Cyclone

Complete Dynamic Multi-cloud Application Management

CYCLONE Architecture and Services for Cloud Applications Developers and Providers

CYCLONE Whitepaper
December 2017

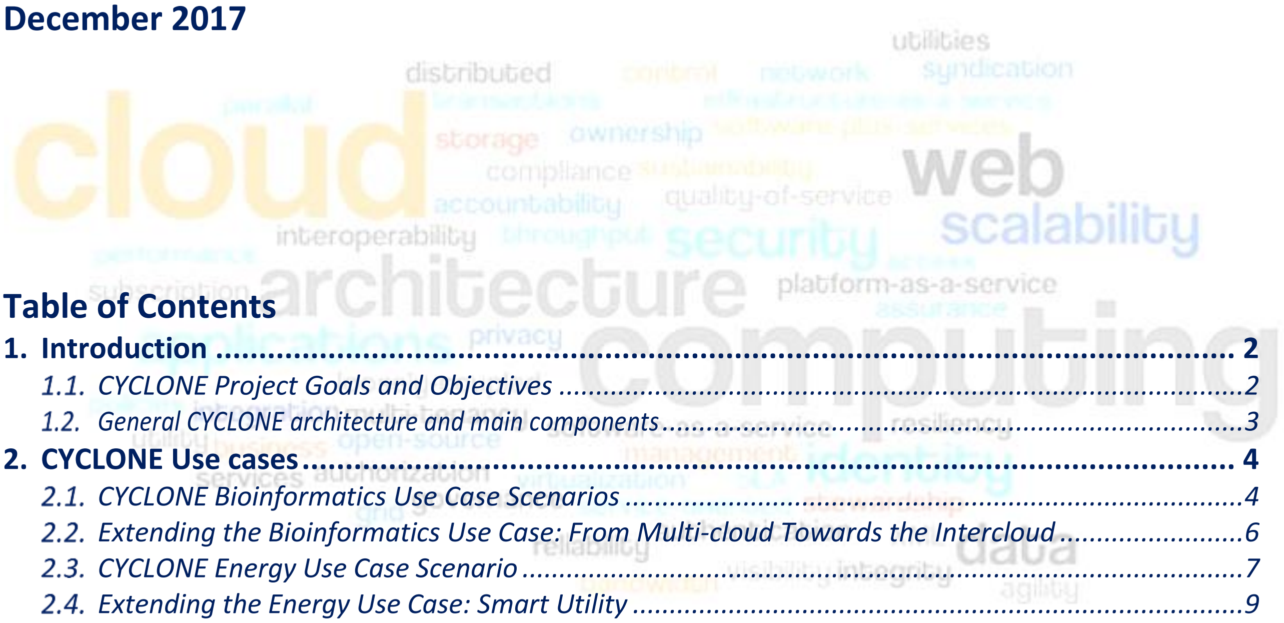


Table of Contents

1. Introduction

The CYCLONE project has well defined target communities that have demonstrated strong interest and are already using the project products for automated cloud services deployment and management that would allow both applications developers and services operators to effectively use cloud based resources by simplifying development, deployment and test of services and applications. The developed tools are extending their functionalities to integrate services from multiple providers and providing user friendly web-based interface for non-computer savvy users.

1.1. CYCLONE Project Goals and Objectives

CYCLONE is a Horizon 2020 innovation action funded by the European Commission which aims at integrating existing cloud management software to allow a unified management of federated clouds. Application service providers (ASPs) develop, deploy, and maintain complex computing platforms within multiple cloud infrastructures to improve resilience, responsiveness and elasticity of their applications.

The CYCLONE project targets the ASPs, providing them with software and tools that

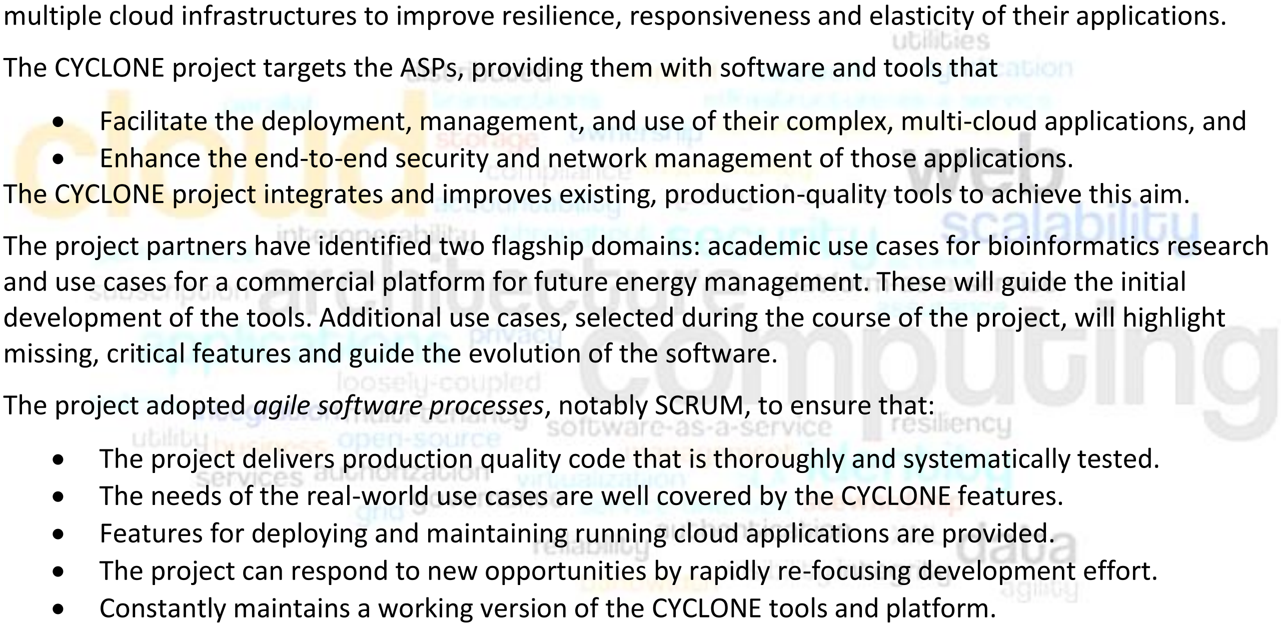
- Facilitate the deployment, management, and use of their complex, multi-cloud applications, and
- Enhance the end-to-end security and network management of those applications.

The CYCLONE project integrates and improves existing, production-quality tools to achieve this aim.

The project partners have identified two flagship domains: academic use cases for bioinformatics research and use cases for a commercial platform for future energy management. These will guide the initial development of the tools. Additional use cases, selected during the course of the project, will highlight missing, critical features and guide the evolution of the software.

The project adopted *agile software processes*, notably SCRUM, to ensure that:

- The project delivers production quality code that is thoroughly and systematically tested.
- The needs of the real-world use cases are well covered by the CYCLONE features.
- Features for deploying and maintaining running cloud applications are provided.
- The project can respond to new opportunities by rapidly re-focusing development effort.
- Constantly maintains a working version of the CYCLONE tools and platform.



The CYCLONE distributed testbed will play a critical role, enabling the full, continuous validation of the software before the foreseen production deployments on a federated bioinformatics cloud.

The main ***objectives*** of the CYCLONE project are:

- *Improve* cloud services in the Infrastructure-as-a- Service (IaaS) layer by integrating network services into the cloud offering, allowing direct control over virtual machine (VM) network accessibility, intra- site data access, and inter-site data transfers.
- *Develop* tools that provide enhanced functionality for cloud providers that agree to federate their resources, such as dynamic allocation of bandwidth between cloud providers and common authentication mechanisms.
- *Provide* tools that allow application developers to take advantage of features like VM coordination within deployments, automated placement of service components, and scaling of service components, essentially providing them with the means to develop a Platform-as-a-Service (PaaS) offering.
- *Provide* software that allows developers to ensure the end-to-end, secure use of data within their application as well as secured access to remote data sources.

- *Demonstrate* that the CYCLONE software meets the needs of concrete academic and commercial use cases, while providing frequent, production-quality releases of that software

1.2. General CYCLONE architecture and main components

Multiple individual use cases and usage scenarios for multi-cloud applications that require cloud and non-cloud resources integration into one intercloud infrastructure that executes a single or multiple enterprise or scientific workflows can be abstracted into general scenario and use case illustrated in Figure 1. It includes two interacting applications, that in general can be multi-cloud, that contain both application related and management components. Application component interacts with end users, management component is controlled by application administrator and interacts with the (inter)cloud management software. The figure also shows Cloud Applications Deployment and Management Software and Tools as an important component to support cloud applications deployment and operation during their whole lifecycle. Intercloud infrastructure should also provide two other components or services: federated access control and security, and intercloud network infrastructure that needs to be provisioned as a part of overall application infrastructure.

Intercloud applications and infrastructure may include multiple existing cloud platforms. In the generally distributed heterogeneous multi-cloud multi-provider environment the problem of applications integration and management is becoming critical and require smooth integration with the application workflow and automation of most of development and operation functions, ideally integration with the application specific development and operation (DevOps) tools. Currently widely used cloud automation tools such as Chef, Puppet allow single cloud provider application deployment. They don't solve problem of multi-cloud resources/services integration and provisioning of inter-cloud network infrastructure.

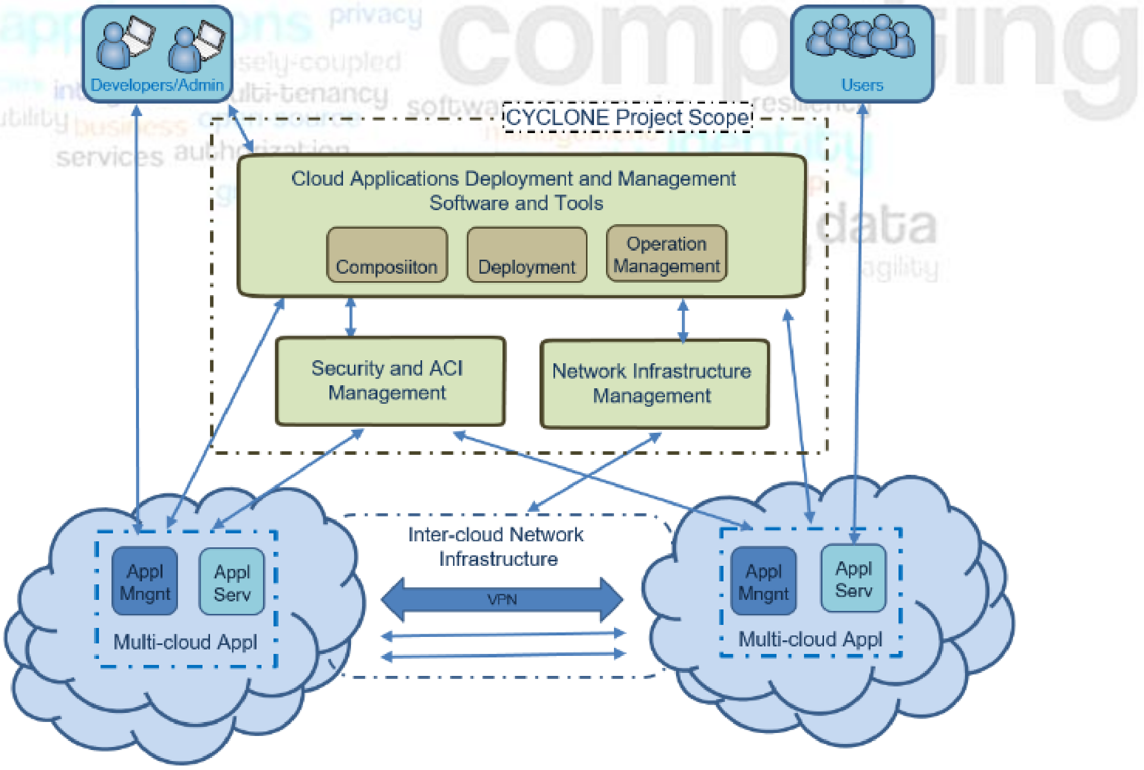

Developers/Admin
Users
CYCLONE Project Scope
Cloud Applications Deployment and Management Software and Tools
Composiiton
Deployment
Operation Management
Security and ACI Management
Network Infrastructure Management
Inter-cloud Network Infrastructure
VPN
Appl Mngnt
Appl Serv
Multi-cloud Appl
Appl Mngnt
Appl Serv
Multi-cloud Appl

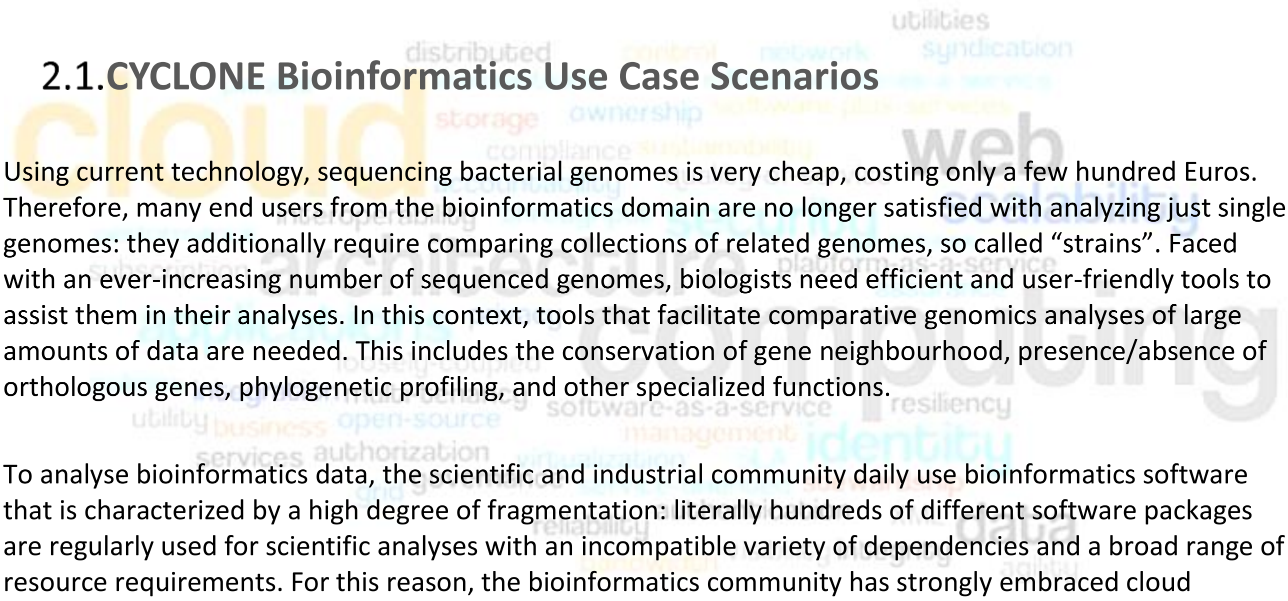
Figure 1. General use case for multi-cloud applications deployment.

The CYCLONE project addressed this problem by leveraging original cloud management platform SlipStream and extending its with necessary functionality and components, in particular for inter-cloud resources deployment and network infrastructure provisioning, enabling federated access control for users and end-to-end security for data transfer, enabling dynamic trust establishment between cloud and application domains.

Intercloud platforms should deliver open integration environment and preferably standardized APIs, protocols, and data formats, allowing for cross-cloud resources interoperability. Practical Intercloud platform development should target two major stakeholders and user communities the Application Service Providers (ASPs) as well as their customers to address real life challenges and problems in a consistent and constructive way.
The effective cloud automation and management platform should allow dynamic cloud resources allocations depending on the workload and application workflow. This task can be solved for single cloud using its native elasticity and load balancing tools, however in intercloud environment such functionality will require involving real cloud platform load (including resources availability) and application monitoring.

2. CYCLONE Use cases

2.1. CYCLONE Bioinformatics Use Case Scenarios



Using current technology, sequencing bacterial genomes is very cheap, costing only a few hundred Euros. Therefore, many end users from the bioinformatics domain are no longer satisfied with analyzing just single genomes: they additionally require comparing collections of related genomes, so called “strains”. Faced with an ever-increasing number of sequenced genomes, biologists need efficient and user-friendly tools to assist them in their analyses. In this context, tools that facilitate comparative genomics analyses of large amounts of data are needed. This includes the conservation of gene neighbourhood, presence/absence of orthologous genes, phylogenetic profiling, and other specialized functions.

To analyse bioinformatics data, the scientific and industrial community daily use bioinformatics software that is characterized by a high degree of fragmentation: literally hundreds of different software packages are regularly used for scientific analyses with an incompatible variety of dependencies and a broad range of resource requirements. For this reason, the bioinformatics community has strongly embraced cloud computing with its ability to provide customized execution environments and dynamic resource allocation.

The French Institute of Bioinformatics – IFB id a project partner and actively using cloudin their research. The CYCLONE consortium has identified several concrete bioinformatics use cases that aim to address some specific well-identified limitations. For example, regarding the key technical areas of the CYCLONE project, Cloud Access Management through cloud proxies and Matchmaking, Brokering, and Mediation of Cloud Resources will provide the IFB with features to access other cloud infrastructures than the current national IFB’s, to integrate the future cloud infrastructures that will be deployed by the regional IFB’s platform, and also to access external cloud infrastructures from academic or commercial providers.

2.1.1. Use case scenario – Securing Human Biomedical Data

This use case is a single virtual machine (VM) application requiring enhanced security features such as a trusted federated authentication mode and a deployment done only on certified (by the French Health Ministry) cloud infrastructure. The use case workflow to ensure data security is illustrated in Figure 2. The cloud appliance NGS- Unicancer is developed by the bioinformatics platform, http://www.synergielyoncancer.fr/ of the Centre Léon Bérard (Lyon, France) in the context of the project

NGS-Clinique (INCA - Institut National du Cancer). It provides a simple web interface to launch the biomedical genomic analysis pipeline. The appliance was enhanced by the Federation Provider developed by Technische Universität Berlin and is ready for on-demand deployment on the IFB's core cloud infrastructure. The user deploys the appliance NGS-Unicancer through the IFB's web interface in "1-click" and uses the CYCLONE federation provider to get access to the VM web interface based on its identity in the federation. The user then easily uploads relevant data, runs the analysis and gets the results.

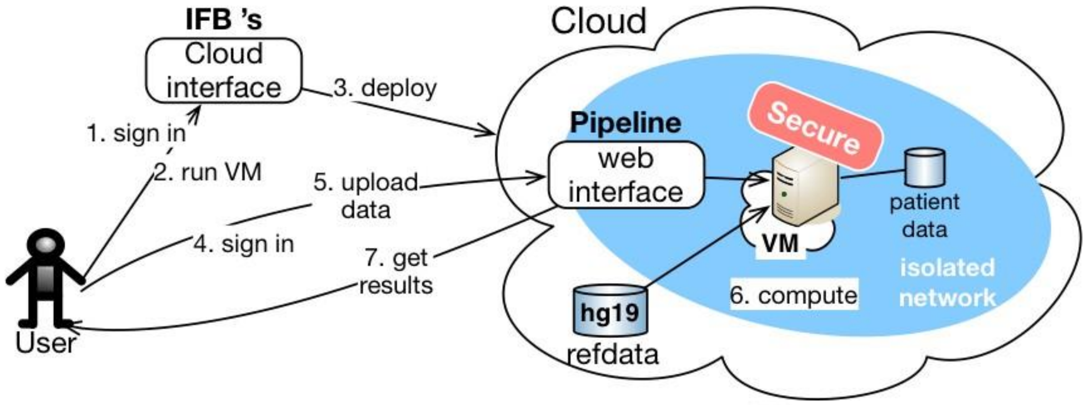


Figure 2. Functional schema of the use case "Securing human biomedical data".

2.1.2. Use case scenario - Cloud Virtual Pipeline for Microbial Genomes Analysis

This use case (which workflow is illustrated in Figure 3) is developed by the platform IFB-MIGALE, http://migale.jouy.inra.fr/ (Jouy-en-Josas, France). The application requires several components: a user web interface, a relational postgreSQL database, and a complete computing cluster with a master and several nodes to perform the data-intensive analyses. This infrastructure already running in a classical static way on bare-metal servers in IFB-MIGALE premises was ported to the cloud and extended with a "1-click" deployment feature by using SlipStream recipes. The image was exported from the IFB's cloud and registered in the StratusLab Marketplace. Afterwards, IFB-core wrote a deployment recipe based on SlipStream that instantiates the complete application with all the required VMs on the CYCLONE infrastructure.

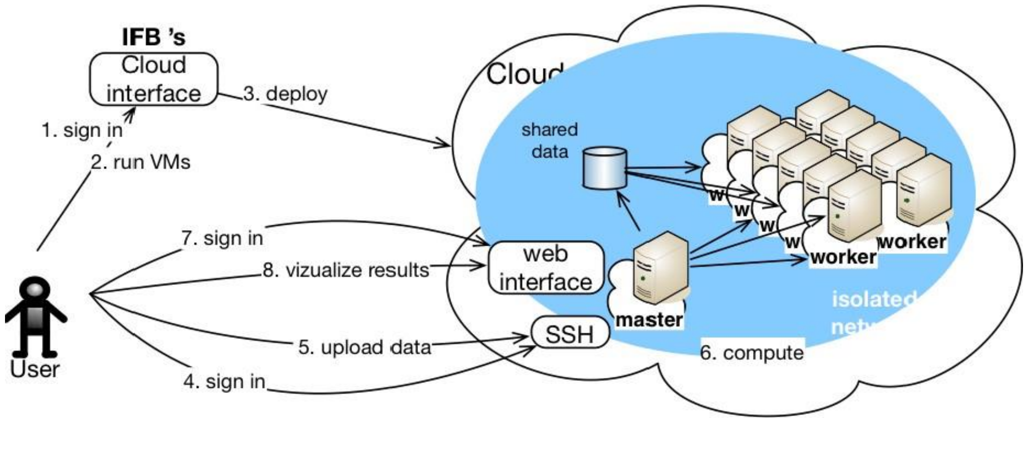


Figure 3 Functional schema of the use case "Cloud virtual pipeline for microbial genomes analysis".

2.2. Extending the Bioinformatics Use Case: From Multi-cloud Towards the Intercloud

Bioinformatics deals with the collection and efficient analysis of biological data, particularly genomic information from DNA sequencers, which become increasingly distributed and may be hosted in different private and public or scientific clouds. The terabytes of raw data, produced by the sequencers for each run, require significant computing resources for analysis that may not be available locally. These sequencers are located at a dozens of places in France, while the users are distributed throughout the country and possibly further afield via international collaborations. Some sequencing centers adopt cloud platform for storing data, large public CPS's and Research Infrastructure (RI) provide cloud based storage of genome data supporting also federated access control with the industry recognised Identify Providers.

Figure 4 shows the bioinformatics application deployed in Cloud 1 in a form of Virtual Private Cloud (VPC) that includes both the actual application that manage the whole scientific workflow and computing cluster. The bioinformatics engineer develops and deploys and application in Cloud 1 using development tools coupled or integrated with an application deployment manager, e.g., SlipStream. The application may use external scientific data and applications located in SciCloud A and B. In case of an excessive workload, some computational tasks can be outsourced to external cloud CloudExt, in particular in a standard cloudburst scenario. Similarly to the original use case definition, Figure 4 includes a scientific data archive for storing obtained results. The data visualisation and collaboration tools could also be hosted in the cloud and provided by specialised SaaS or cloud application providers.

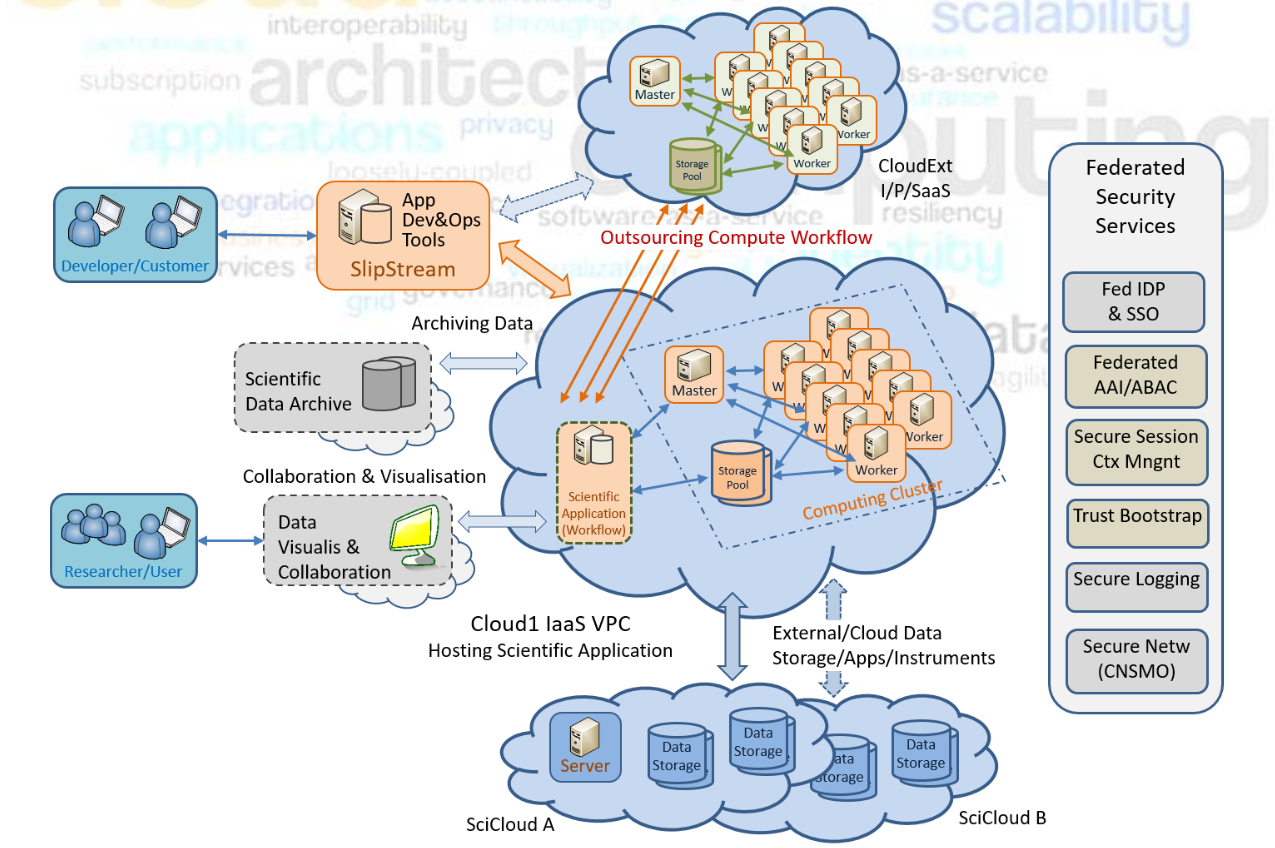


Figure 4: Extended Bioinformatics Use Case

2.3.CYCLONE Energy Use Case Scenario

In order to comply with the “20/20/20” climate change mitigation goals of the EU (http://ec.europa.eu/clima/policies/strategies/2020/index_en.htm), the energy supply has to be changed from conventional fossil energy generation to the increasing usage of renewable energy sources. The integration of CO2 free, volatile and decentralized, renewable energy resources, into the energy system leads to a new approach for energy management systems. To efficiently incorporate the huge number of future participants to the energy system, the integration with new cloud-based ICT technologies is essential.

In the actual situation, conventional power plants are adjustable within a more or less fixed range, so that the energy management is done in a centralised top down approach. To include more and more renewable energy sources in the overall energy system leads to the necessity to restructure the energy supply system. It has to be transformed from a centralized system with large power plants to an increasingly decentralized system of power generation. The integration of CO2 free, volatile, and decentralized, renewable energy resources into the energy system changes its characteristics and forces the underlying power grid to become smarter. This transformed power grids are therefore referred to as smart grids nowadays. Moreover, the integration raises the overall need for an energy management system. Coming from a centralized system where energy is produced when its needed, it changes to a system where energy is produced when sun or wind are available.

Integrating the latest Information and Communication Technology (ICT) for managing the energy components in the grid is becoming more and more essential. ICT from distributed embedded control to Big Data and Cloud Computing is essential for the transition. The decentralized energy production with Decentralized Energy Resources (DER) leads to the necessity to collect measurement data of energy production and consumption in real time all over the grid. Given the sensitivity of the energy system the ICT platform has to fulfil high standards in security, availability and scalability. All these requirements for the data management platform lead to a distributed ICT environment with a trustable and heterogeneous multi-cloud base.

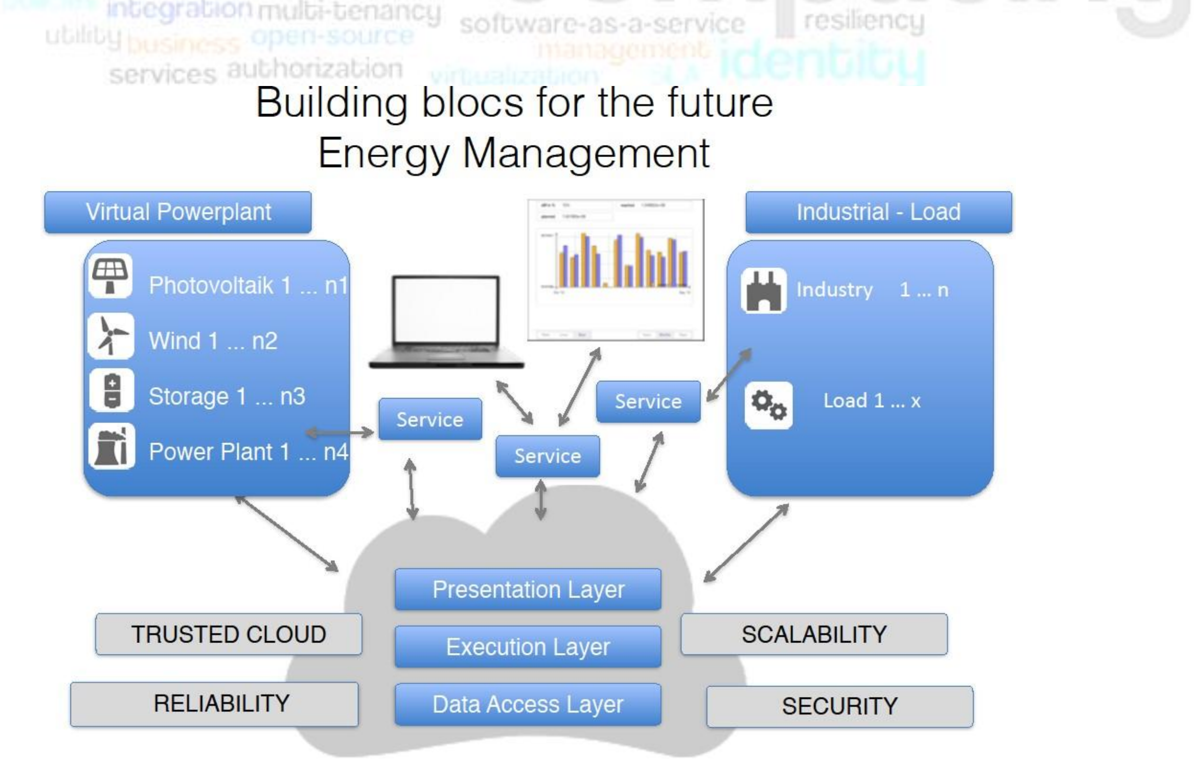


Figure 5. CYCLONE Energy Use Case.

QSC as partner in the CYCLONE consortium has identified several energy use cases which address requirements of an energy management system based on ICT. For example, regarding the critical role of a secure energy supply system, Cloud and application access management is addressed. Transparent multi-cloud resource provisioning enables and facilitates a system architecture that provides the needed resilience, scalability and elasticity for the energy management system.

2.3.1.Use case scenario – Virtual Power Plant

This use case coming from the energy sector (see Figure 5) has the focus on the geographically widely distributed, volatile energy production combined to a Virtual Power Plant (VPP). The main idea of a Virtual Power Plant is to integrate the small, distributed energy resources for the generation of renewable energy and to combine them to one reliable power plant to bring the produced energy into the existing power system. The decentralized energy production leads to the necessity to collect and aggregate measurement data of energy production in real time all over the grid. Concerning the different energy components coming with their own smart metering technologies, the data management has to deal with heterogeneous data.

To build the Virtual Power Plant based on the ICT platform the service for distributed data collection gathers the energy generation data from the different DERs and stores them as raw data on the platform.

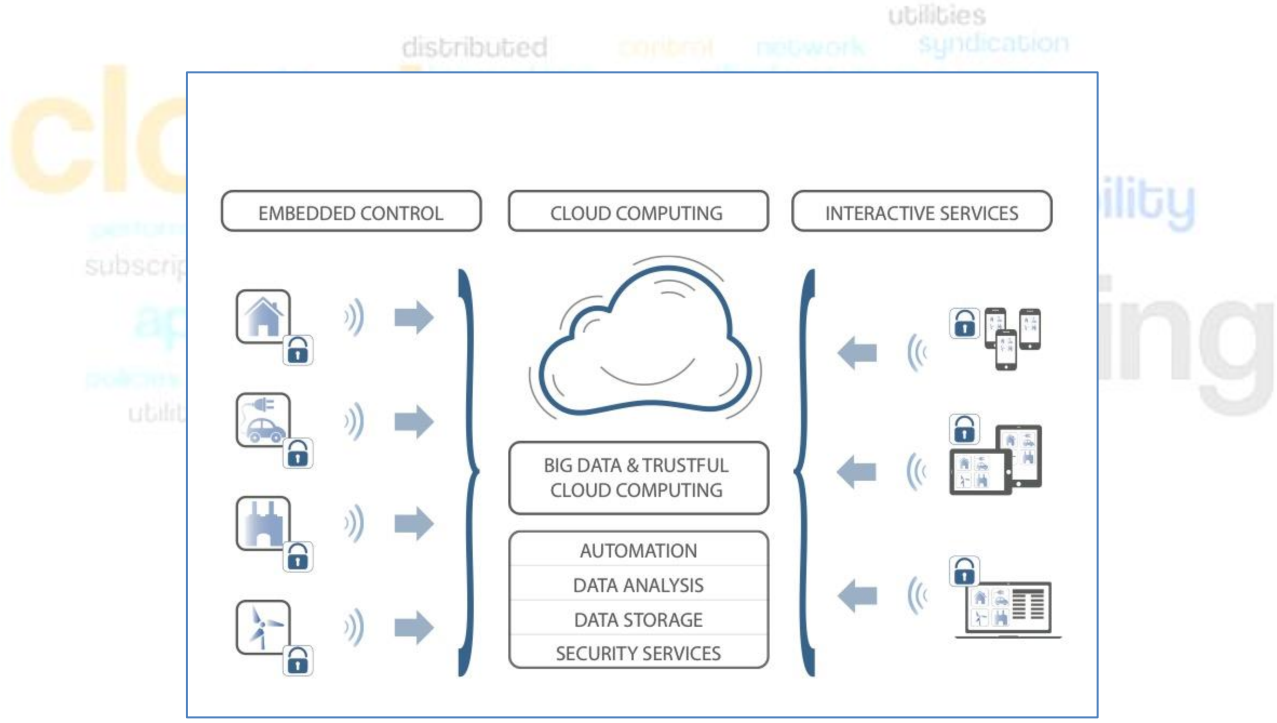


Figure 6. Combining renewable energy resources in a VPP.

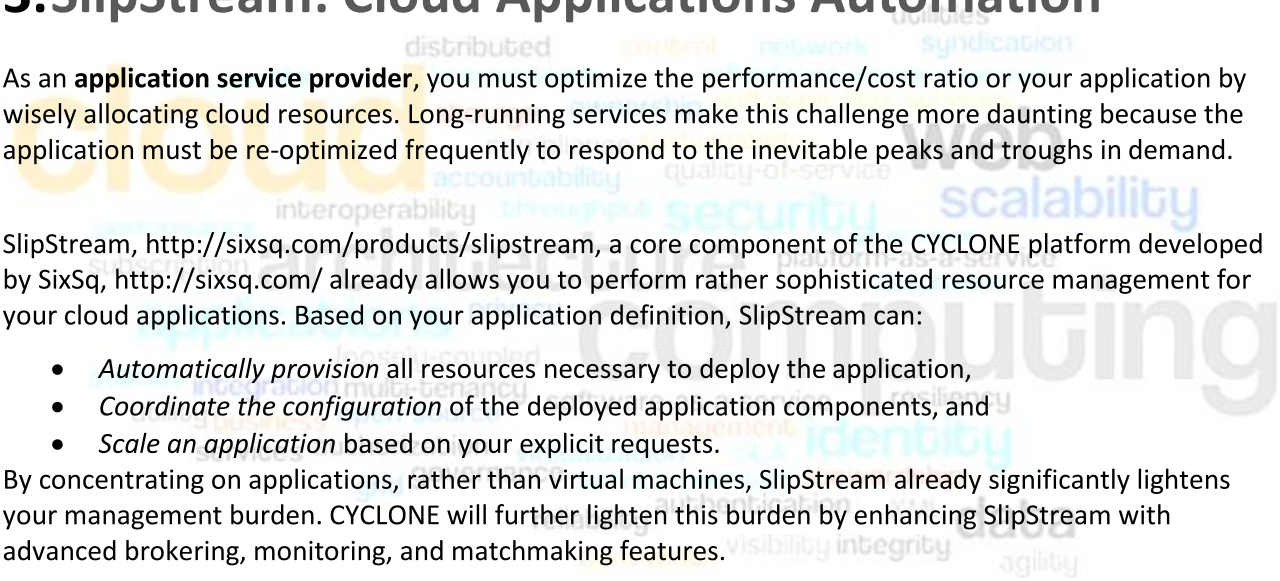
In Figure 6 the connection of the DERs to the ICT-Platform is shown. The location of the DERs can be geographically widely distributed, even over different countries. The DERs are connected via an Internet connection to the platform and send their data to the “Data Collection Service” of the ICT-Platform. With the “Data Collection Service” the data from DERs are stored as authentic raw data. Various aggregation, calculation and visualization services using the collected DER energy generation data are further parts of the ICT system to build the VPP.

2.4. Extending the Energy Use Case: Smart Utility

The energy use case Smart Utility comprises the first use case, the Virtual Power Plant (VPP), where distributed renewable energy sources were aggregated into one virtual power plant. To consider the requirements of future energy management which include also the areas of energy consumption and transport, the use case is extended by including energy consumers and the ICT platform services. The inclusion provides an energy management that enables the partners based on the aggregated energy data collected within the platform to work together with the goal to balance the generation and consumption of energy - thus forming the Smart Utility,

Considering the increasing number of renewable energy resources and the necessity to develop a smart grid management including the generated energy in an efficient and stable way, scalability and elasticity in the node deployment is an important requirement. For new partners joining the platform either on the generation and on the consumption side, the application management system must allow to dynamically deploy the appropriate services in the cloud requested by the partner.

3. SlipStream: Cloud Applications Automation



As an **application service provider**, you must optimize the performance/cost ratio or your application by wisely allocating cloud resources. Long-running services make this challenge more daunting because the application must be re-optimized frequently to respond to the inevitable peaks and troughs in demand.

SlipStream, http://sixsq.com/products/slipstream, a core component of the CYCLONE platform developed by SixSq, http://sixsq.com/ already allows you to perform rather sophisticated resource management for your cloud applications. Based on your application definition, SlipStream can:

- *Automatically provision* all resources necessary to deploy the application,
- *Coordinate the configuration* of the deployed application components, and
- *Scale an application* based on your explicit requests.

By concentrating on applications, rather than virtual machines, SlipStream already significantly lightens your management burden. CYCLONE will further lighten this burden by enhancing SlipStream with advanced brokering, monitoring, and matchmaking features.

Currently with SlipStream, you choose the placement of your application components explicitly; each application deployment requires human intervention. For most applications however, the placement constraints for an application can be expressed as a static "policy", opening up the possibility of automated placement. To do this, two interconnected CYCLONE components will be developed:

- A **brokering database** that contains characteristics of cloud service offers. The database will include a wide variety of information, such as pricing from Cloud Service Providers like Exoscale, https://www.exoscale.ch/, security certifications from organizations like the Cloud Security Alliance, https://cloudsecurityalliance.org/, benchmarks from sites like CloudHarmony, https://cloudharmony.com/, and metrics from the application itself. Having an open schema will allow all of these types of information to be included, allowing you to create rich deployment policies. The implementations will use JSON-LD, http://json-ld.org/, as an interchange format.
- A **matchmaker** that compares an application deployment policy vs. available cloud service offers. The implementations will take an application description and the application deployment policy to find acceptable cloud service offers from the brokering database. The policy can also reference a ranking algorithm (e.g. minimum price) to prioritize the acceptable offers. The resulting prioritized

list of offers can be used for automated placement as well as refined manual placement. The policies will initially be based on DMTF's CIMI filtering syntax https://www.dmtf.org/standards/cloud, and will evolve to meet the needs of the CYCLONE use cases.

To achieve true "1-click" deployments, CYCLONE will extend SlipStream to use these components. You will be able to specify the deployment policy for each component of the cloud application. With this and the application definition, SlipStream will use the Matchmaker to find acceptable placement options and use the results for fully-automated deployment.

Going a step further, application monitoring will be coupled with the automated provisioning to provide fully- automated scaling for applications using the CYCLONE platform. With this you can define self-managing cloud applications, eliminating much of the tedious work involved with cloud application management.

You can already use SlipStream and other components of the CYCLONE platform. If the advanced features interest you, we'd like to talk with you about your use case.

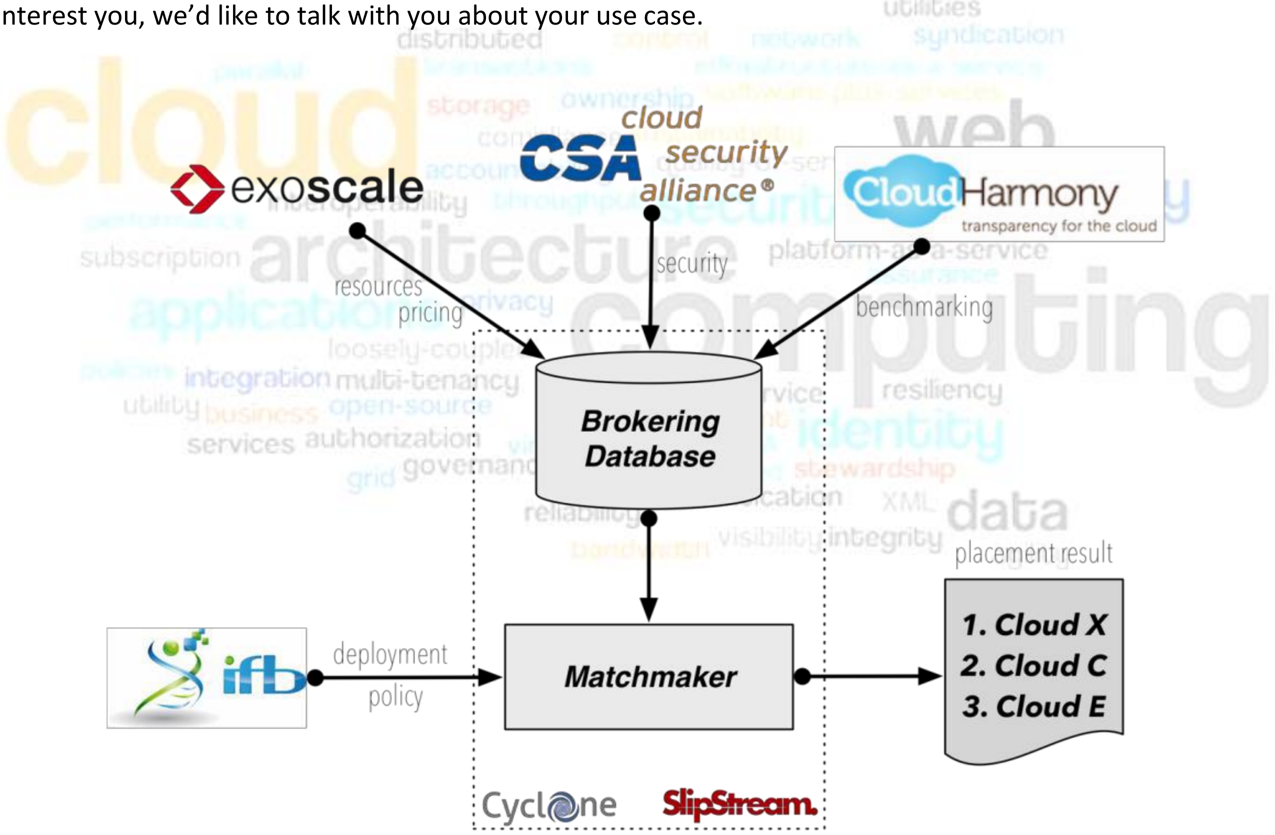


Figure 7: CYCLONE components that automate cloud resource provisioning through SlipStream.

Within CYCLONE, software developers and service operators manage the complete lifecycle of their cloud applications with SlipStream, an open source cloud application management platform. Through its plugin architecture, SlipStream supports most major cloud service providers and the primary open source cloud distributions. By exposing a uniform interface that hides differences between cloud providers, SlipStream facilitates application portability across the supported cloud infrastructures.

To take advantage of cloud portability, developers define “recipes” that transform pre-existing, “base” virtual machines into the components that they need for their application. By reusing these base virtual machines, developers can ensure uniform behaviour of their application components across clouds without having to deal with the time-consuming and error-prone transformation of virtual machine images. Developers bundle the defined components into complete cloud applications using SlipStream facilities for passing information between components and for coordinating the configuration of services.

Once a cloud application has been defined, the operator can deploy the application in “one click”, providing values for any defined parameters and choosing the cloud infrastructure to use. With SlipStream, operators may choose to deploy the components of an application in multiple clouds, for example, to provide geographic redundancy or to minimize latencies for clients. To respond to changes in load, operators may adjust the resources allocated to a running application by scaling the application horizontally (changing the number of virtual machines) or vertically (changing the resources of a virtual machine).

SlipStream combines its deployment engine with an “App Store” for sharing application definitions with other users and a “Service Catalog” for finding appropriate cloud service offers, providing a complete engineering PaaS supporting DevOps processes. All of the features are available through its web interface or RESTful API.

3.1. Nuv.la – Web based configuration and deployment tool

SlipStream, a cloud application management platform, allows developers to define portable cloud applications and for operators to deploy automatically those applications on multiple cloud infrastructures. With SlipStream, the operators can manage the full lifecycle of cloud applications, including provisioning, scaling, migration, and clean up. SixSq releases the SlipStream Community Edition under the Apache 2 license and the source code can be found in the SlipStream organization in GitHub. In addition, SixSq operates a free SlipStream SaaS called Nuvla that can be used to access a number of public clouds.
The Service Catalog, a core feature of SlipStream, contains “offers” from cloud service providers, detailing VM resource configurations, locations, prices, and other information. Developers and operators can attach “policies” that describe constraints to an application. SlipStream then uses those policies to filter the available offers to eliminate those that do not meet the application requirements. The operator can then select any acceptable offer manually or allow SlipStream to choose the least costly offer automatically. These policies are completely general and can be used, for example, to deploy an application to a particular country for legal reasons or to choose a particular combination of CPU cores, RAM, and disk space.

3.2. Functionality used for use cases deployment

The bioinformatics use cases described above principally used SlipStream’s facilities and tools to define applications and its deployment engine through the RESTful API.
The definition of an application component actually consists of a series of recipes that are executed at various stages in the lifecycle of the application. The main recipes, in order, are:

- **Pre-install**: Used principally to configure and initialize the operating system’s package management.
- **Install packages**: A list of packages to be installed on the machine. SlipStream supports the package managers for the RedHat and Debian families of OS.
- **Post-install**: Can be used for any software installation that can not be handled through the package manager.

- **Deployment**: Used for service configuration and initialization. This script can take advantage of SlipStream's "parameter database" to pass information between components and to synchronize the configuration of the components.
- **Reporting**: Collects files (typically log files) that should be collected at the end of the deployment and made available through SlipStream.

There are also a number of recipes that can be defined to support horizontal and vertical scaling that are not used in the defined here use cases. The applications are defined using SlipStream's web interface, the bioinformatics portal then triggers the deployment of these applications using the SlipStream RESTful API.

4. CYCLONE Virtual Private Network service

Modern computing platforms span multiple cloud infrastructures in order to achieve resilience, responsiveness, and elasticity. Most often, they require secured network connectivity, at best automatically managed and available on-demand. However, unless companies pay a significant amount of money for customized cloud infrastructure, many limitations persist in the network services offered by common public cloud vendors: first of all, the networking APIs and procedures differ widely between cloud providers, oftentimes to an incompatible degree. Secondly, tenants have little control over network services and limited visibility over networking resources. This severely limits tenants' flexibility and prevents them from implementing application logic in the network.

CYCLONE networking services rely on CNSMO (Cyclone Network Services Manager and Orchestrator), which is the software component responsible of deploying, configuring and running the networking services in cloud applications.

The OpenNaaS- Network Services Manager/Orchestrator (CNSMO) component has been designed to provide with the network and concrete security services. This provisioning is according to the CYCLONE use cases demands and specific CYCLONE platform needs. One of these services is the Virtual Private Network (VPN) service, which holds a twofold purpose: First, it is in charge of providing the secure connectivity between all client VMs even when these are provisioned by different cloud service providers. Second, the VPN service is considered to establish a secured connection between the CYCLONE user (ASP) and the VMs that put together the application being deployed.

CNSMO is a lightweight micro-services framework developed by the i2CAT Foundation. Leveraging the base concepts of Apache Mesos, CNSMO is a distributed platform defining a management API and service life-cycle, together with an inter-service communication mechanism to support agent discovery. The system is capable of deploying and running multiple services in both local and remote environments.

Although services may run each in a dedicated VM or container, due to requirements imposed by cloud providers, networking services run directly on the user-space of user defined VMs. These services have been carefully designed to have a small fingerprint. With the goal to minimize the impact on the user space, the network services are run in Docker containers.

In order to allow the maximum degree of flexibility for network service deployment and configuration, CNSMO is based on the Software Defined Network paradigm, implementing an overlay network that will exploit the underlying connectivity given by ISPs. Such overlay allows application providers to configure, modify and monitor the networking behavior of virtual machines independently of the inter- and intra-Cloud network (but inheriting the ISP SLAs). CYCLONE's networking solution Multi-Cloud since it is specially designed for distributed scenarios where the virtual machines live in different Data Centers building a multi-domain hybrid cloud.

In the interest to achieve the mentioned features in an unconstrained way, an Ethernet VPN is required to create a private LAN network between the virtual machines hosting the application components. In each of the VMs there is a bridge connecting the physical interface to a virtual switch, which is subscribed to an2 SDN controller that resides in one of the VMs on the same signalling VPN. As a result, the SDN controller is able to manage and monitor the network traffic across VMs instantiated in distributed clouds, regardless of the virtualisation technology.

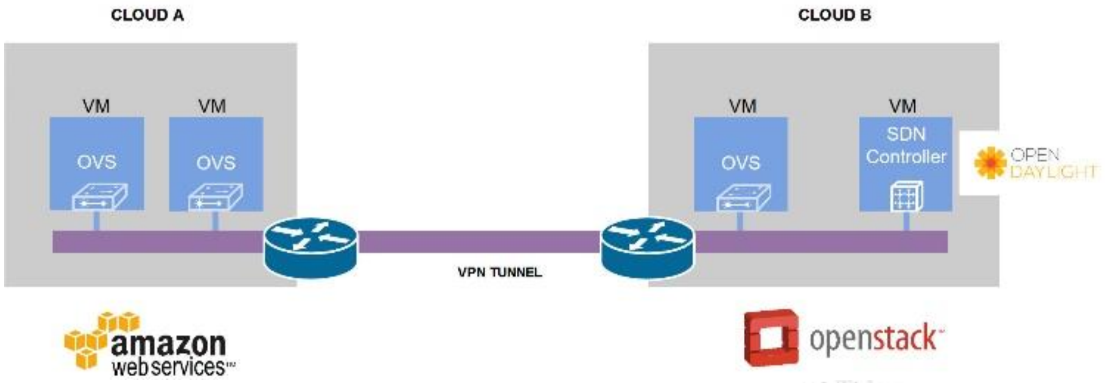


Figure 8. Architecture Overview of the CYCLONE SDN overlay network service.

The OpenNaaS-CNSMO is integrated as part of the SlipStream catalogue. It is instantiated and deployed by SlipStream as part of the overall application deployment to make the network services available to the ASPs. Therefore, it has been fundamental to ensure a proper integration of CNSMO with the SlipStream logic.

4.1.CYCLONE VPN service components

In order to deploy the VPN Service, the following components are required:

- **VPN Server**: This service expects to receive the certificates and configurations of the VPN Server, and provides API to launch the OpenVPN Server software.
- **VPN Clients**: This Service expects the certificates and configurations of the VPN Clients from the Orchestrator and provides API to manage the OpenVPN client software.
- **VPN Configurator**: This service is in charge to provide the configuration files required for the Server and the Clients and generate the required certificates and keys. As input, this service expects the generic configuration of the VPN like VPN Server Port, IP ranges used by the Clients, etc.
- **VPN Orchestrator**: It interacts with the rest of the services in order to configure and run the VPN as a whole. In order to be able to deploy the service, the orchestrator is subscribed to the other components and it won't start to deploy anything before checking that the status of the rest of the components is ready.

4.2.VPN service logic deployment and bootstrapping

Starting from the premise that CNSMO relies on SlipStream to deploy the network services in cloud federated environments, the general bootstrap process, illustrated at Figure 9 is as follows:

Step 1: The CYCLONE user (the ASP) specifies the application profile and requirements. To this end, the ASP must agree with the user of the application on the concrete aspects of the application and has to map them to a recipe in which the details of the required deployment are specified, including the details of the VPN service.

Step 2: Once the application recipe is ready, the user is able to run the application deployment.

Step 3: SlipStream prepares and deploys the requested VMs to run the user's application based on the requested recipe and bootstraps OpenNaaS CNSMO. Thus, CNSMO relies on the VMs deployed by SlipStream to incorporate the VPN service. Each time an application is deployed by SlipStream, it builds component images, runs those images in VMs and, afterwards, runs deployment recipes of each VM.

Step 4: Once CNSMO is launched, the CNSMO SlipStream application contains a deployment recipe with appropriate instructions for CSNMO how to deploy VPN service. The recipe is run by SlipStream inside the CNSMO VM. This recipe gathers information of the deployment from SlipStream and calls the CNSMO API to deploy the VPN service. The recipe then uses the SlipStream to announce to the rest of components that the VPN service has been set up, so SlipStream can resume their deployment.

Step 5: The deployment information is used by CNSMO to determine which services are deployed in which components. In the case of the VPN service, it adds a VPN server VM and allocates ***dockerized*** VPN clients at each of the application VMs.

Step 6: The application deployment is finalized and the application is ready to be used (including the VPN service).

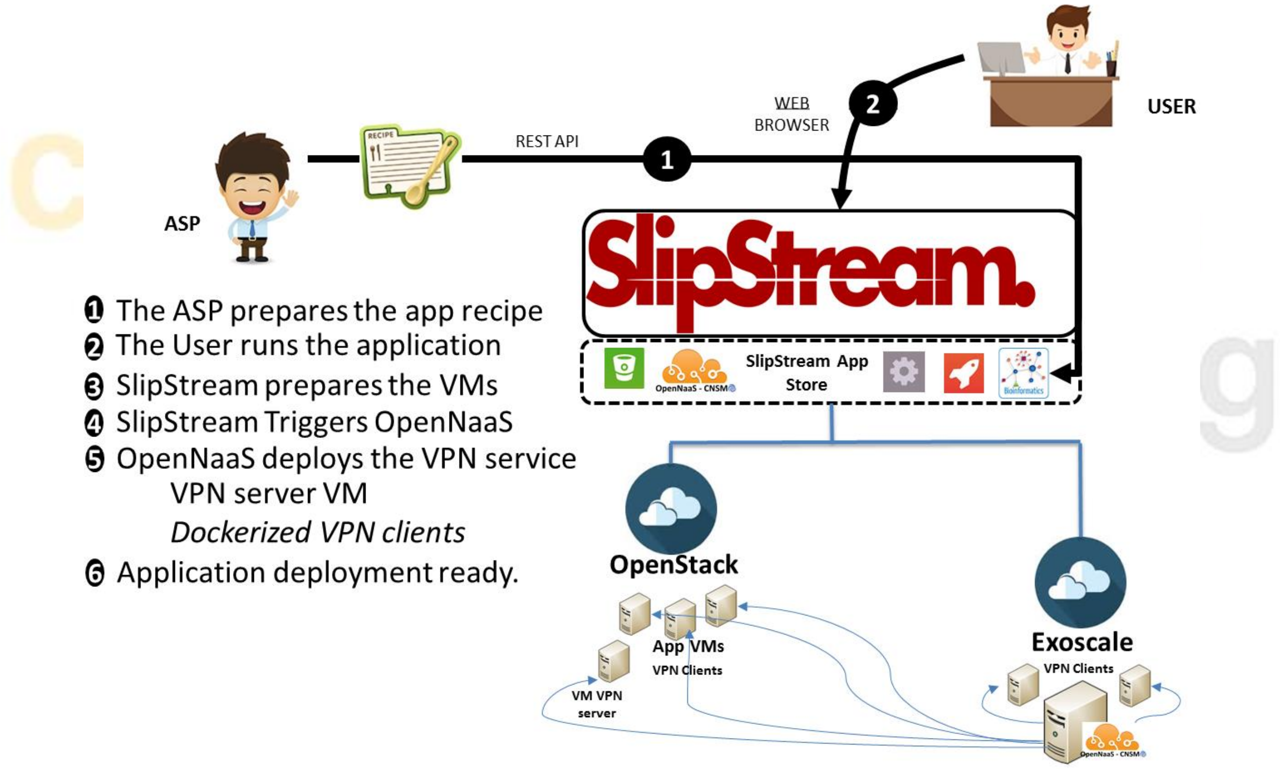


Figure 9. OpenNaaS-CNSMO bootstrapping process to deploy the VPN service.

Thus, the VPN service components are deployed by SlipStream as explained before in a set up consisting of the following VMs:

- **Server VM:** A VM that contains the VPN server CNSMO context, the VPN configurator CNSMO context and the VPN Orchestrator CNSMO context. The role of this VM is to stablish the link between the entire client VMs of the target VPN.
- **Client VMs:** The role of these VMs is just to run the user applications and have a VPN Client CNSMO context. The client VMs are the VMs deployed to run CYCLONE use cases' applications.

5. CYCLONE Federated Security Infrastructure

The CYCLONE security infrastructure aims to enable customer tailored security functionality in federated multi-cloud infrastructures by offering a set of ready-to-use components. The components are selected to best serve the needs of the distributed bioinformatics data processing infrastructure and the distributed cross organization collaborating research teams. The main identified security scenarios include: i) federated identity management, ii) federated authorization management, iii) end-to-end secure data management.

Federated authorization management addresses the collaboration between VM developers and bioinformaticians and the definition of access rules. It will be enabled by lists of users and groups defined at the pre-deployment stage which will be configured on the VMs. The user claims can be used to define a simple configuration with Require Statements in .htaccess on any Apache-hosted application with the OpenIdConnect Module enabled.

The security services are made available to a cloud developer in the form of reusable components that are deployed to provisioned VMs at selected providers during application deployment. The authorization standard eXtensible Access Control Markup Language (XACML) is at the core of security services. Since XACML allows authorization policies to be specified by using attributes, it also allows the use of these attributes in building context information.

The use of XACML as the central component allows easier management of permissions and better separation of concerns. For instance, bioinformatics researchers and admins can get an overview of permissions by looking at the policies and obtain insight about current authorizations to genome data, analysis results or available cloud resources such as VMs. XACML also allows dynamic authorization decisions based on context information. A change in the attributes of a user or a resource is sufficient to have an impact on authorizations.

5.1. CYCLONE Federation Provider: Federated Identities

The Cyclone framework provides the CYCLONE Federation Provider as an approach to ease the hardships of federated multi-cloud identity management. We make special arrangements to ease the integration of pre-existing academic identities that are federated through eduGAIN, as the end-users in many implemented CYCLONE use cases are academic researchers.

From a conceptual perspective, using a centralized authentication server decouples application authentication and reduces the functional footprint of application nodes. As we rely on widely used standards, the integration of Web-based SSO is easier because supporting libraries are widely available. Furthermore, the Federation Provider transforms different user identities into a consistent attribute format (JSON Web Token), decoupling the application node authentication (i.e., OpenID Connect) from the different authentication methods used at the Federation Provider.

From an implementation perspective, the CYCLONE Federation Provider extends and enriches the Keycloak identity and access management solution that is sponsored by RedHat. Keycloak has a rich feature set, mainly single sign-on supporting both SAML2 (as used by eduGAIN) as well as OpenID Connect (as used by many cloud applications). Keycloak can also broker identities, allowing end users to select which credentials they want to use for authentication, even supporting social network logins (e.g. Facebook). Our extensions to Keycloak comprise a data privacy aware session removal, an interface for self-service registration as well as templates that include terms of conditions and data privacy statements for each OpenID connect tenant.

There is a shared Federation Provider Instance in the CYCLONE testbed that is integrated with eduGAIN. Using such a shared instance is beneficial in two ways: first of all, integrating an application with eduGAIN is a manual process that, from our own experience, can take weeks and differs for each university. Second, we offer the eduGAIN user’s identities within an OpenID Connect flow, thus easing the implementation of relying cloud applications. At last, we also provide software sources so that, for example, other ASPs can implement their own Federation Providers with less efforts.

5.2. The CYCLONE PAM module

To access deployed bioinformatics applications via SSH, e.g., to upload research data, every deployed application requires a unique user account and for this, new credentials are established that the end users need to cope with. While this overhead could be reduced by Single Sign On, there is no usable solution for federated SSH login.

The CYCLONE PAM module uses the keyboard-interactive mode of SSH in combination with a custom PAM module to implement such a federated SSH login. The PAM module "pam_openid_connect" starts an embedded web server and displays its URL to the bioinformaticians in the SSH terminal session. When they open the link in their browsers, they are redirected to the CYCLONE Federation Provider where they authenticate with their federated ID using OpenID Connect. After authenticating, the Federation Provider returns the user’s information to the integrated PAM webserver and therefore to the PAM module and the Linux PAM subsystem. The PAM module then compares the user’s account identifier (e.g., email) with a list of user identities allowed to login via the requested system account. This list can be easily modified manually or it can be provided through SlipStream parameters to be used by deployment scripts.

5.3. Dynamic Access Control Infrastructure (DACI)

DACI allows for fine-grained authorizations in multi-tenant multi-cloud applications. The authorization logic employed by DACI is data-centric in that the data is considered to be the sensitive resource to be protected in a distributed application and the rules of access can be specified separately than the application processing it. It employs standards such as eXtensible Access Control Markup Language (XACML) and XML Signatures for interoperability.

The DACI implementation follows for the most part the REST guidelines except certain functionalities implemented as conventional Web services. It also fits to micro services paradigm such that the security services that constitute DACI can be replaced with alternative implementations (e.g. a Token service that uses Web tokens). DACI has been built by using Spring (boot) in Java and employs Redis database server to store policies, identifiers and so on.

The DACI services can be deployed over a cloud environment in two modes:

- **Standalone deployment**: All the services are deployed on the same VM. DACI standalone deployment casts all services as a single component and the installation script invokes all services on their respective ports given (among other information) during the deployment.

- **Distributed deployment**: Each service is deployed as if they will be running over a separate VM. The services then are tied together with an application that acts as an orchestrator. This implies that each service will have its own coordinates for access by external applications.

There can be many architectural reasons to invoke DACI with a standalone or distributed deployment including application design, scalability and security considerations. For instance, one may want to eliminate the network traffic due to DACI internal communication and install DACI on the same VM with the application itself. Similarly, a distributed deployment may enable more flexibilities in architectural decisions.

5.4. Bootstrapping Trust in Federated Clouds

Trust bootstrapping refers to the initialization of a single cloud node or the virtual private cloud with relevant security credentials. The project provides implementation of the proposed Dynamic Infrastructure Trusted Bootstrapping Protocol (DITBP). It leverages the Trusted Platform Module (TPM) and can be effectively implemented using available technologies and tools. TPM is typically available on all CPU boards and supported by majority of cloud platforms.

Among the available proposals, CYCLONE employs **keylime** for bootstrapping trust within cloud nodes and the services running on them. It can be considered as an instance of DITBP that employs a cloud verifier to periodically check the integrity of the node.

Keylime is a software package that allows installing tenant’s secrets to the provisioned VMs. It is particularly useful in environments where the cloud provider is untrusted. Consider a scenario where cloud resources are used to process sensitive data stored in encrypted form. In this scenario, the bootstrapping of the VM with the relevant secret keys is fundamental to detect and prevent any unauthorized access to sensitive data. In addition to installing secret keys, keylime enables periodical checking of the VMs (or the services/applications running on them) to detect any integrity violation or compromises.

One of the most important features of keylime is that it roots trust to hardware (i.e. Trusted Platform Module) to provide high-level of security. The CYCLONE implementation uses the TPM emulator rather than hardware TPM since CYCLONE infrastructure lacks TPM hardware module. There are three steps involved during the deployment of a VM node with keylime:

1. *Key generation* in which the tenant creates a fresh symmetric key K_t for each new node it requests and shares the key with the node and the verifier using secret sharing
2. *Node initiation* that refers to the instantiation of the node with the respective tenant configuration through cloud-init
3. *Key derivation* in which the secrets are installed to cloud nodes according to a secure key derivation protocol.

6. CYCLONE in Action

One of the main cornerstones of CYCLONE is the application of the CYCLONE middleware stack within a broad range of use cases. The following subsections highlight some use cases and show how well the stack fits to the requirements of the use case stakeholders.

6.1. Deploying Bioinformatics Software

The Insyght, a comparative genomic visualization tool, is actively used by the bioinformatics researchers and provides an important use case for cloud based deployment. It consists of 3 components: a pipeline of Perl scripts to compute the required data, a relational database to store these data and the visualization tool itself that queries the relational databases and presents these data in a user-friendly way. The platform automatically launches a set of bioinformatics tools (e.g. BLAST, PSI-BLAST, INTERPROScan) to analyse the data and stores the results of the tools in the relational database (PostgreSQL). These tools use several public reference data collections. A web interface allows the user to consult the results and perform the manual annotation (manual annotation means adding manually metadata and biological knowledge to the genome sequence). The popularity and vast functionality make Insyght a prime candidate for offering it on the IFB Bioinformatics Cloud.

The Insyght deployment comprises two components: a master running the workflow, scheduling the genomes comparisons and storing the result, and several nodes to perform the genomes comparisons. Previously, they were both deployed within a single image that needs to be imported to the target cloud. However, many clouds either do not allow importing custom VM images or do not support images built for other clouds. This challenge can be easily solved using the CYCLONE middleware: using SlipStream IFB developers can create generic deployment recipes that can be deployed to all major cloud platforms, such as the OpenStack cloud used in the CYCLONE testbed.

Deploying each component to different nodes requires one master and several worker nodes according to the size of the genomes dataset to analyze. The CYCLONE middleware stack also helps in making this task easy as Slipstream provides the facilities to deploy and scale heterogeneous applications consisting of different types of VMs. This set of VMs and their data exchange needs to be isolated from other cloud users and VMs for security and operating purposes, for example, to ease the management of the data exchange between the nodes or the NFS exports and mounts. For this purpose, we leverage the VPN service offered by CNSMO.

The CYCLONE Federation Provider and the PAM module provide easy and secure access management for the deployed VMs. They also provide reliable and ubiquitous identity management using user identities from eduGAIN federated identity providers. The PAM module has simplified the access of the end-users to their VMs by liberating them of managing the SSH keys, which can be problematic according to the computing skill of the user and the operating system that is used by them. At last, the PAM module consolidated the security of the cloud infrastructure from both the end-user and the cloud provider point of view.

6.2. Deploying the Energy Use Case

The energy management platform software includes several services for the collection and further processing the energy generation and consumption data. The energy data at the resources and the consumers are collected by embedded systems and sent to the respective cloud service for storage. Application interfaces are provided for access to the data according to the requirements of the partners. In addition, services are included to support visualization, data propagation and aggregation.

For the deployment of the VPP the storage service for each renewable energy resource is deployed to the cloud selected and respectively meeting the claims of the owner/operator. Also, the services and interfaces for data access and visualization are deployed there. The Virtual Power Plant management system is deployed to a dedicated node. In the selection of the various clouds the specifications of the partner according to regulation, company rules and location awareness must be met. Therefore, a multi-cloud deployment management is essential for cloud service brokering and matchmaking. That also applies to the nodes deployed for further partners in the energy system, here the consumers. The services to store, access, visualize and further process the energy consumption data are deployed to dedicated nodes. The deployment of the different nodes is implemented in components in the SlipStream cloud and application management system.

For the deployment of the application each component is deployed to a different node. From the CYCLONE tools SlipStream enables to scale the application and choose appropriate clouds at deployment time.

To enhance the application security and the data exchange between the VPP partner nodes, the nodes are isolated from other cloud users by integrating the CNSMO service for VPN into the VPP application.

The CYCLONE DACI authorization services provide the means to variably define and distribute authorization rules for the partners in the application. They facilitate the definition and the implementation of a secure data access via the provided interfaces.

7. Selected CYCLONE Publications

1. M. Slawik, Y. Demchenko, J. I. Aznar Baranda, R. Branchat, C. Loomis, O. Lodygensky, C. Blanchet. CYCLONE Unified Deployment and Management of Federated, Multi-Cloud Applications. Proceedings of the 5th workshop on Network Infrastructure Services as part of Cloud Computing (NetCloud 2015). 7-10 December, 2015.

2. B. I. Zilci, M. Slawik, A. Küpper. Cloud Service Matchmaking using Constraint Programming. Proceedings of the 24th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE 2015), pp. 63-68, 2015.

3. M. Slawik, B. I. Zilci, F. Knaack, A. Küpper. The Open Service Compendium: Business-pertinent Cloud Service Discovery, Assessment, and Selection. Proceedings of the 12th International Conference on Economics of Grids, Clouds, Systems and Services (GECON 2015). Springer. [Available online] http://arxiv.org/abs/1508.06119

4. José Aznar, Eduard Escalona, Isart Canyameres, Oscar Moya, Albert Vines, CNSMO: A Network Services Manager/Orchestrator Tool for Cloud Federated Environments, Proc. Med-Hoc-Net 2016: Annual Mediterranean Ad Hoc Networking Workshop, Barcelona, Submitted to IEEE Xplore and other Abstracting and Indexing (A&I) databases

5. Mathias, Slawik, Begum Ilke Zilci, Axel Kupper, Yuri Demchenko, Fatih Turkmen, Christophe Blanchet, and Jean-Francois Gibrat, An economical security architecture for multi-cloud application deployments in federated environments, GECON2016 Conference, 20-22 Sept 2016, Athens, Greece.

6. D. Gallico, M. Biancani, C.Blanchet, M. Bedri, J.-F.Gibrat, J.I.A.Baranda, D.Hacker, M.Kourkouli, CYCLONE: a Multi-Cloud Federation Platform For Complex Bioinformatics And Energy Applications, CloudNet 2016 Conference, 3-5 Oct 2016, Pisa, Italy.

7. Mathias Slawik, Begum˙ Ilke Zilci, Axel Kupper, Establishing User-relevant Cloud Service Repositories, Future Generation Computer System (FGCS), Submitted paper, 2016

8. Miroslav Zivkovic, Charles Loomis, Yuri Demchenko, Runtime application performance management for multi-cloud CYCLONE environment, IEEE CloudCom2016 Conference, 12-15 Dec 2016.

9. Yuri Demchenko, Miroslav Zivkovic, Cees de Laat, José Ignacio Aznar Baranda, Christophe Blanchet, Mohamed Bedri, Jean-François Gibrat, Oleg Lodygensky, Mathias Slawik, Ilke Zilci, Rob Branchat, Charles Loomis, CYCLONE: A Platform for Data Intensive Scientific Applications in Heterogeneous Multi-cloud/Multi-provider Environment, Fifth IEEE International Workshop on Cloud Computing Interclouds, Multiclouds, Federations, and Interoperability (Intercloud 2016), In Proc. IEEE International Conference on Cloud Engineering (IC2E), April 4 - 8, 2016, Berlin, Germany

10. Demchenko, Yuri, Fatih Turkmen, Christophe Blanchet, Charles Loomis, Cees de Laat, Cloud Based Big Data Infrastructure: Architectural Components and Automated Provisioning, The 3rd International Symposium on Big Data Principles, Architectures and Applications (BDAA 2016), as part of The International Conference on High Performance Computing and Simulation (HPCS 2016), 18-22 July 2016, Innsbruck, Austria.

11. Demchenko, Yuri, Fatih Turkmen, Ching-Hsien Hsu, Christophe Blanchet, Charles Loomis, Cees de Laat, Cloud Computing Infrastructure for Data Intensive Applications, Book chapter, In book "Big Data Analytics for Sensor-Network Collected Intelligence", Elsevier, 2017 (in production)

12. Yuri Demchenko, Fatih Turkmen, Mathias Slawik, Defining Intercloud Security Framework and Architecture Components for Multi-Cloud Data Intensive Applications. Sixth IEEE International Workshop on Cloud Computing Interclouds, Multiclouds, Federations, and Interoperability (Intercloud 2017), In Proc. 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. Madrid, Spain, May 14-17, 2017

13. Mathias Slawik, Yuri Demchenko, Fatih Turkmen, Alexey Ilyushkin, Cees de Laat Charles Loomis, Christophe Blanchet, CYCLONE: The Multi-Cloud Middleware Stack for Application Deployment and Management, IEEE CloudCom2017 Conference, 11-14 Dec 2017, Hong Kong.

14. Yuri Demchenko, Cloud and Big Data infrastructure security and compliance, Tutorial abstract, IEEE CloudCom2017 Conference, 11-14 Dec 2017, Hong Kong

15. Poster. Intercloud Security Framework and Architecture Components for Multi-Cloud Data Intensive Applications, by Yuri Demchenko, Fatih Turkmen, Cees de Laat (UvA), Eduard Escalona, Mathias Slawik, Christophe Blanchet (CNRS-IFB), Oleg Lodygensky (CNRS-LAL), Cal Loomis (SixSq), Ralf Fischer (QSC), EGI Conference 2017 and INDIGO Summit 2017, 9-12 May 2017, Catania, Italy

16. Poster: CYCLONE Networking Architecture for Multi-cloud Applications, by Eduard Escalona, Albert Viñés (I2CAT), Yuri Demchenko, Fatih Turkmen, Cees de Laat (UvA), Mathias Slawik, Christophe Blanchet (CNRS-IFB), Oleg Lodygensky (CNRS-LAL), Cal Loomis (SixSq), Ralf Fischer (QSC), Presented at GEANT/TERENA Networking Conference TNC17, 29 May – 2 June 2017, Linz, Austria.

17. Poster. CYCLONE Security Middleware: Enabling Security infrastructure for Multi-Cloud Applications, Deployment Automation and Compliance, DI4R 2017 Conference, 30 November – 1 December 2017, Brussels.