Rest Assured

SECURE DATA PROCESSING IN THE CLOUD

Deliverable D3.1

D3.1 - Initial High Level Architecture

Release 1.0

The research leading to these results has received funding from the European Community’s H2020 research and innovation programme under grant agreement n° 731678.

Project Number	:	731678
Project Title	:	RestAssured - Secure Data Processing in the Cloud
Deliverable Type	:	External Report

Title of Deliverable	:	D3.1 - Initial High Level Architecture
Nature of Deliverable	:	External Report
Dissemination Level	:	Public
Contractual Delivery Date	:	30 April 2017
Actual Delivery Date	:	7 May 2017
Contributing WPs	:	All
Editor(s)	:	Eliot Salant

Contributors

Name	Organization	Sections
Andreas Metzger	UDE	Sections 1–9
Zoltan Mann	UDE	Sections 3.1, 5
Stefan Schoenen	UDE	Sections 3.1, 5, 7
Paul Mundt	Adaptant	Sections 3.3.2, 3.3.3
Adam Burns	Adaptant	Sections 3.3.2
Mike Surridge	IT Innovation	Sections 4–5
Ludger Goeke	UDE	Sections 4
Maritta Heisel	UDE	Sections 4
Nazila Gol Mohammadi	UDE	Sections 4
Pete Maynard	IT Innovation	Sections 4
Eliot Salant	IBM	Sections 6

Document History

Version	Date	Comment
V0.1	28.02.2017	Initial draft based on Kickoff meeting discussions
V0.2	29.03.2017	Refined draft based on input from WPs and confcall
V0.4	12.04.2017	Refined draft based on input from WPs and confcall
V0.8	19.04.2017	Refined draft based on comments received
V0.8	28.04.2017	Refined draft after architecture face-2-face meeting
V0.9	02.05.2017	Internal review version
V1.0	07.05.2017	Version submitted to EC

Contents

List of Figures

List of Tables

1 Introduction

Secure cloud computing is key for business success and end user adoption of federated and decentralized cloud services, and as such, is essential to stimulate the growth of the European Digital Single Market whilst ensuring compliance with the EU General Data Protection Regulation (GDPR).

RestAssured aims to provide technical solutions that help the *Data Controllers* in their responsibility in complying with the EU GDPR. A Data Controller is the entity legally responsible for determining the purposes and means of the processing of personal data, whilst ensuring compliance with data protection requirements (see the glossary in Section 8). Indirectly, this may mean that RestAssured will also support *Data Processors* (see Section 8), as – in agreement with Data Controllers – Data Processors perform data processing tasks.

Providing data protection and conformance to emerging digital privacy directives in a cloud environment is a challenging task, due to factors such as the inherently untrusted nature of a public cloud, the geographic distribution of the cloud, and multi-stakeholder systems, where the data belongs neither to the cloud service provider nor to the stakeholder who orchestrates the computation, as well as the highly dynamic changes in cloud services and infrastructures. These concerns mean that privacy and security by design approaches will no longer be sufficient due to uncertainty at design time of how the cloud and privacy requirements may dynamically evolve and change at run time. Therefore, RestAssured is enhancing security-by-design software architectures with novel mechanisms and cloud components for the runtime detection, prediction and prevention of data protection violations.

RestAssured will provide solutions to the specific technical concerns of data protection in the cloud through decentralized data life cycle management, secure enclaves through emerging hardware features, and runtime and predictive enforcement of security policies. In particular, RestAssured builds on recent breakthroughs in secure computation to provide data security and privacy in the cloud. RestAssured is combining four pillars of innovation for its holistic solution of data protection (see Figure 1.1):

- The use of emerging hardware solutions such as Intels SGX to provide secure enclaves for data operations.
- The implementation of sticky policies which will define data access, usage and storage rules.
- Models@runtime aggregate both runtime monitoring and context data to provide an ongoing analysis of data protection compliance in the running system.
- Automated risk management to automatically detect risks to data protection and rapidly determine the cost vs. benefits of alternative protection mechanisms.

This deliverable reports on the results achieved by WP3 “Architecture, platform & methodology” by Month 4 of the project.

WP3 has thee main aims.

- First, WP3 aims to develop the conceptual and technical *RestAssured architecture* to be made publicly available for the implementation of its solution. The development of this architecture will be an iterative process, involving yearly cycles of requirement collection, architectural design, low level design, implementation, testing, and evaluation feedback.
- Second, WP3 will be in charge of setting up and maintaining the *RestAssured testbed*. The testbed will represent the deployment of the prototypical implementation of the RestAssured platform and technical components. Following the RestAssured architecture as reference, this task will integrate the technical solution components from WP4–7.

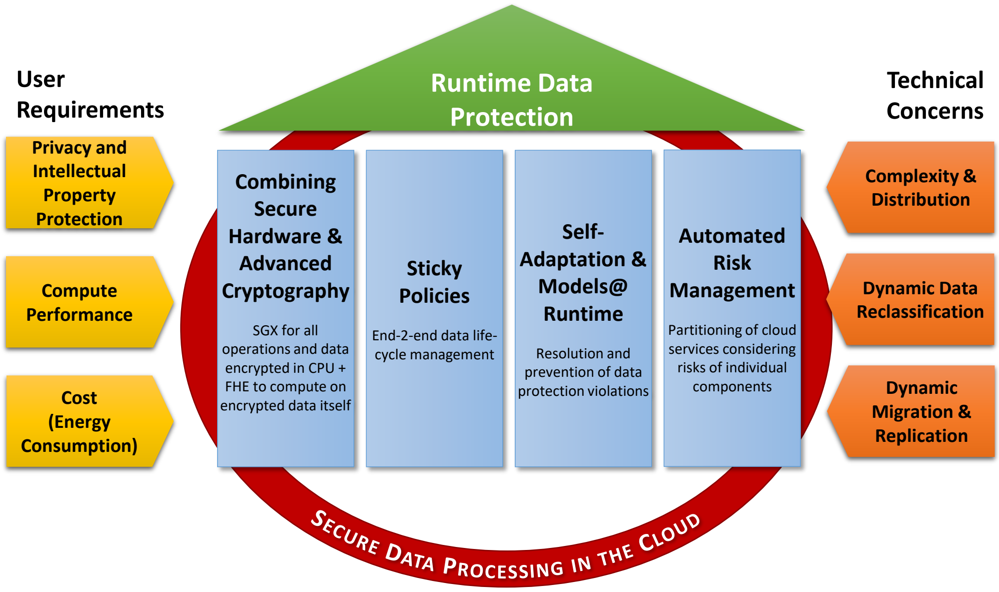

User Requirements
Privacy and Intellectual Property Protection
Compute Performance
Cost (Energy Consumption)
Runtime Data Protection
Combining Secure Hardware & Advanced Cryptography
SGX for all operations and data encrypted in CPU + FHE to compute on encrypted data itself
Sticky Policies
End-2-end data life-cycle management
Self-Adaptation & Models@ Runtime
Resolution and prevention of data protection violations
Automated Risk Management
Partitioning of cloud services considering risks of individual components
Technical Concerns
Complexity & Distribution
Dynamic Data Reclassification
Dynamic Migration & Replication
Secure Data Processing in the Cloud

Figure 1.1: RestAssured Innovation Pillars

- Third, to support both internal alignment on conceptual and procedural concerns of the RestAssured solutions, as well as foster training and uptake of the RestAssured solutions, WP3 will define an overall *RestAssured methodology*. The methodology will consider the complete life-cycle from requirement elicitation, over architectural design, implementation and testing. This life-cycle especially considers quality requirements, internal technical interfaces as well as internal organizational interfaces, and verification activities.

According to the RestAssured timeline, at Month 4 of the project, the initial High Level RestAssured architecture and the design of the RestAssured testbed are defined. As such, this deliverable focuses not only on reporting on these two achieved results, but also gives a first outline of the RestAssured methodology.

The remainder of this deliverable is structured as follows. Section 2 provides an overview of the RestAssured architecture, including its four main views. Sections 3–6, accordingly describe the initial versions of the four main architectural views and how they relate to each other[1]. Section 7 describes the design of the RestAssured testbed. Finally, Section 8 provides a glossary of the key terms and concepts used in this architecture deliverable.

[1] Section 4 includes the outline of the RestAssured methodology

2 Overview of the RestAssured High-Level Architecture (HLA)

The RestAssured High-Level Architecture (HLA) provides a common frame of reference for the development as well as the use of the RestAssured technical building blocks, as developed in WP4–7.

The development of the RestAssured HLA is an iterative process, involving periodic (yearly) cycles of requirement collection, architectural design, low-level design, implementation, testing, and evaluation feedback. Within these longer cycles, shorter increments and refinements of the RestAssured HLA are planned in order to ensure that technology insights (opportunities but also limitations) from creating the technology building blocks are adequately considered. In addition, these cycles facilitate taking into consideration recent developments in technology and the state of the art.

The RestAssured HLA determines the principal functionality for all technical building blocks. It guides the definition of interfaces (APIs) based on a clear understanding of dependencies between the technical building blocks. In particular, the RestAssured HLA is defined using four dedicated, purposefully chosen, complementary architectural views (see Figure 2.1 for an overview). The RestAssured HLA views address different levels of abstraction (conceptual vs. technical) and different life-cycle phases (design-time vs. run-time).

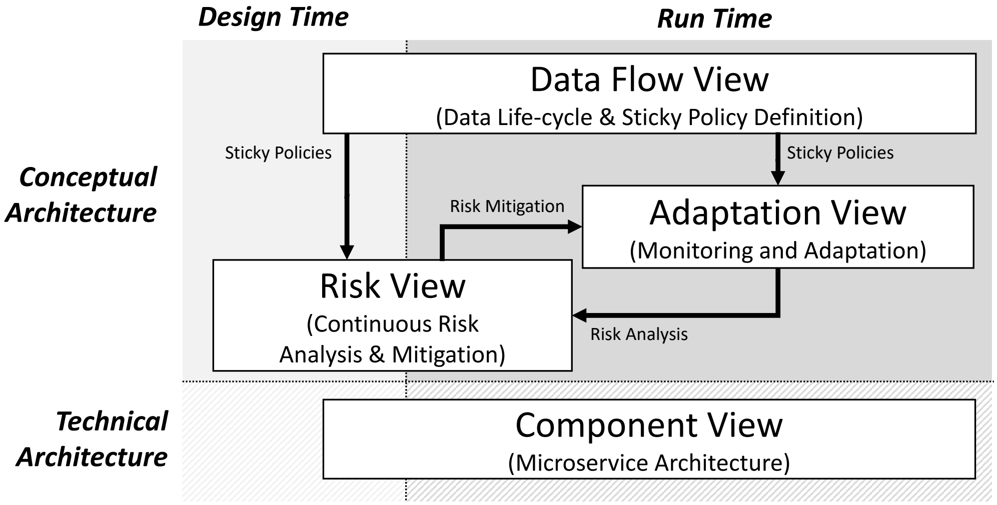


Figure 2.1: Views of RestAssured High Level Architecture

In particular:

- **Data Flow View (Section 3):** This view defines the principal types of secure cloud data processing chains and how they will be supported by the RestAssured technical solutions, taking into account the different stakeholders and their trustworthiness. The Data Flow View is technology-agnostic (the realization of the RestAssured solutions will be defined in other views). The Data Flow View is of particular importance to understand the situation before a RestAssured solution was implemented and how RestAssured can address the issues in current cloud computing scenarios. In particular, the three use cases of RestAssured (WP8) will contribute their relevant scenarios in the form of data flow views, such as to serve as a concrete instantiation of this view and also to serve as "storyboards" for the RestAssured solution demonstrations. In addition, this view will introduce the notion of sticky policies, i.e., policies attached to the data elements, which facilitate observing and managing data protection constraints.

- **Risk Analysis View (Section 4):** This view defines the main (implementation-agnostic) elements of a cloud stack and its applications such as to serve as an input for risk analysis and decision making (WP7). The Risk Analysis View will in particular consider relevant context information and stakeholders that may interact with the cloud system. The Risk Analysis View will be continuously updated (e.g., based on monitoring information channeled via the Adaptation View below).
- **Adaptation View (Section 5):** This view will define the main (implementation-agnostic) elements of a cloud stack and its applications to serve as an abstract representation of the cloud configuration at run time (aka. model@runtime). The main purpose of the Adaptation View is to identify the information that should be maintained by the RestAssured run-time data protection solutions (WP5), to determine, plan and enact adaptations to prevent and counter-act data protection violations. The elements will be contributed from all technical pillars of the RestAssured solutions, i.e. covering WP4-WP7. The Model@runtime View and the Risk Analysis View will be tightly linked, such as to support continuous risk management and mitigation.
- **Component View (Section 6):** This view defines the concrete technical components that will constitute the RestAssured solution, building on the secure cloud data processing environment (WP4) and involving components for run-time data protection assurance (WP5), decentralized data life-cycle management (WP6) and continuous risk assessment and mitigation (WP7).

3 Data Flow View

RestAssured is targeted to address data protection risks in the cloud. To provide a tangible understanding of such potential risks, we start with presenting an abstract data flow through the components and infrastructure containers involved in a typical data processing chain in the cloud. The abstract data flow has purposefully been chosen in such a way as shown to cover a high number of potential data protection risks. The following sub-sections instantiate the Data Flow View for the different use cases addressed in RestAssured.

3.1 Abstract Data Flow View

This section first gives an abstract data flow through the components and infrastructure containers involved in a typical data processing chain in the cloud. In Figure 3.1, we sketch these elements along the data flow of this system, as well as the key stakeholders and their trustworthiness. Trustworthiness leads us to the notion of trusted actors ("white hats") and non-trusted actors ("black hats"); cf. Section 8. The data flow is initiated by a trusted entity and ends with a trusted entity, but may flow through components offered by untrusted entities.

1. The data flow starts with the Data Subject providing his/her data to a Data Controller A.
2. Data Controller A stores the Data Subject's data in a cloud-hosted data base.
3. A software Component A, hosted in an untrusted cloud infrastructure (IaaS) accesses data from the aforementioned data base. Component A is developed and maintained by Data Controller A.
4. Data further flows from Component A to a Component C, which is developed and maintained by an untrusted SaaS Cloud Provider Z.
5. From Component C, data flow reaches the Data Consumer.

In addition, the Data Controller A may deploy a Component B running on the infrastructure of a different Cloud Provider Y. It should be noted that the infrastructure of Cloud Provider Y resides in a non-EU location (such as in the US).

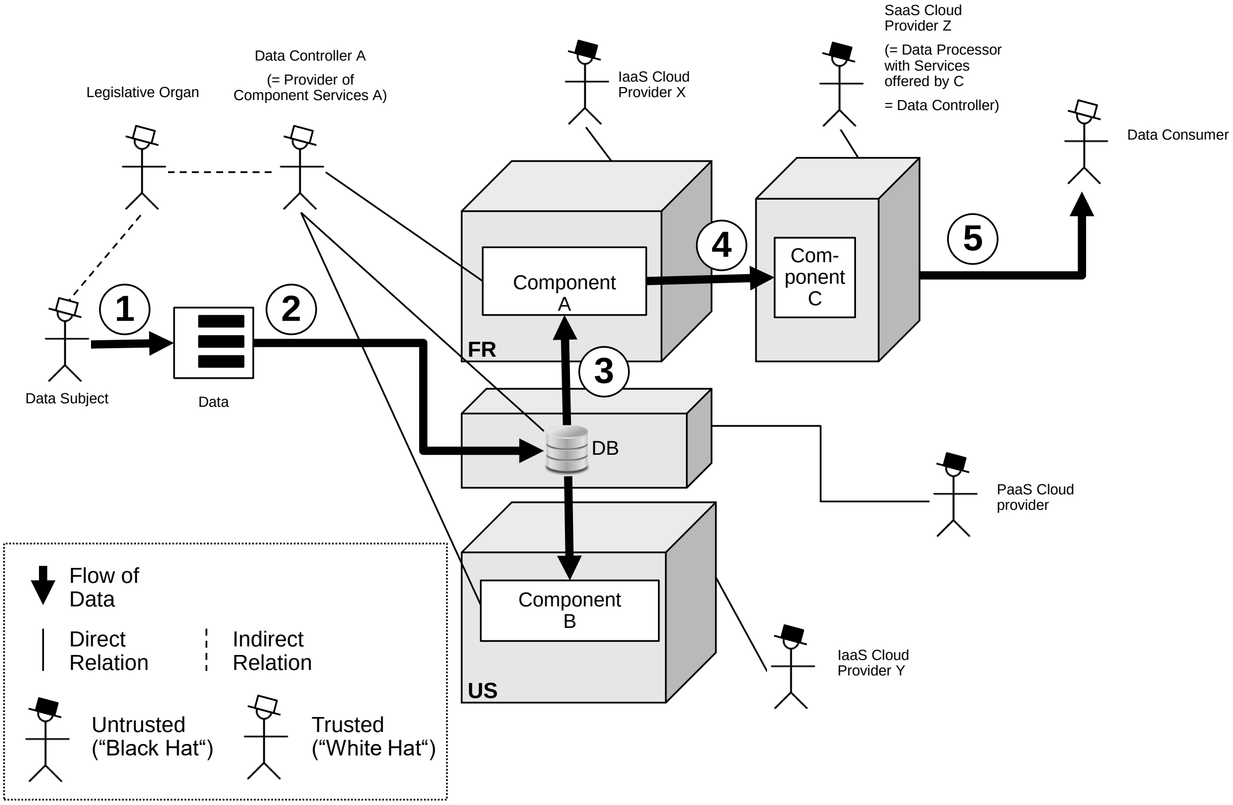


Figure 3.1: Situation without Data Protection Measures

Due to the multi-stakeholder and distributed characteristics of the cloud, the data flow example in Figure 3.1, may lead to different data protection risks if no specific measures are of taken. Figure 3.2 indicates typical risks that may be faced in case the cloud system is deployed on the untrusted elements as indicated above, while not taking any data protection measures. It should be noted that the situation has purposefully been chosen to shown a high number of possible data protection risks.

The architecture and deployment situation explained above, exposes several data protection risks. In particular, the following risks are shown Figure 3.2:

- If data is deployed in non-secure DB, it may be compromised.
- If a software component and thus its data is deployed on a non-secure infrastructure, access to its data may be gained or the code and data may be compromised, e.g., by another tenant on the same infrastructure or even the untrusted infrastructure provider.
- If data is transferred to an untrusted SaaS provider, data protection may be at risk.
- A new Component B is deployed in a non-EU geo-location. In such a case, if Component B aims to access data, this access (and thus data flow to a non-EU geo-location) would violate GPDR geo-location policies.

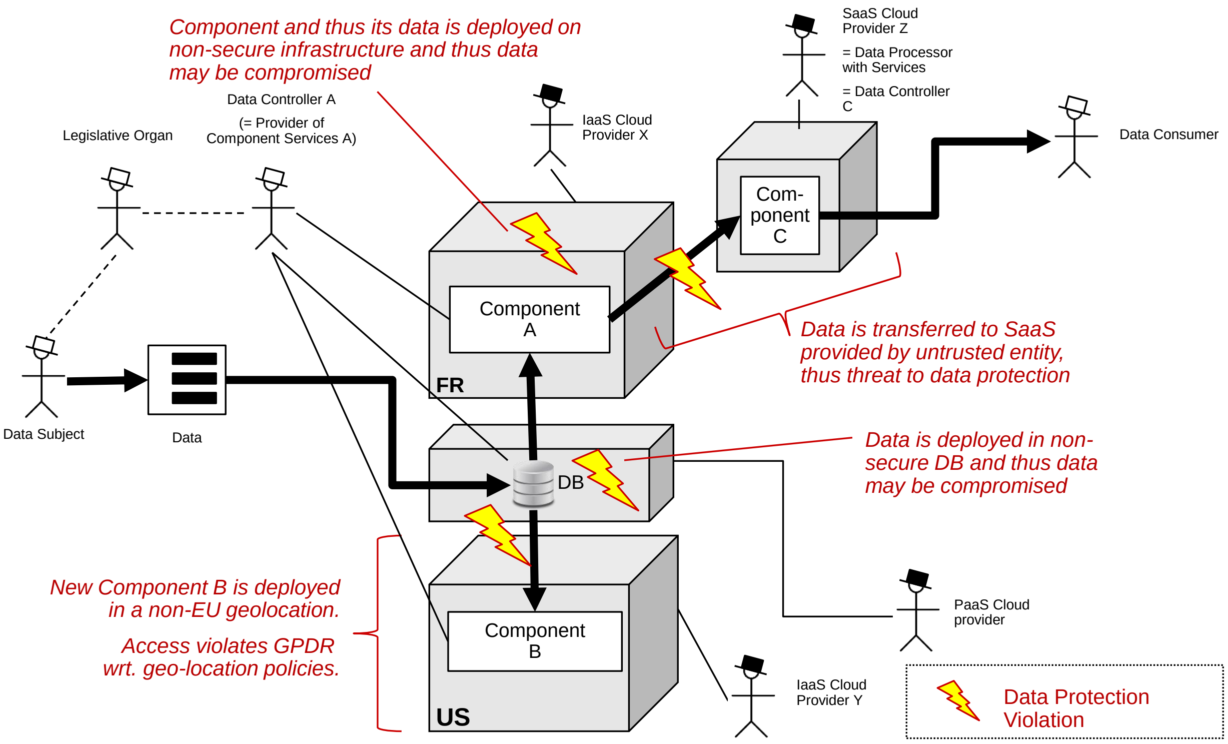

Component and thus its data is deployed on non-secure infrastructure and thus data may be compromised
Data Controller A
(= Provider of Component Services A)
Legislative Organ
IaaS Cloud Provider X
SaaS Cloud Provider Z
= Data Processor with Services
= Data Controller C
Data Consumer
Com-ponent C
Component A
FR
Data Subject
Data
DB
Data is transferred to SaaS provided by untrusted entity, thus threat to data protection
Data is deployed in non-secure DB and thus data may be compromised
New Component B is deployed in a non-EU geolocation.
Access violates GPDR wrt. geo-location policies.
Component B
US
PaaS Cloud provider
IaaS Cloud Provider Y
Data Protection Violation

Figure 3.2: Potential Risks stemming from Situation without Data Protection Measures

How these risks may be addressed when the RestAssured solutions are in place is shown in Figure 3.3. In particular, the yellow bubbles show the decisions that are taken as part of the RestAssured solutions in order to ensure data protection. It is important to note a fundamental assumption for RestAssured, which is that the decisions are taken by trusted entities (“white hats”). Only then can it be ensured that the data protection requirements are met.

Also, due to the implementation-agnostic characteristics of the data flow models, we do not show which components and which actors implement and take such decisions. The diagrams purposefully abstract from these implementation-level details. These details will be provided by the component view in Section 6.

In particular, the following decisions may be taken to address the aforementioned risks:

- By storing the Data in *CryptoDB*, the RestAssured solutions may make use of Homorphic Encryption. CryptoDB offers encrypted storage of data, and allows homomorphic SQL queries on the data base. This means that neither storage nor processing of data will happen in plain text, and thus plain text data cannot be accessed. Homomorphic encryption means that data processing is carried out directly on the encrypted data, i.e., without the need to first decrypt the data (and thus potentially exposing it to attacks).
- By *deploying component code in an SGX enclave*, the RestAssured solutions prevent other tenants, and even the administrator of the untrusted cloud infrastructure (in the scenario) to gain access to code or data. SGX is an intel hardware implementation of secure enclaves, which enable applications to define secure regions of code and data that maintain confidentiality even when an attacker has physical control of the platform and can conduct direct attacks on memory.
- *Data Access* in RestAssured will be governed by the concept of a Data Gatekeeper (see Section 6). The Data Gatekeeper will only allow access to data, provided that the data’s sticky data protection policy matches with the credentials of the requester. In our scenario, data access would only be granted if the requestor resides within the EU and thus would comply with the GDPR geo-location policy.

- Delivering data only in *anonymized* form may be another way that RestAssured ensures protection of sensitive data. In our scenario, anonymization prevents an untrusted service provider from gaining accesses to personal data.

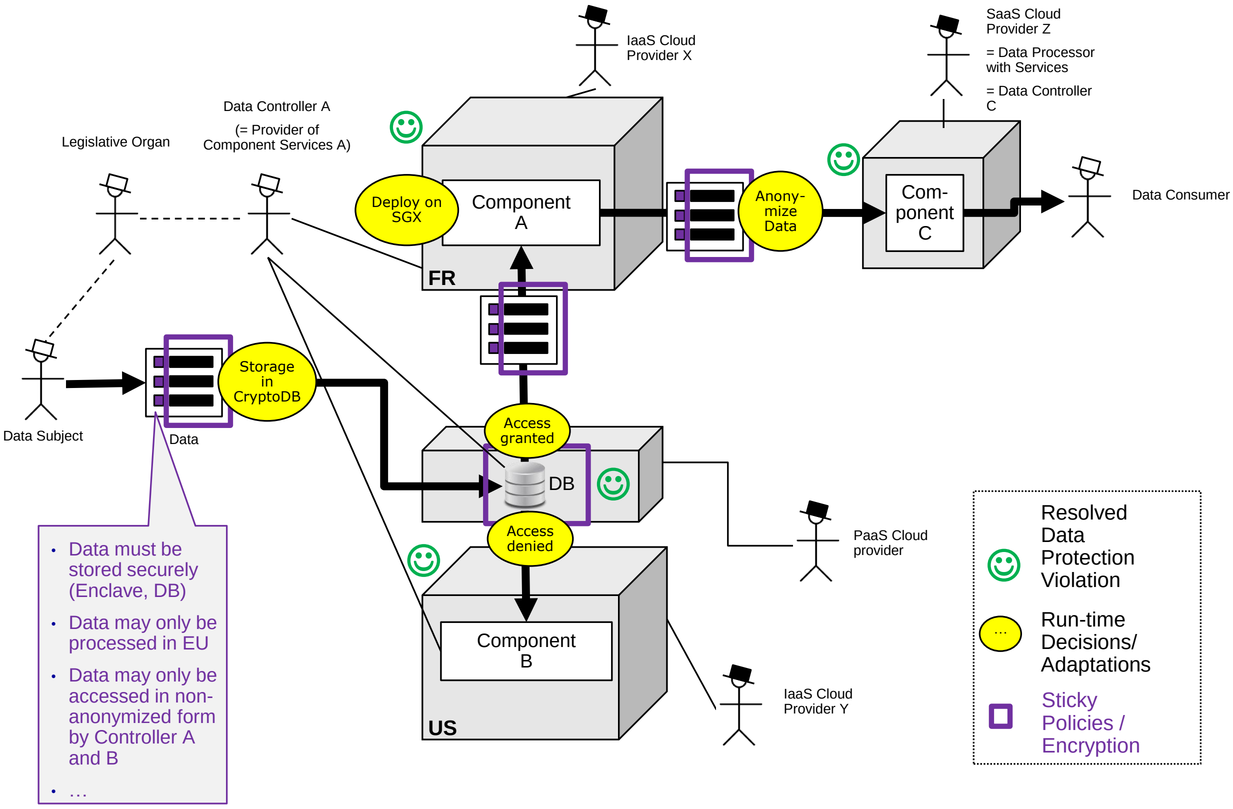

IaaS Cloud Provider X
SaaS Cloud Provider Z
= Data Processor with Services
= Data Controller C
Data Controller A
(= Provider of Component Services A)
Legislative Organ
Deploy on SGX
Component A
FR
Anony-mize Data
Com-ponent C
Data Consumer
Data Subject
Data
Storage in CryptoDB
Access granted
DB
Access denied
PaaS Cloud provider
Component B
US
IaaS Cloud Provider Y
Data must be stored securely (Enclave, DB)
Data may only be processed in EU
Data may only be accessed in non-anonymized form by Controller A and B
…
Resolved Data Protection Violation
Run-time Decisions/ Adaptations
Sticky Policies / Encryption

Figure 3.3: Addressing Risks via RestAssured Conceptual Solutions

An essential concept used to govern data protection and data access is the sticky policy concept. In a nutshell, a sticky policy comes attached to the data asset, and governs the access and allowed use of the data asset. Access to data is only granted in case the decision taken is compliant with the data protection policy. More details on the sticky policy approach are provided in Section 3.3.

3.2 Data Life-cycle and Sticky Data Policies

Figure 3.4 illustrates the steps and roles that are required to successfully implement the data lifecycle taking into account data security and data protection concerns. Data security has to be analysed at the creation of data. This analysis results in a classification of data and a set of data security requirements and privacy preferences that are related to all the steps of the data lifecycle (storage, usage, etc.). The data subject shares these requirements and preferences with the data controller. The data controller specifies the data security and sticky policies that are related to the data and plans the enforcement of these policies and the deployment of the data according to its classification. The data controller relies on a Service Registry that catalogues the various cloud services (IaaS, PaaS, and SaaS) and their security capabilities in order to select the services that comply with the data security policies. Once the security and sticky policies as well as the data deployment and security enforcement plans are agreed upon with the data subject, the data subject can store the data as recommended by the data controller. A data security check is needed at rest and during data processing to ensure that security enforcement being used is effectively addressing the security policies. If not, data deployment and security enforcement must be revised in order to adopt changes and to address new risks.

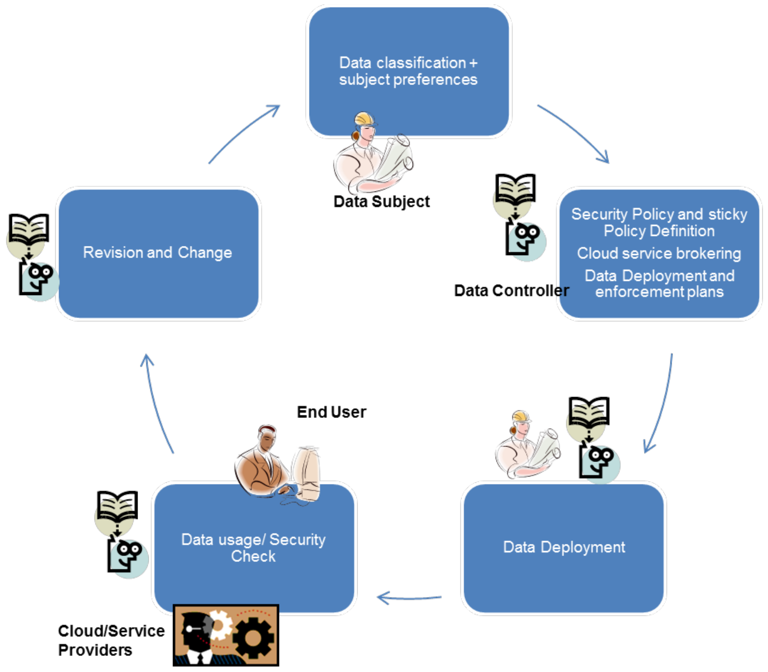
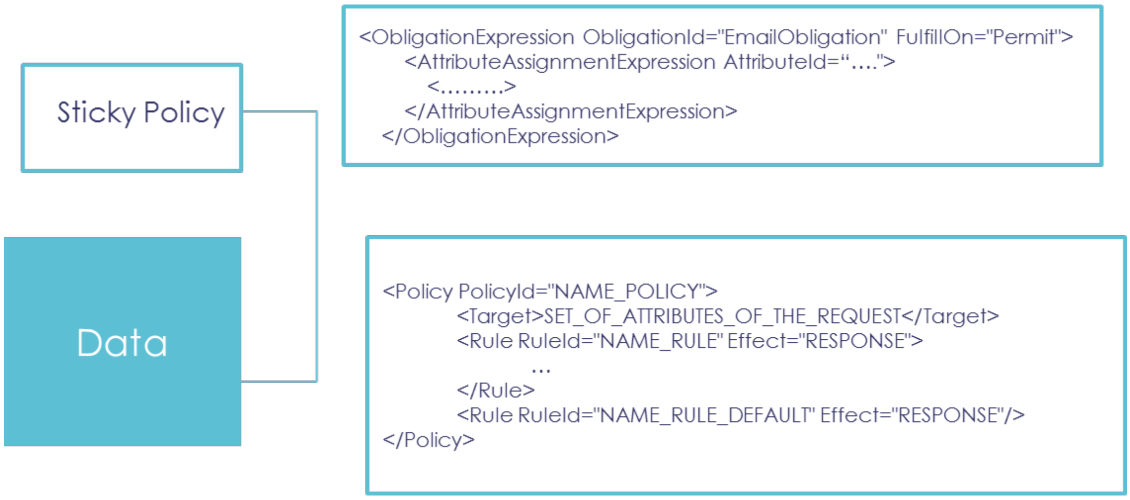

Data classification + subject preferences
Data Subject
Security Policy and sticky Policy Definition
Cloud service brokering
Data Deployment and enforcement plans
Data Controller
Revision and Change
End User
Data Deployment
Data usage/ Security Check
Cloud/Service Providers

Figure 3.4: Data Security Life-Cycle

The RestAssured solutions rely on the sticky policy concept to define and enforce data security and data protection requirements. A sticky policy (see Figure 3.5) is a security policy, which is associated with a piece of data such that access to and use of that data is only possible if the policy has been complied with. Sticky policies can be implemented in several ways.



```
<ObligationExpression ObligationId="EmailObligation" FulfillOn="Permit">
    <AttributeAssignmentExpression AttributeId="….">
       <………>
    </AttributeAssignmentExpression>
 </ObligationExpression>
```

```
<Policy PolicyId="NAME_POLICY">
         <Target>SET_OF_ATTRIBUTES_OF_THE_REQUEST</Target>
         <Rule RuleId="NAME_RULE" Effect="RESPONSE">
                  …
         </Rule>
         <Rule RuleId="NAME_RULE_DEFAULT" Effect="RESPONSE"/>
</Policy>
```

Figure 3.5: Structure of Sticky Policy

XACML (eXtensible Access Control Markup Language[1]) is an OASIS standard that defines an attribute-based access control policy language that can be used to specify the data security policies. A concept within XACML called obligations can be used to express and manage sticky policies. An obligation is a directive to the Policy Enforcement Point (PEP) on what must be carried out when an access is approved by a Policy

[1] see https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

Decision Point (PDP, according to the security policies).

An example of access control policy that that allows a doctor to read the medical records of his patient and an associated obligation that specifies a sticky policy of a patient that wants to receive an alert by email each time his data is accessed could look like this:

```
Allow access to resource MedicalRecord
  if Subject match DoctorOfPatient
  and action is read
  with obligation
  on Permit: SendEmailToPatient(patientID, Subject, time)
```

3.3 Data Flow Views of RestAssured Use Cases

RestAssured plans to deploy the RestAssured solutions across three distinct and diverse use cases, involving actual end-users, thereby demonstrating the broad applicability of the RestAssured solutions.

These uses cases cover different aspects of data protection (intellectual property vs. privacy), involve different forms of distributed cloud architectures (within data centre, federated, as well as along the whole compute continuum / fog). Table 3 provides a characterization of these use cases.

Table 3.1: RestAssured Use Cases and their Characteristics

Use Case	Data Protection Concern	Cloud Architecture	Involved Stakeholders
CARE: Self-directed Social Care	Privacy (sensitive personal identifiable data)	Public, decentralized cloud	Vulnerable adults living at home; social care providers
PAYD: Pay-as-you-Drive / Usage-based Insurance	Privacy (personal identifiable data)	“Fog” (sensors, embedded controllers and storage, edge devices, and cloud)	Citizens; car manufacturers and insurance providers
HPC: High-performance computing for Commercial Use	IPR (business sensitive data)	Data centre (possibly federated)	Commercial enterprises; High performance computing centres

3.3.1 CARE: “Self-directed Social Care” Use Case

The procurement and monitoring of social care is a complex interaction of several stakeholders sharing a citizens sensitive personal data; the service user, their family and carers, social care providers (for-profit companies and not-for-profit charities), and the Local Authority. (In the UK, a Local Authority or LA is the government organisation officially responsible for public services and facilities in a particular locality.) Information is usually stored by the organizations involved (usually the LA and service providers) on premise and is rarely placed in the cloud. Blanket privacy statements to which the citizen is assumed to have consented control the data. In general, local authorities and service providers are required to adhere to data protection laws with little specialist knowledge. The cultural distrust of cloud storage by stakeholders is an obstacle to the efficient sharing and use of personal data necessary to make self-directed support a reality. The RestAssured UK use case concerns the setting up of care to vulnerable adults living at home.

This use case describes the web-based application "Ami" that matches service providing volunteers with people requiring help (clients). Both the volunteers and the clients (on their behalf by third party volunteer organisations) provide personal data (i.e. are data subjects). The application matches the volunteer with clients based on their search criteria (location, interests etc.). When a volunteer finds someone that they'd like to help, the application will advise them which voluntary organisation that client is registered with and invite them to sign up with the same organisation, who will contact the volunteer directly to arrange an interview, security checks, references etc.

Third parties are also allowed to register with Ami to run reports relating to volunteer data, for example to identify client needs in specific areas. Only the data that volunteers have permitted to be shared should be passed to the third party.

The following data flow views explain the situation using the same format as the Abstract data flow views.

Figure 3.6 shows the current data flow for both a data subject (volunteer) registering with the service and data being received by the data consumers (Volunteer organisations, the client - via the organisation and third parties for reporting).

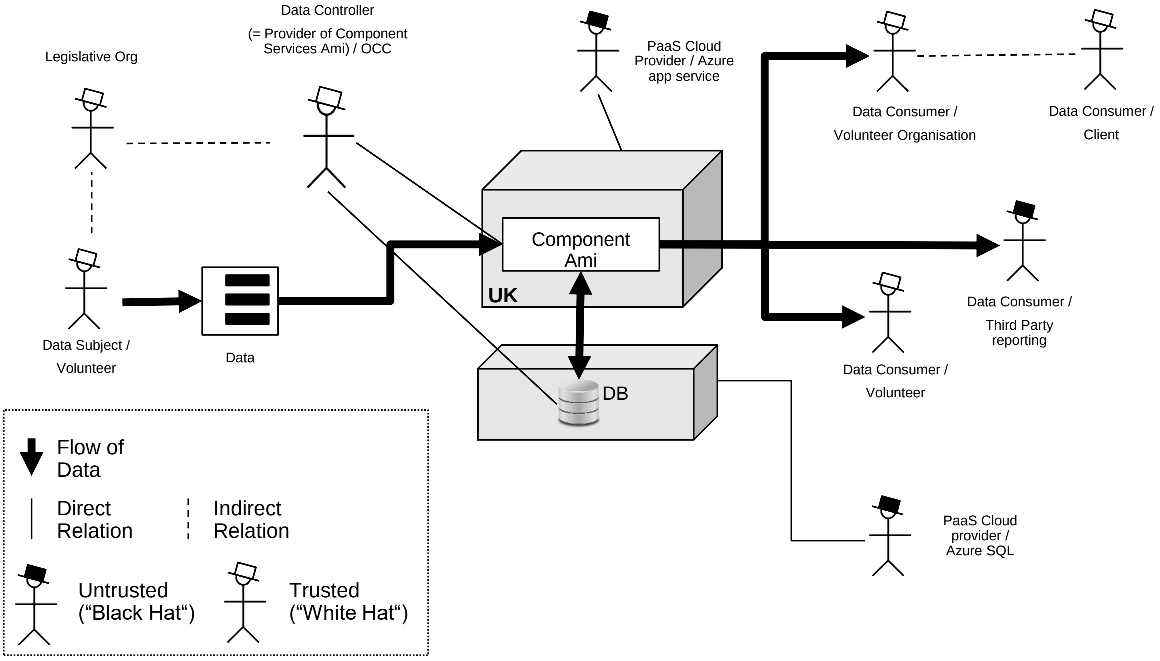

Data Controller
(= Provider of Component Services Ami) / OCC
Legislative Org
PaaS Cloud Provider / Azure app service
Data Consumer / Volunteer Organisation
Data Consumer / Client
Component Ami
UK
Data Subject / Volunteer
Data
DB
Data Consumer / Third Party reporting
Data Consumer / Volunteer
PaaS Cloud provider / Azure SQL
Flow of Data
Direct Relation
Indirect Relation
Untrusted ("Black Hat")
Trusted ("White Hat")

Figure 3.6: Ami current situation

Figure 3.7 below, highlights potential data protection risks for this scenario.

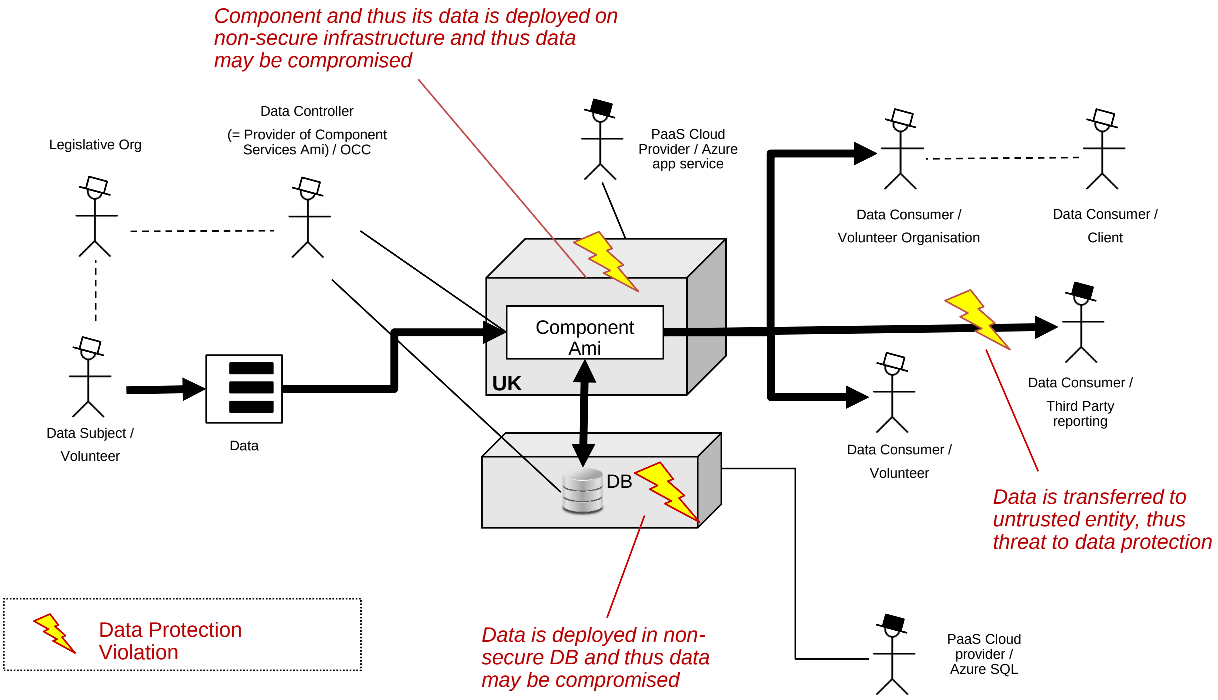

Component and thus its data is deployed on non-secure infrastructure and thus data may be compromised
Legislative Org
Data Controller
(= Provider of Component Services Ami) / OCC
PaaS Cloud Provider / Azure app service
Data Consumer / Volunteer Organisation
Data Consumer / Client
Component Ami
UK
Data Subject / Volunteer
Data
DB
Data Consumer / Third Party reporting
Data Consumer / Volunteer
Data is transferred to untrusted entity, thus threat to data protection
Data Protection Violation
Data is deployed in non-secure DB and thus data may be compromised
PaaS Cloud provider / Azure SQL

Figure 3.7: Ami potential risks

Data is deployed on non-secure infrastructure and thus there is a risk that data may be compromised by untrusted parties while it is stored or during transfer. There is also a risk associated with the transfer of data to the untrusted third party for reporting. Figure 3.8 shows the RestAssured solutions to these risks.

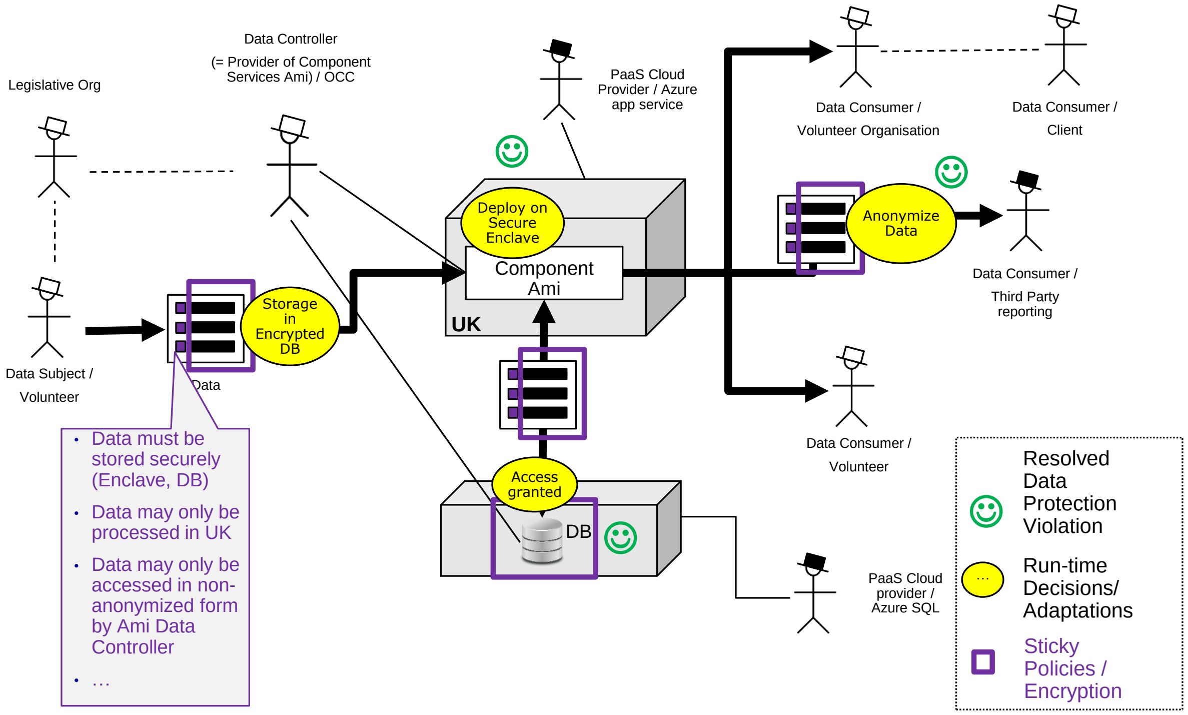

Legislative Org
Data Controller
(= Provider of Component Services Ami) / OCC
PaaS Cloud Provider / Azure app service
Data Consumer / Volunteer Organisation
Data Consumer / Client
Deploy on Secure Enclave
Component Ami
UK
Anonymize Data
Data Consumer / Third Party reporting
Storage in Encrypted DB
Data Subject / Volunteer
Data
Data must be stored securely (Enclave, DB)
Data may only be processed in UK
Data may only be accessed in non-anonymized form by Ami Data Controller
…
Access granted
DB
Data Consumer / Volunteer
PaaS Cloud provider / Azure SQL
Resolved Data Protection Violation
Run-time Decisions/ Adaptations
Sticky Policies / Encryption

Figure 3.8: RestAssured solutions

Through the use of sticky policies, data encryption and secure enclaves data protection violations are resolved.

***

A complementary use case in the same domain as the one described above is the Social Care Finance Portal (SCFP). The SCFP is an online portal for users of social care (clients) to manage payments for services to their local authority (LA). They can view their balance, payment history and debt notices. The service allows users of social care to interact with Local Authorities online. It also allows a representative, acting on behalf of the service user to have full control of their payments. The LA publishes the data to the cloud from their private infrastructure.

Figure 3.9 shows the flow of data from the local authority, which holds the care and financial data for the data subject (client), to the cloud hosted database. The data is then accessed through a cloud hosted application by the data consumer (the client or their representative).

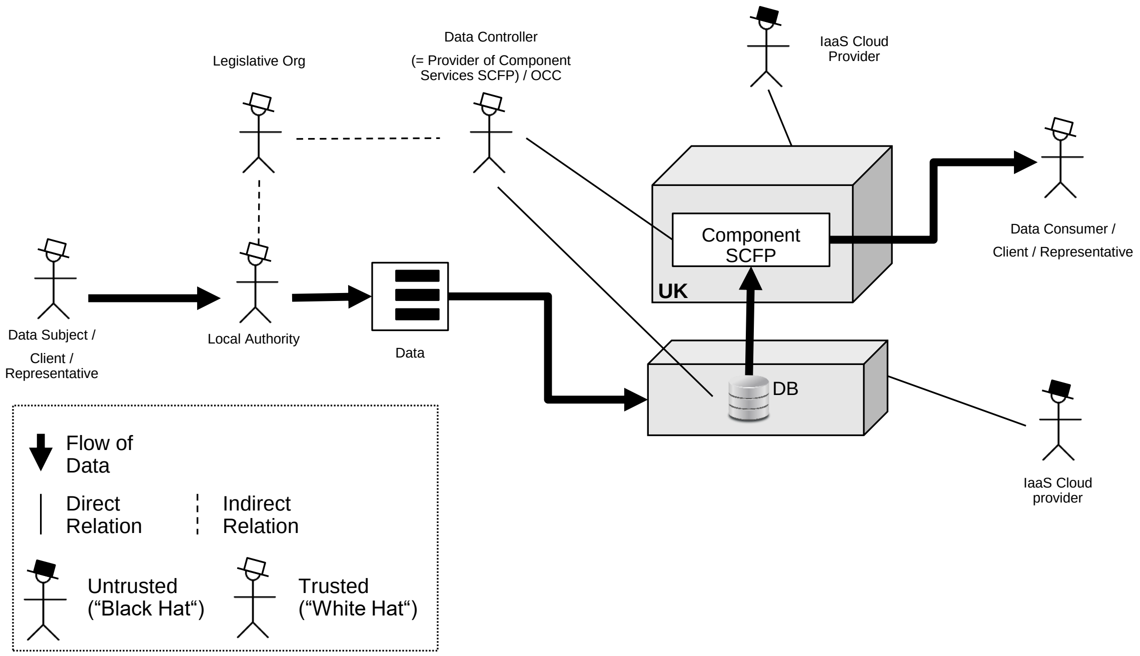


Figure 3.9: Current situation

The data protection risks for this use case are highlighted in 3.10

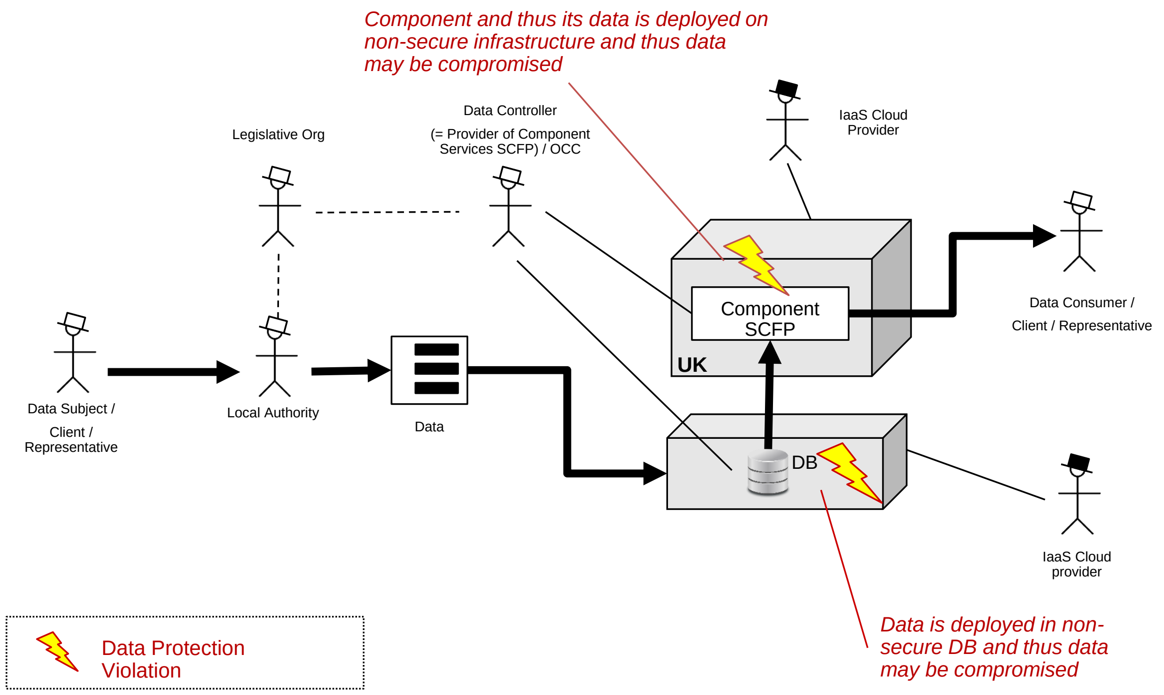

Component and thus its data is deployed on non-secure infrastructure and thus data may be compromised
Legislative Org
Data Controller
(= Provider of Component Services SCFP) / OCC
IaaS Cloud Provider
Component SCFP
UK
Data Consumer / Client / Representative
Data Subject / Client / Representative
Local Authority
Data
DB
IaaS Cloud provider
Data Protection Violation
Data is deployed in non-secure DB and thus data may be compromised

Figure 3.10: Potential risks

Data is deployed on non-secure infrastructure and thus there is a risk that data may be compromised by untrusted parties while it is stored or during transfer.

The RestAssured solutions to these data protection risks are detailed in 3.11

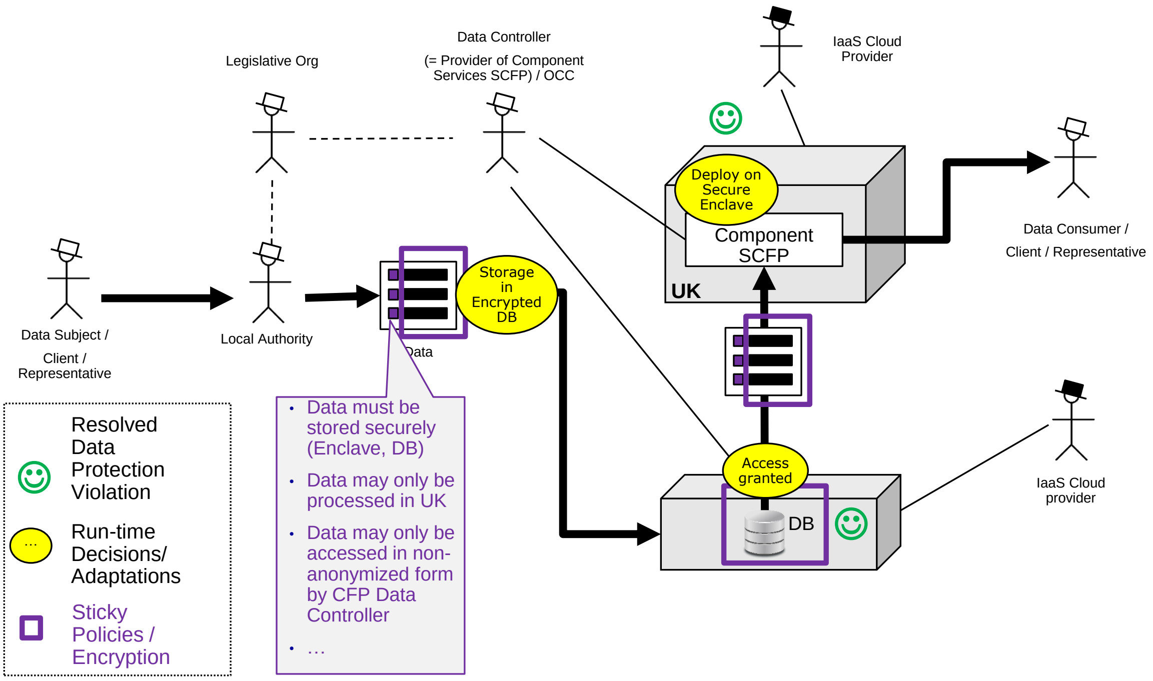

Legislative Org
Data Controller
(= Provider of Component Services SCFP) / OCC
IaaS Cloud Provider
Deploy on Secure Enclave
Component SCFP
UK
Data Consumer / Client / Representative
Data Subject / Client / Representative
Local Authority
Storage in Encrypted DB
Data
Access granted
DB
IaaS Cloud provider
Resolved Data Protection Violation
Run-time Decisions/ Adaptations
Sticky Policies / Encryption
Data must be stored securely (Enclave, DB)
Data may only be processed in UK
Data may only be accessed in non-anonymized form by CFP Data Controller
…

Figure 3.11: RestAssured solutions

Through the use of sticky policies, data encryption and secure enclaves data protection violations are resolved.

3.3.2 PAYD:"Pay as You Drive Insurance" Use Case

Through the combination of telematics and GPS systems data, insurance companies are able to shape a more holistic view of an individual driver's overall risk profile based on insights gleaned from empirical data analysis. While this approach has been successfully engaged in the marketplace in the US for some years under a more relaxed and homogeneous regulatory environment, European insurance and telematics companies have been hesitant to pursue this model without adequate safeguards put in place concerning data protection and privacy rights. With the emergence of IoT and the growth of vehicular sensor networks, this is likely to only be compounded, particularly in light of increased demand from individual drivers for the economic savings and benefits a usage-based insurance model can introduce. Furthermore, as the volume of the sensor data grows, an increasing amount of in-vehicle data processing is expected to take place, which itself becomes subject to privacy and usage control restrictions which heretofore have been left at the discretion of the telematics companies. As 5G begins to emerge and an always-on Connected Car becomes a reality, there will be an increasing push to move away from proprietary cellular connectivity solutions used in existing telematics scenarios today, and to-wards a more open and collaborative environment in which drivers' privacy rights are able to be more explicitly managed, while still drawing on the benefits of a growing IoT ecosystem.

The following data flow views show the current situation, potential risks for data protection and privacy violations, and the RestAssured solutions for this particular use case.

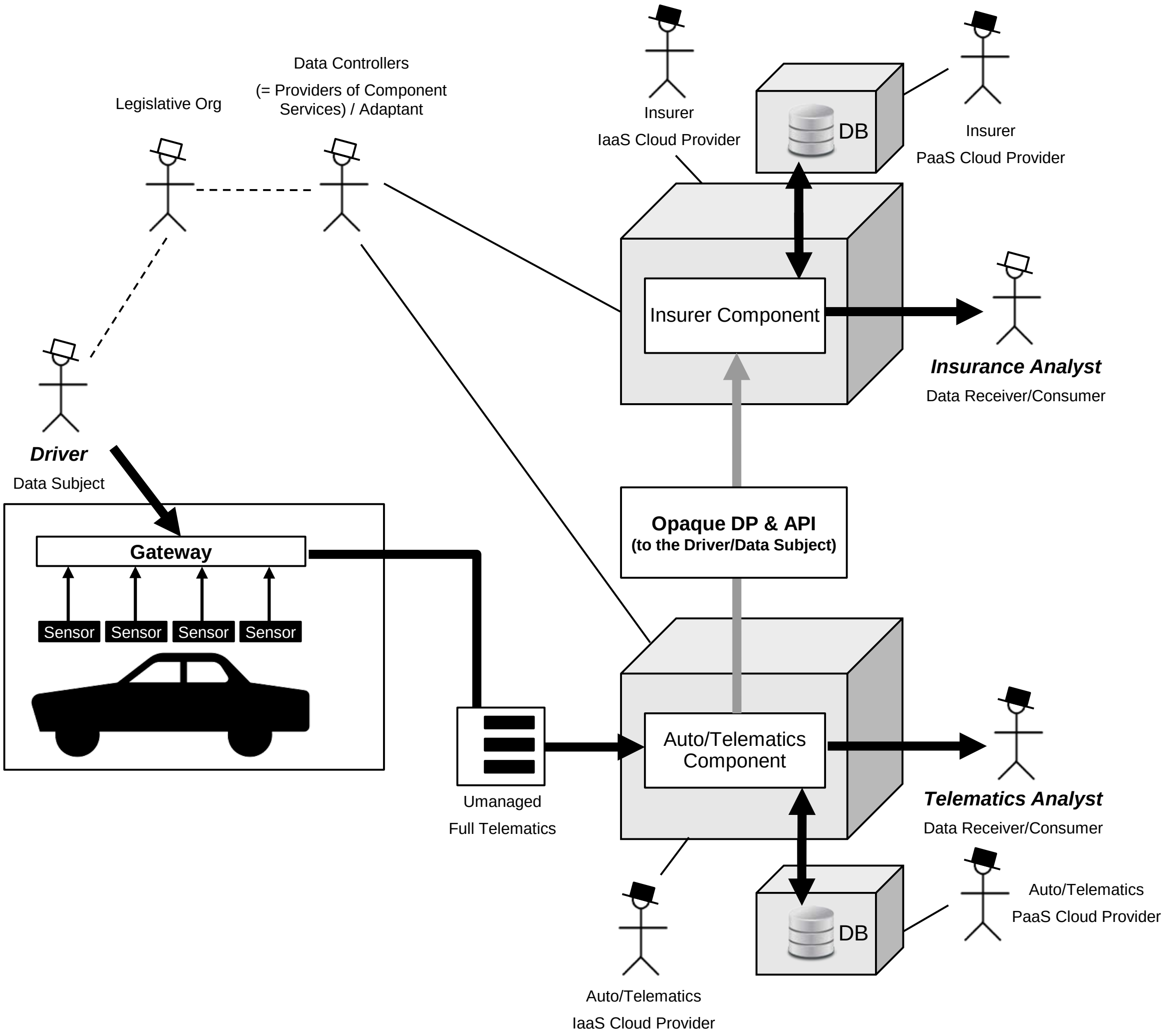


Figure 3.12: Pay-as-you-Drive Insurance - current situation

Figure 3.13 below, highlights potential data protection and privacy risks for this scenario.

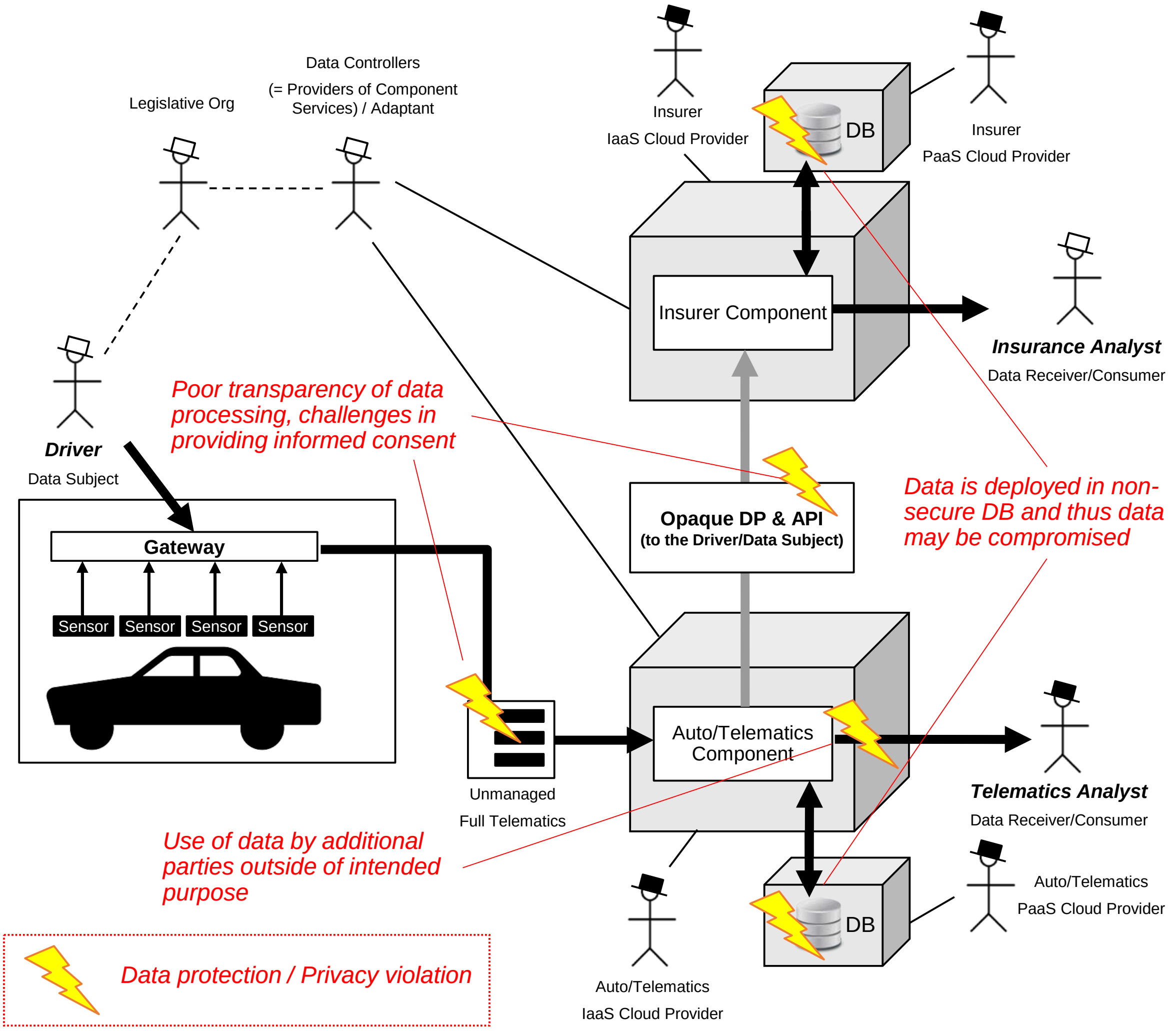

Data Controllers
(= Providers of Component Services) / Adaptant
Legislative Org
Insurer
IaaS Cloud Provider
DB
Insurer
PaaS Cloud Provider
Insurer Component
Insurance Analyst
Data Receiver/Consumer
Driver
Data Subject
Poor transparency of data processing, challenges in providing informed consent
Opaque DP & API
(to the Driver/Data Subject)
Data is deployed in non-secure DB and thus data may be compromised
Gateway
Sensor
Sensor
Sensor
Sensor
Unmanaged
Full Telematics
Auto/Telematics Component
Telematics Analyst
Data Receiver/Consumer
Use of data by additional parties outside of intended purpose
DB
Auto/Telematics
PaaS Cloud Provider
Auto/Telematics
IaaS Cloud Provider
Data protection / Privacy violation

Figure 3.13: Pay-as-you-Drive Insurance - potential risks

Data is deployed on non-secure infrastructure and thus there is a risk that data may be compromised by untrusted parties while it is stored or during transfer. The lack of transparency and general opacity of data processing activities carried out on the individual's data make it difficult for an individual to know what they are consenting to, creating challenges in meeting the burden of informed consent. With data flowing through multiple components and infrastructure owners, there is a further risk associated with the unintended use of data by an untrusted (or otherwise unintended) third party, outside of the purpose for which consent was granted. Figure 3.14 shows the RestAssured solutions to these risks.

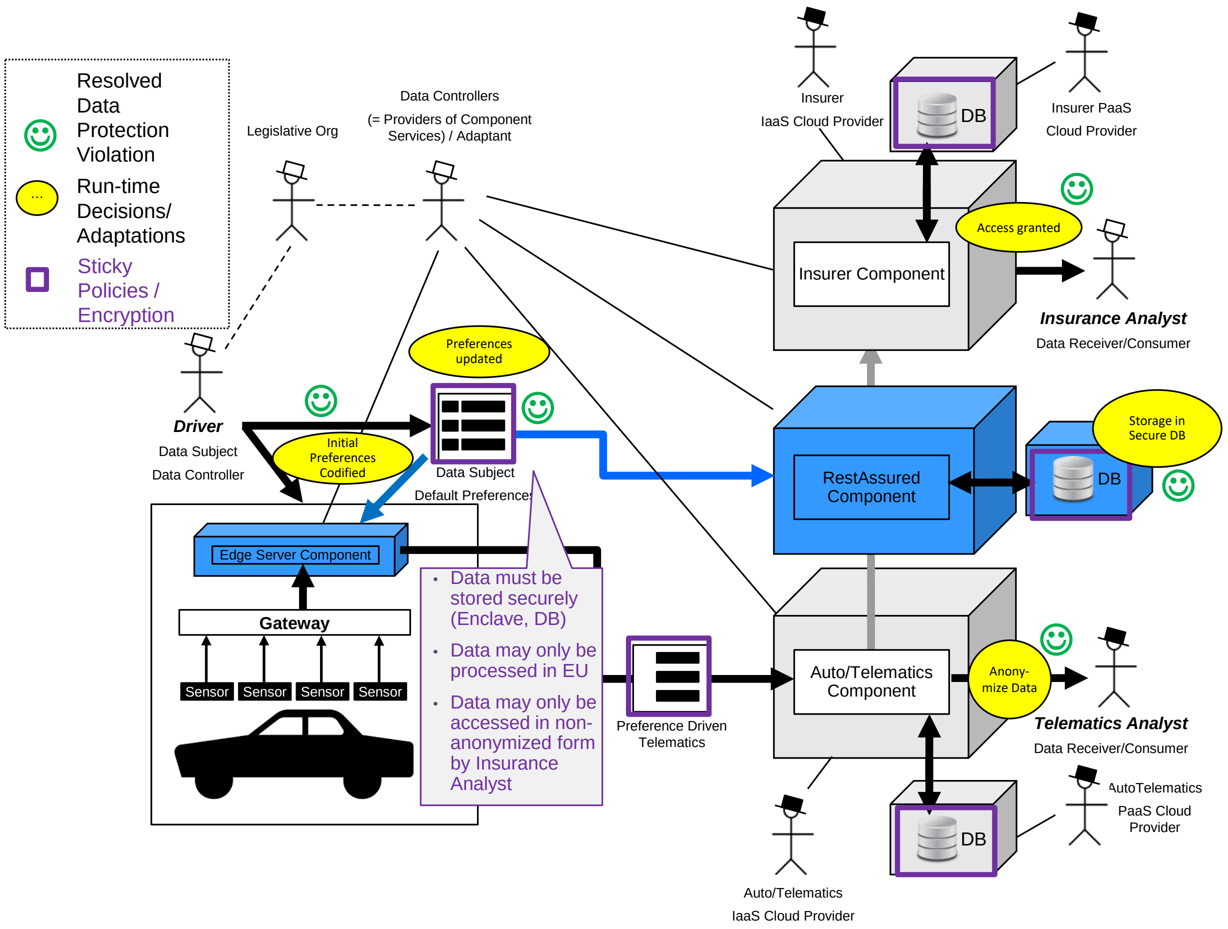


Figure 3.14: Pay-as-you-Drive Insurance - RestAssured solutions

Through the use of sticky policies, data encryption and secure enclaves data protection violations are resolved.

3.3.3 HPC: "High-Performance Computing for Commercial Use" Use Case

Supercomputing centres today face the constant challenge of how to keep the utilization rate of their systems at peak levels across the entire lifecycle of the system in order to maximize the return on investment. As typical system utilization rates are observed in the range of 65%, a significant amount of compute power that can be leveraged is lost. Attempts to increase this utilization rate in the past have culminated in mixing of workloads, physical partitioning of the system by workload type, provisioning parts of the system for more typical IT workloads for internal customers, and opening compute resources up to industry through the Cloud early results of which have seen systems achieve a 99% utilization rate. Industry users who seek to benefit from off-premise supercomputing power however have been reluctant to go all the way to Cloud due to concerns over information security and the potential to have their intellectual property compromised. This concern has most frequently been raised in cases where competitors are able to use the same infrastructure, or an insecure communications link between the sites could be compromised. A partial solution that has typically been utilized to address part of these concerns is a hybrid cloud model where much of the initial cloud-facing simulation data is judged to be precompetitive (and therefore at lesser risk or value), with the finishing touches and key differentiation carried out on-premise (as opposed to a cloud bursting model), building on the previous result. This however creates a need to constantly maintain trusted on-premise compute power, while placing additional costs and resources on monitoring and controlling data transfers, and never being able to fully realize the cost-savings of utilizing off-site compute power.

The following data flow views show the current situation and potential risks for data protection violations, and the RestAssured solutions for this particular use case.

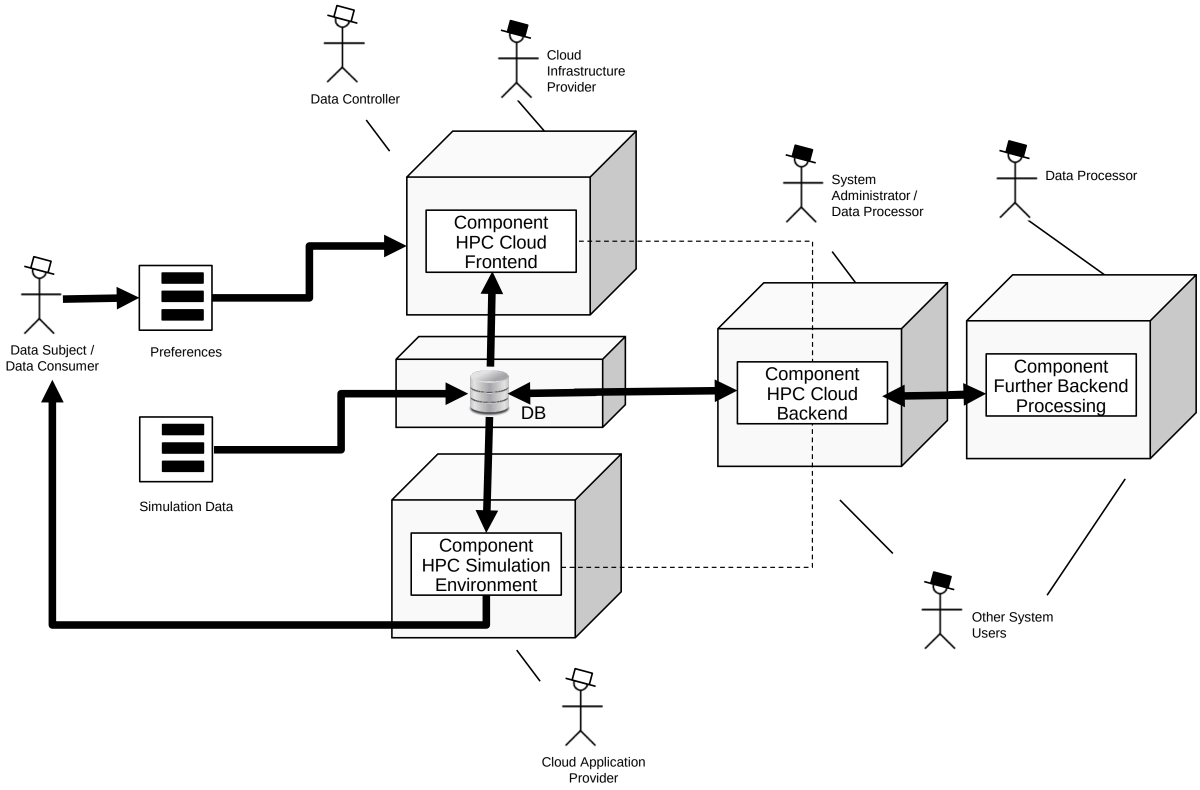

Data Controller
Cloud Infrastructure Provider
System Administrator / Data Processor
Data Processor
Component HPC Cloud Frontend
Data Subject / Data Consumer
Preferences
DB
Component HPC Cloud Backend
Component Further Backend Processing
Simulation Data
Component HPC Simulation Environment
Other System Users
Cloud Application Provider

Figure 3.15: High-Performance Computing for Commercial Use - current situation

Figure 3.16 below, highlights potential data protection risks for this scenario.

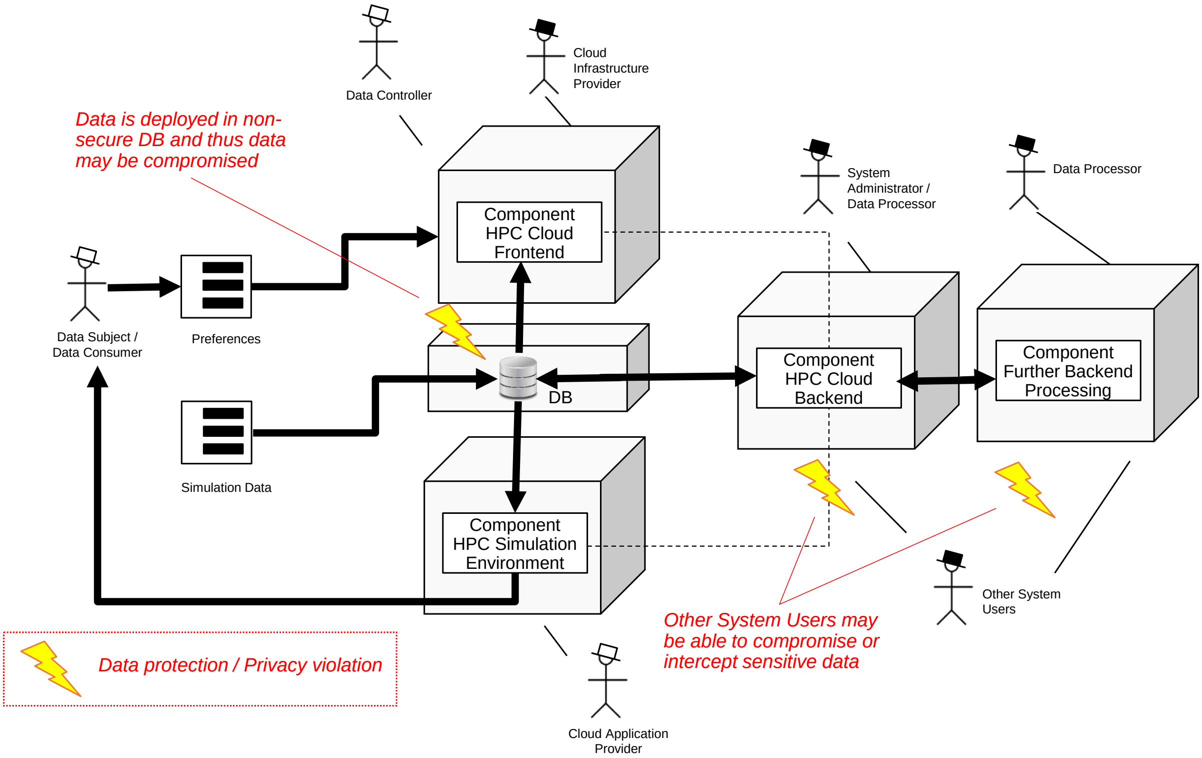

Data Controller
Cloud Infrastructure Provider
Data is deployed in non-secure DB and thus data may be compromised
Component HPC Cloud Frontend
System Administrator / Data Processor
Data Processor
Data Subject / Data Consumer
Preferences
DB
Component HPC Cloud Backend
Component Further Backend Processing
Simulation Data
Component HPC Simulation Environment
Other System Users
Other System Users may be able to compromise or intercept sensitive data
Data protection / Privacy violation
Cloud Application Provider

Figure 3.16: High-Performance Computing for Commercial Use - potential risks

Data is deployed on non-secure infrastructure and thus there is a risk that data may be compromised by untrusted parties while it is stored or during transfer. Other users (tenants) of the system may be able to compromise or intercept sensitive data, limiting the kind of data and extent of processing which may be carried out. In addition to infrastructure risks, simulations may involve multiple processing steps across components and services provided by different untrusted third parties. Figure 3.17 shows the RestAssured solutions to these risks.

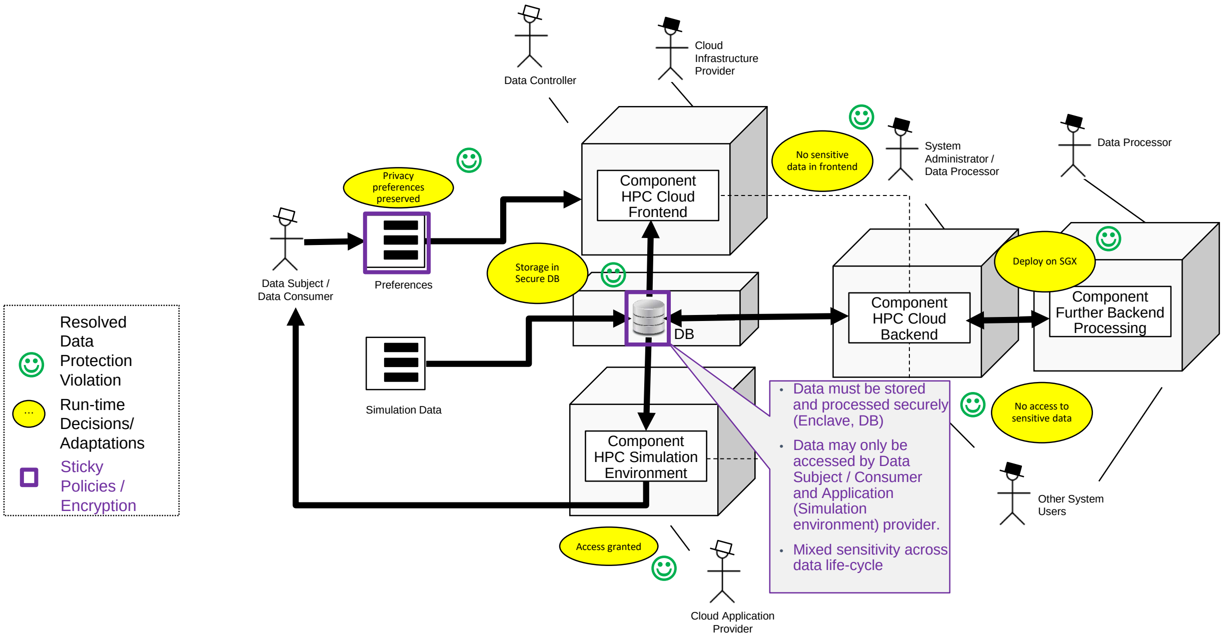

Data Controller
Cloud Infrastructure Provider
No sensitive data in frontend
System Administrator / Data Processor
Data Processor
Privacy preferences preserved
Component HPC Cloud Frontend
Data Subject / Data Consumer
Preferences
Storage in Secure DB
DB
Component HPC Cloud Backend
Deploy on SGX
Component Further Backend Processing
Simulation Data
Component HPC Simulation Environment
Data must be stored and processed securely (Enclave, DB)
Data may only be accessed by Data Subject / Consumer and Application (Simulation environment) provider.
Mixed sensitivity across data life-cycle
No access to sensitive data
Other System Users
Access granted
Cloud Application Provider
Resolved Data Protection Violation
Run-time Decisions/ Adaptations
Sticky Policies / Encryption

Figure 3.17: High-Perforance Computing for Commercial Use - RestAssured solutions

Through the use of sticky policies, data encryption and secure enclaves data protection violations are resolved.

4 Risk Analysis View

4.1 A Note on Risk Management Methodology

As mentioned in Section 1, this report concerns the RestAssured High Level Architecture. The methodology for managing security and privacy risks will be covered in a later deliverable (D7.1). However, at this stage it is sensible to give an outline of the envisaged methodology, which is based on the well-known set of standards built around ISO/IEC 27001. This is summarized in the following diagram:

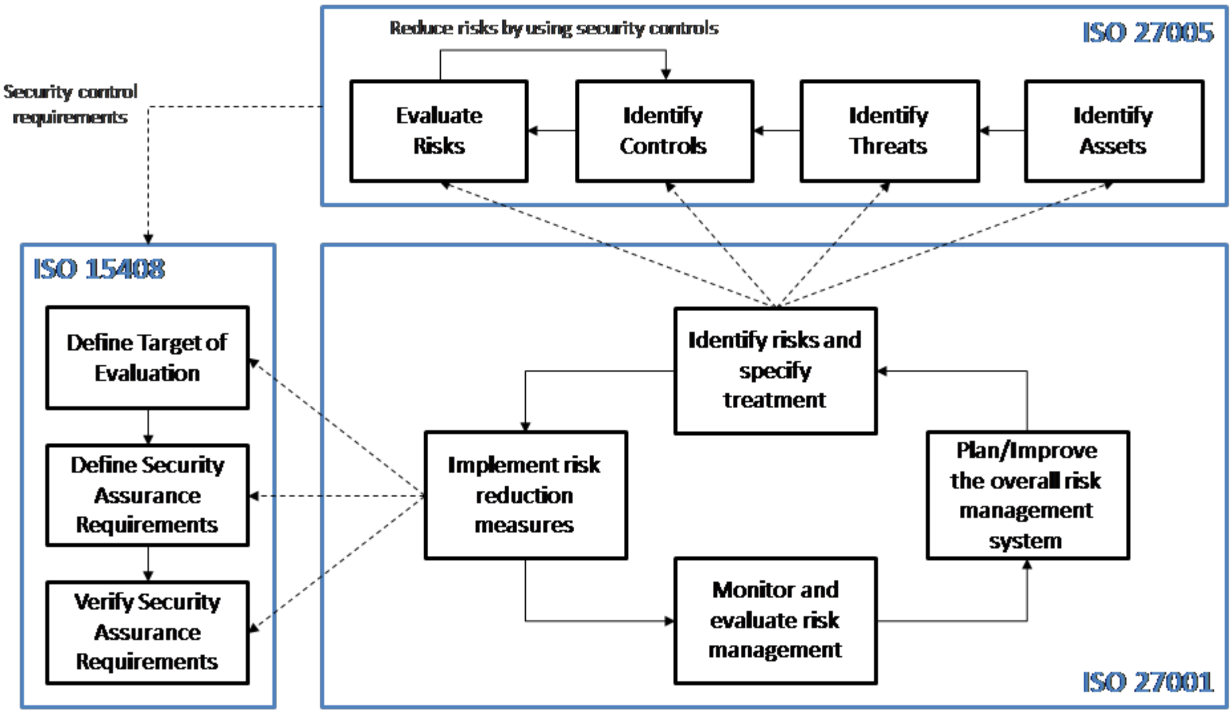

Reduce risks by using security controls
ISO 27005
Security control requirements
Evaluate Risks
Identify Controls
Identify Threats
Identify Assets
ISO 15408
Define Target of Evaluation
Define Security Assurance Requirements
Verify Security Assurance Requirements
Identify risks and specify treatment
Implement risk reduction measures
Plan/Improve the overall risk management system
Monitor and evaluate risk management
ISO 27001

Figure 4.1: Information system risk management standards

The starting point for managing risks is ISO 27001 which specifies how management processes and responsibilities should be handled within an organization. For a given IT system (or set of systems), this approach involves identifying risks and appropriate risk treatments (see below), then implementing those treatments, along with monitoring for incidents and improving the overall management processes where necessary to reduce the frequency or level of incidents in future.

The approach for identifying risks and risk treatments is specified by ISO 27005, which calls for an asset-based analysis to identify threats (i.e. sources of risk). Risks are then assessed in terms of the likelihood that the threat will arise, and the impact it would have on the identified system assets. At that point one has several options:

- accept the risk if it is so unlikely or has so little an impact that it would not be cost-effective to do otherwise;
- avoid the risk by reducing the functionality of the IT system(s) so the threat could not arise;
- transfer the risk to another stakeholder, e.g. by insuring against it, or in some other way agreeing that another stakeholder will assume responsibility and liabilities associated with it; or

- reduce the risk by introducing security controls which reduce the likelihood and/or impact of the associated threat.

Note that risk treatment does not always involve the use of security measures. When it does, this is sometimes referred to as "risk mitigation".

The procedure defined by ISO 27005 is difficult to carry out, requiring a good understanding of both the IT system(s) in which risks must be managed, and the types of threats that may affect security or privacy. It is also time consuming and is typically done during the initial design of a system, or when major upgrades are being considered. In RestAssured we wish to enable continuous risk assessment and run-time activation of risk management mechanisms.

The approach for doing this will involve two main elements:

- performing a conventional risk analysis at design time (i.e., before an application is deployed and used), but using a procedure that generates a machine understandable model of the relevant risks and possible risk reduction measures;
- using this model to support continuous assessment of risks during run time (i.e., once the application is deployed and operational), allowing automated decisions to adjust the control measures so the risk continues to be acceptable, or to avoid the risk if that is not the case.

For example, if personal data has to be transferred between data centres for processing, the design time analysis should produce a model specifying what control strategies (i.e. combinations of security measures) are be acceptable. If the destination data centre lacks some security measures, this will constrain how risks arising from the transfer can be addressed. The RestAssured framework will need to choose a suitable control strategy based on what is available. If none can be found it should prevent the transfer of data (even if it means data processing is delayed) until other means to manage the risk can be identified. See Section 4.4 for more details of how this would work.

4.2 System Assets and Structure

To analyze the risk for a cloud-based system, we must consider not only how that system is structured, but also how it is embedded in its environment. The following figure shows a cloud system analysis pattern. This pattern was developed in the ClouDAT[1] project, and will be adjusted to the needs of the RestAssured project.

[1] http://www.cloudat.de/

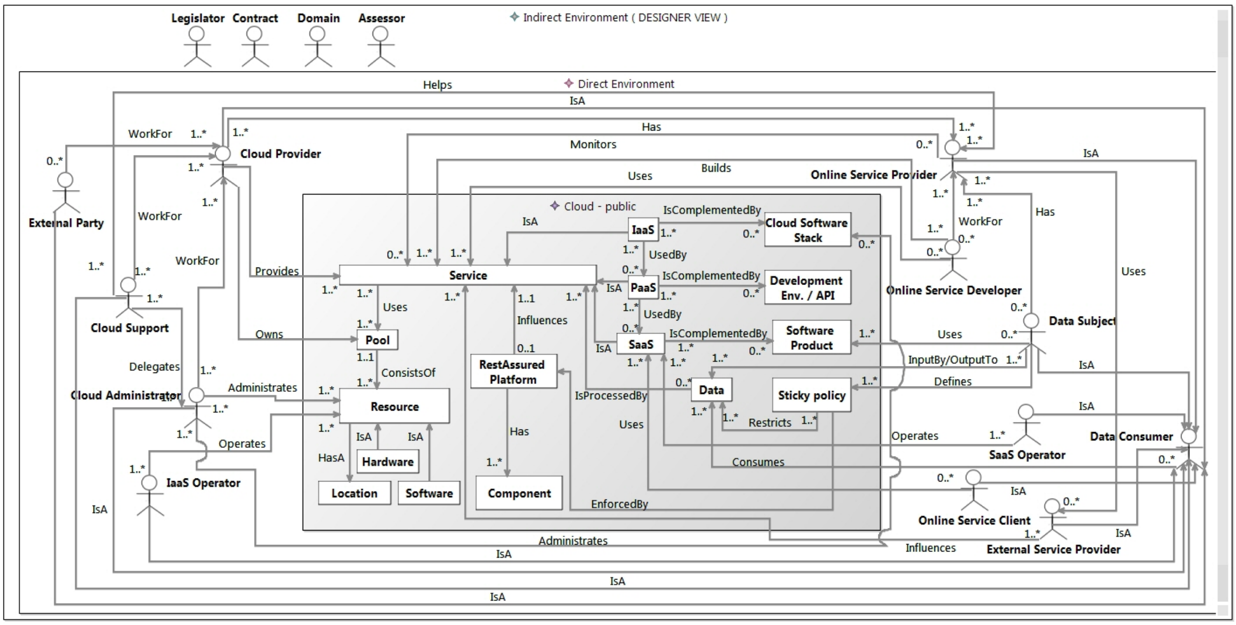


Figure 4.2: RestAssured-Cloud System Analysis Pattern (ReAs-CSAP)

The Cloud System Analysis Pattern (CSAP) represents a pattern for defining the context of a cloud computing service. For this purpose, the relevant elements of the CSAP have to be instantiated. Using a pattern can help ensure that crucial information is not overlooked, e.g. by ensuring that all potentially relevant asset types are considered.

For enabling a use of the CSAP in RestAssured the original CSAP has been extended. This new version of the CSAP is called RestAssured-Cloud System Analysis Pattern (ReAs-CSAP). ReAs-CSAP (shown in Fig. 4.2) enables capturing the context of cloud computing services that are relevant for RestAssured. In the following, we explain the elements of the ReAs-CSAP.

The *indirect environment* of a cloud computing service is described by *Indirect Stakeholders*. These stakeholders do not affect the considered cloud computing service directly by using or interacting with it. Rather they define provisions or laws that a cloud computing service has to comply with. The CSAP provides the following types of Indirect Stakeholders:

- Legislator: Representation of laws and provisions of legislators (e. g. Germany or the European Union) that are relevant for the cloud computing service.
- Domain: Specification of domain-specific provisions and guidelines the cloud computing service has to comply with.
- Contract: Representation of contractual provisions (e. g. Service Level Agreements with customers) that have to be fulfilled by the cloud computing service.
- Assessor: The organization that certifies the level of information security that is implemented by the cloud computing service. In the context of ISO 27001 the Assessor certifies the Information Security Management System (ISMS) for the cloud computing service.

The *direct environment* contains the *Direct Stakeholders* that are relevant for a cloud computing service and the representation of the Cloud itself. Direct Stakeholders are interacting actually with parts of the Cloud in form of cloud computing services and/or its physical resources. These stakeholders can also be directly involved in interactions with the Cloud. A Direct Stakeholder may act as a data controller. A data controller in ReAs-CSAP is modeled as a property of relevant Direct Stakeholders.

Furthermore, Direct Stakeholders can have logical relationships with each other. Similarly to the above-mentioned Indirect Stakeholders the CSAP provides different types of Direct Stakeholders that are explained in the following:

- Cloud Provider: Representation of legal entities that provide a cloud computing service in the form of IaaS, PaaS and/or SaaS that is relevant in the context of a RestAssured scenario. They also own the resources for providing the appropriate cloud computing service(s). Cloud Providers can have associations to the following other types of Direct Stakeholders:
  - Data Subject that makes use the of provided cloud computing service. The personal data of the Data Subject is processed/stored in the used service.
  - Direct Stakeholders of the appropriate type (External Parties, Cloud Support, Cloud Administrators, IaaS Operator) that are working for Cloud Providers.
- Cloud Support: The optional Cloud Support works for the Cloud Provider. It represents the point of contact for Cloud Customers if they have questions or problems regarding the used cloud computing service. Possible problems are delegated to the Cloud Administrators.
- Cloud Administrators: Cloud Administrators work for Cloud Providers. They administrate the resources of the cloud and handle problems that have been reported by customers.
- External parties: Representation of all external parties that work for the Cloud Provider to deliver services that are relevant for or could affect the operation of the considered cloud computing service. For example, external parties could be represented by companies for the maintenance of IT-resources and air-condition or cleaning services.
- IaaS Operator: IaaS Operator performs tasks regarding the operation of an IaaS-service.
- Data Consumer: Representation of every Direct Stakeholder that consumes the Data of a Data Subject.
- Online Service Client: Using the SaaS-service provided by an Online Service Provider. During this usage of cloud service, Online Service Client consumes Data of the Data Subject.
- External Service Provider: Representation of external cloud service providers that are used by the Online Service Provider. An xsExternal Service Provider delivers cloud services to the Online Service Provider.
- SaaS Operator: SaaS Operator perform tasks regarding the operation of an SaaS-service.
- Online Service Provider: Uses cloud computing service that is provided by the Cloud Provider. Online Service Provider are customers of cloud providers but they do not represent the end customers. Rather they use the provided service(s) in form of IaaS (Infrastructure as a Service) or a combination of IaaS and PaaS (Platform as a Service) to develop their own service on the level SaaS (Software as a Service).
- Online Service Developer: Online Service Developer implements SaaS-software for Online Service Provider. For the implementation of the SaaS-software they use the API and the development environment that is provided by the appropriate PaaS-service.
- Data Subject: Representation of end customers (an identified or identifiable natural person) that make use of a cloud computing service in form of
  - IaaS, PaaS or SaaS provided by Cloud Providers or
  - SaaS that is provided by Online Service Provider.

Data Subjects save and/or process their Data in the used cloud computing service.

Beside the Direct Stakeholders, the direct environment also contains the Cloud. The Cloud contains elements that represent the provided cloud computing service(s) and the resources that are necessary for the provision of the service(s). These elements are called Cloud Elements and have different types. The different types of cloud elements are described in the following:

- Service: Central point for referencing all provided cloud computing services.
- IaaS: Representation of the provided IaaS-cloud computing service.
- PaaS: Specification of the provided PaaS-cloud computing service.
- SaaS: Representation of the provided SaaS-cloud computing service.
- Cloud Software Stack: Representation of the software that is necessary for providing the corresponding IaaS-service.
- Development Environment and API: Specification of the API and the development environment that is provided for developing SaaS-software in form of the Software Product.
- Software Product: Represents the software that is provided via the corresponding SaaS-service.
- Pool: Central point for referencing all relevant physical resources of the cloud that are necessary for providing the appropriate cloud services.
- Resource: Central point for referencing all resources in form of Locations, Software and Hardware.
- Location: Representation of all Locations that contain cloud resources (e.g. computing center) or are relevant for the provided cloud computing service in another way (e.g. development site).
- Hardware: Representation of cloud resources in form of necessary Hardware (e.g. server racks or network components).
- Software: Representation of cloud resources in form of necessary Software (e.g. software for managing the cloud or virtualization).
- Data: Specification of the data of the Data Subject that is stored and/or processed in the used cloud computing service. The usage of Data can be Restricted by a Sticky Policy. A Sticky Policy is defined by Data Subject.
- RestAssured Platform: Provides security and privacy mechanisms that influence the cloud Service by enforcing Sticky Policy. The RestAssured Platform provides different Components that implement the appropriate security and privacy mechanisms.

The instantiated Cloud Elements represent so-called Assets that constitute physical or abstract things that have a value for Cloud Providers and their customers. Such assets have to be refined in order to perform a detailed risk analysis, giving rise to further models.

4.3 Machine-understandable Models

The models generated by this analysis will be represented using RDF, capturing meaning in the form of classification hierarchies and other relationships. This allows the use of existing tools to store and analyze the captured knowledge.

The knowledge will be organized in layers, following the approach successfully used in the FP7 OPTET project:

- Core model: this provides an upper ontology for concepts including “asset”, “threat”, “effect” (of threats on assets), and “control” (security measures that reduce the likelihood or impact of threats).
- Domain model: this encodes knowledge of the types of security threats that could affect systems in the domain of interest, which in RestAssured is the domain of distributed, cloud-based IT systems that handle personal data.
- Design-time system model: this encodes the structure of a specific system or application in terms of its assets, based on the Cloud System Analysis Pattern described above. The design time model is expressed in terms of system-specific asset classes, capturing the structure but not the detailed run-time composition of the system.
- Deployment model: this adds asset instances to the design-time model representing the initial run-time configuration of the system on deployment.
- Run-time system model: this provides an up-to-date snapshot of the system composition, taking account of assets that have joined or left the system since deployment, and also the current relationships of assets to each other (e.g. where data assets are stored).

The domain model is typically created by security experts, and in RestAssured the domain model will be one of the project outputs, encoding knowledge from consortium on managing risks in cloud-based systems, including knowledge produced by research in the project.

The core model encodes basic assumptions that can be used by analysis and decision support tools developed for RestAssured. Ideally these tools should be independent of the domain model, so that new security knowledge can be incorporated into the domain model without invalidating any of the tools that will use it.

The design-time model is created during the design time analysis of assets and potential risks. It will support the process of threat identification using machine reasoning to ensure any threats included in the domain model are found where they might arise in the specific system of interest. At the end of the design-time risk analysis stage, the design-time model will include system asset types and an estimate of their value to the system (e.g. to what extent the system would be compromised should each system asset be compromised), a threat catalogue of potential threats to system assets, and a set of control strategies that are considered sufficient to address each threat.

The run-time model is created during run-time based on system monitoring data. It captures the current configuration in terms of system assets and their relationships, and which security controls are available and/or in force to protect each of those assets. The run-time model inherits from the design-time model, so, as the configuration evolves, it will be possible to determine (by reference to design-time elements) what threats may affect each system asset, and what control strategies are considered sufficient. One point worth noting is that the acceptability of a control strategy may also depend on run-time elements, e.g. the policy attached to a specific data item will determine how much protection should be provided against threats to that data item. The design-time model will define the rules for deciding what is acceptable, but the decisions will be made at run-time.

4.4 Risk Analysis Models and Links with other Architectural Views

To summarize, there will be three main models built on a simple core:

- the domain model: a model of security knowledge relevant to RestAssured applications and cloud infrastructure;
- the design-time system model: capturing the structure of an application, including an (auto-generated) risk catalogue and a-priori risk levels and possible control strategies;

- the run-time system model: capturing the current configuration of that application, with current risk levels based on input from monitoring and adaptation components.

The domain model is only used at design time as a knowledge base used in the construction of the design-time system model. The two models that relate directly to the RestAssured architecture are the design-time and run-time models of a RestAssured application. These models will support both design-time and run-time risk assessment and support system adaptation decisions, as shown below:

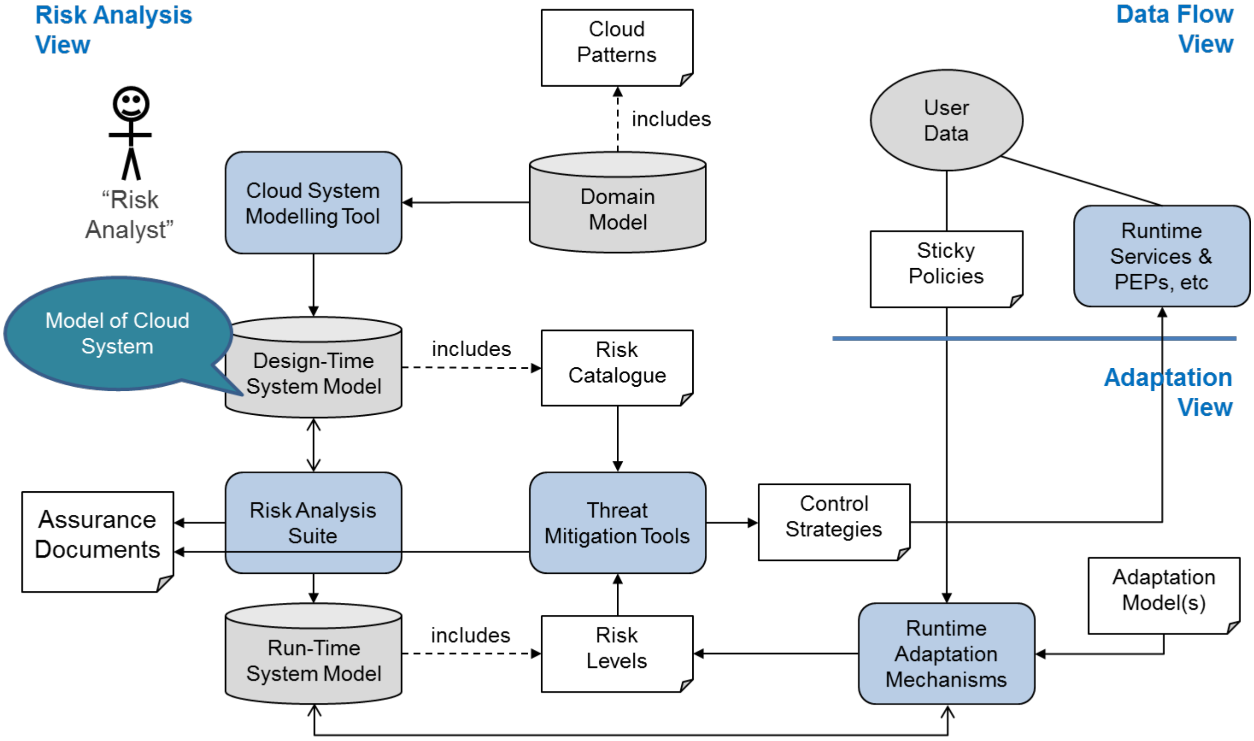


Figure 4.3: Risk analysis models and components

The use of the models to enable each step of the risk analysis procedure at run-time as well as design time is summarized in Table 4.4:

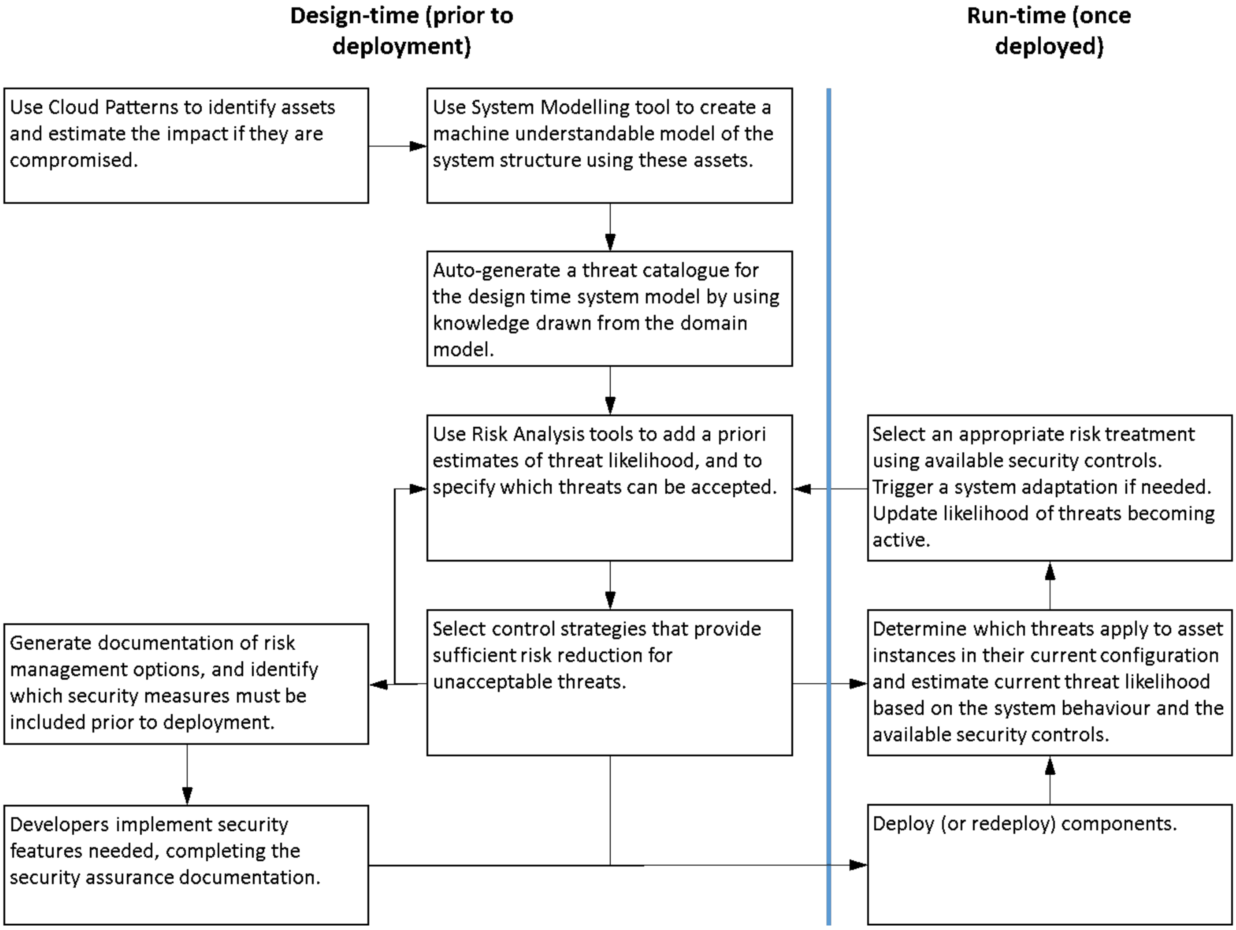

Design-time (prior to deployment)
Run-time (once deployed)
Use Cloud Patterns to identify assets and estimate the impact if they are compromised.
Use System Modelling tool to create a machine understandable model of the system structure using these assets.
Auto-generate a threat catalogue for the design time system model by using knowledge drawn from the domain model.
Use Risk Analysis tools to add a priori estimates of threat likelihood, and to specify which threats can be accepted.
Select control strategies that provide sufficient risk reduction for unacceptable threats.
Generate documentation of risk management options, and identify which security measures must be included prior to deployment.
Developers implement security features needed, completing the security assurance documentation.
Select an appropriate risk treatment using available security controls. Trigger a system adaptation if needed. Update likelihood of threats becoming active.
Determine which threats apply to asset instances in their current configuration and estimate current threat likelihood based on the system behaviour and the available security controls.
Deploy (or redeploy) components.

Figure 4.4: Risk analysis steps at design-time and run-time

5 Adaptation View

One fundamental solution concept of RestAssured is the use of a model of the cloud system and other relevant entities at runtime, called "model@runtime." The main purpose of the model@runtime is to support runtime adaptation, to be elaborated within WP5. This model will provide the basis for interpreting observations, analysing the impact of changes, and responding to potential data protection violations by means of appropriate runtime adaptations.

The meta-model of the model@runtime gives important insight into the concepts relevant for RestAssured at runtime, their attributes and relations. In this respect, it provides a logical view on the RestAssured components and their environment that we call the **Adaptation View**. It complements the other architectural views by a focus on what type of information is needed at runtime for adaptation.

To populate the Adaptation View, each technical WP was asked to provide their "individual" meta-models (concepts) relevant for their technical solutions. These conceptual models served as a basis for alignment and identifying overlaps, and in particular, to identify conceptual links and thus ultimately interfaces among technical solution components.

Figure 5.1 depicts the high-level structuring of the RestAssured model@runtime (thick box in the figure) into 5 sub-models and how these align with the monitoring and adaptation features of RestAssured. In particular, it shows that all five sub-models provide input for monitoring and adaptation, but only three of them (Infrastructure, Applications, and Data) are actually adapted. The meaning of the sub-models is described in Table 5.1.

Table 5.1: Meaning of the sub-models

Infrastructure:	Elements of the physical and virtual computing infrastructure underlying the applications
Applications:	Software above the level of virtual infrastructure and other system services. Includes applications belonging to users of RestAssured technology, RestAssured components, as well as applications of other parties
Data:	Similarly to Applications, this sub-model also refers to data managed by users of RestAssured technology, (meta-)data under control of RestAssured, and data managed by other parties
Goals:	Objectives and requirements that RestAssured aims at satisfying
Actors:	Contains the parties (persons and organizations) and roles relevant for the runtime operation of RestAssured

Figure 5.2 shows a population of this model for initial technical solutions of WP5. These initial technical solutions are concerned with data-protection-aware cloud resource management (e.g., see [?]).

Next, we present more details of each sub-model.

5.1 Infrastructure Sub-Model

Figure 5.3 shows the **Infrastructure** sub-model and its most important relationships with the other sub-models. In this sub-model, an IaaS cloud consists of multiple data centers (DC); each DC consists of

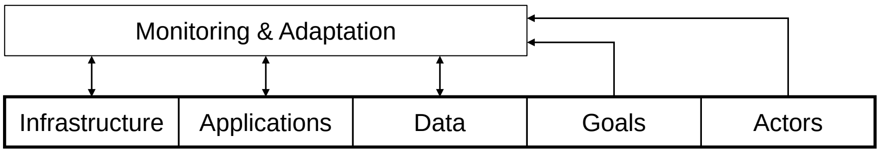


Figure 5.1: High-level structure of the RestAssured model@runtime

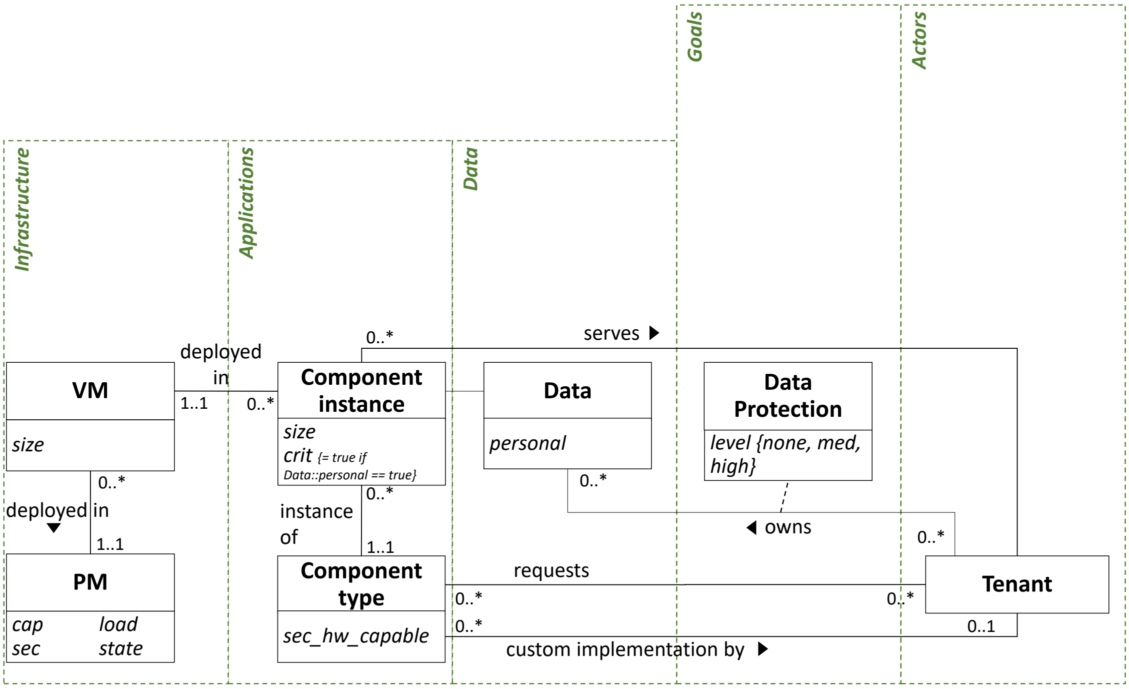


Figure 5.2: Concepts relevant for data-protection-aware adaptive resource management solutions developed in WP5, arranged according to the five sub-models of the Adaptation View

multiple physical machines (PM) each PM may host multiple virtual machines (VM), and each VM may host multiple containers. It should be noted that other deployment scenarios are also possible (e.g., a container could be hosted by a PM directly) that are not shown for the sake of readability. IaaS clouds, DCs, PMs, VMs, and containers are considered infrastructure elements. An IaaS cloud can be accessed through a public or private cloud interface. The former allows access to VMs only, while the latter allows access to all infrastructure elements.

Although attributes are not shown in the figure, it is important to note that an attribute of a PM is whether it supports secure hardware enclaves.

5.2 Applications Sub-Model

Figure 5.4 shows the **Applications** sub-model and its most important relationships with the other sub-models. According to this model, applications consist of multiple components that are linked by connectors. The logical structure of an application is defined by an application template, from which the specific application, component, and connector instances are derived and scaled as necessary.

This is a rather generic model of applications. For the RestAssured technology solutions themselves, more specific types of components (e.g., key management system, attestation service, policy enforcement point) will be determined and described as part of the Component View and its later refinements.

5.3 Data Sub-Model

Figure 5.5 shows the **Data** sub-model and its most important relationships with the other sub-models. The smallest unit of data is the “Attribute value.” Attributes and Records contain multiple attribute values; a Data set contains multiple Attributes and multiple Records. A data set can either be stored or transferred, represented by the respective entities Stored Data Set and Data Flow, both inheriting from Data Set. A Database consists of multiple stored data sets.

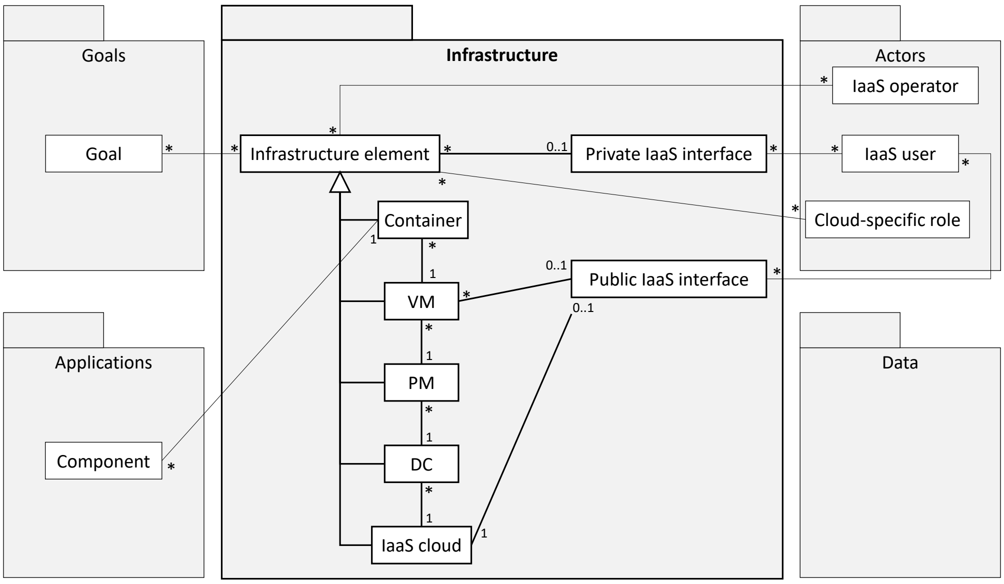


Figure 5.3: Infrastructure sub-model of the Adaptation View

Any Data object can be associated with a piece of Metadata that defines – for example in the form of a sticky policy – the types of operations allowed on the given Data object. Thus, Metadata provide the link to data protection goals from the Goals sub-model.

5.4 Goals Sub-Model

Figure 5.6 shows the **Goals** sub-model and its most important relationships with the other sub-models. It contains the different kinds of goals and requirements that are typical for cloud systems, like high performance, low resource consumption, data protection, user-friendliness, and availability. These high-level goals can be decomposed into more specific ones; in particular, data protection is decomposed into confidentiality, integrity, and authenticity goals. Data protection goals can be encoded in the form of sticky policies which are modeled as Metadata in the Data sub-model.

5.5 Actors Sub-Model

Finally, Figure 5.7 shows the **Actors** sub-model and its most important relationships with the other sub-models. In this sub-model, each Party (person or organization) can have multiple cloud-specific or data-specific Roles. As cloud-specific roles we differentiate users, developers and operators on SaaS (Software as a Service), PaaS (Platform as a Service) or IaaS (Infrastructure as a Service) level. Within data-specific roles, we differentiate data subjects, data producers, data processors, and data controllers. Cloud-specific roles may relate to different infrastructure elements. Similarly, data-specific roles relate to data objects. IaaS users can interact with the infrastructure via a public or private cloud interface. IaaS operators interact with the infrastructure elements they operate. A party in a specific role may have multiple goals.

5.6 Alignment with the Risk analysis view

Obviously, the Adaptation View has some overlaps with the other architectural views. Of particular interest is the overlap with the Run-time model of the Risk Analysis View because both models target run-time activities. The most important difference is, from the adaptation point of view, that risk is just one of the

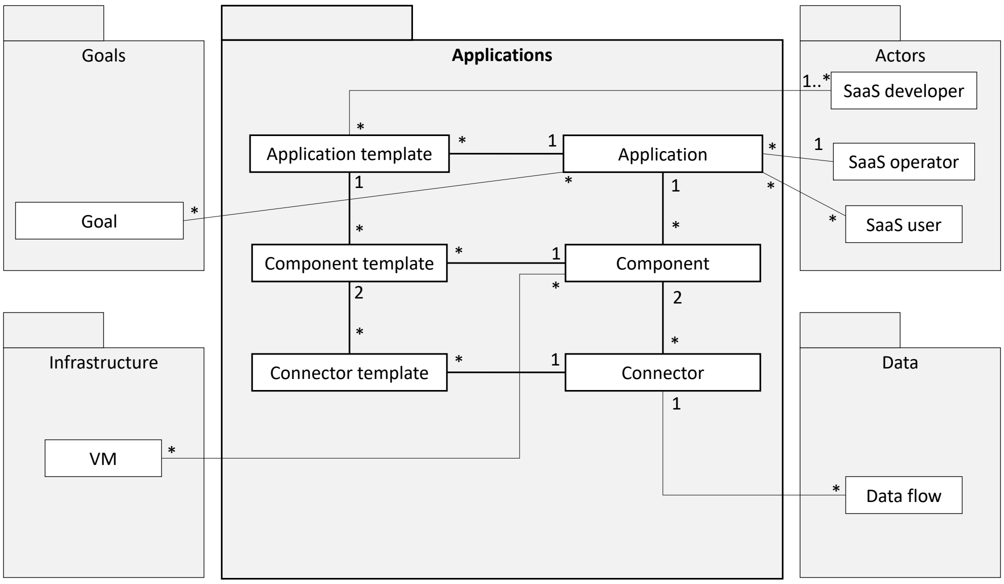


Figure 5.4: Applications sub-model of the Adaptation View

aspects (although a crucial one) that need to be taken into account for adaptation decisions. Other aspects that adaptation needs to account for include performance and costs.

Figure 5.8 elaborates on the relationship and possible cooperation between the two views. One possible contact point arises if the run-time risk assessment activities reveal that the risks associated with the current system configuration are too high. In this case, the Plan and subsequent Execute processes within the adaptation logic (consisting of the processes Monitor, Analyze, Plan, and Execute) can be triggered in order to devise and execute an appropriate adaptation plan that helps to reduce the risks to an acceptable level.

In the second case that involves cooperation between Risk analyses and Adaptation, the adaptation logic is working to come up with an adaptation plan to react to some change (the trigger for which might have originated from Risk analysis or from the Monitoring and Analysis activities within Adaptation). During the Planning process, it is important to assess the risk impact of any proposed changes, for which run-time risk assessment provides the appropriate mechanisms. An adaptation should only be executed after having assured this way that it leads to an acceptable level of risk (which may either mean that risk is *reduced* to or *kept* at an acceptable level, depending on whether the trigger for adaptation was too high risk or something else).

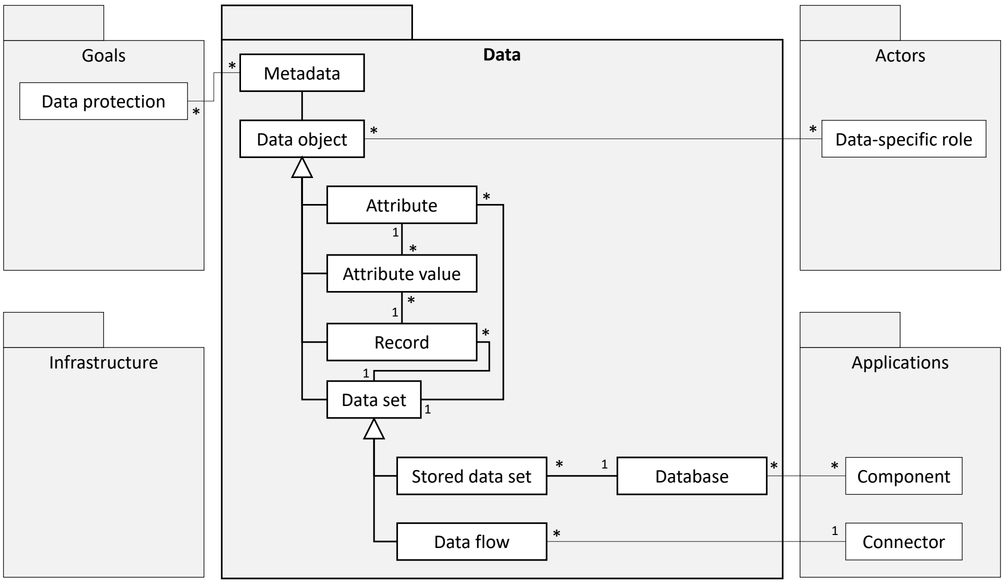


Figure 5.5: Data sub-model of the Adaptation View

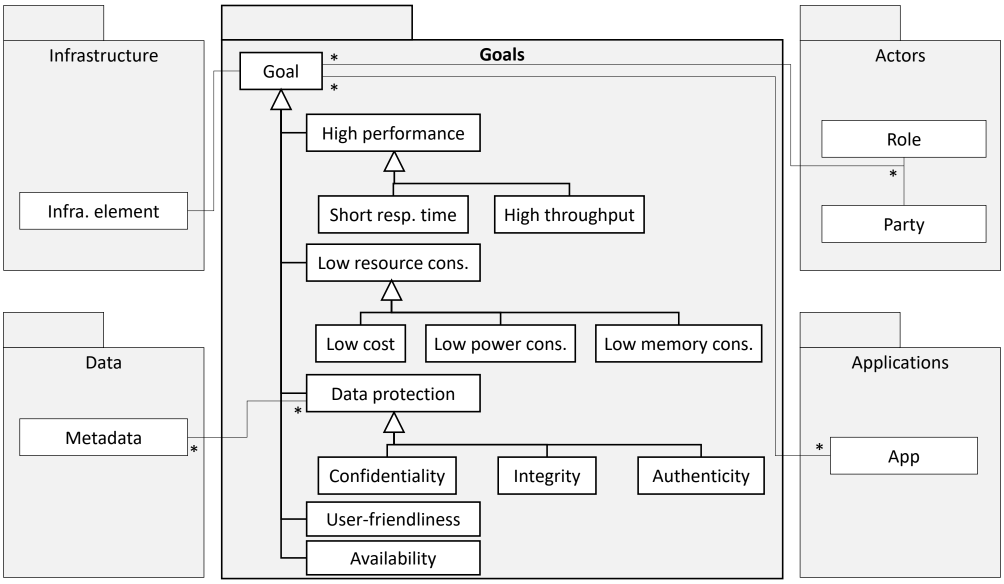


Figure 5.6: Goals sub-model of the Adaptation View

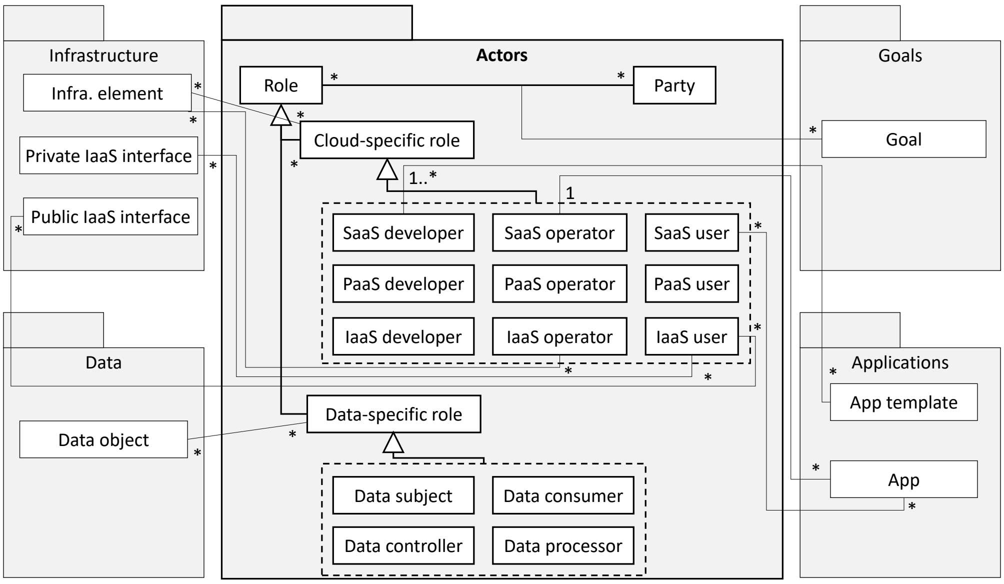


Figure 5.7: Actors sub-model of the Adaptation View

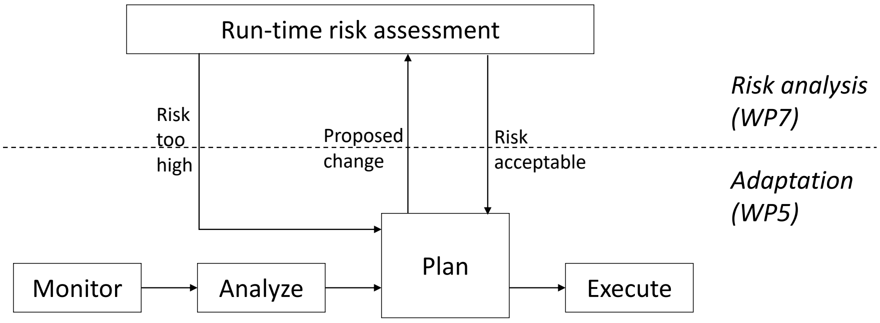


Figure 5.8: Alignment with the Risk analysis view

6 Component View

The RestAssured architecture can conceptually be thought of as a microservices architecture driven by sticky policies. Sticky policies, created by the Data Owner, control access to personal or valuable data even as it migrates across applications or even organizational boundaries. Enabling the Data Owner to have active control over access rights to his data will both encourage use of a wider range of cloud-based services, as well as comply with privacy protecting regulations, such as GDPR. This concept is present in Figure 6.1. This view shows how RestAssured can securely provide sticky policies assurances both in an untrusted (public cloud) environment (blue box) and in a trusted (private cloud) environment (green box). All operations on the data will be carried out by what can conceptually be considered as microservices, which will include SGX-enabled functionality, such as database storage and query, or other PaaS services.

6.1 Design Requirements

A number of requirements and design considerations need to be considered in the design of RestAssured architecture, including:

1. Data which is not allowed to reside in a physical location for whatever reason (e.g. geographical constraints, security reasons . . .) should not even pass through this location in transit.
2. Data access policies may change, and therefore leases for data access rights need to be re-evaluated after a reasonable period of time.
3. Data must traceable to enact “right to be forgotten” measures, and all actions must be non-reputably logged.
4. Public clouds are considered to be inherently untrusted, whereas private clouds are considered to be trusted.
5. Data access policies associated with the data (“sticky policies”) need to conform to the access policies defined on a number of levels, namely the application level, organization level, and legal (legislative) level.
6. Sticky Policies represent preferences of the Data Subject. In case of a conflict between user-defined access policies and organizational/legal access policies, the Data Gatekeeper makes the final decision in the composition of the sticky policies. If the conflict is on the legal level, the legal policies have priority.

6.2 Translating Requirements to Components

The requirements presented above directly helped drive the definition of the components illustrated in Figure 6.1, as explained below. More detailed explanations of the components can be found in Section 6.4.

Requirement 1 means that before data can be transferred either into the RestAssured system, or between RestAssured components, there needs to be confirmation that the transfer destination meets sticky policy requirements. RestAssured therefore defines a component called “the Gatekeeper” which is responsible for locating a conformant target. Requirement 2 means that the Gatekeeper must be able to handle both changes in sticky policies, and changes to the underlying security classification of potential targets. The architecture will not make any assumptions to the physical location of deployment of the Gatekeeper and therefore, by

Requirement 1, the Gatekeeper cannot hold any (potentially sensitive) user data, but rather just supply a target for data transfer.

The need to understand both the available resources and their security classifications leads to the definition of the RestAssured Service Registry. Service Registries are standard components in microservice architectures, and this concept can be extended for RestAssured.

Both the Service Registry and the Gateway need to be secured from external attack (i.e. modification), and therefore these components will either need to run in a secure enclave or a secure environment.

Using the Gatekeeper and Service Registry components to determine the destination for the transfer of data, means that the destination target may potentially change over time, for example in support of a change in access or business policy. This means that the route from the hosted service to the target may change. A secure and flexible manner of data transfer is therefore required, and will be implemented by using cryptographic proof of both the veracity of the requester, and verification that access to the target has indeed been granted, in order to prevent rogue sources from directly accessing RestAssured microservices.

This calls for an additional component, the Policy Enforcement Point which intercepts requests for data access (for either data transfer, or any data copy/read/processing) and uses the Access Policies and the different mechanisms that authenticates the generators of the request, the services that will be used or any relevant context information, decides whether or not the access to data is granted. The enforced security policies can come from the organization, the application, some legislation or the data subject. The security policies coming from the Data Subject would be bounded to the data and stored as Sticky Policies.

6.3 Addressing the RestAssured Goals

As described above, every transfer of data from one RestAssured component to another requires authorization from the Gatekeeper. This authorization can be in the form of a cryptographically signed token and must be accepted by an Policy Enforcement Point to allow access to the microservice. The token can be given an expiration date, which supports Requirement 2. Transaction logging can be done by the Policy Enforcement Points, to meet the logging portion of Requirement 3.

The realization of RestAssured as a microservices-inspired architecture naturally lends itself to the initial objectives as stated in both the DoA and the goals for the ICT-06-2016 call. More specifically, Table 6.1 reproduces the relevant portions of the Objectives table from the original DoA, and shows how the described architecture supports each objective.

Table 6.1: How the RestAssured architecture supports project objectives

Objective	*Description*	*How the architecture supports this*
O1: End-to-end secure cloud architecture and methodology	RestAssured will deliver an end-to-end secure data processing architecture for the cloud, in which for each compute node as well as each data transfer the security and data protection can be assured. *(Innovation pillars 1-4)*	The Gateway will match microservices with sticky policies to guarantee data security policies are met. Additionally, cryptographic mechanisms such as token verification will ensure the integrityof communication between RestAssured components.
O2: Secure cloud data processing and execution environment	By combining FHE and SGX security technologies and making them cloud-ready, RestAssured will deliver secure cloud data processing technologies that are compatible with open source and commercial cloud environments. *(Innovation pillar 1)*	SGX or FHE enabled microservices are supported by the architecture.
O3: Runtime data protection assurance	Using the concept of models@runtime, RestAssured will deliver engines for observation, resolution and prevention of data protection violations. *(Innovation pillar 2)*	Sticky policy driven deployment will guarantee that initial data placement meets with security requirements.
O4: Decentralized data lifecycle management	By enhancing the sticky policy concept to become applicable in dynamic and user-centric situations, RestAssured will deliver advanced means for decentralized management of data lifecycle and data access. *(Innovation pillar 3)*	A microservices-based architecture allows for a much higher level of reliability than a monolithic architecture. This architecture is larger decentralized; the Gatekeeper component can be replicated behind a load balancer. The Service Registry holds data which is largely static, and hence can be designed as a cluster of servers that use a replication protocol to maintain consistency.
O5: Engineering for run-time data protection	Exploiting automated risk management mechanisms, RestAssured will deliver engineering support for automatically deploying cloud services on secure and non-secure processing nodes. *(Innovation pillar 4)*	Model-based engineering methodology Models for capturing multi-stakeholder cloud systems and their security concerns Automated risk management tools for partitioning of cloud services

6.4 RestAssured Components

The *Application Logic* would typically be the cloud-hosted middle tier of a cloud service, which, although depicted in the figure as residing on a public cloud, can of course reside on a private cloud too. The Application Logic connects to backend services (such as a database).

Services, (represented in the diagram as S1, S2 etc.) are typically services in the microservices sense, and may be implemented as Docker containers. On a public (unsecure) cloud, these services may represent the secure enclave-enabled (e.g. SGX-enabled) portion of the hosted application, or they may represent secure enclave-enabled generic services, such as database services. On a public cloud, services may be regular PaaS services (shown in the figure below as managed by Marathon for container orchestration).

Policy Enforcement Points act in a manner analogous to firewalls, and only let authorized, and cryptographically secured requests be delivered to the RestAssured *services*.

The *Gatekeeper* is the component that is responsible for receiving the sticky policies and resource (service) request associated with the Application Logic and potentially supplementing them with additional sticky policies that may come from the business or legislative level. Additionally, the Gatekeeper interacts with the *Service Registry* to obtain a service which meets the requirements from the requesting Application Logic while complying with the potentially supplemented sticky policies. Also, the Gatekeeper is responsible for creating the cryptographic proofs that the Policy Enforcement Point will require to allow for service execution.

The EU sponsored, H2020 project, OPERANDO, has some similarities with the GateKeeper's composition of sticky policies which will be pursued further. From their website (http://www.operando.eu/):

> A key aspect addressed by OPERANDO is the need to simplify privacy for end users (data subjects). OPERANDO will support a simple Privacy Dashboard allowing users to specify their preferences. These will be automatically compared with Online Service Provider (OSP) privacy policies and translated into personal data access control decisions by the PSP (Privacy Service Providers).

In RestAssured, as opposed to OPERANDO, the native application is responsible for supplying the user privacy preferences, which GateKeeper will supplement with installation-specific business or legal requirements to create what OPERANDO calls "privacy policies".

The *Service Registry* is essentially a database describing the available services in the system. The Service Registry needs both to be able to collect information on deployed services, and to be able return a service matching the query from the Gatekeeper.

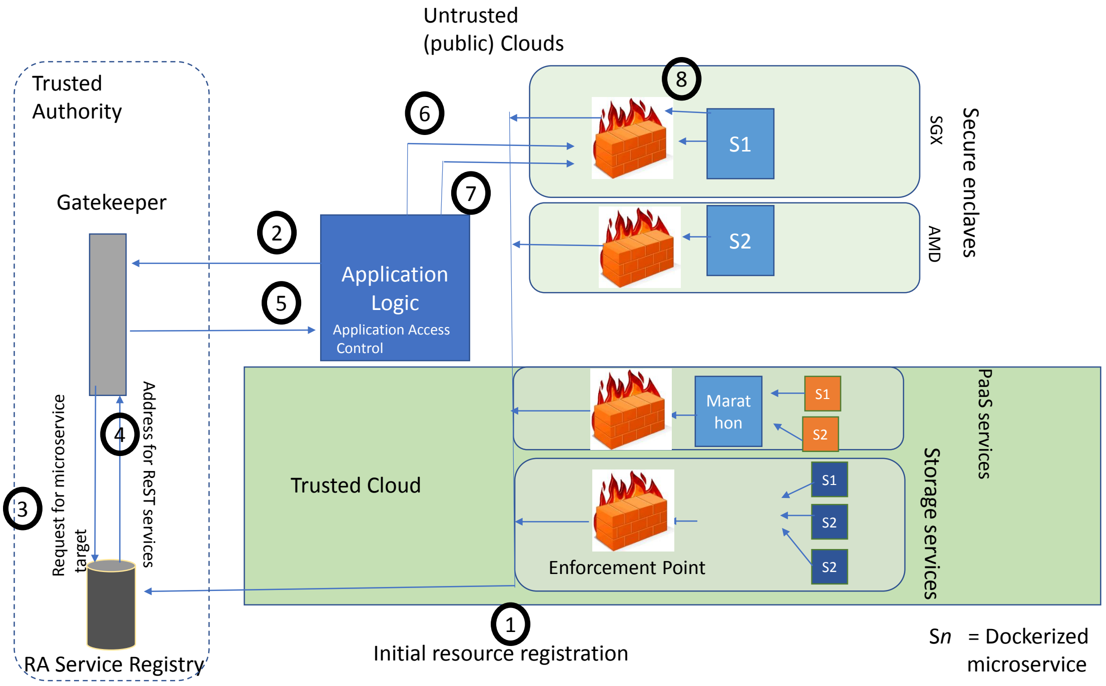


Figure 6.1: The RestAssured Component Architecture

6.5 Orchestration of the RestAssured Components

The following steps correspond to the numbers labeled in Figure 6.1.

1. All services register with the RestAssured Service Registry. This must include a description of what they offer (e.g., SGX-enabled service, AMD enclave, PaaS service, etc.), as well as other meta data like physical location, non-functional (e.g., cost for service), etc.
2. The application requests a service from the Gatekeeper (e.g., SGX enclave running a secure database instance). The sticky policies (representing constraints on the potential target service) for the data are sent, as well as an identifier confirming the properties of the requester.
3. The Gatekeeper adds any additional sticky policies based on company policy, or legal reasons and queries the Service Registry for the appropriate target.
4. The Service Registry returns the address (i.e., exposes an interface, e.g. REST) to the microservice that meets the requirements.
5. The Gatekeeper creates an *access grant* containing the authentication information for the requesting service, informing the calling application about the location of the target.
6. The application sends the access grant and requested data to the target service over a secure link, which gets intercepted by the Policy Enforcement Point. If the token is accepted by the Policy Enforcement Point, the information proceeds through, else a reject message is sent.
7. Secure application execution starts (load data, decrypt, process...).
8. The service returns processed results.

7 Testbed

The testbed will represent the deployment of the prototypical implementation of the RestAssured platform and technical components. Following the RestAssured architecture as reference, this task will integrate the technical solution components from WP4–7. The testbed will there allow demonstrating and testing the solution concepts of the technical workpackes.

7.1 Testbed Requirements

As basis for the set up of the testbed, we collected the technical requirements from the workpackages and use-cases. In Table 7.1 these requirements are listed.

WP/UC	Requirement
WP4	• 3 nodes with SGX supported hardware • 6th generation Intel CPU with SGX BIOS installed with Intel SGX Linux 1.7. • Intel SGX SDK for Linux OS • Intel SGX platform software for Linux • Intel SGX driver for Linux OS • Remote access via VPN
WP5	• Ability to demonstrate adaption – At least 2 physical servers – Ability to start virtual machines – Ability to install and run application components in VMs – Ability to migrate virtual machines
WP6	• Storage and processing capabilities for the policy engine
WP7	• Access via browser
HPC	• OpenStack
PAYD	• Collect sampled data • Simulation of moving cars

CARE	• Microsoft Azure – Azure App Service + web jobs – Azure SQL – Azure Storage – Azure Scheduler • Client Finance Portal – Microsoft Windows Server with IIS 7.5 – Microsoft SQL Server Standard 2008+ – Microsoft Web Platform Installer – Microsoft Web Deploy 2 – .Net Framework 4.5

Table 7.1: Requirements from the workpackages and the use-cases

7.2 Testbed Design

As an initial implementation of the requirements of the WPs and the use cases, we plan to deploy six desktop PCs as the hardware foundation for the testbed. Each PC shall be able to function either as a physical machine in a cloud scenario or as a control and observation station for the testbed users, where a control and observation station is a physical machine on which applications run which enable the testbed user to simulate specific cloud scenarios using the other five physical machines. The control and observation station(s) can be used by the testbed users to control (i.e. create cloud scenarios and react to certain events in these scenarios) and observe (i.e. monitor) all interactions between any physical and virtual machines.

Thereby, different configurations involving the physical machines (and their virtual machines) may be achieved. These possible configurations are shown in Figures 7.1–7.3 respectively.

- *Configuration 1* (Figure 7.1): In this configuration, the testbed user has access to a control and observation station from which he has access to all other physical machines. The control and observation

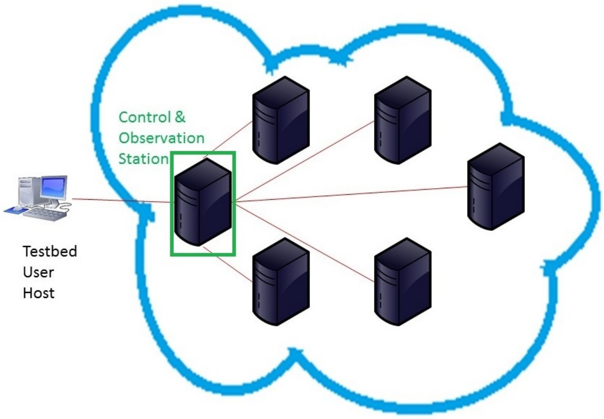


Figure 7.1: Architecture with a control and observation station which also functions as party in the cloud

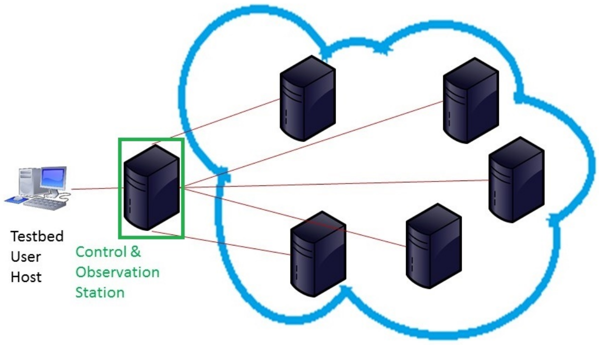


Figure 7.2: Architecture with a control and observation station which does not function as a party in the cloud

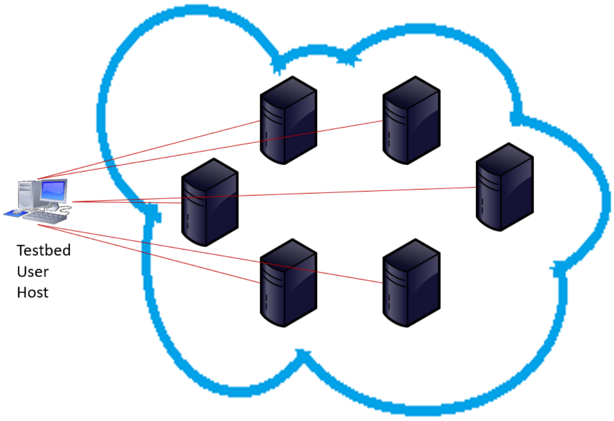


Figure 7.3: Architecture where the testbed user host has direct access to all physical machines

station can also be used as a party in the cloud.

- *Configuration 2* (Figure 7.2): The major difference of this configuration from Configuration 1 is that the control and observation station is not a party in the cloud anymore. Thereby, the performance of the control and observation station might improve which can be crucial, especially if multiple users want to use the testbed at the same time.
- *Configuration 3* (Figure 7.3): This third configuration is not including any control and observation station. In this case the users have direct access to all physical machines.

At this stage of the project, the final decision on the actual configuration still needs to be made, based on which of the layouts is most suitable for the demonstration and test purposes, in particular in light of the RestAssured use cases.

However, it appears that in particular a combination of Configuration 2 and Configuration 3 may be beneficial, i.e., allow to request from the control and observation station also the direct access to one or more machines. Thereby, the machines in direct access may be used for developing and "unit" testing, whilst the other machines may be used to deploy and test in the cloud.

8 Glossary

Term	Definition	Source / References
Data Subject	An identified or identifiable natural person; an identifiable person is one who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his physical, physiological, mental, economic, cultural or social identity;	http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML
Personal Data	“Personal data” shall mean any information relating to an identified or identifiable natural person (’data subject’)	http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML
Data Controller	“controller” shall mean the natural or legal person, public authority, agency or any other body which alone or jointly with others determines the purposes and means of the processing of personal data; where the purposes and means of processing are determined by national or Community laws or regulations, the controller or the specific criteria for his nomination may be designated by national or Community law	http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML
Data Processor	“Processor” shall mean a natural or legal person, public authority, agency or any other body which processes personal data on behalf of the controller;	http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML
Data Processing	any operation or set of operations which is performed upon personal data, whether or not by automatic means, such as collection, recording, organization, storage, adaptation or alteration, retrieval, consultation, use, disclosure by transmission, dissemination or otherwise making available, alignment or combination, blocking, erasure or destruction;	http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML
Data Consumer / Data Recipient	“Recipient” shall mean a natural or legal person, public authority, agency or any other body to whom data are disclosed, whether a third party or not; however, authorities which may receive data in the framework of a particular inquiry shall not be regarded as recipients;	http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML
Third Party	Any natural or legal person, public authority, agency or any other body other than the data subject, the controller, the processor and the persons who, under the direct authority of the controller or the processor, are authorized to process the data;	http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML

Consent	Any freely given specific and informed indication of the wishes of the data subject by which the data subject signifies his agreement to personal data relating to him being processed.	\http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML
Sensitive data	Data that needs to be protected by a organization which may include both personal data and business-related data.	
Service (IaaS, PaaS, SaaS) Provider	An organization that provides a network, storage or processing service.	http://www.pcmag.com/encyclopedia/term/51187/service-provider
Service (IaaS, PaaS, SaaS) Developer	The party (person, company etc.) that designs and implements the necessary software (for SaaS and PaaS) and hardware (for IaaS) of the service. This is a role, meaning that the same party can have multiple roles at the same time.	
Service (IaaS, PaaS, SaaS) Operator	The party (person, company etc.) that operates the service. This is a role, meaning that the same party can have multiple roles at the same time.	
Service (IaaS, PaaS, SaaS) Consumer	The party (person, company etc.) that uses the service. This is a role, meaning that the same party can have multiple roles at the same time.	
Trustworthiness	Measure of a reliability as honest or trustful (probability of failure)	
Trust	A belief that something will behave as expected	
Risk	An effect of uncertainty on objectives where an *effect* is a positive or negative deviation from what is expected.	ISO 31000
Threat	(see ISO 31000 also)	
Vulnerability	An unaddressed weakness or threat to system security	
Sticky policy	Data access policies that travel with the data as it moves across applications or organizations	
Data Gatekeeper	This is the component that receives the a request for a service (e.g. access to storage, a security enclave) and sticky policies from an application, potentially supplements the sticky policies with additional data constraints that come from either legislative or company policy, and attempts to locate a suitable service.	RestAssured
RestAssured Enforcement Point	(Policy enforcement point/policy decision point) Code that sits in front of applications (e.g. legacy databases, data processors...) that checks to make sure that the requested operation is authorized by the Gatekeeper.	RestAssured

RestAssured Service Registry	This is a runtime model of the RestAssured system, tracking the state of both system resources and (potentially) data transfer from one data controller / data processor to another. The Service Registry will receive a request for a resource with given policy constraints, and map to an available resource.	RestAssured
Secure Hardware Enclave	A secure enclave is an address space that is private to the task running in it. Other tasks co-located on the same physical host cannot access this memory, regardless of their runtime priority. Additionally, if this memory space is dumped by an attacker, only jibberish will result.	RestAssured
Trusted Environment	This is a combination of components (namely the Gatekeeper and Service Registry) which run in a protected environment.	RestAssured

Table 8.1: Glossary of Commonly Used RestAssured Terminology

9 Conclusion

This deliverable has provided an initial version of the RestAssured High Level Architecture and the design of the RestAssured testbed as of Month 4 of the project.

The RestAssured High Level Architecture consists of four views that address specific concerns of the RestAssured solutions. On a conceptual level, this deliverable has presented three complementary views: (1) the data flow view, defining the principal types of secure cloud data processing chains; (2) the risk analysis view, defining the main elements of a cloud stack and its applications such as to serve as an input for risk analysis and decision making, and (3) the adaptation view, which defines the main elements of a cloud stack and its applications to serve as an abstract representation of the cloud configuration at run time (aka. model@runtime) to serve as a basis for triggering and enacting adaptations. On a technical level, this deliverable provided an outline of a possible microservices-based implementation of the RestAssured approach.

A first official release of the RestAssured methodology is planned for Month 16 of the project, together with the first public release of the RestAssured architecture and implementation. The final official release of the RestAssured architecture and methodology is planned for Month 27 of the project