= form_hidden('form_id', md5(uniqid(rand(), true))); ?>
6     <table class="table table-bordered table-condensed">
7         <thead>
8             <tr>
9                 <th>Input Media</th>
10                <th>Profile</th>
11                <th>Virtual Filename</th>
12                <th>Media Title</th>
13                <th>Queue</th>
14            </tr>
15        </thead>
16        <tbody>
17            <tr>
18                <td><?= form_dropdown('media_in_id', $medias, $this->input->post('media_in_id')) ?></td>
19                <td><?= form_dropdown('profile_id', $profiles, $this->input->post('profile_id')) ?></td>
20                <td><?= form_input(array('name' => 'virtual_filename', 'class' => 'encoder_string input-large')) ?></td>
21                <td><?= form_input(array('name' => 'title', 'class' => 'encoder_string input-large')) ?></td>
22                <td><?= form_dropdown('queue', $queues, $this->input->post('queue')) ?></td>
23            </tr>
24        </tbody>
25    </table>
26    <?= form_submit('submit', 'Launch job', 'id="launch_job_submit" class="btn btn-primary"'); ?>
27 <?= form_close("\n") ?>
28
29 <script type="text/javascript">
30 /*!-- Post data with AJAX --*/
31 $('#launch_job_submit').click(function() {
32     $.post(
33         "<?= site_url('transform/launch_transform') ?>",
34         $('#form_launch_job').serialize(),
35         function(data) {
36             if (data.errors) {
37                 $('#launch_job_errors').empty();
38                 $('#launch_job_errors').removeClass('hidden');
39                 $('#launch_job_errors').append(data.errors);
40             }
41             else {

```

```

42         window.location = data.redirect;
43     }
44   },
45   'json'
46 );
47 return false;
48 });
49 </script>
```

transform/show.php

```

1  <h1 class="page-header">Transform jobs</h1>
2
3  <div id="transform_jobs">
4      <?= $this->load->view('transform/show_jobs'); ?>
5  </div>
6
7  <script type="text/javascript">
8  /*!-- Set the action for file deletion --*/
9  $('body').on('click', '.revoke', function () {
10      if (confirm('Do you really want to revoke the job "' + $(this).attr('title') + '"?')) {
11          $.get(
12              $(this).attr('href'),
13              function (data) {
14                  window.location = data.redirect;
15              },
16              'json'
17          );
18      }
19      return false;
20  });
21 </script>
22
23 <?php if ($this->user->is_logged()): ?>
24 <?= $this->load->view('transform/launch_transform_form'); ?>
25 <?php endif; ?>
```

transform/show_jobs.php

```

1  <?php if (empty($jobs)): ?>
2
3  <p>Be the first to transform a media !</p>
4
5  <?php else: ?>
6
7  <table class="table table-bordered table-condensed table-striped">
8      <thead>
9          <tr>
10              <th class="span2">Input / output media</th>
11              <th class="span2">Profile / Added by</th>
12              <th class="span2">Added / Started on</th>
13              <th class="span2">Transform node</th>
14              <th>Elapsed</th>
15              <th class="span3">Progress</th>
16              <th>Error</th>
17              <th>Status</th>
18              <th></th>
19      </tr>
20  </thead>
21  <tbody>
22      <?php foreach ($jobs as $k => $job): ?>
23      <tr>
24          <td>
25              <?php
26                  $input = isset($job->media_in->virtual_filename)?$job->media_in->virtual_filename:'';
27                  $output = isset($job->media_out->virtual_filename)?$job->media_out->virtual_filename:'';
28                  echo $input.'<br/>' . $output;
29              ?>
30          </td>
31          <td>
```

```

32      <?php
33      $profile = isset($job->profile->title)?$job->profile->title:'';
34      $user = isset($job->user->name)?$job->user->name:'';
35      echo $profile.'<br/>' . $user;
36      ?>
37  </td>
38  <td>
39      <?php
40      $add_date = isset($job->statistic->add_date)?$job->statistic->add_date:'';
41      $start_date = isset($job->statistic->start_date)?$job->statistic->start_date:'';
42      echo $add_date.'<br/>' . $start_date
43      ?>
44  </td>
45  <td><?= (isset($job->statistic->hostname)?$job->statistic->hostname:'') ?></td>
46  <td>
47      <?php
48      $elapsed = intval(isset($job->statistic->elapsed_time)?$job->statistic->elapsed_time:0);
49      $eta = intval(isset($job->statistic->eta_time)?$job->statistic->eta_time:0);
50      echo gmdate('H:i:s', $elapsed) . '<br/>' . gmdate('H:i:s', $eta);
51      ?>
52  </td>
53  <td>
54      <?php
55      $status = isset($job->status)?strtoupper($job->status):'UNKNOWN';
56      switch ($status) {
57          case 'PENDING': $class = 'progress progress-striped progress-info'; break;
58          case 'PROGRESS': $class = 'progress progress-striped progress-info active'; break;
59          case 'RETRY': $class = 'progress progress-striped progress-warning'; break;
60          case 'SUCCESS': $class = 'progress progress-striped progress-success'; break;
61          case 'FAILURE': $class = 'progress progress-striped progress-danger'; break;
62          case 'REVOKED': $class = 'progress progress-striped progress-info'; break;
63          case 'UNKNOWN': $class = 'progress progress-striped progress-danger'; break;
64          default: $class = 'progress progress-striped progress-info'; break;
65      }
66      ?>
67      <div class="<?= $class ?>">
68          <div class="bar" style="width: <?= isset($job->statistic->percent)?$job->statistic->percent:0 ?>%">
69          </div>
70  </td>
71  <td><?= (isset($job->statistic->error)?print_r($job->statistic->error):'') ?></td>
72  <td>
73      <?php
74      $title = isset($job->statistic->error_details)?$job->statistic->error_details:'';
75      switch ($status) {
76          case 'PENDING': $class = 'label label-warning'; break;
77          case 'PROGRESS': $class = 'label label-info'; break;
78          case 'RETRY': $class = 'label label-warning'; break;
79          case 'SUCCESS': $class = 'label label-success'; break;
80          case 'FAILURE': $class = 'label label-important'; break;
81          case 'REVOKED': $class = 'label label-inverse'; break;
82          case 'UNKNOWN': $class = 'label'; break;
83          default: $class = 'label label-inverse'; break;
84      }
85      ?>
86      <span class="<?= $class ?>" title="<?= $title ?>"><?= $status ?></span>
87  </td>
88  <td>
89      <?php if (($status != 'SUCCESS') && ($status != 'FAILURE') && ($status != 'REVOKED') && ($job->use
90      <a class="revoke" title="<?= $job->_id ?>" href="<?= site_url('transform/revoke/'.$job->_id) ?>"><?php endif; ?>
91      </a>
92  </td>
93  </tr>
94  <?php endforeach; ?>
95  </tbody>
96  </table>
97
98  <?php endif; ?>
99
100 <script type="text/javascript">
101 $(document).ready(function() {
102     setTimeout(
103         function() {
104             $.get(

```

```

105     "<?= site_url('transform/refresh') ?>",
106     function(data) {
107         $('#transform_jobs').html(data);
108     },
109     'html'
110 );
111 },
112 <?= $this->config->item('transform_refresh') * 1000 ?>
113 );
114 });
115 </script>

```

users/add_user_form.php

```

1 <h3>Add an user</h3>
2 <div id="add_users_errors" class="alert alert-error hidden"></div>
3 <?= form_open('users/add_user', array('id' => 'form_add_user')) ?>
4     <?= form_hidden('form_id', md5(uniqid(rand(), true))) ?>
5     <table class="table table-bordered table-condensed">
6         <thead>
7             <tr>
8                 <th>First name</th>
9                 <th>Last name</th>
10                <th>Email</th>
11                <th>Secret</th>
12                <th>Admin platform</th>
13            </tr>
14        </thead>
15        <tbody>
16            <tr>
17                <td><?= form_input(array('name' => 'first_name', 'class' => 'first_name input-medium')) ?></td>
18                <td><?= form_input(array('name' => 'last_name', 'class' => 'last_name input-medium')) ?></td>
19                <td><?= form_input(array('name' => 'mail', 'class' => 'mail input-large')) ?></td>
20                <td><?= form_password(array('name' => 'secret', 'class' => 'secret input-small')) ?></td>
21                <td><?= form_checkbox(array('name' => 'admin_platform', 'class' => 'admin_platform input-medium')) ?>
22            </tr>
23        </tbody>
24    </table>
25    <?= form_submit('submit', 'Add user', 'id="add_user_submit" class="btn btn-primary"'); ?>
26 <?= form_close("\n") ?>
27
28 <script type="text/javascript">
29 /*!<!-- Post data with AJAX -->*/
30 $('#add_user_submit').click(function() {
31     $.post(
32         "<?= site_url('users/add_user') ?>",
33         $('#form_add_user').serialize(),
34         function(data) {
35             if (data.errors) {
36                 $('#add_users_errors').empty();
37                 $('#add_users_errors').removeClass('hidden');
38                 $('#add_users_errors').append(data.errors);
39             }
40             else {
41                 window.location = data.redirect;
42             }
43         },
44         'json'
45     );
46     return false;
47 });
48 </script>

```

users/login_modal.php

```

1 <div class="modal hide fade" id="connexionModal" tabindex="-1" role="dialog" aria-labelledby="connexionModalLabel"
2     <div class="modal-header">
3         <button type="button" class="close" data-dismiss="modal" aria-hidden="true">>x</button>
4         <h3 id="connexionModalLabel">Connexion</h3>
5     </div>

```

```

6   <div class="modal-body">
7     <div id="login_errors" class="alert alert-error hidden">
8     </div>
9
10    <?= form_open('users/login', array('id' => 'loginForm', 'name' => 'loginForm')); ?>
11      <label for="mail">Nom d'utilisateur</label>
12      <?= form_input(array('name' => 'mail', 'id' => 'mail', 'value' => set_value('mail'))); ?>
13
14      <label for="secret">Mot de passe</label>
15      <?= form_password(array('name' => 'secret', 'id' => 'secret'))); ?>
16
17      <br />
18      <?= form_submit('submit', 'Connexion', 'id="login_submit" class="btn btn-primary"'); ?>
19      <?= form_close("\n") ?>
20   </div>
21 </div>
22
23 <script type="text/javascript">
24 /*!-- Give the focus to the mail input -->*/
25 $('#connexionModal').on('shown', function () {
26   $('#mail').focus();
27 });
28 /*!-- Post data with AJAX -->*/
29 $('#login_submit').click(function() {
30   $.post(
31     "<?= site_url('users/login') ?>",
32     $('#loginForm').serialize(),
33     function(data) {
34       if (data.errors) {
35         $('#login_errors').empty();
36         $('#login_errors').removeClass('hidden');
37         $('#login_errors').append(data.errors);
38       }
39       else {
40         window.location = data.redirect;
41       }
42     },
43     'json'
44   );
45   return false;
46 });
47 </script>

```

users/show.php

```

1  <h1 class="page-header">Edit my account</h1>
2
3  <div id="users">
4    <?= $this->load->view('users/show_users') ?>
5  </div>
6
7  <script type="text/javascript">
8  /*!-- Set the action for user deletion -->*/
9  $('body').on('click', '.delete', function () {
10    if (confirm('Do you really want to delete the user "' + $(this).closest('tr').find('.mail').val() + '?'))
11      $.get(
12        $(this).attr('href'),
13        function (data) {
14          window.location = data.redirect;
15        },
16        'json'
17      );
18    }
19    return false;
20  });
21
22 /*!-- Set the action for user edition -->*/
23 $('body').on('click', '.edit', function () {
24   if (confirm('Do you really want to edit the user "' + $(this).closest('tr').find('.mail').val() + '?')) {
25     var post_data = {
26       '<?= $this->security->get_csrf_token_name() ?>': '<?= $this->security->get_csrf_hash() ?>',
27       'first_name': $(this).closest('tr').find('.first_name').val(),

```

```

28     'last_name':      $(this).closest('tr').find('.last_name').val(),
29     'mail':           $(this).closest('tr').find('.mail').val(),
30     'secret':         $(this).closest('tr').find('.secret').val()
31   };
32   if ($(this).closest('tr').find('.admin_platform').length > 0) {
33     post_data['admin_platform'] = (
34       $(this).closest('tr').find('.admin_platform').attr('checked')?true:false
35     );
36   }
37   $.post(
38     $(this).attr('href'),
39     post_data,
40     function (data) {
41       if (data.errors) {
42         $('#users_errors').empty();
43         $('#users_errors').removeClass('hidden');
44         $('#users_errors').append(data.errors);
45       }
46       else {
47         window.location = data.redirect;
48       }
49     },
50     'json'
51   );
52 }
53 return false;
54 });
55 </script>

```

users/show_users.php

```

1  <div id="users_errors" class="alert alert-error hidden"></div>
2
3  <table class="table table-bordered table-condensed">
4    <thead>
5      <tr>
6        <th class="span3">Id</th>
7        <th>First name</th>
8        <th>Last name</th>
9        <th>Email</th>
10       <th>Secret</th>
11       <th class="span2"></th>
12     </tr>
13   </thead>
14   <tbody>
15     <tr>
16       <td><?= $user->_id ?></td>
17       <td><?= form_input(array('name' => 'first_name', 'value' => $user->first_name, 'class' => 'first_name')) ?></td>
18       <td><?= form_input(array('name' => 'last_name', 'value' => $user->last_name, 'class' => 'last_name')) ?></td>
19       <td><?= form_input(array('name' => 'mail', 'value' => $user->mail, 'class' => 'mail input-large')) ?></td>
20       <td><?= form_password(array('name' => 'secret', 'value' => '', 'class' => 'secret input-small')) ?></td>
21     <td>
22       <a class="edit" href="= site_url('users/edit/'.$user-&gt;_id) ?&gt;"&gt;&lt;button class="btn btn-mini btn-primary" type="button"&gt;Edit&lt;/button&gt;
23       &lt;a class="delete" href="<?= site_url('users/delete/'.$user-&gt;_id) ?&gt;"&gt;&lt;button class="btn btn-mini btn-primary" type="button"&gt;Delete&lt;/button&gt;
24     &lt;/td&gt;
25   &lt;/tr&gt;
26 &lt;/tbody&gt;
27 &lt;/table&gt;
28
29 &lt;?php if ($this-&gt;user-&gt;is_admin_platform()): ?&gt;
30
31 &lt;h3&gt;Edit other users&lt;/h3&gt;
32
33 &lt;?php if (empty($users)): ?&gt;
34
35 &lt;p&gt;There is currently no user to edit.&lt;/p&gt;
36
37 &lt;?php else: ?&gt;
38
39 &lt;table class="table table-bordered table-condensed table-striped"&gt;
40   &lt;thead&gt;
41     &lt;tr&gt;
</pre

```

```

42         <th class="span3">Id</th>
43         <th>First name</th>
44         <th>Last name</th>
45         <th>Email</th>
46         <th>Secret</th>
47         <th>Admin platform</th>
48         <th class="span2"></th>
49     </tr>
50   </thead>
51   <tbody>
52     <?php foreach ($users as $k => $user): ?>
53     <tr>
54       <td><?= $user->_id ?></td>
55       <td><?= form_input(array('name' => 'first_name', 'value' => $user->first_name, 'class' => 'first_name')) ?>
56       <td><?= form_input(array('name' => 'last_name', 'value' => $user->last_name, 'class' => 'last_name') ?>
57       <td><?= form_input(array('name' => 'mail', 'value' => $user->mail, 'class' => 'mail input-large')) ?>
58       <td><?= form_password(array('name' => 'secret', 'value' => '', 'class' => 'secret input-small')) ?>
59       <td><?= form_checkbox(array('name' => 'admin_platform', 'value' => 'yes', 'checked' => $user->admin_p
60       <td>
61         <a class="edit" href="= site_url('users/edit/'.$user-&gt;_id) ?&gt;"&gt;&lt;button class="btn btn-mini btn-p
62         &lt;a class="delete" href="<?= site_url('users/delete/'.$user-&gt;_id) ?&gt;"&gt;&lt;button class="btn btn-mini b
63       &lt;/td&gt;
64     &lt;/tr&gt;
65     &lt;?php endforeach; ?&gt;
66   &lt;/tbody&gt;
67 &lt;/table&gt;
68
69 &lt;?php endif; ?&gt;
70
71 &lt;?= $this-&gt;load-&gt;view('users/add_user_form'); ?&gt;
72
73 &lt;?php endif; ?&gt;
74
75 &lt!-- &lt;script type="text/javascript"&gt;
76 $(document).ready(function() {
77   setTimeout(
78     function() {
79       $.get(
80         "&lt;?= site_url('users/refresh') ?&gt;",
81         function(data) {
82           $('#users').html(data);
83           },
84           'html'
85         );
86       },
87       &lt;?= $this-&gt;config-&gt;item('users_refresh') * 1000 ?&gt;
88     );
89   });
90 &lt;/script&gt; --&gt;</pre

```

6.14 OSCIED - FIMS Scripts

6.14.1 common.sh.lu-dep

```

1 set -o nounset # will exit if an uninitialized variable is used
2
3 # Prevent importing N times the following (like C++ .h : #ifndef ... #endif)
4 if ! charmsCommonImported 2>/dev/null; then
5
6 # Constants =====
7
8 SCRIPTS_PATH=$(pwd)
9 BASE_PATH=$(dirname "$SCRIPTS_PATH")
10 FILES_PATH=$BASE_PATH'/scripts_files'
11 GENERATED_PATH=$BASE_PATH'/generated'
12 SPECIFICATION_PATH=$BASE_PATH'/specification'
13 FIMS_SPECS_XSD_PATH=$SPECIFICATION_PATH'/FIMS_V1_0_7'
14 FIMS_SPECS_JAVA_PATH=$GENERATED_PATH'/FIMS_V1_0_7_java'
15 FIMS_SPECS_PHP_PATH=$GENERATED_PATH'/FIMS_V1_0_7_php'
```

```

16 TOOLS_PATH=$BASE_PATH'/tools'
17 SOAP_WEB_PATH=$BASE_PATH'/www'
18 APACHE_WEB_PATH='/var/www/tests'
19
20 # Configuration =====
21
22 # Utilities =====
23
24 setSetting()
25 {
26     if [ $# -eq 4 ]; then key="\${$2\[$3\]}"; value=$4
27     elif [ $# -eq 5 ]; then key="\${$2\[$3\]}\[$4\]"; value=$5
28     else xecho "Usage: ${basename $0}.setSetting file variable (category) name value"
29     fi
30
31     $sudo sed -i "s<$key = .*<$key = '$value';<" "$1"
32     $sudo grep -q "$key = '$value';" "$1" && return $true || return $false
33 }
34
35 charmsCommonImported()
36 {
37     echo > /dev/null
38 }
39 fi

```

6.14.2 menu.sh

```

1 . ./common.sh
2
3 main()
4 {
5     listing=/tmp/$$.list
6     tmpfile=/tmp/$$
7     trap "$sudo rm -f '$listing' '$tmpfile' 2>/dev/null" INT TERM EXIT
8
9     autoInstall dialog dialog
10
11    # Initialize main menu
12    while true
13    do
14        $DIALOG --backtitle 'OSCIED Operations with FIMS' \
15            --menu 'Please select an operation' 0 0 0 \
16            gettools 'download or update useful tools' \
17            install 'install soap test projects' \
18            uninstall 'uninstall soap test projects' \
19            service 'control apache service' 2> $tmpfile
20
21        retval=$?
22        operation=$(cat $tmpfile)
23
24        # Main menu exit door
25        [ $retval -ne 0 -o ! "$operation" ] && break
26
27        techo "Execute operation $operation"
28        eval $operation
29
30        # Main menu pause
31        [ $retval -eq 0 ] && pause
32    done
33 }
34
35 gettools()
36 {
37     autoInstall lftp lftp
38
39     #FIXME useless & not working, huh ! :-( lftp -f 'fims-mirror.sh'
40
41     cd "$TOOLS_PATH" || exit 1
42
43     jaxb='jaxb-ri-2.2.6'
44     zip="$jaxb.zip"

```

```

45  if [ ! -d "$jaxb" ]; then
46    wget -N http://jaxb.java.net/2.2.6/$zip || xecho "Unable to download $zip"
47    unzip $zip
48    rm -f $zip
49  fi
50
51  cd $jaxb/bin/
52  mkdir -p "$FIMS_SPECS_JAVA_PATH"
53  sh xjc.sh -d "$FIMS_SPECS_JAVA_PATH/" "$FIMS_SPECS_XSD_PATH/"
54  cd ../../
55
56  wsdl2php='wsdl2phpgenerator-master'
57  zip="$wsdl2php.zip"
58  if [ ! -d "$wsdl2php" ]; then
59    unzip $zip
60  fi
61
62  cd $wsdl2php
63  find "$FIMS_SPECS_XSD_PATH" -type f -name "*.wsdl" | while read wsdl
64  do
65    name=$(basename "$wsdl" .wsdl)
66    mecho "Generating php : $name"
67    mkdir -p "$FIMS_SPECS_PHP_PATH/$name"
68    ./wsdl2php -i "$wsdl" -o "$FIMS_SPECS_PHP_PATH/$name"
69  done
70  cd ../../
71 }
72
73 install()
74 {
75   pecho 'Install and configure Apache + PHP'
76   eval $install apache2 php5 php5-cli php5-gd php-soap || xecho 'Unable to install Apache + PHP'
77
78   pecho 'Copy SOAP Test Projects'
79   $sudo rsync -rtvh --progress --delete --exclude=.svn "$SOAP_WEB_PATH/" "$APACHE_WEB_PATH/"
80   $sudo chown www-data:www-data "$APACHE_WEB_PATH" -R
81   $sudo find "$APACHE_WEB_PATH" -type f -name "*.php" -exec chmod +x {} \;
82   eval $service apache2 restart
83 }
84
85 uninstall()
86 {
87   pecho 'Uninstall Apache + PHP'
88   eval $purge apache2 php5 php5-cli php5-gd php-soap apache2.2-common
89   eval $autoremove
90   # /etc/apache2/ /var/log/apache2/
91   $sudo rm -rf "$APACHE_WEB_PATH/"
92   $sudo mkdir /var/www/ 2>/dev/null
93 }
94
95 service()
96 {
97   # Initialize service menu
98   while true
99   do
100     $DIALOG --backtitle 'OSCIED Operations with FIMS > Service' \
101       --menu 'Please select an operation' 0 0 0 \
102       restart 'restart apache2 service' \
103       stop   'stop apache2 service' \
104       status 'get apache2 status' 2> $tmpfile
105
106     retval=$?
107     operation=$(cat $tmpfile)
108
109     # Service menu exit door
110     [ $retval -ne 0 -o ! "$operation" ] && break
111
112     techo "Execute operation $operation"
113     eval $service apache2 $operation || xecho "Unable to $operation Apache service"
114
115     # Service menu pause
116     [ $retval -eq 0 ] && pause
117   done

```

```
118 }
119
120 main "$@"
```

6.15 OSCIED - OpenStack Nodes Scripts

6.15.1 files/common

ntp.conf.hack

```
# /etc/ntp.conf, configuration for ntpd; see ntp.conf(5) for help

driftfile /var/lib/ntp/ntp.drift

# Enable this if you want statistics to be logged.
#statsdir /var/log/ntpstats/

statistics loopstats peerstats clockstats
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable

# Specify one or more NTP servers.

# Use servers from the NTP Pool Project. Approved by Ubuntu Technical Board
# on 2011-02-08 (LP: #104525). See http://www.pool.ntp.org/join.html for
# more information.
server 0.ubuntu.pool.ntp.org
server 1.ubuntu.pool.ntp.org
server 2.ubuntu.pool.ntp.org
server 3.ubuntu.pool.ntp.org
server ntp.ubuntu.com iburst
server 127.127.1.0
fudge 127.127.1.0 stratum 10

# Access control configuration; see /usr/share/doc/ntp-doc/html/accept.html for
# details. The web page <http://support.ntp.org/bin/view/Support/AccessRestrictions>
# might also be helpful.
#
# Note that "restrict" applies to both servers and clients, so a configuration
# that might be intended to block requests from certain clients could also end
# up blocking replies from your own upstream servers.

# By default, exchange time with everybody, but don't allow configuration.
restrict -4 default kod notrap nomodify nopeer noquery
restrict -6 default kod notrap nomodify nopeer noquery

# Local users may interrogate the ntp server more closely.
restrict 127.0.0.1
restrict ::1

# Clients from this (example!) subnet have unlimited access, but only if
# cryptographically authenticated.
#restrict 192.168.123.0 mask 255.255.255.0 notrust

# If you want to provide time to your local subnet, change the next line.
# (Again, the address is an example only.)
#broadcast 192.168.123.255
```

```
# If you want to listen to time broadcasts on your local subnet, de-comment the
# next lines. Please do this only if you trust everybody on the network!
#disable auth
#broadcastclient
```

qemu.conf.patch

```
*** qemu.conf.orig      2012-11-23 00:31:58.299027206 +0100
--- qemu.conf          2012-11-23 00:32:17.391516012 +0100
*****
*** 199,210 ****
    # This will only need setting if newer QEMU suddenly
    # wants some device we don't already know about.
    #
! #cgroup_device_acl = [
! #    "/dev/null", "/dev/full", "/dev/zero",
! #    "/dev/random", "/dev/urandom",
! #    "/dev/ptmx", "/dev/kvm", "/dev/kqemu",
! #    "/dev/rtc", "/dev/hpet"
! #]

# The default format for Qemu/KVM guest save images is raw; that is, the
--- 199,210 ---
# This will only need setting if newer QEMU suddenly
# wants some device we don't already know about.
# PATCHED_BY_OSCIED
! cgroup_device_acl = [
!     "/dev/null", "/dev/full", "/dev/zero",
!     "/dev/random", "/dev/urandom",
!     "/dev/ptmx", "/dev/kvm", "/dev/kqemu",
!     "/dev/rtc", "/dev/hpet", "/dev/net/tun"
! ]
```

The default format for Qemu/KVM guest save images is raw; that is, the

cinder/cinder.conf.append

```
1 # PATCHED_BY_OSCIED
2 rabbit_host=TODO
3 rabbit_password=TODO
4 sql_connection=TODO
```

dashboard/local_settings.py.append

```
1 # PATCHED_BY_OSCIED
2 DATABASES = {
3     'default': {
4         'ENGINE': 'django.db.backends.mysql',
5         'NAME': 'dashboard',
6         'USER': 'DASH_USER',
7         'PASSWORD': 'DASH_PASSWORD',
8         'HOST': 'CONTROLLER_PRIVATE_IP',
9         'default-character-set': 'utf8'
10    },
11 }
```

nova/api-paste.ini.hack

```
#####
# Metadata #
#####
[composite:metadata]
use = egg:Paste#urlmap
/: meta
[pipeline:meta]
pipeline = ec2faultwrap logrequest metaapp
[app:metaapp]
paste.app_factory = nova.api.metadata.handler:MetadataRequestHandler.factory

#####
# EC2 #
#####
[composite:ec2]
use = egg:Paste#urlmap
/services/Cloud: ec2cloud
[composite:ec2cloud]
use = call:nova.api.auth:pipeline_factory
noauth = ec2faultwrap logrequest ec2noauth cloudrequest validator ec2executor
keystone = ec2faultwrap logrequest ec2keystoneauth cloudrequest validator
ec2executor
[filter:ec2faultwrap]
paste.filter_factory = nova.api.ec2:FaultWrapper.factory
[filter:logrequest]
paste.filter_factory = nova.api.ec2:RequestLogging.factory
[filter:ec2lockout]
paste.filter_factory = nova.api.ec2:Lockout.factory
[filter:ec2keystoneauth]
paste.filter_factory = nova.api.ec2:EC2KeystoneAuth.factory
[filter:ec2noauth]
paste.filter_factory = nova.api.ec2>NoAuth.factory
[filter:cloudrequest]
controller = nova.api.ec2.cloud.CloudController
paste.filter_factory = nova.api.ec2:Requestify.factory
[filter:authorizer]
paste.filter_factory = nova.api.ec2:Authorizer.factory
[filter:validator]
paste.filter_factory = nova.api.ec2:Validator.factory
[app:ec2executor]
paste.app_factory = nova.api.ec2:Executor.factory

#####
# Openstack #
#####
[composite:osapi_compute]
use = call:nova.api.openstack.urlmap:urlmap_factory
/: oscomputeversions
/v1.1: openstack_compute_api_v2
/v2: openstack_compute_api_v2
[composite:osapi_volume]
use = call:nova.api.openstack.urlmap:urlmap_factory
/: osvolumeversions
/v1: openstack_volume_api_v1
[composite:openstack_compute_api_v2]
use = call:nova.api.auth:pipeline_factory
noauth = faultwrap sizelimit noauth ratelimit osapi_compute_app_v2
keystone = faultwrap sizelimit authtoken keystonecontext ratelimit
osapi_compute_app_v2
keystone_nolimit = faultwrap sizelimit authtoken keystonecontext
osapi_compute_app_v2
```

```
[composite:openstack_volume_api_v1]
use = call:nova.api.auth:pipeline_factory
noauth = faultwrap sizelimit noauth ratelimit osapi_volume_app_v1
keystone = faultwrap sizelimit authtoken keystonecontext ratelimit
osapi_volume_app_v1
keystone_nolimit = faultwrap sizelimit authtoken keystonecontext
osapi_volume_app_v1
[filter:faultwrap]
paste.filter_factory = nova.api.openstack:FaultWrapper.factory
[filter:noauth]
paste.filter_factory = nova.api.openstack.auth:NoAuthMiddleware.factory
[filter:ratelimit]
paste.filter_factory = nova.api.openstack.compute.
limits:RateLimitingMiddleware.factory
[filter:sizelimit]
paste.filter_factory = nova.api.sizelimit:RequestBodySizeLimiter.factory
[app:osapi_compute_app_v2]
paste.app_factory = nova.api.openstack.compute:APIRouter.factory
[pipeline:oscomputeversions]
pipeline = faultwrap oscomputeversionapp
[app:osapi_volume_app_v1]
paste.app_factory = nova.api.openstack.volume:APIRouter.factory
[app:oscomputeversionapp]
paste.app_factory = nova.api.openstack.compute.versions:Versions.factory
[pipeline:osvolumeversions]
pipeline = faultwrap osvolumeversionapp

#####
# Shared #
#####
[filter:keystonecontext]
paste.filter_factory = nova.api.auth:NovaKeystoneContext.factory
[filter:authtoken]
paste.filter_factory = keystone.middleware.auth_token:filter_factory
```

nova/nova.conf.template

```
1 [DEFAULT]
2
3 # Miscellaneous
4 logdir=/var/log/nova
5 state_path=/var/lib/nova
6 lock_path=/var/lock/nova
7 root_helper=sudo nova-rootwrap /etc/nova/rootwrap.conf
8 allow_admin_api=True
9 allow_resize_to_same_host=True
10 verbose=True
11 enabled_apis=ec2,osapi_compute,metadata
12 osapi_compute_extension=nova.api.openstack.compute.contrib.standard_extensions
13
14 # Database (MySQL)
15 sql_connection=mysql://nova:password@CONTROLLER_PRIVATE_IP/nova?charset=utf8
16
17 # Message Queue (RabbitMQ)
18 rabbit_host=CONTROLLER_PRIVATE_IP
19 rabbit_password=password
20
21 # VM Console -> Browser (novnc)
22 novnc_enable=True
23 novncproxy_base_url=http://CONTROLLER_PUBLIC_IP:6080/vnc_auto.html
24 xvpxvncproxy_base_url=http://CONTROLLER_PUBLIC_IP:6081/console
25 novncproxy_port=6080
26 vncserver_proxyclient_address=HOST_PRIVATE_IP
27 vncserver_listen=HOST_PRIVATE_IP
28 my_ip=HOST_PRIVATE_IP
29
```

```

30 # Authentication
31 use_deprecated_auth=False
32 auth_strategy=keystone
33 keystone_ec2_url=http://CONTROLLER_PRIVATE_IP:5000/v2.0/ec2tokens
34
35 # OpenStack APIs
36 cc_host=CONTROLLER_PRIVATE_IP
37 s3_host=CONTROLLER_PRIVATE_IP
38 ec2_host=CONTROLLER_PRIVATE_IP
39 nova_url=http://CONTROLLER_PRIVATE_IP:8774/v1.1/
40 ec2_url=http://CONTROLLER_PRIVATE_IP:8773/services/Cloud
41 ec2_private_dns_show_ip=True
42 dmz_cidr=169.254.169.254/32
43 ec2_dmz_host=CONTROLLER_PRIVATE_IP
44 metadata_host=CONTROLLER_PRIVATE_IP
45 metadata_listen=0.0.0.0
46
47 # Scheduler (nova-scheduler)
48 scheduler_driver=nova.scheduler.simple.SimpleScheduler
49
50 # Compute Service (nova-compute->libvirt->kvm)
51 compute_driver=nova.virt.libvirt.LibvirtDriver
52 libvirt_type=kvm
53 libvirt_cpu_mode=none
54 instance_name_template=instance-%08x
55 instances_path=/var/lib/nova/instances
56 api_paste_config=/etc/nova/api-paste.ini
57 libvirt_use_virtio_for_bridges=True
58
59 # Image Service (Glance)
60 glance_api_servers=CONTROLLER_PRIVATE_IP:9292
61 image_service=nova.image.glance.GlanceImageService
62
63 # Volume Service (Cinder)
64 volume_api_class=nova.volume.cinder.API
65 osapi_volume_listen=CONTROLLER_PRIVATE_IP
66 osapi_volume_listen_port=9000
67
68 # Network Service (quantum)
69 network_api_class=nova.network.quantumv2.api.API
70 quantum_url=http://CONTROLLER_PRIVATE_IP:9696
71 quantum_auth_strategy=keystone
72 quantum_admin_tenant_name=service
73 quantum_admin_username=quantum
74 quantum_admin_password=password
75 quantum_admin_auth_url=http://CONTROLLER_PRIVATE_IP:35357/v2.0
76 force_dhcp_release=True
77 linuxnet_interface_driver=nova.network.linux_net.LinuxOVSIInterfaceDriver
78 firewall_driver=nova.virt.libvirt.firewall.IptablesFirewallDriver
79
80 # Vif-plugging, using Nova Security Groups
81 libvirt_vif_driver=nova.virt.libvirt.vif.LibvirtHybridOVSBridgeDriver
82
83 # Restore VMs on boot
84 resume_guests_state_on_host_boot=True

```

nova/nova-compute.conf.hack

```

1 [DEFAULT]
2 libvirt_type=kvm
3 libvirt_ovs_bridge=br-int
4 libvirt_vif_type=ethernt
5 libvirt_vif_driver=nova.virt.libvirt.vif.LibvirtHybridOVSBridgeDriver
6 libvirt_use_virtio_for_bridges=True

```

6.15.2 files/private/keystone

endpoints

```
# Region Service Public Internal Admin
RegionOne;identity;5000/v2.0;5000/v2.0;35357/v2.0
RegionOne;compute;8774/v2/%(tenant_id)s;8774/v2/%(tenant_id)s;8774/v2/%(tenant_id)s
RegionOne;image;9292/v1;9292/v1;9292/v1
RegionOne;object-store;8080/v1/AUTH_%(tenant_id)s;8080/v1/AUTH_%(tenant_id)s;8080/v1/AUTH_%(tenant_id)s
RegionOne;volume;8776/v1/%(tenant_id)s;8776/v1/%(tenant_id)s;8776/v1/%(tenant_id)s
RegionOne;network;9696/;9696/;9696/
RegionOne;ec2;8773/services/Cloud;8773/services/Cloud;8773/services/Admin
```

roles

```
1 # Name Description
2 admin;
3 Member;
```

services

```
1 # Service Type Description
2 keystone;identity;Identity Service
3 nova;compute;Nova Compute Service
4 glance;image;Image Service
5 swift;object-store;Object Store Service
6 cinder;volume;Block Storage Service
7 quantum;network;Network Service
8 ec2;ec2;EC2 Compatibility Layer
```

tenants

```
1 # Name Description
2 demo;
3 service;
```

users

```
1 # Name Password E-Mail Description
2 on;admin;IMh70Cf1Ry155ACM;david.fischer.ch@gmail.com;OSCIED Administrator
3 on;demo;5Si4ioGRQr7DyUHd;;OSCIED Demo User
4 on;cinder;P9c0oC3duT6HNVMw;;Block Storage Service User
5 off;dashboard;lowNPP3mYXUbYKbX;Dashboard Service User (Only for MySQL)
6 on;glance;O2SakqjhaJ8cdt2p;;Image Service User
7 off;keystone;2FeCJ95Mm434Io5k;;Identity Service User (Only for MySQL)
8 on;nova;Nzseo3CYI28uuIBo;;Compute Service User
9 on;swift;97TWEEdM8vEKA2HAU;;Object Storage Service User
10 on;quantum;3OYd7FVp87hEqrTZ;;Network Service User
```

usersRoles

```
1 # Tenant User Role Comment
2 demo;admin;admin;
3 demo;demo;Member;
4 service;cinder;admin;
5 service;glance;admin;
6 service;nova;admin;
7 service;quantum;admin;
8 service;swift;admin;
```

6.15.3 files/testing

MobileStack-A425-D3.conf

```

1 #!/bin/bash
2 # WARNING: http://forums.openstack.org/viewtopic.php?f=23&t=1397
3 # "(...)" the password for the keystone user had special characters, after
4 # replace with a new an less complex password, everything works fine. (...)"
5
6 TENANT_NAME='demo'
7 CONTROLLER_PUBLIC_IP=$HOST_PUBLIC_IP
8 CONTROLLER_PRIVATE_IP=$HOST_PRIVATE_IP
9 ADMIN_TOKEN='9HYXC3UYCNWZY08'
10 MYSQL_ROOT_PASS='LMbpL05svS6FsASP'
11 RABBIT_PASS='iL6eicw4XxzwO37Q'
12 CINDER_VG='cinder-volumes'
```

MobileStack-A425-D3.interfaces

```

auto lo
iface lo inet loopback

# START PUBLIC
auto eth0
iface eth0 inet dhcp
address 10.10.3.17
netmask 255.255.255.0
gateway 10.10.1.1
dns-nameservers 8.8.8.8
# END PUBLIC

# START PRIVATE
auto eth1
iface eth1 inet static
address 192.168.1.17
netmask 255.255.255.0
# END PRIVATE
```

6.15.4 scripts/common.sh.lu-dep

```

1 set -o nounset # will exit if an uninitialized variable is used
2
3 # Prevent importing N times the following (like C++ .h : #ifndef ... #endif)
4 if ! cloudCommonImported 2>/dev/null; then
5
6 # Constants =====
7
8 SCRIPTS_PATH=$(pwd)
9 BASE_PATH=$(dirname "$SCRIPTS_PATH")
10 FILES_PATH=$BASE_PATH'/files'
11 COMMON_FILES_PATH=$FILES_PATH'/common'
12 PRIVATE_FILES_PATH=$FILES_PATH'/private'
13 NTP_CONF_FILE='/etc/ntp.conf'
14 NTP_HACK_FILE=$COMMON_FILES_PATH'/ntp.conf.hack'
15 NETWORK_CONF_FILE='/etc/network/interfaces'
16 SYSCtrl_CONF_FILE='/etc/sysctl.conf'
17 LIBVIRTD_CONF_FILE='/etc/libvirt/libvirtd.conf'
18 LIBVIRTD_INIT_FILE='/etc/init/libvirt-bin.conf'
19 LIBVIRTD_DEF_FILE='/etc/default/libvirt-bin'
20 QEMU_CONF_FILE='/etc/libvirt/qemu.conf'
21 QEMU_PATCH_FILE=$COMMON_FILES_PATH'/qemu.conf.patch'
22 KEYSTONE_FILES_PATH=$PRIVATE_FILES_PATH'/keystone'
23 KEYSTONE_TENANTS_FILE=$KEYSTONE_FILES_PATH'/tenants'
24 KEYSTONE_USERS_FILE=$KEYSTONE_FILES_PATH'/users'
25 KEYSTONE_ROLES_FILE=$KEYSTONE_FILES_PATH'/roles'
26 KEYSTONE_USERS_ROLES_FILE=$KEYSTONE_FILES_PATH'/usersRoles'
```

```

27 KEYSTONE_SERVICES_FILE=$KEYSTONE_FILES_PATH'/services'
28 KEYSTONE_ENDPOINTS_FILE=$KEYSTONE_FILES_PATH'/endpoints'
29 KEYSTONE_CONF_FILE='/etc/keystone/keystone.conf'
30 GLANCE_API_CONF_FILE='/etc/glance/glance-api.conf'
31 GLANCE_API_PASTE_FILE='/etc/glance/glance-api-paste.ini'
32 GLANCE_REGISTRY_CONF_FILE='/etc/glance/glance-registry.conf'
33 GLANCE_REGISTRY_PASTE_FILE='/etc/glance/glance-registry-paste.ini'
34 NOVA_FILES_PATH=$COMMON_FILES_PATH'/nova'
35 NOVA_CONF_FILE='/etc/nova/nova.conf'
36 NOVA_TEMPL_FILE=$NOVA_FILES_PATH'/nova.conf.template'
37 NOVA_API_PASTE_FILE='/etc/nova/api-paste.ini'
38 NOVA_API_HACK_FILE=$NOVA_FILES_PATH'/api-paste.ini.hack'
39 NOVA_COMPUTE_CONF_FILE='/etc/nova/nova-compute.conf'
40 NOVA_COMPUTE_HACK_FILE=$NOVA_FILES_PATH'/nova-compute.conf.hack'
41 QUANTUM_CONF_FILE='/etc/quantum/quantum.conf'
42 QUANTUM_PASTE_FILE='/etc/quantum/api-paste.ini'
43 QUANTUM_DHCP_CONF_FILE='/etc/quantum/dhcp_agent.ini'
44 QUANTUM_L3_CONF_FILE='/etc/quantum/l3_agent.ini'
45 QUANTUM_OVS_SWITCH_CONF_FILE='/etc/quantum/plugins/openvswitch/ovs_quantum_plugin.ini'
46 ISCSI_CONF_FILE='/etc/default/iscsitarget'
47 CINDER_FILES_PATH=$COMMON_FILES_PATH'/cinder'
48 CINDER_CONF_FILE='/etc/cinder/cinder.conf'
49 CINDER_APPEND_FILE=$CINDER_FILES_PATH'/cinder.conf.append'
50 CINDER_PASTE_FILE='/etc/cinder/api-paste.ini'
51 DASHBOARD_FILES_PATH=$COMMON_FILES_PATH'/dashboard'
52 DASHBOARD_CONF_FILE='/etc/openstack-dashboard/local_settings.py'
53 DASHBOARD_APPEND_FILE=$DASHBOARD_FILES_PATH'/local_settings.py.append'
54 QUANTUM_BUG_FILE='/usr/lib/python2.7/dist-packages/quantum/agent/linux/iptables_manager.py'
55
56 # Configuration =====
57
58 host=$(hostname)
59 HOST_CONFIG_FILE=$(find "$FILES_PATH" -type f -name "$host.conf")
60 HOST_NETWORK_FILE=$(find "$FILES_PATH" -type f -name "$host.interfaces")
61
62 getInterface()
63 {
64     if [ $# -ne 2 ]; then
65         echo "[BUG] Usage: $(basename $0).getInterface type file" >&2
66         exit 1
67     fi
68     content=$(sed 's:\n: :g' "$2")
69     AZ='[A-Z ]*'
70     REPLY_IFACE=$(expr match "$content" ".*START$AZ$1.*auto *([a-z0-9:]*)).*END$AZ$1")
71     REPLY_IP=$(expr match "$content" ".*START$AZ$1.*address *([0-9\.:]*)).*END$AZ$1.*")
72     REPLY_MASK=$(expr match "$content" ".*START$AZ$1.*netmask *([0-9\.:]*)).*END$AZ$1.*")
73     REPLY_GW=$(expr match "$content" ".*START$AZ$1.*gateway *([0-9\.:]*)).*END$AZ$1.*")
74     if [ "$REPLY_IFACE" -a "$REPLY_IP" -a "$REPLY_MASK" ]
75     then return 0
76     else return 1
77     fi
78 }
79
80 if [ ! -f "$HOST_CONFIG_FILE" ]; then
81     echo "Computer '$host' configuration file not found !" >&2
82     exit 1
83 fi
84 if [ ! -f "$HOST_NETWORK_FILE" ]; then
85     echo "Computer '$host' network file not found !" >&2
86     exit 1
87 fi
88
89 if ! getInterface 'PUBLIC' "$HOST_NETWORK_FILE"; then
90     echo 'Unable to detect public network interface' >&2
91     exit 1
92 fi
93 HOST_PUBLIC_IFACE=$REPLY_IFACE
94 HOST_PUBLIC_IP=$REPLY_IP
95 HOST_PUBLIC_MASK=$REPLY_MASK
96 HOST_PUBLIC_GW=$REPLY_GW
97
98 if [ ! "$HOST_PUBLIC_GW" ]; then
99     echo 'Unable to detect public network gateway' >&2

```

```

100    exit 1
101
102
103 if ! getInterface 'PRIVATE' "$HOST_NETWORK_FILE"; then
104     echo 'Unable to detect private network interface' >&2
105     exit 1
106 fi
107 HOST_PRIVATE_IFACE=$REPLY_IFACE
108 HOST_PRIVATE_IP=$REPLY_IP
109 HOST_PRIVATE_MASK=$REPLY_MASK
110 .
111 . "$HOST_CONFIG_FILE"
112
113 # FIXME a better way to handle urls ?
114 CONTROLLER_AUTHZ_URL="http://$CONTROLLER_PUBLIC_IP:5000/v2.0"
115 CONTROLLER_ADMIN_URL="http://$CONTROLLER_PRIVATE_IP:35357/v2.0"
116
117 # Utilities =====
118
119 setSetting()
120 {
121     if [ $# -ne 3 -a $# -ne 4 ]; then
122         xecho "Usage: $(basename $0).setSetting file enabled name [value]"
123     fi
124     local toggle=''
125     if [ $2 -eq $false ]; then toggle='#'; fi
126     if [ $# -eq 3 ]; then
127         $sudo sed -i "s<[# ]*$3<$toggle$3<" "$1"
128         $sudo grep -q "$toggle$3" "$1" && return $true || return $false
129     elif [ $# -eq 4 ]; then
130         $sudo sed -i "s<[# ]*$3 *= *.*<$toggle$3=$4<" "$1"
131         $sudo grep -q "$toggle$3=$4" "$1" && return $true || return $false
132     fi
133 }
134
135 parseId()
136 {
137     id=$(eval "$@" | grep ' id ')
138     id=$(expr match "$id" ".*\([0-9a-z]\{32\}\).*")
139     if [ ! "$id" ]; then return $false; else return $true; fi
140 }
141
142 getId()
143 {
144     if [ $# -ne 2 ]; then
145         xecho "Usage: $(basename $0).getId category name"
146     fi
147     local a=$ADMIN_TOKEN
148     local b=$CONTROLLER_ADMIN_URL
149     id=$(keystone --token $a --endpoint $b ${1}-list | grep " $2 ")
150     id=$(expr match "$id" ".*\([0-9a-z]\{32\}\).*")
151     if [ ! "$id" ]; then return $false; else return $true; fi
152 }
153
154 getPass()
155 {
156     if [ $# -ne 1 ]; then
157         xecho "Usage: $(basename $0).getPass name"
158     fi
159
160     savedIFS=$IFS
161     IFS=';'
162     while read enabled name pass mail description
163     do
164         if [ "$name" = "$1" ]; then return $true; fi
165     done < "$KEYSTONE_USERS_FILE"
166     IFS=$savedIFS
167     return $false
168 }
169
170 tenantCreate()
171 {
172     if [ $# -ne 1 ]; then

```

```

173     xecho "Usage: $(basename $0).tenantCreate name"
174 fi
175 keystone --token $ADMIN_TOKEN --endpoint $CONTROLLER_ADMIN_URL \
176     tenant-create --name "$1"
177 getId 'tenant' "$1"
178 }
179
180 userCreate()
181 {
182     if [ $# -ne 4 ]; then
183         xecho "Usage: $(basename $0).userCreate name pass email description"
184     fi
185     # FIXME add user description
186     keystone --token $ADMIN_TOKEN --endpoint $CONTROLLER_ADMIN_URL \
187         user-create --name "$1" --pass="$2" --email="$3"
188     getId 'user' "$1"
189 }
190
191 roleCreate()
192 {
193     if [ $# -ne 1 ]; then
194         xecho "Usage: $(basename $0).roleCreate name"
195     fi
196     keystone --token $ADMIN_TOKEN --endpoint $CONTROLLER_ADMIN_URL \
197         role-create --name "$1"
198     getId 'role' "$1"
199 }
200
201 serviceCreate()
202 {
203     if [ $# -ne 3 ]; then
204         xecho "Usage: $(basename $0).serviceCreate name type description"
205     fi
206     # FIXME add service description
207     keystone --token $ADMIN_TOKEN --endpoint $CONTROLLER_ADMIN_URL \
208         service-create --name "$1" --type="$2"
209     getId 'service' "$1"
210 }
211
212 endpointCreate()
213 {
214     if [ $# -ne 5 ]; then
215         xecho "Usage: $(basename $0).endpointCreate region serviceId publicUrl internalUrl adminUrl"
216     fi
217     keystone --token $ADMIN_TOKEN --endpoint $CONTROLLER_ADMIN_URL \
218         endpoint-create --region "$1" --service-id=$2 \
219         --publicurl=$3 --internalurl=$4 --adminurl=$5
220 }
221
222 userRoleAdd()
223 {
224     if [ $# -ne 3 ]; then
225         xecho "Usage: $(basename $0).userRoleAdd tenantId userId roleId"
226     fi
227     keystone --token $ADMIN_TOKEN --endpoint $CONTROLLER_ADMIN_URL \
228         user-role-add --tenant-id=$1 --user-id=$2 --role-id=$3
229 }
230
231 tokenGet()
232 {
233     if [ $# -eq 2 ]; then
234         keystone --os-username="$1" --os-password="$2" \
235             --os-auth-url=$CONTROLLER_ADMIN_URL token-get
236     elif [ $# -eq 3 ]; then
237         keystone --os-username="$1" --os-password="$2" --os-tenant-name="$3" \
238             --os-auth-url=$CONTROLLER_ADMIN_URL token-get
239     else
240         xecho "Usage: $(basename $0).tokenGet username password [tenantName]"
241     fi
242 }
243
244 imageCreateQcow2()
245 {

```

```

246 if [ $# -ne 7 ]; then
247     xecho "Usage: $(basename $0).imageCreateQcow2 username password tenantName public imageName imageFile imag
248 fi
249
250 archive=$(basename "$7")
251 name=$(basename "$archive" .tar.gz)
252 [ $4 -eq $true ] && public='--public' || public=''
253 tmp=$(mktemp -d)
254 cd $tmp || return $false
255 wget -N "$7"
256 tar -zxvf "$archive"
257 [ ! -f "$6" ] && return $false
258 id=$(glance --os-username="$1" --os-password="$2" --os-tenant-name="$3" \
259   --os-auth-url=$CONTROLLER_AUTHZ_URL image-create \
260   $public --name "$5" --container-format ovf --disk-format qcow2 < "$6")
261 id=$(expr match "$id" ".*\([0-9a-z]\{8\}-[0-9a-z]\{4\}-[0-9a-z]\{4\}-[0-9a-z]\{4\}-[0-9a-z]\{12\}\)\.*")
262 if [ ! "$id" ]; then return $false; else return $true; fi
263 }
264
265 imageCreateAll()
266 {
267 if [ $# -ne 4 ]; then
268     xecho "Usage: $(basename $0).imageCreateAll username password tenantName imageUrl"
269 fi
270
271 archive=$(basename "$4")
272 name=$(basename "$archive" .tar.gz)
273 mkdir /tmp/images 2>/dev/null
274 cd /tmp/images || return $false
275 wget -N "$4"
276 tar -zxvf "$archive" || return $false
277 imageCreateLinuxKernel "$1" "$2" "$3" "${name}-vmlinuz" || return $false; kernelId=$id
278 imageCreateLinuxRamdisk "$1" "$2" "$3" "${name}-loader" || return $false; ramdiskId=$id
279 imageCreateLinux "$1" "$2" "$3" "${name}.img" $kernelId $ramdiskId || return $false
280 }
281
282 imageCreateLinuxKernel()
283 {
284 if [ $# -ne 4 ]; then
285     xecho "Usage: $(basename $0).imageCreateLinuxKernel username password tenantName imagePath"
286 fi
287 id=$(glance --os-username="$1" --os-password="$2" --os-tenant-name="$3" \
288   --os-auth-url=$CONTROLLER_AUTHZ_URL image-create \
289   --name='tty-linux-kernel' --disk-format='aki' --container-format='aki' < "$4")
290 id=$(expr match "$id" ".*\([0-9a-z]\{8\}-[0-9a-z]\{4\}-[0-9a-z]\{4\}-[0-9a-z]\{4\}-[0-9a-z]\{12\}\)\.*")
291 if [ ! "$id" ]; then return $false; else return $true; fi
292 }
293
294 imageCreateLinuxRamdisk()
295 {
296 if [ $# -ne 4 ]; then
297     xecho "Usage: $(basename $0).imageCreateLinuxRamdisk username password tenantName imagePath"
298 fi
299 id=$(glance --os-username="$1" --os-password="$2" --os-tenant-name="$3" \
300   --os-auth-url=$CONTROLLER_AUTHZ_URL image-create \
301   --name='tty-linux-ramdisk' --disk-format='ari' --container-format='ari' < "$4")
302 id=$(expr match "$id" ".*\([0-9a-z]\{8\}-[0-9a-z]\{4\}-[0-9a-z]\{4\}-[0-9a-z]\{4\}-[0-9a-z]\{12\}\)\.*")
303 if [ ! "$id" ]; then return $false; else return $true; fi
304 }
305
306 imageCreateLinux()
307 {
308 if [ $# -ne 6 ]; then
309     xecho "Usage: $(basename $0).imageCreate username password tenantName imagePath kernelId ramdiskId"
310 fi
311 glance --os-username="$1" --os-password="$2" --os-tenant-name="$3" \
312   --os-auth-url=$CONTROLLER_AUTHZ_URL image-create \
313   --name='tty-linux' --disk-format='ami' --container-format='ami' \
314   --property kernel_id=$5 --property ramdisk_id=$6 < "$4"
315 }
316
317 cloudCommonImported()
318 {

```

```

319     echo > /dev/null
320 }
321 fi

```

6.15.5 scripts/cinder.sh

```

1  ./common.sh
2
3 main()
4 {
5     ok=$false
6     if [ $# -gt 0 ]; then
7         mode=$1
8         shift # remove mode parameter
9         eval "$mode" "$@"
10    fi
11
12   if [ $ok -eq $false ]; then
13       xecho "Usage: $(basename $0) mode\n\tmode : { install, installData, uninstall, service, try }"
14   fi
15 }
16
17 install()
18 {
19     if [ $# -ne 0 ]; then
20         xecho "Usage: $(basename $0) install"
21     fi
22     ok=$true
23
24     which vgdisplay > /dev/null || xecho 'LVM must be installed !'
25     $sudo vgdisplay "$CINDER_VG" > /dev/null || xecho "LVM VG $CINDER_VG not found !"
26
27     user='cinder'
28     getPass "$user" || xecho "Unable to get $user password"
29
30     pecho 'Install and configure Cinder Volume Service'
31     eval $install linux-headers-generic # required to compile kernel modules
32     eval $install cinder-common cinder-api cinder-scheduler cinder-volume iscsitarget \
33         iscsitarget-dkms open-iscsi python-cinder python-cinderclient tgt \
34         || xecho 'Unable to install Cinder Volume Service'
35
36     pecho 'Replace embedded sqlite database by MySQL'
37     $sudo rm /var/lib/cinder/cinder.sqlite 2>/dev/null
38     db="mysql://$user:$pass@$CONTROLLER_PRIVATE_IP/cinder?charset=utf8"
39
40     pecho 'Configure iSCSI'
41     a=$ISCSI_CONF_FILE
42     $sudo cp $a $a.old
43     $sudo sed -i 's/false/true/g' $a
44
45     a=$CINDER_PASTE_FILE
46     pecho "Configure Cinder ('basename $a')"
47     $sudo cp $a $a.old
48     setSetting $a $true 'service_host'      "$CONTROLLER_PUBLIC_IP" || xecho '1'
49     setSetting $a $true 'auth_host'        "$CONTROLLER_PRIVATE_IP" || xecho '2'
50     setSetting $a $true 'admin_tenant_name' 'service' || xecho '3'
51     setSetting $a $true 'admin_user'       "$user" || xecho '4'
52     setSetting $a $true 'admin_password'   "$pass" || xecho '5'
53
54     a=$CINDER_CONF_FILE
55     pecho "Configure Cinder ('basename $a')"
56     $sudo cp $a $a.old
57     if ! $sudo grep -q 'PATCHED_BY_OSCIED' $a; then
58         $sudo sh -c "cat '$CINDER_APPEND_FILE' >> $a"
59     fi
60     setSetting $a $true 'rabbit_host'      "$CONTROLLER_PRIVATE_IP" || xecho '1'
61     setSetting $a $true 'rabbit_password'  "$RABBIT_PASS" || xecho '2'
62     setSetting $a $true 'sql_connection'   "$db" || xecho '3'
63     setSetting $a $true 'iscsi_helper'     'ietadm' || xecho '4'
64     setSetting $a $true 'volume_group'    "$CINDER_VG" || xecho '5'
65
66     service restart

```

```

67
68     pecho 'Initialize the new Cinder Volume Service database'
69     $sudo cinder-manage db sync || xecho 'Unable to initialize database'
70
71     service restart
72 }
73
74 installData()
75 {
76     xecho 'NOT IMPLEMENTED'
77     #FIXME : to check cinder setup, create & list
78     try create --display_name test 1
79 }
80
81 uninstall()
82 {
83     if ! which cinder > /dev/null; then
84         xecho 'Cinder must be installed !'
85     fi
86
87     if [ $# -ne 0 ]; then
88         xecho "Usage: $(basename $0) uninstall"
89     fi
90     ok=$true
91
92     pecho 'Uninstall Cinder Volume Service'
93     eval $purge cinder-common cinder-api cinder-scheduler cinder-volume iscsitarget \
94         iscsitarget-dkms open-iscsi python-cinder python-cinderclient tgt
95     eval $autoremove
96     $sudo rm -rf /etc/cinder/ /var/lib/cinder/ /var/log/cinder/
97 }
98
99 service()
100 {
101     if ! which cinder > /dev/null; then
102         xecho 'Cinder must be installed !'
103     fi
104
105     if [ $# -ne 1 ]; then
106         xecho "Usage: $(basename $0) service operation\n\toperation : { start, stop, restart, ... }"
107     fi
108     ok=$true
109
110     op=$1
111     eval $service tgt      $op || xecho "Unable to $op tgt service"
112     eval $service iscsitarget $op || xecho "Unable to $op iscsitarget service"
113     eval $service cinder-api   $op || xecho "Unable to $op cinder api service"
114     eval $service cinder-volume  $op || xecho "Unable to $op cinder volume service"
115     eval $service cinder-scheduler $op || xecho "Unable to $op cinder scheduler service"
116 }
117
118 try()
119 {
120     if ! which cinder > /dev/null; then
121         xecho 'Cinder must be installed !'
122     fi
123     ok=$true
124
125     getPass 'admin' || xecho 'Unable to get admin password'
126     export OS_TENANT_NAME=$TENANT_NAME
127     export OS_USERNAME=admin
128     export OS_PASSWORD=$pass
129     export OS_AUTH_URL=$CONTROLLER_AUTHZ_URL
130     echo "Volumes"; cinder list
131     if [ $# -gt 0 ]; then
132         cinder "$@"
133     else
134         yesOrNo 1 'create a new volume'
135         if [ $REPLY -eq $true ]; then
136             readLine 'Volume name';      name=$CHOICE
137             readLine 'Volume size (GB)'; size=$CHOICE
138             cinder create --display_name "$name" "$size"
139         fi

```

```

140     fi
141 }
142
143 main "$@"

```

6.15.6 scripts/configure.bootstrap.sh

```

1  . ./common.sh
2
3  # Must be done before (bootstrap of the bootstrap)
4  # sudo apt-get install subversion
5  # svn co https://claire-et-david.dyndns.org/prog/OSCIED/components/cloud/nodes \
6  #   --username=oscied --password#####
7
8  pecho "Configure passwordless sudo for user $USER"
9  noppasswd="$USER ALL=(ALL) NOPASSWD: ALL"
10 if ! $Sudo grep -q "$noppasswd" /etc/sudoers
11 then $Sudo sh -c "echo '$noppasswd' >> /etc/sudoers"
12 else recho 'sudoers already modified'
13 fi
14
15 pecho 'Bootstrap : Add automatic call to setup script'
16 if [ ! -e ${HOME}/setup.sh ]
17 then ln -s $(pwd)/setup-helper.sh ${HOME}/setup.sh
18 else recho 'symlink to setup.sh already created'
19 fi
20 if ! $Sudo grep -q 'setup.sh' /etc/rc.local; then
21   boot="su $USER -c 'cd ${HOME} && sh setup.sh startup' &&\n"
22   $Sudo sed -i "s:^exit 0:$boot:" /etc/rc.local
23 else
24   recho 'call to setup script already added'
25 fi
26 $Sudo chmod +x /etc/rc.local
27
28 pecho 'Upgrade the system'
29 eval $update
30 eval $upgrade
31 eval $distupgrade
32 eval $autoremove
33
34 pecho 'Configure the bootloader'
35 # Fix ticket #42 : Sometimes grub doesn't start default "option" automatically.
36 fail='GRUB_RECORDFAIL_TIMEOUT=2'
37 if ! grep -q "$fail" /etc/default/grub; then
38   $Sudo sed -i "s:GRUB_TIMEOUT=.*:GRUB_TIMEOUT=2:" /etc/default/grub
39   $Sudo sh -c "echo '$fail' >> /etc/default/grub"
40   $Sudo update-grub
41 else
42   recho 'bootloader already modified'
43 fi
44
45 pecho 'Install some tools'
46 #sudo grep fr /usr/share/i18n/SUPPORTED
47 $Sudo locale-gen fr_CH.UTF-8
48 eval $install linux-headers-generic openssh-server
49
50 recho 'Job done'

```

6.15.7 scripts/configure.network.sh

```

1  . ./common.sh
2
3  pecho 'Configuring Network Interfaces'
4  a=$NETWORK_CONF_FILE
5  $Sudo mv $a $a.old
6  $Sudo cp "$HOST_NETWORK_FILE" $a
7  eval $service networking restart || xecho 'Unable to restart networking service'
8
9  $Sudo ifup $HOST_PUBLIC_IFACE

```

```

10 $sudo ifup $HOST_PRIVATE_IFACE
11
12 pecho 'Enable IP Forwarding'
13 setSetting $SYSCTL_CONF_FILE $true 'net.ipv4.ip_forward' '1' || xecho '1'
14 $sudo sysctl net.ipv4.ip_forward=1
15
16 pecho 'Install Network Utilities'
17 eval $install vlan bridge-utils || xecho 'Unable to install network utilities'
18
19 pecho 'Install and configure Network Time Protocol'
20 eval $install ntp || xecho 'Unable to install ntp'
21 $sudo cp -f "$NTP_HACK_FILE" "$NTP_CONF_FILE" # FIXME node cfg is different
22 eval $service ntp restart || xecho 'Unable to restart ntp service'
23
24 exit 0 # FIXME bridge for flat
25 if $sudo grep -q $NETWORK_BR_IFACE $NETWORK_CONF_FILE; then
26
27     pecho "Setup network bridge interface $NETWORK_BR_IFACE"
28
29     pecho 'Create VM Networking Bridge'
30     $sudo brctl addbr $NETWORK_BR_IFACE
31     # FIXME restart maybe not needed
32     eval $service networking restart || xecho 'Unable to restart networking service'
33
34     $sudo ifup $NETWORK_BR_IFACE
35 fi
36
37 # FIXME TODO FOR QUANTUM VLAN
38
39 #eval $install bridge-utils vlan || xecho 'Unable to install network utilities'
40 #pecho 'Load 8021q module to support VLAN'
41 #$sudo modprobe 8021q

```

6.15.8 scripts/configure.startup.sh

```

1 . ./common.sh
2
3 # FIXME do nothing !

```

6.15.9 scripts/dashboard.sh

```

1 . ./common.sh
2
3 main()
{
4     ok=$false
5     if [ $# -gt 0 ]; then
6         mode=$1
7         shift # remove mode parameter
8         eval "$mode" "$@"
9     fi
10
11     if [ $ok -eq $false ]; then
12         xecho "Usage: $(basename $0) mode\n\tmode : { install, uninstall, service }"
13     fi
14 }
15
16 install()
{
17     if [ $# -ne 0 ]; then
18         xecho "Usage: $(basename $0) install"
19     fi
20     ok=$true
21
22     user='dashboard'
23     getPassword "$user" || xecho "Unable to get $user password"
24
25     pecho 'Install and configure OpenStack Dashboard'
26     $sudo mkdir -p /var/log/apache2 2>/dev/null # fixes bug apache2 restart failed

```

```

29 eval $install memcached libapache2-mod-wsgi openstack-dashboard || \
30     xecho 'Unable to install OpenStack Dashboard'
31
32 eval $remove openstack-dashboard-ubuntu-theme || \
33     xecho 'Unable to remove OpenStack Dashboard Ubuntu theme'
34
35 pecho 'Replace embedded cache backend by memcached'
36 # FIXME todo
37
38 pecho 'Configure OpenStack Dashboard'
39 a=$DASHBOARD_CONF_FILE
40 if ! $sudo grep -q 'PATCHED_BY_OSCIED' $a; then
41     $sudo cp $a $a.old
42     b="s:CONTROLLER_PRIVATE_IP:$CONTROLLER_PRIVATE_IP:g"
43     c="s:DASH_USER:$user:g"
44     d="s:DASH_PASSWORD:$pass:g"
45     $sudo sh -c "sed '$b;$c;$d' '$DASHBOARD_APPEND_FILE' >> '$a'"
46 else
47     # FIXME not keep in sync if something changed ...
48     recho 'dashboard configuration already done'
49 fi
50
51 service restart
52 }
53
54 uninstall()
55 {
56     if [ $# -ne 0 ]; then
57         xecho "Usage: $(basename $0) uninstall"
58     fi
59     ok=$true
60
61     pecho 'Uninstall OpenStack Dashboard'
62     eval $purge memcached libapache2-mod-wsgi openstack-dashboard apache2
63     eval $autoremove
64     $sudo rm -rf /etc/openstack-dashboard/ /var/log/apache2/ #+ memcached
65 }
66
67 service()
68 {
69     if [ $# -ne 1 ]; then
70         xecho "Usage: $(basename $0) service operation\n\toperation : { start, stop, restart, ... }"
71     fi
72     ok=$true
73
74     op=$1
75     eval $service apache2 "$op" || xecho "Unable to $op dashboard service"
76 }
77
78 main "$@"

```

6.15.10 scripts/glance.sh

```

1  ./common.sh
2
3 main()
4 {
5     ok=$false
6     if [ $# -gt 0 ]; then
7         mode=$1
8         shift # remove mode parameter
9         eval "$mode" "$@"
10    fi
11
12    if [ $ok -eq $false ]; then
13        xecho "Usage: $(basename $0) mode\n\tmode : { install, installData, uninstall, service, try }"
14    fi
15 }
16
17 install()
18 {

```

```

19  if [ $# -ne 0 ]; then
20      xecho "Usage: $(basename $0) install"
21  fi
22  ok=$true
23
24  user='glance'
25  getPass "$user" || xecho "Unable to get $user password"
26
27  pecho 'Install and configure Glance Image Service'
28  eval $install glance glance-api glance-common python-glanceclient \
29      glance-registry python-glance || xecho 'Unable to install Glance Image Service'
30
31  pecho 'Replace embedded sqlite database by MySQL'
32  $sudo rm /var/lib/glance/glance.sqlite 2>/dev/null
33  db="mysql://$user:$pass@$CONTROLLER_PRIVATE_IP/glance?charset=utf8"
34
35  pecho 'Configure Glance API'
36  a=$GLANCE_API_CONF_FILE
37  $sudo cp $a $a.old
38  setSetting $a $true 'sql_connection' "$db" || xecho '1'
39  setSetting $a $true 'enable_v2_api' 'True' || xecho '2'
40  setSetting $a $true 'notifier_strategy' 'rabbit' || xecho '3'
41  setSetting $a $true 'rabbit_host' "$CONTROLLER_PRIVATE_IP" || xecho '4'
42  setSetting $a $true 'rabbit_password' "$RABBIT_PASS" || xecho '5'
43  # FIXME todo swift related informations (if installed)
44  setSetting $a $true 'auth_host' "$CONTROLLER_PRIVATE_IP" || xecho '6'
45  setSetting $a $true 'admin_tenant_name' 'service' || xecho '7'
46  setSetting $a $true 'admin_user' "$user" || xecho '8'
47  setSetting $a $true 'admin_password' "$pass" || xecho '9'
48  setSetting $a $true 'flavor' 'keystone' || xecho '10'
49
50  # FIXME and glance cache to speedup nodes !
51
52  pecho 'Configure Glance Registry'
53  a=$GLANCE_REGISTRY_CONF_FILE
54  $sudo cp $a $a.old
55  setSetting $a $true 'sql_connection' "$db" || xecho '1'
56  setSetting $a $true 'auth_host' "$CONTROLLER_PRIVATE_IP" || xecho '2'
57  setSetting $a $true 'admin_tenant_name' 'service' || xecho '3'
58  setSetting $a $true 'admin_user' "$user" || xecho '4'
59  setSetting $a $true 'admin_password' "$pass" || xecho '5'
60  setSetting $a $true 'flavor' 'keystone' || xecho '6'
61
62  service restart
63
64  pecho 'Initialize the new Glance Image Service database'
65  $sudo glance-manage db_sync || xecho 'Unable to initialize database'
66
67  service restart
68 }
69
70 installData()
71 {
72     if ! which glance > /dev/null; then
73         xecho 'Glance must be installed !'
74     fi
75
76     ok=$false
77     if [ $# -eq 1 ]; then
78         eval "data_$1"
79     fi
78
80     if [ $ok -eq $false ]; then
81         xecho "Usage: `basename $0` installData mode\n\tmode : { all, ubuntu_12_04_UEC, verify }"
82     fi
83 }
84
85
86 data_all()
87 {
88     ok=$true
89     ubuntu_12_04_UEC
90     verify
91 }

```

```

92
93 data_ubuntu_12_04_UEC()
94 {
95     ok=$true
96     pecho "Add Ubuntu 12.04 UEC image"
97     getPass 'admin' || xecho 'Unable to get admin password'
98     tar='ubuntu-12.04-server-cloudimg-amd64.tar.gz'
99     imageCreateQcow2 'admin' "$pass" "$TENANT_NAME" $true 'Ubuntu 12.04 UEC' \
100    'precise-server-cloudimg-amd64.img' \
101    "http://uec-images.ubuntu.com/releases/precise/release/$tar" || exit 1
102 }
103
104 data_verify()
105 {
106     ok=$true
107     pecho "Verify Glance's insallation"
108     #getPass 'admin' || xecho 'Unable to get admin password'
109     #tar='http://smoser.brickies.net/ubuntu/ttlinux-uec/ttlinux-uec-amd64-12.1_2.6.35-22_1.tar.gz'
110     #imageCreateAll 'admin' "$pass" "$TENANT_NAME" "$tar" || exit 1
111     #recho "No problem 'till now"
112     recho 'Verification skipped'
113 }
114
115 uninstall()
116 {
117     if [ $# -ne 0 ]; then
118         xecho "Usage: $(basename $0) uninstall"
119     fi
120     ok=$true
121
122     pecho 'Uninstall Glance Image Service'
123     eval $purge glance glance-api glance-common python-glanceclient glance-registry python-glance
124     eval $autoremove
125     $sudo rm -rf /etc/glance/ /var/lib/glance/ /var/log/glance/
126 }
127
128 service()
129 {
130     if ! which glance > /dev/null; then
131         xecho 'Glance must be installed !'
132     fi
133
134     if [ $# -ne 1 ]; then
135         xecho "Usage: $(basename $0) service operation\n\toperation : { start, stop, restart, ... }"
136     fi
137     ok=$true
138
139     op=$1
140     eval $service glance-api      "$op" || xecho "Unable to $op glance api service"
141     eval $service glance-registry "$op" || xecho "Unable to $op glance registry service"
142 }
143
144 try()
145 {
146     if ! which glance > /dev/null; then
147         xecho 'Glance must be installed !'
148     fi
149     ok=$true
150
151     getPass 'admin' || xecho 'Unable to get admin password'
152     export OS_USERNAME=admin
153     export OS_PASSWORD=$pass
154     export ADMIN_TOKEN=$ADMIN_TOKEN
155     export SERVICE_TOKEN=$ADMIN_TOKEN
156     export SERVICE_ENDPOINT=$CONTROLLER_ADMIN_URL
157     export OS_AUTH_URL=$CONTROLLER_AUTHZ_URL
158     getId tenant $TENANT_NAME || exit 1; tId=$id
159     echo "Images"; glance --os-tenant-id "$tId" image-list
160 }
161
162 main "$@"

```

6.15.11 scripts/install.prerequisites.sh

```

1 . ./common.sh
2
3 pecho 'Enable the Cloud Archive repository for Ubuntu'
4 release=$(cat /etc/lsb-release | grep DISTRIB_CODENAME | cut -d'=' -f2)
5 if [ "$release" = 'precise' ]; then
6   deb='http://ubuntu-cloud.archive.canonical.com/ubuntu'
7   $sudo sh -c "echo '$deb precise-updates/folsom main' > /etc/apt/sources.list.d/folsom.list"
8   eval $install ubuntu-cloud-keyring || xecho 'Unable to install ubuntu cloud keyring'
9 else
10   recho "Ubuntu Cloud Archive not necessary for release $release"
11 fi
12
13 pecho 'Enable OpenStack Ubuntu Testers repository for Ubuntu'
14 deb='http://ppa.launchpad.net/openstack-ubuntu-testing/folsom-trunk-testing/ubuntu'
15 $sudo sh -c "echo '$deb quantal main' > /etc/apt/sources.list.d/folsom-testing.list"
16 $sudo apt-key adv --recv-keys --keyserver keyserver.ubuntu.com '81DCD8423B6F61A6' || \
17 xecho 'Unable to receive gpg key'
18
19 eval $update
20 eval $upgrade
21 eval $install debconf-utils python-cliff python-greenlet python-mysqldb python-pyparsing \
22     python-sqlalchemy || xecho 'Unable to install common prerequisites'
23
24 # http://serverfault.com/questions/252333/list-all-packages-from-a-repository-in-ubuntu-debian
25 #packages='ubuntu-cloud.archive.canonical.'
26 #packages+=`com_ubuntu_dists_precise-updates_folsom_main_binary-amd64_Packages`
27 #cat /var/lib/apt/lists/$packages | grep Package

```

6.15.12 scripts/keystone.sh

```

1 . ./common.sh
2
3 main()
4 {
5   ok=$false
6   if [ $# -gt 0 ]; then
7     mode=$1
8     shift # remove mode parameter
9     eval "$mode" "$@"
10  fi
11
12  if [ $ok -eq $false ]; then
13    xecho "Usage: $(basename $0) mode\n\tmode : { install, installData, uninstall, service, try }"
14  fi
15 }
16
17 install()
18 {
19   if [ $# -ne 0 ]; then
20     xecho "Usage: $(basename $0) install"
21   fi
22   ok=$true
23
24   user='keystone'
25   getPass "$user" || xecho "Unable to get $user password"
26
27   pecho 'Install and configure Keystone Identity Service'
28   eval $install keystone python-keystone python-keystoneclient || xecho 'Unable to install Keystone'
29
30   pecho 'Replace embedded sqlite database by MySQL'
31   $sudo rm /var/lib/keystone/keystone.db 2>/dev/null
32   db="mysql://$user:$pass@$CONTROLLER_PRIVATE_IP/keystone?charset=utf8"
33
34   pecho 'Configure Keystone'
35   a=$KEYSTONE_CONF_FILE
36   $sudo cp $a $a.old
37   setSetting $a $true 'admin_token' "$ADMIN_TOKEN" || xecho '1'
38   setSetting $a $true 'connection' "$db" || xecho '2'
39

```

```

40     service restart
41
42     pecho 'Initialize the new Keystone Identity Service database'
43     $sudo keystone-manage db_sync || xecho 'Unable to initialize database'
44 }
45
46 installData()
47 {
48     if ! which keystone > /dev/null; then
49         xecho 'Keystone must be installed !'
50     fi
51
52     ok=$false
53     if [ $# -eq 1 ]; then
54         eval "data_\$1"
55     fi
56
57     if [ $ok -eq $false ]; then
58         mode='mode : {all, tenants, users, roles, usersRoles, services, endpoints, verify }'
59         xecho "Usage: $(basename $0) installData mode\n\t$mode"
60     fi
61 }
62
63 data_all()
64 {
65     ok=$true
66     tenants
67     users
68     roles
69     usersRoles
70     services
71     endpoints
72     verify
73 }
74
75 data_tenants()
76 {
77     ok=$true
78     pecho 'Create Keystone Tenants'
79     savedIFS=$IFS
80     IFS=';'
81     while read tenant description
82     do
83         if echo "$tenant" | grep -q '#'; then
84             continue
85         fi
86         mecho "Create Keystone's tenant $tenant"
87         if ! tenantCreate "$tenant"; then
88             xecho "Unable to create tenant $tenant"
89         fi
90     done < "$KEYSTONE_TENANTS_FILE"
91     IFS=$savedIFS
92 }
93
94 data_users()
95 {
96     ok=$true
97     pecho 'Create Keystone Users'
98     savedIFS=$IFS
99     IFS=';'
100    while read enabled name pass mail description
101    do
102        if echo "$enabled" | grep -q '#'; then continue; fi
103        if echo "$enabled" | grep -q 'off'; then continue; fi
104        mecho "Create Keystone's user $name"
105        if ! userCreate "$name" "$pass" "$mail" "$description"; then
106            xecho "Unable to create user $name"
107        fi
108    done < "$KEYSTONE_USERS_FILE"
109    IFS=$savedIFS
110 }
111
112 data_roles()

```

```

113  {
114    ok=$true
115    pecho 'Create Keystone Roles'
116    savedIFS=$IFS
117    IFS=';'
118    while read role description
119    do
120      if echo "$role" | grep -q '#'; then
121        continue
122      fi
123      mecho "Create Keystone's role $role"
124      if ! roleCreate "$role"; then
125        xecho "Unable to create role $role"
126      fi
127    done < "$KEYSTONE_ROLES_FILE"
128    IFS=$savedIFS
129  }
130
131  data_usersRoles()
132  {
133    ok=$true
134    pecho 'Assign Keystone Roles to Users'
135    savedIFS=$IFS
136    IFS=';'
137    while read tenant user role comment
138    do
139      if echo "$tenant" | grep -q '#'; then
140        continue
141      fi
142      mecho "Give role $role to user $user into tenant $tenant"
143      getId 'tenant' "$tenant" || xecho 'Unable to get tenant id'; tenantId=$id
144      getId 'user'    "$user"   || xecho 'Unable to get user id';   userId=$id
145      getId 'role'   "$role"  || xecho 'Unable to get role id';  roleId=$id
146      if ! userRoleAdd "$tenantId" "$userId" "$roleId"; then
147        xecho "Unable to give role $role to user $user into tenant $tenant"
148      fi
149    done < "$KEYSTONE_USERS_ROLES_FILE"
150    IFS=$savedIFS
151  }
152
153  data_services()
154  {
155    ok=$true
156    pecho 'Create Keystone Services'
157    savedIFS=$IFS
158    IFS=';'
159    while read service type description
160    do
161      if echo "$service" | grep -q '#'; then
162        continue
163      fi
164      mecho "Create Keystone's service $service"
165      if ! serviceCreate "$service" "$type" "$description"; then
166        xecho "Unable to create service $service"
167      fi
168    done < "$KEYSTONE_SERVICES_FILE"
169    IFS=$savedIFS
170  }
171
172  data_endpoints()
173  {
174    ok=$true
175    pecho 'Create Keystone Endpoints'
176    savedIFS=$IFS
177    IFS=';'
178    while read region service public internal admin
179    do
180      if echo "$region" | grep -q '#'; then
181        continue
182      fi
183      mecho "Create Keystone's endpoint $region:$service"
184      public="http://$CONTROLLER_PUBLIC_IP:$public"
185      internal="http://$CONTROLLER_PRIVATE_IP:$internal"

```

```

186     admin="http://$CONTROLLER_PRIVATE_IP:$admin"
187     getId 'service' "$service" || xecho 'Unable to get service id'; serviceId=$id
188     if ! endpointCreate "$region" "$serviceId" "$public" "$internal" "$admin"; then
189         xecho "Unable to create endpoint $region:$service"
190     fi
191     done < "$KEYSTONE_ENDPOINTS_FILE"
192     IFS=$savedIFS
193 }
194
195 data_verify()
196 {
197     ok=$true
198     pecho "Verify Keystone's installation"
199     getPass 'admin' || xecho 'Unable to get admin password'
200     tokenGet 'admin' "$pass" || xecho 'Unable to get a token for admin user'
201     tokenGet 'admin' "$pass" $TENANT_NAME || \
202         xecho "Unable to get a token for admin user in tenant $TENANT_NAME"
203     recho "No problem 'till now"
204 }
205
206
207 uninstall()
208 {
209     if [ $# -ne 0 ]; then
210         xecho "Usage: $(basename $0) uninstall"
211     fi
212     ok=$true
213
214     pecho 'Uninstall Keystone Identity Service'
215     eval $purge keystone python-keystone python-keystoneclient
216     eval $autoremove
217     $sudo rm -rf /etc/keystone/ /var/lib/keystone/ /var/log/keystone/
218 }
219
220 service()
221 {
222     if ! which keystone > /dev/null; then
223         xecho 'Keystone must be installed !'
224     fi
225
226     if [ $# -ne 1 ]; then
227         xecho "Usage: $(basename $0) service operation\n\toperation : { start, stop, restart, ... }"
228     fi
229     ok=$true
230
231     op=$1
232     eval $service keystone "$op" || xecho "Unable to $op keystone service"
233 }
234
235 try()
236 {
237     if ! which keystone > /dev/null; then
238         xecho 'Keystone must be installed !'
239     fi
240     ok=$true
241
242     # mysqldump -u root -p keystone > keystone.sql
243
244     getPass 'admin' || xecho 'Unable to get admin password'
245     export OS_USERNAME=admin
246     export OS_PASSWORD=$pass
247     #export ADMIN_TOKEN=$ADMIN_TOKEN
248     export SERVICE_TOKEN=$ADMIN_TOKEN
249     export SERVICE_ENDPOINT=$CONTROLLER_ADMIN_URL
250     export OS_AUTH_URL=$CONTROLLER_AUTHZ_URL
251     getId user 'admin' || exit 1; uId=$id
252     getId tenant $TENANT_NAME || exit 1; tId=$id
253     #env | sort | grep -v LS_COLOR
254     #exit 0
255     echo "Tenants";      keystone tenant-list
256     echo "Users";       keystone user-list
257     echo "Roles";       keystone role-list
258     echo "Users Roles"; keystone user-role-list --user-id $uId --tenant-id $tId

```

```

259     echo "Services";      keystone service-list
260     echo "Endpoints";    keystone endpoint-list
261     echo "EC2 Credentials"; keystone ec2-credentials-list --user-id $uId
262 }
263
264 main "$@"

```

6.15.13 scripts/kvm.sh

```

1  . ./common.sh
2
3 main()
4 {
5     ok=$false
6     if [ $# -gt 0 ]; then
7         mode=$1
8         shift # remove mode parameter
9         eval "$mode" "$@"
10    fi
11
12    if [ $ok -eq $false ]; then
13        xecho "Usage: $(basename $0) mode\n\tmode : { install, uninstall, service }"
14    fi
15}
16
17 install()
18 {
19     if [ $# -ne 0 ]; then
20         xecho "Usage: $(basename $0) install"
21     fi
22     ok=$true
23
24     pecho 'Install and configure KVM hypervisor'
25     eval $install kvm libvirt-bin pm-utils || xecho 'Unable to install hypervisor'
26     if ! $sudo grep -q 'PATCHED_BY_OSCIED' ${QEMU_CONF_FILE}; then
27         if ! $sudo grep -q '/dev/net/tun' ${QEMU_CONF_FILE}; then
28             $sudo patch -c ${QEMU_CONF_FILE} "${QEMU_PATCH_FILE}" || xecho 'Unable to configure hypervisor'
29         else
30             # FIXME better check
31             recho 'Hypervisor configuration seem OK'
32         fi
33     else
34         recho 'Hypervisor configuration already updated'
35     fi
36
37     pecho 'Check if KVM virtualization is supported'
38     autoInstall cpu-checker kvm-ok
39     if ! kvm-ok > /dev/null; then
40         xecho 'Hardware virtualization support is required'
41     fi
42
43     pecho 'Load KVM virtualization kernel module'
44     $sudo modprobe kvm
45     $sudo modprobe kvm-intel
46     if ! lsmod | grep -q 'kvm'; then
47         xecho 'Unable to load KVM kernel module'
48     fi
49
50     pecho 'Disable default virtual bridge'
51     $sudo virsh net-destroy default
52     $sudo virsh net-undefine default
53
54     pecho 'Allow live migration'
55     a=${LIBVIRTD_CONF_FILE}
56     $sudo cp $a $a.old
57     setSetting $a $true 'listen_tls' '0' || xecho '1'
58     setSetting $a $true 'listen_tcp' '1' || xecho '2'
59     setSetting $a $true 'auth_tcp' "'none'" || xecho '3'
60
61     a=${LIBVIRTD_INIT_FILE}
62     $sudo cp $a $a.old

```

```

63 setSetting $a $true 'env libvirtd_opts' '"-d -l"' || xecho '4'
64
65 a=$LIBVIRTD_DEF_FILE
66 $sudo cp $a $a.old
67 setSetting $a $true 'libvirtd_opts' '"-d -l"' || xecho '4'
68
69 service restart
70 }
71
72 uninstall()
73 {
74     if [ $# -ne 0 ]; then
75         xecho "Usage: $(basename $0) uninstall"
76     fi
77     ok=$true
78
79     pecho 'Uninstall KVM hypervisor'
80     eval $purge kvm libvirt-bin
81     eval $autoremove
82     $sudo rm -rf /etc/kvm/ /etc/libvirt/ /var/log/libvirt/
83 }
84
85 service()
86 {
87     if [ $# -ne 1 ]; then
88         xecho "Usage: $(basename $0) service operation\n\toperation : { start, stop, restart, ... }"
89     fi
90     ok=$true
91
92     op=$1
93     eval $service libvirt-bin "$op" || xecho "Unable to $op hypervisor service"
94 }
95
96 main "$@"

```

6.15.14 scripts/mysql.sh

```

1 . ./common.sh
2
3 main()
4 {
5     ok=$false
6     if [ $# -gt 0 ]; then
7         mode=$1
8         shift # remove mode parameter
9         eval "$mode" "$@"
10    fi
11
12    if [ $ok -eq $false ]; then
13        xecho "Usage: $(basename $0) mode\n\tmode : { install, uninstall, service }"
14    fi
15 }
16
17 install()
18 {
19     if [ $# -ne 0 ]; then
20         xecho "Usage: $(basename $0) install"
21     fi
22     ok=$true
23
24     pecho 'Install and configure MySQL'
25     sql='mysql-server mysql-server' # Tip : http://ubuntuforums.org/showthread.php?t=981801
26     echo "$sql/root_password select $MYSQL_ROOT_PASS" | $sudo debconf-set-selections
27     echo "$sql/root_password_again select $MYSQL_ROOT_PASS" | $sudo debconf-set-selections
28     $sudo mkdir /etc/mysql 2>/dev/null
29     eval $install mysql-server || xecho 'Unable to install MySQL'
30
31     # Now MySQL will listen to incoming request of any source
32     recho 'FIXME security (listen to private network only ?)'
33     $sudo sed -i 's/127.0.0.1/0.0.0.0/g' /etc/mysql/my.cnf
34

```

```

35 root=$MYSQL_ROOT_PASS
36
37 # Fix ticket #57 : Keystone + MySQL = problems
38 mysql -uroot -p"$root" -e "DROP USER ''@'localhost'; DROP USER ''@'$hostname'';" 
39 mysql -uroot -p"$root" -e "GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' WITH GRANT OPTION;" 
40 mysql -uroot -p"$root" -e "SET PASSWORD FOR 'root'@'%' = PASSWORD('$MYSQL_ROOT_PASS');" 
41
42 service restart
43
44 pecho 'Create Cinder Volume Service database & user'
45 user='cinder'
46 getPassword "$user" || xecho "Unable to get $user password"
47 mysql -u root -p"$root" -e "CREATE DATABASE cinder;" 
48 mysql -u root -p"$root" -e "GRANT ALL ON cinder.* TO '$user'@'%' IDENTIFIED BY '$pass';" 
49
50 pecho 'Create Dashboard Service database & user'
51 user='dashboard'
52 getPassword "$user" || xecho "Unable to get $user password"
53 mysql -u root -p"$root" -e "CREATE DATABASE dashboard;" 
54 mysql -u root -p"$root" -e "GRANT ALL ON dashboard.* TO '$user'@'%' IDENTIFIED BY '$pass';" 
55
56 pecho 'Create Glance Image Service database & user'
57 user='glance'
58 getPassword "$user" || xecho "Unable to get $user password"
59 mysql -u root -p"$root" -e "CREATE DATABASE glance;" 
60 mysql -u root -p"$root" -e "GRANT ALL ON glance.* TO '$user'@'%' IDENTIFIED BY '$pass';" 
61
62 pecho 'Create Keystone Identity Service database & user'
63 user='keystone'
64 getPassword "$user" || xecho "Unable to get $user password"
65 mysql -u root -p"$root" -e 'CREATE DATABASE keystone;' 
66 mysql -u root -p"$root" -e "GRANT ALL ON keystone.* TO '$user'@'%' IDENTIFIED BY '$pass';" 
67
68 pecho 'Create Nova Compute Service database & user'
69 user='nova'
70 getPassword "$user" || xecho "Unable to get $user password"
71 mysql -uroot -p"$root" -e 'CREATE DATABASE nova;' 
72 mysql -uroot -p"$root" -e "GRANT ALL PRIVILEGES ON nova.* TO '$user'@'%' IDENTIFIED BY '$pass';" 
73
74 pecho 'Create Quantum Network Service database & user'
75 user='quantum'
76 getPassword "$user" || xecho "Unable to get $user password"
77 mysql -u root -p"$root" -e "CREATE DATABASE quantum;" 
78 mysql -u root -p"$root" -e "GRANT ALL ON quantum.* TO '$user'@'%' IDENTIFIED BY '$pass';" 
79 }
80
81 uninstall()
82 {
83     if [ $# -ne 0 ]; then
84         xecho "Usage: $(basename $0) uninstall"
85     fi
86     ok=$true
87
88     pecho 'Uninstall MySQL'
89     eval $purge mysql-client-5.5 mysql-client-core-5.5 mysql-common mysql-server mysql-server-5.1 \
90     mysql-server-5.5 mysql-server-core-5.5
91     eval $autoremove
92     sudo rm -rf /etc/mysql/ /var/lib/mysql/ /var/log/mysql/
93 }
94
95 service()
96 {
97     if ! which mysql > /dev/null; then
98         xecho 'MySQL must be installed !'
99     fi
100
101    if [ $# -ne 1 ]; then
102        xecho "Usage: $(basename $0) service operation\n\toperation : { start, stop, restart, ... }"
103    fi
104    ok=$true
105
106    op=$1
107    eval $service mysql "$op" || xecho "Unable to $op MySQL service"

```

```

108 }
109
110 main "$@"

```

6.15.15 scripts/nova.sh

```

1 . ./common.sh
2
3 main()
{
4     ok=$false
5     if [ $# -gt 0 ]; then
6         mode=$1
7         shift # remove mode parameter
8         eval "$mode" "$@"
9     fi
10
11    if [ $ok -eq $false ]; then
12        xecho "Usage: $(basename $0) mode\n\tmode : { install, uninstall, service, try }"
13    fi
14}
15
16
17 install()
{
18     ok=$false
19     if [ $# -eq 1 ]; then
20         eval "install_$1"
21     fi
22
23     if [ $ok -eq $false ]; then
24         xecho "Usage: `basename $0` install mode\n\tmode : { standalone, controller, compute }"
25     fi
26}
27
28
29 install_standalone()
{
30     ok=$true
31
32     user='nova'
33     getPass "$user" || xecho "Unable to get $user password"
34
35     pecho 'Install and configure Nova Compute Service'
36     eval $install nova-api nova-cert nova-common nova-compute nova-compute-kvm nova-doc \
37         nova-consoleauth nova-scheduler nova-novncproxy novnc python-nova python-novaclient || \
38     xecho 'Unable to install Nova Compute Service'
39
40     pecho 'Replace embedded sqlite database by MySQL'
41     sudo rm /var/lib/nova/nova.sqlite 2>/dev/null
42     db="mysql://$user:$pass@$CONTROLLER_PRIVATE_IP/nova?charset=utf8"
43
44
45     pecho 'Configure Nova API'
46     a=$NOVA_API_PASTE_FILE
47     sudo cp $a $a.old
48     setSetting $a $true 'auth_host'      "$CONTROLLER_PRIVATE_IP" || xecho '1'
49     setSetting $a $true 'admin_tenant_name' 'service'          || xecho '2'
50     setSetting $a $true 'admin_user'      "$user"             || xecho '3'
51     setSetting $a $true 'admin_password' "$pass"            || xecho '4'
52
53     user='quantum'
54     getPass "$user" || xecho "Unable to get $user password"
55
56     pecho 'Configure Nova on a Standalone'
57     a=$NOVA_CONF_FILE
58     sudo cp $a $a.old
59     b="s:$CONTROLLER_PUBLIC_IP:$CONTROLLER_PUBLIC_IP:g"
60     c="s:$CONTROLLER_PRIVATE_IP:$CONTROLLER_PRIVATE_IP:g"
61     d="s:HOST_PUBLIC_IP:$HOST_PUBLIC_IP:g"
62     e="s:HOST_PRIVATE_IP:$HOST_PRIVATE_IP:g"
63     sudo sh -c "sed '$b;$c;$d;$e' '$NOVA_TEMPL_FILE' > '$a'"
64     setSetting $a $true 'sql_connection'           "$db"          || xecho '1'
65     setSetting $a $true 'rabbit_password'        "$RABBIT_PASS" || xecho '2'

```

```

66     setSetting $a $true 'quantum_admin_tenant_name' 'service'      || xecho '3'
67     setSetting $a $true 'quantum_admin_username'      "$user"      || xecho '4'
68     setSetting $a $true 'quantum_admin_password'      "$pass"      || xecho '5'
69
70     pecho 'Configure Nova Compute'
71     a=$NOVA_COMPUTE_CONF_FILE
72     $sudo mv $a $a.old
73     $sudo cp $NOVA_COMPUTE_HACK_FILE $a
74
75     service restart
76
77     pecho 'Initialize the new Nova Compute Service database'
78     $sudo nova-manage db sync || exit 1
79
80     service restart
81 }
82
83 uninstall()
84 {
85     if [ $# -ne 0 ]; then
86         xecho "Usage: $(basename $0) uninstall"
87     fi
88     ok=$true
89
90     pecho 'Uninstall Nova Compute Service'
91     eval $purge nova-* novnc python-nova*
92     eval $autoremove
93     $sudo rm -rf /etc/nova/ /var/lib/nova/ /var/log/nova/
94 }
95
96 service()
97 {
98     if ! which nova > /dev/null; then
99         xecho 'Nova must be installed !'
100    fi
101
102    if [ $# -ne 1 ]; then
103        xecho "Usage: $(basename $0) service operation\n\toperation : { start, stop, restart, ... }"
104    fi
105    ok=$true
106
107    op=$1
108    eval $service nova-api      "$op" || xecho "Unable to $op nova api service"
109    eval $service nova-cert     "$op" || xecho "Unable to $op nova cert service"
110    eval $service nova-compute   "$op" || xecho "Unable to $op nova compute service"
111    eval $service nova-consoleauth "$op" || xecho "Unable to $op nova console auth service"
112    #NOVA-VOLUME eval $service nova-volume      "$op" || xecho "Unable to $op nova volume service"
113    eval $service nova-scheduler   "$op" || xecho "Unable to $op nova scheduler service"
114 }
115
116 try()
117 {
118     if ! which nova > /dev/null; then
119         xecho 'Nova must be installed !'
120     fi
121
122     ok=false
123     if [ $# -gt 1 ]; then
124         eval "try_$1"
125     fi
126
127     if [ $ok -eq $false ]; then
128         xecho "Usage: `basename $0` try mode\n\tmode : { nova, manage }"
129     fi
130 }
131
132 try_nova()
133 {
134     ok=$true
135
136     # sh nova.sh try nova network create --label=yahoo --fixed_range_v4=10.10.10.0/24 \
137     # --bridge=br100 --dns1=8.8.8.8 --project_id=1c226c717e3d43e89415230e5e17c286
138     # sh nova.sh try nova boot myUbuntu --image 9fbebc50-2376-4f6e-8492-a3048bc32ef9 --flavor 2

```

```

139
140  getPassword 'admin' || xecho 'Unable to get admin password'
141  export OS_NO_CACHE=1 # fixes bug https://bugs.launchpad.net/python-novaclient/+bug/1020238
142  export OS_TENANT_NAME=$TENANT_NAME
143  export OS_USERNAME=admin
144  export OS_PASSWORD=$pass
145  #export ADMIN_TOKEN=$ADMIN_TOKEN
146  #export SERVICE_TOKEN=$ADMIN_TOKEN
147  #export SERVICE_ENDPOINT=$CONTROLLER_ADMIN_URL
148  export OS_AUTH_URL=$CONTROLLER_AUTHZ_URL
149
150  if [ $# -eq 0 ]; then
151      echo "Hosts"; nova host-list
152      echo "Hypervisors"; nova hypervisor-list
153      echo "Images"; nova image-list
154      echo "Servers"; nova list
155      echo "Keypairs"; nova keypair-list
156      echo "Networks"; nova network-list
157      echo "Volumes"; nova volume-list
158      echo "Volumes Types"; nova volume-type-list
159      echo "Flavors"; nova flavor-list
160      echo "Services"; $sudo nova-manage service list
161      # echo "Quotas"; $sudo nova-manage quota
162  else
163      nova --os-username=$OS_USERNAME --os-password=$OS_PASSWORD \
164          --os-auth-url=$OS_AUTH_URL --os-tenant-name=$OS_TENANT_NAME "$@"
165  fi
166 }
167
168 try_manage()
169 {
170     ok=$true
171
172     # sh nova.sh try nova network create --label=yahoo --fixed_range_v4=10.10.10.0/24 --bridge=br100 \
173     #     --dns1=8.8.8.8 --project_id=1c226c717e3d43e89415230e5e17c286
174     # sh nova.sh try nova boot myUbuntu --image 9fbefc50-2376-4f6e-8492-a3048bc32ef9 --flavor 2
175
176     getPassword 'admin' || xecho 'Unable to get admin password'
177     export OS_NO_CACHE=1 # fixes bug https://bugs.launchpad.net/python-novaclient/+bug/1020238
178     export ADMIN_TOKEN=$ADMIN_TOKEN
179     export SERVICE_TOKEN=$ADMIN_TOKEN
180     export SERVICE_ENDPOINT=$CONTROLLER_ADMIN_URL
181     $sudo nova-manage "$@"
182 }
183
184 main "$@"

```

6.15.16 scripts/open-vswitch.sh

```

1 . ./common.sh
2
3 main()
4 {
5     ok=$false
6     if [ $# -gt 0 ]; then
7         mode=$1
8         shift # remove mode parameter
9         eval "$mode" "$@"
10    fi
11
12    if [ $ok -eq $false ]; then
13        xecho "Usage: $(basename $0) mode\n\tmode : { install, uninstall, services }"
14    fi
15 }
16
17 install()
18 {
19     if [ $# -ne 0 ]; then
20         xecho "Usage: $(basename $0) install"
21     fi
22     ok=$true
23

```

```

24  pecho 'Install and configure Open vSwitch'
25  eval $install linux-headers-generic # required to compile kernel modules
26  eval $install openvswitch-switch openvswitch-datapath-dkms
27  eval $service openvswitch-switch start || xecho 'Unable to start Open vSwitch'
28
29  pecho 'Create quantum open-vSwitch agent required OVS bridge br-int'
30  $sudo ovs-vsctl add-br br-int || xecho 'Unable to create bridge br-int'
31
32  # FIXME todo a network watchdog -> fallback in case of necessity !
33
34  pecho 'Create quantum l3 agent required OVS bridge br-ex'
35  $sudo ovs-vsctl add-br br-ex || xecho 'Unable to create bridge br-ex'
36  $sudo ovs-vsctl br-set-external-id br-ex bridge-id br-ex || xecho 'TODO'
37  $sudo ovs-vsctl add-port br-ex $HOST_PUBLIC_IFACE || \
38    xecho "Unable to attach br-ex to $HOST_PUBLIC_IFACE"
39
40  pecho "Enable bridge br-ex and reset interface $HOST_PUBLIC_IFACE"
41  $sudo ifconfig br-ex up
42  $sudo ifconfig br-ex $HOST_PUBLIC_IP netmask $HOST_PUBLIC_MASK
43  $sudo ifconfig $HOST_PUBLIC_IFACE 0
44  $sudo route add default gw $HOST_PUBLIC_GW dev br-ex
45
46  pecho 'Update network interfaces configuration'
47  if ! grep -q 'br-ex' $NETWORK_CONF_FILE; then
48    a=$NETWORK_CONF_FILE
49    n=$HOST_PUBLIC_IFACE
50    $sudo cp $a $a.old
51    b='auto PUBLIC_IFACE'
52    c='iface PUBLIC_IFACE inet manual'
53    d='pre-up ip link set \'$IFACE up'
54    e='post-down ip link set \'$IFACE down'
55    f='auto br-ex'
56    $sudo sed -i "s:auto $n:$b\n$c\n$d\n$e\n$f:;s:$n:br-ex:g;s:PUBLIC_IFACE:$n:g" $a
57  else
58    recho 'Network interfaces configuration already updated'
59  fi
60
61  service restart
62 }
63
64 uninstall()
{
65
66  if [ $# -ne 0 ]; then
67    xecho "Usage: $(basename $0) uninstall"
68  fi
69  ok=$true
70
71  # FIXME avoid breaking network (insert old network config + ifconfig)
72  pecho 'Uninstall Open vSwitch'
73  eval $purge openvswitch-switch openvswitch-datapath-dkms
74  eval $autoremove
75  $sudo rm -rf /etc/openvswitch* /var/log/openvswitch/
76  $sudo cp -f "$HOST_NETWORK_FILE" $NETWORK_CONF_FILE
77  $sudo ifconfig br-ex down
78  $sudo ifconfig $HOST_PUBLIC_IFACE $HOST_PUBLIC_IP netmask $HOST_PUBLIC_MASK
79  $sudo route add default gw $HOST_PUBLIC_GW
80  eval $service networking restart
81 }
82
83 service()
{
84
85  if [ $# -ne 1 ]; then
86    xecho "Usage: $(basename $0) service operation\n\toperation : { start, stop, restart, ... }"
87  fi
88  ok=$true
89
90  op=$1
91  eval $service networking "$op" || xecho "Unable to $op networking service"
92 }
93
94 main "$@"

```

6.15.17 scripts/quantum.sh

```

1 . ./common.sh
2
3 main()
4 {
5     ok=$false
6     if [ $# -gt 0 ]; then
7         mode=$1
8         shift # remove mode parameter
9         eval "$mode" "$@"
10    fi
11
12    if [ $ok -eq $false ]; then
13        xecho "Usage: $(basename $0) mode\n\tmode : { install, uninstall, service, try }"
14    fi
15 }
16
17 install()
18 {
19     ok=$false
20     if [ $# -eq 1 ]; then
21         eval "install_$1"
22     fi
23
24     if [ $ok -eq $false ]; then
25         xecho "Usage: `basename $0` install mode\n\tmode : { standalone, controller, compute }"
26     fi
27 }
28
29 install_standalone()
30 {
31     ok=$true
32
33     user='quantum'
34     getPass "$user" || xecho "Unable to get $user password"
35
36     # quantum-plugin-openvswitch-agent <-> nova-compute
37     pecho 'Install and configure Quantum Network Service'
38     eval $install quantum-common quantum-server quantum-plugin-openvswitch \
39         quantum-plugin-openvswitch-agent python-quantum python-quantumclient || \
40     xecho 'Unable to install Quantum Network Service'
41
42     pecho 'Replace embedded sqlite database by MySQL'
43     sudo rm /var/lib/quantum/quantum.sqlite 2>/dev/null
44     db="mysql://$user:$pass@$CONTROLLER_PRIVATE_IP/quantum?charset=utf8"
45
46     pecho 'Configure Quantum Open vSwitch Agent'
47     a=$QUANTUM_OVSWATCH_CONF_FILE
48     Sudo cp $a $a.old
49     Sudo sed -i 's/Default://g' $a
50     setSetting $a $true 'sql_connection'      "$db"          || xecho '1' # Set sql connection
51     setSetting $a $true 'enable_tunneling'   'True'        || xecho '2' # Enable tunnelling
52     setSetting $a $true 'tenant_network_type' 'gre'        || xecho '3' # Mode tunneling
53     setSetting $a $true 'tunnel_id_ranges'   '1:1000'     || xecho '4' # Set range of id's
54     setSetting $a $true 'integration_bridge' 'br-int'     || xecho '5' # Set integration bridge
55     setSetting $a $true 'tunnel_bridge'      'br-tun'     || xecho '6' # Set tunneling bridge
56     setSetting $a $true 'local_ip'           "$HOST_PRIVATE_IP" || xecho '7' # Only if agent running
57
58     a=$QUANTUM_CONF_FILE
59     pecho "Configure Quantum ($(basename $a))"
60     Sudo cp $a $a.old
61     plugin='quantum.plugins.openvswitch.ovs_quantum_plugin.OVSShadowPluginV2'
62     setSetting $a $true 'core_plugin'       "$plugin"      || xecho '1'
63     setSetting $a $true 'auth_strategy'    'keystone'    || xecho '2'
64     setSetting $a $true 'rabbit_host'      "$CONTROLLER_PRIVATE_IP" || xecho '3'
65     setSetting $a $true 'rabbit_password'  "$RABBIT_PASS" || xecho '4'
66
67     a=$QUANTUM_PASTE_FILE
68     pecho "Configure Quantum ($(basename $a))"
69     Sudo cp $a $a.old
70     setSetting $a $true 'auth_host'        "$CONTROLLER_PRIVATE_IP" || xecho '1'
71     setSetting $a $true 'admin_tenant_name' 'service'    || xecho '2'

```

```

72 setSetting $a $true 'admin_user'      "$user"           || xecho '3'
73 setSetting $a $true 'admin_password'   "$pass"          || xecho '4'
74
75 eval $service quantum-server restart || xecho 'Unable to restart quantum server service'
76 eval $service quantum-plugin-openvswitch-agent restart || \
77     xecho 'Unable to restart quantum open-vSwitch agent'
78
79 pecho 'Install and configure Quantum DHCP/L3 Agent'
80 eval $install quantum-dhcp-agent quantum-13-agent || \
81     xecho 'Unable to install Quantum DHCP/L3 Agent'
82
83 pecho 'Configure Quantum DHCP Agent'
84 a=$QUANTUM_DHCP_CONF_FILE
85 $sudo cp $a $a.old
86 setSetting $a $true 'use_namespaces' 'True' || xecho '1'
87
88 pecho 'Configure Quantum Layer 3 Agent'
89 a=$QUANTUM_L3_CONF_FILE
90 $sudo cp $a $a.old
91 setSetting $a $true 'auth_url'        "$CONTROLLER_ADMIN_URL" || xecho '1'
92 setSetting $a $true 'auth_region'     'RegionOne'         || xecho '2'
93 setSetting $a $true 'admin_tenant_name' 'service'        || xecho '3'
94 setSetting $a $true 'admin_user'      "$user"           || xecho '4'
95 setSetting $a $true 'admin_password'   "$pass"          || xecho '5'
96 setSetting $a $true 'metadata_ip'     "$CONTROLLER_PUBLIC_IP" || xecho '6'
97 setSetting $a $true 'use_namespaces'   'True'           || xecho '7'
98
99 #pecho 'Initialize the new Quantum Network Service database'
100 #$sudo quantum-manage db_sync || xecho 'Unable to initialize database'
101
102 g='https://github.com/mseknbilel/OpenStack-Folsom-Install-guide/blob/stable/GRE'
103 pecho 'Fix a known bug of quantum (when using a single router)'
104 cecho "URL : $g/Tricks%26Ideas/modify_iptables_manager.rst"
105 $sudo sed -i 's:/sbin/iptables:g' $QUANTUM_BUG_FILE
106
107 service restart
108 }
109
110 uninstall()
111 {
112     if [ $# -ne 0 ]; then
113         xecho "Usage: $(basename $0) uninstall"
114     fi
115     ok=$true
116
117     pecho 'Uninstall Quantum Network Service'
118     eval $purge quantum-
119     eval $autoremove
120     $sudo rm -rf /etc/quantum/ /var/lib/quantum/ /var/log/quantum/
121 }
122
123 service()
124 {
125     if ! which quantum > /dev/null; then
126         xecho 'Quantum must be installed !'
127     fi
128
129     if [ $# -ne 1 ]; then
130         xecho "Usage: $(basename $0) service operation\n\toperation : { start, stop, restart, ... }"
131     fi
132     ok=$true
133
134     op=$1
135     eval $service quantum-server    "$op" || xecho "Unable to $op quantum server service"
136     eval $service quantum-dhcp-agent "$op" || xecho "Unable to $op quantum dhcp agent service"
137     eval $service quantum-13-agent   "$op" || xecho "Unable to $op quantum 13 agent service"
138     eval $service quantum-plugin-openvswitch-agent "$op" || \
139         xecho "Unable to $op quantum open-vSwitch agent"
140 }
141
142 try()
143 {
144     if ! which quantum > /dev/null; then

```

```

145     xecho 'Quantum must be installed !'
146 fi
147 ok=$true
148
149 # sh nova.sh try nova network create --label=yahoo --fixed_range_v4=10.10.10.0/24 --bridge=br100 \
150 #   --dns=8.8.8.8 --project_id=1c226c717e3d43e89415230e5e17c286
151 # sh nova.sh try nova boot myUbuntu --image 9fbebc50-2376-4f6e-8492-a3048bc32ef9 --flavor 2
152
153 getPass 'admin' || xecho 'Unable to get admin password'
154 export OS_TENANT_NAME=$STENANT_NAME
155 export OS_USERNAME=admin
156 export OS_PASSWORD=$pass
157 #export ADMIN_TOKEN=$ADMIN_TOKEN
158 #export SERVICE_TOKEN=$ADMIN_TOKEN
159 #export SERVICE_ENDPOINT=$CONTROLLER_ADMIN_URL
160 export OS_AUTH_URL=$CONTROLLER_AUTHZ_URL
161
162 if [ $# -eq 0 ]; then
163     # echo "Hosts";      nova host-list
164     # echo "Hypervisors"; nova hypervisor-list
165     # echo "Images";     nova image-list
166     # echo "Servers";    nova list
167     # echo "Keypairs";   nova keypair-list
168     echo oups
169 else
170     quantum --os-username=$OS_USERNAME --os-password=$OS_PASSWORD \
171         --os-auth-url=$OS_AUTH_URL --os-tenant-name=$OS_TENANT_NAME "$@"
172 fi
173 }
174
175 main "$@"

```

6.15.18 scripts/rabbitmq.sh

```

1  ./common.sh
2
3 main()
4 {
5     ok=$false
6     if [ $# -gt 0 ]; then
7         mode=$1
8         shift # remove mode parameter
9         eval "$mode" "$@"
10    fi
11
12    if [ $ok -eq $false ]; then
13        xecho "Usage: $(basename $0) mode\n\tmode : { install, uninstall, service }"
14    fi
15 }
16
17 install()
18 {
19     if [ $# -ne 0 ]; then
20         xecho "Usage: $(basename $0) install"
21     fi
22     ok=$true
23
24     pecho 'Install and configure RabbitMQ Messaging Queue Server'
25     eval $install rabbitmq-server || xecho 'Unable to install RabbitMQ'
26     $sudo rabbitmqctl change_password guest "$RABBIT_PASS" || xecho 'Unable to configure RabbitMQ'
27 }
28
29 uninstall()
30 {
31     if [ $# -ne 0 ]; then
32         xecho "Usage: $(basename $0) uninstall"
33     fi
34     ok=$true
35
36     pecho 'Uninstall RabbitMQ'
37     $sudo mkdir /var/log/rabbitmq 2>/dev/null # fix rabbitmq-server package uninstall error

```

```

38     eval $purge rabbitmq-server
39     eval $autoremove
40     $sudo rm -rf /etc/rabbitmq/ /var/log/rabbitmq/
41 }
42
43 service()
44 {
45     if [ $# -ne 1 ]; then
46         xecho "Usage: $(basename $0) service operation\n\toperation : { start, stop, restart, ... }"
47     fi
48     ok=$true
49
50     op=$1
51     eval $service rabbitmq-server "$op" || xecho "Unable to $op rabbitmq server service"
52 }
53
54 main "$@"

```

6.15.19 scripts/setup.sh

```

1 . ./common.sh
2
3 pecho 'Detected network configuration :'
4 cecho "Public iface : $HOST_PUBLIC_IFACE $HOST_PUBLIC_IP $HOST_PUBLIC_MASK $HOST_PUBLIC_GW"
5 cecho "Private iface : $HOST_PRIVATE_IFACE $HOST_PRIVATE_IP $HOST_PRIVATE_MASK"
6
7 main()
8 {
9     ok=$false
10    if [ $# -gt 0 ]; then
11        mode=$1
12        shift # remove mode parameter
13        eval "$mode" "$@"
14    fi
15
16    if [ $ok -eq $false ]; then
17        n1="\n      "
18        n2="\t: "
19        m='{ bootstrap, startup, standalone, controller, compute, uninstall, service }'
20        a="${n1}bootstrap${n2}Enable automatic setup on boot"
21        b="${n1}startup${n2}Keep system up-to-date on boot"
22        c="${n1}standalone${n2}Install an OpenStack Standalone (all-in-one)"
23        d="${n1}controller${n2}Install an OpenStack Controller"
24        e="${n1}compute${n2}Install an OpenStack Compute Node"
25        f="${n1}uninstall${n2}Uninstall OpenStack components"
26        g="${n1}service${n2}Run OpenStack components services init script"
27        xecho "Usage: $(basename $0) mode\n\tmode : $m\n$a$b$c$d$e$f$g"
28    fi
29 }
30
31 bootstrap()
32 {
33    if [ $# -ne 0 ]; then
34        xecho "Usage: $(basename $0) bootstrap"
35    fi
36    ok=$true
37
38    ./configure.bootstrap.sh
39 }
40
41 startup()
42 {
43    if [ $# -ne 0 ]; then
44        xecho "Usage: $(basename $0) startup"
45    fi
46    ok=$true
47
48    ./configure.startup.sh
49 }
50
51 standalone()

```

```

52  {
53    if [ $# -ne 0 ]; then
54      xecho "Usage: $(basename $0) standalone"
55    fi
56    ok=$true
57
58    #./configure.network.sh          || exit 1
59    ./install.prerequisites.sh     || exit 1
60    ./rabbitmq.sh      install    || exit 1
61    ./mysql.sh        install    || exit 1
62    ./keystone.sh      install    || exit 1
63    ./keystone.sh      installData all || exit 1
64    ./glance.sh       install    || exit 1
65    ./glance.sh       installData all || exit 1
66    ./cinder.sh       install    || exit 1
67    ./kvm.sh         install    || exit 1
68    ./open-vswitch.sh install    || exit 1
69    ./quantum.sh      install standalone || exit 1
70  # ./cinder.sh      installData    || exit 1
71  # ./swift.sh       install standalone // exit 1
72    ./nova.sh        install standalone || exit 1
73    ./dashboard.sh    install    || exit 1
74  }
75
76 controller()
77 {
78  if [ $# -ne 0 ]; then
79    xecho "Usage: $(basename $0) controller"
80  fi
81  ok=$true
82
83  ./configure.network.sh          || exit 1
84  ./install.prerequisites.sh     || exit 1
85  ./rabbitmq.sh      install    || exit 1
86  ./mysql.sh        install    || exit 1
87  ./keystone.sh      install    || exit 1
88  ./keystone.sh      installData all || exit 1
89  ./glance.sh       install    || exit 1
90  ./glance.sh       installData all || exit 1
91  # ./open-vswitch.sh install    || exit 1
92  # ./quantum.sh      install controller // exit 1
93  ./cinder.sh       install    || exit 1
94  # ./cinder.sh      installData    || exit 1
95  # ./swift.sh       install controller // exit 1
96  ./nova.sh        install controller || exit 1
97  ./dashboard.sh    install    || exit 1
98  }
99
100 compute()
101 {
102  if [ $# -ne 0 ]; then
103    xecho "Usage: $(basename $0) compute"
104  fi
105  ok=$true
106
107  ./configure.network.sh          || exit 1
108  ./install.prerequisites.sh     || exit 1
109  # ./open-vswitch.sh install    || exit 1
110  # ./quantum.sh      install compute // exit 1
111  ./kvm.sh         install    || exit 1
112  ./nova.sh        install compute || exit 1
113  }
114
115 uninstall()
116 {
117  if [ $# -ne 0 ]; then
118    xecho "Usage: $(basename $0) uninstall"
119  fi
120  ok=$true
121
122  ./mysql.sh       uninstall
123  ./rabbitmq.sh    uninstall
124  ./cinder.sh      uninstall

```

```

125 ./dashboard.sh      uninstall
126 ./glance.sh        uninstall
127 ./keystone.sh      uninstall
128 ./nova.sh          uninstall
129 #./swift.sh         uninstall
130 ./quantum.sh       uninstall
131 ./open-vswitch.sh  uninstall
132 ./kvm.sh           uninstall
133 }
134
135 service()
136 {
137     if [ $# -ne 1 ]; then
138         xecho "Usage: $(basename $0) service operation\n\toperation : { start, stop, restart, ... }"
139     fi
140     ok=$true
141
142     # FIXME If a component is not installed = error
143     op=$1
144     ./mysql.sh          service "$op" || exit 1
145     ./rabbitmq.sh       service "$op" || exit 1
146     ./cinder.sh         service "$op" || exit 1
147     ./dashboard.sh      service "$op" || exit 1
148     ./glance.sh         service "$op" || exit 1
149     ./keystone.sh       service "$op" || exit 1
150     ./nova.sh          service "$op" || exit 1
151     #./swift.sh          service "$op" || exit 1
152     ./quantum.sh        service "$op" || exit 1
153     ./open-vswitch.sh   service "$op" || exit 1
154     ./kvm.sh            service "$op" || exit 1
155 }
156
157 main "$@"

```

6.15.20 scripts/setup-helper.sh

```

1  {
2     echo "OSCIED Setup Helper - $(date)"
3     # FIXME this is hardcoded, a better solution ?
4     cd nodes || { echo '[ERROR] nodes directory not found !' 1>&2; exit 1; }
5     svn cleanup && svn update --username='oscied' --password='41nVGq07GmSgK9Ud' --non-interactive \
6         --accept theirs-full
7     sudo chown $UID:$UID . -R
8     cd scripts && sh setup.sh "$@" || echo '[ERROR] Unable to execute setup.sh properly' 1>&2
9 } 2>&1 | tee setup.log

```

6.15.21 scripts/tests.sh

```

1  # MY TRIALS/TESTS WITH QUANTUM
2
3  # nova-network XOR quantum
4  # http://docs.openstack.org/folsom/openstack-network/admin/content/nova_with_quantum.html
5
6  # Quantum L3 agent cannot ping directly, must use :
7  # -> ip nets ns exec qrouter... ip addr list
8  # -> ip nets ns exec qrouter... ping <fixed_ip>
9  # http://docs.openstack.org/trunk/openstack-network/admin/content/install_quantum-l3.html
10
11 # Common security group workflow
12 # http://docs.openstack.org/trunk/openstack-network/admin/content/securitygroup_workflow.html
13
14 . ./common.sh
15
16 getId tenant $TENANT_NAME || xecho '1'; tId=$id
17 parseId sh quantum.sh try net-create --tenant-id $tId demoNet          || xecho '2'; nId=$id
18 parseId sh quantum.sh try subnet-create --tenant-id $tId demoNet 50.50.1.0/24 || xecho '3'; sId=$id
19 parseId sh quantum.sh try router-create --tenant-id $tId demoRouter      || xecho '4'; rId=$id
20 sh quantum.sh try router-interface-add $rId $sId || exit 1
21

```

```

22 exit 0
23 sh quantum.sh try net-create --tenant-id 4e2c131ac12946658ffe984feb5ce0e sharednet1 --shared \
24   --provider:network_type flat --provider:physical_network physnet1
25
26 nid=
27
28 sh quantum.sh try subnet-create --tenant-id $tid sharednet1 192.168.1.0/24
29
30 sid=
31
32 sh nova.sh try nova boot --image 'Ubuntu 12.04 UEC' --flavor 1 --nic net-id=$nid vmS

```

6.15.22 scripts/tokengen.sh

```

1 randpass 26 $false $true
2 echo

```

6.16 FFmpeg Documentation

6.16.1 Available Codecs

```

1 Codecs:
2 D..... = Decoding supported
3 .E..... = Encoding supported
4 ..V.... = Video codec
5 ..A.... = Audio codec
6 ..S.... = Subtitle codec
7 ....S.. = Supports draw_horiz_band
8 ....D. = Supports direct rendering method 1
9 ....T = Supports weird frame truncation
10 -----
11 D V D 4xm          4X Movie
12 D V D 8bps         QuickTime 8BPS video
13 D A D 8svx_exp    8SVX exponential
14 D A D 8svx_fib    8SVX fibonacci
15 EV    a64multi     Multicolor charset for Commodore 64
16 EV    a64multi5   Multicolor charset for Commodore 64, extended with 5th color (colram)
17 DEA D aac          Advanced Audio Coding
18 D A D aac_latm    AAC LATM (Advanced Audio Codec LATM syntax)
19 D V D aasc         Autodesk RLE
20 DEA D ac3          ATSC A/52A (AC-3)
21 EA    ac3_fixed    ATSC A/52A (AC-3)
22 D A D adpcm_4xm   ADPCM 4X Movie
23 DEA D adpcm_adx   SEGA CRI ADX ADPCM
24 D A D adpcm_ct    ADPCM Creative Technology
25 D A D adpcm_ea    ADPCM Electronic Arts
26 D A D adpcm_ea_maxis_xa ADPCM Electronic Arts Maxis CDROM XA
27 D A D adpcm_ea_r1  ADPCM Electronic Arts R1
28 D A D adpcm_ea_r2  ADPCM Electronic Arts R2
29 D A D adpcm_ea_r3  ADPCM Electronic Arts R3
30 D A D adpcm_ea_xas ADPCM Electronic Arts XAS
31 D A D adpcm_ima_amv ADPCM IMA AMV
32 D A D adpcm_ima_apc ADPCM IMA CRYO APC
33 D A D adpcm_ima_dk3 ADPCM IMA Duck DK3
34 D A D adpcm_ima_dk4 ADPCM IMA Duck DK4
35 D A D adpcm_ima_ea_eacs ADPCM IMA Electronic Arts EACS
36 D A D adpcm_ima_ea_sead ADPCM IMA Electronic Arts SEAD
37 D A D adpcm_ima_iss ADPCM IMA Funcom ISS
38 DEA D adpcm_ima_qt ADPCM IMA QuickTime
39 D A D adpcm_ima_smjpeg ADPCM IMA Loki SDL MJPEG
40 DEA D adpcm_ima_wav ADPCM IMA WAV
41 D A D adpcm_ima_ws ADPCM IMA Westwood
42 DEA D adpcm_ms    ADPCM Microsoft
43 D A D adpcm_sbpro_2 ADPCM Sound Blaster Pro 2-bit
44 D A D adpcm_sbpro_3 ADPCM Sound Blaster Pro 2.6-bit
45 D A D adpcm_sbpro_4 ADPCM Sound Blaster Pro 4-bit
46 DEA D adpcm_swf   ADPCM Shockwave Flash

```

| | | | |
|-----|-------|----------------|---|
| 47 | D A D | adpcm_thp | ADPCM Nintendo Gamecube THP |
| 48 | D A D | adpcm_xa | ADPCM CDROM XA |
| 49 | DEA D | adpcm_yamaha | ADPCM Yamaha |
| 50 | DEA D | alac | ALAC (Apple Lossless Audio Codec) |
| 51 | D A D | als | MPEG-4 Audio Lossless Coding (ALS) |
| 52 | D A D | amrnb | Adaptive Multi-Rate NarrowBand |
| 53 | D A D | amrwb | Adaptive Multi-Rate WideBand |
| 54 | DEV | amv | AMV Video |
| 55 | D V D | anm | Deluxe Paint Animation |
| 56 | D V D | ansi | ASCII/ANSI art |
| 57 | D A D | ape | Monkey's Audio |
| 58 | DES | ass | Advanced SubStation Alpha subtitle |
| 59 | DEV D | asv1 | ASUS V1 |
| 60 | DEV D | asv2 | ASUS V2 |
| 61 | D A D | atrac1 | Atrac 1 (Adaptive TRansform Acoustic Coding) |
| 62 | D A D | atrac3 | Atrac 3 (Adaptive TRansform Acoustic Coding 3) |
| 63 | D V D | aura | Auravision AURA |
| 64 | D V D | aura2 | Auravision Aura 2 |
| 65 | DEV D | avrp | Avid 1:1 10-bit RGB Packer |
| 66 | D V D | avs | AVS (Audio Video Standard) video |
| 67 | D V D | bethsoftvid | Bethesda VID video |
| 68 | D V D | bfi | Brute Force & Ignorance |
| 69 | D A D | binkaudio_dct | Bink Audio (DCT) |
| 70 | D A D | binkaudio_rdft | Bink Audio (RDFT) |
| 71 | D V | binkvideo | Bink video |
| 72 | D V D | bintext | Binary text |
| 73 | DEV D | bmp | BMP image |
| 74 | D A D | bmv_audio | Discworld II BMV audio |
| 75 | D V | bmv_video | Discworld II BMV video |
| 76 | D V D | c93 | Interplay C93 |
| 77 | D V D | camstudio | CamStudio |
| 78 | D V D | camtasia | TechSmith Screen Capture Codec |
| 79 | D V D | cavs | Chinese AVS video (AVS1-P2, JiZhen profile) |
| 80 | D V D | cdgraphics | CD Graphics video |
| 81 | D V D | cinepak | Cinepak |
| 82 | DEV D | cljr | Cirrus Logic AccuPak |
| 83 | D A D | cook | COOK |
| 84 | D V D | cyuv | Creative YUV (CYUV) |
| 85 | DEA D | dca | DCA (DTS Coherent Acoustics) |
| 86 | D V D | dfa | Chronomaster DFA |
| 87 | D V | dirac | BBC Dirac VC-2 |
| 88 | DEV D | dnxhd | VC3/DNxHD |
| 89 | DEV | dpx | DPX image |
| 90 | D A D | dsicinaudio | Delphine Software International CIN audio |
| 91 | D V D | dsicinvideo | Delphine Software International CIN video |
| 92 | DES | dvbsub | DVB subtitles |
| 93 | DES | dvdsub | DVD subtitles |
| 94 | DEV D | dvvideo | DV (Digital Video) |
| 95 | D V D | dxa | Feeble Files/ScummVM DXA |
| 96 | D V D | dxtory | Dxtory |
| 97 | DEA D | eac3 | ATSC A/52 E-AC-3 |
| 98 | D V D | eacmv | Electronic Arts CMV video |
| 99 | D V D | eamad | Electronic Arts Madcow Video |
| 100 | D V D | eatgg | Electronic Arts TGQ video |
| 101 | D V | eatgv | Electronic Arts TGV video |
| 102 | D V D | eatqi | Electronic Arts TQI Video |
| 103 | D V D | escape124 | Escape 124 |
| 104 | D V D | escape130 | Escape 130 |
| 105 | DEV D | ffv1 | FFmpeg video codec #1 |
| 106 | DEVSD | ffvhuff | Huffyuv FFmpeg variant |
| 107 | DEA D | flac | FLAC (Free Lossless Audio Codec) |
| 108 | DEV D | flashsv | Flash Screen Video |
| 109 | DEV D | flashsv2 | Flash Screen Video Version 2 |
| 110 | D V D | flic | Autodesk Animator Flic video |
| 111 | DEVSD | flv | Flash Video (FLV) / Sorenson Spark / Sorenson H.263 |
| 112 | D V D | fraps | Fraps |
| 113 | D V D | frwu | Forward Uncompressed |
| 114 | DEA D | g722 | G.722 ADPCM |
| 115 | DEA | g723_1 | G.723.1 |
| 116 | DEA D | g726 | G.726 ADPCM |
| 117 | D A D | g729 | G.729 |
| 118 | DEV D | gif | GIF (Graphics Interchange Format) |
| 119 | D A D | gsm | GSM |

| | | | |
|-----|--------|------------------|--|
| 120 | D A D | gsm_ms | GSM Microsoft variant |
| 121 | DEV D | h261 | H.261 |
| 122 | DEVS D | h263 | H.263 / H.263-1996 |
| 123 | D VSD | h263i | Intel H.263 |
| 124 | EV | h263p | H.263+ / H.263-1998 / H.263 version 2 |
| 125 | D V D | h264 | H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10 |
| 126 | D V D | h264_vdpau | H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10 (VDPAU acceleration) |
| 127 | DEVSD | huffyuv | Huffyuv / HuffYUV |
| 128 | D V D | idcinvideo | id Quake II CIN video |
| 129 | D V D | idf | iCEDraw text |
| 130 | D V D | iff_byterun1 | IFF ByteRun1 |
| 131 | D V D | iff_ilbm | IFF ILBM |
| 132 | D A D | imc | IMC (Intel Music Coder) |
| 133 | D V D | indeo2 | Intel Indeo 2 |
| 134 | D V | indeo3 | Intel Indeo 3 |
| 135 | D V | indeo4 | Intel Indeo Video Interactive 4 |
| 136 | D V | indeo5 | Intel Indeo Video Interactive 5 |
| 137 | D A D | interplay_dpcm | DPCM Interplay |
| 138 | D V D | interplayvideo | Interplay MVE video |
| 139 | DEV | j2k | JPEG 2000 |
| 140 | DEV D | jpegls | JPEG-LS |
| 141 | D V D | jkv | Bitmap Brothers JV video |
| 142 | D V | kgv1 | Kega Game Video |
| 143 | D V D | kmvc | Karl Morton's video codec |
| 144 | D V D | lagarith | Lagarith lossless |
| 145 | DEA D | libgsm | libgsm GSM |
| 146 | DEA D | libgsm_ms | libgsm GSM Microsoft variant |
| 147 | EA | libmp3lame | libmp3lame MP3 (MPEG audio layer 3) |
| 148 | DEV D | libopenjpeg | OpenJPEG based JPEG 2000 encoder |
| 149 | DEV | libschroedinger | libschroedinger Dirac 2.2 |
| 150 | DEA D | libspeex | libspeex Speex |
| 151 | EV | libtheora | libtheora Theora |
| 152 | EA | libvorbis | libvorbis Vorbis |
| 153 | DEV | libvpx | libvpx VP8 |
| 154 | EV | libx264 | libx264 H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10 |
| 155 | EV | libx264rgb | libx264 H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10 RGB |
| 156 | EV | ljpeg | Lossless JPEG |
| 157 | D V D | loco | LOCO |
| 158 | D A D | mace3 | MACE (Macintosh Audio Compression/Expansion) 3:1 |
| 159 | D A D | mace6 | MACE (Macintosh Audio Compression/Expansion) 6:1 |
| 160 | D V D | mdec | Sony PlayStation MDEC (Motion DECoder) |
| 161 | D V D | mimic | Mimic |
| 162 | DEV D | mjpeg | MJPEG (Motion JPEG) |
| 163 | D V D | mjpegb | Apple MJPEG-B |
| 164 | D A D | mlp | MLP (Meridian Lossless Packing) |
| 165 | D V D | mmvideo | American Laser Games MM Video |
| 166 | D V D | motionpixels | Motion Pixels video |
| 167 | D A D | mp1 | MP1 (MPEG audio layer 1) |
| 168 | D A D | mplfloat | MP1 (MPEG audio layer 1) |
| 169 | DEA D | mp2 | MP2 (MPEG audio layer 2) |
| 170 | D A D | mp2float | MP2 (MPEG audio layer 2) |
| 171 | D A D | mp3 | MP3 (MPEG audio layer 3) |
| 172 | D A D | mp3adu | ADU (Application Data Unit) MP3 (MPEG audio layer 3) |
| 173 | D A D | mp3adufloat | ADU (Application Data Unit) MP3 (MPEG audio layer 3) |
| 174 | D A D | mp3float | MP3 (MPEG audio layer 3) |
| 175 | D A D | mp3on4 | MP3onMP4 |
| 176 | D A D | mp3on4float | MP3onMP4 |
| 177 | D A D | mpc7 | Musepack SV7 |
| 178 | D A D | mpc8 | Musepack SV8 |
| 179 | DEVS D | mpeg1video | MPEG-1 video |
| 180 | D V DT | mpeg1video_vdpau | MPEG-1 video (VDPAU acceleration) |
| 181 | DEVS D | mpeg2video | MPEG-2 video |
| 182 | DEVS D | mpeg4 | MPEG-4 part 2 |
| 183 | D V DT | mpeg4_vdpau | MPEG-4 part 2 (VDPAU) |
| 184 | D VSDT | mpegvideo | MPEG-1 video |
| 185 | D V DT | mpegvideo_vdpau | MPEG-1/2 video (VDPAU acceleration) |
| 186 | D VSDT | mpegvideo_xvmc | MPEG-1/2 video XvMC (X-Video Motion Compensation) |
| 187 | DEVSD | msmpeg4 | MPEG-4 part 2 Microsoft variant version 3 |
| 188 | D VSD | msmpeg4v1 | MPEG-4 part 2 Microsoft variant version 1 |
| 189 | DEVSD | msmpeg4v2 | MPEG-4 part 2 Microsoft variant version 2 |
| 190 | D V D | msrle | Microsoft RLE |
| 191 | DEV D | msvideo1 | Microsoft Video-1 |
| 192 | D V D | mszh | LCL (LossLess Codec Library) MSZH |

| | | | | | |
|-----|-----|---|---|------------------|---|
| 193 | D | V | D | mjpeg | Mobotix MxPEG video |
| 194 | DEA | D | | nellymoser | Nellymoser Asao |
| 195 | D | V | D | nuv | NuppelVideo/RTJPEG |
| 196 | DEV | D | | pam | PAM (Portable AnyMap) image |
| 197 | DEV | D | | pbm | PBM (Portable BitMap) image |
| 198 | DEA | D | | pcm_aaw | PCM A-law |
| 199 | D | A | D | pcm.bluray | PCM signed 16 20 24-bit big-endian for Blu-ray media |
| 200 | D | A | D | pcm_dvd | PCM signed 20 24-bit big-endian |
| 201 | DEA | D | | pcm_f32be | PCM 32-bit floating point big-endian |
| 202 | DEA | D | | pcm_f32le | PCM 32-bit floating point little-endian |
| 203 | DEA | D | | pcm_f64be | PCM 64-bit floating point big-endian |
| 204 | DEA | D | | pcm_f64le | PCM 64-bit floating point little-endian |
| 205 | D | A | D | pcm_lxf | PCM signed 20-bit little-endian planar |
| 206 | DEA | D | | pcm_mulaw | PCM mu-law |
| 207 | DEA | D | | pcm_s16be | PCM signed 16-bit big-endian |
| 208 | DEA | D | | pcm_s16le | PCM signed 16-bit little-endian |
| 209 | D | A | D | pcm_s16le_planar | PCM 16-bit little-endian planar |
| 210 | DEA | D | | pcm_s24be | PCM signed 24-bit big-endian |
| 211 | DEA | D | | pcm_s24daud | PCM D-Cinema audio signed 24-bit |
| 212 | DEA | D | | pcm_s24le | PCM signed 24-bit little-endian |
| 213 | DEA | D | | pcm_s32be | PCM signed 32-bit big-endian |
| 214 | DEA | D | | pcm_s32le | PCM signed 32-bit little-endian |
| 215 | DEA | D | | pcm_s8 | PCM signed 8-bit |
| 216 | D | A | D | pcm_s8_planar | PCM signed 8-bit planar |
| 217 | DEA | D | | pcm_u16be | PCM unsigned 16-bit big-endian |
| 218 | DEA | D | | pcm_u16le | PCM unsigned 16-bit little-endian |
| 219 | DEA | D | | pcm_u24be | PCM unsigned 24-bit big-endian |
| 220 | DEA | D | | pcm_u24le | PCM unsigned 24-bit little-endian |
| 221 | DEA | D | | pcm_u32be | PCM unsigned 32-bit big-endian |
| 222 | DEA | D | | pcm_u32le | PCM unsigned 32-bit little-endian |
| 223 | DEA | D | | pcm_u8 | PCM unsigned 8-bit |
| 224 | D | A | D | pcm_zork | PCM Zork |
| 225 | DEV | D | | pcx | PC Paintbrush PCX image |
| 226 | DEV | D | | pgm | PGM (Portable GrayMap) image |
| 227 | DEV | D | | pgmyuv | PGMYUV (Portable GrayMap YUV) image |
| 228 | D | S | | pgssub | HDMV Presentation Graphic Stream subtitles |
| 229 | D | V | D | pictor | Pictor/PC Paint |
| 230 | DEV | D | | png | PNG image |
| 231 | DEV | D | | ppm | PPM (Portable PixelMap) image |
| 232 | DEV | D | | prores | Apple ProRes |
| 233 | D | V | D | prores_lgpl | Apple ProRes (iCodec Pro) |
| 234 | D | V | D | ptx | V.Flash PTX image |
| 235 | D | A | D | qcelp | QCELP / PureVoice |
| 236 | D | A | D | qdm2 | QDesign Music Codec 2 |
| 237 | D | V | D | qdraw | Apple QuickDraw |
| 238 | D | V | D | qpeg | Q-team QPEG |
| 239 | DEV | D | | qtrle | QuickTime Animation (RLE) video |
| 240 | DEV | D | | r10k | AJA Kona 10-bit RGB Codec |
| 241 | DEV | D | | r210 | Uncompressed RGB 10-bit |
| 242 | DEV | | | rawvideo | raw video |
| 243 | DEA | D | | real_144 | RealAudio 1.0 (14.4K) encoder |
| 244 | D | A | D | real_288 | RealAudio 2.0 (28.8K) |
| 245 | D | V | D | r12 | RL2 video |
| 246 | DEA | D | | roq_dpcm | id RoQ DPCM |
| 247 | DEV | D | | roqvideo | id RoQ video |
| 248 | D | V | D | rpza | QuickTime video (RPZA) |
| 249 | DEV | D | | rv10 | RealVideo 1.0 |
| 250 | DEV | D | | rv20 | RealVideo 2.0 |
| 251 | D | V | D | rv30 | RealVideo 3.0 |
| 252 | D | V | D | rv40 | RealVideo 4.0 |
| 253 | D | A | D | s302m | SMPTE 302M |
| 254 | DEV | | | sgi | SGI image |
| 255 | D | A | D | shorten | Shorten |
| 256 | D | A | D | sipr | RealAudio SIPR / ACELP.NET |
| 257 | D | A | D | smackaud | Smacker audio |
| 258 | D | V | D | smackvid | Smacker video |
| 259 | D | V | D | smc | QuickTime Graphics (SMC) |
| 260 | DEV | D | | snow | Snow |
| 261 | D | A | D | sol_dpcm | DPCM Sol |
| 262 | DEA | D | | sonic | Sonic |
| 263 | EA | | | sonicls | Sonic lossless |
| 264 | D | V | D | sp5x | Sunplus JPEG (SP5X) |
| 265 | DES | | | srt | SubRip subtitle |

| | | | |
|-----|-------|-----------------|--|
| 266 | D V D | sunrast | Sun Rasterfile image |
| 267 | DEV D | svq1 | Sorenson Vector Quantizer 1 / Sorenson Video 1 / SVQ1 |
| 268 | D VSD | svq3 | Sorenson Vector Quantizer 3 / Sorenson Video 3 / SVQ3 |
| 269 | DEV D | targa | Truevision Targa image |
| 270 | D VSD | theora | Theora |
| 271 | D V D | thp | Nintendo Gamecube THP video |
| 272 | D V D | tierTEXseqvideo | Tiertex Limited SEQ video |
| 273 | DEV D | tiff | TIFF image |
| 274 | D V D | tmv | 8088flex TMV |
| 275 | D A D | truehd | TrueHD |
| 276 | D V D | truemotion1 | Duck TrueMotion 1.0 |
| 277 | D V D | truemotion2 | Duck TrueMotion 2.0 |
| 278 | D A D | truespeech | DSP Group TrueSpeech |
| 279 | D A D | tta | True Audio (TTA) |
| 280 | D A D | twinvq | VQF TwinVQ |
| 281 | D V D | txd | Renderware TXD (TeXture Dictionary) image |
| 282 | D V D | ultimotion | IBM UltiMotion |
| 283 | D V D | utvideo | Ut Video |
| 284 | DEV D | v210 | Uncompressed 4:2:2 10-bit |
| 285 | D V D | v210x | Uncompressed 4:2:2 10-bit |
| 286 | DEV D | v308 | Uncompressed packed 4:4:4 |
| 287 | DEV D | v410 | Uncompressed 4:4:4 10-bit |
| 288 | D V | vb | Beam Software VB |
| 289 | D V D | vble | VBLE Lossless Codec |
| 290 | D V D | vcl | SMPTE VC-1 |
| 291 | D V D | vcl_vdpau | SMPTE VC-1 VDPAU |
| 292 | D V D | vclimage | Windows Media Video 9 Image v2 |
| 293 | D V D | vcrl | ATI VCR1 |
| 294 | D A D | vmdaudio | Sierra VMD audio |
| 295 | D V D | vmddvideo | Sierra VMD video |
| 296 | D V D | vmnc | VMware Screen Codec / VMware Video |
| 297 | DEA D | vorbis | Vorbis |
| 298 | D VSD | vp3 | On2 VP3 |
| 299 | D V D | vp5 | On2 VP5 |
| 300 | D V D | vp6 | On2 VP6 |
| 301 | D V D | vp6a | On2 VP6 (Flash version, with alpha channel) |
| 302 | D V D | vp6f | On2 VP6 (Flash version) |
| 303 | D V D | vp8 | On2 VP8 |
| 304 | D V D | vqavideo | Westwood Studios VQA (Vector Quantized Animation) video |
| 305 | D A D | wavesynth | Wave synthesis pseudo-codec |
| 306 | D A D | wavpack | WavPack |
| 307 | D A | wmalossless | Windows Media Audio 9 Lossless |
| 308 | D A D | wmapro | Windows Media Audio 9 Professional |
| 309 | DEA D | wmav1 | Windows Media Audio 1 |
| 310 | DEA D | wmav2 | Windows Media Audio 2 |
| 311 | D A D | wmavoice | Windows Media Audio Voice |
| 312 | DEVSD | wmv1 | Windows Media Video 7 |
| 313 | DEVSD | wmv2 | Windows Media Video 8 |
| 314 | D V D | wmv3 | Windows Media Video 9 |
| 315 | D V D | wmv3_vdpau | Windows Media Video 9 VDPAU |
| 316 | D V D | wmv3image | Windows Media Video 9 Image |
| 317 | D V D | wnv1 | Winnov WNV1 |
| 318 | D A D | ws_snd1 | Westwood Audio (SND1) |
| 319 | D A D | xan_dpcm | DPCM Xan |
| 320 | D V D | xan_wc3 | Wing Commander III / Xan |
| 321 | D V D | xan_wc4 | Wing Commander IV / Xxan |
| 322 | D V D | xbin | eXtended BINary text |
| 323 | D V D | x1 | Miro VideoXL |
| 324 | DES | xsub | DivX subtitles (XSUB) |
| 325 | DEV D | xwd | XWD (X Window Dump) image |
| 326 | DEV D | y41p | Uncompressed YUV 4:1:1 12-bit |
| 327 | D V | yop | Psygnosis YOP Video |
| 328 | DEV D | yuv4 | Uncompressed packed 4:2:0 |
| 329 | DEV D | zlib | LCL (LossLess Codec Library) ZLIB |
| 330 | DEV D | zmbv | Zip Motion Blocks Video |
| 331 | | | |
| 332 | | | Note, the names of encoders and decoders do not always match, so there are |
| 333 | | | several cases where the above table shows encoder only or decoder only entries |
| 334 | | | even though both encoding and decoding are supported. For example, the h263 |
| 335 | | | decoder corresponds to the h263 and h263p encoders, for file formats it is even |
| 336 | | | worse. |

6.16.2 Available Filters

| | | |
|----|---------------|---|
| 1 | Filters: | |
| 2 | aconvert | A->A Convert the input audio to sample_fmt:channel_layout:packed_fmt. |
| 3 | aformat | A->A Convert the input audio to one of the specified formats. |
| 4 | amerge | AA->A Merge two audio streams into a single multi-channel stream. |
| 5 | anull | A->A Pass the source unchanged to the output. |
| 6 | aresample | A->A Resample audio data. |
| 7 | ashowinfo | A->A Show textual information for each audio frame. |
| 8 | asplit | A->AA Pass on the audio input to two outputs. |
| 9 | astreamsync | AA->AA Copy two streams of audio data in a configurable order. |
| 10 | earwax | A->A Widen the stereo image. |
| 11 | pan | A->A Remix channels with coefficients (panning). |
| 12 | silencedetect | A->A Detect silence. |
| 13 | volume | A->A Change input volume. |
| 14 | abuffer | ->A Buffer audio frames, and make them accessible to the filterchain. |
| 15 | aevalsrc | ->A Generate an audio signal generated by an expression. |
| 16 | amovie | ->A Read audio from a movie source. |
| 17 | anullsrc | ->A Null audio source, return empty audio frames. |
| 18 | abuffersink | A-> Buffer audio frames, and make them available to the end of the filter graph. |
| 19 | anullsink | A-> Do absolutely nothing with the input audio. |
| 20 | blackframe | V->V Detect frames that are (almost) black. |
| 21 | boxblur | V->V Blur the input. |
| 22 | copy | V->V Copy the input video unchanged to the output. |
| 23 | crop | V->V Crop the input video to width:height:x:y. |
| 24 | cropdetect | V->V Auto-detect crop size. |
| 25 | delogo | V->V Remove logo from input video. |
| 26 | deshake | V->V Stabilize shaky video. |
| 27 | drawbox | V->V Draw a colored box on the input video. |
| 28 | drawtext | V->V Draw text on top of video frames using libfreetype library. |
| 29 | fade | V->V Fade in/out input video. |
| 30 | fieldorder | V->V Set the field order. |
| 31 | fifo | V->V Buffer input images and send them when they are requested. |
| 32 | format | V->V Convert the input video to one of the specified pixel formats. |
| 33 | frei0r | V->V Apply a frei0r effect. |
| 34 | gradfun | V->V Debands video quickly using gradients. |
| 35 | hflip | V->V Horizontally flip the input video. |
| 36 | hqdn3d | V->V Apply a High Quality 3D Denoiser. |
| 37 | lut | V->V Compute and apply a lookup table to the RGB/YUV input video. |
| 38 | lutrgb | V->V Compute and apply a lookup table to the RGB input video. |
| 39 | lutyuv | V->V Compute and apply a lookup table to the YUV input video. |
| 40 | mp | V->V Apply a libmpcodecs filter to the input video. |
| 41 | negate | V->V Negate input video. |
| 42 | noformat | V->V Force libavfilter not to use any of the specified pixel formats for the input to t |
| 43 | null | V->V Pass the source unchanged to the output. |
| 44 | ocv | V->V Apply transform using libopencv. |
| 45 | overlay | VV->V Overlay a video source on top of the input. |
| 46 | pad | V->V Pad input image to width:height[:x:y[:color]] (default x and y: 0, default color: |
| 47 | pixdescstest | V->V Test pixel format definitions. |
| 48 | scale | V->V Scale the input video to width:height size and/or convert the image format. |
| 49 | select | V->V Select frames to pass in output. |
| 50 | setdar | V->V Set the frame display aspect ratio. |
| 51 | setpts | V->V Set PTS for the output video frame. |
| 52 | setsar | V->V Set the pixel sample aspect ratio. |
| 53 | settb | V->V Set timebase for the output link. |
| 54 | showinfo | V->V Show textual information for each video frame. |
| 55 | slicify | V->V Pass the images of input video on to next video filter as multiple slices. |
| 56 | split | V->VV Pass on the input to two outputs. |
| 57 | swapuv | V->V Swap U and V components. |
| 58 | thumbnail | V->V Select the most representative frame in a given sequence of consecutive frames. |
| 59 | tinterlace | V->V Perform temporal field interlacing. |
| 60 | transpose | V->V Transpose input video. |
| 61 | unsharp | V->V Sharpen or blur the input video. |
| 62 | vflip | V->V Flip the input video vertically. |
| 63 | yadif | V->V Deinterlace the input image. |
| 64 | cellauto | ->V Create pattern generated by an elementary cellular automaton. |
| 65 | color | ->V Provide an uniformly colored input, syntax is: [color[:size[:rate]]]. |
| 66 | frei0r_src | ->V Generate a frei0r source. |
| 67 | life | ->V Create life. |
| 68 | mandelbrot | ->V Render a Mandelbrot fractal. |
| 69 | movie | ->V Read from a movie source. |
| 70 | mptestsrc | ->V Generate various test pattern. |
| 71 | nullsrc | ->V Null video source, return unprocessed video frames. |

```

72  rgbtessrc      | ->V   Generate RGB test pattern.
73  testsrc        | ->V   Generate test pattern.
74  buffersink     V->|   Buffer video frames, and make them available to the end of the filter graph.
75  nullsink       V->|   Do absolutely nothing with the input video.
76  buffer         | ->V   Buffer video frames, and make them accessible to the filterchain.

```

6.16.3 Available Formats

File formats:

| | |
|----|--|
| D. | = Demuxing supported |
| .E | = Muxing supported |
| -- | |
| E | 3g2 3GP2 format |
| E | 3gp 3GP format |
| D | 4xm 4X Technologies format |
| D | IFF IFF format |
| D | ISS Funcom ISS format |
| D | MTV MTV format |
| DE | RoQ raw id RoQ format |
| E | a64 a64 – video for Commodore 64 |
| D | aac raw ADTS AAC |
| DE | ac3 raw AC-3 |
| D | act ACT Voice file format |
| D | adf Artworx Data Format |
| E | adts ADTS AAC |
| DE | adx CRI ADX |
| D | aea MD STUDIO audio |
| DE | aiff Audio IFF |
| DE | alaw PCM A-law format |
| DE | alsa ALSA audio output |
| DE | amr 3GPP AMR file format |
| D | anm Deluxe Paint Animation |
| D | apc CRYO APC format |
| D | ape Monkey's Audio |
| D | applehttp Apple HTTP Live Streaming format |
| DE | asf ASF format |
| E | asf_stream ASF format |
| DE | ass Advanced SubStation Alpha subtitle format |
| DE | au SUN AU format |
| DE | avi AVI format |
| E | avm2 Flash 9 (AVM2) format |
| D | avs AVS format |
| D | bethsoftvid Bethesda Softworks VID format |
| D | bfi Brute Force & Ignorance |
| D | bin Binary text |
| D | bink Bink |
| DE | bit G.729 BIT file format |
| D | bmv Discworld II BMV |
| D | c93 Interplay C93 |
| DE | caf Apple Core Audio Format |
| DE | cavsvideo raw Chinese AVS video |
| D | cdg CD Graphics Format |
| E | crc CRC testing format |
| DE | daud D-Cinema audio format |
| D | dfa Chronomaster DFA |
| DE | dirac raw Dirac |
| DE | dnxhd raw DNxHD (SMPTE VC-3) |
| D | dsicin Delphine Software International CIN format |
| DE | dts raw DTS |
| DE | dv DV video format |
| D | dv1394 DV1394 A/V grab |
| E | dvd MPEG-2 PS format (DVD VOB) |
| D | dxa DXA |

| | |
|---------------------------|---|
| D ea | Electronic Arts Multimedia Format |
| D ea_cdata | Electronic Arts cdata |
| DE eac3 | raw E-AC-3 |
| DE f32be | PCM 32 bit floating-point big-endian format |
| DE f32le | PCM 32 bit floating-point little-endian format |
| DE f64be | PCM 64 bit floating-point big-endian format |
| DE f64le | PCM 64 bit floating-point little-endian format |
| D fbdev | Linux framebuffer |
| DE ffm | FFM (FFserver live feed) format |
| DE ffmetadata | FFmpeg metadata in text format |
| D film_cpk | Sega FILM/CPK format |
| DE filmstrip | Adobe Filmstrip |
| DE flac | raw FLAC |
| D flic | FLI/FLC/FLX animation format |
| DE flv | FLV format |
| E framecrc | framecrc testing format |
| E framemd5 | Per-frame MD5 testing format |
| DE g722 | raw G.722 |
| DE g723_1 | raw G.723.1 |
| D g729 | G.729 raw format demuxer |
| E gif | GIF Animation |
| D gsm | raw GSM |
| DE gxf | GXF format |
| DE h261 | raw H.261 |
| DE h263 | raw H.263 |
| DE h264 | raw H.264 video format |
| D ico | Microsoft Windows ICO |
| D idcin | id Cinematic format |
| D idf | iCE Draw File |
| DE image2 | image2 sequence |
| DE image2pipe | piped image2 sequence |
| D ingenient | raw Ingenient MJPEG |
| D ipmovie | Interplay MVE format |
| E ipod | iPod H.264 MP4 format |
| E ismv | ISMV/ISMA (Smooth Streaming) format |
| D iv8 | A format generated by IndigoVision 8000 video server |
| DE ivf | On2 IVF |
| D jack | JACK Audio Connection Kit |
| D jv | Bitmap Brothers JV |
| DE latm | LOAS/LATM |
| D lavfi | Libavfilter virtual input device |
| D libcdio | |
| D libdc1394 | dc1394 A/V grab |
| D lmlm4 | lmlm4 raw format |
| D loas | LOAS AudioSyncStream |
| D lxf | VR native stream format (LXF) |
| DE m4v | raw MPEG-4 video format |
| E matroska | Matroska file format |
| D matroska,webm | Matroska/WebM file format |
| E md5 | MD5 testing format |
| DE microdvd | MicroDVD subtitle format |
| DE mjpeg | raw MJPEG video |
| E mkvtimestamp_v2 | extract pts as timecode v2 format, as defined by mkvtoolnix |
| DE mlp | raw MLP |
| D mm | American Laser Games MM format |
| DE mmf | Yamaha SMAF |
| E mov | MOV format |
| D mov,mp4,m4a,3gp,3g2,mj2 | QuickTime/MPEG-4/Motion JPEG 2000 format |
| E mp2 | MPEG audio layer 2 |
| DE mp3 | MPEG audio layer 3 |
| E mp4 | MP4 format |
| D mpc | Musepack |
| D mpc8 | Musepack SV8 |

| | |
|--------------|--|
| DE mpeg | MPEG-1 System format |
| E mpeg1video | raw MPEG-1 video |
| E mpeg2video | raw MPEG-2 video |
| DE mpegs | MPEG-2 transport stream format |
| D mpegtstraw | MPEG-2 raw transport stream format |
| D mpegvideo | raw MPEG video |
| E mpjpeg | MIME multipart JPEG format |
| D msnwctcp | MSN TCP Webcam stream |
| DE mulaw | PCM mu-law format |
| D mvi | Motion Pixels MVI format |
| DE mxf | Material eXchange Format |
| E mxf_d10 | Material eXchange Format, D-10 Mapping |
| D mxg | MxPEG clip file format |
| D nc | NC camera feed format |
| D nsv | Nullsoft Streaming Video |
| E null | raw null video format |
| DE nut | NUT format |
| D nuv | NuppelVideo format |
| DE ogg | Ogg |
| DE oma | Sony OpenMG audio |
| DE oss | Open Sound System playback |
| D pmp | Playstation Portable PMP format |
| E psp | PSP MP4 format |
| D psxstr | Sony Playstation STR format |
| D pulse | Pulse audio input |
| D pva | TechnoTrend PVA file and stream format |
| D qcp | QCP format |
| D r3d | REDCODE R3D format |
| DE rawvideo | raw video format |
| E rcv | VC-1 test bitstream |
| D rl2 | RL2 format |
| DE rm | RealMedia format |
| D rpl | RPL/ARMovie format |
| DE rso | Lego Mindstorms RSO format |
| DE rtp | RTP output format |
| DE rtsp | RTSP output format |
| DE s16be | PCM signed 16 bit big-endian format |
| DE s16le | PCM signed 16 bit little-endian format |
| DE s24be | PCM signed 24 bit big-endian format |
| DE s24le | PCM signed 24 bit little-endian format |
| DE s32be | PCM signed 32 bit big-endian format |
| DE s32le | PCM signed 32 bit little-endian format |
| DE s8 | PCM signed 8 bit format |
| DE sap | SAP output format |
| D sbg | SBaGen binaural beats script |
| E sdl | SDL output device |
| D sdp | SDP |
| E segment | segment muxer |
| D shn | raw Shorten |
| D siff | Beam Software SIFF |
| DE smjpeg | Loki SDL MJPEG |
| D smk | Smacker video |
| D sol | Sierra SOL format |
| DE sox | SoX native format |
| DE spdif | IEC 61937 (used on S/PDIF - IEC958) |
| DE srt | SubRip subtitle format |
| E svcd | MPEG-2 PS format (VOB) |
| DE swf | Flash format |
| D thp | THP |
| D tiertexseq | Tiertex Limited SEQ format |
| D tmv | 8088flex TMV |
| DE truehd | raw TrueHD |
| D tta | True Audio |

| | |
|---------------------|---|
| D tty | Tele-typewriter |
| D txd | Renderware TeXture Dictionary |
| DE u16be | PCM unsigned 16 bit big-endian format |
| DE u16le | PCM unsigned 16 bit little-endian format |
| DE u24be | PCM unsigned 24 bit big-endian format |
| DE u24le | PCM unsigned 24 bit little-endian format |
| DE u32be | PCM unsigned 32 bit big-endian format |
| DE u32le | PCM unsigned 32 bit little-endian format |
| DE u8 | PCM unsigned 8 bit format |
| D vcl | raw VC-1 |
| D vcltest | VC-1 test bitstream format |
| E vcd | MPEG-1 System format (VCD) |
| D video4linux2,v4l2 | Video4Linux2 device grab |
| D vmd | Sierra VMD format |
| E vob | MPEG-2 PS format (VOB) |
| DE voc | Creative Voice file format |
| D vqf | Nippon Telegraph and Telephone Corporation (NTT) TwinVQ |
| D w64 | Sony Wave64 format |
| DE wav | WAV format |
| D wc3movie | Wing Commander III movie format |
| E webm | WebM file format |
| D wsaud | Westwood Studios audio format |
| D wsvqa | Westwood Studios VQA format |
| DE wtv | Windows Television (WTV) |
| D wv | WavPack |
| D x11grab | X11grab |
| D xa | Maxis XA File Format |
| D xbin | eXtended BINary text (XBIN) |
| D xmvc | Microsoft XMV |
| D xwma | Microsoft xWMA |
| D yop | Psygnosis YOP Format |
| DE yuv4mpegpipe | YUV4MPEG pipe format |

HTTP ROUTING TABLE

| | | |
|------------|---|---|
| / | GET /, 133 | GET /transform/profile/count, 123 GET /transform/profile/id/(id), 134 PATCH /transform/profile/id/(id), 135 PUT /transform/profile/id/(id), 135 DELETE /transform/profile/id/(id), 137 GET /transform/queue, 125 |
| /flush | POST /flush, 130 | |
| /index | GET /index, 130 | |
| /media | GET /media, 130 POST /media, 130 GET /media/HEAD, 129 GET /media/count, 129 GET /media/id/(id), 141 PATCH /media/id/(id), 142 PUT /media/id/(id), 142 DELETE /media/id/(id), 143 GET /media/id/(id)/HEAD, 141 | |
| /publish | POST /publish/callback, 127 GET /publish/job, 128 POST /publish/job, 128 GET /publish/job/HEAD, 123 GET /publish/job/count, 123 GET /publish/job/id/(id), 139 DELETE /publish/job/id/(id), 141 GET /publish/job/id/(id)/HEAD, 134 GET /publish/queue, 127 | |
| /publisher | GET /publisher/queue, 126 | |
| /transform | POST /transform/callback, 124 GET /transform/job, 126 POST /transform/job, 126 GET /transform/job/HEAD, 123 GET /transform/job/count, 123 GET /transform/job/id/(id), 137 DELETE /transform/job/id/(id), 139 GET /transform/job/id/(id)/HEAD, 133 GET /transform/profile, 124 POST /transform/profile, 124 | |

PYTHON MODULE INDEX

0

orchestra, 121