Rest Assured

SECURE DATA PROCESSING IN THE CLOUD

Deliverable D8.2
D8.2 - First Validation Results
Release 0.2

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      : D8.2 - First Validation Results
Nature of Deliverable     : External Report
Dissemination Level       : Public
Contractual Delivery Date : 30 June 2018
Actual Delivery Date      : 30 June 2018
Contributing WPs          : WP3, WP4, WP5, WP6, WP7, WP8
Editor(s)                 : Paul Mundt, Jodie Hatfield
## Contributors

<table>
<thead>
<tr>
<th>Name</th>
<th>Organization</th>
<th>Sections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paul Mundt</td>
<td>Adaptant</td>
<td>Sections 1–5</td>
</tr>
<tr>
<td>Jodie Hatfield</td>
<td>Adaptant</td>
<td>Sections 1–5</td>
</tr>
<tr>
<td>Zoltan Mann</td>
<td>UDE</td>
<td>Sections 2.2.2, 4</td>
</tr>
<tr>
<td>Eliot Salant</td>
<td>IBM</td>
<td>Section 2.2.1</td>
</tr>
<tr>
<td>James Greig</td>
<td>OCC</td>
<td>Section 3.2.2</td>
</tr>
<tr>
<td>Mike Surridge</td>
<td>IT Innovation</td>
<td>Section 2.2.4</td>
</tr>
<tr>
<td>Toby Wilkinson</td>
<td>IT Innovation</td>
<td>Section 2.2.4</td>
</tr>
</tbody>
</table>
## Document History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>V0.1</td>
<td>01.01.2018</td>
<td>Document created with initial layout</td>
</tr>
<tr>
<td>V0.2</td>
<td>30.06.2018</td>
<td>First draft for cross-WP peer review</td>
</tr>
</tbody>
</table>
Contents

List of Figures ........................................................................................................... 6
List of Tables .............................................................................................................. 7
List of Abbreviations ................................................................................................. 8

1 Introduction ............................................................................................................. 9
  1.1 Purpose and Scope ............................................................................................... 9
    1.1.1 Deferral of the HPC Use Case ...................................................................... 10
    1.1.2 Alignment with Project Phases and Architecture Version ......................... 11

2 Software Component Validation ............................................................................ 12
  2.1 Approach ............................................................................................................. 12
  2.2 Technical Work Package Validation Results ..................................................... 13
    2.2.1 Validation of WP4: Secure cloud data processing and execution environments .................................................................................. 13
    2.2.2 Validation of WP5: Run-time data protection assurance ............................ 13
    2.2.3 Validation of WP6: Decentralized data life-cycle management .................. 14
    2.2.4 Validation of WP7: Engineering for run-time data protection .................. 15
  2.3 Evaluation Summary .......................................................................................... 15

3 Detailed Use Case Validation ................................................................................ 16
  3.1 Approach ............................................................................................................ 16
  3.2 Use Case Evaluations .......................................................................................... 17
    3.2.1 Validation of the Pay-As-You-Drive Insurance (PAYD) Use Case ............... 17
    3.2.2 Validation of the Self-directed Social Care (CARE) Use Case .................... 23
    3.2.3 Validation of the High Performance Computing (HPC) Use Case .............. 27
  3.3 Evaluation Summary .......................................................................................... 27

4 Feedback Cycle and Next Steps .......................................................................... 28
  4.1 Use Case Contributions to the RestAssured Architecture .................................. 28
  4.2 Use Case Requirements for the Next Phase ....................................................... 28

5 Conclusion ............................................................................................................. 29

References ............................................................................................................... 30
List of Figures

Figure 1.1: RestAssured Project Phases ................................................................. 11

Figure 3.1: PAYD Conceptual Data Model .......................................................... 18

Figure 3.2: PAYD Software Architecture ........................................................... 20

Figure 3.3: SCANT Conceptual Architecture ..................................................... 24

Figure 3.4: SCANT Software Architecture ....................................................... 25
List of Tables

Table 2.1: Validation objectives for technical work packages .................................................. 12

Table 3.1: RestAssured Use Case Characteristics ................................................................. 16

Table 3.2: Components of the PAYD use case testing environment .................................. 19

Table 4.1: Common Use Case Requirements for Phase 2 ..................................................... 28

Table 4.2: Specific Use Case Requirements for Phase 2 ....................................................... 28
List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>CARE</td>
<td>Self-directed Social Care</td>
</tr>
<tr>
<td>CSAP</td>
<td>Cloud System Analysis Pattern</td>
</tr>
<tr>
<td>GDPR</td>
<td>Regulation (EU) 2016/679</td>
</tr>
<tr>
<td>HPC</td>
<td>High Performance Computing</td>
</tr>
<tr>
<td>PAYD</td>
<td>Pay-As-You-Drive Insurance</td>
</tr>
<tr>
<td>PHYD</td>
<td>Pay-How-You-Drive Insurance</td>
</tr>
<tr>
<td>SCANT</td>
<td>Social Care Analysis of Needs Tool</td>
</tr>
<tr>
<td>SGX</td>
<td>Intel Software Guard Extensions</td>
</tr>
<tr>
<td>SSM</td>
<td>System Security Modeller</td>
</tr>
<tr>
<td>UBI</td>
<td>Usage-Based Insurance</td>
</tr>
</tbody>
</table>
1 Introduction

Purpose and Scope

The purpose of this document is to present the preliminary validation results of the RestAssured solution and technologies, both individually as standalone components, as well as through their integrated application in a diverse set of project use cases, building on the initial use case validation plan outlined in Deliverable D8.1[7].

A holistic validation approach is realized through the inclusion of both component-level and use case-level validation, while also considering the extent to which the RestAssured solutions are able to address the evolving requirements of the use cases, and the extent to which the use cases contribute to the overall technical direction of the project.

To this extent, a number of unique use cases from different application domains have been pursued. Each use case has been selected both for its own application- and domain-specific data protection concerns, as well as the differences in cloud architecture in order to demonstrate the flexibility of the RestAssured components in different deployment scenarios (these are elaborated in Table 3.1 but for convenience are briefly summarized below):

- **PAYD** Pay-As-You-Drive Insurance concentrates on securing data streams from connected car sensors, edge devices and untrusted cloud infrastructures used as telematics for usage based automotive insurance with emphasis on compliance to data protection regulations, such as Regulation (EU) 2016/679 (GDPR)[1].

- **CARE** Self-directed Social Care concentrates on securing sensitive personal data of vulnerable adults living at home across untrusted, decentralised cloud infrastructures with emphasis on compliance to data protection regulations, such as the GDPR.

- **HPC** High Performance Computing concentrates on securing intellectual property and data integrity across the data lifecycle utilising untrusted infrastructure from cloud service providers where commercial competitors may also be clients.

Validation of the individual software components developed within the technical work packages, together with an assessment of overall requirements and the extent to which they have been satisfied is elaborated in Chapter 2, a detailed drill-down into the use case environments, their implementation, and their validation results are provided in Chapter 3 with the feedback cycle between versions of the architecture finally elaborated in Chapter 4.
Deferral of the HPC Use Case

Use Case Prioritization for the GDPR

With the coming into force of the GDPR during this stage of the project, a strategic decision was made to prioritize the use cases that involve the processing and handling of personal data such that market barriers can be reduced and a higher degree of innovation potential can be realized by the consortium members carrying out the implementation of the PAYD and CARE use cases.

SGX Scalability and Performance Limitations

Current scalability and performance limitations in the current Intel Software Guard Extensions (SGX) technology identified early in the project has also limited the extent to which the efficacy and appropriateness of secure enclave-based approaches in addressing the data protection concerns of High Performance Computing workloads can be assessed. Alternative implementations of secure enclaves (such as those being developed by AMD) have gradually come to market during the first phase of the project, and are expected to provide a more appropriate starting point upon which to begin the implementation of the HPC use case.

⚠️ HPC Use Case Rescheduling for Phase 2

Owing to the prioritization of use cases with an emphasis on the processing of personal data on account of the GDPR coming into force during Phase 1 of the project, and the scalability limitations identified in the current SGX technology, the High Performance Computing use case, including its validation planning, will be deferred until the second half of the project, and will be elaborated in the D8.3 Final Validation Plan.
Alignment with Project Phases and Architecture Version

While the RestAssured use case and validation activities span across the entire project life-cycle, in order to align with the project release methodology and the incremental evolution of the overall architecture, use cases are elaborated with increasing levels of complexity and validated against each version of the architecture.

A central objective of WP8: Use Cases and end-user validation is to feed back use case requirements and validation results and align regularly and iteratively with both the overall architecture of RestAssured (WP3) as well as at the component level, through the core technology development work packages (WP4 - WP7). This objective is spread out across project development phases (Phase 1 and 2), with the use cases providing two cycles of feedback and requirements shaping prior to the final maturation phase of the project (Phase 3). The project phases and their relation to the project timeline are further outlined in Figure 1.1.

The project phases, as outlined in the description of work, have the following scope:

- **Phase 1 (run-time observation)**: Establish fundamental, baseline solutions (such as enhancing cloud computing with SGX, the observation of privacy policy violations, as well as handling and use of static sticky policies);

- **Phase 2 (run-time re-configuration)**: Focus on the more challenging and advanced aspects of RestAssured, including the use of the security enhanced cloud for data-intensive computation (e.g. Spark), the mitigation and prevention of privacy policy violations by cloud reconfiguration, as well as dynamic sticky policy management;

- **Phase 3 (stabilization and packaging)**: dedicated to stabilizing and packaging the RestAssured solutions.

*This Deliverable principally focuses on validating Phase 1 results and providing feedback for Phase 2 of the project.* These results, in turn, will form the basis of the final validation plan (D8.3), against which the Phase 2 results will be validated. A final feedback cycle will occur when the Phase 2 results are validated, allowing for a final round of feedback to be incorporated during Phase 3.
2 Software Component Validation

Approach

In accordance with the high-level objective of Phase 1 (run-time observation) (see Section 1.1.2 for reference) to provide tools and methods that can be directly used for the run-time observation and compliance assessment of cloud systems, this Chapter aims to assess the extent to which the technical work packages have both contributed to this goal and the establishment of a fundamental technological baseline for the project upon which exploitation can be realized. More specifically, this includes the enhancement of Cloud Computing with SGX-based Secure Enclaves, the observation of privacy policy violations, and the use of static sticky policies in governing and controlling data access.

In pursuit of the overarching Phase 1 objective, validation of the software components developed within the technical work packages has been carried out in a couple of different ways:

- Each of the technical work packages (WPs 4-7) have identified a set of specific requirements and validation objectives for the first phase of the project (these are further elaborated in Table 2.1 below).
- Each of the use cases have integrated and built on a subset of the RestAssured componentry in different ways in line with their specific requirements. Use case-specific utilization of RestAssured components is further elaborated in Chapter 3.

<table>
<thead>
<tr>
<th>WP#</th>
<th>Description</th>
<th>Phase 1 Validation Objective(s)</th>
</tr>
</thead>
</table>
| WP4 | Secure cloud data processing and execution environments | • Validate the security of the Secure Enclaves  
• Validate the operations of the Query Gateway and Data Gatekeeper |
| WP5 | Run-time data protection assurance | • Run-time detection of data protection vulnerabilities |
| WP6 | Decentralized data lifecycle management | • Generation and use of static sticky policies in controlling data access |
| WP7 | Engineering for run-time data protection | • Run-time detection of data protection and security risks |

The following sections (Section 2.2.1 – Section 2.2.4) look at the specific validation results of each of the technical work packages, with each section providing specific information about how these criteria have been addressed.
Technical Work Package Validation Results

Validation of WP4: Secure cloud data processing and execution environments

Validation of the components in WP4, Secure Cloud Data Processing and Execution Environment, has two aspects:

- Validate the security of the secure enclaves, and
- Validate the operations of the Query Gateway and Data Gatekeeper

The secure enclaves, based on SGX technology, have been verified by showing that data which is written into the enclave can be correctly read back out. Additionally, data protection itself was demonstrated by:

1. doing a dump of the data-at-rest, which is encrypted by an SGX installation running on a secure local machine and then stored on the cloud, and
2. for data-in-processing, by doing a dump of running process memory for Opaque Spark.

In both cases, it was verified that the data was in fact encrypted and hence protected.

Validation of WP5: Run-time data protection assurance

The results achieved in WP5 in the first project period have been validated in multiple ways:

- The risk pattern approach for detecting data protection violations in the cloud was validated by applying it first to a generalized version of the RestAssured use cases as captured by the data-flow view of D3.1[6]. This application of the approach was described in the publication Schoenen et al.: Using risk patterns to identify violations of data protection policies in cloud systems, 13th International Workshop on Engineering Service-Oriented Applications and Cloud Services (WESOACS) 2017[9].

- The meta-model for cloud run-time models was validated by applying it to the same generalized version of the RestAssured use cases. This application of the meta-model is described in the publication Mann et al.: Towards a run-time model for data protection in the cloud, Modellierung 2018[5].

- Both the risk pattern approach and the cloud meta-model were further applied specifically to both the Pay as You Drive and the Self-Directed Social Care use cases. This validation step is described in detail in deliverable D5.1[3], Chapter 4.

- To further validate the applicability of the risk pattern approach and the cloud meta-model, they have been applied to create a catalog of risk patterns based on data protection vulnerabilities described in the literature. The resulting risk pattern catalog, consisting of 45 risk patterns, as well as the methodology to create them and the lessons learned are described in the Modelling Data Protection Vulnerabilities of Cloud Systems using Risk Patterns technical report[8] available on the RestAssured project page[1].

All these validation steps reinforced the conclusion that the proposed risk pattern approach, together with the underlying meta-model, are applicable to detect data protection vulnerabilities at run time.
Validation of WP6: Decentralized data life-cycle management

The outputs from WP6 in Phase 1 are:

- A methodology for managing the Data Life Cycle in a multi-tenant environment.
- An Ontology for defining sticky policies.
- A **Decision Point**, which interprets the sticky policies and provides a decision for queries concerning several data subjects, or fine-grained decisions for queries concerning individual data subjects.
- A **Data Protection Contract Manager**, able to generate usage contracts defining the processing of personal data by Service Providers.
- A **Sticky Policy Manager**, able to generate, update, display and transpose the security requirements of data subjects into sticky policies.

These outputs are integrated in the RestAssured environment for further validation testing by the CARE and PAYD use cases. A detailed look at the modeling of sticky policies within each of the use case environments is provided in Deliver D6.1[2] and so is not repeated here.

For the validation of the individual components developed within WP6, the following validation steps have been carried out:

1. The **Data Protection Contract Manager** generates a **Data Protection Contract** that reflects the different usages of personal data carried out by each use case.

2. The **Sticky Policy Manager** generates and allows updating **Sticky Policies** for data subjects in each of the project use cases. Data subjects can specify their preferences and consent settings based on the planned usages outlined by the Service Providers in the **Data Protection Contract**. These settings are then translated into **Sticky Policies**, following the defined **Ontology**.

3. The **Decision Point** is used in the RestAssured environment by the **Query Gateway**, in order to filter the authorized personal data.

4. The **Decision Point** takes **Decisions** based on the content of the **Sticky Policies**, and forwards this decision to the **Query Gateway**, in order to enforce this decision.

5. The **Data Protection Contracts** are matched against an XSD schema, and then signed with an XML signature (the Sticky Policies can also be signed, in order to validate the integrity of the policies).

**Validation Summary**

The registration of a new **Data Protection Contract** has been tested, resulting in valid contracts against the XSD schema, and has been used successfully by the **Sticky Policy Manager**. The generation and update of static sticky policies has been tested and validated in each of the use cases The **Decision Point** has been tested with each of the use cases, resulting in different decisions being taken when policies are updated or modified (reflected in the CARE use case), or when queries involve different sources of data for which the data subject has defined additional restrictions (reflected in the PAYD use case).

As a result of this, the stated Phase 1 objective for WP6 (as per Table 2.1) can be seen to have been successfully achieved.
Validation of WP7: Engineering for run-time data protection

The primary outputs of WP7 (phase 1 of RestAssured) are two tools:

- **Cloud System Analysis Pattern (CSAP)**, which takes a high-level approach to modeling a system within its context.
- **System Security Modeller (SSM)**, which takes a more detailed view of a system, capturing the assets of the system, the relationships between them, and the threats to those assets.

CSAP has been validated by applying it to the CARE and PAYD use cases. The application to the CARE use case is described in Sections 3.1.2.4 and 3.1.2.5 of D7.1[10], and the application to the PAYD use case in Section 3.1.2.6 of D7.1.

SSM is a reasonably mature tool that has been validated on a previous H2020 project (5G-ENSURE[2]). The extensions and enhancements to SSM developed in Phase 1 of RestAssured (described in Sections 3.2 and 4.2 of D7.1) primarily concern the computation of risk-levels for identified threats. The main challenge addressed is the inclusion of secondary effect cascades and attack paths in the algorithms used to evaluate risks from each threat. This work has been validated by constructing a detailed model of the CARE use case, in conjunction with OCC, and verifying that the risks found, if left untreated, represent major security concerns in that use case.

**Evaluation Summary**

Based on the criteria set forth for Phase 1 evaluation, the technical work packages, in close cooperation with the PAYD and CARE use cases can be seen to have achieved their stated objectives (as per Table 2.1 above). Furthermore, during the evaluation of each work package, and the component-level integration and validation by the use cases, specific limitations were further identified (the details of which are further explored in the following chapter on use case validation - Chapter 3), providing a firm basis upon which to enter into Phase 2 of the project and upon which to validate the continuing maturity of individual components throughout the remainder of the project life-cycle.
3 Detailed Use Case Validation

Approach

Use case validation of RestAssured is carried out across a set of distinct and diverse use cases, involving actual end-users, which have been selected for their unique characteristics, including data protection concerns (protection of intellectual property vs. protection of personal data and privacy), forms of distributed cloud architectures (within the data centre, federated, as well as along the whole compute continuum / fog), and variety of involved stakeholders. Table 3.1 below provides a brief overview of the unique characterization of each of the RestAssured use cases:

Table 3.1: RestAssured Use Case Characteristics

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Data Protection Concern</th>
<th>Cloud Architecture</th>
<th>Involved Stakeholders</th>
</tr>
</thead>
<tbody>
<tr>
<td>CARE: Self-directed Social Care</td>
<td>Privacy and de-anonymization (sensitive personally identifiable data)</td>
<td>Public, decentralized cloud</td>
<td>• Vulnerable adults living at home&lt;br&gt;• Social care providers</td>
</tr>
<tr>
<td>PAYD: Pay-As-You-Drive Insurance</td>
<td>Privacy (personally identifiable data)</td>
<td>Sensors, edge devices, and cloud</td>
<td>• Drivers&lt;br&gt;• Telematics providers&lt;br&gt;• Insurance providers</td>
</tr>
<tr>
<td>HPC: High Performance Computing for Commercial Use</td>
<td>IPR (business sensitive data)</td>
<td>Data centre (possibly federated), hybrid cloud</td>
<td>• Commercial enterprises&lt;br&gt;• Supercomputing centres&lt;br&gt;• Cloud providers</td>
</tr>
</tbody>
</table>

The breadth of the use cases and their validation demonstrate the applicability of the RestAssured framework to many other business value networks and data lifecycle models, and form a set of representative case studies where data passing through multiple stakeholders can be secured in public cloud infrastructures whilst maintaining security standards for the treatment of sensitive data, as well as compliance to data protection regulations such as the GDPR.

The following sections look at the specific validation results of each of the use cases. A brief description of each use case is provided, together with a number of scenarios that demonstrate the main tensions of the use case, and the way in which the application of RestAssured technologies allows the use case challenges to be addressed. The results of the validation are then briefly summarized, together with an overview of the specific test environment in which the use case was constructed and validated. Finally, any next steps for the use case are identified and elaborated.
Use Case Evaluations

Validation of the Pay-As-You-Drive Insurance (PAYD) Use Case

Use Case Description

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:

- **PAYD**: Pay-As-You-Drive Insurance 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.

- **PHYD**: Pay-How-You-Drive Insurance 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.

PAYD is used by following main classes of user:

**Insurance Users** involved in the assessment of risk and the provisioning of risk-adjusted policies based on empirical driving performance of personal drivers. Wishes to run typical Usage-Based Insurance (UBI) queries periodically on the drivers' driving data in order to adjust the driver's risk score.

**Drivers** of vehicles interested in the use of vehicle telematics to reduce their insurance premiums, but that want additional control over how their personal data is collected and used leveraging their new-found rights under the GDPR, as well as in finding out more about their own driving performance.

Data Model

OpenXC[^1], developed by Ford and Bug Labs as an open standard for accessing vehicle data, has been selected as the basis of the PAYD Data Model, as this provides direct interoperability with a growing ecosystem of both open source software and hardware for interacting with vehicle data, as well as access to advanced instrumentation data that will be explored in Phase 2 of the project.

[^1]: http://openxcplatform.com
In the case of the PAYD use case, the data model involves 3 tables:

**Person** contains sensitive 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 about the vehicle, such as its vehicle identification number (VIN).

**Event** contains telemetry that forms part of a Journey, including sensitive information such as geolocational positioning data, speed, acceleration and braking patterns, etc.

and is further elaborated in Figure 3.1:

![Figure 3.1: PAYD Conceptual Data Model](http://restassuredh2020.eu/)

Of particular note is that while each of these tables contains different types of data, with the exception of the Person data, none of the Auto or Event data is inherently a threat to the individual’s privacy until such a time that the data is linked by a join operation across the tables. In RestAssured, not only is the sensitive data encrypted, the entire series of inner joins that create the opportunity to single out the individual through linkability or inference across the tables within a single complex query is run in an encrypted environment, with only the results explicitly consented to by the driver being returned in the query result.

**Purpose limitation** is achieved by requiring each query to be run on a targeted service endpoint in the Query Gateway for which the purpose of data access and collection is codified as part of the requesting URI - assessed and decided upon by the Data Gatekeeper. The data subject is therefore able to pre-consent to a number of different purposes and usages, providing fine-grained control over the types and amount of data that can be accessed on a case-by-case basis. The requester then knows precisely what can be done with the data, and in which context, facilitating their compliance with the GDPR.
The individual components of the PAYD environment are elaborated in Table 3.2 below:

**Table 3.2: Components of the PAYD use case testing environment**

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>openxc-vehicle-simulator</td>
<td>An application for generating simulated OpenXC vehicle trace data in real-time</td>
</tr>
<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>
</tr>
<tr>
<td>openxc2mysql</td>
<td>A conversion utility for non-interaction conversion of OpenXC vehicle traces and insertion into an unencrypted MySQL/MariaDB database</td>
</tr>
<tr>
<td>PAYD insurer client</td>
<td>The insurer-facing client application where risks to be assessed for a specific driver are selected and an adjusted risk score is provided</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>
</tr>
<tr>
<td>PAYD API gateway</td>
<td>The main application-facing [API gateway that arbitrates access to multiple API/Edge service endpoints.</td>
</tr>
<tr>
<td>Consent manager</td>
<td>An end-user facing application that models consent requests, obtains fine-grained consent from the driver, and informs the creation of sticky policies</td>
</tr>
</tbody>
</table>

While the software architecture outlines the PAYD runtime environment, one aspect that has been omitted is the method in which driving data is generated and inserted into the data stores. For the purpose of the demonstration, driving data from a streaming source is pre-loaded into the Encrypted data store directly - as the RestAssured technologies mature, this will be adjusted to include periodic batch updating. In preparation for the validation, the pre-loading of the Encrypted data store is achieved by the following process:

1. The *openxc-vehicle-simulator* generates a stream of synthetic driving data in real-time

2. *openxc2mysql* intercepts this data stream and provides a local unencrypted copy of the data for interrogation by the Driver (e.g. on a trusted data store, such as the Driver’s smartphone, through the PAYD driver app)

3. *openxc2event* receives the data stream, converts it into a RestAssured event record, and submits it to the encrypted data store (e.g. in an untrusted environment, such as public cloud)

Once this is done, all access to the driving data is carried out by the respective clients through the PAYD API gateway, which communicates directly with the Query Gateway, in line with the defined high-level software architecture.
The PAYD software architecture is elaborated in Figure 3.2 below:

**Figure 3.2: PAYD Software Architecture**

**Use Case Scenarios**

As of June 2018, PAYD supports the following scenarios:

**Insurer Query Scenario**  In this scenario, the insurer uses the PAYD insurer client application to calculate an adjusted risk score for a specific driver based on a number of specific risk-weighted queries that each involve the analysis of different sources of sensitive data.

While the insurer is able to select which queries are important to them to carry out, the extent to which information is provided, and in what granularity, is controlled by the Data Protection Contract, itself modelled from the consent and privacy preferences provided by the driver.

**A Note on the Business Impact of Withholding Consent under the GDPR**

As the data subject may opt-out of disclosing data that the insurer considers integral for the calculation of a specific risk and query (e.g. the time of day that a driver typically drives), a high risk may be automatically attached for a given risk if the information is withheld, ultimately preventing the query from being satisfied. This functionality is considered an integral part of the exchange, as it permits a balance to be found between the data subject exercising their rights over their data, and the insurer in still being able to assess risk in the case of selective disclosure or consent being withheld.
Currently a number of sample queries are carried out by the PAYD insurer client, these include:

**Listing 3.1: Determine if a driver has been driving at night throughout their journey**

```sql
SELECT Person.PersonID, Person.FullName, Person.EmailAddress, e.timestamp from Events as e inner join Auto on e.AutoID = Auto.AutoID inner join Person on Auto.OwnerID = Person.PersonID GROUP BY Person.PersonID, Person.FullName, Person.EmailAddress, e.timestamp HAVING (HOUR(e.timestamp) > 20) AND (Person.PersonID = XX);
```

**Listing 3.2: Determine if a driver driving at night has exceeded a specified speed**

```sql
SELECT Person.PersonID, Person.FullName, Person.EmailAddress, e.timestamp, MAX(e.vehicle_speed) from Events as e inner join Auto on e.AutoID = Auto.AutoID inner join Person on Auto.OwnerID = Person.PersonID, Person.FullName, Person. EmailAddress, e.timestamp HAVING (HOUR(e.timestamp) > 20) AND (MAX(e.vehicle_speed > 100.00)) AND (Person.PersonID = XX);
```

**Listing 3.3: Determine if a driver falls within a high-risk age range and exceeds a specified speed**

```sql
SELECT Person.PersonID, Person.FullName, Person.EmailAddress, Person.DateOfBirth, MAX(e.vehicle_speed) FROM Events AS e INNER JOIN Auto ON e.AutoID = Auto.AutoID INNER JOIN Person ON Auto.OwnerID = Person.PersonID GROUP BY Person.PersonID, Person.FullName, Person. EmailAddress, Person.DateOfBirth HAVING (MAX(e.vehicle_speed) > 100.00) AND (YEAR(CURRENT_DATE) − YEAR(DateOfBirth)>= 16) AND (YEAR(CURRENT_DATE) − YEAR(DateOfBirth) <= 24) AND (Person.PersonID = XX);
```

In addition to determining if such a scenario has occurred, each of the results can be further counted and filtered, allowing for further refinement of the risk calculation over time.

**Driver Query Scenario** In this scenario, the driver, wishing to assess their own driving performance over time, or to determine when maintenance on the vehicle may be needed, is able to run their own queries directly. In addition to providing the driver with information that can benefit them directly, the driver is able to experiment with different types of analyses across the full range of data, increasing their awareness over their own data (as well as gaining situational awareness), and directly influencing the kinds of information they would be willing to share with the insurer over time.

**Validation Results**

Initial testing with the current prototype implementation demonstrates that it is already possible for individuals to provide fine-grained and purpose-limited data access control over their driving data, and that the insurer is able to assess and adjust a risk score regardless of the individual’s privacy preferences and consent parameters.
These aspects are further demonstrated through the following simple validation test:

1. Insurers can only see a subset of the Driver’s information, based on the permission in the Data Protection Contract.
2. The Insurer opens the insurer client application and enters the parameters for the Driver they wish to assess.
3. The Driver has provided consent for a subset of data accesses and data collection purposes, each of which are reflected in the Data Protection Contract.
4. We show how the Insurer can see a given subset of driver data relative to the consent preferences of the Driver, but not other data. While the Insurer is provided with an adjustment based on the risk assessed, none of the underlying information is disclosed in the assessment.
5. The Insurer wishes to assess a number of other Drivers, each with their own unique privacy preferences and consent parameters. The process is repeated for each Driver assessed, with the risk assessment changing accordingly.

Next Steps

In Phase 1 of the project, owing to limitations in Opaque, geolocational and instrumentation data were excluded from the validation. The first version of the PAYD validation has focused primarily on basic UBI queries that can influence the overall risk assessment of the driver, while leaving the more complex analysis of dynamic geolocational and instrumentation data (more typical of the more advanced PHYD model) to the second phase of the project. Furthermore, the following specific technical challenges will also be pursued:

- Support for fine-grained risk assessment through e.g. selective disclosure/dynamic masking of rows relative to individual consent preferences policy settings.
- Support for changing drivers of a vehicle, each with different privacy preferences, within the same journey (this carries with it a high risk of identification through inference, owing to vehicle location data and timestamps, which will be the focus of future work).
- Support for more complex queries through the Query Gateway, as well as the querying of geolocational and instrumentation data in Opaque, allowing more complex PHYD models to be implemented.
- Changing and withdrawing consent over time (e.g. as a result of the driver’s own analysis of their driving performance).
- Standards-based consent receipts to facilitate interoperability with a growing ecosystem of consent-aware tools and services.
- End-to-end authentication and security between RestAssured system components, including across a microservices service mesh (e.g. with istio-auth and SPIFFE).

---

3. https://istio.io
4. https://spiffe.io
Validation of the Self-directed Social Care (CARE) Use Case

Use Case Description

Ami, developed and operated by Oxford Computer Consultants, is an online service in the United Kingdom that connects (i) lonely people who need help and (ii) volunteers offering help. Matching volunteers to people needing care is based on information such as the place where a person lives and their needs. These pieces of information are displayed only in obfuscated form, so as to preserve the users’ privacy. The information about people with loneliness and related needs is valuable to local authorities, who are responsible for supplying social care to persons in need within their areas.

Social Care Analysis of Needs Tool (SCANT) is a tool to assist the local authorities in identifying unmet needs, whilst also preserving the privacy of the potentially vulnerable Ami users. For instance, local authorities can query with SCANT the number of Ami users with particular needs in a broad geographical region, however, individual Ami users who did not consent to the disclosure of their data will remain anonymous to the local authorities. The registration of data subjects takes place with a special user interface (Ami/SCANT registration tool), which forwards the information about user consent to the data gatekeeper. By using this proposed architecture, data protection policies of the Ami users can be captured and enforced throughout the data life-cycle. The stored sensitive data are protected against unauthorized access. Queries of local authorities are modified automatically on the fly so that the data of Ami users who did not consent to the analytical use of their data are excluded from the results. This guarantees that local authorities never get access to data of Ami users who did not consent to this. Local authorities can still work with data of Ami users who did consent to the disclosure of their data as well as with aggregated data of Ami users who consented to this. This automatic fine-grained access control is a major advantage of the architecture. Furthermore, through continuous monitoring, risk assessment, and adaptation, changes to the underlying infrastructure can be handled transparently.

SCANT is used by two main classes of user:

Local Authority User works for regional government and have responsibility for providing adequate social care to certain classes of vulnerable citizens. They may need to obtain statistical data about the distribution of different care needs within their area of responsibility in order to plan distribution of services.

Service Provider User works for a private company or charitable organisation, which provides social care services to their clients, often on behalf of the local authorities. They can benefit from access to statistical data about the distribution of different care needs in order to plan marketing activities.

The software architecture of SCANT is based on the assumption that the SCANT tool itself cannot be entirely trusted. It may be hosted on a location, such as a public cloud, that provides opportunities for hostile actors to interfere. There is also the possibility of an “inside threat” amongst the staff with access to the SCANT tool at the Local Authority or Service Provider. Instead the SCANT tool delegates access to the sensitive data to the Query Gateway components. As can be seen in Figure 3.4, the SCANT tool sends its query to the Query Gateway, which uses the Data Gatekeeper to verify the data subject’s consent to the usage requested.

The SCANT tool itself is designed as an easily deployable web app, which could be rolled out on user premises. It is a python web app, hosted on a Nginx server process in a docker container. These technology choices were made to provide the following benefits.

Python The python language provides great flexibility while we are developing the SCANT tool. The current iteration of the tool is a prototype to demonstrate the feasibility of the RestAssured approach, rather than a finished product. The ability to make revisions quickly is a more valuable feature than the maintainability and guarantees of formal correctness of a more structured language. The availability of excellent open source Python libraries is also a valuable asset.
Flask  Flask is a lightweight and relatively unopinionated framework for building web apps. It has proved ideal for our use case and has allowed a rapidly evolving prototype to mature as the project has progressed.

Nginx  Flask comes with an in-built development web server, but this is not adequate for a production system. Nginx is a performant and robust web server that handles the web request servicing and thread management for our app.

Docker  Docker provides isolation of our system from dependency on the details of the host platform. Our app is deployed as a pre-built Docker image. This allows us to assemble our system dependencies independent of the host operating system.

The SCANT code is responsible for parsing the user’s inputs, and using this to assemble a query relating to the need of interest and the region under study.

Use Case Scenarios

As of June 2018, SCANT supports the following three scenarios:

1. Local Authority (LA) scenario
2. Provider Scenario
3. Data Subject Preference Update

Local Authority scenario  In this scenario the Local Authority User uses the SCANT tool to identify 1) a need of interest (e.g. transport, companionship etc.) and 2) a region of interest (here defined as a UK postcode region or outcode e.g. OX1 - the postcode region for central Oxford.)

The user input thus supplied is then parsed and used to create a SQL query of the form:

```
SELECT PersonId FROM escant WHERE ((Outcode='OX1') AND (NeedId=2))
```
This query provides a list of relevant person ids that: i) Match the query ii) Have given consent for the appropriate usage. It is important that although the interference with the SCANT tool could modify the query sent, this will not allow access to sensitive data as the primary security is enforced by the data contract. This contract allows access to the following fields:

1. **PersonId** - Unique identifier for a person.
2. **Outcode** - Postcode region in which a person dwells.
3. **NeedId** - Unique identifier for a need (e.g. Transport etc.)
4. **IsTopNeed** - *Optional* is this the client’s most pressing need?

Thus, no matter what queries are sent using a corrupted SCANT tool, the Query Gateway still restricts access to the above non-sensitive data, preserving client confidentiality and privacy.

**Provider Scenario**  From a technical perspective, the Service Provider User performs exactly the same operations as the LA user above, the only difference is that the data subject has to provide consent for them to do so. Data subjects may supply consent for LA and Provider users independently. These are considered different *Usages* and consent is tracked independently by the Data Gatekeeper.

**Data Subject Preference Update**  In order for consent to be meaningful, it must be possible for the Data Subject to grant, withhold or withdraw it. For the purposes of the prototype SCANT, a proof-of-principle additional user interface has been included to allow Data Subjects to register with SCANT, or to modify their consent to the 2 main usages supported (LA and Provider). The SCANT user interface integrates with
the Data Gateway (the upload component of the Query Gateway) and the Data Gatekeeper to allow a data subject to decide which, if either, of the two usages they will permit. In a production system, this usage would be fulfilled by the Ami system. However, for simplicity and security, Ami integration will not be supported until the basic RestAssured system has been validated and found secure.

Validation Results

Initial tests with the current prototype SCANT have produced encouraging results. It is already possible for us to demonstrate the key functionality of the system - that changes to the sticky policies associated with the data subjects consent lead to changes in the results of queries that reflect the appearance or disappearance of clients from the pool of consenting data subjects. The simple validation tests we can demonstrate involves:

Different usages

1. Select a known data subject ahead of time, and noting their needs and outcode, ensure they have given consent.
2. Log in as LA user and perform a query for the outcode and needs of the noted subject, demonstrate the count of results.
3. Log in as Provider User and perform a query for the outcode and needs of the noted subject, demonstrate the count of results.
4. Log in as the data subject, and navigate to the sticky policy management screen on the Data Gatekeeper.
5. Change the usage permissions to allow LA use, but not provider use.
6. Log in as LA user and perform a query for the outcode and needs of the noted subject, demonstrate that the count of results does not change.
7. Log in as Provider user and perform a query for the outcode and needs of the noted subject, and demonstrating the count of results has decreased by one.

No consent

1. Select a known outcode ahead of time, for which there is only 1 data subject.
2. Log in as the data subject, and navigate to the sticky policy management screen on the Data Gatekeeper.
3. Change the usage permissions to disallow all uses.
4. Log in as LA user and perform a query for the outcode and needs of the noted subject, demonstrate that the count of results is zero.
5. Log in as Provider User and perform a query for the outcode and needs of the noted subject, demonstrate that the count of results is zero.
6. Log in as the data subject and navigate to the sticky policy management screen on the Data Gatekeeper.
7. Change the usage permissions to allow LA use, but not provider use.
8. Log in as LA user and perform a query for the outcode and needs of the noted subject, demonstrate that the count of results has increased by one.
9. Log in as Provider user and perform a query for the outcode and needs of the noted subject, demonstrate that the count of results does not change.

New Subject

1. Select a known outcode ahead of time, for which there is only 1 data subject.
2. Log in as the data subject and navigate to the sticky policy management screen on the Data Gatekeeper.
3. Change the usage permissions to disallow all uses.
4. Log in as LA user and perform a query for the outcode and needs of the noted subject, demonstrate that the count of results is zero.
5. Log in as Provider User and perform a query for the outcode and needs of the noted subject, demonstrate that the count of results is zero.
6. Log in as a new data subject, recording their postcode as matching the noted outcode.
7. Change the usage permissions to allow LA use, but not provider use.
8. Log in as LA user and perform a query for the outcode and needs of the noted subject, demonstrate that the count of results has increased by one.
9. Log in as Provider user and perform a query for the outcode and needs of the noted subject, demonstrate that the count of results does not change.

These simple scenarios demonstrate that the RestAssured system is capable of supporting basic queries, whilst maintaining data subjects’ privacy

Next Steps

The next steps we will work on are:

- Authentication between RestAssured system components.
- Additional query complexity in the Query Gateway
- Support for a more complex scenario involving multiple data subject types - for example matching volunteer helpers with clients in need.

Validation of the High Performance Computing (HPC) Use Case

HPC Use Case Rescheduling for Phase 2

Owing to both the current limitations of the SGX technology as well as the more commercially-oriented data protection concerns of the HPC use case and the timing of the GDPR coming into force during the first phase of the project, the use cases that place an emphasis on personal data concerns have been prioritized for the first phase validation. The validation of the HPC use case will, therefore, be deferred until the second phase of the project. For additional information on this decision, refer to the explanatory notes on page 10.

Evaluation Summary

At the current state of the art, RestAssured technologies are demonstrably able to support the fundamental needs of both the CARE and PAYD use cases. While each of the use cases represent similar challenges for the underlying RestAssured technologies, each has uniquely utilized the componentry in different ways to address the particular challenges presented, demonstrating the flexibility of the RestAssured technologies and methodology. Each use case has further elaborated concrete expectations for the continued evolution of RestAssured, and has identified real requirements for both the architecture and the individual technical components for the second phase of the project that will enable the use cases to be further enriched and, ultimately, brought closer to market.
4 Feedback Cycle and Next Steps

Use Case Contributions to the RestAssured Architecture

Insights from the use cases were a major driver for both the initial development and the continuous improvement of the RestAssured architecture. The data flow view of the initial RestAssured architecture, described in deliverable D3.1[6], Section 3, was based on an abstraction of the PAYD and CARE use cases. This abstraction was later used in other work packages (e.g., WP5) as well to validate the proposed approaches.

Later on, the architecture was consolidated to the view described in D3.2[4], Section 2.3. For this, it was vital to identify the context entities (data subject, data controller, application, data store) with which the RestAssured run-time system has to interact, as well as the specific requirements regarding the interaction with these context entities. This process, described in D3.2, Section 2.2, was also largely based on the insights from the use cases. Finally, the proposed architecture was validated by instantiating it in the CARE use case (D3.2, Section 2.4).

Use Case Requirements for the Next Phase

Throughout the course of the use case validation activities, a number of limitations were encountered and elaborated for follow-up during the next phase of the project. While each use case has its own specific set of feature enhancements that will be pursued as the technologies mature, a number of points of commonality with specific requirements for the technical work packages were identified. These are briefly summarized for convenience in Table 4.1 below:

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Responsible WP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additional query complexity in the Query Gateway</td>
<td>WP4, WP6</td>
</tr>
<tr>
<td>Dynamic changes to and withdrawal of consent by data subjects</td>
<td>WP5–WP7</td>
</tr>
<tr>
<td>End-to-end authentication between RestAssured system components</td>
<td>WP3, WP4–WP7</td>
</tr>
<tr>
<td>Support for multiple and different types of data subjects</td>
<td>WP3, WP4–WP7</td>
</tr>
</tbody>
</table>

Specific requirements for the technical work packages raised by the use cases individually are further elaborated in Table 4.2 below:

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Requesting Use Case</th>
<th>Responsible WP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Protection Contract interoperability with consent receipt standard</td>
<td>PAYD</td>
<td>WP4, WP6</td>
</tr>
<tr>
<td>Dynamic row masking and filtering based on data subject preferences</td>
<td>PAYD</td>
<td>WP4, WP6</td>
</tr>
<tr>
<td>Support for geospatial, date and time data types</td>
<td>PAYD</td>
<td>WP4, WP6</td>
</tr>
</tbody>
</table>
5 Conclusion

This document identified the first set of validation goals and criteria relevant for measuring the success and relevance of the overall architecture and the individual software components in addressing the needs of the use cases.

Given the iterative nature of the project development phases (as outlined in section 1.1.2, not all components are able to be assessed to the full extent of their capabilities at this phase. Gaps that have been identified and requirements that have been refined based on the initial validation have been contributed back for the next phase of the project. A more comprehensive validation of the components and acceptance testing of V2 of the overall architecture will be carried out in the final validation plan (D8.3) and reported on in the next version of this document at the end of the project.
References


