MITIGATE

EUROPEAN COMMISSION

MITIGATE

Multidimensional, IntegraTed, rIsk assessment framework and dynamic, collaborative risk manaGement tools for critical information infrAstrucTrurEs

www.mitigateproject.eu

Grant Agreement No.653212
Topic: H2020-DS-2014-01
"**Risk Management and Assurance Models**"
Innovation Action

Deliverable D2.3

System Architecture and Technical Specifications

Contractual Date of Delivery:	M8 / April 2016
Editor:	Panagiotis Gouvas (SingularLogic)
Work-package:	2
Distribution / Type:	PU / Other
Version:	1.0
File:	D2.3_v1.0.docx

Project co-funded by the European Union within the Horizon 2020 Programme

Abstract

This deliverable reflects the outcomes of task T2.5 and provides details regarding the high level architecture of the MITIGATE risk management system. More specifically, this deliverable provides information regarding the various components that comprise the high level architecture and elaborates on the interdependencies between them. Per each component a specific analysis is performed regarding the offered functionalities and the component-interactions that are required.

MITIGATE envisages to deliver a unified environment where sophisticated risk management features will be offered taking under consideration complex supply chain services that are executed among organizations. Such features include collaborative risk assessment, intuitive visualization, simulation and open intelligence analytics. These features are offered based on the smooth integration of the aforementioned components. The componentization and the identification of components' interdependencies along with their interactions, that this deliverable provide, constitute an essential input to D3.1 where normative interfaces will be defined.

Executive Summary

MITIGATE aims to provide a holistic solution regarding risk management in the frame of port supply chain services. To do so, several services have to be provided such as collaborative risk assessment, advanced simulation and visualization of potential attacks, open intelligence analysis services etc. Such services can be achieved though the usage of existing isolated tools; yet MITIGATE envisages to create a unified environment targeting security analysts where these services will be seamlessly offered. In order to achieve this goal, MITIGATE has formulated a high level architecture that consists of eight coarse grained components that complement each other. These components include:

- an Asset Modelling & Visualization component that allows users to declare their assets along with the cyber relationship and serialize this declaration in a strict format which will be addressed as "Asset Cartography". Each organization that participates in a supply chain service will use this component in order to create its own cartography which will be automatically linked to available vulnerabilities and attack-types;
- a Supply Chain Service Modelling component that allows security analysts to model the supply chain services that are performed by their organizations while also allowing to provide the mapping of the existing assets with the various processes that are defined in the frame of SCSs;
- a Simulation & Game Theory component that it is responsible for the discovery of attack paths given a specific asset cartography and the calculation of the best defensive strategy regarding the protection of a specific asset based on game theoretical principles;
- a Collaborative Risk Assessment component which is responsible to guide the security analyst in order to perform the appropriate steps that are required for the conduction of a risk assessment for a specific supply chain service so as to be inline with the methodology as defined in D2.2[1];
- an Open Intelligence and Big-Data Analytics component that is responsible to provide near real-time notifications regarding potential vulnerabilities that are related to the assets that exist in the asset cartography of one organization through the text-mining of open sources;
- a Notification and Reporting component that is responsible to provide push notifications to the security analyst regarding any type of messages that are raised from the time-consuming operations such as the conduction of a vulnerability assessment, the calculation of risks, the processing of open information sources etc.;
- an Administration component that is responsible for the management and the consistency of the various 'enumerations' that are required by all the other components (e.g. vulnerabilities, attack-types and business partners);
- an Access Control and Privacy component that provides security guarantees in a horizontal manner to all the other components

Finally, it should be noted that the architecture is complemented by a persistency layer and a pub/sub system. The persistency layer consists of two types of databases; one relational that is used in order to store fully structured data and one NoSQL that is used in order to store semi-structured data that change frequently (e.g. Vulnerability reports). The pub/sub system is used in order to decouple the communication of the components and more specifically to eliminate any blocking communication that may be required.

It should be clarified that the purpose of this deliverable is not to provide normative interfaces. Instead, based on the fine-grained componentization and the identification of components' interdependencies and interactions, this deliverable will provide an essential input to D3.1 where normative interfaces will be formulated.

Version History

Version	Date	Comments, Changes, Status	Authors, contributors, reviewers
0.1	12/02/2016	First version of the ToC	N. Polemi, S. Papastergiou, A. Karantjias, N. Drosos
0.2	23/02/2016	Second version of the ToC	N. Drosos
0.3	30/03/2016	Contribution form the partners	A. Duzha, M. Bianchi, F. Campana, T. Erler
0.4	10/4/2016	Asset Modelling, Supply Chain Service Management, Collaborative Risk Assessment	S. Papastergiou, A. Karantjias, P. Gouvas
0.5	19/04/2016	Simulation & Game Theory	H. Mouratidis, E. Rekleitis, M. Pavlidis, S. Schauer
0.6	21/04/2016	Notification & Reporting, Access Control	N. Drosos, P. Gouvas
0.7	21/4/2016	Open Inteliigence & Big Data Analytics, Administration	A, Duzha, F. Campana
0.8	25/04/2016	Release for Internal Review	P. Gouvas
0.9	27/04/2016	Comments from Reviewer	A. Karantjias, , S. Papastergiou
1.0	30/04/2016	Final Release	P. Gouvas

Contributors

First Name	Last Name	Partner	Email
Nineta	Polemi	UPRC	dpolemi@gmail.com
Spyros	Papastergiou	UPRC	spyrospapastergiou@gmail.com
Athanassios	Karantjias	UPRC	thanos.karantjias@gmail.com
Nikos	Drosos	SiLo	ndrosos@singularlogic.eu
Panagiotis	Gouvas	SiLo	pgouvas@gmail.com
Armend	Duzha	MAGG	armend.duzha@maggioli.it
Manuel	Bianchi	MAGG	manuel.bianchi@maggioli.it
Flavio	Campana	MAGG	flavio.campana@maggioli.it
Timo	Erler	Franhofer	timo.erler@iml.fraunhofer.de
Stefan	Schauer	AIT	Stefan.Schauer@ait.ac.at
Haris	Mouratidis	UoB	H.Mouratidis@brighton.ac.uk
Michalis	Pavlidis	UoB	M.Pavlidis@brighton.ac.uk
Evangelos	Rekleitis	UoB	E.Rekleitis@brighton.ac.uk

Glossary

AC	Access Complexity
AV	Access Vector
CIL	Cumulative impact level
CVL	Cumulative vulnerability level
CVSS	Common Vulnerability Scoring System
ICIL	Individual Chain impact level
ICVL	Individual Chain vulnerability level
IVL	Individual Vulnerability Level
NIST	National Institute of Standards and Technology
NVD	National Vulnerability Database
RM	Risk Management
PIL	Propagated impact level
PVL	Propagated vulnerability level
SC	Supply Chain
SCS	Supply Chain Service
SCSBP	SCS Business Processes

Table of Contents

List of Figures

1 Introduction

1.1 Scope and objectives

The purpose of this deliverable is to provide details regarding the high level architecture of the MITIGATE risk management system. Towards these lines, the overall architecture is presented along with a fine grained componentization scheme. MITIGATE will provide a set of functional features such as collaborative risk assessment, visualization, simulation and open intelligence analytics. All these features rely on discrete components that implement the aforementioned functionality. However these components have to collaborate to each other since the end-most functionalities of MITIGATE will be achieved through the synergy of these individual components. Therefore, beyond the elaboration of the high level architecture, the identification of the component interdependencies is one of the major objectives of the current deliverable.

Furthermore, since most of the components rely on existing software artefacts it is extremely essential to identify the complementarity or any potential conceptual mismatch regarding the data structures that the various component interactions may raise. Since the aim of MITIGATE is to come up with a unified environment where all aforementioned services are offered seamlessly it is extremely important to identify the interactions between the various components. It could be argued that this is the most crucial aspect of the deliverable since the identification of interactions will drive the process of creating normative interfaces. Based on the fine-grained componentization and the identification of components' interdependencies and interactions this deliverable will provide an essential input to D3.1 where normative interfaces will be defined for all services.

1.2 Structure of the Deliverable

The deliverable is structured as follows: initially the high level architecture is provided (chapter 2). Based on the high level architecture the functional components that comprise the architecture are briefly discussed. From chapter 3 until chapter 10 each one of the components are analysed separately. For the sake of comprehensiveness and clarity, each of the components is analysed using a specific template. The template consists of four parts. The first part relates to the brief description of the business logic of the component while the second summarizes the discrete functionalities that are offered. The third part elaborates on the internal architecture (or control flow) of the component while the last part lists the various interactions that each component may have from other components. Special emphasis has been given in maintaining a common granularity between the 8 components.

2 High Leve Architecture

2.1 Overview of the Architecture

As already mentioned, MITIGATE system aims to provide a holistic solution regarding risk management in the frame of port supply chain services. To this end, specific set of services need to be developed and integrated in a seamless manner. Such services include assessment of risk in a collaborative manner among organizations, advanced simulation and visualization of potential attacks and advanced reports from open intelligence analysis services. In order to achieve the goal of developing a unified system, a high level architecture has been defined. This architecture is presented on Figure 1.

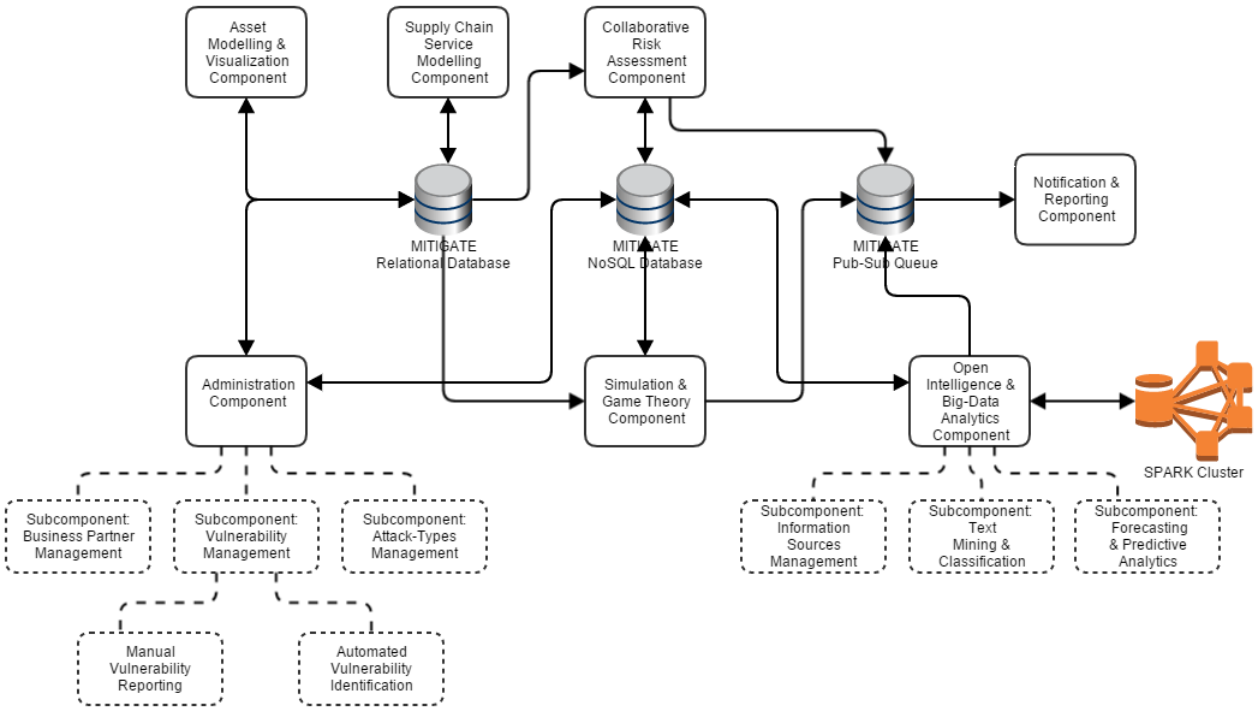

Figure 1 - MITIGATE High Level Architecture

As it is depicted there are seven main components that comprise the MITIGATE system namely **a)** the Asset Modelling & Visualization, **b)** the Supply Chain Service Modelling, **c)** the Simulation & Game Theory, **d)** the Collaborative Risk Assessment, **e)** the Open Intelligence and Big-Data Analytics, **f)** the Notification and Reporting, **g)** the Administration and **h)** the Access Control and Privacy Component which is not depicted on Figure 1 since it is totally horizontal to all the other components. Although chapters 2 to 10 elaborate on each one of these components separately, the next section will provide a short description of each component.

2.2 Bird's Eye View on Components

Initially, the **Asset Modelling & Visualization** component allows security analysts (also mentioned as assessors in the frame of the deliverable) to declare their assets along with the cyber relationships. This declaration will be serialized in a strict format which will be introduced by the project and will be addressed as "Asset Cartography". The creation of a valid asset cartography within the frame of an organization is the first step towards the realization of a collaborative risk assessment. Each

organization that participates in a supply chain service will use this component in order to create its own cartography. The cartography will be automatically linked to available vulnerabilities and attack-types that are relevant to the individual assets that are declared. The cartography along with the linked information will be intuitively visualized by a graph rendering subcomponent of this component.

The **Supply Chain Service Modelling** component allows users to model the supply chain services that are performed by their organizations. More specifically, supply chain services consist of various business processes that are performed in a synergetic way among different business partners. Each business partner has a predefined role in the supply chain service which requires the 'participation' of specific cyber assets. Towards these lines, this component relies on the output of the Asset Modelling component since it allows to map assets that are already defined in the asset cartography of each organization with the processes that these assets are involved. This 'mapping' will play a significant role during the calculation of risks.

The **Simulation & Game Theory** component has a twofold goal. On the one hand it is responsible for the discovery of attack paths given a specific asset cartography and a specific supply chain service and on the other hand it is responsible to propose the best defensive strategy regarding the protection of a specific asset based on game theoretical principles. Both of these features provide significant added value to the final solution.

The **Collaborative Risk Assessment** component is responsible to guide the security analyst in order to perform the appropriate steps that are required for the conduction of a risk assessment for a specific supply chain service. More specifically, MITIGATE introduced a detailed multi-step processes, in the frame of D2.2 [1], in order to calculate SCS risks. These steps have to be executed in a guided way in order to stay in-line with the defined methodology. This component offers all supportive features that are required for an error-free execution of the methodology.

The **Open Intelligence and Big-Data Analytics** component is responsible to provide near real-time notifications regarding potential vulnerabilities that are related to the assets that exist in the asset cartography of one organization. These notifications will be generated based on the text-processing of open sources. However, such mining techniques are extremely computational intensive; thus the component will rely on a big-data framework (SPARK[3]) in order to achieve linear scalability.

The **Notification and Reporting** component is responsible to provide push notifications to the security analyst regarding any type of messages that are published in the pub/sub queue. Since MITIGATE involves many time-consuming operations (e.g. the conduction of a vulnerability assessment, the calculation of risks, the processing of open information sources) every time that such an operation is completed a specific message is placed in a predefined topic of the pub/sub queue. The specific component consumes all messages that relate to notification topics and presents them in a structured way to the user.

The **Administration** component is responsible for the management and the consistency of the various 'enumerations' that are required by all the other components. Such enumerations include mainly vulnerabilities, attack-types and business partners. This component also implements the semi-automated update of these enumerations from open sources.

The **Access Control and Privacy** component provides security guarantees in a horizontal manner to all the other components. More specifically, since the information that is provided and processed

(e,g, asset cartography, attack paths, risk calculations etc) is extremely sensitive, the specific component undertakes the responsibility of implementing the appropriate authentication, authorization and encryption schemes that are required in order to protect MITIGATE services and data end-to-end.

Finally, it should be noted that the architecture is complemented by a persistency layer and a pub/sub system. These components are totally supportive; hence they are not analysed in the following chapters. However, it should be clarified that the persistency layer consists of two types of databases; one relational (Mysql[4]) and one NoSQL(MongoDB[5]). The relational database is used in order to store fully structured data that change rarely (e.g. credentials, business partners) while the NoSQL is used in order to store semi-structured data that change frequently (e.g. Vulnerability reports). The pub/sub system (ActiveMQ[6]) is used in order to decouple the communication of the components and more specifically to eliminate any blocking communication that may be required. Elimination of blocking communication is a prerequisite for the creation of scalable system.

3 Asset Modelling & Visualization Component

3.1 Business Logic

As already mentioned, the Asset Modelling & Visualization component allows users to declare their assets along with the cyber relationship in order to create the so-called “Asset Cartography”. A valid asset cartography within the frame of an organization is the first step towards the realization of a collaborative risk assessment since each organization that participates in a supply chain service will use this component in order to create its own cartography. In addition, the cartography will be automatically linked to available vulnerabilities and attack-types that are relevant to the individual assets that are declared. To do so, the vulnerabilities and attack-types that are synchronized by open sources (e.g. CVEDetails[7]) will be directly usable from this component. Such information is managed by the Administration component as it will be examined below.

Furthermore, assets that are registered to a cartography participate in supply chain services. Therefore, the cartography per se along with the linked information (i.e. vulnerabilities, attack types and relevant SCSs) have to be intuitively visualized using a graph visualization modality. The specific component will offer such visualization functionality. Such graphical representation makes the information easier to present and comprehend, alleviating some of the analysis burden from the assessor’s point of view.

3.2 Functional Description

Functionality ID	C1.F1
Short Title	Creation of Asset Cartography
Description	Declaration of assets along their cyber-relationships and automatic linking of each asset with known vulnerabilities and attack-types (in our case attack-types and threats coincide)
Pre-Conditions	The vulnerabilities and attack types that are automatically linked to asset-types are synchronized in the MITIGATE persistency engine
Post-Conditions	The cartography is persisted to the NoSQL database using a common format

Functionality ID	C1.F2
Short Title	Visualization of SCS assets interdependencies
Description	Creation of a visual representation of the SCS asset interdependencies model. The SCS asset interdependencies model will be extracted in a suitable vector or bitmap image graphic file and forwarded to the MITIGATE interface to be presented to the user. The visualization shall comprise a suitably detailed view of the model, focusing on the participating cyber assets and their interrelations.
Pre-Conditions	Creation of the SCS asset interdependencies model in the Supply Chain Service Modelling component

Post-Conditions	Export of bitmap or vector graphic

Functionality ID	C1.F3
Short Title	Visualization of attack paths
Description	Creation of a visual representation of the SCS attack paths. The discovered attack paths will be highlighted as an overlay of the SCS asset interdependencies model. This overlay can be extracted in a suitable vector or bitmap image graphic file and forwarded to the MITIGATE interface to be presented to the user. The visualization shall comprise a suitably detailed view of the SCS asset interdependencies model overlaid by the attack paths.
Pre-Conditions	Discovery of attack paths using the Simulation and Game Theory Component
Post-Conditions	Export of bitmap or vector graphic

3.3 Component Architecture

The high-level control flow related to the Asset Modelling and Visualization Component is presented on Figure 2.

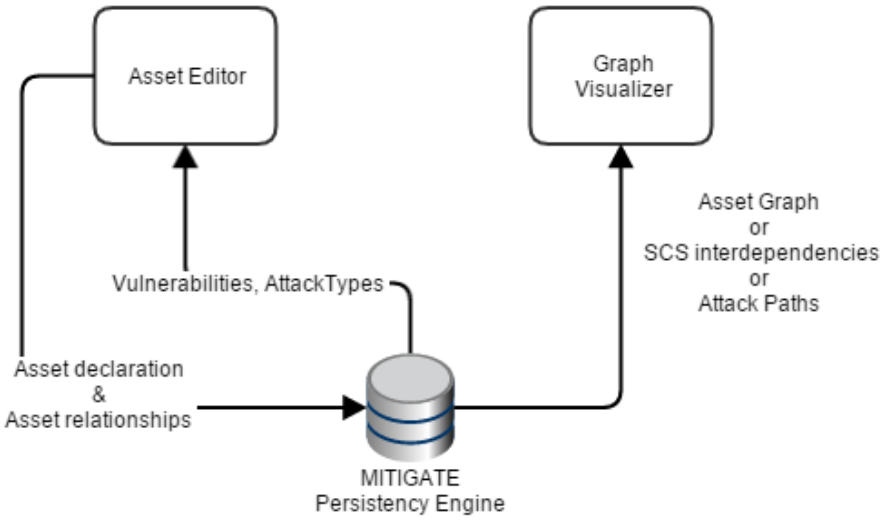

Figure 2 – High level control flow of Asset Modelling & Visualization component

As it is depicted, there are two major subcomponents, namely the asset editor and the graph visualizer. The first subcomponent is responsible for the proper declaration of an asset in the persistency engine. 'Proper' refers to the fact that many of the fields that have to be filled out will be autocompleted based on the existing registered enumeration in the system. E.g. the registration of a new mailserver should automatically prompt the user to select one of the available ones based on the vendorname. Furthermore, upon selection of the vendor a specific version should be selected. However, based on the selected version and the vulnerability records (that are replicated in the

persistency engine from open sources such as CVE details) the exact vulnerabilities that relate to the declared asset should automatically be inherited. This process should be seamlessly handled by the Asset Editor.

Moreover, the Graph Visualizer will be responsible to render all types of graphs that relate to the system. Such types include the asset cartography per se, the supply chain interdependency of assets and the attack paths. The two last models are generated by the Simulation and Game Theory component as it will be examined.

3.4 Interaction Between Components

3.4.1 Internal relationships

Component	Type	Description
MITIGATE Relational & NoSQL database	Bidirectional	Creation/modification of XML/JSON artefacts that represent a valid asset cartography & loading of SCS artefacts
Access Control & Privacy Component	Bidirectional	Consumption of Authentication & Authorization services
Simulation & Game Theory Component	GET	Acquisition of required data for the successful construction of attack paths

3.4.2 External relationships

N/A

4 Supply Chain Service Modelling Component

4.1 Business Logic

The Supply Chain Service Modelling component allows users to model the supply chain services that are performed by their organizations. Therefore, this component offers an authoring environment where a security analyst (a.k.a. assessor) can model various business processes that are performed collaboratively with several business partners. The most crucial functionality of this authoring environment is the ability to correlate the various assets that are captured in the frame of a cartography with specific business processes. This 'mapping' will play a significant role during the conduction of risk assessment.

4.2 Functional Description

Functionality ID	C2.F1
Short Title	Modelling of Supply Chain Services
Description	Allow an analyst to create and manage supply chain services which includes management of business processes and management of business partners that have a role in this processes
Pre-Conditions	N/A
Post-Conditions	Creation of SCS model and related Business Processes

Functionality ID	C2.F2
Short Title	Associate SCS with Cyber Assets
Description	Allow an analyst to associate specific business processes with existing assets that are declared using the Asset Modelling component
Pre-Conditions	Assets should already exist in a valid cartography
Post-Conditions	An enriched model of SCS is created which is used by the Simulation & Game Theory Component

4.3 Component Architecture

The control flow of the Supply Chain Service Modelling Component is presented on Figure 3.

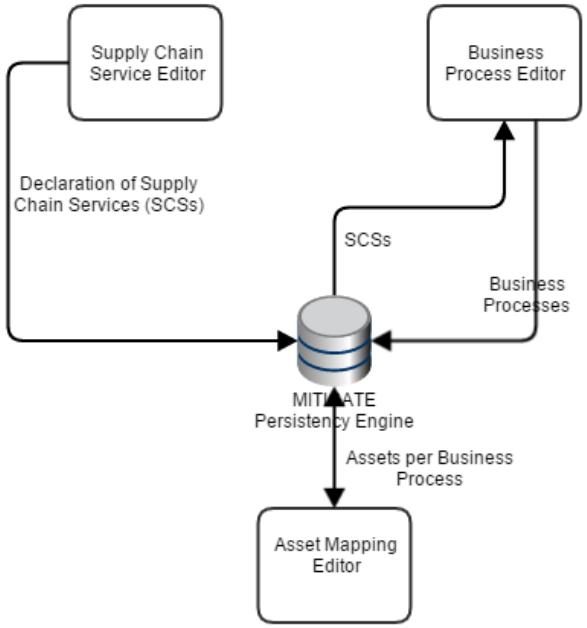

Figure 3 – Control flow of Supply Chain Service Modelling Component

As it is depicted on the figure, there are three major subcomponents related to this component; namely the SCS editor, the Business Process editor and the Asset Mapping Editor. The first editor is used to declare supply chain services and associate business partners that collaborate in the frame of a SCS. The business process editor is used in order to populate specific business processes that relate to a SCS and map the business partners to specific roles per business process. The third editor is used in order for each organization to declare which assets are used per each process based on the previous assignment.

4.4 Interaction Between Components

4.4.1 Internal relationships

Component	Type	Description
MITIGATE Relational & NoSQL database	Bidirectional	Creation/modification of XML/JSON artefacts that represent SCS artefacts
Access Control & Privacy Component	Bidirectional	Consumption of Authentication & Authorization services
MITIGATE Pub/Sub Queue	POST	Publish a message that new assets have been associated to a SCS

4.4.2 External relationships

N/A

5 Simulation & Game Theory Component

5.1 Business Logic

The Simulation and Game Theory component has a multidisciplinary role. Initially, it is used for the creation of the SCS asset interdependencies model. The model is a comprehensive graph representation of the SCS system, containing all the necessary information for the discovery and identification of the chain of sequential vulnerabilities on different assets that arise from consequential multi-steps attacks initiated from the Entry Points in order to exploit the vulnerabilities of the Target Points. The second main functionality has to do with the actual discovery and identification of the so-called vulnerability chains or paths. Path discovery is offered in two versions. One takes into account existing security controls that may mitigate discovered vulnerabilities, effectively identifying only unmitigated attack paths. The other assumes no security controls are in place and can be used for comparative analysis to evaluate the effectiveness and hence cost effectiveness of existing security controls. It should be noted that this component creates the graphs in a way that the visualization component that is already examined can render them. The third functionality is related to identification of the best defensive strategy regarding the protection of a specific asset based on game theoretical principles.

5.2 Functional Description

Functionality ID	C3.F1
Short Title	Modelling of SCS assets interdependencies
Description	Using the input provided by the assessor regarding the assets that support the SCS under examination and their dependencies the component will create a model of the SCS system, upon which risk analysis will run.
Pre-Conditions	Supply Chain Service's business processes identification Business partners' association SCS cyber assets identification Confirmed vulnerabilities identification Zero-day vulnerabilities identification
Post-Conditions	Creation of SCS asset interdependencies model

Functionality ID	C3.F2
Short Title	Discovery of attack paths (irrespectively of existing security controls)
Description	Based on the created SCS asset interdependencies model and the chosen attacker profile the component will find all attack paths from the designated entry to the target point(s), assuming no security controls are active in the system. The results can be used comparatively to evaluate the functional and economic effectiveness of the existing security controls.

Pre-Conditions	Creation of SCS asset interdependencies model Attacker profile identification Designation of entry and target point(s)
Post-Conditions	List of all attack paths in the SCS asset interdependencies model from the entry to the target point(s), assuming no existing security controls

Functionality ID	C3.F3
Short Title	Discovery of active attack paths
Description	Based on the created SCS asset interdependencies model and the chosen attacker profile the component will find all attack paths from the designated entry to the target point(s), not already mitigated by existing security controls. The results will be used from the risk analysis component to calculate the vulnerability and risk levels
Pre-Conditions	Creation of SCS asset interdependencies model Attacker profile identification Designation of entry and target point(s)
Post-Conditions	List of all attack paths in the SCS asset interdependencies model from the entry to the target point(s), not mitigated by existing security controls

Functionality ID	C3.F4
Short Title	Identification of Optimal Mitigation Actions and Worst Case Damage
Description	Computation of a Nash equilibrium for a number of potential attack strategies and a number of respective defence strategies. Therefore, a game between the attacker and the defender (admin of the infrastructure) is simulated for a given number of rounds. The aim is to minimize the expected damage. In this context, the algorithm is able to deal with uncertain payoffs (damage caused by a specific attack and defence action). The output is a list of defence strategies (e.g., security measures) together with the probability distribution indicating how they should be applied. Additionally, the worst case damage is described.
Pre-Conditions	Identification of attack strategies Identification of defence strategies Payoffs (description of damage for all combinations of attack and defence strategy)

Post-Conditions	Optimal defence strategy Worst case attack strategy Worst case payoff (damage)

5.3 Component Architecture

The Simulation and Game Theory component can be thought of as a generic component that consists of three subcomponents; the modelling subcomponent, the visualization subcomponent and the game-theory subcomponent. The first being responsible for the creation of the SCS asset dependencies model and the discovery of the possible attack paths, the second for the creation and export of visual representations of the model and the discovered paths. The flow between the first two subcomponents is visualized on Figure 4 while the control flow for the game theory subcomponent is presented on Figure 5.

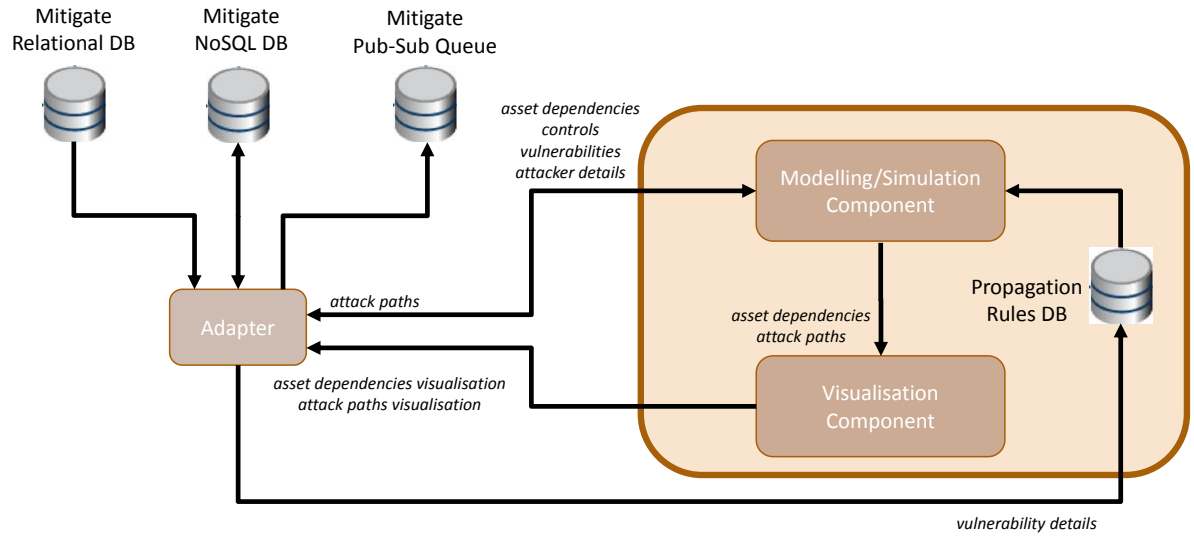

Figure 4 – Control flow for the Simulation and Visualization subcomponents

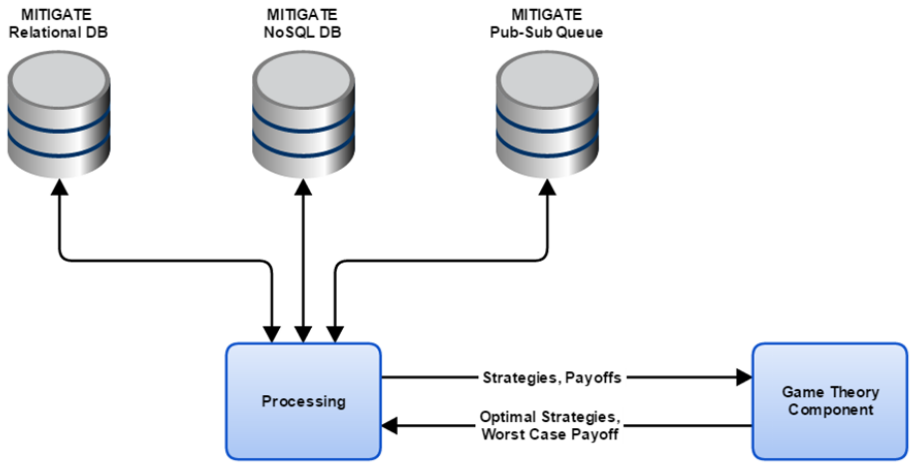

Figure 5 - Control flow for the Game Theory subcomponent

5.4 Interaction Between Components

5.4.1 Internal relationships

Component	Type	Description
MITIGATE Pub-Sub Queue	Bidirectional	Acquisition of queued tasks directed towards the modelling subcomponent and publication of results.
MITIGATE Relational and NoSQL DB	GET	Acquisition of required data for the successful execution of tasks such asset cartographies and SCSs
Access Control & Privacy Component	Bidirectional	Consumption of Authentication & Authorization services

5.4.2 External relationships

N/A

6 Collaborative Risk Assessment

6.1 Business Logic

The main goal of this component is to support the methodology that has to be followed in order to perform risk assessment. More specifically, as thoroughly analyzed in D2.2 [1] a multi-step process has been formulated in order perform risk calculation, impact analysis and risk mitigation. This process will be supported by MITIGATE through the seamless interoperation of the specific component with all the other components. The component is addressed as 'collaborative' because the calculations for one risk assessment rely on information that is provided by different organizations for the same SCS.

6.2 Functional Description

Functionality ID	C4.F1
Short Title	Definition of the scope of the Supply Chain Risk Assessment
Description	The initial step of the assessment, where the risk assessor selects the Supply Chain Service under consideration and defines the goal, the scope, and the outcome of the assessment.
Pre-Conditions	A valid set of asset-cartographies and SCSs that relate to these assets should be defined
Post-Conditions	N/A

Functionality ID	C4.F2
Short Title	Vulnerability Analysis
Description	Confirmed and zero-day Vulnerabilities that exist in the cyber assets of the SCS will be identified, based on data extracted from existing repositories of vulnerabilities, vulnerability scanners and from the list of the applied security controls identified in the Enhanced Business Partner Declaration. Each individual vulnerability is first assessed and attack paths that would allow an attacker to reach one or more target assets from one or more entry points/assets are discovered. An assessment is performed to calculate the cumulative vulnerability level of the attack paths. Furthermore, a complementary vulnerability measurement (i.e. the propagated vulnerability level) is computed under the assumption that an attacker will successfully penetrate the SCS to a predefined depth starting from one or more pre-selected entry points/assets
Pre-Conditions	A valid set of asset-cartographies and SCSs that relate to these assets should be defined
Post-Conditions	All types of vulnerabilities are calculated

Functionality ID	C4.F3
Short Title	Impact Analysis

Description	First the impact of the successful exploitation of each confirmed and zero-day vulnerability is estimated for individual assets and then the cumulative impact and propagated impact are computed
Pre-Conditions	A valid set of asset-cartographies and SCSs that relate to these assets should be defined
Post-Conditions	The impact analysis is presented to the security analyst

Functionality ID	C4.F4
Short Title	Risk Calculation
Description	Risk estimates are computed for individual assets and the commutative risk and propagated risk are assessed
Pre-Conditions	A valid set of asset-cartographies and SCSs that relate to these assets should be defined
Post-Conditions	The risk calculations are presented to the security analyst

Functionality ID	C4.F5
Short Title	Risk Mitigation
Description	The risk assessment values are compared against specific criteria (set and agreed by all business partners), in order to select the optimum security controls; following a game theoretic approach
Pre-Conditions	A valid set of asset-cartographies and SCSs that relate to these assets should be defined The criteria should be pre-populated
Post-Conditions	The optimal controls are presented to the security analyst

6.3 Component Architecture

The control flow regarding the Collaborative Risk Assessment component is presented on Figure 6. As it is depicted, the specific component consists of three subcomponents; namely the Risk Assessment Configuration the Risk Calculation and the Reporting subcomponent. The first subcomponent is responsible to initialize a risk assessment by supporting the already defined methodological steps. To do so, the subcomponent has to interact with the persistency engine in order to load SCS models and relevant asset cartographies. The second subcomponent performs the actual calculations which require interaction with the Simulation and Game Theory component in order to resolve relevant attach paths. Since this is a time consuming process, the security analyst is notified for the results in an asynchronous way through the usage of a predefined topic in the pub/sub component.

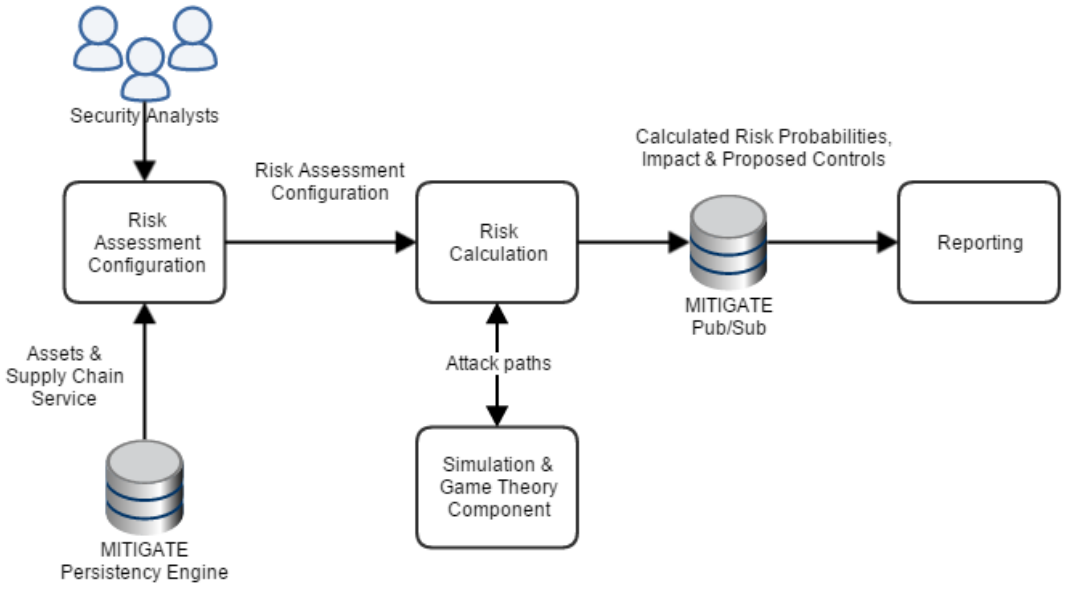

Figure 6 - Control Flow regarding Collaborative Risk Assessment Component

6.4 Interaction Between Components

6.4.1 Internal relationships

Component	Type	Description
MITIGATE Relational and NoSQL Database	Bidirectional	The initialization of a Risk Assessment requires access to the database of asset cartographies, SCSs and vulnerabilities
MITIGATE Pub-Sub Queue	Bidirectional	Since the risk calculation is a time consuming process the pub/sub is notified upon the finalization of a calculation
Simulation & Game Theory Component	GET	The calculation of SCS assets' interdependencies and the attack paths are an essential prerequisite of the risk assessment
Access Control & Privacy Component	Bidirectional	Consumption of Authentication & Authorization services

6.4.2 External relationships

N/A

7 Open Intelligence & Big Data Analytics Component

7.1 Business Logic

The main purpose of the Open Intelligence & Big Data Analytics component can be summarized as follows:

- Retrieves and indexes data from external sources like social media (e.g. Twitter) or RSS-Feeds
- Extracts necessary information to infer information related to risks and possible actions that have to be taken
- Analyzes & processes the indexed data based on the SCS and the asset cartography models
- Performs data classification which is propagated to other MITIGATE components

The algorithms that have to be executed in order to achieve the aforementioned functionality are computational intensive; hence the analysis has to be supported by a computation cluster. On top of these services, the specific component integrates some predictive analysis algorithms that can be used to predict some valuable trends related to cyber-security.

7.2 Functional Description

Functionality ID	C5.F1
Short Title	Information Sources Management
Description	Users must be able to configure data sources by defining required parameters via web interface.
Pre-Conditions	N/A
Post-Conditions	Data sources and specific polling mechanisms are automatically configured

Functionality ID	C5.F2
Short Title	External Data Extraction
Description	The Open Intelligence component requires a connection to the external sources. Therefore, connections to different sources will be implemented. The sources can be separated in three different groups. Connections to the following types are required: • Social media • RSS-Feeds • CVE repositories
Pre-Conditions	Data sources are configured.
Post-Conditions	Data are stored in the NoSQL repository

Functionality ID	C5.F3
Short Title	Indexing of data
Description	Upon reception the data are indexed in order to be efficiently process able.
Pre-Conditions	Pre-Conditions are: • Data retrieval successful • Defined structure or keywords for Data Processing • Parameters describing the meta-information need to be available (e.g. length of necessary content)
Post-Conditions	Classification of information is required to increase the usability of data value.

Functionality ID	C5.F4
Short Title	Classification Of Extracted Data
Description	Indexed Data are classified using text-mining algorithms. An example for text processing could be to determine if a content contains zero-day vulnerability.
Pre-Conditions	The following pre-conditions are needed: • Data processing successful • Description of key words to identify risks and vulnerability within the data • Unified classification of risks to order incidents in categories
Post-Conditions	A structured set of information is pushed to the pub/sub

Functionality ID	C5.F5
Short Title	MITIGATE Connector
Description	The MITIGATE connector enables the connectivity of other components to the Open Intelligence component. It will be able to transmit results of analysis to defined recipients.
Pre-Conditions	The Classification of data has to be completed.
Post-Conditions	A specific topic in the pub/sub is notified for a new item that has to be processed by the security analyst

7.3 Component Architecture

The Open Intelligence component uses extensive resources to store and analyze the data. Hence, the infrastructure to support it must satisfy the following requirements:

- *Scalability*: Big Data infrastructure must be horizontal scalable, in order to parallelize data computation and ease the upgrade of the infrastructure by just adding new worker servers.
- *Data replication*: multiple copies of the same data must exists, in order to achieve both resilience and increased availability.
- *Locality-aware computation*: data should be processed on the same node where it resides every time is possible, in order to minimize network bandwidth usage.
- 3rd-party software must be *open source.*

In order to fulfill such requirements, components will be deployed in a cluster environment (see Figure 7). The overall infrastructure can be analyzed identifying two operational layers:

- *Manager layer*: applications in this layer have the task of managing the coordination of jobs and the fair assignment of resources. Manager components must be aware of the network topology and guarantee data locality.
- *Worker layer*: applications in this layer have the task of storing and computing data. These components are built to support heavy workloads and achieve good performance.

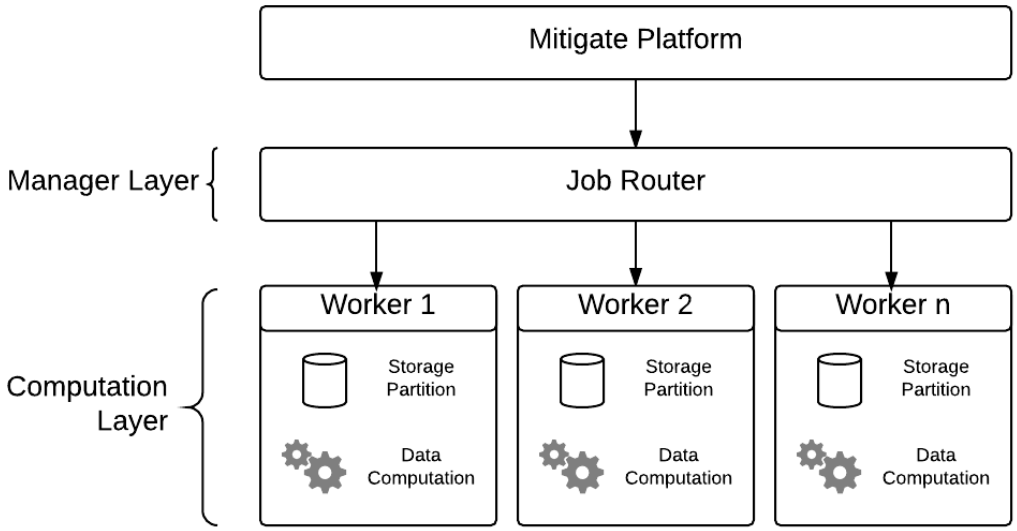

Figure 7 - MITIGATE Cluster that will support the Open Intelligence component

The Open Intelligence component consists of several subcomponents (see Figure 8), which can be described as followed:

Information Source Management. It contains the user interface the external data manager settings. Correctly authenticated users will be able to set various parameters, like scanning start time and frequency, enable or disable sources, and source specific configurations.

External Data Manager. It enables connections to external data sources. The sources can be of different categories, such as Social Media. The groups can also enable different data sources. The qualified data will be received form source and delivered to a subcomponent called Text-Processing & Classification. This subcomponent contains five different packages, one for each type of information source, currently:

- CVE Details
- NIST NVD
- Twitter

- Reddit
- Generic RSS feeds

Every component will fetch data from its own source, then normalize it and check existing information through their own specific id (e.g.: CVE id, Twitter post id) when possible to avoid duplication. Then it will store new, non-duplicate information in the correct repository: untrusted sources will be available for subsequent analysis, while trusted source will be stored directly as vulnerabilities.

Text Mining. This subcomponent consists of two modules and realizes the Text Processing and Classification. First, the data will be processed and analyzed by the Text Processing package. The analysis is focused on parameters and other analytics-requirements. After analyzing the data, the Text Classification will take place.

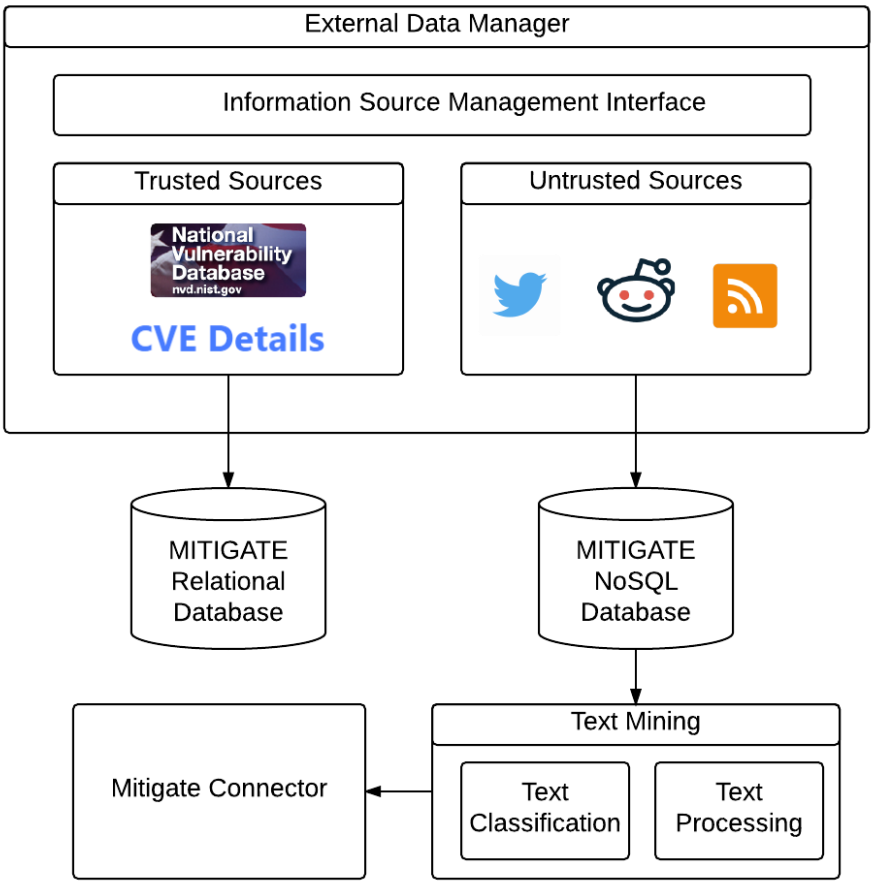

Figure 8 - Open Intelligence subcomponents

7.4 Interaction Between Components

7.4.1 Internal relationships

The Open Intelligence component will interact with the following internal components:

Component	Type	Description
MITIGATE NoSQL Database	Bidirectional	Storing/receiving the results

MITIGATE Relational Database	Bidirectional	Automated creation/update of vulnerabilities from CVE repositories
MITIGATE Pub/Sub component	PUSH	Text analysis component will notify MITIGATE operators of possible vulnerabilities that have been found in social media channels
Access Control & Privacy Component	Bidirectional	Consumption of Authentication & Authorization services

7.4.2 External relationships

The Open Intelligence component will interact with the following external sources:

Component	Type	Description
Social Media	GET	Retrieving information from social media (e.g. Twitter)
RSS-Feed	GET	Retrieving information from RSS-Feeds
Open Repositories	GET	Retrieving information from repositories (e.g. CVEDetails repository)

8 Notification & Reporting Component

8.1 Business Logic

The Notification and Reporting component offers horizontal services to all the other components. Its purpose is to provide a single-point of interaction for the users of the MITIGATE system. A core architectural decision that has been taken prior to the componentization is that all time-consuming interactions among components will be asynchronous. Therefore, the pub/sub component will be used, yet this raises some issues regarding the user experience. More specifically, the output of the computational intensive and time-consuming operations has to be promptly viewable to the system users since they may want to react on the output. For example, if the Open Intelligence component identifies a new zero-day vulnerability that is exploited widely a risk assessment has to be re-executed in order to evaluate potential risks or enforce mitigation actions. The component will offer both reporting and notification features.

8.2 Functional Description

Functionality ID	C6.F1
Short Title	Definition of a new log-type
Description	The component will be extensible enough so as new log-types will be on-boarded dynamically. Each new log-type has to be mapped to a specific topic of the pub/sub component since the Notification and Reporting component will subscribe automatically to this topic.
Pre-Conditions	Log entries are added in the pub/sub based on specific pre-defined topics.
Post-Conditions	N/A

Functionality ID	C6.F2
Short Title	Configuration of a notification handler
Description	Since components rely on the pub/sub system to report their logs in an asynchronous manner, users will be able to define synchronous ‘callbacks’ for specific log-entries. These callbacks pre-assume the existence of notification handlers such as email and SMS that will undertake the task of synchronous delivery.
Pre-Conditions	The notification handlers (email, SMS) expose an API that is compatible with the platform.
Post-Conditions	N/A

Functionality ID	C6.F3
Short Title	Navigation to ‘faceted’ logs & Visualization

Description	The MITIGATE user will have the capability to navigate among historical logs and search for specific log-entries. The logs will be pre-indexed for efficient search. Furthermore, during indexing each pub/sub topic will be considered as a different facet in order to allow more intuitive search. Using these advanced search functionalities many reports can be produced.
Pre-Conditions	Log entries are added in the pub/sub based on specific pre-defined topics.
Post-Conditions	N/A

8.3 Component Architecture

A more fine-grained architecture of the Notification and Reporting component is presented on Figure 9. As it is depicted the main subcomponents are

- the topic subscriber, that controls the information flow
- the indexer which stores the logs in a specific format so as to be efficiently retrievable
- the faceted search and reporting subcomponents that allows intuitive navigation on the logs
- the notification handler that performs synchronous communication which will be configured by the users.

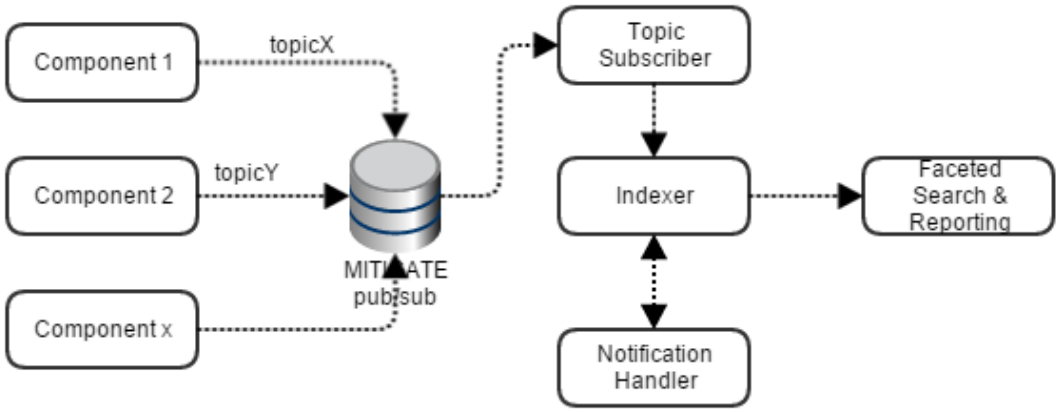

Figure 9 - Fine-grained architecture of the Notification and Reporting component

8.4 Interaction Between Components

8.4.1 Internal relationships

Component	Type	Description
MITIGATE Pub-Sub Queue	Bidirectional	All log entries are submitted to the pub/sub and are consumed by the Notification and Reporting component

Access Control & Privacy Component	Bidirectional	Consumption of Authentication & Authorization services

8.4.2 External relationships

Component	Type	Description
Email Server, SMS Gateway, etc	POST	Any system that can be used for synchronous PUSH of information can be used

9 Administration Component

9.1 Business Logic

The Administration component is responsible for the management and the consistency of the various 'enumerations' that are required by all the other components. Such enumerations include mainly vulnerabilities, attack-types and business partners. This component also implements the semi-automated update of these enumerations from open sources. Based on the MVC pattern, this component enables the manipulation of data through specific SCRUD (Search, Create, Read, Update, Delete) interfaces.

9.2 Functional Description

Functionality ID	C7.F1
Short Title	SCRUD Entity management
Description	The Administration component will offer SCRUD (Search, Create, Read, Update, Delete) functionalities for common entities of the system, such as Business Partners, Vulnerabilities, Attack types
Pre-Conditions	Prior to each delete a logical consistency check will be performed in order to verify that a deleted item is not participating in a SCS or an asset cartography
Post-Conditions	Entity is created/read/updated/deleted.

Functionality ID	C7.F2
Short Title	SCRUD Entity management
Description	The Administration component will offer SCRUD (Search, Create, Read, Update, Delete) functionalities for common entities of the system, such as Business Partners, Vulnerabilities, Attack types
Pre-Conditions	Prior to each delete a logical consistency check will be performed in order to verify that a deleted item is not participating in a SCS or an asset cartography
Post-Conditions	Entity is created/read/updated/deleted.

9.3 Component Architecture

The subcomponents of the Administration component are presented on Figure 9. As it is depicted, per each entity that a SCRUD API is required a specific Data Access Object (a.k.a. DAO) handler has to be populated. However, this handler never interacts with the raw databases since as already mentioned MITIGATE relies on multiple database engines. Instead each handler will interact with a subcomponent that is called Transactional Manager which will be responsible to reassure the consistency and the integrity of objects that are managed. Assurance of referential integrity between the different engines is the most crucial functionality of this component.

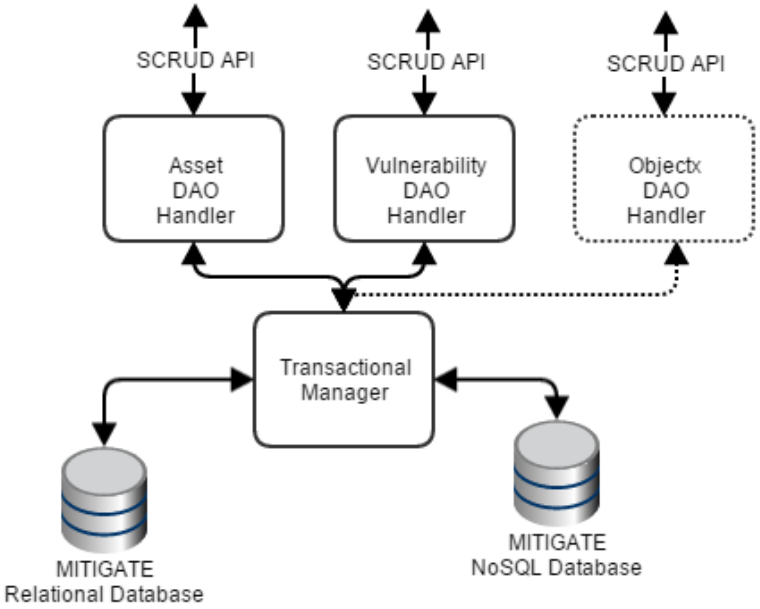

Figure 10 - Subcomponents of Administration

9.4 Interaction Between Components

9.4.1 Internal relationships

The Administration component will interact with the following internal components.

Component	Type	Description
MITIGATE Relational and NoSQL Database	Bidirectional	Read/create/update/delete of specific entities from relational database
MITIGATE Open Intelligence Component	Bidirectional	Backend operators will receive notifications about possible vulnerabilities found by Big Data Text Mining component
Access Control & Privacy Component	Bidirectional	Consumption of Authentication & Authorization services

9.4.2 External relationships

There are no relationships with external components.

10 Access Control & Privacy Component

10.1 Business Logic

The Access Control & Privacy component undertakes the following:

- The build of the required authentication logic of the MITIGATE platform and its privacy services. It is the main component that undertakes the proper satisfaction of all aspects on online authentication procedure.
- The secure and timely management of on-boarding (provisioning) and off-boarding (de-provisioning) of users and the proper extension of those processes to all involved MITIGATE services.
- The proper management of users credentials and MITIGATE applications
- Support of Roles, User Groups and assign permission privileges based on the MITIGATE access & privacy policy

It is the core component of the MITIGATE platform since it supports controlled release of sensitive data and services, ensuring that all end users' requests are validated before releasing data to them. Upon receipt of an end-user request this component attempts to authenticate the requesting entity and upon successful authentication it implements authorization mechanisms in order to validate the request and provide specific access permissions to individual MITIGATE services and data. If the validation or authorization fails, the end-user request is denied. In addition, this component is responsible of implementing the proper mechanisms to protect data confidentiality, integrity and non-repudiation.

The Access Control & Privacy Component allows MITIGATE platform to perform the following:

- Enable users to use their credentials and gain direct access into a MITIGATE service and/or data
- Enable account creation with the provision of a unique email address, that will be used in order to activate the account
- Enable forgotten password functionality
- Enforce a strict Password Policy, enhancing the security of the implemented MITIGATE services
- Set privacy options for each data field inserted and managed from the system itself
- Audit and capture information about events, such as the user actions
- Records audit events for page views
- Securely manage user-sessions within the MITIGATE system, integrating strong Session Management mechanisms
- Protect user sessions against unauthorized high-jacking by other users
- Ensure that sensitive data will not be transmitted by a web application as URL parameters (exception: session identifiers)
- Enforce strong cryptographic mechanisms. Specifically, use of SSL v3[8] / TLS 1.x[9] with server authentication and encryption (with a minimum key length of 128 bits)
- Ensure for any implemented Web Service in the MITIGATE system that all requests will be protected from manipulation/replay attacks while on the public, unprotected communication path
- Authenticate the recipient of a web service request by the sender of this request

- Ensure for integrity and confidentiality insurance reasons that SOAP[10]/REST[11] requests or replies that are not handled immediately and need to be buffered in the DMZ will keep their message integrity and confidentiality.

10.2 Functional Description

Functionality ID	C8.F1
Short Title	Authenticate MITIGATE users
Description	The Access Control & Privacy Component attempts to authenticate the requesting entity validating the credentials chosen from the end-user.
Pre-Conditions	The user must have been successfully registered on the MITIGATE platform
Post-Conditions	Upon Successful authentication the user will be properly authorized on the system

Functionality ID	C8.F2
Short Title	Authorize MITIGATE users
Description	The Access Control & Privacy Component attempts to authorize the requesting entity applying a given role, based on the group in which the user belongs and following the Role-based policy supported from the MITIGATE platform
Pre-Conditions	The user must have been successfully authenticated within the MITIGATE platform
Post-Conditions	Upon successful authorization a unique session identifier will be generated and sent to the user

Functionality ID	C8.F3
Short Title	Session Generation
Description	The Access Control & Privacy Component attempts to generate a unique session identifier. The session cookie(s) will be used to transmit the session identifier to the user
Pre-Conditions	The user must have been successfully authenticated & authorized within the MITIGATE platform
Post-Conditions	N/A

Functionality ID	C8.F4
Short Title	Session Validation

Description	The Access Control & Privacy Component attempts to verify the session identifier provided to the user, at every attempt of the last mentioned to execute an application logic function
Pre-Conditions	A unique session identifier must have been successfully generated and provided to the user
Post-Conditions	Upon successful validation of the unique session identifier, the MITIGATE system will allow or deny access based on the access privileges of the user's role

Functionality ID	C8.F5
Short Title	Enforce Web Services Security Mechanisms
Description	Each element and attribute value communicated in a web service request will be described with the expected data type, the expected length and/or range, and whenever appropriate a formal specification describing the acceptable data. All web services that does not contain a formal specification of input data, black/white listings will be used to stop illegal characters from being passed to the webserver. In case that no such black / white listing will be made, the web services themselves will be hardened against possible injection attacks. Last but not least, definitions of Web Services will not use "unbounded" attributes.
Pre-Conditions	N/A
Post-Conditions	N/A

Functionality ID	C8.F6
Short Title	Enforce Encryption and protection mechanisms on data
Description	Depending on the type of data (e.g. a file, JSON object, raw string, etc.) and its annotations, the Access Control & Privacy Component will determine the appropriate mechanisms to be used in order to encrypt the data. Multiple mechanisms for protection of various data types will be implemented. The choice how to protect each item stored in the database is made at run-time, by matching a protection mechanism specific to the item's data type and privacy level
Pre-Conditions	N/A
Post-Conditions	Data will be securely stored on the MITIGATE back-end systems.

10.3 Component Architecture

The Access Control & Privacy component consists of several related subcomponents as depicted on Figure 10.

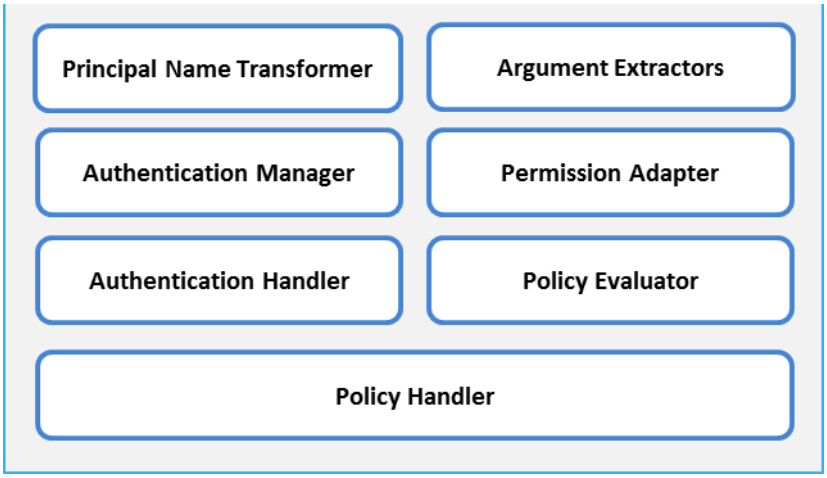

Figure 11 – Access Control and Privacy subcomponents

- The ***Principal Name Transformer*** subcomponent, which undertakes the required transformation of the user id string, typed into the login form, into a tentative Principal Name to be validated by a specific type of Authentication Handler
- The ***Authentication Manager***, which acts as the entry point into the Access Control & Privacy component. This manager accepts one or more credentials and delegates authentication to the Authentication Handler.
- The ***Argument Extractors***, which are responsible to examine the http request for parameters that describe the authentication request such as the requesting service, etc. Extractors should support a number of supported authentication protocols and each shall create appropriate instances of the Web application service that contains the results of the extraction.
- The ***Authentication Handler***, which undertakes the authentication of a single credential and reports one of three possible results:
  1) Success
  2) Failure
  3) Not Attempted

  The Authentication Handler may support many common kinds of authentication systems such as common Database, LDAP technologies[12], JAAS[13], Oauth 1.0/2.0[14], OpenID[15], X.509[16] client SSL certificate, etc.
- The ***Permission Adapter***, which is the proper subcomponent for authorization in the core MITIGATE system. It evaluates the policy from a particular data source and delivers a Permission Collection that contains a set of permissions granted. The Permission Adapter subcomponent implements the following:
  - The mechanisms that initialize the adapter with all required relevant parameters
  - The mechanisms that is called whenever the Permissions with particular Principals is requested
  - The access-token generator, which acts as a machine-readable response that describes the data items approved/rejected and the exact terms on which access is given
- ***Policy Evaluator,*** the functionality of which includes granular evaluation of privileges, down to specific data fields and services, against the MITIGATE system Policy.

- ***Policy Handler*****,** which optimizes storage by ensuring that any binary is only stored once on the file system. Rather than storing the file in its original name under a specific path, it creates a checksum of the file (MD5[17] and SHA1[18]) and renames it to its checksum. All the metadata about a file is then stored in MITIGATE system repository. Using checksum based storage, any operation done on an artefact (copy, move, delete) is actually implemented by changing the metadata stored in the MITIGATE repository. Moreover, Policy Handler integrates the proper mechanisms used for managing encryption keys effectively and integrating encryption. Specifically, it undertake the secure storage, handling and deletion of encryption keys as well as the monitoring of the whole encryption procedure.

10.4 Interaction Between Components

10.4.1 Internal relationships

The following table depicts the internal MITIGATE relationships of the Access Control & Privacy Component with other components:

Component	Type	Description
MITIGATE Relational and NoSQL Database	Bidirectional	Provides transparent encryption/decryption functionalities to sensitive data
Asset Modelling and Visualization Component	Bidirectional	Provide Authentication & Authorization services
Supply Chain Service Modelling Component	Bidirectional	Provide Authentication & Authorization services
Simulation and Game Theory Component	Bidirectional	Provide Authentication & Authorization services
Collaborative Risk Assessment Component	Bidirectional	Provide Authentication & Authorization services
Open Intelligence and Big Data Analytics Component	Bidirectional	Provide Authentication & Authorization services
Administration Component	Bidirectional	Provide Authentication & Authorization services, as well as encryption services to specific data stored in the database
Notification and Reporting Component	Bidirectional	Provide Authentication & Authorization services

10.4.2 External relationships

The relationships of the Access Control & Privacy Component established with external to MITIGATE platform components or systems have as follows:

Component	Type	Description

End-User client	Bidirectional	Provide Authentication & Authorization (session cookies) and cryptographic services (i.e. SSL)

11 Conclusions

The purpose of the current deliverable was to provide details regarding the architecture of the MITIGATE risk management system. MITIGATE envisages to deliver a unified environment where sophisticated risk management features will be offered taking under consideration port supply chain services that are executed among organizations. Such features include collaborative risk assessment, visualization, simulation and open intelligence analytics. All these features rely on discrete components that implement the aforementioned functionality. As already discussed, the MITIGATE high level architecture consists of eight components namely the Asset Modelling & Visualization component, the Supply Chain Service Modelling component, the Simulation & Game Theory component, the Collaborative Risk Assessment component, the Open Intelligence and Big-Data Analytics component, the Notification and Reporting component, the Administration component and finally the Access Control and Privacy component. These components complement each other in order to achieve the endmost goals of the project.

One of the critical goals of the deliverable was to identify the components' interdependencies and interactions. Based on the fine-grained componentization and the identified interactions, current deliverable will provide an essential input to WP3 where normative models will be defined.

References

[1] Mitigate Deliverable 2.2 - Evidence-Driven Maritime Supply Chain Risk Assessment Approach
[2] Mitigate Deliverable 3.1 (to appear 6/2016) – Specification on the Risk Assessment, Mitigation and Simulation Functionalities
[3] SPARK Big Data Platform - Available Online: http://spark.apache.org/
[4] Mysql Relational Database- Available Online: https://www.mysql.com/
[5] MongoDB NoSQL Database- Available Online: https://www.mongodb.org/
[6] ActiveMQ Pub/Sub system - Available Online: http://activemq.apache.org/
[7] CVEDetails - Available Online: http://www.cvedetails.com/
[8] SSL v3 Protocol - Available Online: https://tools.ietf.org/html/rfc6101/
[9] TLS 1.x Protocol- Available Online: https://tools.ietf.org/html/rfc5289/
[10] SOAP Specification - Available Online: https://www.w3.org/TR/soap/
[11] Pautasso, Cesare; Zimmermann, Olaf; Leymann, Frank (April 2008), "RESTful Web Services vs. Big Web Services: Making the Right Architectural Decision", 17th International World Wide Web Conference (WWW2008) (Beijing, China)
[12] LDAP Specification - Available Online: https://tools.ietf.org/rfc/rfc4511.txt
[13] Java Authentication & Authorization: https://docs.oracle.com/javase/8/docs/technotes/guides/security/jaas/JAASRefGuide.html
[14] Oauth Protocol - Available Online: http://oauth.net/2/
[15] OpenID - Available Online: http://openid.net/
[16] X.509 Specification - Available Online: https://tools.ietf.org/html/rfc5280/
[17] MD5 Algorithm - Available Online: https://tools.ietf.org/html/rfc1321/
[18] SHA1 Algorithm - Available Online: https://tools.ietf.org/html/rfc3174/