
# Secure management of IoT devices lifecycle through identities, trust and distributed ledgers

# D2.3 Updated Threat Modeling Module

## Document Summary Information

| | | | |
|---|---|---|---|
| **Grant Agreement No** | 101020416 | **Acronym** | ERATOSTHENES |
| **Full Title** | Secure management of IoT devices lifecycle through identities, trust and distributed ledgers | | |
| **Start Date** | 01/10/2021 | **Duration** | 42 months |
| **Project URL** | www.eratosthenes-project.eu | | |
| **Deliverable** | D2.3 - Updated Threat Modeling Module | | |
| **Work Package** | WP2 | | |
| **Contractual due date** | 30/06/2023 | **Actual submission date** | |
| **Nature** | Software | **Dissemination Level** | PU |
| **Responsible author** | Dimitri Van Landuyt (KUL) | **Lead Beneficiary** | KUL |
| **Authors** | Dimitri Van Landuyt (KUL), Stef Verreydt (KUL) | | |
| **Internal reviewers** | Rohit Bohara (DW), Blaž Podgorelec (TUG) | | |

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No 101020416.

### *Revision history (including peer reviewing & quality control)*

| Version | Issue Date | % Complete | Changes | Contributor(s) |
|---|---|---|---|---|
| v0.1 | 23/05 | 10% | Initial Deliverable Structure + introduction | Dimitri Van Landuyt (KUL) |
| v0.2 | 25/05 | 15% | Work on outline and deliverable scope | Dimitri Van Landuyt (KUL) |
| v0.4 | 01/06 | 20% | Details and overview of simulators (first version) | Dimitri Van Landuyt (KUL) |
| v0.5 | 05/06 | 25% | MQTT topics table (first version) | Dimitri Van Landuyt (KUL) |
| v0.6 | 10/06 | 50% | Inclusion of step-by-step tutorial | Stef Verreydt (KUL) |
| v0.7 | 13/06 | 75% | Discussion of adaptation layer design | Dimitri Van Landuyt (KUL) |
| v0.8 | 14/06 | 80% | Scientific positioning + conclusion | Dimitri Van Landuyt (KUL) |
| v0.9 | 15/06 | 86% | Finalization of deliverable for internal review | Dimitri Van Landuyt (KUL) |
| v0.95 | 20/06 | 90% | Addressing internal reviewer #1 comments | Stef Verreydt (KUL) |
| v0.97 | 21/06 | 95% | Addressing internal reviewer #2 comments | Stef Verreydt (KUL) |
| v1.0 | 22/06 | 100% | Release | Dimitri Van Landuyt (KUL) |

### *Disclaimer*



### *Copyright message*

# Table of Contents



# List of Figures



# List of Tables

# Glossary of terms and abbreviations used

| Abbreviation / Term | Description |
|---|---|
| API | Application Programming Interface |
| C2V2X | Cellular V2X |
| DFD | Data flow diagram |
| DID | Device ID/Device Identifier |
| DSRC | Dedicated Short-Range Communications |
| EMF | Eclipse Modeling Framework |
| GEMV2 | Geometry-based, Efficient propagation Model for Vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication |
| GUI | Graphical user interface |
| ICTM | Instance-Centric Threat Modeling |
| IEEE | Institute of Electrical and Electronics Engineers |
| ITS-G5 | European standard for vehicular communications (IEEE 802.11p) |
| MQTT | OASIS messaging standard designed for IoT |
| PDP | Policy decision point |
| PoC | Proof-of-concept |
| PTV Vissim | Traffic flow simulation software package developed by PTV Planung Transport Verkehr AG |
| SPARTA | Security and Privacy Threat Modeling for Automated Threat elicitation and Risk-driven Threat prioritzation |
| TA | Trust agent |
| TL2V | Traffic light-to-Vehicle |
| TMB | Trust Manager & Broker |
| TMRA | Threat modeling and risk assessment |
| V2TL | Vehicle-to-Traffic light |
| V2V | Vehicle-to-Vehicle |
| V2X | Vehicle-to-Any |
| VEINS | Vehicles In Network Simulation |

# 1 Executive Summary

An update of the development state of the Threat Modeling and Risk Assessment (TMRA) service is provided. This work extends upon the results presented prior in Deliverable D2.2: "Prototype of Threat Modeling and Deployer of Trust Agents".

More specifically (and in line with the earlier deliverable),

- A technical and implementation-centric discussion is provided of the TMRA component with specific attention on integration and adoption (e.g., in the pilot studies) and the architectural role of an adapter component in this regard.
- Reference and explanation are given on the proof-of-concept implementations that have been made available, with emphasis on how to execute and adopt (tutorial steps).
- Roadmap and outline for experimentation and further evaluation is discussed, focusing on feasibility and performance impact.

# 2 Introduction

The Threat Modeling and Risk Assessment (TMRA) service represents one of the key components in the broader umbrella of the trust-enabling technologies development in WP2 of ERATOSTHENES. TMRA maintains a digital twin representation of the system in the form of application-level DFDs that are continually synchronized with the running and operational system at the basis of system events of relevance (e.g., announcement of new devices, specific access requests being made). Levering automated threat elicitation technologies, a threat model is automatically maintained which is capable to report on risks on various levels: at the level of elements, at the level of the nature of the risk (threat type), and aggregated values at the level of the entire system. These risk scores can be used to inform specific decisions, to influence the trust calculation algorithms and as a motivator to make adaptive decisions to dynamically reduce risk (e.g., enable specific security controls). As such, the TMRA component itself is a central element of a self-adaptive risk-aware security mechanism which is capable of making more optimal dynamic decisions at run time, as opposed to the static one-size-fits-all decisions otherwise made at design time. The deliverable describes the second development iteration of the TMRA component (M15 to M21). Whereas the first iteration had focused on initial prototyping (cf, Deliverable D2.2 Prototype of Threat Modeling and Deployer of trust agents), this development iteration focused on architectural extension towards better adoptability and integrability of the component (Application Adapter layer) and providing the necessary resources (step-by-step tutorial) to promote the potential uptake in the ERATOSTHENES pilot studies. Finally, we outline ongoing activities on scientific evaluation of the TMRA component.

## 2.1 Mapping ERATOSTHENES Outputs

Purpose of this section, is to map ERATOSTHENES Grand Agreement commitments, both within the formal Deliverable and Task description, against the project's respective outputs and work performed.

Table 1: Adherence to ERATOSTHENES GA Deliverable & Tasks Descriptions

| ERATOSTHENES GA Component Title | ERATOSTHENES GA Component Outline | Respective Document Chapter(s) | Justification |
|---|---|---|---|
| **DELIVERABLE** | | | |
| D2.3. Updated Threat Modeling Module [KUL, M21, Other (Software)] [from: Chapter 3, section 3.1.2, Description of the Work packages (WP2)] | This deliverable includes the updated outputs from T2.2 about threat modeling module of TBM for dynamic end-to-end threats analysis. | Outcomes of the second development iteration are presented in Section 4. | T2.2 focuses on a novel per-instance threat analysis approach. Deliverable D2.2 presented the first prototype outcomes. In an incremental fashion, this deliverable presents the outcomes of the subsequent phase of demonstrator and prototype development (M14 to M21) |
| **TASKS** | | | |

| | | | |
|---|---|---|---|
| Task 2.2: Threat modeling for continuous Risk Assessment [Chapter 3, section 3.1.2, Description of the Work packages (WP2)] | This task will establish key innovations in (self-adaptive) threat modeling automation, as well as the development and validation thereof. These include: (a) Middleware for the construction, maintenance and evolution of a system model that is reflective of the current state of the operational distributed IoT system. | Section 4.2 | The MQTT connector that is part of the demonstrator and creates a shadow model (digital twin) of the system. |
| | This model will be based on abstractions that are common in threat modeling (data flows, data stores, entities) and will leverage additional information to establish trust boundaries. In addition, overlays will be created to attach trust scores to each of these elements. The necessity of employing different abstractions such as individual sensors or sensor management capabilities will be investigated. | Section 4.2 | A traditional DFD is extended with an instance-layer which is modeled in the same abstractions. The translation of application-level events into specific edits/manipulations of the instance-layer model is tackled in this deliverable. |
| | For this, different reflective agents will be deployed across the distributed, multi-organisational IoT infrastructure that can, in a decentralized fashion, share relevant state and structure information about their operational environments. | NA | Distributed and large scale deployment of these connectors will be investigated in future development of the TMRA framework. |
| | (b) A threat elicitation engine able to act upon this system model and generate the different threats that are theoretically applicable or feasible, based on knowledge bases which are augmented with knowledge on weaknesses/common-threats in the IoT context (cfr. Task 1.1). [..] | Section 4.2 | The current implementation performs per-instance threat elicitation, generates possible threat scenarios and supports risk calculation on a per-device manner. Next extensions will be provided in terms of the supported knowledge bases and in function of the application cases that drive the effort. |

## 2.2 Deliverable Overview and Report Structure

The deliverable starts with an executive summary (sect. 1). This Section (sect. 2) introduces the deliverable, explains the position of its content in the broader context of the Description of Work (DoW), maps the discussed prototype and technology to the ERATOSTHENES outputs, and discusses how the reviewer comments were addressed. Section 3 then provides an in-depth positioning of the work in the context of the overarching architecture as developed in WP1.

The main deliverable context (incl. description of the work and improvements made since the previous version) is provided in Section 4. This section gives a step-by-step on how to run the technology, discusses the architectural translation layer introduced in the previous development iteration, and outlines plans and selection of simulation environments for extensive validation and evaluation.

Section 5 discusses the scientific novelty and innovation of the work, whereas Section 6 concludes the deliverable.

## 2.3 Progress Since the First Deliverable Version

The development iteration reported upon in this deliverable focuses on improvements made to the TMRA component that fit in four different categories:

1. **Expressiveness**: The TMRA component has been updated to work with a more comprehensive and complete list of MQTT events (cf. Table 3).
2. **Documentation**: a step-by-step tutorial is provided on how to install, deploy and use TMRA.
3. **Architecture**: Design has been improved, in terms of adding a capability to flexibly translate application-specific MQTT events into generic and reusable model changes (adaptation layer).
4. **Evaluation**: Study & Exploration of viable V2X simulation environment for more extensive evaluation and validation of the TMRA component; planning of and initial steps towards extensive experimental evaluation of the TMRA component is discussed.

## 2.4 Adherence to 1st EC Review Comments and Recommendations

*This section applies to those deliverables that are submitted after the first review.*

This section summarizes responses and document updates following the EC review that took place on xxx. All reviewers' comments were effectively taken under consideration and details for each (and the related document updates) have been included in the table below:

| **Review Comment(s)**<br>(as provided by the reviewers) | **Adherence and Document Update**<br>(Short reply and reference to the chapter that details the reply) |
|---|---|
| Availability and accessibility of the prototype and its source code. | The docker image for the threat modeling and risk assessment PoC has been made available on the project Docker image registry (Nexus[1]), the image for this deliverable is *"tmra:mqttv3-0.2"*. This initial packaged version is pre-populated with a DFD of the Smart Vehicles case and some example devices.<br><br>The source code has been synchronized in the project's GitLab environment (https://ci-cysec.eng.it/gitlab/ERATOSTHENES/tmra). |
| | |
| | |

[1] https://ci-cysec.eng.it/nexus/

# 3 Architecture Orientation and Industrial Requirements

This section discusses the role of the TMRA component in the broader ERATOSTHENES architecture.

## 3.1 Architectural Positioning and design decisions

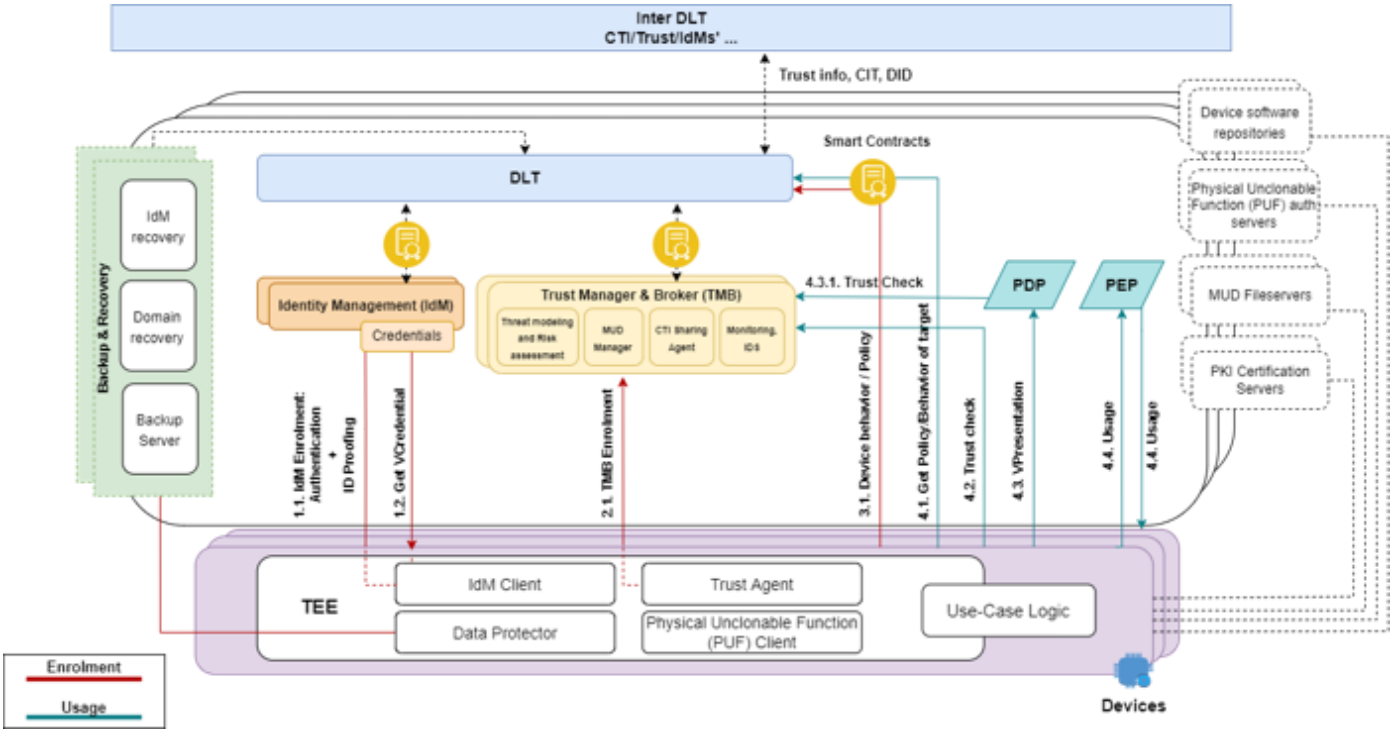


Figure 1. Graphical overview of the overall ERATOSTHENES architecture (adopted from Deliverable D1.3). TMRA is part of the Trust Manager & Broker, depicted at the center of the figure.

This deliverable provides the description of a key component in the edge layer of the ERATOSTHENES technology stack of which the overall architecture is depicted in Figure 1.

The **Threat Modeling and Risk Assessment (TMRA)** module is part of the Trust manager & broker sub-system (TMB) and maintains a digital twin system model of the devices (instances) in complement to a system model that characterizes the overall design of a case (system layer). At the basis of both models, a threat-centric risk assessment is performed, and the TMRA will emit a risk score per device, specific to threat types (e.g., identity spoofing, tampering with interactions by an intermediary, tampering with software on the IoT device itself, etc).

The instance-centric model maintained in TMRA is kept in sync with both (i) the class-level design level (DFD) and (ii) the operational system. The first type of constraints is kept by instantiating meta-model elements and adhering to the meta-model constraints as imposed by the design model. The second type of synchronization is performed through a comprehensive and expressive layer of MQTT topics and events to which the TMRA component subscribes.

This risk score can then be used in the Trust Manager and Broker (TMB) to calculate an overall trust score that is then consulted by the PDP dynamically to regulate access to devices/information and the system or application as a whole.

## 3.2 Business, Industrial Positioning and End-User Requirements

*The positioning of this component hasn't changed. For convenience, we reprint the positioning that was already presented in D2.2.*

The Threat Modeling and Risk Assessment (TMRA) module performs risk-centric calculations at the basis of a system model that is co-maintained and co-evolved with the operational system (a digital twin model). It provides application-level specification/design of the system and augments it with per-instance models in which concrete device details are kept. As such, this mechanism contributes to the trust and risk calculation that is performed by the Trust Management

Broker (TMB), and this technology contributes to addressing the requirements listed in Table 2 (adopting the numbering scheme from Deliverable D1.2).

Table 2: Mapping between Pilot case requirements and the TMRA component.

| Req ID | Description | How addressed |
|---|---|---|
| P1_NFR_03 | Accuracy of trust model to estimate direct/indirect trust attributes | TMRA calculates a risk score on a per-device/process or system entity basis and allows this at the level of specific threat scenarios (e.g. spoofing, tampering, etc). This detailed analysis and its reported outcomes allows for a fine-grained expression of trust. |
| P1_NFR_13 | Consistent representation of trust and trust relationships in automotive (vehicle and smart city) IoT networks | Trust relations between vehicles and between vehicles and infrastructural components can be expressed in TMRA and taken into account in the risk calculations. |
| P2_FR_01 | Consistent representation of trust relationships in personalized health devices | TMRA keeps track of trust-enhancing technologies and all sorts of trust-related information. These are expressed in the DFD-centric model that is kept on a per-device basis, allowing to express differences and interrelations between devices in this regard. |
| P2_FR_02 | Per-device trust calculation and calibration of trust in devices | TMRA keeps track (in the instance layer) on a per-device basis, which elements of trust-enhancements are active and evaluates how they reduce security risk from a threat modeling perspective. |
| P2_FR_06 | The trust score measurement is assigned to every device and service, internal or external to the infrastructure. | TMRA keeps track of trust-enhancing technologies and all sorts of trust-related information. These are expressed in the DFD-centric model that is kept on a per-device basis, allowing to express differences and interrelations between devices in this regard. |
| P2_FR_07 | Threat analysis depending on trust and context must be held to decide whether interactions/data sharing will be allowed | TMRA keeps track (in the instance layer) on a per-device basis, which elements of trust-enhancements are active and evaluates how they reduce security risk from a threat modeling perspective. It also contextualizes interactions between devices and takes the potential impact into account. |
| P2_FR_11 | Initial Trust Score assignment to IoT devices during the enrollment phase | The first version of the TMRA prototype described in this deliverable focuses on bootstrapping of the models. This is done at the basis of device announcement MQTT events. When new devices emerge, an initial model instance is created with trust values that reflect initial uncertainty. |
| P2_NFR_05 | Accuracy of trust model to estimate direct/indirect trust Attributes | Although more concrete trust attributes still need to be articulated and validated at this stage, as the shadow model of a device is maintained at the basis of actual events (life-cycle events or interactions), we will be able to express changes in trust correctly. |
| P3_NFR_04 | Accuracy of trust model to estimate trust attributes of 3rd-devices | Although more concrete trust attributes still need to be articulated and validated at this stage, as the shadow model of a device is maintained at the basis of actual events (life-cycle events or interactions), we will be able to express changes in trust correctly. |

## 3.3 Methodology

As this development iteration has focused on adoptability and integrability, we have taken an approach of architectural design to improve the aspects of coupling to application-specific details, and separation of concerns. The implementation has also focused on versatility of the core primitives to express model edits in the application-level DFD. In addition, we have improved documentation of code and included a tutorial.

## 3.4 Code Availability

The source code for the TMRA component is available at https://ci-cysec.eng.it/gitlab/ERATOSTHENES/tmra . The repository includes a readme file with general instructions.

# 4 Threat Modeling and Risk Assessment service (TMRA)

As discussed above in section 3, the TMRA component represents an operational service that is subscribed to a number of relevant MQTT topics and updates the digital twin threat model accordingly. Based on a necessary degree of expressiveness and accuracy, TMRA can then provide a run-time assessment of risk, specific to individual devices or components, and specific in terms of the nature of the potential threat being considered (spoofing, tampering, etc).

The current PoC is based on the connected vehicles case. Here, an overview is provided on the specific steps taken for the construction, maintenance and evolution of a system model that is reflective of the current state of a connected vehicles system. The overall structure of the TMRA is shown in Figure 2. A step-by-step tutorial on how to use the current PoC is provided in section 4.1. The application adapter is discussed in section 4.2, and section 4.3 describes the ongoing approach to test and validate this component by leveraging existing simulations of V2X environments.

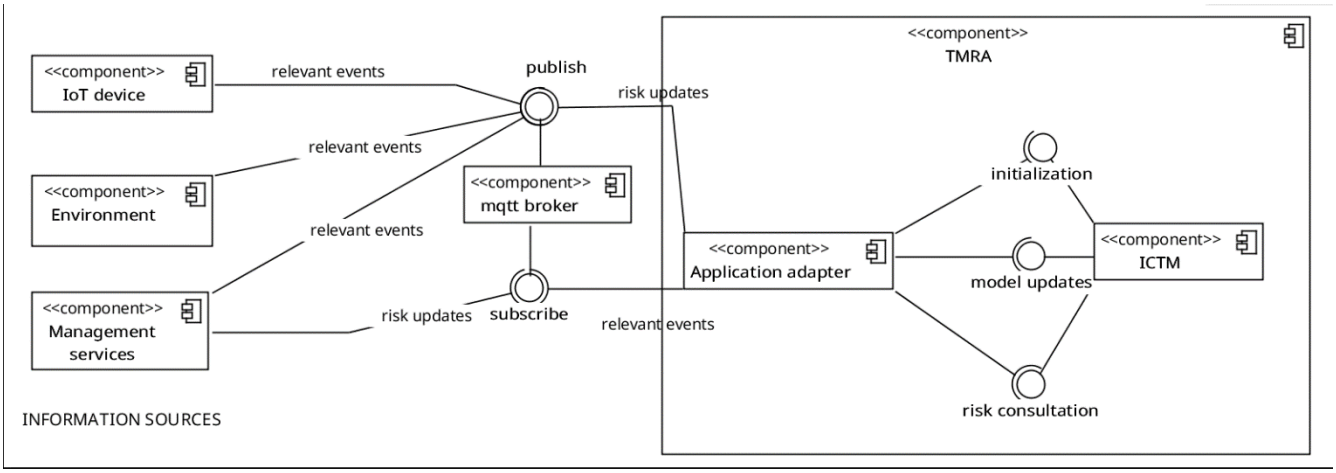


Figure 2: Architectural overview of TMRA, in relation to diverse information sources (depicted on the left) and the MQTT broker which governs the flow of information.

## 4.1 Step-by-step tutorial

The following subsections detail how to use the demonstrator version of the TMRA components.

### 4.1.1 Setup

Testing the TMRA component requires only a running MQTT broker. Example setups for an MQTT broker and the TMRA component itself are provided shortly here.

#### *4.1.1.1 MQTT broker*

The TMRA component was developed with MQTT version 3 in mind (not version 5). For our tests, we ran the eclipse-mosquitto MQTT broker (which supports both versions 3 and 5) locally in a docker container, as explained here: https://hub.docker.com/_/eclipse-mosquitto . Concretely, the command used to start the MQTT broker container is:

*docker run -it -p 1883:1883 -p 9001:9001 -v*
*"path/to/conf/mosquitto.conf":/mosquitto/config/mosquitto.conf eclipse-mosquitto*

Where the mapped ports denote how to reach the container via the host (I.e., via ports 1883 and 9001 on the host machine), and *path/to/conf/mosquitto.conf* points to a file with the following contents:

*persistence false*
*log_dest stdout*

*allow_anonymous true*
*connection_messages true*

*listener 1883 0.0.0.0*

#### 4.1.1.2 TMRA

To run the TMRA container (*tmra:mqttv3-0.2* on the ERATOSTHENES docker repository), the following command can be used:

*docker run --rm --name tmra tmra:mqttv3-0.2 broker-ip:1883*

Where the final parameter (*broker-ip:1883*) points to the MQTT broker. For example, if the broker is run locally in a docker container as explained above, the full command would be

*docker run --rm --name tmra tmra:mqttv3-0.2 "tcp://host.docker.internal:1883"*

### 4.1.2 Workflow

#### 4.1.2.1 Supported topics

The current TMRA component supports the following MQTT topics:

Table 3: TMRA MQTT topics

| Event | Topic | Payload |
|---|---|---|
| **New device** | `device/deviceID/announcement/new` | Device type (currently only 'Vehicle') |
| **Device removed** | `device/deviceID/announcement/remove` | |
| **Device sends data to other device** | `dataflow/deviceID/announcement/new` | SenderID;RecipientID |
| **Device parameter changed** | `device/deviceID/announcement/parameter/parameter_name` | New parameter value |

The *deviceID* is, as the name suggests, the ID of the device to which the event relates. For example, if a new device with ID *'D1'* would enter the system, then the topic would be *device/D1/announcement/new*. In the case of the dataflow event, the deviceID relates to the sender of the data, and the ID of the recipient is passed in the payload. For example, if *'D1'* sends data to *'D2'*, then the topic is *dataflow/D1/announcement/new*, and the payload would contain *'D2'*.

#### 4.1.2.2 Using the PoC

The TMRA is subscribed to the topics listed in Table 3 for all potential devices using wildcards, (e.g. *device/.*/announcement/new*). To test the TMRA component, al that needs to be done is to send events to the MQTT broker. The actions taken by the TMRA for each of the topics are described shortly in what follows.

In general, if an event is relevant for the threat model, the application adapter will a) trigger the required model updates and risk recalculation with the ICTM b) query the ICTM component for the threats and risk scores, and c) publish those outputs to the MQTT broker.

**New Device.** The model in the PoC was prepopulated with two vehicles and a traffic light, as shown below.

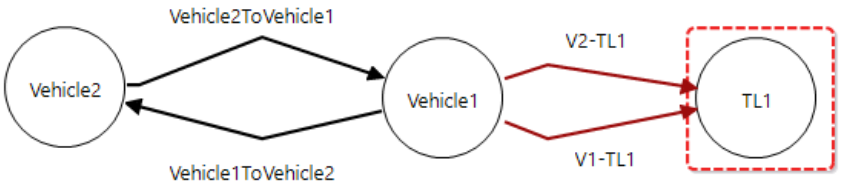


Figure 3: Prepopulated model

Now, if a new vehicle enters the system, for example *'Vehicle3'*, the following event would be published to the MQTT broker:

*device/Vehicle3/announcement/new*

The payload would contain the string 'Vehicle' to denote that the newly added device is a vehicle (and not a traffic light, for example). The broker forwards that event to the TMRA component, as it is subscribed to *device/.*/announcement/new*. The application adapter then parses this event (i.e., extracts the ID from the event) and updates the model. Then, the application adapter instructs the ICTM component to recalculate the threats and their risks. Finally, these risk values are published to the MQTT broker. For each vehicle, the application adapter publishes an event containing the total risk for that element, as well as the risk per threat type. Note that risk calculation is work in progress. With the current PoC, risk calculation always returns zero. The full console output produced by the TMRA component is shown in Figure 4.

```
New message received device/Vehicle3/announcement/new
topic = device/Vehicle3/announcement/new
Adding new vehicle.
Vehicle 'Vehicle3' added : ConnectedVehiclesModel.impl.VehicleImpl@401d22f4 (name: Vehicle3, description: null) (uuid: 045a7fcc-e81b-40ba-83cd-a6ad5215c2b2, time_last_auth: Wed Jun 14 17:05:09 UTC 2023)
Recalculating risk...
36 threats found:
  - Spoofing threat on ConnectedVehiclesModel.impl.VehicleToVehicleImpl@6dc17b83 (name: Vehicle1ToVehicle2, description: null) with a risk of 0 (0,0)
  - Spoofing threat on ConnectedVehiclesModel.impl.VehicleToVehicleImpl@5e0826e7 (name: Vehicle2ToVehicle1, description: null) with a risk of 0 (0,0)
  - Tampering threat on ConnectedVehiclesModel.impl.VehicleToTrafficLightImpl@3ce1e309 (name: V1-TL1, description: null) with a risk of 0 (0,0)
  - Spoofing threat on ConnectedVehiclesModel.impl.VehicleToTrafficLightImpl@47af7f3d (name: V2-TL1, description: null) with a risk of 0 (0,0)
  - Tampering threat on ConnectedVehiclesModel.impl.VehicleToTrafficLightImpl@47af7f3d (name: V2-TL1, description: null) with a risk of 0 (0,0)
  - Tampering threat on ConnectedVehiclesModel.impl.VehicleToVehicleImpl@5e0826e7 (name: Vehicle2ToVehicle1, description: null) with a risk of 0 (0,0)
  - Tampering threat on ConnectedVehiclesModel.impl.VehicleToVehicleImpl@6dc17b83 (name: Vehicle1ToVehicle2, description: null) with a risk of 0 (0,0)
  - Spoofing threat on ConnectedVehiclesModel.impl.VehicleToTrafficLightImpl@3ce1e309 (name: V1-TL1, description: null) with a risk of 0 (0,0)
  - Spoofing threat on ConnectedVehiclesModel.impl.Traffic_LightImpl@5158b42f (name: TL1, description: null) (uuid: null) with a risk of 0 (0,0)
  - Spoofing threat on ConnectedVehiclesModel.impl.VehicleImpl@402bba4f (name: Vehicle2, description: null) (uuid: null, time_last_auth: null) with a risk of 0 (0,0)
  - Spoofing threat on ConnectedVehiclesModel.impl.VehicleImpl@402bba4f (name: Vehicle2, description: null) (uuid: null, time_last_auth: null) with a risk of 0 (0,0)
  - Tampering threat on ConnectedVehiclesModel.impl.VehicleImpl@3427b02d (name: Vehicle1, description: null) (uuid: null, time_last_auth: Fri Sep 16 07:57:15 UTC 2022) with a risk of 0 (0,0)
  - Spoofing threat on ConnectedVehiclesModel.impl.VehicleImpl@3427b02d (name: Vehicle1, description: null) (uuid: null, time_last_auth: Fri Sep 16 07:57:15 UTC 2022) with a risk of 0 (0,0)
  - Spoofing threat on ConnectedVehiclesModel.impl.Traffic_LightImpl@5158b42f (name: TL1, description: null) (uuid: null) with a risk of 0 (0,0)
  - Tampering threat on ConnectedVehiclesModel.impl.Traffic_LightImpl@5158b42f (name: TL1, description: null) (uuid: null) with a risk of 0 (0,0)
  - Tampering threat on ConnectedVehiclesModel.impl.VehicleImpl@3427b02d (name: Vehicle1, description: null) (uuid: null, time_last_auth: Fri Sep 16 07:57:15 UTC 2022) with a risk of 0 (0,0)
  - Tampering threat on ConnectedVehiclesModel.impl.VehicleImpl@3427b02d (name: Vehicle1, description: null) (uuid: null, time_last_auth: Fri Sep 16 07:57:15 UTC 2022) with a risk of 0 (0,0)
  - Tampering threat on ConnectedVehiclesModel.impl.Traffic_LightImpl@5158b42f (name: TL1, description: null) (uuid: null) with a risk of 0 (0,0)
  - Tampering threat on ConnectedVehiclesModel.impl.VehicleImpl@402bba4f (name: Vehicle2, description: null) (uuid: null, time_last_auth: null) with a risk of 0 (0,0)
  - Tampering threat on ConnectedVehiclesModel.impl.VehicleImpl@402bba4f (name: Vehicle2, description: null) (uuid: null, time_last_auth: null) with a risk of 0 (0,0)
  - Spoofing threat on ConnectedVehiclesModel.impl.VehicleImpl@3427b02d (name: Vehicle1, description: null) (uuid: null, time_last_auth: Fri Sep 16 07:57:15 UTC 2022) with a risk of 0 (0,0)
  - Tampering threat on ConnectedVehiclesModel.impl.VehicleImpl@3427b02d (name: Vehicle1, description: null) (uuid: null, time_last_auth: Fri Sep 16 07:57:15 UTC 2022) with a risk of 0 (0,0)
  - Spoofing threat on ConnectedVehiclesModel.impl.VehicleImpl@3427b02d (name: Vehicle1, description: null) (uuid: null, time_last_auth: Fri Sep 16 07:57:15 UTC 2022) with a risk of 0 (0,0)
  - Spoofing threat on ConnectedVehiclesModel.impl.VehicleImpl@3427b02d (name: Vehicle1, description: null) (uuid: null, time_last_auth: Fri Sep 16 07:57:15 UTC 2022) with a risk of 0 (0,0)
  - Tampering threat on ConnectedVehiclesModel.impl.VehicleToTrafficLightImpl@3ce1e309 (name: V1-TL1, description: null) with a risk of 0 (0,0)
  - Spoofing threat on ConnectedVehiclesModel.impl.VehicleToTrafficLightImpl@47af7f3d (name: V2-TL1, description: null) with a risk of 0 (0,0)
  - Tampering threat on ConnectedVehiclesModel.impl.VehicleToTrafficLightImpl@47af7f3d (name: V2-TL1, description: null) with a risk of 0 (0,0)
  - Spoofing threat on ConnectedVehiclesModel.impl.VehicleToTrafficLightImpl@3ce1e309 (name: V1-TL1, description: null) with a risk of 0 (0,0)
  - Tampering threat on ConnectedVehiclesModel.impl.VehicleImpl@3427b02d (name: Vehicle1, description: null) (uuid: null, time_last_auth: Fri Sep 16 07:57:15 UTC 2022) with a risk of 0 (0,0)
  - Spoofing threat on ConnectedVehiclesModel.impl.VehicleImpl@3427b02d (name: Vehicle1, description: null) (uuid: null, time_last_auth: Fri Sep 16 07:57:15 UTC 2022) with a risk of 0 (0,0)
  - Tampering threat on ConnectedVehiclesModel.impl.VehicleImpl@3427b02d (name: Vehicle1, description: null) (uuid: null, time_last_auth: Fri Sep 16 07:57:15 UTC 2022) with a risk of 0 (0,0)
  - Spoofing threat on ConnectedVehiclesModel.impl.VehicleImpl@3427b02d (name: Vehicle1, description: null) (uuid: null, time_last_auth: Fri Sep 16 07:57:15 UTC 2022) with a risk of 0 (0,0)
  - Tampering threat on ConnectedVehiclesModel.impl.VehicleToTrafficLightImpl@3ce1e309 (name: V1-TL1, description: null) with a risk of 0 (0,0)
  - Spoofing threat on ConnectedVehiclesModel.impl.VehicleToTrafficLightImpl@47af7f3d (name: V2-TL1, description: null) with a risk of 0 (0,0)
  - Tampering threat on ConnectedVehiclesModel.impl.VehicleToTrafficLightImpl@47af7f3d (name: V2-TL1, description: null) with a risk of 0 (0,0)
  - Spoofing threat on ConnectedVehiclesModel.impl.VehicleToTrafficLightImpl@3ce1e309 (name: V1-TL1, description: null) with a risk of 0 (0,0)
Vehicle: ConnectedVehiclesModel.impl.VehicleImpl@3427b02d (name: Vehicle1, description: null) (uuid: null, time_last_auth: Fri Sep 16 07:57:15 UTC 2022)
Publishing message: 0.0 to topic device/Vehicle1/riskscore
Publishing message: 0.0 to topic device/Vehicle1/Tampering/riskscore
Publishing message: 0.0 to topic device/Vehicle1/Spoofing/riskscore
Vehicle: ConnectedVehiclesModel.impl.VehicleImpl@402bba4f (name: Vehicle2, description: null) (uuid: null, time_last_auth: null)
Publishing message: 0.0 to topic device/Vehicle2/riskscore
Publishing message: 0.0 to topic device/Vehicle2/Tampering/riskscore
Publishing message: 0.0 to topic device/Vehicle2/Spoofing/riskscore
Vehicle: ConnectedVehiclesModel.impl.VehicleImpl@401d22f4 (name: Vehicle3, description: null) (uuid: 045a7fcc-e81b-40ba-83cd-a6ad5215c2b2, time_last_auth: Wed Jun 14 17:05:09 UTC 2023)
Publishing message: 0.0 to topic device/Vehicle3/riskscore
Publishing message: 0.0 to topic device/Vehicle3/Tampering/riskscore
Publishing message: 0.0 to topic device/Vehicle3/Spoofing/riskscore
```

Figure 4: TMRA console output on new device event

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No 101020416.

The updated model is persisted in the TMRA container (*files/model.sparta*) and can also be visualized using SPARTA[2] as shown below.

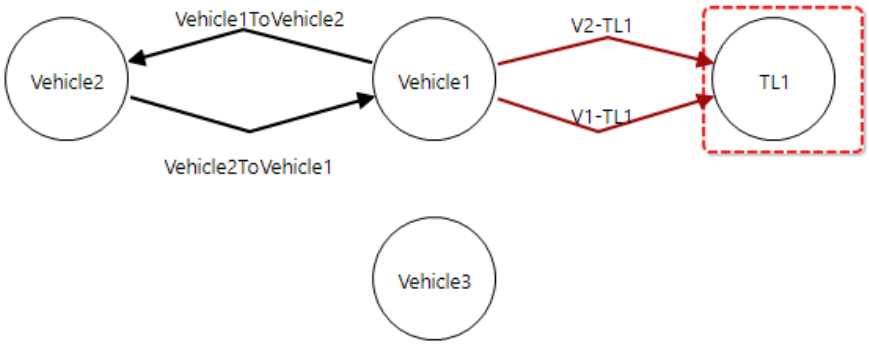


Figure 5: model after new vehicle event

**New data flow.** If, for example, Vehicle1 and Vehicle3 exchange data or interact with each other, the following event would be published:

*dataflow/Vehicle1/announcement/new*

The payload of this message would be "*Vehicle1;Vehicle2*". Again, the application adapter first updates the model by adding the new data flow, before instructing the ICTM component to recalculate the risks. The updated model is shown below.

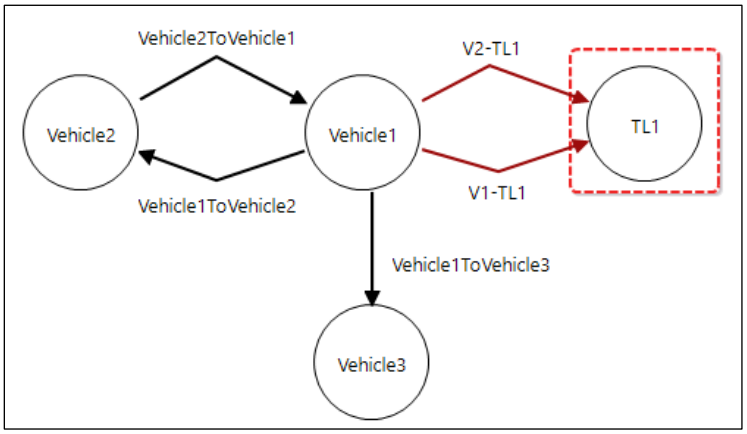


Figure 6: model after new data flow event

**Device removed.** If, after a while, Vehicle3 is removed from the system (e.g., drives away from the traffic light), the following event would be published:

*device/Vehicle3/announcement/remove*

This would return the model to the state shown in Figure 3: Prepopulated model, and similar risk scores would be published.

**Parameters changed.** The model also supports parameters for devices. For example, the time at which a device was last authenticated may have an influence on the chance of a spoofing threat (and, indirectly, the trust in that device).

[2] SPARTA documentation can be found here: https://docs.sparta.distrinet-research.be/main/index.html.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No 101020416.

In the current PoC, vehicles have an auth parameter to show how parameters can be changed. Risk calculation taking into account such parameters is work in progress. The current MQTT topics allow to change parameters for specific vehicles as follows:

*device/Vehicle3/announcement/parameters/auth*

Where the payload would contain a string with the date and time. To parse timestamps, the TMRA component leverages the *java.util.Date* and *java.text.SimpleDateFormat* libraries. Concretely, the following date format is used: *new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss'Z'")* .

## 4.2 MQTT translation layer

As already shown in Figure 1, MQTT is adopted as the main communication protocol between the diverse producers and consumers (in many cases of the integration, these are the different ERATOSTHENES technologies that contribute to risk reduction and trust) to share information about relevant events. MQTT is the de facto standard for lightweight communication in IoT sensor based applications and implements a number of relevant security controls.

The full set of topics to which interested parties can subscribe and to which events are published in MQTT is dynamic and broad, but also highly specific to the technology at hand (cf. the documentation of the diverse ERATOSTHENES components), and to the application case being implemented.

To allow independent implementation and evolution (and general decoupling) of the TMRA component from such emerging and often-changing event and topic type structures, and to ensure versatility and applicability vis a vis the different application cases (ERATOSTHENES pilots and beyond), we adopt an adapter-driven translation approach.

**Synchronization stage**. In the MQTT translation layer, the main component is the `Application Adapter` which subscribes to the events of relevance for risk assessment in the application. For each type of incoming event (call 1 and 1.1), a translation or conversion of the event (call 2) is performed into an executable edit or change of the instance-level DFD model. Finally, the `Application Adapter` executes the corresponding model edit at the level of the instance layer DFD (call 3).

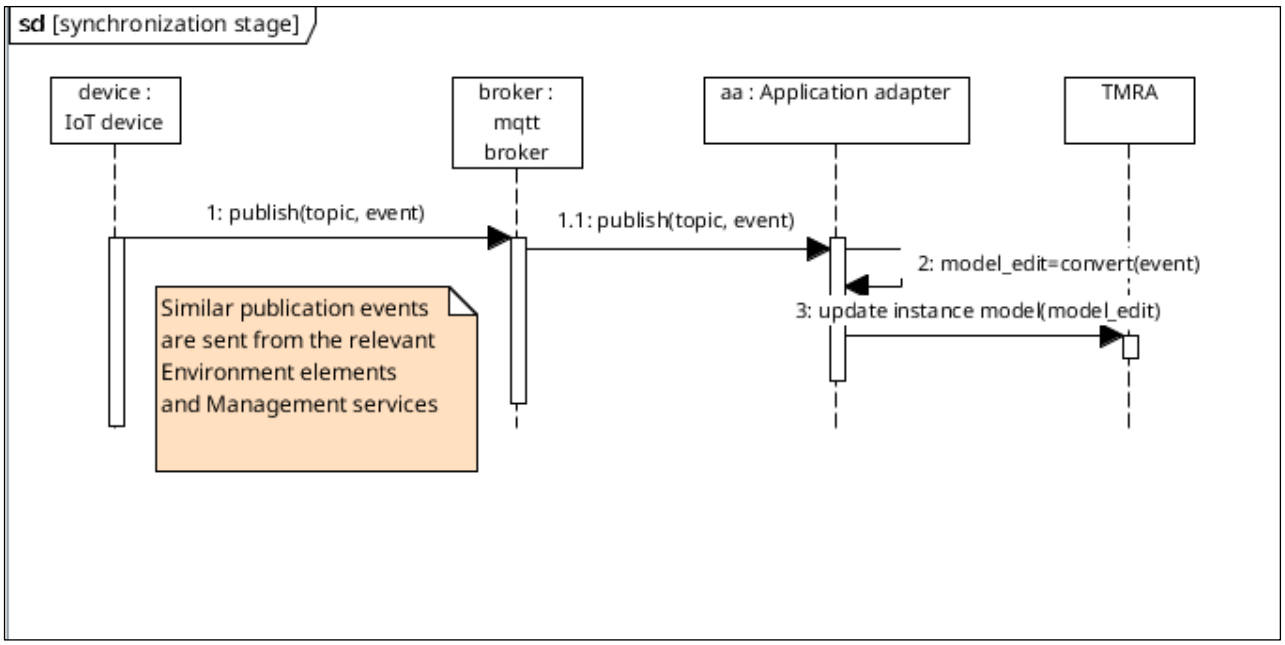


Figure 7. Sequence diagram depicting the role of the `Application Adapter` (MQTT translation layer) in converting MQTT events into executable edits of the instance DFD model and enacting these changes in the actual model.

**Initialization stage**. The process of continuous synchronization explained above works after the following initialization steps have been performed: first, an application DFD that corresponds to the logic of the actual application in which the risk assessment will be done. The creation of an application-level DFD is a prerequisite and a necessity for the TMRA component to correctly contextualize the risks and assess elements of likelihood and impact. It is static and expresses the entity classes such as cars and traffic lights. Loading the application-level DFD is the first step depicted in the sequence diagram (call 1).
Second, an initial, empty instance-level DFD is created that will be linked to the application-level DFD, which will adhere to the constraints and concepts encapsulated therein (call 2).

Finally, the Application Adapter will register for all the relevant topics of interest with the `mqtt broker` (call 3), which is the point at which this component will start relaying all the relevant events to the `Application Adapter`.

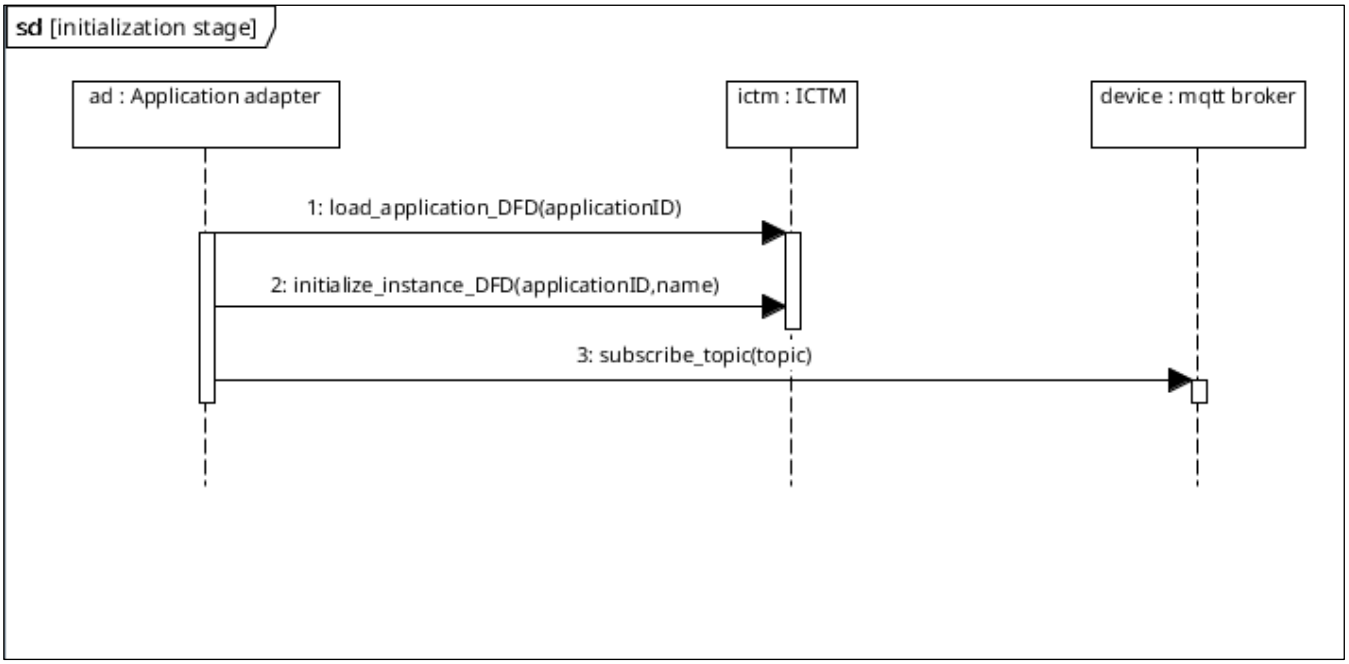


Figure 8. Sequence diagram depicting the architectural role of the `Application Adapter` (MQTT translation layer) in the initialization stage, to create empty instance DFD in line with the selected application DFD and to subscribe to relevant topics.

**Risk publication and consultation**. The final stage involves obtaining and providing access to the risk values on a per-element basis. For each element in the DFD, the `Application Adapter` is capable of regularly sending updates to specific topics via MQTT to the broker. In the current implementation, these events are fired whenever a change in risk score is detected, but as an alternative, this can also be done on a more regular basis.

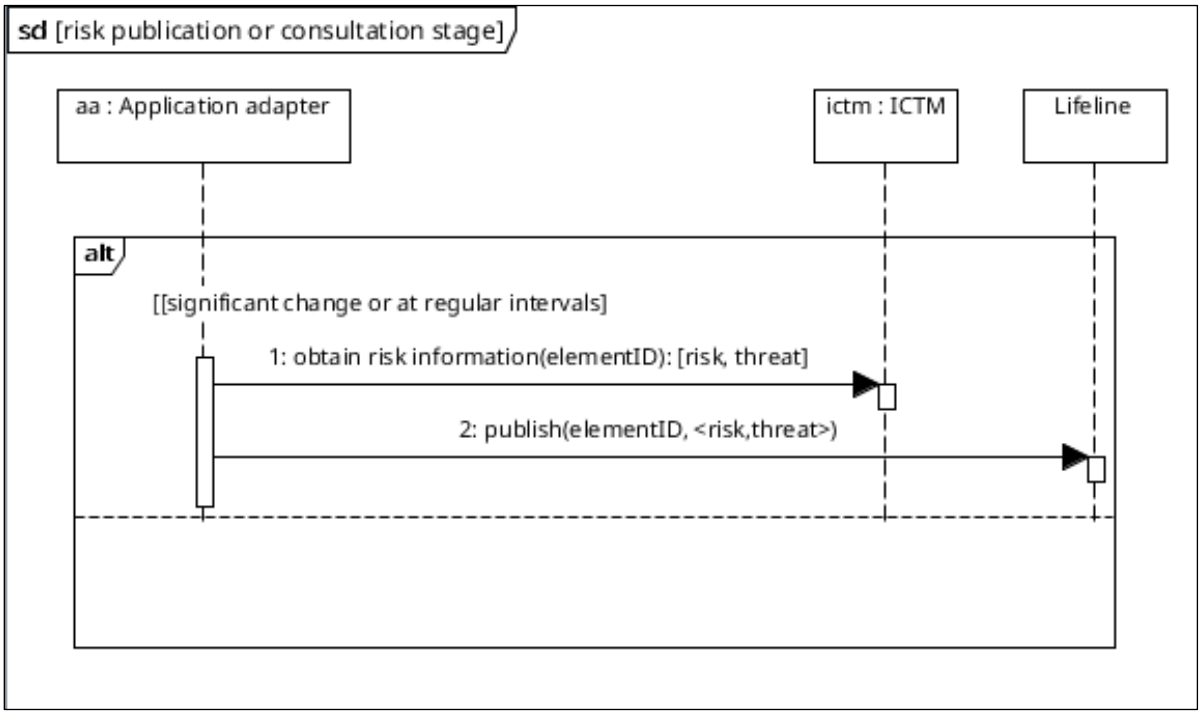


Figure 9. Sequence diagram depicting the architectural role of the Application Adapter (MQTT translation layer) in the risk publication stage, to inform interested parties to significant (configurable) changes to the risk levels of individual elements in the DFD

Risk publication is performed via the following topic structures:

- device/DID/risk
- device/DID/threatype/risk

in which the DID refers to the unique Device Identifier, which is replaced by an actual device identifier.

Interested parties such as the TMB can then subscribe to these types of events to be updated/informed after every risk assessment about the outcomes, on a per-device or per-device/per-threat type basis.

## 4.3 Outlook: simulation-based validation

The initial validations of the TMRA have been conducted in the context of the connected Vehicles (V2V and V2X) pilot case. These initial steps focus on showing the integration and applicability within a realistic application case. In ongoing validation efforts, the more extensive validation of the TMRA and its ability to handle a realistic workload involves the use of V2V simulators.
To this extent, an explorative study of a number of existing V2V simulators has been performed, for example those described in a report by the Global Cybersecurity Institute [2].
A short summary of the major simulators, as described by that report, is provided here:

- **WiLabV2XSim**: simulates the resource allocation algorithms in cellular V2X (C-V2X) and the medium contention algorithm in IEEE 802.11p (DSRC/ITS-G5).
- **GEMV² :** a highly scalable geometry-based channel propagation modeling tool for V2V. Allows modeling and visualization of V2V signal propagation.
- **VEINS**: open-source framework for running network-layer simulations of vehicular networks.
- **VeReMi**: publicly available dataset of vehicle messages recorded from VEINS simulations and specifically designed to test designs for misbehavior detection systems that might be used in V2V infrastructures.

- **PTV Vissim**: commercial traffic simulator was developed for studying traffic flow and management from the perspective of transportation system engineering.

Out of these simulators, the **VEINS** and **VeReMi** simulators fit best with the overall goal of ERATOSTHENES, as they are open source and operate at a similar abstraction level (network layer). In the next development efforts, a simulator-based evaluation effort is being set up, looking at:

- The **practical feasibility** of translating the events of such simulation scenarios to MQTT events supported by the TMRA component.
- The **performance cost** of synchronizing the models (digital twin) to the actual real-world state.
- The **performance cost** of querying and utilizing the risk estimation outcomes.

The outcomes and completion of these efforts will be reported in D2.9 (M36).

# 5 Research and Scientific Innovation

The motivation of scientific novelty was already conducted in D2.2. To avoid needless repetition, we refer the interested reader to this specific deliverable.

While the additional design aspects (decoupling of application and TMRA) provide little scientific value, but focus on pragmatic aspects of integrability and reusability, the results of the scientific performance and feasibility efforts for which the first steps are described in Section 4 are considered essential in terms of completing an extended version of the initial publication [1] which is planned in M22 of the project.

# 6 Conclusions

This deliverable describes the improvements and extensions made to the TMRA component in its second implementation cycle. It predominantly focuses on showing a more in-depth step-by-step tutorial on how to adopt and run the component (documentation which is highly relevant for the Pilot and integration cases). In addition, the architectural refinement of the Application Adapter is discussed, which allows the TMRA core components to remain agnostic of application-specific details, leads to a better separation of concerns and higher reusability and versatility of the TMRA component as a whole.
Finally, the next steps into further validation and evaluation (feasibility and performance) are outlined.

# 7 References

[1] Verreydt, S., Van Landuyt, D., Joosen, W. (2023). Expressive and Systematic Risk Assessments with Instance-Centric Threat Models. In: *SAC '23: Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*. doi: 10.1145/3555776.3577668

[2] Global Cybersecurity Institute. (2021). V2V Simulators and Related Software. url: https://www.rit.edu/wisplab/sites/rit.edu.wisplab/files/2021-12/V2V%20Simulators%20and%20Related%20Software.pdf