



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 D4.1

Security Infrastructure Specification and Initial Implementation

Due date: 31/12/2015

Submission date: 18/12/2015

Deliverable leader: TU Berlin (TUB)

Editors list: Slawik, Mathias (TUB), Demchenko, Yuri (UvA)

Dissemination Level

<input checked="" type="checkbox"/>	PU: Public
<input type="checkbox"/>	PP: Restricted to other programme participants (including the Commission Services)
<input type="checkbox"/>	RE: Restricted to a group specified by the consortium (including the Commission Services)
<input type="checkbox"/>	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	Loomis, Charles
QSC AG	QSC	
Technische Universität Berlin	TUB	Slawik, Mathias
Fundacio Privada I2CAT, Internet I Innovacio Digital A Catalu- nya	I2CAT	
Universiteit Van Amsterdam	UVA	Demchenko, Yuri
Centre National De La Recherche Scientifique	CNRS	

Change history

Version	Date	Partners	Description/Comments
1	29/05/2015	TUB	Initial creation
2	28/10/2015	UVA, TUB	Updated document with Feedback from Y. Demchenko (UVA)
3	07/12/2015	UVA, TUB	Consolidated document
4	15/12/2015	TUB	Ready for internal review
5	17/12/2015	TUB, LAL, SixSq	Internal review and revision. Final version for submission.

Table of Contents

1. Introduction and Overview	8
1.1. <i>Linked Deliverables</i>	<i>8</i>
1.2. <i>CYCLONE Environment and Requirements</i>	<i>9</i>
1.3. <i>Objectives for the Security Architecture</i>	<i>10</i>
1.3.1. Global Security Objectives	10
1.3.2. Cloud Application Security Objectives	10
1.3.3. Cloud Deployment Security Objectives	10
1.3.4. Cloud Infrastructure Security Objectives	10
1.3.5. Cloud Network Security Objectives	10
1.4. <i>Security Actors and Use Cases</i>	<i>10</i>
1.4.1. Use Case: Federated Authentication	11
1.4.2. Use Case: VM Deployment	12
1.4.3. Use Case: Federated Authorization	13
1.5. <i>Application Lifecycle (simplified)</i>	<i>15</i>
1.6. <i>Infrastructure Implications for Cloud Security</i>	<i>16</i>
2. Components and Deployment.....	17
2.1. <i>Federation Provider</i>	<i>18</i>
2.1.1. Namespaces and User Attributes Mapping	18
2.1.2. Federation Provider Deployment	19
2.2. <i>Shibboleth IP & eduGAIN</i>	<i>19</i>
2.3. <i>SlipStream.....</i>	<i>21</i>
2.4. <i>IaaS platform (LAL Cloud, IRT OpenStack)</i>	<i>21</i>
2.5. <i>OpenNaaS & Network Resources.....</i>	<i>21</i>
2.6. <i>Distributed Logging.....</i>	<i>21</i>
2.7. <i>Bioinformatics Applications</i>	<i>21</i>
3. Federation Provider (FP) Security Modelling and Threat Analysis.....	22
3.1. <i>External Dependencies.....</i>	<i>22</i>
3.2. <i>Trust Levels</i>	<i>22</i>
3.3. <i>Entry Points.....</i>	<i>23</i>
3.3.1. Assets	23
3.3.2. Data Flow Diagram.....	24
3.3.3. Threat List	26

4. Future Activities.....	29
4.1. Cloud Application End-to-End security.....	29
4.2. End-to-end HTTP security through the Trusted Cloud Transfer Protocol (TCTP)	29
4.3. SSH-Login into the Cloud Services	29
4.4. Using the network topology for controlling TCTP.....	29
4.5. Transparent deployment of TCTP	29
4.6. Security Lifecycle Management and Trust Bootstrapping	30
4.7. Secure VM teardown	30
4.8. Non-browser HTTP-based identity federation and delegation.....	30
4.9. Activities to be fit into the use cases.....	30
4.9.1. Distributed Authorization using XACML	30
4.9.2. Location-based Access Control	31

Figures Summary

Figure 1: Federated Authentication Use Case	11
Figure 2: VM Deployment Use Case	12
Figure 3: Federated Authorization	13
Figure 4: Federated Authorization Mechanisms	14
Figure 5: Components Overview	17
Figure 6: Deployment Environment	18
Figure 7: eduGAIN search interface.....	19
Figure 8: eduGAIN details of CYCLONE Federation Provider.....	20
Figure 9: Data Flow Diagram	24

Executive Summary

This document summarizes the specification and the initial implementation of the basic CYCLONE security infrastructure. Its goal is enabling security within federated, multi-cloud solutions through the deployment of secure applications and services as well as automating the configuration of security properties.

On the conceptual layer, it consists of a model of CYCLONE actors, use cases, and components. Those are instantiated within the CYCLONE software stack to provide required functionality to realize the CYCLONE use cases. As a matter of course, we apply security best practices as well as a design mindful of security implications for the architecture. We emphasize this by conducting a thorough and structured security analysis on the Federation Provider, which we also plan on offering as-a-service in the future. We tried to keep this deliverable focused and terse. We link other deliverables and materials for further information.

1. Introduction and Overview

The basic motivation for the creation of the CYCLONE security infrastructure is to enable holistic security functionality in multi-cloud deployments and federated cloud architectures. This holistic security functionality should benefit both end-users, as well as their corresponding organizations by meeting relevant requirements and offering a set of best-practice and ready-to-use components.

There is a great diversity in the IT security infrastructures of different organizations. It is therefore a challenging task to create the CYCLONE security infrastructure in a generic way, as the adaptability of the infrastructure components is highly dependent on the maturity and the scope of the IT security infrastructure of the organizations adopting CYCLONE.

This document first gives an overview about the CYCLONE environment and the requirements of the use cases. Afterwards, the security architecture as well as its objectives, components and deployment are presented. We analyse security threats and countermeasures for the current deployment of the Federation Provider, as this is the pivotal element of the architecture and we also plan on offering it as-a-service in the future. The document concludes with an outlook on upcoming security-related activities within CYCLONE.

1.1. Linked Deliverables

This document references other previously published deliverables, which can be found [on our website](http://www.cyclone-project.eu/deliverables.html)¹. The following table presents relevant deliverables and their related content:

Number	Title	Related content
D3.1	Evaluation of Use Cases	Description of CYCLONE Use Cases as well as their requirements.
D5.1	Functional Specification of the E2E Network Service Model	Detailed description of OpenNaaS and the network service requirements it addresses within CYCLONE.
D6.1	Complex Application Description Specification	Summary of SlipStream and the application description it uses.
D6.2	Specification of Interfaces for Brokering, Deployment, and Management	Summary of SlipStream approaches to the application lifecycle, deployment, management, and brokering.
D7.1	Description of Testbed	Overview of the testbed architecture, as well as the network connectivity and all the employed software.
D7.2	Overlay with Focus on Component Manager	Focussed presentation of the role of SlipStream within the testbed, as well as the demonstrators realized in Year 1.

¹ <http://www.cyclone-project.eu/deliverables.html>

1.2. CYCLONE Environment and Requirements

The CYCLONE Environment and its Requirements were derived from the Deliverable D3.1, the evaluation of the use cases. In summary, it consists of the environments of the two use cases: first, the environment of the French Bioinformatics Institute (IFB) for the Use Case 1 and second, the environment of the energy use case. For brevity, this document focuses on the IFB environment as the energy use case has quite similar requirements.

There are a number of requirements stated in Deliverable D3.1, from which only a small number pertains to the global Security Infrastructure of CYCLONE. The following table shows the functional and non-functional requirements for the security infrastructure, their D3.1 ID, a short description, and how we address them within the security architecture:

Type	Name (D3.1 ID)	Description	Addressed by
Functional	Federated Identity Management (4, 7, 11, 22, 30)	Every cloud service access needs to use a Federated Identity Management. This also includes the Bioinformatics VM web interface, the storage, the SSH login, and the X2Go service.	The CYCLONE Federation Provider and the Federated Authentication use case scenario.
	Federated Authorization Management (5, 7, 11, 22, 30, 31, 32)	Every cloud service authorization has to rely on the federated identity. This also includes the Bioinformatics VM web interface, the storage, the SSH login, and the X2Go service. Furthermore the authorization should support the definition and use of groups as well as custom lists of user identities.	The Federated Authorization use case scenario.
	End-to-end secure data management (8)	Any data management task (e.g., uploading data) has to be secured end-to-end.	In general, TLS+VPN. When using proxies, TCTP (see 4.2, p. 29).
Non-functional	User Interfaces (1,2,3)	The user interfaces MUST support web technologies. It MAY support mobile devices.	All of the components are accessible via web and, when possible, have a responsive layout.
	Public Cloud Security (13)	All security functionality should not “break down” if applied in a trusted public cloud.	The general conception is not relying on a specific cloud model. Also, the security infrastructure does not state any requirements or demands which would weaken the overall level of security.

1.3. Objectives for the Security Architecture

There are a number of different security objectives for the Security Architecture, divided into global objectives and cloud application, cloud deployment, and cloud infrastructure objectives.

1.3.1. Global Security Objectives

- Offer a federated log-in system for reusing user identities, e.g., eduGAIN² federated user identities.
- Offer a distributed logging for debugging and auditing purposes.

1.3.2. Cloud Application Security Objectives

- Provide easy-to-use integration mechanisms of the Federation Provider into cloud applications, e.g., into the IFB Bioinformatics Cloud, in order to raise the application's security level by integrating a secure and tested solution for web authentication.
- Use the federated user identities for authorization purposes.
- Secure the transmission of sensitive data, such as the biomedical data.
- Use the distributed logging within cloud applications.

1.3.3. Cloud Deployment Security Objectives

- Integrate the federated log-in system into the SlipStream deployment manager.
- Use the federated identities within SlipStream ACLs.
- Offer a secure deployment of the security components via SlipStream.
- Use the distributed logging within SlipStream, e.g., for debugging deployments or auditing usage of cloud resources.

1.3.4. Cloud Infrastructure Security Objectives

- Use the distributed logging within the LAL cloud infrastructure for debugging VM operations and auditing VM host processes.

1.3.5. Cloud Network Security Objectives

- Enable a software-defined network within the cloud infrastructure, e.g., configure the firewall rules.

1.4. Security Actors and Use Cases

Within the security architecture, the main interaction is between the **Cloud Service** and the **Cloud Service User**. Within CYCLONE there are multiple services, e.g., **Deployed Applications** (e.g., a Bioinformatics VM), **SlipStream**, the **Logging**, an **IaaS Platform** (e.g., LAL Cloud), and the **IFB Portal**. There are also different Cloud Service User instantiations, the **Bioinformatician**, a **VM Developer**, as well as a **Cloud Operator**. We do not model the actor "Cloud Service Provider", as requirement 13 from D3.1 was to be independent of any specific cloud model.

² <http://services.geant.net/edugain/Pages/Home.aspx>

1.4.1. Use Case: Federated Authentication

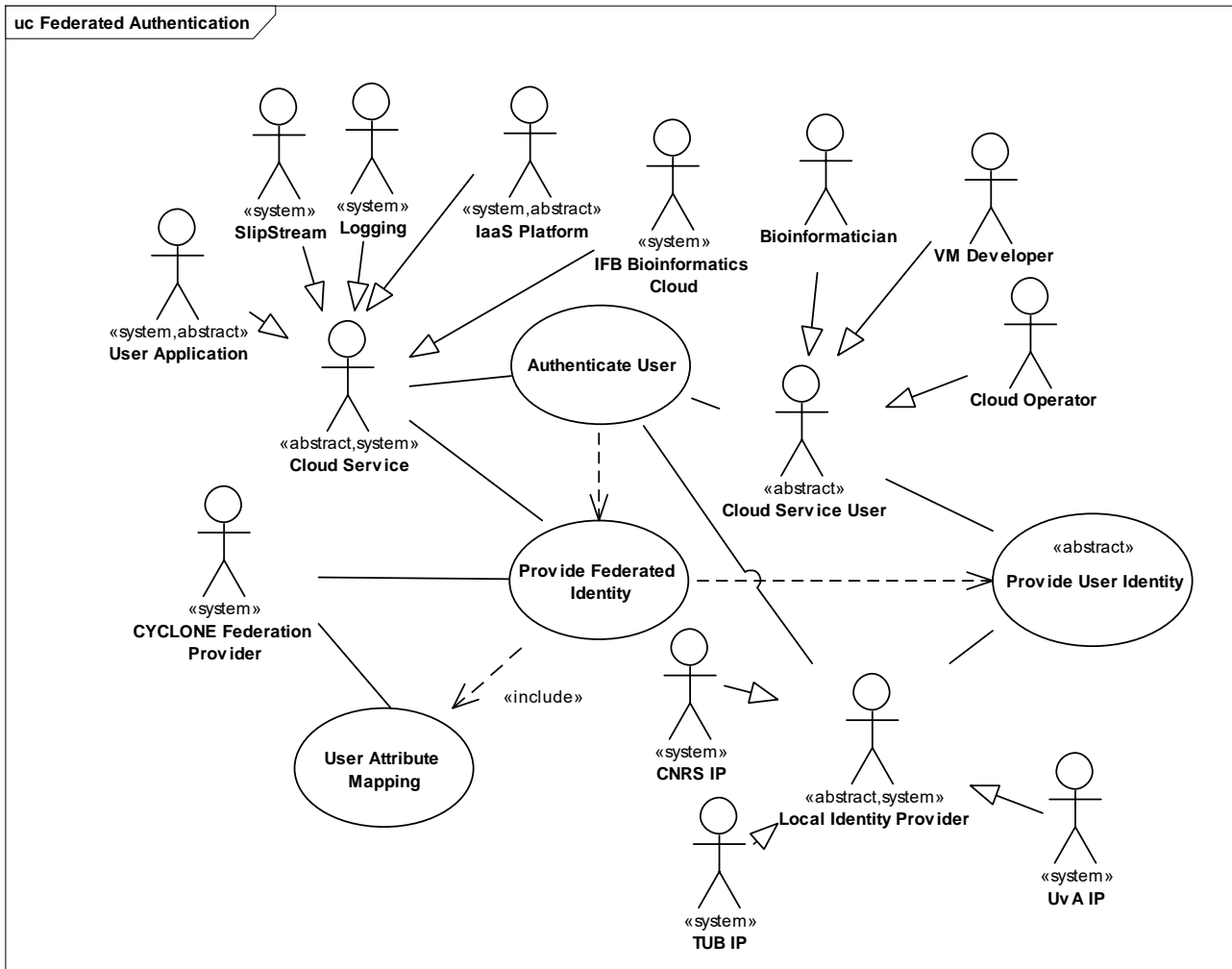


Figure 1: Federated Authentication Use Case

Figure 1 shows an overview of the Federated Authentication Use Case: any CYCLONE system actor can perform federated user authentication: the deployed user applications, SlipStream, the Logging system, the IaaS platform, as well as the IFB Bioinformatics Cloud. The CYCLONE Federation Provider provides federated user identities through relying on the local identity providers to authenticate users and provide their identity via the SAML 2.0 Web Authentication Workflow. The CYCLONE Federation Provider supports all Identity Providers of eduGAIN, for example, TU Berlin, CNRS, and UvA. Providing a federated identity includes user attribute mapping, which means transforming SAML user assertions into JSON Web Token claims. This can be configured in detail, either globally or client/service specific.

1.4.2. Use Case: VM Deployment

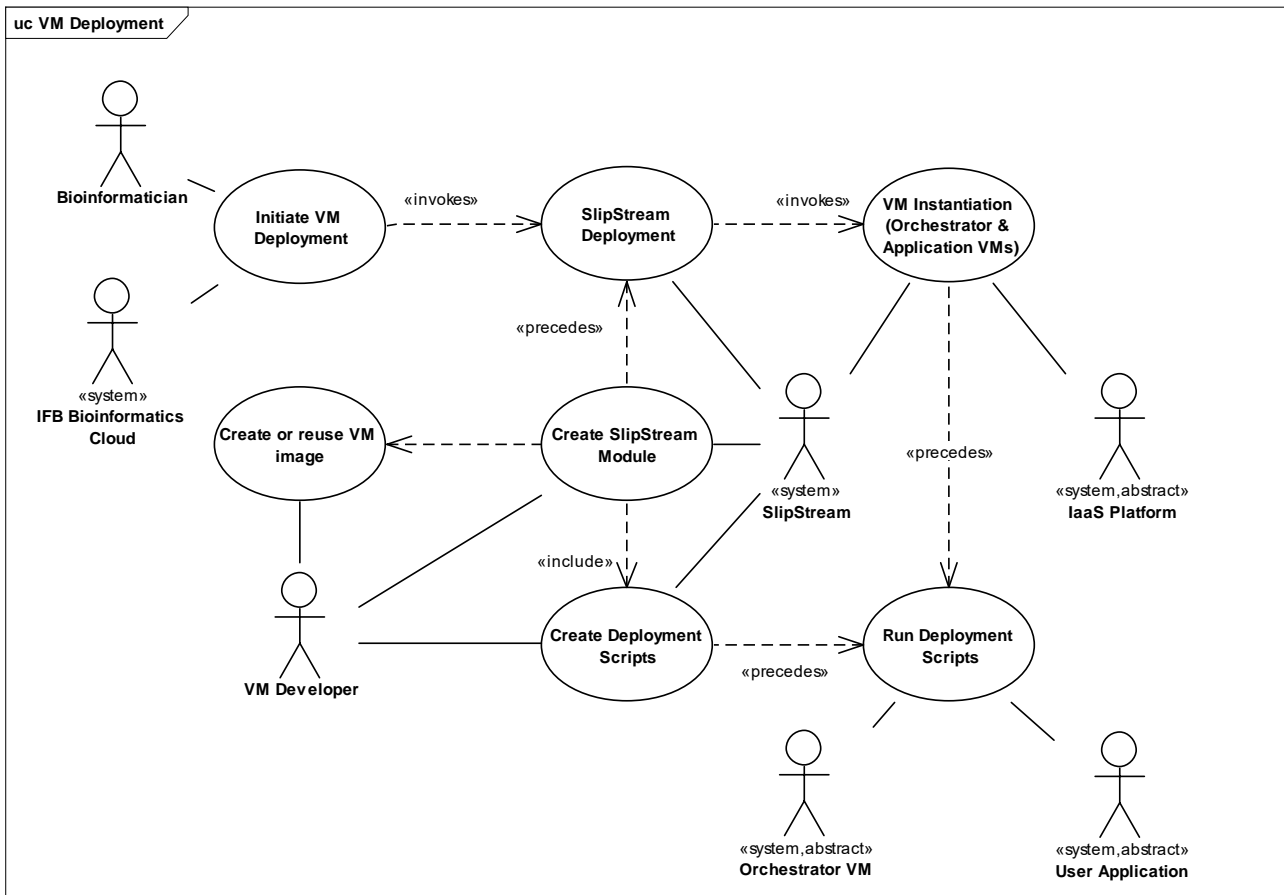


Figure 2: VM Deployment Use Case

In order to better understand the federated authorization use case, we first describe how VM deployment is handled in CYCLONE, as illustrated in Figure 2: As a first step, a VM developer needs to create a SlipStream module, which consists of a VM image as well as deployment scripts. As a module base, VM developers can either use a set of predefined images (e.g., Ubuntu LTS) or create their own.

The IFB Bioinformatics Cloud provides facilities for Bioinformaticians to initiate the SlipStream module deployment. This deployment creates application VMs and an Orchestrator VM on a configured IaaS platform (e.g., LAL Cloud). The Orchestrator configures the deployed VMs through a deployment script run. After this process completes, the Bioinformaticians get notified through the IFB Bioinformatics Cloud portal.

1.4.3. Use Case: Federated Authorization

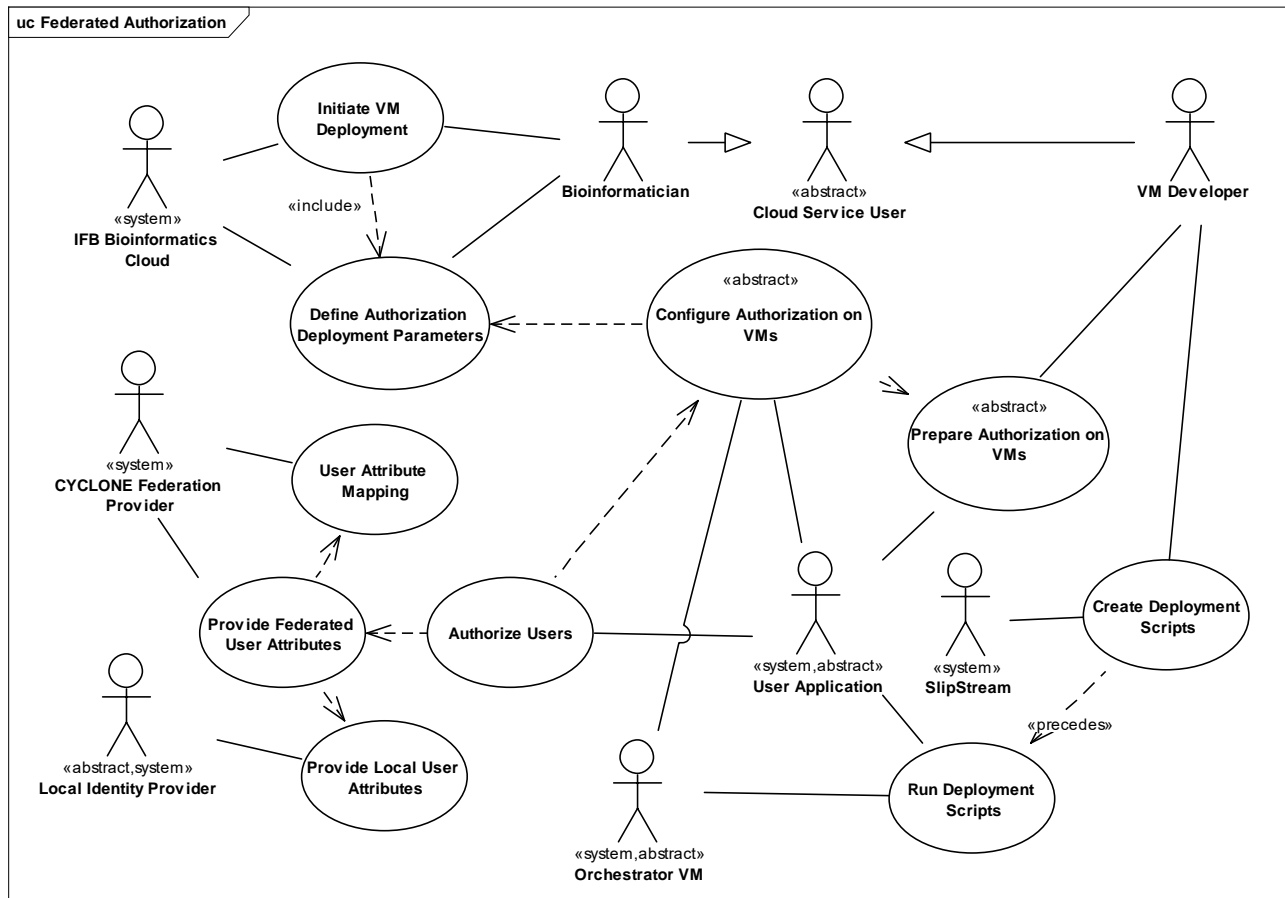


Figure 3: Federated Authorization

The preceding Figure 3 shows the Federated Authorization Use Case: As a first step, VM developers have to prepare their VMs to use authorization based on federated user identities, as described more in detail in the next paragraph. When Bioinformaticians initiate the deployment of VMs, they can define deployment parameters relevant for the authorization. These include, for example, a list of users and groups which should be able to have access to the machines. Those parameters are referred to within the deployment scripts so that the orchestrator VM can configure the authorization on the machines accordingly.

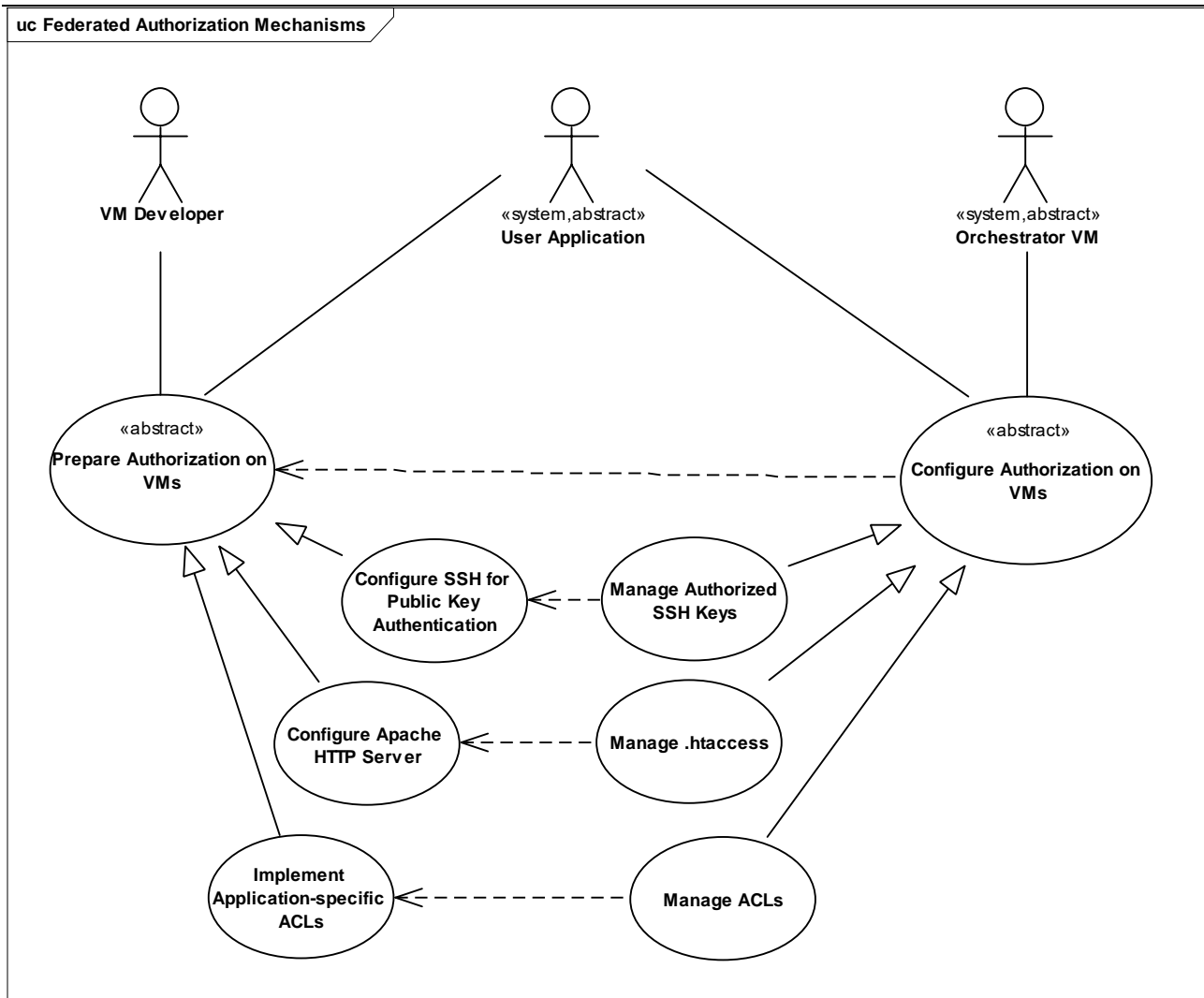


Figure 4: Federated Authorization Mechanisms

Based on the use case description, we foresee three different methods for federated authorization, which are shown in the preceding Figure 4:

1. SSH Public Key Authorization

Users are able to persist their public SSH keys in the IFB Cloud. The keys of the users who the deployer selected are copied into the `~/ssh/authorized_users` in order to give them access to the machine. Some services are offered through X2Go³ remote desktops, which support log in using the same mechanism.

³ <http://wiki.x2go.org/doku.php/start>

2. Apache HTTP Server Authentication & Authorization

The [mod_auth_openidc](https://github.com/pingidentity/mod_auth_openidc)⁴ module enables the Apache HTTP Server to use the CYCLONE Federation Provider for federated user authentication. The regular [Require-Statements](https://httpd.apache.org/docs/2.4/en/mod/mod_authz_core.html#require)⁵ can then be used, for example, in the [.htaccess](https://httpd.apache.org/docs/2.4/howto/htaccess.html)⁶ file configuration in order to permit or deny users access to the system, based on the issued user claims. This enables easy AA functionality for all Apache-hosted applications, e.g., PHP and Python solutions.⁷

3. Application-specific ACLs

Any application which is OpenID Connect – enabled can implement their own ACLs. In the CYCLONE ecosystem, SlipStream is an example of such an application. With the new login module of SlipStream, it relies on federated user identities to enforce SlipStream ACLs.

As some IFB users are unfamiliar with SSH and SSH key handling, we'll work on an alternative solution for federated authorization and authentication for SSH in Year 2, which we describe in Section 4.3 (SSH-Login into the Cloud Services) on page 29.

1.5. Application Lifecycle (simplified)

The application lifecycle is mostly covered by the use case VM Deployment (Section 1.4.2, page 12). As it consists of quite a lot of steps, one could define a great number of phases. To keep things simple, we instead use a reduced set of four phases for describing VM deployment in CYCLONE:

Phase 1 "Preparation"

In phase 1, the VM developer creates a deployable SlipStream module. This module would use deployment parameters to configure the instance-specifics, e.g., the permitted users and groups, the URLs of backend systems - logging, federation provider, and so on. It should also include components to be able to integrate with other CYCLONE components, e.g., the logging and the Federation Provider.

Phase 2 Deployment and Configuration

Phase 2 is initiated by a Bioinformatician which instructs the IFB Bioinformatics Cloud to deploy the VM using SlipStream. The IFB Bioinformatics Cloud provides SlipStream with all required parameters (e.g., authorization-related). The whole deployment and configuration workflow is described in detail in the [SlipStream documentation](#)⁸.

Phase 3 Main operation, with optional scaling and configuration changes

In this phase, the deployed VM can be used by authorized users. When it is running, the VM can be scaled up or down by SlipStream⁹. As a scaling script can also update the machine configuration, changed configurations can be applied by issuing a scale command to SlipStream, possibly without changing the number of VMs ("null" scaling).

⁴ https://github.com/pingidentity/mod_auth_openidc

⁵ https://httpd.apache.org/docs/2.4/en/mod/mod_authz_core.html#require

⁶ <https://httpd.apache.org/docs/2.4/howto/htaccess.html>

⁷ In D7.2 we show how Wordpress, an example PHP application, can use this mechanism for federated authorization.

⁸ <http://ssdocs.sixsq.com/en/latest/index.html>

⁹ More information on scaling can also be found in the advanced SlipStream tutorial on ssdocs.sixsq.com

Phase 4 VM teardown

Phase 4 is initiated by a Bioinformatician which instructs the IFB Bioinformatics Cloud to instruct SlipStream to tear down the VMs. As is required by the CYCLONE use cases, we'll create a mechanism which ensures that all sensitive data is thoroughly deleted when a VM is torn down, as described in Section 4.7 (Page 30).

1.6. Infrastructure Implications for Cloud Security

Any CYCLONE user must have a federated identity, e.g., the Bioinformatician, VM Developer, and Cloud Operator. This simplifies the implementation of the application authentication, as there is only one authentication mechanism for all user applications and applications do not need to handle user registration and password reset, for example. This is no drawback: if the CYCLONE Security Architecture would be implemented in other (e.g., company-)environments, the CYCLONE Federation Provider also supports regular "non-federated" identity sources, such as LDAP and a local user database.

2. Components and Deployment

This chapter provides an overview about the components and their deployment. It starts with a general overview before providing a summary of each component.

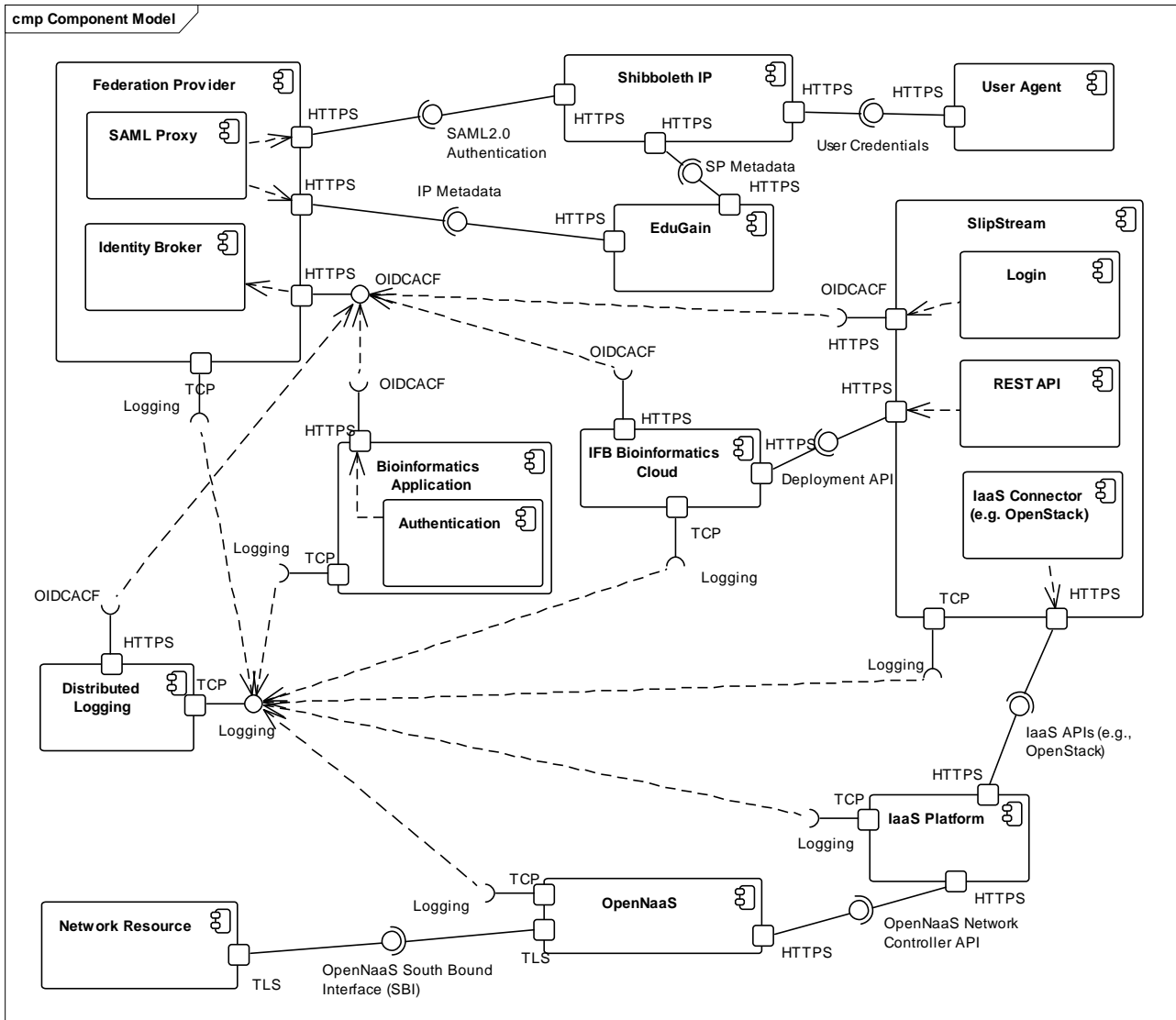


Figure 5: Components Overview

The preceding Figure 5 shows the main components of CYCLONE as well as their interconnection. The Federation Provider provides the OpenID Connect Authentication Code Flow (OIDCASF) which is used for federated authentication of users to the Bioinformatics Applications, the IFB Bioinformatics Cloud, SlipStream and the Distributed Logging. In order to provide this, it is dependent on both the local Shibboleth Identity Providers, as well as the eduGAIN Federation.

SlipStream provides the REST API, which is used by the bioinformatics cloud for initiating deployments. Within SlipStream, the IaaS connector invokes the respective IaaS APIs (e.g., OpenStack), which in turn uses OpenNaaS as a Network Controller for configuring Network resources accordingly. All relevant applications log into the distributed logging via TCP, i.e., the Bioinformatics Application, the Bioinformatics Cloud, SlipStream, IaaS platform, and OpenNaaS.

The following Figure 6 shows the environment where the components are currently deployed. For details on the environment, see D7.1, the description of the CYCLONE testbed, and D7.2, the description of the role of SlipStream within the Testbed.

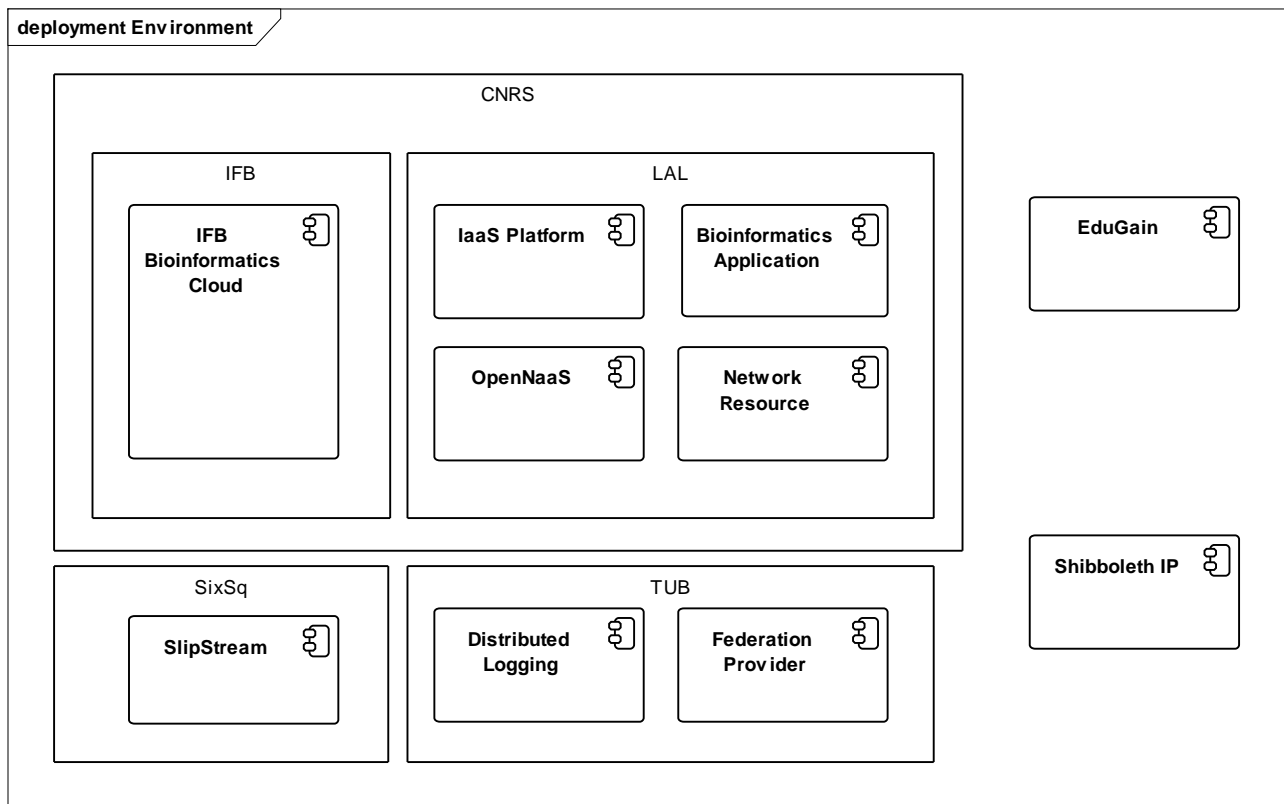


Figure 6: Deployment Environment

2.1. Federation Provider

The Federation Provider lets cloud service users login using their eduGAIN identities. It provides relying parties an OpenID Connect Authentication Code Flow, which transmits user claims using a signed JSON Web Token (JWT)¹⁰. Deliverable 7.2 contains sections on the Federation Provider as well as information about demos. The source code is available in the CYCLONE organization on GitHub [6].

2.1.1. Namespaces and User Attributes Mapping

The Federation Provider has numerous options for mapping eduGAIN SAML attributes to JWT claims. Currently, the following claims are mapped from eduGAIN attributes:

- A unique user identifier (`eduPersonPrincipalName`)
- The home organization's domain name (`schacHomeOrganization`)
- User's relationship(s) to their institution (`eduPersonAffiliation`)
- The preferred name when displaying entries (`displayName`)

The attribute mapping is quite flexible, allowing global as well as client-specific mappings.

¹⁰ JWT is a lightweight SAML-alternative for web- and browser-based SSO, see <http://jwt.io/>

2.1.2. Federation Provider Deployment

As there is only one single Federation Provider for the current CYCLONE ecosystem, this section only describes how it would be deployed again in the same setting. Other settings would need respective adjustments, which we currently cannot foresee.

The Federation Provider is deployed from a [Docker-based GitHub repository¹¹](#). There is an example Keycloak configuration file (keycloak-export.json) which contains many default and exemplary roles, clients, and users. We have adjusted this to our needs, putting in signed certificates as well as the SAML-based integration with eduGAIN through the PHP library [SimpleSAMLphp¹²](#). Currently, TUB has deployed the Federation Provider manually on a server at the SNET chair using the Docker repo. We have created a SlipStream module which we'll use for deployment on OpenStack as soon as the Interoute testbed is available for deployment. Then, there will be two deployments: the stable production deployment on the CYCLONE Interoute OpenStack installation and a regular development deployment at TUB.

2.2. Shibboleth IP & eduGAIN

All eduGAIN Identity Providers are based on [Shibboleth¹³](#). eduGAIN provides the metadata of all registered Identity and Service Providers, which can be queried using [a web interface¹⁴](#):

eduGAIN
PART OF THE GEANT SERVICES PORTFOLIO

HOME | eduGAIN STATUS | CONTACT US | FORMALITIES | TOOLS | HELP

eduGAIN Technical Site

Entity type
☐ All ☒ Identity Providers ☐ Service Providers ☐ Attribute Authorities (standalone)

Entity category filter
No filter

Federation filter (you may select multiple)
All federations
Armenia - AFIRE
Austria - AConet Identity Federation
Belgium - Belnet Federation

Entity filter
CYCLONE
Enter multiple words, and the match will be against any of them, put several words in double quotes to match the whole phrase, precede words or phrases with plus signs to make them required in the search or minus signs to filter out the unwanted (more info).

SAML 2.0 support filter
No filter

Show Clear form

Global statistics

All entities:	2509
IdPs:	1485
SPs:	1025
Standalone AAs:	3

☐ EntityId search
☒ Whole entity multiword search
☐ Reverse whole entity multiword search

Listing of all Service Providers (1)

SP	Entity ID	Entity details
1 CYCLONE Federation Provider Registrar: DFN AAI Org: Technische Universität Berlin	https://federation.cyclone-project.eu/samlbridge/module.php/saml/sp/metadata.php/cyclone-saml-bridge	

DANTE European Commission Information Society and Media Credits Legal

Figure 7: eduGAIN search interface

¹¹ <https://github.com/cyclone-project/cyclone-federation-provider-apache-oidc-demo>

¹² <https://simplesamlphp.org/>

¹³ <https://shibboleth.net/>

¹⁴ <https://technical.eduGAIN.org/entities.php>

This list also contains the metadata of the CYCLONE Federation Provider, which is registered as an eduGAIN Service Provider:

Entity details
Show XML
Close

Entity information

Entity ID:	https://federation.cyclone-project.eu/samlbridge/module.php/saml/sp/metadata.php/cyclone-saml-bridge
Registrar:	DFN AAI

Language: en

Display Name:	Technische Universität Berlin
Name:	e160
URL:	http://www.tu-berlin.de


Language: de

Display Name:	Technische Universität Berlin
Name:	e160
URL:	http://www.tu-berlin.de

Contact details

technical	Mathias Slawik; mail: mathias.slawik@tu-berlin.de
support	Mathias Slawik; mail: mathias.slawik@tu-berlin.de
administrative	Mathias Slawik; mail: mathias.slawik@tu-berlin.de

Service Provider information



Language: en

Service Name:	CYCLONE Federation Provider
Display Name:	CYCLONE Federation Provider
Description:	The CYCLONE Test Federation Provider is used for testing and demonstrating the technologies developed within the CYCLONE project
Privacy policy:	https://secure.cyclone-project.eu/sp_privacy_policy.html

Language: de

Service Name:	CYCLONE Federation Provider
Display Name:	CYCLONE Federation Provider
Description:	Der CYCLONE Test Federation Provider dient zum Testen und zur Demonstration der im CYCLONE – Projekt entwickelten Technologien

Protocols

urn:oasis:names:tc:SAML:1.1:protocol; urn:oasis:names:tc:SAML:2.0:protocol

Requested attributes

displayName (SAML:2.0)	urn:oid:2.16.840.1.113730.3.1.241 (required)
eduPersonAffiliation (SAML:2.0)	urn:oid:1.3.6.1.4.1.5923.1.1.1.1 (required)
eduPersonPrincipalName (SAML:2.0)	urn:oid:1.3.6.1.4.1.5923.1.1.1.6 (required)

Figure 8: eduGAIN details of CYCLONE Federation Provider

2.3. SlipStream

Much information about SlipStream, its usage, and existing demos can be found in D3.1, D6.1, D6.2, and D7.2.

Within the Security Architecture, SlipStream is used for deployment of Bioinformatics applications on the LAL Cloud, configuring federated authorization through deployment parameters, logging in users via OIDCACF as well as logging all relevant output in the distributed logging system.

2.4. IaaS platform (LAL Cloud, IRT OpenStack)

There are two IaaS platforms in CYCLONE: LAL provides a StratusLab platform for Bioinformatics deployment, which could potentially be migrated to OpenStack in Year 2. IRT provides an OpenStack based deployment testbed.

In the current conception of the Use Case deployment, users of the IFB Bioinformatics Cloud won't be interacting directly with the LAL APIs, but only with the Bioinformatics Portal. This portal uses its own SlipStream account to deploy VMs. Therefore, no Bioinformatics User can ever retrieve the LAL cloud credentials as they do not have access to the SlipStream account of the Bioinformatics Cloud.

2.5. OpenNaaS & Network Resources

A wealth of information on OpenNaaS and the devices it controls can be found in Deliverable D5.1, the functional specification of the E2E network service model. Within the security architecture, OpenNaaS is used directly by the IaaS platform to configure the Network Resources to implement the network topology required by the deployed VMs. There is no federated login to OpenNaaS, as only the IaaS platform is a consumer of OpenNaaS APIs.

2.6. Distributed Logging

Deliverable D7.2 describes the current deployment of the distributed logging. The source code of the logging system is available in the CYCLONE organization on GitHub [6].

By relying on the widely used ELK stack we hope to provide an easy-to-use distributed logging solution. The main use cases for the logging would be the diagnosis of errors and providing data for possible security audits. We have transferred the TRESOR implementation of multi-tenancy to CYCLONE, i.e., only people from the same tenant (identified by the `schacHomeOrganization` claim) can see the logs of their tenant. Based on a possible refinement of the use case requirements, we'll work on integrating a more comprehensive access control within the logging.

2.7. Bioinformatics Applications

The description, deployment, and CYCLONE-extensions of the bioinformatics applications is covered by D3.1, D7.1, and D7.2 where we already show how applications can be extended by federated login, authentication, as well as authorization. There are also other demo applications, such as Wordpress, for which we have demonstrated its extension by the CYCLONE federated authentication in D7.2.

3. Federation Provider (FP) Security Modelling and Threat Analysis

This Chapter applies the [OWASP Application Threat Modelling¹⁵](https://www.owasp.org/index.php/Application_Threat_Modeling) methodology onto the main component of the CYCLONE security architecture, the Federation Provider. It is based on the following steps:

1. Application decomposition
2. Definition of entry points, assets, and trust levels
3. Creation of a data flow diagram
4. Determining and ranking threats
5. Identification of countermeasures and mitigation strategies

As mentioned before, the Federation Provider is currently deployed in the TUB environment, which will later serve as the development environment, as the productive deployment of the Federation Provider will be performed within the IRT OpenStack Testbed. As threat analysis is performed most reasonably within an existing deployment, we chose to analyse the current deployment, instead of speculating on the future architecture. An eventual Threat Analysis based on the IRT environment will show the similarities and differences and therefore the threats dependent and independent of the deployment environment.

3.1. External Dependencies

ID	Dependency
D1	The Federation Provider is deployed on a hardened VM with sufficient security for login and file system access.
D2	The database of the FP is an embedded H2 database ¹⁶ with persistent disk-based tables.
D3	The Federation Provider Deployment relies on a Docker Compose deployment.
D4	A securely managed SSL reverse proxy is in front of the Wildfly Application Server.

3.2. Trust Levels

ID	Name	Description
T1	Anonymous Web User	A user who is connected to Keycloak but has not provided credentials
T2	User with eduGAIN Identity	A user who has logged in via an eduGAIN identity
T3	Relying Party	A third party which relies on Keycloak for authenticating users
T4	FP Admin	A Federation Provider administrator
T5	FP DevOp	A Federation Provider Developer/Operator

¹⁵ https://www.owasp.org/index.php/Application_Threat_Modeling

¹⁶ A SQL database implemented in Java, most often used as an embedded DB: <http://www.h2database.com/>

3.3. Entry Points

ID	Name	Description	Trust Levels
E1	Wildfly Application Server (HTTP)	The HTTP interface to the Wildfly application server powering the FP. It is reverse proxied by a secure SSL proxy.	
E1.1	OpenID Connect API	The API used by relying parties to authenticate federated users via the FP	T3
E1.2	Administration Console	The Servlet used for administering the FP	T4
E1.3	Log-in Screen	The FP log-in screen	T1, T2, T4
E1.4	Account Manager	The account management used by users to manage their own accounts	T2, T4
E2	Apache	The HTTP interface to an Apache HTTP server containing the SimpleSamlPHP bridge to eduGAIN	
E2.1	SimpleSamlPHP Bridge	The SimpleSamlPHP Bridge (SP) for eduGAIN	T1, T2, T4

3.3.1. Assets

ID	Name	Description	Trust Levels
A1	FP Database	Assets regarding the H2 database of the FP	
A1.1	User List	A list of all known users and their home organizations	T4
A1.2	Client List	A list of all clients, containing their certificates, redirect URIs, etc.	T4
A1.3	Keycloak Config	The general configuration of Keycloak	T5
A2	OpenID Connect API	Assets regarding the OpenID Connect API	
A2.1	Personal Data	The personal data of the logged in users, provided by eduGAIN.	T3
A3	SimpleSamlPHP	Assets regarding the SimpleSamlPHP Bridge to eduGAIN	
A3.1	SAML configuration	The certificate and metadata for the eduGAIN integration.	T5

3.3.2. Data Flow Diagram

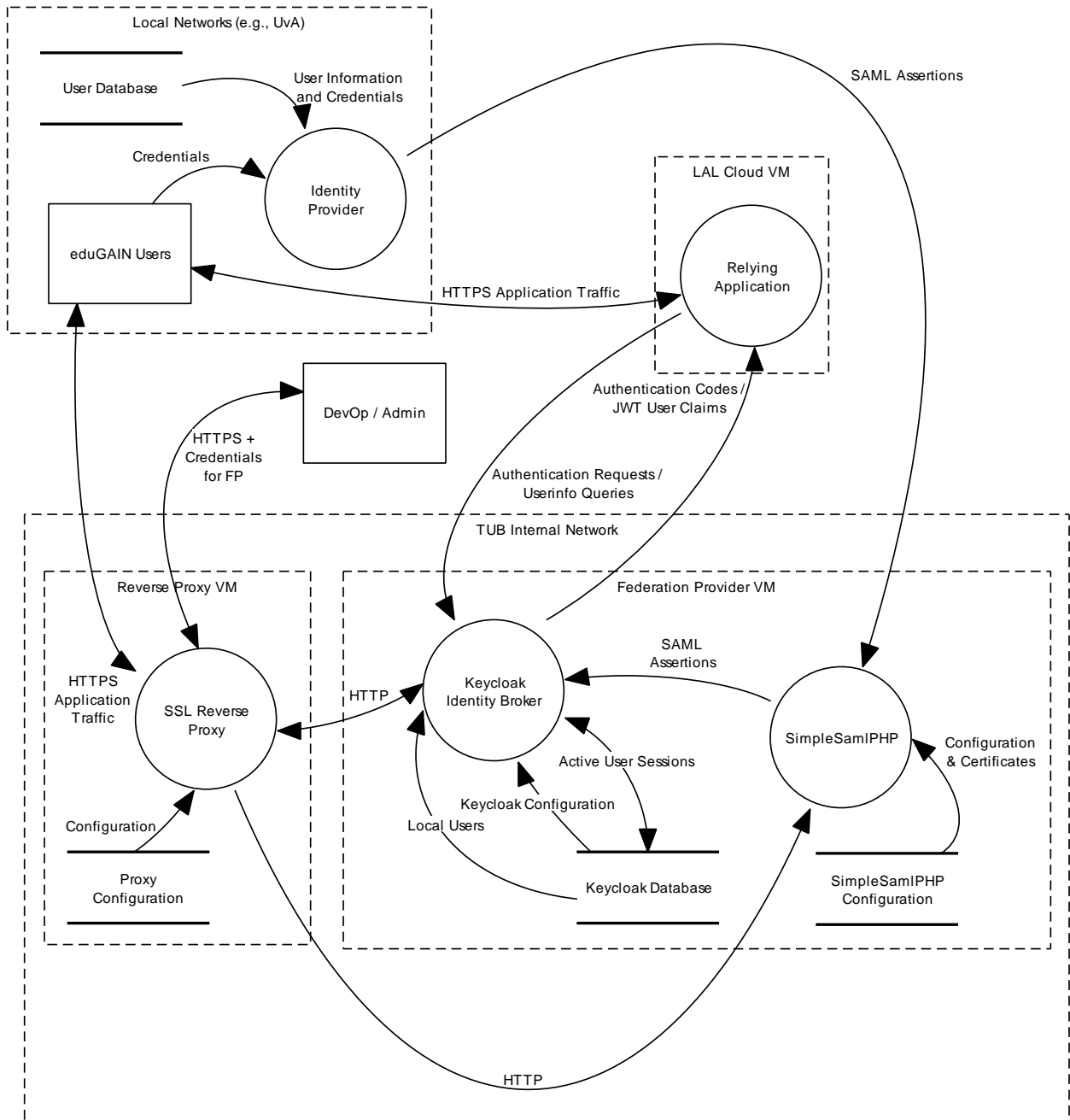


Figure 9: Data Flow Diagram

The Data Flow within the Federation Provider is displayed in the preceding Figure 9.

There are five distinct Security Domains:

1. TUB Internal Network

The network between the Reverse Proxy VM and the Federation Provider VM. It is exclusively owned, managed, and secured by TU Berlin. The communication path between both VMs can be considered secure, i.e., there are a number of controls in place so that eavesdropping on the communication is not possible, allowing the use of HTTP for inter-VM traffic.

2. Reverse Proxy VM

The Reverse Proxy VM serves as a TLS reverse-proxy for all SNET-managed publicly accessible services.

3. Federation Provider VM

The Federation Provider VM hosts both the Keycloak Identity Broker as well as the SimpleSamlPHP bridge to eduGAIN Identity Providers. It is accessed via reverse-proxied HTTP requests from the Reverse Proxy VM.

4. LAL Cloud

All Bioinformatics Applications are deployed within VMs on the LAL Cloud.

5. Local Networks (e.g., UvA)

Most eduGAIN users access cloud applications from a local network.

In order to authenticate against relying applications, eduGAIN users interact with the Federation Provider through the SSL Reverse Proxy. The Proxy retrieves its configuration (e.g., SSL certificates, reverse IPs) from the local file system and relays traffic to the Keycloak Identity Broker as well as the SimpleSamlPHP bridge.

On the Federation Provider VM, the Keycloak Identity Broker handles the main functionality, e.g., creating and signing JWTs, attribute mapping, as well as implementing the OpenID Connect Authentication Code Flow. It is accompanied by the SimpleSamlPHP bridge, which offers SAML-based eduGAIN federation. The Identity Broker loads local users (FP DevOps / Admins) as well as its configuration (e.g., certificates, clients) from a local H2 database. Active FP User Sessions are also persisted there.

According to OpenID Connect, relying applications initiate authentication requests, which cause eduGAIN users to authenticate against their local Identity Providers. Those providers return SAML assertions to the SimpleSamlPHP bridge, which relays them to the Identity Broker. It returns Authentication Codes in response to the authentication requests which allows the relying applications to query for user information at the Identity Broker.

3.3.3. Threat List

The following table contains a list of threats, their possible causes as well as a mitigation strategy. The last column designates threats whose mitigation is already implemented (I, highlighted green) or designed but not yet implemented or validated in the current development deployment (D, highlighted blue). For information on DREAD please see the [OWASP wiki on this topic](#). For rating, we use [Table 3.6 from the Microsoft threat modelling document](#).

Threat	Cause	Mitigation	Accidental	Malware	Script Kiddie	Curious A.	Motivated Attacker	Criminals	Damage Potential	Reproducibility	Exploitability	Affected Users	Discoverability	DREAD Value	High/Medium/Low Threat	Designed or Implemented
Impersonation of any user to relying applications	Stolen private key for Keycloak JWT certificate.	Deployment on hardened VM.		X	X	X	X	X	3	3	2	3	2	13	H	D
	Broken encryption through too little entropy.	Use of high-quality entropy sources.					X	X	3	1	1	3	1	9	L	D
Impersonation of eduGAIN user to relying applications	Applications accept tokens, without verifying them with the FP. Tokens could be manufactured, or swapped with different issuers or subjects.	In OIDC/ACF, user agents do not supply tokens, but codes, which only the application can transform into tokens. Therefore, applications should never allow user agents to send them tokens, but only use (possibly encrypted) sessions.					X	X	3	3	3	3	3	15	H	D
	Server impersonation to client	Every client should validate the SSL connection to the FP and also received tokens using a shared secret or PKI certificates.					X	X	3	3	2	3	3	14	H	D

Threat	Cause	Mitigation	Accidental	Malware	Script Kiddie	Curious A.	Motivated Attacker	Criminals	Damage Potential	Reproducibility	Exploitability	Affected Users	Discoverability	DREAD Value	High/Medium/Low Threat	Designed or Implemented
Impersonation of DevOps and Admins	Password guessing by brute force	Strong password policy, Fail2Ban integration, Increasing Hash Iterations, and possibly enforcing Keycloak two-factor authentication with time-based one-time pass.				X	X	X	3	3	3	3	3	15	H	D
	Clickjacking on Keycloak login page	Usage of X-Frame-Options and Content-Security-Policy by Keycloak to protect against clickjacking.				X	X	X	3	3	2	3	2	13	H	I
Leaking user information (Displayed user name, email address, user organisation, user affiliation [staff, student, ...])	Access Token Redirect ([5] 16.8)	Access Token should be audience and scope restricted, Keycloak has to validate audience and scope.	X	X	X	X	X	X	2	3	2	3	1	11	M	I
	Stolen access token by Man-in-the-Middle attack	Enforce SSL on all connections, use trusted server certificates, use secure networks if possible.					X	X	2	3	1	3	1	10	M	I
	Compromised / replayed keycloak access code	Cryptographically strong random value prevents guessing. Access codes cannot be reused, preventing replay attacks. Also, the lifetime of access codes is very short.					X	X	2	1	1	3	1	8	L	I
Revealing of sensitive information (e.g., what service is accessed, which client is used, ...)	Plaintext OpenID Connect requests	Encryption of OpenID Connect requests	X	X	X	X	X	X	2	3	3	3	3	14	H	D

Threat	Cause	Mitigation	Accidental	Malware	Script Kiddie	Curious A.	Motivated Attacker	Criminals	Damage Potential	Reproducibility	Exploitability	Affected Users	Discoverability	DREAD Value	High/Medium/Low Threat	Designed or Implemented
Misuse of Keycloak as Open Redirector	No redirect URL validation	Only allow specific redirect URLs for each client.	X	X	X	X	X	X	1	3	3	1	3	11	M	D
Client abuse	Server response repudiation, e.g., by not signing the response.	Response signing with non-repudiatable key by server. Enforced signature validation on client.					X	X	1	3	3	3	3	13	H	D
Server abuse	Client request repudiation, e.g., when clients do not sign their request	Request signing with non-repudiatable key by client. Enforced signature validation on server.					X	X	1	3	3	3	3	13	H	D
Further exposure to security and privacy threats	Request disclosure	Encryption of OpenID Connect request in an encrypted JWT within the request and request_uri parameter (see Sec 16.1 of [5])	X	X	X	X	X	X	1-3	3	3	3	3	13-15	H	D
	CSRF attack on Keycloak and OpenID Connect	Oauth 2.0 and Keycloak state cookies are matched against transmitted state parameter.					X	X	1-3	2	2	1	1	7-9	L	I

The threat analysis shows, besides many others, two most important Federation Provider threats: brute-force attack on administrators' passwords, as well as lacking validation of tokens by cloud applications. Before conducting this analysis, both were also our estimated attack vectors. There was a [famous blog article in March 2015](#) showing that many JWT libraries have critical vulnerabilities allowing attackers to bypass the verification step. Since then, [jwt.io](#) offers an overview about the supported verification steps of different JWT libraries.

4. Future Activities

After the Use Case analysis was completed, all of the security activities were roughly scheduled into the project years. Some of them need to have their validation defined, as they have not been required yet by the use cases. These are described in the following sections.

4.1. Cloud Application End-to-End security

End-to-end security denotes that within a communication path no intermediary can access transferred data. This is especially relevant within HTTP-based cloud computing environments, as even if communication is secured by TLS, some HTTP intermediaries, such as load balancers, proxies and gateways terminate TLS connections and thus have access to possibly confidential data.

4.2. End-to-end HTTP security through the Trusted Cloud Transfer Protocol (TCTP)

Certain Bioinformatics VMs will be extended by the Trusted Cloud Proxy in order to allow them to be accessed with end-to-end security, i.e., from the bioinformatics cloud application user's browser to the server process in the VM.

4.3. SSH-Login into the Cloud Services

Within the federated authorization use case, the SSH login depends on Bioinformaticians managing SSH keys on their machines as well as on the Cloud Portal. Not every end user is able to perform these tasks well and it requires manual effort. We'll work on creating a PAM module which allows people to use their federated identity for easy login to deployed VMs.

4.4. Using the network topology for controlling TCTP

If the network is controlled by OpenNaaS, the TCTP functionality could be enabled or disabled based on the network topology, e.g., it should be disabled, if the user already uses a VPN tunnel to connect to a privately owned infrastructure, and enabled, if the access is managed by cloud intermediaries, e.g., cloud proxies and gateways.

4.5. Transparent deployment of TCTP

The Cloud Deployment Manager should be able to transparently add TCTP functionality to application deployments using the Distributed Cloud Proxy in "Reverse Proxy & TCTP Server" mode.

4.6. Security Lifecycle Management and Trust Bootstrapping

The cloud based services are deployed, operated and destroyed according to the services lifecycle that may include either only their deployment in cloud or also continuous development, deployment and operation process also known as DevOps. When deploying cloud based services and applications, the security services need to be also deployed.

Security lifecycle management can also solve another important issue in building consistent security services for on-demand provisioned applications and infrastructures such as initial trusted key or certificates distribution and trusted infrastructure bootstrapping. This research will be based on the initial work by UvA published in [1] and [2].

4.7. Secure VM teardown

When VMs are torn down, certain legal provisions require the complete removal of any remaining data, e.g., transient VM storage, existing backups, and more. CYCLONE will allow the definition of such teardown policies which need to be implemented by the respective IaaS solution, e.g., OpenStack.

4.8. Non-browser HTTP-based identity federation and delegation

As we see in CYCLONE it is very challenging to provide federated authentication for non-browser HTTP applications, e.g., RESTful command-line tools: while there is the SAML 2.0 Enhanced Client Profile (ECP), it is unsupported by a number of tools and federations, most notably eduGain. We are using OpenID Connect, where there is a so called “Direct Access Grant”, but this works only for local IDPs and not for a whole federation. Of course there is also Kerberos, but it is not designed for easy federation and HTTP applications. So we’re not able to rely on any established mechanism for this purpose and need to find a new mechanism to enable, for example, SlipStream CLI tools to use a federated identity.

Besides this, there is also no accepted and reliable delegation mechanism between SlipStream and the underlying IaaS providers. Even if people would be able to login to SlipStream using their federated identity, it is still very hard to delegate their cloud resource access to a third-party component.

4.9. Activities to be fit into the use cases

The project proposal and the Declaration of Action contain a preliminary use case analysis which assumes the need for a wide spectrum of functionalities fitting to a great variety of use cases and application scenarios. In Year 1 we prioritized providing needed functionality for the bioinformatics and energy use cases.

The following sections present further security functionality which needs to be fit into additional use cases in the future, mostly pertaining to complex authorization scenarios. Work Package 4 will include these activities in a meaningful way into the next use cases addressed in Years 2 and 3.

4.9.1. Distributed Authorization using XACML

Distributed multi-domain authorisation and XACML based policy expression allow for complex authorisation decision making while changing inter-domain security context, including delegation of authority and rights.

Typical use case for such scenario is a complex distributed resource reservation in multi-domain large scale applications. The project will look for such applications amongst partner’s networks and first of all in such

areas and complex multi-domain network provisioning (like GEANT network) or data intensive scientific or production workflow of high definition video production.

Meantime additional background research on the distributed multi-domain authorisation and XACML based policy expression will be done.

4.9.2. **Location-based Access Control**

TUB proposed the use of Location-based Access Control in order to meet further authorization and general security requirements within certain use cases, such as within the German Health Centres addressed by TRESOR. We'll look into translating the research we presented in [3] and [4] to further CYCLONE use cases in Years 2 and 3.

References

- [1] Yuri Demchenko, Diego R. Lopez, Joan 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. ISBN 978-1-4244-9348-7
- [2] Peter Membrey, Canh, Ngo, Yuri Demchenko, Cees de Laat, Trusted Virtual Infrastructure Bootstrapping for On Demand Services. The 7th International Conference on Availability, Reliability and Security (AREs 2012), 20-24 August 2012, Prague. ISBN 978-0-7695-4775-6
- [3] Zickau, S. and Thatmann, D. and Ermakova, T. and Repschläger, J. and Zarnekow, R. and Küpper, A. (2014). Enabling Location-based Policies in a Healthcare Cloud Computing Environment. Proceedings of the 3rd IEEE International Conference on Cloud Networking (IEEE CloudNet). IEEE.
- [4] Thatmann, D. and Slawik, M. and Zickau, S. and Küpper, A. (2012). Towards a Federated Cloud Ecosystem: Enabling Managed Cloud Service Consumption. Economics of Grids, Clouds, Systems, and Services, GECON 2012. Springer, 223-233.
- [5] Sakimura, N. and Bradley, J. and Jones, M. and de Medeiros, B. and Mortimore, C. (2014). OpenID Connect Core 1.0 incorporating errata set 1. Retrieved on 2015-12-15 from http://openid.net/specs/openid-connect-core-1_0.html
- [6] GitHub organization "cyclone-project". <https://github.com/cyclone-project>

Appendix A Abbreviations

ACL	Access Control List
ELK	Elasticsearch Kibana Logstash
IaaS	Infrastructure-as-a-Service
JWT	JSON Web Token
SAML	Security Assertion Markup Language
SDN	Software Defined Networks
TCTP	Trusted Cloud Transfer Protocol
VM	Virtual Machine

<END OF DOCUMENT>