The research leading to these results has received funding from the European Community’s H2020 research and innovation programme under grant agreement n° 731678.
<table>
<thead>
<tr>
<th>Project Number</th>
<th>731678</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Title</td>
<td>RestAssured - Secure Data Processing in the Cloud</td>
</tr>
<tr>
<td>Deliverable Type</td>
<td>External Report</td>
</tr>
<tr>
<td>Title of Deliverable</td>
<td>Final RestAssured Handbook</td>
</tr>
<tr>
<td>Nature of Deliverable</td>
<td>External Report</td>
</tr>
<tr>
<td>Dissemination Level</td>
<td>Public</td>
</tr>
<tr>
<td>Contractual Delivery Date</td>
<td>31 October 2019</td>
</tr>
<tr>
<td>Actual Delivery Date</td>
<td>30 November 2019</td>
</tr>
<tr>
<td>Contributing WPs</td>
<td>All</td>
</tr>
<tr>
<td>Editor(s)</td>
<td>John Boyle</td>
</tr>
</tbody>
</table>
## Contributors

<table>
<thead>
<tr>
<th>Name</th>
<th>Organization</th>
<th>Sections</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Boyle</td>
<td>OCC</td>
<td>Sections 0.1, 1, 0.2, 3</td>
</tr>
<tr>
<td>Charlton Dowding</td>
<td>OCC</td>
<td>Sections 2.3.3</td>
</tr>
<tr>
<td>Zoltan Mann</td>
<td>UDE</td>
<td>Section 2.3.1</td>
</tr>
<tr>
<td>Nazila Gol Mohammadi, Ludger Goecke</td>
<td>UDE</td>
<td>Section A.3</td>
</tr>
<tr>
<td>Paul Mundt</td>
<td>Adaptant</td>
<td>Sections 2.3.2, 0.2</td>
</tr>
<tr>
<td>Eliot Salant</td>
<td>IBM</td>
<td>Sections 1, 2, 0.2, A.5</td>
</tr>
<tr>
<td>Cyrille Piatte</td>
<td>Thales</td>
<td>Sections A.4, 2, 0.2</td>
</tr>
<tr>
<td>Toby Wilkinson, Mike Surridge, Stephen Phillips</td>
<td>ITInnov</td>
<td>Sections A.3, 2, 0.2</td>
</tr>
</tbody>
</table>
## Document History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>October 2018</td>
<td>Initial version</td>
</tr>
<tr>
<td>V2</td>
<td>October 2019</td>
<td>Second version</td>
</tr>
</tbody>
</table>
Contents

List of Figures ........................................................................ 7
List of Tables ......................................................................... 9
0.1 Introduction ...................................................................... 10
0.2 RestAssured Handbook V2 .............................................. 12
1 Overview of RestAssured ..................................................... 14
2 How to Build a Business Offering based on RestAssured Technology . 15
  2.1 Introduction ................................................................. 15
  2.2 Components ............................................................... 16
    2.2.1 TruCE framework ................................................ 16
    2.2.2 Apache Parquet Modular Encryption ..................... 16
    2.2.3 System Security Modeller .................................... 16
    2.2.4 Runtime Risk Evaluator ....................................... 17
    2.2.5 Thales Components ............................................. 17
  2.3 Building Solution Examples ........................................... 19
    2.3.1 Enabling run-time data protection assurance .......... 19
    2.3.2 Integrating the Pay-As-You-Drive Use Case .......... 20
    2.3.3 Risk assessing the WaveCloud Architecture .......... 24
3 Conclusions ....................................................................... 28
A Appendices ....................................................................... 29
  A.1 References to Open Source Software .............................. 30
    A.1.1 Parquet encryption ............................................. 30
    A.1.2 Keycloak .......................................................... 30
    A.1.3 Apache Jena ..................................................... 30
    A.1.4 Jersey ............................................................... 30
    A.1.5 Hashicorp Vault ............................................... 30
    A.1.6 Apache Kafka ................................................... 30
    A.1.7 Apache Zookeeper ............................................ 30
    A.1.8 Apache Spark .................................................. 30
  A.2 References to API Documentation .................................. 31
    A.2.1 Data Gatekeeper: Data Protection Decision Point .... 31
    A.2.2 Data Gatekeeper: Sticky Policy Manager .............. 31
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.2.3</td>
<td>Data Gatekeeper: Data Protection Contract Manager</td>
<td>33</td>
</tr>
<tr>
<td>A.2.4</td>
<td>Data Gatekeeper: Context Manager</td>
<td>33</td>
</tr>
<tr>
<td>A.3</td>
<td>Risk Models</td>
<td>35</td>
</tr>
<tr>
<td>A.3.1</td>
<td>The CSAP Tool</td>
<td>38</td>
</tr>
<tr>
<td>A.3.2</td>
<td>System Security Modeller</td>
<td>43</td>
</tr>
<tr>
<td>A.3.3</td>
<td>The Runtime Risk Evaluator</td>
<td>48</td>
</tr>
<tr>
<td>A.4</td>
<td>Data Lifecycle</td>
<td>50</td>
</tr>
<tr>
<td>A.5</td>
<td>Configuring a RESTASSURED-like solution for capturing data as encrypted Parquet files</td>
<td>54</td>
</tr>
<tr>
<td>A.5.1</td>
<td>Starting Kafka and Zookeeper</td>
<td>54</td>
</tr>
<tr>
<td>A.5.2</td>
<td>Setting up Vault for key management</td>
<td>54</td>
</tr>
</tbody>
</table>
List of Figures

Figure 1: PAYD Conceptual Data Model ................................................................. 10

Figure 2.1: PAYD Conceptual Data Model ............................................................ 21

Figure 2.2: PAYD v2 Software Architecture .......................................................... 22

Figure 2.3: WaveCloud model construction ............................................................ 25

Figure 2.4: SecureConfig control assertion ............................................................ 26

Figure 2.5: High risk threat assessment ................................................................. 27

Figure A.1: Standardised Information Risk Management Process (Fig 2 from ISO 27005:2018) ......................................................... 36

Figure A.2: Organisational Dependencies in Risk Assessment .................................. 37

Figure A.3: A Risk Assessment Methodology for Cloud-Based Software and Services ................................................................. 38

Figure A.4: Editor mode of the CSAP-tool ............................................................... 39

Figure A.5: Adding a Cloud Element and defining its name .................................... 40

Figure A.6: Equality of the name and instance type of an added Cloud Element ............ 40

Figure A.7: Instantiation of a Cloud Element of the instance type Data in a ReAs-CSAP ................................................................. 41

Figure A.8: Representation of the instantiated Cloud Element of the instance type Data in a ReAs-CSAP ................................................................. 41

Figure A.9: Definition of Properties of the instantiated Cloud Element of the instance type Data ................................................................. 42

Figure A.10: Asset palette, canvas, and relationships ................................................. 43

Figure A.11: Misbehaviour impact levels, likelihoods, and risk levels ......................... 44

Figure A.12: Trustworthiness levels ........................................................................ 44

Figure A.13: Remaining threats in the SCANT use case after applying standard controls ................................................................. 45

Figure A.14: Finding the root cause ...................................................................... 46

Figure A.15: Treating the root cause ....................................................................... 47

Figure A.16: Treating the GDPR compliance threat .................................................. 48
List of Tables

Table 2.1: Components developed within the PAYD use case ................................. 23
0.1 Introduction

The RESTAssured handbook is an online, multi-media handbook that explains how to use the RESTAssured solutions in open source as well as commercial cloud environments.

V1 of the handbook was released in December 2018 after Y1 of the project. It has been regularly updated during the project and this report brings together with the updated technical releases and the additional materials produced during the 2nd part of the project.

The purpose of each of the RESTAssured components is described in outline together with links to the RESTAssured code libraries; open-source components; licencing agreements.

The handbook includes as set of use case descriptions designed to show how to apply and validate the RESTAssured technologies. The Handbook therefore explains in a hands-on fashion how the RESTAssured solutions may be used and will thus be an important means to achieve uptake and impact in practice.

Specifically, the handbook covers the following: section I is an overview of RESTAssured for those new to the project objectives, technologies and delivery platforms. This is accompanied by a set of project videos, screenshots or descriptions in sections A.3 and A.4. These cover:

- an overview of the RESTAssured concept illustrated through the Social Care use case,
- an introduction to the RESTAssured risk models and risk modeling tools,
- an overview of the individual RESTAssured components.

Figure 1: RestAssured PHYD video
Section 2 explains how to build a RESTASSURED business solution and offering. Each of the technical components is first described and then the application of those components to the business cases is summarised. For each use case we describe the architecture of the solution (as applied to the use case) and how the solution is actually implemented including clear reference to the code libraries and how they are called.

The handbook includes a set of appendices which reference the open source libraries used and the RESTASSURED APIs.
0.2 RestAssured Handbook V2

Specific additions to V2 of the Handbook are broken down into the new technical components and the new application of these technical components to the use cases.

1. New technical components:
   
   (a) IBM has implemented encryption for Apache Parquet files, which is a major contribution by RestAssured to the Open Source community. Apache Parquet is a columnar storage format widely used within the Hadoop ecosystem (e.g. including Spark). Storing database information in columnar format allows for much more efficient loading of Big Data into an analytic engine, such as Spark SQL, since only those columns required for the data are required to be transferred from disk to memory. IBM has implemented encryption of Parquet files, where the decryption of the data is now done within the Spark SQL engine. This allows for Big Data files to securely reside in a public cloud, and then be utilized by a Spark SQL engine running either in a trusted cloud, or within a secure enclave, such as the AMD enclave.

   (b) IT Innovation has added metrics for financial cost and performance overheads for security measures to the SSM knowledge base. At the time of writing, it has not yet been possible to add support for cost or performance impact estimation in the SSM tool itself.

   (c) IT Innovation has extended domain models used by SSM to improve modelling of data flows and data lifecycle including anonymization. The extensions use machine inference to deduce the flows of data between producer, consumer and storage processes, and determines the relationships of intermediate processes and the data flow. This simultaneously improves support for the data lifecycle model from WP6, and assists SSM users by making it easier to define the interactions between processes and data.

   (d) IT Innovation has implemented an interface for run-time risk evaluation, developing new methods to calculate system risk levels on the fly, based upon adaptations that align with variation points in a system’s design. This is supported by a new component from IT Innovation that performs a mapping between SSM model variants and the run-time model from WP5.

   (e) Thales has integrated the Data Gatekeeper with an OpenID Connect server. This allows receiving information about authenticated end-users of the Rest-Assured system, whether they are Data Consumers, Data Subjects or Service Providers.

   (f) Thales has extended the reasoning capabilities of the Data Protection Decision Point, providing a matching against a context ontology, thus providing Context Based Access Control.

   (g) Adaptant has extended the PAYD applications with support for the OAuth 2.0 protocol, which will be further extended for OpenID Connect for authenticated communication with the Date Gatekeeper by different roles.

2. New applications of the technical components to the use cases:

   (a) High performance computing for commercial enterprises use case will be used to illustrate authentication and authorization between components, providing end-to-end auditability across the data-lifecycle, as well as exploring the application of RestAssured technologies to Apache Spark and its related components (Spark SQL, MLlib).

   (b) PAYD use case will explore the use of adaptation to handle country-level changes, service portability through amendment of the data protection contract, and will further explore shifting away from a centralized Data Gatekeeper model in order to enable decision making and enforcement at the Edge.
(c) IBM has implemented a version of the Pay-How-You Drive (PHYD) use case to illustrate the use of Apache Parquet file encryption. This allows for data files, such as those holding automobile telematics data, as well as policy holder personal data, to securely reside in a public cloud, and then be utilized by an automobile insurer either in a trusted cloud, or within a secure enclave, such as the AMD enclave.

(d) Social care has been used to illustrate the automated risk management for run-time data protection tools
1 Overview of RestAssured

RestAssured provides solutions to specific technical concerns of data protection in the cloud (such as geo-location restrictions on personal data), which are imposed by the dynamic, multi-stakeholder and decentralized nature of federated cloud systems. These concerns mean that privacy and security by design approaches are no longer sufficient, due to uncertainty at design time of how the cloud and privacy requirements may dynamically evolve and change at run time. To this end, RestAssured provides novel mechanisms and cloud architectures for the runtime detection, prediction and prevention of data protection violations.

RestAssured assures the protection of sensitive business and citizen data in the cloud by combining four pillars of innovation: (1) the development of encryption for Parquet files for data-at-rest security, together with cloud enablement of hardware enclaves for protected data processing, (2) sticky policies for decentralized data lifecycle management, (3) models@runtime for data protection assurance, and (4) automated risk management for run-time data protection.

RestAssured solutions are being demonstrated through three use cases driven by project partners; High Performance Computing for commercial enterprises; Pay As You Drive usage based insurance; and self-directed Social care for vulnerable adults and social care providers.

The RestAssured Handbook explains, in a hands-on fashion, how the RestAssured solutions may be used and is thus an important means to achieve uptake and impact in practice.

RestAssured has released to YouTube a short video which explains the main concepts of RestAssured through the context of its social care use case. This demonstration video, “RestAssured In Action: An example use case”, can be found at: https://restassuredh2020.eu/wp-content/uploads/2018/07/RestAssured-demo.mp4

A more technical video of the IBM implementation of a Pay-How-You-Drive use case for automobile insurance can be found at: https://restassuredh2020.eu/wp-content/uploads/2019/09/IBM_PHYD_demo_movie.mp4

A deeper look at the RestAssured technologies not shown in the videos for brevity considerations can be found in the Appendices, specifically:

- Risk Models (Appendix A.3)
- Data Lifecycle (Appendix A.4)
2 How to Build a Business Offering based on RestAssured Technology

2.1 Introduction

REST_ASSURED offers a blueprint to allow third parties to create a Cloud infrastructure which both offers a high degree of security for hosted applications, as well as giving Cloud infrastructure and Service Providers a means to help meet GDPR requirements. Additionally, a number of REST_ASSURED components such as the Risk Analysis modeler (CSAP) can be used as standalone tools.

This handbook will serve as a guide to create a REST_ASSURED -like system, and is meant to be used in conjunction with the more technical and detailed project documentation which can be found on the project website, https://restassuredh2020.eu/publications.

A high-level view of the architecture and the individual components is presented in the D3.2 and D3.3 High Level Architecture and Methodology deliverables. These documents should be considered as a starting point to gain a conceptual understanding of the goals of the REST_ASSURED system and the interactions between the software components.

The actual programming interfaces (APIs) or low-level specifications for each component are described in the scientific reports for each technical work package, namely D4.1 for Secure Enclaves, D5.1 for runtime data protection assurance, D6.1 for data lifecycle security management, and D7.1 for the security and privacy monitoring component.

Some components have been implemented either in full or in part as open source and are publicly available. References to project-produced open source can be found in Appendix A.1. For those components that are proprietary, developers can either create alternative versions based on the published specifications, or contact REST_ASSURED to learn about licensing possibilities.
2.2 Components

2.2.1 TruCE framework

Owner: IBM
License: Apache License 2.0
Link: https://github.com/IBM/sgx-trust-management

Purpose:
Intel’s SGX enclaves add a layer of hardware-based security to cloud computing. However, working with SGX is difficult, and requires a high degree of expertise. To simplify this process, and enable the use of SGX hardware in RESTASSURED-like platforms, IBM developed Trust Management Framework (or TruCE for short - ‘Trust in Cloud Enclaves’). TRuCE handles all aspects of remote attestation and secret delivery process in Intel SGX enclaves. The framework enables application developers to focus on the application code, performing attestation by a simple API call. Additional calls are available for easy-to-use enclave sealing and secret passing to enclaves.

2.2.2 Apache Parquet Modular Encryption

Owner: IBM
License: Apache License 2.0

Purpose: IBM has developed the standard for adding modular encryption to Apache Parquet files. Parquet is the most widely used format in the Apache Hadoop ecosystem. The Apache Parquet community is currently working on the development of the code to implement this standard.

2.2.3 System Security Modeller

Owner: IT Innovation
License: Proprietary
Link: Not open source

Purpose:
System Security Modeller (SSM) is a tool for performing a risk assessment of a system at design-time (i.e. before deployment) in line with ISO 27005. SSM is a web based application with a canvas onto which a user can drag icons representing the assets in a system, and these can be linked together to express how those assets are related. The result is a design-time model of the system. SSM can perform a detailed risk analysis of a design-time model, identify threats to assets, and the consequent risks to the system. The user is able to explore the model to identify the root causes of identified risks, and apply security controls to eliminate or mitigate the threats from which the risks arise. SSM recalculates the risk levels to take account of the security controls the user has selected, and the process is repeated until the risk level is acceptable to the user. The resultant set of security controls (and the assets to which they must be applied) constitutes a detailed set of security requirements that the deployed system should meet.
2.2.4 Runtime Risk Evaluator

**Owner:** IT Innovation  
**License:** Proprietary  
**Link:** Not open source (due to its dependence on SSM)  
**Purpose:**
Runtime Risk Evaluator is a software component whose primary objective is to perform risk analysis of systems at runtime (i.e. during the operation of the system). It is deployed as a software service, usable via a set of REST APIs. These allow initial design time risk models (created using the System Security Modeller SSM) to be uploaded (and updated dynamically) and set the initial context for the running system. Subsequent changes to the system (and/or environment) will trigger potential adaptations of the system’s behaviour e.g. to increase performance or maintain dependability – Risk@Runtime analyses every potential adaptation (input via the risk analysis API) in terms of a new runtime model of the system and calculates the new risk levels of this future state. The component can greenlight the suggested adaptation if it meets an acceptable risk level, or highlight where threats are greatest and need to be further mitigated by an adaptation.

2.2.5 Thales Components

Several assets of the Data Gatekeeper has been developed and used. They are wrapped in REST Web Services, using Jersey ([A.1.4]) framework. They must be deployed in trusted environments.

2.2.5.1 Data Protection Contract Manager

**Owner:** Thales  
**License:** Proprietary  
**Link:** Not open source  
**Purpose:**
The Data Protection Contract Manager is responsible for the registration of the Service providers, in Data Protection Contracts. It collects the type of data that is needed by the Service Providers and their planned usage. The Data Protection Contracts between the Data Gatekeeper and the Service Providers are generated, signed and stored.

2.2.5.2 Sticky Policy Manager

**Owner:** Thales  
**License:** Proprietary  
**Link:** Not open source  
**Purpose:**
The Sticky Policy Manager component is responsible for the registration of the Data Subject. It collects the data subject security preferences through a Graphical User Interface. The Sticky Policy Manager component translates the data subject requirements into Sticky Policies, and bounds them to the personal data. The Sticky Policies are generated in RDF (Resource Description Framework) Format, using the framework Apache Jena ([A.1.3]), released under Apache License 2.0. This component is also responsible for signing and storing the Sticky Policies, in a TripleStore. The TripleStore is also furnished by the framework Apache Jena.
2.2.5.3 Context Manager

**Owner:** Thales  
**License:** Proprietary  
**Link:** Not open source  

**Purpose:**  
The Context Manager component is responsible for context information management. It collects information from the Adaptation, Monitoring, Authentication and Data Protection Enforcement Point in order to generate a Context associated with a query. This context is generated as a Named Graph, in RDF format. It is stored in a Triplestore, furnished by the framework Apache Jena. The Context is extracted and matched in the Sticky Policies in order to compute the set of authorized data related to a query.

2.2.5.4 Data Protection Decision and Enforcement Points

**Owner:** Thales  
**License:** Proprietary  
**Link:** Not open source  

**Open Source Alternatives:** Authzforce: [https://github.com/authzforce](https://github.com/authzforce), Open Policy Agent: [https://openpolicyagent.org](https://openpolicyagent.org)  

**Purpose:**  
The Data Protection Decision Point component is responsible for extracting and combining the organizational access control policies and sticky policies. It will output a response: grant or deny, for the processing of the personal data within the Rest Assured environment. The Data Protection Decision Point component is able to deliver fine-grained access control decisions, based on the individual sticky policies.

2.2.5.5 Authentication Component

**Owner:** 3rd party open source  
**License:** Apache License 2.0  
**Link:** keycloak website: [https://www.keycloak.org](https://www.keycloak.org)  

**Open Source Alternatives:** Any other OpenID Connect implementation: [https://openid.net/developers/certified/](https://openid.net/developers/certified/)  

**Purpose:**  
The Authentication component is an authentication service, using an implementation of the OpenID Connect protocol.
2.3 Building Solution Examples

2.3.1 Enabling run-time data protection assurance

Work package 5 addresses the problem of run-time data protection assurance by detecting changes in data protection risks during run time, mitigating them by appropriate adaptations, and performing other adaptations (e.g., for performance reasons) in such a way that data protection policies remain satisfied. The elaborated solution approaches are described in detail in deliverables D5.1-D5.3.

To build a solution incorporating the elaborated techniques for run-time data protection assurance, the following steps must be made:

- **Refining the meta-model.** The proposed approach for detecting and mitigating data protection risks is based on a meta-model. The run-time model of the cloud system as well as the risk patterns and adaptations depend on this meta-model. D5.1 describes a possible meta-model. This meta-model may have to be customized though, depending on the specific cloud system to be targeted. As an example, the meta-model in D5.1 assumes that application components are deployed in virtual machines. If the given system uses containers instead of or in addition to virtual machines, the meta-model has to be modified or extended accordingly. As described in D5.2, the implementation of WP5 uses the open-source Eclipse Modeling Framework\(^1\) for creating and storing the meta-model.

- **Identification of risk patterns and adaptation rules.** A further design-time activity that is required as preparation for the run-time data protection assurance is the elaboration of a catalog of relevant risk patterns. An initial catalog of risk patterns has been described and made publicly available under [https://restassuredh2020.eu/wp-content/uploads/2018/06/Modelling-Data-Protection-Vulnerabilities-of-Cloud-Systems-using-Risk-Patterns-Technical-Report.pdf](https://restassuredh2020.eu/wp-content/uploads/2018/06/Modelling-Data-Protection-Vulnerabilities-of-Cloud-Systems-using-Risk-Patterns-Technical-Report.pdf). Similarly to the meta-model, this catalog can be used as a basis but may have to be extended or modified according to the specific technologies used and their potential vulnerabilities. For identifying – and continually updating – vulnerabilities, public databases such as the Common Vulnerabilities and Exposures (CVE) database\(^2\) can be used as source. As described in D5.2, the implementation of WP5 uses the open-source Henshin library\(^3\) for creating and storing risk patterns. In addition, also rules for possible adaptations can be captured in the form of Henshin model transformation rules.

- **Implementation of adapters for monitoring and adaptation execution.** Detecting changed data protection risks during run time presupposes that the used run-time model is kept in line with the configuration of the cloud system by means of monitoring. The monitoring system described in D5.1 handles this in a generic way and depends on specific monitoring adapters for the used systems. A monitoring adapter extracts real-time monitoring information from the monitored system and forwards that information to the monitoring gateway of the RESTASSURED Adaptation component. The monitoring adapter has to be implemented in a system-specific way for the used applications and infrastructure management systems, as different systems offer different interfaces or necessitate different monitoring probes to extract information about their state. Similarly, the RESTASSURED Adaptation component uses an execution gateway that offers a generic interface for executing adaptations. This generic interface has to be mapped on specific APIs offered by the systems to be adapted by means of system-specific execution adapters.

- **Optional: integration with run-time management tools.** The RESTASSURED run-time data protection assurance approach can work autonomously, by automatically detecting and mitigating data

\(^1\)https://www.eclipse.org/modeling/emf/
\(^2\)https://cve.mitre.org/
\(^3\)https://www.eclipse.org/henshin/
protection risks. However, in some cases, it can be useful to use an appropriate alarm mechanism to notify the system administrators and provide information about the found risk pattern to them. This can be in the form of a dashboard or by integration with the incident reporting system of the organization.

- **Optional: integration with design-time tools.** The run-time model may be initialized with a deployment plan created during the design phase with the help of some deployment planning tool. One possibility for such integration could be through a standardized language for cloud deployments. The implementation of WP5 currently supports TOSCA (Topology and Orchestration Specification for Cloud Applications) for this purpose.

2.3.2 Integrating the Pay-As-You-Drive Use Case

2.3.2.1 Background

In the past traditional car insurance policies have used basic static information about driver (age, gender, profession and previous claim history) and automobile (brand, model, year of manufacture) upon which to base premiums for insurance. The introduction of vehicle telematics has enabled a number of new *Usage-Based Insurance (UBI)* pricing models to emerge, allowing for a more fine-grained approach to risk assessment to be carried out derived from empirical driving data:

- **Pay-As-You-Drive (PAYD)** describes an automotive UBI product with a pricing model based on distance driven based on telemetric data collected while the car is being driven. The telemetric data set collected for determining the policy premium for such products is often comparatively narrow, mostly limited to odometer readings (distance travelled) and more recently in some cases including geopositioning (GPS) data sets of date, time, speed, direction and location to be used for simplistic usage analysis.

- **Pay-How-You-Drive (PHYD)** describes a more advanced UBI product with a pricing model based on driver behaviour analysis. The telemetric data set collected for determining the policy premium is a far wider telematic data set (with more resolution and accuracy) and it is processed more deeply than traditional PAYD. Not only using more finely sampled location, time of day, and distance travelled, but also taking in telematics streams of car controls (e.g. steering wheel, brakes) instrumentation (tachometer) and other sensors, a driver behavioural profile can be built up algorithmically by back end analysis (such as machine learning) in cloud infrastructures.

Unless an explicit reference is made to a specific model (e.g. PHYD), this document will otherwise use the term PAYD to refer interchangeably to all the above insurance models and associated technical components and back-end cloud analysis models.

2.3.2.2 Compliance and End-User Acceptance Challenges

The key challenges for these models are in the areas of compliance with applicable data protection legislation (including, but not limited to, the GDPR) and in facilitating end-user acceptance. Key concerns considered by the use case, therefore, are:

1. Increasing end-user acceptance by enabling the end-user to control all aspects of data collection, aggregation, and sharing.

2. Ensuring compliance with data processing regardless of changes in physical location of the end-user.

2.3.2.3 Data Model
At the beginning of the project, OpenXC\textsuperscript{\textregistered} developed by Ford and Bug Labs as an open standard for accessing vehicle data, was selected as the basis of the PAYD Data Model. Unfortunately, OpenXC appears to have become largely abandoned throughout the course of the project, and this model has subsequently been built on and extended to support the evolving needs of the use case implementation. With that in mind, compatibility with the original data model is addressed through conversion utilities developed within the project, ensuring that the PAYD implementation is still able to benefit from interoperability with the existing ecosystem of OpenXC tools (such as the OpenXC Vehicle Simulator).

In the case of the PAYD use case, the data model is primarily comprised of the following tables:

**Person** contains basic personal information about the Driver, including name, gender, date of birth, etc.

**Auto** contains general information about the vehicle involved in a Journey, including identifying information and characteristics about the vehicle, such as its:

1. vehicle identification number (VIN), from which the make, model, and year can be derived; and
2. license plate number, from which a range of registration information can be obtained in different jurisdictions.

**Event** contains events that occurred during the course of a Journey. During Phase 1 of the project, this has been limited primarily to simple telematics and sensor data, including geolocational positioning data, accelerometer/gyroscope readings, etc. In Phase 2 of the project, this has been extended to contain data across a range of categories, including locational and biometric.

**Journey** contains a series of Events for a given Journey, involving a Person (the Driver) and an Auto.

and is further elaborated in Figure 2.1 below:

![Figure 2.1: PAYD Conceptual Data Model](http://openxcplatform.com)
2.3.2.4 PAYD Software Architecture & Component Overview

The PAYD software architecture is briefly elaborated in Figure 2.2 below:

**Figure 2.2: PAYD v2 Software Architecture**

**Consent Management** while notably absent as a single logical component, is something that is embedded across a number of other components, and is itself comprised of a number of smaller purpose-specific components:

- The user-facing UI for application embedding and consent receipt generation (as JSON)
- A library (or backend service) capable of producing a signed consent receipt as a JWT-encoded token (as well as providing token introspection capabilities for other services)
- A Data Protection Decision Point based on consent receipt analysis and/or JWT token introspection

Within the context of the PAYD use case, the UI component is embedded directly into the PAYD driver app, while changes in consent result in changes in token issuance/revocation state which can subsequently be decided upon by the Data Protection Decision Point. The state of consents for a given service can at any time be validated and attested by service providers through the PAYD insurer dashboard.

The individual components of the PAYD environment are elaborated in Table 2.1 below:
### Table 2.1: Components developed within the PAYD use case

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>openxc2event</td>
<td>A conversion utility for non-interactive conversion of OpenXC vehicle traces to RESTASSURED event records (JSON), for insertion into the encrypted data store</td>
<td>GPL</td>
</tr>
<tr>
<td>openxc2mysql</td>
<td>A conversion utility for non-interactive conversion of OpenXC vehicle traces and insertion into a MySQL/MariaDB database (may be encrypted or unencrypted)</td>
<td>GPL</td>
</tr>
<tr>
<td>Go-SQL-REST-Driver</td>
<td>A driver for Golang’s database/sql package to support SQL over REST API endpoints, such as the RESTASSURED Query Gateway, or Apache Drill.</td>
<td>Apache 2.0</td>
</tr>
<tr>
<td>Region Router</td>
<td>A router and middleware for region-aware endpoint routing based on GeoIP and ISO 3166-1 alpha-2 country code definition.</td>
<td>Apache 2.0</td>
</tr>
<tr>
<td>PAYD insurer dashboard</td>
<td>The insurer-facing dashboard where risks to be assessed for a specific driver are selected and consent settings for drivers can be verified.</td>
<td>Proprietary</td>
</tr>
<tr>
<td>PAYD driver app</td>
<td>The driver-facing application that presents the latest risk and policy information to the driver, allows them to manage their consent settings, and learn more about their own driving behaviour.</td>
<td>Proprietary</td>
</tr>
<tr>
<td>PAYD API gateway</td>
<td>The main application-facing API gateway that arbitrates access to multiple API/Edge service endpoints.</td>
<td>Proprietary</td>
</tr>
<tr>
<td>Consent manager</td>
<td>A set of components for modelling consent requests, obtaining fine-grained consent from the driver, and informing the creation of sticky policies: The user-facing UI for application and consent receipt generation. A library capable of producing a signed consent receipt as a JWT-encoded token.</td>
<td>Apache 2.0</td>
</tr>
</tbody>
</table>
While the specific implementation within the PAYD driver app and PAYD insurer dashboard remain Proprietary, open source reference implementations have been provided for all components. A reference implementation for the adaptation component, as used by the PAYD API Gateway, and still being refined, will further be released together with an OpenAPI definition before the end of the project. A reference Docker Compose stack for the aforementioned components will also be provided prior to the conclusion of the project in order to facilitate third-party experimentation with open source components and RESTAssured methodologies.

For convenience, all components developed and released as open source by Adaptant will be made permanently available and easily discoverable through Adaptant’s GitHub Account:
https://github.com/orgs/adaptant-labs/teams/restassured/repositories

2.3.3 Risk assessing the WaveCloud Architecture

2.3.3.1 Background

This use case example summarises the process of using the RESTAssured System Security Modeller tool (SSM), as detailed in section A.3.2, to risk assess the WaveCloud architecture.

WaveCloud is a serverless AWS cloud-based solution developed by OCC that stores, analyses, and presents a user’s measurement data through a web portal. It also has an accompanying mobile application, which allows a user to connect to a WaveGo unit, take a measurement, store it locally, and then upload it to the cloud. The details of what a measurement consists of is not relevant here, as we are simply considering the sensitivity of the data.

This process is broken down into the following stages:

- Constructing the WaveCloud model from its assets and connections/relationships
- Applying standard security measures (controls)
- Assessing threats

2.3.3.2 Constructing the Model

The appropriate assets are identified from the Asset Palette and the model is constructed. Asset relationships are then formed in the relevant directions, and thus data paths are created.

For example, the Portal Server asset in Figure 2.3 hosts the Web Portal process which in turn serves and receives Measurement Data. The relationships explain both the logical (e.g. a server hosts a DynamoDB database) and the inferred connections or networks between the components (e.g. a mobile app communicates via the internet, to a API Gateway).
Trustworthiness and impact levels are set to appropriate values for the primary assets, for example the Measurement Data which is considered sensitive, is given a High Impact level for loss of confidentiality.

### 2.3.3.3 Applying Security Measures

Standard security measures are applied, where applicable, to match the current system.

Examples of the controls applied include:

- **AntiMalware** - installed on the host servers
- **Logging** - the WaveGo app, API Gateway and Lambda Functions maintain auditable activity records
- **Redundancy** - as standard in an AWS cloud hosted environment
- **Sandboxing** - processes that have restricted access to the host environment
- **SecureConfig** - the basic security configuration changes applied to a host asset, such as changing default passwords for example (See Figure 2.4)
• SoftwarePatching - software is maintained by the automatic application of software patches on the AWS servers

• X.509 - private keys and certificates are used to establish trusted and secure communication between assets

![SecureConfig control assertion](image)

**Figure 2.4: SecureConfig control assertion**

### 2.3.3.4 Assessing Threats

The constructed model is then validated and the risks calculated. High risk threats are assessed first, Figure 2.5 shows a highlighted threat pertaining to an exploitation of privileges risk at the WaveGo App that threatens the Measurement Data.
Recommended controls can be applied and the risks re-calculated to observe the reduction in risk, adjustments can then be made to the model to simulate design changes and to iteratively assess their impact. The results of this process can then be used to determine what action should be taken in the design of the real system to reduce the risk of threats as identified in the model.
3 Conclusions

This RESTAssured Handbook outlines our progress in developing a replicable way of applying the RESTAssured technologies.

We have described in a hands-on manner how to use the RESTAssured components through a set of implemented use cases. We have ensured a full coverage of the material required to replicate our work through a set of appendices which reference the open source libraries used and the RESTAssured APIs.

As such, the handbook is an important means to achieve uptake and impact in practice. Project dissemination activities to the end of the project include the wide-spread dissemination of the handbook through the Common Dissemination Boost cluster; the participation of partners in European events and internally within our organizations’ research departments and implementation teams.
A Appendices
A.1 References to Open Source Software

A.1.1 Parquet encryption

The changes to Parquet for the IBM-led encryption can be found in the Apache github under: https://github.com/apache/parquet-format/tree/apache-parquet-format-2.3.1 and in: https://github.com/apache/parquet-mr/tree/apache-parquet-1.8.2

A.1.2 Keycloak

Open source implementation of the OpenID Connect protocol. The documentation can be found at: http://www.keycloak.org The source code is available on github at: https://github.com/keycloak/keycloak

A.1.3 Apache Jena

Open source framework for building Linked Data applications. The documentation is available from the website: https://jena.apache.org/ The source code is available on github at: https://github.com/apache/jena

A.1.4 Jersey

Open Source Framework for developing RESTful Web Services in Java. The documentation and latest release are available at: https://jersey.github.io/

A.1.5 Hashicorp Vault

Vault is used as a Key Management System (KMS) in RESTASSURED. Documentation and the latest release code can be found at: https://www.vaultproject.io/

A.1.6 Apache Kafka

Kafka is a distributed streaming platform which is used as to allow for streaming of data from an endpoint (for example, transmitted telematics data in the PHYD use case) to the Data Gateway for capture as encrypted Parquet files. https://kafka.apache.org/

A.1.7 Apache Zookeeper

Zookeeper is used by Kafka to maintain its configuration information. https://zookeeper.apache.org/

A.1.8 Apache Spark

Apache Spark is an analytics engine for Big Data. In particular, RESTASSURED uses the Spark Streaming and Spark SQL components for data capture from a Kafka queue and subsequent data write in Parquet format, as well as an SQL client to query the data from the Parquet files. https://spark.apache.org/
A.2 References to API Documentation

A.2.1 Data Gatekeeper: Data Protection Decision Point

The following endpoints are part of the Data Protection Decision Point component. They are the core intelligence of the Data Gatekeeper, responsible for taking decision on processing and treatment of personal data, depending on context information and personal/individual requirements on privacy. Thus, allowing a fine grained access control respecting privacy requirements of every individual for which personal data are processed.

A.2.1.1 Aggregated Authorization with Context

- GET https://DATAGATEKEEPER-ADDRESS/decisionpoint/services/{serviceID}/usage/{usageID}/aggregatedAuthorizationContext

This endpoint delivers the set of authorized entries for a query through a service. This endpoint is called by the Enforcement Point, implemented as a Query Gateway component, specifying the serviceID and usageID path parameters, as well as the set of concerned entries in the SQL Tables as query parameters. This endpoint also asks the Context Manager to update and compute the related context information. The endpoint answers with the list of authorized entries (may be empty) associated with the registered primary identifier of the service serviceID databases, depending on the computed context. The list of concerned data must be sent as query parameters.

A.2.1.2 Specific Authorization with Context

- GET https://DATAGATEKEEPER-ADDRESS/decisionpoint/services/{serviceID}/usage/{usageID}/dataSubject/{dataSubjectID}/askAuthorizationContext

This endpoint delivers a signed JWT for a query on a specific data subject. This endpoint is called by the Enforcement Point, implemented as a Query Gateway component, specifying the serviceID, dataSubjectID and usageID as path parameters. This endpoint also asks the Context Manager to update and compute the related context information. The endpoint answers with a signed JWT stating which data are allowed for the data subject under the computed context. Thus implementing fine grained authorization. In the case where there is no authorized data, an empty json object is sent.

A.2.1.3 Simple Authorization

- GET https://DATAGATEKEEPER-ADDRESS/decisionpoint/services/{serviceID}/usage/{usageID}/dataSubject/{dataSubjectID}/simpleDecision

This endpoint delivers a Grant or Deny response for a query. This implement a coarse grained access control.

A.2.2 Data Gatekeeper: Sticky Policy Manager

These endpoints are part of the Sticky Policy Manager component. They should be accessible for the Data Subject itself, as they allow to generate, update, delete Sticky Policies linked with personal data owned by the Data Subject. For convenience, some parts of the API can be accessed through a Graphical User Interface.
A.2.2.1 Graphical Representation

- **GET** [https://DATAGATEKEEPER-ADDRESS/decisionpoint/dataSubject/{dataSubjectID}/displayPolicy](https://DATAGATEKEEPER-ADDRESS/decisionpoint/dataSubject/{dataSubjectID}/displayPolicy)

This endpoint displays a graphical representation of the security policy linked with the personal data of dataSubjectID.

A.2.2.2 Service Registry

- **GET** [https://DATAGATEKEEPER-ADDRESS/decisionpoint/listServices](https://DATAGATEKEEPER-ADDRESS/decisionpoint/listServices)

This endpoint lists the services registered on the Data Gatekeeper.

A.2.2.3 Data Subject Creation

- **POST** [https://DATAGATEKEEPER-ADDRESS/decisionpoint/services/request/registerFirst](https://DATAGATEKEEPER-ADDRESS/decisionpoint/services/request/registerFirst)

This endpoint creates a data subject as a graph with personal identifiers in the TripleStore of the Sticky Policy Manager.

A.2.2.4 Creating Sticky Policy

- **POST** [https://DATAGATEKEEPER-ADDRESS/decisionpoint/createFirstPolicy](https://DATAGATEKEEPER-ADDRESS/decisionpoint/createFirstPolicy)

This endpoint populates a security Sticky Policy with the privacy requirements of the Data Subject.

A.2.2.5 Updating Sticky Policy

- **POST** [https://DATAGATEKEEPER-ADDRESS/decisionpoint/updatePolicy](https://DATAGATEKEEPER-ADDRESS/decisionpoint/updatePolicy)

This endpoint updates a Sticky Policy.

A.2.2.6 Delete Sticky Policy

- **DELETE** [https://DATAGATEKEEPER-ADDRESS/decisionpoint/deletePolicy](https://DATAGATEKEEPER-ADDRESS/decisionpoint/deletePolicy)

This endpoint deletes a part or all of the Sticky Policy related to a Data Subject. The identity of the Data Subject must be passed as query parameter.

A.2.2.7 Signature Sticky Policy

- **GET** [https://DATAGATEKEEPER-ADDRESS/signPolicy](https://DATAGATEKEEPER-ADDRESS/signPolicy)

This endpoint sign the Sticky Policy related to a Data Subject. The identity of the Data Subject must be passed as query parameter.

A.2.2.8 Validate Sticky Policy

- **GET** [https://DATAGATEKEEPER-ADDRESS/validatePolicy](https://DATAGATEKEEPER-ADDRESS/validatePolicy)

This endpoint validates the signature of the Sticky Policy related to a Data Subject. The identity of the Data Subject must be passed as query parameter.
A.2.3  Data Gatekeeper: Data Protection Contract Manager

The following endpoints are parts of the Data Protection Contract Manager. They are responsible for registering Service Providers to the Rest Assured Data Gatekeeper data lifecycle manager. They should be accessible to the Service Providers in order for them to register the capabilities and usages of personal data. A Graphical User Interface is furnished to the Service Providers in order for them to register.

A.2.3.1  Display Data Protection Contract

- **GET** `https://DATAGATEKEEPER-ADDRESS/dataprotectioncontract/services/{serviceID}/displayContract`

This endpoint displays the content of a Data Protection Contract.

A.2.3.2  Build Template Contract

- **POST** `https://DATAGATEKEEPER-ADDRESS/dataprotectioncontract/buildContract`

This endpoint get as input Post Parameters and builds a template for data protection contract in the GUI. This template will then be filled by the Service Provider in the generate contract endpoint.

A.2.3.3  Generate Contract

- **POST** `https://DATAGATEKEEPER-ADDRESS/dataprotectioncontract/generateContract`

This endpoint can be based on the template from Build Template Contract endpoint. This endpoint generates a Data Protection Contract from Post Parameters. It checks the generated contract against a XSD scheme, and if valid, an XML signature is generated for the contract. The contract is then stored by the Data Protection Contract Manager.

A.2.4  Data Gatekeeper: Context Manager

The following endpoints are parts of the Context Manager. They are responsible for creating, updating and deleting Context Information, related to queries on personal data. They are accessible from the Data Protection Decision Point.

A.2.4.1  Creating Context

- **POST** `https://DATAGATEKEEPER-ADDRESS/decisionpoint/services/request/createContext`

This endpoint generates a Named Graph related to a Data Subject and a Service ID in which context information is stored in RDF format.

A.2.4.2  Update Context

- **POST** `https://DATAGATEKEEPER-ADDRESS/decisionpoint/services/request/updateContext`

This endpoint updates a Named Graph storing context information with up-to-date context information.
A.2.4.3 Delete Context

- **DELETE** [https://DATAGATEKEEPER-ADDRESS/decisionpoint/services/request/deleteContext](https://DATAGATEKEEPER-ADDRESS/decisionpoint/services/request/deleteContext)

This endpoint deletes a Named Graph storing context information related to a Data Subject and a Service ID.
A.3 Risk Models

For performing a risk assessment for a cloud computing application, it is necessary to identify the primary and secondary assets that play a role in the system and the relationships between them and obtain all relevant information about the context of the considered application. The context analysis of a cloud computing service and its underlying system is a difficult task because of the variety of different types of information that have to be considered. This context information includes: i) Legal, regulatory and/or contractual requirements that are relevant for a cloud computing service (indirect stakeholders), ii) relations to other involved cloud computing services, iii) high-level cloud system components that support the involved cloud computing services, iv) data that is processed by the cloud computing services and v) stakeholders that interact directly with the cloud computing services and/or the underlying cloud system components.

RESTAssured has developed three tools to support this analysis, two for use prior to deployment and the third at run-time:

- The RESTAssured CSAP patterns and associated CSAP tool for applying these patterns to capture the context for a cloud application.
- The System Security Modeller (SSM) tool and associated RESTAssured domain model used to model the high level system composition in terms of related primary and secondary assets.
- The Runtime Risk Evaluator (RRE) tool, which maps between a model of the running system (provided by RESTAssured monitoring and adaptation components) and SSM model variants, using SSM as a run-time risk calculation service.

CSAP is a high-level context-oriented approach that employs patterns to identify the relationships between the system and the stakeholders (see RESTAssured Deliverable D7.3). Here stakeholders include Data Subjects, Data Controllers, and Cloud Providers. CSAP represents a pattern for defining the context of a cloud computing service. Using the CSAP 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 CSAP has been extended to include a new pattern. This new pattern of the CSAP is called RestAssured-Cloud System Analysis Pattern (ReAs-CSAP).

SSM employs graph-based models to model the assets of a system and the relationships between them. A domain specific catalogue of threat patterns captures the possible threats within a domain, and through pattern matching the specific threats in a system model can be identified. Moreover, by specifying the trustworthiness levels of certain attributes of system assets, the likelihood of those threats can be computed. This includes two mechanisms by which the effects of threats can be propagated: automatic secondary effect chaining, and loss of trustworthiness which opens up potential attack paths whereby attackers may penetrate further into the system. These steps allow the likelihood of threats (and more importantly, threat effects) to be computed. Impact levels can then be specified for the effects on primary assets, and combined with the computed likelihood of these effects to obtain risk levels.

RRE also uses graph-based algorithms to map between a model of assets detected in the running system by the RESTAssured monitoring and adaptation components, and an SSM model variant corresponding to the current configuration and obtained by restricting the design-time model (which should in principle encompass all possible run-time configurations). Once assets have been mapped to an appropriate restricted SSM model variant, the RRE component uses SSM as a risk level calculator. This is used in two ways: to update risk levels when the running system undergoes a significant change (e.g. a change in system composition, or in the security measures currently enabled), and to forecast risk levels in advance of any changes that may be proposed by the RESTAssured adaptation system. In either case, the calculated risk
levels are returned to the adaptation system along with information about the threats causing these risk levels, and measures for addressing these threats. The adaptation system can then use this information to formulate system adaptation proposals to address increasing risks, or decide whether a previously proposed change should be implemented in the running system.

The two offline or design-time tools are aligned with the ISO 27005 standard procedure for risk assessment, incorporating risk identification, risk analysis and risk evaluation, as shown in Figure A.1.

For most of the first year, our working assumption was that the RESTASSURED risk assessment methodology would start with a pattern-based approach for context analysis (using CSAP), followed by a model-based approach to identify the primary and secondary assets and relationships (using SSM). The RESTASSURED pattern used with the CSAP approach conform to the General Data Protection Regulation, supports identification of high level assets and stakeholders, and documentation of the scope and boundaries of the cloud computing application or service, conforming to the requirements of the ISO 27005 standard. Then the SSM model captures the system structure in terms of asset relationships, allowing auto-generation of a threat catalogue, and calculation of risk levels taking into account the interdependencies between primary and secondary assets. With this idea in mind, once the first prototypes were available, we started to analyse how to transfer information from CSAP to SSM, and then how to use this information for dynamic (run-time) updating of the risk level calculation.
However, when the RESTASSURED validation partners began investigating how to use these tools, the feedback provided made it clear that the situation is more complex. In reality, the service operator requires information from two other sources in order to carry out the risk assessment:

- From the supplier of (application) software and services they will use, given that the majority of service operators will obtain these from a software vendor or in the form of Software as a Service (SaaS).
- From the cloud resource provider where the application will be hosted.

The overall flow of information is illustrated by Figure A.2. The stakeholders on either side are suppliers to the service operator in the middle. Their input is derived from their own analysis of security risks, but this cannot be synchronised with that of the service operator. In practice, the suppliers will normally conduct their analysis in advance, and disclose information from it as needed to address any concerns raised by the service operator.

The transfer of input between these organisations in a usable form is non-trivial. The starting point for service operators is often the supplier’s ISO 27001 Statement of Applicability which defines which of the supplier’s services are covered by their ISO 27001 compliant information system risk management system. Beyond this, the suppliers normally provide information only as needed to address specific concerns raised by the service operator.

RESTASSURED has therefore developed a more integrated, model-based risk assessment methodology that incorporates security information exchange between organisations. This is shown in Figure A.3.

In most cases, the cloud resource providers are large organisations like Amazon, who provide a significant amount of input, but this is included as standard in manuals and usage agreements and is not tailored for specific service operators. Input from the right of Figure A.3 is therefore a set of standard security properties and usage restrictions from the selected cloud resource provider.

The RESTASSURED methodology for pre-deployment risk assessment is designed primarily to support those in the centre-left of Figure A.3. The starting point is therefore the software/service provider, who has to make assumptions about how their software will be used, and then assess potential risks to determine...
which control measures they should incorporate into their software design. The high level asset interdependency models used by SSM are the most natural starting point, as they can be related more easily to a software architecture and design, making them easier for a software company to create and also to use. The software/service provider is also often best placed to integrate information from a candidate cloud resource provider, e.g. by including security measures provided by the cloud platform in their model.

The resulting model captures the software/service providers assumptions about the use of their products and services, as well as the security measures included by design (and implemented by the software/service provider) or by default (using standard security features from the cloud resource provider). This can be sent to the service operator, allowing them to elaborate, using the RESTAssured CSAP pattern and tools to identify external stakeholders and boundaries, so these can be added to the SSM model to create a model of the initial deployment. This model then becomes the starting point for dynamic updating of the model by the RRE component once the application has been deployed in the cloud.

The CSAP, SSM and RRE tools are described in more detail in RESTAssured Deliverable D7.3.

A.3.1 The CSAP Tool

This tool provides a graphical editor that supports the creation and instantiation of Cloud System Analysis Patterns (CSAP). To this end, the tool provides two editors, namely the Designer-editor and the User-editor. These editors are explained in the following sections.

A.3.1.1 The Designer-editor

The Designer-editor of the CSAP-tool enables designers to create specific Cloud System Analysis Patterns (CSAP). The designers can build a specific cloud system analysis pattern in two different ways:
• by extending the original CSAP (see Figure A.4) or
• by creating an empty CSAP that contains only an Indirect Environment, a Direct Environment and a Cloud.

In case of starting with an empty CSAP, designers can create their own CSAP by adding appropriate types of Indirect Stakeholders, Direct Stakeholders and Cloud Elements. Furthermore, associations between Direct Stakeholders and Cloud Elements as well as between Direct Stakeholders among each other can be created. If the original CSAP is used as basis, already existing CSAP-elements can be modified and/or deleted. Figure A.5 shows the definition of a new Cloud Element that is added to an empty CSAP. In the designer-editor the name of a CSAP-element can be as the same as its instance type (see Figure A.6).

A.3.1.2 The User-editor

In the User-editor, any defined pattern can be instantiated. The instantiation of our ReAs-CSAP is described in D7.1. Figure A.7 shows the instantiation of a Cloud Element of instance type Data during the instantiation of the ReAs-CSAP. This Cloud Element instance specifies customers’ data and therefore is named Customer Data.

The instantiation of the Cloud Element is marked by displaying the type name Data surrounded by angle brackets under the name of the Cloud Element (see Fig. A.8). For a subset of the properties of the instantiated Cloud Element, the corresponding values can be specified in the according dialogue during its instantiation (see Fig. A.7). For all properties, except for the instance type, the corresponding values can be assigned in a property panel (see Fig. A.9).

---

Figure A.5: Adding a Cloud Element and defining its name

Figure A.6: Equality of the name and instance type of an added Cloud Element
Figure A.7: Instantiation of a Cloud Element of the instance type Data in a ReAs-CSAP

Figure A.8: Representation of the instantiated Cloud Element of the instance type Data in a ReAs-CSAP
<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud Element Customer Data</td>
<td></td>
</tr>
<tr>
<td>Considered In Ra</td>
<td>false</td>
</tr>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>Descriptive Doc</td>
<td></td>
</tr>
<tr>
<td>Instance Type</td>
<td>Instance Type Data</td>
</tr>
<tr>
<td>Is Asset</td>
<td>true</td>
</tr>
<tr>
<td>Is Instance</td>
<td>true</td>
</tr>
<tr>
<td>Is Refining</td>
<td>false</td>
</tr>
<tr>
<td>Is Service Layer</td>
<td>false</td>
</tr>
<tr>
<td>Location</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>Owner</td>
<td></td>
</tr>
<tr>
<td>Rationale For Not Considered In Ra</td>
<td></td>
</tr>
<tr>
<td>Rationale For Not Is Asset</td>
<td></td>
</tr>
<tr>
<td>Uuid</td>
<td>_1831EDHwaEiIMfWht_pOxjg</td>
</tr>
</tbody>
</table>

Figure A.9: Definition of Properties of the instantiated Cloud Element of the instance type Data
A.3.2 System Security Modeller

System Security Modeller (SSM) is a web based tool for performing design-time risk assessment of systems.

A.3.2.1 Constructing the Initial Model

The tool has a canvas on which the user can create a model of the system by dragging assets from a palette [Fig. A.10 – (1)] on to the canvas [Fig. A.10 – (2)] and adding the relationships between them [Fig. A.10 – (3)]. Once this process is complete SSM applies machine reasoning techniques to:

- Check that the model is syntactically correct.
- Added certain inferred assets that exist by virtue of the relationships between the assets. For example this can include logical entities like paths between networks.
- Find threats that exist within the model using a knowledge base of threat patterns.

A.3.2.2 Identifying Primary Assets

The next step is to identify the primary assets in the system, and specify the impact levels for the possible misbehaviours of those assets. By default SSM assigns relatively low impact levels to asset misbehaviours. This is entirely appropriate for supporting assets (as per ISO 27005), but not appropriate for the primary assets of the system. In Fig. A.11 the Subject Data has been identified as a primary asset, and Loss of Confidentiality has been given a High impact level [Fig. A.11 – (1)].
A.3.2.3 Applying System Specific Knowledge

The threats identified by SSM have likelihoods, and these in turn determine the likelihood of the asset misbehaviours that they cause [Fig. A.11 – (2)]. The combination of the impact level of an asset misbehaviour, and the corresponding likelihood of that misbehaviour, determine the risk of that misbehaviour [Fig. A.11 – (3)]. (See Section 3.2.5 of D7.1 for details of this calculation.)

The threat likelihoods are determined by the trustworthiness of assets, or more specifically, by the trustworthiness levels of certain attributes of those assets. Again, SSM chooses default values for these attributes, but the user can adjust these based upon their understanding of the system they are modelling. For example in Fig. A.12, which is taken from a model of the SCANT use case (Refer to D8.2), the AMI LAN is provisioned in a public cloud. As this may be shared with other tenants outside of the control of the SCANT operator, the user trustworthiness level is set to Low [Fig. A.12 – (1)].
A.3.2.4 Adding Standard Security Measures

At this stage SSM will likely have identified a large number of potentially high risk threats. The next step is to systematically apply standard security measures:

- Software patching of all hosts (physical or virtual).
- Authenticate and encrypt all client-service communications.
- Use encrypted storage for the sensitive data.

In Fig. [A.13] we have applied these controls to the SCANT use case model. SSM’s reasoning determines that this addresses most of the threats, but leaves one high risk security threat [Fig. A.13 – (1)], and one GDPR compliance threat [Fig. A.13 – (2)].

Figure A.13: Remaining threats in the SCANT use case after applying standard controls
A.3.2.5 Treating High Risk Threats

The high risk security threat can be examined in the Threat Explorer [Fig. A.14 – (1)], where we see that this is a primary threat. Primary threats have entry points corresponding to trustworthiness attributes of the involved assets (see Section 3.2.2 of D7.1 for more details). Here the relevant trustworthiness attribute is the trustworthiness of the users of the Spark DB. Something is driving the trustworthiness level down from Very High to only Medium. We can investigate this further using the Misbehaviour Explorer [Fig. A.14 – (2)], where we see another threat that is the root cause of the threat that we are investigating [Fig. A.14 – (3)].

![Figure A.14: Finding the root cause](image)

The root cause threat arises from the fact that the AMI Server is provisioned in a public cloud. The management of the server is outside the control of the SCANT operator, and thus the data processed by the Spark DB hosted on the AMI Server is potentially exposed. The available control strategy is run the SparkDB in a secure enclave on the AMI Server [Fig. A.15 – (1)].
A.3.2.6 Treating GDPR Compliance Threats

The GDPR compliance threats can be examined in the Compliance Explorer [Fig. A.16 – (1)], and each individual threat examined in turn [Fig. A.16 – (2)]. For the SCANT use case, SSM has identified that the Spark DB may process the Subject Data in a way that is not compliant with the GDPR. The available control strategy is to apply Sticky Policies to the Subject Data, and enforce them at the Spark DB [Fig. A.16 – (3)].
A.3.3 The Runtime Risk Evaluator

The Runtime Risk Evaluator (RRE) is the last and therefore the least mature risk assessment tool produced by REST Assured. It is designed to be used with the REST Assured adaptation framework, which is also a research prototype framework. The main components of the RRE are shown in conjunction with adaptation components in Figure A.17.

Figure A.16: Treating the GDPR compliance threat

Figure A.17: Run-time risk assessment in RestAssured
The purpose of the RRE tool is to take as input a run-time system model supplied by the adaptation components, and report back the risk level for the configuration described by the run-time system model, and a list of threats giving rise to the calculate risk level, along with possible control strategies. The run-time system model provided will either be a snapshot of the current system following a dynamic change that may cause a change in the risk level, or a proposed new configuration that would be the result of a system adaptation being considered by the adaptation system.

Both the input and output must be related to assets described in the run-time system model, in order that the adaptation system can interpret the results and decide what action should be taken. The tasks performed by the RRE tool itself are therefore:

1. Analysis of the assets defined in the run-time model, to determine which of the restricted variants of the original SSM design time model most closely matches the current configuration.

2. Mapping between the run-time model assets, and assets defined in the selected SSM model variant.

3. Passing any changes in risk factors (estimates of asset trustworthiness, and the monitored status of security measures) to SSM, and invoking its risk level calculation algorithm.

4. Extracting information from the selected SSM model variant describing the risk level and relevant threats and mapping back to the run-time model assets before sending results to the adaptation planner component.

The only graphical user interface used by RRE is SSM itself. This provides the risk dashboard component shown in Figure A.17, which is also the means by which information can be introduced manually, e.g. if security measures are used whose status may change but cannot be monitored by the adaptation components.
A.4 Data Lifecycle

In the RESTASSURED project, the Data Gatekeeper manages the data protection policies and the services governing the data life-cycle. It is responsible for deciding, based on the available policies and various constraints such as legislation, context, location, etc., which operations are allowed on which piece of data and to enforce user data protection policies in a decentralized computational environment. The objective of the Data Gatekeeper is to securely manage the complete data lifecycle in a decentralized cloud setting.

The Data Gatekeeper works as follow: Service provider offers a contract, specifying its service and the constraints it applies to the data. In parallel, the data subject specifies its requirements on the processing of its data. Both specifications are combined and encoded in a sticky policy. The sticky policy is attached to the data. The security policy is thus consulted before each use of the data, and a corresponding context is computed. The combination of Context information and Sticky Policy generate a decision on data access.

The Data Gatekeeper is composed of 6 logical components:

1. The Data Protection Contract Manager is responsible for the registration of the Service providers, in Data Protection Contracts. It collects the type of data that is needed by the Service Providers and their usage (see figures A.18 and A.19). The Data Protection Contracts between the Data Gatekeeper and the Service Providers are generated, signed and stored, as shown in figure A.20.

![Data Protection Contract](image-url)
2. The Sticky Policy Manager component allows a Data Subject to register its preferences on the processing of its personal data, as shown in figure A.21. In order to specify the possible options for data processing, the structure of the Data Protection Contract of the requested service is used. The Data Subject can verify the authorizations that he/she delivers to a service as seen in figure A.22 and update these preferences. These preferences are translated into sticky policies that are logically bounded.
to personal data and checked before processing the data. The Sticky Policy Manager is able to sign and store sticky policies.

Figure A.21: Change Preferences

Figure A.22: Display Authorization

3. The Data Protection Decision Point component is responsible for combining the various organizational access control policies, sticky policies and context information in order to grant or deny the
processing of the data. The decision is forwarded to the Privacy Enforcement Point component.

4. The Data Protection Enforcement Point component is responsible for intercepting the request for personal data from Services, forwarding the request to the Data Protection Decision Point component and applying the decision made by this component. A Data Protection Enforcement Point is locally deployed for each Service, in front of each databases storing personal information.

5. The Authentication component is responsible for authenticating a data consumer requesting data through a service. When a data consumer logs in on a service with authentication delegation, he is directed to the Authentication component. He logs in on the Authentication component and is redirected to the service, being authenticated. The Authentication component is an authentication service, using an implementation of the OpenID Connect protocol.

6. The Context Manager Component is responsible for storing updating and retrieving context information related to the Rest-Assured system, in order for the Data Protection Decision Point to take decision based also on information about the environment of the queries. Depending on the context associated with a query, different filters on personal data are triggered and applied to the decision.
A.5 Configuring a RESTAssured-like solution for capturing data as encrypted Parquet files

The IBM Pay-How-You-Drive scenario as seen in the video [https://www.youtube.com/watch?v=IrcN-mcg7ds](https://www.youtube.com/watch?v=IrcN-mcg7ds) demonstrates how streaming telematics data can be securely transmitted to a RESTAssured system running in a secure enclave, and consequently written out as encrypted Parquet files. All the software required to run this scenario exists as Open Source - either from third parties or as project-produced Open Source. (See Sections A.1 and 2.2).

A.5.1 Starting Kafka and Zookeeper

From the Zookeeper main directory:
> bin/zoopkeeperserverstart.sh config/zookeeper.properties &

From the Kafka main directory:
> sudo bin/kafkaserverstart.sh /kafka/2.11.1.0/config/server.properties

A.5.2 Setting up Vault for key management

1. Add to ~/.bashrc: export VAULT_ADDR="http://127.0.0.1:8200"
2. Confirm that there is a config.hcl file, eg:

```yaml

backend "file" {
  path = "data"
}

listener "tcp" {
  tls_disable = 1
}

mlock_disable=true
```

Start up the Vault server:
> vault server config=config.hcl &
> vault operator init

The previous command will give the Vault root token and unseal keys. Save them in a secure location. Use the unseal keys with:
> vault operator unseal (do this 3 times with different keys) then -
> vault secrets enable kv

Set up the encryption keys in Vault:
> run VaultInterface.java with required changes for vaultToken and keys in keyMap.put etc. (See IBM files)

Configure PyCharm for Spark - set SPARK_HOME to the top-level Spark directory
Add pyspark.py jar FOR VERSION 2.30
Edit the ~/spark2.3.0/conf/spark-defaults.conf file for correct paths

Start the IBM Data Gateway:
run data_gateway_encrypted.py

Connect the streaming data source to the Kafka queue and start streaming. Files will now be captured as
encrypted Parquet files.