CyberKit4SME
Democratizing a Cyber Security Toolkit for SMEs and MEs
Project Nº 883188

D2.2
Technical specifications

Responsible: aurelien.lecamus@inetum.world
Document Reference: D2.2
Dissemination Level: Public
Version: 1.1
Date: 15/09/2021
Executive Summary

The main goal of this deliverable is to identify a technical architecture and specifications for the toolkit to work as a whole and to be integrated within the information system of SMEs. This in turn allows to identify how we can architecture and build the test bed in order to work on the tools development and to enable validation scenarios to be carried out. This also enables technical partners to define what they will need to develop or enhance, in terms of interfaces, in the tasks associated with WP4 and WP5. In order to complete their vision regarding these tasks, the main SMEs expectations regarding the general usage of the toolkit have been assessed and listed within the document, allowing technical partners to identify what features they can develop or enhance to meet these expectations. In parallel, the main features technical partners expect to work on during the project have also been listed. The validation scenarios carried out in the scope of WP3 will allow to confront these expectations with reality and to refine the functional specifications for the toolkit.
### Contributors Table

<table>
<thead>
<tr>
<th>Version</th>
<th>DOCUMENT SECTION</th>
<th>AUTHOR(S)</th>
<th>REVIEWER(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>ALL (save sections listed below)</td>
<td>Aurélien LECAMUS (Inetum) Leïla RABHI (Inetum)</td>
<td>Maya ANDERSON (IBM) Aurélien LECAMUS (Inetum)</td>
</tr>
<tr>
<td>1.0</td>
<td>IV.1.1, IV.6.1, V.3.1 (Keenaï)</td>
<td>Mathieu BERTRAND (Inetum)</td>
<td>Maya ANDERSON (IBM) Aurélien LECAMUS (Inetum)</td>
</tr>
<tr>
<td>1.0</td>
<td>IV.1.2, IV.6.2, V.3.2 (SSM)</td>
<td>Stephen PHILLIPS (IT Innov)</td>
<td>Maya ANDERSON (IBM) Aurélien LECAMUS (Inetum)</td>
</tr>
<tr>
<td>1.0</td>
<td>IV.1.3, IV.6.3, V.3.3 (SDS)</td>
<td>Maya ANDERSON (IBM)</td>
<td>Maya ANDERSON (IBM) Aurélien LECAMUS (Inetum)</td>
</tr>
<tr>
<td>1.0</td>
<td>IV.1.4, IV.6.4, V.3.4 (SL)</td>
<td>Stefano DE ANGELIS (UoS Cyber)</td>
<td>Maya ANDERSON (IBM) Aurélien LECAMUS (Inetum)</td>
</tr>
<tr>
<td>1.0</td>
<td>IV.1.5, IV.6.5, V.3.5 (CJML)</td>
<td>Ragnhild HALVORSRUD (SINTEF)</td>
<td>Maya ANDERSON (IBM) Aurélien LECAMUS (Inetum)</td>
</tr>
<tr>
<td>1.0</td>
<td>IV.4.2 Testbed architecture</td>
<td>Stephen PHILLIPS (IT Innov)</td>
<td>Maya ANDERSON (IBM) Aurélien LECAMUS (Inetum)</td>
</tr>
<tr>
<td>1.0</td>
<td>IV.5.1. SSM &lt;&gt; Keenaï</td>
<td>Stephen PHILLIPS (IT Innov) Mathieu BERTRAND (Inetum)</td>
<td>Maya ANDERSON (IBM) Aurélien LECAMUS (Inetum)</td>
</tr>
<tr>
<td>1.0</td>
<td>IV.5.2. SDS &gt; Keenaï</td>
<td>Maya ANDERSON (IBM) Mathieu BERTRAND (Inetum)</td>
<td>Maya ANDERSON (IBM) Aurélien LECAMUS (Inetum)</td>
</tr>
<tr>
<td>1.0</td>
<td>IV.5.3. Keenaï &lt;&gt; Service Ledger</td>
<td>Mathieu BERTRAND (Inetum) Stefano DE ANGELIS (UoS Cyber)</td>
<td>Maya ANDERSON (IBM) Aurélien LECAMUS (Inetum)</td>
</tr>
<tr>
<td>1.0</td>
<td>IV.5.4. SSM &lt;&gt; Service Ledger</td>
<td>Stefano DE ANGELIS (UoS Cyber) Stephen PHILLIPS (IT Innov)</td>
<td>Maya ANDERSON (IBM) Aurélien LECAMUS (Inetum)</td>
</tr>
<tr>
<td>1.0</td>
<td>IV.5.5. CJML &lt;&gt; SSM</td>
<td>Stephen PHILLIPS (IT Innov) Ragnhild HALVORSRUD (SINTEF)</td>
<td>Maya ANDERSON (IBM) Aurélien LECAMUS (Inetum)</td>
</tr>
<tr>
<td>1.1</td>
<td>IV.1.4 SL</td>
<td>Stefano DE ANGELIS (UoS Cyber)</td>
<td>Aurélien LECAMUS (Inetum)</td>
</tr>
<tr>
<td>1.1</td>
<td>IV.3.3. Security within the toolkit</td>
<td>Stephen PHILLIPS (IT Innov) Stefano DE ANGELIS (UoS Cyber) Mathieu BERTRAND (Inetum)</td>
<td>Stephen PHILLIPS (IT Innov) Aurélien LECAMUS (Inetum)</td>
</tr>
<tr>
<td>1.1</td>
<td>IV.3.4. Authentication and Authorization</td>
<td>Stephen PHILLIPS (IT Innov) Aurélien LECAMUS (Inetum)</td>
<td>Aurélien LECAMUS (Inetum)</td>
</tr>
<tr>
<td>1.1</td>
<td>IV.4.2 Testbed architecture</td>
<td>Stephen PHILLIPS (IT Innov) Aurélien LECAMUS (Inetum)</td>
<td>Aurélien LECAMUS (Inetum)</td>
</tr>
</tbody>
</table>
IV. ARCHITECTURE .......................................................... 14
IV.1. Architecture of each tool ............................................. 14
  IV.1.1. Keenaï .............................................................. 14
  IV.1.2. SSM ............................................................... 15
  IV.1.3. SDS ............................................................... 17
  IV.1.4. SL ................................................................. 19
  IV.1.5. CJML ............................................................. 20
IV.2. Functional architecture ............................................. 21
IV.3. General architecture ................................................. 22
  IV.3.1. Conceptual architecture ...................................... 22
  IV.3.2. System requirements overview .............................. 23
  IV.3.3. Security within the toolkit .................................. 24
  IV.3.4. Authentication and Authorization .......................... 25
IV.4. Deployment architecture ......................................... 26
  IV.4.1. General deployment architecture ........................... 26
  IV.4.2. Testbed architecture ......................................... 27
IV.5. Interaction between tools .......................................... 29
  IV.5.1. SSM <> Keenaï ................................................ 29
    IV.5.1.a. Data Models ............................................. 29
    IV.5.1.b. Building and Using a System Model ................... 31
    IV.5.1.c. Identifying Divergent Operational Events .......... 31
    IV.5.1.d. Operational Risk Assessment .......................... 32
      IV.5.1.d.1. Software Vulnerability Data from OpenVAS .... 32
ANNEX 1 – PRESENTATION OF THE TOOLKIT COMPONENTS 

Secure Data Services ............................................. 45
Keenaï ........................................................................ 46
System Security Modeller .......................................... 47
CJML ........................................................................ 48
Service Ledger ......................................................... 49
Table of Tables

Table 1. List of requirements ...........................................................................................................11
Table 2. System requirements ..........................................................................................................23

Table of Figures

Figure 1. Keenaï general architecture ..............................................................................................14
Figure 2. General data processing workflow ....................................................................................15
Figure 3. Component architecture of the System Security Modeller ..............................................16
Figure 4. Deployment diagram for the System Security Modeller ..................................................16
Figure 5. SDS High-level architecture .............................................................................................17
Figure 6. SDS with HTTP API ..........................................................................................................18
Figure 7. SDS as a library ................................................................................................................18
Figure 8. High level overview of the SL CTI sharing platform architecture ..................................19
Figure 9 Detailed architecture of a SL instance ..............................................................................20
Figure 10. Component architecture – tools for human and organisational risk models ...............21
Figure 11. Project functional architecture diagram .........................................................................21
Figure 12. Technical conceptual architecture ..................................................................................22
Figure 13. Deployment architecture ...............................................................................................26
Figure 14. High level testbed architecture ......................................................................................27
Figure 15. Targeted CJML application architecture ........................................................................44
Figure 16. Presentation of the SDS ................................................................................................46
Figure 17. Sources and global functionalities ..................................................................................46
Figure 18. CJML targets a wide user group ......................................................................................48
Figure 19. Customer journey and touchpoints ................................................................................49
Figure 20. Presentation of the Service Ledger ..................................................................................50
# Table of Acronyms and Definitions

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>CERT</td>
<td>Computer Emergency Response Team</td>
</tr>
<tr>
<td>CJML</td>
<td>Customer Journey Modelling Language</td>
</tr>
<tr>
<td>CoI</td>
<td>Community of Interest</td>
</tr>
<tr>
<td>CSIRT</td>
<td>Computer Security Incident Response Team</td>
</tr>
<tr>
<td>CTI</td>
<td>Cyber Threat Intelligence</td>
</tr>
<tr>
<td>EPS</td>
<td>Event Per Second</td>
</tr>
<tr>
<td>GDPR</td>
<td>General Data Protection Regulation</td>
</tr>
<tr>
<td>IoC</td>
<td>Indicator of Compromise</td>
</tr>
<tr>
<td>ISO</td>
<td>International Organization for Standardization</td>
</tr>
<tr>
<td>ME</td>
<td>Micro Enterprise</td>
</tr>
<tr>
<td>NIS</td>
<td>Network and Information System</td>
</tr>
<tr>
<td>SaaS</td>
<td>Software as a Service</td>
</tr>
<tr>
<td>SDS</td>
<td>Secure Data Services</td>
</tr>
<tr>
<td>SIEM</td>
<td>Security Information and Event Management</td>
</tr>
<tr>
<td>SL</td>
<td>Service Ledger</td>
</tr>
<tr>
<td>SME</td>
<td>Small and Medium sized Enterprise</td>
</tr>
<tr>
<td>SSM</td>
<td>System Security Modeller</td>
</tr>
<tr>
<td>STIX</td>
<td>Structured Threat Information Expression</td>
</tr>
<tr>
<td>TAXII</td>
<td>Trusted Automated Exchange of Intelligence Information</td>
</tr>
</tbody>
</table>
I. INTRODUCTION

I.1. Purpose and organisation of the document

This document aims to specify the architecture of the toolkit, the interactions between its components and the external interfaces of the toolkit for SMEs to interact with it. These specifications will serve as a basis for installing the toolkit components in the testbeds and launch the validation scenarios testing. They will be enhanced over time as more feedbacks are gathered from said scenarios.

In addition, it has been decided to include in this document a general analysis of the functional specifications for the toolkit and the main features that are envisioned to be developed. It allows giving more visibility on the toolkit component roadmaps to the various stakeholders. It also helps identify whether there were any oversights in the technical specifications.

I.2. Scope and audience

This document is applicable to the CyberKit4SME project until the end of the project.

This is a public document.
II. CONTEXT AND OBJECTIVES

II.1. Overview of the project

CyberKit4SME aims to democratize a kit of cyber security tools and methods enabling SMEs/MEs to:

- Increase awareness of cybersecurity risks, vulnerabilities and attacks; Monitor and forecast risks;
- Manage risks using organisational, human and technical security measures with greater confidence; and
- Collaborate and share information in a collective security and data protection effort.

Tools developed in the project are:

- Semi-automated ISO 27005 threat identification and risk mitigation analysis, using a knowledge base of technical and human/organisational risk factors;
- Encryption and isolation tools to protect data being stored, processed or exchanged;
- Security information and event management, using multiple data sources for threat detection and diagnosis,
- Blockchain tools for SMEs/MEs to share intelligence and incident reports with supply chain partners and with CERTs.

II.2. Objectives of the project

CyberKit4SME will make its tools cheaper and more usable by SME/ME, by

1) Exploiting synergies between tools in the kit to simplify the use of each;
2) Sharing information to increase the data available for threat detection and diagnosis at each SME/ME;
3) Embedding intelligence (e.g. machine reasoning and data analytics), to fill gaps in inputs and automate tasks such as risk analysis and security configuration.

The project will also use its tools and cyber range demos to train SMEs/MEs to identify their top threats and recognise and address them with greater confidence. Results will be validated by SME/ME in four critical sectors: Finance, Health Care, Energy and Transport.

Outcomes include reducing the time/cost of cyber security awareness and protection, simplifying meeting and demonstrating compliance with NIS Directive and GDPR, protecting distributed assets from cloud services to edge devices, and engaging in secure supply chains with larger organisations.

The project will also collaborate with related research projects and disseminate widely in the scientific community and in SME networks.

II.3. Regulatory constraints

CyberKit4SME aims to help SMEs and MEs demonstrate compliance with any regulation that imposes requirements for cyber security or data protection. In each sector there are regulations at European level (e.g. eIDAS in Health Care) or national level (e.g. BAIT requirements in the German Finance sector). Here we focus on EU regulations that apply to all the validation sectors: the NIS Directive and the GDPR.
### III. REQUIREMENTS

The requirements identified during task 2.1 have been taken into account throughout this document.

#### III.1. Synthesis of requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GR_01</td>
<td>General</td>
<td>Be able to generate a report over the past N weeks to show their clients the types and severity of threats that were seen and which we might help them with</td>
</tr>
<tr>
<td>GR_02</td>
<td>General</td>
<td>Be able to query the system for threats over a given period</td>
</tr>
<tr>
<td>GR_03</td>
<td>General</td>
<td>Have a tool to be able to monitor for and identify potential threats and mitigation strategies in the existing system</td>
</tr>
<tr>
<td>ACR_01</td>
<td>Access Control</td>
<td>Have a utility to be able to manage new users when they come to the system (passwords, role-based access, access to different types of data etc.)</td>
</tr>
<tr>
<td>ACR_02</td>
<td>Access Control</td>
<td>Have a utility to be able to remove users when they leave (revoke access / passwords, clear up user-generated reports, etc.)</td>
</tr>
<tr>
<td>AR_01</td>
<td>Auditing</td>
<td>Have a utility to be able to query the entire system to identify potential threats and mitigation on demand</td>
</tr>
<tr>
<td>AR_02</td>
<td>Auditing</td>
<td>Have a utility to be able query the entire system to identify potential threats and mitigation running permanently as a background process and raising alerts if issues have been identified</td>
</tr>
<tr>
<td>AR_03</td>
<td>Auditing</td>
<td>Have a standards-compliant audit tool to query access requests and denial</td>
</tr>
<tr>
<td>AR_04</td>
<td>Auditing</td>
<td>Have a penetration testing process to be run on demand</td>
</tr>
<tr>
<td>IR_01</td>
<td>Interoperability</td>
<td>Be able to integrate services and / or data from different systems in accordance with relevant standards</td>
</tr>
<tr>
<td>PSR_01</td>
<td>Physical Security</td>
<td>Have a utility to be able to manage physical access to sensitive areas (such as server rooms)</td>
</tr>
<tr>
<td>PSR_02</td>
<td>Physical Security</td>
<td>Have the means to secure data in transit between servers and between servers and 3rd party devices (i.e., during remote installation)</td>
</tr>
<tr>
<td>PSR_03</td>
<td>Physical Security</td>
<td>Have a utility to identify and secure different virtual areas of a server (i.e., secure data, algorithms, etc. to different levels according to their sensitivity)</td>
</tr>
<tr>
<td>PR_01</td>
<td>Provenance</td>
<td>Have the means to track where data have been sourced and what actions have been performed on the data since (i.e., generate a complete provenance record)</td>
</tr>
<tr>
<td>PR_02</td>
<td>Provenance</td>
<td>Have the means to guarantee delivery of data / code without interference to its intended destination</td>
</tr>
<tr>
<td>PR_03</td>
<td>Provenance</td>
<td>Have the means to track a user’s activities during access sessions</td>
</tr>
<tr>
<td>SDR_01</td>
<td>Software Development</td>
<td>Have the means to protect the development environment (including code, training / test data, etc.)</td>
</tr>
<tr>
<td>SDR_02</td>
<td>Software Development</td>
<td>Have the means to protect the intellectual property of their developmental environment (e.g., architecture, technical documentation)</td>
</tr>
<tr>
<td>SDR_03</td>
<td>Software Development</td>
<td>Have the means to check software for internal security exposures and robustness before release</td>
</tr>
<tr>
<td>ID</td>
<td>Category</td>
<td>Description</td>
</tr>
<tr>
<td>-----</td>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SER_01</td>
<td>Software</td>
<td>Have the means to secure module-to-module transfers and communication in the field (e.g., where modules in a distributed environment exchange information / data and / or communicate back to our own development environment)</td>
</tr>
<tr>
<td>SER_02</td>
<td>Software</td>
<td>Have the means to protect software (and data) during execution</td>
</tr>
<tr>
<td>NFR_01</td>
<td>Non Functional</td>
<td>Have the means to identify potential threats from human behaviours</td>
</tr>
<tr>
<td>NFR_02</td>
<td>Non Functional</td>
<td>Have the means to be able to differentiate threats to their customers and those which affect them internally</td>
</tr>
<tr>
<td>NFR_03</td>
<td>Non Functional</td>
<td>Receive training on cybersecurity for their own engineers / employees</td>
</tr>
<tr>
<td>NFR_04</td>
<td>Non Functional</td>
<td>Receive training on cybersecurity for their customers to be offered as a service</td>
</tr>
<tr>
<td>NFR_05</td>
<td>Non Functional</td>
<td>Provide customers with support for how to identify and protect against threats they may encounter (e.g., to recover when problems arise)</td>
</tr>
<tr>
<td>NFR_06</td>
<td>Non Functional</td>
<td>Provide training and support for their own employees to encourage security-aware behaviours</td>
</tr>
<tr>
<td>NFR_07</td>
<td>Non Functional</td>
<td>Have the means to validate compliance with appropriate standards and regulations (ISO 27001, GDPR etc)</td>
</tr>
<tr>
<td>NFR_08</td>
<td>Non Functional</td>
<td>Have a checklist (both manual and automated) for a given process, e.g., onboarding a customer; removing a customer etc.</td>
</tr>
<tr>
<td>NFR_09</td>
<td>Non Functional</td>
<td>Have the means to check the integrity of a deliverable (e.g., that software components are the same as those sent from our system to the customer when they are deployed at the customer location)</td>
</tr>
<tr>
<td>NFR_10</td>
<td>Non Functional</td>
<td>Communication between modules being deployed on the net (like edge devices) shall be secured</td>
</tr>
<tr>
<td>NFR_11</td>
<td>Non Functional</td>
<td>Data integrity between modules being deployed on the net (like edge devices) shall be ensured</td>
</tr>
<tr>
<td>SR_01</td>
<td>Security</td>
<td>Communication between modules shall be secured</td>
</tr>
<tr>
<td>SR_02</td>
<td>Security</td>
<td>Communication between modules shall involve a mutual authentication</td>
</tr>
<tr>
<td>SR_03</td>
<td>Security</td>
<td>Data confidentiality shall be ensured</td>
</tr>
<tr>
<td>SR_04</td>
<td>Security</td>
<td>Data integrity shall be ensured</td>
</tr>
<tr>
<td>SR_05</td>
<td>Security</td>
<td>Password management shall be ensured</td>
</tr>
<tr>
<td>SR_06</td>
<td>Security</td>
<td>User activity shall be tracked and logged</td>
</tr>
<tr>
<td>SR_07</td>
<td>Security</td>
<td>The communication between sources and collectors (into the SIEM) shall be encrypted</td>
</tr>
<tr>
<td>SR_08</td>
<td>Security</td>
<td>High availability shall be ensured for the vital modules of the toolkit (like the SIEM)</td>
</tr>
<tr>
<td>SR_09</td>
<td>Security</td>
<td>A multilevel security access shall be supported</td>
</tr>
</tbody>
</table>

Table 1. List of requirements
III.2. General usage

Requirements encompassing the general usage of the toolkit are too specific to be detailed here, but they have been taken into account through the listing of the main general expectations from users of the toolkit in section V.2.3. General usage.

The requirements in question are:

- GR_01
- GR_02
- GR_03
- AR_01
- AR_02
- AR_03
- PSR_03
- PR_01
- PR_02
- PR_03
- SER_01
- SER_02

III.3. Integration

Requirements regarding integration have been taken into account and are covered in section IV.5. Interaction between tools.

The requirements in question are:

- IR_01
- PSR_02

III.4. Access management

Requirements regarding access management have been taken into account and are covered in sections IV.3.4. Authentication and Authorization and IV.6.6. User manager.

The requirements in question are:

- ACR_01
- ACR_02

III.5. Security

Requirements concerned with the security of the toolkit itself have been taken into account and are covered in section IV.3.3. Security within the toolkit and in section IV.3.4. Authentication and Authorization.

The requirements in question are:

- PSR_02
- NFR_10
- NFR_11
- SR_01
- SR_02
- SR_03
- SR_04
- SR_05
- SR_06
- SR_07
- SR_08
- SR_09
III.6. Out of scope

Some requirements have been identified as not directly within the scope of the toolkit, either because they are non-functional (NFR_03 to NFR_08) and cannot be addressed by the toolkit or because they deal with area of security not encompassed by the project, namely application security (SDR_01 to SDR_03), physical security (PSR_01) and penetration testing (AR_04).

The requirements in question are:

- NFR_03
- NFR_04
- NFR_05
- NFR_06
- NFR_07
- NFR_08
- AR_04
- PSR_01
- SDR_01
- SDR_02
- SDR_03
IV. ARCHITECTURE

IV.1. Architecture of each tool

For each tool:
- A presentation (taken from D2.1 – Technical Requirements) has been included in Annex 1 – Presentation of the toolkit components;
- The installation, operation and administration have been described in D2.1 – Technical Requirements in paragraph V. INSTALLATION & OPERATION.

IV.1.1. Keenaï

Keenaï is an adaptive solution and its architecture can be defined according to the environment:
- Full distributed environment: n servers (VMs/Docker containers) running in parallel
- A configuration (resources, number of nodes, …) depending on the context (expected EPS, topology, …)

The Keenaï environment is based on a set of modules providing specific processing functions and grouped into several categories:
- A set of collectors
- A Central Analysis Cluster:
  - A centralization module / archiving raw logs
  - A centralization module / recording events
  - A central correlation engine
  - An Alerting module / real time notification
- An Administration Console

![Keenaï general architecture](Figure 1. Keenaï general architecture.)
The main technologies used in Keenaï are:

- Based on a Big Data stack (Hadoop / Yarn)
- Apache Kafka: distributed data bus (logs, events, alerts, models, information)
- Elasticsearch: indexing and visualization of data in real time
- Apache Flink: distributed processing and analysis
- Hadoop Distributed File System (HDFS): log storage, metric result storage for Machine Learning, …
- Apache Spark: batch processing
- Tomcat: web administration console
- MySQL: database for console data (users, profiles, rules, filters, …)
- Logstash: log standardization

The general workflow of the processing of data (events & configuration) is described in figure below:

![General data processing workflow](image)

**Figure 2. General data processing workflow.**

### IV.1.2. SSM

The System Security Modeller is a Java application with a REST API for client applications. The primary client application is a web-based user interface which utilises ReactJS for a single-page application.

User authentication is handled by an external Keycloak component (using the OpenID Connect Authorization Code Flow). Users are identified as having a “user” role and in addition may have the “admin” role. The role(s) permit access by the SSM to different parts of the API accordingly. Authorization data is held by the SSM (not Keycloak).

Data for the SSM is persisted in two different databases:

1. a MongoDB database storing authorization data (who has what rights on what system model) and model meta-data (last edit date, etc);
2. a JenaTDB database holding the knowledgebases of threats and controls (also known as “domain models”) and the system models drawn by the users.

A high-level component architecture of the SSM can be found in Figure 3.

The SSM is deployed in containers using Docker Compose or Kubernetes. The application is split into four containers:

1. A reverse proxy (nginx) through which the client communicates. The proxy also hosts the user documentation website.
2. The SSM Java application deployed in Tomcat. The service also serves the ReactJS client to the user’s web browser. The JenaTDB database is hosted in the same container and communication with JenaTDB is done in-process.
4. MongoDB.
In addition, the prototype software to provide a risk analysis of an operational model is currently implemented (in Python) in a separate micro-service called the “SSM-adaptor”. When deployed in Kubernetes, data persistence must be provided by Kubernetes persistent volumes. A deployment diagram may be found in Figure 4.

Figure 3. Component architecture of the System Security Modeller.

Figure 4. Deployment diagram for the System Security Modeller.
IV.1.3. SDS

The Secure Data Service (SDS) enables to store data and consume data in a secure and safe manner. The use cases use the SDS API in order to import/export data from Hybrid Cloud storage and to run analytic SQL queries and machine learning data processing on the data. The SDS exposes a REST API to the use cases, and in addition it sends data access events and security alerts to the SIEM (Keenai).

At the heart of the Secure Data Service (SDS) is parquet encryption, which is the new open standard for data encryption and integrity verification\(^1\). Data imported by SDS can be persisted even in untrusted storage in Parquet format\(^2\), which is an analytics-friendly columnar format for large-scale data. Parquet files containing sensitive information can be protected by the modular encryption mechanism that encrypts and authenticates the file data and metadata - while reaping all the benefits of Parquet format functionality, e.g. columnar projection, predicate pushdown, encoding and compression.

Master keys used for the encryption are stored in Hashicorp Vault\(^3\). In use cases, in which the partners are reluctant to store encryption keys in the public cloud, the Vault will be on customer premises, together with the SDS, whereas the data can reside in the public cloud, since it is encrypted. Access to data is governed by controlling access to the encryption master keys.

The general architecture of SDS and its interactions is presented in Figure 5.

![Figure 5. SDS High-level architecture](image)

The SDS is driven by an engine, which can be Apache Spark\(^4\), Apache Hudi\(^5\) (for table formats) or similar. Spark enables to import data from multiple data formats, and in particular from CSV files, which most use cases use, and to write them into Parquet and encrypted Parquet. Engines such as Hudi allow the SDS to work with table formats.

As part of its data access flows, the SDS will send data access events and security alerts to SIEM for further analysis.

There are different options for deploying and using SDS.

The first option is to deploy it as a service on use case partner premises, and to use its HTTP API, as shown in Figure 6.

---

1. [https://github.com/apache/parquet-format/blob/apache-parquet-format-2.7.0/Encryption.md](https://github.com/apache/parquet-format/blob/apache-parquet-format-2.7.0/Encryption.md)
2. [https://parquet.apache.org/](https://parquet.apache.org/)
4. [https://spark.apache.org/](https://spark.apache.org/)
5. [https://hudi.apache.org/](https://hudi.apache.org/)
In such a deployment, there is an SDS Gateway, implemented using the Play Framework and exposing methods, such as sql-query, insert-data, bulk-import in the HTTP API. The SDS Engine implements these methods using Apache Spark and Apache Hudi, with the help of an internal Table catalogue, where different tables can be configured, e.g. parquet files, CSV files and Hudi tables. The Java implementation of Parquet Modular Encryption (PME) is used for writing and reading encrypted parquet files with privacy and integrity protection, that enable secure and efficient work with encrypted data that is stored either in the cloud or on-prem. PME with key management tools can use either encryption keys managed by application, or keys managed by a Key Management Service, such as Hashicorp Vault, which also provides role-based access control to the encryption keys. Access control to encryption keys enables fine-grained access control to the data. For example, a large parquet can contain data collected from different sources, but various personas accessing it will get access only to the columns, to which master keys they are allowed access in Vault.

Another deployment option is as part of the use case tooling, as a library for securely reading and writing parquet data. It is shown in Figure 7.

In this deployment option, SDS is deployed as a library for reading and writing parquet data. We are working with the Open Source Apache Arrow community on the design and implementation of exposing Parquet Modular Encryption with Key Management Tools in PyArrow. The use case will be able to read...
and write parquet files and convert them to pandas, modin, etc. Then machine learning can efficiently run on this encrypted data with such common advanced tools as TensorFlow, Ray, etc. Again, as in the previous deployment option, encryption keys are managed either by application or by a Key Management Service such as Hashicorp Vault.

**IV.1.4. SL**

The Service Ledger (SL) is a middleware blockchain-enabled platform that offers programmable services that directly or indirectly interact with one or more decentralised applications. SL enables MEs/SMEs to collaborate and share CTI over a fully decentralised network without any centralised owner of data. The overview of the infrastructure is depicted in Figure 8. The Organisations (MEs and SMEs) and the Local Authorities share CTI and cybersecurity-related events via SL who indeed represents a gateway to the decentralised network. SL uses a decentralised data store system (to be defined between IPFS – InterPlanetary File System, and Solid Project) to store data. Stored data will then be linked to a blockchain network that will distribute trust among participants, and guarantee security, immutability and reliability of data shared.

![Figure 8. High level overview of the SL CTI sharing platform architecture](image)

Each Organisation and Local Authority runs a SL instance to participate in the decentralised network. Figure 9 illustrates the technical architecture of a single SL instance. The SL instance embeds a Web Application Server developed with the Node.js framework (we leave open the possibility of migrating to Django for compatibility with other tools). SL provides a data encryption service that manages the encryption of CTI data that needs to be shared. This component can be integrated with the IBM SDS tool. The web application is also provided with a PostgreSQL database to store user-related information.

SL embeds a TAXII 2.1 server (component not decided yet – might be the OASIS implementation of TAXII Medallion) to enable the sharing of CTI in STIX format, using the TAXII standard. The TAXII server exposes REST API that will be invoked by users via the Client Interface and forward data to the server via REST API.

The primary SL client interface is a web-based client application (ReactJS) that enables users to interact with SL’s functionalities. In addition, users can interact with SL via ad-hoc REST APIs.

SL’s users could have either the “admin” role for administration and configuration functionalities, or “user” role for common usage. The management of users is still uncertain due to the fully decentralised nature of the architecture. The usage of third party trusted authorities for identity management (e.g., Keycloak) represents a single point of control that should be avoided in a fully-decentralised network. A
better solution will be the implementation of a DAO – Decentralised Autonomous Organisation via encryption rules governed by the blockchain network. Nevertheless, this functionality is ambitious and of difficult implementations, and in case of difficulties a classic centralised (or semi-centralised) approach will be used for identities management and access control rules.

The SL server application directly communicates with an embedded blockchain node and data store node respectively for (i) the secure store of sharing events, and (ii) the persistence of data. SL relies on the Algorand blockchain platform for its principal functionalities of data sharing. An Algorand node is embedded into SL and configured by SL instance administrators. The Algorand node instance maintains a wallet with all the blockchain accounts and their public/private key pairs and communicates via the public Algorand blockchain via the embedded APIs. The SL’s blockchain functionalities are implemented via the Algorand SDK (JavaScript and Python sdks).

The data store functionality has not been implemented yet. It will be a peer-to-peer decentralised storage network and SL will embed one Data Store node to persist data accordingly. The principal candidates for building such a decentralised storage network are IPFS and Solid Project.

SL will be deployed as a multiservice containerised architecture (using Docker). Each SL instance will run 5 docker containers:

- The SL application deployed within a NodeJS server exposing the endpoint REST APIs and the frontend client interface accessible via web browser;
- A TAXII server listening for incoming connections from the frontend APIs, that communicates data straightforward to the server application;
- A PostgreSQL database;
- Algorand node full docker image: Algorand sandbox;
- The Data store node docker image (to be defined).

**Figure 9 Detailed architecture of a SL instance**

### IV.1.5. CJML

The Customer Journey Modelling Language (CJML) is used to model and visualise human behaviour and work processes in terms of **user journeys**, i.e., it is centred around the human activities and interactions throughout a process. A CJML model shows the actor’s steps (in terms of actions and communications) throughout the process. The models are used to illustrate hypothetical or real situations, for example best practices, but can also represent previous cyber attacks or situations to avoid.

This tool will be used to address human- and organisational aspects and more generally to contribute to increase awareness of cybersecurity risks, vulnerabilities and attacks. Figure 10 shows a high-level overview of the various components, which will be continuously improved in regular updates. The catalogue of scenarios consists of generic models and SME-specific models relating to the validation scenarios. In its current form, CJML consists of an initial collection of cybersecurity scenarios, templates (Microsoft PowerPoint and stencils in Microsoft Visio) and the graphical elements (bitmap and vector format).
Throughout the project, CJML will make use of various diagram types that serve different purposes. As an example, the journey diagram may be used to emphasize a deviation from best practices or vulnerability associated with a certain user (employee) behaviour.

**Figure 10. Component architecture – tools for human and organisational risk models**
(dotted lines represent future elements)

CJML aims at a broad user group in the SMEs to set the focus on the human element and errors made by employees. A module for training will be provided, in the form of workshops with the SME partners. Furthermore, an online module for training of new users will also be developed.

Currently, we develop software stencils in an open-source format to support diagramming with CJML, but we also consider developing a HTML5-based software tool for visualisation and validation of models.

**IV.2. Functional architecture**

The functional architecture as defined at the beginning of the project (see below) has been the basis for the definition of the interactions between the toolkit components.

**Figure 11. Project functional architecture diagram**

Some interactions are expected to evolve, and the functional architecture diagram will be updated once the interactions will be clearer, after a few iterations of the validation process.
IV.3. General architecture

IV.3.1. Conceptual architecture

The conceptual architecture showing how the various components will interact is presented below in Figure 12. Technical conceptual architecture.
### IV.3.2. System requirements overview

<table>
<thead>
<tr>
<th>Category</th>
<th>OS</th>
<th>CPU</th>
<th>RAM</th>
<th>Storage</th>
<th>Additional information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secure Data Services</td>
<td>Any (runs docker)</td>
<td>4 cores</td>
<td>16 GB</td>
<td>12GB</td>
<td>Lib : Java 8 (preferably 11)</td>
</tr>
<tr>
<td>Keenaï central platform</td>
<td>Any (runs docker)</td>
<td>From 8 to 64 cores</td>
<td>From 32 to 128 GB</td>
<td>4 TB (depending on EPS and retention period)</td>
<td>Hardware : Depending on the target: number of Event Per Second, Information System topology, the time for log/event retention, high availability or not, … A typical Keenaï instance : ~10 VMs/Docker Containers ~5 KEPS centralized High availability taken into consideration Prerequisites: - Docker installed - Web browser to access the Keenaï Administration Console</td>
</tr>
<tr>
<td>Keenaï collector</td>
<td>Any (runs docker)</td>
<td>4</td>
<td>8</td>
<td>4GB</td>
<td>Hardware : For a small (e.g. 10) number of users and occasional use, the hardware requirements are quite light-weight and the System Security Modeller software</td>
</tr>
<tr>
<td>System Security Modeller</td>
<td>Any (runs docker)</td>
<td>4</td>
<td>8</td>
<td>4GB</td>
<td>Software: - Docker and docker compose - Web browser (HTML5 support)</td>
</tr>
<tr>
<td>Customer Journey Modelling Language</td>
<td>Any (runs docker)</td>
<td>Web browser (HTML5 support)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service Ledger</td>
<td>Any (runs docker)</td>
<td>6</td>
<td>8 GB</td>
<td>1TB</td>
<td>Hardware : Minimum requirements for running a SL instance and the deployment of an Algorand participation node connected on the public main network and/or on a private network</td>
</tr>
</tbody>
</table>

Table 2. System requirements
IV.3.3. Security within the toolkit

In order to ensure the security within the toolkit, the interfaces between tools will rely on standard secure protocols. So will the displayed interfaces for integration, use and administration.

This means that the communication both within the toolkit and between the toolkit and the SME information system will be encrypted, ensuring data confidentiality and integrity for data in motion.

As some components of the toolkit will be co-hosted in a datacentre and others will be installed on SME premises, some interactions will take place over internet. In addition to the use of encrypted protocols, firewalls, proxies and authentication (through certificates) will be implemented.

Concerning data in use and data at rest, the technical partners are in charge of the security within their toolkit components and are mindful that, given its nature, the security of the application data is of the utmost importance. Authentication and authorisation is of course required (see below) and in addition:

- SSM: instances of the service are not shared between organisations, reducing the chance of a software flaw exposing confidential data to another organisation’s users. The service and database are deployed in a container and an analysis of security is being performed along with the implementation of measures such as container scanning at build time to identify vulnerabilities. We expect to have the service pen-tested in a separate project.
- CJML: no software artifact for CJML currently exists. Any documentation created using CJML should be kept as secure as necessary using usual procedures. If a software solution is created, it will be made secure by design.
- Keenai:
  - Strong authentication and password management (16 characters or more with at least one lower case, one upper case, one figure and one special character);
  - Access control (role management for data and feature access control) allowing to deferentiatce between admin rights and analyst rights;
  - TLS-encrypted communication between log sources and collectors;
  - Mutual TLS encryption for communication between modules (collectors / Central analysis Cluster, ... with strong cyber suite);
  - Encrypted communication on the admin console web portal (with strong cyphersuite);
  - Integrity controls (on logs, events, user activity traces, ...);
  - Password protection;
  - Hardening and securing of the Docker base: user rights, file access rights, removal of unused tools, ...;
  - Web interface protection against attacks and regular pentesting campaigns web: SQL injection, XSS, CSRF, ...;
  - Traceability and recording of user activity;
  - High availability of Keenai.
- SDS: instances of the service are not shared between organisations, and the sensitive data of the SMEs is stored by the SDS encrypted either on-premise or in the cloud. In order to be able to read the data, the user has to give SDS the access token that is used by SDS to send requests to the Key Management Service (KMS) to wrap/unwrap encryption keys. The KMS authorises the requests according to the permissions to use the master keys that are granted to the access token according to KMS policies.
- Service Ledger: data shared on SL is stored over decentralised system combining blockchain and decentralised storage – no centralised server – and properly encrypted with secure cryptographic standards. Data is shared ensuring integrity, confidentiality and immutability using the blockchain’s and decentralised storage append only data structures so that data cannot be deleted or overwritten. To cope with the need of data superseding caused by possible false positives, the SL will use a data versioning mechanism. Such a mechanism assigns versions to any data stored through SL – usually combining timestamps with additional details. For each data shared, the SL will expose dedicated APIs for retrieving all its versions and its latest
version. Users and tools consuming SL sharing services must always consider the latest version of data.

High availability and/or resiliency mechanism will also be studied for components of the toolkit that need to keep running.

The toolkit can rely on the SIEM in order to track user activity.

### IV.3.4. Authentication and Authorization

The CyberKit4SME toolkit is a loosely-coupled set of tools such that an SME can pick and choose which elements to use but also combine them easily when appropriate. Authentication and authorisation both have a role to play in enabling such flexibility and ease of use.

As well as flexibility in terms of choice of tool, we aim to provide flexibility in terms of the location of deployments. For instance, Keenaï and SSM are both primarily offered as SaaS (hosted outside of the SME’s infrastructure) but if desired, an SME could install either system on-premise. Generally, if a service is deployed on-premise then it is useful to integrate that service with a local identity management system such as LDAP or Active Directory. In contrast, for corporate SaaS systems, the expectation is that users generally must make additional new accounts in the cloud service (though there do exist paid-for systems such as Google Cloud Identity which can securely synchronise local identity data with a cloud-based system).

Where tools are integrated and work closely together it is more important to reduce friction and at least have the same username/password for tools and ideally have single sign-on (SSO). An example use-case is that of Keenaï and SSM where the cyber-analyst user is likely to need to move quickly from one tool to the other while investigating an alert.

We have settled on using KeyCloak for authorisation as it is a mature and fully-featured open source tool which will enable the desired flexibility and integration. Relevant features which may be used are:

- federation between KeyCloak instances (providing deployment flexibility)
- synchronisation with LDAP and Active Directory (ideal for on-prem deployments)
- kerberos bridge (lets an on on-prem service reuse a workstation account login)
- two factor authentication (“2FA”) (for added security)
- client adaptors for many languages (making integration easier)

The SSM already uses KeyCloak for authentication and we plan to use KeyCloak for authentication for Keenaï, SDS and Service Ledger (and potentially CJML). In the Keenaï/SSM use case sketched above, with both services deployed in the same domain (on-prem or a single SaaS site) the desired SSO behaviour will be possible and the user able to transition from one web application to the other.

Technically, SSM is a “client” of KeyCloak and uses the Open ID Connect (OIDC) “authorization code flow” to manage the communication between the user’s browser, KeyCloak and SSM. Keycloak also supports the other OIDC flows, and we will determine which is most appropriate for each tool, setting each one up as additional “clients”. SSM also uses KeyCloak to store roles for user accounts (“admin” and “user”) to do role-based access control (RBAC) on the API methods. In SSM authorisation on individual model resources is managed locally to the SSM service (such fine-grained authorisation data should not be managed in KeyCloak). Each tool brings different requirements for authorisation and roles, complicated by the fact that Keenaï is a multi-tenant service whereas SSM (for instance) is single-tenant. We will analyse the requirements to determine whether role information can be shared across tools.

Finally, the requirements and technical implementation will of course be modulated by the SME validation work in WP3 as he work is more firmly defined. We have so far taken a general view based on the state of the art and general expectations but each SME may also bring other aspects to consider.

---

6 [https://www.keycloak.org/docs/6.0/server_admin/#features](https://www.keycloak.org/docs/6.0/server_admin/#features)
IV.4. Deployment architecture

IV.4.1. General deployment architecture

The toolkit deployment diagram is displayed in Figure 13 with the components deployed on SME premises (left and side) and the mutualised ones (right-hand side).

![Deployment architecture diagram](image)

**Figure 13. Deployment architecture**
IV.4.2. Testbed architecture

The project’s testbed will:

- provide a central location for the centralised services, namely Keenaï and the SSM, for use in validation;
- provide a location to develop and test the integration between Keenaï, SDS, SL and the SSM;
- host a Test Lab to provide representative services (including SDS) and data for monitoring and asset discovery purposes (see IV.5.1.b. below).

An indicative architecture of the testbed may be seen in Figure 14. High level testbed architecture.

Both Keenaï and the SSM will be deployed using Docker in Virtual Machines. The Keenaï deployment will also include a node for the Service Ledger blockchain. The assets in the test lab are likely to be virtual machines on a separate VLAN.

In addition to the machines and services shown in the Figure, the necessary agents to collect logging data (from Windows OS) for Keenaï and for network scanning and discovery will be installed in the test lab and on the SME sites.

Some datasets will be required in order to test the various features of Keenaï. These datasets will in the first stages come from existing opensource datasets (such as the Mordor datasets7). In addition to these existing datasets, a set of machines, organised in a network, representative of an actual information system will be implemented in Inetum’s own testbed or on the project testbed. Finally, for the last stages

7 https://github.com/OTRF/Security-Datasets
of testing, actual datasets extracted from the information system of SMEs might be used in order to confirm that the implementation of the SIEM matches the existing assets. If the SMEs data have to leave the information system of the SMEs in order for the final tests to be carried out, the data will be anonymised through a protocol that is yet to be documented.
IV.5. Interaction between tools

As the SMEs might choose to implement only part of the toolkit, and since the toolkit is meant to be integrated with the widest possible range of existing solutions, as well as with one another, the aim is to keep the various components of the toolkit loosely coupled.

That would allow an SME to replace one of the components by another that they would have for instance already purchased previously.

To achieve this end, the toolkit components are kept as much in line as possible with the standards of the market in terms of formats (e.g. JSON), protocols (e.g. https, ssh, …) and interfaces (e.g. REST, …).

IV.5.1. SSM <> Keenaï

Integration between the SSM and Keenaï can provide some powerful advantages for operational monitoring of incidents and risk, with each tool enriching the context for the other.

The two systems take quite different approaches to modelling or describing an IT system. The origins of the SSM are in the security-by-design space with the idea that a system should be modelled and secured before it is deployed operationally. In contrast, Keenaï, as a SIEM, takes the approach of monitoring and making sense of what is found in the traces left in log files across the IT system. Essentially one specifies what should be and one interprets what is.

Commonly, the SSM is actually used not at the design stage but to model an existing system in order to understand the system’s risks and what additional security controls should be added to reduce the risk, or to create a baseline for understanding a proposed change to the operational system (which brings back in the security-by-design aspect).

By bringing live operational information from Keenaï into the SSM we can more accurately model the operational system.

By bringing information from the SSM into Keenaï we can compare and contrast what Keenaï is seeing in the monitoring data (“what is”) with what is expected in the design (“what should be”), adding useful context and highlighting divergence.

To combine Keenaï and the SSM there needs to be a mapping between the data models of the two pieces of software. This is discussed below, followed by a prioritised list of subsequent integrations. The precise nature of these integrations will be developed in the testbed and validated with the SME partners as we better understand their difficulties and the appropriate solutions. Indeed, the prioritisation may also change.

Any integration will be done through well-defined open interfaces, using open standards where appropriate in order to maintain the loose coupling philosophy and thus provide maximum flexibility for users of the software.

IV.5.1.a. Data Models

Keenaï, through its interpretation of log files has an understanding of the following assets types:

- hosts (identified by IP address and hostnames);
- services (identified as network ports on hosts) and that could be associated to applications (22/ssh, 80/http, 443/https, 3306/mysql, …); processes (identified by the path to a binary on a host);
- users (identified by individual user account IDs);
- files (identified by a URL or a full path on a host);
- networks (from firewall monitoring);
- spaces (defined manually or based on IP address).

Some relationships between assets are also understood, for instance:

- which services are hosted by which host;
D2.2 Technical specifications

- the existence of communication between hosts and/or services (from analysis of service logs and firewall logs);
- which processes are launched on a host;
- which files are created/modified/deleted on a host.

The main knowledge base deals with events (actions, transactions, activities, ...) that records all the associated and related entities.

These relationships between assets can easily be done using queries / filters inside Keenaï or using the Elasticsearch JSON API provided.

The SSM has a much more complete picture of the system: a lot of detail is necessary for the correct identification of threats. The asset types available in the SSM follow a hierarchical model and vary according to the configuration but generally include:

- hosts
  - fixed hosts (server, router)
  - mobile hosts (laptop, smart phone, tablet)
  - virtual hosts (VM, virtual router)
  - sensors
- services and processes (with sub-classes database, email client, web browser)
  - note that in contrast to Keenaï, in the SSM there is no distinction at the asset level between a “process” and a “service”: services are just processes that communicate with other processes over a network
- stakeholder roles (adult, child, human, organisation)
- data (with sub-classes genetic data, health data, special category data)
- networks (with sub-classes wired LAN, WiFi LAN, VLAN, Cellular network, internet)
- spaces (data centre, private space, public space, legal jurisdiction)

In SSM system models, the assets placed on the canvas should be considered to be classes of asset representing one or more identical assets in the real system. So for instance, not every identical desktop computer (on the same network) is placed into the model as a threat to one is a threat to all, and all should be protected with the same controls (such as anti-virus, etc). This raises a complication for mapping from Keenaï’s model to the SSM model. Keenaï will identify individual user accounts by account ID for instance but in an SSM model perhaps only a few user assets would be present, representing the few different user roles.

The asset to asset relationships in the SSM are detailed and too numerous to be documented here, but briefly, they do model which services are hosted by which host (as in Keenaï) and specify all expected process-process communication.

As can be seen, there is a great deal of overlap between the models of assets and relationships held in Keenaï and the SSM. To provide any integration value one model must be mapped to the other and to achieve this the SSM permits “additional properties” to be set on any asset. These additional properties are key/value pairs and can be used to make note of information that is not required by the SSM’s risk analysis, but which is needed to make such a mapping. For instance:

- hosts can record the IP address and hostname;
- processes can record the port they are available on;
- processes can record the CPE\(^8\) identifier associated with the software (i.e. software name and version).

By adding such additional properties, the two models can be mapped (to a degree) and the power of the two tools combined.

\[^8\text{Common Platform Enumeration: https://nvd.nist.gov/products/cpe}\]
In addition to modelling the assets and relationships, both systems have a model of threats to the system. The SSM describes each potential threat with a couple of sentences, with a few threats citing the OWASP top 10 list⁹. It would be helpful in both tools to add links to other threat taxonomies such as the MITRE ATT&CK framework¹⁰, providing a further link between the tools and also to external data.

**IV.5.1.b. Building and Using a System Model**

The SSM cannot provide any analysis until a detailed model of the system has been created, and from previous studies it is already known that creating the model can be difficult. Currently Keenaï has knowledge about the system through its analysis of log data (see above). It should be possible to use data and the knowledge from Keenaï to help bootstrap the SSM model. Rather than starting a model from a blank canvas, even just having a set of (mostly disconnected) assets that Keenaï is aware of placed on the canvas would be a useful start.

Furthermore, a new feature in Keenaï is planned which will allow the environment to be dynamically discovered and to provide context automatically (automatic discovery of Operating Systems and open ports / services). At the collector level, a periodic and scheduled discovery of IS hosts can be performed with a tool such as nmap¹¹ or OpenVAS¹².

In Keenaï, there are a lot of benefits using enrichment and context addition:

- Provide SIEM users with detailed information about the IS:
  - Enrich the operation view (visualization), for instance, using a fragment of the SSM model to provide context
  - Enrich the reporting view and specific dashboards
  - Advanced search using search engine
- Trigger Correlation Rules according to context information
- Trigger Alarms according to context information

Based on this discovery feature, Keenaï will be able to share this knowledge with the SSM to help build the model more easily, and later draw upon the model when appropriate.

**IV.5.1.c. Identifying Divergent Operational Events**

The SSM’s system model defines all expected asset-asset connections. This is necessary in order to find all paths in a system that an attacker or other failure may propagate through. Keenaï should be able to make use of the rich data in the SSM model to query events seen in a live system.

For instance, given an event, a query sent to the SSM model could identify whether the event was suspicious or not. For instance:

- unexpected process to process communication;
- access to unexpected ports on hosts;
- unexpected access to a sensitive asset (data or process);
- interaction with a process by an unexpected user;
- interaction with a process at an unexpected time.

Aside from the first example, all of these would need some enrichment of the SSM system model. For example, the SSM system model does not currently record which port is used by a process, or which assets are “sensitive”. Individual users recognised by Keenaï would need to be mapped to the roles described in the SSM (see above). The SSM model does not currently have any understanding of time but, in the same way that additional properties can be added to assets in the SSM model, we could extend this so that additional properties could be added to relationships. In this way, information about

---

⁹ Top 10 web application security risks: https://owasp.org/www-project-top-ten/
¹⁰ MITRE ATT&CK: https://attack.mitre.org/
¹¹ NMAP, the Network Mapper: https://nmap.org/
¹² OpenVAS, Open Vulnerability Assessment Scanner: https://openvas.org/
when a process to process communication was expected (or not) could be added to the relationship and queried by Keenaï.

IV.5.1.d. Operational Risk Assessment

The SSM’s risk analysis is performed on the basis of an analysis of what threats to the system are present (resulting from the topology of the model), the impact of the threat consequences on the system’s primary assets (specified by the modeller based on an understanding of the business) and the likelihood of each threat occurring. A threat likelihood determines the likelihood of an adverse effect and thereby determines the risk, based on the standard concept of risk being proportional to impact and likelihood.

The likelihood of a threat is computed based on a variety of factors, including a model of how likely each asset is to fail in a variety of modes, which vary according to asset type. When an SSM system model is used at design time these likelihoods are set to default values but at operation time they can be updated to reflect the current situation.

For instance, a “process” asset (representing a running software process such as a web server) has attributes representing how likely it is to be free of vulnerabilities that can be exploited by an attacker. Most software is buggy, so the likelihood of a vulnerability at design time is set to “medium” by default which will lead the SSM to recommend various controls be put in place to mitigate the risk of an attacker exploiting such a hypothetical weakness should it arise. When modelling an operational system, we can do better than this and where vulnerabilities are known for a particular software asset set the likelihood of a weakness to “very high”, or set the attributes to “very low” where no vulnerabilities are known of. Given a system model configured to reflect the operational system, a new risk assessment can be executed and recommendations for mitigating the risk provided.

The intention is to use the SSM in CyberKit4SME for operational risk assessment, and to provide an updated risk assessment along with recommendations for security controls to Keenaï. This information would then be logged in Keenaï, reviewed by an operator and acted on as appropriate.

To sources of data to update an operational model are described below.

IV.5.1.d.1. Software Vulnerability Data from OpenVAS

In the ProTego project\textsuperscript{13} which focuses on data-protection in hospitals and care centres, IT Innovation have been developing a prototype to update an operational model (representing the current system state) with vulnerabilities discovered through active scanning by OpenVAS. In CyberKit4SME we intend to consolidate this work. Keenaï also integrates with OpenVAS and so a single OpenVAS instance will be able to feed data to both the SSM and Keenaï. In ProTego, the OpenVAS data was first processed by another component before being sent on to the SSM. It is an open question whether the SSM should take a vulnerability feed via Keenaï or directly from OpenVAS but, in line with the principles of loose coupling of components, it is likely that the SSM integration will be directly with OpenVAS.

Detail on the existing OpenVAS integration may be found in ProTego D4.3, but in summary:

1. OpenVAS performs a scan of the processes on a network, detecting versions of software present (described by CPE) and uses the CVE database\textsuperscript{14} to look up whether vulnerabilities are present. Each CVE is described through a CVSS vector\textsuperscript{15} which further describes the vulnerability, for instance whether the attacker needs to be authenticated or not or on the local network or not.

2. Further information from OpenVAS links each CVE to a weakness type described through a CWE\textsuperscript{16}.

3. The vulnerability data is mapped onto assets in the SSM’s system model.

4. The SSM updates the attributes in the system model based on the presence (or not) of the different vulnerabilities, using the information in the CVSS vector and the CWE.

\textsuperscript{13} ProTego H2020 project: https://protego-project.eu/
\textsuperscript{14} Common Vulnerability Enumeration: https://cve.mitre.org/
\textsuperscript{15} Common Vulnerability Scoring System: https://nvd.nist.gov/vuln-metrics/cvss/v2-calculator
\textsuperscript{16} Common Weakness Enumeration: https://cwe.mitre.org/
D2.2 Technical specifications

5. The SSM recomputes the risks present in the system model.
6. The SSM models the addition of various controls to determine which ones are most effective at mitigating the current risks.
7. The SSM provides the updated risk levels and control recommendations to an operator (via other software).

IV.5.1.d.2. Security Events from Keenaï

Keenaï's monitoring and analysis of the operational system raises security alerts for the operator to review and to assign a security incident category to. Some of these events would be appropriate for updating the operational model held in the SSM in a similar way to that described above for OpenVAS data.

The default list of incident categories is:

- Exploitation of a vulnerability
- Elevation of privileges
- Data exfiltration
- Viral propagation
- Use of a persistence mechanism
- Denial of service
- Unauthorized access to a resource
- Identity theft
- Actions not in accordance with the security policy

Today in Keenaï, the association of an alert to an incident category is made manually by the user (during the analysis and the qualification of an incident). We plan, in the very near future, to associate automatically in real-time the incident category defined in the rule (detection/correlation rule) model to the raised alert.

When the SIEM raises an alert because it detects something suspicious about a user, a host or a process associated to suspicious signals, it could be useful for the SSM to have the knowledge of the potential threat for this asset.

IV.5.1.e. Further Potential Integrations

IV.5.1.e.1. Mapping from an SSM attack path to Keenaï rules.

Keenaï uses several techniques to analyse and correlate events.

Among them, it uses a scenario correlation engine with a logico-temporal approach (based on rules called chronicles or sequences) allowing the detection of attack sequences by writing specific and time-described scenarios. This correlation engine is based on time signatures whose principles are as follows:

- The succession of patterns based on filters describing the different steps of the scenario (Event A followed by Event B followed by Event C, ...),
- Events involved (A, B, C) can be correlation events,
- The use of time constraints between 2 patterns (I want to see B in the 10 minutes following A),
- The links between the different patterns (ex: IP src A = IP src B, ...).

When a rule matches a sequence of events, it raises an alert.

The attack path identified at the SSM level is close to a sequence in Keenaï, so it could be used to help the SIEM user create rules with the following point of attention:
Technical specifications

- A step involved in the rule definition and deduced from the attack path may refer to a log or a combination of logs, for instance:
  - Access to a RPM service from Internet
  - Followed by Access to RPM server
  - Followed by communication to DMZ
  - ...
- We need links to follow the graph based on any property (IP address, user name, process, ...) so the SSM needs to know this information to send it to Keenaï
- It is not necessary to consider all the steps (the whole graph) but some specific steps

At the moment, it seems difficult to automatically create a rule based on the graph provided by the SSM, so this feature is presented here as an additional possibility for which low priority is set.

**IV.5.1.e.2. Taking into account a SSM risk at the level of Keenaï**

Based on the idea that the SSM knows that there is a risk on a host, a service/application or a user, we can imagine the following workflow:

- The SSM sends an information about a risk on an asset
- The SIEM takes into account this information and associates a risk level or a score (high) to this entity (or these entities)
- Several uses of this risk knowledge could be considered in Keenaï:
  - On the existing rules: a severity increment could be performed (automatic adjustment of the severity level) when an alert is detected for a high scored entity. For instance, in Keenaï, if a use case (a rule) is triggered and an alert is created with severity 2:medium, if the related entity (the host, user, service, ...) is associated to a high score then the severity becomes 1:high.
  - A starting point for graph investigation,
  - A starting point for security reports,
  - When an alert is sent with the full associated detail (event information including context, logs, ...) the detail related to the risk provided by the SSM could be added. It can also be visible in the detail of an alert, in the operation view where the full content of an event is displayed (including information about CVE, vulnerability reports, IoC, ...) with risk information from the SSM.

**IV.5.2. SDS > Keenaï**

At the SIEM level, the SDS can be considered as a source of events/alerts like any sources the SIEM deals with, so the integration of the tools SDS and Keenaï can be based on a standard log collection using Syslog.

In this way on the SDS, a Syslog service (as a client) can be used in order to send SDS events in real-time to a Keenaï collector that already manages a Syslog service (as a server) able to consume events and to process them in Keenaï.

**IV.5.3. Keenaï <> Service Ledger**

The Keenaï / Service Ledger integration will be based on the use of a REST API compliant with the TAXII standard.
The communication between the tools is bi-directional and the exchange of information may be considered in both directions:

- From Service Ledger to Keenaï: Keenaï pulls information from Service Ledger (either manually or automatically) to retrieve a list of known IoC (and associated properties like an ip-src, ip-dst, hostname, filename, a checksum, ...) and to update its real-time detection module;

- From Keenaï to Service Ledger: Keenaï sends information related to a new or existing IoC to enrich the knowledge base and security alerts.

Once imported in Keenaï, IoC are transformed into signatures that are automatically deployed in the system. Theses signatures are used to enrich events in real-time and are based on:

- The identification of the matching attribute (IP, hostname, filename, ...) considered as the detected key;

- The addition of a sl_ioc_id as an enrichment property.

IoC could be imported:

- On demand;

- Or scheduled once per day for instance.

IV.5.4. SSM <> Service Ledger

Currently, no direct integration between Service Ledger and SSM has been identified. This does not mean there exists no link at all: data from Service Ledger will be conveyed to Keenaï and potentially from there to SSM (both as described above). A direct link between the tools (whether through technical or human integration) may be defined later, as the usage of the tools by the SME partners in the validation proceeds.

IV.5.5. CJML <> SSM

We are currently exploring how CJML and SSM can support each other. CJML is a process-oriented language based around humans and their actions, while SSM diagrams are predominantly technical, include few humans, and are primarily static but have some limited temporal aspects. While CJML has a high abstraction level when it comes to technical details and is designed to be understood by the general population, SSM is focused on technical details and dependencies.

Just considering the diagrams themselves, the clarity and relative simplicity of CJML diagrams could be combined with the technical detail of the SSM models to better describe a situation. We can see the utility of starting with CJML and getting more detail from a related SSM diagram or vice versa, starting from the SSM and using CJML to get a simpler view.

Starting with CJML, one possible connection is that the SSM can be used to “zoom in” on a touchpoint in a CJML scenario to understand the technical detail and explore mitigations. In this way CJML can provide an overview of a detailed SSM model and help build the broader picture by relating an SSM model to employees and their work processes.

Starting with the SSM, an attack path, or a particular attack technique could be better described using CJML, either as part of prepared training material or in some automated fashion. As noted above, the SSM models are primarily static, but the attack path does have a temporal aspect (one activity occurring before another) so is a potential candidate for integration.

The exact technical integration between these tools is under investigation. If the attack path diagramming is to be automated, then some work will need to be done in the SSM to summarise the quite detailed data currently in the model before interacting with any CJML software. The CJML metamodel description will also need to be extended. A component would be required to generate and layout CJML diagrams from such a description.
IV.6. External interfaces

In this chapter the various external interfaces, either existing or planned, are described for each toolkit component.

IV.6.1. Keenaï

- Web interface to access the Keenaï Administration Console (HTTPS/443)
- Integration of log sources:
  - The transmission mechanism in push mode from log sources is agentless and mainly based on Syslog (but not only) with the ability to secure communications. If necessary (this is the case with Windows systems), an additional agent (nxLog) is installed in order to monitor Windows event logs and specific files
  - Format: any format (raw logs, JSON, CSV, XML, multiline, …)
  - Protocol:
    - Syslog (TLS, TCP or UDP)
    - On the Keenaï collector side, a Logstash agent is provided authorizing the use of tens of additional protocols (Apache Kafka, file, TCP, HTTP, SNMPTRAP, S3, WMI, and many others…)
  - Port: Syslog (by default, common ports used are TLS/6514, TCP/601, UDP/514) or any other ports based on protocols accepted by Logstash
- Integration with other tools and third-party solutions:
  - Alerting: SMTP, SNMP, Web service, ticketing, …
  - Vulnerability scanner (OpenVAS) using the OMP (OpenVAS Management Protocol)
  - LDAP (OpenLDAP, Active Directory) for data enrichment
  - Based on structured flat files (CSV structure) with the ability to automatically load this content

IV.6.2. SSM

- SSM web interface supporting model management, limited user management and the modelling process:
  - Creating/deleting models
  - Defining the assets and their relationships (drag and drop UI)
  - Defining the impact of adverse effects (such as loss of confidentiality) on primary assets
  - Defining the expected likelihood of different failure modes
  - Finding the threats to the system and available security controls
  - Defining which security controls are present
  - Executing the risk calculation
  - Viewing and navigating the attack paths that have been discovered
- SSM API
  - The API is used by the web interface for all operations and is implemented as a RESTful interface using JSON
- SSM-adaptor API
  - The prototype work done in the ProTego project for operational risk modelling is partly implemented in a separate micro-service which receives formatted vulnerability data
through a RESTful JSON API, manages the updating and querying of the operational model and places alerts and control suggestions on a Kafka queue.

- Keycloak API
  - User authentication and management is provided by a Keycloak service

### IV.6.3. SDS

- SDS services exposed using HTTP API (REST API where possible) in one deployment option, and as method calls in another deployment option.
- Integration with assets to protect – data protected by SDS can reside in an object store, in a distributed file system or in any other storage supported by Apache Spark DataSource API.
  - Compatible technology – S3 API, POSIX FS, HDFS
  - Format – parquet
  - Integration process – SDS should be configured with the appropriate drivers, such as S3a for S3, and table definition should include a full path to the appropriate storage
- Key Management Service – Hashicorp Vault provides encryption key wrapping and unwrapping via its HTTP API

### IV.6.4. Service Ledger

- REST API endpoints: Exposed to the SL’s users for interacting with the TAXII server and exchange STIX data - data sharing actions (pull/push). The list of available APIs and a documentation will be available to users via browser (Swagger tool).
- SL will expose a Web interface where admins and users can login and interact with SL’s functionalities. The interface will be accessible via web browser.

### IV.6.5. CJML

A general version of CJML is available through SINTEF's web site\(^\text{17}\). In context of the CyberKit4SME project we tailor the general version to the field of cyber security.

Most users of CJML prefer to use templates in Microsoft PowerPoint for making models and diagrams. An alternative is to use stencils made in Microsoft Visio, but this requires a license.

During the WP3 Baseline analysis (Task 3.1), preliminary versions of the tool components were made available for the SMEs through the project's SharePoint site. Although the focus was not on the tool itself, it was used to illustrate situations and scenarios relevant for the SMEs' management of cyber security. For the validation, we will develop components for training and expand the catalogue of scenarios (generic and SME specific). From earlier experience in interacting with users, workshops are well suited to introduce the tool and make the SMEs able to generate their own scenarios for increased awareness among their employees.

Currently we are exploring open-source diagramming software (e.g. diagrams.net) and a HTML5-based software tool. However, from earlier experience we see that using PPT is a great advantage for several reasons: (1) everyone is familiar with the tool; and (2) they are free to draw what they want to explain without any constraints. These observations are important as CJML aims at a broad user group, in principle all employees in a company.

If it is determined that generating CJML diagrams from within the SSM software is useful and possible, then it is most likely that a software library to generate these diagrams would be the best approach for integration with the SSM.

However, if a fully-fledged service including a user interface for drawing diagrams is deemed necessary then a service-based interface for generating diagrams would be the best solution.

**IV.6.6. User manager**

As of now, each toolkit component has a separate user access management interface. Federation of access is under study and will be defined during the next stages.
V. FUNCTIONAL OVERVIEW

V.1. Our approach

The approach is the following:

1. Create a functional specification basis (object of this section)
   a. From the partner SMEs perspective (section V.2.)
      i. Identify the business pains that the SMEs face regarding security
      ii. Identify within the SMEs the user roles for the toolkit
      iii. Identify the general specifications that would alleviate the identified business pains
   b. From the technical partners perspective (section V.3.)
      i. Identify the features that would answer to the requirements listed in D2.1
      ii. Identify the features that would serve the project best based on their understanding of it and on the knowledge of their tools (ergonomics, interactions between tools of the toolkit, alignment with standards, ...)

2. Enrich the basis along the way through feedbacks from the SMEs and trials and errors from technical partners
   a. Enrichment of the requirements
   b. Further detailing of the already identified specifications
   c. Adjustment of an existing specification
   d. Identification of a missing specification
   e. Removal of a specification (not needed anymore, unachievable given the means, ...)

It ensues that the present section will not comprise an exhaustive set of specifications, but will provide a general direction for the evolution of the toolkit.

V.2. SME perspective

V.2.1. Business Pains to address

The business pains concerning information security identified during the workshops with SMEs and thanks to their feedback are quite in line with the ones usually identified with companies of other sizes. They are listed below:

- Comply with various standards and regulations (e.g. GDPR) and be in a position to provide evidence in case of an audit.
- Manage their security and data privacy and be confident about their levels.
- Ensure the confidentiality and integrity of stored data, moving data and data in use.
- Ensure the availability if their services.
- Be protected from security events on their information system or at least detect them and be notified in order to react in a timely manner.
- Ensure the security of their deliverables (e.g. software)
- Raise the awareness of their employees.

The difference with bigger companies lies more in the significance of a specific risk for the whole information system and the SME itself, as well as the means at disposal in order to address said risk.
D2.2 Technical specifications

As a result, the risks from one SME to the next might vary drastically, so as their expectations about security. The aim of the toolkit as of now is to address the core, most common security concerns and expectations. It will need to be completed with other security tools.

V.2.2. Roles

The main roles regarding the interactions with the toolkit that have been identified within the SMEs are the following:

- **Management roles**
  The management roles include people in position such as CEO, CISO, DPO and the likes.
  It is assumed that their main expectations will be to get reports and KPIs about security and compliance.
  Managers are not expected to have a high knowledge in IT, let alone in cyber security.

- **Toolkit technical user**
  The toolkit technical users will most likely be the IT administrators and the persons in IT security positions.
  Their main activities identified yet will be to use the toolkit in order to update the risk models, analyse data from the SIEM, integrate new assets to the protection tools, share security intelligence and produce dashboards and reports.
  Toolkit technical users are expected to have an average knowledge in IT, but not necessarily in cyber security.

- **Toolkit functional user**
  The toolkit functional users are people working directly in the line of business of the SME. They might not directly interact with the toolkit, but they might benefit from their outputs or help enhance the relevance of the inputs (e.g. risk analysis with regard to business).
  Their position will depend on the business of the SME and might include sales and pre-sales team, development or operation team, etc.
  Their main activities will be to control the adequation of the toolkit inputs and feedbacks and the reality of the business and to get security feedbacks in order to improve their work.
  Toolkit functional users are not expected to have a high knowledge in IT, let alone in cyber security.

- **Toolkit administrator**
  The toolkit administrators will be carrying out the configuration and operation of the toolkit. This role will most likely be assumed by IT administrator within the SME.
  Their main activities identified yet will be to configure the tools, ensure that they are running properly and intervene when they are not, and manage the accesses.
  Toolkit administrators are expected to have an average knowledge in IT, but not necessarily in cyber security.

As we are dealing with SMEs, the line between the various roles might be a theoretical one. The SME employees being few by definition, they tend to fill in multiple roles.

V.2.3. General usage

The purpose of this chapter is to list the expected usage that we identified with the SMEs in order to ensure that we do not overlook any feature to implement in the toolkit component roadmaps.

It is therefore only a high-level overview of what might be needed and is not meant to exhaustive nor very detailed.
D2.2 Technical specifications

The security specific aspects including what is the expected scope of protection and of detection has not been included since it is very specific to each SME and will be taken into account individually in the validation scenarios.

V.2.3.a. Reporting

The following items are interesting for management roles:

- Periodic reports
  - Issues/attacks reports taking into account the associated risks
  - Compliance / Risk coverage reports and associated mitigation plans
  - KPIs regarding the level of security and its evolution
- Real-time dashboards
  - Health of a specific scope (endpoints, network, application, software delivery process, …)
- Notifications / Alerts in case of a security event and associated remediation plans

V.2.3.b. Use

The following items are expected to be carried out in a simple manner for the toolkit technical users:

- High-level configuration of the tools
- Production of reports and dashboard
- Browse reports and dashboards

Tool specific expectations:

- Risk modelling
  - Model threats from employee behaviour
  - Model threats from customer behaviour
- Risk analysis
  - Create risk analyses
  - Get overviews of the risk
  - Get overviews of the mitigation plans to implement
  - Update risks on the running information system
  - Raise alerts if new risks arise or if existing risks become more critical
- Protection
  - Secure a specific scope depending on the associated risk
  - Protect data in motion
  - Protect data in use
  - Protect data at rest
- Detection
  - Track access and denial of access to software and to infrastructure
  - Track access and use of data
  - Detect security incidents and be notified
  - Investigate potential and recognised security incidents
- Information sharing
  - Get intelligence about existing security threats
V.2.3.c. Administration

The following items are expected to be carried out in a simple manner for the toolkit administrators:

- Deployment of the toolkit components
- Integration into the existing Information system
- Addition or removal of a user
- Low-level configuration of the tools

Integration impact to the existing architecture and to the performance of the information system should be low and non-existent.

V.3. Technical partners perspective

The purpose of this chapter is to list the features that have been identified by the technical partners as needing to be developed or enhanced in order to improve the toolkit components for the target audience of SMEs.

It does therefore not include the existing features and is not meant to be exhaustive.

V.3.1. Keenaï

In the scope of the project, new features will be implemented in Keenaï:

- MITRE ATT&CK compatibility: the integration of the MITRE ATT&CK knowledge base / framework in Keenaï will help users to better understand the motivation behind an attacker’s actions and tactics. It will facilitate the investigation by adding context to the different phases of an attack and will also be useful in mitigation and impact reduction. This feature includes the addition of correlation rules and uses cases and a prebuilt content mapped to MITRE ATT&CK.

- Investigation module: it will allow an investigation based on graphs. The graph will be built according to the logs received by the module. The user will be able to identify the different entities present in the logs as well as the relationships between them. The idea will be to interact with the graph (clicking on nodes) without having to perform textual and list-oriented searches. It is a more natural approach and closer to the functioning of humans who think by entities (server, file, user, ...) and by relationships between them (a user has accessed a file ...) and not using lists which is the way to analyse/search usually proposed in a SIEM.

- Knowledge:
  - Vulnerability Management: enhancement of the existing feature with the possibility to launch on demand and schedule scan tasks directly from the SIEM.
  - Discovery mode: to discover hosts and services automatically in order to add knowledge and to enhance real-time detection

- Usability:
  - Parser creation enhancement
  - Dashboard catalogue enhancement
  - Detection ruleset/catalogue enhancement
V.3.2. SSM

Usability:
- The interface for navigating attack paths through the system will be improved, helping the user understand cause and effect and where best to place security controls.
- We will add a way to automatically build part of the system model so that the user does not have to start with a blank canvas.
- Other enhancements determined to be needed through validation and the evolution of the SME’s requirements.

General features:
- Differentiation between internal concerns and those relating to an SME’s customers.

Alignment with standards, providing further context for the users’ understanding:
- Control strategies will be aligned with ISO 27001 Appendix A.
- Compliance with GDPR will be modelled.
- We will investigate aligning the threats in the knowledgebase with MITRE ATT&CK.

Reporting:
- Reports aligned with ISO 27001 will be added.
- Further reports may be added according to requirements determined through validation.

Operational risk assessment:
- The existing prototype integration with live vulnerability data from OpenVAS will be consolidated and improved.
- Events from Keenaï will be used to improve the operational model where appropriate.
- Alerts containing problems and potential mitigations will be raised.

Note that some of these items may be completed in other parallel projects as part of a wider software roadmap. Which work has been charged to this project will of course be clearly communicated in future reports.

V.3.3. SDS

The Secure Data service (SDS) enables to store data and consume data in a secure and safe manner. It is a new service, but it is based on the recently published technology of Parquet Modular Encryption. In the scope of the project the following new features are targeted to be developed:

- An SDS package for easy deployment, so that Use cases can easily install SDS on their premises.
- Dataset integrity protection – currently the integrity protection provided by Parquet Modular Encryption is on a file-level, and doesn’t allow to detect modification on the dataset level.
- Parquet Modular Encryption in PyArrow – to enable python developers to use Parquet Modular Encryption for writing and reading encrypted files with privacy and integrity protection.
- Generation of audit and security events, e.g. data integrity violations, access to encryption keys for reading encrypted data, failed attempts to access encryption keys.

V.3.4. SL

SL offers data sharing futures while preserving security, data integrity, privacy and business confidentiality. The SL’s features that will be implemented for the project are listed below.

- Secure data sharing: CTI shared on a distributed data storage (DDS), and secured over the blockchain.
D2.2 Technical specifications

- **Tokenisation of CTI**: CTI data will be secured on the blockchain as digital tokens. Any token will point at its relative data source stored on DDS. Tokenised CTI are immutable and tamper proof representations of shared data. This consent SMEs to share and collect trusted and verifiable data;

- **Privacy preserving sharing**: SMEs can decide to store anonymised data on the DDS to preserve privacy. In this case, the blockchain’s tokenised versions of CTI data will be properly anonymised too;
  - **Anonymous transactions**: Data secured on the blockchain will be stored through one-shot blockchain accounts (anonymous);

- **Custom sharing groups**: SMEs will have the possibility to share data amongst custom groups to preserve business confidentiality. This future will take advantage of data encryption for DDS (data will be accessible only by group members) and private/permissioned blockchain networks.

V.3.5. CJML

CJML currently exists in a descriptive format and falls under the category of a domain-specific modelling language. However, CJML is described in terms of a meta-model with concepts and attributes specified, and work is in progress to extend CJML to become executable. This can be achieved in different ways, depending on the purpose and the specific user needs. We are currently exploring CJML in its conceptual form with the users (SMEs) and with other tool owners to search for effective interfaces and transitions to other tools.

Should it be determined that a drawing tool is necessary, CJML will be distributed as a web application that runs on a web server under the HTTPS protocol. For the client to access the application, an HTML5-capable browser will be needed. CJML’s design and development is inspired by the diagrams.net functionality and operation while using elements from its open source framework. In the scope of the project the following new features are planned to be developed:

- **Data Store**: In our case, it is a SQL Server database. There are two separate databases used - one for storing Users information and the other for Property Information. This arrangement is done to isolate the security information (e.g. Username, Roles, etc.) from the application database.

- **Data Access Layer**: This layer provides entities, using which read-write operations can be performed easily.

- **Web API Service Layer**: Since the data needs to be exposed to the web and to various clients, we need open standards for data communication. This is possible using JavaScript Object Notation (JSON).

- **UI Layer**: This layer is implemented using HTML5 for Data Display and accepting values.

The relationship between the new features and the formed targeted architecture is visualized in Figure 15.

![Figure 15. Targeted CJML application architecture](image-url)
ANNEX 1 – PRESENTATION OF THE TOOLKIT COMPONENTS

This chapter presents the tools that were selected to be part of the CyberKit4SME toolkit as presented in chapter III. PRESENTATION OF THE TOOLKIT SOLUTIONS of deliverable D2.1 - Requirements analysis.

In each case, the tools are based on previous and ongoing research by the relevant partner. The main focus in CyberKit4SME will be to adapt the tools and models to make them accessible by SMEs & MEs without the need for scarce and expensive specialist cyber security expertise.

Five tools are presented:

- The Secure Data Services (SDS), developed by IBM
- Keenaï, developed by Inetum
- The System Security Modeller (SSM), developed by the University of Southampton
- The Customer Journey Modelling Language (CJML), developed by the SINTEF
- The Service Ledger (SL), developed by the University of Southampton

Secure Data Services

The Secure Data Services (SDS), are services that protect data across its lifecycle as it is stored, accessed and used. This includes facilities to govern data access and validation across the whole data lifecycle. The usage of the SDS in the context of a use-case is presented in Figure 16. Usage examples:

- Database engine: data insert / delete / get / query
- Getting data from standard sources (streams and file formats)
- Encrypted export/import of bulk data
- Anonymized export of bulk data
- Deleting individual records (e.g., for GDPR)
- Cloud backup and ransomware protection
- Use for data and/or metadata

- Applicability in wide SME ecosystem
  - demo inside existing CyberKit4SME use cases
    - e.g. database, data import/export
  - demo as additions to existing CyberKit4SME use cases
    - e.g. cloud backup
  - demo in new use cases (modelled after existing, or other typical SME apps)
    - all features

- Properties
  - free open source technology
    - no download or maintenance charges
  - easy to use
  - (almost) transparent data security support
  - standard interfaces and formats
  - helps to leverage hybrid cloud
    - offload SME IT tasks to public cloud: less expensive, less headache
    - no cloud lock-in! use / switch to any public cloud
    - keep extra-sensitive data on-premises
Keenaï

Keenaï is a tool that allows monitoring, through a single entry point, the whole security of the Information System. The solution is part of the SIEM's category (Security Information and Event Management).

The development of Keenaï started in 2009. It is under the responsibility of the Inetum Cybersecurity Business Unit (R&D located in Rennes, France).

Keenaï has received the French state support and security certification is in progress.

Keenaï main objectives:
- To record continuously the Information System activities,
- To detect internal and external attacks in real-time,
- To identify and reduce security threats.

The figure below (Figure 17) presents possible sources and the global functionalities of Keenaï:

![Figure 16. Presentation of the SDS](image)

![Figure 17. Sources and global functionalities](image)
D2.2 Technical specifications

Keenaï offers the main following features:

- Collection of the events produced by the applications and appliances (both physical and virtual) of the IS: Firewalls, antivirus, web servers, business applications, Intrusion Detection Systems...
- Centralisation and storage of the events forwarded by the collection modules,
- Standardization, enrichment, analysis and correlation of security events,
- The synthetic presentation (dashboards, statistics, ...) of the IS security,
- The sending of alarms to notify incidents of security.

The Keenaï solution is multi-tenant and can be deployed both on SaaS or on-premise. The data collected is processed and analysed in real-time thanks to a scalable and fully distributable architecture.

The analysis is mainly based on:

- Context addition
- Signature/Correlation engines
- Threat Intel integration
- Machine Learning unsupervised and supervised algorithms combination

System Security Modeller

The System Security Modeller (SSM) is a risk assessment tool for socio-economic systems. It combines a drag-and-drop graphical interface for drawing system models with an innovative machine-reasoning engine and detailed domain knowledgebase of threats and countermeasures to create a comprehensive view of the risks to a system and how to mitigate them.

The SSM automates much of the risk assessment procedure described in ISO 27005 and thereby supports ISO 27001 compliance. Through automation, a risk assessment is made methodical and reproducible and a security analyst may do a better job in less time.

To use the SSM the procedure is:

- Draw a model of the system, including relevant assets (networks, hosts, processes, data, people, places) and their relationships (such as which process uses what data).
  - This requires an understanding of the physical/virtual infrastructure (network, hosts), the software and data used by a company and the environment (people, places).
- Identify the primary assets for the business (generally data and processes) and indicate the impact on the business that failures in those assets (such as loss of confidentiality) would cause.
  - This requires an understanding of the business.
- The SSM then finds the threats to the system automatically using the built-in knowledgebase and through its understanding of attack-paths and threat cascades.
  - This would normally be done (imperfectly) by a trained security analyst.
- Specify what control measures are already in place (such as passwords, firewalls, etc).
  - This requires an understanding of the information-security measures already in place.
- The SSM then computes the risk of every threat to the system automatically.
  - This is very hard to do by hand and would be done by a security analyst. It involves the use of the specified impacts, the inter-connectedness of the assets (to see how failures in the secondary assets affect the primary assets) and an understanding of how the controls that are in place effect the likelihood of each threat.
D2.2 Technical specifications

- Add additional controls suggested by the SSM and recompute the risk until the residual risk is acceptable for the business.
  - This requires an understanding of what information-security measures are possible for implementation.

All together the ISO 27005 risk assessment procedure is a complex process, requiring knowledge about many aspects of a business. Such a procedure has at times been considered too difficult for SMEs to manage, hence simpler descriptions of necessary security measures such as the UK’s Cyber-Essentials have been introduced by regulators.

The System Security Modeller goes a long way to automating and simplifying the procedure and in Cyberkit4SME the additional needs of SMEs will be understood and addressed so that SMEs can also be supported in performing risk assessments of their systems and implementing the controls appropriate to their risks.

CJML

The Customer Journey Modelling Language (CJML) is a visual language for the modelling and visualisation of services and work processes in terms of user journeys. The adoption of user journey methods has progressed rapidly among public and private service providers to emphasize the human element in digitalisation of our society’s service systems. CJML is centred around the human activities throughout a process, and the humans can have various roles (e.g., a customer, an employee, a citizen, a patient, etc.). Example of journeys are online shopping, tax reporting, or hiring a new employee. A user journey is a model of the steps in the process that are visible or can be encountered by the user.

The target group of the modelling language includes service providers, but also researchers and consultants, see Figure 18. The core elements of CJML targets a wide and heterogeneous user group and does not require a technical background. CJML differs from general-purpose modelling languages in two principal ways:

1. It models the service process from the end user’s point of view
2. It aims at being intuitive for all users, and does not require a technical background

CJML enables modelling of service processes that extends over time and involve one or more end users interacting with service providers.

![Figure 18. CJML targets a wide user group](image)

CJML consists of terminology, a visual notation, and supporting methods and tools (guidelines, examples, templates). The basic concepts of the language are customer journeys which consists of a chain of touchpoints. The touchpoints constitute the basic elements of the journey, and their visual appearance depends on the type of diagram used, see Figure 19.

The simple journey diagram is suitable for journeys with few actors, and it emphasises potential deviations in the actual journey as compared to the planned journey. The swimlane diagram is useful...
D2.2 Technical specifications

for journeys involving several actors, and thereby identifies both the initiator and recipient of a touchpoint.

![Journey diagram and Swimlane diagram]

**Figure 19. Customer journey and touchpoints**

CJML describes user journeys in two states: The hypothetical state (planned journey) and the execution state (actual journey). This makes it suitable to compare reality against theory (what was supposed to happen). As an example, it is possible to use the journey diagram type to emphasise and compare best-practice behaviour against unwanted behaviour. For example, the diagram can show how an employee is expected to react to a phishing attack, compared against the unfortunate situation of being victim to a phishing attack and the consequences that may occur.

A scenario is a good starting point to make a CJML model, and the essential steps to make a model is:

- Identify the actors that have a role in the scenario
- Identify the communication points (instances of communications) among the actors and their essential attributes (the sender, the receiver, the communication channel, time etc)
- Identify essential actions (non-communicating events)
- Depending on the diagram type chosen, insert the touchpoints and connect them.

Service Ledger

The Service Ledger (SL) is a blockchain-as-a-service platform that offers programmable blockchain-enabled services that apply in several application scenarios.

Use case example: enabling the sharing of security information between different entities/organisations

- The organisations involved can share CTI information to improve the awareness and the defence against threats and malicious activities

SL platform is used for sharing CTI coming from

- the security and monitoring features of the SIEM tools
- the local authorities CERT/CSIRTs that aim to share CTI (like IoC and TTPs)
- Advantages of a solution based on blockchain
- Blockchain technology distributes trust and preserves strong integrity on shared CTI
- SL permits privacy preserving security information sharing allowing the deployment of private, permissioned blockchains – specific Communities of Interest (CoIs)

SL is built on top of the Algorand blockchain platform

Algorand\(^\text{18}\) is one of the most valuable blockchain platforms. It’s a secure, decentralised, and scalable solution

Service Ledger Main components (see Figure 20):

\(^{18}\) https://www.algorand.com
D2.2 Technical specifications

- **Configuration manager**: Algorand deployer and configurer
- **Privacy manager**: policy-based privacy rules definition (to be finalised in WP5)
- **Data access control manager**: Programmable access control rules for CTI read/write operations
- **Orchestrator**: Connect the blockchain node with either a permissioned or permissionless Algorand network
- **Blockchain explorer**: user friendly dashboard for the blockchain inspection
- **Smart contract application manager**: deployer of Algorand’s client applications that interact with the blockchain

\[\text{Figure 20. Presentation of the Service Ledger}\]

- SL embeds a software module for the CTI sharing
  - Algorand’s programmable smart contract able to persist and transact the blockchain CTI based on the STIX 2.0\(^{19}\) standard
  - TAXII 2.* protocol embedded in SL for the exchange of STIX CTI
    - CTI consumers can access CTI resources on SL over HTTPS
    - CTI producers can publish (and share) CTI resources on SL over HTTPS
- Possible usage of SL
  - An organisation take part in the sharing platform according to its privacy requirements
    - Configuration of Algorand blockchain instance and the privacy policies
  - Enable the sharing of CTI data either publicly or within a particular CoI
    - Configure the orchestrator to participate in an existing permissioned or permissionless blockchain
  - The organisation shares (or accesses) STIX data through the embedded client application by using the SL APIs – TAXII protocol

\(^{19}\) https://docs.oasis-open.org/cti/stix/v2.0/cs01/part1-stix-core/stix-v2.0-cs01-part1-stix-core.html