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 D4.3

CYCLONE Secure Action and Resource Models

Due date:	31/12/2015
Submission date:	18/12/2015
Deliverable leader:	TU Berlin (TUB)
Editors list:	Zilci, Ilke (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	R. Branchat, C. Loomis
QSC AG	QSC	
Technische Universitaet Berlin	TUB	I. Zilci, M. Slawik
Fundacio Privada I2CAT, Internet I Innovacio Digital A Catalunya	I2CAT	J. Aznar, I. Canyameres
Universiteit Van Amsterdam	UVA	Y. Demchenko
Centre National De La Recherche Scientifique	CNRS	O. Lodygensky

Change history

Version	Date	Partners	Description/Comments
0.1	18.11.2015	TUB	Layout the general structure to start discussion on contents with partners
0.2	19.11.2015	TUB	Rearrange the structure and chapter introductions after discussions
0.3	08.12.2015	TUB	Add glossary and describe TUB security extensions
0.4	11.12.2015	SIXSQ	Added description of SlipStream security extensions
0.5	14.12.2015	I2CAT, LAL	Added description of OpenNaaS and Stratuslab security extensions
0.6	16.12.2015	SIXSQ	Internal review
1.0	17.12.2015	TUB, SIXSQ	Final edit and review
2.0	22.12.2015	TUB	Final version ready

Table of Contents

Figures Summary

Executive Summary

This document serves as a reference for the current state of the extended CYCLONE security APIs within the CYCLONE security infrastructure.

It starts with a glossary with general and security specific definitions. The glossary aims to set a basis for a common vocabulary while discussing security in multi-cloud environments. The CYCLONE use cases establish essential security requirements on the CYCLONE APIs. The main goal of the APIs is to make CYCLONE compatible to the existing use case environments. Moreover, CYCLONE aims to ease the integration with prospective multi-cloud application users and service providers. The descriptions of the security endpoints and the interactions between the components focus on the current implementation of the authentication mechanisms. The APIs implement current standards for service provider APIs and the established best-practice for the CYCLONE multi cloud application users.

1. Introduction

The CYCLONE security infrastructure aims to provide holistic security functionality in multi-cloud deployments. This implies that the CYCLONE components should carry out security operations as transparently as possible to the stakeholders outside the CYCLONE system. Ideally, end-users log into CYCLONE with an existing account, their organizations continue using their existing identity providers and other security mechanisms. Moreover, cloud service providers can offer their services over CYCLONE without the need to change their security mechanisms. The diversity of IT infrastructures on both sides makes it a challenging task for CYCLONE to meet their specific requirements.

To achieve the integration and the needed compatibility, CYCLONE identifies four actions:

1. Develop a common vocabulary to discuss security in multi-cloud environments,
2. Adjust or extend existing CYCLONE components security APIs to build the CYCLONE system,
3. Follow established standards in practical CYCLONE use cases, and
4. Investigate forthcoming extensions.

This document describes the current state of the execution of the actions listed above. First, it provides the initial CYCLONE Glossary. Following this, it explains all CYCLONE security extensions APIs and the security protocols implemented focusing on two components at a time. It summarizes the open questions in the final chapter.

This deliverable is strongly linked to D3.1 and D4.1 which explain more on its context. The relation of the security extensions to the use cases is explained in greater detail in Deliverable 4.1 [D4.1]. The use cases are described in greater detail in Deliverable 3.1 [D3.1].

2. CYCLONE Glossary

This glossary aims to describe the roles and terms in multi-cloud application management platforms in general and the entities in CYCLONE specifically in a unified and consistent way. CYCLONE aims to develop a vocabulary for discussing security in multi-cloud environments.

2.1. General Terms

Cloud Computing	Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models. (NIST SP 800-145)
Cloud service	A service offered using the cloud computing model.
Cloud deployment models (NIST)	There are four distinct cloud deployment models according to NIST: *Private cloud.* The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises. *Community cloud.* The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises. *Public cloud.* The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider. *Hybrid cloud.* The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or

	proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).
Intercloud	The general model for a cloud infrastructure that combines multiple heterogeneous clouds from multiple providers and typically also includes campus/enterprise infrastructure and non-cloud resources. Intercloud model may use federated cloud model or implement more specific common control and management functions to create a kind of Intercloud virtual private cloud.
Federated cloud	The cloud infrastructure that involves multiple heterogeneous clouds from different providers that use a federation mechanism to share, access and control combined infrastructure and services. Federated cloud typically combines multiple private clouds and may include also private cloud. Federation members remain independent however having common policy in resources sharing and access control, including federated identity management. Cloud federation may include provider side federation and customer side federation. Community clouds will most probably adopt the federated cloud model.
Cloud federation	A cloud federation consists of cloud resources which can be consumed in a collective fashion by cloud end-users coming from different organizations. There are a lot of examples in the academic sector, for example, institutions offering resources (e.g. VMs) for a diverse set of participating research institutions. The “rendez-vous” service which CYCLONE uses to conduct Tutorials is another example of a federated service: It is operated by Renater and offered collectively to all participating research institutions. Cloud federation is realized by different mechanisms, such as federated identity (e.g. OpenID, Shibboleth) as well as delegated resource access (e.g. OAuth).
Multi-cloud deployment	A cloud service deployment which spans multiple clouds. An example would be a web application using resources provisioned on both EC2 and Azure.
DevOps	DevOps (development and operations) is an enterprise software development phrase used to mean a type of agile relationship between Development and IT Operations. The goal of DevOps is to change and improve the relationship by advocating better communication and collaboration between the two business units.

2.2. Security Terms

Single sign-on (SSO)	Single sign-on (SSO) is a property of access control of multiple related, but independent software systems. With this property a user logs in once and gains access to all systems without being prompted to log in again at each of them.
Federated identity management (FIDM)	Federated identity management (FIDM) is an arrangement that can be made among multiple enterprises that lets subscribers use the same identification data to obtain access to the networks of all enterprises in the group. The use of such a system is sometimes called identity federation.
X.509 Certificate	Security Certificate format, typically referred to identity certificate used for authentication. Relates to Public Key Infrastructure (PKI)
Attribute Certificate	A digital certificate that binds a set of descriptive data items, other than a public key, either directly to a subject name or to the identifier of another certificate that is a public-key certificate.
Access Control List (ACL)	A mechanism that implements access control for a system resource by enumerating the identities of the system entities that are permitted to access the resource.
Authority **Attribute Authority**	An entity, responsible for the issuance of certificates or security assertions. More specificly, a Security Token Service is an issuing and validating authority for security tokens.
Certificate Authority (CA)	An entity that issues digital certificates (especially X.509 certificates) and vouches for the binding between the data items in a certificate.
Credential(s)	Data that is transferred or presented to establish either a claimed identity or the authorizations of a system entity.
Federated authentication	Federated authentication is using the user identity from the home organization either by re-direct authentication to the home organization authentication service or using one of federated identity providers to obtain your user identity or authentication assertion.
Role-Based Access Control (RBAC)	A form of identity-based access control where the system entities that are identified and controlled are functional positions in an organization or process.
Assertion **SAML Assertion**	An assertion is a package of information that supplies one or more statements made by a SAML authority. SAML defines three different

	kinds of assertion statements that can be created by a SAML authority.
SAML	Security Assertion Markup Language (SAML) is an OASIS standard that defines security assertions format, protocol, bindings, profiles and metadata.
XACML	eXtensible Access Control Markup Language (XACML) is an XML based policy language.
Delegation **Delegation of Authority (DoA)**	Delegation is the process of a computer user granting some (or all) of their access rights to another user or service. In Role-Based Access Control models, delegation of authority involves delegating roles that users can assume or the set of permissions that they can acquire, to other users.
Shibboleth	Shibboleth is federated identity-based authentication and authorization infrastructure based on Security Assertion Markup Language (SAML).
Attribute namespace	Attribute namespaces are a naming convention for attributes, in particular used for security purposes: authentication and authorization. Namespace expression depends on the attributes format and can use one of the forms: XML URN, URI/URL, or ASN.1. Namespace definitions should ensure that attributes are unique in the context of the administrative domain.
Access Manager	The Access Manager grants authorized end-users the right to use a service, while preventing access to non-authorized end-users.

2.3. Cloud roles

Cloud (Service) Provider	A cloud provider is a person or an organization; it is the entity responsible for making a service available to interested parties. A Cloud Provider acquires and manages the computing infrastructure required for providing the services, runs the cloud software that provides the services, and makes arrangement to deliver the cloud services to the Cloud Consumers through network access. A cloud provider offers cloud services to cloud consumers. They either rely on other providers for the needed infrastructure (e.g., RedHat OpenShift PaaS relies on Amazon IaaS), or provide the infrastructure themselves (e.g., Google and Amazon).
IaaS Provider **PaaS Provider** **SaaS Provider**	Cloud providers can be differentiated by the cloud service model they use, e.g., IaaS (Infrastructure-as-a-Service, e.g., VMs), PaaS (Platform-as-a-Service, e.g., an environment for Docker containers), and SaaS (Software-as-a-Service, e.g., CRM solutions such as Salesforce Sales Cloud).

Infrastructure Provider	An infrastructure provider provides a “non-cloud” infrastructure for cloud providers and other consumers. For example, a data center operator provides colocation and servers to cloud providers and other companies. It can rely on other infrastructure providers, e.g., a global network provider for connectivity.
Cloud Application Developer	The Cloud Application Developer implements and maintains applications which provide required business functionality. They can be custom applications, customized vendor software, or open source solutions. Some developers are DevOps, which means they also fulfill the role of cloud operators.
Cloud Carrier	An intermediary that provides connectivity and transport of cloud services from Cloud Providers to Cloud Consumers. A typical role for telecom provider.
Cloud Application Operator	Cloud Operators carry out operational and maintenance activities of cloud solutions, e.g. deployment tasks. These operations can be highly automated, e.g., the deployment of cloud solutions is highly automated to allow demand-oriented scaling of applications. Due to this automation, many cloud application developers are becoming cloud operators, who are then called DevOps.
Cloud Consumer	The term “cloud consumer” refers to both the end-users (e.g., biomedical users, medical practitioners) and their respective organizations.
Cloud Application User	The person who interacts with a cloud based application in order to use it.
Cloud Broker	A cloud broker is an entity that manages the use, performance and delivery of cloud services and negotiates relationships between cloud providers and cloud consumers. Cloud broker may also include developers of integration functions.

3. Overview of CYCLONE Security APIs

This section explains the security interactions between CYCLONE components in the Component Model Diagram in D4.1 “Security infrastructure specification and initial implementation” Chapter 3.1. [D4.1]. It is not a full API reference, includes only the methods relevant for the security integration and links to the descriptions of protocols which CYCLONE components implement.

3.1. CYCLONE Federation Provider Security APIs

3.1.1. eduGAIN

eduGAIN is a service for interfederating educational identity providers [EduGAIN]. Currently TUB via DFNAAI and CNRS via France - Fédération Éducation-Recherche are connected to eduGAIN.

As shown in the Component Model in the link between SAML Proxy and eduGAIN, SAML Proxy issues a GET request to DFNAAI to get the Identity Providers Metadata:

```
GET https://www.aai.dfn.de/fileadmin/metadata/DFN-AAI-Basic-metadata.xml
```

```
GET https://www.aai.dfn.de/fileadmin/metadata/DFN-AAI-eduGAIN+idp-metadata.xml
```

As shown in the Component Model in the link between Shibboleth IP and eduGAIN, the educational Identity Provider, in this example tubIT (TU Berlin's Identity Provider that uses Shibboleth), issues a GET request to DFNAAI to get the Service Providers Metadata:

```
GET https://www.aai.dfn.de/fileadmin/metadata/DFN-AAI-eduGAIN+sp-metadata.xml
```

3.1.2. SAML Proxy

SAML Proxy (SimpleSAMLphp) is registered at eduGAIN as a service provider with the URL which provides the metadata:

https://federation.cyclone-project.eu/samlbridge/module.php/saml/sp/metadata.php/cyclone-saml-bridge

There are two advantages of using the SAML Proxy: It fixes the problem that the identity broker assigns one SAML endpoint URL for each registered identity provider. It offers an identity provider discovery service: it parses the IP Metadata list and shows all to the user for selection.

SimpleSAMLphp has no built in support for OpenID Connect Authorization Code Flow (ACF) and JSON Web Tokens (JWTs). Therefore, it does not meet the needs of CYCLONE alone and it is combined with Keycloak.

3.1.3. Identity Broker

The identity broker (Keycloak) implements the Open ID Connect ACF to allow service providers which participate in the CYCLONE federation accept authentication via eduGAIN. Moreover, it creates JWTs using the SAML assertions. JWTs are substantially less complex than SAML Responses. We expect that this simplifies the integration for service providers.

It can be accessed at:

`https://federation.cyclone-project.eu/auth`.

Deliverable 7.2 “Overlay with focus on component manager”, Section 4.2.3. “Authentication with Keycloak” explains the API for Open ID Connect ACF [D7.2].

3.2. SlipStream

SlipStream is a gateway to multi-cloud deployment automation, remaining independent from the underlying IaaS platform used. In CYCLONE, users with accounts on eduGAIN member identity providers must be able to log in to the SlipStream web interface. On the side interfacing with IaaS providers, the current solution is based on storage of user credentials.

3.2.1. Authentication Service

Until recently, SlipStream has authenticated users based on username/password pairs stored in an internal database. On a successful login request, the SlipStream server returns an authentication cookie (to be included in subsequent requests) that can be used to access SlipStream resources.

The generated cookie is more than a simple “session” identifier; it contains a token that enumerates the user’s identity and roles. Because the token contains this information, all authorization decisions can be done “locally”, avoiding call outs to an external service or accesses to a database for every authorization decision. To avoid various security issues, the token is time-limited, tied to the client’s IP address, and encrypted. The “local” evaluation of the token verifies these constraints before evaluating the Access Control List (ACL) to decide whether to accept or to deny a particular action on a resource.

A simplified view of this process is shown in Figure 1; the initial POST in this case contains the username and password of the user trying to authenticate.

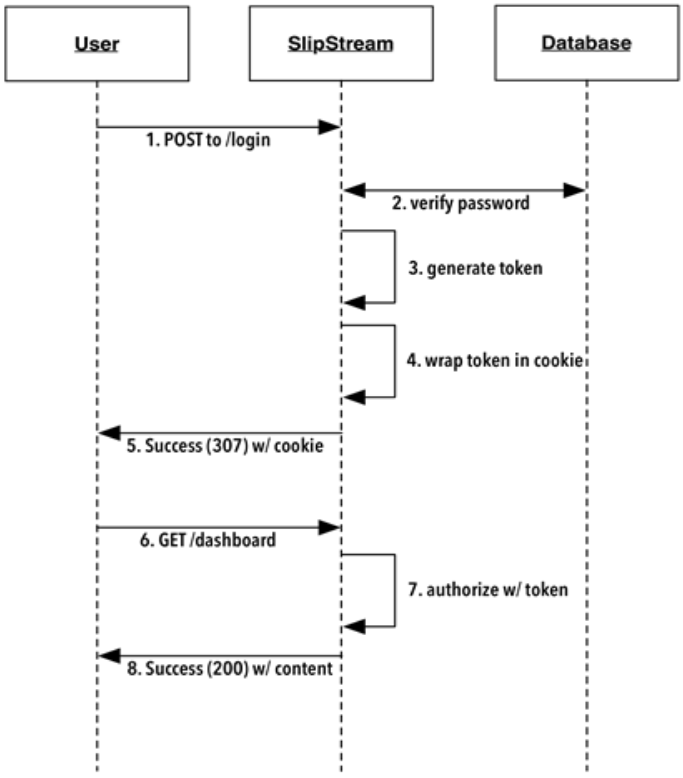

Figure 1 Legacy Authentication with SlipStream

While this method is effective for standalone deployments of SlipStream, it makes integration much more difficult in environments where user authentication (and management) is handled by an external service. As such environments are more the rule rather than the exception, the SlipStream authentication code has been refactored to support this use case. SlipStream now uses a separate, plugin-based service for authentication. This allows flexible support for external authentication mechanisms, in addition to the support for the previous username/password authentication. The service can support multiple authentication methods simultaneously.

Within CYCLONE, the primary objective is to support the federated identity provider developed within the project and based on Keycloak. However, as a first validation of the new authentication server, the developers decided to incorporate the GitHub authentication mechanism into SlipStream. The workflow for the GitHub authentication is shown in Figure 2. This flow implements OAuth 2.0. Authorization Code Flow in which the user logs into the authentication server to define the authorization rules for the access token for the application. In this case, SlipStream is the application and GitHub is the authentication server. SlipStream can access user resources with the access token. As the overall authentication workflow is very similar to that for the CYCLONE federated identity provider, support for that should come quickly after the authentication service has been validated.

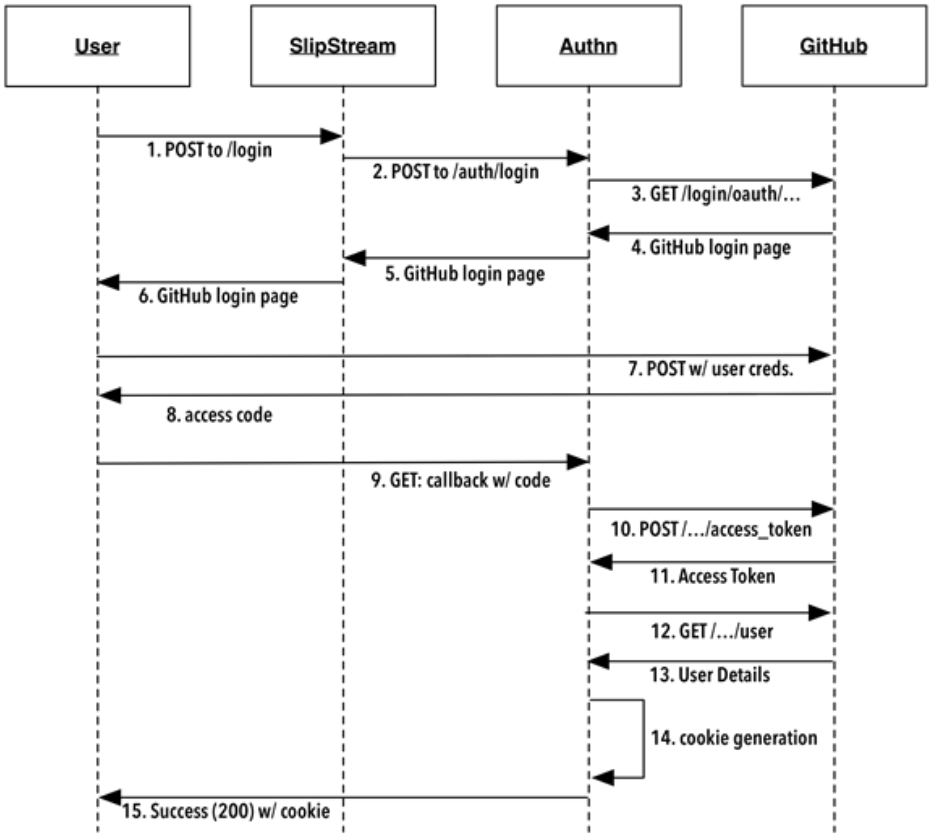

Figure 2 SlipStream using GitHub Authentication

3.2.2. Delegation

In order to manage users' cloud applications, SlipStream must access cloud services on behalf of the users. Because most cloud services do not currently support delegation, this requires that SlipStream must store the users' credentials directly. The only exception at the moment is the EGI Cloud Infrastructure, where SlipStream can use delegated proxy certificates rather than the primary credentials of the user.

Within CYCLONE the goal is to have the IaaS providers, SlipStream, and the cloud application to use a common federated identity provider. This can be achieved without too much difficulty in each of these layers individually. However, using the federated identity within SlipStream to access IaaS providers is a challenging problem. The login workflow for the federated identity provider requires human access to an external "login" page. Consequently, simple storage of the users' primary credentials will not work because SlipStream would not be able to "login" automatically to manage the users' cloud resources. Either a "command line" authentication workflow would need to be supported or a delegation scheme would need to be developed. How to resolve this problem is an open question. However, work on a solution can be postponed for the time being as this is not an impediment to using the CYCLONE platform by the current use cases. Until this problem is solved, users can store their normal IaaS provider credentials in their SlipStream profiles.

3.2.3. SlipStream Login at StratusLab

SlipStream acts as a client to IaaS infrastructures and uses these infrastructures' services and protocols as the end user's representative. It is the end user's responsibility to delegate such empowerment to SlipStream by storing its credentials within SlipStream.

StratusLab, on the other hand is an IaaS middleware permitting to deploy and maintain a cloud platform in order to permit end users to create and use virtualized components such as *virtual machine* and *virtual disk*. In order to use StratusLab facilities, end users must be registered. End user registration is under the platform administrator's responsibility. As soon as the registration is accepted, the end user can start using the platform, presenting its validated credentials.

StratusLab end users must store their credentials within SlipStream to automate IaaS usage. StratusLab makes no difference, and is even not able to see any difference, if the user uses the IaaS directly or through delegation.

3.3. OpenNaaS

3.3.1. North Bound Interface –NBI

OpenNaaS provides a northbound REST + XML API. It offers all the functionality of OpenNaaS through a multi-tenant interface. Any external or internal software module can access to it through the management network interface. It offers security through HTTPS and "HTTP Basic Authentication" (RFC 2617 [RFC2617] and RFC 7235 [RFC 7235]) with configured users and passwords per instance.

3.3.2. South Bound Interface – SBI

OpenNaaS can communicate with network elements through any network protocol. Current implemented protocols include Netconf (RFC 6241) [RFC6241], REST + XML and JSON for many different devices, SOAP for AutoBAHN bandwidth-on-demand Géant provisioning tool or W-onesys for ROADM devices. Each protocol has its own security constraints, for instance in the Netconf implementation OpenNaaS uses SSH as transport based on user and password credentials. Depending on the specific SBI protocol that is used to communicate with the network resource, OpenNaaS will be extended to include such SBI (if needed) including the associated security.

3.3.3. Interface to the ELK Logging Stack

OpenNaaS network service management tool should be able to log its messages to the Elasticsearch [Elasticsearch], Logstash [Logstash] & Kibana [Kibana] (ELK) logging stack, so that the logs can be monitored

from a central point and uniformly managed. This is currently the single identified feature that would be desirable to be integrated within the whole security ecosystem for OpenNaaS.

The logging stack was formerly configured by TUB for TRESOR which involved a single identity provider. For CYCLONE and its federated identity requirements, the configuration scripts are extended to provide each user with only the logs which belong to the user's organization at the web browser based dashboard.

To be able to run the CYCLONE security components, there are no specific network configurations identified yet.

4. CYCLONE Components Security Interactions

This section explains how CYCLONE components and external actors take part in the implementation of the standard protocols SAML 2.0 and OpenID Connect ACF. Section 4.1 focuses on the eduGAIN member identity providers which provide SAML 2.0 Authentication. Section 4.2 describes the CYCLONE Federation Provider and service provider interactions which mostly go over OpenID Connect Authorization Code Flow.

4.1. SAML 2.0 Authentication

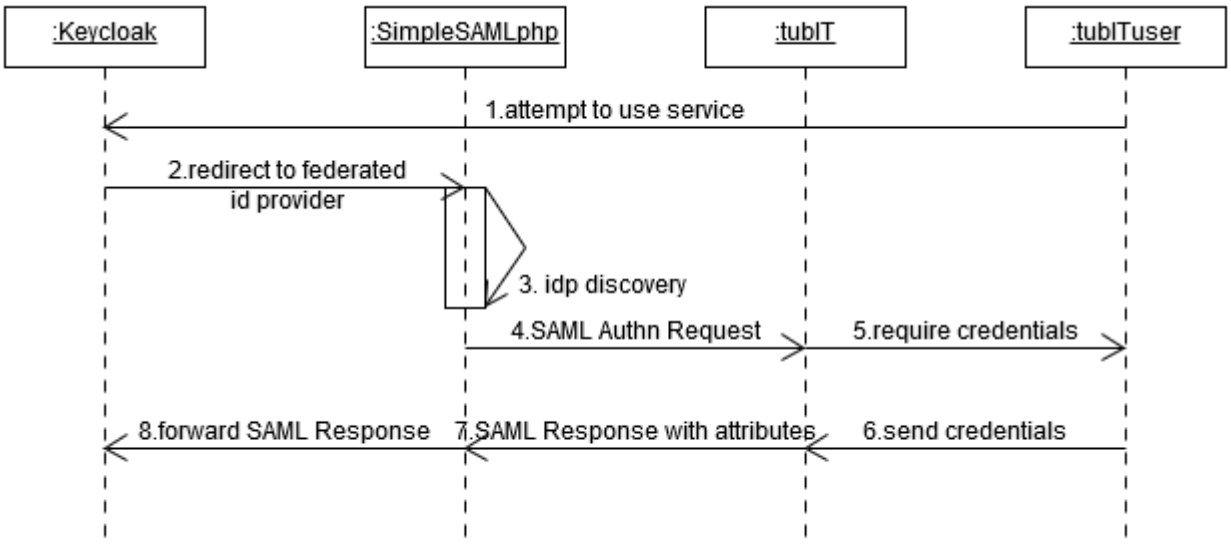

Figure 3 SAML 2.0 Authentication in CYCLONE

When a user attempts to access a service that has integrated the authentication via identity broker, the authentication is executed conform to the standard SAML 2.0 Authentication [SAML2].

Between the SAML Proxy (SimpleSAMLphp) and the educational identity providers participating in the eduGAIN service via their federations, there is trust since both are registered in the corresponding metadata lists.

Upon redirect of a service to SimpleSAMLphp, an Identity Provider Discovery is performed (a cloud application user selects an identity provider from the provided list). SimpleSAMLphp sends a SAML <AuthnRequest> to the selected IdP. The IdP redirects the cloud application user to login. The IdP then returns the results of the authentication in a SAML <Response> to SimpleSAMLphp.

4.2. OpenID Connect Authorization Code Flow (OIDCACF)

In CYCLONE, we have multiple OpenID Connect clients which interact with the identity broker (Keycloak [Keycloak]) following the standard OIDCACF: SlipStream login, IFB bioinformatics cloud, bioinformatics applications, and distributed logging. By "clients" we mean their browser based dashboards. This allows cloud application users to login to applications via browsers and their identities from eduGAIN member identity providers.

Figure 4 shows how the Federation Provider – SlipStream authentication service will implement OpenID Connect ACF. The details on the POST request bodies are described on our GitHub repository for the Federation Provider [FedProGitHub].

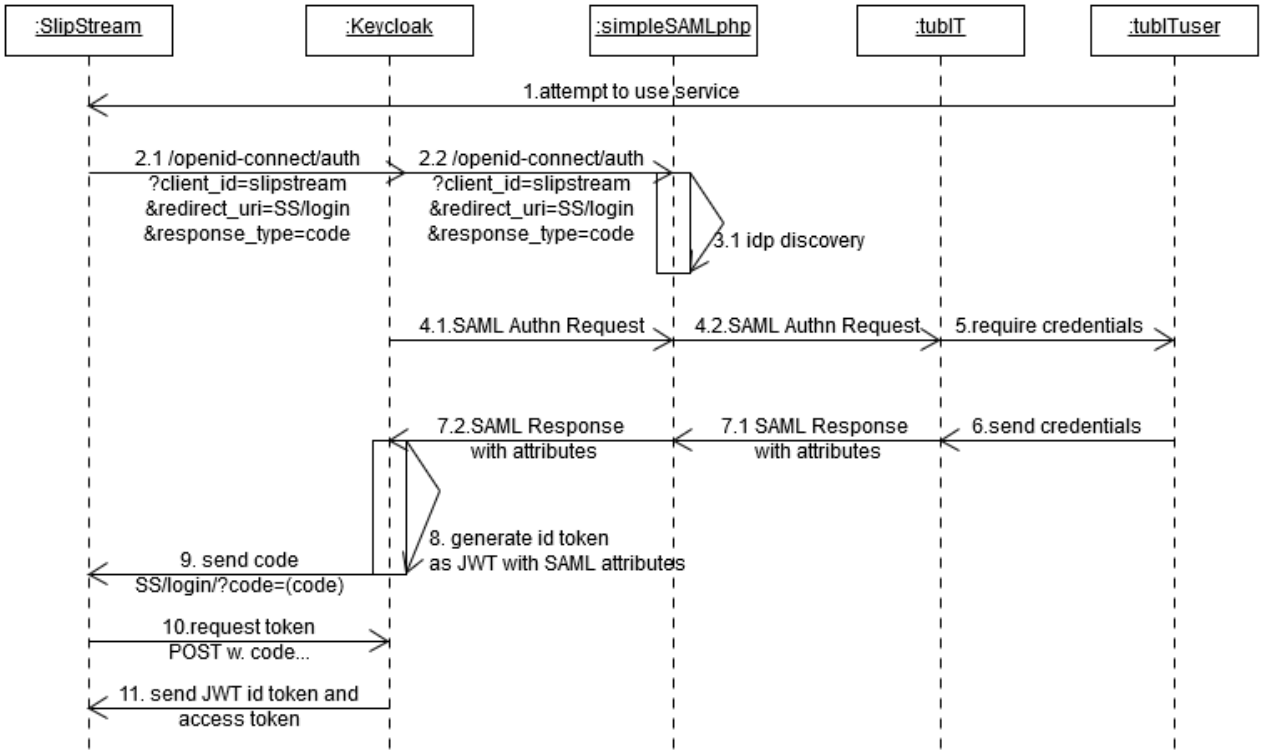

Figure 4 SlipStream OpenID Connect ACF

5. Outlook

This document describes the current state of the tasks that CYCLONE has identified as necessary to achieve a well-integrated and interoperable security infrastructure. First, it provides the initial CYCLONE Glossary. Following this, it explains all CYCLONE security extension APIs and the security protocols implemented focusing on two components at a time.

CYCLONE follows an iterative approach to the implementation of the use cases. Therefore, the current deployed security infrastructure described in this document is the primary source for identifying the following steps. For the integration of the CYCLONE federation provider and eduGAIN member identity providers, we will investigate if there is a need to retrieve more user attributes for authorization decisions. This might lead to a more complex attribute management and mapping along with policies. For the CYCLONE federation provider and SlipStream authentication integration, the next step is to implement the OpenID Connect ACF on the SlipStream side and define tests for this interaction. For the current use cases which work with browser logins and existing IaaS accounts, the current system meets the needs. However, the delegation of user rights to SlipStream for command line applications is an open question and identified as a long term problem to tackle. On the integration of OpenNaaS and StratusLab, current work evaluates the use of OpenNaas multi-tenancy over its usage by only StratusLab to perform required network configurations on the network infrastructure.

The solutions provided in this document will be further developed and documented in the future deliverables D4.4 and D4.5.

References

[EduGAIN] http://services.geant.net/edugain/About_eduGAIN/Pages/Home.aspx

[RFC2617] HTTP Authentication: Basic and Digest Access Authentication. Available at: https://www.ietf.org/rfc/rfc2617.txt

[RFC7235] Hypertext Transfer Protocol (HTTP/1.1): Authentication. Available at: https://tools.ietf.org/html/rfc7235

[RFC6241] Network Configuration Protocol (NETCONF). Available at: https://tools.ietf.org/html/rfc6241

[Elasticsearch] https://www.elastic.co/products/elasticsearch

[Logstash] https://www.elastic.co/products/logstash

[Kibana] https://www.elastic.co/products/kibana

[SAML2] https://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf

[FedProGitHub] https://github.com/cyclone-project/cyclone-federation-provider

[Keycloak] http://keycloak.github.io/docs/userguide/keycloak-server/html/identity-broker.html#identity-broker-overview

[D4.1] CYCLONE Deliverable 4.1 Security infrastructure specification and initial implementation, Available at: http://www.cyclone-project.eu/assets/images/deliverables/Security infrastructure specification and initial implementation.pdf

[D3.1] CYCLONE Deliverable 3.1 Evaluation of Use Cases

[D7.2] CYCLONE Deliverable D7.2 Overlay with focus on component manager, Available at: http://www.cyclone-project.eu/assets/images/deliverables/Overlay with focus on component manager.pdf

Abbreviations

WP	Work Package
WPL	Work Package Leader
ACF	Authorization Code Flow
OIDCACF	Open Id Connect Authorization Code Flow
JWT	JSON Web Token
JSON	JavaScript Object Notation

<END OF DOCUMENT>