

Open-Source Cloud Infrastructure for Encoding and Distribution

By David Fischer,
Prof. : Andrés Revuelta
EBU : Bram Tullemans

In collaboration with



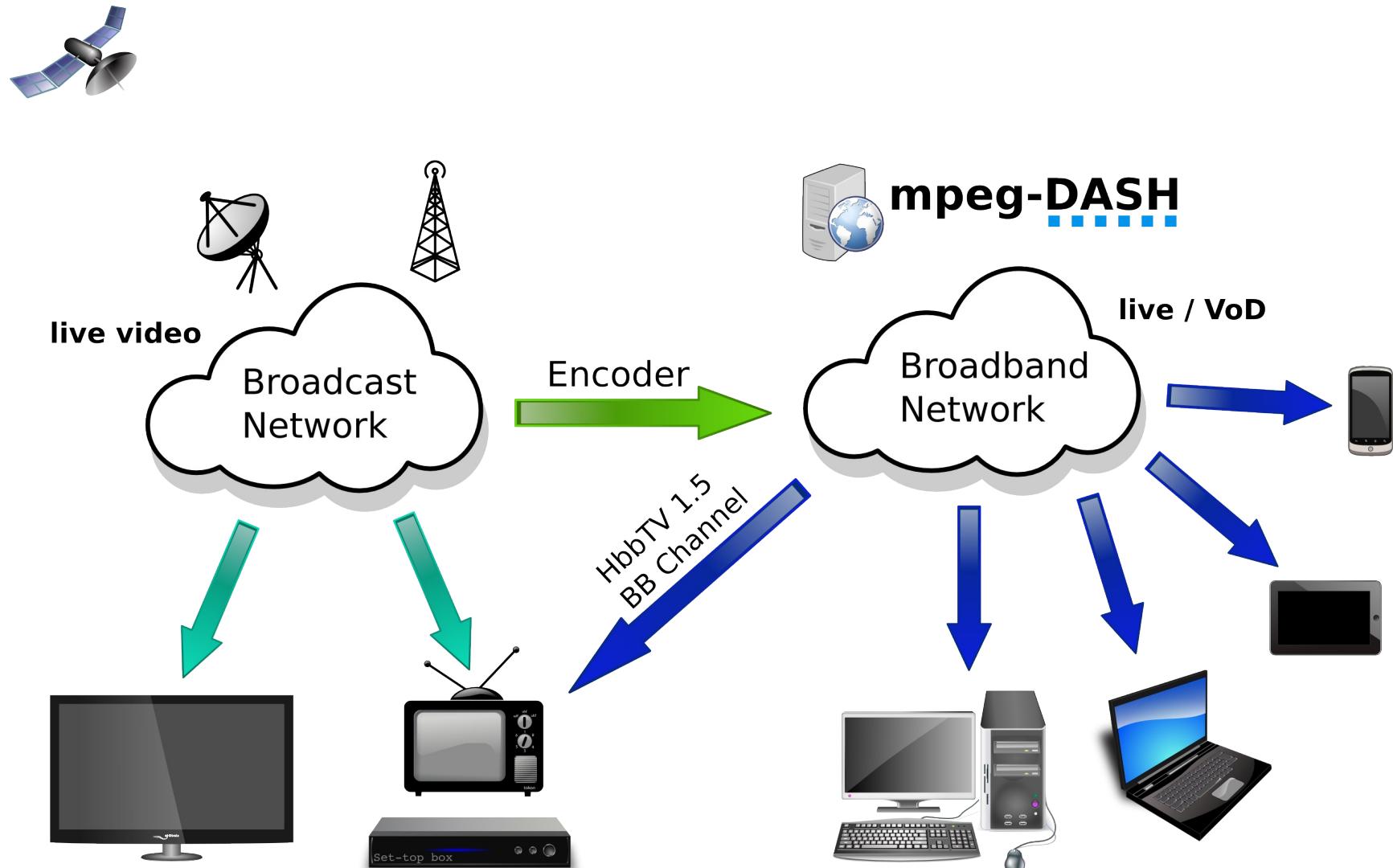
hepia

Haute école du paysage, d'ingénierie
et d'architecture de Genève

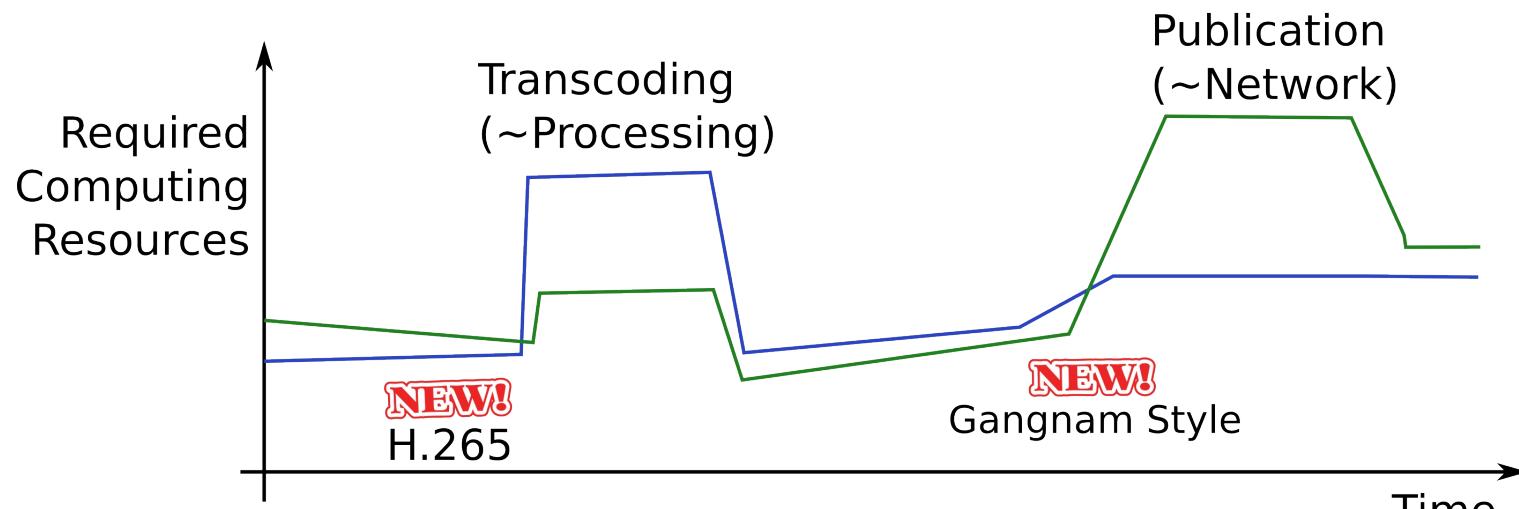
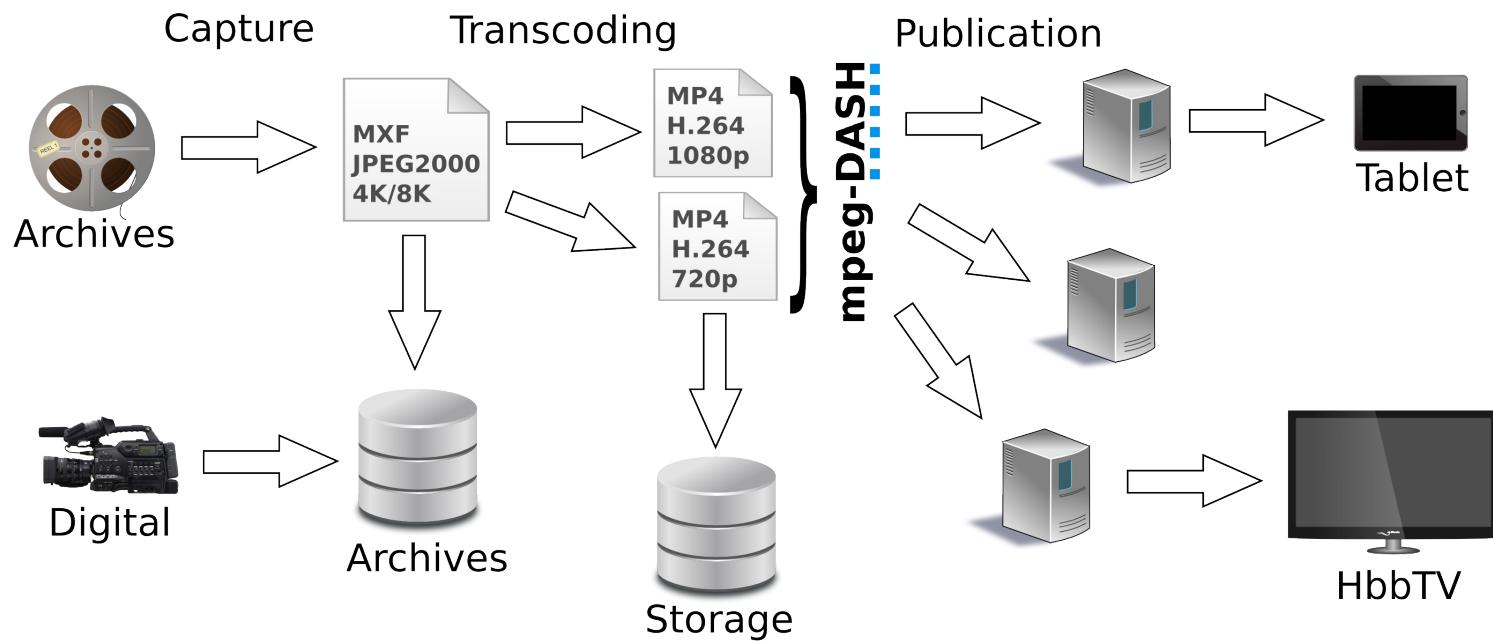
Plan

- Introduction
- OSCIED
- Layers of OSCIED
- Servers Layer
- Clouds Layer
- Services Orchestration Layer
- Application (Platform) Layer
- Conclusion
- Future
- Master HES-SO MA
- Questions & Answers

Introduction - Context



Introduction – Motivation



OSCIED

- Providing a scalable OSS media platform to members of EBU
- A platform based on cloud-era OSS technologies, dedicated to :
 - **Transcoding** of a wide collection of medias to new formats
 - **Online publication** of popular medias
- A platform that is designed to allow :
 - Deployment of services in-house, entreprise IT infrastructure
 - Scaling of transcoding and publication points to public clouds (IaaS)
 - Ease of scale-up or scale-down of services
 - Separation of development, testing and production environments
 - Future features or capabilities (MPEG-DASH, ...)

Layers of OSCIED (\neq OSI)

Application

Deployment

Clouds / IaaS

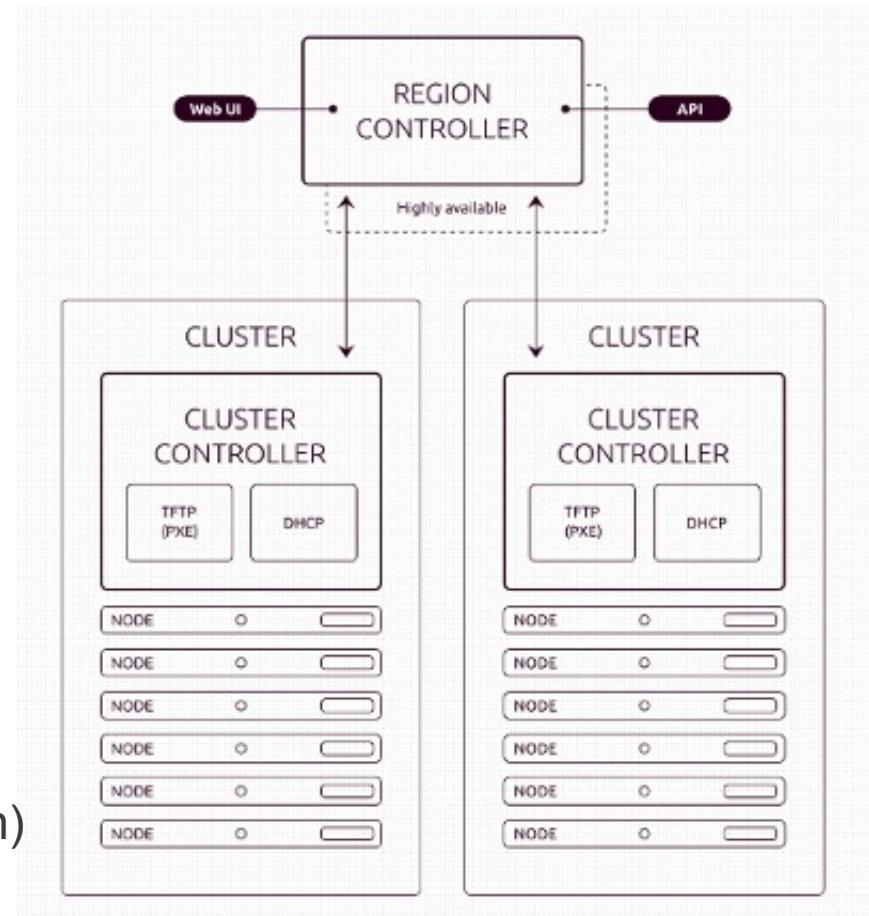
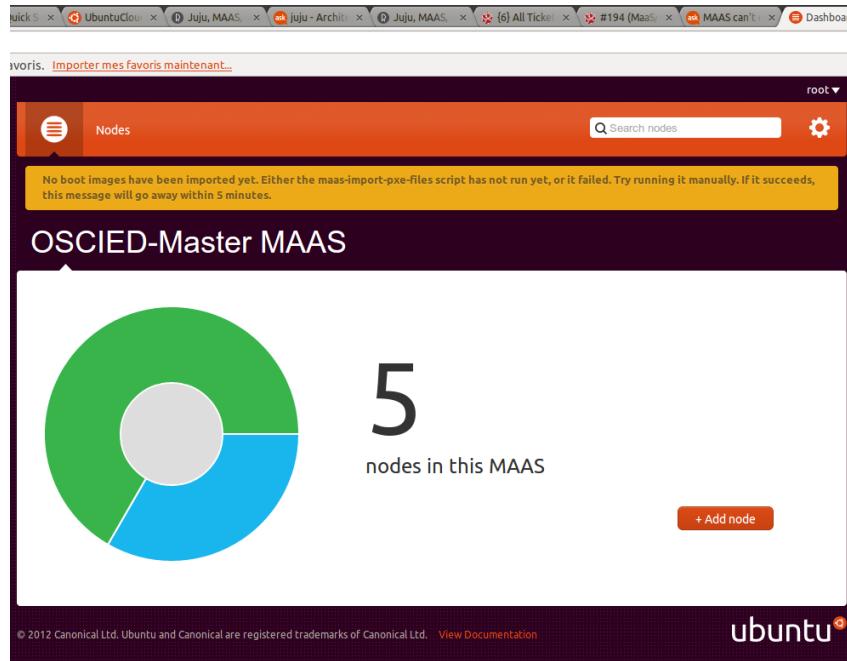
Servers

Servers Layer

- A setup of 4x 1U servers chosen based on :
 - Project's CAPEX (the cash)
 - Optimization of configuration based on needs
 - Ubuntu Server's certified HW list for Dell Servers
- Dell PowerEdge R420
 - 2x Xeon E5-2430 2.20GHz, 15MB Cache, 7.2GT/s, 6C
 - 4x 4GB RDIMM 1333MT/s x8 Data Width
 - HW RAID PERC H310 Mini + 2x 1TB 7.2K SATA



Servers Layer (MaaS)

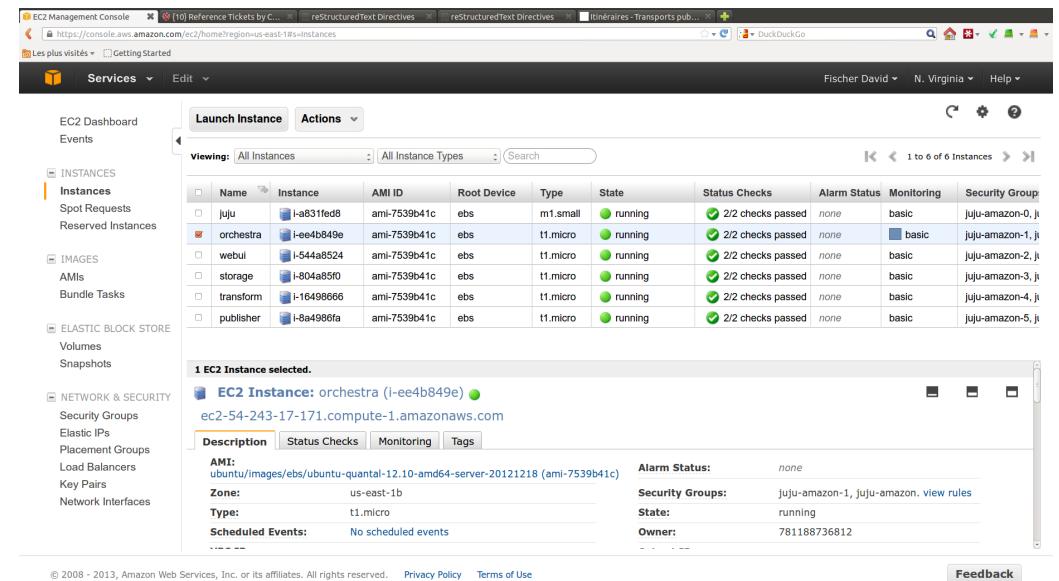


- ✓ Ease of servers provisioning (PXE install)
- ✓ Driveable through an API (enable automation)
- ✓ Developed to interact with JuJu

Clouds Layer (Amazon, ...)

« Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers. »

- Manageable through an **API** or **WUI**
- Compute / network services :
 - EC2 – Elastic Compute Cloud**
 - EMR – Cluster Hadoop
 - Route 53 – DNS
- Storage/CDN services :
 - S3 – Simple Storage Service**
 - Glacier – Low-cost Archive Storage
 - EBS – Elastic Block Storage
 - CloudFront – Content Delivery Net.
- Database services :
 - RDS – Relational DB Services
 - DynamoDB – NoSQL data store
- Management services :
 - AWS Management Console**

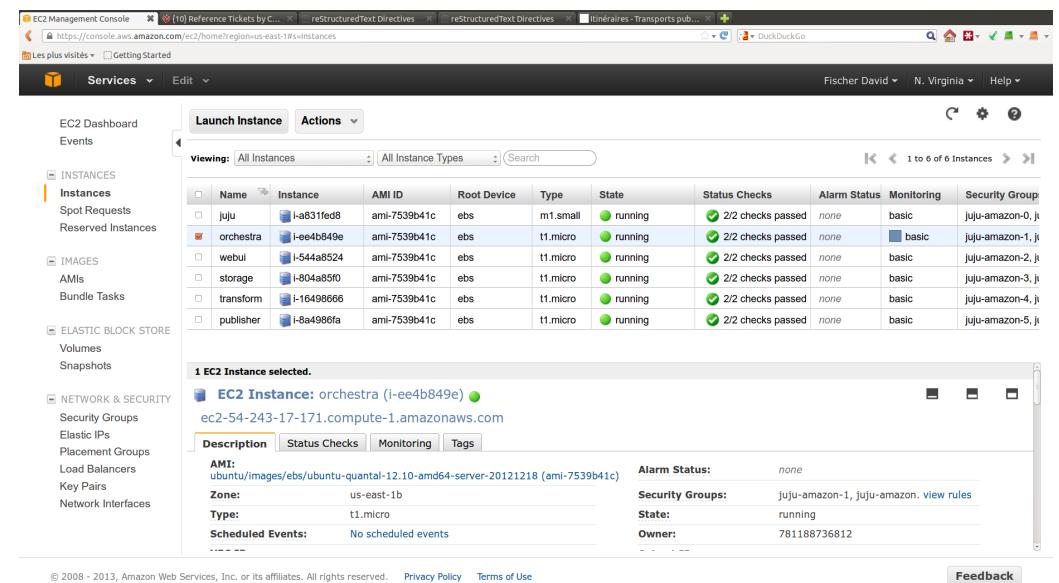


VM Instances can be launched **On-demand**, **Reserved** (for 2 years) or **Spot** (if price below 0.1\$/h). Instances Type : Define #CPU, #MB RAM, iops ... \$/h

Clouds Layer (Amazon, ...)

« Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers. »

- Manageable through an **API** or **WUI**
- Compute / network services :
 - EC2 – Elastic Compute Cloud**
 - .
 - .
- Storage/CDN services :
 - S3 – Simple Storage Service**
 - .
 - .
 - .
- Database services :
 - .
 - .
- Management services :
 - AWS Management Console**



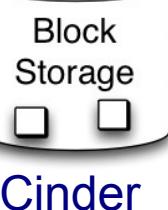
VM Instances can be launched **On-demand**, **Reserved** (for 2 years) or **Spot** (if price below 0.1\$/h). Instances Type : Define #CPU, #MB RAM, iops ... \$/h

Clouds Layer (OpenStack)



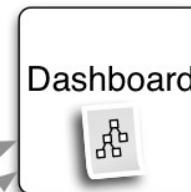
Nova services
(compute, network, image ...)

Quantum



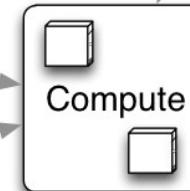
- ✓ Amazon EC2 compatible
- ✓ Open-Source, wide community
- ✓ Pluggable design
- ✓ (-) Documentation

Horizon

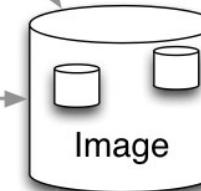


Rackspace's Swift
object storage

Compute



Glance



AWS S3
compatible :

- Ceph
- GlusterFS
- Swift

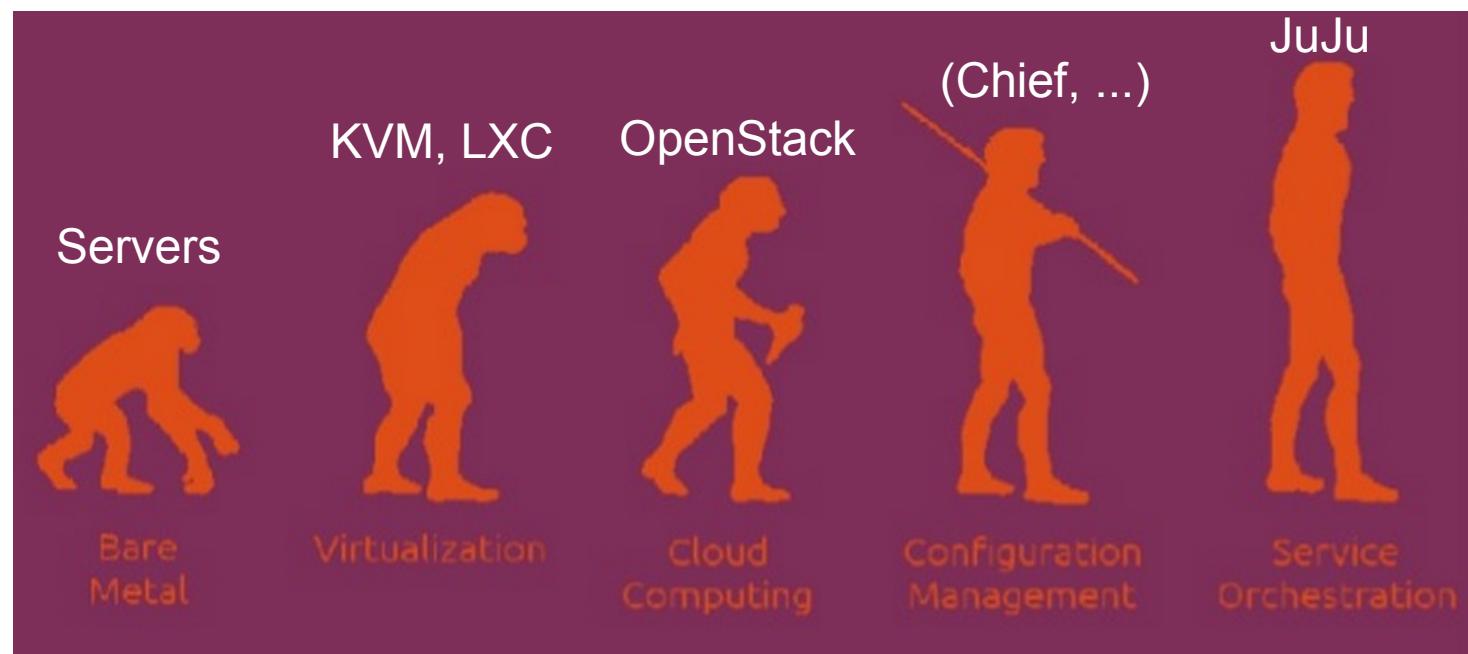


<http://ken.pepple.info>

Identity

Keystone

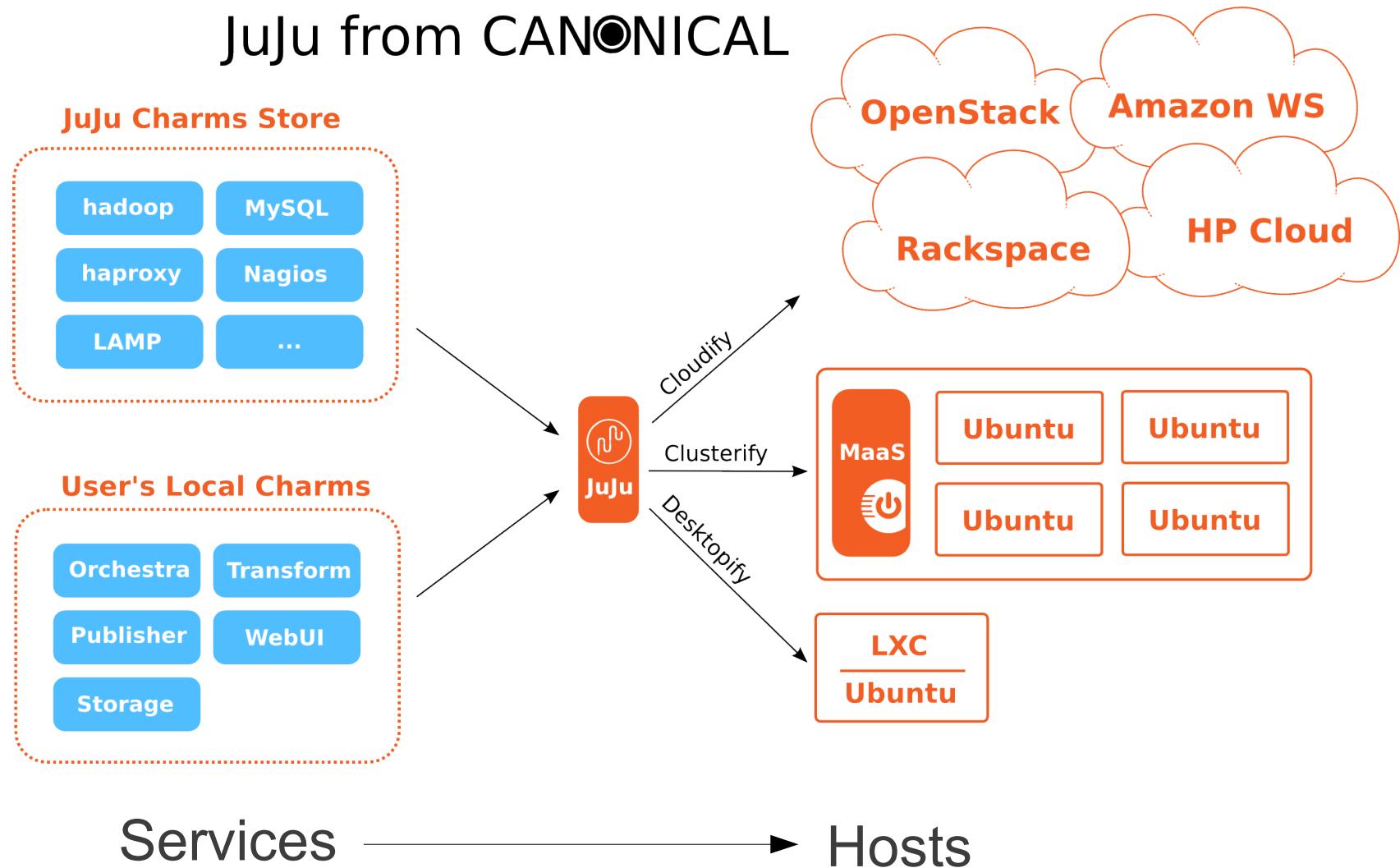
Services Orchestration ...



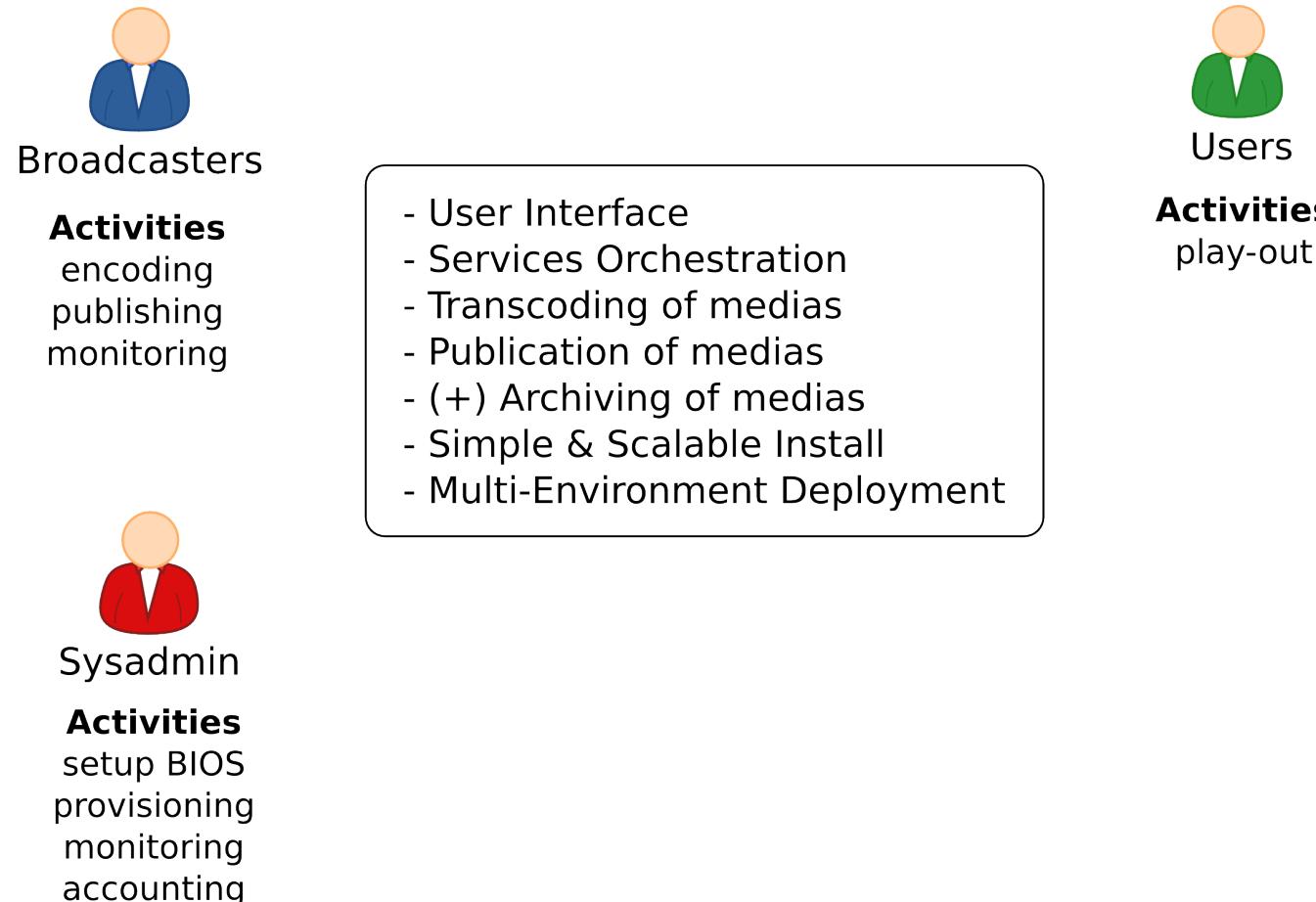
Copyright Hardik Dalwadi

Services Orchestration ...

JuJu from CANONICAL



Design Process (1/6)

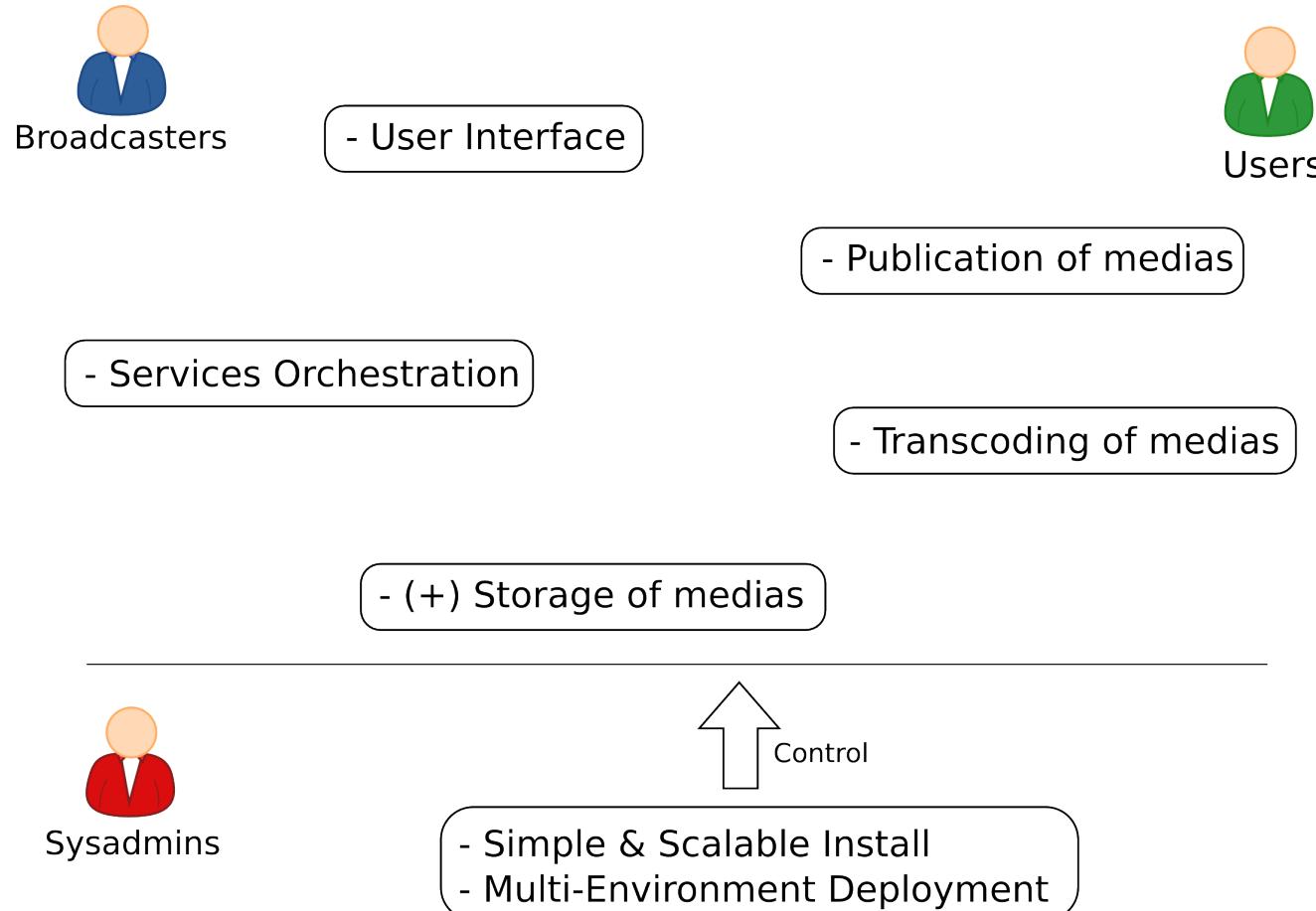


Design process steps

Listing of features based on uses-cases

Listing of actors and activities

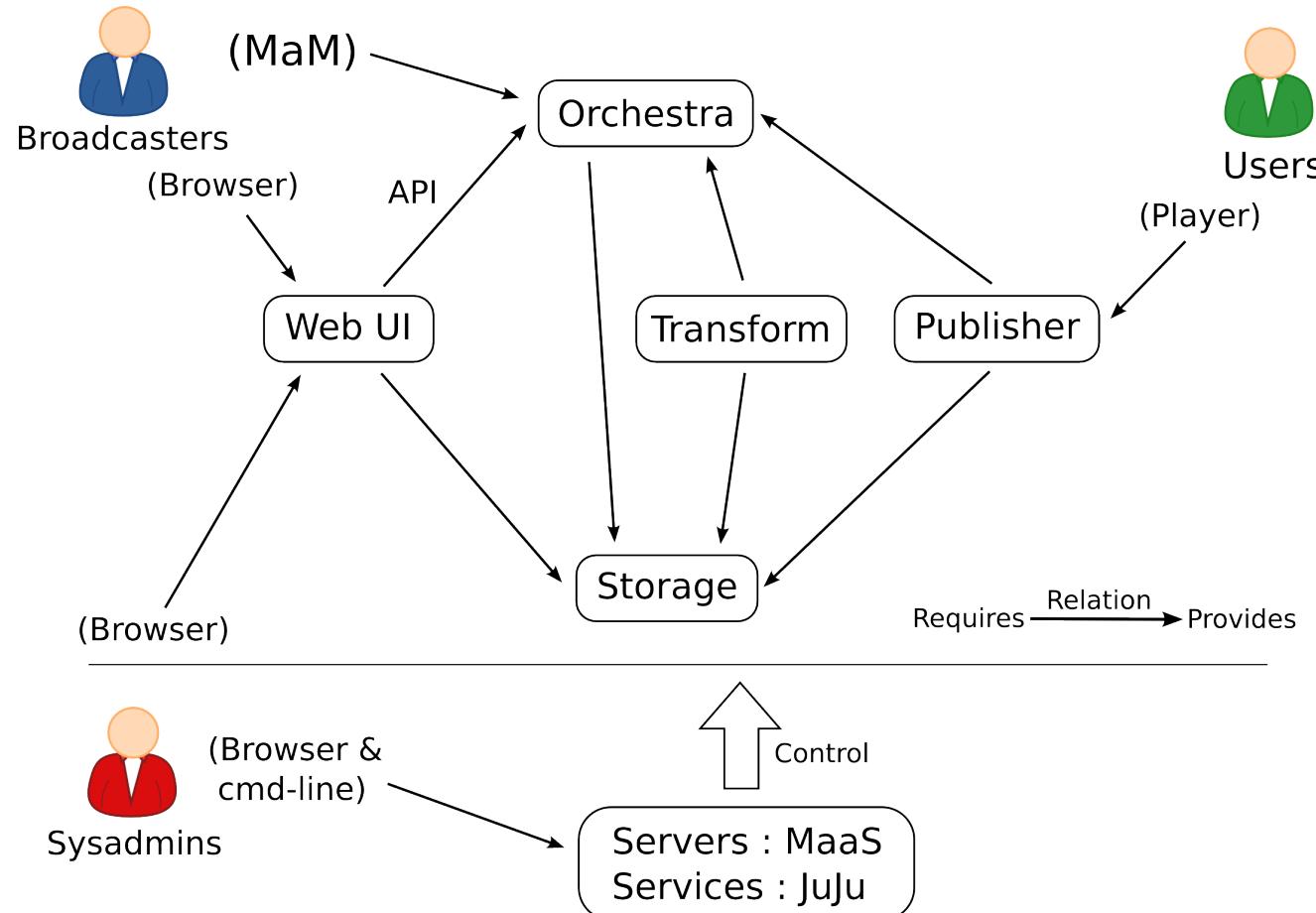
Design Process (2/6)



Design process steps

Separation of services into interconnected components
Selection of an OSS deployment tool (JuJu & MaaS)

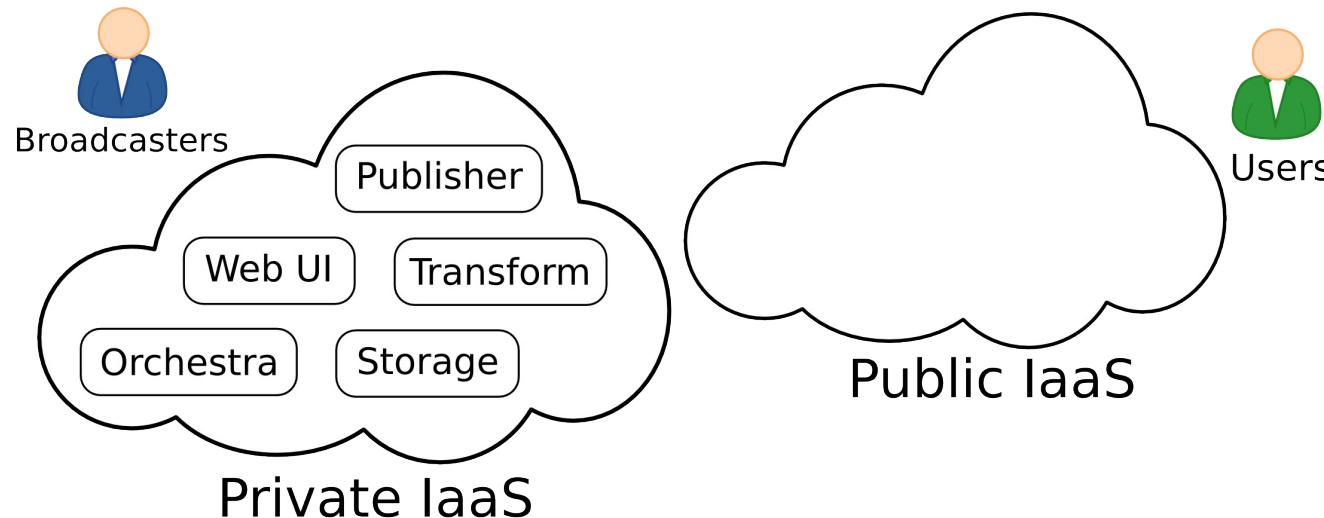
Design Process (3/6)



Design process steps

Encapsulation of services into Charms (+relations)
Selection of OSS sub-components (e.g. Flask)

Design Process (4/6)

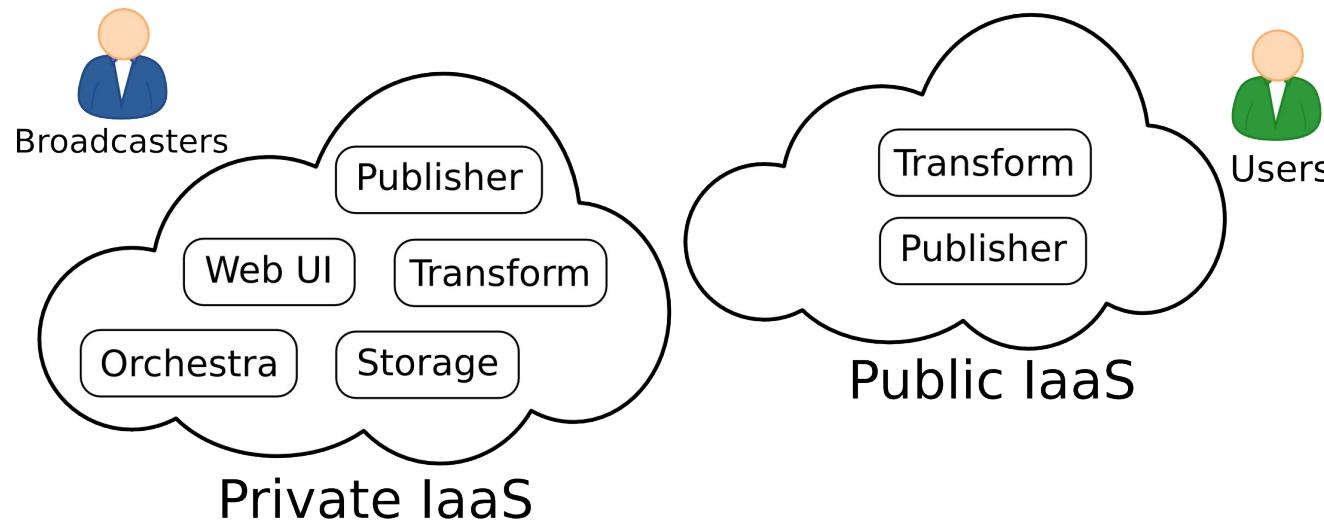


Design process steps

Development of charm's hooks

Deployment of charms for hooks test & debug

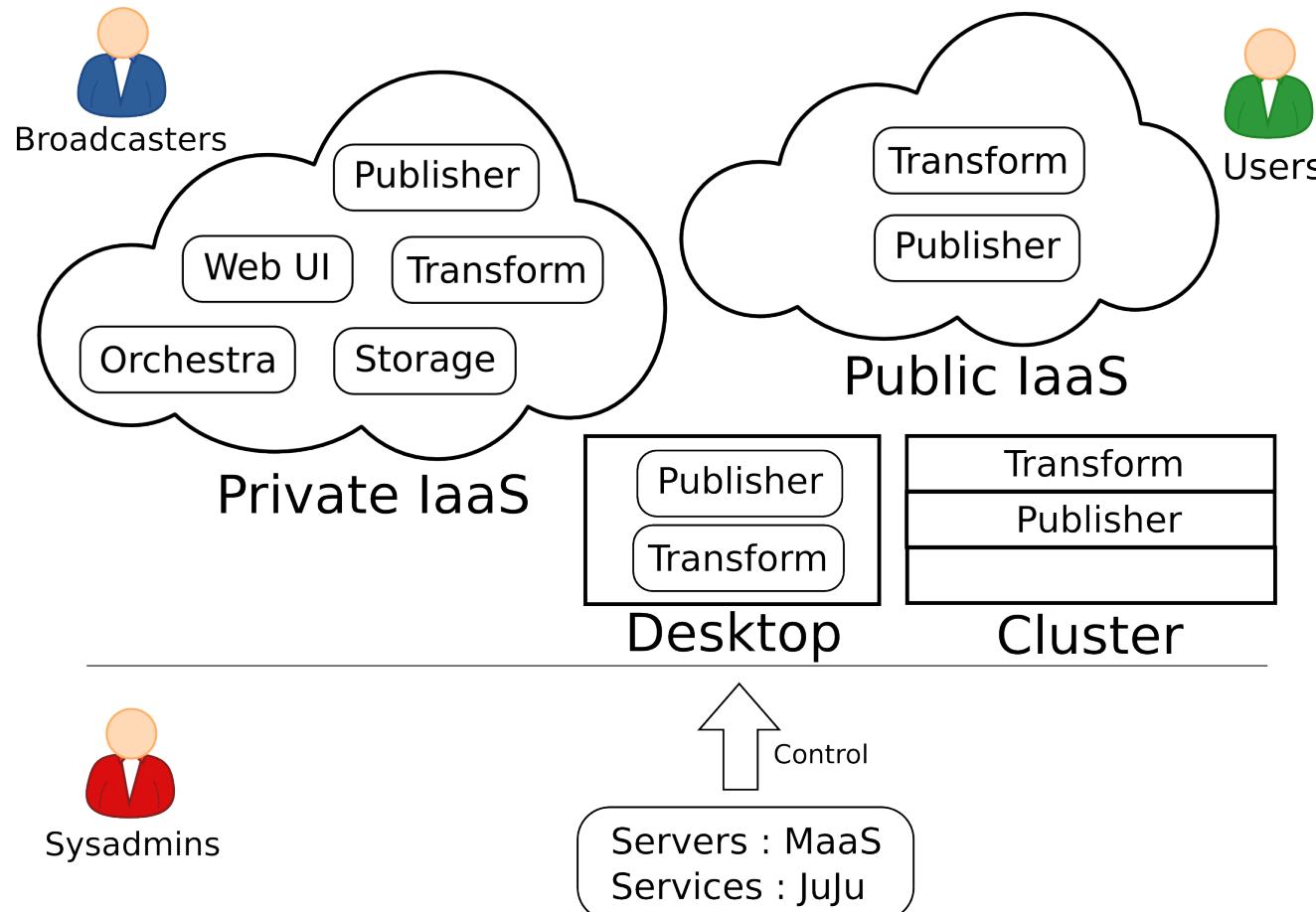
Design Process (5/6)



Design process steps

Adaptation of charms to allow // env. deployment
Development of application-level features

Design Process (6/6)

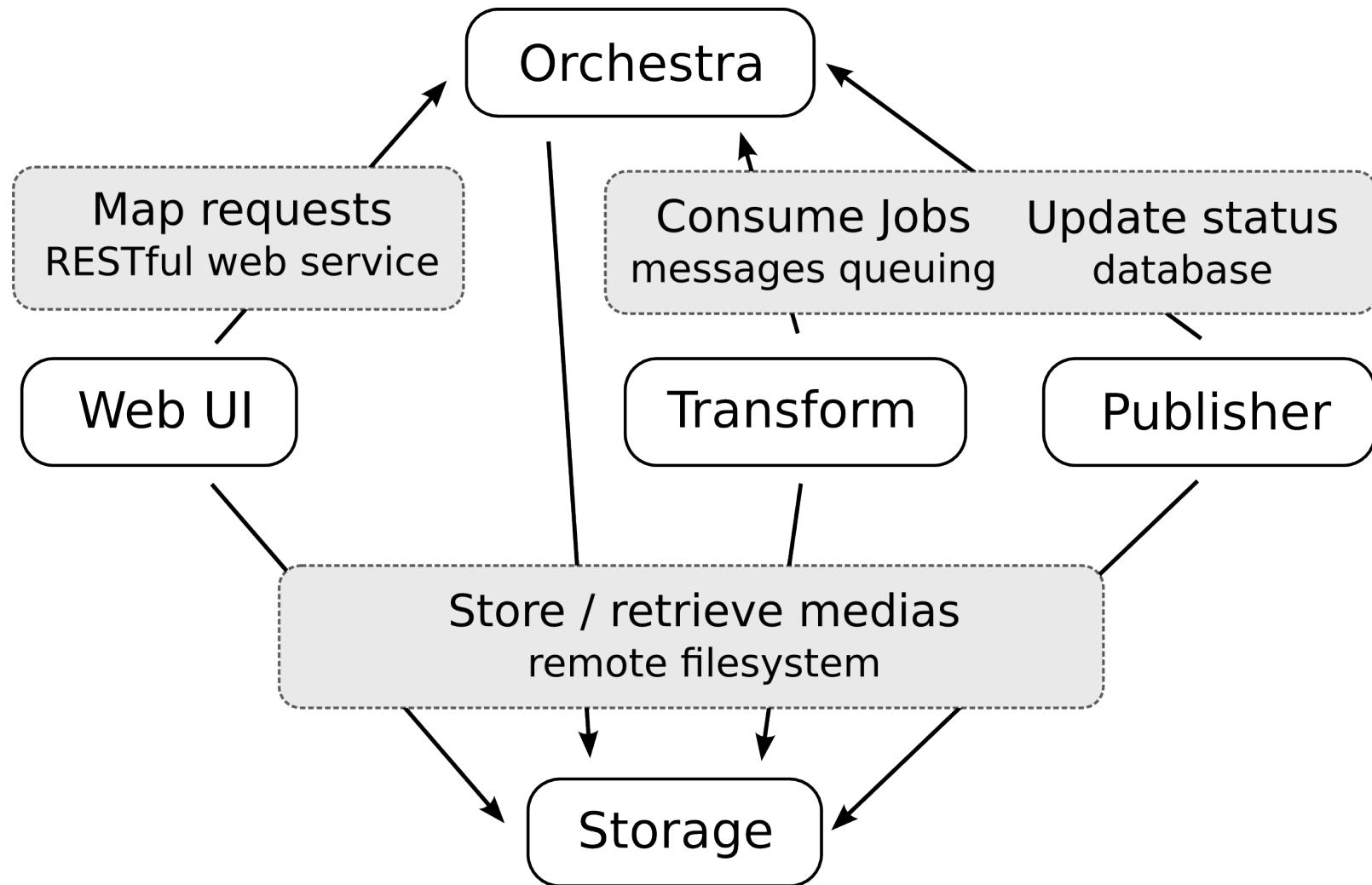


Design process steps

Improvements based on 1st demo EBU's feedback
Multi-environment setup for demo (+debug)

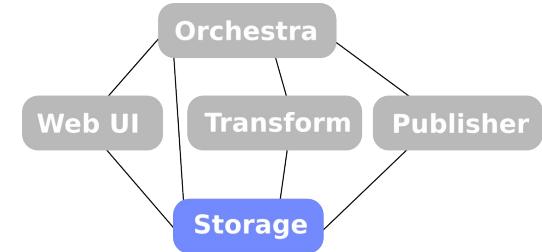
Application Architecture

Relation between components



Application Architecture

Highly-available shared storage



Description	Kind of access OS API	Complex.	Regions	Stability Mgmt.	Scalability Security	Miscellaneous
Swift S3 storage	OS, HTTP S3 API	High	(?)	(?) (?)	High Proxy	Object < 5GB
Ceph (all-in-one)	FS, BS, OS S3 API	Medium	(?)	(?) (?)	High (?)	
LS4 media storage	OS + metadata REST API	High	Yes	Unit test (?)	High (?)	No more updates
NFS filesystem	FS	Low	No	Stable Simple	Low Yes	
GlusterFS	FS, OS, CIFS, HTTP, FTP S3, Swift API	Low	Yes	Stable Simple	High Yes	Most scalable filesystem

OS : Object storage
 BS : Block storage
 FS : Filesystem

Systèmes de fichiers distribués -
 comparaison sur la grille Grid 5000
 JF. Garcia, F. Lévigne, M. ...

Application Architecture

Highly-available shared storage

Charm encapsulating a (simple) GlusterFS server

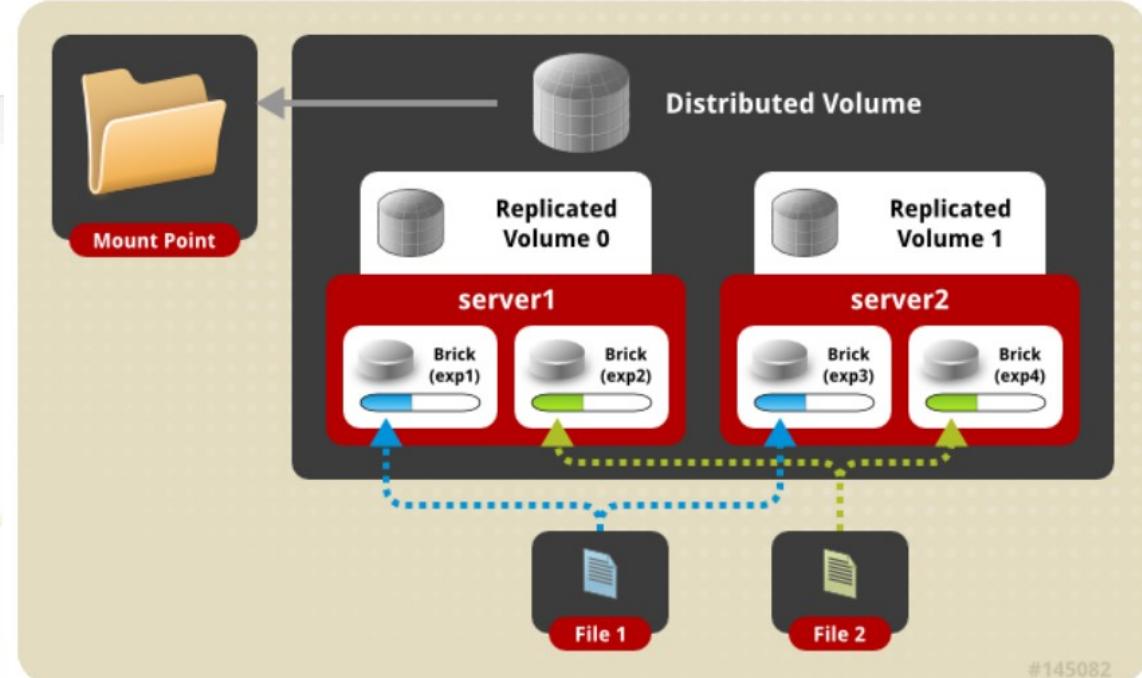
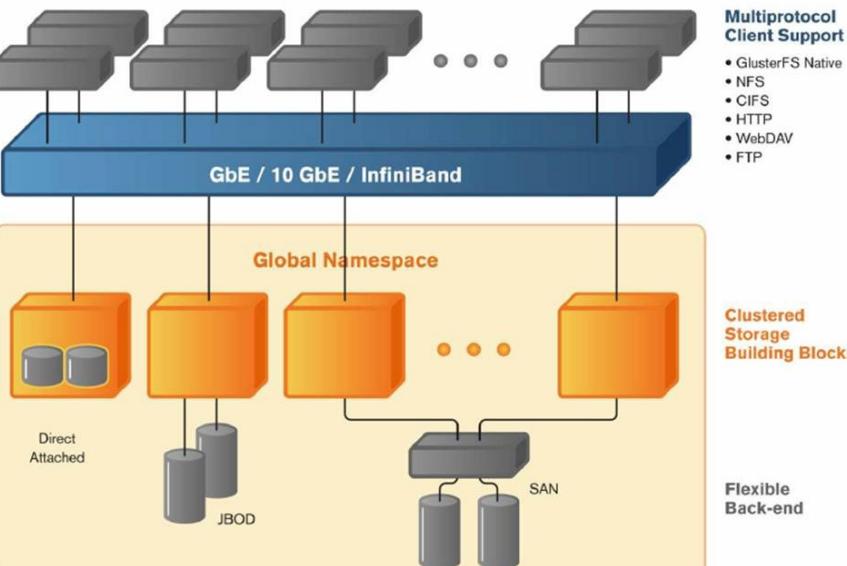
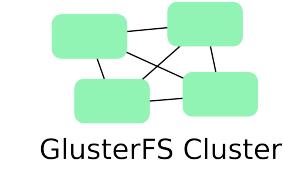
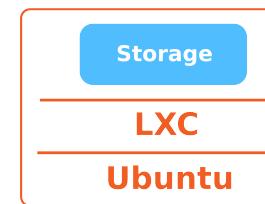
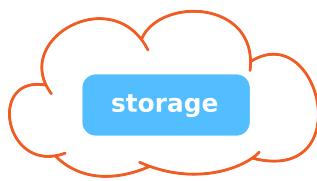


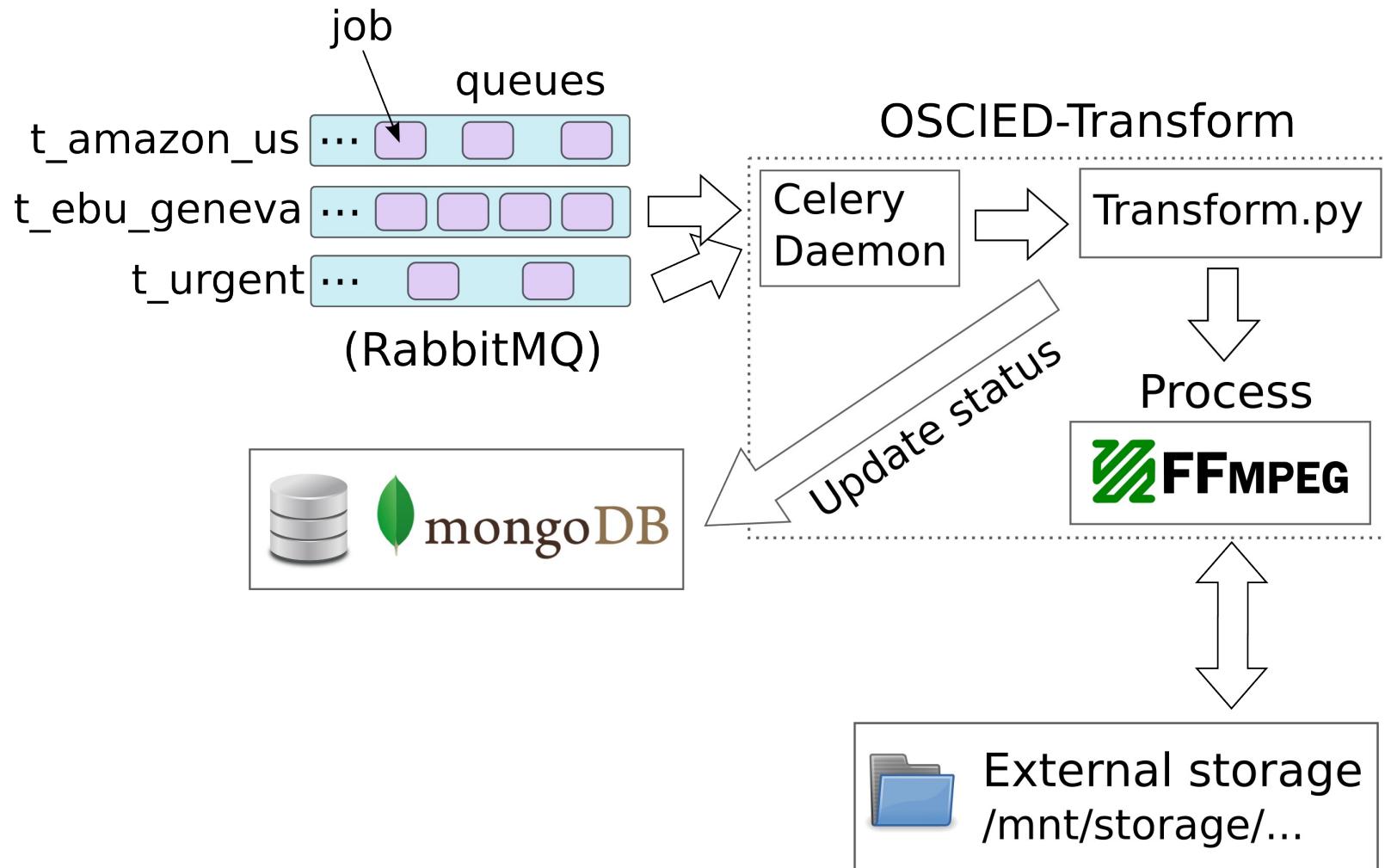
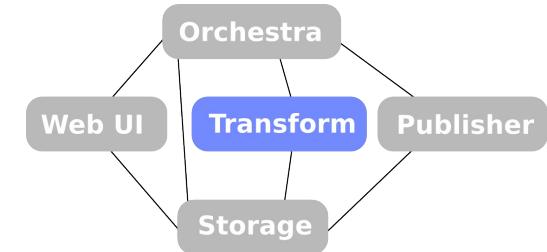
Figure 5.5. Illustration of a Distributed Replicated Volume

Implementations of the Storage Service



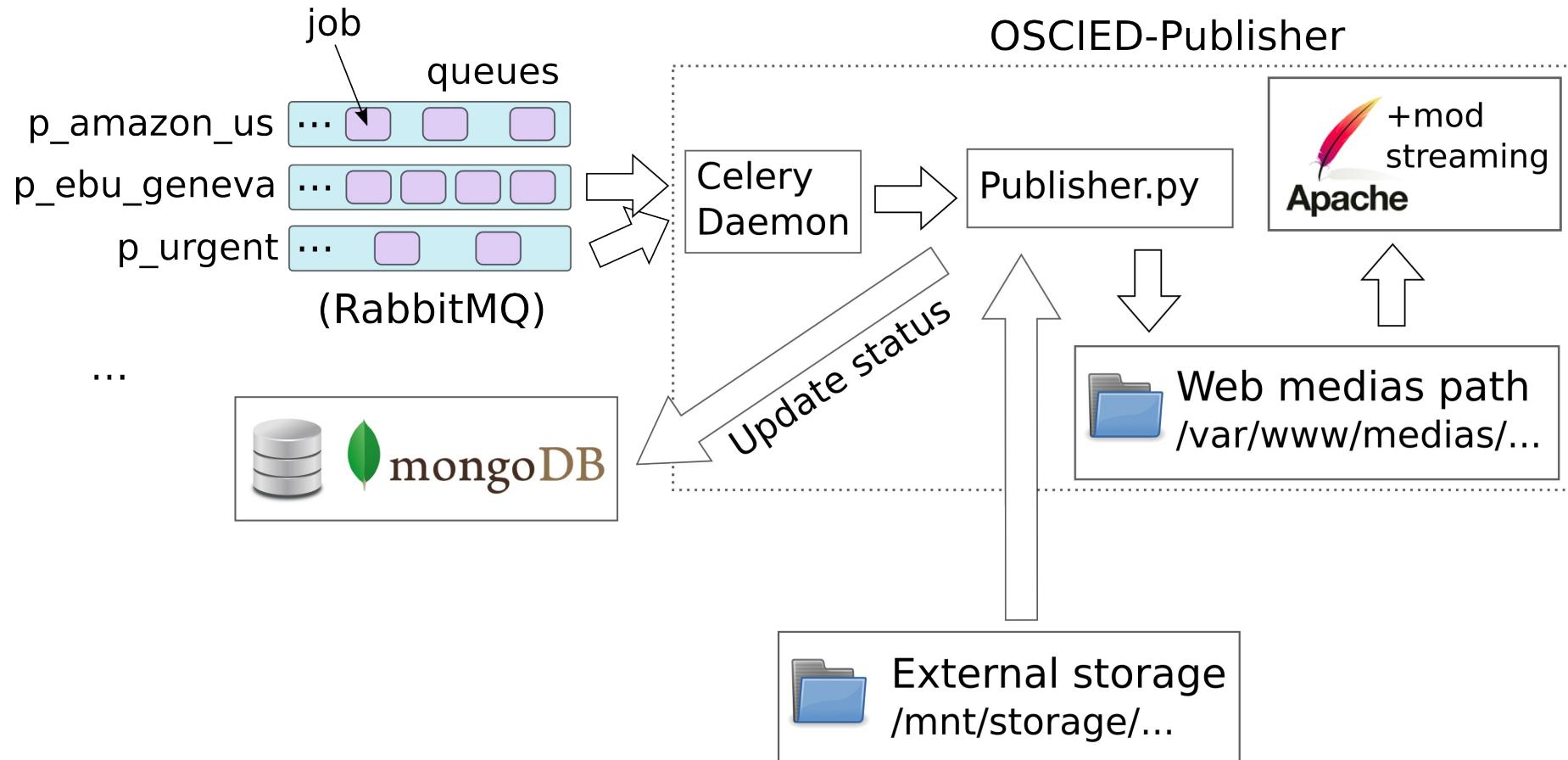
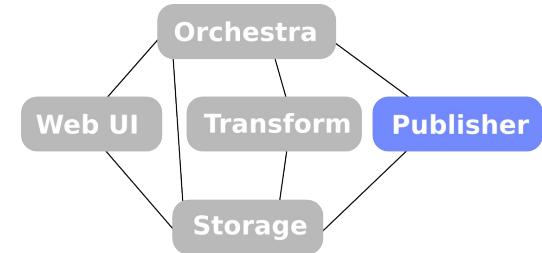
Application Architecture

Media transformation service



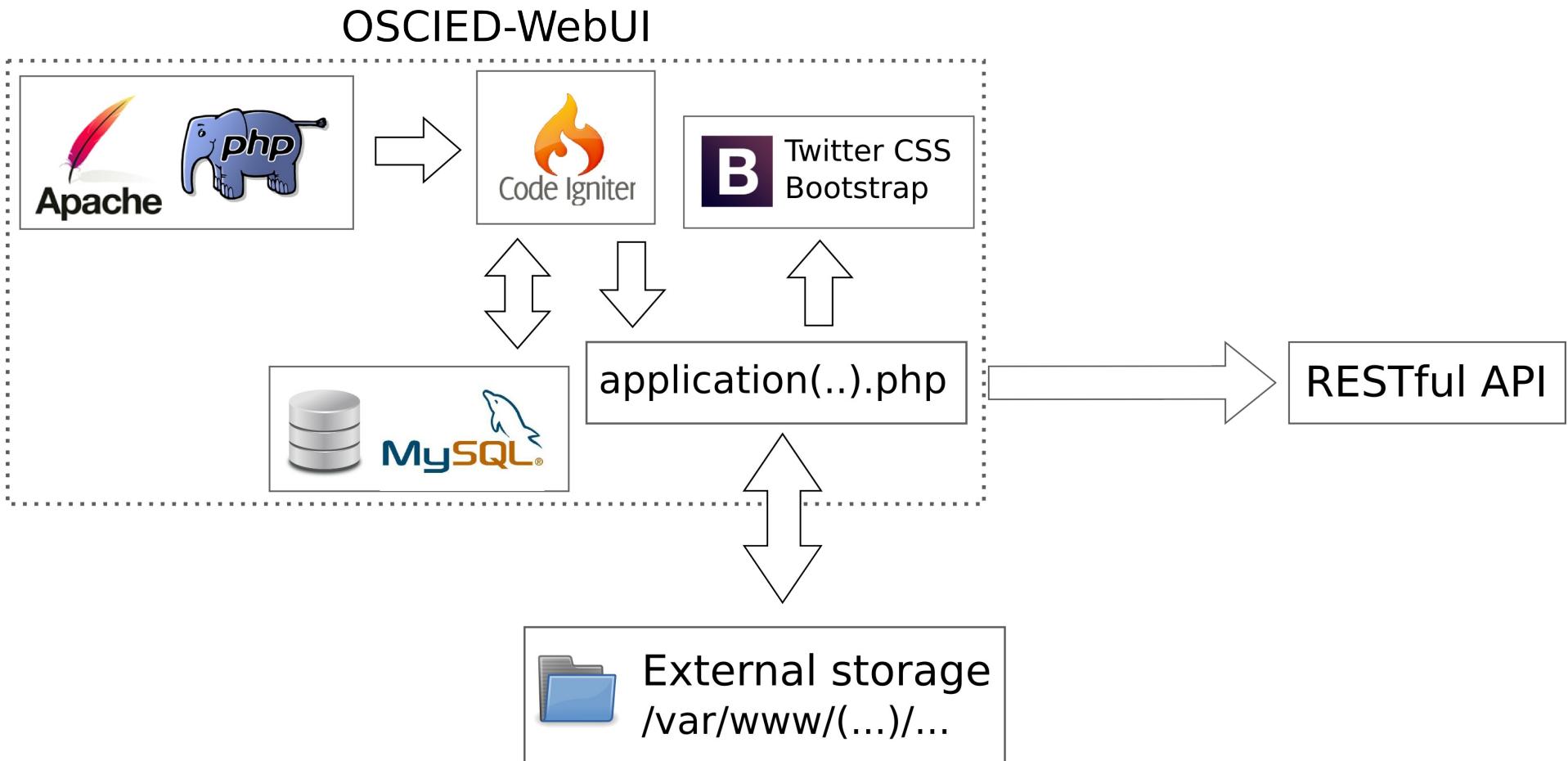
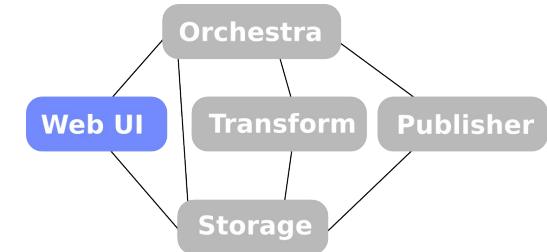
Application Architecture

Media publication service



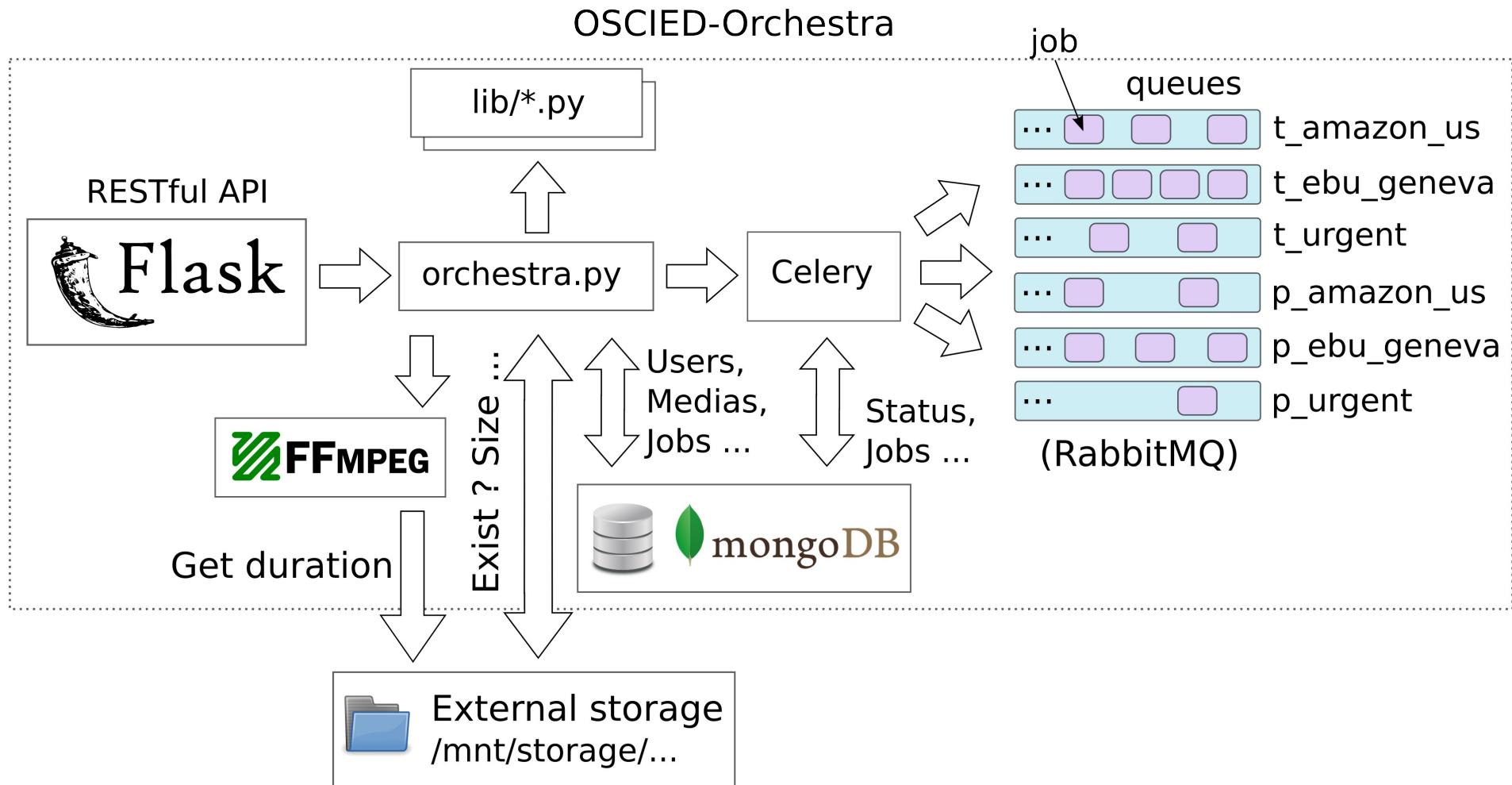
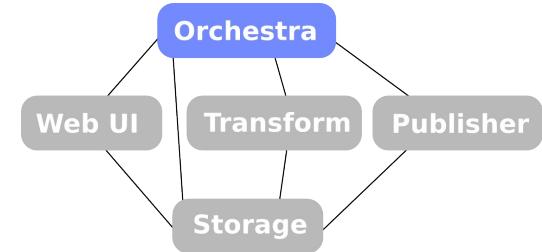
Application Architecture

User-friendly web user interface

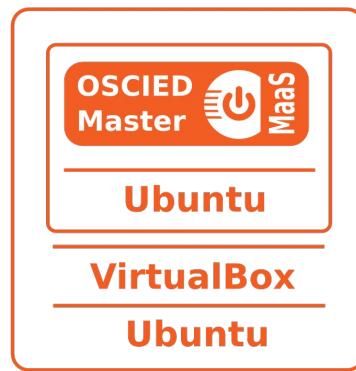
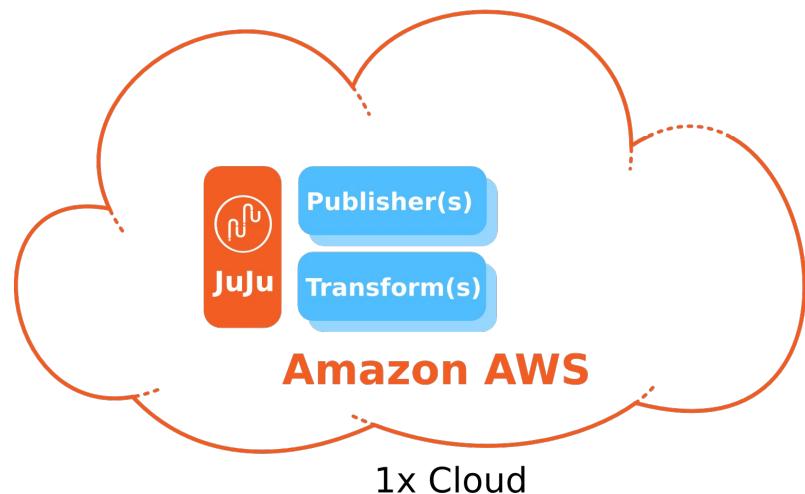


Application Architecture

The brain of OSCIED



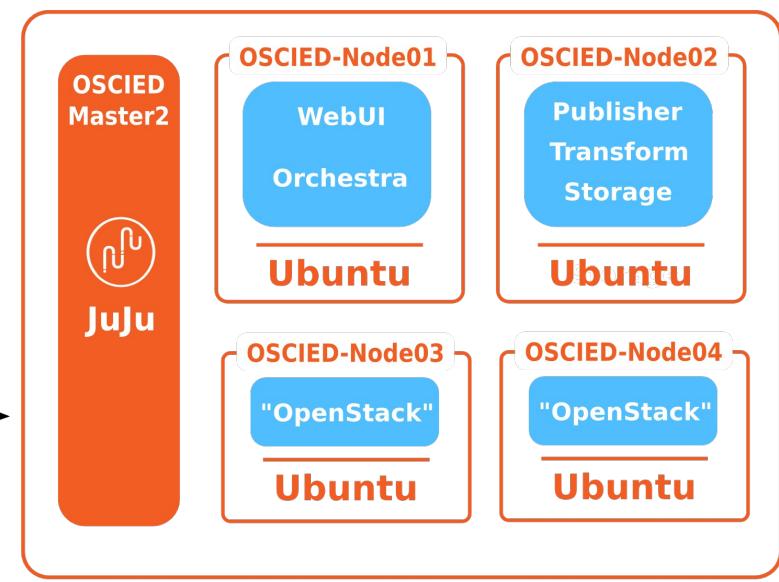
OSCIED D' Day Setup



1x Desktop

Source	Destination	Service
:5000	OSCIED-Node01:5000	Orchestra API
:5672	OSCIED-Node01:5672	Orchestra RabbitMQ
:27017	OSCIED-Node01:27017	Orchestra MongoDB
:7000	OSCIED-Master:80	MaaS Master
:7001	OSCIED-Node01:80	Web UI
:7002	OSCIED-Node02:80	Publisher
:24007	OSCIED-Node02:24007	Storage
:24009	OSCIED-Node02:24009	Storage

NAT/FW



1x Desktop + 4x Dell R420 Servers

Conclusion – Checklist

- [OK] Providing a scalable OSS media platform to members of EBU
- [OK] A platform based on cloud-era OSS technologies, dedicated to :
 - [OK] **Transcoding** of a wide collection of medias to new formats
 - [OK] **Online publication** of popular medias
- [OK] A platform that is designed to allow :
 - [OK] Deployment of services in-house, entreprise IT infrastructure
 - [Bonus] Deployment into Clusters and Standalone Desktop Computers
 - [Bonus] Can be interconnected to haproxy, nginx, Charms Store
 - [OK] Scaling of transcoding and publication points to public clouds (IaaS)
 - [OK] Ease of scale-up or scale-down of services
 - [OK] Separation of development, testing and production environments
 - [OK] Future features or capabilities (MPEG-DASH, ...)
- [OK] Setup of MaaS ([] → OpenStack)
- [OK] Ticket management system (TRAC) + code versioning (SVN)

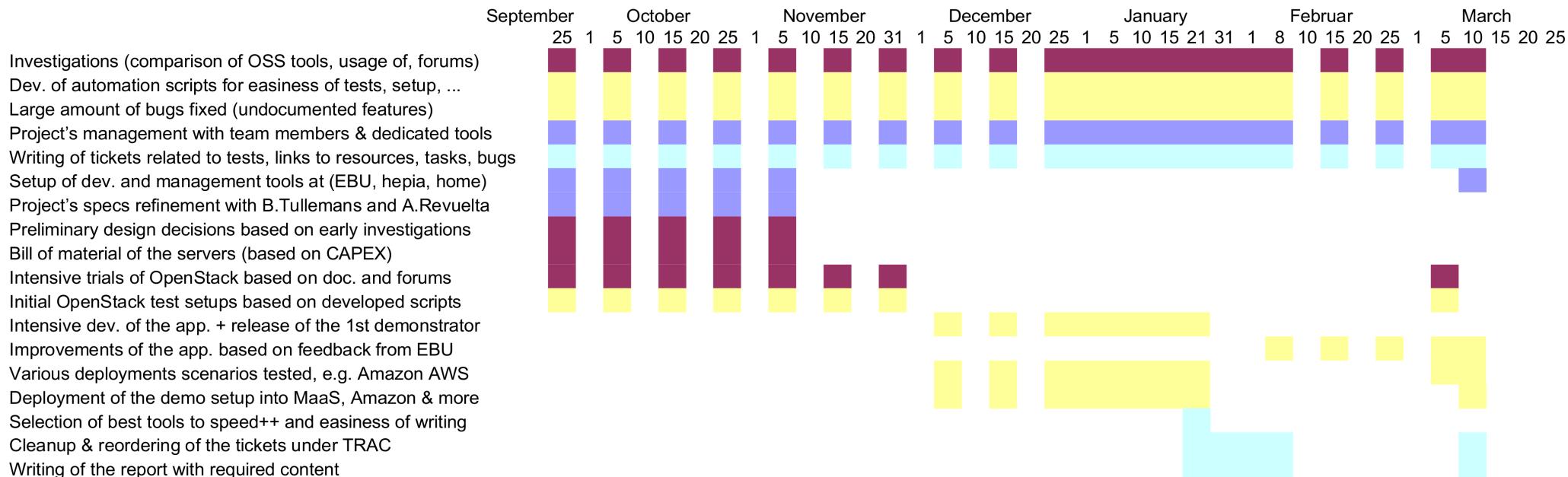
Conclusion – The Project

- Setup OpenStack is a complex, time-taking task. Documentation is not synced with releases, ~breaks compatibility.
- Canonical's MaaS and JuJu allow to deploy, connect and manage servers & services in a simple manner !
- This demonstrator is an OSS media platform allowing EBU's members :
 - To use in-house IT resources mixed with necessary amount of public cloud resources
 - To easily adapt scale of transcoding and publication services
 - To optimize costs of services :
 - By adapting CAPEX / OPEX balance based on business needs
 - By avoiding « vendor lock-in » to dedicated media services (SaaS) or even Cloud providers (IaaS)
 - To test it, use it, even add missing features as it is Open-Source licensed
- Developed application is designed to being simple to use but rather powerful :
 - Deployment is made easy, thanks to JuJu and developed charm's hooks
 - Is elastic, balance of private & public resources is easy and can be automated
 - Proof of concept : Open-Source based Encoding and Distribution Platform

Future

- OSCIED can grow into a full-fledged media platform with professional management layers :
 - Enhancements and future features are already listed on TRAC.
 - Improve security and avoid SPOF
 - MPEG-DASH is the hottest feature
 - Open Broadcast Encoder for delivering MPEG2-TS feeds
 - Add of automatic scalability functions (e.g. based on business rules)
 - Adding native imports of distribution files to reduce latency
- OSCIED was and will be an opportunity for me to work on topics such as OSS, cloud, medias, cluster storage ... :)

Master HES-SO MA - Planning

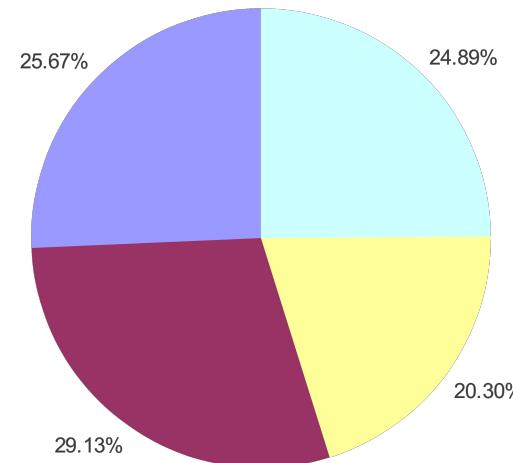


Project Management
 Investigations
 Developments & Tests
 Documentation

Master HES-SO MA - Skills

- **Project Management**
 - Meetings, mails
 - Svn, TRAC tickets ...
 - Goals refinement
- **Investigations**
 - Selection of OSS tools
 - Design of OSCIED
 - Forums & documentation
 - Source-code decipher
- **Developments & Tests**
 - Scripting (automation)
 - Setup of demo & tools
 - Development (application)
 - Charms hooks, ...
 - Application-level logic
 - Tests (deployment, debug)
- **Documentation**
 - Report (PDF + HTML)
 - Presentation
 - TRAC tickets
 - Orchestra API

TIC Profile



Questions & Answers

