COLLEGES SCIENCES
BRETAGNE POUR L'INGENIEUR
LOIRE ET LE NUMERIQUE

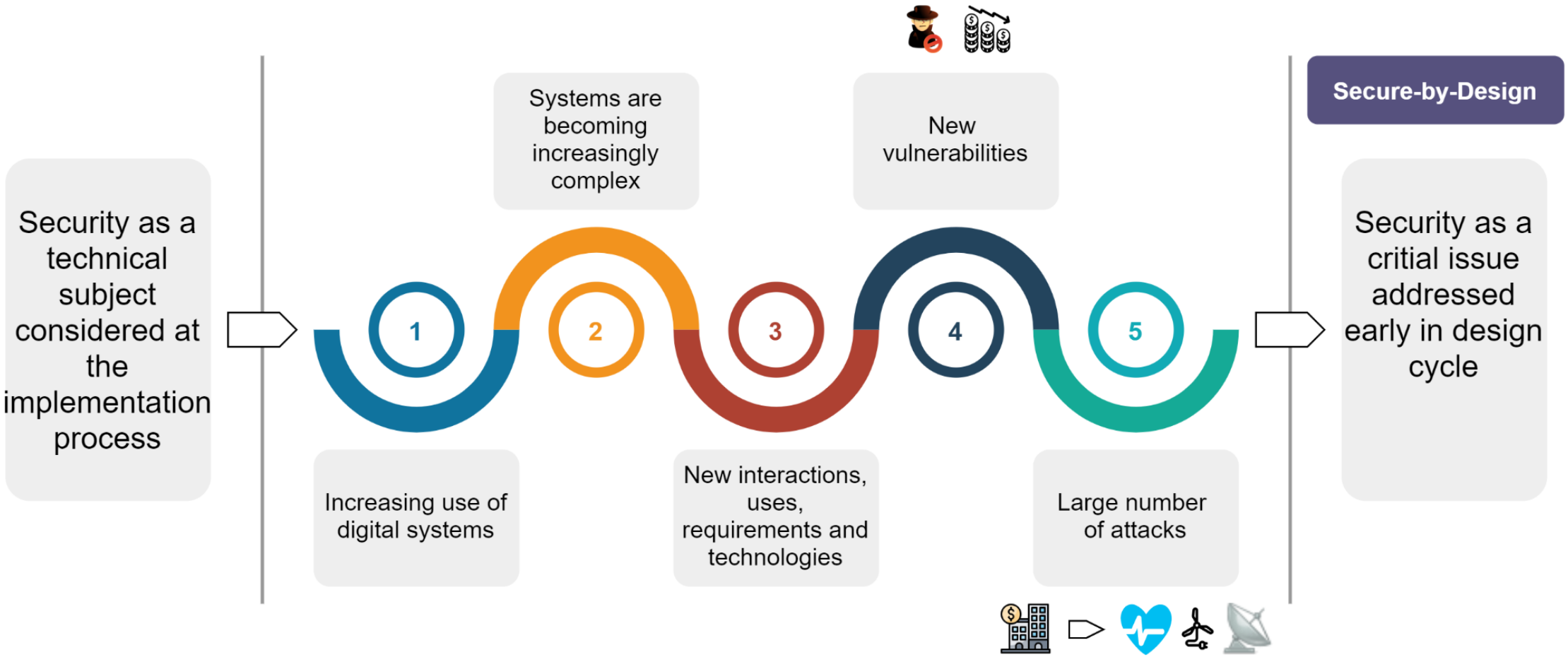
Towards a unifying framework for the specification, formalization and analysis of secure hardware and software architectures

Hiba Hnaini

Objective

Design and **evaluate** a new multi-paradigm security modeling approach and its engineering framework (engineering process and tooling) allowing the **specification**, **formalization** and **analysis** of **secure hardware and software architectures.**

Context



Innovative nature of the project

We want to

Design secure systems

Using a unifying framework (Specification, Formalization, Analysis)

That is applicable in real cases

The path to designing secure systems is long
Need federated approach
(Different levels of abstraction and viewpoints)

No unifying framework for the multiple languages

Technology transfer has a significantly lower efficiency outside limited test facilities

Going beyond a simple mix of solutions

Using different modeling and programming formalisms (Multiparadigm)

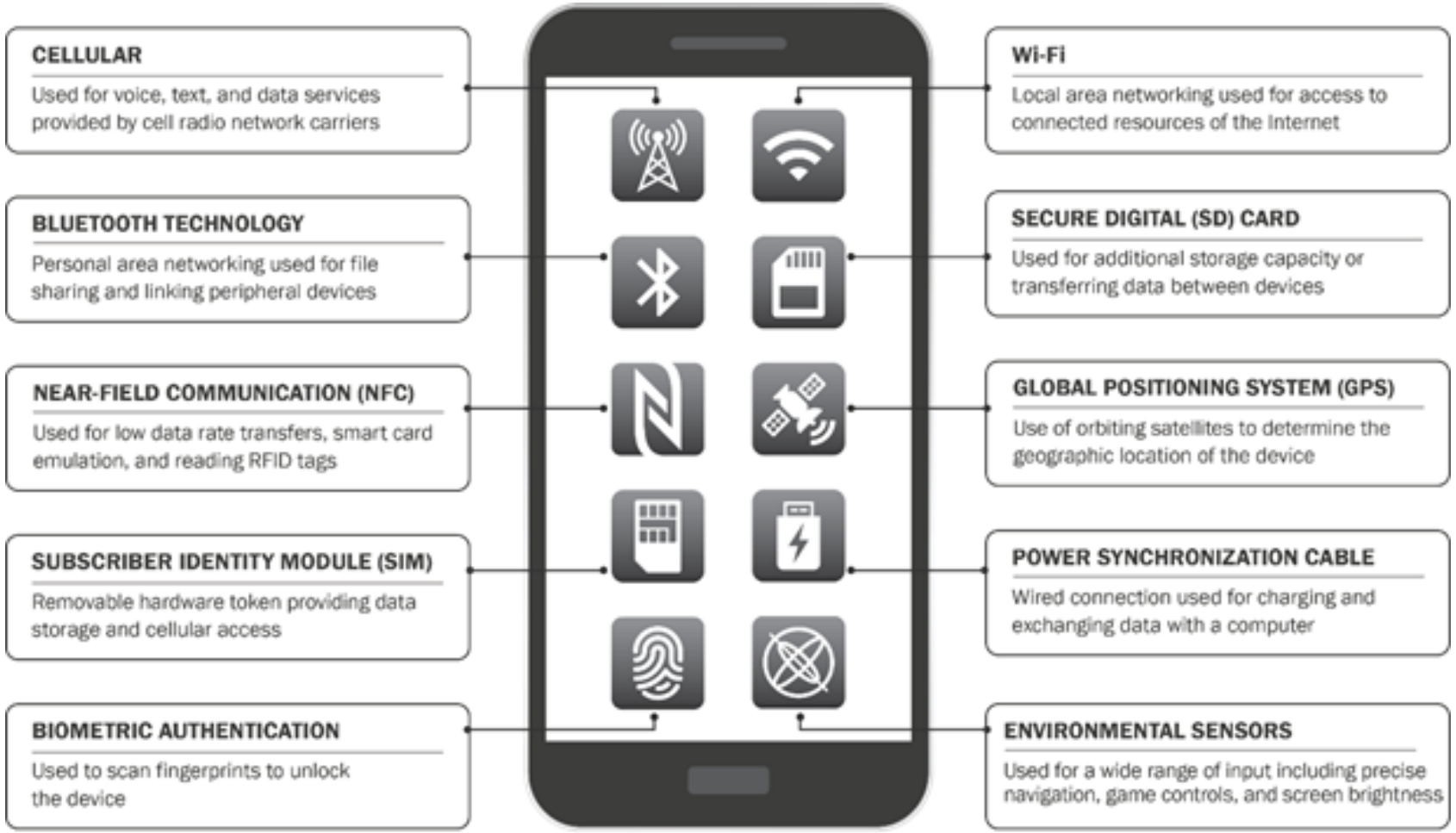
Create a unifying framework (Specification, Formalization, Analysis) that supports:

Reusability of the approach (Separate between specification and analysis)

+

Develop reference experiments to affirm the applicability and usefulness in real cases

Example – A Smart Phone or a Family of Smart Phones



Example – A Smart Phone or a Family of Smart Phones

Security Requirements according to:

Requirements for OEM regarding Smartphone Security (bund.de)

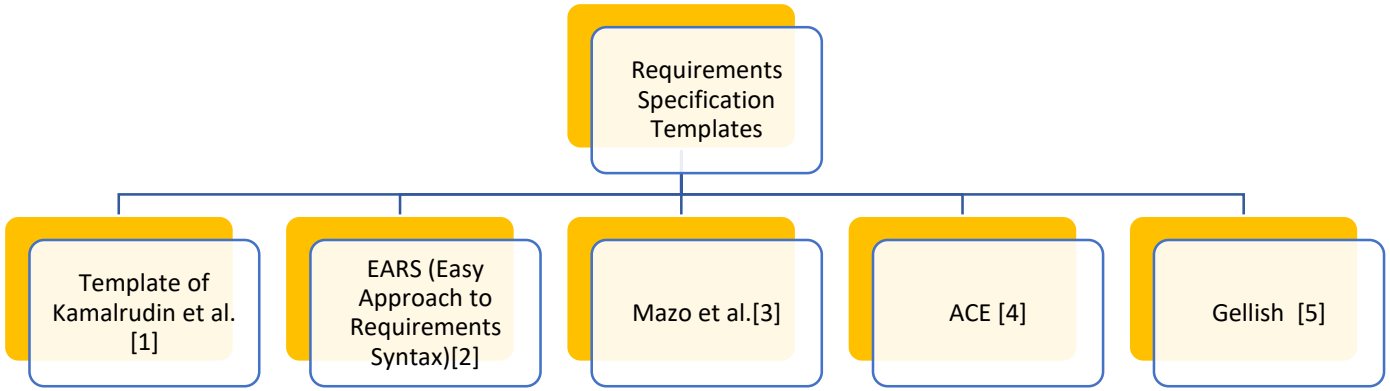
(defined by the German Federal Office for Information Security)

Security Criteria	Number of Requirements
Maintainability	2
Access Control	6
Integrity	2
Privacy	5
Authorization	1
Resilience to Attacks	3
Immunity	1
Availability	1
Confidentiality	4
Location Privacy	1

Example – A Smart Phone or a Family of Smart Phones

The main security criteria in this domain are:	Confidentiality Integrity Privacy Availability
Requirements from the Document:	Req1: From the network perspective the use of the newest Radio Canal Ciphering Algorithms has very high priority Devices supporting these algorithms are better protected. Req2: The HSE must be used to store critical user data. Req3: In case of bootloader unlock, all user data must be deleted in a safe way. Req4: The processor must support hardware based security functions for attack defense. This includes mechanisms for ROP/JOP mitigation, attacks to speculative execution and memory encryption, and separation mechanisms (TEE). Req 5: All new devices must be provided with the latest OS available at release time.

Security Requirements Specification



[1] Kamalrudin, Massila & Mustafa, Nuridawati & Sidek, Safiah. (2018). A Template for Writing Security Requirements. 10.1007/978-981-10-7796-8_6.

[2] A. Mavin, P. Wilkinson, A. Harwood and M. Novak, "Easy Approach to Requirements Syntax (EARS)," 2009 17th IEEE International Requirements Engineering Conference, 2009, pp. 317-322, doi: 10.1109/RE.2009.9.

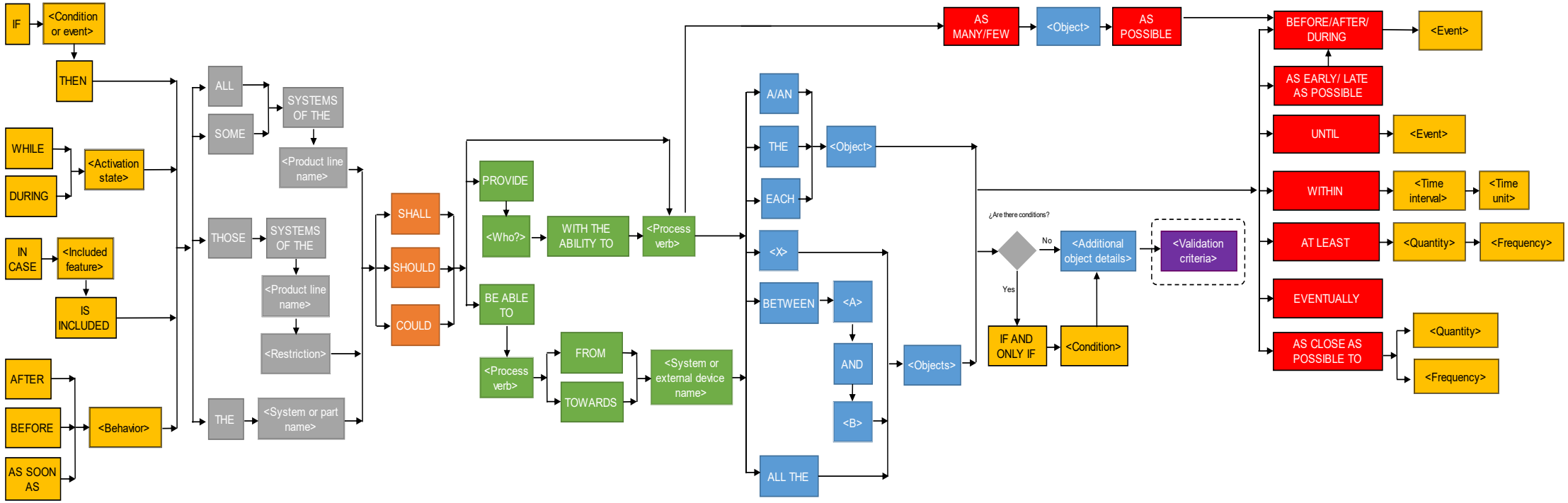
[3]] Mazo, Raúl & Jaramillo, Carlos & Vallejo, Paola & Medina, Jhon. (2020). Towards a new template for the specification of requirements in semi-structured natural language. Journal of Software Engineering Research and Development. 8. 3. 10.5753/jserd.2020.473.

[4] Fuchs, Norbert E., et Rolf. Schwitter. « Attempto Controlled English (ACE).» CLAW 96: proceedings of the First International Workshop on Controlled Language Applications. 1996.

[5] van Renssen, Andries. (2011). Modeling of Textual Requirements in a Gellish Universal Database.. 102-115.

Security Requirements Specification

Mazo et al. template [1]



[1] Mazo, Raúl & Jaramillo, Carlos & Vallejo, Paola & Medina, Jhon. (2020). Towards a new template for the specification of requirements in semi-structured natural language. Journal of Software Engineering Research and Development. 8. 3. 10.5753/jserd.2020.473.

Security Requirements Specification

Mazo et al. template [1]

Req1: The <Cellular Interface>$_{\text{system or part name}}$ <should>$_{\text{priority}}$ <ensure>$_{\text{process verb}}$ <data>$_{\text{object}}$ <**confidentiality by Radio Canal Ciphering Algorithms** >$_{\text{additional object information}}$

Req2: The <HSE>$_{\text{system or part name}}$ <should>$_{\text{priority}}$ <ensure> $_{\text{process verb}}$ <data>$_{\text{object}}$ <**integrity by storing security critical data >**$_{\text{additional object information}}$

Req3: < **W**hen the bootloader is unlocked **>** $_{\textbf{condition}}$ The <App Processor>$_{\text{system or part name}}$ <should>$_{\text{priority}}$ <ensure> $_{\text{process verb}}$ <data>$_{\text{object}}$ <**privacy by deleting user data>**$_{\text{additional object information}}$

…

Req5: <All new devices>$_{\text{systems}}$ <should>$_{\text{priority}}$ <ensure>$_{\text{process verb}}$ <availability of data>$_{\text{object}}$ <by having the latest OS at release.> $_{\text{additional info}}$

[1] Mazo, Raúl & Jaramillo, Carlos & Vallejo, Paola & Medina, Jhon. (2020). Towards a new template for the specification of requirements in semi-structured natural language. Journal of Software Engineering Research and Development. 8. 3. 10.5753/jserd.2020.473.

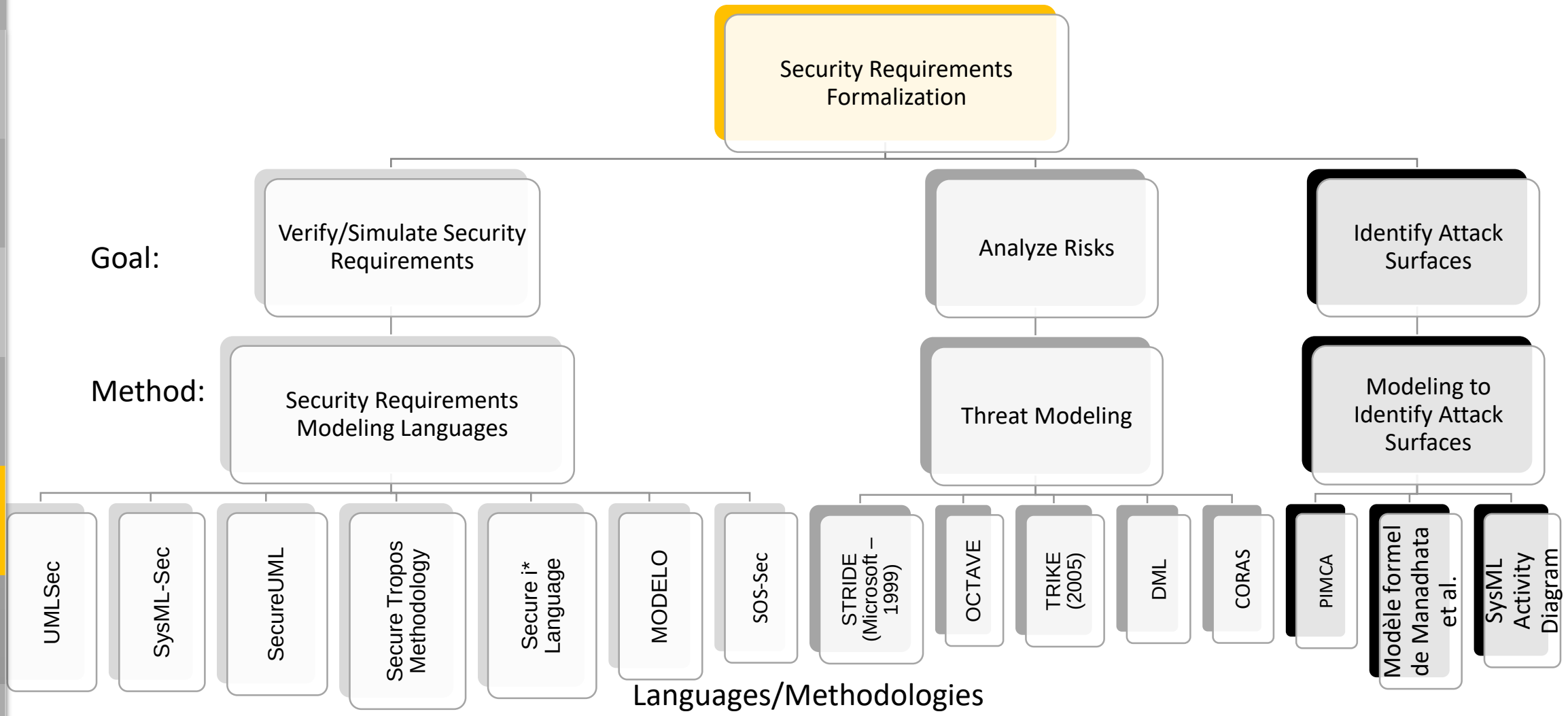
Security Requirements Specification

Mazo et al. template[1]

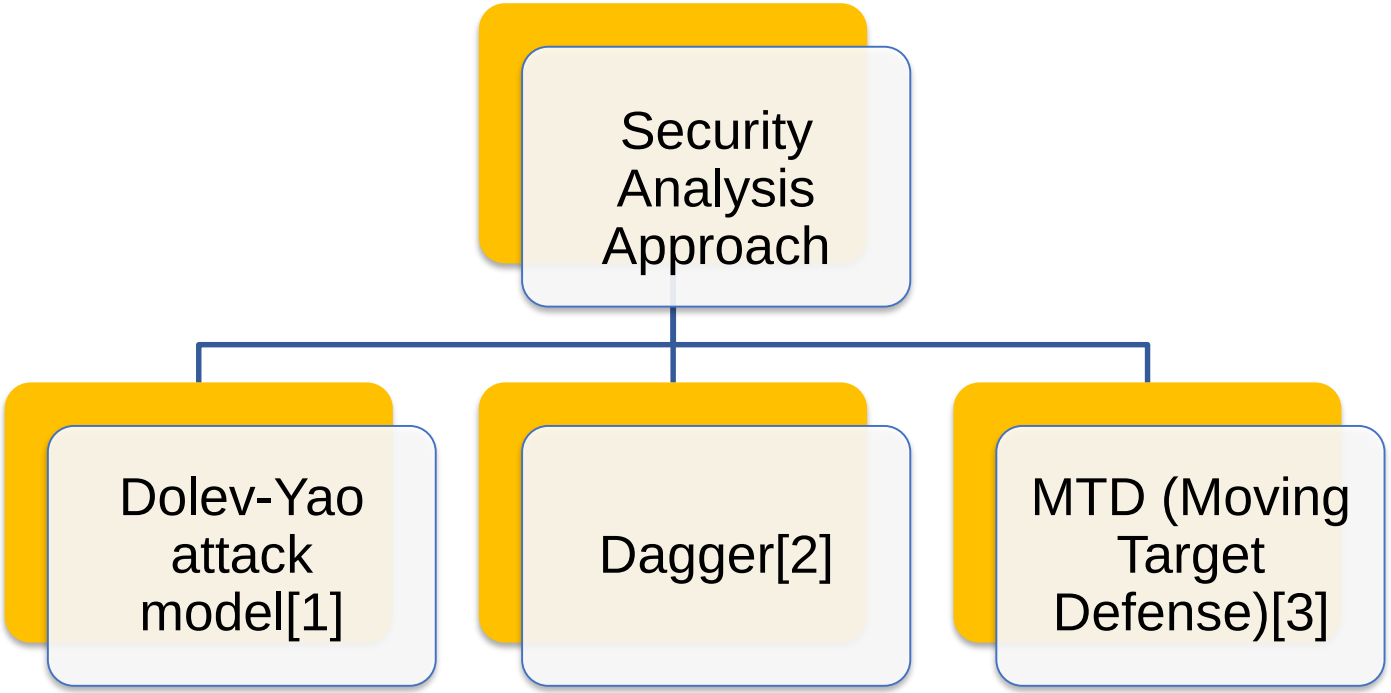
Advantages	Disadvantages
Requirements Specification	Security Criteria considered as additional object info.
Structured Natural Language	Security Mechanism considered as additional object info.
Applies to a family of systems and auto-adaptive systems	

[1] Mazo, Raúl & Jaramillo, Carlos & Vallejo, Paola & Medina, Jhon. (2020). Towards a new template for the specification of requirements in semi-structured natural language. Journal of Software Engineering Research and Development. 8. 3. 10.5753/jserd.2020.473.

Security Requirements Formalization



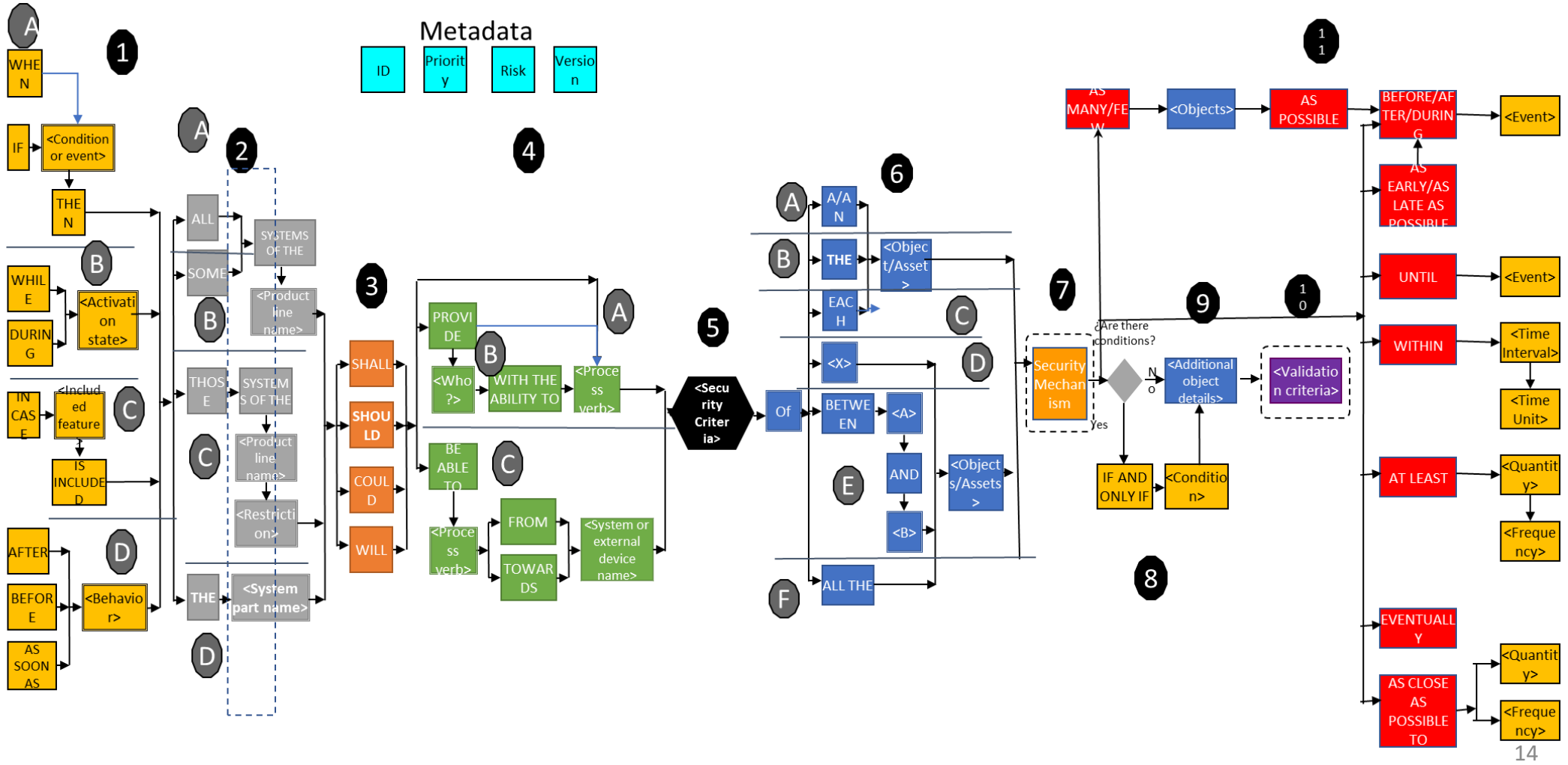
Security Analysis



[1]Cervesato, Iliano. (2001). The Dolev-Yao Intruder is the Most Powerful Attacker
[2]Peterson, Elisha. (2016). Dagger: Modeling and visualization for mission impact situation awareness. 25-30. 10.1109/MILCOM.2016.7795296.
[3]Lei, Cheng & Zhang, Hong-Qi & Jinglei, Tan & Zhang, Yu-Chen & Liu, Xiao-Hu. (2018). Moving Target Defense Techniques: A Survey. Security and Communication Networks. 2018. 1-25. 10.1155/2018/3759626.

Security Requirements Specification

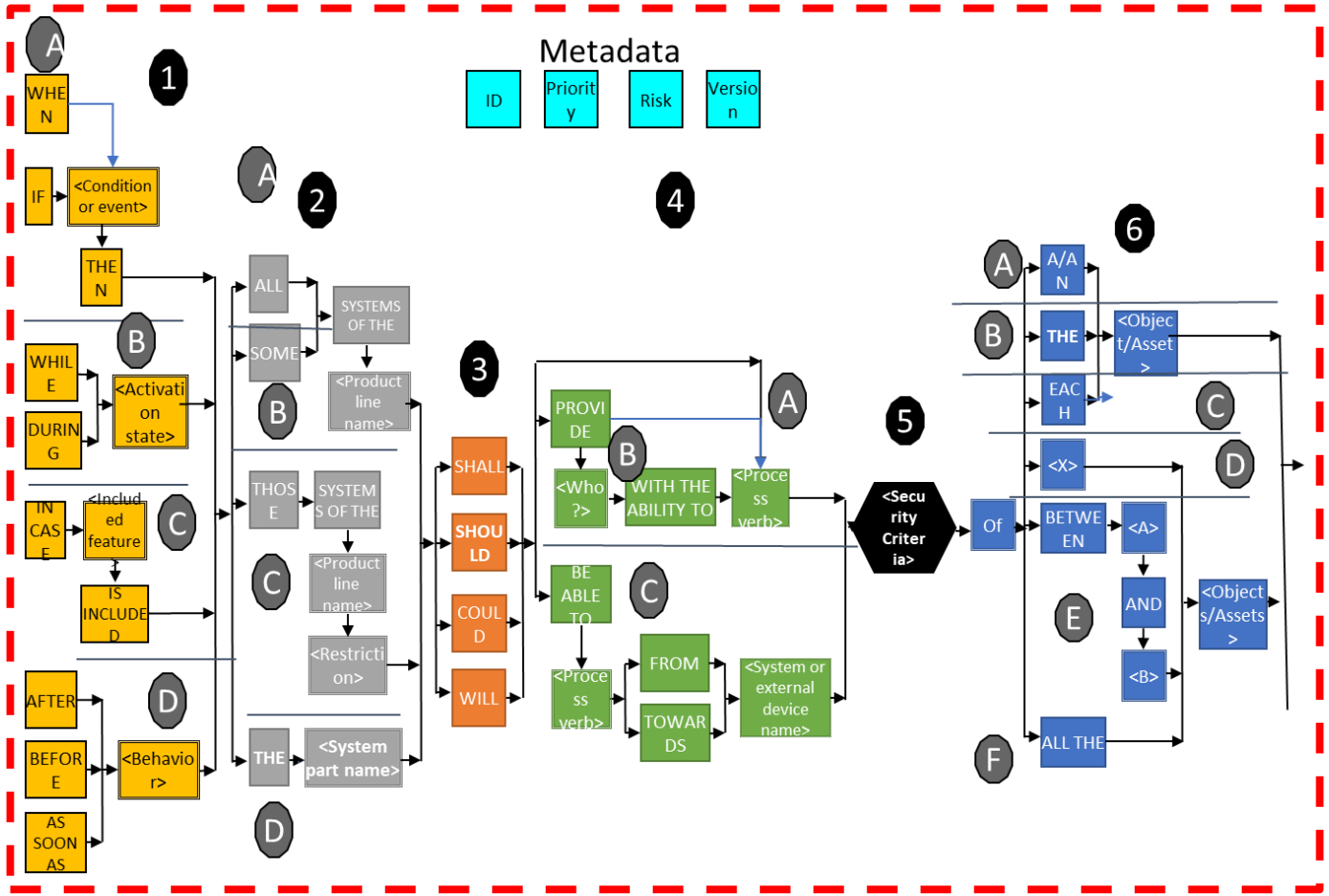
SECRET : Template based on the new template for the specification of requirements in semi-structured natural language



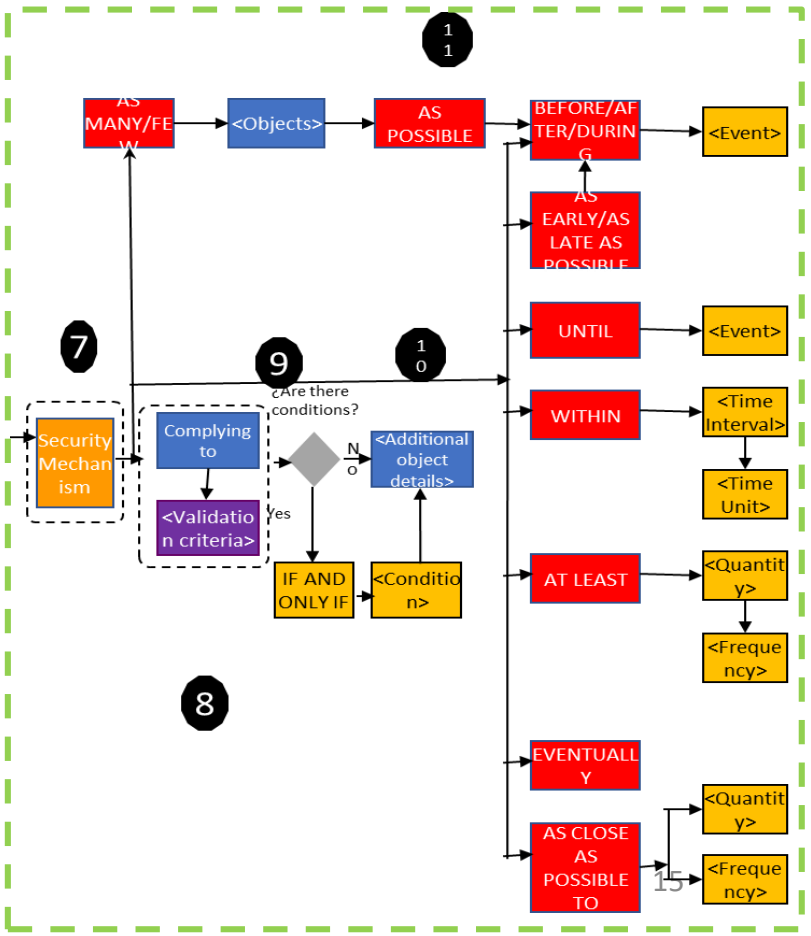
Security Requirements Specification

SECRET : Template based on the new template for the specification of requirements in semi-structured natural language

Problem Space

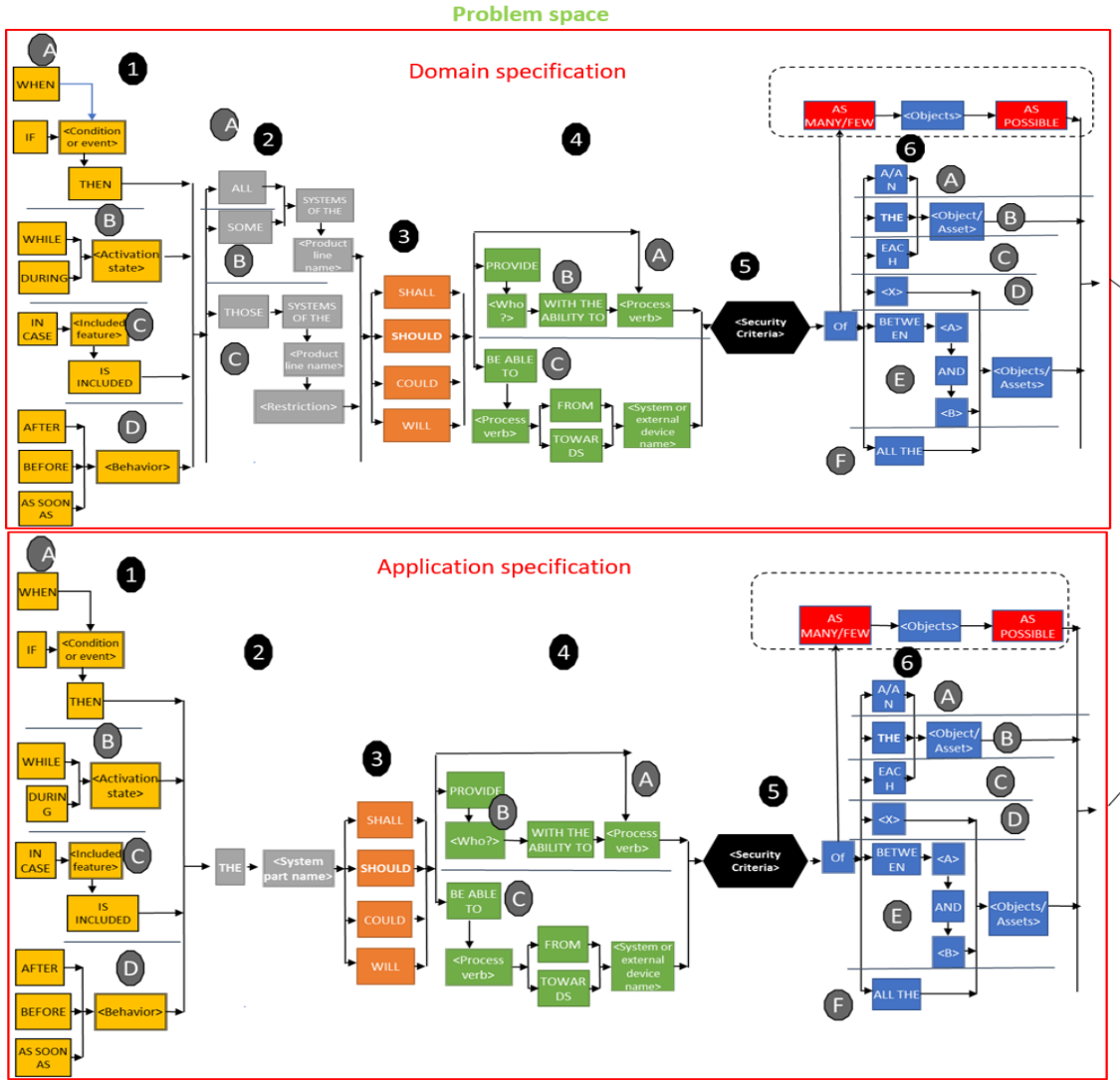


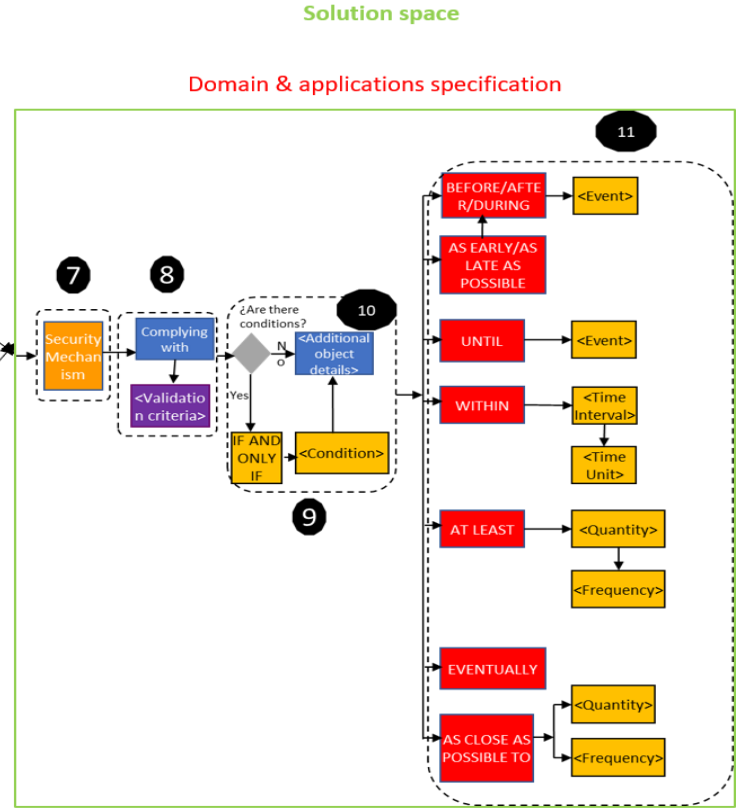
Solution Space



Security Requirements Specification

SECRET : Template based on the new template for the specification of requirements in semi-structured natural language





Security Requirements Specification

SECRET : Template based on the new template for the specification of requirements in semi-structured natural language –

Creating a microservice for text suggestion based on SECRET

Security Requirements Specification

SECRET : Template based on the new template for the specification of requirements in semi-structured natural language –

Creating a microservice for text suggestion based on SECRET

Application Security Requirements

SecurityRequirement
Draft
Req1 - 1
High
The system

Status: Draft

Description: The system

shall
should
could
will

Domain Security Requirements

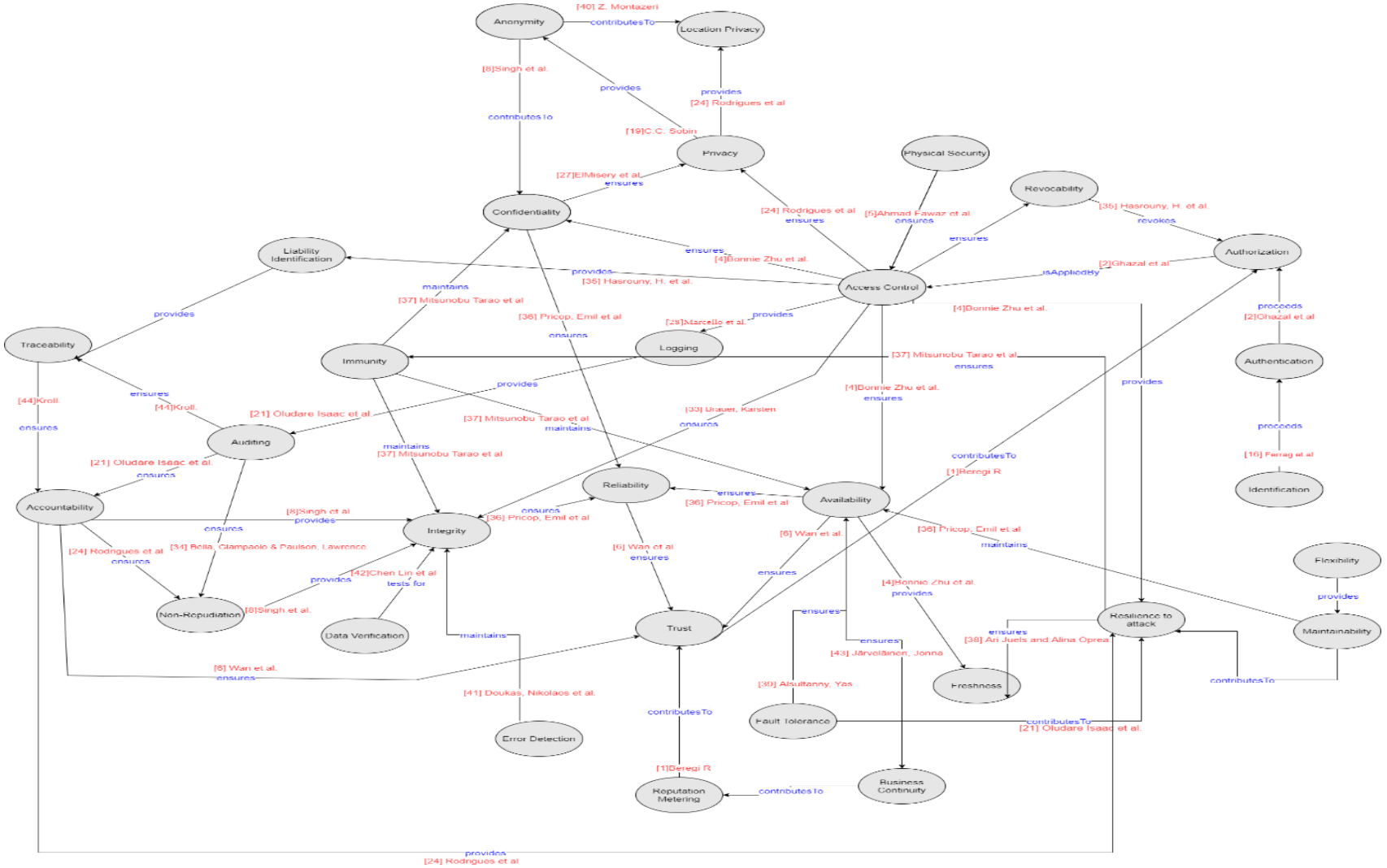
SecurityRequirement
Draft
Req1 - 1
High
undefined

Description

If
When
While
During
In Case
After
Before
As soon as
All
Some
Those

Security Requirements Specification

Ontology of Security Criteria V2



Ontology of Security Criteria – Proof of Concept

Writing Requirements Without the Typology (4 requirements only):

Req1: The <Cellular Interface>$_{\text{system or system part}}$ <should>$_{\text{priority}}$ <ensure>$_{\text{process verb}}$ <confidentiality>$_{\text{security criteria}}$ of <data>$_{\text{asset to protect}}$ <by Radio Canal Ciphering Algorithms>$_{\text{security mechanism}}$

Req2: The <HSE>$_{\text{system or system part}}$ <should>$_{\text{priority}}$ <ensure>$_{\text{process verb}}$ <integrity>$_{\text{security criteria}}$ of <data>$_{\text{asset to protect}}$ <by storing security critical data>$_{\text{security mechanism}}$

...

Security Requirements Specification

Ontology of Security Criteria – Proof of Concept

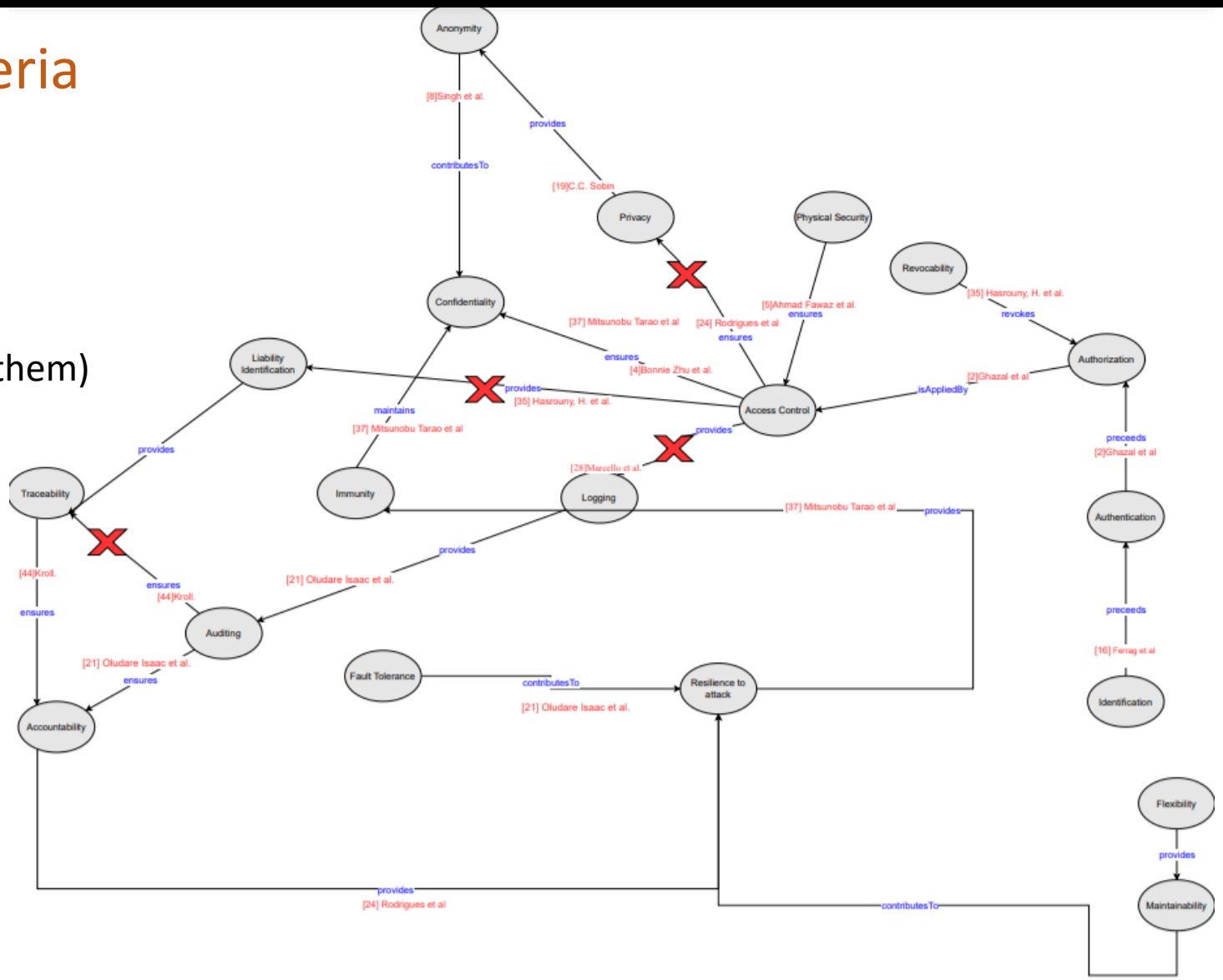
Keep Confidentiality

+

The criteria that contribute to it (all of them)

+

Break loops with repetitive criteria



Ontology of Security Criteria – Proof of Concept

Writing Requirements With the Ontology:

Req1: The <Cellular Interface>$_{\text{system or system part}}$ <should>$_{\text{priority}}$ <ensure>$_{\text{process verb}}$ <confidentiality>$_{\text{security criteria}}$ of <data>$_{\text{asset to protect}}$ <by Radio Canal Ciphering Algorithms>$_{\text{security mechanism}}$

- Req1.1: The <Cellular Interface>$_{\text{system or system part}}$ <should>$_{\text{priority}}$ <ensure>$_{\text{process verb}}$ <access control>$_{\text{security criteria}}$ of <data>$_{\text{asset to protect}}$ <.....>$_{\text{security mechanism}}$
- Req1.2: The <Cellular Interface>$_{\text{system or system part}}$ <should>$_{\text{priority}}$ <ensure>$_{\text{process verb}}$ <authorization>$_{\text{security criteria}}$ of <users>$_{\text{asset to protect}}$ <.....>$_{\text{security mechanism}}$

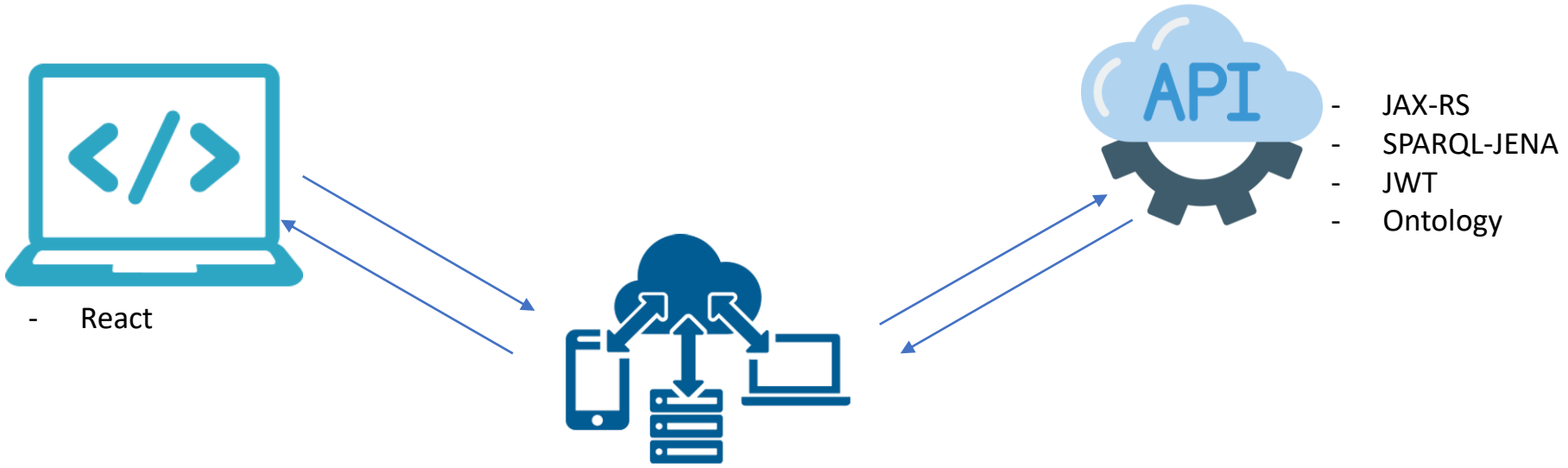
Req2: The <HSE>$_{\text{system or system part}}$ <should>$_{\text{priority}}$ <ensure>$_{\text{process verb}}$ <integrity>$_{\text{security criteria}}$ of <data>$_{\text{asset to protect}}$ <by storing security critical data>$_{\text{security mechanism}}$

...

Ontology of Security Criteria

API of Security Criteria Ontology – Web Interface:

A web interface is created to simplify the step of ontology evaluation for experts:



- React

- JAX-RS
- SPARQL-JENA
- JWT
- Ontology

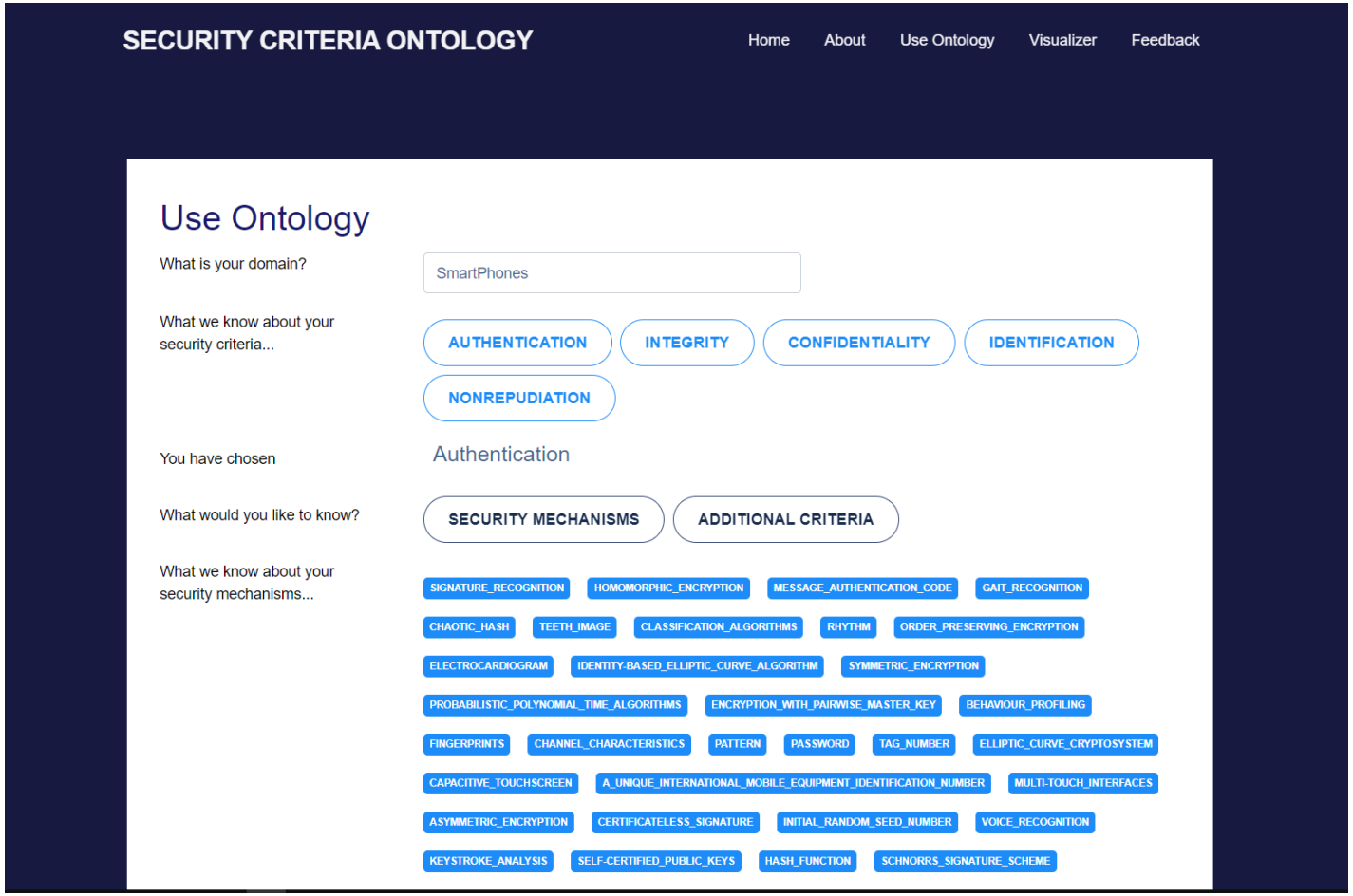
- NodeJS Proxy
- Add CORS to Response

Ontology of Security Criteria

API of Security Criteria Ontology – Web Interface:

A web interface is created to simplify the step of ontology evaluation for experts:

- Get the security mechanisms of security criteria in a domain

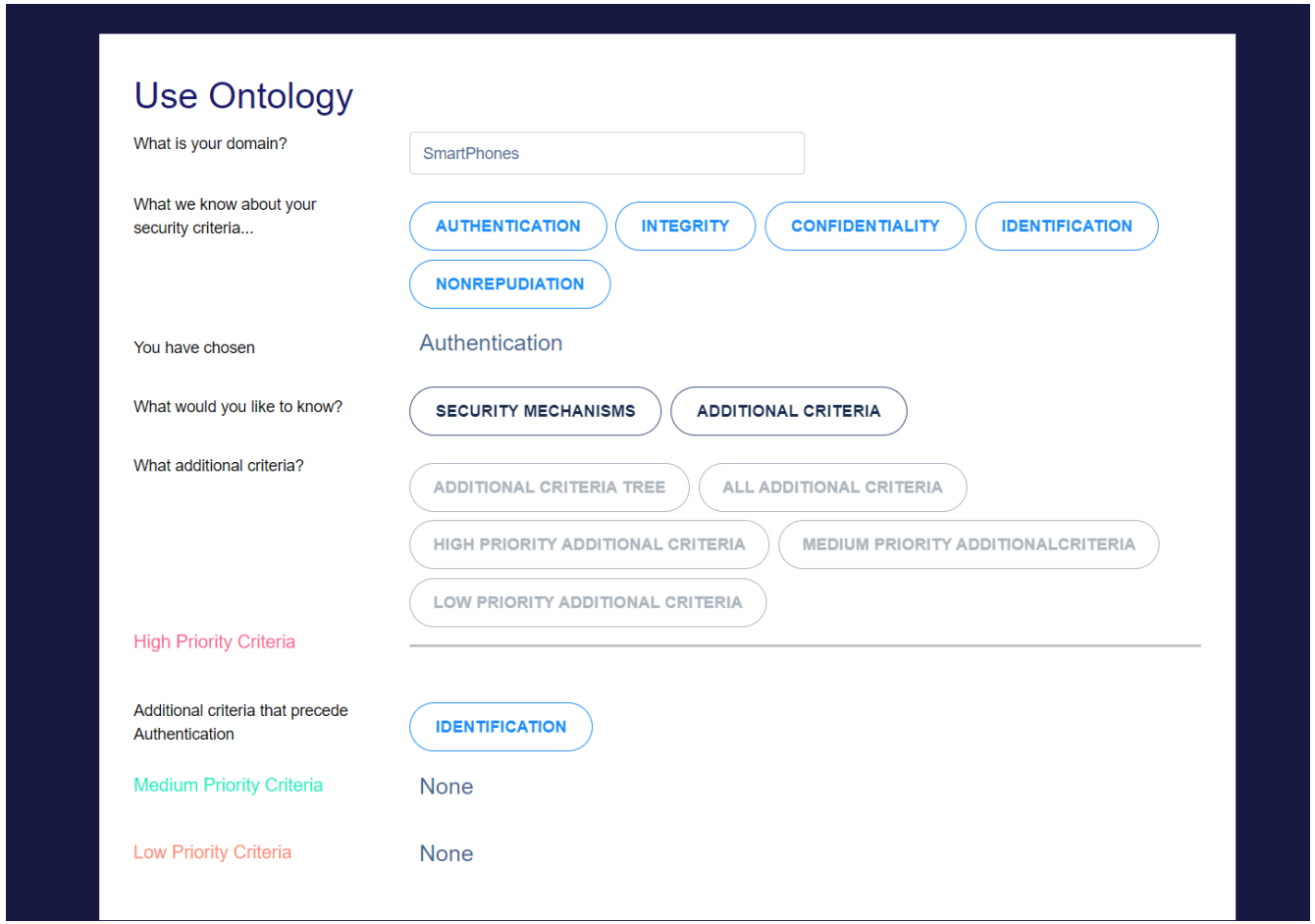


Ontology of Security Criteria

API of Security Criteria Ontology – Web Interface:

A web interface is created to simplify the step of ontology evaluation for experts:

- Get the suggested criteria of security criteria in a domain



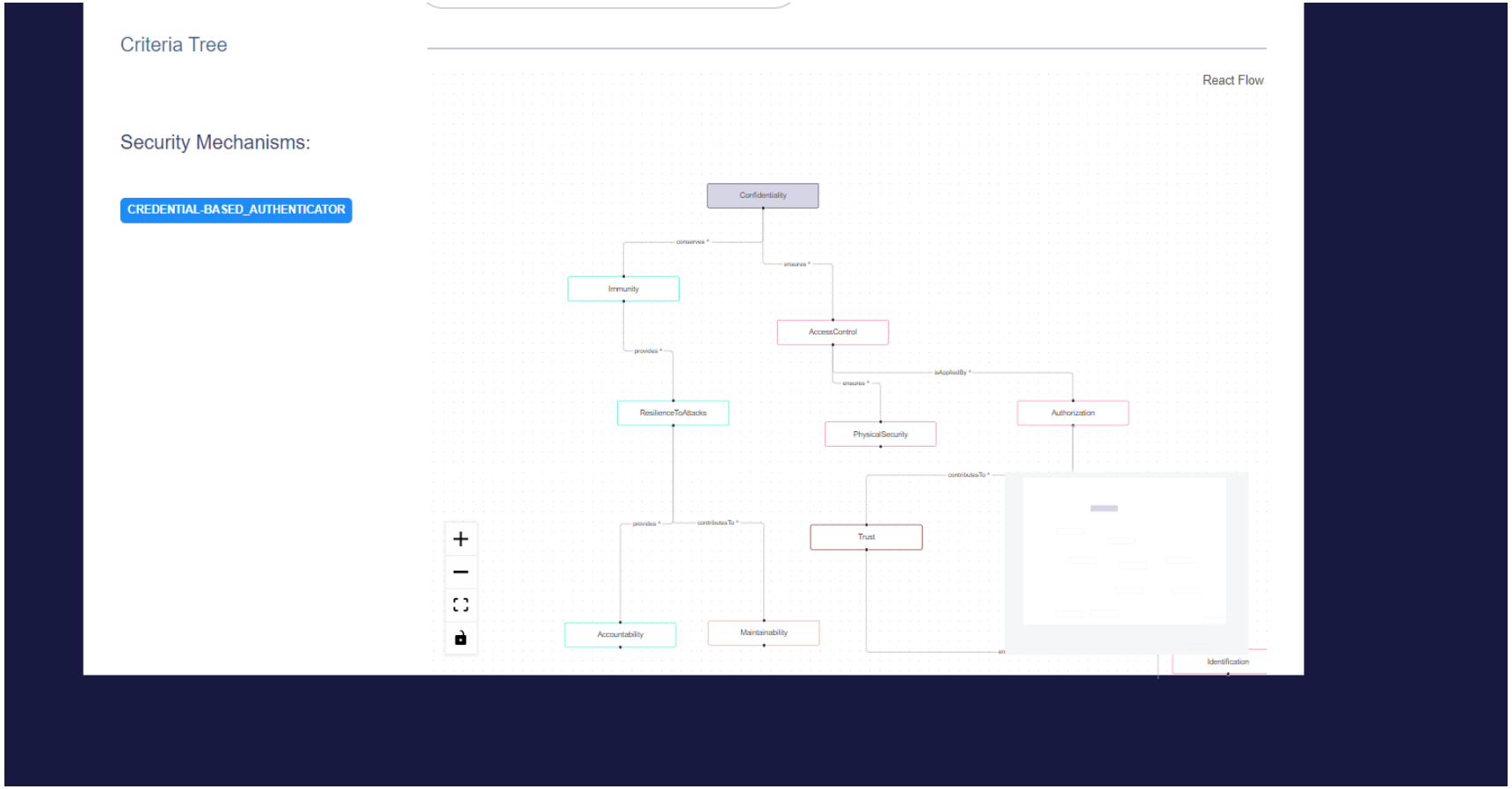
Security Requirements Specification

Ontology of Security Criteria

API of Security Criteria Ontology – Web Interface:

A web interface is created to simplify the step of ontology evaluation for experts:

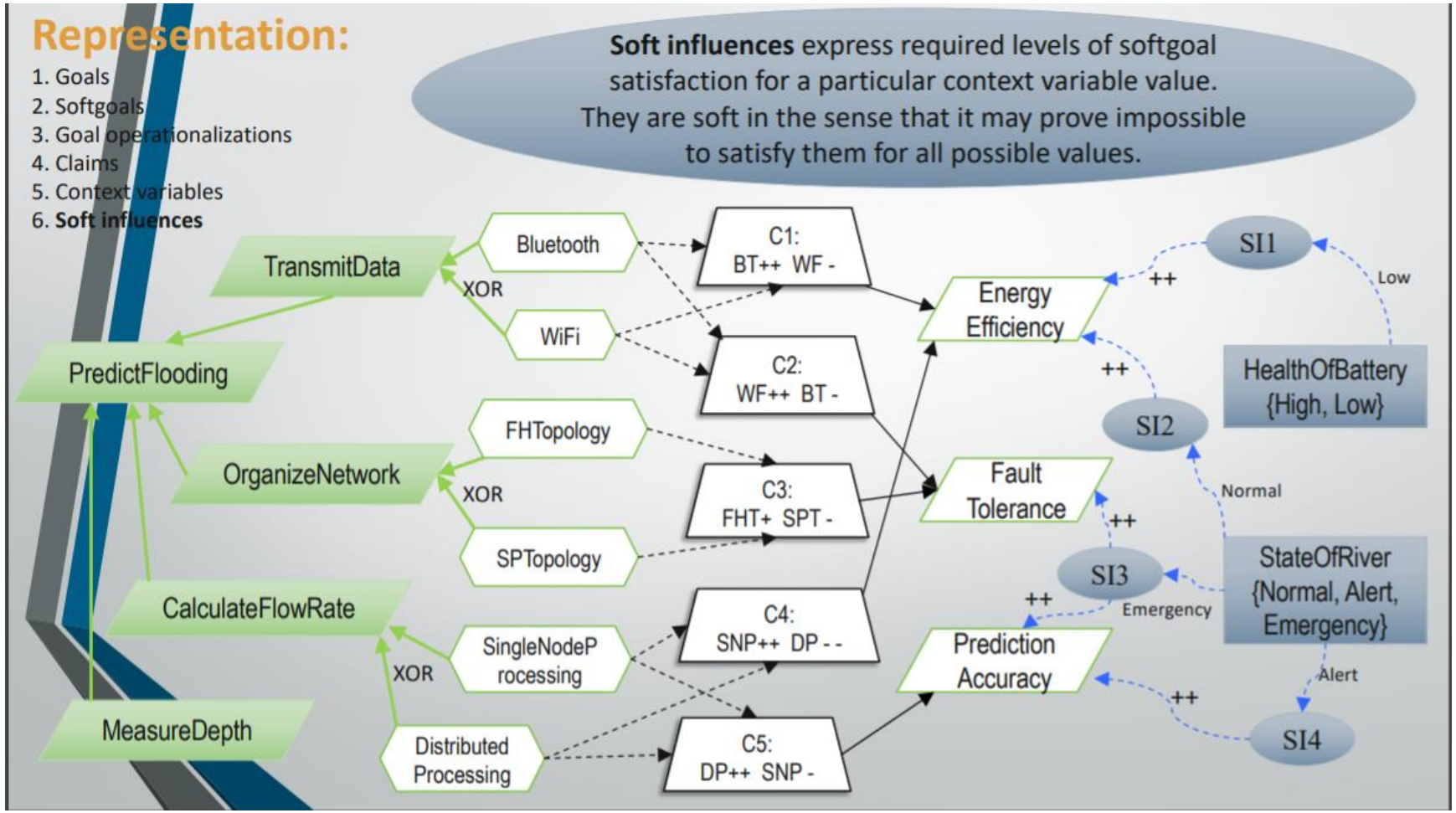
- Get the tree of additional security criteria
- Example: If you need confidentiality, then it is advised that you define sub-requirement with the immunity, access control … security criteria.



Security Requirements Formalization

SERENA

Soyer et al. Example



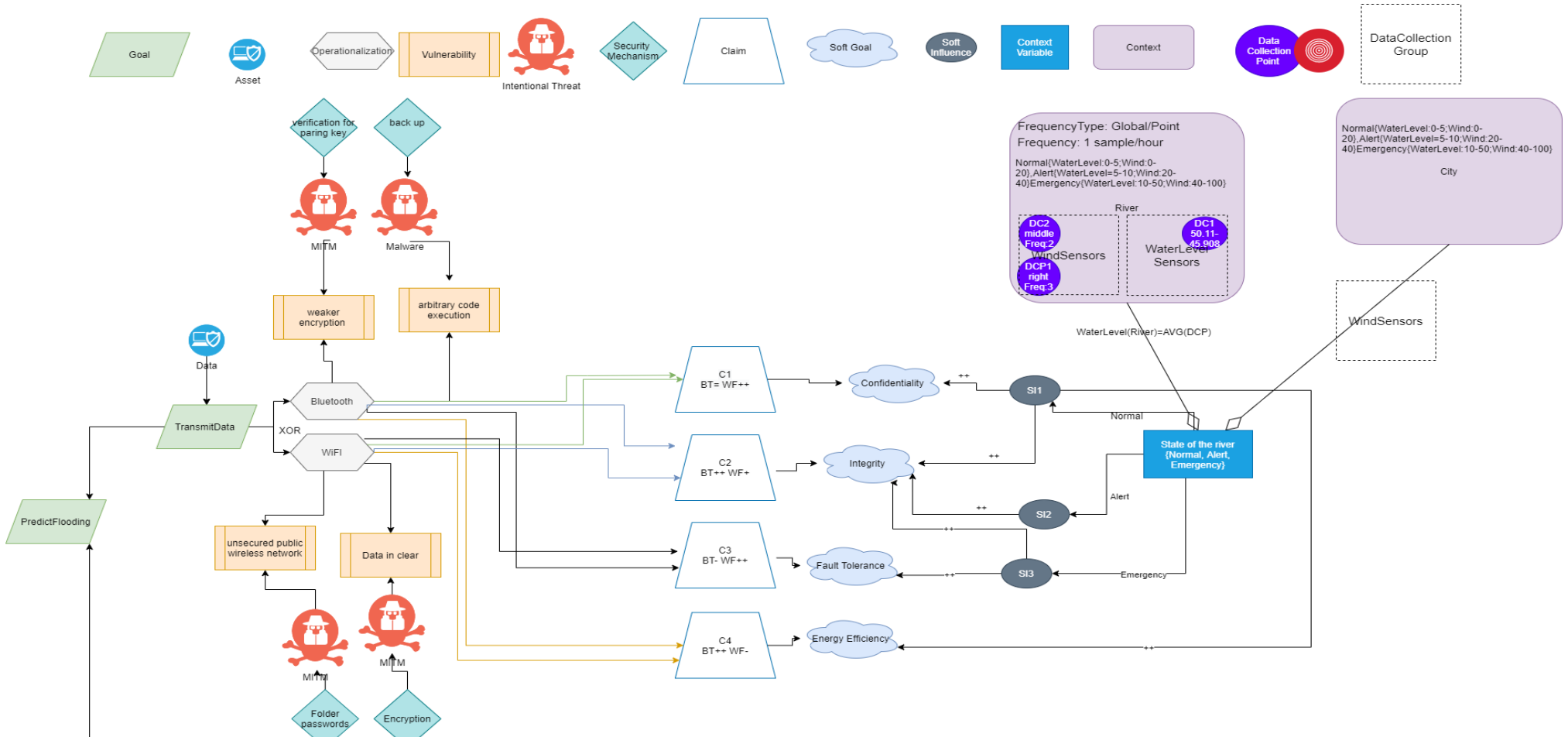
Security Requirements Formalization

SERENA

Example

Context has:

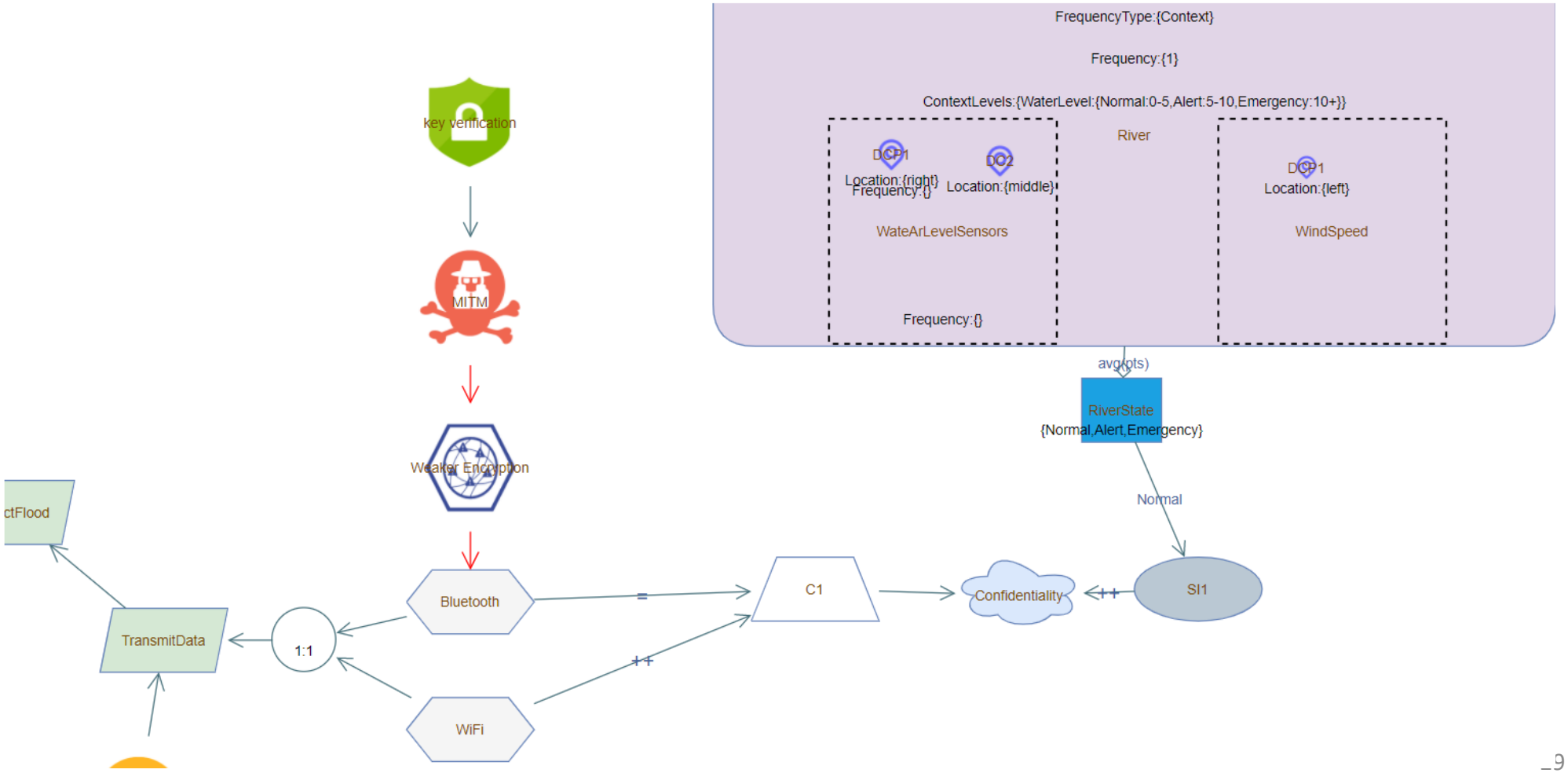
- Frequency if global for all data collection points
- ContextLevels
- DataCollectionGroups (According to the sensor or data needed) that have:
  - Frequency if not global in context
  - Data Collection Points that have:
    - Location
    - Frequency if not global in context or DataCollectionGroup



Security Requirements Formalization

SERENA

Example in VariaMos



Security Requirements Formalization

SERENA

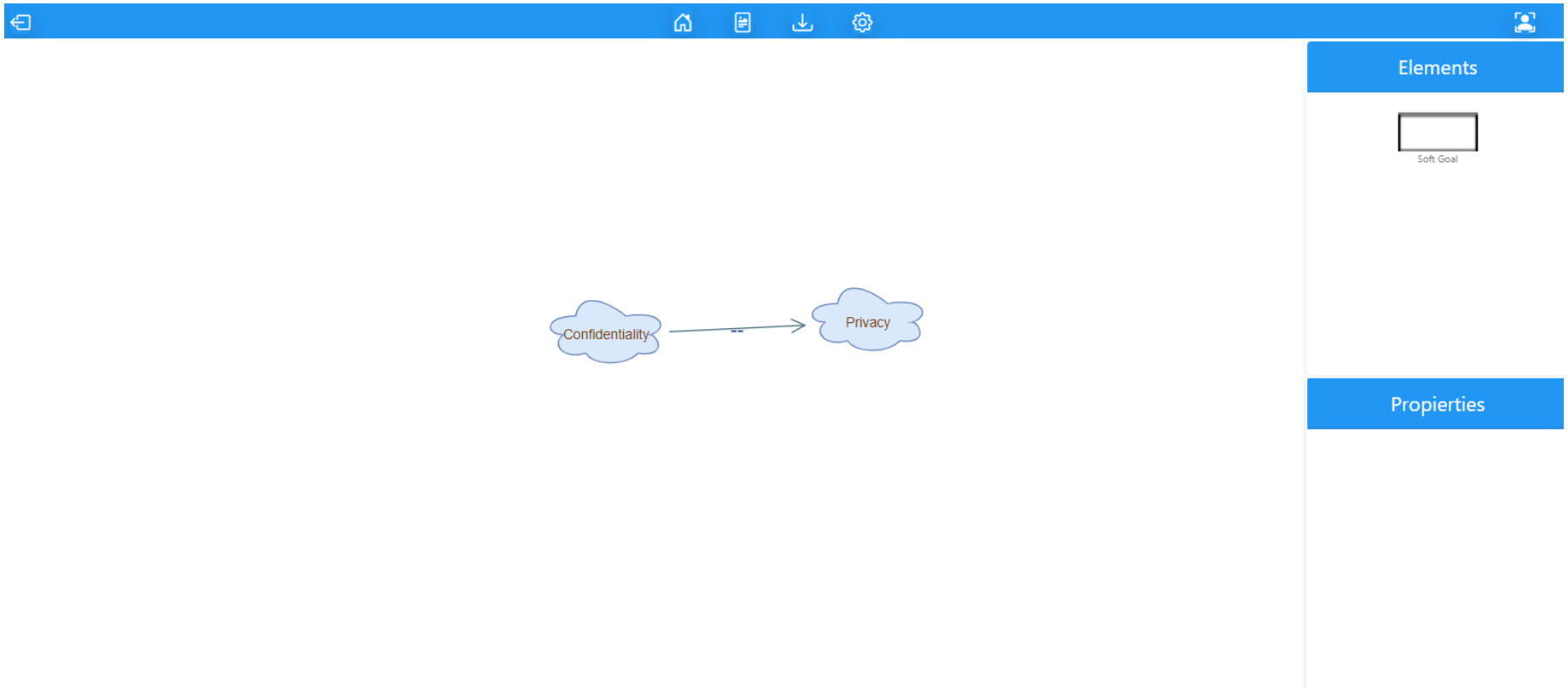
Security Criteria Diagram (Like asset diagram in CORAS)?

Let the user add the goals they need then use the ontology in a risk mode
(For example, the user needs confidentiality and privacy in the system. If a threat denies (to a certain level) confidentiality, it also denies privacy with a level that depends on the relationship defined between confidentiality and privacy in the ontology (level of confidentiality denial divided by the level of relationship).
Another idea: suggest additional security criteria?

Security Requirements Formalization

SERENA

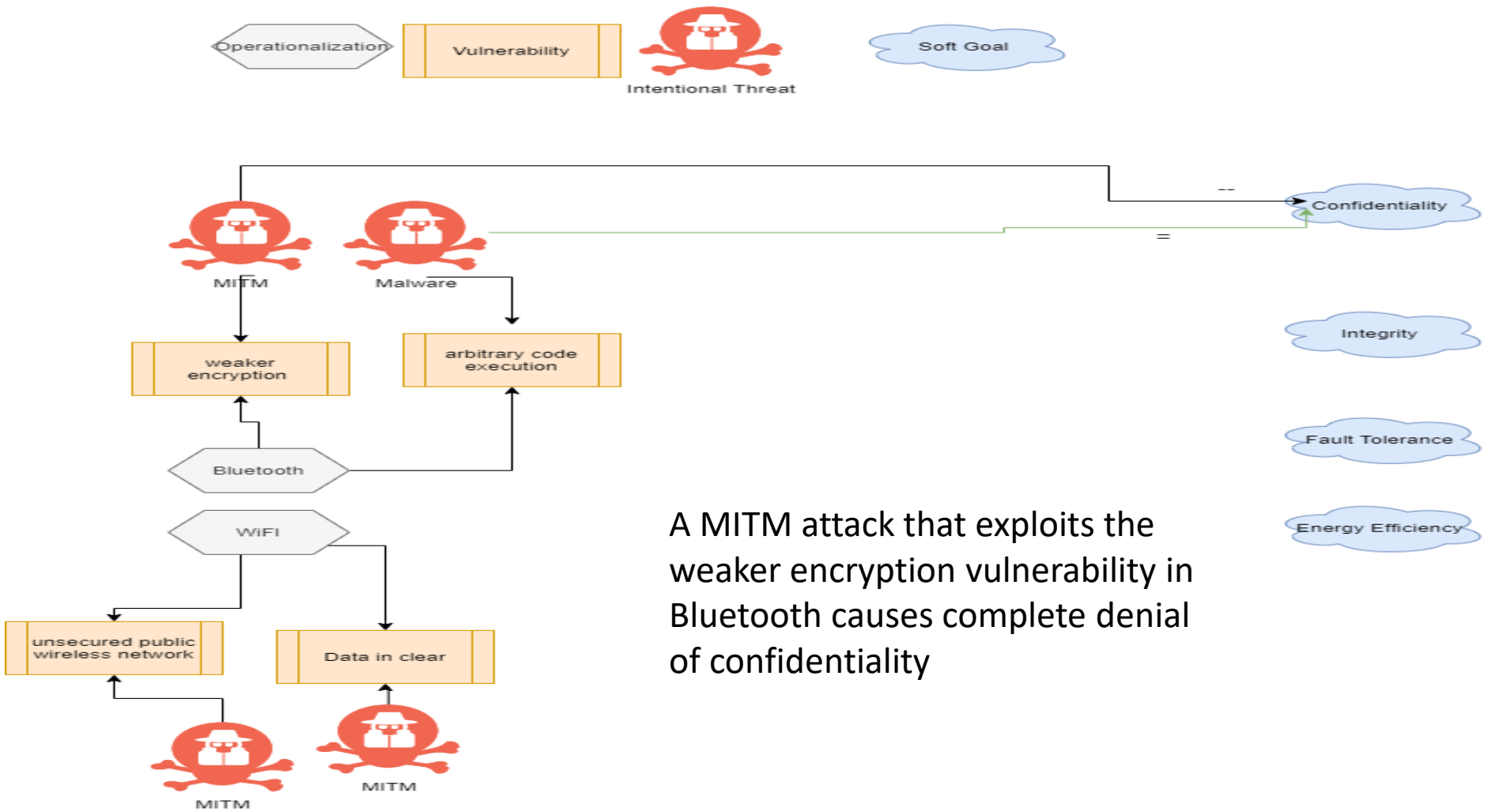
Security Criteria Diagram (Like asset diagram in CORAS)?



Security Requirements Formalization

SERENA

Risk Diagram Suggestion

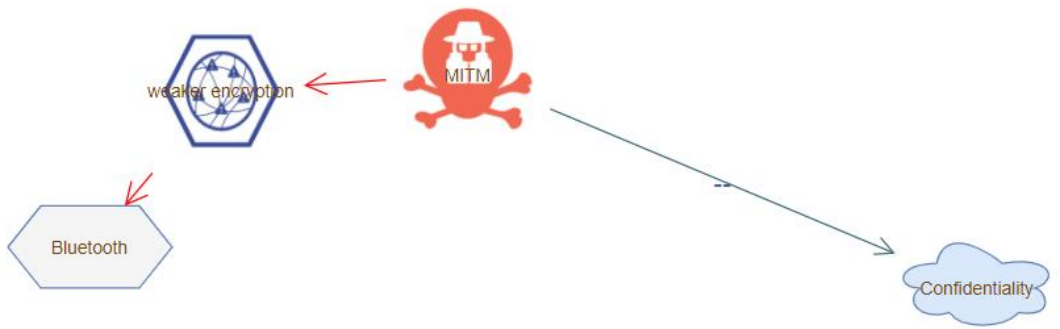


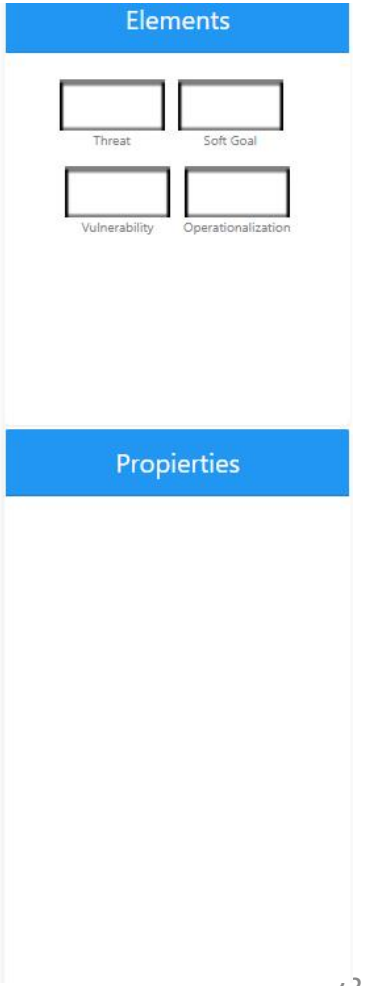
A MITM attack that exploits the weaker encryption vulnerability in Bluetooth causes complete denial of confidentiality

Security Requirements Formalization

SERENA

Risk Diagram Suggestion

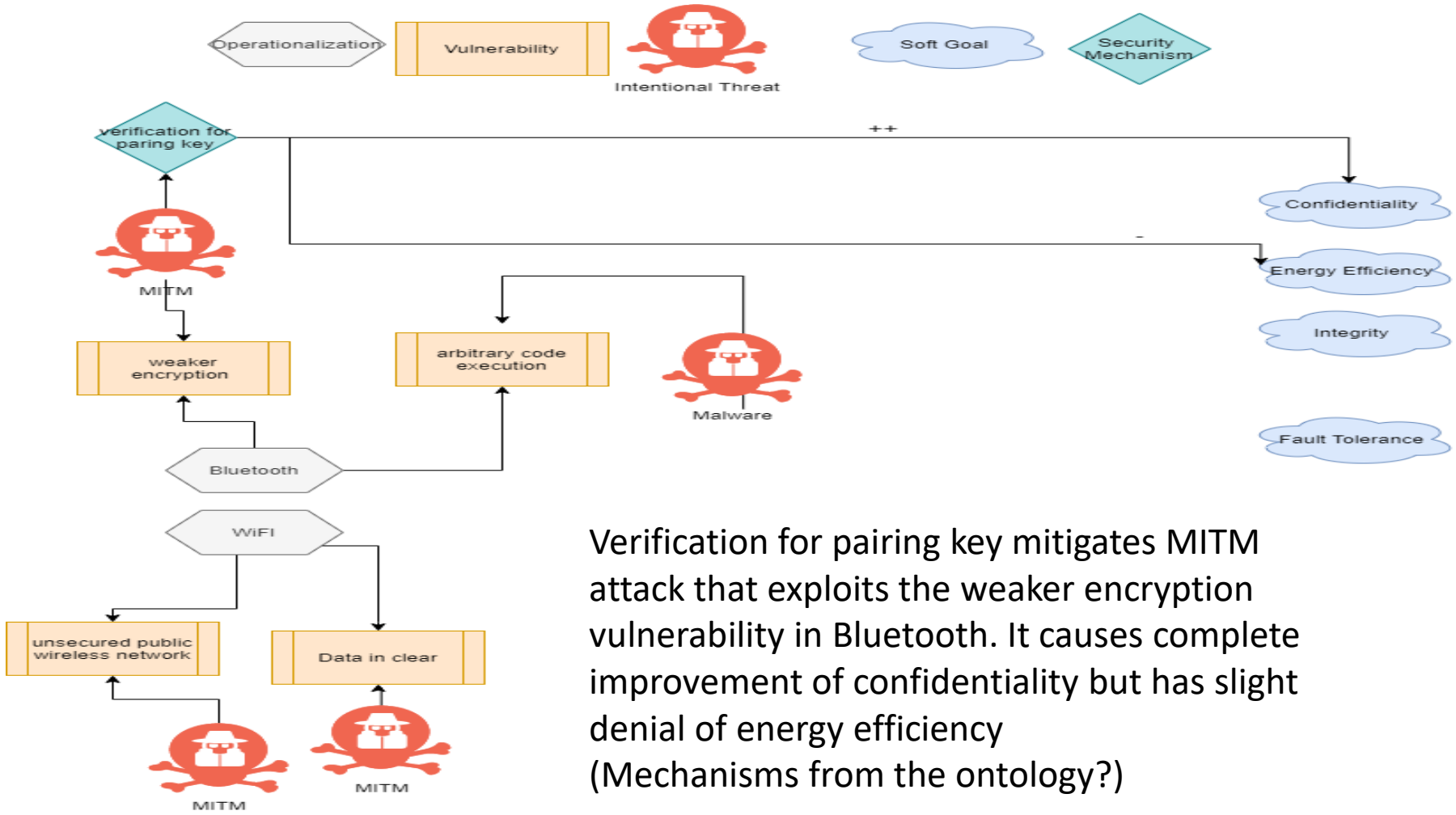




Security Requirements Formalization

SERENA

Treatment Diagram Suggestion

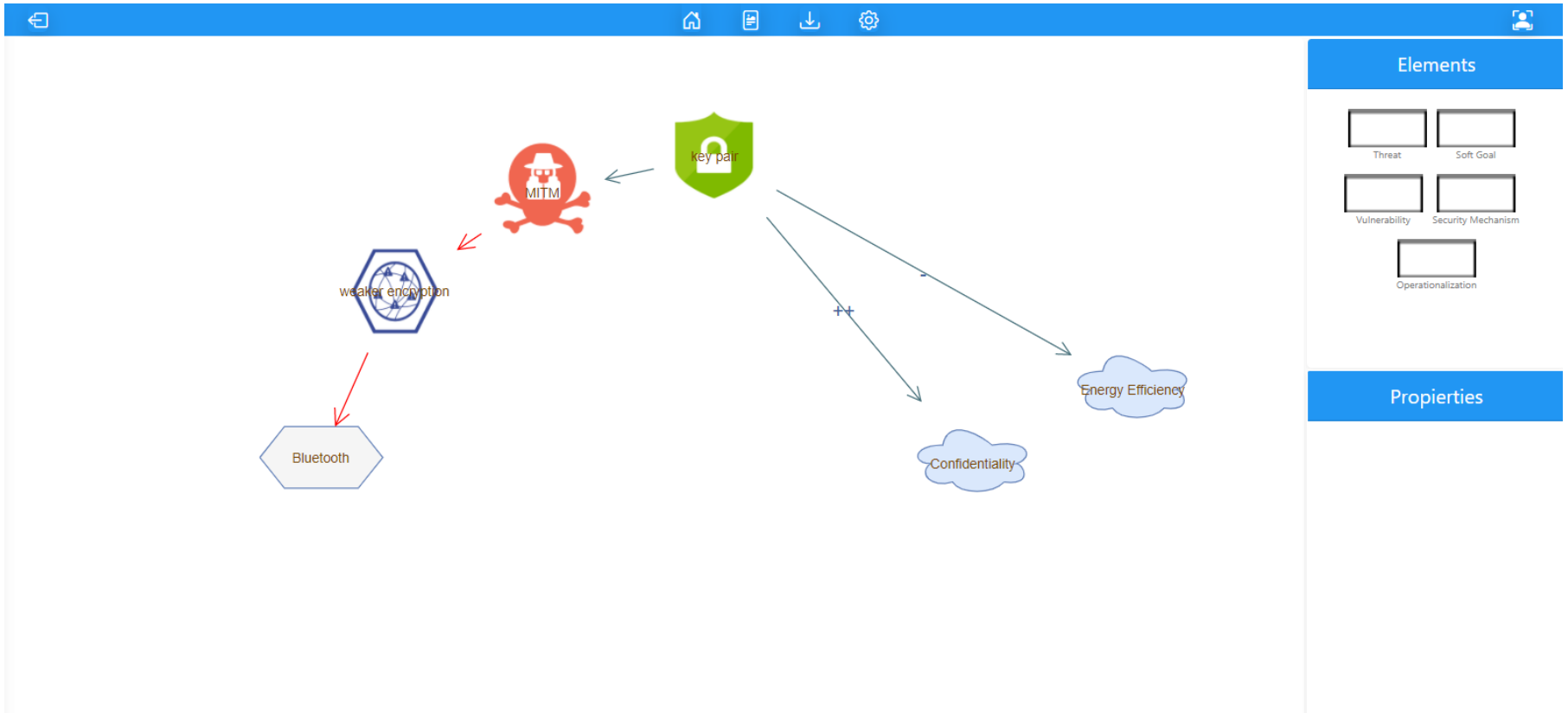


Verification for pairing key mitigates MITM attack that exploits the weaker encryption vulnerability in Bluetooth. It causes complete improvement of confidentiality but has slight denial of energy efficiency
(Mechanisms from the ontology?)

Security Requirements Formalization

SERENA

Treatment Diagram Suggestion

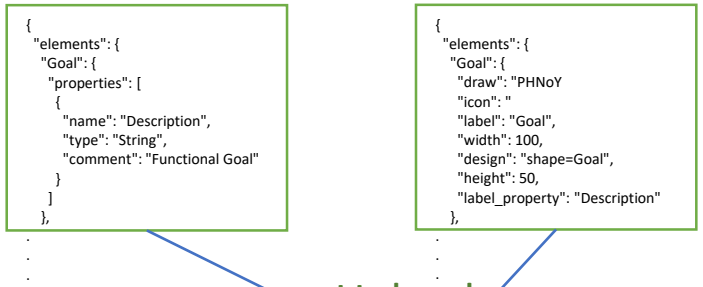


Security Analysis

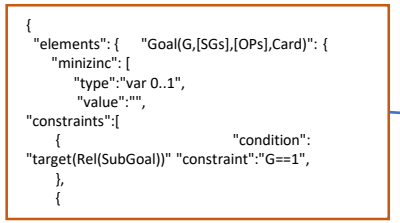
SERENA Using Variamos

1- Abstract Syntax – Language definition (Elements & Relationships & Restrictions) (JSON)

1- Concrete Syntax – Graphical Definition of the Language (JSON)



Upload



Upload

VARIAMOS

2- SERENA Model

3-Semantic Syntax – How is the language translated (JSON). Example: How to translate a SecureKAOS model into minizinc code.

Variamos Model JSON + Semantic Definition JSON + Name of the model + Type of the model (domain or application)

Translator (minizinc)

Minizinc code of the model

1 - I want to create a new language!

2- I want to create a model using this language!

3- I want to analyze the model I created

Security Analysis

SERENA Using Variamos