Cyclone

Newsletter #3

European Commission

Complete & Dynamic Multi-cloud Application Management

Introduction

CYCLONE is a Horizon 2020 innovation action funded by the European Commission which aims at integrating existing cloud management software to allow a unified management of federated clouds. Application service providers (ASPs) develop, deploy, and maintain complex computing platforms within multiple cloud infrastructures to improve resilience, responsiveness and elasticity of their applications. These applications are often designed to scale automatically in response to demand and to permit live upgrades of the underlying software.

A number of challenges are addressed by the CYCLONE project. In this newsletter we describe our approach and solutions for the following two, namely,

1. ***Multi-cloud deployment and scaling of federated applications.*** A challenging factor for application deployment and exploitation within the CYCLONE infrastructure is its highly dynamic nature, which raises the need for application and infrastructure managing solutions that quickly adapt to, or even anticipate, changing circumstances. Based on the use case scenarios described at CYCLONE deliverables and the corresponding requirements, we present how CYCLONE development addresses some of these. In particular, in this newsletter we describe the proposed solution for auto-scaling of cloud application to maintain a given Quality-of-Service (QoS). We briefly describe the two core components of the solution: SlipStream and an Analytics Engine.
2. ***Compliance analysis for bioinformatics applications deployed in the cloud.*** One of the aims of the CYCLONE project is to follow security compliance recommendations and standards in the CYCLONE developments. Leveraging on the Newsletter #2, we provide more details about the HIPAA/HITECH rules that create a basis for Protected Health Information (PHI) or Personally Identifiable Information (PII). We also explain mapping of different industry related standards to the general cloud compliance guidelines and best practices by Cloud Security Alliance (CSA).

Table of contents

CYCLONE at a glance

Contract number	*644925*
Call Identifier	*H2020-ICT-2014-1*
Duration	*January 2015 –December 2017*
Funding scheme	*Innovation action*
Budget	*3.84 M€*
EC Contribution	*2.84 M€*
Topic	*Advanced Cloud Infrastructures and Services (ICT-07-2014)*

contact@cyclone-project.eu

FOLLOW US

Twitter
@H2020_CYCLONE

LinkedIn
https://www.linkedin.com/groups/8259424

Web Site
http://www.cyclone-project.eu/

GitHub
https://github.com/cyclone-project

Application Performance Management for Multi-Cloud CYCLONE Environment

Charles Loomis Chief Technical Officer (SixSq)
CYCLONE Technical Coordinator and WP6 Leader
Miroslav Živković Researcher (UvA)
WP6 Member

Cloud computing infrastructures are a recognized alternative to buying and maintaining custom IT infrastructures on premise. The primary economic attraction is the ability to pay for the computational resources only when they are needed—a shift from capital expenditures to operational expenditures. The flexibility to change the allocated resource dynamically, known as elasticity, is the basis of the utility computing model.

The dynamic provisioning of virtualized resources offered by cloud computing infrastructures allows applications deployed in a cloud environment to increase and decrease *automatically* the allocated resources. The main purpose of this capability, known as auto-scaling, is to minimize automatically the resources allocated to an application while satisfying the varying workload and performance constraints. The need for auto-scaling is particularly important during workload peaks, during which applications may need to scale up to extremely large-scale systems. *Non-functional aspects, such as cloud application cost and performance, have a strong impact on the management and auto-scaling of cloud applications.*

We present here the application management platform for the CYCLONE that takes advantage of cloud federation to avoid the limitations of a single cloud provider, while maintaining a good quality of service (QoS) for the customers and minimizing infrastructure costs for the service provider. We first describe the motivating use cases and then identify the requirements of the system. Next, we describe the components of the CYCLONE application management system (centered on SlipStream) that contribute to the auto-scaling solution. This system provides the means, but not the knowledge, to scale a system automatically. The knowledge is embedded in the algorithms used to scale the application and we explain briefly how to integrate these algorithms (using Analytics Engine) to be used in CYCLONE for scaling applications based on both functional and non-functional requirements.

Use Cases

To motivate the CYCLONE auto-scaling framework, several use cases have been identified. We present here two use cases, and interested reader is referred to CYCLONE D6.3 deliverable for an detailed overview of all considered use-cases The former use-case highlights the general functionality (independent of the selected application) required of the system. The latter use case is a real-world one that further illustrates the requirements of the system and demonstrates the utility of the solution.

Generic Use Case: Monitoring

Performance volatility on cloud infrastructures is common and such volatility can lead to changes in application performance that need to be addressed. Typically, the performance deteriorates abruptly and the response time for an application increases significantly. Monitoring is essential to detect such situations and overall performance management of cloud applications. When the monitoring determines that the performance has changed significantly, the CYCLONE platform may request scaling up the application's resources or redeployment of application components elsewhere. Volatility can also lead to improved performance. Monitoring can also detect these situations and trigger a process to scale down the application.

Use case: Scientific Image Processing

Publicly funded research creates an immense amount of data that has general social, academic, and commercial value. However, finding viable business models that keep the maintenance costs for the public reasonable has been an obstacle to widespread availability of open data. SixSq is exploring solutions with European Space Agency (ESA) in which public data is hosted on European cloud infrastructures and partially or fully monetized to reduce the need for public subsidies.

Technically, a viable solution requires:

- Detailed knowledge of the storage locations of dataset components,
- Means of placing analysis applications near the data of interest, and
- Ranking of multiple providers based on price or other characteristics.

All but the first requirement are already features of the CYCLONE brokering and matchmaking components. Integrated data management is a crucial feature for this use case that is planned for SlipStream in the last year of the project. Advanced networking may also play a role here, if significant bandwidth is required for remote access to datasets.

Requirements

Based on the use cases, a set of requirements for the auto-scaling features was derived. Many of the requirements were already identified in other CYCLONE developments. Overall, these requirements require minor adjustments to existing features of SlipStream, rather than significant additional development. The requirements are as the following:

- Triggering of scaling actions of an application based on application metrics using simple, predefined algorithms (e.g. adding node based on machine load).
- Triggering of scaling actions of an application based on application metrics defined by the developer of the application
- Ability to publish application-specific benchmarks of cloud providers into the Service Catalog or Open Service Compendium.
- Placement based on static characteristics (e.g. geographical location) of a cloud service provider.
- Placement based on dynamic VM monitoring information from SlipStream itself.
- Placement based on external information pushed into the SlipStream Service Catalog or Open Service Compendium.
- Placement based on the join of all information associated with a given cloud service provider.
- Ranking of selected cloud service providers based on predefined algorithms (e.g. price).
- Ranking based on algorithms provided by the application developer and/or the application operator.
- Ability to trigger notifications/alerts through SlipStream.
- Ability to trigger scaling actions from within the application.
- Ability to search the Service Catalog and Open Service Compendium manually to see the results from various policies and to ideally then associate those policies with applications.

Proposed solution

The proposed solution for auto-scaling of cloud application to maintain a given Quality of Service (QoS) is illustrated in Figure 1. There are two core components of the solution: SlipStream and an Analytics Engine. SlipStream handles all aspects related to resource management for the application. The Analytics Engine encapsulates the knowledge necessary to decide what actions are required (if any) given the current state of an application.

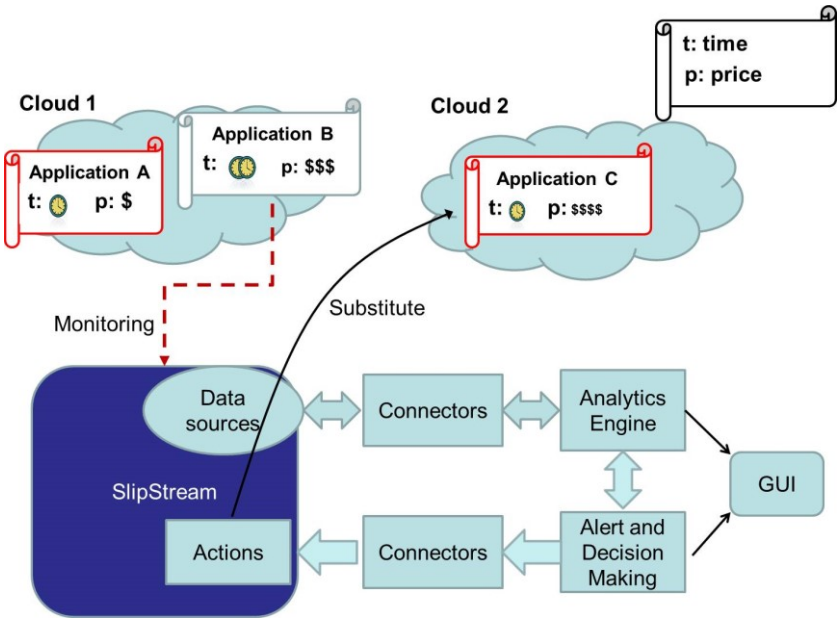

Figure 1: The QoS-control Architecture and Interaction with SlipStream

As illustrated in the figure, there are three functionally equivalent applications (A, B, C) that are deployed within two different cloud infrastructures (one and two). Based on current conditions, application requirements, and user preferences, SlipStream ranks possible placement solutions to present the user with an ordered list of alternatives. In this case, application B deployed within cloud infrastructure 1 was the optimal choice.

When the deployment begins, SlipStream starts the resource monitoring. Once the application is operational, it too can provide application metrics, such as performance or response time. These metrics create a time-series of data that is streamed to the Analytics Engine for analysis. The Engine currently supports two different mechanisms for data analysis: anomaly detection (discussed below) and application optimization (via resource scaling, component migration, or application substitution).

Anomaly detection

Anomalies are items, events or observations that do not conform to an expected pattern or other items in a dataset. For example:

- Significant deviations in time series metrics,
- Significant changes in message rates,
- Rare or unusual log messages, or
- Unusual user behavior.

If the expected or normal patterns of behavior are known, anomalies can be identified. Given a time-series, the Engine models the time-series producing an expected value that may be later consumed by the anomaly detection methods. The anomaly detection methods include both outlier and change-point detection algorithms. For an outlier detection, the algorithms identify a time-series element for which the observed value is significantly different from the expected value from the rest of the time series.

For change-point detection algorithms, the goal is to identify for a given time-series of observations whether the current point (or one in the near past) represents a change in the series. One desires to detect a potential change as soon as possible. In general, change-point detection methods monitor some test statistic, which is based on the observations, and issue an alarm if this test statistic exceeds a certain threshold, such that the calculated probability of not detecting a change-point is kept below a certain predefined value, for example, $\alpha = 5\%$. Statistically speaking, the change-point detection methods perform a hypothesis test for every time step. Change-point detection methods can be used to detect a change in mean or a change in the distribution of the time series, for example

Application Optimization

As for the optimization, these algorithms are used to provide optimal policies that take as input the cost functions, response-time distributions, and (theoretically) reward/penalty functions. These policies further specify when to substitute one application for another and which application to substitute. The value to be optimized may change by application and actor; for example, it may be maximizing the revenue, for the provider), or minimizing the costs, for the application owner.

The optimization algorithms may also be triggered by the anomaly detection, for instance when a performance anomaly has been identified and a new policy must be derived. Once a new policy is available, a decision is then made as to whether to apply the newly derived policy or not. The decision-making process may be completely automated. If changes are necessary, the specified actions are then applied to the deployed application by SlipStream. In Figure 2, a decision is made to substitute Service B in Cloud one by Service C in Cloud two.

SlipStream Components for Auto-scaling: Architecture

Non-functional requirements strongly influence the choice of cloud service providers (CSPs) for a given application deployment. These requirements include:

- Security guarantees or certifications of CSPs,
- Historical and current operational quality, and
- Application performance or response under varying loads.

The set of non-functional and operational requirements are often formalized as part of Service Level Agreements (SLAs) between the cloud application owners and cloud service providers. SlipStream, the CYCLONE component that deals with cloud application lifecycle management, provides the means to monitor such agreements and to react in case of violations.

The high-level architecture of SlipStream consists of four functional blocks:

- **Application Description Repository** Allows application developers to describe the functional and non-function requirements, software installation/configuration, and parameters for cloud components and multi-component cloud applications.
- **Service Catalog** Provides information concerning the administrative, financial, and operational characteristics of CSPs. The information is collected into "offers" that can then be selected based on both functional and non-functional requirements.
- **Deployment Engine** Combines application and CSP information to choose and then provision appropriate resources for an application. This engine handles the full lifecycle of the application including the initial deployment, scaling actions, and termination.
- **Monitoring** SlipStream monitors the state of all deployed cloud resources, allowing the deployment engine to control the state of resources, to raise alerts for abnormal conditions, and to provide usage information.

Users can access these functional blocks through a comprehensive API, a web browser interface, and a command line client. The web browser interface is roughly organized around the four functional blocks described above. The API is a REST-based API running over the HTTP(S) protocol. Figure 2 shows the functional blocks of the SlipStream server, access methods, and the underlying CSPs that provide resources for users' cloud applications.

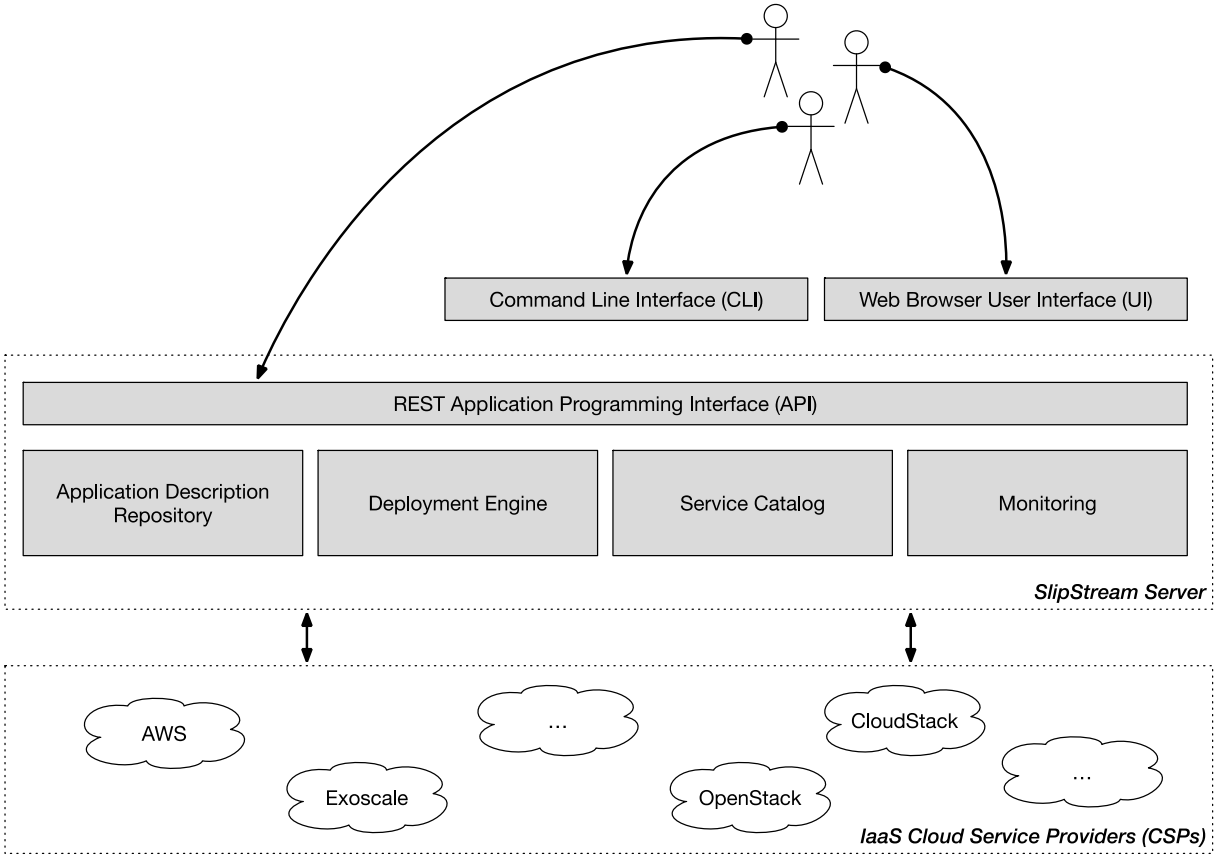

Figure 2: Functional Blocks of SlipStream

Compliance Analysis for Bioinformatics Applications in CYCLONE

Yuri Demchenko Senior Researcher (UvA)
CYCLONE WP2 Leader

Next to the general IT infrastructure compliance and data protection standards and regulations, there are standards related to the healthcare and genomic research, such as HIPAA/HITECH, Genetic Information Nondiscrimination Act (GINA), and Genomic Data Sharing (GDS). We provide here more details about the HIPAA/HITECH rules that form a basis for Protected Health Information (PHI) or Personally Identifiable Information (PII). We also explain mapping of different industry related standards to the general cloud compliance guidelines and best practices by Cloud Security Alliance (CSA).

Compliance and Security

Compliance and security are related and in some cases interchangeable. ***Security*** is commonly defined as a set of technical, physical, and administrative controls in order to ensure normal operation of a system or application. ***Compliance*** is a certification or confirmation that the system or an organization meets the requirements of the specified standards, established legislation, regulatory guidelines or industry best practices that can be jointly defined as compliance framework.

A compliance framework can include business processes and internal controls the organization has in place to adhere to these standards and requirements. The compliance framework should also map different requirements to internal controls and processes to eliminate redundancies.

Why is compliance important for a cloud environment? When moving to the cloud, the organization moves from an internal security and operational environment (that may not be formally defined) to the external operational security that will become a part of the SLA with the Cloud Service Provider (CSP). Compliance in this case will define the expected level of security and the assurances given by the CSPs.

Although the main IT and security standards are already adopted by the cloud community, the challenge to achieve the compliance for cloud based applications and/or solutions remains, due to the following:

- Existing audit requirements are not designed for virtualized distributed environment.
- Lack of visibility in cloud; large CSP such as Amazon and Google are “walled/curtained gardens”. Requirements to allow CSP audit may involve Non-Disclosure Agreement (NDA) and risk of provider lock-in.

It is a common practice in cloud security that CSPs implement Shared Responsibility Model that splits responsibility for the security of different layers and components between CSP and a customer. For example, Amazon Web Services (as an IaaS cloud provider) ensures the security of the cloud infrastructure and cloud platform services: facilities, physical security of datacenter, network infrastructure, virtualization platform and infrastructure. The customer is responsible for security of the following components: Amazon Machine Instances (AMI), OS, and applications, data in transit, data at rest, and data stores, credentials, policies and configurations. The customer is specifically responsible to comply with the Acceptable Use Policy (AUP), ensure correct use of the cloud platform, and for the security updates and patches of the guest OS and installed applications. Similarly, when developing cloud based applications, the applications developer must analyze and ensure compliance of the end user applications with the industry related compliance requirements.

Data security and protection is also a shared responsibility that involves

- The cloud provider’s responsibility to ensure secure data storage, processing and transfer and well as provide necessary security mechanisms to enable application level security.
- The application developer’s responsibility to correctly implement the application security in the cloud multi-tenant virtualized environment (often referred to as Security Development Lifecycle and defined by a number of industry standards and guidelines) to protect user data and personal information, integrate applications security with the provided cloud platform security services and mechanisms, and provide necessary and easy usable security services for end user to correctly use application security.
- The end user’s responsibility to ensure the security of their application access client (typically browser with hosting OS), access credentials and data.

Case study: HIPAA/HITECH compliance and genomic research

The HIPAA Security Rule contains the standards that must be applied to safeguard and protect electronic Protected Health Information (ePHI) or Electronic Health Record (EHR) when it is at rest and transferred. The rules apply to anybody or any system that has access to confidential patient data or personal identifiers which reveal the identity of an individual. There are three parts of the HIPAA Security Rule: technical safeguards, physical safeguards and administrative safeguards, which are presented in a form of HIPAA compliance checklists, see HIPAA Compliance Checklist, http://www.hipaajournal.com/hipaa-compliance-checklist/ and HIPAA Security Checklist, http://www.hipaaone.com/wp-content/uploads/2013/05/HIPAA-Security-Checklist.pdf.

At the moment, HIPAA-related regulations are organised in a form of the HIPAA Omnibus Rule that includes few additional documents and rules to the original HIPAA, as, for example, Health Information Technology for Economic and Clinical Health (HITECH).

The Technical Safeguards concern the technology that is used to protect ePHI and provide access to the data. The general requirement is that ePHI, whether at rest or in transit, must be encrypted according to the NIST standards once it travels beyond an organization's internal firewalled servers. Thereafter organizations are free to select whichever mechanisms they find the most appropriate, in order to:

- Implement means of access control that include assigning a centrally-controlled unique username and a PIN code for each user, but also establishing procedures to govern the release or disclosure of ePHI during an emergency.
- Introduce a mechanism to authenticate ePHI in order to confirm whether ePHI has been altered or destroyed in an unauthorized manner.
- Implement tools for encryption and decryption to ensure that devices used by authorized users have the functionality to encrypt and decrypt messages.
- Introduce activity audit controls to register attempted access to ePHI and record what is done with that data once it has been accessed.
- Facilitate automatic logoff after a pre-defined period of time. This prevents unauthorized access of ePHI should the device be left unattended.

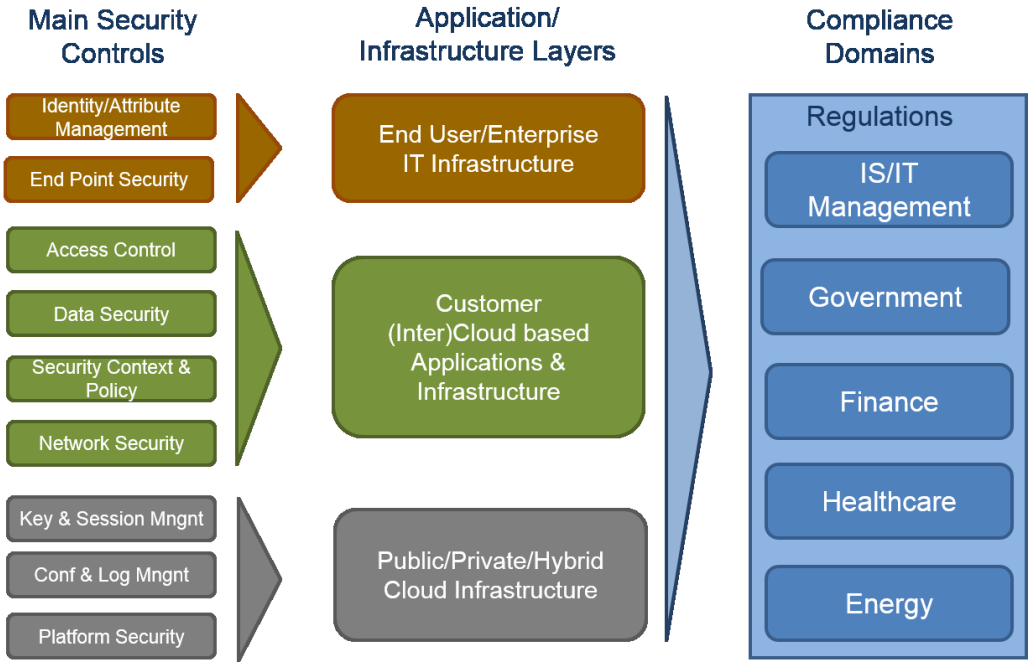

Figure 3: Relations between security controls, customer applications infrastructure and the main compliance domains

In the US, genomic data privacy and security is regulated by the National Institute of Health (NIH) Genomic Data Sharing (GDS) policy that addresses both the submission of genomic data by NIH-funded researchers to an NIH repository and the subsequent access and use of that data by either investigator. The researchers conducting a human genomic research must adhere to the Common Rules and have their research approved by an Institutional Review Board (IRB). GDS requires that informed consent is obtained from the genomic research participants. The investigators that want to download controlled access data from a NIH data repository must sign a Data Use Certification Agreement and abide by the NIH Genomic Data User Code of Conduct.

Mapping between HIPAA/HITECH safeguards and CSA Cloud Security controls

Clouds are the environment for processing genomic data, in particular when using Next Generation Sequencing (NGS) algorithms that allow for parallelisation and scalability. However, a cloud platform must be checked for compliance with the security, organisation and auditing regulations and requirements that are expressed in a form of controls, safeguards or rules. The biomedical data protection is regulated by HIPAA/HITECH, GINA, GDS and related regulations. The compliance to these regulations is not sufficient as they are strongly founded on more general IT and Information Systems security.

The main tool to assess cloud platform compliance for general and specific security and operational requirements is the CSA's Consensus Assessments Initiative Questionnaire (CAIQ) that includes the Cloud Controls Matrix (CCM) and is a part of the CSA

GRC Stack. CAIQ provides mapping of the CCM cloud security controls to 32 different industry specific standards, regulations and guidelines, in particular HIPAA/HITECH that is presented by administrative, physical and security safeguards.

HIPAA/HITECH-defined administrative, physical and security safeguards are mapped to the following CAIQ question groups:

- Application & Interface Security (Application Security, Data Integrity)
- Audit Assurance & Compliance (Audit Planning, Independent Audit)
- Business Continuity Management & Operational Resilience (Business Continuity Planning)
- Data Security & Information Lifecycle Management (E-commerce Transactions, Ownership/stewardship, Secure disposal)
- Datacenter Security (Asset Management, Offsite authorization, Policy, User access)
- Encryption & Key Management (Key Generation, Encryption)
- Governance and Risk Management (Risk Assessments, Management Program, Policy enforcement and review)
- Human resource (multiple controls)
- Identity & Access Management (User Access Policy, Segregation of duties, User Access Restriction/Authorization, User ID credentials and revocation)
- Infrastructure & Virtualization Security (Audit Logging / Intrusion Detection, Segmentation, Wireless security)
- Security Incident Management, E-Discovery & Cloud Forensics (Incident Management, Incident Response Legal Preparation)
- Threat and Vulnerability Management (Antivirus / Malicious Software, Vulnerability / Patch Management)

The presented list of security related questions may represent requirements and guidelines for application developers to develop consistent security services which must be integrated with cloud provider platform security as well as user or customer site/organisational security and facilities.

The CYCLONE consortium

Interoute S.p.A. Italy, www.interoute.com

Fundacio Privada I2CAT, Internet I Innovació Digital a Catalunya Spain, www.i2cat.es/en/

Centre National de la Recherche Scientifique France, www.cnrs.fr

QSC AG Germany, www.qsc.de

SixSq Sàrl, Switzerland, www.sixsq.com

Technische Universität Berlin, Germany, www.tub.de

Universiteit van Amsterdam, The Netherlands, www.uva.nl

Project coordinator
Matteo Biancani
Interoute S.p.A.
Tel: +39 (0)6 61 52 40 33
Fax: +39 (0)6 61 52 40 99

Technical coordinator
Eduard Escalona
i2CAT

European Commission

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 644925