A coordinated framework for cyber resilient supply chain systems over complex ICT infrastructures

D6.2 IT-1 FISHY release validated

Document Identification			
Status	Final	**Due Date**	28/02/2022
Version	1.0	**Submission Date**	10/03/2022

Related WP	WP6	**Document Reference**	D6.2
Related Deliverable(s)	D6.1, D5.1	**Dissemination Level (*)**	PU
Lead Participant	ATOS	**Lead Author**	Antonio Álvarez
Contributors	SYN, XLAB, UPC, TUBS, Sonae, Capgemini, STS	**Reviewers**	Guillermo Jiménez Prieto, Araceli Rojas Morgan (CAPGEMINI)
			Antonis Gonos (OPTIMUM)

Keywords:
Pilot scenario, validation activities, validation metrics, use cases, proof of concept, results

This document is issued within the frame and for the purpose of the FHISY project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 952644. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the European Commission.

This document and its content are the property of the FISHY Consortium. All rights relevant to this document are determined by the applicable laws. Access to this document does not grant any right or license on the document or its contents. This document or its contents are not to be used or treated in any manner inconsistent with the rights or interests of the FISHY Consortium or the Partners detriment and are not to be disclosed externally without prior written consent from the FISHY Partners.

Each FISHY Partner may use this document in conformity with the FISHY Consortium Grant Agreement provisions.

(*) Dissemination level: **PU**: Public, fully open, e.g. web; **CO:** Confidential, restricted under conditions set out in Model Grant Agreement; **CI:** Classified, **Int =** Internal Working Document, information as referred to in Commission Decision 2001/844/EC.

Document Information

List of Contributors	
Name	Partner
Antonio Álvarez Romero	ATOS
Ana Machado Silva	SONAE
José Duarte	SONAE ARAUCO
Eleni Leligou	SYN
Panagiotis Athanasoulis	SYN
Panagiotis Karkazis	SYN
Guillermo Jiménez Prieto	CAPGEMINI
Jan Antic	XLAB
Grigorios Kalogiannis	STS
Andreas Zacharakis	STS
Andreas Miaoudakis	STS
Manolis Chatzimpyrros	STS
Eva Marín Tordera	UPC
Ayaz Husain	UPC
Mounir Bensalem	TUBS

Document History			
Version	Date	Change editors	Changes
0.1	2022-01-19	ATOS, SYN	ToC and initial structure
0.2	2022-02-10	SONAE, SYN, CAPGEMINI, ATOS	Input to sections 2, 3, 4
0.3	2022-02-11	ATOS, SYN	Input to section 1 and to the table of acronyms
0.4	2022-02-16	SYN, ATOS	Addition of figures to section 2.3
0.5	2022-02-18	SONAE, ATOS	Addition of acronym to table, refinement in sections 1.4, 3.2, 3.3. Input to section 3.4
0.6	2022-02-24	ATOS, SYN, CAPGEMINI, XLAB, STS	Refinement of section 2 and 4
0.7	2022-02-25	ATOS	Input to section 1. Refinements in sections 2, 3, 4. Input to sections 5 and 6.
0.8	2022-03-01	ATOS, SYN, UPC	Refinements in section 2 and 5
0.9	2022-03-02	ATOS, SONAE	Update in list of acronyms. Update in section 2.3. Refinements in sections 3 and 5.
0.10	2022-03-03	ATOS, TUBS	Refinement in section 2.3
0.11	2022-03-04	ATOS, CAPGEMINI	Acronyms added. Refinements in sections 4 and 5.

0.12	2022-03-04	ATOS, SONAE	Refinement in section 3.2
0.13	2022-03-04	ATOS, STS	Refinement in section 2.3
0.14	2022-03-04	ATOS	Document layout
0.15	2022-03-07	ATOS, CAPGEMINI	Refinement in section 4.2 and 4.3. Document ready for QA process.
0.16	2022-03-08	ATOS, OPTIMUM, CAPGEMINI	Document review and addressing comments
0.17	2022-03-09	ATOS, SONAE, CAPGEMINI	Document review and addressing comments
1.0	2022-03-10	ATOS	FINAL VERSION TO BE SUBMITTED

Quality Control		
Role	Who (Partner short name)	Approval Date
Deliverable leader	Antonio Álvarez (ATOS)	10/03/2022
Quality manager	Juan Andrés Alonso (ATOS)	10/03/2022
Project Coordinator	Antonio Álvarez (ATOS)	10/03/2022

Table of Contents

List of Tables

List of Figures

List of Acronyms

Abbreviation / acronym	Description
DID	Decentralised Identifier
EDC	Enforcement and Dynamic Configuration
ELK	Elastic search, Logstash and Kibana
F2F	Farm to Fork
IRO	Intent-based Resilience Orchestrator
JSON	JavaScript Object Notation
K8S	Kubernetes
NED	Network Edge Device
OEM	Original Equipment Manufacturer
PoC	Proof-of-Concept
POD	*Pods* are the smallest deployable units of computing that can be created and manage in Kubernetes
RAE	Risk Assessment Engine
SACM	Security Assurance & Certification Management
SADE	Securing Autonomous Driving function at the Edge
SIA	Secure Infrastructure Abstraction
SSID	Service Set Identifier
TIM	Trust & Incident Manager
UC	Use Case
UML	Unified Modelling Language
UTC	Universal Time Coordinated
UUID	Universally Unique Identifier
VAT	Vulnerability Assessment Tool
WBPTV	Wood-based Panels Trusted Value-chain

Executive Summary

Deliverable D6.2, "IT-1 FISHY release validated" reports the first iteration in the process of deploying, validating and assessing the FISHY Platform in the three use cases. For each pilot, the specific vertical applications are presented and then the different security enhancements enabled by the FISHY platform are described. A very visual approach with architectural figures and several screenshots of the work performed has been used throughout the document. The interplay among components is shown providing evidence of their use in each pilot. A first wave of feedback has been generated. Firstly, it has been compiled at pilot level and then there is also a global aggregation. All three pilots find relevant how the FISHY platform can contribute for a better cyber resilience concerning their supply chains. The feedback obtained is positive and encourages further work integrating more elements and designing more complex use cases. This will be carried out during the second iteration and will feed the showcasing activities as the project becomes more mature.

1 Introduction

1.1 Purpose of the document

Deliverable D6.2 is the first report about the activities that were performed in the framework of Task T6.3 (PoC Deployment and Demonstration) and Task T6.4 (Validation and Assessment). These two tasks started simultaneously in M15 (November 2021) and this report covers the work done until M18 (February 2022), when IT-1 concludes and the intermediate version of the FISHY Platform is available. An update of this deliverable (D6.4) will be prepared and submitted in M36 after the IT-2 version of the FISHY platform has been released.

As the focal point of this document is the validation of FISHY-IT1 in the three different use cases, the reader will find information about the validation activities carried out in the three pilots: F2F (SYN, OPT), WBPTV (SONAE) and SADE (CAPGEMINI). For each pilot, we will present the specific vertical applications and then we will address the security enhancements enabled by the FISHY Platform. Then the outcomes are evaluated and a first wave of feedback is generated, first at pilot level and then there is an aggregation at project level.

1.2 Relation to other project work packages

This deliverable highly interrelates with D6.1 [1] which describes the integration and validation methodology and planning as well as the threats and attacks to be detected. There is a bidirectional relationship with the rest of technical WPs (WP2, 3, 4 and 5). On one hand, the WP6 is about piloting and benefits from the previous technical work carried out in the aforementioned WPs. On the other hand, the insights of this first round of testing, validation and assessment is relevant to guide the further work in the technical WPs 2-5 for the subsequent months towards produced FISHY IT-2 to better match the use case requirements.

1.3 Structure of the document

This document is organised in the following major chapters:

- **Chapter 1:** introduces the document
- **Chapter 2-4:** These chapters report the validation activities for FISHY IT-1 in each one of the three FISHY use cases (F2F, WBPTV and SADE).
- **Chapter 5: Result consolidation.** In this chapter, the feedback from the three use cases is consolidated and organised per FISHY component so that the feedback that will be provided to WP2-5 is coherent.
- **Chapter 6: Conclusions** This chapter provides the conclusions of this deliverable.

1.4 Glossary adopted in this document and clarification of terms

In this section we clarify that:

- **FISHY use case:** FISHY has selected and described in its DoA three different supply chains (F2F, WBPTV and SADE) which can use/exploit the FISHY platform. There are appropriate partners within the FISHY Consortium to pilot and test the FISHY platform in one instance of each considered supply chain. Namely, Optimum and Synelixis for the piloting of the F2F use case, SONAE for the piloting of the WBPTV use case and Capgemini Engineering (ex. ALTRAN) for the piloting of the SADE use case.
- **UC-Use Case**. In this deliverable, the acronym UC refers to the formally described "use cases" as they are defined using the Unified modelling language (UML). We studied the use of the FISHY platform and came up with UML diagrams for each FISHY use case in order to capture detailed requirements and rigorously define elaborate (UML-compliant) use cases that would drive the testing of the FISHY platform and its components. For this purpose, we try to differentiate it from the FISHY use cases which are in essence supply chain instances using FISHY platform. In many cases, to stress the difference we refer to FISHY use case vs. UML-compliant use cases.

2 FISHY validation in Farm to Fork supply chain

2.1 Introduction

In this chapter, we focus on the validation of FISHY IT-1 in the Farm-to-Fork supply chain. Considering as starting point the description of this use case in D6.1 [1], in the following section we first describe the deployment details and the relevant challenges we faced. In section 2.3 we describe in detail how FISHY detects the threats/attacks defined in D6.1, also providing evidence and feedback with respect to FISHY IT-1 in section 2.4.

2.2 Farm-to-Fork (F2F) vertical application

In the Farm to Fork supply chain, to protect the F2F platform, we (SYN, OPT) have implemented the components that deliver to the FISHY platform information from four distinct points of the deployed F2F platform. The “security probes” (marked as entry point 1, 2, 3, 4a and 4b), which have been described in [1], of the F2F platform are shown in the following Figure 1. These data are sent to FISHY platform in the form of a JSON object which will include the following fields: UUID (Unique Universal ID), Timestamp (UTC timestamp), Type, Metadata.

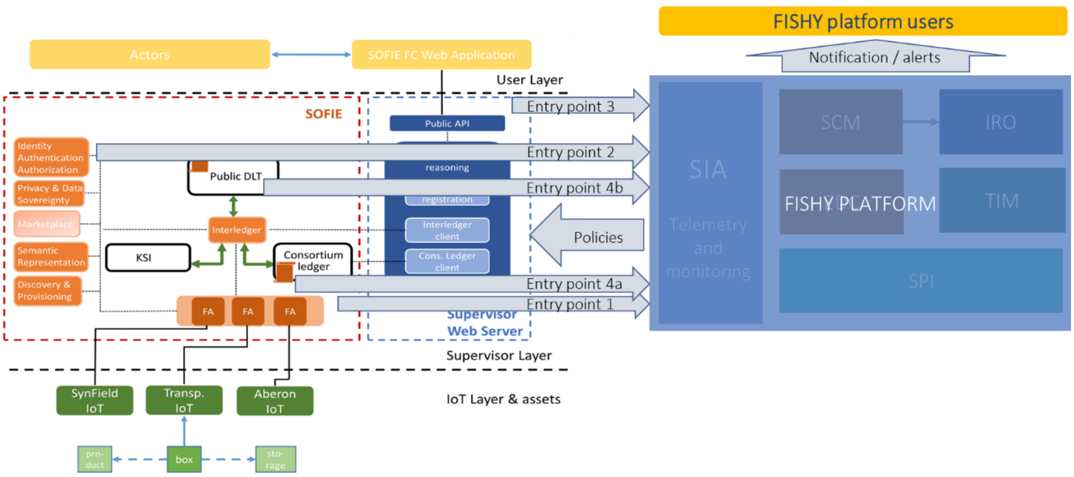


Figure 1: The F2F platform and its interconnection with the FISHY platform

2.3 FISHY-enabled security enhancement in F2F supply chain

In the F2F case, SYN/OPT consider that four types of attacks are of interest and have defined the metadata that are passed to FISHY to enable their detection. These are:

- Type 1: Unauthorised device –wallet ID level
  - Metadata: {Attacker wallet ID, Expected Legitimate Wallet ID, Device name}
- Type 2: Unauthorised device – Decentralised Identifier DID level (with DID characterizing the device)
  - Metadata: {Attacker DID, Device name, Jwt}
- Type 3: Unauthorised User
  - Metadata: {username, IP}
- Type 4: Attack to Blockchain node
  - Metadata: {IP, port, incident type}

With the FISHY platform IT-1, we have validated the detection of threats through two different data flows which are described below.

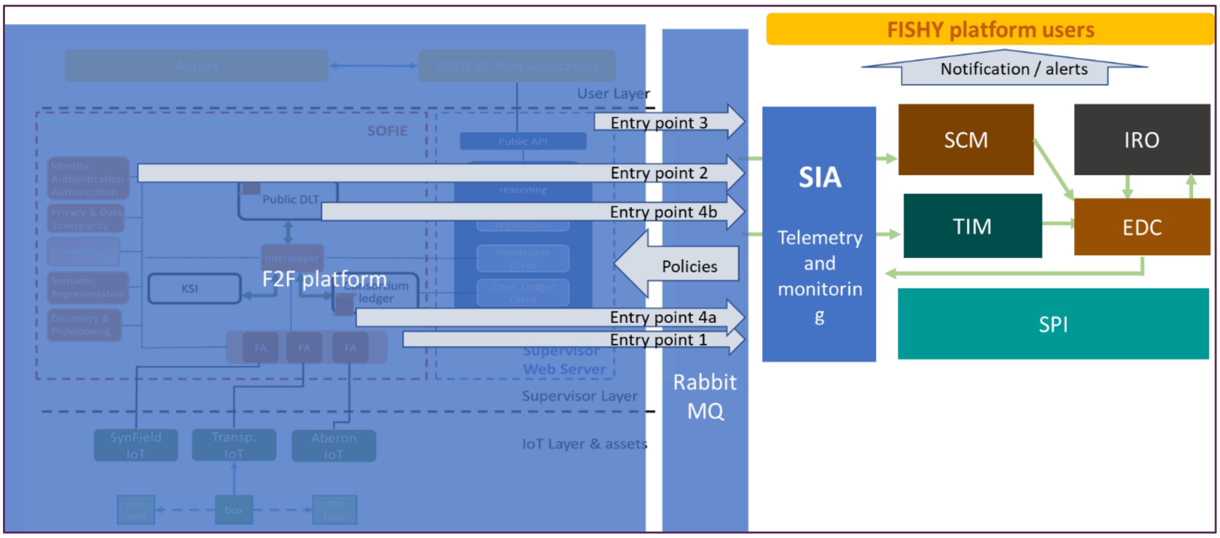


Figure 2. Flow of information enabling threat detection

As shown in Figure 2, the messages from the RabbitMQ that has been deployed for the exchange of this information are consumed by SCM, developed by STS and TIM (and more specifically Wazuh and PMEM tool in TIM, developed by XLAB and UPC respectively) to detect different types of attacks. An instance of the RabbitMQ through which the information captured from the F2F platform passes to FISHY platform components is shown in Figure 3.

Data Flow A: followed to detect threat types 1 and 4

For attack types 1 and 4, SCM is responsible for monitoring the JSON objects and when the following two rules hold, the user is notified through the IRO/Dashboard. For this to happen, SCM has been integrated with the RabbitMQ (see Figure 3) and then the SCM presents through the FISHY dashboard the results to the FISHY users (see Figure 4) which is the F2F supply chain operator in this case. Finally, the FISHY platform user can get a result from the auditing of the infrastructure he administers (Auditing and Dashboard components in action). It is worth mentioning that the EDC is responsible for deciding and enforcing policies (being informed by SCM through the central repository). The operation of EDC is described in more detail in the data flow B.

Table 1: Security rules applied for attack types 1 and 4

Type	RULE
1	If Attacker wallet ID appears more than *Threshold1.1* times in *Threshold 1.2* hours, then • FISHY **notifies/alerts** F2F supply chain operator and/or • FISHY **notifies** IoT Island operator and/or • FISHY **enforces** Wallet ID ban (i.e., the F2F SOFIE platform will no longer consider keeping information coming from this wallet ID).
4	If IP appears more than *Threshold4.1* times in *Threshold4.2* hours, then • FISHY **notifies** F2F supply chain operator providing the IP and port number • FISHY **enforces** IP ban (i.e., the F2F SOFIE platform will no longer accept access request from this specific IP).

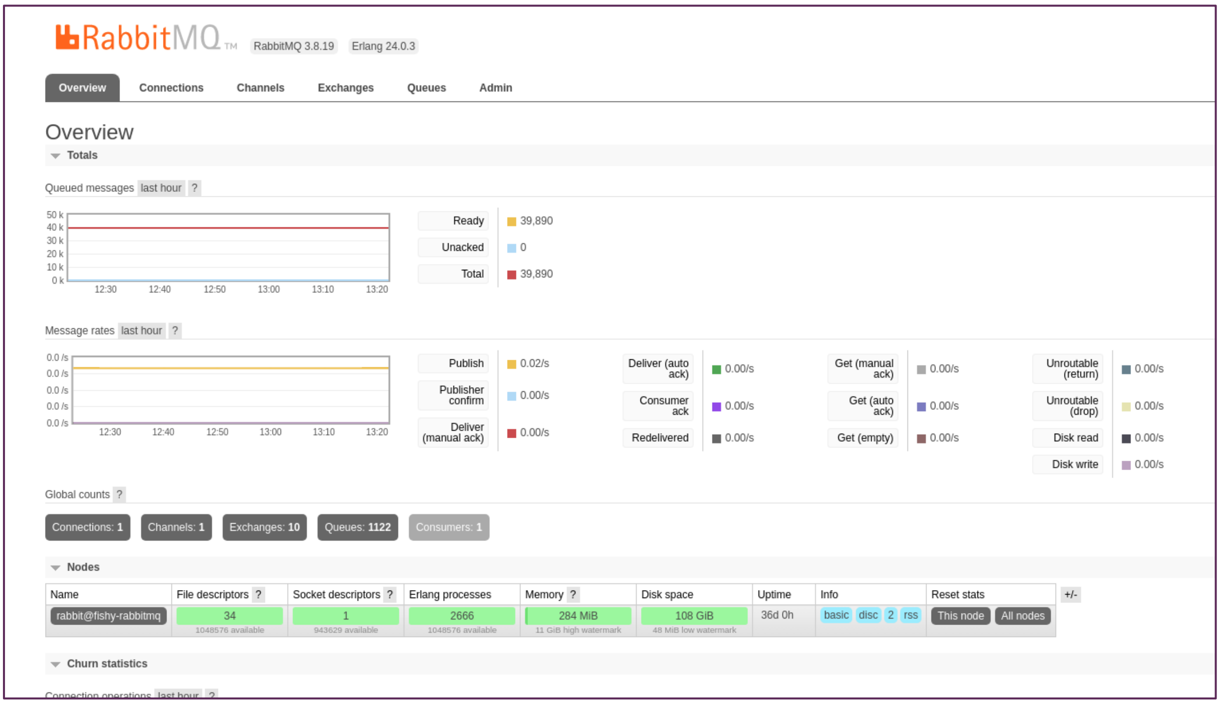


Figure 3. The RabbitMQ that enables the information dissemination between FISHY components and F2F platform

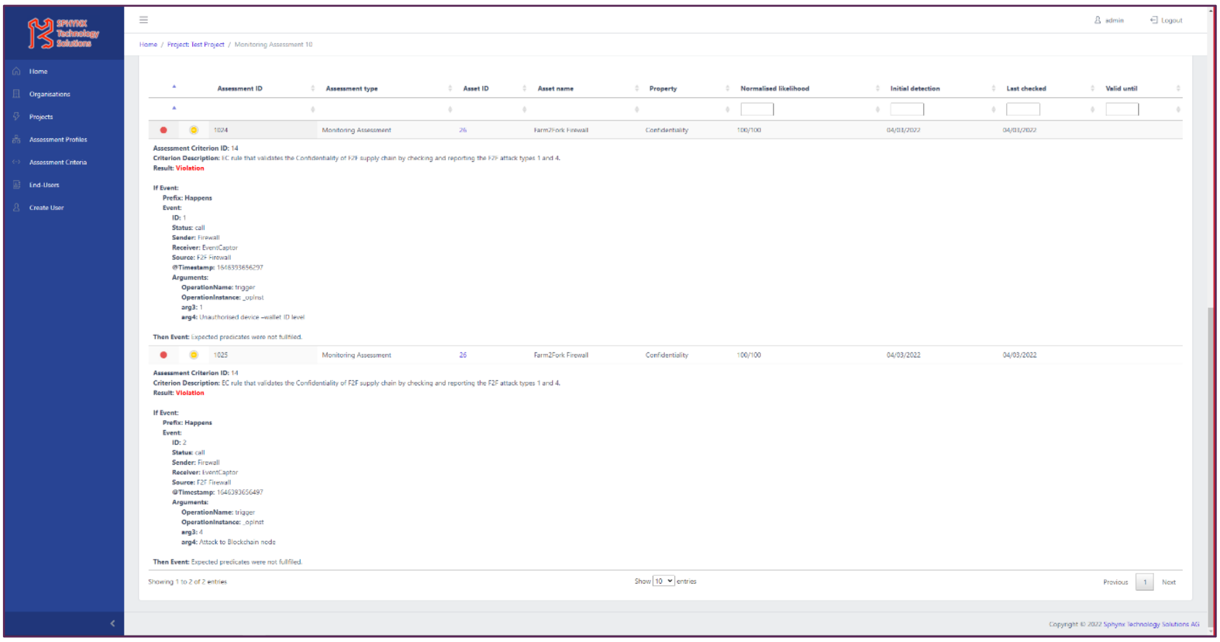


Figure 4. Screenshots from the dashboard showing the threat detection of type 1 and type 4

Data Flow B: followed to detect threat types 2 and 3

In this case, the messages from the RabbitMQ are consumed by TIM (lead developer: XLAB) and when the two rules (shown in the table below) hold, TIM communicates with IRO through the central threat/ attack repository which operates in pub-sub mode. This in turn notifies a) the FISHY platform user and b) the EDC to process the event and suggest the enforcement of a relevant policy. The integration with TIM- Wazuh tool is evident in Figure 5 where the reception of events from Synelixis' platform is shown, while for the integration with PMEM component we show here Figure 6, where PMEM component deployed in Synelixis' infrastructure analyses network traffic. The detection of events is shown to the user through IRO dashboard (shown in Figure 8). These events trigger the definition of specific policies (e.g. IP ban) by EDC which are passed to the F2F platform for enforcement through a RabbitMQ as shown in Figure 9 and Figure 10.

Table 2: Security rules applied for attack types 2 and 3

Type	RULE
2	If Attacker DID appears more than *Threshold2.1* times in *Threshold2.2* hours, then • FISHY **notifies/alerts** F2F supply chain operator providing the relevant log info (Attacker DID, Device name) and/or • FISHY **notifies** IoT Island operator and/or • FISHY **enforces** DID ban (i.e., the F2F SOFIE platform will no longer consider keeping information coming from this DID).
3	If IP appears more than *Threshold3.1* times in *Threshold3.2* hours, then • FISHY **notifies** F2F supply chain operator providing the relevant log info (username, IP) and/or • FISHY **enforces** IP ban (i.e., the F2F SOFIE platform will no longer accept access request from this specific IP).

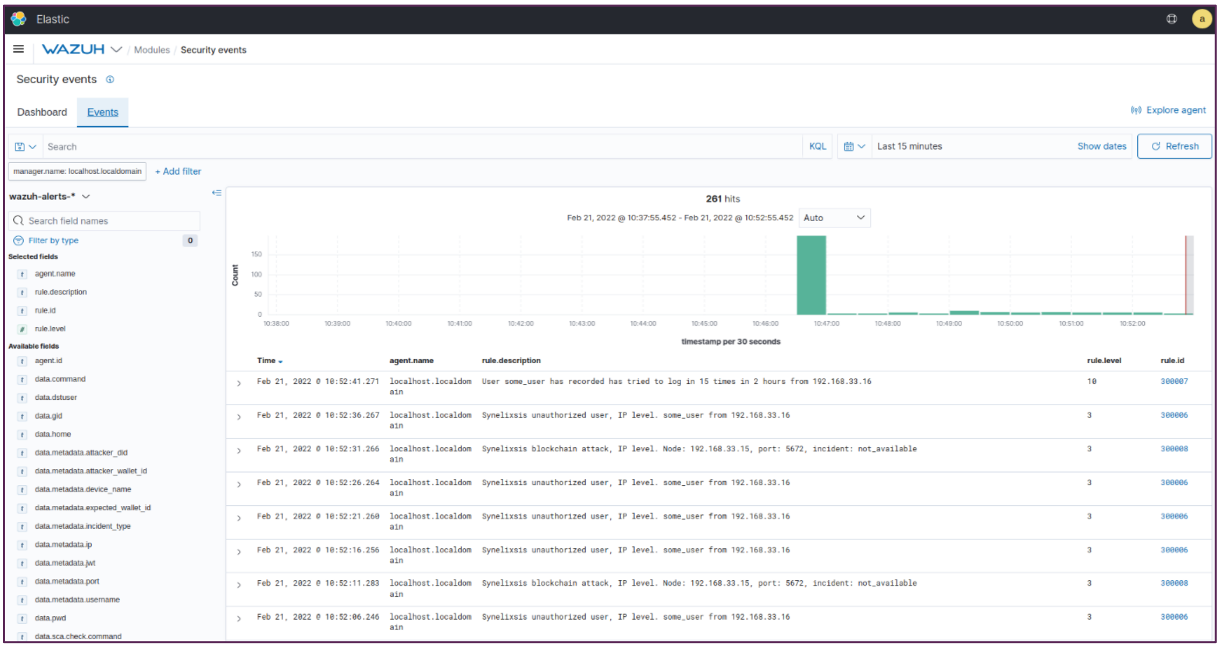


Figure 5. Evidence of TIM - Wazuh integration with Synelixis´ IT platform

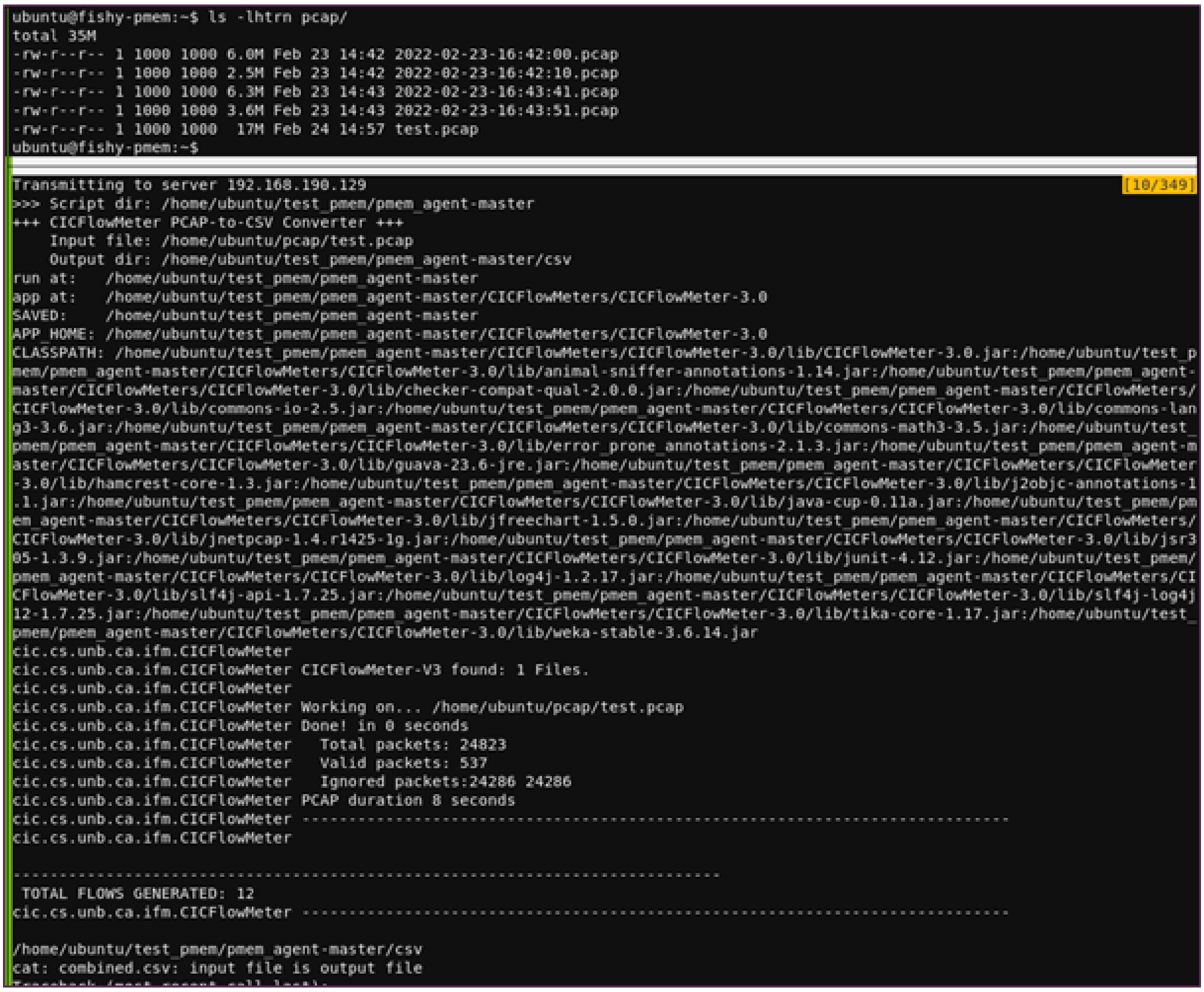


Figure 6. Evidence of TIM - PMEM integration with Synelixis´ IT platform

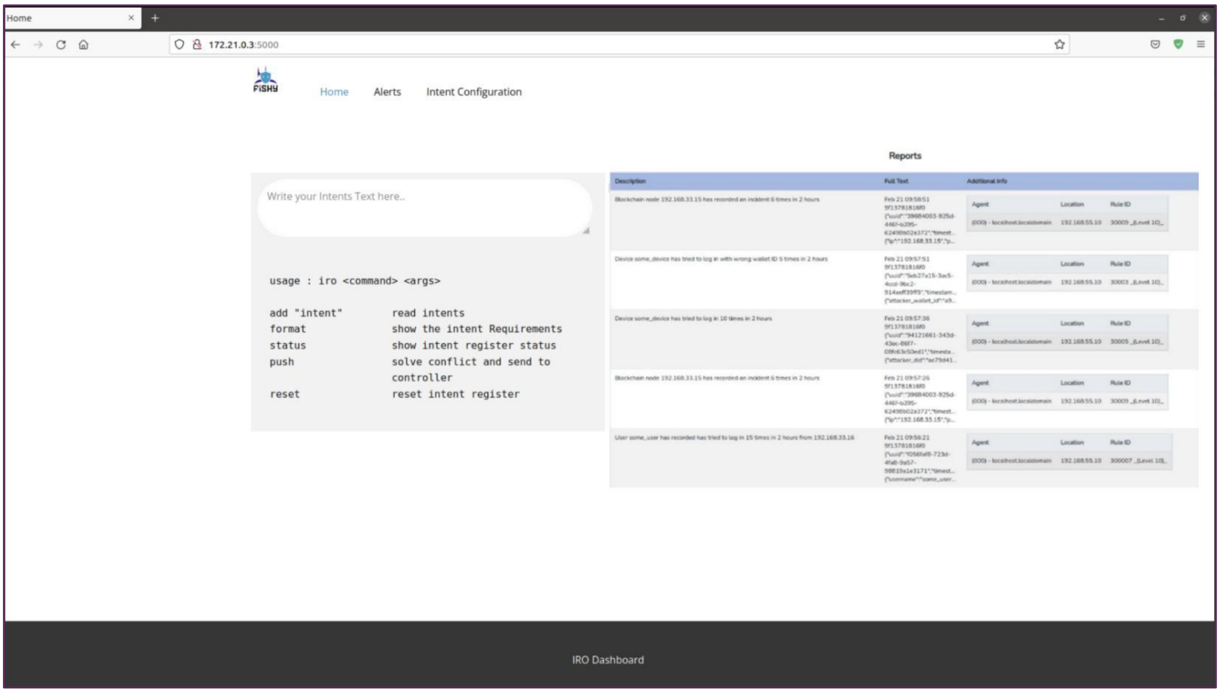


Figure 7. Indicative screen of IRO dashboard showing the detected events

Reports

Description	Full Text	Additional info
Blockchain node 192.168.33.15 has recorded an incident 6 times in 2 hours	Feb 21 09:58:51 9f13781816f0 {"uuid":"39684003-925d-446f-b395-62498b02a372","timest… {"ip":"192.168.33.15","p…	Agent: (000) - localhost.localdomain; Location: 192.168.55.10; Rule ID: 30009 _(Level 10)_
Device some_device has tried to log in with wrong wallet ID 5 times in 2 hours	Feb 21 09:57:51 9f13781816f0 {"uuid":"5eb27a15-3ac5-4ccd-9bc2-914aeff39ff5","timestam… {"attacker_wallet_id":"a9…	Agent: (000) - localhost.localdomain; Location: 192.168.55.10; Rule ID: 30003 _(Level 10)_
Device some_device has tried to log in 10 times in 2 hours	Feb 21 09:57:36 9f13781816f0 {"uuid":"94121661-343d-43ec-86f7-08fc63c50ed1","timesta… {"attacker_did":"ae79d41…	Agent: (000) - localhost.localdomain; Location: 192.168.55.10; Rule ID: 30005 _(Level 10)_
Blockchain node 192.168.33.15 has recorded an incident 6 times in 2 hours	Feb 21 09:57:26 9f13781816f0 {"uuid":"39684003-925d-446f-b395-62498b02a372","timest… {"ip":"192.168.33.15","p…	Agent: (000) - localhost.localdomain; Location: 192.168.55.10; Rule ID: 30009 _(Level 10)_
User some_user has recorded has tried to log in 15 times in 2 hours from 192.168.33.16	Feb 21 09:56:21 9f13781816f0 {"uuid":"f056faf8-723d-4fa8-9a57-98819a1e3171","timest… {"username":"some_user…	Agent: (000) - localhost.localdomain; Location: 192.168.55.10; Rule ID: 300007 _(Level 10)_

Figure 8. Indicative screen of IRO dashboard showing the detected events (zooming in the description of the attacks)

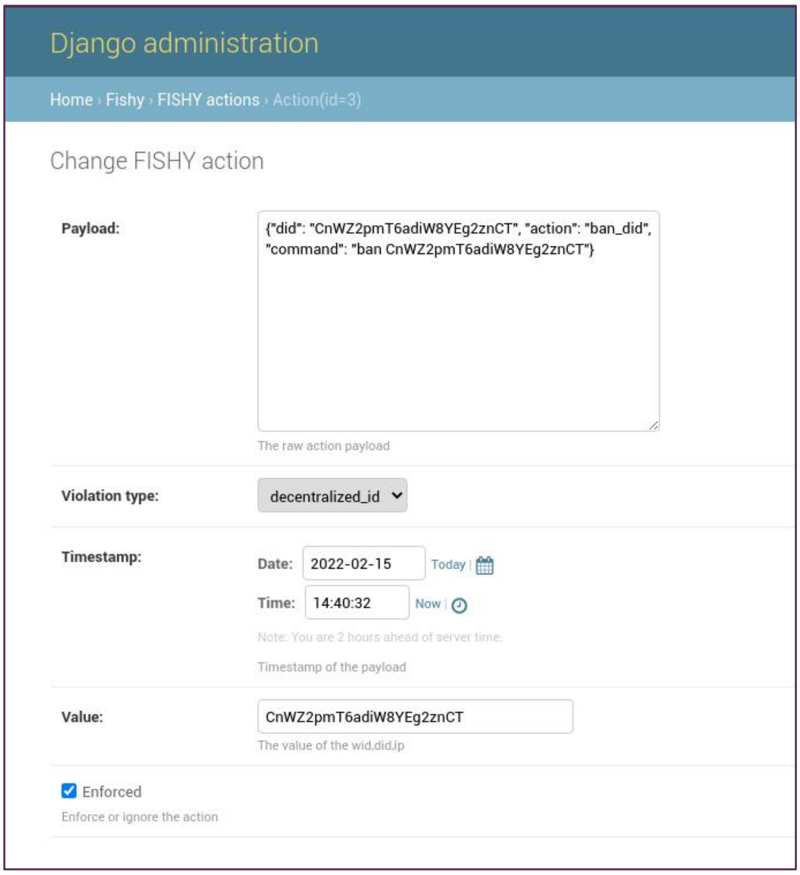


Figure 9. The RabbitMQ receives the action that should be performed as decided by EDC (rule 2)

```
ubuntu@sofie-supervisor:~$ docker logs -f fishy-action-parser
 [*] Waiting for messages. To exit press CTRL+C
 [x] Received {'action': 'ban_ip', 'ip': '5.203.235.131', 'command': 'iptables -A INPUT -s 5.203.235.131 -j DROP'}
 [x] Created action=1
```

Figure 10. The RabbitMQ receives the action that should be performed by the EDC (e.g. IP Ban)

Regarding the metrics that have been defined in D6.1, as shown in the following table the target values for both the technical and business metric have been achieved.

Table 3: Pilot metrics for the Farm-to-Fork case

Metric ID	Metric description	Type	Target value	Achieved value
SC1_B1	Number of interledger technologies supported	Business and technical	2	3
SC1_T1	Number/Types of threats that can be detected	Technical	3	4

2.4 Feedback

The integration of the F2F IT platform with FISHY platform and components was smooth and used state-of-the-art tools for the exchange of information between the two platforms like RabbitMQ. The FISHY components that were validated in the F2F use case and the relevant experience follows:

Validation of SCM: the integration and validation of SCM was smooth. The SCM component receives the information from F2F platform and monitors the conditions defined by the F2F user. SCM was found to be flexible as it supports multiple types of rules that F2F use cases needed. The respective results are shown in a dedicated dashboard. A potential improvement would be to allow the user to define the rules for attack detection through a dedicated graphical interface.

Validation of TIM: the integration and validation of TIM was equally smooth. The TIM component receives the information from F2F platform and Wazuh tool monitors the conditions defined by the F2F user. This component was found to be flexible as it supports multiple types of rules that F2F use cases needed. The respective results are shown in a dedicated dashboard. A potential improvement would be to allow the user to define the rules for attack detection through a dedicated graphical interface. Additionally, with respect to PMEM component, this was deployed in F2F infrastructure (namely in Synelixis' premises) and analyses the information relevant to the internal network where the platform is deployed. This is then passed to Machine-learning algorithms enabling anomaly detection. A concern that was raised and is relevant to the commercialisation of the PMEM component is whether the company operating the F2F solution would be willing in exposing the information captured from its internal network to the PMEM operator. This is a point that has to be clarified as the potential customer may be concerned about revealing sensitive information. (In our case, we analyse the PMEM code and we were sure of what is processed.) For the validation phase, the component was deployed in Synelixis´ infrastructure and the network information did not flow outside it. This has to be guaranteed (and accordingly marketed) for PMEM to be commercialised.

Validation of EDC: the operation of EDC was validated. The policies to enforce were co-decided by the F2F operator and EDC designers/developers. The policies to be enforced are communicated to the F2F IT platform. Knowledge of the network structure of the F2F IT platform is necessary to enable enforcement of policies in specific points/devices (firewalls, router, etc..).

Validation of IRO/dashboard: the operation of IRO/dashboard was validated as it collected the results/events detected by TIM and SACM, allowing the F2F operator create a clear understanding of what happens in the infrastructure it operates. A potential improvement anticipated to arrive at IT-2 is to allow the operator set specific rules for threat detection.

3 FISHY validation in Wood-based Panel Trusted Value-Chain

3.1 Introduction

Following the detailed description of the Wood-based Panels Trusted Value-Chain scenario and use cases in deliverable D6.1 [1], the following section describes the work developed to ensure the validation of FISHY in iteration 1 (IT-1), as well as challenges and improvement opportunities detected.

3.2 Wood-based Panel Trusted Value vertical application

As described in D6.1, in the Wood-based Panels Trusted Value-Chain scenario, several components were to be implemented to deliver to the FISHY platform information from three distinct points of the deployed Sonae Arauco's IoT platform, as detailed in the following figure:

(1) – Collects information on Network Infrastructure (WLAN Controller).
(2) – Collects information from the systems devices of the IoT Infrastructure that are located, some on-prem and others in Azure Cloud.
(3) – Collects information on IoT Hub.

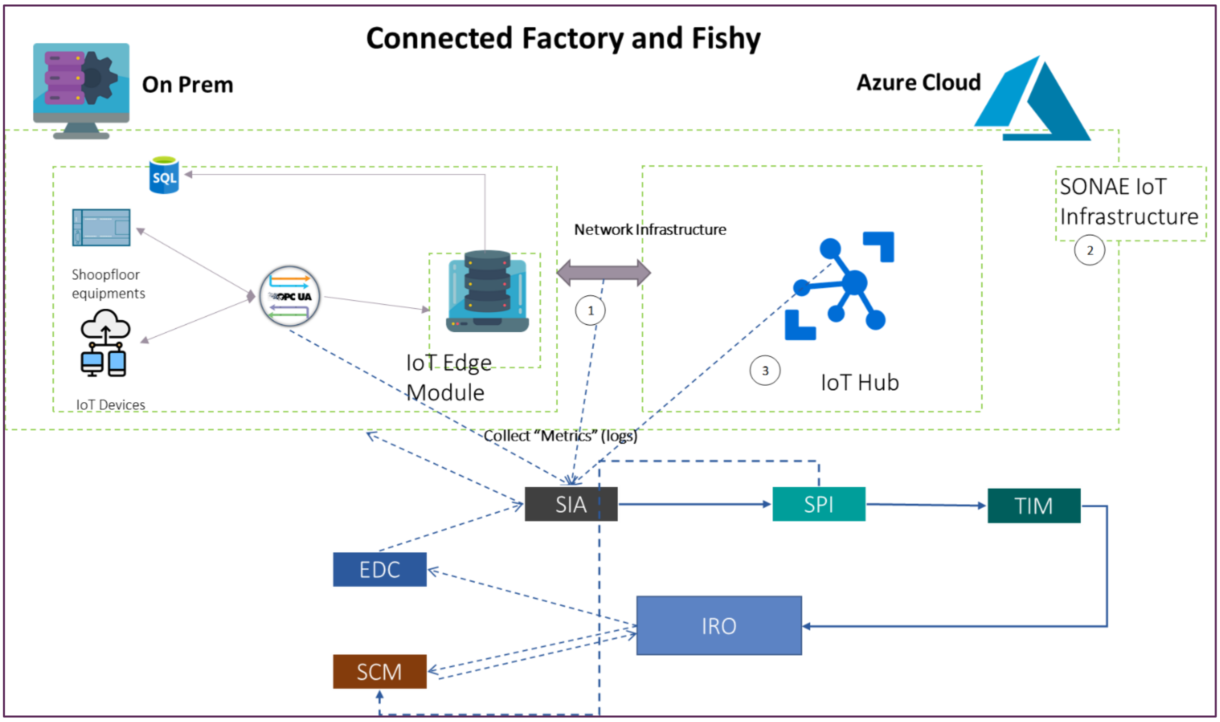


Figure 11. The connected factory architecture and its interconnection with the FISHY Platform

For IT-1, the deployment was focused on the TIM component. This component was used to ensure:

- Vulnerabilities assessment (VAT).
- Risk estimation (RAE).
- Monitoring and testing of supply chain - intrusion detection (XL-SIEM).

Deployment details are described in the following section.

3.3 Security enhancements in WBPTV pilot

For IT-1 in the WBPTV case, three types are distinguished and are meant to detect different types of attacks/threats.

- Type 1: Unauthorised device – IoT device
  - Metadata: {Time Stamp, Client Mac Address, Base Radio Mac Address, SSID, Client IP Address, Message}
- Type 2: Process incident – IoT Hub
  - Metadata: {Time Stamp, Count, total, minimum, maximum, average, resource ID}
- Type 3: Unauthorised access – Windows system
  - Metadata: {Time Stamp, Event ID, User ID; Device ID}

For each type, the following security rules will be applied:

Table 4. Security rules applied per each type of attack

Type	RULE
1	If Attacker IoT device appears in the network (SSID), then • FISHY **notifies/alerts** WBPTV supply chain operator/administrator with the following information (Time Stamp, Client Mac Address (attacker device), Base Radio Mac Address, Message, Client IP Address, Message).
2	If threat IoT Hub appears more than *Threshold2.1* times in *Threshold2.2* hours, then • FISHY **notifies/alerts** WBPTV supply chain operator/administrator providing the relevant log info (Time Stamp, Resource ID, Minimum, Maximum, Average) and/or • FISHY **notifies** process engineer.
3	If Unauthorized access IoT appears in one Windows system, then • FISHY **notifies/alerts** WBPTV supply chain operator/administrator with the following information (Time Stamp, Event ID, User ID, Client IP Address).

Data flows are represented in Figure 12, with the specific section of the architecture highlighted with (number). These threats/attacks mentioned earlier are detected as follows:

For Type 1 attack/threat

- Wlan Controller (5) (Figure 13 and Figure 14) monitors in real time the network and send all events to FISHY (4) via "Cyber-Agent Docker" (3). Whenever an event of a connection and authentication occurs in the network (SSID), FISHY then confirms if the "Client Mac Address" of the device is authorized to connect. If not authorized, FISHY starts the incident process.

For Type 2 attack/threat

- Azure monitors in real time a process and sends events to FISHY (4) via the IoT HUB (Figure 15) through the "Cyber-Agent docker" (3). Whenever an adverse event occurs in IoT platform, FISHY starts the incident process.

For Type 3 attack/threat

- Windows servers (1) exchange event logs information with FISHY (4). Whenever one unauthorized access been identified, FISHY starts the incident process.

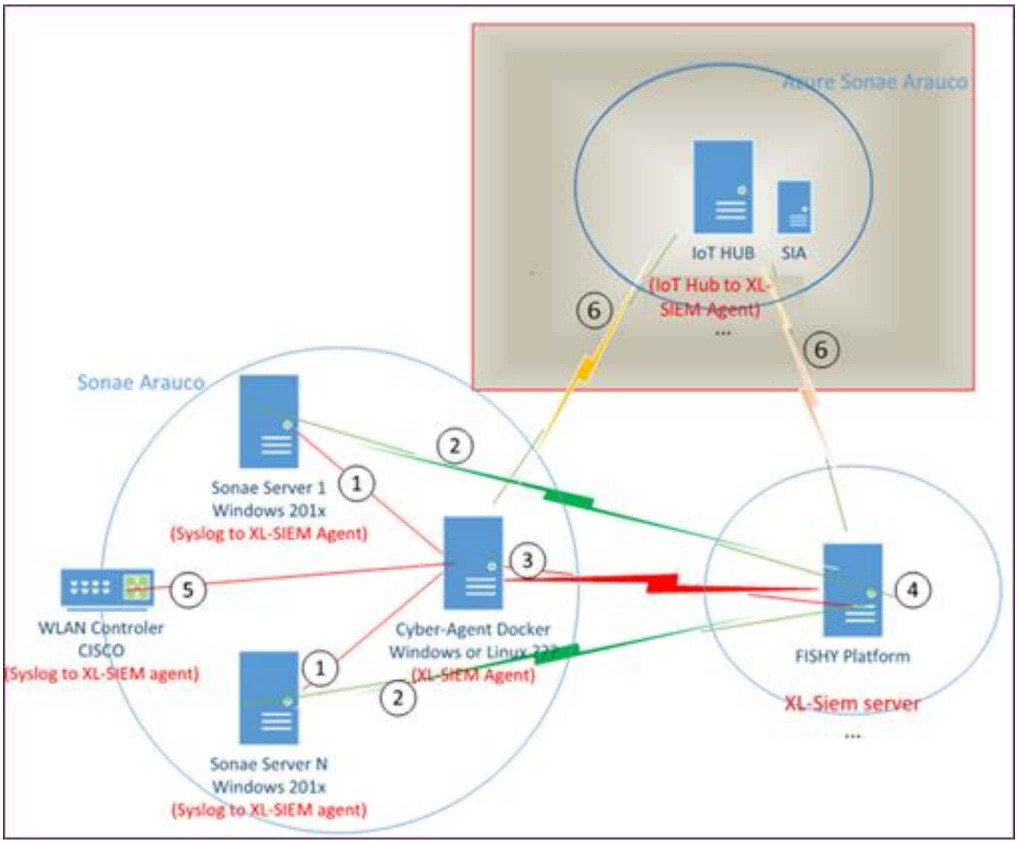


Figure 12. Details of the FISHY platform integrations for IT-1 at the Wood-based panels Trusted value-chain use case

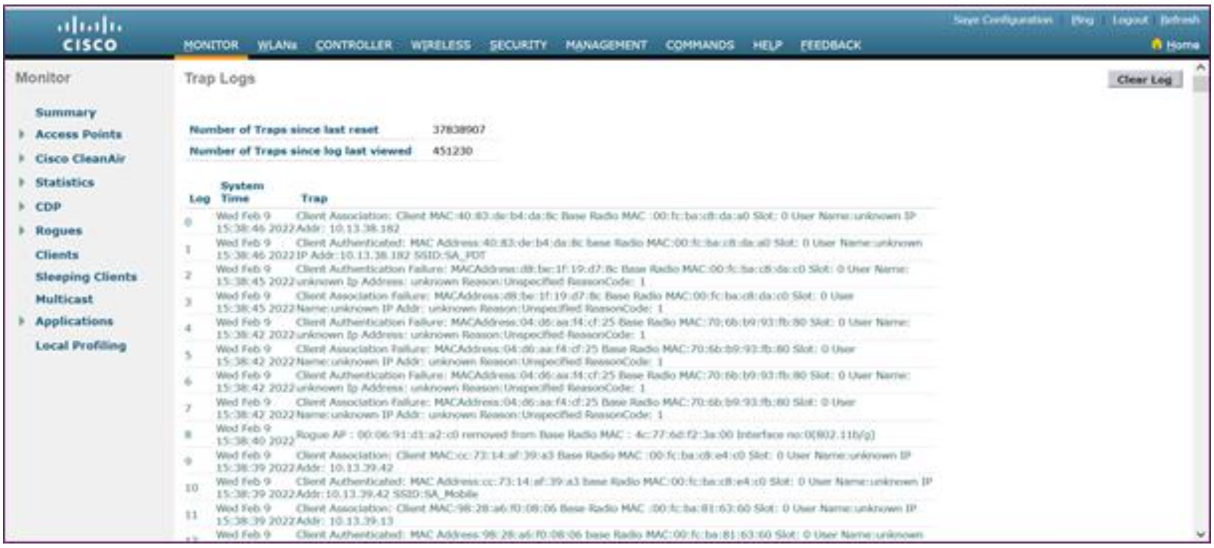


Figure 13. Screenshot of syslog of WLAN Controller sending logs to TIM (XL-SIEM module) – use case scenario 1

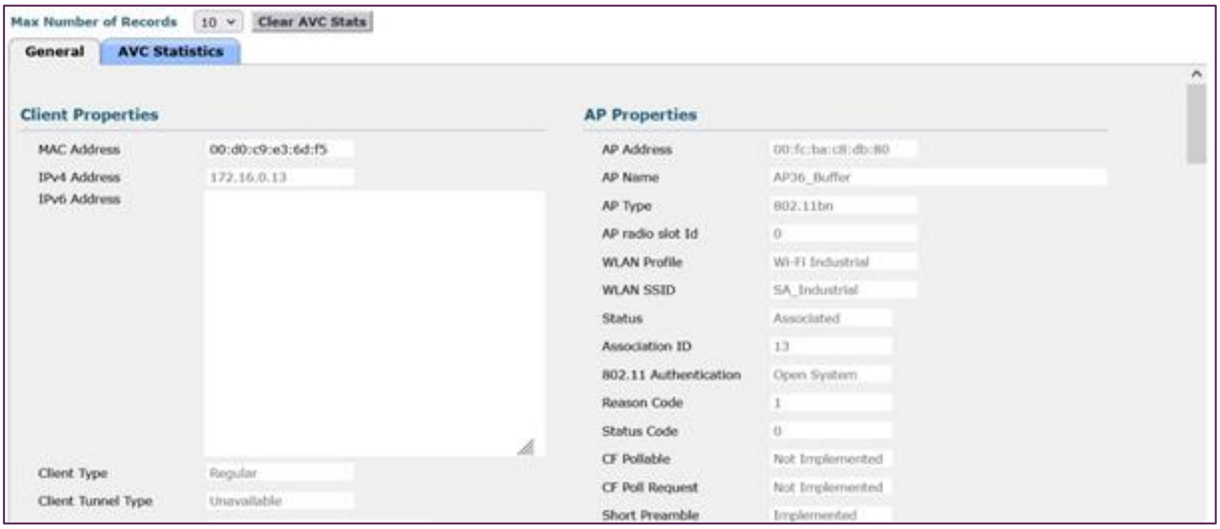


Figure 14. Registered IoT device information set from WLAN Controller to TIM (XL-SIEM module) – use case scenario 1

Figure 15. Telemetry logs sent from IoT Hub to TIM (XL-SIEM module) – use case scenario

3.4 Feedback

This initial iteration was useful to signal some important topics, both from the technical and non-technical perspective, that are highly relevant in a project such as FISHY.

From a technical perspective, SONAE ARAUCO's infrastructure is reliant on Microsoft technology and Windows machines. This proved a challenge when it came to integrate some modules of FISHY that were designed to run using Linux machines. Project partners were, however, able to overcome these obstacles.

Also, integrating the different modules proved somewhat challenging to all partners involved. The reason is two-fold: (i) on the one hand, it was not always easy to access the right documentation to know how to successfully deploy each of the modules; (ii) on the other hand, FISHY is a very ambitious project in the sense that the first real-context validation stage happens quite soon, considering that this is an R&D project. This means that preparation of the validation work started whilst the architecture refinement and integration work were also progressing, which actually implied that a lot of effort was demanded from both, use case partners and technical partners, to put everything together and ensure that all spoke the same language and had the same understanding of what would be the outcome of iteration 1 (IT-1). Nonetheless, that effort was guaranteed by all partners involved thus ensuring the right conclusion of IT-1.

This feedback will surely be helpful not only in shaping iteration 2 of the FISHY validation, but also when it comes to considering the exploitation strategy of FISHY.

4 FISHY validation in Securing Autonomous Driving Function at the edge (SADE)

4.1 Introduction

In this chapter we will focus on the validation of the SADE use case for iteration 1 as described in deliverable D6.1 [1].

In this first iteration, the use case we validate is the third one. This use case focuses on certifying software versions that are safe according to the manufacturer of the IoT device managed by the FISHY platform.

We have also focused on deploying the necessary services for this validation in the 5Tonic environment, the 5G laboratory located in Leganes (Spain).

4.2 SADE vertical application

For the validation of the use case, we have an ecosystem of services. These services are deployed at different points of the infrastructure, depending on the need and the nature of the services. We can differentiate several points for this validation: Cloud (5Tonic), 5G EDGE (ENSCONCE) and the outside world.

Several services will be in the cloud, others in the EDGE where the vehicles are connected. Finally, the vehicle will be connected in the outside world via 5G.

Next image shows a diagram of the services required for the validation of SADE UC3 for iteration 1.

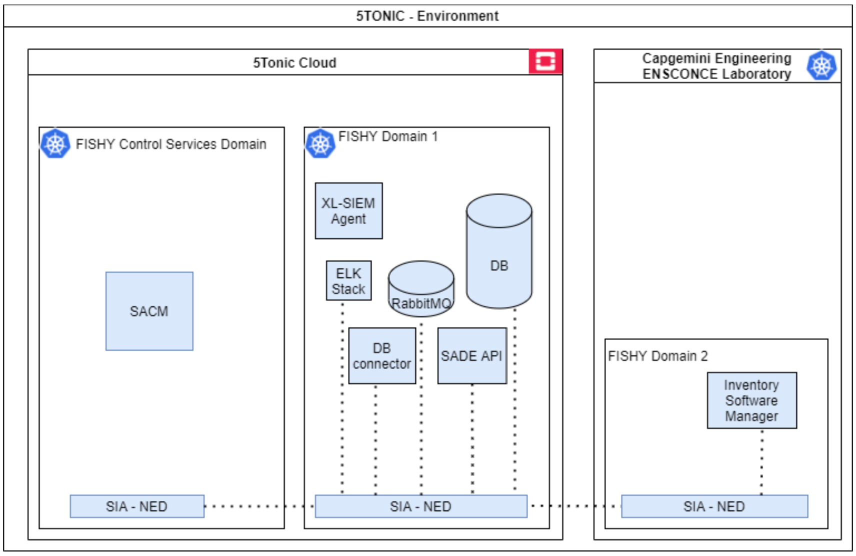


Figure 16. Services that will be deployed for the validation of SADE use-case IT-1

In this first validation, the tools that we will be able to integrate are the SIA and the SACM.

The first component (SIA) is the core of the ecosystem of services that we are going to deploy.

We will have 3 Sandboxes representing 3 different domains (FISHY control services, FISHY domain1, FISHY domain2).

The services required for SADE use case validation are deployed in different domains:

- Fishy control services (Cloud 5Tonic Leganés):

- SACM

- Fishy domain 1 (Cloud 5Tonic Capgemini Engineering Lab Madrid):
  - RabbitMQ
  - ELK stack
  - SADE API
  - Database
  - Database API Connector

```
admin-fishy@fishy-domain-1:~$ kubectl get pods
NAME                                READY   STATUS
fishy-db-7d5c79fd5d-f2gb8           1/1     Running
fishy-db-connector-68948f456b-xcgqk 1/1     Running
fishy-elk-56bc6ff99b-l6tlb          1/1     Running
fishy-rabbit-cdc7d774-8qhml         1/1     Running
fishy-sade-api-6c55b69fb7-ptx2x     1/1     Running
ned-domain-1                        1/1     Running
admin-fishy@fishy-domain-1:~$
```

Figure 17. List of services deployed inside the SIA Fishy-domain-1 related to SADE use case

- Fishy domain 2 (ENSCONCE 5G EDGE 5Tonic Capgemini Engineering Lab Madrid):
  - Software Inventory Service

```
root@CONTROLLER8935:~# kubectl get pods --all-namespaces |grep fishy
ec-u570230f98000010-5744e5b91000006   ec-fishysia-5d87bb9d85-rdf8l                1/1   Running
ec-u570230f98000010-5744e5b91000006   ec-fishyswclisuperc-55f84f5c86-8gkd4        1/1   Running
ec-u570230f98000010-5744e5b91000006   ec-fishyswsrvsuperc-7f8bf55f7-25hns         1/1   Running
root@CONTROLLER8935:~#
```

Figure 18. List of services deployed as the FISHY-donmain-2

The integration with the SIA is based on a deployment of services inside the sandboxes of each domain.

Each service has a fixed IP address. Communication between the services uses this address, that belongs to a specific subnet and it is managed by the NED of the SIA domain. Cloud deployments are done as virtual machines using Openstack by default. The most complicated integration concerns the integration inside the EDGE as part of the ENSCONCE platform (a Kubernetes-based Edge Compute Platform).

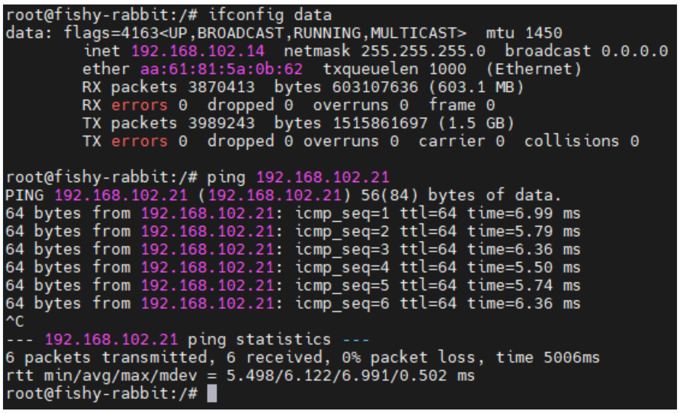
In this first phase of integration, the NED has been separated from the sandbox, deploying the component as a standalone application using the ENSCONCE web portal.

After the provisioning, point-to-point connections have been created between the other FISHY domains (Fishy control services and Fishy domain1). The deployments of the NED and the Software Inventory service have been manually edited to associate some interfaces, allowing the communication through those interfaces with the services located in the three domains.

```
root@ec-fishyswclisuperc-55f84f5c86-8gkd4:/# ifconfig data
data: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1450
        inet 192.168.102.21  netmask 255.255.255.0  broadcast 0.0.0.0
        ether 0e:2c:d6:14:d3:aa  txqueuelen 1000  (Ethernet)
        RX packets 3123079  bytes 335794691 (335.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3196179  bytes 342200967 (342.2 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@ec-fishyswclisuperc-55f84f5c86-8gkd4:/# ping 192.168.102.14
PING 192.168.102.14 (192.168.102.14) 56(84) bytes of data.
64 bytes from 192.168.102.14: icmp_seq=1 ttl=64 time=6.17 ms
64 bytes from 192.168.102.14: icmp_seq=2 ttl=64 time=6.34 ms
64 bytes from 192.168.102.14: icmp_seq=3 ttl=64 time=6.23 ms
64 bytes from 192.168.102.14: icmp_seq=4 ttl=64 time=6.46 ms
64 bytes from 192.168.102.14: icmp_seq=5 ttl=64 time=5.82 ms
64 bytes from 192.168.102.14: icmp_seq=6 ttl=64 time=6.27 ms
64 bytes from 192.168.102.14: icmp_seq=7 ttl=64 time=5.62 ms
^C
--- 192.168.102.14 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6000ms
rtt min/avg/max/mdev = 5.619/6.127/6.456/0.276 ms
root@ec-fishyswclisuperc-55f84f5c86-8gkd4:/#
```

Figure 19. Example of connectivity from a service in Fishy-domain-2 to another one in Fishy-domain-1



```
root@fishy-rabbit:/# ifconfig data
data: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1450
        inet 192.168.102.14  netmask 255.255.255.0  broadcast 0.0.0.0
        ether aa:61:81:5a:0b:62  txqueuelen 1000  (Ethernet)
        RX packets 3870413  bytes 603107636 (603.1 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3989243  bytes 1515861697 (1.5 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@fishy-rabbit:/# ping 192.168.102.21
PING 192.168.102.21 (192.168.102.21) 56(84) bytes of data.
64 bytes from 192.168.102.21: icmp_seq=1 ttl=64 time=6.99 ms
64 bytes from 192.168.102.21: icmp_seq=2 ttl=64 time=5.79 ms
64 bytes from 192.168.102.21: icmp_seq=3 ttl=64 time=6.36 ms
64 bytes from 192.168.102.21: icmp_seq=4 ttl=64 time=5.50 ms
64 bytes from 192.168.102.21: icmp_seq=5 ttl=64 time=5.74 ms
64 bytes from 192.168.102.21: icmp_seq=6 ttl=64 time=6.36 ms
^C
--- 192.168.102.21 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5006ms
rtt min/avg/max/mdev = 5.498/6.122/6.991/0.502 ms
root@fishy-rabbit:/#
```

Figure 20. Example of connectivity from a service in Fishy-domain-2 to another one in Fishy-domain-1

The second component we integrated is the SACM. The integration is based on the deployment of the component. SACM will collect information about the software versions of the IoT devices in the vehicle. This information is stored in the RabbitMQ deployed in Fishy domain 1. SACM will get the list of versions of each component by making a request to the SADE REST API. Once all the data is obtained, it will check if everything is correct or if there is a problem with non-certified versions.

The current integration does not contemplate actions on the infrastructure in this first phase.

Below is the communications diagram, the dotted lines are communications through the SIA/NED.

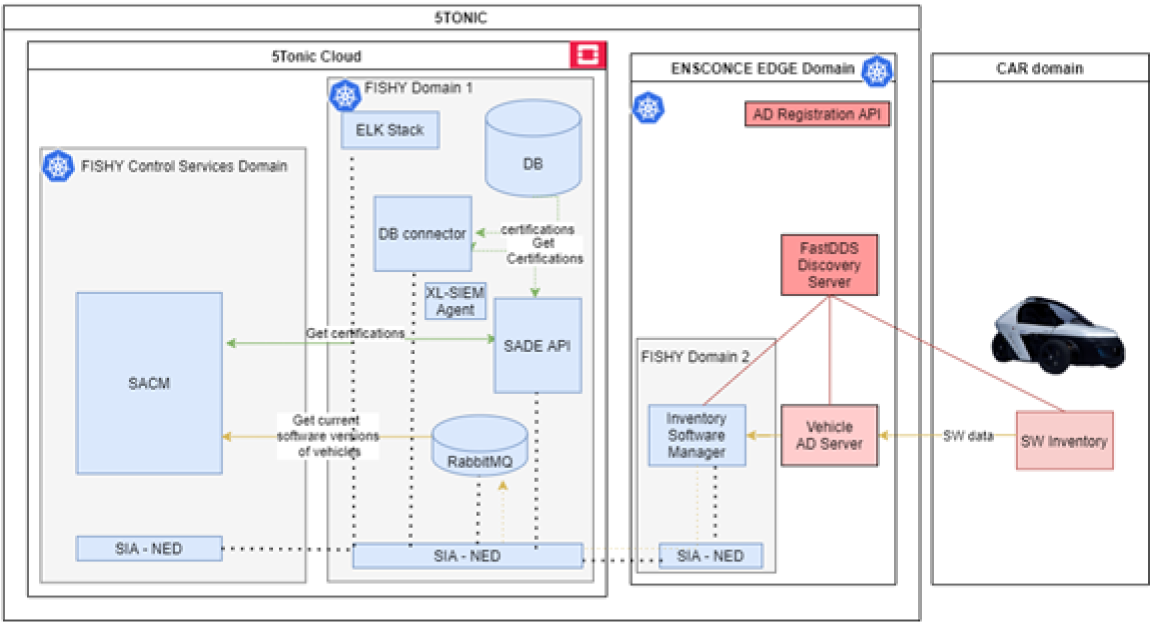


Figure 21. Diagram of the use case 3 of SADE validation for IT-1

Another integration is with RAE and XL-SIEM components. This integration is not related to the UC3 Software Patch certification, XL-SIEM and RAE will be deployed to calculate the cyber risk exposure in the SADE use case.

4.3 Security enhancements in SADE pilot

The following table shows an example of information that OEMs add using FISHY dashboard to certify its software versions. This information is stored in the data base.

Table 5. Example of information OEMs add using the FISHY dashboard to certify their software versions

Model	TempMeterXXX
SW Version	1.1235
Safe Update Link (optional)	https://company.com/updates/TempMeterXXX/1.1235/firmware.bin
Update checksum (optional)	5a000ca5302b19ae8c7a66149f3e1e98

Data from vehicles will be sent to FISHY in the form of a JSON object which will include: UUID (Unique Universal ID, Timestamp (UTC timestamp) and Metadata.


```
{
”metadata”: {
        ”sw_data”: [{
                ”manufacturer”: ”Capgemini Engineering”,
                ”model”: ”TempMeterXXX”,
                ”sw_version”: ”1.1235”,
                ”serial_number”: ”sensor_ht:257d0001XXXX”,
            },
            {
                ”manufacturer”:  ”Capgemini Engineering”,
                ”model”: ”CamSensorXXX”,
                ”sw_version”: ”0.1”,
                ”serial_number”: ”sensor_cam:1d101s”,
            }
            ],
        ”vin”: ”0000-0000-0000-0001”,
        ”timestamp”: ”1624003974”,
},
 “UUID”: “”
}
```


Figure 22. JSON object including vehicle data in SADE use case

SADE will send this information to a RabbitMQ exchange, deployed in the Sandbox of the Fishy domain 1 as a k8s POD.

- SACM must get JSON messages and parses the received information.
- SACM compares with SW certification versions provided by OEMs that can be recovered from the SADE API using REST.

RULES

- There is one rule that checks if one version received is not certified:
  - FISHY notifies/alerts users related to the compromised vehicle.
  - FISHY enforces Update* policy against SADE Service (REST API module)

* If an updated version model is certified and contains a safe link for an update, that link must be provided; if not, our service will start a recall notification. FISHY just does not send any link in the POST request.

On the other hand, data collectors send logs to XL-SIEM. XL-SIEM in turn sends elaborated events and alarms to RAE that can calculate in real-time the cyber risk exposure.

An agent of the XL-SIEM is deployed as part of the FISHY appliance and sends logs for the XL-SIEM to detect those attacks. This agent is in charge of obtaining the log files from a number of services related to SADE use case and will make them available to the RAE.

Log files collected are from:

- RabbitMQ server.
- NGINX + gunicorn SADE API
- NGINX + gunicorn DB connector API

The agent will be deployed in the same CLOUD infrastructure (same domain) as the other services of the use case, allowing access to the logs by mapping volumes to a common directory, which is accessible by the agent.

Actions like software update or send notifications to car's owner won't be implemented for the iteration 1.

4.4 Feedback

This first iteration has been a first approach to how to validate the FISHY platform globally.

Some key points have been identified that pose a challenge in demonstrating full integration. The best thing about having several validation points is that allows this identification and subsequent milestones to be tackled.

As far as the use case is concerned, we have solved some integration difficulties due to the situation we are in and the fact that the components are still in the development phase. However, the great work of the partners has facilitated the deployment of the components and the integration with the use case. Also, the definition of the flows has allowed to consolidate the architecture of the use case solution. The biggest complication comes when it comes to integrate a component as part of the solution platform, as is the case of the SIA. The way to integrate next components will be thoroughly monitored in the following months.

There are still many tools to be integrated and we are well on our way and the focus is well set for the second iteration.

We believe that the beginning of the integration is the most difficult stage of a project. As the project progresses, the integration with the rest of the components will become more fluid.

5 Feedback consolidation

The following table summarizes the most important feedback items collected per FISHY Platform main building block and also for the FISHY Platform as a whole.

Table 6. Feedback consolidation

FISHY Component	F2F	WBPTV	SADE
SCM	Flexible component. Supports multiple rules. A dedicated GUI for the user to define the rules would be appreciated.	Not included in IT-1	Easy to deploy with Docker. Supports multiple rules. Could be difficult to define a rule with a lot of conditions.
TIM	Provides rich functionality enabling also machine-learning based threat detection based on PMEM. Overall, flexible component which supports multiple rules and autonomous anomaly detection.	Used for the purpose of vulnerabilities assessment, risk estimation and intrusion detection in the infrastructure	XL-SIEM is flexible and allows to collect several kinds of logs. Could be extended with plugins to support more log types. Easy to deploy with Docker.
EDC	Policies decided in collaboration between infrastructure operator and EDC designers / developers. Knowledge about the infrastructure is indispensable to make good use of EDC.	Not included in IT-1	Not included in IT-1
IRO	The collection of results and events works properly The operator has clear understanding of what happens in the infrastructure The feature of configuring rules for threat detection would be appreciated	It was used to access analytics, notifications and configuration of different user profiles	Not included in IT-1
SPI	Not included	Not included in IT-1	Not included in IT-1
SIA-NED	Not included	Not included in IT-1	Its integration was not easy with other platforms different than the sandbox. Could be great to have a DHCP in the sandbox to

FISHY Component	F2F	WBPTV	SADE
			assign IP addresses to the components. Auto configure on boot would be appreciated, when the Sandbox is rebooted, all connectivity is lost in the services.
FISHY Platform	The integration run smoothly and using RabbitMQ for information exchange was a good choice. Good impression overall of the FISHY Platform and its potential	Problems derived from the use of Windows were overcome Early piloting implied the need for close communication with ongoing development and integration activities. Good impression overall of the FISHY Platform and its potential	Good impression overall of the FISHY Platform and its potential. Once all the tools are integrated as a single platform, the possibilities and ease of use will be significantly increased.

6 Conclusions

In this document we have described the deployments made in the infrastructures of the three pilot partners to connect those infrastructures to the FISHY Platform. In addition, some elementary integration tests have been run, collecting evidence about the correct interplay among the components chosen by each pilot. These tests have been based on what was envisioned and designed in T6.1 and T6.2, eventually documented in D6.1. Being elementary, they reflect some of the challenges the pilots are facing in terms of supply chain cyber resilience. They are a good starting point for more complex use cases within the pilots that will be addressed during the next months. Specific demos will be built within WP6 for showcasing the platform in different contexts.

The pilot partners have compiled a first wave of feedback which is highly useful for the technical activities in WP2, WP3, WP4 and WP5. There is a positive overall impression and the FISHY concept has shown to make sense. The early PoCs create a good rationale to state that there is a way forward and that the effort put into the project is worthwhile. The different tools are used in at least one of the pilots and are expected to be integrated in more during IT-2.

At this point the four tasks of WP6 will run in parallel until M32 (April 2023). During this time there will be a close loop between the use case settings and the definition of the demo strategy (in T6.1 and T6.2) on one side, and the execution of tests and validation of components and results on the other side (in T6.3 and T6.4). In M24 (August 2022) D6.3 will be submitted as an intermediate milestone of T6.1 and T6.2. Once T6.1 and T6.2 will conclude in M32, then T6.3 and T6.4 will continue until the end of the project in M36 (August 2023), when D6.4 will be submitted and WP6 will close.

7 References

[1] FISHY, D6.1 “Use cases settings and demonstration strategy”, 2021