# CYCLONE: The Multi-Cloud Middleware Stack for Application Deployment and Management

Mathias Slawik
Berlin Institute of Technology, Germany
mathias.slawik@tu-berlin.de

Christophe Blanchet
CNRS IFB
christophe.blanchet@france-bioinformatique.fr

Yuri Demchenko, Fatih Turkmen, Alexy Ilyushkin, Cees de Laat
University of Amsterdam, The Netherlands
{ y.demchenko, F.Turkmen, A.Ilyushkin, C.T.A.M.deLaat}@uva.nl

Charles Loomis
SixSq Sàrl
cal@sixsq.com

*Abstract*— **DevOps teams have to consider many technology and platform aspects when developing, deploying and operating cloud based applications: application deployments need to work everywhere on different cloud platforms, identities need to come from anywhere, and networks need to connect to anyone. The CYCLONE middleware is a holistic middleware stack that allows deploying and managing cloud based applications on multiple clouds and multiple cloud platforms. It includes a deployment manager, a practical identity federation, as well as a network manager that connects VMs independent of any specific infrastructure. This article explains the CYCLONE middleware stack, and what it can offer for application developers and operators. The paper describes in details the main bioinformatics use cases that evolve from a single VM installation for simple microbial research to multicloud infrastructure for advanced genomic resource. The paper also describes the CYCLONE federated identity management and access control infrastructure that significantly simplifies access for institutional users.**

***Keywords CYCLONE cloud automation platform for scientific application, SlipStream Cloud Management platform, Bioinformatics cloud based applications, CYCLONE Network Services Manager and Orchestrator (CNSMO), Overlay on-demand network provisioning, Intercloud Access Control, Trust Bootstrapping in Federated Clouds.***

## I. INTRODUCTION

The CYCLONE European Innovation Action focuses on three main areas that present challenges in multi-cloud settings: (1) application deployment and management, (2) authentication and authorization based on federated identities, and (3) software-defined network management. The project outcome is the integrated CYCLONE software stack that comprises preexisting production-ready software as well as our own extensions tackling multi-cloud issues. The whole stack contains solely open-source software in order to maximize its utility and also provide a basis for further collaboration. All of our developments are hosted in our GitHub Repository at github.com/cyclone-project and we provide comprehensive documentation on our website, www.cyclone-project.eu. Our goal is to apply the stack to enhance diverse use cases in academic and commercial production- and near-production settings to ensure the eventual applicability of the projects results and highlight CYCLONE's utility in existing DevOps environments.

## II. CYCLONE FLAGSHIP USE CASE: BIOINFORMATICS

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

One of the CYCLONE use case partners, the French Institute of Bioinformatics (L'Institut Français de Bioinformatique), consists of 32 bioinformatics platforms (PF) spanning the entire French territory as well as a national hub, the "UMS 3601–IFB-core", that is the representative in the project. The IFB has deployed a cloud infrastructure on its own premises at IFB-core and aims to deploy a federated cloud infrastructure over the regional PFs. This cloud infrastructure is devoted to the French life science community, research, and industry, with services for the management and analysis of life science data. More concretely, Bioinformaticians can use this "IFB Bioinformatics Cloud" to deploy VMs containing useful bioinformatics research tools.

As we'll show in the course of this article, the CYCLONE middleware stack is fully utilized in the use case, managing the deployment of the applications, authenticating and authorizing application users using with their federated identities, as well as interconnecting the components of clustered tools securely. Therefore, the Bioinformatics use case is an excellent fit to the challenge areas addressed by CYCLONE.

## III. CYCLONE MIDDLEWARE COMPONENTS

There is a lot of information already available about the components of the CYCLONE cloud middleware, for example, on our website [5] and within the other publications mentioned in the introduction. In the following, we will concentrate on the most important components for the integration into the bioinformatics use case and explain them briefly. This will guide the explanation of the subsequent "CYCLONE in Action" section.

### A. *Nuvla: Deploy Applications on any Cloud*

SlipStream, a cloud application management platform, allows developers to define portable cloud applications and for operators to deploy automatically those applications on multiple cloud infrastructures. With SlipStream, the operators can manage the full lifecycle of cloud applications, including provisioning, scaling, migration, and clean up. SixSq releases the SlipStream Community Edition under the Apache 2 license and the source code can be found in the SlipStream organization in GitHub. In addition, SixSq operates a free SlipStream SaaS called Nuvla [7] that can be used to access a number of public clouds.

The Service Catalog, a core feature of SlipStream, contains "offers" from cloud service providers, detailing VM resource configurations, locations, prices, and other information. Developers and operators can attach "policies" that describe constraints to an application. SlipStream then uses those policies to filter the available offers to eliminate those that do not meet the application requirements. The operator can then select any acceptable offer manually or allow SlipStream to choose the least costly offer automatically. These policies are completely general and can be used, for example, to deploy an application to a particular country for legal reasons or to choose a particular combination of CPU cores, RAM, and disk space.

### B. *CYCLONE Federation Provider: Federated Identities*

The Cyclone framework provides the CYCLONE Federation Provider as an approach to ease the hardships of federated multi-cloud identity management. We make special arrangements to ease the integration of preexisting academic identities that are federated through eduGAIN [8], as the end-users in many implemented CYCLONE use cases are academic researchers.

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

From an implementation perspective, the CYCLONE Federation Provider extends and enriches the Keycloak identity and access management solution [9] that is sponsored by RedHat. Keycloak has a rich feature set, mainly single sign-on supporting both SAML2 (as used by eduGAIN) as well as OpenID Connect (as used by many cloud applications). Keycloak can also broker identities, allowing end users to select which credentials they want to use for authentication, even supporting social network logins (e.g. Facebook). Our extensions to Keycloak comprise a data privacy aware session removal, an interface for self-service registration as well as templates that include terms of conditions and data privacy statements for each OpenID connect tenant.

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

### C. *PAM Module and Xpra Wrapper: WebSSO solution*

Many users that leverage multiple clouds face the problem of having a large number of user accounts to use with different services. As we detailed in the last chapter, this problem can be reduced using web-based single sign-on. However, no SSO implementation can be used satisfyingly for Secure Shell Login. This problem is amplified in the Bioinformatics use case as researchers share datasets and results by letting other people log into the VMs. Currently, involved researchers need to create a new user account for every person with whom they share data with.

Using the CYCLONE PAM module "pam_openid_connect" [10] allows SSH login using the federated identities of the end users, e.g., the Bioinformatics researchers. It is integrated with the SSH server through the PAM subsystem. The keyboard-interactive mode of SSH allows the PAM module to display a URL of an ephemeral web server started for this particular login session. When users follow this link, it initiates a regular Open ID Connect Authentication Code Flow with the CYCLONE Federation Provider that returns user attributes as a JSON Web Token to this ephemeral web server. The current implementation compares the user's email with a list of allowed emails in a file. This file can be edited by the bioinformaticians as well as created at deployment time by SlipStream. We consider our solution to be as simple as possible to solve the concrete requirements of the bioinformaticians, namely, to allow another user shell access to shared VMs. We forsee that further requirements could lead to the need to implement the Intercloud Access Control Infrastructure, explained later in this article.

Other bioinformatics software is provided as regular desktop applications. There are some preexisting tools that allow remote access to desktop applications, for example, Xpra [11]. The Xpra client and server can communicate via an SSH connection, possibly established using the CYCLONE PAM module. However, the manual setup and coordination of Xpra and SSH is not always done easily by the end users. To provide an easy way for the bioinformaticians to use remote desktop applications authenticated with their federated identity, we provide a desktop "wrapper" around both tools, available at [12]. We use Electron [13], a tool provided by GitHub to "Build cross platform desktop apps with JavaScript, HTML, and CSS" in order to lower our implementation effort and to provide the wrapper to a large range of different users and devices.

### D. *OpenNaaS CNSMO: Connect all the Clouds*

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

CYCLONE provides network services to cloud-based applications using OpenNaaS CNSMO (CYCLONE Network Services Manager and Orchestrator) that was presented in [14], available online at [15]. It is far more lightweight than comparable solutions such as Apache Mesos while still providing the essential network management APIs. The system is capable of deploying, configuring, and running multiple network services in both private and public environments. The most significant CNSMO feature is that it is agnostic to the underlying IaaS provider, running on top of any cloud service and being OS independent. Thus, CNSMO integrates networking aspects over federated clouds and allows tenants to request network services and manage them. CNSMO leverages Docker containers for easy deployment and management.

The CNSMO services can invoke themselves and are stateless and independent. Any CNSMO service can potentially launch any other presently developed service. In effect, CNSMO is lightweight, distributed, and modular: Lightweight service agents coordinate themselves by communicating through a distributed system state. CNSMO features a modular micro-service architecture that is scalable and extendable: agents are atomic single-purpose units. The CNSMO architecture is detailed in [16].

## IV. CYCLONE IN ACTION

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

### A. *Deploying Bioinformatics Software*

In [17] Lacroix, et al. present Insyght, a comparative genomic visualization tool that consists of 3 components: a pipeline of Perl scripts to compute the required data, a relational database to store these data and the visualization tool itself that queries the relational databases and presents these data in a user-friendly way. The platform automatically launches a set of bioinformatics tools (e.g. BLAST, PSI-BLAST, INTERPROScan) to analyse the data and stores the results of the tools in the relational database (PostgreSQL). These tools use several public reference data collections. A web interface allows the user to consult the results and perform the manual annotation (manual annotation means adding manually metadata and biological knowledge to the genome sequence). The popularity and vast functionality make Insyght a prime candidate for offering it on the IFB Bioinformatics Cloud.

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

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

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

### B. *Creating VPNs Over any Cloud*

CNSMO is integrated with Nuv.la, relying on the orchestrator functionality of SlipStream and being published as a module on Nuv.la. CNSMO on Nuv.la consists of a VM image that is able to run the CNSMO network services. A SlipStream “recipe” contains the deployment instructions to be run by the SlipStream orchestrator as a set of scripts. A single SlipStream application component can run any number of CNSMO agents (one agent in each of the application VMs) depending on the network service(s) that have been selected to be deployed together with the application. Figure 1 shows the integration of CNSMO in the SlipStream market place.

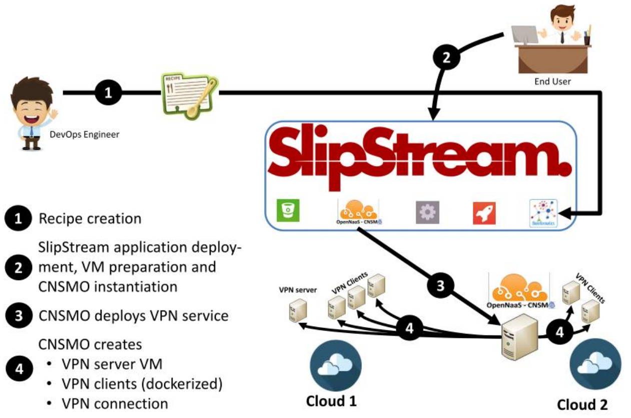

Figure 1. Integration of CNSMO with SlipSream

Until now, we implemented three network services: a multi-cloud VPN, a firewall, as well as a load balancer. We selected these services as they were the first to be requested within the flagship CYCLONE complex applications created by the use case partners. They have been fully implemented, tested, integrated, and validated over several cloud infrastructures. In the following, we’ll explain the use of CNSMO on the example of the VPN service. The general bootstrapping process consists of the following steps (see Figure 1):

0. Before deploying an application, the SlipStream recipe needs to be created that includes the network services to be deployed.

1. When the application deployment is initiated through SlipStream, the SlipStream Orchestrator VM instantiates the application VM images as new VMs on the target cloud and executes the respective deployment recipes. Part of this deployment is the CNSMO image that contains a systemd lifecycle service that runs CNSMO.

2. Once CNSMO is launched, it provides an API so that SlipStream can control the deployment of the network services. To this end, the CNSMO SlipStream application uses a deployment recipe that includes the appropriate instructions for CNSMO to deploy the chosen networking services, in this case a VPN. It is important to clarify that the recipe is run by SlipStream inside the CNSMO VM. For instance, for the concrete case of a VPN network service, the SlipStream command line client calls the CNSMO API to deploy the VPN service (the VPN server and the VPN clients).

3. The SlipStream deployment parameters are used by CNSMO to determine which services to deploy, e.g., the VPN service. For the exemplary distributed VPN service, CNSMO carries out the following steps:

- It creates the VPN server VM.
- It creates the VPN clients inside the VMs of the application that have been deployed by SlipStream.
- It launches the VPN server to configure the VPN clients and establish the VPN service.
- Finally, CNSMO uses the SlipStream command line client to announce to the rest of the components that the networking service has been set up, so that SlipStream can resume its deployment (running their deployment script) and give back control of the deployment.

### C. *Using Academic Identities in Research Prototypes*

The “Internet of Services Lab” (IoSL) is a teaching project of one of the participating institutions where students work in groups of three to six, implementing software related to numerous research projects and topics. Within this teaching project, there are different areas where the application of CYCLONE provides numerous benefits:

1. Rapid provisioning of resources for student projects: Students require resources for conducting their projects, e.g., Virtual Machines. In the current set-up it is a manual procedure to provision those resources. By leveraging the CYCLONE testbed as well as our deployment tools we minimize the required effort of the student supervisors considerably.

2. Utilization of SlipStream modules for reproducible application deployments: After students finish their course it is often problematic for other students and their supervisors to pick up their work. Most often, documentation is lacking and software versioning is not reliable, if it is even available. By building upon SlipStream modules we create application deployments that can be easily reproduced, extended, and scaled. Also, students will learn how to structure their applications to leverage cloud characteristics, e.g., how to create immutable application deployments.

3. Integration of the CYCLONE Federation Provider: From our experience, every built demonstrator has its own user management. Furthermore, they often do not follow security best practices. By integrating the Federation Provider into each demonstrator, students learn about federated identity and are also liberated from implementing their own user management, as all students and supervisors will be able to login to the demonstrators via eduGAIN.

## V. From Multi-cloud Towards the Intercloud

### A. *Extending the CYCLONE Bioinformatics Use Case*

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 2 shows the bioinformatics application deployed in Cloud 1 in a form of Virtual Private Cloud (VPC) that includes both the actual application that manage the whole scientific workflow and computing cluster. The bioinformatics engineer develops and deploys and application in Cloud 1 using development tools coupled or integrated with an application deployment manager, e.g., SlipStream. The application may use external scientific data and applications located in SciCloud A and B. In case of an excessive workload, some computational tasks can be outsourced to external cloud CloudExt, in particular in a standard cloudburst scenario. Similarly to the original use case definition, Figure 3 includes a scientific data archive for storing obtained results. The data visualisation and collaboration tools could also be hosted in the cloud and provided by specialised SaaS or cloud application providers.

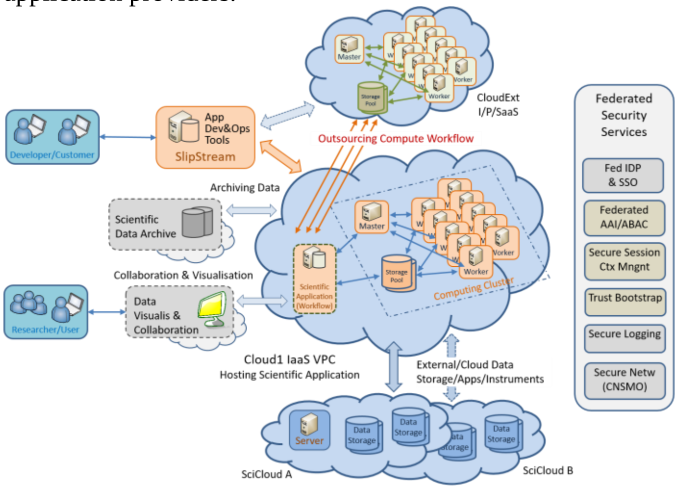

Figure 2: Extended Bioinformatics Use Case

Suggested security services are combined into the federated security services stack, as depicted on the right side of Figure 2, that include currently existing CYCLONE security services and those that are required for multi-cloud applications that are discussed in the next sections.

### B. *Intercloud Access Control Infrastructure*

Besides the basic modes of "VM-local" authentication and authorization required by the current Bioinformatics use cases, there are a number of extensions to the Intercloud access control infrastructure that are conceivable. Particular examples include federated access control and identity management approaches at the Intercloud level that are possibly integrated with cloud-native access control services, e.g., AWS AIM (Access and Identity Management) or Azure AAD (Azure Active Directory).

Besides this, we mainly investigated the possibility to enable multi-tenant access control with centralized decision points. On the one side, a single point of control with multiple enforcement points would simplify permission management when policies apply to multiple deployments and applications. It also enables authorizations with finer granularity. One the other side, this requires additional efforts, for example, establishing a rule vocabulary over all participating applications, installation, integration and maintenance of the required components, establishing means to authoring rules, and integrating externalized authorization in all participating applications. We concluded that it is highly dependent on the specific use case if the benefits outweigh the required efforts.

In order to support future use cases that predominantly benefit from centralized access rules management, we prepared multiple components that can be easily integrated into diverse scenarios:

- The PDP service mostly follows the XACML REST profile [18]. It accepts authorization decisions, evaluates a set of rules and returns an authorization decision to be enforced by the relying applications.
- The token service acts as a temporal credentials authority that issues and verifies temporal tokens that contain or reference a security context of a specific application session. The tokens managed by the service can be used for authorization services. Currently, the token service supports X.509 certificates and a custom token format for multi-tenant access control based on attributes. Application developers can also define their own token formats to meet their needs.
- The context management service manages the trust relations between cloud tenants that share resources and leverages the security tokens from the token service. It is a fundamental service for implementing the policy decision point in multi-tenant access control and managing the delegations between tenants.

The multi-tenant access control can also be implemented in a distributed manner, offering a RESTful service to its clients for authorizations. This mode is useful in settings where the security services are consumed in a distributed way 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 [19], 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. However, it is less complete in terms of XACML specification coverage. The second one is ATT-XACML that has better coverage of the

XACML standard and provides REST interfaces for XACML component functionality but is not optimized for performance.

### *C. Bootstrapping Trust in Federated Clouds*

Trust bootstrapping refers to the initialization of a single cloud node or the virtual private cloud with relevant security credentials. Our previous work [20,21] proposes the Dynamic Infrastructure Trusted Bootstrapping Protocol (DITBP). It leverages the Trusted Platform Module (TPM) and can be effectively implemented using available technologies and tools. TPM is typically available on all CPU boards and supported by majority of cloud platforms.

Among the available proposals, CYCLONE employs **keylime** [22] for bootstrapping trust within cloud nodes and the services running on them. It can be considered as an instance of DITBP that employs a cloud verifier to periodically check the integrity of the node. There are three steps involved during the deployment of a VM node with keylime:

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

## VI. FUTURE DEVELOPMENT

Our goal in creating the CYCLONE middleware stack is helping DevOps teams solve their current challenges in federated multi-cloud environments. As we applied the stack for the benefit of the highlighted use case stakeholders, we already achieved this goal, albeit in a small scale. We will continue applying the stack in further use cases and extend it as necessary. These use cases span a wide range of DevOps environments in order to strengthen and maturate the stack. The proposed solutions are already used by other projects and tools outside of CYCLONE in order to preserve our results after the end of the project funding.

## ACKNOWLEDGEMENT

The authors wish to thank the members of the bioinformatics platforms Centre Léon Bérard (Lyon, France) and IFB-MIGALE (Jouy-en-Josas, France) to provide their bioinformatics applications as respective use cases "Biomedical data analysis" and "Bacterial genomes analysis". This work is supported by the CYCLONE Horizon 2020 Innovation Action CYCLONE, funded by the European Commission under grant number 644925.

## REFERENCES

[1] 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.

[2] G. Succi and M. Marchesi, *Extreme Programming Examined (XP)*. Addison-Wesley Longman, Amsterdam, 2001.

[3] Y. Demchenko et al., "CYCLONE: A Platform for Data Intensive Scientific Applications in Heterogeneous Multi-cloud/Multi-provider Environment," in *2016 ieee International Conference on Cloud Engineering Workshop*, 2016, pp. 154–159.

[4] M. Slawik et al., "An Economical Security Architecture for Multi-cloud Application Deployments in Federated Environments," in *Proceedings of the 13th International Conference on Economics of Grids, Clouds, Systems and Services*, 2016.

[5] The CYCLONE project, "CYCLONE - Home." [Online]. Available: http://www.cyclone-project.eu/. [Accessed: 17-Jan-2017].

[6] S. Blank, "Perfection by subtraction – the minimum feature set." [Online]. Available: https://steveblank.com/2010/03/04/perfection-by-subtraction-the-minimum-feature-set. [Accessed: 10-Jan-2017].

[7] "Nuvla." [Online]. Available: https://nuv.la.

[8] GÉANT, "EduGAIN." [Online]. Available: http://www.geant.org/Services/Trust_identity_and_security/eduGAIN..

[9] RedHat, "Keycloak Open Source Identity and Access Management." [Online]. Available: http://www.keycloak.org/.

[10] E. Berdonces Bonelo and B. Brancotte, "pam_openid_connect module." [Online]. Available: https://github.com/cyclone-project/cyclone-python-pam. [Accessed: 11-Jan-2017].

[11] "Xpra home page." [Online]. Available: http://xpra.org/.

[12] E. Berdonces Bonelo, "Xpra-electron-client." [Online]. Available: https://github.com/cyclone-project/xpra-electron-client.

[13] "Electron - build cross platform desktop apps with javascript, html, and css." [Online]. Available: http://electron.atom.io/.

[14] J. Aznar et al., "CNSMO: A Network Services Manager/Orchestrator tool for cloud federated environments," in *2016 Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, 2016, pp. 1–5.

[15] OpenNaaS, "OpenNaaS CNSMO." [Online]. Available: http://opennaas.org/opennaas-cnsmo/. [Accessed: 06-Jan-2017].

[16] J. Aznar et al., "Specification of network management and service abstraction: CYCLONE Deliverable D5.2." [Online]. Available: http://www.cyclone-project.eu/assets/images/deliverables/Specification of network management and service abstraction.pdf. [Accessed: 06-Jan-2017].

[17] T. Lacroix1 et al., "Insyght: Navigating amongst abundant homologues, syntenies and gene functional annotations in bacteria, it's that symbol!" *Nucleic Acids Research*, 2014.

[18] REST Profile of XACML v3.0 Version 1.0, Committee Specification 02, 23 November 2014 [online] http://docs.oasis-open.org/xacml/xacml-rest/v1.0/cs02/xacml-rest-v1.0-cs02.pdf

[19] SNE-XACML, "A high performance xacml pdp engine." [Online]. Available: https://github.com/canhnt/sne-xacml.

[20] Demchenko, Y., C. de Laat, O. Koeroo, D. Groep, Re-thinking Grid Security Architecture. Proceedings of IEEE Fourth eScience 2008 Conference, December 7–12, 2008, Indianapolis, USA. Pp. 79-86. IEEE Computer Society Publishing. ISBN 978-0-7695-3535-7

[21] P. Membrey et al., "Trusted virtual infrastructure bootstrapping for on demand services," in *Seventh international conference on availability, reliability and security, ARES 2012*, 2012, pp. 350–357.

[22] N. Schear et al., "Bootstrapping and maintaining trust in the cloud," in Proceedings of the 32nd Conf. on computer security applications, ACSAC 2016, Los Angeles, USA, Dec. 5-9, 2016, 2016, pp. 65–77.