SOCCRATES

SOC & CSIRT Response to Attacks & Threats based on attack defense graphs Evaluation Systems

D6.2

Initial version of the SOCCRATES platform

Deliverable type:	Demonstrator
Contributing work packages:	WP6 Prototypes, Integration and Testing
Due date of deliverable:	30/11/2021
Submission date:	30/11/2021
Dissemination level:	PU

Responsible organisation:	ATOS
Editor:	Jesús Villalobos Nieto, Susana González Zarzosa
Revision:	1.0

Abstract	This deliverable describes the initial version of the SOCCRATES Platform prototype with the integration of the different components through the SOCCRATES Orchestrator and Integration Engine.
Keywords:	Security Automation, Orchestrator & Integration Engine, Integration, Use Case

Management summary

This deliverable provides a description of the initial version of the SOCCRATES platform that is mainly the result of Task 6.2, focused on the integration of the different components developed in WP3, WP4 and WP5 with the SOCCRATES Orchestrator & Integration Engine and web frontend developed in T6.1.

In D6.1 a first prototype was described that was an initial proof of concept of the mechanisms included in the SOCCRATES platform to provide orchestration and reconfiguration capabilities of the infrastructure through the usage of the Cortex framework and the BPMN engine Activiti. This deliverable D6.2 describes a second prototype with the initial version of the SOCCRATES platform that will be used for testing of the components integration and evaluation of the two pilots included in the project.

This deliverable describes in detail the implementation and deployment of the Orchestrator & Integration Engine component, which is the core of the SOCCRATES platform. And also how each of the other SOCCRATES components are integrated through the implementation of different Cortex analyzers and/or responders.

Additionally, the SOCCRATES workflows to cover the five SOCCRATES use cases have been reviewed and updated according to the swim lane diagrams described in D2.3. The use cases are described in detail with the SOCCRATES components and the Cortex analyzers/responders involved in each of them and also how they are triggered. The updated use case 5 (the response on deployment of a new system in the infrastructure) is included in the demonstration of the execution, to illustrate with an example how the integration of the SOCCRATES platform is done.

Finally, this deliverable includes the lessons learned during the integration process of this initial version of the SOCCRATES platform together with the plan to fulfill the requirements for the final version of the SOCCRATES platform, that will be reported in the next D6.4.

Contributor(s)	Ruben Trapero (ATOS) Jesús Villalobos Nieto (ATOS) Susana González Zarzosa (ATOS) Martin Eian (MNM) Agron Bajraktari (AIT) Paul Smith (AIT) Christophe Kiennert (IMT) Maciej Kosz (VTF) Ashish Kumar Dwivedi (KTH) Mathias Ekstedt (KTH) Erick Ringdahl (FRS)

Reviewer(s)	Frank Fransen (TNO)

Security Assessment	**See deliverables table for security assessment requirements**
Approval Date	25/11/2021
Remarks	Minor comment.

TABLE OF CONTENTS

1 INTRODUCTION – RATIONALE OF THIS DOCUMENT 7

1.1 The SOCCRATES project 7
1.2 This deliverable 8
1.3 Structure of this deliverable 8

2 THE INITIAL SOCCRATES PLATFORM 9

2.1 Overview 9
2.2 The SOCCRATES Orchestrator Engine 9
2.2.1 Orchestrator Core 9
2.2.2 Orchestrator Workflow Start Events 14
2.2.3 Logging system 16
2.3 The SOCCRATES Integration Engine 17
2.3.1 Business Impact Analyzer 19
2.3.2 Respone Planner 20
2.3.3 Attack Defense Graph 20
2.3.4 Course of Action 21
2.3.5 Automatic Reconfiguration 22
2.3.6 Advanced Attack Detection 22
2.4 The integration environment 23

3 SOCCRATES WORKFLOWS 28

3.1 Use Case 1: Response on Detected Ongoing Attack 28
3.2 Use Case 2: Response on Reception of New Cyber Threat Intelligence 31
3.3 Use Case 3: Response on Discovery of New Vulnerable Assets 32
3.4 Use Case 4: Response on Discovery of System Configuration Change 34
3.5 Use Case 5: Response on Deployment of New Systems in Infrastructure 34

4 SOCCRATES PLATFORM DEMONSTRATOR 36

4.1 Triggering workflow for UC5 36
4.2 Service task execution 38
4.3 Orchestrator to Cortex Analyzer / Responder 39
4.4 User task execution 41
4.5 Exclusive Gateways 44
4.6 Automatic Response 45

5 SUMMARY AND NEXT STEPS 46

6 REFERENCES 47

7 ABBREVIATIONS 48

8 ANNEX I : ANALYZERS & RESPONDERS REQUESTS AND RESPONSES 50

8.1.1 Business Impact Analyzer 50
8.1.2 Respone Planner 51
8.1.3 Attack Defense Graph 53
8.1.4 Course of Action 54

8.1.5 Automatic Reconfiguration 55
8.1.6 Advanced Attack Detection 56
9 ANNEX II : DATA SHARED BETWEEN TASKS IN UC5 WORKFLOW 57

TABLE OF FIGURES

Figure 1-1 – The SOCCRATES platform 7
Figure 2-1 - Activiti process definition (workflow) for UCs 2 to 5 10
Figure 2-2 - Activiti workflow editor - Setting the invoked Java Class 10
Figure 2-3 - SOCCRATES Orchestrator Core - Java Project Structure 11
Figure 2-4 - Service task example 12
Figure 2-5 - Orchestrator Core execution for debugging/testing 13
Figure 2-6 - Orchestrator Core terminal output 13
Figure 2-7 - Orchestrator Core input throught terminal 14
Figure 2-8 - Message definitions 14
Figure 2-9 - Message deifnitions editor 15
Figure 2-10 - Start message event association with message reference 15
Figure 2-11 - SOCCRATES Orchestrator Database Log sample 16
Figure 2-12 - Process definition editor - Task id 17
Figure 2-13 - SOCCRATES Orchestrator and Integration Engine environment diagram 23
Figure 2-14 - SOCCRATES Integration Engine docker compose file 25
Figure 2-15 - FTP Folder to deploy Cortex Analyzers 26
Figure 2-16 - Enabling and configuring Cortex Analyzers 26
Figure 2-17 - Testing the Analyzers with Cortex Dashboard 27
Figure 2-18 - Testing the Analyzers using Cortex API 27
Figure 2-19 - Example of report obtained from a Cortex Analyzer 27
Figure 3-1 - UC1 workflow - Analysis Stage 28
Figure 3-2 - UC1 workflow - Preliminary Response phase 29
Figure 3-3 - UC1 workflow - Assess full extent 29
Figure 3-4 - UC1 workflow - Final containment phase 30
Figure 3-5 - UC1 input triggering message 30
Figure 3-6 - UC2-5 Workflow 31
Figure 3-7 - UC2 input triggering message 32
Figure 3-8 - UC3 input triggering message 33
Figure 3-9 - UC4 input triggering message 34
Figure 3-10 - UC5 input triggering message 34
Figure 4-1 - Request to Activiti API for triggering UC5 36
Figure 4-2 - Activiti UI - Running processes 37
Figure 4-3 - UC5 process execution status diagram 38
Figure 4-4 - Service Task execution log 38
Figure 4-5 - Cortex UI - Jobs History 39
Figure 4-6 - Cortex UI - Responder Job Report 40
Figure 4-7 - Use Case traces in logging table 40
Figure 4-8 - UC5 execution - Process Diagram 41
Figure 4-9 - User task form properties 42
Figure 4-10 - Retrieving form data using Activiti API 43
Figure 4-11 - Process diagram - Exclusive Gateway 44
Figure 4-12 - Exclusive gateway - Path flow condition 44
Figure 4-13 - Automatic Response - Email example 45

1 Introduction – Rationale of this document

This section introduces the SOCCRATES project and defines the goals of this deliverable.

1.1 The SOCCRATES project

SOCCRATES (SOC & CSIRT Response to Attacks & Threats based on attack defence graphs Evaluation Systems) is an EU funded project under the Horizon2020 programme that has the following main challenge:

How can SOC and CSIRT operations effectively improve their capability in detecting and managing response to complex cyber-attacks and emerging threats, in complex and continuously evolving ICT infrastructures while there is a shortage of qualified cybersecurity talent?

The main objective of SOCCRATES is to develop and implement a security automation and decision support platform (‘**the SOCCRATES platform’)** that will significantly improve an organisation’s capability (usually implemented by a SOC and/or CSIRT) to quickly and effectively detect and respond to new cyber threats and ongoing attacks.

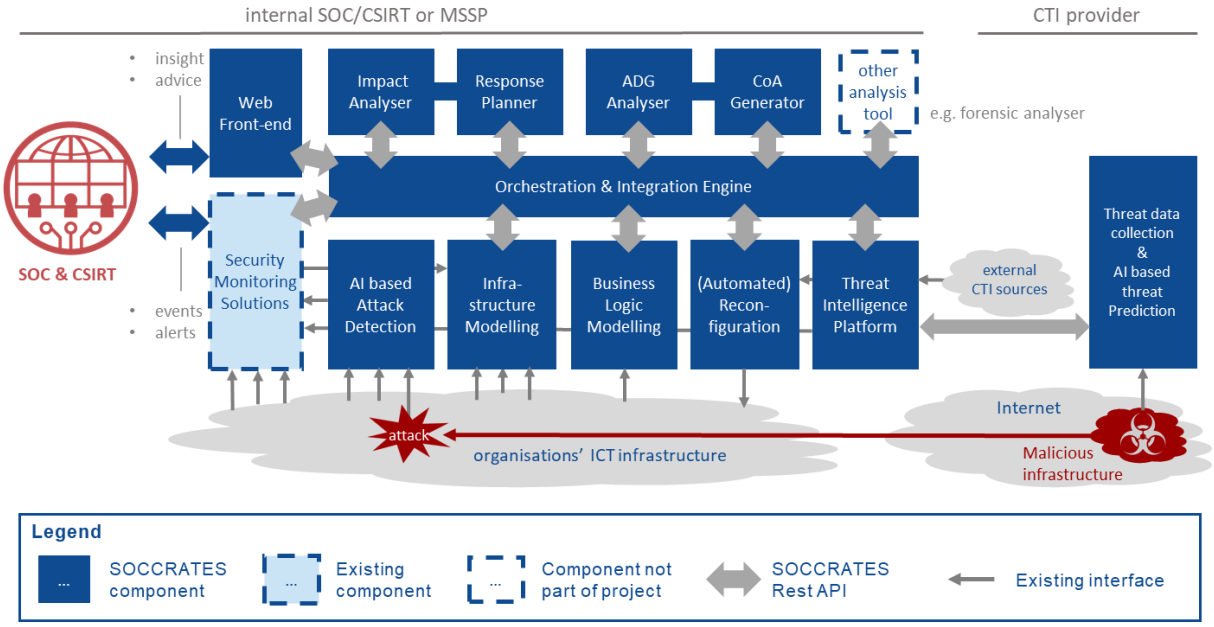


Figure 1-1 – The SOCCRATES platform

The SOCCRATES platform (see Figure 1-1) consists of an orchestrating function and a set of innovative components for automated infrastructure modelling, attack detection, cyber threat intelligence utilization, threat trend prediction, and automated analysis using attack defence graphs and business impact modelling to aid human analysis and decision making on response actions, and enable the execution of defensive actions at machine-speed.

SOCCRATES has the following concrete project objectives:

1. Deliver the SOCCRATES platform consisting of an orchestration function and a unique integration of innovative background solutions that seamlessly work together.
2. Show that the SOCCRATES platform can improve SOC operations by evaluating the SOCCRATES platform in two diverse real-life pilot environments.
3. Examine and illustrate the benefits of automation for selected SOC activities to help manage the cyber security skills gap in organizations.
4. Prepare for successful exploitation by the SOCCRATES partners of the individual innovated components and the integrated SOCCRATES platform in commercial products that are offered to the market and are available for the European (business) community.

Please visit http://www.soccrates.eu for more information on the SOCCRATES project.

1.2 This deliverable

This deliverable is the second report of WP6, framed mostly on results of Task 6.2 (Integration of SOCCRATES Platform). The goal of this document is to describe the initial version of the SOCCRATES platform with the integration of the different components with the SOCCRATES Orchestrator & Integration Engine. Additionally, this deliverable includes the description of the SOCCRATES workflows implementation for the five use cases defined in SOCCRATES and a demonstration of the execution of one of these use cases from the perspective of the Orchestrator & Integration Engine.

1.3 Structure of this deliverable

The structure of this deliverable is as follows. Section 2 describes the implementation of the initial version of the SOCCRATES Platform with the integration of the components with the Orchestrator & Integration Engine, which is in charge of executing the workflows and interconnecting the different SOCCRATES components. Section 3 describes the different workflows implemented for the different use cases, detailing how to trigger them and which SOCCRATES components are involved during its execution. Section 4 focuses on the demonstration of the SOCCRATES platform prototype through the execution of the UC5, which is triggered when a new asset is detected in the monitored infrastructure.

2 The initial SOCCRATES Platform

2.1 Overview

The SOCCRATES Platform is composed, as described in section 2 of D2.3 (Frank Fransen, 2021), by a set of components (depicted in Figure 1-1) responsible for providing the different capabilities that should support the platform based on the use cases defined in the project.

The individual description of each component is available in the different deliverables produced in WP3, WP4 and WP5. In particular, the initial ADG based Attack Analysis prototype was reported in D3.1 (Alkkiomäki, 2021), the initial prototype of the AI-based Attack Detection and Threat Intelligence Platform components were reported in D4.1 (Chiscop, 2020) and D4.2 (Kijewski, 2020), and the Business Logic Modelling and Impact Analyzer and Response Planner components in D5.2 (Kiennert, 2021).

In this chapter of the deliverable, we are going to focus on the integration of the components in the SOCCRATES Platform performed through the Orchestration and Integration Engine (OIE). It consists of two main subcomponents, the SOCCRATES Orchestrator Engine and the SOCCRATES Integrator Engine which includes the Cortex framework. In next sections, the implementation of these subcomponents is described in detail and how it is used to build the SOCCRATES Platform integrating all their components.

2.2 The SOCCRATES Orchestrator Engine

The SOCCRATES Orchestrator Engine is, together with the SOCCRATES Integration Engine, one of the core components of the SOCCRATES platform. It is responsible for providing the capability of managing and triggering the different stages that need to be followed in the workflows associated to operations related to an incident detection or an incident response.

As it was described in D6.1 (Trapero, 2020), the SOCCRATES Orchestrator Engine integrates Activiti[1] as workflow engine to manage the execution of the security automation and decision processes included in the BPMN files defined in the project. The three components of Activiti (Activiti Engine, Activiti Admin and Activiti Rest) that need to be deployed and the details about Activiti installation and configuration are also included in section 2.3.1 of that deliverable D6.1.

2.2.1 Orchestrator Core

As described in D6.1 (Trapero, 2020), the different use cases are implemented using Activiti as BPMN workflows, or as Activiti calls them, *process definitions*.

[1] https://www.activiti.org/

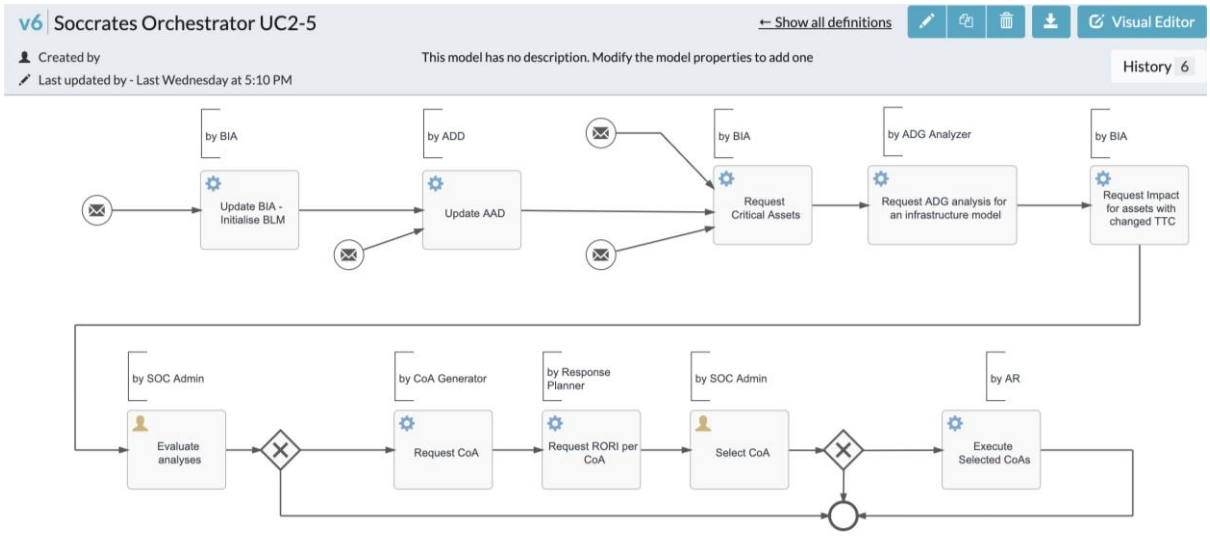


Figure 2-1 - Activiti process definition (workflow) for UCs 2 to 5

The workflows are triggered by external components through the Activiti REST API and are executed in the Activiti engine, who is in charge of invoking the different tasks and controling the flow between them. Those tasks invokes Java classes that have been developed to implement the requirements of the SOCCRATES Orchestrator for each step of the use cases.

In the Activiti UI we can edit the workflows and set the invoked Java class for each service task. This can be done by setting the "Class" field, as shown in Figure 2-2.

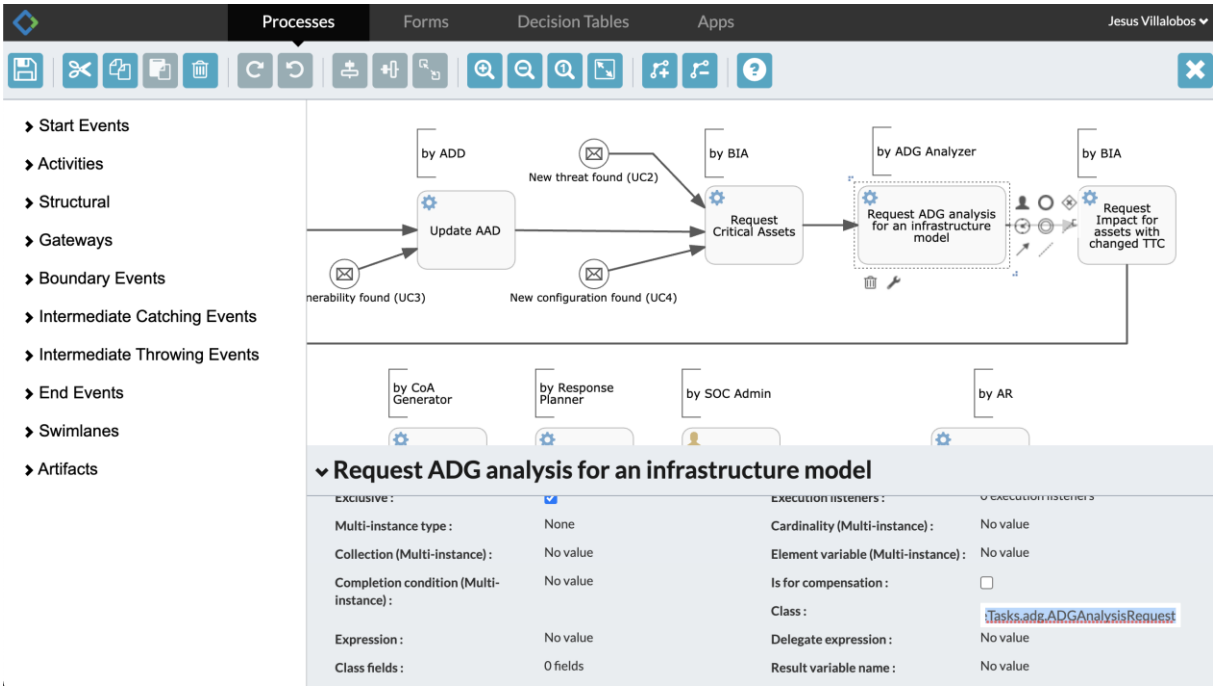


Figure 2-2 - Activiti workflow editor - Setting the invoked Java Class

2.2.1.1 Java Project

To implement these Java classes, we have created a Java project with the core of the SOCCRATES Orchestrator, which includes all the executable tasks and dependencies, such as utils, data models, etc.

A brief description of the structure of this project is detailed below for each package of the project. A screenshot of the project structure is shown in Figure 2-3.

Figure 2-3 - SOCCRATES Orchestrator Core - Java Project Structure

- **UCStarter**: Main class used to run a local instance of the Activiti-engine and to trigger the different UCs with example data input. It is mainly used for debugging and testing purposes.
- **activiti**: Interface to interact with the running Activiti engine.
- **cortex**: Interface to interact with Cortex Responders and Analyzers.
- **dataSource**: Interfaces with the data sources (e.g. Databases connections).
- **entity**: Data models mapped with database tables. Includes the specific orchestrator tables, excluding the Activiti ones.
- **listeners**: The listener classes are invoked when some event happens during the execution of the workflows. For example, when a use case (workflow) starts, a listener is invoked to register in database it has started.
- **model**: Data model used for the requests and responses with the different Cortex Analyzers and Responders. This data model is based on the description of the different Cortex Analyzers and Responders provided in D2.3 (Frank Fransen, 2021).
- **serviceTasks**: This package contains all the service tasks invoked form the Activiti workflows. An example of a service task is provided later in section 2.2.1.2.
- **utils**: Provides different methods to manipulate Dates, JSONs, log information, etc.

2.2.1.2 Service Tasks

The service tasks are invoked from the Activiti engine during the execution of the workflows. Each service task performs a set of different tasks required for each step of the workflow.

Generally, each service task performs at least the following actions:

1. Access to the required execution variables of the workflow (data from other previous steps that was stored as a global variable).
2. Prepare the request to a Cortex Analyzer/Responder.
3. Log the request into the database.
4. Perform the request to the Cortex Analyzer/Responder.
5. Check the response from the Cortex Analyzer/Responder.
6. Store the response as a global execution variable, so other Service Tasks can access to it.
7. Log the response in database.

In Figure 2-4, an example of the service task is shown to generate the Course of Actions.

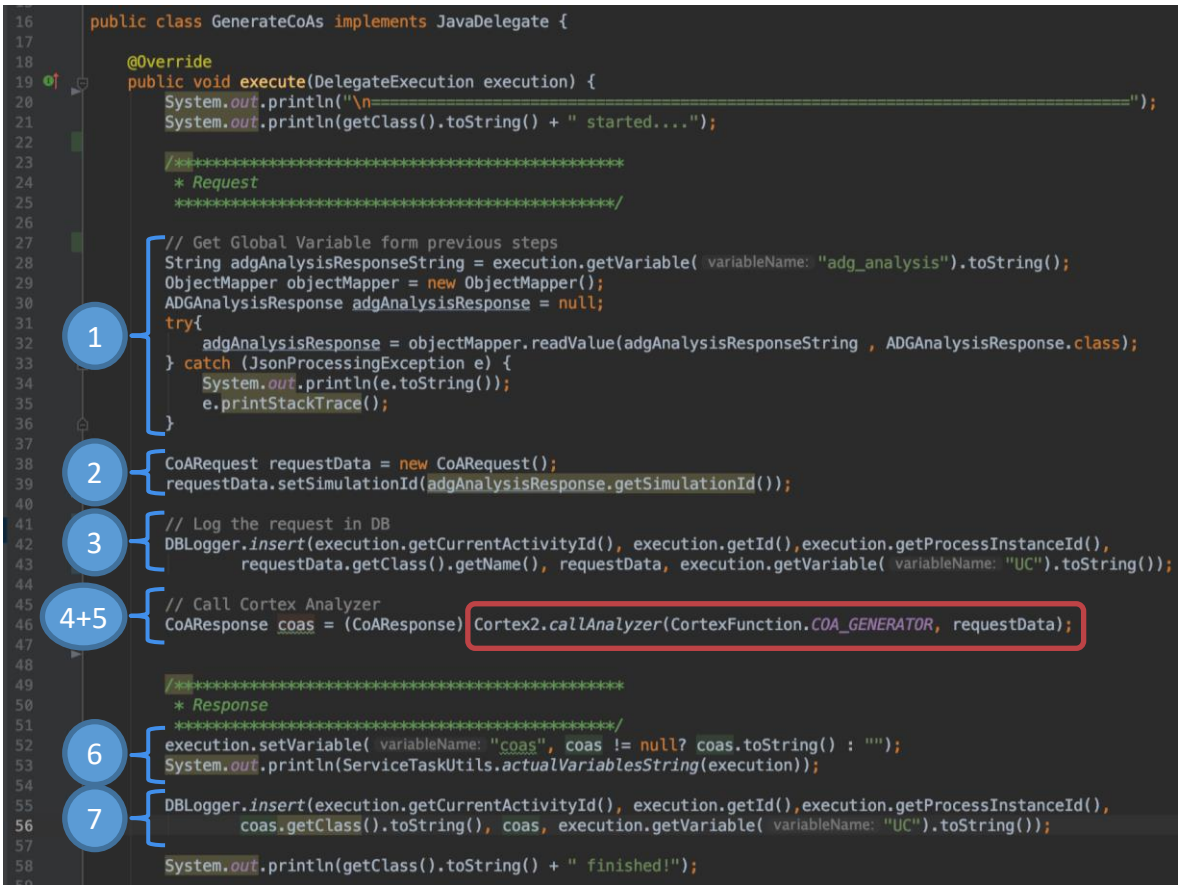


Figure 2-4 - Service task example

2.2.1.3 Running Orchestrator Core in Testing/Debbuging mode

As introduced in section 2.2.1.1, the Orchestrator Core has a main Class named **UCStarter,** which is able to run a local Activiti Egine instance that offers the possibility of debugging the execution of the workflows in real-time. It is not meant to be run as a standalone instance,

since it requires access to an already running Activiti Database.It does include the Activiti Engine and is able to run Activiti workflows without the whole Activiti installation described in D6.1 in sections 2.3.1 and 2.4.

As shown in Figure 2-5, to execute this main class we pass two arguments: the UC number and the user who starts it. It starts the UC3 by passing a preconfigured example input message to the workflow, simulating the request to the Activiti REST API that the IMC component would do to trigger the UC3 when a new vulnerability is found.

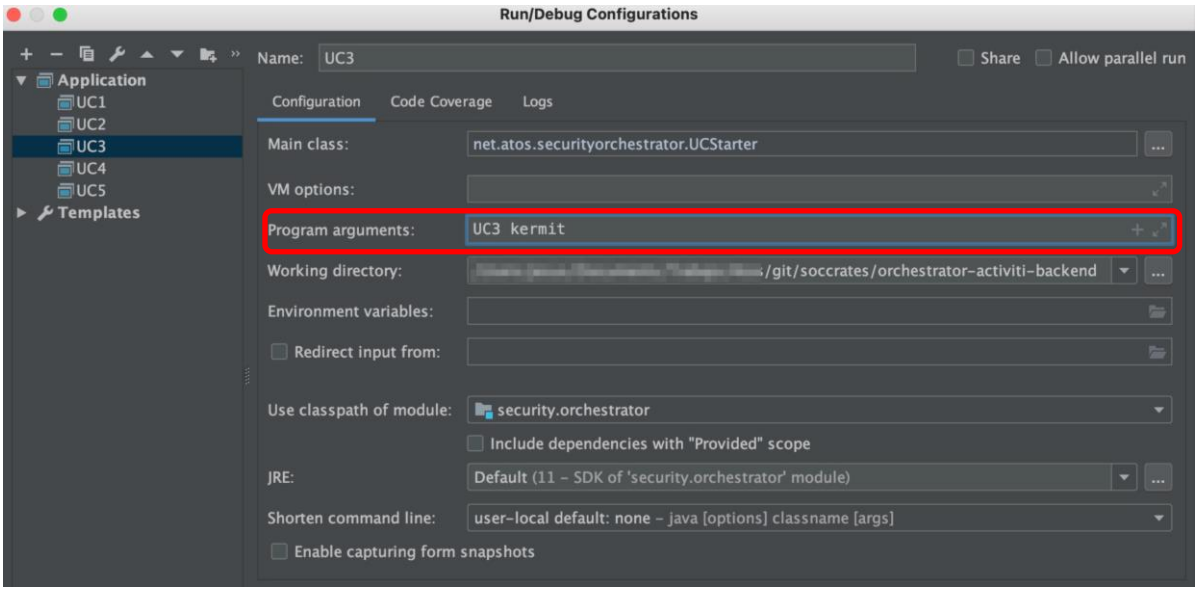


Figure 2-5 - Orchestrator Core execution for debugging/testing

This capability allows us to test and debug the execution of the different workflows, having the ability of getting internal variables values, checking the execution steps, etc. This ability is really helpful for the development and integration processes since it eases the task of finding bugs and errors. Figure 2-6 shows an example of the output in the terminal.

```
class soccrates.atos.orchestrator.serviceTasks.aad.UpdateAADModel started....
Hibernate:
    insert
    into
        orchestrator_log
        (created, data, execution_id, class, process_instance_id, task_id, type, use_case)
    values
        (?, ?, ?, ?, ?, ?, ?, ?)
[input_vulnerability_uc3, input_message, UC]
class soccrates.atos.orchestrator.serviceTasks.aad.UpdateAADModel finished!

================================================================================
class soccrates.atos.orchestrator.serviceTasks.adg.ADGAnalysisRequest started....
Task Name: adg_analysis
processInstanceId: 1647501
processDefinitionId: SOCCRATES_ORCHESTRATOR_UC25:32:1232645
currentActivitiId: adg_analysis
superExecutionId: null
parentId: 1647501
eventName: null
id: 1647503
processInstanceBusinessKey: null
rootProcessInstanceId: 1647501
Hibernate:
    insert
```

Figure 2-6 - Orchestrator Core terminal output

Furthermore, it allows us to interact with the terminal to simulate the inputs that the user would give through the Orchestrator UI, what enables us to complete the workflows locally with simulated data inputs.

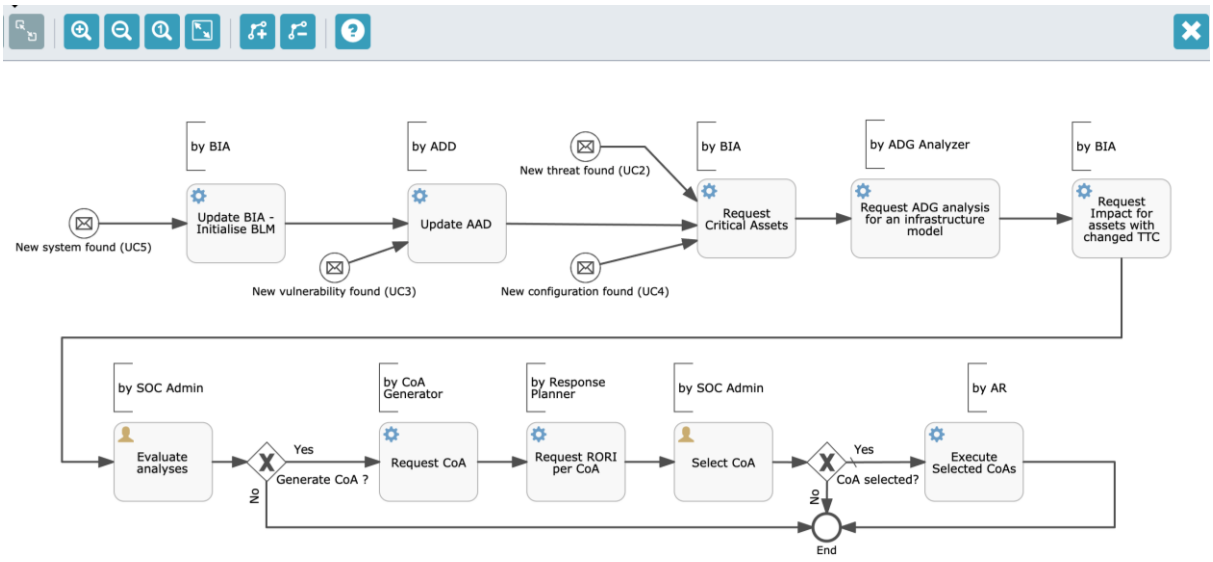
```
[adg_analysis, input_vulnerability_uc3, impact_analysis, input_message, UC]
class soccrates.atos.orchestrator.serviceTasks.impactAnalyser.GetBusinessImpactForAssets finished!
Process started with process instance id [1650001] key [SOCCRATES_ORCHESTRATOR_UC25]
Active outstanding tasks: [0]
Active outstanding tasks: [1]
Processing Task [Evaluate analyses]
mustGenerateCoA?
true
adg_analysis_results?
```

Figure 2-7 - Orchestrator Core input throught terminal

2.2.2 Orchestrator Workflow Start Events

To start a workflow, Activiti provides the "Start message events", which has the capability of starting a workflow when certain message is received through the Activiti REST API.

To configure the messages that can start a workflow, we declare the "message definitions" from the Activiti workflow editor, as shown in Figure 2-8 and marked in red.



Soccrates Orchestrator UC2-5

Process identifier :	SOCCRATES_ORCHESTRAT ...	Name :	Soccrates Orchestrat ...
Documentation :	No value	Process author :	No value
Process version string (documentation only) :	No value	Target namespace :	http://www.activiti. ...
Execution listeners :	0 execution listeners	Event listeners :	0 event listeners
Signal definitions :	No signal definitions configured	Message definitions :	4 message definitions

Figure 2-8 - Message definitions

The message definitions can be considered as IDs that are linked to the workflow. We can add, create or delete message definitions from the Activiti interface, as shown in Figure 2-9.

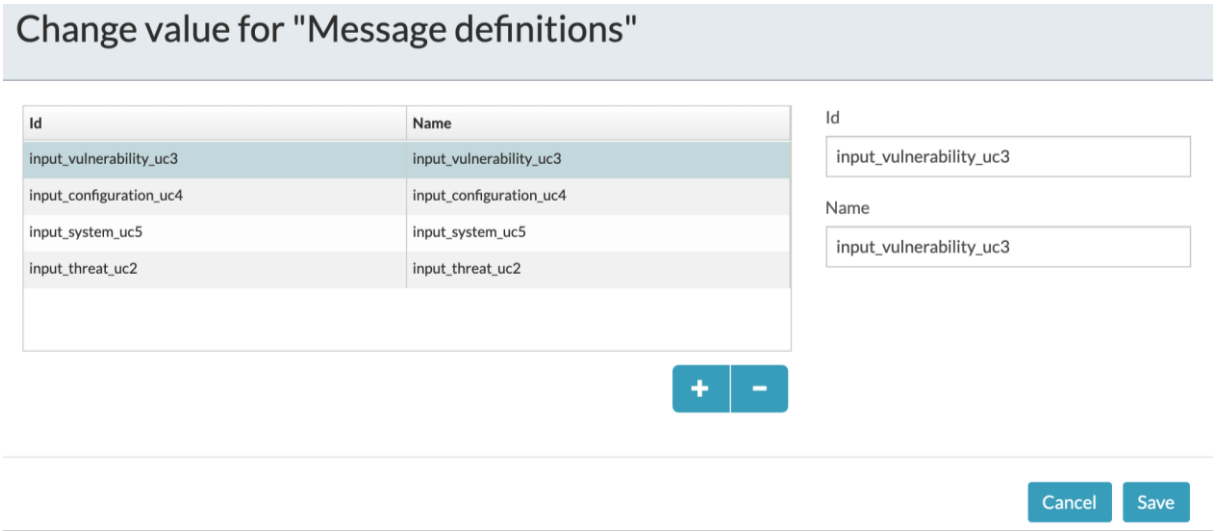


Figure 2-9 - Message deifnitions editor

Once the message definitions is defined, we can link the « start message events » with them. When a message is received through the Activiti API with this message ID, the workflow will be triggered starting from this event. The Figure 2-10 depicts how to associate the « start message event » with the message definition (i.e. Message reference).

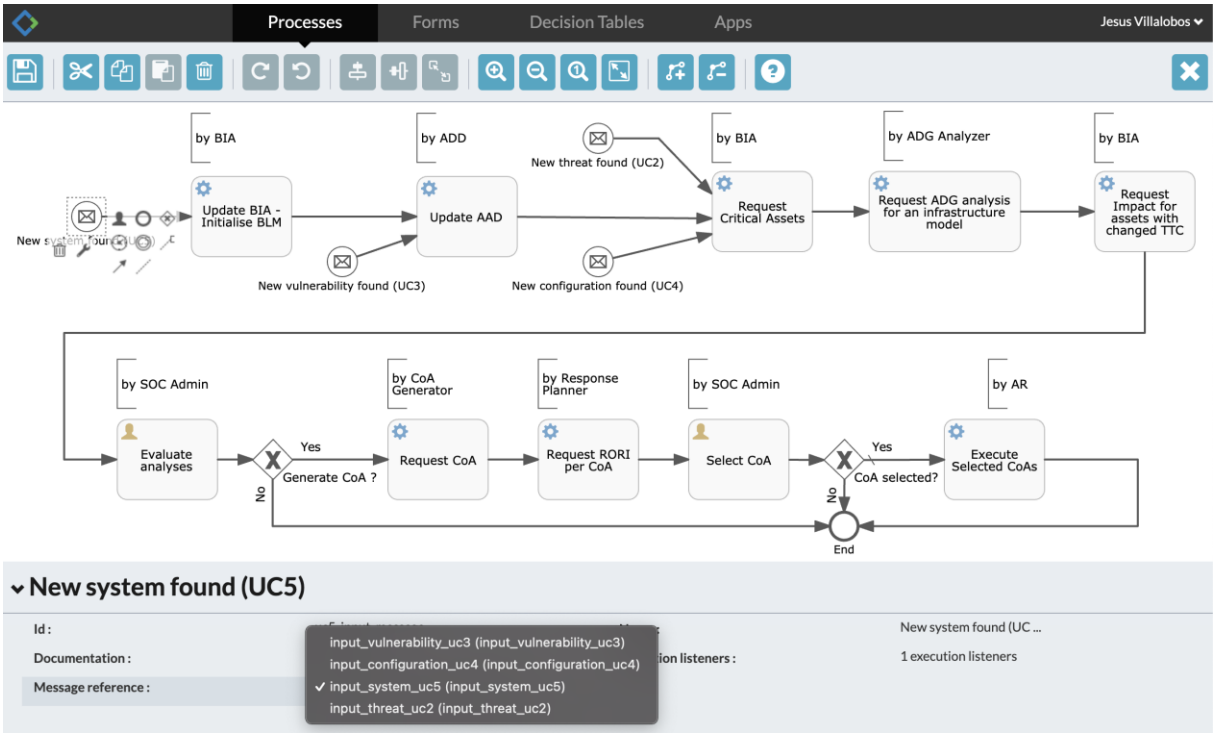


Figure 2-10 - Start message event association with message reference

Sending a message to the Activiti API using one of the message definitions already defined, will trigger the workflow execution of one of the UCs. The triggering of use cases is defined in the section 3 of this document, where an example of each triggering message is included.

2.2.3 Logging system

A first version of a logging system has been implemented in the Orchestrator Core, which logs in a database the activity of the Service Tasks and their interactions with the Cortex Analyzers & Responders.

Those logs can be used for generating metrics, statistics, for locating errors and bugs, etc. In this first version, the logging system is able to log:

- When a workflow starts / ends, through Activiti Listeners implemented in the Orchestrator Core.
- When a User Task starts / ends, through Activiti Listeners implemented in the Orchestrator Core.
- The requests the Service Tasks perform to the Cortex Analyzers / Responders.
- The responses from the Cortex Analyzers / Responders to the Service Tasks.

It is foreseen to extend this list and log more events, like when a use case stops because an error, or others related to the UI, like when the SOC analyst access to the user tasks.

In Figure 2-11 it is depicted an example of the logs generated during the execution of the workflow for the UC3.

id	executio...	process_i...	use_case	created	data	task_id	type	class
1...	2027503	2027501	5	2021-11-09 17:02:15	{"mustGenerateCoA"...		UC_end	class soccrates.atos.orchestrator.listeners.UCListener
1...	2027503	2027501	5	2021-11-09 17:02:09	{"description":"Selec...	execute_coas	task_service_request	soccrates.atos.orchestrator.model.AREXecuteCoasRequest
1...	2027503	2027501	5	2021-11-09 17:02:08	{"mustGenerateCoA"...	select_coa	task_user_end	class soccrates.atos.orchestrator.listeners.UserTaskListener
1...	2027503	2027501	5	2021-11-09 17:00:56	{"adg_analysis":"{\"si...	select_coa	task_user_start	class soccrates.atos.orchestrator.listeners.UserTaskListener
1...	2027503	2027501	5	2021-11-09 17:00:54	{"initialTTC":{"228.R...	request_coa	task_service_response	class soccrates.atos.orchestrator.model.CoAResponse
1...	2027503	2027501	5	2021-11-09 17:00:52	soccrates.atos.orche...	request_coa	task_service_request	soccrates.atos.orchestrator.model.CoARequest
1...	2027503	2027501	5	2021-11-09 17:00:51	{"adg_analysis":"{\"si...	evaluate_analyses	task_user_end	class soccrates.atos.orchestrator.listeners.UserTaskListener
1...	2027503	2027501	5	2021-11-09 17:00:09	{"adg_analysis":"{\"si...	evaluate_analyses	task_user_start	class soccrates.atos.orchestrator.listeners.UserTaskListener
1...	2027503	2027501	5	2021-11-09 17:00:08	{"business_impacts":...	impact_analyzer...	task_service_response	soccrates.atos.orchestrator.model.ImpactAnalysisBusinessResponse
1...	2027503	2027501	5	2021-11-09 17:00:02	{"business_name":n...	impact_analyzer...	task_service_request	soccrates.atos.orchestrator.model.ImpactAnalysisBusinessRequest
1...	2027503	2027501	5	2021-11-09 17:00:01	{"simulationId":"307...	adg_analysis	task_service_response	class soccrates.atos.orchestrator.model.ADGAnalysisResponse
1...	2027503	2027501	5	2021-11-09 16:59:59	{"date":"2021-11-0...	adg_analysis	task_service_request	soccrates.atos.orchestrator.model.ADGAnalysisRequest
1...	2027503	2027501	5	2021-11-09 16:59:58	{"critical_assets":["a"...	critical_assets	task_service_response	class soccrates.atos.orchestrator.model.BIA_CriticalAssetsResponse
1...	2027503	2027501	5	2021-11-09 16:59:52	{"Threshold":0.6}	critical_assets	task_service_request	class soccrates.atos.orchestrator.model.BIA_CriticalAssetsRequest
1...	2027503	2027501	5	2021-11-09 16:59:46	{"host":"10.0.0.1","ty...	update_aad	task_service_request	soccrates.atos.orchestrator.model.AADNewSystemRequest
1...	2027503	2027501	5	2021-11-09 16:59:40	{"newAsset":["comp...	initialise_blm	task_service_request	soccrates.atos.orchestrator.model.UpdateBIARequest
1...	2027503	2027501	5	2021-11-09 16:59:38	{"input_system_uc5"...		UC_start	class soccrates.atos.orchestrator.listeners.UCListener

Figure 2-11 - SOCCRATES Orchestrator Database Log sample

The fields of the database table that stores the logs are defined as follows:

- **id**: Identifier of the log entry.
- **created**: Creation date of the log entry.
- **execution_id**: Identifier of the execution of the workflow. Unique between all executions.
- **process_instance_id**: Similar to the execution_id, it identifies the execution of the process definition (the workflow). Unique for each process instance.
- **task_id**: Identifier of the task that produced the log entry. It's empty in the case of "UC_start" and "UC_end" events. For entries generated by service tasks and user tasks it uses the id of the task defined in the Activiti workflow editor, as shown in the Figure 2-12.
- **type**: Defines the type of log entry. The defined types are:
    - ***UC_start***: Used when a use case starts, before any task is invoked.
    - ***UC_end***: Used when a use case ends, after all tasks in the workflow have been invoked.
    - ***task_service_request***: Before a service task invokes a Cortex Analyzer/Responder, its request data is logged here.

- ***task_service_response:*** When a service task receives the response from a Cortex Analyzer, it logs the returned data.
- ***task_user_start:*** Logs the moment when the orchestrator starts processing a user task, before being available for the UI.
- ***task_user_end:*** Logs the moment when the orchestrator finishes a user task, before receiving the feedback from the UI.

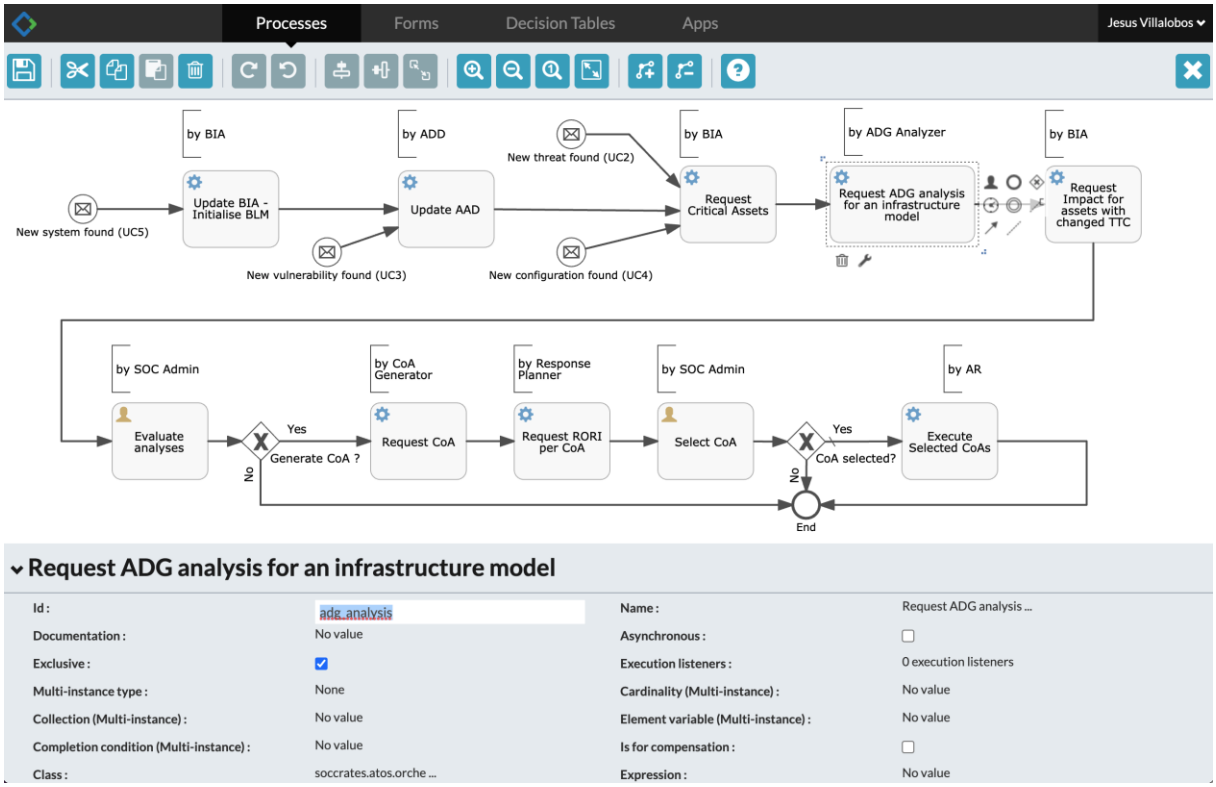


Figure 2-12 - Process definition editor - Task id

2.3 The SOCCRATES Integration Engine

As it was described in D6.1 (Trapero, 2020), Cortex[2] is the technology chosen in SOCCRATES for the integration with the SOCCRATES Orchestrator Engine of the different tools included in the SOCCRATES platform. Each tool vendor has implemented its own Cortex Analyzer or Responder to enable this integration, as it was described in detail in section 2.3.2 of D6.1.

The following components of the SOCCRATES platform are integrated with the Orchestrator and Integration Engine through Cortex analyzers or responders. The complete description of these components is included in the deliverable D2.3 – System Architecture & Interface Specification Final Version (Frank Fransen, 2021):

- **Attack Defence Graph Analyzer (ADG)**
  The ADG Cortex analyzer is the interface between the OIE and the Attack Defence Graph analyzer. Additionally, on one side it provides the communication of the component with the Threat Intelligence Platform and the SIEM to get Indicators of Compromise, and on the other

[2] https://github.com/TheHive-Project/Cortex

hand with the Infrastructure Modelling Component to request and infrastructure model (see Figure 3.6 of D2.3)

- **Course of Action Generator (CoA)**
  The CoA Cortex analyzer is used to invoke through the OIE the functionality provided by the CoA Generator to suggest to the SOC analyst potential defences included in the model that could be activated as a feedback for the ADG analyzer.
- **Business Impact Analyzer (BIA) & Business Logic Modelling**
  Two Cortex analyzers provide the interface to invoke the Business Impact assessment for affected or potentially affected assets and for containment. These analyzers will interact with the Infrastructure Modelling Component and the Business Impact Analyzer as described in section 3.7 of D2.3. A Cortex responder is also provided to trigger the update of a model.
- **AI based Attack Detection (AAD)**
  The OIE interacts with the AAD component through the AAD Cortex responder to notify about changes in the infrastructure (in particular, new vulnerabilities or assets detected) using the REST API provided. The architecture is shown in Figure 3.2 of D2.3.
- **Automated Reconfiguration (AR)**
  Through Cortex responders, this component allows to interact with the IT support or IT infrastructures to perform some mitigation action depending on the selection done by the SOC analyst. Current version of the demonstrator only includes a Cortex responder to send an email to the IT support (or the email configured) with the selected course of action and information about the analysis performed.
- **Response Planner (RP)**
  The RP Cortex analyzers are the interfaces used to evaluate the Return On Response Investment (RORI) for CoAs in the different use cases, and to generate containment CoAs in the case of the Use Case 1 where it is required an attack response.

The Infrastructure Modelling Component (IMC) does not include Cortex analyzers or responders since it only interacts indirectly with the SOCCRATES Orchestrator and Integration Engine. As it is described in section 3.3 of deliverable D2.3, three analyzers of the Response Planner component (the ADG Cortex Analyser, the BIA Cortex Analyser and the Containment CoA Cortex Analyser) invoke directly the APIs of the IMC to interact with this component.

Following sections describe in detail each of these tools and the analyzers and responders associated to them. The table below summarizes them classified by component.

Table 2-1 : SOCCRATES Cortex Analyzers and Responders

Component	Type	Analyzer ID	Function	UC - Phase
BIA	Analyzer	BIA.1	Assess Bussiness Impact for affected asset	UC1 – Initial analysis phase
	Analyzer	BIA.1	Assess BIA for potentially affected assets	UC1 - Initial analysis phase
	Analyzer	BIA.1	Assess BIA for containment	UC1 - Containment determine strategy
	Responder	BIA.2	Trigger the execution of a new model	UC5

	Analyzer	BIA.3	Assess BIA	UC2, UC3, UC4, UC5 – Request impact for assets with changed TTC
RP	Analyzer	RP.1	Generate containment CoA	UC1 - Containment determine strategy
	Analyzer	RP.2	Generate RORI per CoA	UC1 - Containment determine strategy
	Analyzer	RP.2	Generate RORI per CoA	UC2, UC3, UC4, UC5
ADG	Analyzer	ADG.1	Generate next step analysis	UC1 - Initial analysis phase
	Analyzer	ADG.1	Determine attack path. Internally it communicates with TIP and SIEM to get IoC	UC1 - Containment full extent
	Analyzer	ADG.1	Request ADG analysis for an infrastructure model	First task after triggering workflow: - UC2: New threat - UC3: New vulnerability - UC4: New config change - UC5: New asset
CoA	Analyzer	CoA.1	Generate CoA to prevent initial attack path	UC1 - Containment full extent UC2, UC4, UC5
AR	Responder	AR.1	Send email with CoA	all UC / all CoAs
AAD	Responder	AAD.1	Report information detected	UC3 and UC5

2.3.1 Business Impact Analyzer

Analyzer / Responder ID	BIA.1
Type (Analyzer / Responder)	Analyzer
Component	Business Impact Analyzer
Function(s)	Assess BIA for affected asset Assess BIA for potentially affected assets
Cortex Analyzer / Responder name	SOCCRATES_Business_GetCritical_Analyzer_1_0
Cortex Analyzer / Respoder ID	fa1c2d6dd21eb1e407f9f407286b49be

Analyzer / Responder ID	BIA.2
Type (Analyzer / Responder)	Analyzer
Component	Business Impact Analyzer
Function(s)	Assess BIA
Cortex Analyzer / Responder name	SOCCRATES_Business_Impact_analyzer_1_0
Cortex Analyzer / Respoder ID	f4e47987f12d771a7508b18817b432d3

Analyzer / Responder ID	BIA.3
Type (Analyzer / Responder)	Responder
Component	Business Impact Analyzer
Function(s)	Trigger the execution of a new model
Cortex Analyzer / Responder name	SOCCRATES_Business_Impact_responder_1_0
Cortex Analyzer / Respoder ID	9db22dd73574180f3b17402fcf3f5736

The Business Impact Analyzer relies on two Cortex Analyzers and one Cortex Responder:

- A Cortex Analyzer (SOCCRATES_Business_GetCritical_Analyzer_1_0) which fetches the list of critical assets, and return a list of assets (identified by the IMC UUID) that have a criticality value superior to the threshold value given as input (whose default value is 0.66).
- A Cortex Analyzer which computes the business impact of one or more external events. The impact is computed by default on the most important business entity (called Business Company).
- A Cortex Responder which triggers an update of the Business Logic model following the addition of a new asset in the infrastructure. This Cortex Responder is only called during Use Case 5.

Examples of requests to the function "bia_fetchCriticalAssets" of the **SOCCRATES_Business_GetCritical_Analyzer_1_0**, the function "bia_analysis" of the **SOCCRATES_Business_Impact_analyzer_1_0** and the function "bia_updateModel" of the **SOCCRATES_Business_Impact_responder_1_0**, and their responses can be found in section 8.1.1 of Annex I.

2.3.2 Respone Planner

Analyzer / Responder ID	RP.1
Type (Analyzer / Responder)	Analyzer
Component	Response Planner
Function(s)	Generate containment CoA
Cortex Analyzer / Responder name	SOCCRATES_RP_CONTAINMENT_COA_GENERATOR
Cortex Analyzer / Respoder ID	9f4654ab7b7c6f679fb9bac69ebddd20

Analyzer / Responder ID	RP.2
Type (Analyzer / Responder)	Analyzer
Component	Response Planner
Function(s)	Ranks CoAs by RORI
Cortex Analyzer / Responder name	SOCCRATES_RP_COA_RANKING_ANALYZER
Cortex Analyzer / Respoder ID	a57e8fcb972c297f5206457d5e20dbd3

The Response Planner relies on two Cortex Analyzers:

- A Cortex Analyzer (SOCCRATES_RP_CONTAINMENT_COA_GENERATOR) to generate CoAs following the reception of an event from the SIEM or from REASENS
- A Cortex Analyzer (SOCCRATES_RP_COA_RANKING_ANALYZER) to rank the CoAs according to the RORI value and produce a CACAO playbook for each CoA

An example of a request to the function "generateContainmentCoas" of the **SOCCRATES_RP_CONTAINMENT_COA_GENERATOR** analyser and to the function "coagen" of the **SOCCRATES_Business_Impact_responder_1_0** responder, and their responses can be found in section 8.1.2 of Annex I.

2.3.3 Attack Defense Graph

Analyzer / Responder ID	ADG.1
Type (Analyzer / Responder)	Analyzer
Component	Attack Defense Graph
Function(s)	Request ADG analysis for an infrastructure model
Cortex Analyzer / Responder name	SOCCRATES_ADG_ANALYZER
Cortex Analyzer / Respoder ID	ea168ce0f257d03ddf3c1839b3fb5cab

The functionality provided by the ADG to the SOCCRATES platform is utilized via the ADG Cortex Analyzer and its main function "simulation". The simulation function exposed by the Cortex Analyzer handles all Use Cases provided by the ADG and takes a set of optional parameters depending on which Use Case in the workflow that has been triggered. All parameters are optional and the ADG assigns default values for all omitted parameters. The core functionality, configuration and function parameters are defined and explained in deliverable D2.3 in chapter 3.4 Attack Defence Graph Analyzer.

Below is an example of a request to the ADG Cortex Analyzer and the function "simulation". In this example we will send a request to conduct a simulation on the current environment with an attacker that has the capabilities of APT-28 and starts with user execution on a workstation (0a6b636c-3dc8). The simulated attacker will attempt to destroy data in our customer database (5c3b425c-1dc5) using a newly discovered vulnerability (CVE-2021-26427) on an Exchange server (6749be31-f8aa) in the network. The response to the simulation request has been intentionally truncated to conserve space. The full response specification can be found in the deliverable mentioned above.

An example of a request to the function "simulation" of the **SOCCRATES_ADG_ANALYZER** analyser and its response can be found in section 8.1.3 of Annex I.

2.3.4 Course of Action

Analyzer / Responder ID	CoA.1
Type (Analyzer / Responder)	Analyzer
Component	Course of Action
Function(s)	Generate CoA to prevent initial attack path Request CoA for a given incident
Cortex Analyzer / Responder name	SOCCRATES_Coa_Generator_1_0
Cortex Analyzer / Respoder ID	f8769db7c0256b1820102dc20e84bbd1

The Course of Action (CoA) generator is used to tackle the problem of selection of countermeasures against cyber-attacks in IT infrastructure. The CoA generator suggests and activate a list of defense-steps which are disabled in the model of IT infrastructure. The process of CoA generation is an iterative process that operates defense-steps in the attack graphs enhancing the IT infrastructure security to the security operation center (SOC) analyst. The attack graphs generated by Attack Defence Graph (ADG) Analyzer that works by creating a feedback loop with the CoA Analyzer [1]. The process of CoA generation is optimized by considering initial attack path generated by the ADG Analyzer and get the efforts of the attacker, repeat the procedure to find the new attack graph for further analysis. The aforementioned iterative process is performed by the CoA generator without the involvement of the orchestration and integration engine (OIE).

CoA Cortex Analyzer is implemented to execute the CoA generator using orchestration and integration engine. The CoA generator needs a simulation id to begin its execution. The simulation id is a pointer to the attack defense graph generated from the model of IT infrastructure. Another input for the CoA Cortex Analyzer is the budget that is an optional parameter used for selecting the monetary costs. The mandatory parameter is simulation id to execute CoA Cortex Analyzer whereas the budget for defenses is considered as an optional. In case, budget is not provided to CoA generator, it assume zero budget.

An example of a request to the function "coagen" of the **SOCCRATES_Coa_Generator_1_0** analyser and its response can be found in section 8.1.4 of Annex I.

2.3.5 Automatic Reconfiguration

Analyzer / Responder ID	AR.1
Type (Analyzer / Responder)	Responder
Component	Mailer_1_0
Function(s)	Sends an email to the SOC analyst with information regarding the case and the selected CoA
Cortex Analyzer / Responder name	Mailer_1_0
Cortex Analyzer / Respoder ID	be1f213893fb57a8dd4cddf9eb545216

The Automatic Reconfiguration relies on a Cortex responder, based on the code for the Mailer responder available at TheHive Project[3], to provide the functionality to send an email to the SOC analyst with the list of selected course of actions and information related (such as the RORI, the business impact or the report URL).

An example of a request to the function "ar_mailer" of the **Mailer_1_0** responder and its response can be found in section 8.1.5 of Annex I.

2.3.6 Advanced Attack Detection

Analyzer / Responder ID	AAD.1
Type (Analyzer / Responder)	Responder
Component	Advanced Attack Detection
Function(s)	Report information detected (i.e., new vulnerabilities, new assets)
Cortex Analyzer / Responder name	SOCCRATES_AAD_RESPONDER
Cortex Analyzer / Respoder ID	a5bce2c4d2821f443cf499062572432d

The AAD component relies on the following Cortex Responder:

- SOCCRATES_AAD_RESPONDER triggers the AAD (Reasens) via the AAD REST API whenever a new asset or new vulnerability has been found.

Upon receiving information about new assets or vulnerabilites by the Activiti engine, the responder sends a POST request to the AAD REST API, which in turn forwards it to Reasens via Kafka. An acknowledgment is then sent back to the responder containing either an error or success message.

[3] https://github.com/TheHive-Project/Cortex-Analyzers/blob/master/responders/Mailer/mailer.py

Examples of requests to the the **SOCCRATES_AAD_RESPONDER** responder for new vulnerability or new assets and their responses can be found in section 8.1.6 of Annex I.

2.4 The integration environment

The environment prepared for the **SOCCREATES Orchestrator** and the **SOCCRATES Integration Engine** is deployed at ATOS premises.

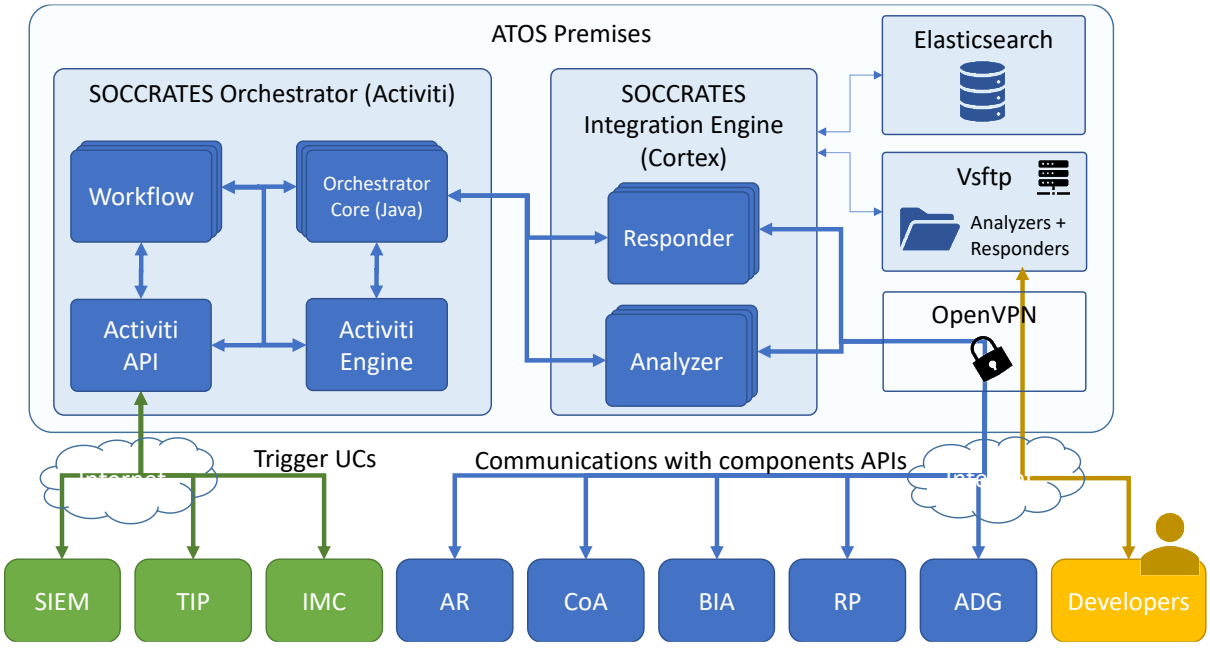


Figure 2-13 - SOCCRATES Orchestrator and Integration Engine environment diagram

The Figure 2-13 shows the different components that compose the integration environment at ATOS premises and the connections betweem them and other SOCCRATES components:

- The first one is the **SOCCRATES Orchestrator**, which is composed of an Activiti installation, the Orchestrator Core as a Java Library, and the implemented workflows for the different use cases. Here, the Activiti API receives the requests from other SOCCRATES components to trigger the workflows for the use cases.
- The **SOCCRATES Integration Engine** is mainly composed of a Cortex installation within the developed Analyzers and Responders that integrates the SOCCRATES components with the platform. It is deployed as a Docker container using docker-compose within other components like the Vsftpd server and the Elasticsearch.
- The **Elasticsearch** server is used by Cortex to store the configurations and the jobs history (i.e. the calls and responses done to the Responders and the Analyzers).
- The **Vsftpd** server is used to expose the Cortex Analyzers and Responders repository to the developers. This allows runtime changes to the analyzers and responders, easing the integration process.
- Finnally, the **OpenVPN** server is used to connect the SOCCRATES components running at their own premises (e.g. : BIA, ADG,…) with the SOCCRATES Platform which is running in the integration environment. This secures the connections between the SOCCRATES Integration Engine and the components APIs.

The **SOCCRATES Orchestrator Core** is deployed in the running Activiti installation at ATOS premises. For that, it is required to compile the Java Project described in previous section 2.2.1.1 to generate a "jar" library which contains all the Java Classes of the Orchestrator Core and its dependencies.

Once the library is compiled, it is packed in the file **soccrates-orchestrator-core-jar-with-dependencies.jar**, which needs to be copied to the folder WEB-INF/lib located under each of the Activiti components folders (/webapps/activiti-app, /webapps/activiti-admin and /webapps/activiti-rest) and then restart the Tomcat service.

This Java library is then accessible by the workflows running in the Activiti Engine, which are able to invoke the Service Tasks Java Classes that perform the requests to the Cortex Analyzers and Responders through their APIs.

Although this initial version of the SOCCRATES platform is deployed as it is described there, it is foreseen to generate a docker container for the SOCCRATES Orchestrator Engine to simplify the process in order to ease the deployment process in the pilots.

The deployment of the **SOCCRATES Integration Engine** is done using a Cortex docker image and configured following the instructions described in section 2.3.2 of D6.1. Additionally to Cortex, the docker-compose file used (see Figure 2-14) includes the deployment of Elasticsearch[4] and a Vsftp server[5], which has been included to ease the development and integration phases of the Cortex Analyzers and Responders.

[4] https://www.elastic.co/es/elasticsearch/

[5] https://security.appspot.com/vsftpd.html

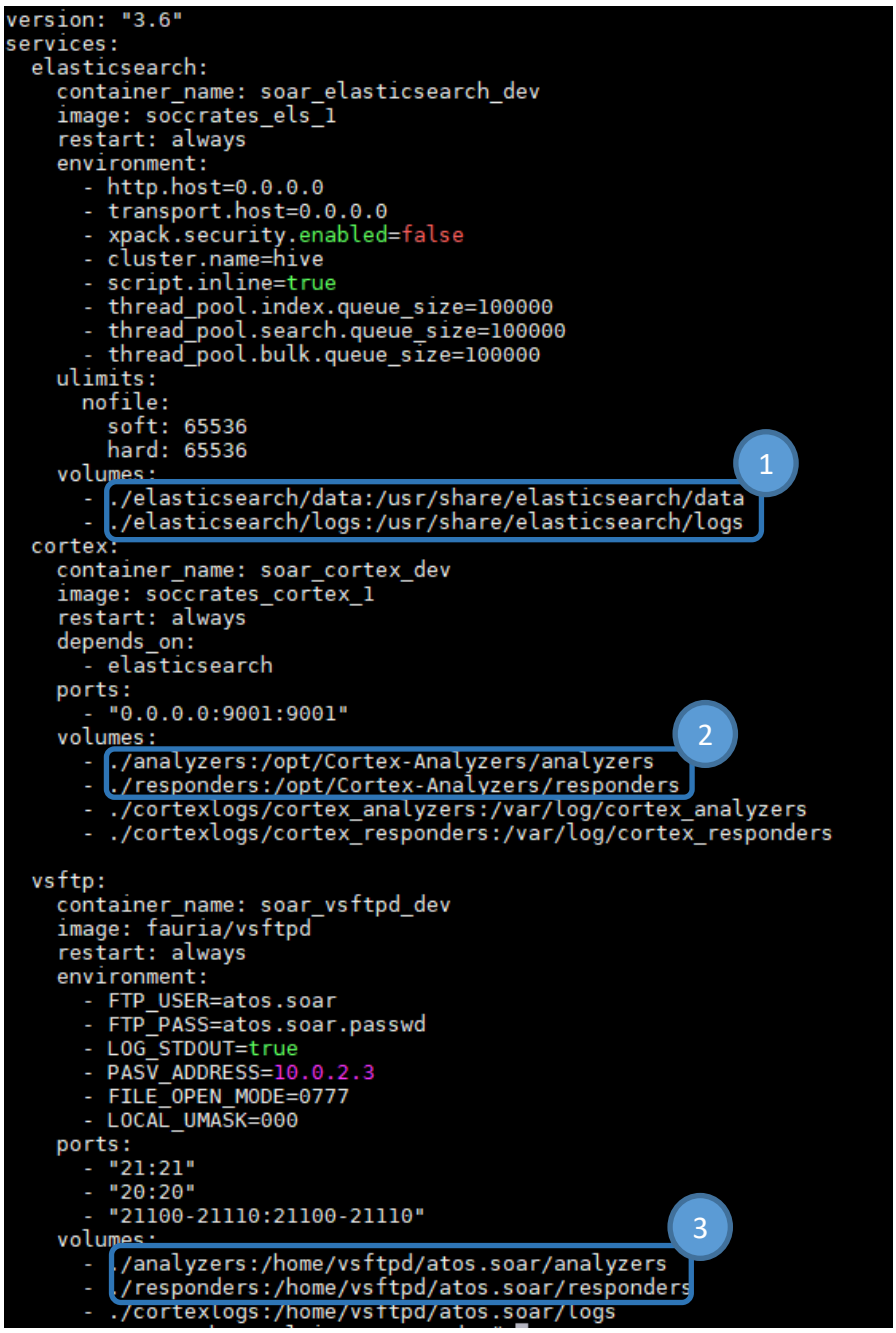


Figure 2-14 - SOCCRATES Integration Engine docker compose file

It is important to note that the elasticsearch data and log folders have been moved to a volume to make them persistent ("1" mark in Figure 2-14). The first one, data folder, stores the history of jobs triggered in Cortex and the different configuration parameters of the Responders and Analyzers. In that way, we can preserve this data even if the Docker container is deleted.

Additionally, volumes have been created to store the Cortex analyzers and responders, which are also shared with the Vsftp service (marks "2" and "3" in Figure 2-14). The access to the Vsftp service is done through VPN using the following command (or arternatively it can also be used an FTP client): ftp://user:pass@10.0.2.3:21

This FTP access gives developers the ability to directly access the analyzers and responders to update them during the development phase, as shown in Figure 2-15.

Figure 2-15 - FTP Folder to deploy Cortex Analyzers

For the deployment of new Cortex Analyzers and Responders to the SOCCRATES platform it has been defined an integration methodology with the following steps:

1. Access to SOCCRATES tested is done through VPN connection.
2. Files are uploaded via FTP into the analyzers/responders folder (see Figure 2-15).
3. Analyzers/Responders are enabled and configured through the Cortex Dashboard, which is available at http://10.0.2.3:9001 (see Figure 2-16).

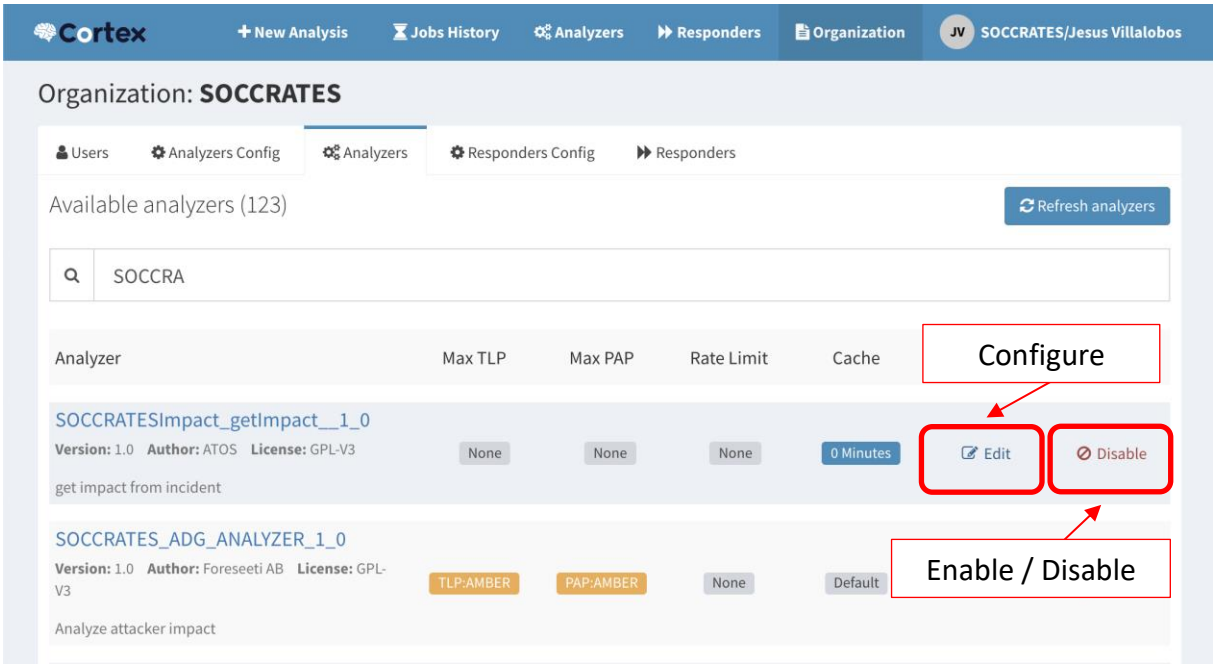


Figure 2-16 - Enabling and configuring Cortex Analyzers

Analyzers can be tested using the Cortex Dashboard (see an example in Figure 2-17) or invoking the Cortex API (see an example using the tool Postman[6] in Figure 2-18), as it is also described in section 2.3.3 of D6.1.

[6] https://www.postman.com/

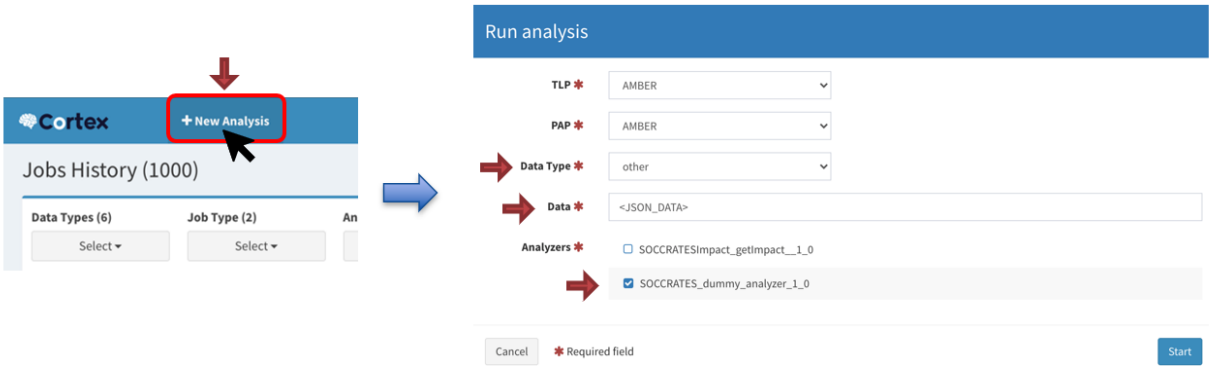


Figure 2-17 - Testing the Analyzers with Cortex Dashboard

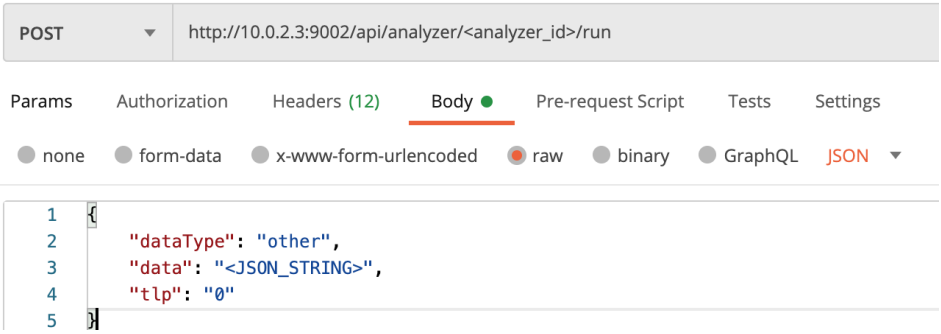


Figure 2-18 - Testing the Analyzers using Cortex API

As a summary, the following requests can be done to the Cortex API:

- List Analyzers/Responders (for example, to obtain the ID of an analyzer):
  GET http://10.0.2.3:9002/api/<analyzer|responder>
- Run Analyzer/Responder:
  POST http://10.0.2.3:9002/api/<analyzer|responder>/<ANALYZER_ID|RESPONDER_ID>/run
- Get Report:
  GET http://10.0.2.3:9002/api/job/<JOB_ID>/waitreport?atMost=1minute

It is worth mentioning that Cortex does not support JSON arrays as response from Analyzers. Consequently, when an Analyzer needs to return an array, it has to be wrapped with a JSON object and put the array in the field “results” under “full”. An example of result returned by a Cortex analyzer is shown in Figure 2-19.

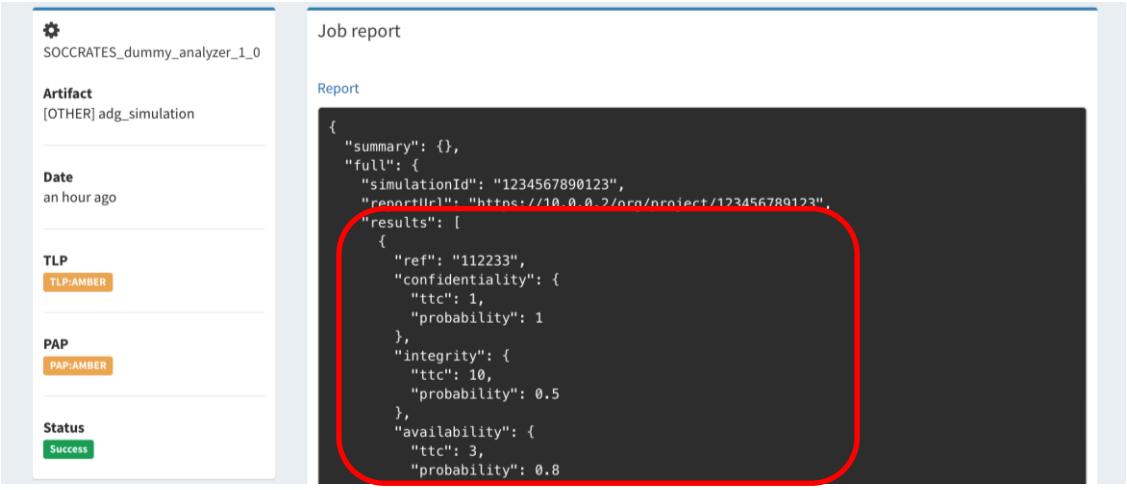


Figure 2-19 - Example of report obtained from a Cortex Analyzer

3 SOCCRATES workflows

3.1 Use Case 1: Response on Detected Ongoing Attack

The Use Case 1 represents the response of the SOCCRATES platform to attacks against the monitored infrastructure and detected by the SIEM or by the AI-based Attack Detection (AAD) component. A detailed description of this use case is in section 4.2 from deliverable D2.1 (Eian, 2019).

The BPMN workflow diagram created in Activiti for the UC1 has been separated by sub processes, in order to differentiate the different stages described in detail in deliverable D2.3 (Frank Fransen, 2021) section 4.2. Each sub process has a set of tasks (user tasks or service tasks), in addition to start and end events.

The first stage is the "Analysis" stage, which is depicted in the Figure 3-1 and corresponds to the section 4.2.2 of deliverable D2.3.

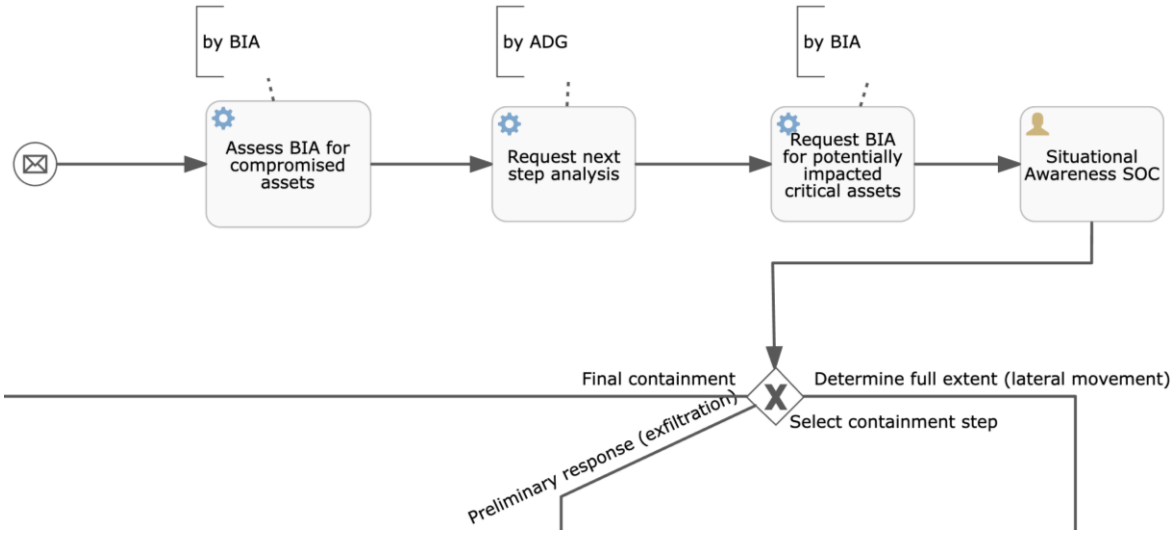


Figure 3-1 - UC1 workflow - Analysis Stage

After the user task "Situational Awareness SOC" there is an Exclusive Gateway, which controls the path of the flow based on the feedback provided by the user in this user task. The three paths connects to 3 different sub processes, one for each stage.

The first sub process is used for the "Preliminary response" phase, which is shown in Figure 3-2 and described in detail in deliverable D2.3 section 4.2.3.

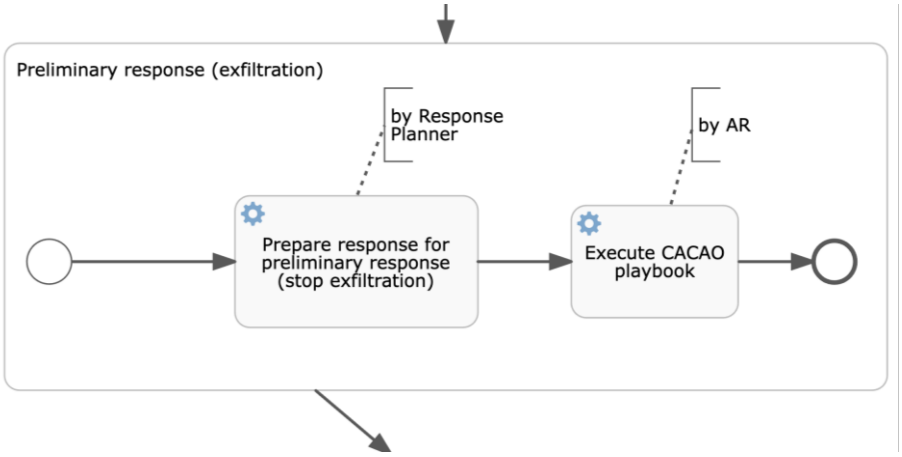


Figure 3-2 - UC1 workflow - Preliminary Response phase

The following sub process of the UC1 workflow is used for the phase "Assess full extent", which is shown in Figure 3-3 and described in section 4.2.4 of deliverable D2.3. This sub process includes a user task to retrieve from the SOC analyst affected IPs that are related to the detected incident.

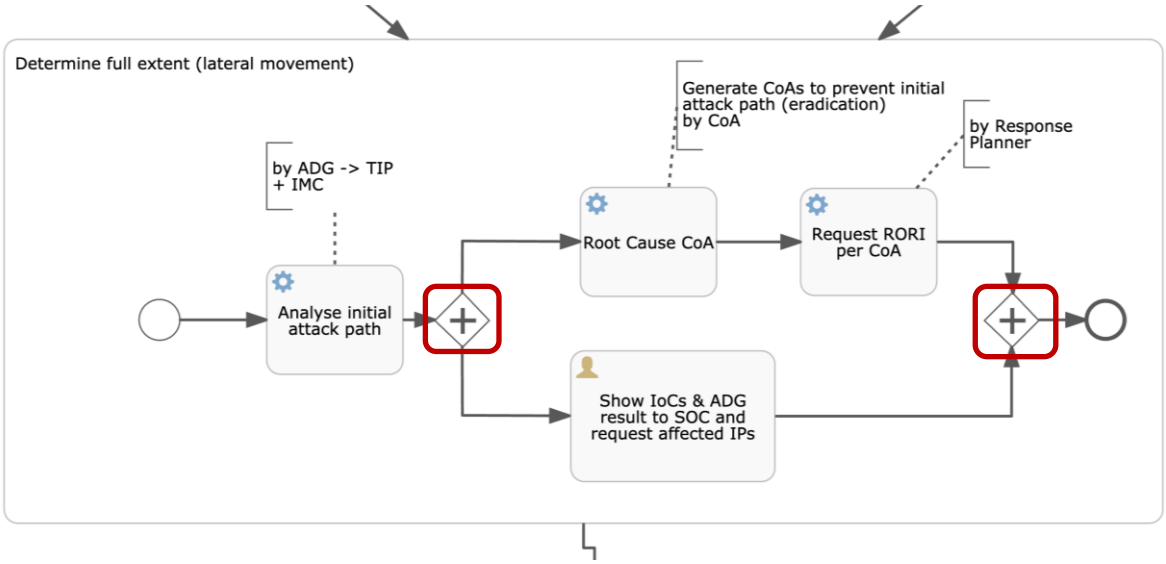


Figure 3-3 - UC1 workflow - Assess full extent

The two elements marked in red in Figure 3-3 are « Parallel gateways », which are used to split the flow into two different paths that are executed in parallel at the same time. It is worth mentioning that the second *parallel gateway* waits for the flows from the different paths before continuing the flow to finish the sub process.

The last sub process is used for the phase « Final containment », which is shown in Figure 3-4 and detailed in section 4.2.5 of deliverable D2.3. After the execution of this sub process the UC1 workflow finishes at the « End » event.

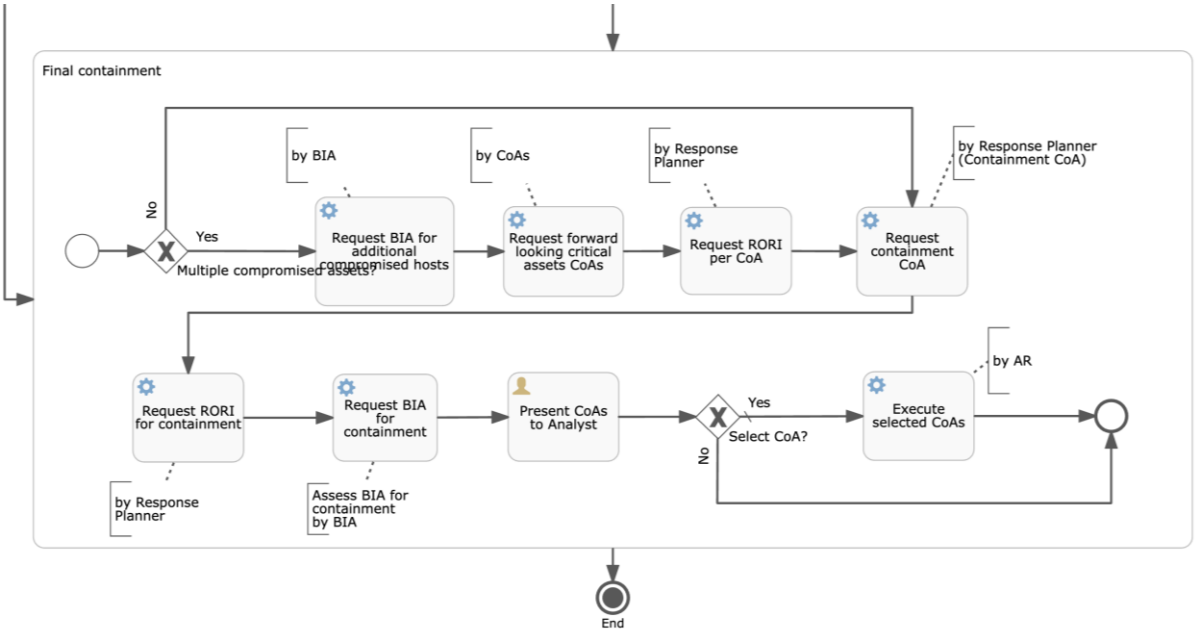


Figure 3-4 - UC1 workflow - Final containment phase

The input message that triggers the use case 1 is "**input_incident_uc1**". This message is sent to the OIE component using the REST API provided as shown below and described with more detail in D2.3:

```
curl --location --request POST 'http://10.0.2.15:5404/activiti-rest/service/runtime/process-instances' \
--header 'Authorization: Basic U0lFTV92dGY6dnRmLnNvY2NyYXRlcy4yMDIx' \
--header 'Content-Type: application/json' \
--data-raw '{
    "message": "input_incident_uc1",
    "variables": [
        {
            "name": "input_incident_uc1",
                  "value":  "{\"Time\":\"\",\"Title\":\"\",\"Category\":\"\",\"Urgency\":\"\",\"Source\":\"\",\"Destination\":\"\"}"
        }]}'
```

Figure 3-5 - UC1 input triggering message

Following table summarizes the analyzers and responders involved in this use case with their functionalities:

Table 3-1 – Summary of Cortex analyzers and responders involved in UC-1

Analyzer/Responder	Component	Function
Analysis stage		
BIA.3	Business Impact Analyzer	Assess BIA for compromised assets
ADG.1	ADG Analyzer	Request next step analysis
BIA.3	Business Impact Analyzer	Request BIA for potentially impacted critical assets
Preliminary response stage		

RP.1	Response Planner	Prepare response for preliminary response (stop exfiltration)
AR.1	Automated Reconfiguration	Execute CACAO playbook
Assess full extent stage		
ADG.1	ADG Analyzer	Analyse initial attack path
CoA.1	CoA Generator	Root Cause CoA
RP.2	Response Planner	Request RORI per CoA
Final containment stage		
BIA.3	Business Impact Analyzer	Request BIA for additional compromised hosts
CoA.1	CoA Generator	Request forward looking critical assets CoAs
RP.2	Response Planner	Request RORI per CoA
RP.1	Response Planner	Request containment CoA
RP.2	Response Planner	Request RORI for containment
BIA.3	Business Impact Analyzer	Request BIA for containment
AR.1	Automated Reconfiguration	Execute selected CoAs

3.2 Use Case 2: Response on Reception of New Cyber Threat Intelligence

Use Case 2 is one of the use cases defined in SOCCRATES to increase the cyber resilience which is focused on the response that need to be triggered when it is received in the SOCCRATES platform a new Indicator of Compromise through the Threat Intelligence Platform. The diagram with the workflow included in this Use Case 2 is shown in Figure 3-6, which was created using the Activiti workflow editor, introduced in deliverable D6.1 section 2.3.1.

This BPMN workflow diagram also includes the Use Cases 3 to 5 since all the tasks included in this use case to mitigate threats are also in the other ones.

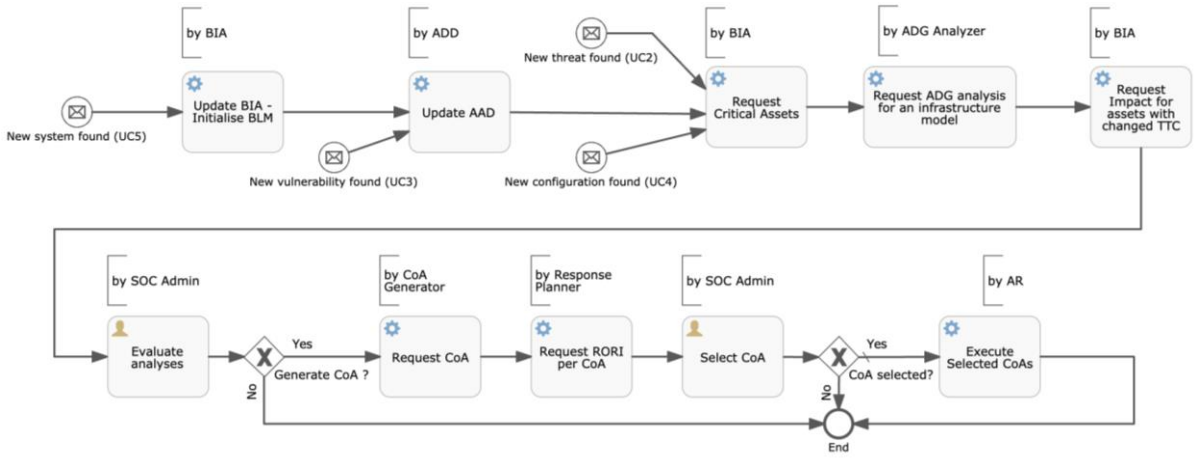


Figure 3-6 - UC2-5 Workflow

The input message that triggers the use case 2 is "**input_threat_uc2**". This message is generated by the TIP when receives some cyber threat notification sent from external CTI sources

or from the Threat Data Collection & Threat Prediction component, operated externally to the SOCCRATES Platform.

An example of the request done by TIP to the Activiti API to trigger the use case is shown below:

```
curl --location --request POST 'http://10.0.2.3:8080/activiti-rest/service/runtime/process-instances' \
--header 'Authorization: Basic <token>' \
--header 'Content-Type: application/json' \
--data-raw '{
    "message": "input_threat_uc2",
    "variables": [
        {
            "name": "input_threat_uc2",
            "value": "                    {\"Content\":\"9334c5c75c7639d2db62091242226924783ba61ae3fe5d5895dd5fa97e45c898\", \"Type\":\"vulnerability\", \"ID\": \"cve-2010-0738\", \"Date\": \"2021-11-04T13:45:05.098Z\"}"
        }]}'
```

Figure 3-7 - UC2 input triggering message

In section 4.3 of deliverable D2.3, it is described step by step the details of the workflow followed when the use case 2 is invoked.

Following table summarizes the analyzers and responders involved in this use case with their functionalities:

Table 3-2 : Summary of Cortex analyzers and responders involved in UC-2 and UC-4

Analyzer/Responder	Component	Function
BIA.1	Business Impact Analyzer	Fetch critical assets from BIA
ADG.1	ADG Analyzer	Request ADG analysis for an infrastructure model (including the infrastructure model before and after the event).
BIA.3	Business Impact Analyzer	Request impact for assets with changed time to compromise (TTC).
CoA.1	CoA Generator	Generate containment CoAs (in case the SOC admin has decided to continue with the response to the incident)
RP.2	Response Planner	Request RORI per CoA
AR.1	Automated Reconfiguration	Responder to send mail with the selected CoA in case it has been selected the SOC user

3.3 Use Case 3: Response on Discovery of New Vulnerable Assets

The main purpose of Use Case 3 is to trigger a response when it is discovered a new vulnerability that can affect some of the assets monitored. The input message that triggers the use case 3 is "**input_vulnerability_uc3**". This message is automatically generated by the Infrastructure Modelling Component when it is created an "affects" fact connecting a vulnerability object with a software object, meaning that a vulnerability has been discovered in a software which is installed in an asset.

An example of the request done by the IMC to the Activiti API to trigger the use case is shown below:

```
curl --location --request POST 'http://10.0.2.3:8080/activiti-rest/service/runtime/process-instances' \
--header 'Authorization: Basic <token>' \
--header 'Content-Type: application/json' \
--data-raw '{
    "message": "input_vulnerability_uc3",
    "variables": [
        {
            "name": "input_vulnerability_uc3",
            "value": " {\"Host\": \"\",\"Software\": \"Mozilla Firefox 88.0 (x64 en-US):88.0\", \"Vulnerability\": \"cve-2010-0738\", \"Date\":\"2021-11-04T13:45:05.098Z\"} "
        }]}'
```

Figure 3-8 - UC3 input triggering message

Following table summarizes the analyzers and responders involved in this use case with their functionalities:

Table 3-3 - Summary of Cortex analyzers and responders involved in UC-3

Analyzer/Responder	Component	Function
AAD.1	AAD Analyzer	Update AAD with information of the new vulnerable asset found in the infrastructure.
BIA.1	Business Impact Analyzer	Fetch critical assets from BIA
ADG.1	ADG Analyzer	Request ADG analysis for an infrastructure model (including the infrastructure model before and after the event).
BIA.3	Business Impact Analyzer	Request impact for assets with changed TTC
CoA.1	CoA Generator	Generate containment CoAs (in case the SOC admin has decided to continue with the response to the incident)
RP.2	Response Planner	Request RORI per CoA
AR.1	Automated Reconfiguration	Responder to send mail with the CoA in case this has been selected by the SOC admin

3.4 Use Case 4: Response on Discovery of System Configuration Change

The Use Case 4 is triggered when a change in the configuration of the assets is detected by the Infrastructure Modelling Component. The input message that triggers this use case is "**input_configuration_uc4**". The workflow followed when the message is received by the OIE is the same to the use case 2. Consequently, the same analyzers and responders included in Table 3-2 are used in this use case. The main difference is in the information provided in the input message that will be used in the ADG analysis.

An example of the request done by the IMC to the Activiti API to trigger the use case is shown below:

```
curl --location --request POST 'http://10.0.2.3:8080/activiti-rest/ser-
vice/runtime/process-instances' \
--header 'Authorization: Basic <token>' \
--header 'Content-Type: application/json' \
--data-raw '{
    "message": "input_configuration_uc4 ",
    "variables": [
        {
            "name": "input_configuration_uc4",
            "value": "{\"HostName\": \"www.example.com\", \"Date\": \"2021-11-
04T13:45:05.098Z\"}"
        }]}'
```

Figure 3-9 - UC4 input triggering message

3.5 Use Case 5: Response on Deployment of New Systems in Infrastructure

Use Case 5 focuses on providing a response when a new asset is added to the monitored infrastructure and detected by the Infrastructure Modelling Component. The input message that triggers the use case 5 is "**input_system_uc5**".

An example of the request done by the IMC to the Activiti API to trigger the use case is shown below:

```
curl --location --request POST 'http://10.0.2.3:8080/activiti-rest/ser-
vice/runtime/process-instances' \
--header 'Authorization: Basic <token>' \
--header 'Content-Type: application/json' \
--data-raw '{
    "message": "input_system_uc5",
    "variables": [
        {
            "name": "input_system_uc5",
            "value": "{\"HostName\": \"www.example.com\", \"Date\": \"2021-11-
04T13:45:05.098Z\"}"
        }]}'
```

Figure 3-10 - UC5 input triggering message

Following table summarizes the analyzers and responders involved in this use case with their functionalities:

Table 3-4 - Summary of Cortex analyzers and responders involved in UC-5

Analyzer/Responder	Component	Function
BIA.2	BIA	Update BIA model and initialize Business Logic Model
AAD.1	AAD Responder	Update AAD with information of the new assets added to the infrastructure
BIA.1	Business Impact Analyzer	Fetch critical assets from BIA
ADG.1	ADG Analyzer	Request ADG analysis for an infrastructure model (including the infrastructure model before and after the event).
BIA.3	Business Impact Analyzer	Request impact for assets with changed TTC
CoA.1	CoA Generator	Generate containment CoAs (in case the SOC admin has decided to continue with the response to the incident)
RP.2	Response Planner	Request RORI per CoA
AR.1	AR	Responder to send mail with the CoA in case this has been selected by the SOC admin

4 SOCCRATES Platform demonstrator

4.1 Triggering workflow for UC5

To demonstrate the actual version of the SOCCRATES Platform the workflow for the Use Case 5 has been chosen, since it has common steps with other UCs (2 to 4) and involves more tasks than the others. To initiate the workflow of the UC5, we make a request to the Activiti API sending the corresponding message, which simulates the behaviour of the Infrastructure Modelling Component when a new asset is found in the infrastructure.

To perform the request to the Activiti API we use Postman[17], which provides a graphical tool to store and execute requests to APIs. The Figure 4-1 shows the result of executing the request that triggers the workflow for the UC5.

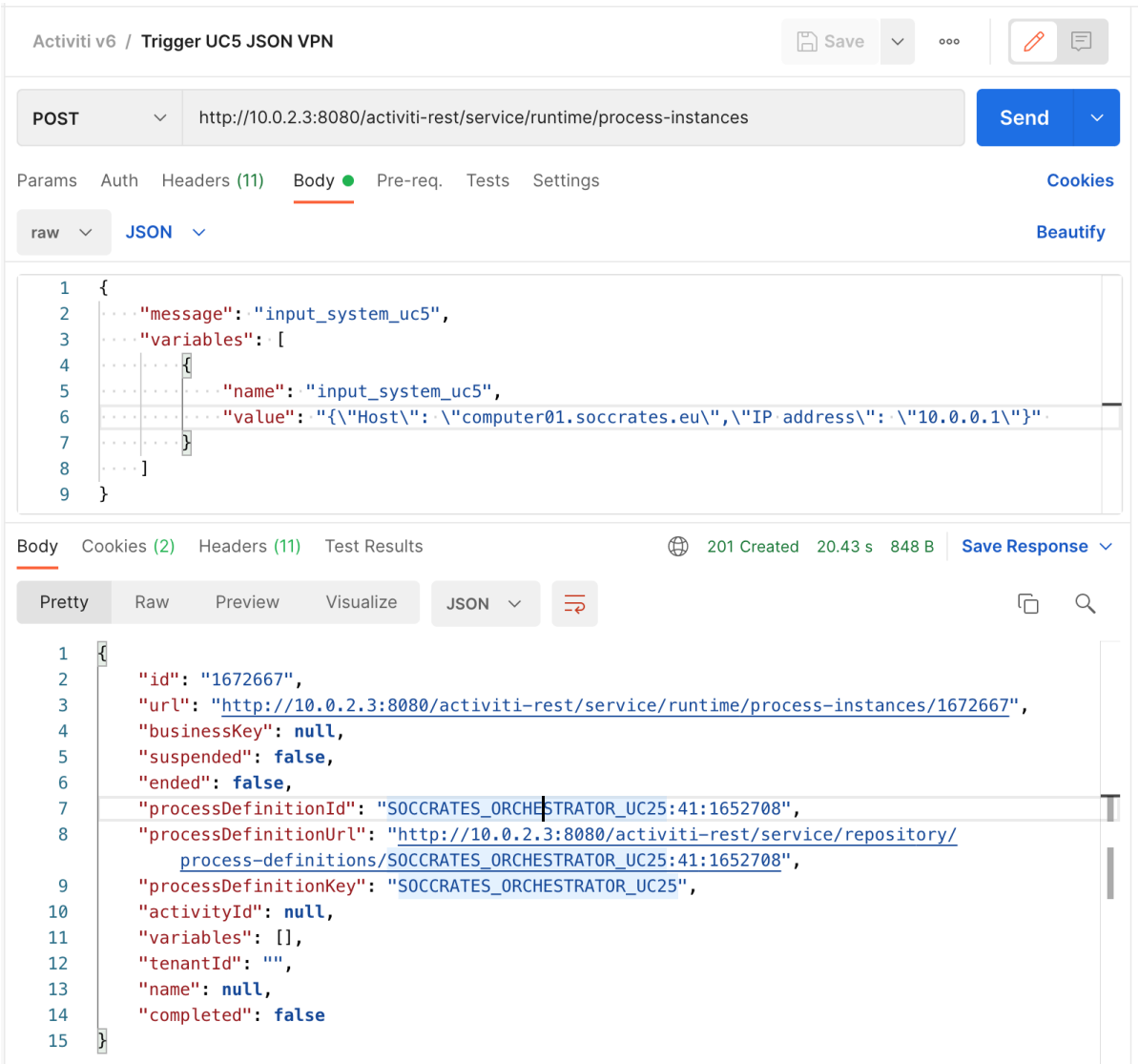


Figure 4-1 - Request to Activiti API for triggering UC5

[7] https://www.postman.com/

Classification level: Public

The information relative to the situation that triggers the workflow is stored under field "value" as a JSON string.

The data format of the response is defined by Activiti and is the same for all use cases. When the execution is successful, it returns a HTTP code "201 Created" with information about the execution, which is not really relevant for the triggering entity (for the IMC in this case), apart of the returned HTTP code.

Once triggered, Activiti creates a new "Process Instance", which refers to a workflow execution. In the Activiti UI we can see the list of running processes, as shown in Figure 4-2.

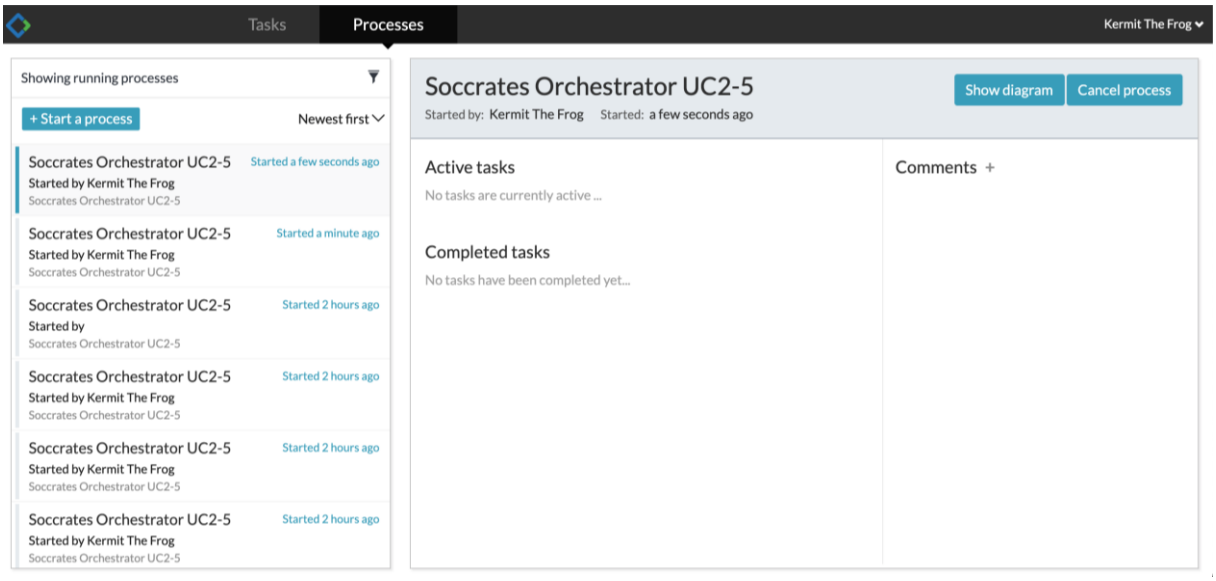


Figure 4-2 - Activiti UI - Running processes

In the list, we see the name of the running processess, that is "Soccrates Orchestrator UC2-5". As described in section 3.2, there is a single workflow for the use cases from 2 to 5. As consequence, we see in Activiti the name of the workflow, but not the refered use case. This is something that will be addressed in the Orchestrator UI, where it will be easily distingable the use case of each executed workflow.

By clicking in the button « Show diagram » a popup is shown with a diagram depicting the actual execution status, as shown in Figure 4-3. In this diagram, we can see the input message that started the workflow marked with a blue border. If we check the workflow in the Activiti workflow editor, we can see it refers to the input message for the UC5 (see Figure 3-10).

The actual service task being executed is outlined with a green border. In Figure 4-3 it can be appreciated that it is being executed the service task « Update BIA – Initialise BLM », which connects to the Business Impact Analyzer component through its Cortex Responder to intialise the Business Logic Model.

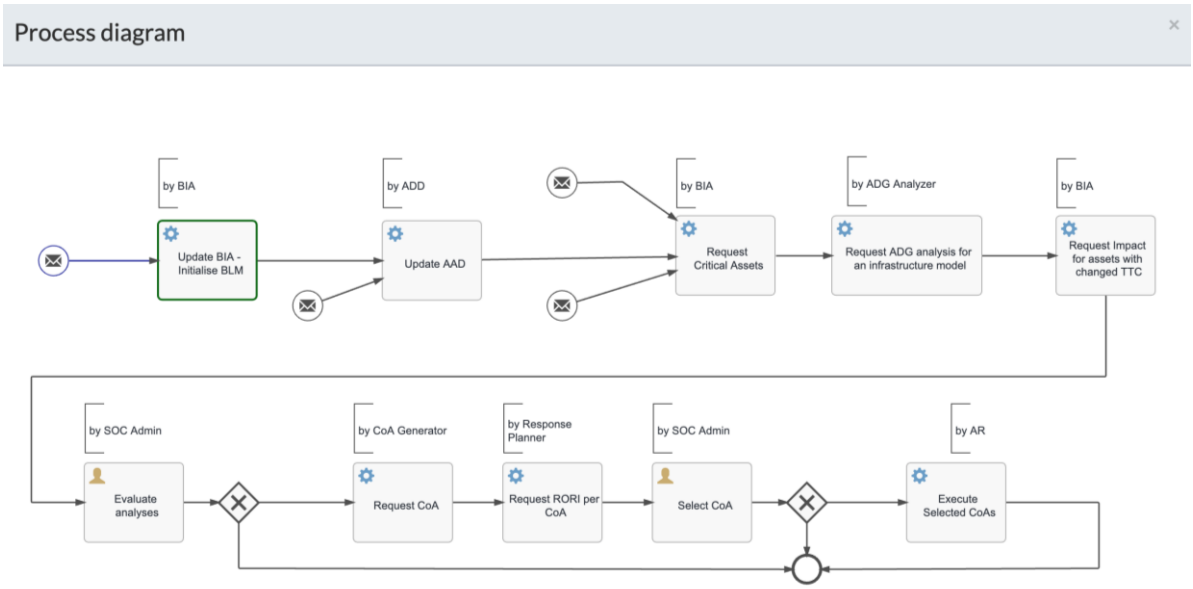


Figure 4-3 - UC5 process execution status diagram

4.2 Service task execution

In the Ochestrator Core terminal output we can see the traces of the execution of the different service tasks involved in the UC5 workflow. In Figure 4-4, they are shown the traces (logs) of the orchestrator core that detail some of the steps and results of each service task execution. In red, it is marked the list of the execution variables, which are available for every service task of the workflow execution. This endows the service tasks with the capability of sharing data between them.

```
================================================================================
class soccrates.atos.orchestrator.serviceTasks.impactAnalyser.InitialiseBLM started....
Invoking Cortex: bia_updateModel
Response code: 200
[input_system_uc5, input_message, UC]
class soccrates.atos.orchestrator.serviceTasks.impactAnalyser.InitialiseBLM finished!

================================================================================
class soccrates.atos.orchestrator.serviceTasks.aad.UpdateAADModel started....
Hibernate:
    insert
    into
        orchestrator_log
        (created, data, execution_id, class, process_instance_id, task_id, type, use_case)
    values
        (?, ?, ?, ?, ?, ?, ?, ?)
[input_system_uc5, input_message, UC]
class soccrates.atos.orchestrator.serviceTasks.aad.UpdateAADModel finished!

================================================================================
```

Figure 4-4 - Service Task execution log

When the first service task of the workflow is executed (i.e. : InitialiseBLM), there are only 3 variables available : « input_system_uc5 », « input_message » that stores the data of the triggering message, and the « UC » variable that stores the user case number being executed. Those variables are used for the different services tasks to build the requests for the Cortex Analyzers and Responders and, in some cases, to behave differently depending on the use case.

4.3 Orchestrator to Cortex Analyzer / Responder

As described in section 2.2.1.2, the service tasks communicate with other SOCCRATES components through Cortex Analyzers and Responders. For that, each service task prepares the request to the corresponding Cortex Analyzer or Responder using the data extracted from the execution variables.

There are two ways to monitor the interaction between the Orchestrator and Cortex. The first one, is using the Cortex UI, which provides a history of the executed jobs (i.e. : calls to Analyzers/Responders), where the user can navigate through them and check the request and response data (the job report). The Figure 4-5 shows the Jobs History view from the Cortex UI, where we can see the status of the job and the input data.

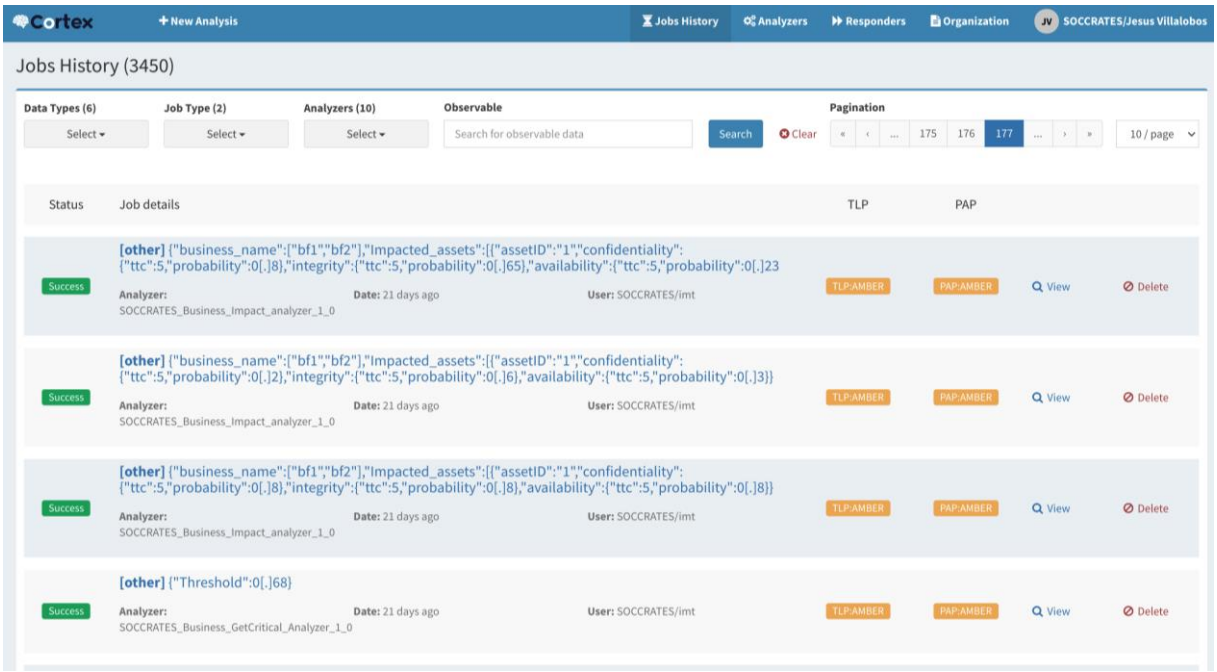


Figure 4-5 - Cortex UI - Jobs History

By clicking in the « View » button we open the job report, shown in the Figure 4-6. Cortex, stores the result of the Responder/Analyzer in the field « full » of the report JSON.

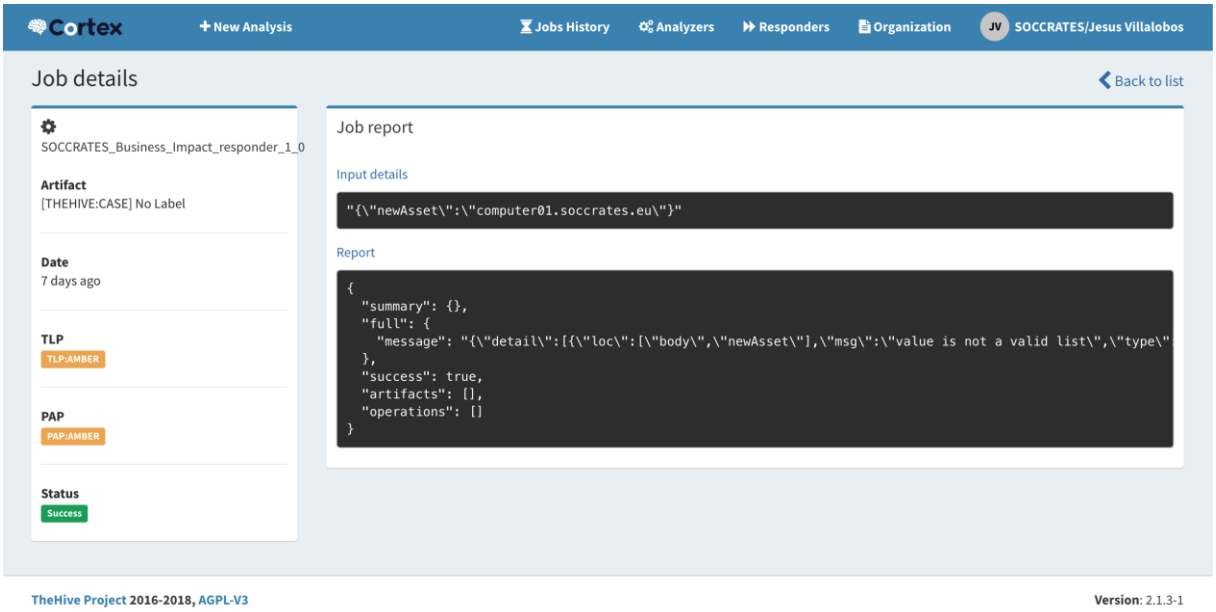


Figure 4-6 - Cortex UI - Responder Job Report

The second way to monitor the communications between the Orchestrator and Cortex is checking the database logging table, described in section 2.2.3. When the corresponding service task performs the request to Cortex, it stores the request data in the field « data » of the logging table using JSON format. Additionally, when it gets the response from Cortex, it also stores the returned data in a new entry. In the Figure 4-7 we see an example of the entries created after the execution of the whole UC5.

In the Figure 4-7, the entry corresponding to the request performed by the first service task executed is marked in red, the « initialise_blm » service task. Since Cortex Responders are not expected to return any usable data, there is no entry for the response of this call.

data	task_id	type	class	use_case
{"mustGenerateCoA":"true","coas":"[1]","adg_analysis":"{...		UC_end	class soccrates.atos.orchestrator.listeners.UCListener	5
{"mustGenerateCoA":"true","coas":"[1]","adg_analysis":"{...	select_coa	task_user_end	class soccrates.atos.orchestrator.listeners.UserTaskListener	5
{"adg_analysis":"{\"simulationId\":\"307666084811239...	select_coa	task_user_start	class soccrates.atos.orchestrator.listeners.UserTaskListener	5
{"coas":[{"efficiency":"170.731","coaTTC":{"228.Read":[1...	request_coa	task_service_response	class soccrates.atos.orchestrator.model.CoAResponse	5
soccrates.atos.orchestrator.model.CoARequest@72224107	request_coa	task_service_request	soccrates.atos.orchestrator.model.CoARequest	5
{"adg_analysis":"{\"simulationId\":\"307666084811239...	evaluate_analyses	task_user_end	class soccrates.atos.orchestrator.listeners.UserTaskListener	5
{adg_analysis={"simulationId":"307666084811239","re...	evaluate_analyses	task_user_start	class soccrates.atos.orchestrator.listeners.UserTaskListener	5
{"business_impacts":[{"business_name":"bf1","impact_pr...	impact_analyzer_assets	task_service_response	soccrates.atos.orchestrator.model.ImpactAnalysisBusinessResponse	5
{"business_name":["bf1","bf2","bp1","Company"],"attack...	impact_analyzer_assets	task_service_request	soccrates.atos.orchestrator.model.ImpactAnalysisBusinessRequest	5
{"simulationId":"307666084811239","reportUrl":"https:...	adg_analysis	task_service_response	class soccrates.atos.orchestrator.model.ADGAnalysisResponse	5
{"date":"2021-11-03","entryPoints":[{"ref":"INTERNET","t...	adg_analysis	task_service_request	soccrates.atos.orchestrator.model.ADGAnalysisRequest	5
{"critical_assets":["a","b","d"],"report_url":"https://10.0....	critical_assets	task_service_response	class soccrates.atos.orchestrator.model.BIA_CriticalAssetsResponse	5
{"Threshold":0.6}	critical_assets	task_service_request	class soccrates.atos.orchestrator.model.BIA_CriticalAssetsRequest	5
{"host":"10.0.0.1","type":"newAsset","timestamp":"2021...	update_aad	task_service_request	soccrates.atos.orchestrator.model.AADNewSystemRequest	5
{"newAsset":["computer01.soccrates.eu"]}	initialise_blm	task_service_request	soccrates.atos.orchestrator.model.UpdateBIARequest	5
{"input_system_uc5":{\"Host\":\"computer01.soccrate...		UC_start	class soccrates.atos.orchestrator.listeners.UCListener	5

Figure 4-7 - Use Case traces in logging table

In the case of a service tasks that uses Cortex Analyzers, we can find request and response entries, like the example marked in green in the Figure 4-7, corresponding to the execution of the « adg_analysis » service task.

4.4 User task execution

While progressing the execution of the workflow, the different tasks are executed and the calls to the components through Cortex Analyzers and Responders are performed.

When the workflow arrives to a **user task**, it waits until a user completes it. In the Figure 4-8, it is shown the « Process diagram » in this status, where the user task box is outlined with a green border.

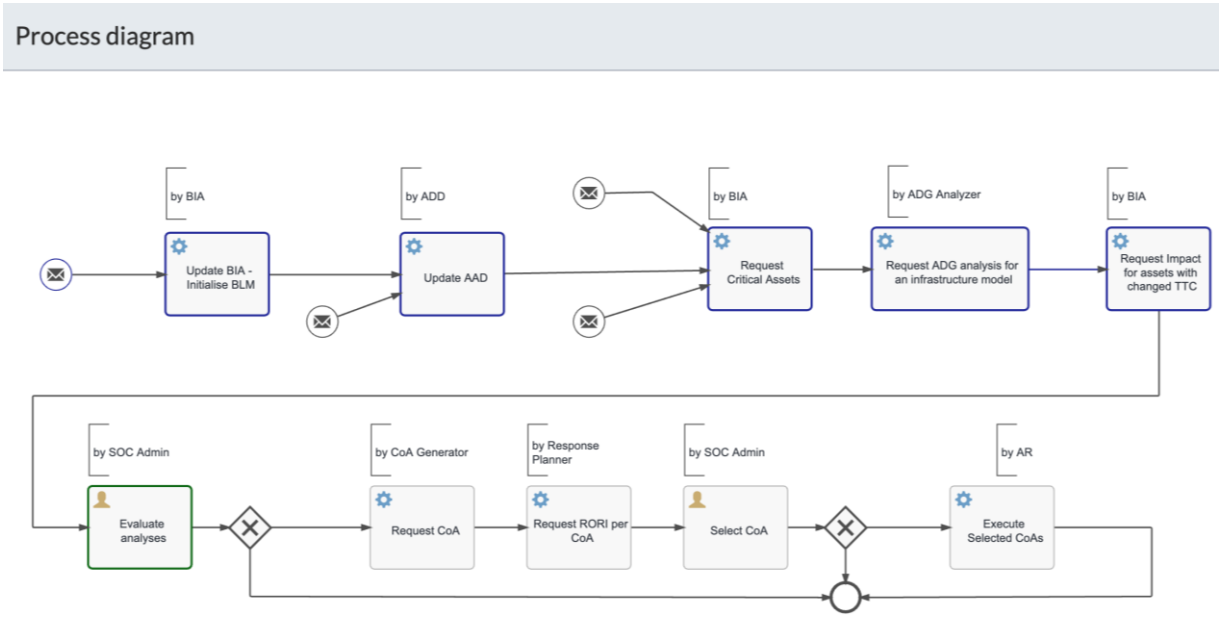


Figure 4-8 - UC5 execution - Process Diagram

In the user tasks, information related with the use case is shown to the user to help him to take the decision. This information is obtained from the execution variables, like the *triggering message* to show the related asset, or the responses from the Cortex Analyzers. Additionally, it is also needed to have an input variable to store the user decission taken based on the presented information.

Those variables and parameters are specified in the Activiti workflow editor as « Form properties ». In the Figure 4-9, it is shown the configured « Form properties » for the User Task « Evalute analyses », which is meant to show to the user the results from the ADG Analysis and the BIA and get feedback about if the system should generate Course of Actions for the presented situation or not. In the field « Variable » of the Figure 4-9, it is set the execution variable where this data is taken from. In the case of the « mustGenerateCoA » form property, this field « Variable » is empty, since it is used to store the user input and not as output to present information in the form.

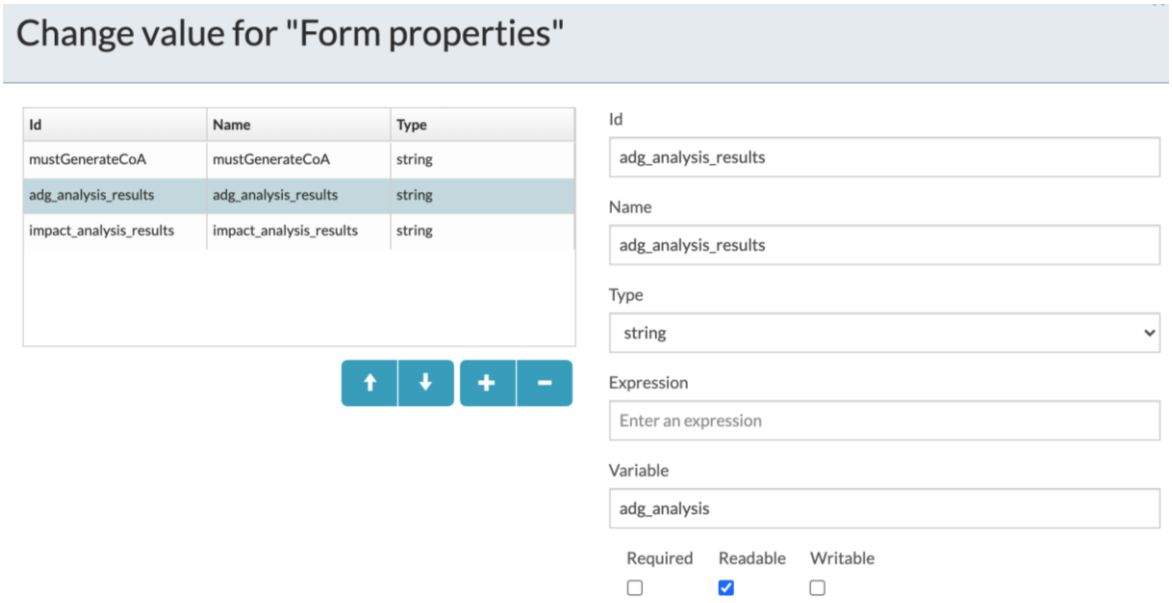


Figure 4-9 - User task form properties

Activiti provides a form builder for the user tasks, which is really limited in the Community version and does not cover the needs of the SOCCRATES Orchestrator. To address it, it is foreseen to include a « user task viewer » in the SOCCRATES Orchestrator Web Frontend where these forms will be presented to the user to get his/her feedback.

For the actual version of the Orchestrator, we can simulate the interaction between the Orchestrator UI and Activiti user tasks using its API, with which we can retrieve the user task form properties and then send back the user decision. In the Figure 4-10, it is shown the retrieval of the form properties for building the form related to the user task « Evaluate analysis » (see Figure 4-8). A JSON with the form properties is returned, with the values of the execution variables defined previously for them.

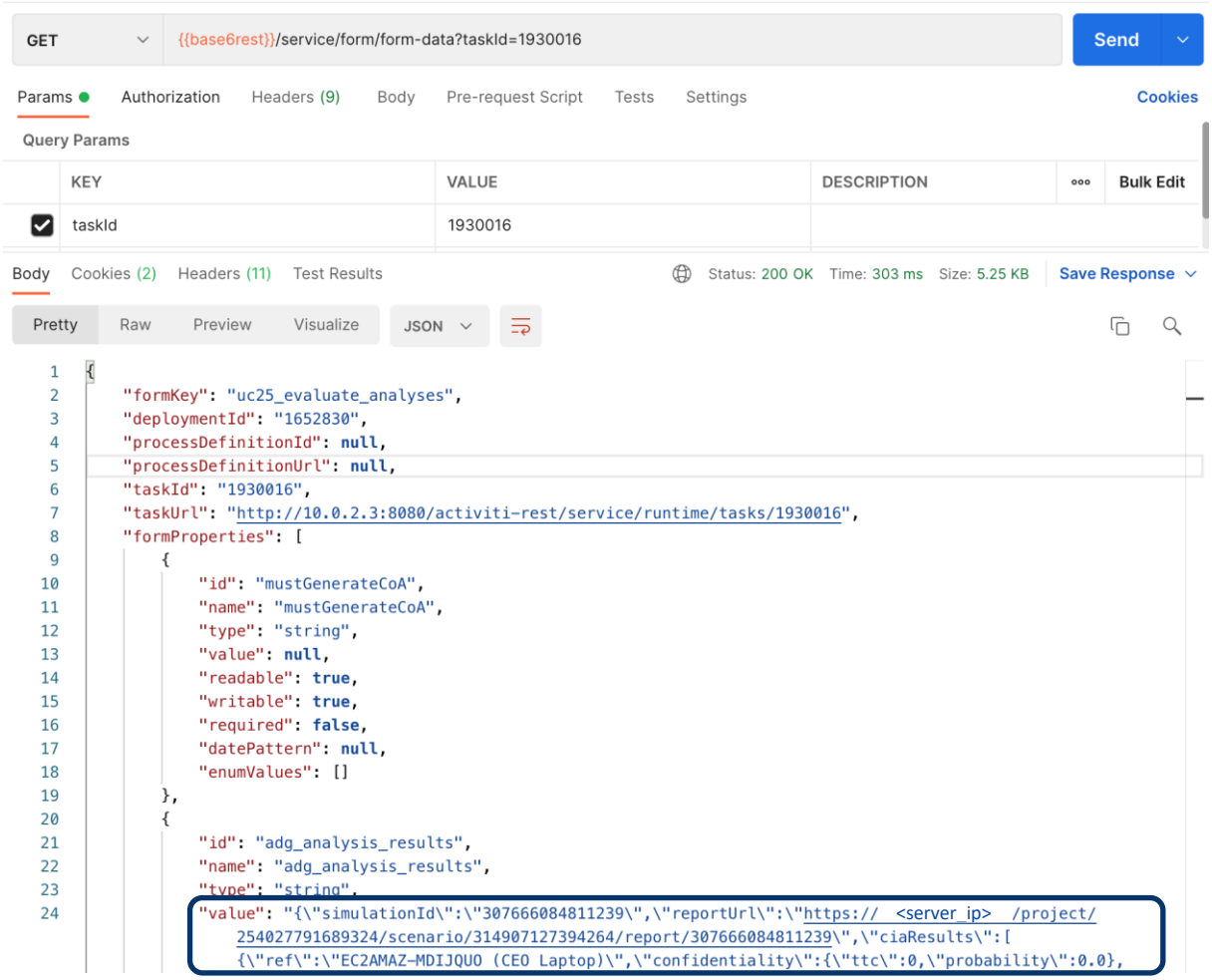


Figure 4-10 - Retrieving form data using Activiti API

To send the user feedback, we make a request to the Activiti API specifying a value to the form property « mustGenerateCoA » :

```
curl --location --request POST 'http://10.0.2.3:8080/activiti-rest/ser-
vice/form/form-data' \
--header 'Cache-Control: no-cache' \
--header 'Content-Type: application/json' \
--data-raw '{
  "taskId" : "1930016",
  "properties" : [
    {
      "id" : "mustGenerateCoA",
      "value" : true
    }
  ]
}'
```

Then, Activiti stores the given value as an execution variable that can be used in further service tasks.

4.5 Exclusive Gateways

The exlusive gateways in BPMN are elements that controls the path to be taken during the execution of the workflow. In the Figure 4-11 it is shown the status of the workflow after the user decision. Marked in red it is one of the exclusive gateways that uses the value of the variable « mustGenerateCoA » (based on the user decission) to continue the workflow to the next task or finishing it.

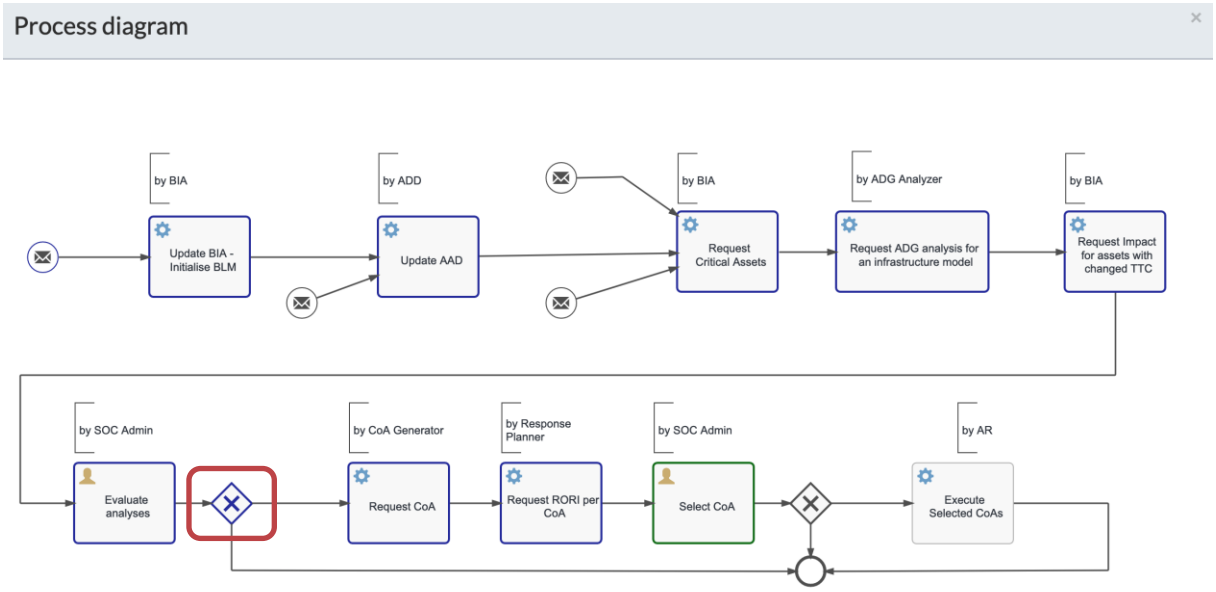


Figure 4-11 - Process diagram - Exclusive Gateway

In the Figure 4-12, it is shown the configured « Flow condition » that evaluates the path « Yes » to continue the workflow to the next task.

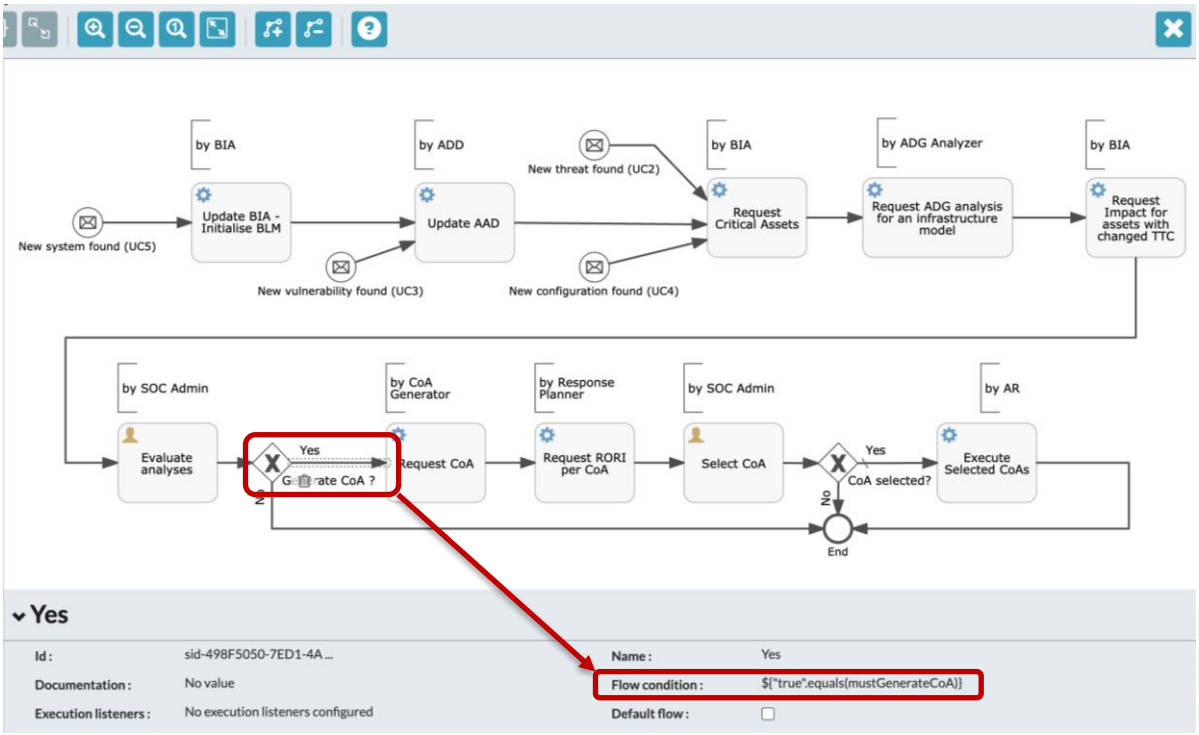


Figure 4-12 - Exclusive gateway - Path flow condition

The condition can be summarized as: if the variable *mustGenerateCoA* is equal to "true", then return true and continue the workflow execution for this path.

4.6 Automatic Response

Continuing with the execution of the workflow of the UC5, the other service tasks and user tasks are executed in a similar way as described before. In Annex 9, it has been included all the requests and responses between the different service tasks included in the UC5 workflow.

It is worth describing the last service task, which takes actions based on the selected CoA. This task uses the Cortex Responder named « Mailer_1_0 » to send an email to the SOC Analyst with the description of the selected CoA, whithin some information about the triggering message and a link the the tool.

Figure 4-13 - Automatic Response - Email example

Additionally, it is foreseen that this task is extended to include other actions via Cortex Responders, like opening a ticket in a ticketing system such as Jira, or doing changes in the monitored infrastructure by executing OpenC2 commands. Those actions would require to perform new requests to other Cortex Responders from the same task, which would determine what actions should be taken depending on the configuration and the selected Course of Action.

5 Summary and Next Steps

This deliverable has presented the initial outcome of task T6.2, describing the results obtained related to the development of the SOCCRATES Orchestrator & Integration Engine and the integration of the components in an initial version of the SOCCRATES Platform. It has been described the development of the Orchestrator Core, which is in charge of the execution of the different tasks of the BPMN workflows. Also it has been described the creation of the two BPMN workflows for covering the different use cases, one of them covering the UC2 to UC5 which have common tasks and differentiating the UC by the start event messages. Additionally it has been described the integration of the components in the SOCCRATES platform through Cortex Analyzers and Responders for the different Use Cases, besides the development of the first prototype of the platform available for testing in pilots. Finally, it has been presented a demonstration of the execution of the UC5 in the Orchestrator within a detailed explanation of the most significant elements involved on it.

In the next deliverable D6.3 ("SOCCRATES Testing and Validation Report"), to be delivered in M29, it will be reported the testing and verification results of the SOCCRATES Platform, including the evaluation of the functionality of individual components.

The deliverable D6.4 ("Final version of the SOCCRATES Platform"), to be reported in M34, will include the pending tasks to be completed by the last 12 months of the project. This includes the development of the SOCCRATES web frontend which integrates the UI of the different components in a single portal, besides the integration of Keycloak as authentication and authorization provider, to be used by all the tools incorporated in the SOCCRATES web frontend. Additionally, this deliverable will report the adaptation of the Orchestrator Logging System and UI for the obtainment of metrics that will be collected from the pilots. Moreover, it will include the pending integration of the Automated Reconfiguration Cortex Responder (AR.2), which is being defined based on pilots capabilities. Finally, this deliverable will describe the deployment of the SOCCRATES Orchestrator and Integration Engine in the pilots and in the validation environment, within a description of the adaptations required for this process.

6 References

Alkkiomäki, M. K. (2021). *D3.1 Initial ADG based Attack prototypes.* European Commission.

Chiscop, I. (2020). *D4.1 AI-based Attack Detection to Detect Advanced Threats.* European Commission.

Eian, M. (2019). *D2.1 Use cases definition & pilot sites requirements.* European Commission.

Frank Fransen, S. T. (2021). *SOCCRATES D2.3 System Architecture & Interface Specification Final version .* European Commission.

Kiennert, C. (2021). *D5.2 Business Logic Modelling and Impact Analyser & Response Planner - Initial Prototype.* European Commission.

Kijewski, P. (2020). *D4.2 Threat Identification and Threat Trend Prediction - Initial Prototype.* European Commission.

Trapero, R. (2020). *D6.1 Initial version of the SOCCRATES Platform Orchestration, Reconfiguration and Front-end.* European Commission.

7 Abbreviations

This glossary serves as inventory of abbreviations used in the document.

Acronym	Description
AAD	AI based Attack Detection
ACT	semi-Automated Cyber Threat intelligence
ADG	Attack Defence Graph
AEF	Argus Event Format
AI	Artificial Intelligence
AIT	AIT Austrian Institute of technology
API	Application Programming Interface
APT	Advance Persistent Threat
AR	Automated Reconfigration
ATOS	ATOS Spain
AV	AntiVirus
BIA	Business Impact Analyser
BLM	Business Logic Modelling
BPMN	Business Process Model and Notation
CACAO	Collaborative Automated Course of Action Operations
CC	Command and Control
CERT	Computer Emergency Response Team
CMDB	Configuration Management Database
CSIRT	Computer Security Incident Response Team
CoA	Course of Action
CTI	Cyber Threat Intelligence
DC	DataCentre
DGA	Domain Generated Algorithm
DNS	Domain Name System
EDR	Endpoint Detection and Response
ELK	Elasticsearch/Logstash/Kibana
FRS	Foreseeti
FSC	F-secure
FTP	File Transfer Protocol
ICT	Information and Communication Technology
IDS	Intrusion Detection System
JSON	JavaScript Object Notation
IMC	Infrastructure Modelling Component
IMT	Institut Mines Télécom – Télécom SudParis
INTF	Interface
IoC	Indicators of Compromise
IP	Internet Protocol
IPS	Intrusion Prevention System
IRM	Incident Response and Management
ITIL	Information Technology Infrastructure Library
KTH	Kungliga Tekniska högskolan - Royal Institute of Technology
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol

M_n	Infrastructure Model (at time *n*)
MNM	mnemonic
MSSP	Managed Security Service Provider
MTTD	Mean Time To Detection
NOC	Network Operations Centre
OIE	Orchestrator & Integration Engine
OT	Operational Technology
OS	Operating System
RORI	Return on Response Investment
RP	Response Planner
SDN	Software Defined Network
SHS	Shadowserver
SIEM	Security information and event management
SOAR	Security Orchestration, Automation and Response
SOC	Security Operation Centre
SOCCRATES	SOC & CSIRT Response to Attacks & Threats based on attack defence graph Evaluation Systems
SSL	Secure Sockets Layer
TAP	Test Access Point
TIP	Threat Intelligence Platform
TLS	Transport Layer Security
TNO	Nederlandse Organisatie voor toegepast natuurwetenschappelijk onderzoek
TTC	Time To Compromise
UC	Use Case
UI	User Interface
UUID	Universally Unique IDentifier
VLAN	Virtual LAN
VM	Virtual Machine
VPN	Virtual Private Network
VTF	Vattenfall

8 ANNEX I : Analyzers & Responders Requests and Responses

8.1.1 Business Impact Analyzer

Table 8-1: Request and response example for BIA.1 Analyzer

Request data	{"Threshold":0.6}
Response data	{ "summary": {}, "full": { "critical_assets": ["a", "b"] }, "success": true, "artifacts": [], "operations": [] }

Table 8-2 - Request and response example for BIA.2 Analyzer

Request data	{ "business_name": ["bf1", "Company"], "Impacted_assets": [{ "assetID": "2", "confidentiality": { "ttc": 5, "probability": 0.8 }, "integrity": { "ttc": 5, "probability": 0.5 }, "availability": { "ttc": 5, "probability": 0.6 } }] }
Response data	{ "summary": {}, "full": { "business_impacts": [{ "business_name": "bf1", "impact_probability": 0.27, "impact_criticality": 8.644990662583584, "critical_time": 10 }, { "business_name": "Company", "impact_probability": 0.24, "impact_criticality": 0.8026005596974961, "critical_time": 18 }], "report_url": "https://10.0.0.2/org/project/123456789" }, "success": true, "artifacts": [], "operations": [] }

	}

Table 8-3 - Request and response example for BIA.3 Responder

Request data	"{\"newAsset\":[\"computer01.soccrates.eu\"]}"

Response data	{
	"summary": {},
	"full": {
	"message": "\"Model updated\""
	},
	"success": true,
	"artifacts": [],
	"operations": []
	}

8.1.2 Respone Planner

Table 8-4 - Request and response example for RP.1 Analyzer

Request data	{
	"siem_label": "TA0001",
	"sourceIPs": ["192.168.0.3", "192.168.0.10"],
	"destinationIPs": ["123.45.67.89", "123.98.76.54"]
	}

Response data	{
	"summary": {},
	"full": {
	"containment": true,
	"coas": [
	{
	"defenses": [
	{
	"ref": "112244",
	"defenseName": "Isolate",
	"defenseInfo": "Network Segmentation",
	"mitreRef": "M1030",
	"hosts": [
	"192.168.0.3",
	"192.168.0.10"
	]
	}
	]
	},
	{
	"defenses": [
	{
	"ref": "112244",
	"defenseName": "Isolate",
	"defenseInfo": "Network Segmentation",
	"mitreRef": "M1030",
	"hosts": [
	"192.168.0.3",
	"192.168.0.10"
	]
	},
	{
	"ref": "112255",
	"defenseName": "Filter",
	"defenseInfo": "Filter Network Traffic",
	"mitreRef": "M1037",
	"flows": [
	{
	"src": "192.168.0.3",
	"dst": "123.45.67.89"
	}
	]
	}
	]
	}

```
    ]
  },
  "success": true,
  "artifacts": [],
  "operations": []
}
```


Table 8-5- Request and response example for RP.2 Responder

Request data


```
{
  "siem_label":"TA0001",
  "initialTTC":{"228.Read":[19,26,34],"325.NetworkRespondConnect":[23,30,40]},
  "containment":false,
  "coas":[
   {
    "coaTTC":{"228.Read":[17976931348623157,17976931348623157,17976931348623157],
              "325.NetworkRespondConnect": [17976931348623157,17976931348623157,17976931348623157]},
    "time_cost":{"1":355,"2":564,"3":22,"4":895,"5":1336},
    "monetary_cost":{"1":87355,"2":564,"3":228750,"4":8955,"5":8788},
    "defenses":[
              {"ref":"119","defenseName":"authenticated","defenseInfo":"Setup authentification","mitreRef":"Ref"},
              {"ref":"100","defenseName":"Disabled","defenseInfo":"Disabled Application","mitreRef":"Ref"}]
   }
   ]
}
```


Response data


```
{
  "summary": {},
  "full": {
   "ranked_coas": [
    {
     "RORI": 6173.048024629743,
     "containment": false,
     "coaTTC": {
      "228.Read": [
       17976931348623156,
       17976931348623156,
       17976931348623156
      ],
      "325.NetworkRespondConnect": [
       17976931348623156,
       17976931348623156,
       17976931348623156
      ]
     },
     "monetary_cost": {
      "1": 87355,
      "2": 564,
      "3": 228750,
      "4": 8955,
      "5": 8788
     },
     "defenses": [
      {
       "ref": "119",
       "defenseName": "authenticated",
       "defenseInfo": "Setup authentification",
       "mitreRef": "Ref"
      },
      {
       "ref": "100",
       "defenseName": "Disabled",
       "defenseInfo": "Disabled Application",
       "mitreRef": "Ref"
      }
     ],
     "business_impact": 0.56,
     "playbook": {
      "type": "playbook",
      "spec_version": "1.0",
      "created_by": "Soccrates_Response_Planner",
      "description": "This playbook will apply the CoA in the system",
```

```
        "playbook_types": [
          "prevention"
        ],
        "features": {
          "parallel_processing": true
        },
        "created": "Thu Oct 28 22:28:16 2021",
        "id": "soccrates-playbook-1635452896070112425",
        "name": "Soccrates CoA Playbook 1635452896070112425",
        "workflow_start": "start--soccrates-playbook-1635452896070112425",
        "workflow": {
          "start--soccrates-playbook-1635452896070112425": {
            "type": "start",
            "on_completion": "parallel--soccrates-playbook-1635452896070112425"
          },
          "parallel--soccrates-playbook-1635452896070112425": {
            "type": "parallel",
            "name": "Apply defenses of the CoA",
            "next_steps": [
              "defense1--soccrates-playbook-1635452896070112425",
              "defense2--soccrates-playbook-1635452896070112425"
            ]
          },
          "defense1--soccrates-playbook-1635452896070112425": {
            "type": "single",
            "name": "authenticated",
            "description": "Setup authentification",
            "on_completion": "end--soccrates-playbook-1635452896070112425",
            "target": "119",
            "commands": [
              {
                "type": "manual",
                "command": "Setup authentification on 119"
              }
            ]
          },
          "defense2--soccrates-playbook-1635452896070112425": {
            "type": "single",
            "name": "Disabled",
            "description": "Disabled Application",
            "on_completion": "end--soccrates-playbook-1635452896070112425",
            "target": "100",
            "commands": [
              {
                "type": "manual",
                "command": "Disabled Application on 100"
              }
            ]
          },
          "end--soccrates-playbook-1635452896070112425": {
            "type": "end"
          }
        }
      },
    }
  ],
  "report_url": "10.8.0.4:8001/1635452896070197593"
 },
 "success": true,
 "artifacts": [],
 "operations": []
}
```


8.1.3 Attack Defense Graph

Table 8-6- Request and response example for ADG.1 Analyzer

Request data	{ "threatActor": "APT-28", "entryPoints": [{

	"ref": "0a6b636c-3dc8",
	"type": "UserExecution"
	}
	],
	"highValueAssets": [
	{
	"ref": "5c3b425c-1dc5",
	"type": "DataDestruction"
	}
	],
	"vulnerabilities": [
	{
	"ref": "6749be31-f8aa",
	"type": "CVE-2021-26427"
	}
	]
	}

Response data	{
	"simulationId": "123456789123",
	"reportUrl": "https://soc.securicad.com/project/75463/report/123456789123",
	"ciaResults": [
	{
	"ref": "5c3b425c-1dc5",
	"confidentiality": {
	"ttc": "inf",
	"probability": 0
	},
	"integrity": {
	"ttc": "12",
	"probability": 0.75
	},
	"availability": {
	"ttc": "12",
	"probability": 0.75
	}
	}
	],
	"ciaResultsDelta": [
	{
	"ref": "5c3b425c-1dc5",
	"confidentiality": {
	"ttc": "0",
	"probability": 0
	},
	"integrity": {
	"ttc": "0",
	"probability": 0
	},
	"availability": {
	"ttc": "0",
	"probability": 0
	}
	}
	],
	"results": { intentionally omitted }
	}

8.1.4 Course of Action

Table 8-7 - Request and response example for CoA.1 Analyzer

Request data	{
	"simulationId": "123456789123"
	}

Response data	{
	"initialTTC": {
	"48639a8b-8fcd-4802-af3f-f8f28cd24901.Read": [
	20,
	23,
	35

```
      ],
      "58639a8b-8fcd-4802-af3f-f8f28cd24901.Eavesdrop": [
        23,
        30,
        40
      ]
    },
    "results": [
    {
      "efficiency": "170.585",
      "coaTTC": {
        "48639a8b-8fcd-4802-af3f-f8f28cd24901.Read": [
          20,
          23,
          35
        ],
        "58639a8b-8fcd-4802-af3f-f8f28cd24901.Eavesdrop": [
          1.7976931348623157E+308,
          1.7976931348623157E+308,
          1.7976931348623157E+308
        ]
      },
      "monetary_cost": {
        " EavesdropDefense_mc ": 20
      },
      "defenses": [
        {
          "ref": " 48639a8b-8fcd-4802-af3f-f8f28cd24901",
          "defensename": " EavesdropDefense ",
          "defenseInfo": " Protects from eavesdrop attacks "
        }
      ],
      "report_url":                "https://10.0.0.2/org/project/123456789123/scenario/232236696159292/report/112654926483686"
    },
    {
      "efficiency": "272.808",
      "coaTTC": {
        "48639a8b-8fcd-4802-af3f-f8f28cd24901.Read": [
          1.7976931348623157E+308,
          1.7976931348623157E+308,
          1.7976931348623157E+308
        ],
        "58639a8b-8fcd-4802-af3f-f8f28cd24901.DataNotPresent": [
          1.7976931348623157E+308,
          1.7976931348623157E+308,
          1.7976931348623157E+308
        ]
      },
      "monetary_cost": {
        " DataNotPresent_mc ": 10
      },
      "defenses": [
        {
          "ref": "0a6b636c-3dc8-42bd-8414-cfa062aea574",
          "defensename": " DataNotPresent ",
          "defenseInfo": "Model the probability of data"
        }
      ],
      "report_url":                "https://10.0.0.2/org/project/123456789123/scenario/232236696159292/report/337386199836050"
    }
]}
```


8.1.5 Automatic Reconfiguration

Table 8-8 - Request and response example for AR.1 Responder

Request data	{ "description": "Selected CoA details: \\n\\tBusiness Impact: 0.56\\n\\tContainment: false\\n\\tPlaybook: Soccrates CoA Playbook(prevention)\\n\\t\\t - This step will Patch the appli-cation on selected hosts\\n\\t\\t

	(target: 112233)\\n\\t\\t - This step will enable Two Factor Authentication\\n\\t\\t (target: 445566)\\n\\n\\nReport URL: https://10.0.0.2/org/project/1234567890\\n\\nTriggering Message: {\\\"Host\\\": \\\"computer01.soccrates.eu\\\",\\\"IP address\\\": \\\"10.0.0.1\\\"}", "title": "CoAs suggested by SOCCRATES platform as Response on Discovery of System Configuration Change (UC-5)", "tags": ["mail:it_support@company.com"], "username": "soccrates.manager@gmail.com" }

Response data	{
	"summary": {},
	"full": {
	"message": "message sent"
	},
	"success": true,
	"artifacts": [],
	"operations": [
	{
	"type": "AddTagToCase",
	"tag": "mail sent"
	}
	]
	}

8.1.6 Advanced Attack Detection

Table 8-9- Request and response example for AAD.1 Responder (new vulnerability)

Request data	{
	"dataType": "thehive:case",
	"data": "{
	\"timestamp\" : \"2021-05-08T18:04:12Z\",
	\"type\" : \"newVulnerability\",
	\"host\" : \"host01.soccrates.eu\",
	\"vulnerabilities\": [
	{\"software\": \" Microsoft Edge (Chromium-based)\",\"cve\": \" CVE-2021-33741\"}
	]
	}",
	"tlp": "0"
	}

Response data	{
	"summary": {},
	"full": {
	"message": "Successfully sent event to Reasens"
	},
	"success": true,
	"artifacts": [], "operations": [] }

Table 8-10 - Request and response example for AAD.1 Responder (new asset)

Request data	{
	"dataType": "thehive:case",
	"data": "{\"timestamp\" : \"2021-05-09T18:06:22Z\", \"newAsset\":[\"host02.soccrates.eu\"]}"
	}

Response data	{
	"summary": {},
	"full": {
	"message": "Successfully sent event to Reasens"
	},
	"success": true, "artifacts": [], "operations": [] }

9 ANNEX II : Data shared between tasks in UC5 workflow

Description		Internal Variable
Data input		input_system_uc5
JSON	{ "HostName": "computer01.soccrates.eu", "Date": "2021-11-04T13:45:05.098Z" }	
Request to SOCCRATES_Business_Impact_responder_1_0		-
JSON	{ "newAsset": ["computer01.soccrates.eu"] }	
Request to SOCCRATES_AAD_RESPONDER_1_0		-
JSON	{ "host": "10.0.0.1", "type": "newAsset", "timestamp": "2021-11-04T13:45:05.098Z" }	
Request to SOCCRATES_Business_GetCritical_Analyzer_1_0		-
JSON	{ "Threshold": 0.54 }	
Response from SOCCRATES_Business_GetCritical_Analyzer_1_0		bia_critical_assets
JSON	{ "critical_assets": ["a", "b", "d"], "report_url": "https://10.0.0.2/org/project/123456789" }	
Request to SOCCRATES_ADG_ANALYZER		-
JSON	{ "date": "2021-10-28", "threatActor": { "threatActor": "APT-28", "capabilities": ["T1110[.]002", "T1087"] }, "entryPoints": [{ "ref": "INTERNET", "type": "Connect" }], "highValueAssets": [{ "ref": "EC2AMAZ-MDIJQUO (CEO Laptop)", "type": "Confidentiality"	

```
      }
   ],
   "vulnerabilities":
   [{
        "ref": "6749be31-f8aa-484c-9aa1-21bb4ae0ef82",
        "cve": "CVE-2021-1234"
     }]
}
```


Response from SOCCRATES_ADG_ANALYZER	adg_analysis

JSON


```
{
   "simulationId": "179693909062924",
   "reportUrl":       "https://<activiti_server_ip>/project/254027791689324/scenario/192139535621048/report/179693909062924",
   "results": {
    "attacker": {
     "metaconcept": "Attacker",
     "id": 7578,
     "oid": "294010606823422",
     "name": "Attacker",
     "description": null
    },
    "threshold": 100,
    "simid": "179693909062924",
    "results": {
     "date": "2021-10-28 07:39:56.094731",
     "confidentiality": 0,
     "integrity": 0,
     "availability": 0,
     "risk": 0,
     "maxrisk": 5,
     "max": 1.7976931348623157e+308,
     "top hosts": [],
     "top networks": [],
     "top datastores": [],
     "risks": [
      {
       "metaconcept": "System",
       "object_id": "49",
       "object_name": "EC2AMAZ-MDIJQUO (CEO Laptop)",
       "attackstep": "FullAccess",
       "ttc5": "1.7976931348623157E308",
       "ttc50": "1.7976931348623157E308",
       "ttc95": "1.7976931348623157E308",
       "probability": "0.0",
       "consequence": "5",
       "values": [
        1.7976931348623157e+308,
        ...
       ],
       "attackstep_id": "49.FullAccess"
      }
     ],
     "imperfectdefenses": 0,
     "team_vulnerabilities": [],
     "tag_results": {}
    },
    "threat_summary": [],
    "chokepoints": [],
    "missing": [],
    "report_url":      "https://<activiti_server_ip>/project/254027791689324/scenario/192139535621048/report/179693909062924"
   },
   "ciaResults": [
```

```
    {
     "ref": "EC2AMAZ-MDIJQUO (CEO Laptop)",
     "availability": {
      "ttc": 0,
      "probability": 0
     },
     "confidentiality": {
      "ttc": 0,
      "probability": 0
     },
     "integrity": {
      "ttc": 0,
      "probability": 0
     }
    }
   ],
   "ciaResultsDelta": {}
  }
```


Request to SOCCRATES_Business_Impact_analyzer_1_0	-

JSON


```
{
   "Impacted_assets":
   [
      {
         "assetID": "2",
         "confidentiality":
         {
            "ttc": 5,
            "probability": 0.8
         },
         "integrity":
         {
            "ttc": 5,
            "probability": 0.8
         },
         "availability":
         {
            "ttc": 5,
            "probability": 0.8
         }
      }
   ]
}
```


Response from SOCCRATES_Business_Impact_analyzer_1_0	impact_analysis

JSON


```
{
   "business_impacts":
   [
      {
         "business_name": "bf1",
         "impact_probability": 0.32,
         "impact_criticality": 8.56704875182555,
         "critical_time": 10.4
      },
      {
         "business_name": "bf2",
         "impact_probability": 0.26,
         "impact_criticality": 2.4977408861112362,
         "critical_time": 5.4
      },
      {
         "business_name": "bp1",
         "impact_probability": 0.32,
         "impact_criticality": 0.9247724179961048,
         "critical_time": 12.4
```

<table>
<tr><td></td><td><pre>
      },
      {
        "business_name": "Company",
        "impact_probability": 0.28,
        "impact_criticality": 9.807794186006355,
        "critical_time": 21.4
      }
    ],
    "report_url": "https://10.0.0.2/org/project/123456789"
}
</pre></td></tr>
<tr><td colspan="2">User input</td><td>mustGenerateCoA</td></tr>
<tr><td>data</td><td colspan="2"><true|false></td></tr>
<tr><td colspan="2">Request to SOCCRATES_Coa_Generator_1_0</td><td>-</td></tr>
<tr><td>JSON</td><td colspan="2"><pre>
{
  "simulationId": "307666084811239"
}
</pre></td></tr>
<tr><td colspan="2">Response from SOCCRATES_Coa_Generator_1_0</td><td>coas</td></tr>
<tr><td>JSON</td><td colspan="2"><pre>
{
  "initialTTC": {
    "48639a8b-8fcd-4802-af3f-f8f28cd24901.Read": [20,23,35],
    "58639a8b-8fcd-4802-af3f-f8f28cd24901.Eavesdrop": [23,30,40]
  },
  "results": [
  {
    "efficiency": "170.585",
    "coaTTC": {
      "48639a8b-8fcd-4802-af3f-f8f28cd24901.Read": [
        20,
        23,
        35
      ],
      "58639a8b-8fcd-4802-af3f-f8f28cd24901.Eavesdrop": [
        1.7976931348623157E+308,
        1.7976931348623157E+308,
        1.7976931348623157E+308
      ]
    },
    "monetary_cost": {
      " EavesdropDefense_mc ": 20
    },
    "defenses": [
      {
        "ref": " 48639a8b-8fcd-4802-af3f-f8f28cd24901",
        "defensename": " EavesdropDefense ",
        "defenseInfo": " Protects from eavesdrop attacks "
      }
    ],
    "report_url": "https://10.0.0.2/org/project/123456789123/scenario/232236696159292/report/112654926483686"
  },
  {
    "efficiency": "272.808",
    "coaTTC": {
      "48639a8b-8fcd-4802-af3f-f8f28cd24901.Read": [
        1.7976931348623157E+308,
        1.7976931348623157E+308,
        1.7976931348623157E+308
      ],
      "58639a8b-8fcd-4802-af3f-f8f28cd24901.DataNotPresent": [
        1.7976931348623157E+308,
        1.7976931348623157E+308,
        1.7976931348623157E+308
</pre></td></tr>
</table>

```
        ]
      },
      "monetary_cost": {
        " DataNotPresent_mc ": 10
      },
      "defenses": [
        {
          "ref": "0a6b636c-3dc8-42bd-8414-cfa062aea574",
          "defensename": " DataNotPresent ",
          "defenseInfo": "Model the probability of data"
        }
      ],
      "report_url":                "https://10.0.0.2/org/project/123456789123/scenario/232236696159292/report/337386199836050"
    }
]}
```


Request to SOCCRATES_RP_COA_RANKING_ANALYZER	-

JSON


```
{
  "siem_label": "TA0001",
  "initialTTC":
  {
    "48639a8b-8fcd-4802-af3f-f8f28cd24901[.]Read":  [ 19, 26, 34 ],
    "325[.]NetworkRespondConnect": [ 23, 30, 40 ]
  },
  "containment": false,
  "coas":
  [
    {
      "efficiency": "170[.]585",
      "coaTTC":
      {
        "48639a8b-8fcd-4802-af3f-f8f28cd24901[.]Read": [18, 25, 34 ],
        "325[.]NetworkRespondConnect":
        [ 17976931348623157, …]
      },
      "time_cost": { "1": 355, "2": 564, "3": 22, "4": 895, "5": 1336},
      "monetary_cost": { "1": 4464, "2": 564, "3": 556, "4": 895, "5": 1336 },
      "defenses":
      [
        {
          "ref": "100",
          "defenseName": "Disabled",
          "defenseInfo": "Info",
          "mitreRef": "Ref"
        }
      ]
    },
    {
      "efficiency": "272[.]808",
      "coaTTC":
      {
        "228[.]Read": [ 17976931348623157, .. ],
        "325[.]NetworkRespondConnect": [ 17976931348623157,… ]
      },
      "time_cost": { "1": 355, "2": 564, "3": 22, "4": 895, "5": 1336 },
      "monetary_cost": { "1": 87355, "2": 564, "3": 228750, "4": 8955, "5": 8788 },
      "defenses":
      [
        {
          "ref": "119",
          "defenseName": "authenticated",
          "defenseInfo": "Info",
          "mitreRef": "Ref"
```

```
        },
        {
          "ref": "100",
          "defenseName": "Disabled",
          "defenseInfo": "Info",
          "mitreRef": "Ref"
        }
      ]
    }
  ]
}
```


Response from SOCCRATES_RP_COA_RANKING_ANALYZER	rori

JSON


```
{
  "ranked_coas": [
   {
    "RORI": 10003.873665927717,
    "containment": false,
    "coaTTC": {
     "228.Read": [
      18,
      25,
      34
     ],
     "325.NetworkRespondConnect": [
      17976931348623156,
      17976931348623156,
      17976931348623156
     ]
    },
    "monetary_cost": {
     "1": 4464,
     "2": 564,
     "3": 556,
     "4": 895,
     "5": 1336
    },
    "defenses": [
     {
      "ref": "100",
      "defenseName": "Disabled",
      "defenseInfo": "Info",
      "mitreRef": "Ref"
     }
    ],
    "business_impact": 0.56,
    "playbook": {
     "type": "playbook",
     "spec_version": "1.0",
     "created_by": "Soccrates_Response_Planner",
     "description": "This playbook will apply the CoA in the system",
     "playbook_types": [
      "prevention"
     ],
     "features": {
      "parallel_processing": true
     },
     "created": "Thu Oct 28 15:49:40 2021",
     "id": "soccrates-playbook-1635428980436982664",
     "name": "Soccrates CoA Playbook 1635428980436982664",
     "workflow_start": "start--soccrates-playbook-1635428980436982664",
     "workflow": {
      "start--soccrates-playbook-1635428980436982664": {
       "type": "start",
```

```
        "on_completion": "parallel--soccrates-playbook-1635428980436982664"
       },
       "parallel--soccrates-playbook-1635428980436982664": {
        "type": "parallel",
        "name": "Apply defenses of the CoA",
        "next_steps": [
         "defense1--soccrates-playbook-1635428980436982664"
        ]
       },
       "defense1--soccrates-playbook-1635428980436982664": {
        "type": "single",
        "name": "Disabled",
        "description": "Info",
        "on_completion": "end--soccrates-playbook-1635428980436982664",
        "target": "100",
        "commands": [
         {
          "type": "manual",
          "command": "Info on 100"
         }
        ]
       },
       "end--soccrates-playbook-1635428980436982664": {
        "type": "end"
       }
      }
     },
     "id": 0,
     "name": "COA 1"
    },
    {
     "RORI": 6173.048024629743,
     "containment": false,
     "coaTTC": {
      "228.Read": [ 17976931348623156, …],
      "325.NetworkRespondConnect": [ 17976931348623156, … ]
     },
     "monetary_cost": {
      "1": 87355,
      "2": 564,
      "3": 228750,
      "4": 8955,
      "5": 8788
     },
     "defenses": [
      {
       "ref": "119",
       "defenseName": "authenticated",
       "defenseInfo": "Info",
       "mitreRef": "Ref"
      },
      {
       "ref": "100",
       "defenseName": "Disabled",
       "defenseInfo": "Info",
       "mitreRef": "Ref"
      }
     ],
     "business_impact": 0.56,
     "playbook": {
      "type": "playbook",
      "spec_version": "1.0",
      "created_by": "Soccrates_Response_Planner",
      "description": "This playbook will apply the CoA in the system",
```

```
        "playbook_types": [
         "prevention"
        ],
        "features": {
         "parallel_processing": true
        },
        "created": "Thu Oct 28 15:49:40 2021",
        "id": "soccrates-playbook-1635428980437267358",
        "name": "Soccrates CoA Playbook 1635428980437267358",
        "workflow_start": "start--soccrates-playbook-1635428980437267358",
        "workflow": {
         "start--soccrates-playbook-1635428980437267358": {
          "type": "start",
          "on_completion": "parallel--soccrates-playbook-1635428980437267358"
         },
         "parallel--soccrates-playbook-1635428980437267358": {
          "type": "parallel",
          "name": "Apply defenses of the CoA",
          "next_steps": [
           "defense1--soccrates-playbook-1635428980437267358",
           "defense2--soccrates-playbook-1635428980437267358"
          ]
         },
         "defense1--soccrates-playbook-1635428980437267358": {
          "type": "single",
          "name": "authenticated",
          "description": "Info",
          "on_completion": "end--soccrates-playbook-1635428980437267358",
          "target": "119",
          "commands": [
           {
            "type": "manual",
            "command": "Info on 119"
           }
          ]
         },
         "defense2--soccrates-playbook-1635428980437267358": {
          "type": "single",
          "name": "Disabled",
          "description": "Info",
          "on_completion": "end--soccrates-playbook-1635428980437267358",
          "target": "100",
          "commands": [
           {
            "type": "manual",
            "command": "Info on 100"
           }
          ]
         },
         "end--soccrates-playbook-1635428980437267358": {
          "type": "end"
         }
        }
       },
       "id": 1,
       "name": "COA 2"
      }
     ],
     "report_url": "10.8.0.4:8001/1635428980437337204"
    }
```


User input	selectedCoAs
data	<coa_id>

Request to Mailer_1_0	

JSON	{
	"description": "Selected CoA details: \n\n\nNo selected CoAs\n\nTriggering Message: {\"Host\": \"computer01.soccrates.eu\",\"IP address\": \"10.0.0.1\"}",
	"title": "CoAs suggested by SOCCRATES platform as Response on Discovery of System Configuration Change (UC-5)",
	"tags": [
	"mail:<user>@gmail.com"
	],
	"username": "soccrates.manager@gmail.com"
	}