Security by Design: An asset-based approach to bridge the gap between architects and security experts

Nan (Zhang) Messe

7th October 2021

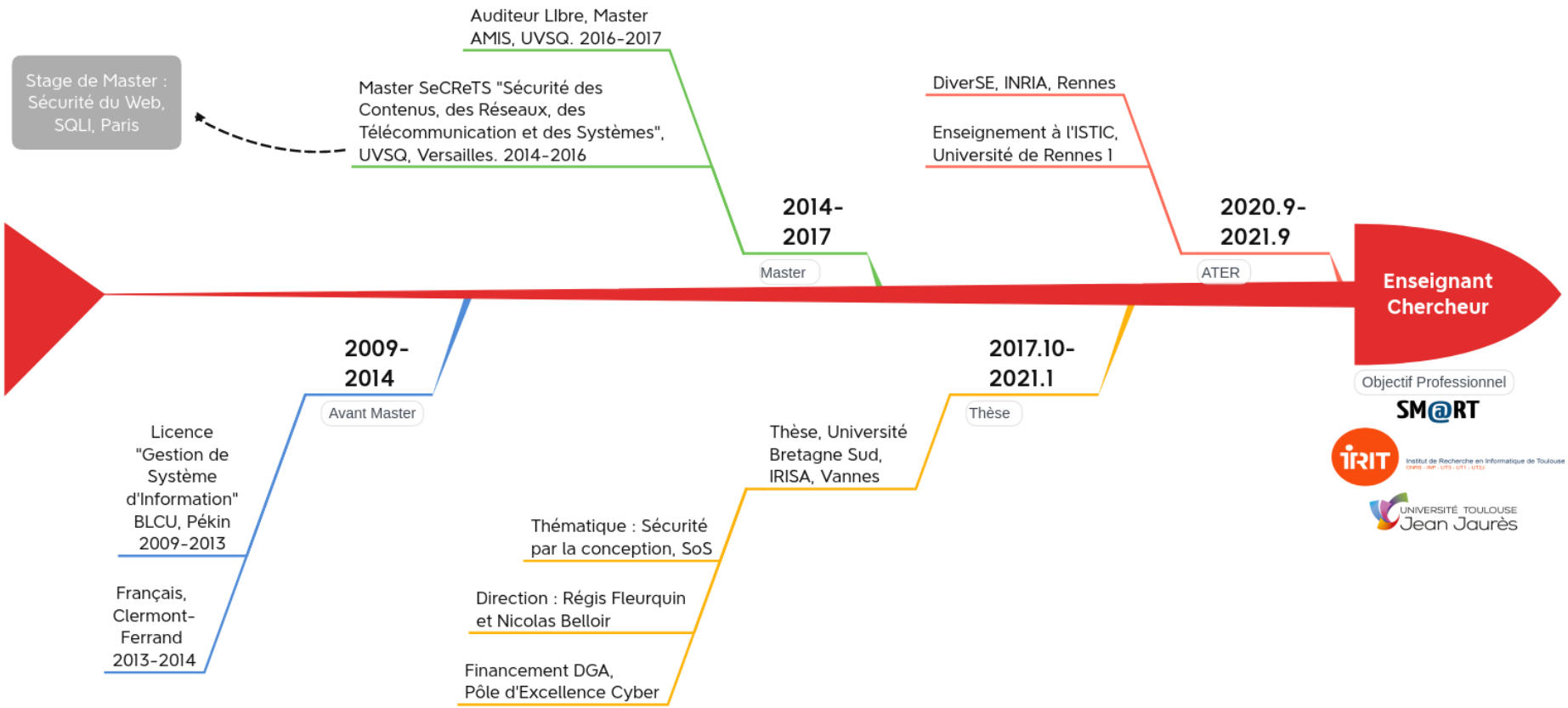
SM@RT

IRIT Institut de Recherche en Informatique de Toulouse
CNRS - INP - UT3 - UT1 - UT2J

UNIVERSITÉ TOULOUSE Jean Jaurès

1. https://nanzhangmesse.github.io/

Self Presentation

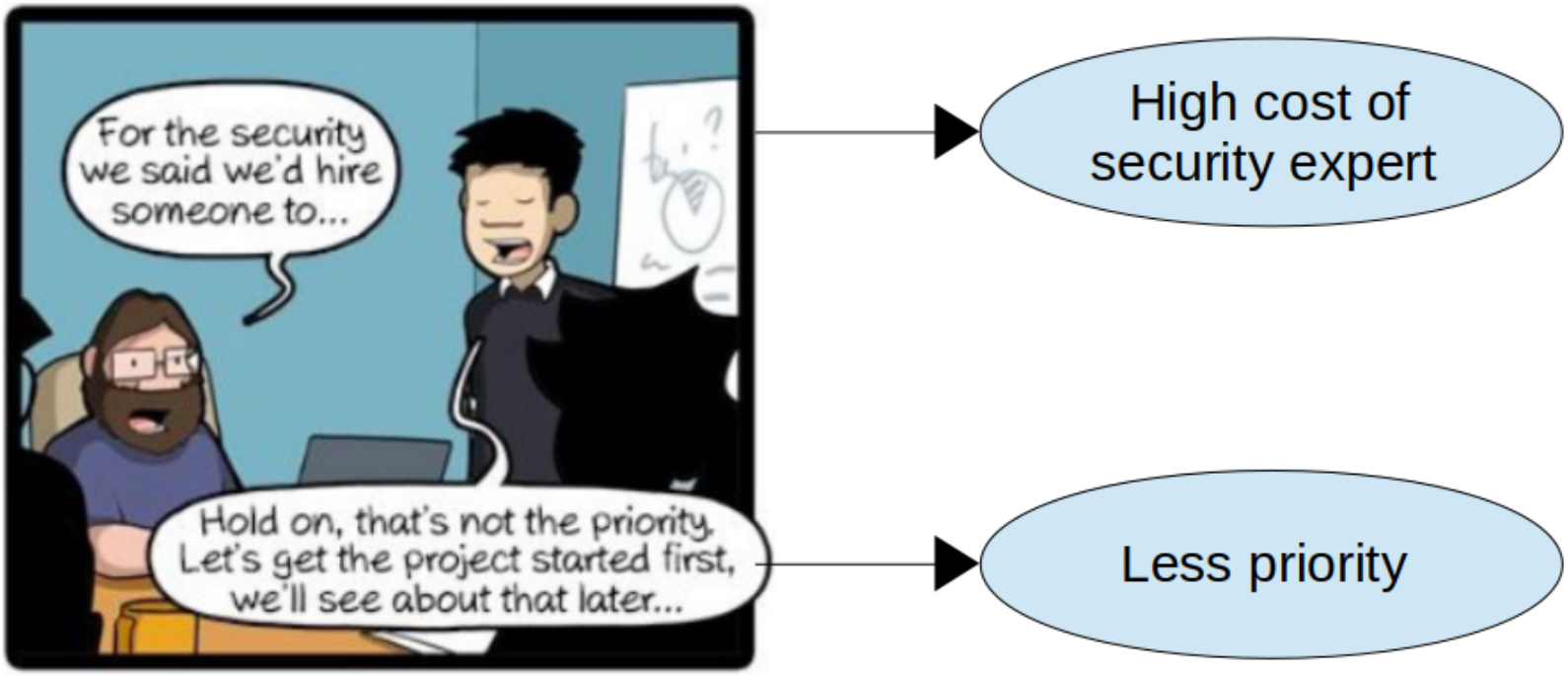


Agenda

1. **Problem statement**
2. **Contribution**:
   - **Contribution 1**: a security assistance
   - **Contribution 2**: a structured threat modeling
3. **Conclusion and perspectives**

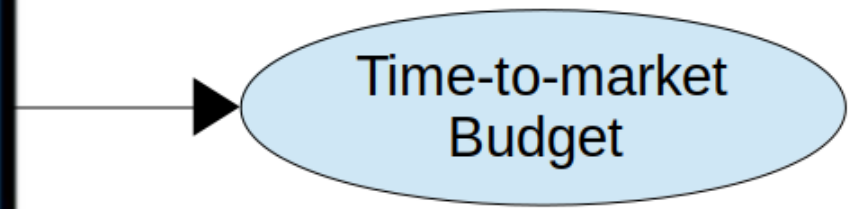
1. Problem statement
   - Problem statement
   - State-of-the-art
   - Objective
2. Contribution 1. Security assistance
3. Contribution 2. Structuring threat modeling
4. Conclusion

Problem statement



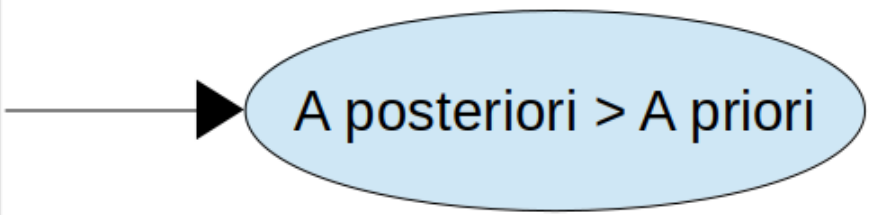
2. Valentina Casola et al. "A novel Security-by-Design methodology : modeling and assessing security with a quantitative approach". In : *Journal of Systems and Software* 163 (mai 2020).

Problem statement



3. Adam Shostack. "Experiences Threat Modeling at Microsoft". In : *Proceedings of the MODSEC@MoDELS workhop*. 2008.

Problem statement



4. A. Van Den Berghe et al. "A lingua franca for security by design". In : *3rd Annual IEEE Cybersecurity Development Conference (IEEE SecDev)*. IEEE COMPUTER SOC, nov. 2018, p. 69-76.

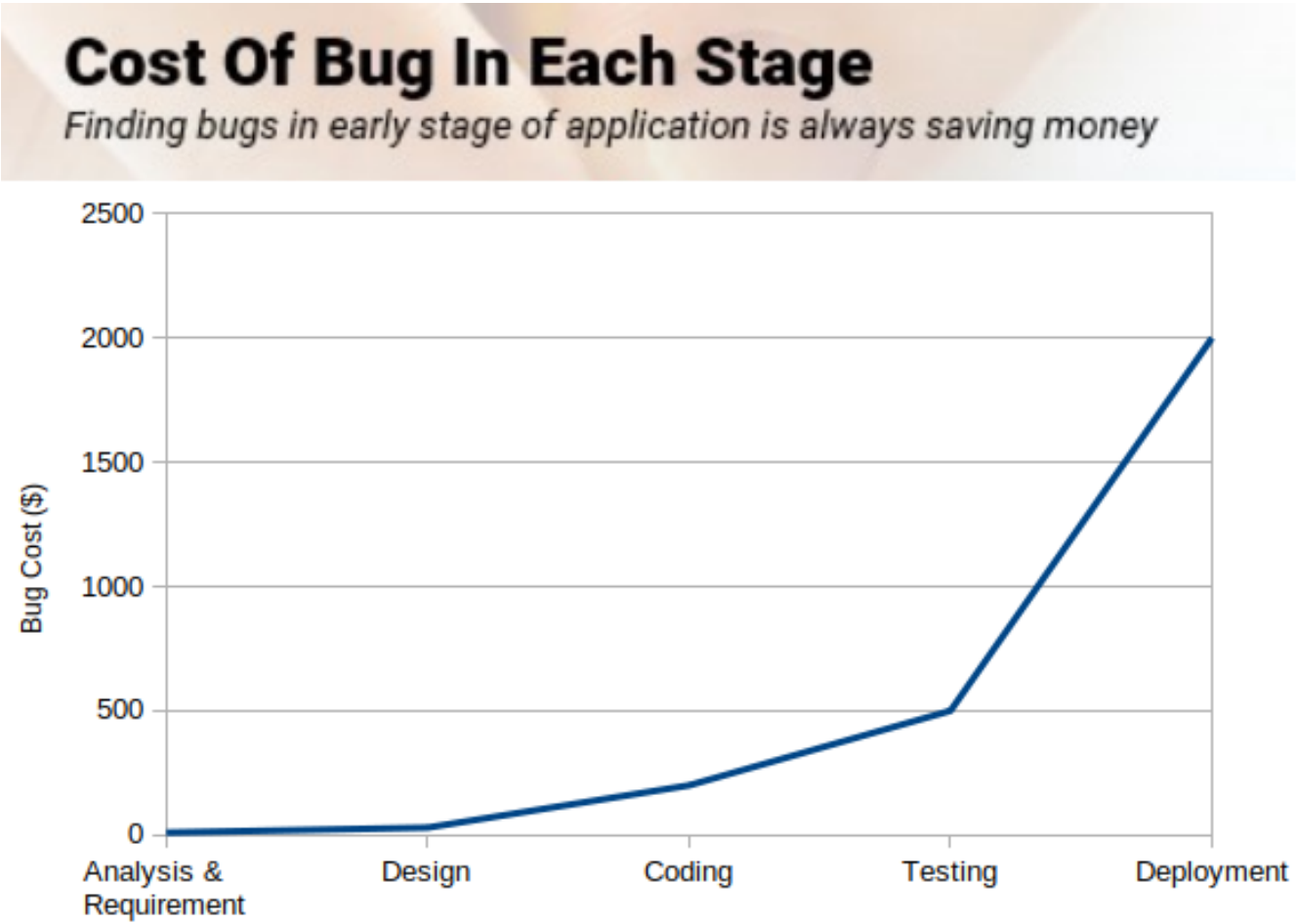
Problem statement

Organized and sophisticated attacks

High variability of threats

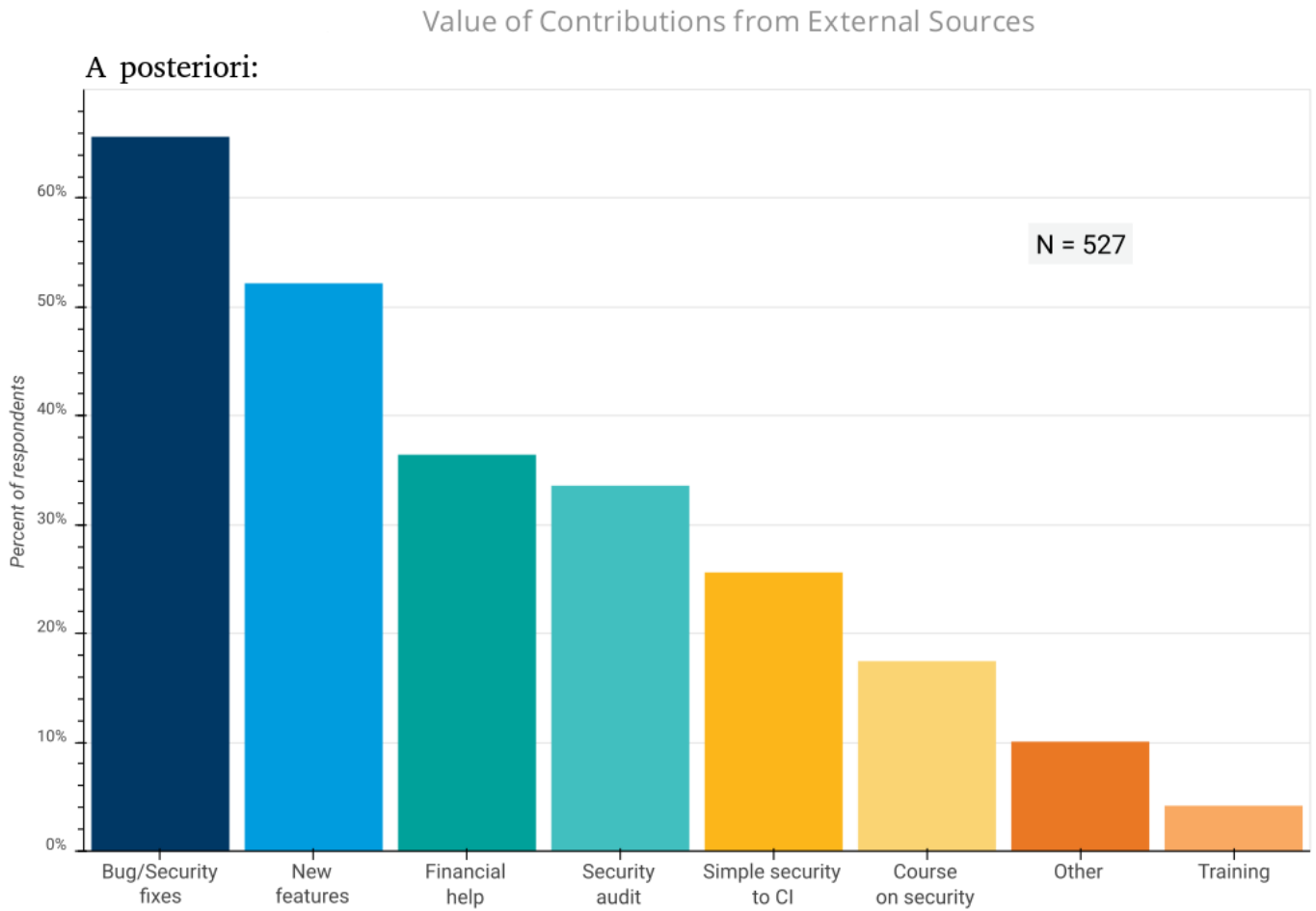
5. Microsoft Security Response Center. *Guidance for WannaCrypt attacks*. 2017.
6. Symantec Security Response. *Petya ransomware : Here's what you need to know*. 2017.
7. TechTarget. *BlueKeep (CVE-2019-0708)*. Juin 2019.

Problem statement



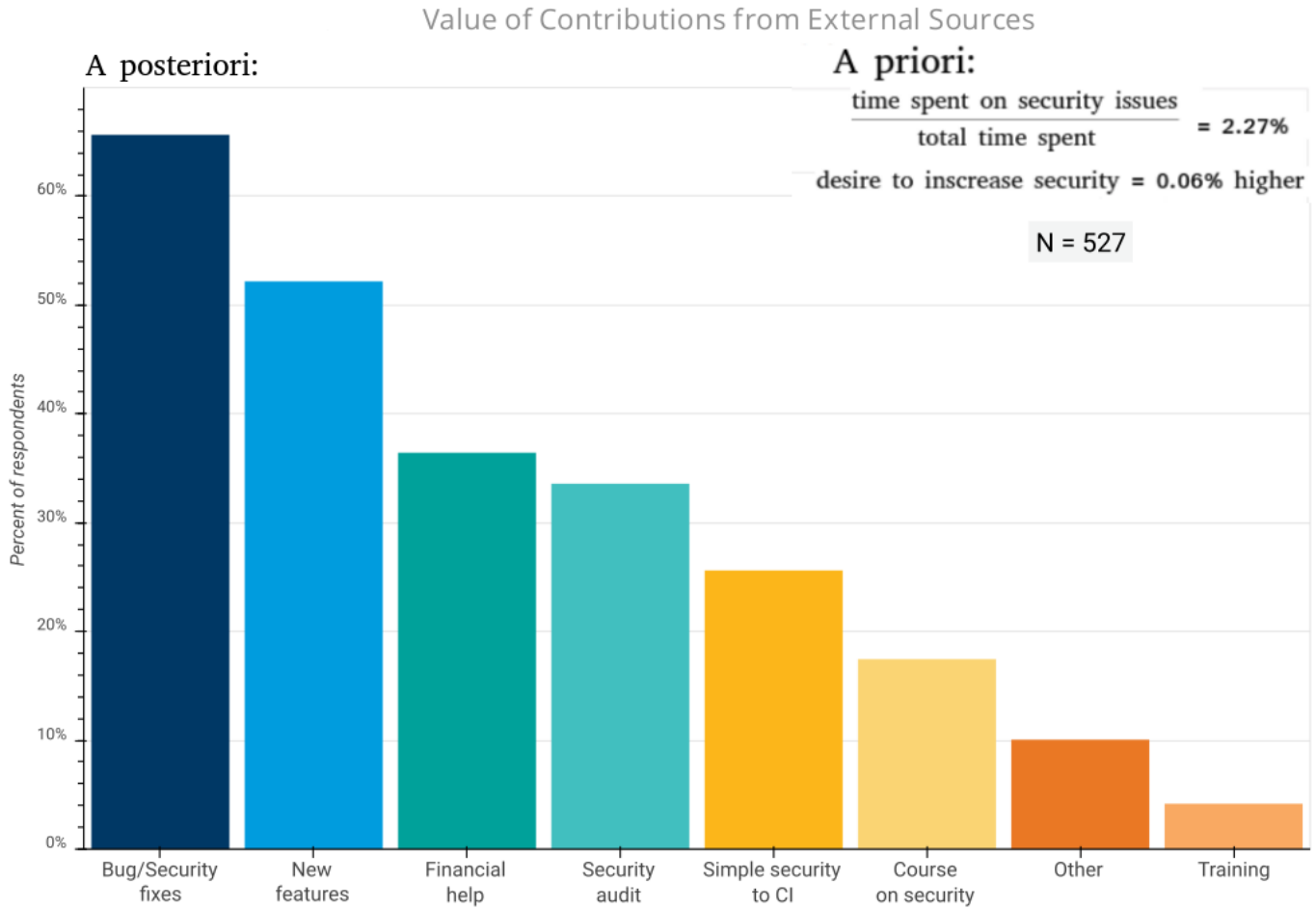
8. SaiRaj Mahesh. *Why Should Companies migrate from SDLC to Secure SDLC*. Nov. 2018. url : https://medium.com/@sairajmahesh/why-should-companies-migrate-from-sdlc-to-secure-sdlc-4264ea52be1f.

Problem statement



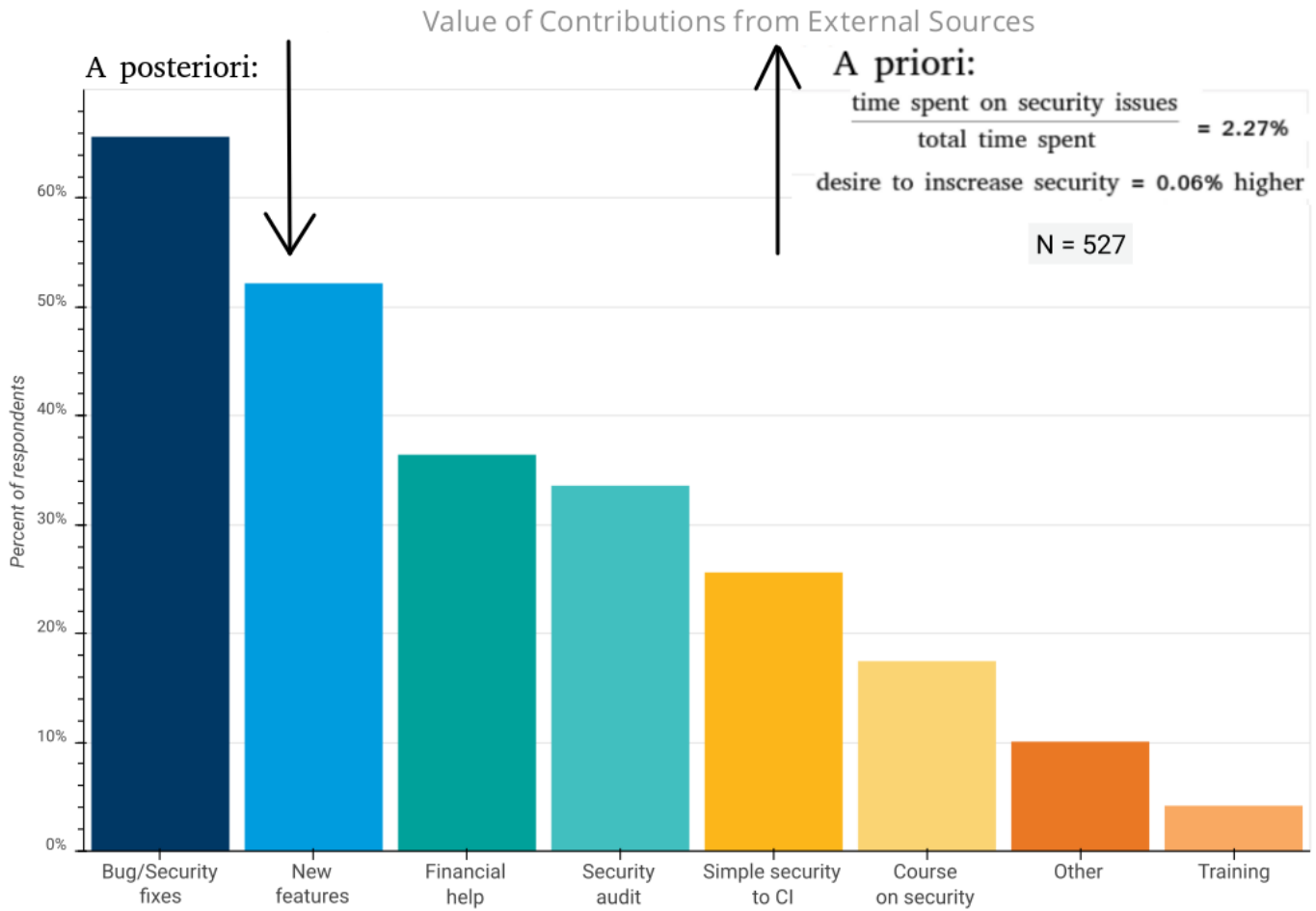
9. The Linux Foundation. *Report on the 2020 FOSS Contributor Survey*. standard. The Linux Foundation & The Laboratory for Innovation Science at Harvard, déc. 2020, p. 1-96.

Problem statement



9. The Linux Foundation. *Report on the 2020 FOSS Contributor Survey*. standard. The Linux Foundation & The Laboratory for Innovation Science at Harvard, déc. 2020, p. 1-96.

Problem statement



9. The Linux Foundation. *Report on the 2020 FOSS Contributor Survey*. standard. The Linux Foundation & The Laboratory for Innovation Science at Harvard, déc. 2020, p. 1-96.

Security-by-design

Domain Expertise

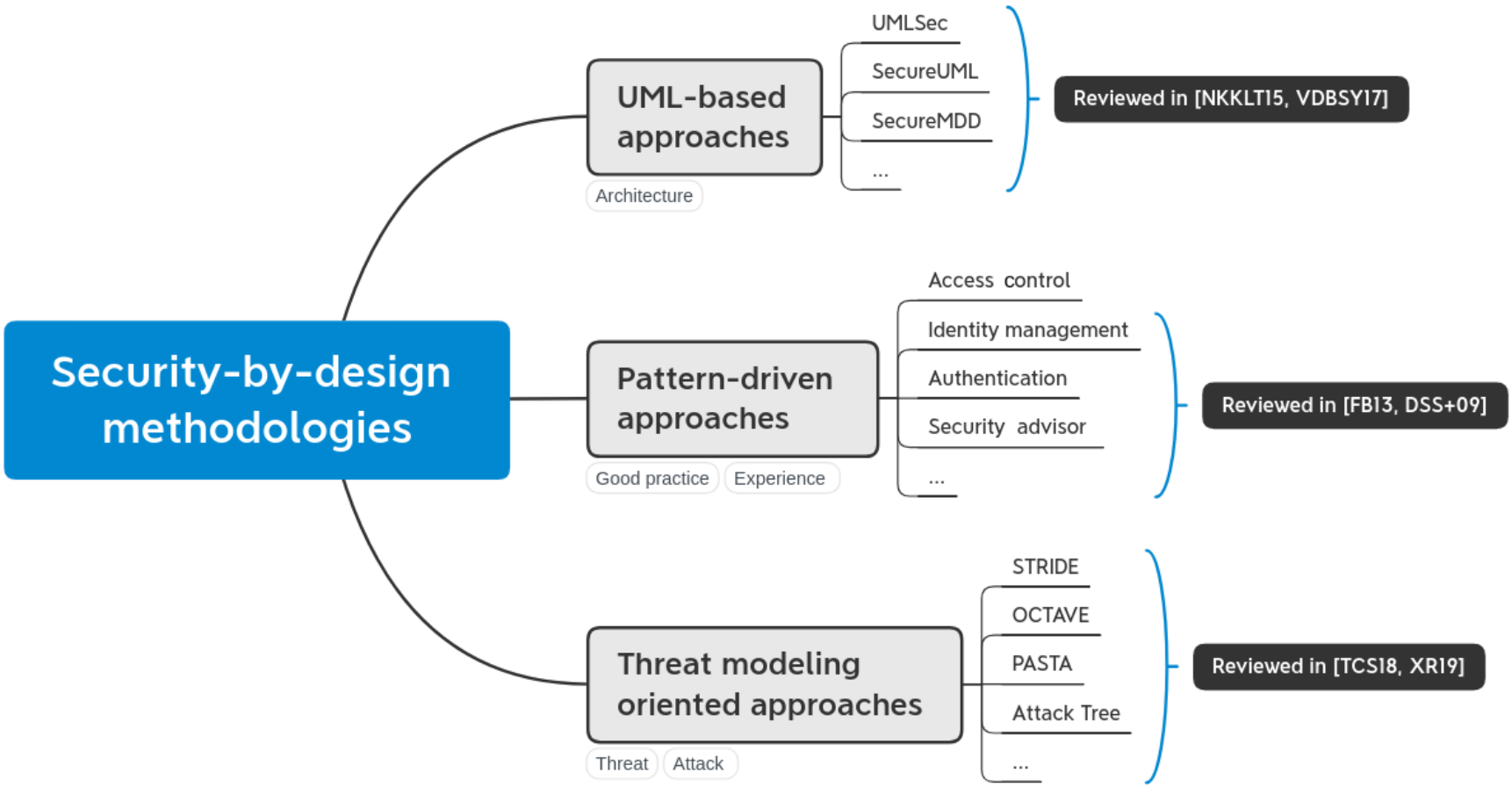
Security-by-design

Security Expertise

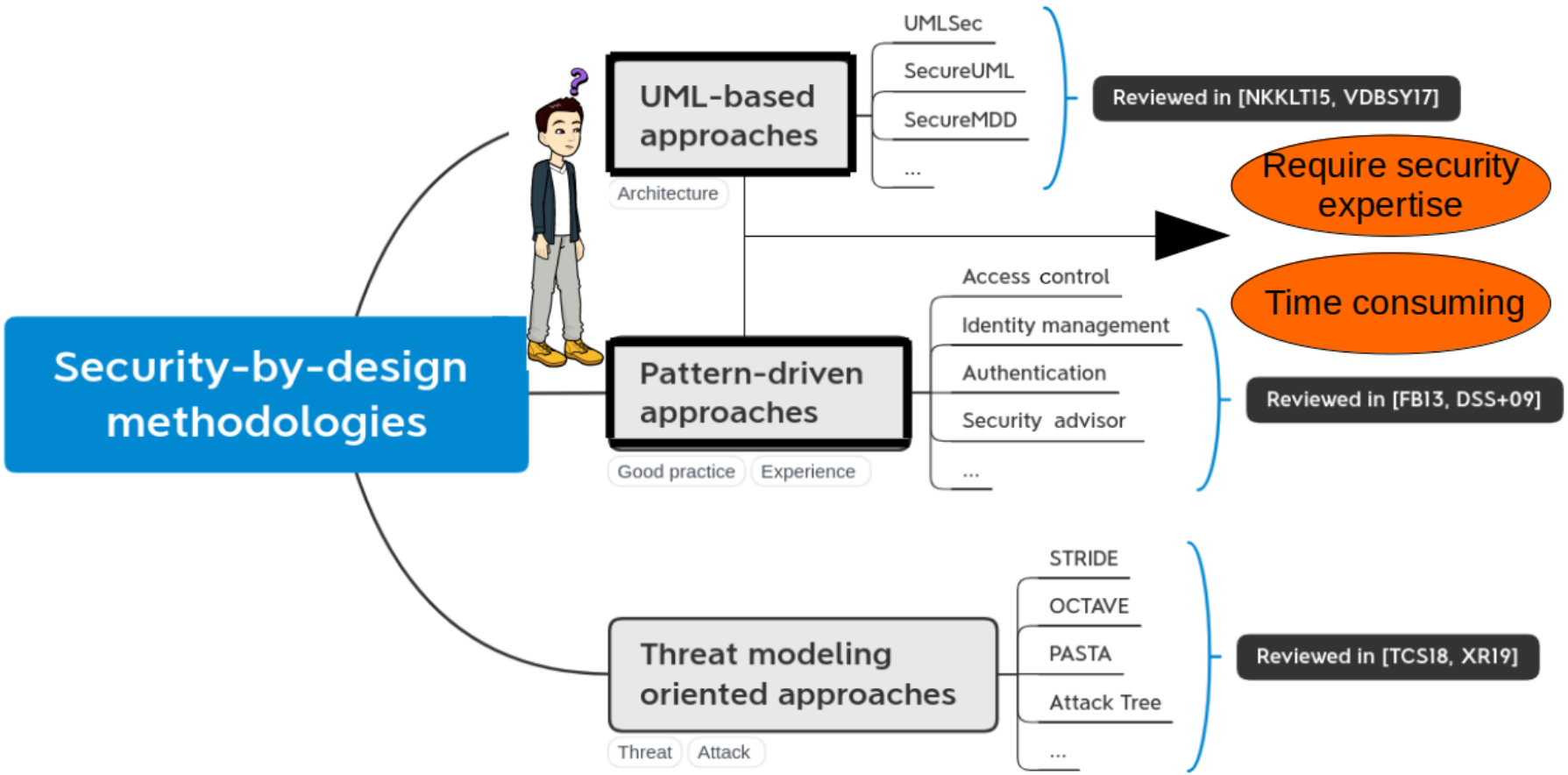
"Considering security as early as the design phase of the software development process"
(A Priori)

10. Michael Waidner, Michael Backes et Jörn Müller-Quade. "Development of secure software with security by design". In : *Fraunhofer Institute for Secure Information Technology, SIT technical reports, SIT-TR-2014-03* 12 (2014).

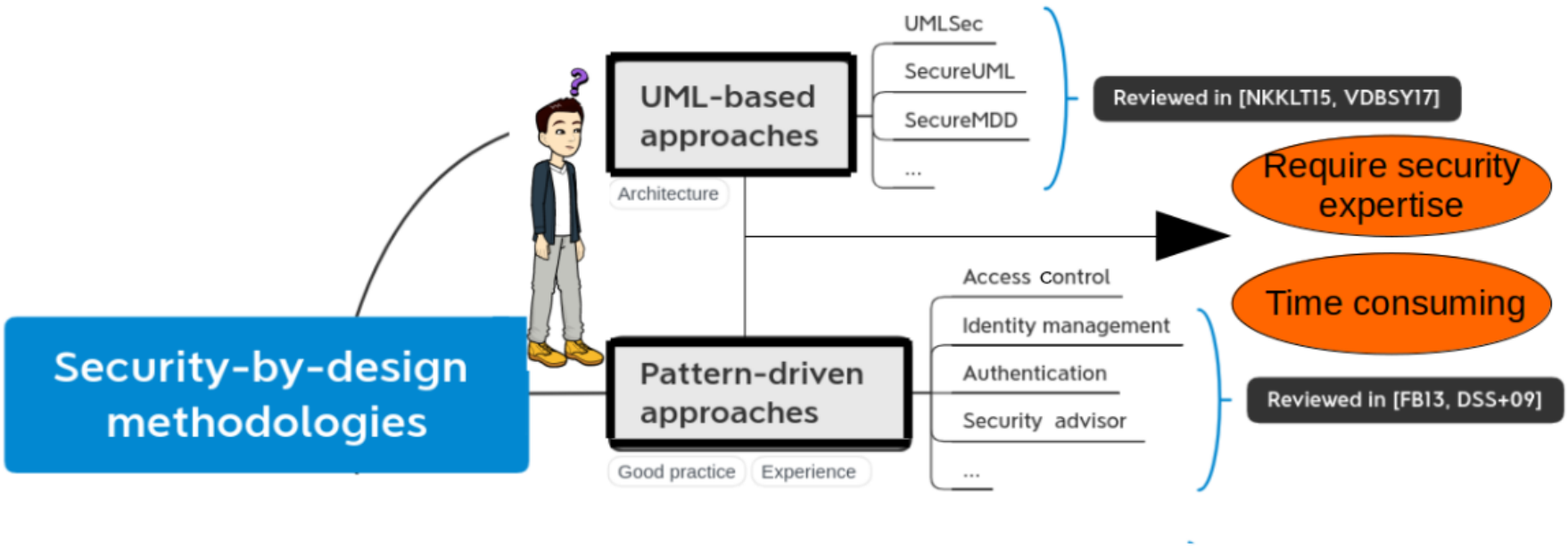
Current security-by-design methodologies



Current security-by-design methodologies

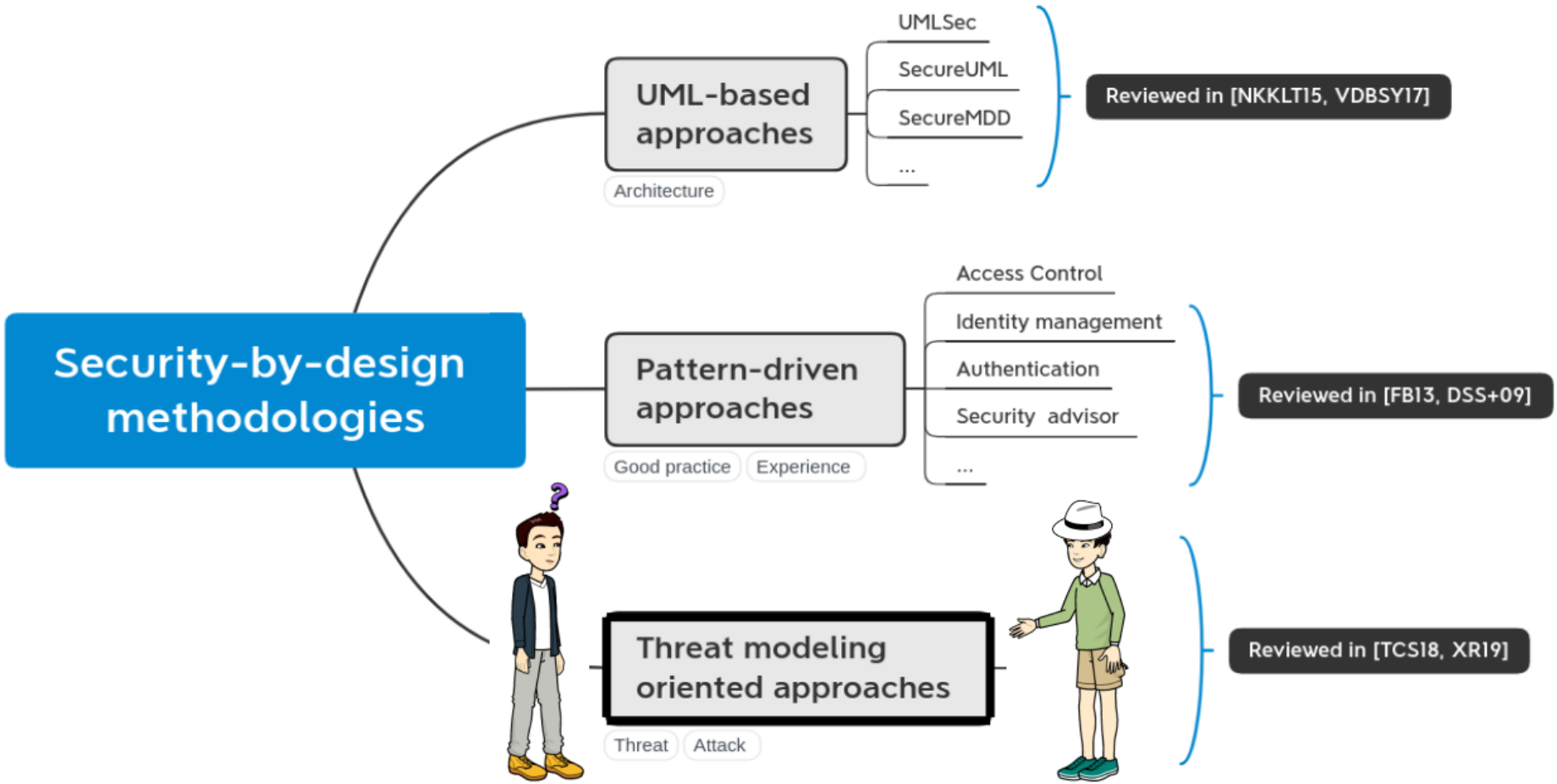


Current security-by-design methodologies

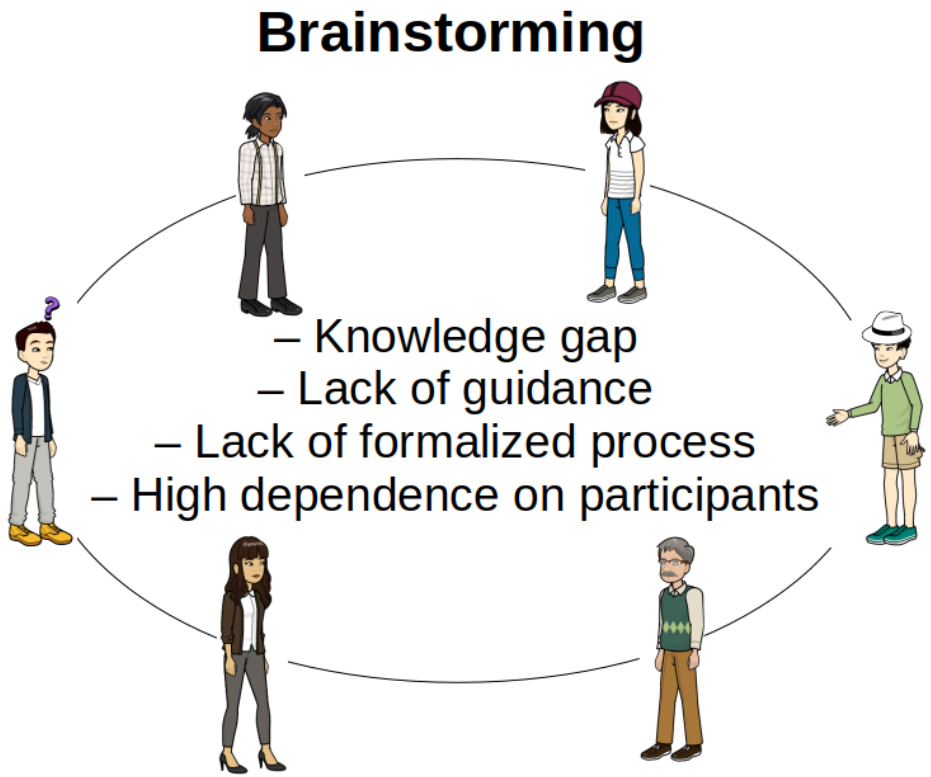


RQ1 : How can we assist architects who have limited security knowledge to integrate security aspects at the design phase (from the architect' viewpoint) ?

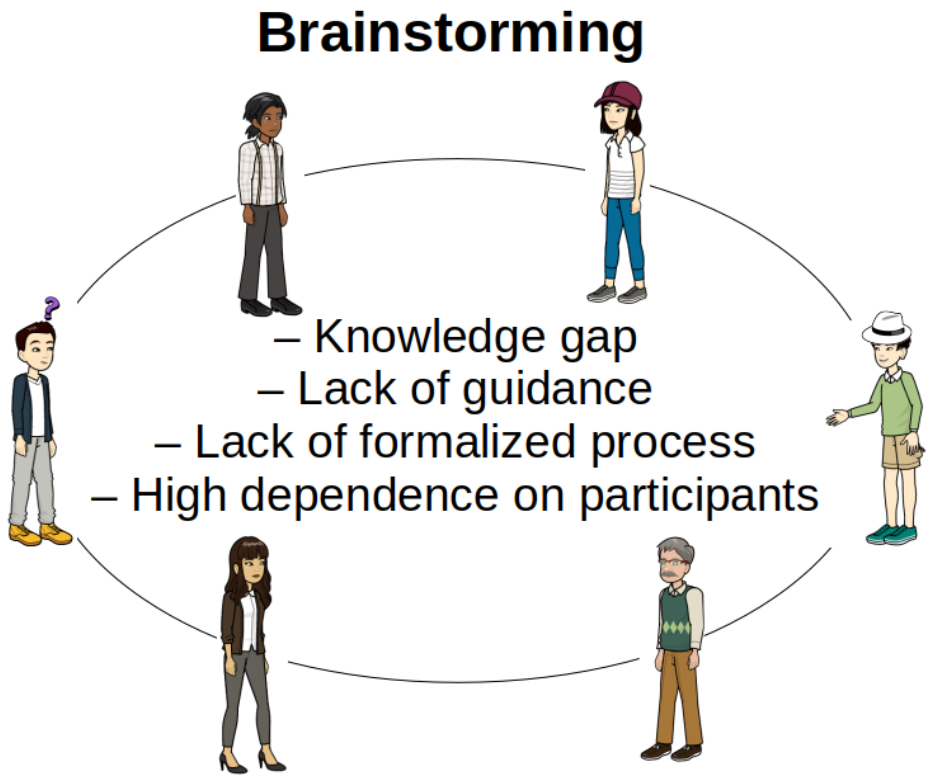
Current security-by-design methodologies



Threat modeling limitations



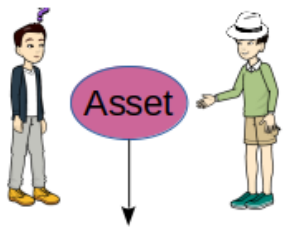
Threat modeling limitations



RQ2 : How can we assist security experts to structure the threat modeling process to ensure the security-by-design (from security expert's viewpoint) ?

An inventory of industrial threat modeling processes

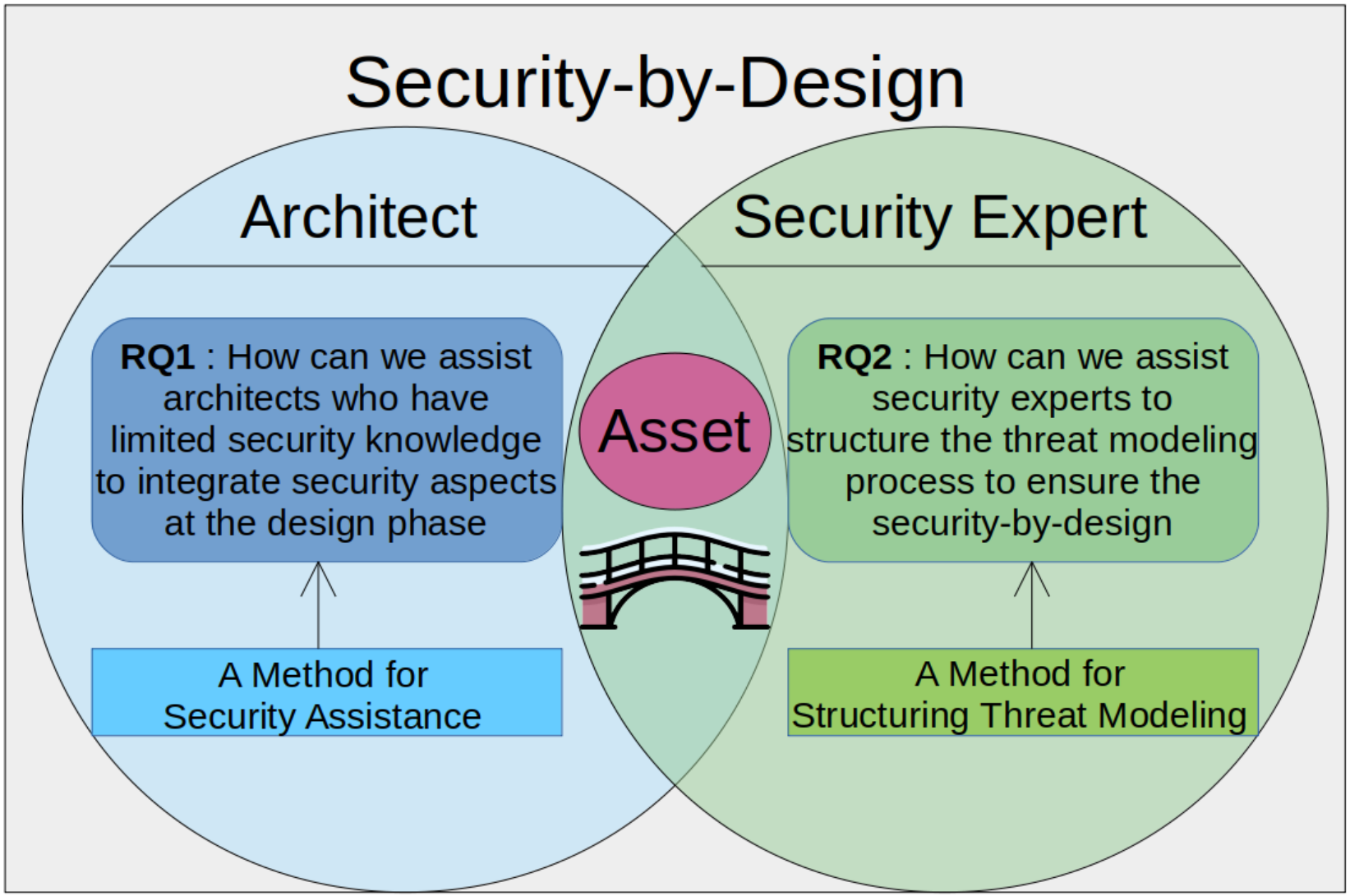
Phase	Asset Identification			Threat Enumeration				Threat Prioritization		Mitigation	
Activity / Paper	Identify security goal	Model domain	Identify asset	Identify threat	Enumerate &document threat	Describe attacker	Identify vulnerability	Rate threat	Assess risk	Mitigation	Verification
Torr (2005) [15]		X		X						X	X
Shostack (2008) [12]		X			X					X	X
Scandariato (2013) [11]		X		X	X						
Beckers (2013) [1]		X	X	X	X	X					
Dhillon (2011) [4]		X		X					X	X	
Steven (2010) [13]	X	X		X			X				
Kamatchi (2016) [6]		X	X	X	X			X			



'Anything that has value to an organization'

11. Joel Brenner. "ISO 27001 : Risk management and compliance". In : *Risk management* 54.1 (2007), p. 24.

Bridging the gap between architects and security experts



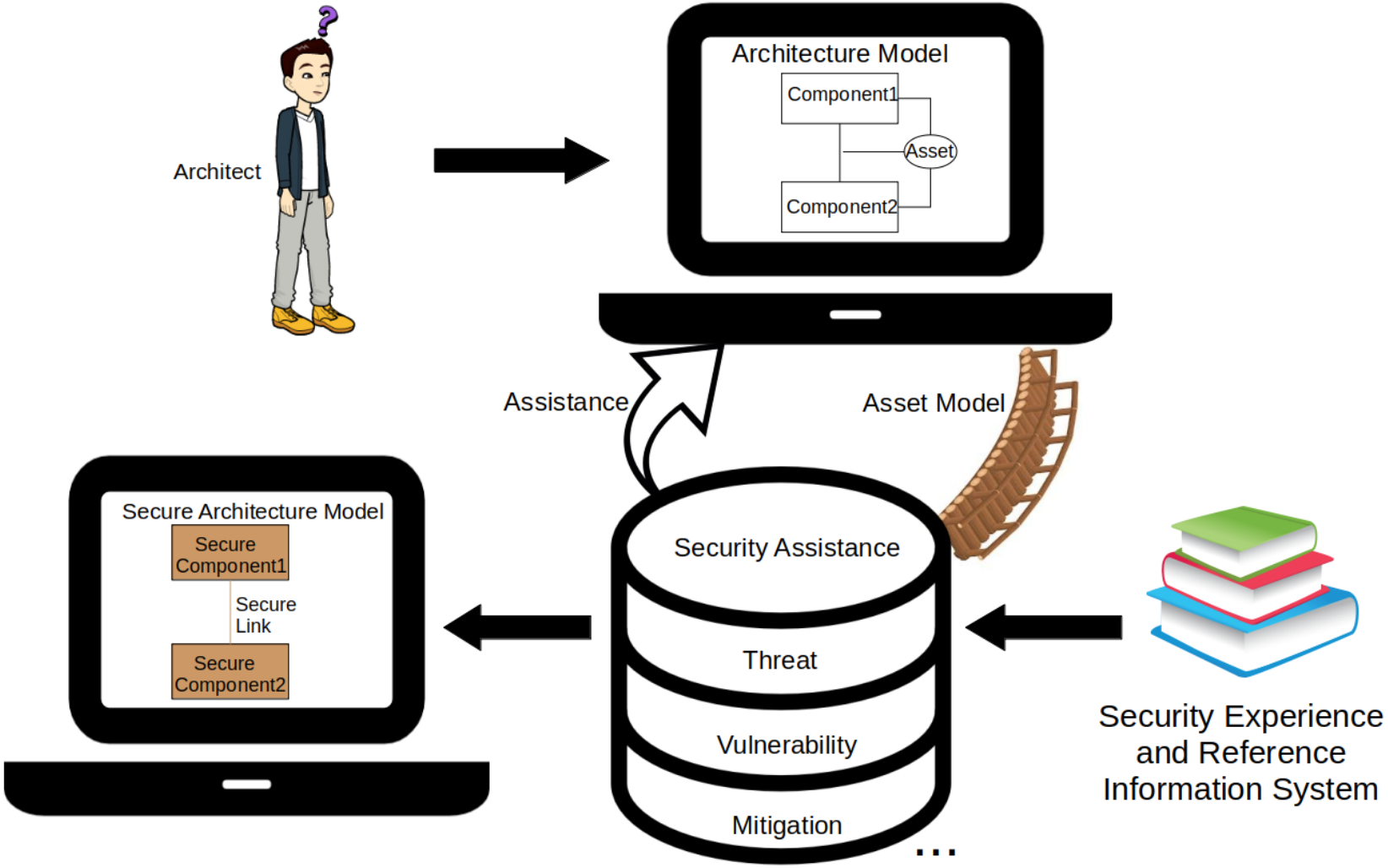
1 Problem statement

2 Contribution 1. Security assistance
- General description
- Foundation of the security assistance
- Security assistance enactment

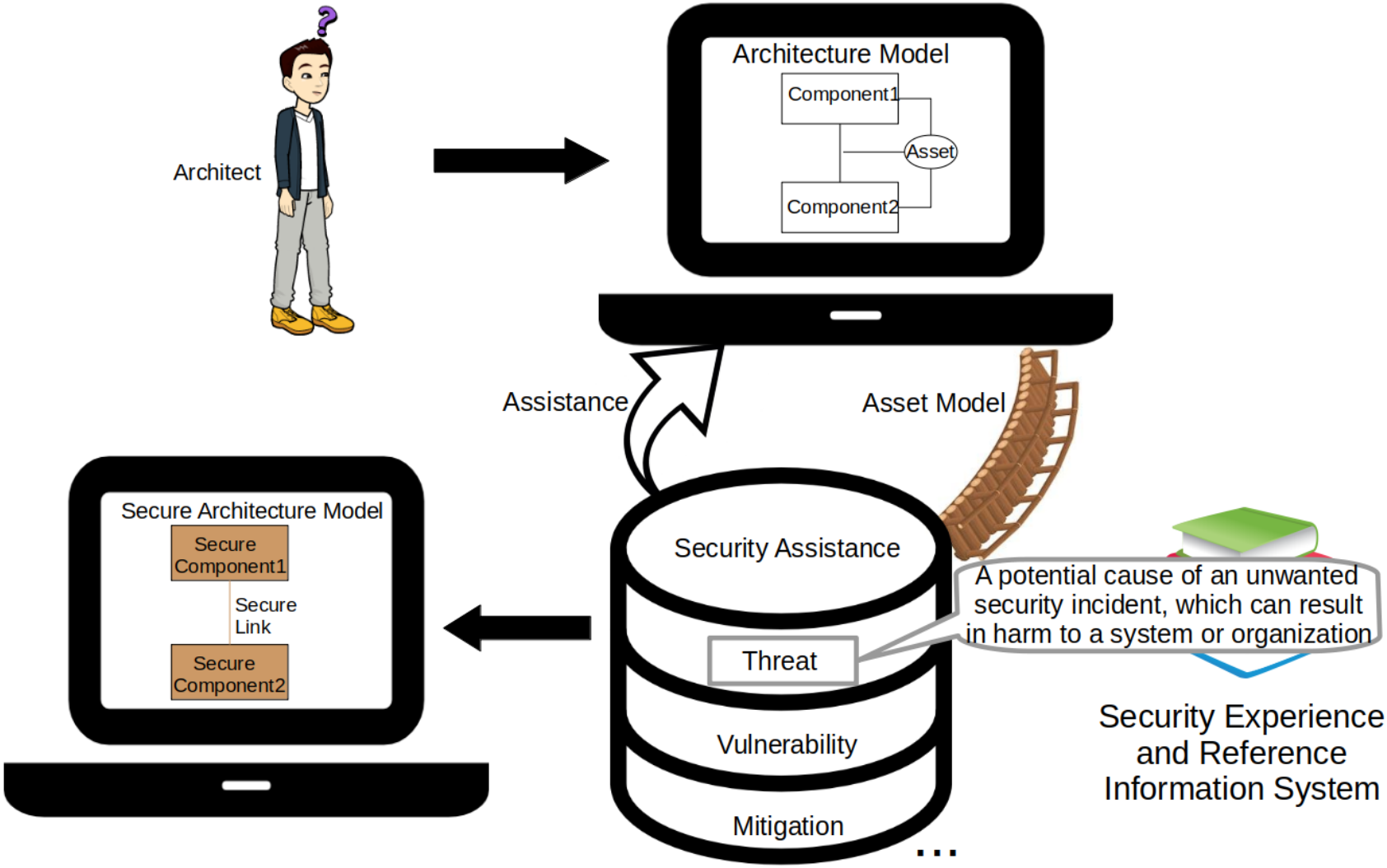
3 Contribution 2. Structuring threat modeling

4 Conclusion

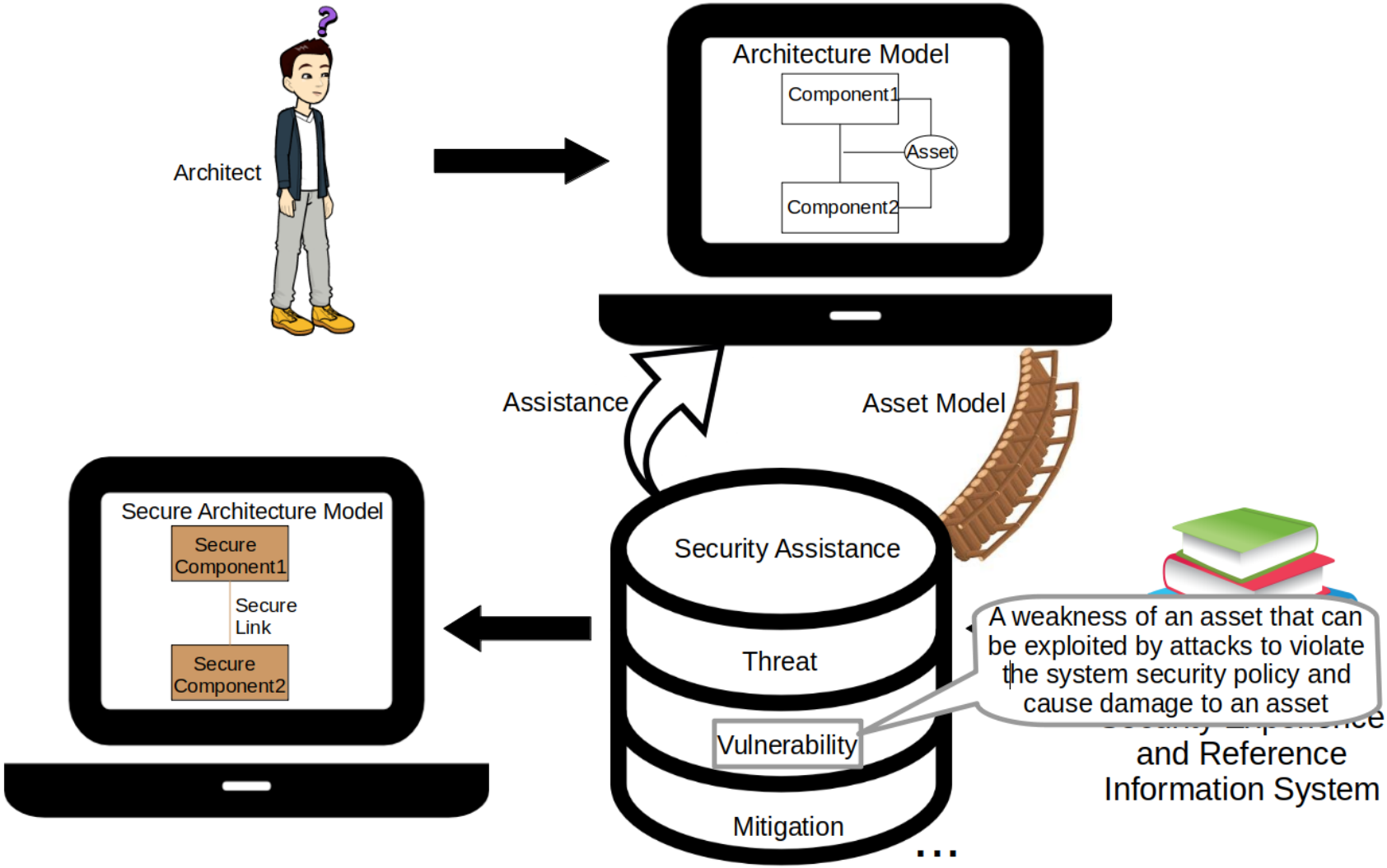
The objective of the security assistance



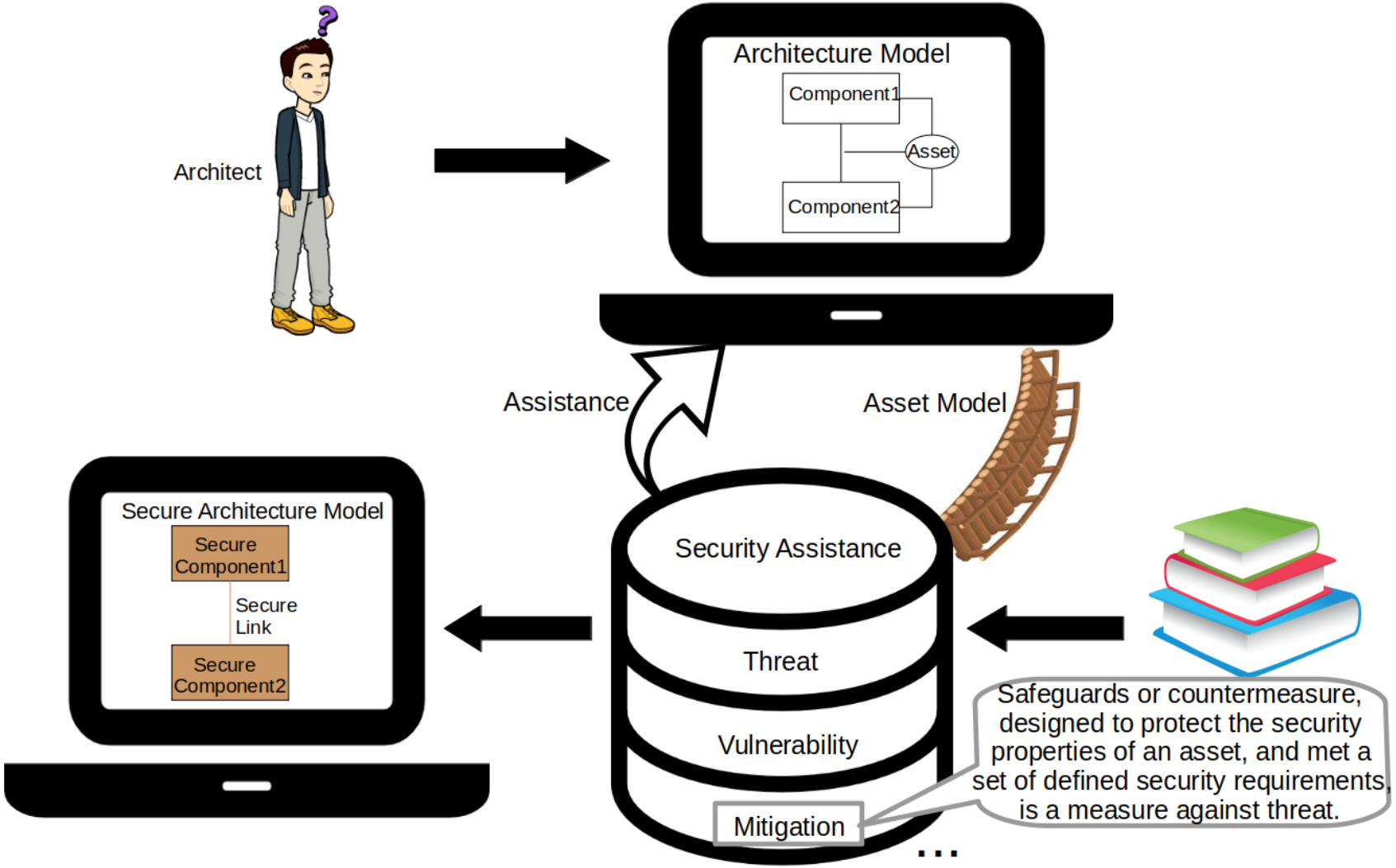
The objective of the security assistance



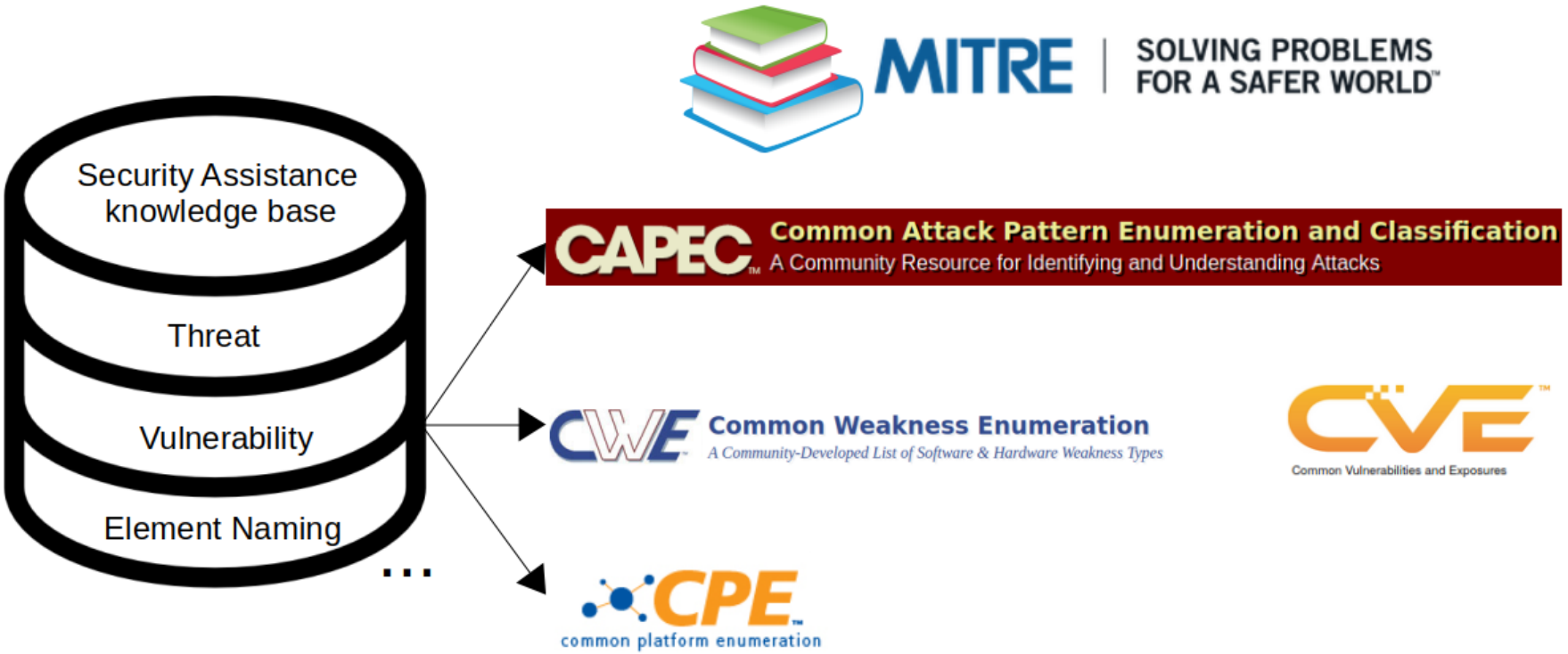
The objective of the security assistance



The objective of the security assistance

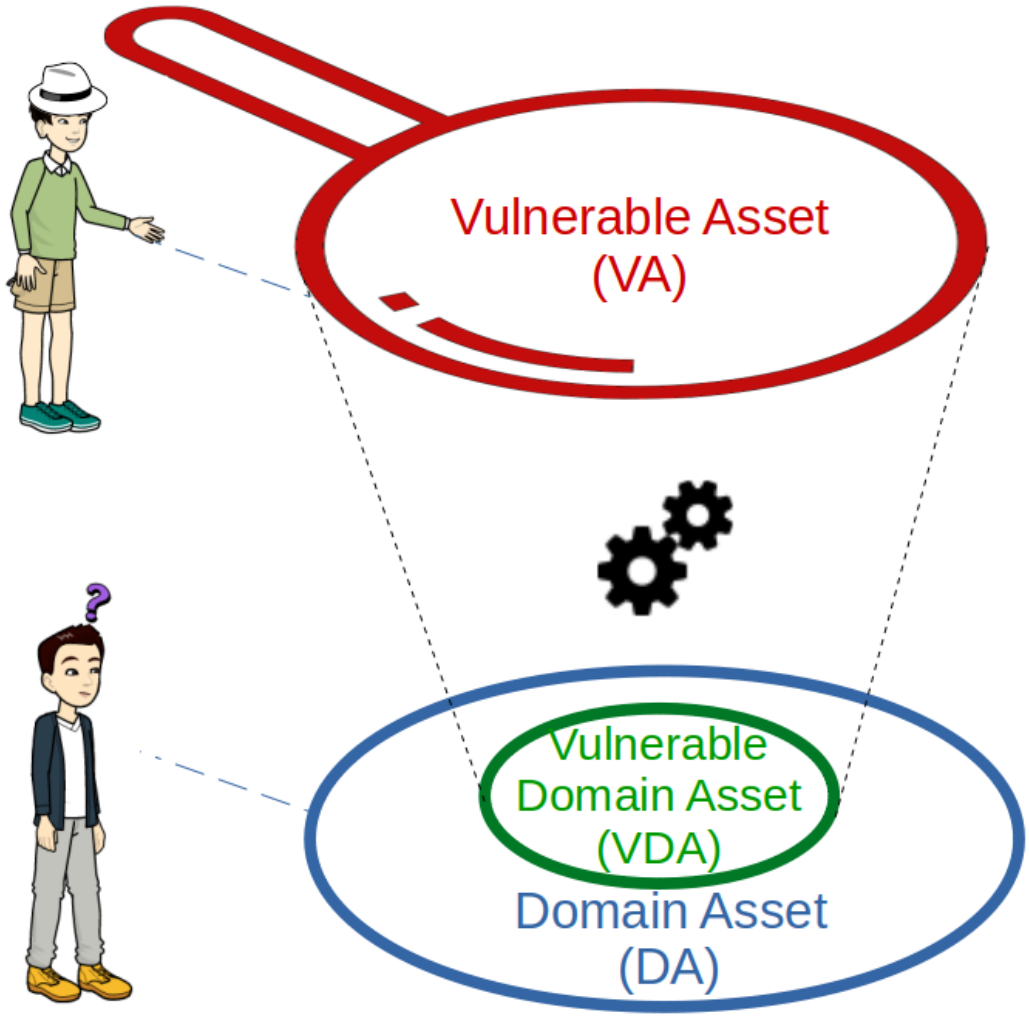


The security assistance knowledge background



12. **CAPEC** : capec.mitre.org/
13. **CWE** : cwe.mitre.org/
14. **CVE** : cve.mitre.org/
15. **CPE** : cpe.mitre.org/

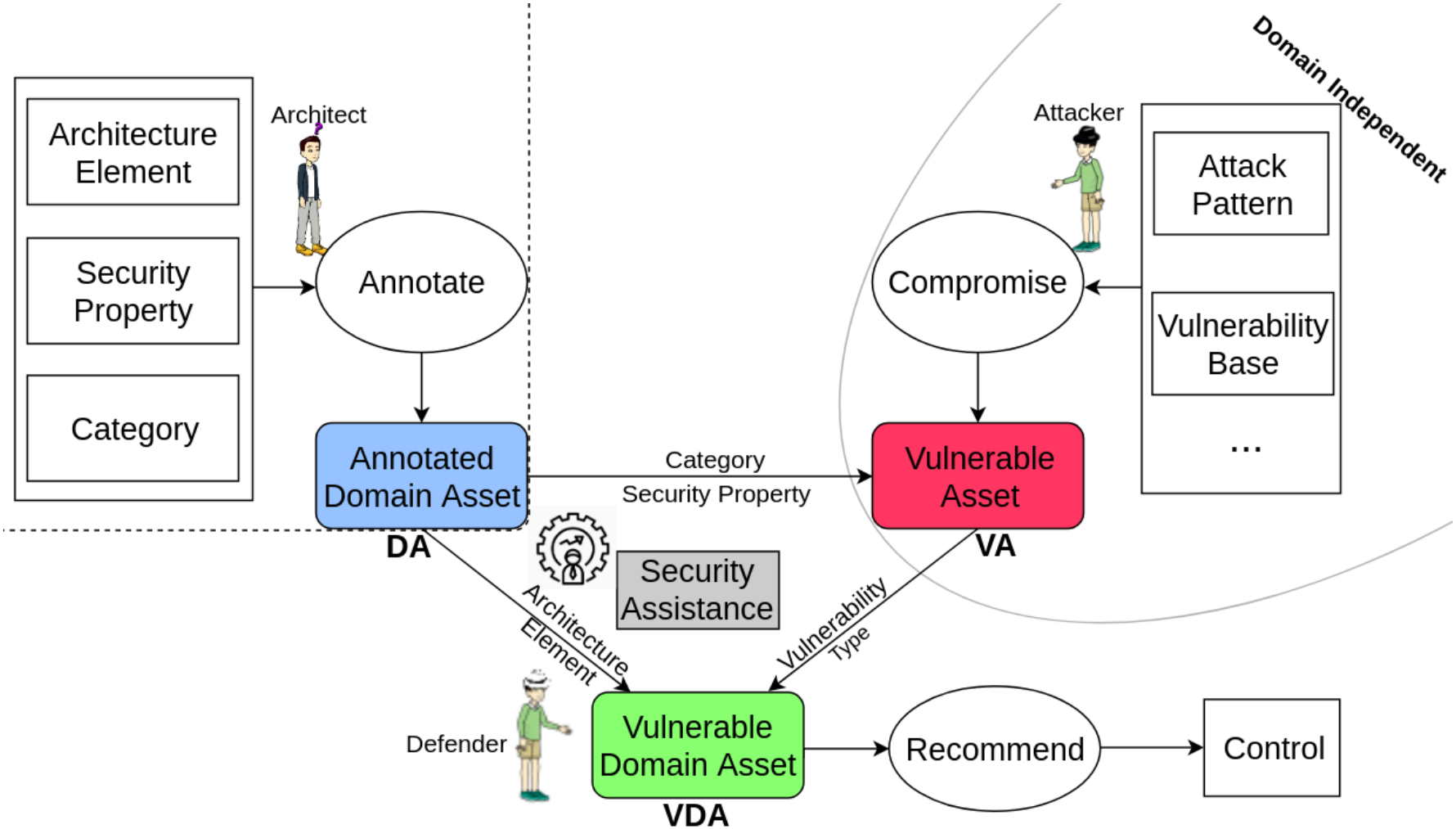
A novel refinement of "asset"



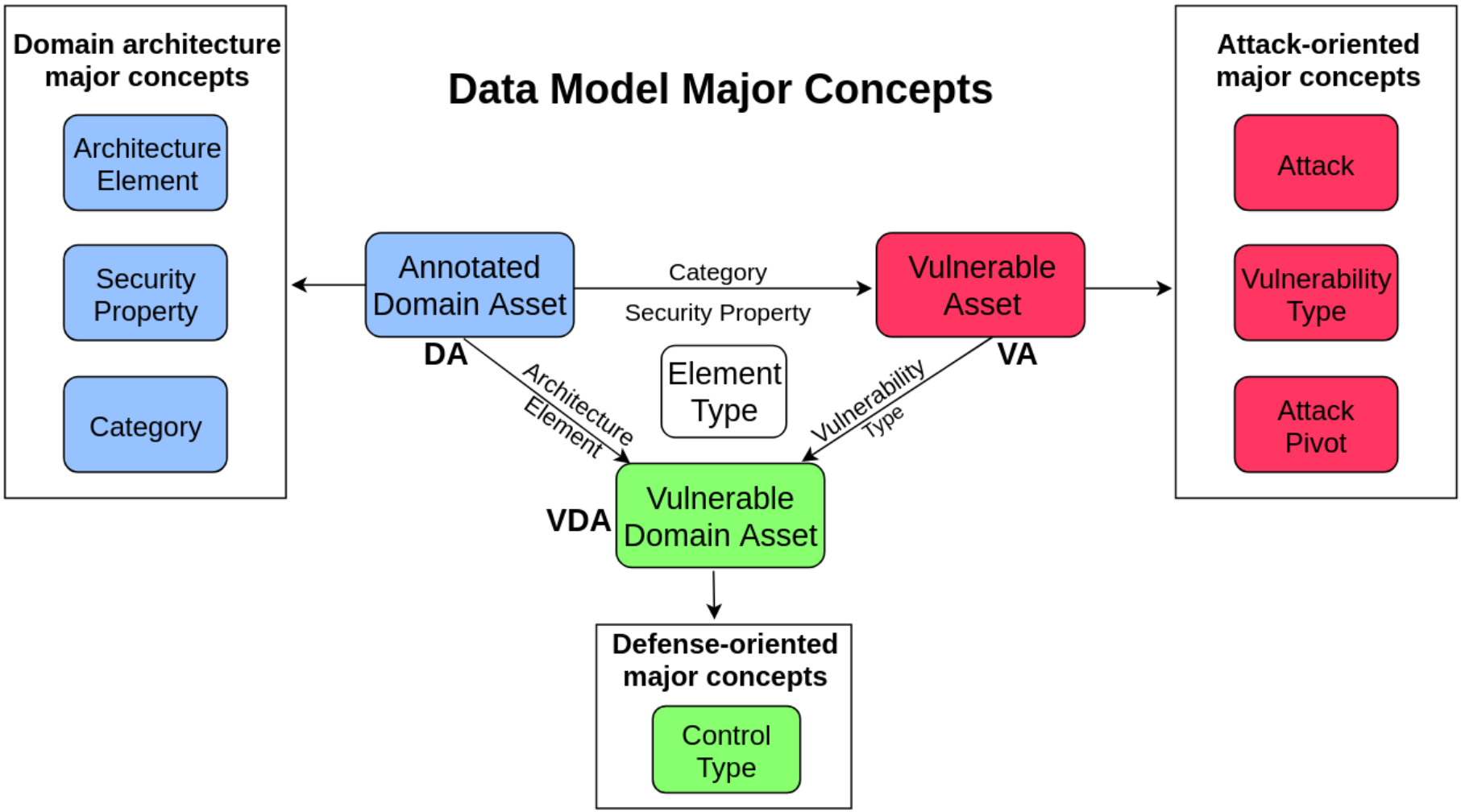
Asset	Definition
Domain Asset (**DA**)	Anything that has value for domain experts, towards the fulfilment of the function and goal of system, together with the assurance of its properties.
Vulnerable Asset (**VA**)	Anything that has value for security experts. It has vulnerabilities that can be menaced by threats.
Vulnerable Domain Asset (**VDA**)	Anything that has value for domain experts, but also has vulnerabilities that can be menaced by threats.

16. **Asset: anything that has value to an organization. (ISO27001).**

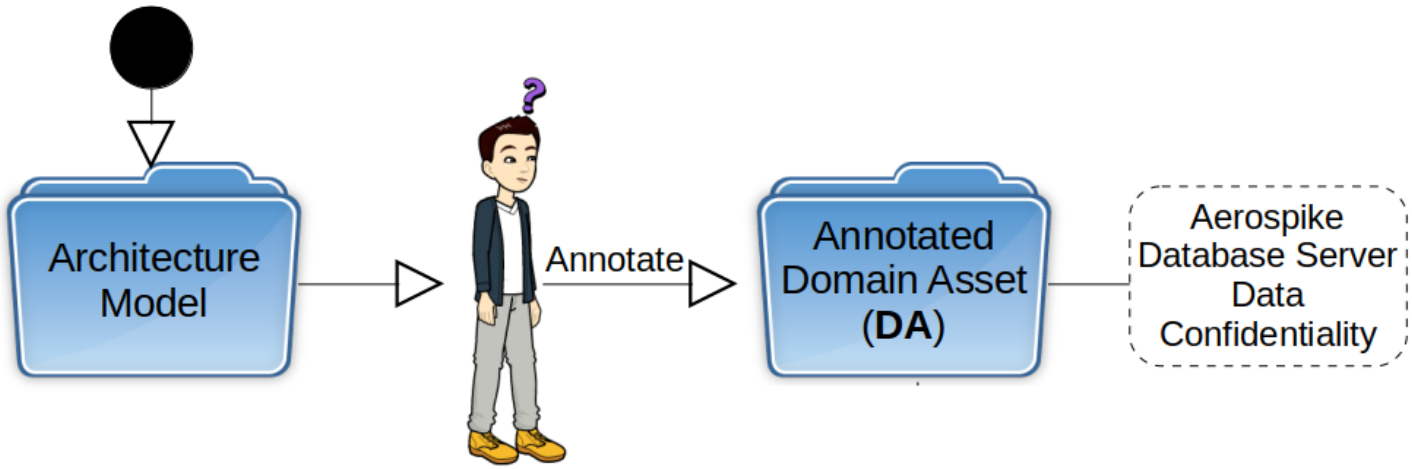
Asset-based 3-view security assistance framework



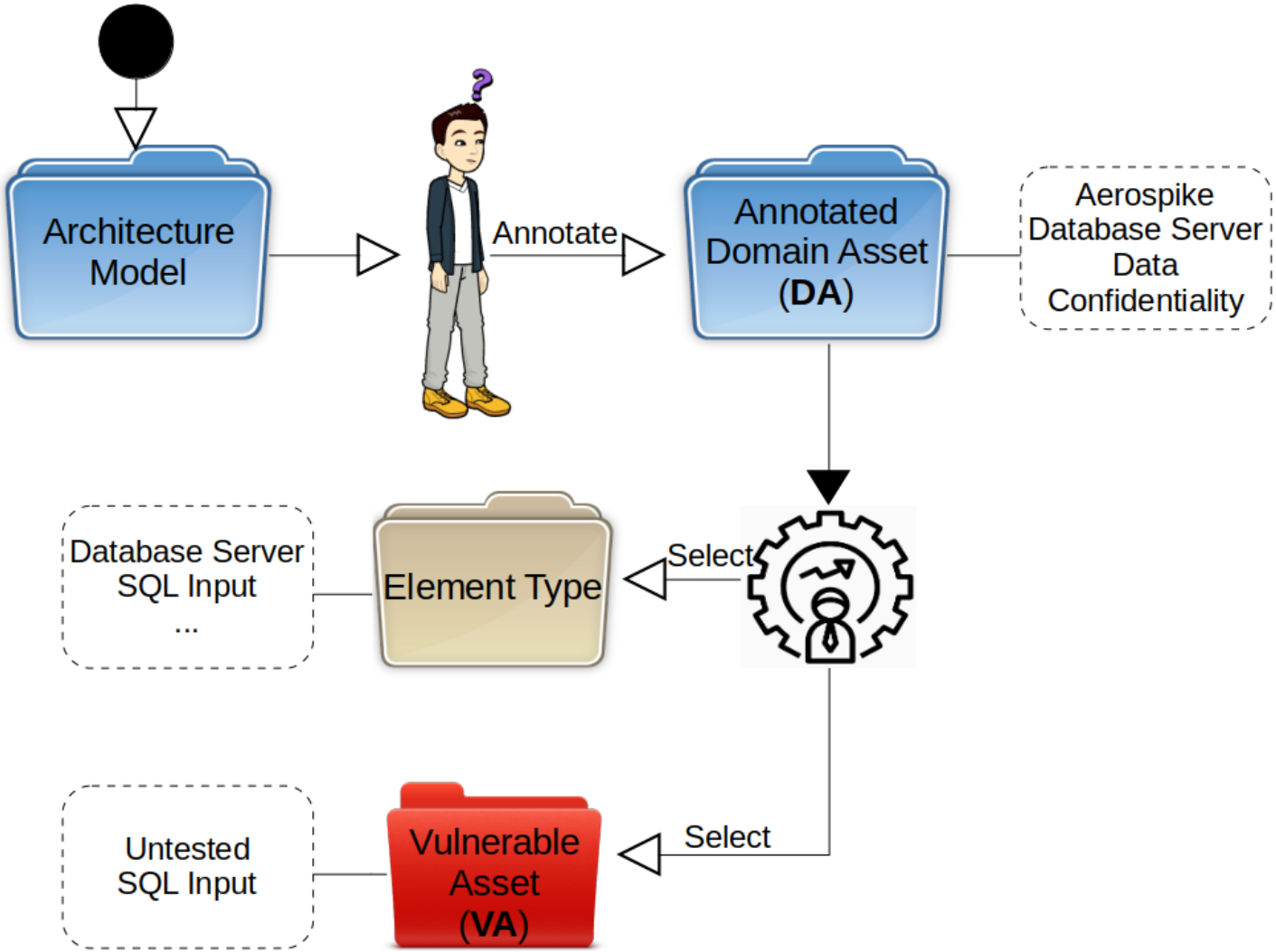
The assistance data model



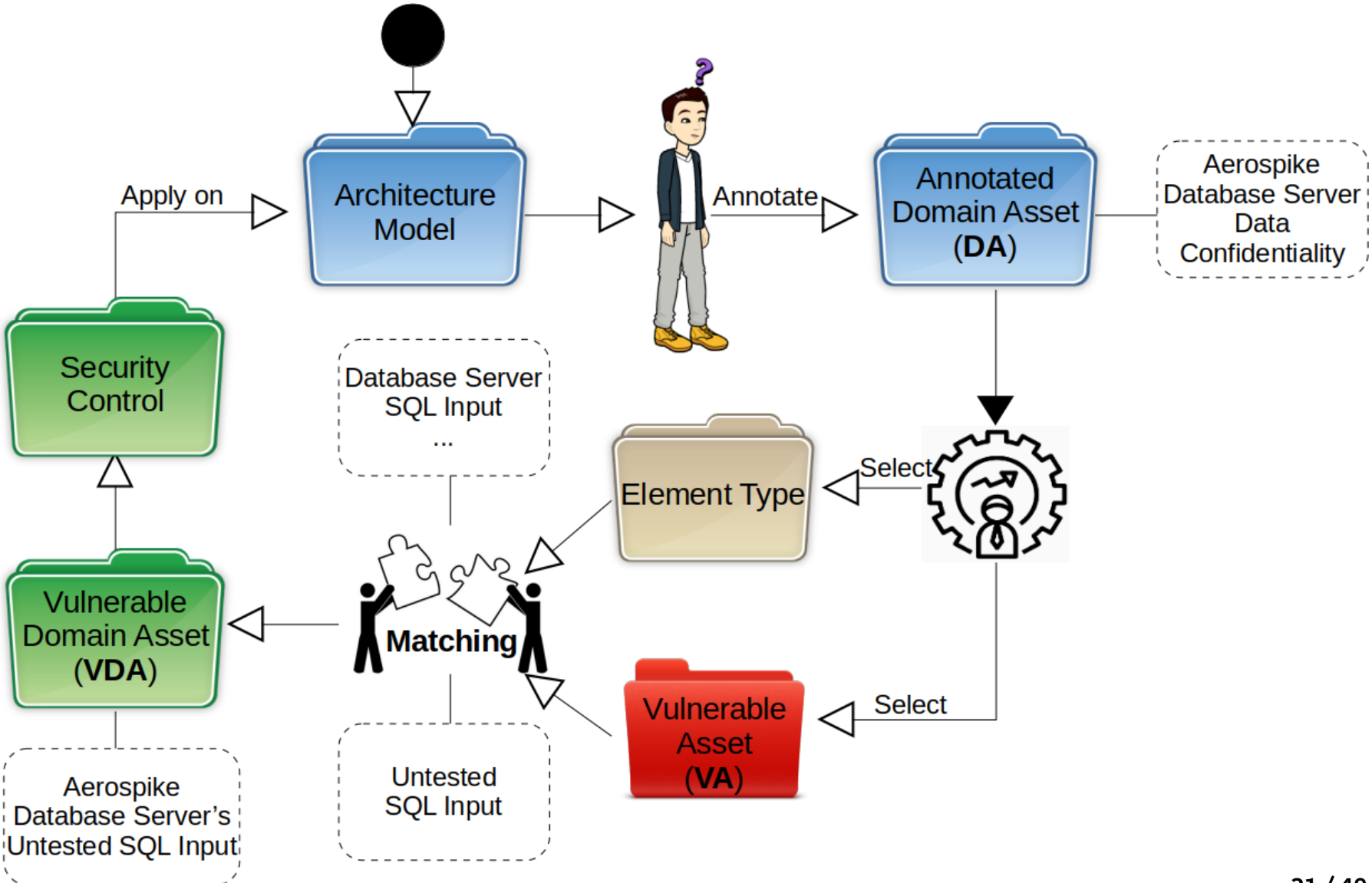
A BPMN-based process: major tasks



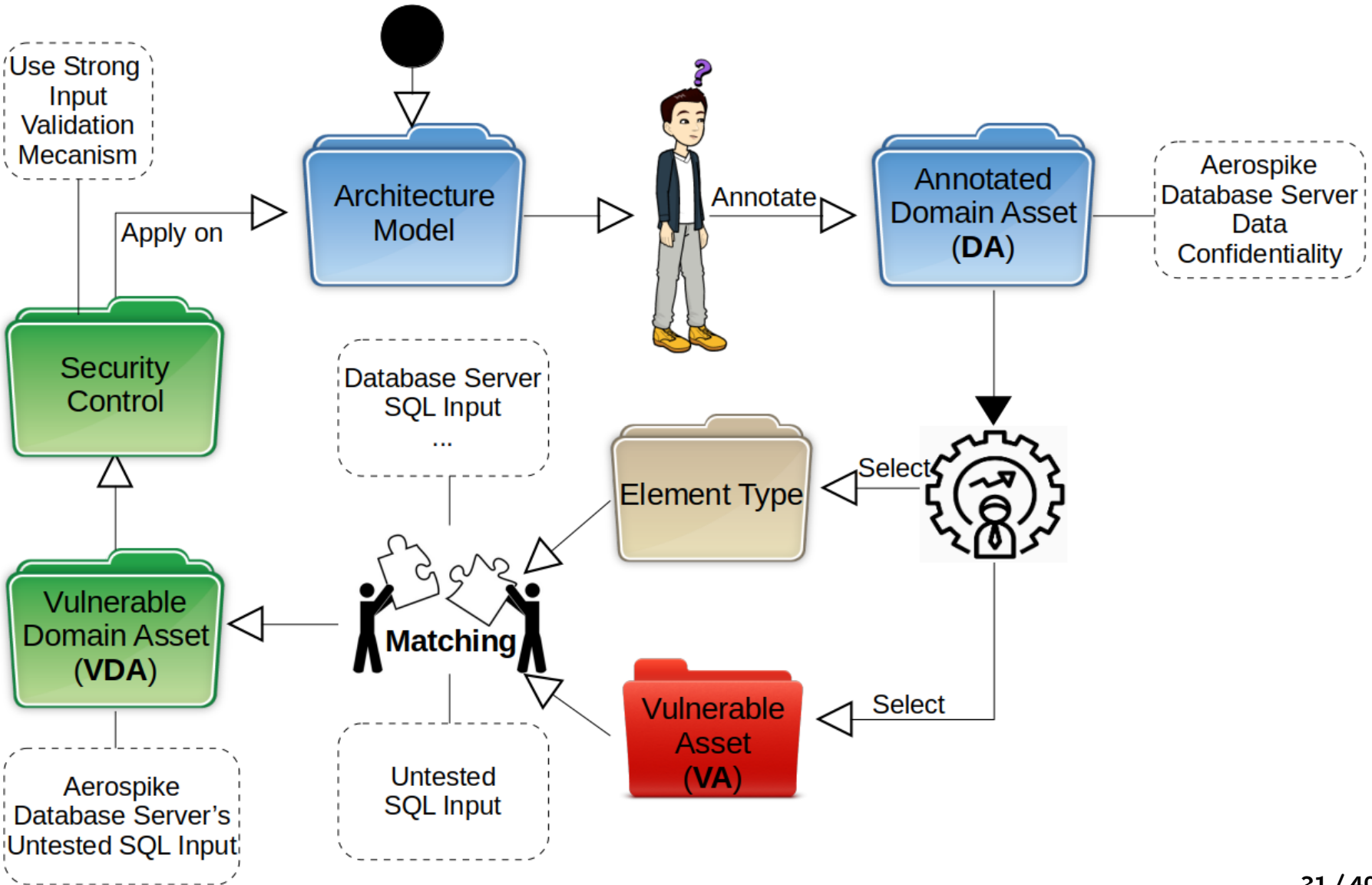
A BPMN-based process: major tasks



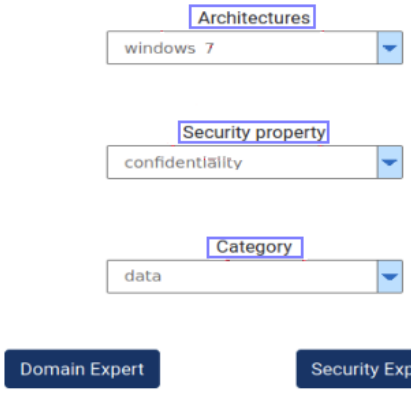
A BPMN-based process: major tasks



A BPMN-based process: major tasks



A web application assistance prototype tool for architects

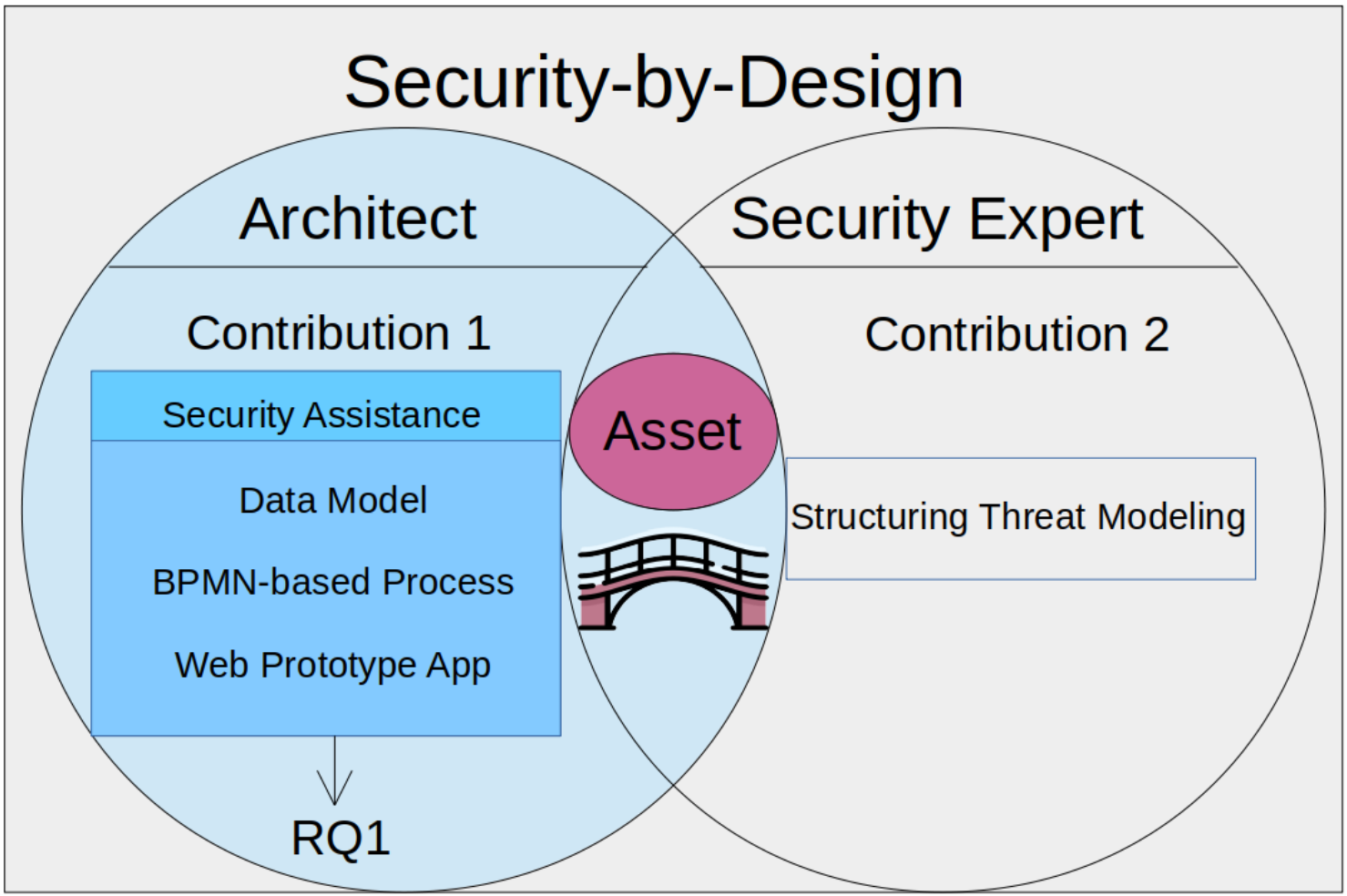


Results of architecture modeling

VDA	VULNERABILITY	CONTROL
sql_statement	improper_neutralization_of_special_elements_used_in_an_sql_command	• strong_input_validation • use_of_parameterized_queries_or_stored_procedures • use_of_custom_error_pages
	improper_neutralization_of_special_elements_in_output_used_by_a_downstream_component	• enforce_principle_of_least_privilege • harden_registry_server_and_file_access_permissions • implement_communications_to_and_from_the_registry_using_secure_protocols • strong_input_validation • use_of_parameterized_queries_or_stored_procedures • use_of_custom_error_pages
	improper_input_vallidation	• use_input_validation_before_writing_to_a_web_log • validate_all_log_data_before_it_is_output • strong_input_validation • use_of_parameterized_queries_or_stored_procedures • use_of_custom_error_pages
	incorrect_comparison	• strong_input_validation • use_of_parameterized_queries_or_stored_procedures • use_of_custom_error_pages
	improper_enforcement_of_message_or_data_structure	• strong_input_validation • use_of_parameterized_queries_or_stored_procedures • use_of_custom_error_pages

17. http ://share-irisa.univ-ubs.fr/abs4sos/

Contribution 1 global view

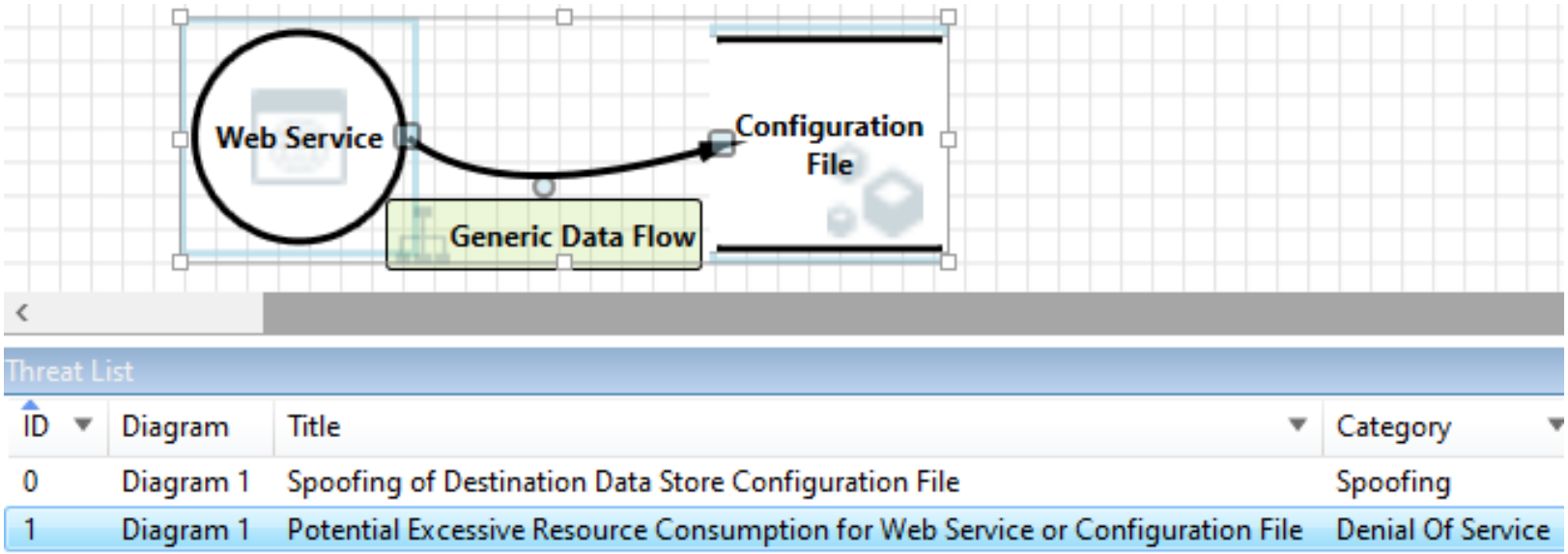


18. Messe Nan et al. "An Asset-Based Assistance for Secure by Design". In : *2020 27th Asia-Pacific Software Engineering Conference (APSEC)*. ***Core rank: B***.

1. Problem statement
2. Contribution 1. Security assistance
3. Contribution 2. Structuring threat modeling
   - Threat modeling in practice
   - Reference model
   - Asset identification process
   - Vulnerable Asset library
   - Asset identification process integrating with Microsoft illustration
4. Conclusion

Microsoft SDL threat modeling process

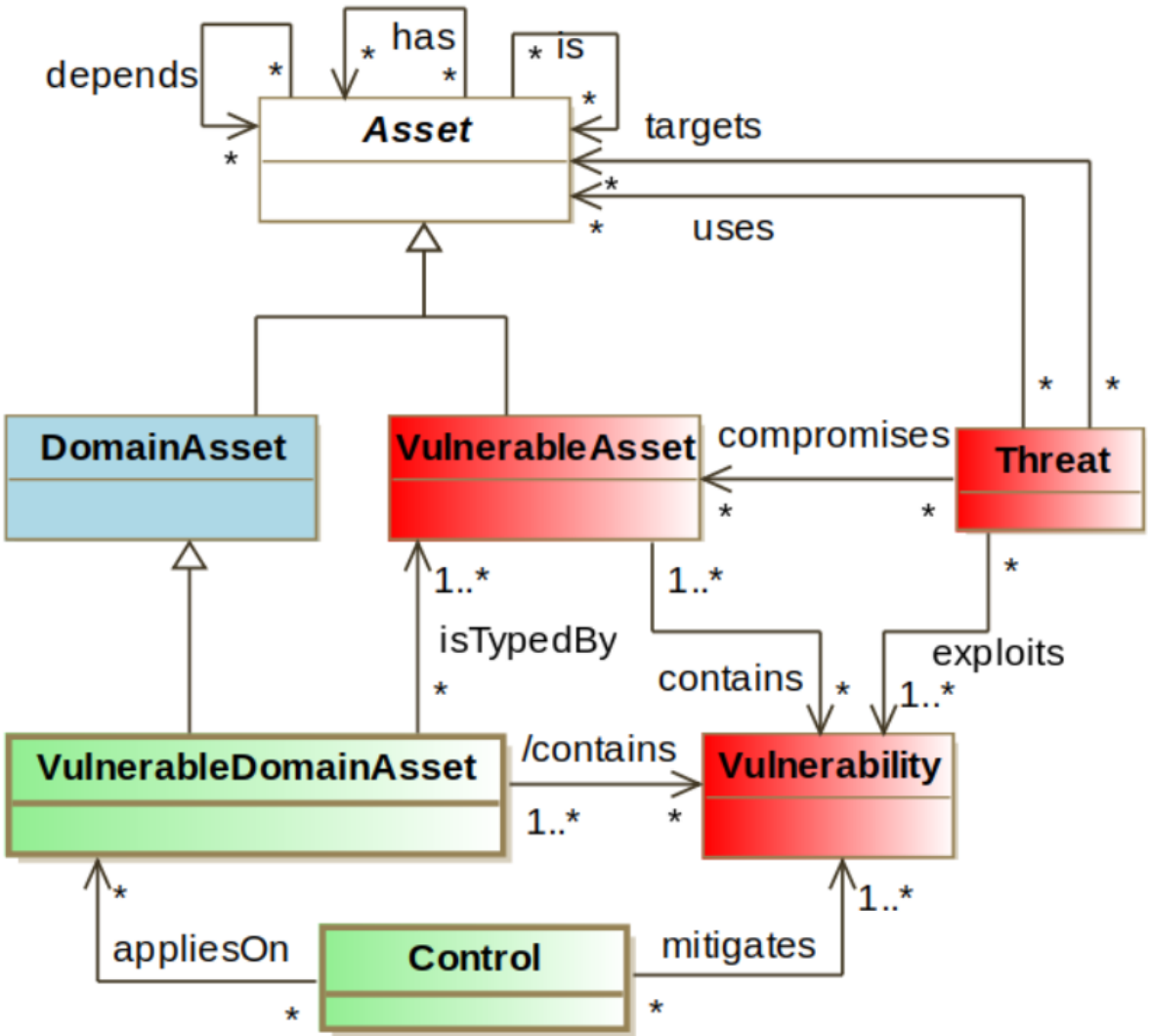
WebSphere Application Server Version 7.0 :



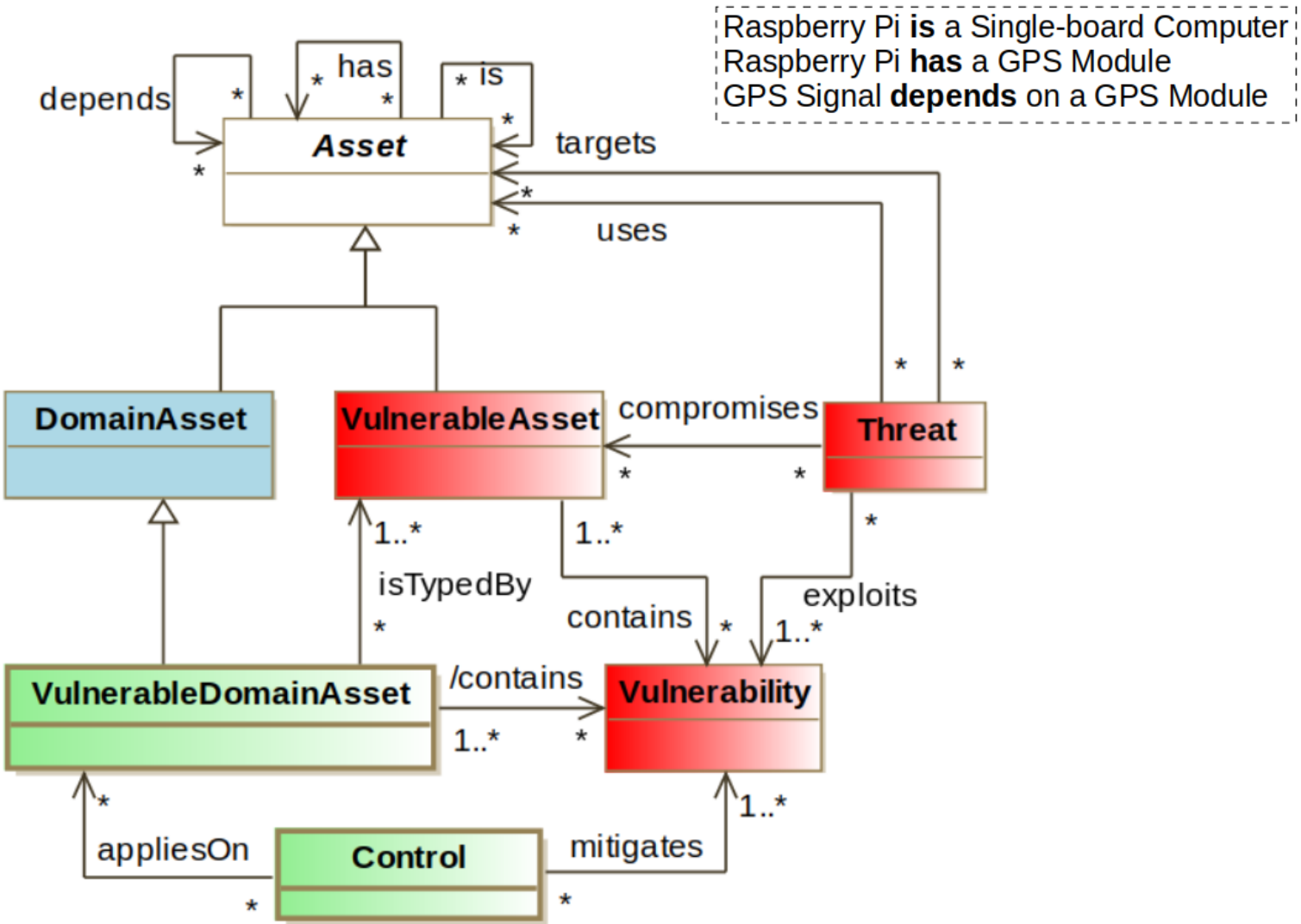
Microsoft SDL threat modeling tool

19. https ://www.microsoft.com/en-us/download/details.aspx ?id=49168

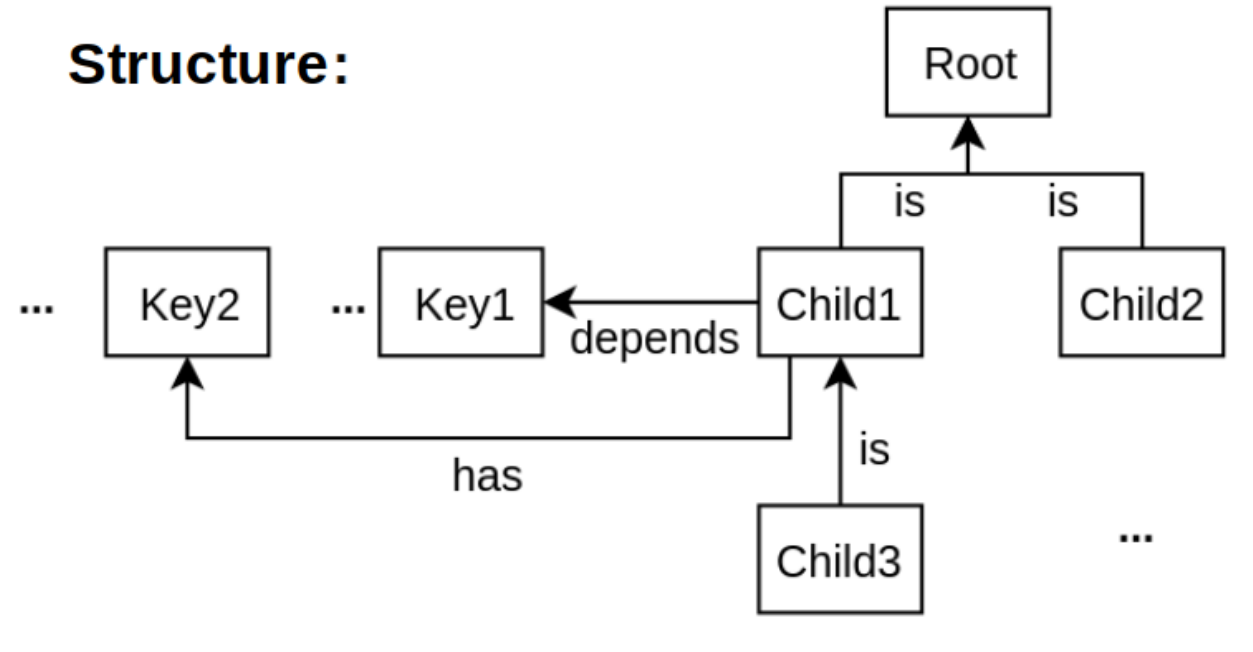
An asset-based reference model



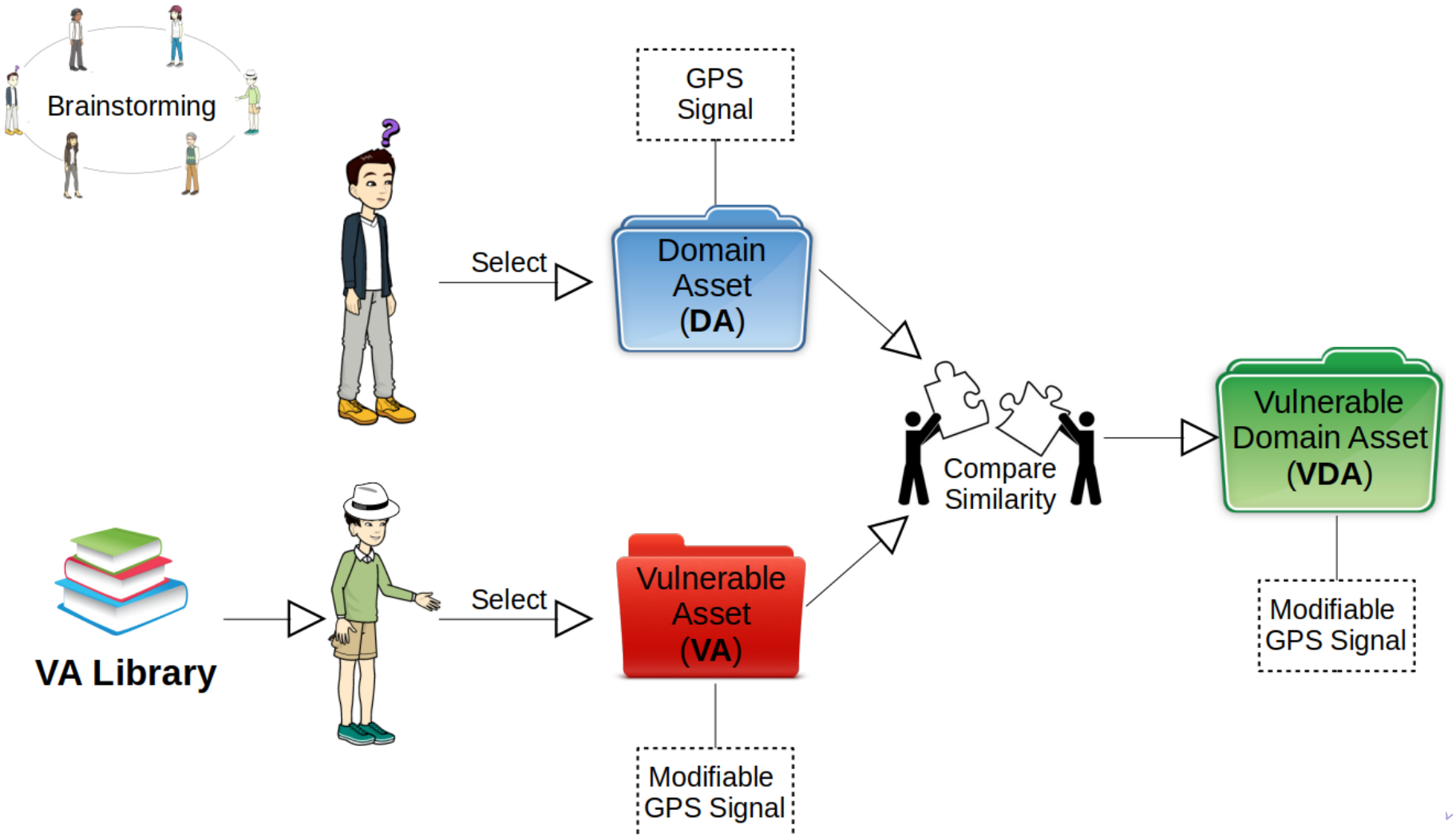
An asset-based reference model



The B-Tree structure



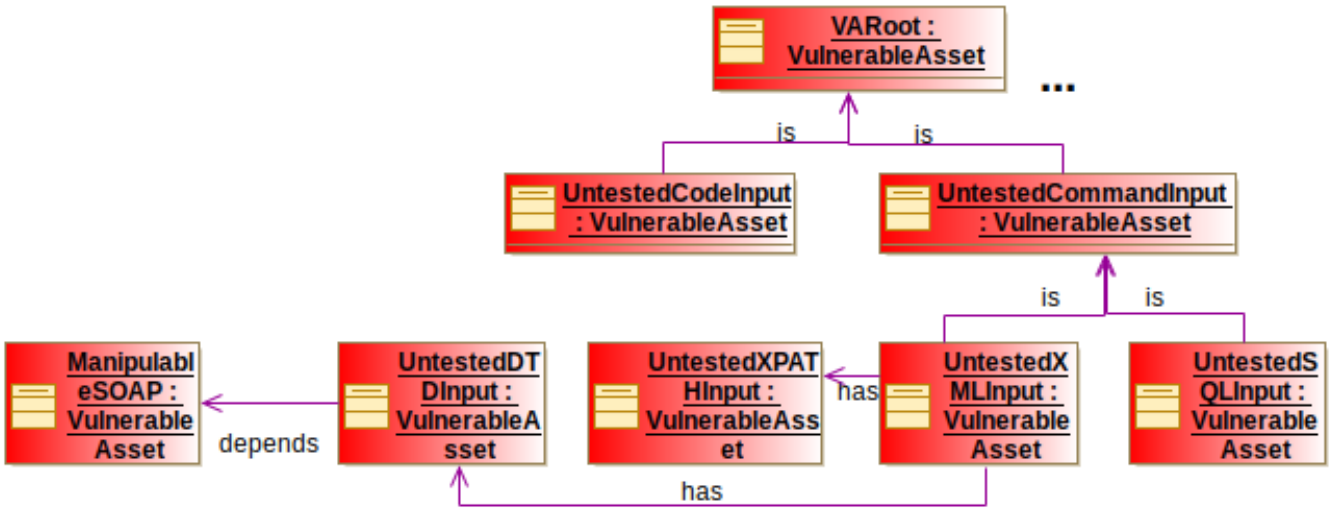
Asset identification process: major tasks

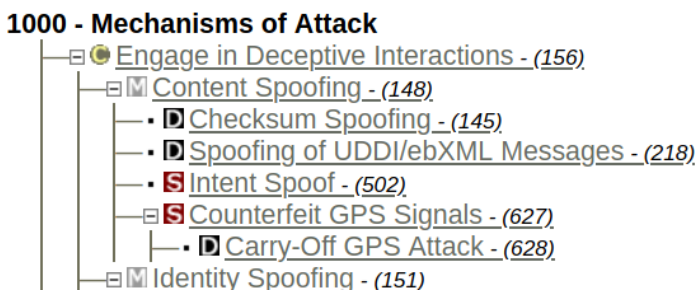


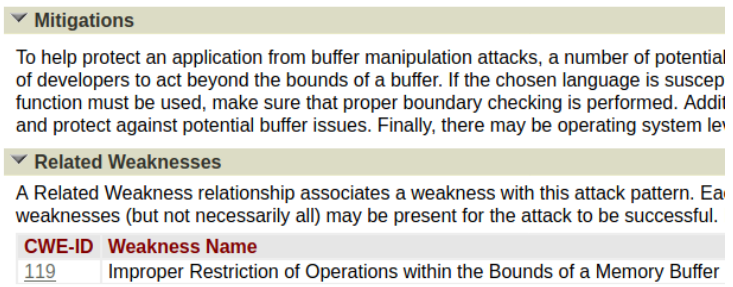
Building VA library

VA Library

Extraction of VA from CAPEC respecting B-Tree structure







20. https ://capec.mitre.org/

VA extraction rules

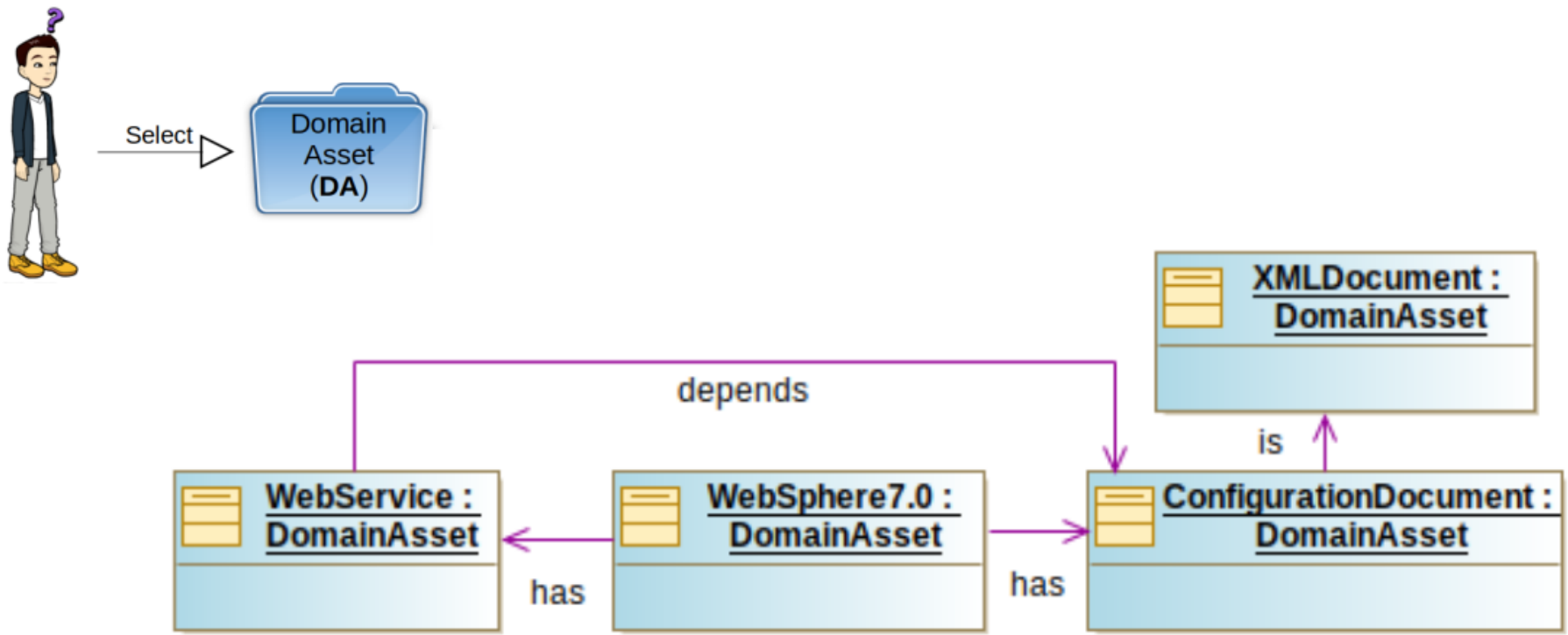
- **VA Rule**: '*Keyword*' $\rightarrow$ '**VA**'
  Rule example: **VA** + '*manipulation*' (Ex. '**Web service protocol** *manipulation*') ;

VA extraction rules

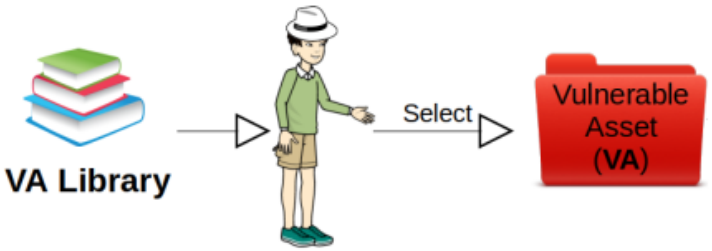
- **VA Rule**: '*Keyword*' $\rightarrow$ '**VA**'
  Rule example: **VA** + '*manipulation*' (Ex. '**Web service protocol** *manipulation*') ;
- **Relation Rule**: '*childOf*' | '*canFollow*' $\rightarrow$ '*is*' | '*has*' | '*depends*'
  Rule example: '**SOAP** manipulation' *is* a *childOf* '**web services protocol** manipulation'

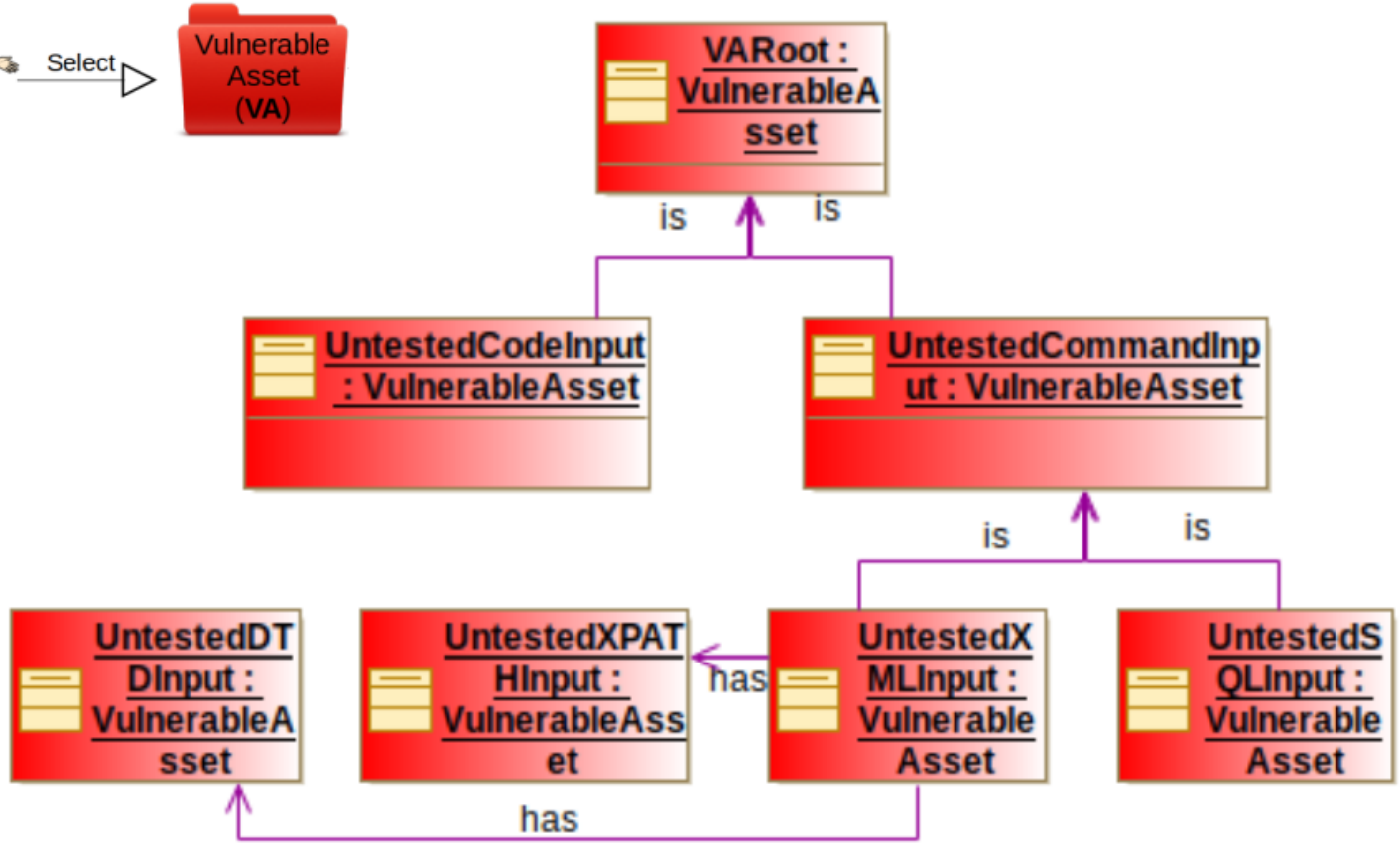
Asset identification process illustration: I. Domain Asset (DA)

WebSphere Application Server Version 7.0:

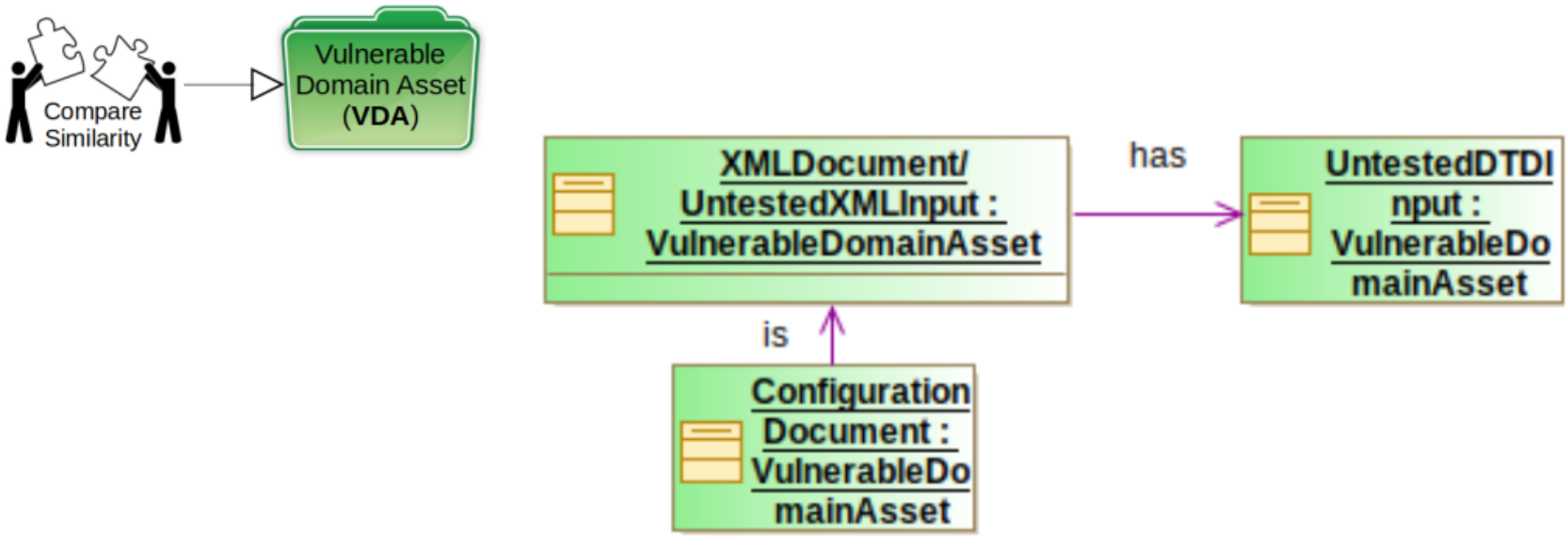


II. Vulnerable Asset (VA)





III. Vulnerable Domain Asset (VDA)



Result: 14 threats found
(*XML Schema Poisoning, XML Ping of the Death, XML Entity Expansion, XML Entity Linking, Spoofing of UDDI/ebXML Messages, XML Routing Detour Attacks, XML External Entities Blowup, XML Attribute Blowup, XML Nested Payloads, XML Oversized Payloads, XML Injection, XML Quadratic Expansion, XML Flood, DTD Injection*).

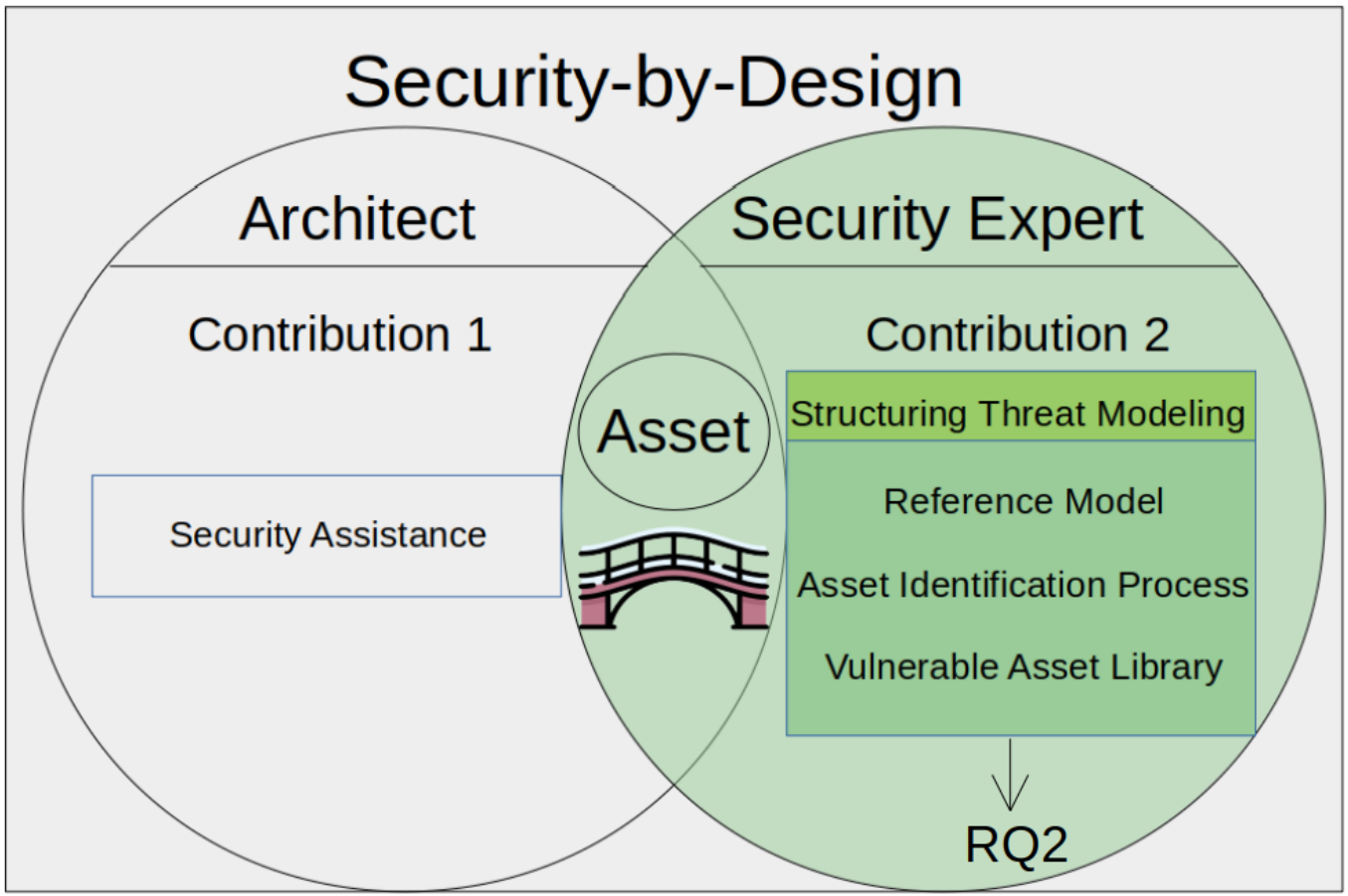
A reusable BASH prototype for security experts

KEYWORD	VULNERABILITY_ASSET	CAPEC_ID
Manipulate	Registry Information	203
Manipulate	Human Behavior	416
Manipulate	Timing and State	172
Manipulate	Data Structures	255
Manipulate	System Resources	262
Leveraging	Race Conditions	26
Leveraging	Race Conditions via Symbolic Links	27

Figure – An excerpt of BASH application result

21. https://github.com/lunanan/ArchwareExtraction

Contribution 2 global view



22. Messe Nan et al. "Asset-Oriented Threat Modeling". In : *2020 19th IEEE International Conference On Trust, Security And Privacy In Computing And Communications (TrustCom).* ***Core rank: A en 2020***.

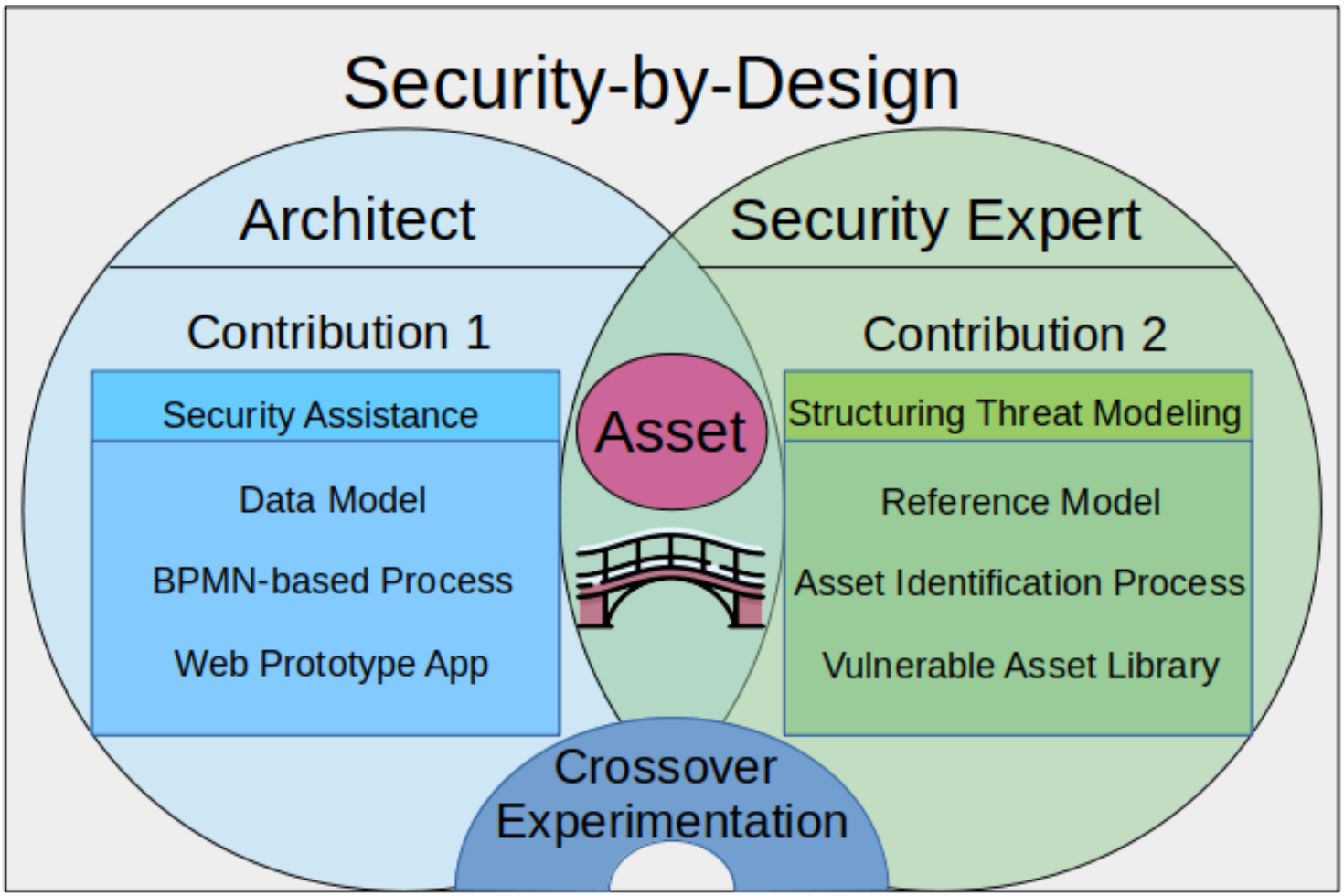
1 Problem statement

2 Contribution 1. Security assistance

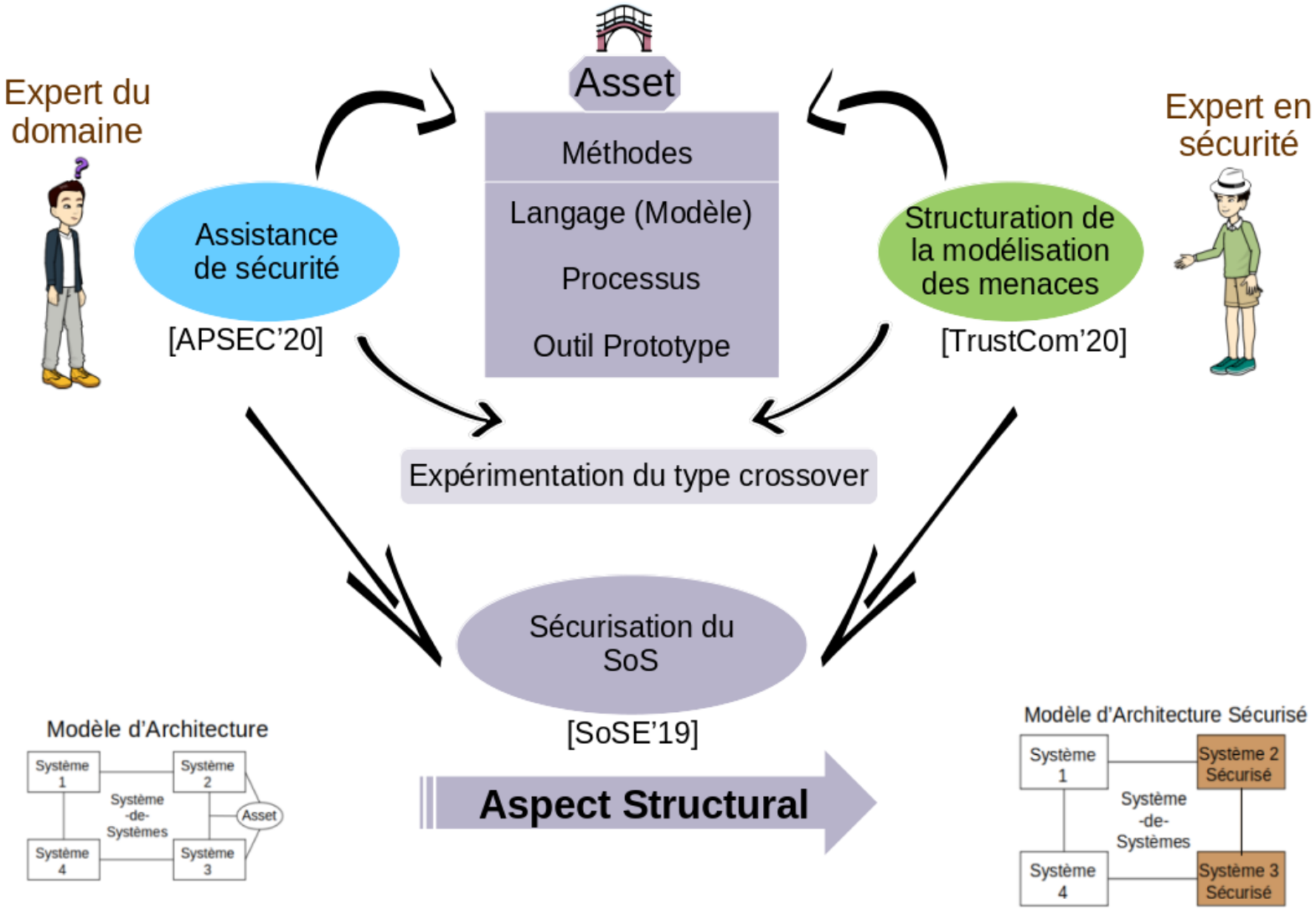
3 Contribution 2. Structuring threat modeling

4 Conclusion
- Contribution
- Perspective

Bridging the gap between architects and security experts



High level view



Perspectives – In the short term (ongoing works)

Contribution 2 Structuring threat modeling	VA library	Automating VA extraction (GATE)
	Similarity comparison	Semantic matching (S-match)
	Evaluation	Applying in cloud domain (DiverSE)
Asset		Asset interdependence (Tampere University)

Table – Ongoing works

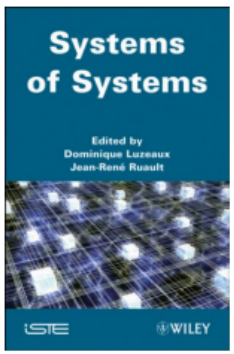
Perspectives – In the long term

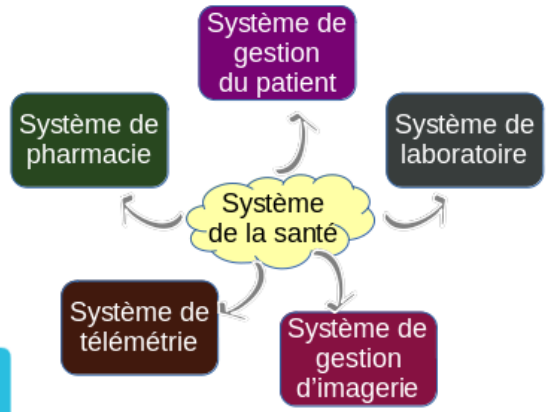
Contribution 1 Security assistance	Concept extension	Adding *Risk* concept
		Extending concept *SecurityProperty*
	Element type formalization	Investigating alternatives to CPE (typing mechanism)
		Developing a modeling phase tool suite (SysML profil)
Contribution 2 Structuring threat modeling	VA library	Including organizational and human aspect
Asset		Coupling with Attack Tree

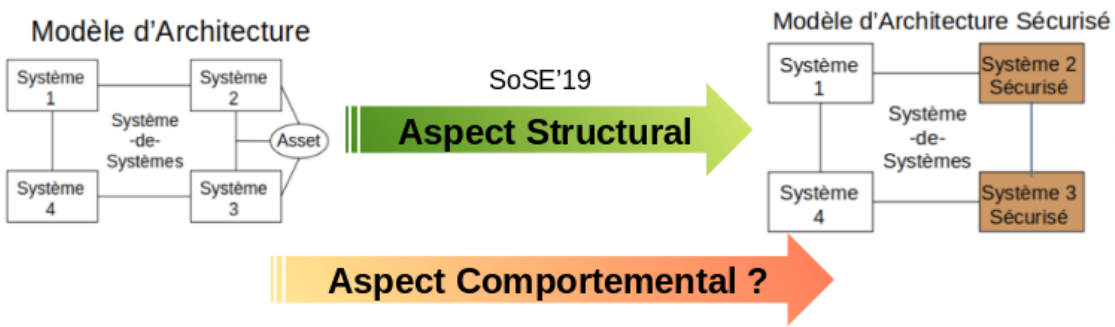
Table – Perspectives in the long term

23. Christopher Schmitz, André Sekulla et Sebastian Pape. "Asset-centric analysis and visualisation of attack trees". In : *Graphical Models for Security - 7th International Workshop, GraMSec@CSF 2020, Boston, MA, USA, Virtual Conference, June 22, 2020, Revised Selected Papers*. T. 12419. LNCS. Springer, nov. 2020, p. 45-64. doi : 10.1007/978-3-030-62230-5_3. url : https://link.springer.com/chapter/10.1007%2F978-3-030-62230-5_3.

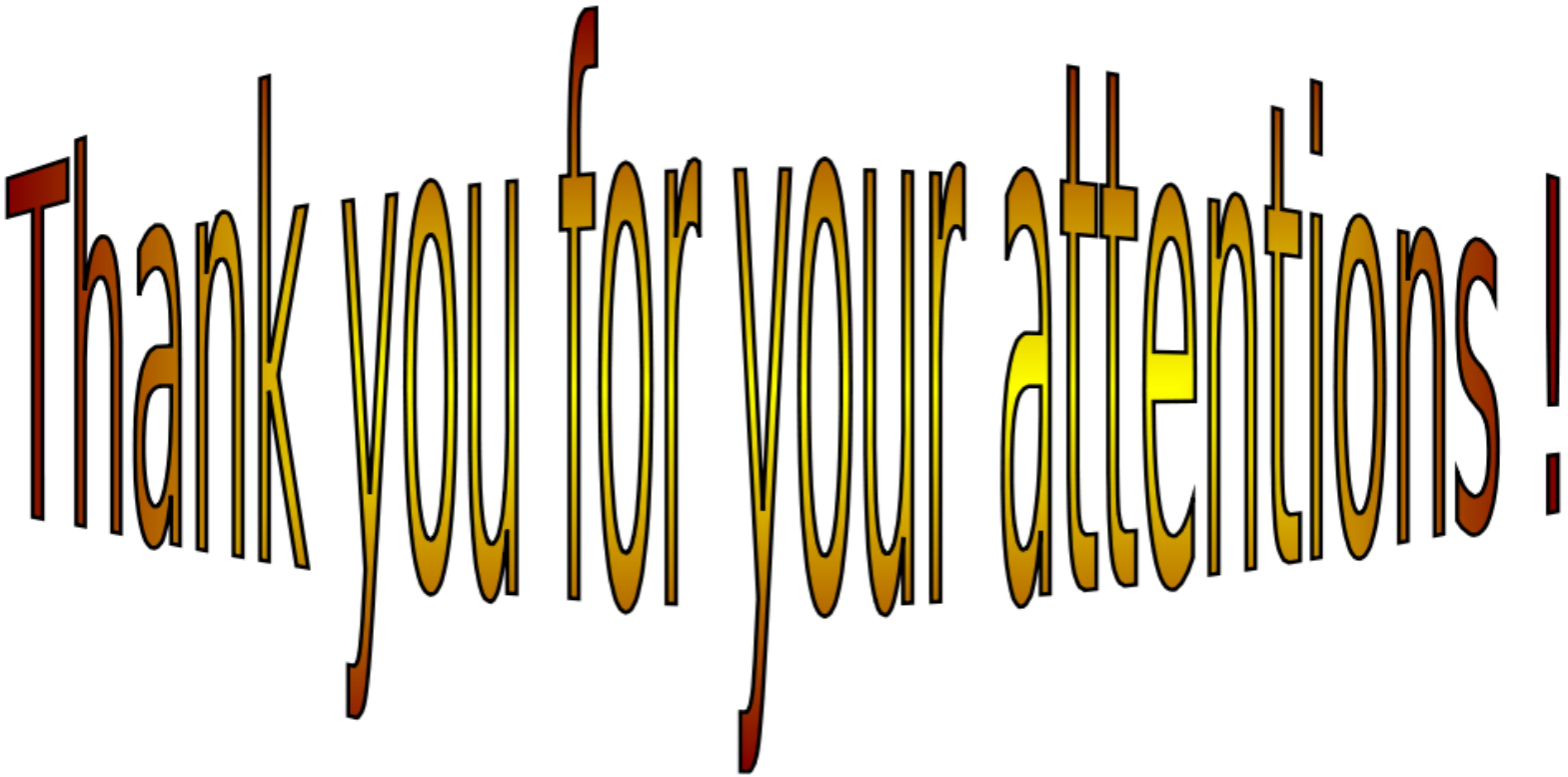
Systems-of-systems







Thank you for your attention



Major publications

1. Nan Messe, Vanea Chiprianov, Nicolas Belloir, Jamal El-Hachem, Régis Fleurquin and Salah Sadou. Asset-oriented threat modeling. 2020 19th IEEE International Conference On Trust, Security And Privacy In Computing And Communications (**TrustCom**), 2020. **Core rank : A**
2. Nan Messe, Nicolas Belloir, Vanea Chiprianov, Jamal El-Hachem, Régis Fleurquin and Salah Sadou. An asset-based assistance for secure by design. 2020 27th Asia-Pacific Software Engineering Conference (**APSEC**), 2020. **Core rank : B**
3. Nan Messe, Nicolas Belloir, Vanea Chiprianov, Imane Cherfa, Régis Fleurquin, and Salah Sadou. Development of secure system of systems needing a rapid deployment. In 14th Annual Conference System of Systems Engineering, **SoSE** 2019, Anchorage, AK, USA, May 19-22, 2019, pages 152–157. **IEEE**, 2019

Annexe – Collaboration I

- Collaboration avec Gurvan Le Guernic (DiverSE, DGA)
  - Sujet : Une ontologie pour les services cryptographiques
  - Domaines de recherche : cryptographie, ontologie, méta-modèle
  - Taux d'avancement : 50% (un papier est en cours de rédaction avec la soumission prévue pour CSF, 2021 ; un stage dans le cadre de ce travail commence le 1er avril)
- Collaboration avec Stéphanie Challita (DiverSE) et Olivier Barais (DiverSE)
  - Sujet : Un méta-modèle ou une ontologie des bonnes pratiques pour assurer la sécurité du Cloud
  - Domaines de recherche : Cloud, sécurité, ontologie, méta-modèle
  - Taux d'avancement : 20% (un stage dans le cadre de ce travail commence 1er mai)

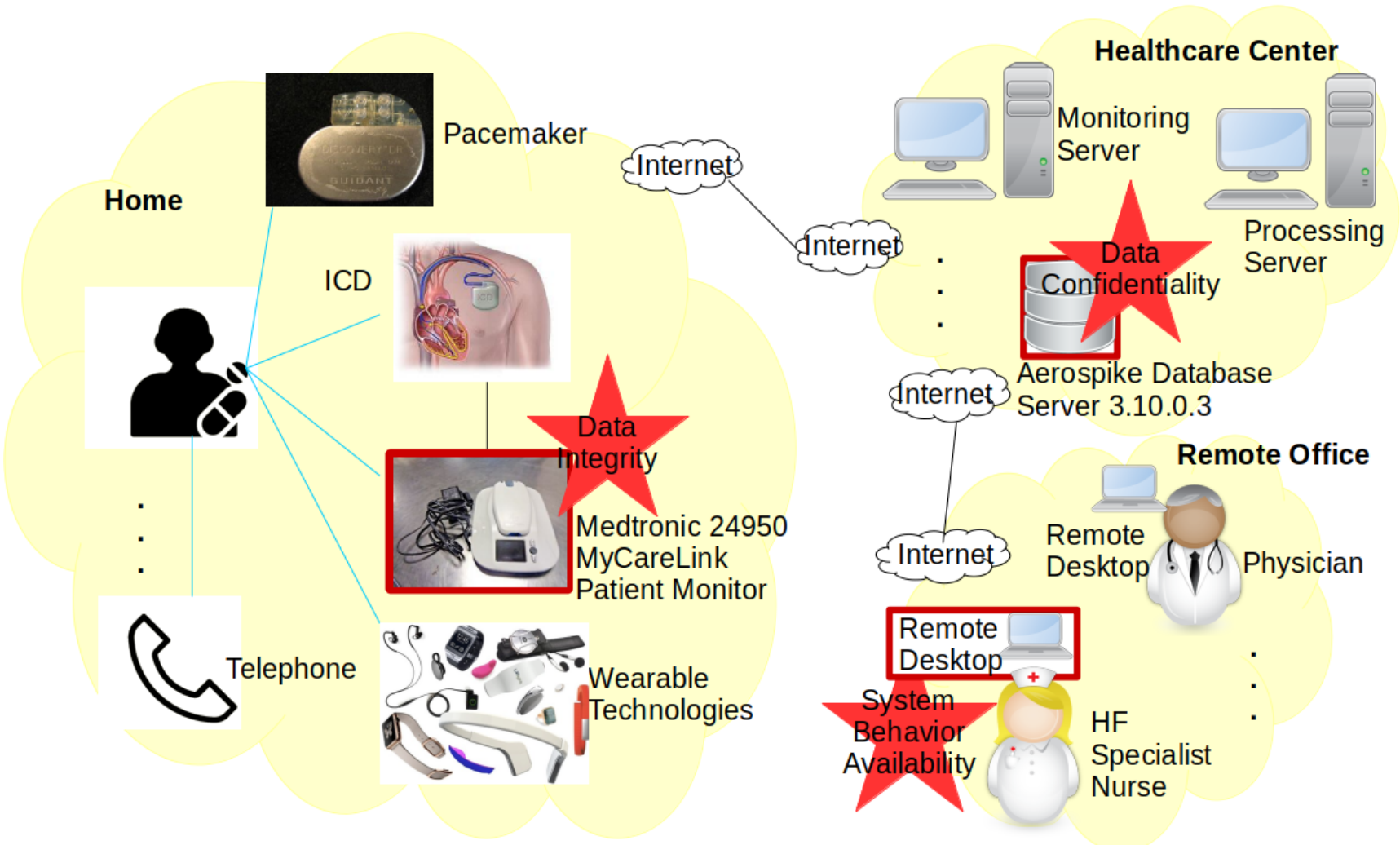
Annexe – Collaboration II

- Collaboration avec Sophie Pinchinat (LogicA), Didier Vojtisek (DiverSE, LogicA), Laurent Collet (*Startup* Sya, Digital) et William Ragot (*Startup* Sya, Digital)
  - Projet : ATSyRA V2
  - Domaines de recherche : analyse des risques, système basé sur des *assets*
  - Description : AYSyRA V1 est un outil qui permet de définir à la fois le modèle de domaine et l'arbre d'attaque utilisé pour analyser les menaces de sécurité potentielles sur ce modèle. Avec le *startup* Sya, on travaille sur ATSyRA V2, qui intègre la partie de la vulnérabilité impliquée dans l'arbre d'attaque. Ma thèse permet de fournir cette partie d'identification des vulnérabilités.
  - Taux d'avancement : 30% (une bourse Bingo initiée par la DGA est en cours de demande pour réaliser ce projet)

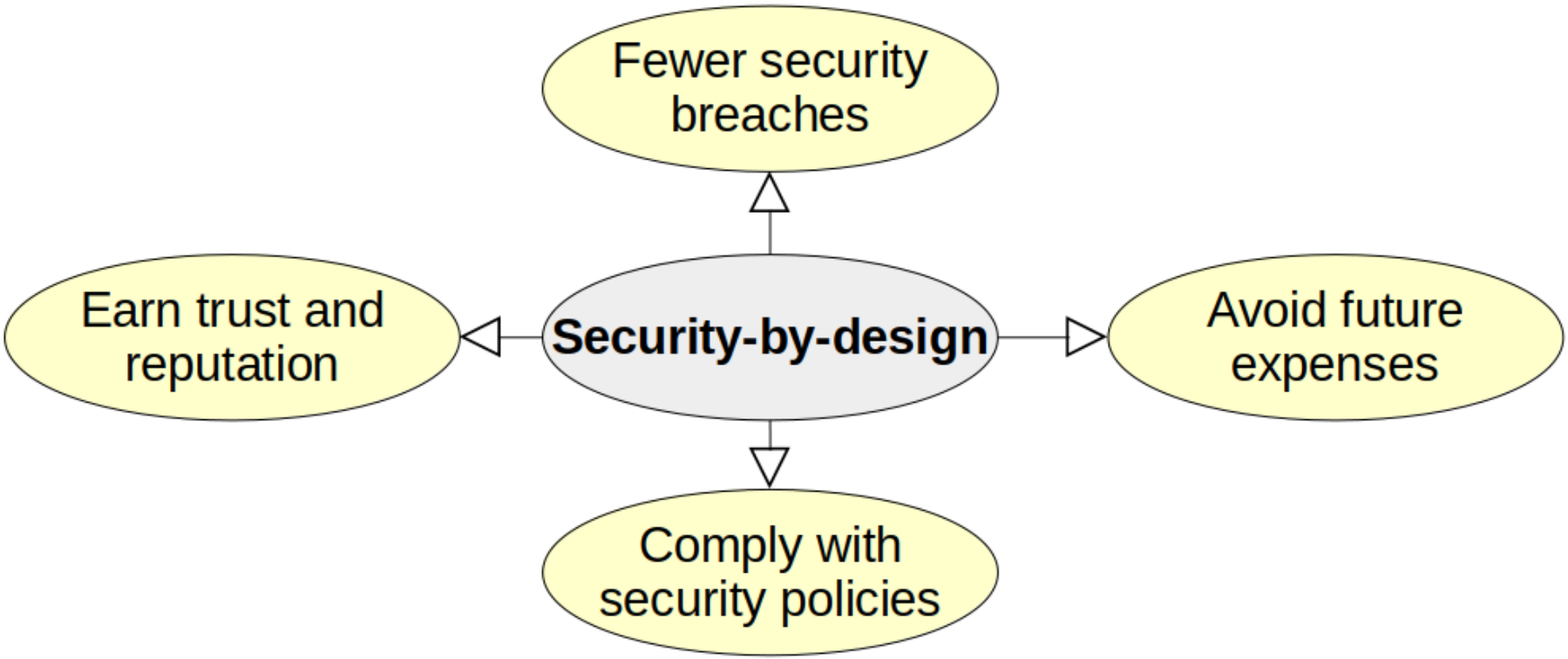
Annexe – Collaboration III

- Collaboration avec Benoît Combemale (DiverSE), Salah Sadou (ArchWare) et Raounak Benabidallah (DiverSE)
  - Sujet : Proposer des indicateurs de sécurité pour aider les développeurs à prendre des décisions
  - Domaines de recherche : métriques de sécurité, vulnérabilité, projets open-sources
  - Taux d'avancement : 5%
- Collaboration avec Eric Coatanea (**Université de Tampere, Finlande**), Salah Sadou et Raounak Benabidallah
  - Sujet : Identification des contradictions entre les propriétés de sécurité dans le modèle d'architecture
  - Domaines de recherche : cybersécurité, modélisation, graphe de causalité, TRIZ
  - Taux d'avancement : 10%
- Participation à SLIMFAST projet de l'équipe DiverSE (intervention 5%)

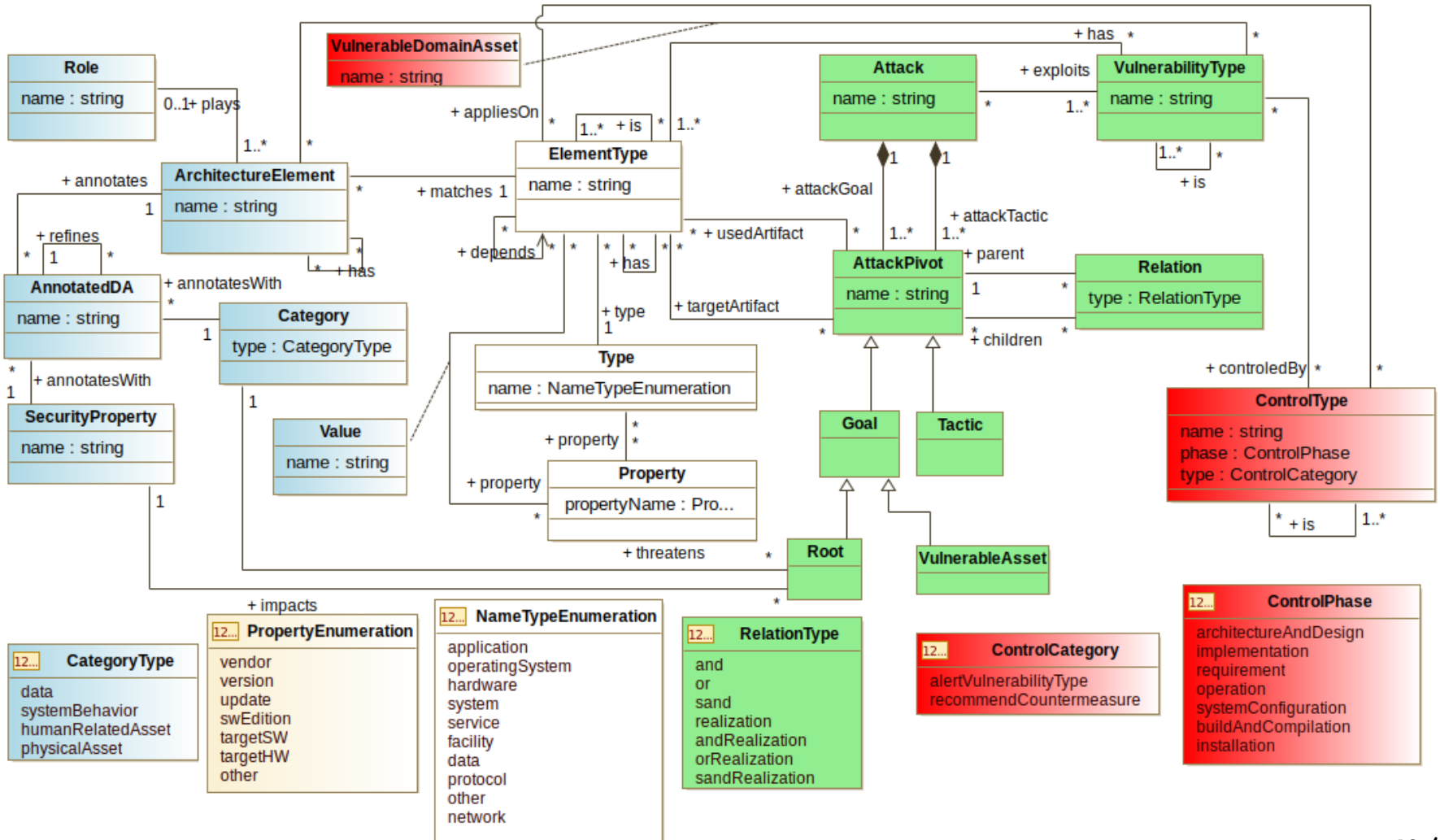
Security-by-design



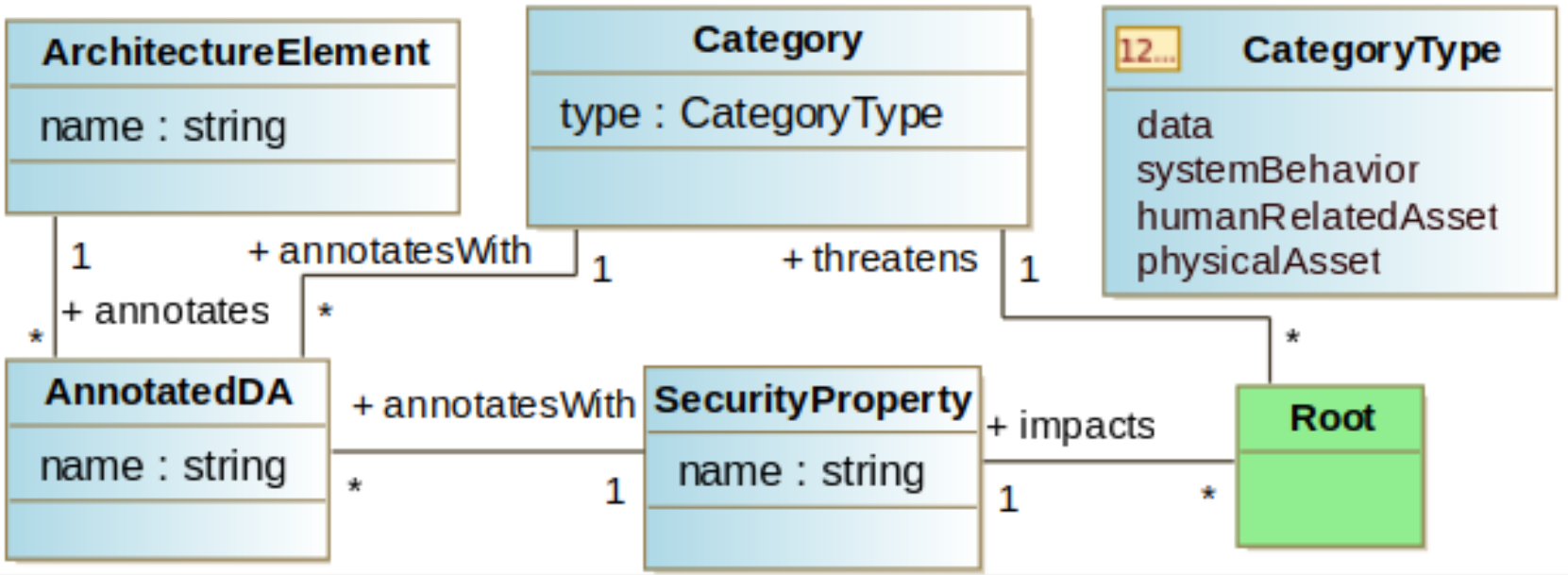
Security-by-design



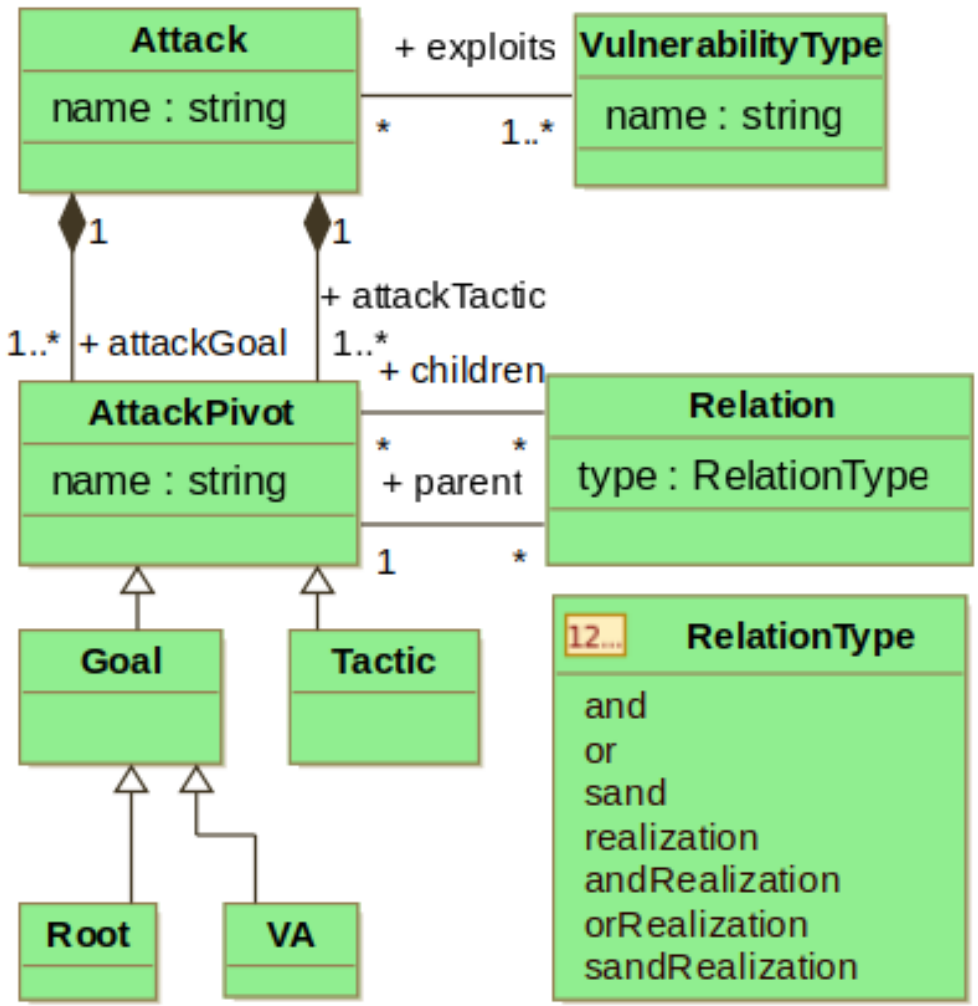
Appendices – Security assistance data model



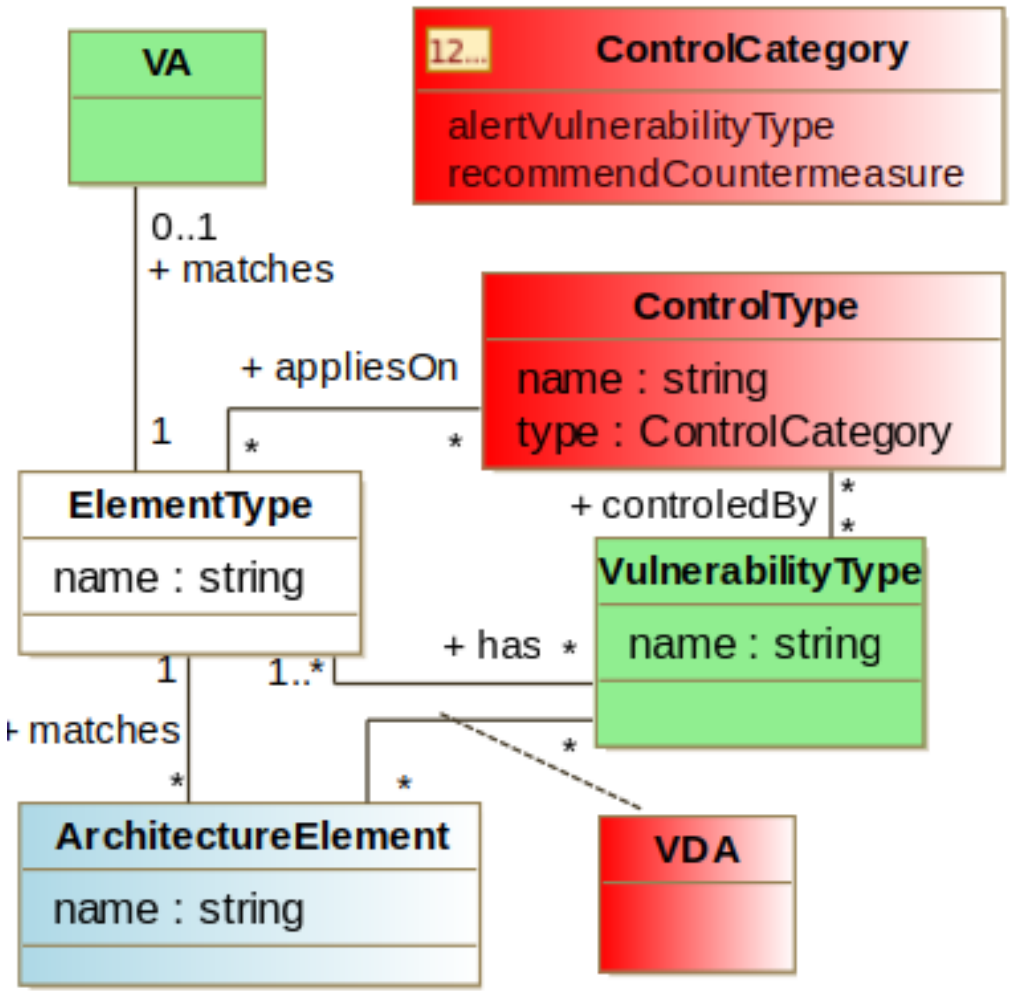
Appendices – The assistance data model : domain architecture specific aspects



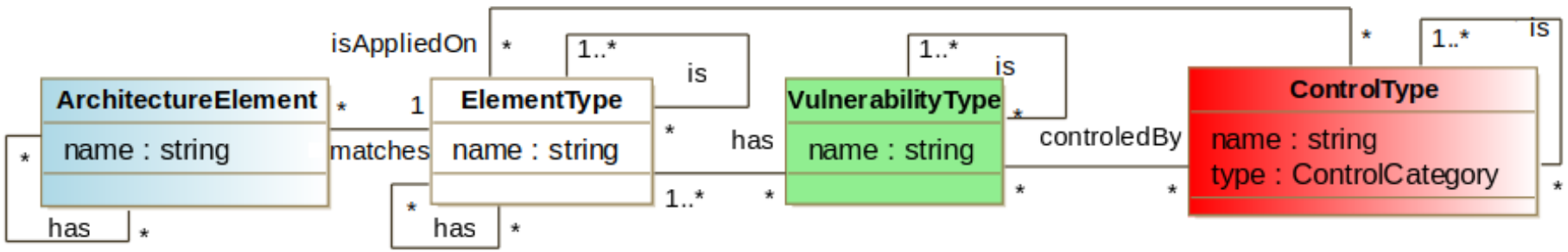
Appendices – The assistance data model : attack specific aspects



Appendices – The assistance data model : defense specific aspects



Appendices – The assistance data model : refinement and structural mechanisms

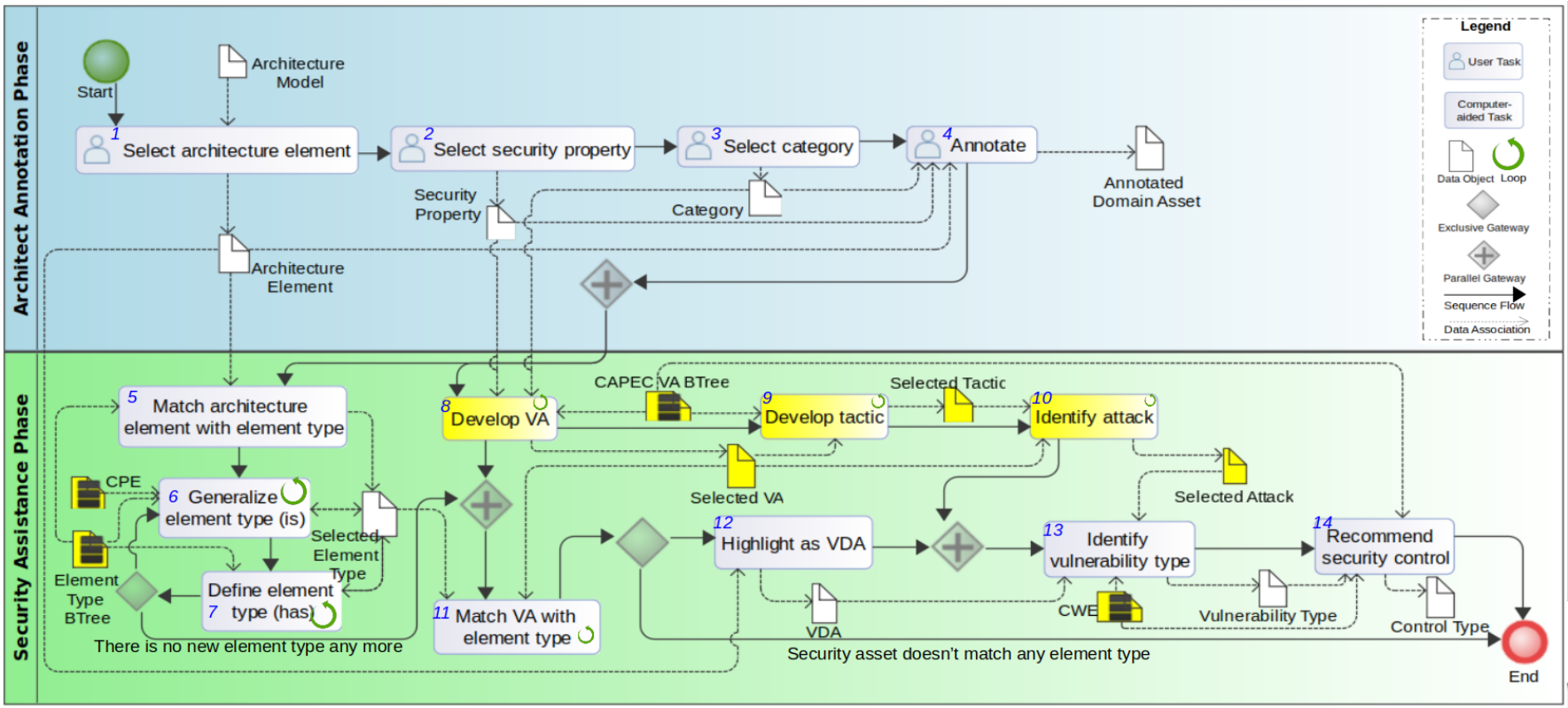


Deal with both abstract and concrete architecture elements' security aspect.

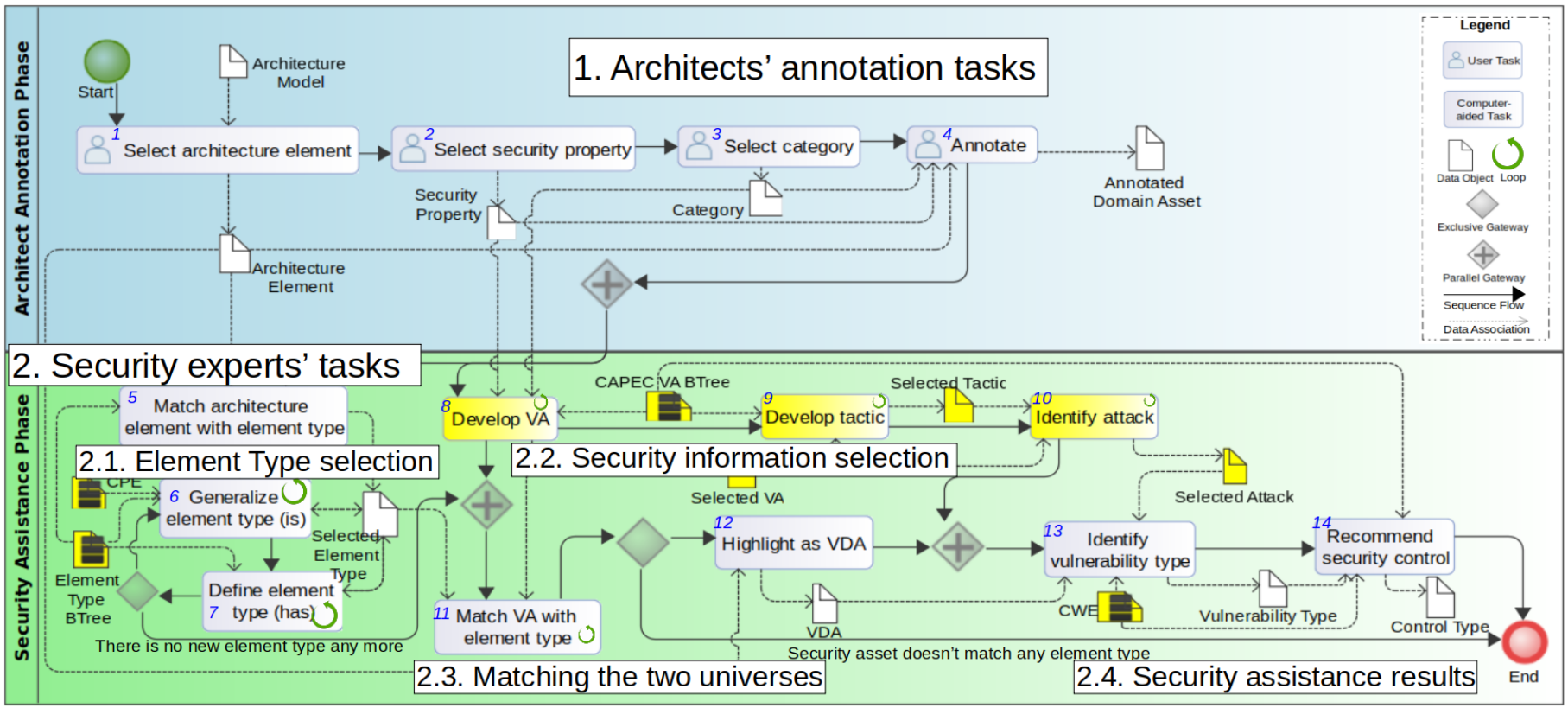
Appendices – The assistance data model : concept definition

Aspect	Concept	Definition
Domain Architecture	AnnotatedDA	A domain asset (or architecture element) that is annotated with a security property and a category.
	ArchitectureElement	An element that the architect wants to protect. It is represented in the architecture model.
	SecurityProperty	The security objective that the architect wants to protect, such as confidentiality, integrity and availability.
	Category	The category of the AnnotatedDA to be protected, such as data, system behavior, human related asset and physical asset.
Attack	Attack	An attempt to destroy, expose, alter, disable, steal or gain unauthorized access to or make unauthorized use of an asset.
	AttackPivot	An attack goal or an attack tactic.
	Goal	An aim or purpose of an attack.
	Tactic	A planned way of conducting an attack
	Root	A pair of a security property with a category, which is the final goal of an attack.
	VA	Anything that is valuable to a security expert. It contains vulnerabilities that can be exploited by attacks.
	VulnerabilityType	A vulnerability, a weakness or a design error that may result in an undesirable event, whose exploitation can compromise the involved VA.
Defense	VDA	Anything that has value for domain experts, but also has vulnerabilities that can be exploited by attacks.
	ControlType	Safeguards or countermeasure, designed to protect the security properties of an asset, and met a set of defined security requirements, is a measure against threat.
Refinement and Structural	ElementType	The general types of elements that we can use as components during the architecture modeling.

Appendices – The assistance process



Appendices – The assistance process



Assistance algorithm

Algorithm 1 Assistance Process

Input

ae, sp, c; /* architecture element, security property, category*/
array ET of n strings; /*list of element types in assistance database*/
2-dimensional array VA[root, va]; /*list of vulnerable assets under different roots in assistance database*/
map CC[container, component]; /*a map of element types with has-a relation*/
map VV[va, vt]; map VC[vt, ct];

Output

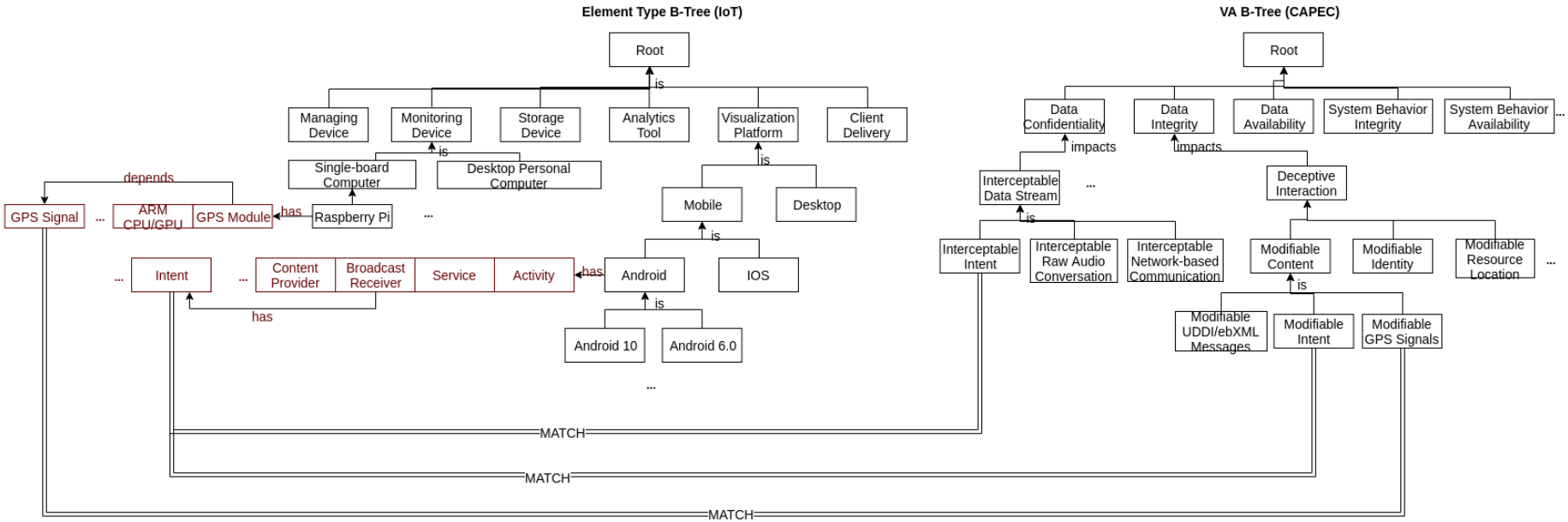
VDA[]:vulnerable domain asset list; VT[]:vulnerability type list; CT[]:control list;

```
1: Declare arrayList LS
2: for i=0 to n -1 do
3:    if ae=ET[i] then
4:       LS.add(ET[i]);
5:    end if
6: end for
7: for all element type j in LS do
8:    if LS[j].parent != NULL then
9:       LS.add(LS[j].parent);
10:   end if
11: end for
12: for all element type j in LS do
13:   if LS[j] in CC.container then
14:      LS.add(CC.component);
15:   end if
16: end for
17: VA.root← c+sp;
18: for all element type j in LS do
19:   if LS[j] in VA.va then
20:      VDA.add(LS[j]);
21:   end if
22: end for
23: for all vda in VV do
24:   VT.add(VV.vulnerabilityType);
25: end for
26: for all vt in VC do
27:   CT.add(VC.ct);
28: end for
```

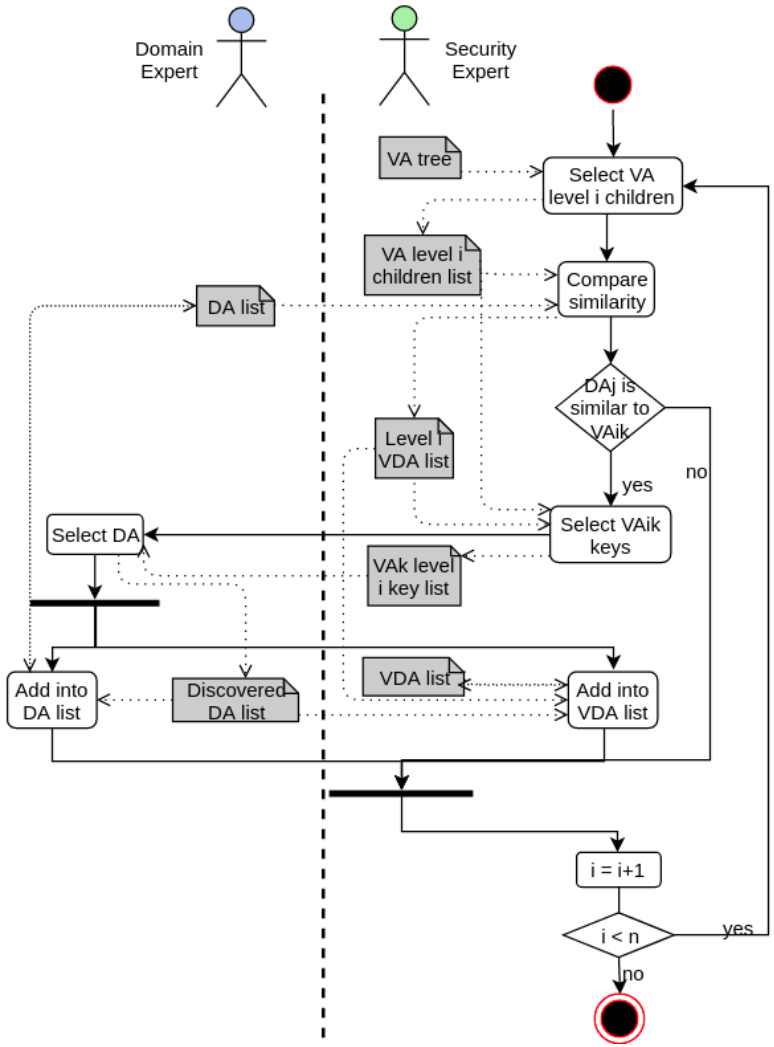
Appendices – Assistance enactment : Application of the security assistance on the motivating example : the results

(4) Annotated DA				Element Type List		(8) VA	(9) Tactic	(10) Attack	(11,12) VDA	(13) Vulnerability Type	(14) Control Type
ID	(1) Architecture Element	(2) Security Property	(3) Category	(5,6) is	(7) has						
DA1	Medtronic 24950 MyCareLink Patient Monitor (Concrete)	Integrity	Data	-Medtronic 24950 MyCareLink Patient Monitor -Monitor -Mobile	Hard-coded Password ...	-Information -Configuration Detail -Software Structure and Composition -Compiled Object -Executable -Machine Instructions -Hard-coded Credential -Hard-coded Password ...	-Reverse engineering -White box reverse engineering -Read sensitive strings within an executable ...	-Hard-coded password realized by reading sensitive strings within an executable ...	-Medtronic Monitor Hard-coded Password ...	-Use of hard-coded credentials (CWE-798) ...	-Utilize a first login mode -Store credentials outside of the code in a strongly protected encrypted configuration file or database... (Concrete)
DA2	Aerospike Database Server 3.10.0.3 (Concrete)	Confidentiality	Data	-Aerospike database server 3.10.0.3 -Aerospike Database Server -Database Server -Server	-SQL Statement ...	-Data Input Interpretation -Command Input Interpretation -SQL Statement ...	-Blind SQL statement -Command line execution through SQL injection ...	-SQL statement compromised by command line execution through SQL injection ...	-Aerospike Database SQL Statement	-Improper input validation (CWE-20) ...	-Use an "accept known good" input validation strategy... (Abstract)
DA3	Remote Desktop (Abstract)	Availability	System Behavior	-Remote desktop -Desktop	-Web Browser -XML Parser...	-Application functionality -Appropriate memory allocation -XML parser...	-XML entity expansion -XML quadratic expansion...	-XML parser compromised by XML entity expansion...	-Remote Desktop XML Parser	-missing XML validation (CWE-112) ...	-always validate XML input against a known XML Schema or DTD... (abstract)

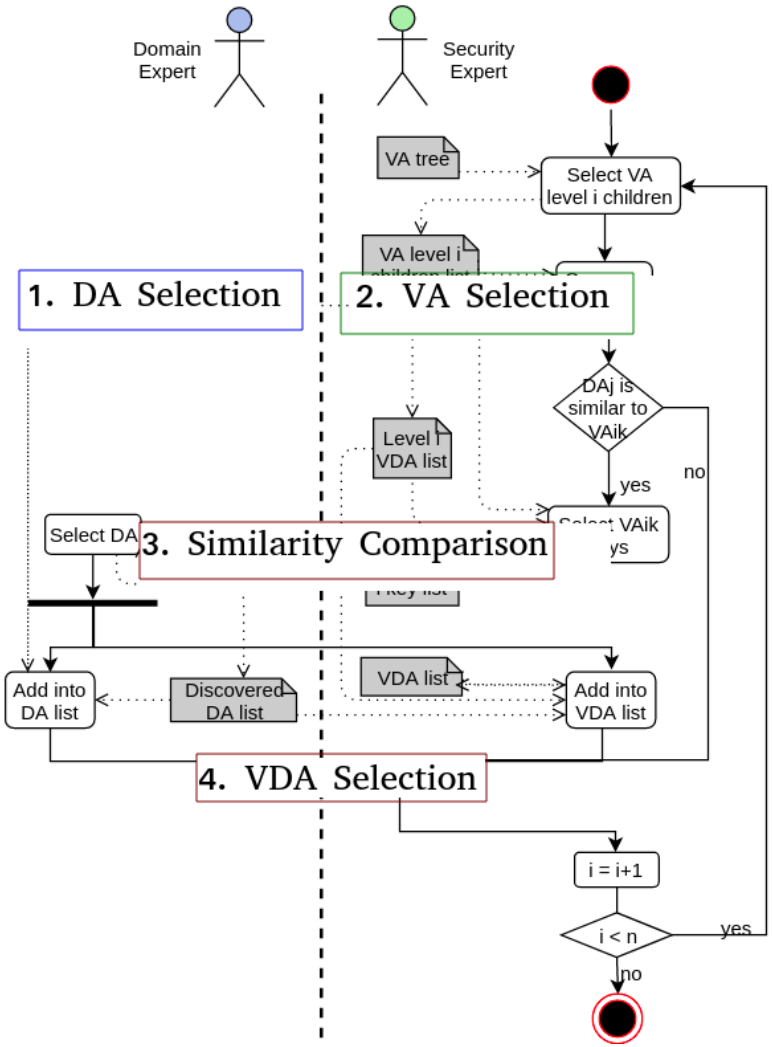
Appendices – DA B-Tree matches VA B-Tree : An example



Appendices – Asset identification process



Appendices – Asset identification process



Some rules to extract VAs and their relations basing on CAPEC

- **Rule 1** : 'contaminate' | 'poison' | 'leverage' | 'manipulate' | 'abuse' | 'exploit' | 'misuse' + VA (Ex. 'Poison web service registry') ;
- **Rule 2** : VA + 'manipulation' | 'poisoning' | 'tampering' | 'alteration' (Ex. 'Web service protocol manipulation') ;
- **Rule 3** : VA + 'injection' | 'inclusion' | 'insertion' ; VA = 'Untested' + VA + 'Input' (Ex. 'XML injection', VA = 'UntestedXMLInput') ;
- **Rule 4** : 'childOf' $\rightarrow$ 'is' | 'has' (Ex. '**SOAP** manipulation' *is* a '**web services protocol** manipulation' ; '**XML** injection' *has* '**DTD** injection') ;
- **Rule 5** : 'canFollow' $\rightarrow$ 'depends'.

Appendices – Keywords in three clusters

```
#Découper les mots simples et comparer avec cette liste de mots clés. (VA + mot clé)
extractKeyword1 <- function(){
  res <- c("Spoofing","Phishing","Hijacking","Overlay","Squatting",
           "Monitoring","Flood","Splitting","Smuggling","Tampering",
           "Bypass","Abuse","Overflow","Poisoning","Disabling",
           "Seizure","Jamming","Blocking","Alteration","Analysis",
           "Impersonation", "Manipulation", "Expansion", "Linking",
           "Blowup", "Fragmentation", "Misuse", "Exploitation",
           "Altered","Injection", "Pollution", "Inclusion",
           "Insertion", "Scanning", "Discovery", "Footprinting",
           "Fingerprinting", "Probe")
  return(res)
}
```

```
#Ne pas découper le mots composés pour extractKeyword2,
#seulement localiser en text mining. (mot clé + VA)
extractKeyword2 <- function(){
  res <- c("Spoofing of ","Exploiting Incorrectly","Modification of",
           "Collect Data from","Pretexting via","Bypassing of ")
  return(res)
}
```

```
#Découper les mots simple et comparer avec cette liste de mots clés. (mot clé + VA)
extractKeyword3 <- function(){
  res <- c("Manipulate","Leveraging","Manipulating","Disabling","Accessing",
           "Intercepting","Modifying","Counterfeit","Fake the", "Exploit",
           "Using", "Leverage", "Bypassing", "Poison", "Infected", "Contaminate",
           "Detect","Probe", "Capture", "Sniffing")
  return(res)
}
```

Proof-of-Concept

Proof-of-Concept	***Security Assistance***	***Structuring Threat Modeling (Integration)***
Comparison With	***Microsoft SDL Tool***	***Microsoft Threat Modeling Process***
Quality Result	Capability of Identifying Threat, Vulnerability, Security Control	
Quantity Result	Number of Threats, Number of Vulnerabilities, Number of Security Controls	
Usefulness	Questionnaire	

Table – Proof-of-concept experimentation

Crossover experimentation

- What ?

Sequence \ Period	Period 1	Period 2
Group 1 : AB (Sequence 1)	Treatment A	Treatment B
Group 2 : BA (Sequence 2)	Treatment B	Treatment A

Table – AB-BA-Crossover Trial

24. Sira Vegas, Cecilia Apa et Natalia Juristo. "Cross-Over Designs in Software Engineering Experiments : Benefits and Perils". In : *IEEE Transactions on Software Engineering* 42 (jan. 2015), p. 1-1. doi : 10.1109/TSE.2015.2467378.

Crossover experimentation

- What ?

Sequence \ Period	Period 1	Period 2
Group 1 : AB (Sequence 1)	Treatment A	Treatment B
Group 2 : BA (Sequence 2)	Treatment B	Treatment A

Table – AB-BA-Crossover Trial

- Why ?
  - Control the variability among subjects

24. Sira Vegas, Cecilia Apa et Natalia Juristo. "Cross-Over Designs in Software Engineering Experiments : Benefits and Perils". In : *IEEE Transactions on Software Engineering* 42 (jan. 2015), p. 1-1. doi : 10.1109/TSE.2015.2467378.

Crossover experimentation

- What ?

Sequence \ Period	Period 1	Period 2
Group 1 : AB (Sequence 1)	Treatment A	Treatment B
Group 2 : BA (Sequence 2)	Treatment B	Treatment A

Table – AB-BA-Crossover Trial

- Why ?
  - Control the variability among subjects
- How ?
  - Good practices
  - Two case studies → avoid carryover threat

24. Sira Vegas, Cecilia Apa et Natalia Juristo. "Cross-Over Designs in Software Engineering Experiments : Benefits and Perils". In : *IEEE Transactions on Software Engineering* 42 (jan. 2015), p. 1-1. doi : 10.1109/TSE.2015.2467378.

Crossover experimentation – security assistance

	Period 1		**Period 2**	
Group 1 : AB (Sequence 1)	*Treatment A Microsoft SDL tool (Case study 1)*		*Treatment B security assistance (Case study 2)*	
	Number of Threats	Number of Vulnerabilities	Number of Threats	Number of Vulnerabilities
Subject 1	4	X	24	52
Subject 4	15	X	21	49
Subject 5	15	X	28	70
Group 2 : BA (Sequence 2)	*Treatment B security assistance (Case study 1)*		*Treatment A Microsoft SDL tool (Case study 2)*	
	Number of Threats	Number of Vulnerabilities	Number of Threats	Number of Vulnerabilities
Subject 2&3	10	39	14	X
Subject 6	16	12	6	X
Subject 7	16	9	7	X

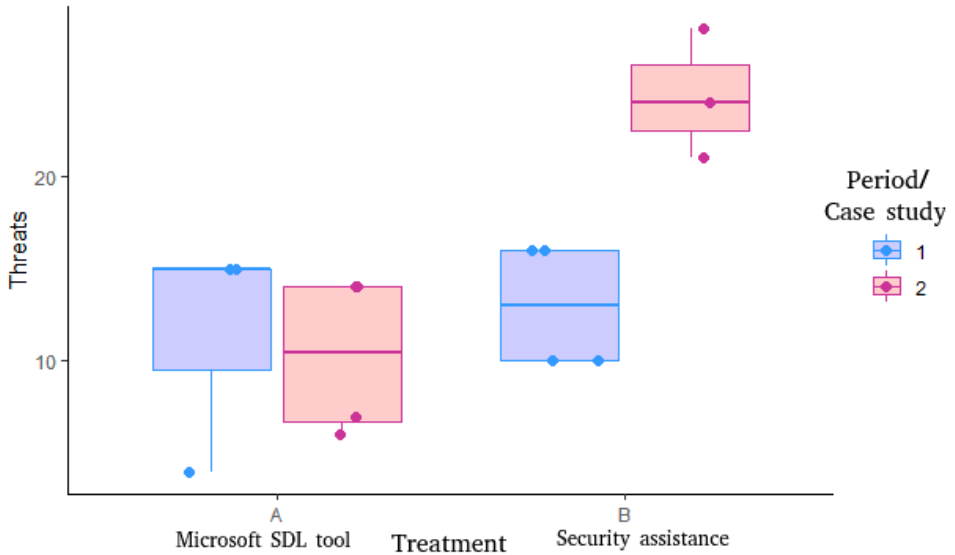
Table – Security assistance crossover study results

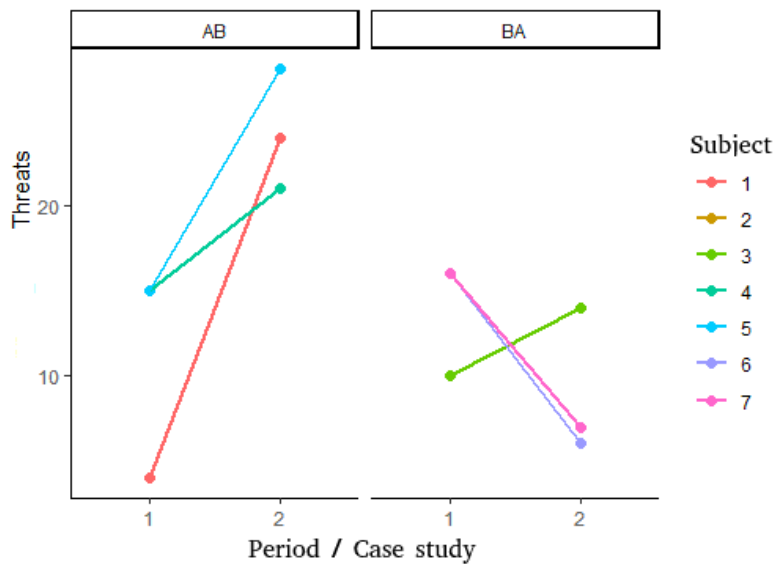
The quality analysis

Tool \ Capability	Identify critical threats	Threats reference	Identify vulnerabilities	Propose security controls
Microsoft SDL Tool	X	STRIDE		
Security Assistance	X	CAPEC/CWE (MITRE)	X	X

Table – Tool quality comparison

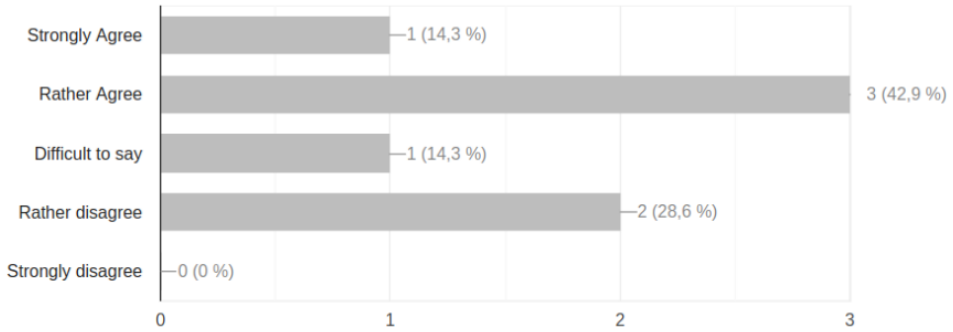
The quantity analysis



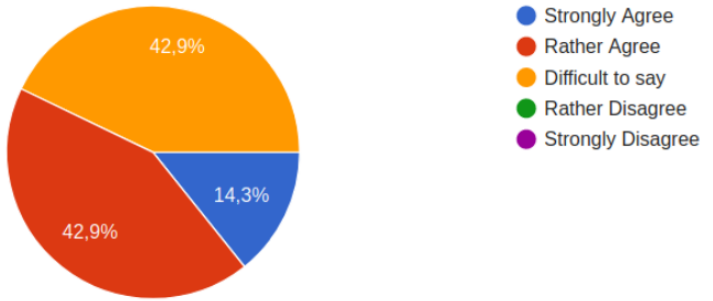


The usefulness evaluation – questionnaire

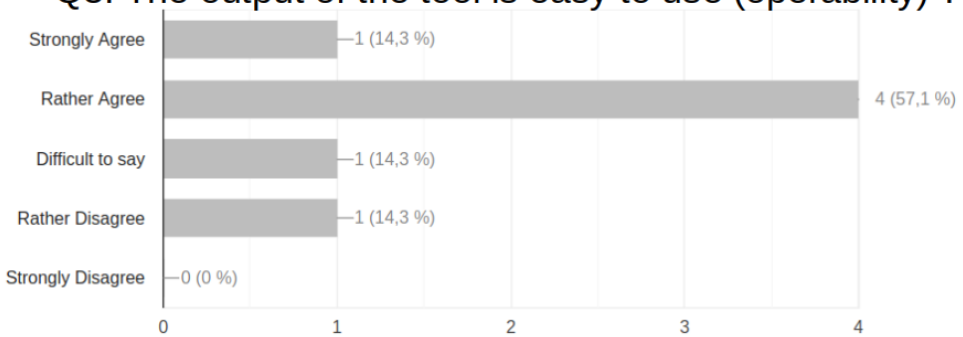
Q1: The assistance tool is easy to use (operability)?



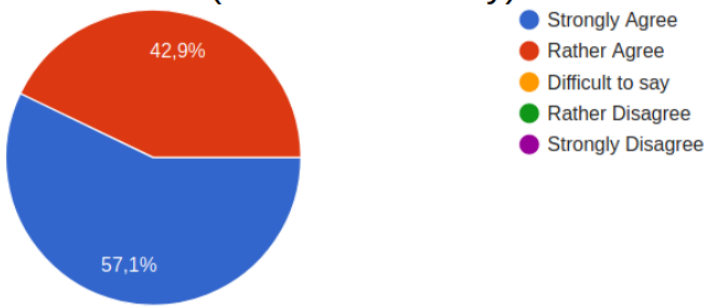
Q2: The instruction for the assistance tool is easy to understand (learnability) ?



Q3: The output of the tool is easy to use (operability) ?



Q4: The output of the tool is easy to understand (understandability) ?



Crossover experimentation planning – threat modeling process

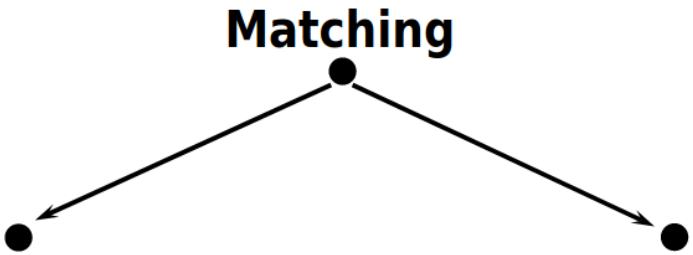
Period / Sequence	**Period 1**	**Period 2**
Group 1: AB (Sequence 1)	*Treatment A* : Microsoft SDL threat modeling process (Case study 1)	*Treatment B* : integration with asset identification process (Case study 2)
Group 2: BA (Sequence 2)	*Treatment B* : integration with asset identification process (Case study 1)	*Treatment A* : Microsoft SDL threat modeling process (Case study 2)

- **Subject**: a group participants during the brainstorming session
- **Response variable**: the process
- Main experiment **factor**: the process with two **treatments**

Matching

Matching

Matching: given two graph-like structures (e.g., concept hierarchies or ontologies), produce a mapping between the nodes of the graphs that semantically correspond to each other



Syntactic Matching

- Relations are computed between labels at nodes
- R = {x∈[0,1]}

Note: all previous systems are syntactic…

Semantic Matching

- Relations are computed between concepts at nodes
- R = { =, ⊒, ⊑, ⊥, ⊓}

Note: First implementation CTXmatch [Bouquet et *al. 2003*]