Cyclone

Complete Dynamic Multi-cloud Application Management

Project no. 644925

Innovation Action

Co-funded by the Horizon 2020 Framework Programme of the European Union

Call identifier: H2020-ICT-2014-1

Topic: ICT-07-2014 – Advanced Cloud Infrastructures and Services

Start date of project: January 1st, 2015 (36 months duration)

Deliverable D4.2

Multi-cloud Security

Due date:	31/12/2016
Submission date:	30/12/2016
Deliverable leader:	TUB
Editors list:	Mathias Slawik (TUB), Ilke Zilci (TUB)

Dissemination Level

☒	PU:	Public
□	PP:	Restricted to other programme participants (including the Commission Services)
□	RE:	Restricted to a group specified by the consortium (including the Commission Services)
□	CO:	Confidential, only for members of the consortium (including the Commission Services)

List of Contributors

Participant	Short Name	Contributor
Interoute S.P.A.	IRT	
Sixsq SARL	SIXSQ	
QSC AG	QSC	
Technische Universitaet Berlin	TUB	Mathias Slawik, Ilke Zilci, Dirk Thatmann
Fundacio Privada I2CAT, Internet I Innovacio Digital A Catalunya	I2CAT	José Aznar Baranda, Isart Canyameres
Universiteit van Amsterdam	UVA	Fatih Turkmen, Yuri Demchenko
Centre National De La Recherche Scientifique	CNRS	

Change history

Version	Date	Partners	Description/Comments
0.4	Dec 13 2016	TUB	Revised content considerably
0.5	Dec 21 2016	TUB, i2CAT	Final draft ready for internal review
0.5	Dec 27 2016	TUB	Comment internal review addressed
0.6	Dec 29 2016	UvA	Additional multi-cloud analysis and use cases definition included
Final	Dec 29 2016	I2CAT. Interoute	Final version submitted to EC

Executive Summary

This deliverable summarises the development of the CYCLONE security services that comprise security infrastructure motivated by the initial set of use cases defined in WP3 and deliverable D3.1 and provides suggestions for further extension of the basic use cases and security services to address needs for multi-cloud application platform and corresponding security services. The document refers to specific use cases requirements that indicate need for multi-cloud applications platforms that will require corresponding multi-cloud security services. The document presents analysis of the potential multi-cloud use cases that include a general multi-cloud and Intercloud use case and the proposed bioinformatics use case extension that identifies necessary functional application infrastructure and security components that would allow using distributed cloud based resources and data sets, including possible application workflow migration or outsourcing to external cloud. The presented analysis confirm benefits of consistent implementation of the federated multi-cloud security model that can be potentially integrated with the currently widely adopted by the major cloud service providers the federated access control and federated identify management model.

The deliverable provides overview of the security services that can be used in multi-cloud applications such as currently implemented federated identify management using eduGAIN, secure shell login using eduGAIN federated identities, and new services being developed such as multi-domain Attribute Based Access Control, security services lifecycle management and trust bootstrapping for virtualised cloud environment.

The deliverable provides suggestions for further CYCLONE security infrastructure development and their evaluation in the testbed deployment.

Table of Contents

Figures Summary

Tables Summary

1. Introduction and overview

This document summarises CYCLONE developments that have been carried out during Y2. It both provides a high-level overview on the multi-cloud security functions as well as concrete information on their implementation and testbed deployment. This report refers to the previous WP4 and WP3 deliverables that defines and influence the CYCLONE security infrastructure:

- D4.1 and D4.3 (M10) established the basis for this document, defining the general CYCLONE security components and architecture [1. 2]
- D3.1 (M10) identified the basic CYCLONE use cases and specified requirements to CYCLONE components, including security related [3]
- D4.4 (M24) provides consolidated information on the CYCLONE security components and their operation [4].

1.1. Scope of CYCLONE multi-cloud security: providing use case security

Computer security is an enormously varied field of computer science and cloud security uses the best practices and recent developments in computer security ensuring compliance with the corresponding computer and network security standards. Cloud Services Providers (CSP) implement shared responsibility model where cloud provider insures security of their cloud services that include security of cloud infrastructure (i.e. virtualised computing, storage and network resource/services), platform (for hosting customer applications) or applications when providing IaaS, PaaS or SaaS services (see AWS example in [5]). The customers are responsible for the security of their applications and services starting from OS and platform or container and including user accounts management and access control for both application developers and users.

In developing security services for cloud based applications, we focus on customer controlled security services and rely on the security compliance of the cloud platforms that is ensured by the providers. Considering the general scope of project, we focus our efforts on providing specific security services for cloud based applications and correspondingly for application development, deployment and management tools and platforms. The CYCLONE security services are motivated by the CYCLONE use cases and are focused on such missing functionality as authentication and authorization for customer developed applications on ALL cloud layers (e.g., web-based single sign-on as well as SSH login) using federated identities in academic settings.

The CYCLONE security architecture relies on the lower layer and cloud infrastructure security services and provides applications related security services and practical tools that can be adapted and applied for specific applications and implementation platforms – either singularly or in combination. Modular construction and simplicity allows their reusability and composability, including their easy integration with the production-grade tools and established industry-recognized standards, e.g., Keycloak and OpenID Connect.

With the critical need to deliver operational security infrastructure at the early project stage to allow working with the CYCLONE use cases, the security development followed the extreme programming approach in delivering first the core and basic functionality (following “Do the simplest thing that could possibly work”) motivated by the use cases requirements that have being implemented. Further security infrastructure and services development should be motivated by new use cases or new required functionality of already operated use cases as they evolve, in particular, bioinformatics use cases when they involve more resources and data source extend scope of the genome research. Such extension and development is seen in using multi-cloud scientific and user data (see D3.1 use cases requirements) as well as integrating external cloud based applications, in particular for outsourcing voluminous computation to specialised scientific clouds. This is especially important in the context of the recent EU initiative on European Open Science Cloud (EOSC) which defined as a cloud of services for scientific applications [6].

1.2. Limitations

Extending current CYCLONE security infrastructure will be based on the current existing and deployed security services and components with adding new components for managing security services, identity and access credentials and security context in multi-cloud and multi-domain environment, which will use commonly accepted and supported by major cloud providers the federated security model.

However there are some limitations in multi-cloud security that still exist:

- **Using federated identities in non-browser scenarios.**

  OpenID Connect is best used within browser-based scenarios, e.g., web single sign-on. For command line usage, the OpenID Connect Direct Access Grant was designed to query for a token directly, specifying the account name and password in the process. However, implementing this would require support by the EduGAIN Identity Providers that is not there yet. For SAML 2.0 there is the "Enhanced Client Profile" (ECP) which was designed for enabling federated identities to be used on the command line, e.g., for SSH login. However, while Shibboleth supports SAML 2.0 ECP, none of the 1,446 Identity Providers in EduGAIN support the "Reverse SOAP (PAOS) Binding" (urn:oasis:names:tc:SAML:2.0:bindings:PAOS) required for ECP.

- **Multi-cloud account management**

  SlipStream currently manages clouds on behalf of the users, persisting their credentials for later use, e.g., to instantiate applications on different clouds. What SlipStream could do in addition is implementing yet more management functions, e.g., updating credit card details or analysing cloud invoices. Currently, no public cloud does expose those functions via an APIs, it is very challenging to implement this in a reliable and secure manner.

1.3. Document structure

This document is structured as follows. Section 2 provides reference to the general multi-cloud and inter-cloud use case and analyses multi-cloud extension for the bioinformatics use case 3 – Live remote cloud processing of sequencing data which provides a basis for general multi-cloud security requirements definition. Section 3 describes the high-level approach to multi-cloud security functions and components which is followed by details about their implementation. Section 4 provides information about the testbed deployment. Finally, section 5 explains the future planned activities.

2. Multi-cloud Use Cases Analysis

This section will provide background for discussing multi-cloud security and corresponding security components in section 3. The section refers to the general multi-cloud and inter-cloud use case defined in the authors previous research on Intercloud[1] In Architecture Framework (ICAF) and Intercloud Federation Framework which are used to formulate general requirements to multi-cloud security, including conceptual need for inter-cloud network infrastructure that needs to be provisioned as a part of multi-cloud customer application. The section describes possible multi-cloud extension for the bioinformatics use case 3 – Live remote cloud processing of sequencing data which can respond to future use of distributed multi-cloud data and multi-cloud deployment of scientific applications.

2.1. General multi-cloud and Intercloud use cases

The following general use cases motivated the definition of the general Intercloud Architecture Framework (ICAF) [7] and Intercloud Federation Framework [8] in the previous authors research:

(1) Enterprise IT infrastructure evolution and migration to cloud that will require both the integration of the legacy infrastructure or private cloud components with the external public cloud components with further outsourcing specialised services to multiple cloud providers.

(2) Large project-oriented scientific infrastructures (capable of handling big data) including dedicated transport network infrastructure that need to be provisioned on-demand

Figure 1 illustrates the typical e-Science or enterprise collaborative infrastructure that includes enterprise proprietary and cloud based computing and storage resources, instruments, control and monitoring system, visualization system, and users represented by user clients and typically residing in real or virtual campuses

The main goal of the enterprise or scientific infrastructures is to support the enterprise or scientific workflows and operational procedures related to processes monitoring and data pro- cessing. Cloud technologies allow to simplify building such infrastructures and provision them on-demand. Figure 2 illustrates how an example enterprise or scientific workflow can be mapped to cloud based services and next deployed and operated as an instant Intercloud infrastructure. It contains cloud infrastructure segments IaaS (VR3-VR5) and PaaS (VR6, VR7), separate virtualised resources or services (VR1, VR2), two interacting campuses A and B, and interconnecting them network infrastructure that in many cases may need to use dedicated network links for guaranteed performance data traffic isolation.

[1] We use word “Intercloud” in a capitalized form denoting this as a term used previous published works and industry standards.

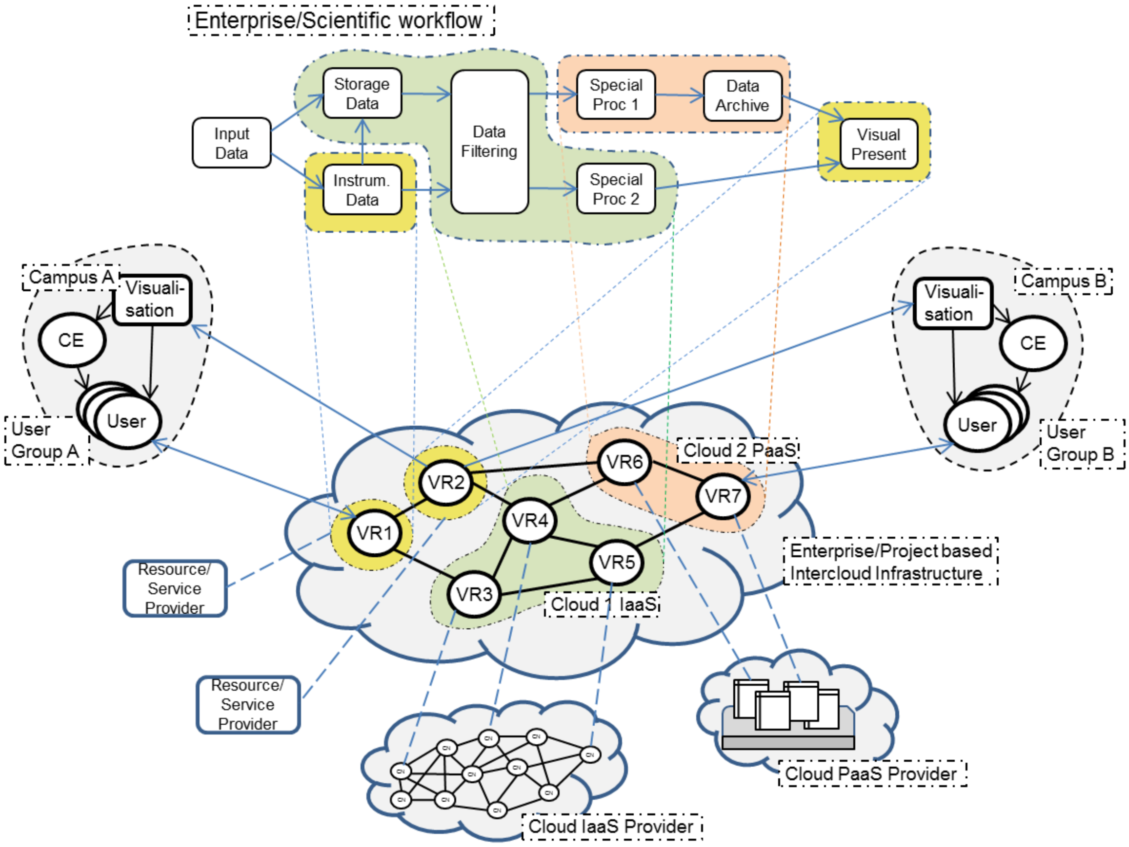

Figure 1. Enterprise or project oriented collaborative multi-cloud infrastructure created to support enterprise or scientific workflow.

Figure 2 presents an infrastructure centric view that demonstrates an important role of the interconnecting inter-cloud network infrastructure that is represented by FADI (Federated Access and Delivery Infrastructure) to indicate common approach to use federated access control in multi-provider multi-domain environment. Practical FADI implementation in heterogeneous multi-cloud Intercloud environment may benefit from using such technologies as multi-domain VPN which can be powered by Software Defined Networks (SDN) for provisioning overlay FADI network on demand. Such functionality is generally supported by the CYCLONE CNSMO that can provide inter-cloud network services (see section 3.7).

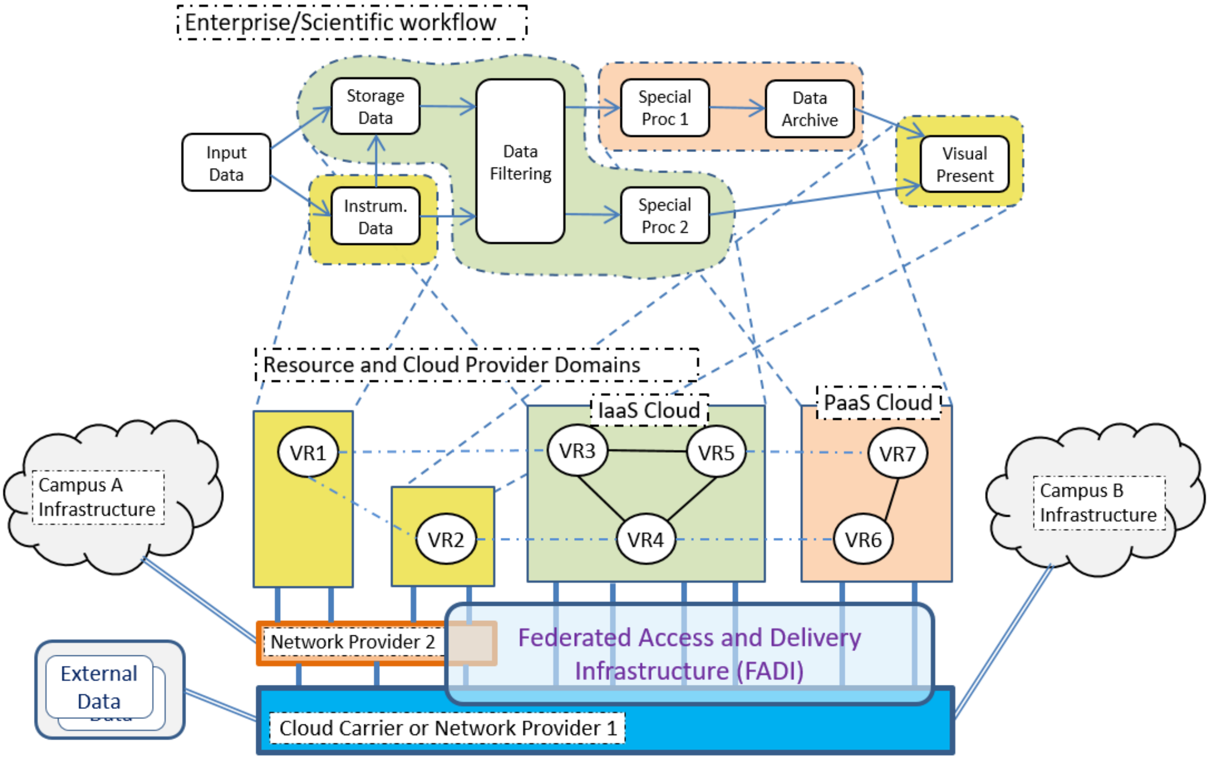

Figure 2. Infrastructure centric view of the enterprise or scientific multi-cloud infrastructure. The FADI is introduced as federated network infrastructure that inter-connects multiple cloud based and non-cloud resources.

2.2. General requirements for multi-cloud security services

The discussed above general use case for multi-cloud applications infrastructure together with the knowledge of the cloud security best practices allows us to specify the following general requirements to multi-cloud and Intercloud security infrastructure:

- Provide access control, security credentials and security context management for multi-cloud applications deployment, operation and management, in general covering all application lifecycle.
- Allow users and applications (internally and on behalf of users) to access all distributed multi-cloud resources using single credentials that should be federated with the individual cloud credentials and access control mechanisms.
- Application based access control must be integrated with the cloud based security services and implement in a consistent way the shared security responsibility model that is defined and implemented by cloud services providers as a standard cloud services security model.
- Support federated access control and resource management model, allowing integration with the cloud federation services.
- Support multi-cloud secure logging services.
- Ensure data protection during the whole data handling lifecycle, including data transfer between different clouds and security domains as well as data storage in-rest.

- Provide secure trust bootstrapping for the provisioned on-demand cloud based security services that should bind the deployed security services to the applications runtime environment and virtualisation platform, to prevent unauthorised virtual environment cloning.

The proposed multi-cloud and Intercloud security requirements can provide a basis for defining specific requirements to CYCLONE multi-cloud security infrastructure that should be aligned with the corresponding use cases.

2.3. Extending CYCLONE use cases to multi-cloud scenarios

CYCLONE use cases requirements includes four specific requirements to address need for multi-cloud applications and adopt generically multi-cloud nature of scientific data and increasing multi-source industrial data (requirements numbering follows original numbering in Table 3-1 Deliverable D3.1 [3]).

Excerpt from Table 3-1 Common list of use case requirements Deliverable D3.1

ID	Title	Description
20	Multi-clouds deployment of complex application	A complex bioinformatics application requires to be deployed over two or more cloud infrastructures to obtain the necessary computing resources.
21	Multi-clouds distribution of community reference datasets	The deployment of a bioinformatics workflow over two or more cloud infrastructures requires that the collections of public reference data used during the treatment is available in all of these clouds
24	Dynamic Network resource allocation	Deploying a complex bioinformatics application requires the distribution of the user data in a secure way in several cloud infrastructures. The user data can be files, relational or NoSQL databases.
25	Multi-clouds distribution of user data	Deploying a complex bioinformatics application requires the distribution of the user data in a secure way in several cloud infrastructures. The user data can be files, relational or NoSQL databases.

Initially defined CYCLONE use cases doesn’t explicitly require multi-cloud implementation limiting required cloud infrastructure to single cloud service provider, however mentioned above requirements imply that some use cases may evolve to multi-cloud implementation. To provide a context for the CYCLONE multi-cloud security infrastructure definition we will extend the Bioinformatics use case UC3 - Live remote cloud processing of sequencing data to its possible implementation in multi-cloud environment as illustrate din Figure 3.

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 dozen 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 3 shows the bioinformatics application deployed in Cloud1 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 application in Cloud1 using development tools coupled or integrated with the SlipStream cloud automation tools. The application may use external scientific data and applications located in SciCloud A and B. In case of excessive workload, some computational tasks can be outsourced to external cloud CloudExt, in particular in a standard cloudburst scenario. Similarly to original use case definition, Figure 3 includes Scientific Data archive for storing obtained scientific results data. Application user bioinformatician researchers may use data visualisation and collaboration tools that all can be hosted in cloud and provided by specialised SaaS or cloud applications providers.

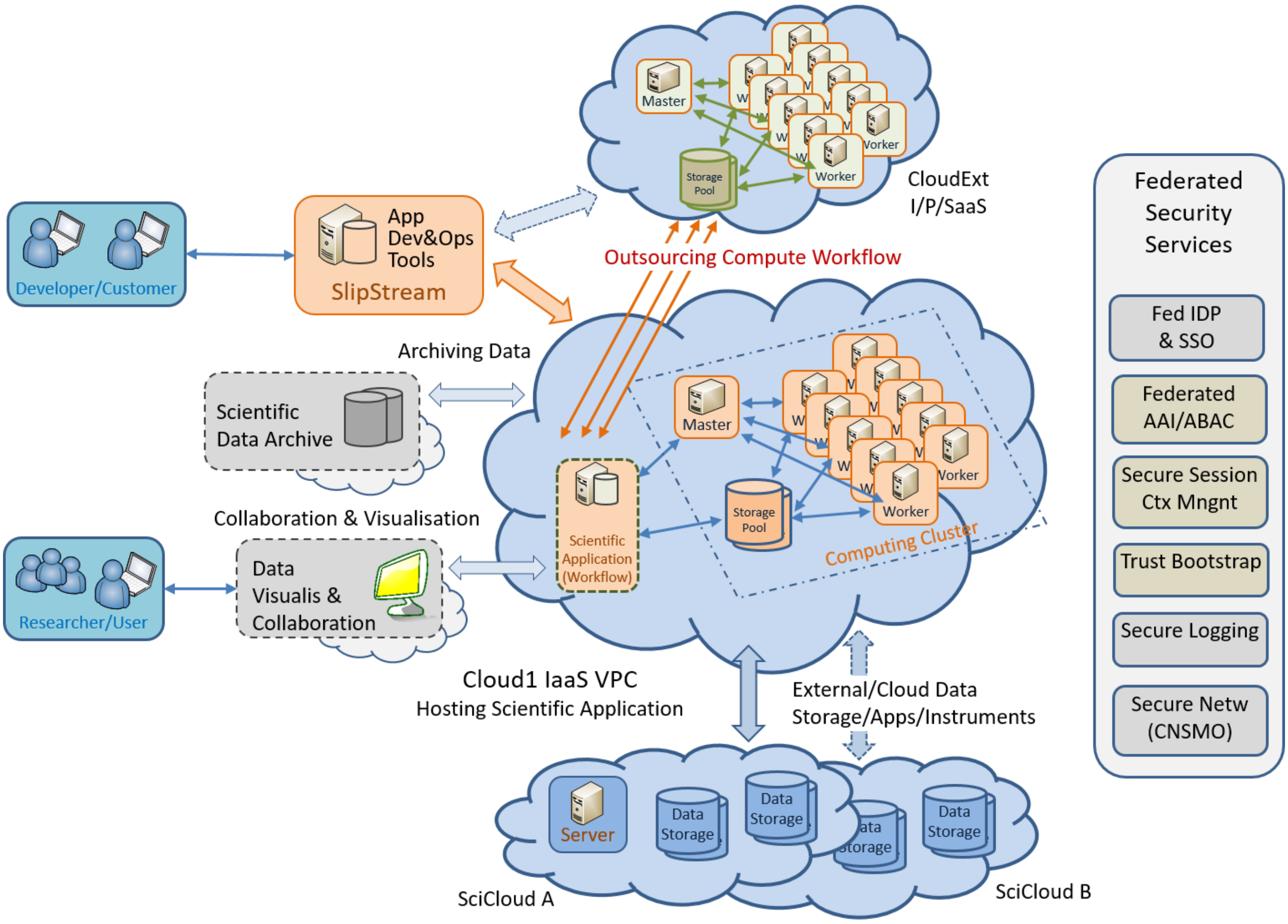

Figure 3. Bioinformatics use case 3 “Live remote cloud processing of sequencing data” extended to multi-cloud environment.

Suggested security services are combined into the federated security services stack, as depicted on the right side of Figure 3, that include currently existing CYCLONE security services and those that are required for multi-cloud applications which are discussed in the next section.

3. CYCLONE Approaches to Multi-cloud Security

This section explains the diverse focus areas of our approach to multi-cloud security. While explicating each area, we put special emphasis on the relation of the respective approach to the state of the art as well as other tools in the respective domains.

3.1. Federated Identity Management in Research Environments using eduGAIN

In CYCLONE Bioinformatics use cases, there are two main scenarios of utilising cloud for collaborative research. The first one focuses on authentication to web applications with dashboards which are deployed on research clouds. The second scenario focuses on sharing SSH access to virtual machines on the cloud in the post-deployment phase. This chapter explores alternative solutions and compares them to the current CYCLONE Federation Provider focusing on their suitability for the use cases.

3.1.1. Alternatives to CYCLONE Federation Provider Base Technologies

Another solution approach which was conceptually evaluated in the CYCLONE context is to use Kerberos V5 and interconnect Kerberos domains using Kerberos' Domain-Trust mechanisms.

Kerberos is a protocol which enables SSO on system level services. Web services were not foreseen in its original RFC. However, Microsoft proposed the "Simple and Protected GSSAPI Negotiation Mechanism" (SPENGO)(RFC 4559, 4178). SPNEGO brings the system level Kerberos ticket available on the Client (from the local ticket cache) and previously retrieved from a Key Distribution Center (KDC) to the browser for a further authentication to web services. SPENGO helps to negotiate what common GSSAPI mechanisms are available at the web service, selects one and then dispatches all further security operations to it. SPNEGO must be supported by the Browser on the client side. At the time of writing, all major browsers such as IE, Mozilla, Mozilla Firefox, Konqueror, Chrome and Safari support SPENGO.

3.1.1.1. Technical Evaluation of Trusted Kerberos Domains for CYCLONE

SPENGO supports two authentication modes. For CYCLONE use cases, GSSAPI (Kerberos) is of special interest. Using Kerberos tickets for Web-based authentication requires a GSSAPI enabled web service (e.g. WS-Security/SOAP or Django app with GSSAPI configuration). SPENGO is a Microsoft driven negotiation mechanism. However, since there are open source browsers which include SPNEGO and FLOSS web frameworks which are GSSAPI compatible, we assume that a Linux Kerberos client, Linux Kerberos and Linux Web Server works out of the box with SPNEGO. This is the combination of systems which are mostly used in our use cases. Therefore, interconnected Kerberos Domains meet the basic requirements of CYCLONE.

However, there are some drawbacks of such a deployment which prevent it from being ideal for CYCLONE use cases. Unfortunately, Kerberos tickets are by design incompatible among Microsoft and Linux Kerberos implementations. This causes unnecessary complexity especially in heterogeneous Windows/Linux/MacOS landscapes: i.e. the need to install the MIT Kerberos Client for Windows on Windows Client in order to connect to a Linux Kerberos KDC.

Another disadvantage when using Kerberos is the need to open several network ports for setting up the kerberos realm trust relations. Different ports for inbound and outbound traffic have to be opened to establish trust relations and use a Kerberos SSO/authentication during runtime. This includes ports such as 389, 135, 88 or/and 445. In a SPENGO-related web-service scenario, the web service has to communicate with the Kerberos KDC using port 88. This is not required in plain SAML/Shibboleth or OpenID Connect/JWT related authentication scenarios, which has been a major requirement in Cyclone.[2]

To manage Kerberos tickets each university or research organization needs to run a Kerberos instance which is trusted by others. The participating nodes in the whole system, deployment manager Nuv.la, Cloud Service Providers and Cloud application providers should be kerberized in order to complete the authentication. Each of these nodes can host an own KDC or register at a trusted KDC.

In multi cloud environments, compatibility and interoperability might be an issue when using Kerberos since the clouds (OpenStack-based, Azure or AWS) could require different Kerberos implementations. Cross domain solutions exist, however these are possible to roll out only with multiple intermediary components which translate between the incompatible tickets on the web level and system level, resulting in a bulky system according to our estimations.

3.1.1.2. Usability of GSSAPI vs OpenId Connect from the perspective of Cloud Application Providers

As mentioned in the Section … there several web service technologies (WS-Security, SOAP) which allow web applications to to work with GSSAPI and Kerberos tickets. In addition, mod_auth_kerberos with Apache can be used to move this task from the application itself to the web server. However, for RESTful web services OpenID Connect is more popular in the community due to active use and documentation by the communities.

Cloud application providers, especially SaaS, work with web interfaces and web technologies such as SAML and JWT. We can report this from our own experience with Keycloak and its perceived ease of use by our developers. A translation between JWTs and Kerberos tickets could be an option, however its complexity is not fully evaluated by the authors at the time of writing.

3.1.1.3. Usability from the perspective of End users (eduGAIN users, bioinformaticians)

For our use cases, CYCLONE Federation Provider uses the technologies which address most valued requirements of the functionality and the existing technical environment. On the client side of eduGAIN users, currently installed systems work without further installations or configurations.

3.1.2. CYCLONE Approach

Our current solution with the CYCLONE Federation Provider and CYCLONE PAM module allows easy set up of centralized authentication to web applications as well as sharing SSH access to customized virtual machines on cloud.

CYCLONE Federation Provider enables web applications to easily integrate since it provides the OpenID Connect and JSON Web Tokens (JWTs). These are state-of-the-art technologies which are supported by major web frameworks with libraries which are actively updated. Moreover, with this solution existing eduGAIN member identity providers are integrated to the CYCLONE ecosystem with minimal configuration effort.

Please also refer to Section 1.2 Limitations on other features of the technologies which CYCLONE currently utilises which are not available in the specific environment CYCLONE currently works in.

3.2. Secure Shell Login using eduGAIN Federated Identities

In multi-cloud environments, every new cloud introduces a new user account, increasing the number of passwords that end users must deal with. This overhead can be reduced using Single Sign On (SSO) of

[2] See sample flow: http://www.oracle.com/technetwork/articles/idm/weblogic-sso-kerberos-1619890.html

federated identities. However, there is no satisfying SSO implementation for Secure Shell Login that can be used web scale.

This challenge - implementing SSO using federated identities (e.g., for bioinformaticians) - is the focal area of the CYCLONE PAM Module. For the CYCLONE bioinformatics use case, using federated eduGAIN identities for SSH login is the most obvious way of implementing SSO at CNRS, since each researcher already has an EduGAIN identity, bound to their institutional email address.

There are two actors involved:

1. “sequencing engineer” who starts the deployment
2. “bioinformatician” who needs to access the deployed VMs

All explained workflows are strongly related to UC3 as described in Deliverable D3.1.

3.2.1. Workflow WITHOUT PAM module

First, the sequencing engineer deploys and instantiates one or multiple images. Afterwards, the sequencing engineer logs in via SSH to the VM and adds new system users for all bioinformaticians who need SSH access to the machine. This is done using public key authentication with the key that is deposited on the bioinformatics portal. After this configuration is done, bioinformaticians can now log in to the VM to collaborate, debug, or provide other support.

The scheme described above poses some problems:

- Bioinformaticians prefer not to use public key authentication due to usability issues
- Sequencing engineers have to use SSH configuration tools directly to manage and map bioinformaticians to system users
- In multi-cloud deployments, manual key distribution by sequencing engineers is not feasible, due to its required efforts

3.2.2. CYCLONE approach

The CYCLONE solution consists of using the keyboard-interactive mode of SSH in combination with a custom PAM module, implemented using the Python PAM bridge for simplicity. This PAM module "pam_openid_connect" starts an embedded web server and displays its URL to the bioinformaticians. When they click on the link they authenticate via the CYCLONE Federation Provider following the regular OpenID Connect flow. After authenticating with the web server, it returns user’s information to the PAM module and the Linux PAM subsystem. There is a list of email addresses from the users that are allowed to login. This list can be modified manually, or provided through SlipStream parameters.

3.2.3. Comparison to Other Solutions

The sequencing engineers could configure password authentication for SSH. However, this has pitfalls both in terms of security and usability:

1. Each bioinformatician receives a new set of username and password for the specific VM instance.
2. The sequencing engineer must securely share the username and password with the target bioinformatician
3. The number of passwords that bioinformaticians have to know in this case is not easily manageable.
4. The sequencing engineer has to manage usernames and passwords of bioinformaticians who received access to the VM.

All of this can lead to increased efforts as well as security leaks.

However, using the CYCLONE PAM Module, there are many advantages:

- The password does not leave the domain of the IDP, therefore it is secure against possible exposure to external systems.
- For sequencing engineers and bioinformaticians it is much easier to use.
- Only the sequencing engineer has to generate a key pair that is automatically distributed to the machines.
- A list of bioinformaticians' email addresses that need access can be easily distributed by the CYCLONE deployment manager, Nuv.la.

Yet, our approach has the main limitation, that the client devices need to have a browser and an SSH client installed. However, this is always the case in the Bioinformatics use case as well as in other areas.

3.3. Multi-cloud logging, auditing, and monitoring

Multi-cloud environments provide quite challenging environments for logging, auditing, and monitoring. Especially when applications span multiple clouds, there are several requirements regarding these functions:

- **Common mapping for heterogeneous sources.** In multi-cloud environments, log messages can originate from a diverse set of systems and services. Thus, the logging system needs to provide means to map those messages onto a common set of attributes.
- **Flexible deployment for manifold topologies.** Cloud applications can be deployed in a diverse range of topologies that impact the performance of logging services. Logs should be sent to the nearest consumer to keep the logging system performing well. Therefore, logging system topologies need to be flexible enough to be in line with application topologies.
- **Keeping log access control in line with application access control.** Log messages can contain sensitive and possibly personal data that needs to be protected. At best, the access control to the logging should reflect the same access control that is applied for the service access.

3.3.1. The CYCLONE approach

Within CYCLONE we make use of the ELK stack, as already explained in Deliverables D4.1 and D4.3 as well as our publications. ELK is especially suited for solving the multi-cloud challenge, as it features many characteristics that address the challenges set in the preceding section, especially:

- **Flexible input and filter plugins.** These provide both a comprehensive interface to many of the services used in CYCLONE, e.g., a Syslog interface for daemon logs, database connectors for common database solutions, as well as generic JSON APIs for application specific log subsystems, e.g., Log4j. Processing those logs using Logstash filters such as the mutate filter[3] are also quite flexible, creating an easy to use platform for mapping heterogeneous data sources onto a common schema.
- **Relaying and aggregation.** As Logstash can send its logs to another Logstash instance, creating a cascaded log system that relays log messages to upstream log servers is easily conceivable. Logstash can also aggregate data so that, for example, multiple instances local to the data center would keep

[3] https://www.elastic.co/guide/en/logstash/current/plugins-filters-mutate.html

the raw logs and at the same time a company-wide instance would collect statistics over all Logstash instances.

- **Flexible access control.** The access control mechanisms are quite flexible and can be defined and customized by the Kibana operators. Static access control rules are quite simple to define. For example, the current access control rule compares the Kibana user's domain with a certain field in the logs. If they match access is granted. Other access control rules, e.g., based on a comparison of tags or user names are equally simple to implement. If cloud users require more flexibility, possibly having distributed access control using externally defined rules, an extension with an XACML PEP sound quite feasible.

3.3.2. Comparison to other approaches

There are simpler approaches for scenarios requiring less functionality, for example, using a remote-capable syslogger, such as rsyslog[4]. This works quite well when applications are homogeneous, e.g., all components support syslog, they have simple topologies, e.g., run all in the same data center, and when they don't require sophisticated access control.

Another approach is using systemd's journald together with systemd-journal-remote [5] that supports collecting logs from other systems using either a simple line-based format or JSON. Journald features a compact on-disk representation as well as a good set of tools for interacting with the service.

However, as the capabilities of both approaches are limited, they require a lot of effort to support multi-cloud scenarios as well as the ELK stack. Yet, both approaches are complementary, as they can also log to Logstash.

Other solutions focus more on the monitoring part, such as Nagios[6], Icinga[7], and Munin[8]. As their main feature, they aggregate data from diverse systems, displaying them in a miscellaneous set of graphs. Most often, SNMP is used to connect to diverse devices, such as routers and printers. However, they cannot be used for logging as they operate on numeric values and states. Using the aggregation functions of Logstash and Elasticsearch and the graphing capabilities of Kibana, similar solutions can be created. It is dependent on the concrete use case and its requirements which solution would be more fitting.

3.4. Attribute-based Access Control

In CYCLONE bioinformatics use cases, our investigation of the need for centralized access rules management is ongoing. A single point of control with multiple enforcement points would simplify the management of permissions in use cases where certain policies apply to multiple deployments or applications.

CYCLONE could potentially employ dynamically provisioned access control infrastructure (DACI) in two modes of operation:

- **Standalone**: A single VM deployment of XACML PDP is accessed/used by different applications. The authorization mechanism in this mode is comparable to PAM module where local users are evaluated according to their credentials when accessing local resources on the same VM. The PDP engine is

[4] http://www.rsyslog.com/storing-and-forwarding-remote-messages/
[5] https://www.freedesktop.org/software/systemd/man/systemd-journal-remote.html
[6] https://www.nagios.org/
[7] https://www.icinga.com/
[8] http://munin-monitoring.org/

executed as a Java library (i.e. jar) together with the application which may be in turn part of larger application or pipeline (as in the cases of bioinformatics workflows).

- **Distributed**: Shared REST-based Service for authorizations. This mode is useful in settings where the security services are consumed in a distributed manner among multiple VM nodes or multiple disparate applications on the same VM.

Currently, there are two XACML implementations that are integrated and provided with example service consumer clients in CYCLONE. The first one, SNE-XACML [9], is a PDP implementation with basic functionalities for PAP and PEP. It is optimized in terms of performance and well integrated with the rest of the components in DACI. However, it is less complete in terms of XACML specification coverage. The second one is ATT-XACML [10] which has better coverage of the XACML standard and provides REST interfaces for XACML component functionality but is not optimized for performance.

For the second mode of operation of DACI, we have the following interfaces for each component of DACI:

PDP Service: The authorization service follows the guidelines from XACML REST profile [11] with certain adaptations. It has the following interfaces:

Method	Resource	Description
GET	…/pdps/{tenantId}	Instantiation of a Policy Decision Point (PDP) for the selected tenant. It may also be used to return the entry point if the entry point needs not to have fixed entry point pattern.
POST	…/pdps/{tenantId}/pdp	Evaluation of a request against the policy. It returns a Response object.

Table 1 PDP Service Interfaces

Inline with the REST profile, our authorization service assumes that the authorization requests and responses are either [XACMLMedia] types as given in [12] or JSON objects following [13].

Tenant Management Service: The tenant management service provides various functionality including tenant management (addition/deletion) and policy administration point (PAP) of XACML. The interfaces of this service are provided in Table 2.

Method	Resource	Description
POST	…/tenants/{tenantId}	Creates a new tenant.
GET	…/tenants/{tenantId}	Returns the information of the requested tenant
POST	…/pdps/{tenantId}/policies/{policyId}	Stores a new policy for the selected tenant
GET	…/pdps/{tenantId}/policies/{policyId}	Returns the specified policy of the given tenant

Table 2 Tenant Management Service Interfaces

Token Service: The token service acts like a certificate authority that issues and verifies tokens of certain formats. The tokens managed by the service can be used for authorization services and in CYCLONE, we mainly support two types of certificates; X509 certificates and a simple attribute/value pair. Table 3 presents the interfaces for token service.

Method	Resource	Description
POST	…/tokens/{tenantId}	Issues a new token for the specified tenant
GET	…/tokens	Checks the validity of the provided token

Table 3 Token Service Interfaces

Context Management Service: The trust relations between tenants through sharing policies are managed through this service. It is a fundamental service for the proper function PDP service when employed in multi-tenant access control mode, i.e. not only as a decision point but also a point of checking delegations. Table 4 presents the interfaces of context management service.

Method	Resource	Description
GET	…/contexts	Returns a context response according to the provided context request

Table 4 Context Management Service

3.4.1. Integration and Usability Considerations

We plan to integrate DACI services as Slipstream components that can be chosen during application deployment. With the integration, we assume that the applications consuming DACI authorization service implement the Policy Enforcement Point (PEP) of XACML that receives an access request from the application pipeline, creates a DACI request and forwards to Policy Decision Point (PDP) for evaluation. For CYCLONE defined usecases, we will provide a generic PEP implementation that can be adapted/integrated to by usecase owners.

3.5. Managing the Lifecycle of Security Services

In order to ensure that the security services mentioned in Section 3.4 are developed in good quality and remain in good state, we employ a simplified version of the security services lifecycle management (SSLM) approach presented in [14]. SSLM models the stages of a security service lifecycle from provisioning to decommissioning, and allows for systematic development and management. The lifecycle of each service instance is identified by a session id.

In SSLM, there are the following stages:

- **Service Request:** In the current deployment of security services in Slipstream, the service request starts with the deployment request of the application which employs security services as components. This stage is used to bind all other stages and related security context. The Request stage may optionally include SLA negotiation which will become a part of the binding agreement to start on-demand service provisioning.
- **Deployment stage** begins in parallel to the deployment of the application. The security services are deployed to resources preconfigured in automation tool scripts according to user's cloud settings and are made accessible with the relevant API interfaces (e.g. REST).
- During **Operation stage** the security services provide security functionality (e.g. authorizations) to the deployed applications/services and maintain the access or usage sessions for the user that deployed the application over the cloud.
- **Decommissioning stage** ensures that all sessions are terminated, data are cleaned up and session security context is recycled.

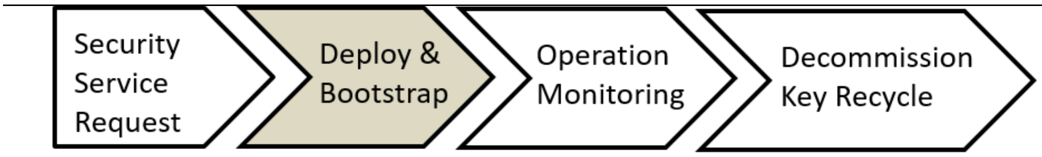

Figure 4. Security Services Lifecycle Management model

In the original SSLM model, there are two additional stages: Reservation stage and Registration&Synchronisation stage. The former corresponds to specification of deployment recipes in the multi-cloud setting. The services can be deployed over the same or different (multiple) provider resources and thus the binding of the parameters (e.g. Slipstream's "ss-get" and "ss-set" parameters) are handled by the cloud automation tool. The latter refers to possible scenarios with the provisioned security services migration or failover. These scenarios are also addressed by the cloud automation tool and do not need to be considered in the service lifecycle.

3.6. Bootstrapping Trust in Federated Clouds

Trust bootstrapping refers to initialization of cloud nodes with relevant secrets. This functionality and service has been researched in the previous authors work [15]. CYCLONE employs keylime [16] for bootstrapping trust within cloud nodes and the services running on them. While most bioinformatics use cases are deployed over trusted clouds, CYCLONE software stack can be used to automate deployments over untrusted public clouds. keylime provides the necessary means to ensure that the cloud resources remain in good state during the computation (through integrity monitoring) and to pass secrets (e.g. keys, tokens) from tenants to his/her nodes in a secure way.

Most trust bootstrapping approaches including keylime rely on Trusted Computing Group's Trusted Platform Module (TPM). TPM is a dedicated cryptographic processor that allows key generation, integrity checks, disk encryption and similar services over a platform. Integrity measures (aka *attestations*) obtained from the operating system, applications and alike of the platform are hashed in PCRs (Platform Configuration Register) in an incremental manner with TPM_Extend() operation. The TPM can then be queried by a verifier for these PCRs to check the integrity of the platform element (e.g. applications) through TPM_Quote() operation. The TPM_Quote() operation results with a quote (report on integrity). TPM uses the endorsement key (EK), a unique asymmetric key generated/assigned by the manufacturer, to sign attestation identity key (AIK) which is used to sign the quotes.

In order to bootstrap cloud nodes with security keys and initialize them for integrity monitoring, the tenants rely on a service called Cloud Verifier (CV) [17] that acts as an intermediary between tenants and their nodes. CV is mainly responsible for periodically checking the integrity of resources and it can live in either tenant's or cloud provider's premises. There are three steps involved in keylime to establish trust:

Key Generation: The tenant creates a fresh symmetric key K_t for each new node it wants to request. The initialization data *d* for the node is encrypted with this key (Enc_{K_t}(d)). The tenant then performs secret sharing over K_t as follows: (1) It creates a random value *V* that has the same size with K_t, (2) It computes $U = K_t \oplus V$. Here *U* is shared with the cloud node and *V* is shared with the CV. If the CV can successfully verify the cloud node for integrity, then it passes *V* to cloud node as well.

Node initiation: The cloud provider instantiates a new VM for the tenant with the information (Enc_{K_t}(d)) the tenant sent. The data *d* can be initialization metadata such as *cloud-init* script. The provider in return sends unique resource id (uuid) and IP address of the node to tenant. After receiving these information from the provider, the tenant contacts to CV with ⟨uuid, V, IP, port, policy⟩ where policy is a TPM policy that defines the relevant register information (i.e. PCR). Note here that the communication between the tenant and CV is done through a secure channel such as pre-established TLS.

Key Derivation Protocol: The final step involves the communications with the cloud node. The first communication is between CV and the cloud node where CV asks for a quote. CV first sends a nonce and the PCR values it is interested in. The cloud node responds with a quote that is signed by the AIK available to it. As can be noted from the key generation step, the successful validation of the quote is used to share *V* (in encrypted form) with the cloud node. The second communication is performed between the tenant and cloud node to exchange *U*. The tenant first sends a fresh nonce to cloud node asking for a quote. If the cloud node's quote is successfully verified then the tenant shares *U* (in encrypted form) along with a hash of node's ID so that the cloud node can also verify the tenant. Note here that the keys *U* and *V* are shared in encrypted form over an untrusted network. This means the communicating parties have a pre-established secret. This is achieved through ephemeral keys (NK) through deep quotes to TPM. More specifically, a virtual PCR (PCR #16) of cloud node's TPM is used to store and bind asymmetric keys between the communicator and the cloud node.

Error! Reference source not found. presents the protocol that underlies keylime's key derivation.

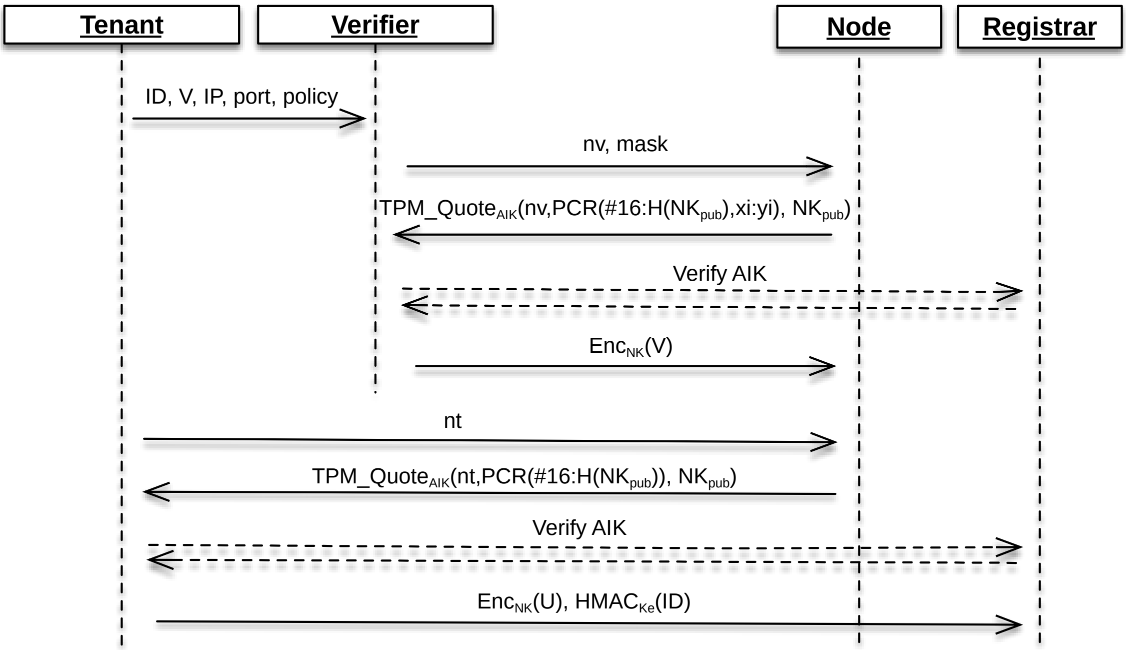
Tenant
Verifier
Node
Registrar
ID, V, IP, port, policy
nv, mask
TPM_Quote$_{AIK}$(nv,PCR(#16:H(NK$_{pub}$),xi:yi), NK$_{pub}$)
Verify AIK
Enc$_{NK}$(V)
nt
TPM_Quote$_{AIK}$(nt,PCR(#16:H(NK$_{pub}$)), NK$_{pub}$)
Verify AIK
Enc$_{NK}$(U), HMAC$_{Ke}$(ID)

Figure 5. Bootstrapping Keys in keylime (courtesy of [16])

3.6.1. Usability and Integration Considerations

In CYCLONE, keylime can be used for bootstrapping nodes with keys over untrusted clouds. However certain adaptations are necessary in order to employ its architectural model. In particular, the prototype of keylime has been implemented over Xen hypervisor's TPM features. A service that provides interaction (e.g. activation and virtual TPM creation/association) with the hardware TPM is executed in cloud provider premises. Adoption of keylime in CYCLONE testbed may require a re-implementation of this service for the existing hypervisors such as KVM in CYCLONE.

3.7. Multi-cloud Network Services

WP5 Deliverable D5.2 presents several network services that also relate to multi-cloud security: the CNSMO Firewalling and VPN services. Please see D5.2 for the comprehensive description of the preliminary version of these services including both development and deployment.

Clearly neither VPN nor firewalling are *new* services – there are many implementations at different levels available. However, the services are not included for their novelty but for the very easy **integration of multi-cloud networking into application deployments**.

The integration of the VPN service with the Deployment Manager (SlipStream) is a highly novel aspect. The **C**YCLONE **N**etworking **S**ervices **M**anager and **O**rchestrator (**CNSMO**) is included as part of the SlipStream application recipe, deploying network services onto the VMs, configuring them and setting them up at runtime. The integration of these services with the regular SlipStream deployment procedure - without impacting the SlipStream software or processes - entails benefits for both the application service providers, getting additional services for free, as well as the application deployment managers, who can extend their offered services by these network and security services in their market places.

The multi-cloud aspect is also remarkable. The services are being deployed over multiple clouds at the application level in a homogeneous way, avoiding the potential limitations (e.g., incompatibilities) of working with several cloud infrastructure providers. These services are deployed on any cloud and enable, for example, creating new VPN clients in case of applications scaling up.

Thus, the value of these services lies in the capability to be deployed together with complex cloud applications by integrating CSNMO with SlipStream in distributed, multi-cloud environments. This is one of the innovative aspects that CYCLONE delivers to enhance the secure provisioning of complex applications in federated clouds.

3.8. Multi-cloud Deployment

SlipStream leaves most of the detailed security decisions/enforcement to the application layer. The exception may be seen as the integration of 'firewall' rules from the defined application topology, this topic will be described in detail in WP5/6 deliverables. Among new features to be developed from which the ones related to identity management are:

- allowing SlipStream to act as an IdP (probably OpenID),
- continuing to enhance the user management capabilities around groups and roles,
- enhancing the handling of ACLs for resources (particularly to facilitate secure sharing of cloud credentials).

4. Evaluation of Security Components in Testbed Deployments

This section shows how we deployed the components to the testbeds and use them in the use cases.

4.1. IFB Use of Federated Identity in Testbed Deployment (IFB)

With the deployment on the testbed, we gained insights on further requirements both on technical and user related issues. The first one is the ports. In IFB Cloud and OpenStack@LAL, the firewall allows only one or two ports to external access. At other target clouds, there might be other ports available. Therefore, we extended the implementation with the ability to load an array of ports from a configuration file and start the web server at one of those. The second problem was the compatibility of the Python PAM to system dependencies.

4.2. Multi-cloud network services

The preliminary versions of the network services have been implemented, integrated within CNSMO, deployed together with SlipStream, and successfully tested over multiple clouds. Later versions of these services will eventually incorporate additional features to support, for instance, the scaling of applications. The evaluation and demonstration of the VPN and firewalling services has been recorded on video and is available on YouTube:

- VPN service: https://www.youtube.com/watch?v=34nqiouZly4
- Firewalling service: https://www.youtube.com/watch?v=e7JcFwJz-bA

The Figure below shows the deployment workflow on Exoscale and OpenStack clouds as it has been validated.

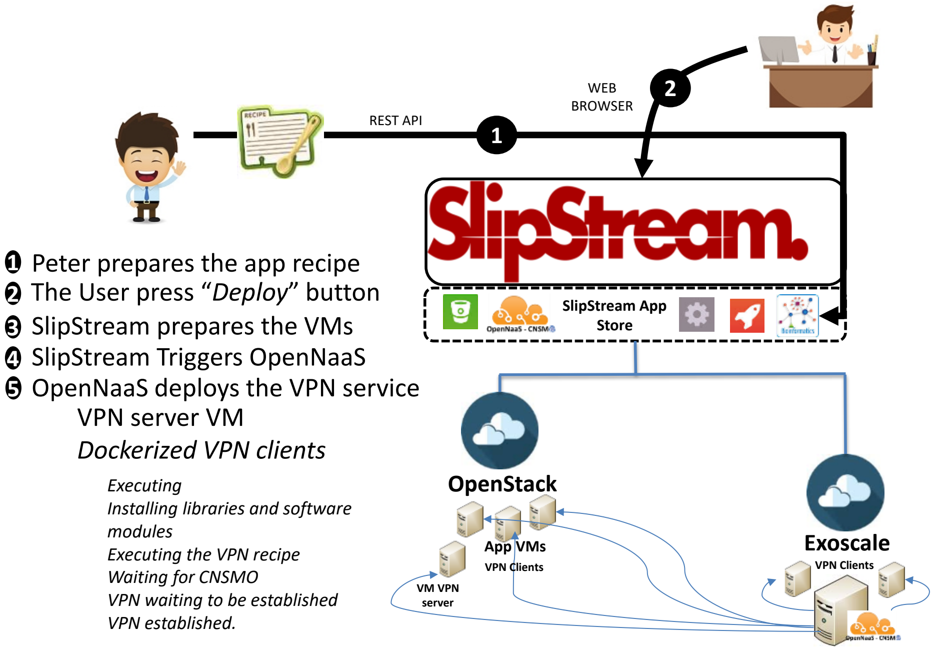
WEB BROWSER
2
REST API
1
RECIPE
SlipStream
SlipStream App Store
OpenNaaS - CNSMO
Bioinformatics
❶ Peter prepares the app recipe
❷ The User press "*Deploy*" button
❸ SlipStream prepares the VMs
❹ SlipStream Triggers OpenNaaS
❺ OpenNaaS deploys the VPN service
VPN server VM
Dockerized VPN clients
Executing
Installing libraries and software modules
Executing the VPN recipe
Waiting for CNSMO
VPN waiting to be established
VPN established.
OpenStack
App VMs
VPN Clients
VM VPN server
Exoscale
VPN Clients
OpenNaaS - CNSMO

Figure 6: VPN service multi-cloud deployment workflow

5. Summary & Outlook

This deliverable described our recent concepts related to cloud security which the WP4 was focused in Y2 on, compared these concepts to selected alternative solutions with focus on CYCLONE use cases. The document summarised the result of non-exhaustive survey of the possible alternatives for federated identify management and access control at the level applications and cloud services and suggested further investigation of the Kerberos' Domain Trust mechanisms for multi-cloud CYCLONE use cases.

The report provides suggestions for further extension of the basic use cases and security services to address needs for multi-cloud application platform and corresponding security services. The document refers to specific use cases requirements that indicate need for multi-cloud applications platforms that will require corresponding multi-cloud security services. The document presents analysis of the potential multi-cloud use cases that include a general multi-cloud and Intercloud use case and the proposed bioinformatics use case extension that identifies necessary functional application infrastructure and security components that would allow using distributed cloud based resources and data sets, including possible application workflow migration or outsourcing to external cloud. The presented analysis confirm benefits of consistent implementation of the federated multi-cloud security model that can be potentially integrated with the currently widely adopted by the major cloud service providers the federated access control and federated identify management model.

The deliverable provides overview of the security services that can be used in multi-cloud applications such as currently implemented federated identify management using eduGAIN, secure shell login using eduGAIN federated identities, and new services being developed such as multi-domain Attribute Based Access Control, security services lifecycle management and trust bootstrapping for virtualised cloud environment.

In Y3, we will investigate the use cases further for multi cloud implementation and corresponding requirements.

References

[1] CYCLONE Deliverable D4.1: Security Infrastructure Specification and initial Implementation, CYCLONE Project, December 2015

[2] CYCLONE Deliverable D4.3: CYCLONE Secure Action and Resource Models, CYCLONE Project, December 2015

[3] CYCLONE Deliverable D3.1: Evaluation of Use Cases, CYCLONE Project, October 2015

[4] CYCLONE Deliverable D4.4: Consolidated CYCLONE secure action and resource models, December 2016

[5] Todorov, D. & Ozkan, Y. 'AWS security best practices', Amazon Web Services [Online]. Available from: http://media.amazonwebservices.com/AWS_Security_Best_Practices.pdf

[6] Realising the European Open Science Cloud. First report and recommendations of the Commission High Level Expert Group on the European Open Science Cloud, November 2016 [online] https://ec.europa.eu/research/openscience/pdf/realising_the_european_open_science_cloud_2016.pdf

[7] Demchenko, Y., M. Makkes, R.Strijkers, C.Ngo, C. de Laat, Intercloud Architecture Framework for Heterogeneous Multi-Provider Cloud based Infrastructure Services Provisioning, The International Journal of Next-Generation Computing (IJNGC), Volume 4, Issue 2, July 2013

[8] Y.Demchenko, C. Lee, C.Ngo, C. de Laat, Federated Access Control in Heterogeneous Intercloud Environment: Basic Models and Architecture Patterns. In Proc IEEE International Conference on Cloud Engineering (IC2E), March 11, 2014, Boston, USA

[9] Nabil Schear, Patrick T. Cable II, Thomas M. Moyer, Bryan Richard, Robert Rudd, "Bootstrapping and Maintaining Trus[5t in the Cloud", Annual Computer Security Applications Conference (ACSAC), 2016

[10] Joshua Schiffman, Yuqiong Sun, Hayawardh Vijayakumar, Trent Jaeger: Cloud Verifier: Verifiable Auditing Service for IaaS Clouds. SERVICES 2013: 239-246

[11] XACML REST Profile, http://docs.oasis-open.org/xacml/xacml-rest/v1.0/cs02/xacml-rest-v1.0-cs02.pdf, last accessed on 12th December 2016.

[12] eXtensible Access Control Markup Language (XACML) Media Type. November 2013. IETF RFC 7061. http://tools.ietf.org/html/rfc7061.

[13] JSON Profile of XACML 3.0 Version 1.0, December 2014, OASIS Committee Specification, http://docs.oasis-open.org/xacml/xacml-json-http/v1.0/xacml-json-http-v1.0.html

[14] Canh Ngo, Yuri Demchenko, Cees de Laat: Decision Diagrams for XACML Policy Evaluation and Management. Computers & Security 49: 1-16 (2015)

[15] ATT XACML Implementation, https://github.com/att/XACML

[16] Yuri Demchenko, D.R. Lopez, J.A. Garcia Espin, Cees de Laat, "Security Services Lifecycle Management in On-Demand Infrastructure Services Provisioning", International Workshop on Cloud Privacy, Security, Risk and Trust (CPSRT 2010), 2nd IEEE International Conference on Cloud Computing Technology and Science (CloudCom2010), 30 November - 3 December 2010, Indianapolis, USA.

[17] Canh Ngo, Peter Membrey, Yuri Demchenko, Cees de Laat: Policy and Context Management in Dynamically Provisioned Access Control Service for Virtualized Cloud Infrastructures. ARES 2012: 343-349

6. Abbreviations and Definitions

6.1. Definitions

Term	Definition

6.2. Abbreviations

B2B	Business to Business
CSP	Cloud Service Provider
DC	Data Center
E2E	End to End
GSSAPI	Generic Security Services API
IaaS	Infrastructure-as-a-Service
ICAF	Intercloud Architecture Framework
ICFF	Intercloud Federation Framework
IPR	Intellectual Property Rights
IT	Information Technology
MaaS	Metal as a Service
NaaS	Network-as-a-Service
Net-HAL	Network Hardware Abstraction Layer
NFV	Network Function Virtualization
PaaS	Platform-as-a-Service
PC	Project Coordinator
PMB	Project Management Board
PoP	Point of Presence
SaaS	Software-as-a-Service
SCI	Smart Core Interworks
SDN	Software Defined Networks
SP	Service Provider
TC	Technical Coordinator

TCTP	Trusted Cloud Transfer Protocol
TMB	Technical Management Board
WP	Work Package
WPL	Work Package Leader

<END OF DOCUMENT>