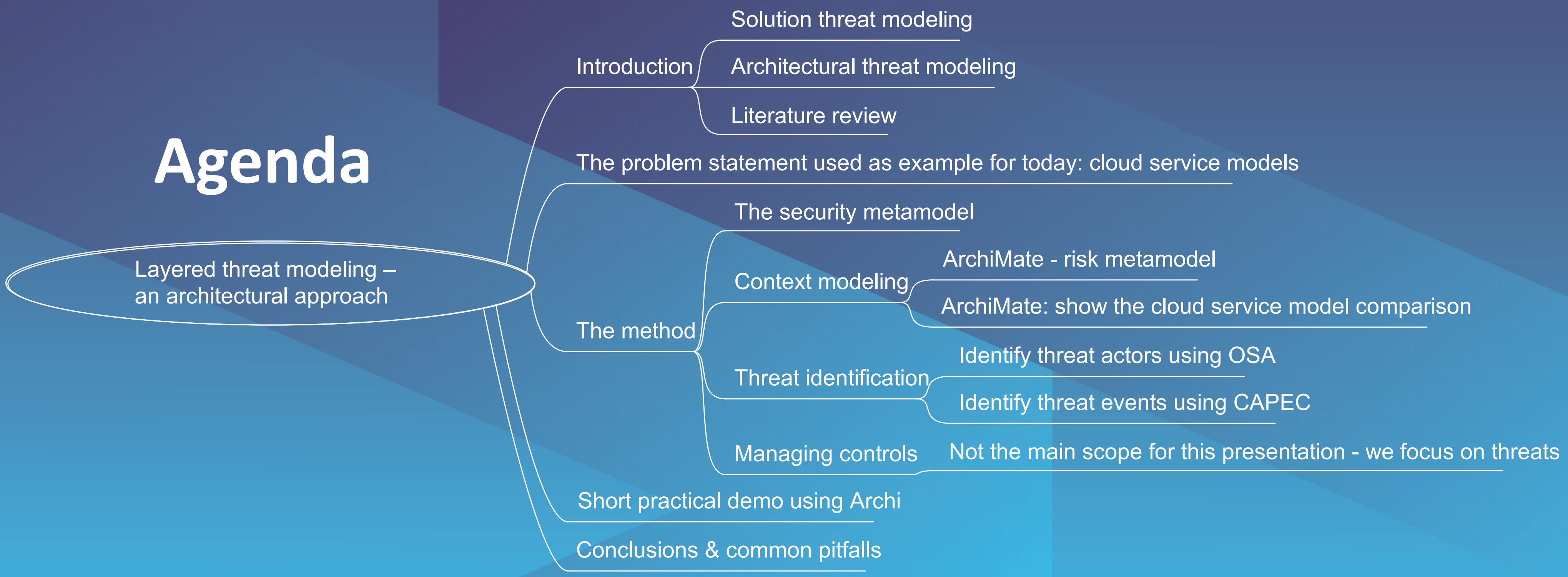
OWASP 2022
VIRTUAL
APPSEC
JUN 6-10

PRESENTED BY: MICHAEL BOEYNAEMS / **CO-CREATOR**: MATHIJS HUBRECHTSEN

Layered threat modeling - an architectural approach

Agenda



Layered threat modeling – an architectural approach

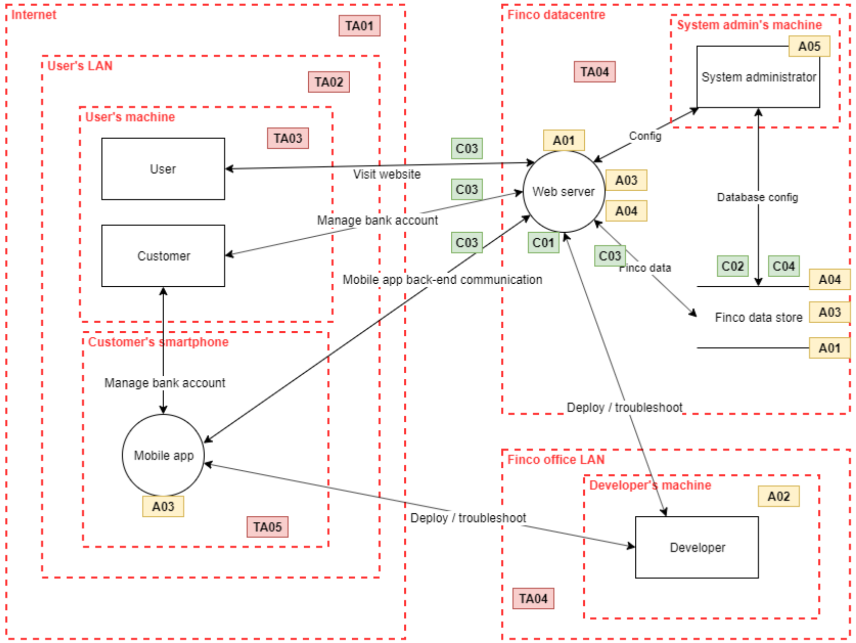
- Introduction
  - Solution threat modeling
  - Architectural threat modeling
  - Literature review
- The problem statement used as example for today: cloud service models
- The method
  - The security metamodel
  - Context modeling
    - ArchiMate - risk metamodel
    - ArchiMate: show the cloud service model comparison
  - Threat identification
    - Identify threat actors using OSA
    - Identify threat events using CAPEC
  - Managing controls
    - Not the main scope for this presentation - we focus on threats
- Short practical demo using Archi
- Conclusions & common pitfalls

OWASP 2022
VIRTUAL
APPSEC
JUN 6-10

Introduction

You all know about solution threat modeling

- A solution threat model is **focused on a single solution**.
- Various notations can be used: DFDs, UML diagrams, …
- Various techniques can be used: STRIDE, LINDDUN, …



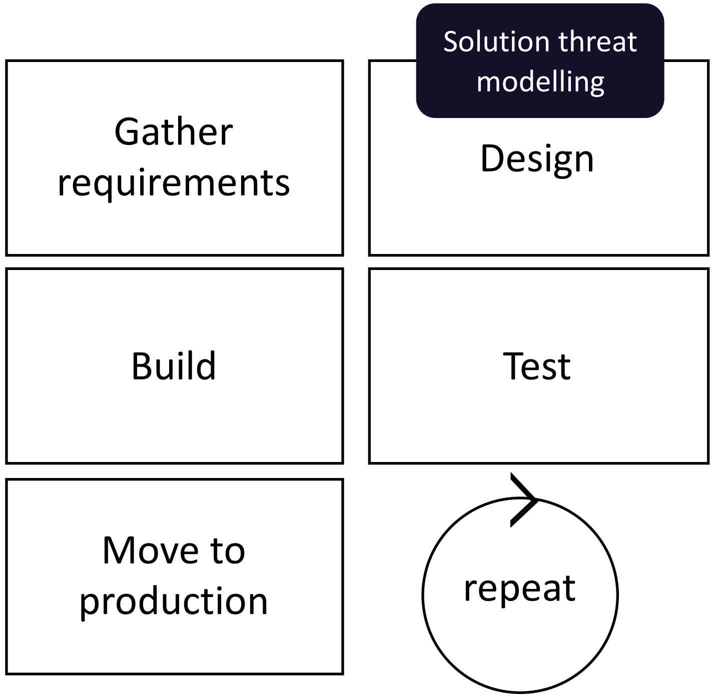
Assets	
ID	Description
A01	User credentials
A02	Source code
A03	Bank account information
A04	Database credentials
A05	Root credentials

Threat Actors	
ID	Description
TA01	Unauthenticated external user (Internet attacker)
TA02	Unauthenticated internal user (LAN attacker)
TA03	Malicious customer
TA04	Malicous employee
TA05	Attacker with jail-broken device

Security Controls	
ID	Description
C01	Authentication
C02	Password hashing
C03	TLS (in transit)
C04	Database encryption (at rest)

You all know about solution threat modeling

- A solution threat model is **created during the design or build phase**.



Solution threat modelling

Gather requirements

Design

Build

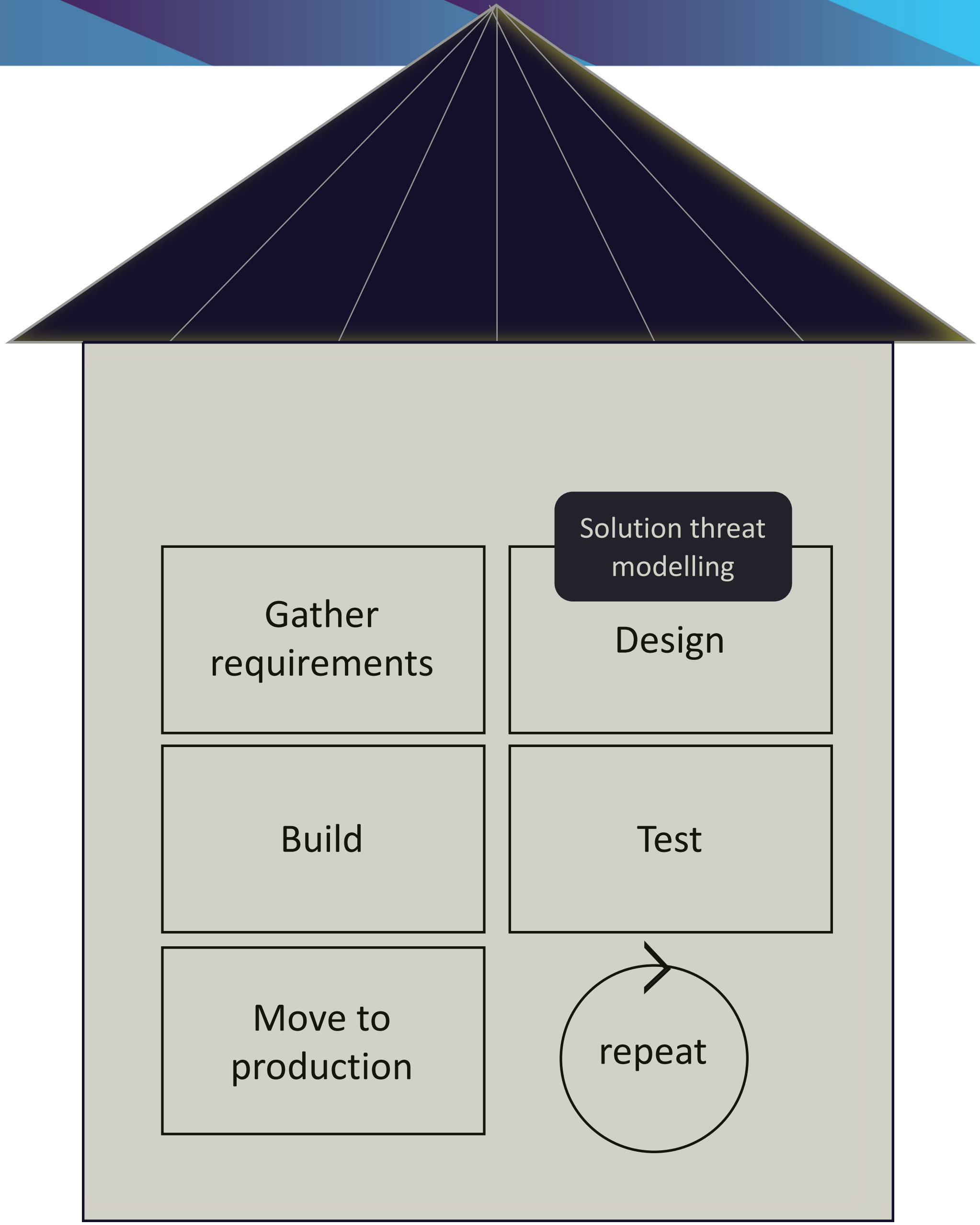
Test

Move to production

repeat

You all know about solution threat modeling

A solution threat model helps you to securely design a barn...

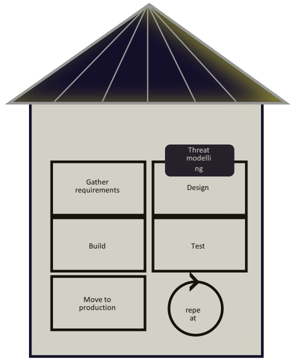


… but don’t you want to architect an entire FARM?

Two layers of threat modeling

Software (security) architect

- Helps design one barn
- Employs solution threat modeling
- Defines system and development security controls

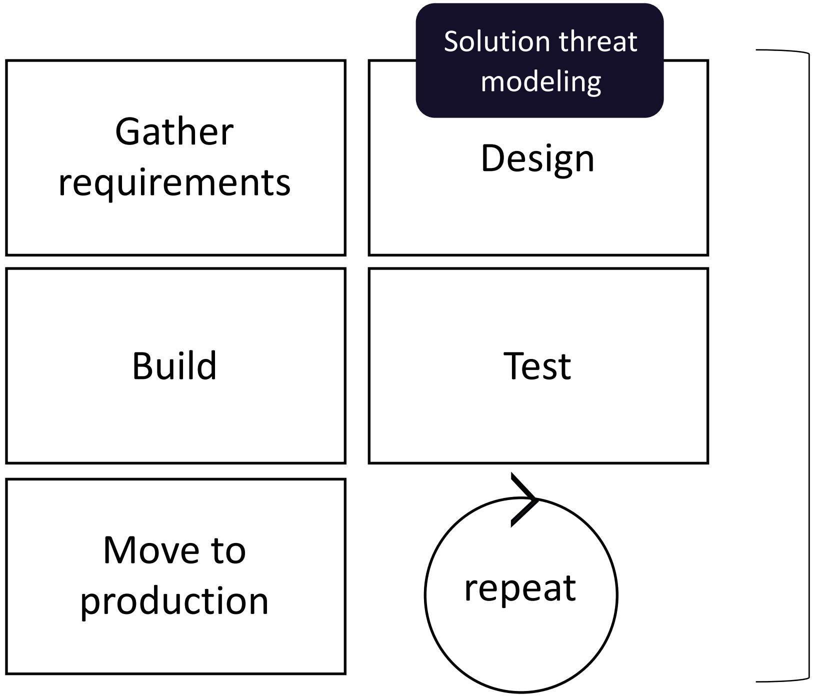


Enterprise (security) architect

- Helps design a complete farm
- Employs architectural threat modeling
- Defines security objectives, principles and generic security controls

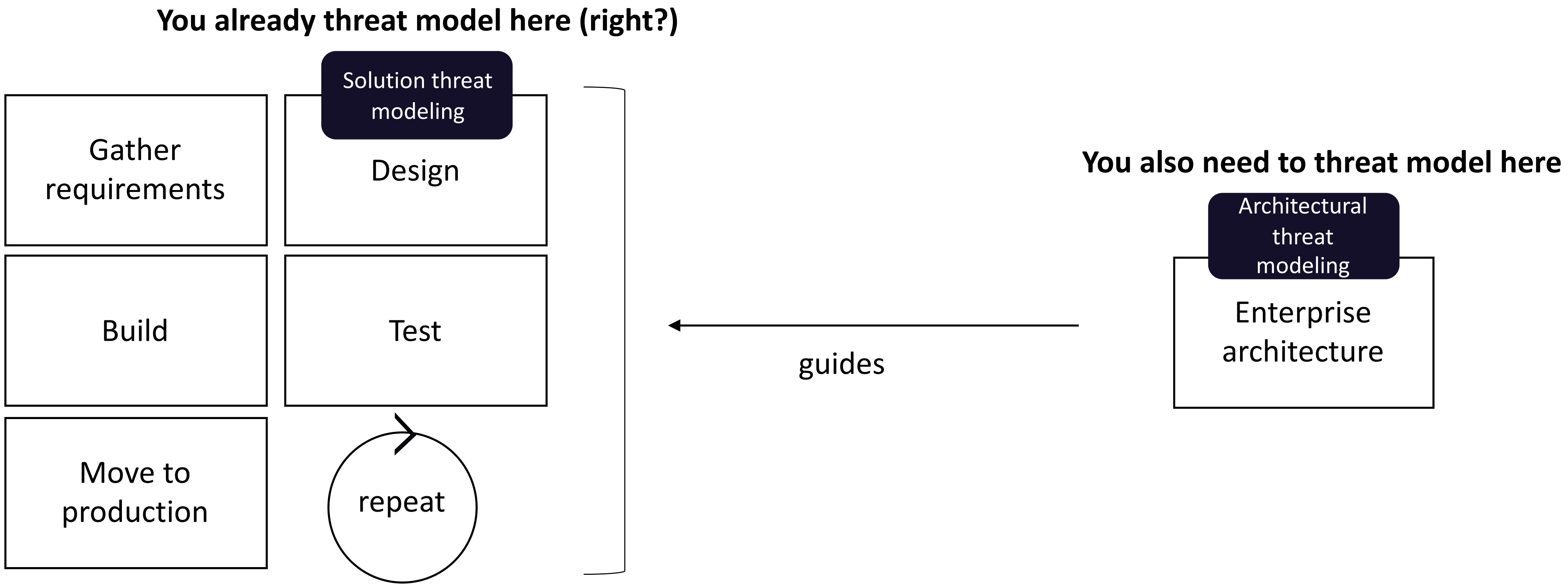
Two layers of threat modeling

You already threat model here (right?)



Designing one barn
'Solution threat modeling'

Two layers of threat modeling

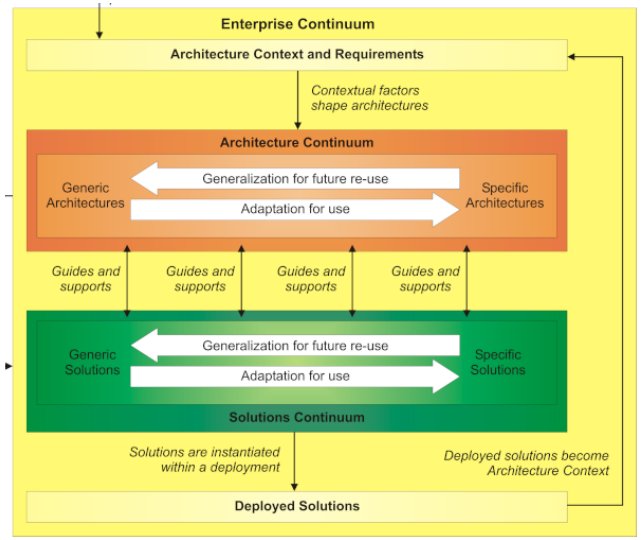


Designing one barn
'Solution threat modeling'

Designing an entire farm
'Architectural threat modeling'

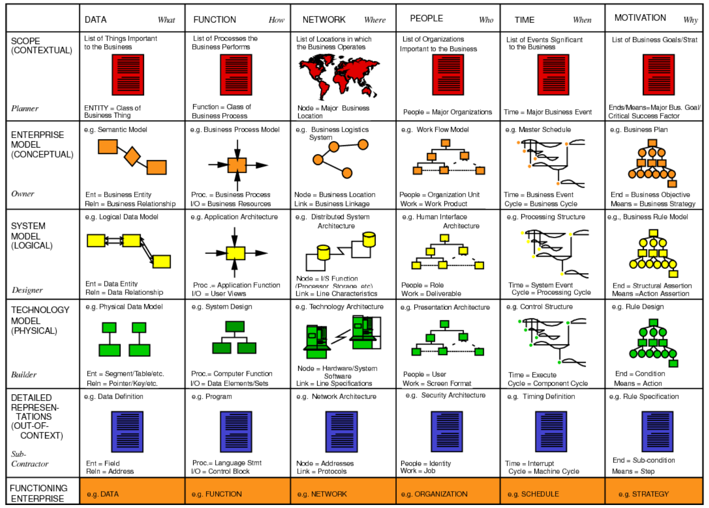
Two layers of threat modeling

The distinction between EA threat modeling and solution threat modeling is confirmed by a lot of frameworks



TOGAF

SABSA (blurred for licensing reasons)

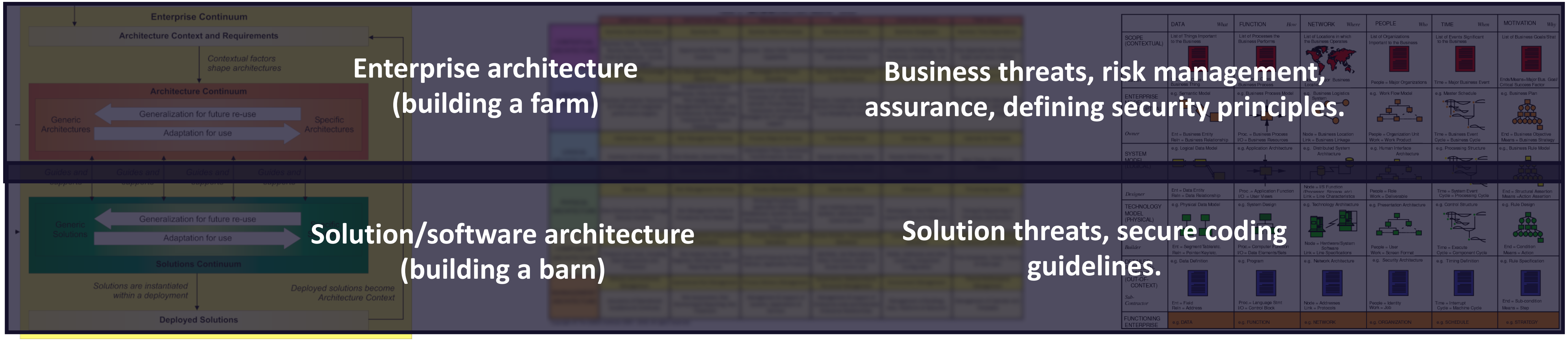


	DATA *What*	FUNCTION *How*	NETWORK *Where*	PEOPLE *Who*	TIME *When*	MOTIVATION *Why*
SCOPE (CONTEXTUAL) *Planner*	List of Things Important to the Business ENTITY = Class of Business Thing	List of Processes the Business Performs Function = Class of Business Process	List of Locations in which the Business Operates Node = Major Business Location	List of Organizations Important to the Business People = Major Organizations	List of Events Significant to the Business Time = Major Business Event	List of Business Goals/Strat Ends/Means=Major Bus. Goal/ Critical Success Factor
ENTERPRISE MODEL (CONCEPTUAL) *Owner*	e.g. Semantic Model Ent = Business Entity Reln = Business Relationship	e.g. Business Process Model Proc. = Business Process I/O = Business Resources	e.g. Business Logistics System Node = Business Location Link = Business Linkage	e.g. Work Flow Model People = Organization Unit Work = Work Product	e.g. Master Schedule Time = Business Event Cycle = Business Cycle	e.g. Business Plan End = Business Objective Means = Business Strategy
SYSTEM MODEL (LOGICAL) *Designer*	e.g. Logical Data Model Ent = Data Entity Reln = Data Relationship	e.g. Application Architecture Proc .= Application Function I/O = User Views	e.g. Distributed System Architecture Node = I/S Function (Processor, Storage, etc) Link = Line Characteristics	e.g. Human Interface Architecture People = Role Work = Deliverable	e.g. Processing Structure Time = System Event Cycle = Processing Cycle	e.g., Business Rule Model End = Structural Assertion Means =Action Assertion
TECHNOLOGY MODEL (PHYSICAL) *Builder*	e.g. Physical Data Model Ent = Segment/Table/etc. Reln = Pointer/Key/etc.	e.g. System Design Proc.= Computer Function I/O = Data Elements/Sets	e.g. Technology Architecture Node = Hardware/System Software Link = Line Specifications	e.g. Presentation Architecture People = User Work = Screen Format	e.g. Control Structure Time = Execute Cycle = Component Cycle	e.g. Rule Design End = Condition Means = Action
DETAILED REPRESEN-TATIONS (OUT-OF-CONTEXT) *Sub-Contractor*	e.g. Data Definition Ent = Field Reln = Address	e.g. Program Proc.= Language Stmt I/O = Control Block	e.g. Network Architecture Node = Addresses Link = Protocols	e.g. Security Architecture People = Identity Work = Job	e.g. Timing Definition Time = Interrupt Cycle = Machine Cycle	e.g. Rule Specification End = Sub-condition Means = Step
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY

Zachman

Two layers of threat modeling

The distinction between EA threat modeling and solution threat modeling is confirmed by a lot of frameworks



Enterprise architecture (building a farm)

Business threats, risk management, assurance, defining security principles.

Solution/software architecture (building a barn)

Solution threats, secure coding guidelines.

TOGAF

SABSA (blurred for licensing reasons)

Zachman

An example:
What threats do we face in cloud service models and which security principles must be followed to manage these threats?

Introducing the cloud problem statement

- Essential characteristics:
  - On demand self-service
  - Broad network access
  - Resource pooling
  - Rapid elasticity
  - Measured services
- Service models
  - Software as a service
  - Platform as a service
  - Infrastructure as a service
- Deployment models
  - Private cloud
  - Community
  - Public cloud
  - Hybrid cloud

		SaaS	PaaS	IaaS	On-prem
Responsibility always retained by customer	Information and data	🟨	🟨	🟨	🟨
	End user devices	🟨	🟨	🟨	🟨
	Accounts and identities	🟨	🟨	🟨	🟨
Responsibility varies by type	Identity infrastructure	◩	◩	🟨	🟨
	Applications	■	◩	🟨	🟨
	Network controls	■	◩	🟨	🟨
Responsibility transfers to cloud provider	Operating system	■	■	🟨	🟨
	Physical hosts	■	■	■	🟨
	Physical network	■	■	■	🟨
	Physical datacenter	■	■	■	🟨

■ Cloud provider 🟨 Customer ◩ Shared

Source: Microsoft

OWASP 2022
VIRTUAL
APPSEC
JUN 6-10

Architectural threat modeling

Step 0: you need a metamodel

ISSRM mapped to threat model concepts and ArchiMate elements.

ISSRM (1)	Threat model concepts (2)	TOGAF/ArchiMate (3)	ArchiMate metamodel used in this talk
Asset	Asset	Resource	Resource
Business Asset	Business Asset	Any Business element	Any Business element
IS Asset	IS Asset	Any Application or Technology element	Any Application or Technology element
Security Objective	Security Objective	Driver	Driver
Risk	Risk	Assessment	Assessment
Event	Event	Assessment	Event
Impact	Impact	Assessment	Assessment
Threat	/	Assessment	See threat event / threat agent
/	Threat event	/	Event
/	Threat agent	/	Actor
Vulnerability	Vulnerability	Assessment	Assessment
Risk Treatment	Risk Treatment	Goal	Course of action
Security Requirement	Security Requirement	Requirement	Requirement
Control	Control	Core element ('implemented control')	Core element

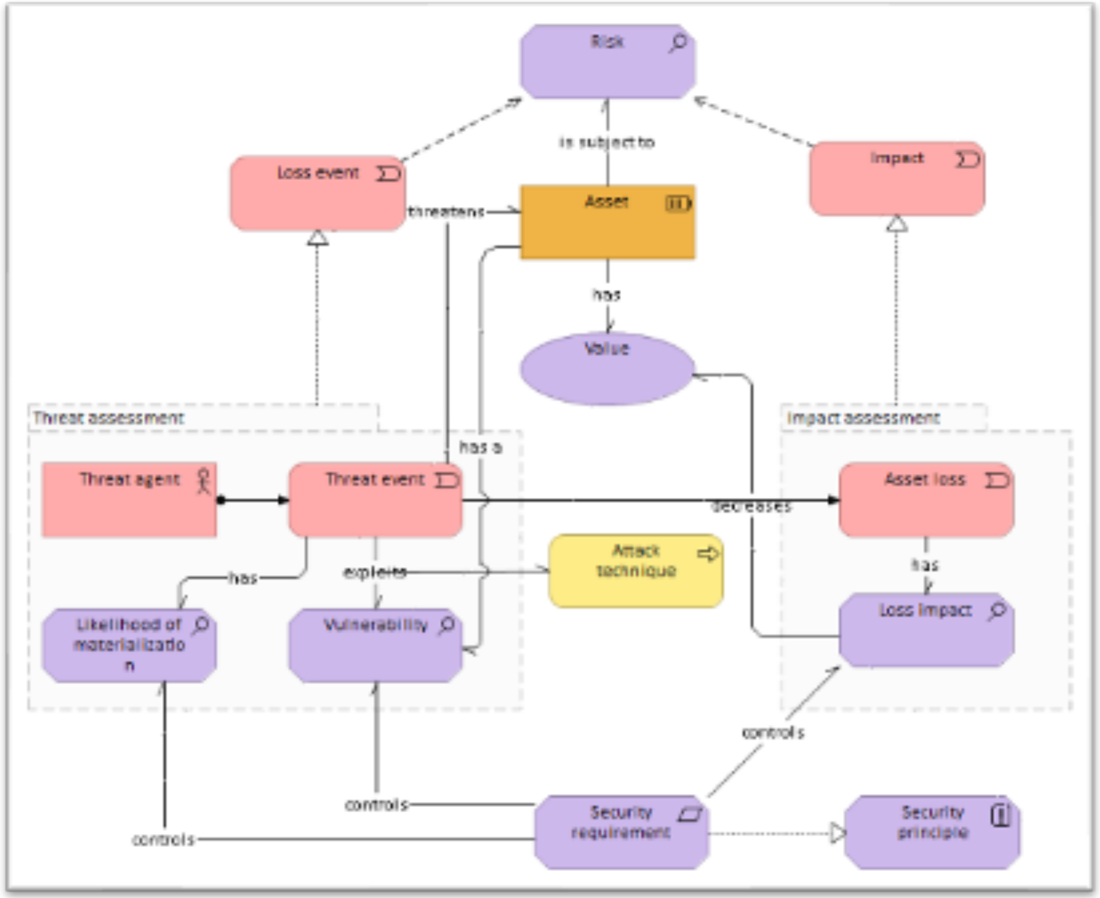
Sometimes the concept 'attack' is also used. Note that every attack possibly leads to a threat, but not every threat is linked to an attack.

(1) E. Dubois, P. Heymans, N. Mayer, R. Matulevičius: A Systematic Approach to Define the Domain of Information System Security Risk Management (ISSRM), in Intentional Perspectives on Information Systems Engineering, S. Nurcan, C. Salinesi, C. Souveyet, J. Ralyté, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010 (pp.289-306).
(2) Based on NIST, Shostack, The Open Group
(3) The Open Group, How to Model Enterprise Risk Management and Security with the ArchiMate® Language

Step 0: you need a metamodel

ISSRM mapped to threat model concepts and ArchiMate elements.

Resulting in a metamodel that we can use in practice.



ArchiMate metamodel used in this talk

Step 1: you need an architectural model

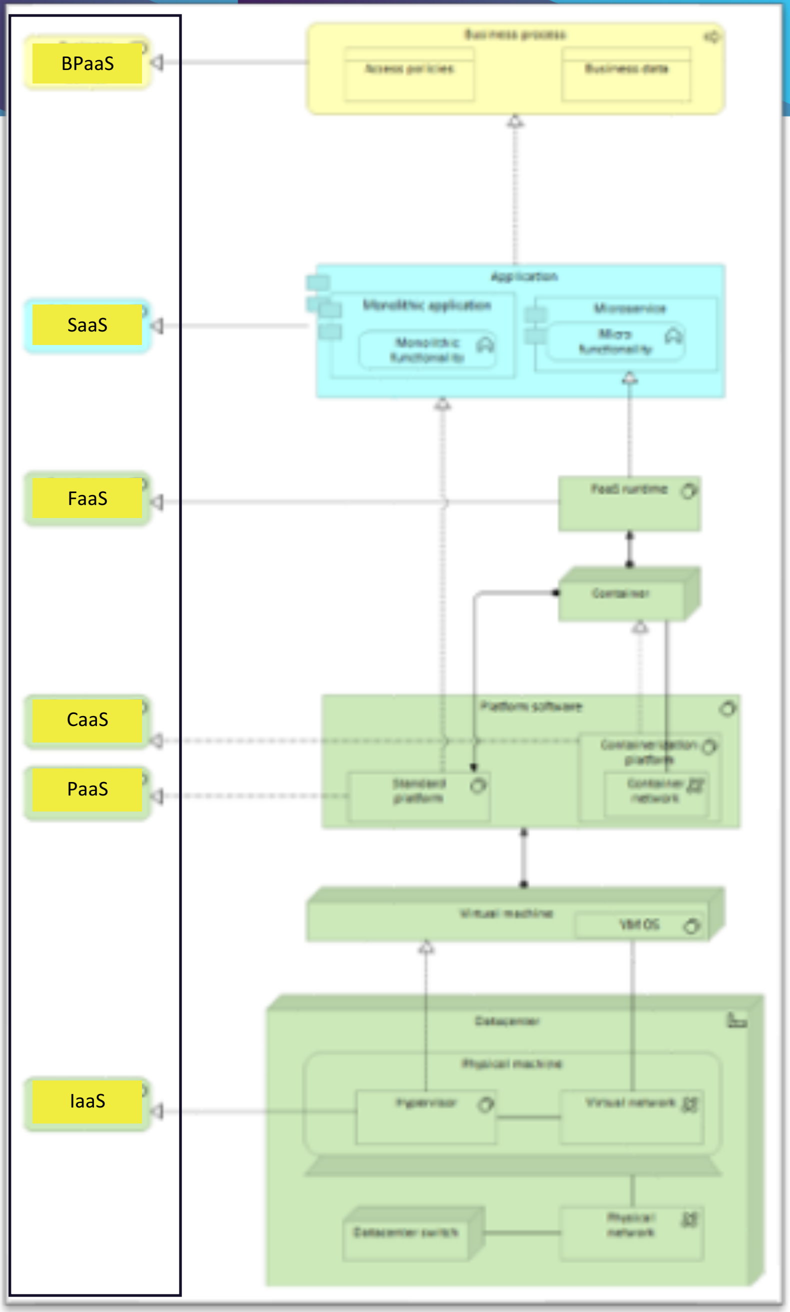
We use the ArchiMate notation as it

- Is a de facto standard for (enterprise) architectural modeling;
- It facilitates linking between business, applicative, infrastructural, and data architectures;

"In effect, ArchiMate describes the structure of cities, while UML describes the structure of houses and office buildings. Both are needed, and they solve different problems. In that way, they do not intersect at all. Unfortunately, the diagramming notations are not so consistent." - Nick Malik ,2009

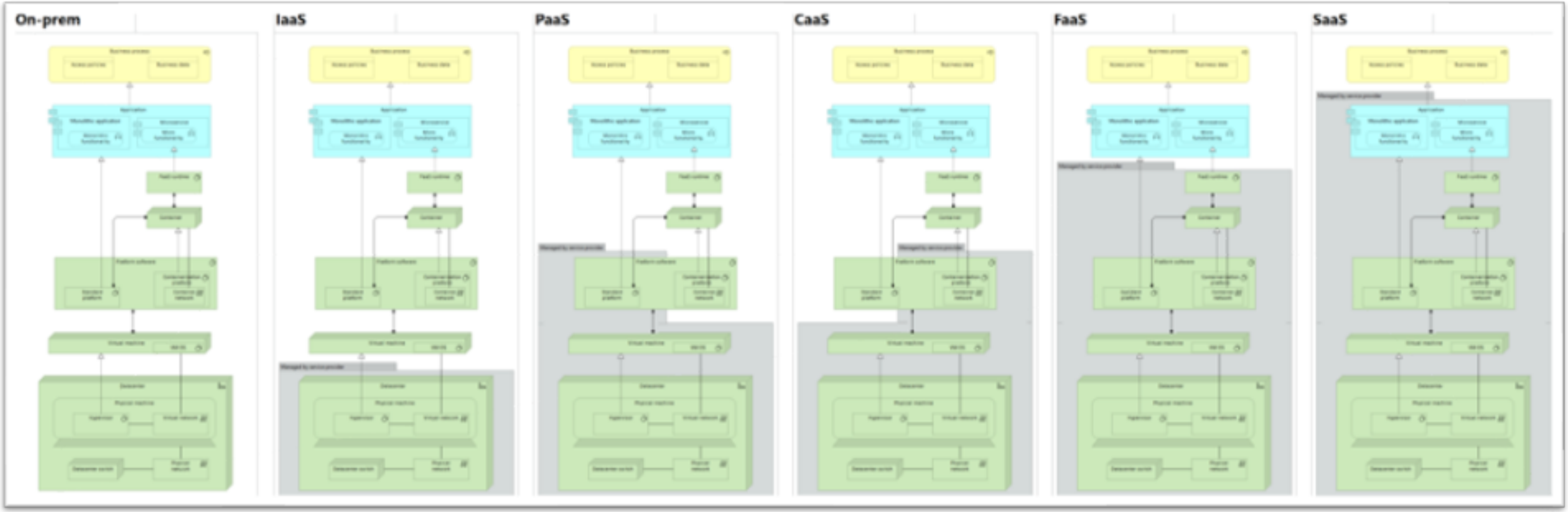
https://docs.microsoft.com/en-us/archive/blogs/nickmalik/will-there-be-a-battle-between-archimate-and-the-uml

Cloud service models - responsibility



Generic cloud architecture

Step 1: you need an architectural model

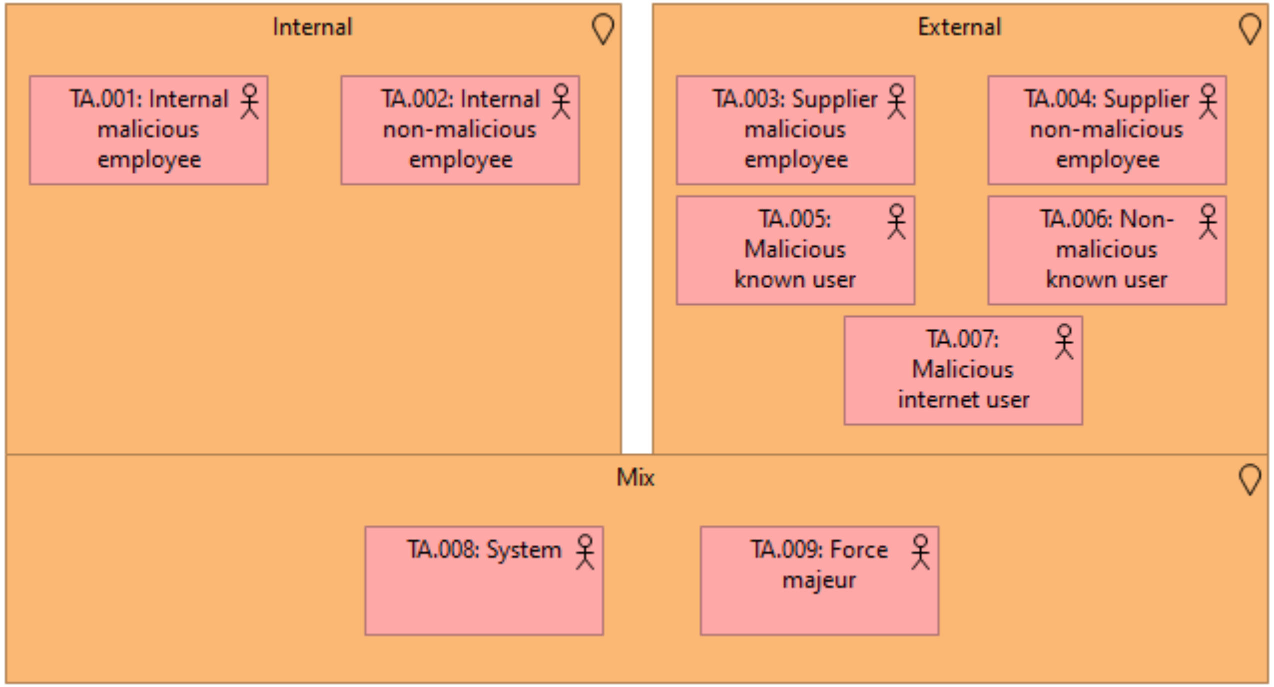


Cloud service models - responsibility

▨ Managed by service provider

Step 2a: you need to identify threat actors

We loosely base threat actor identification on the OSA threat classification method



Internal

TA.001: Internal malicious employee

TA.002: Internal non-malicious employee

External

TA.003: Supplier malicious employee

TA.004: Supplier non-malicious employee

TA.005: Malicious known user

TA.006: Non-malicious known user

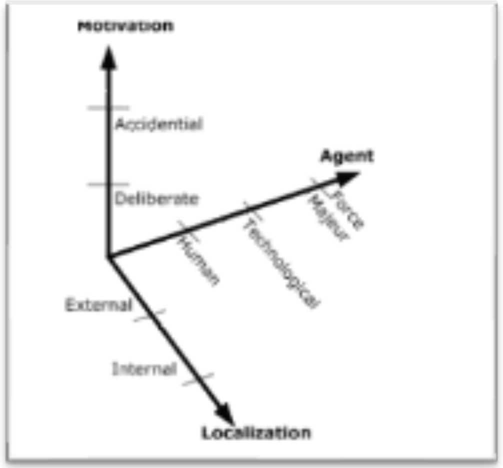
TA.007: Malicious internet user

Mix

TA.008: System

TA.009: Force majeur

Threat actors

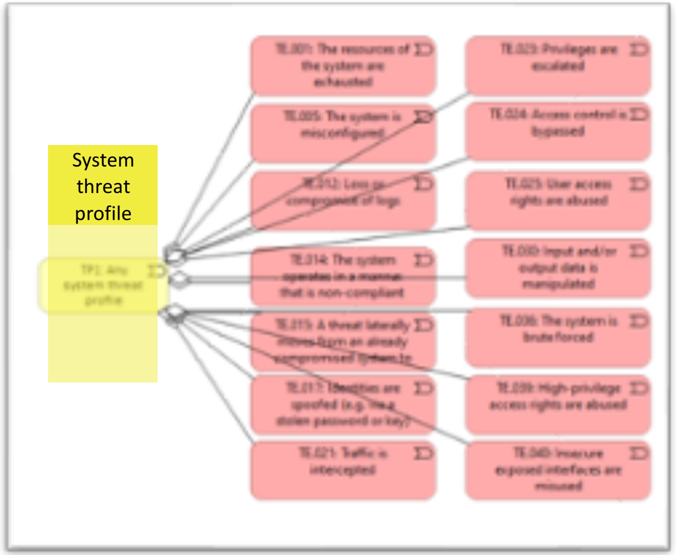


Threat classification method

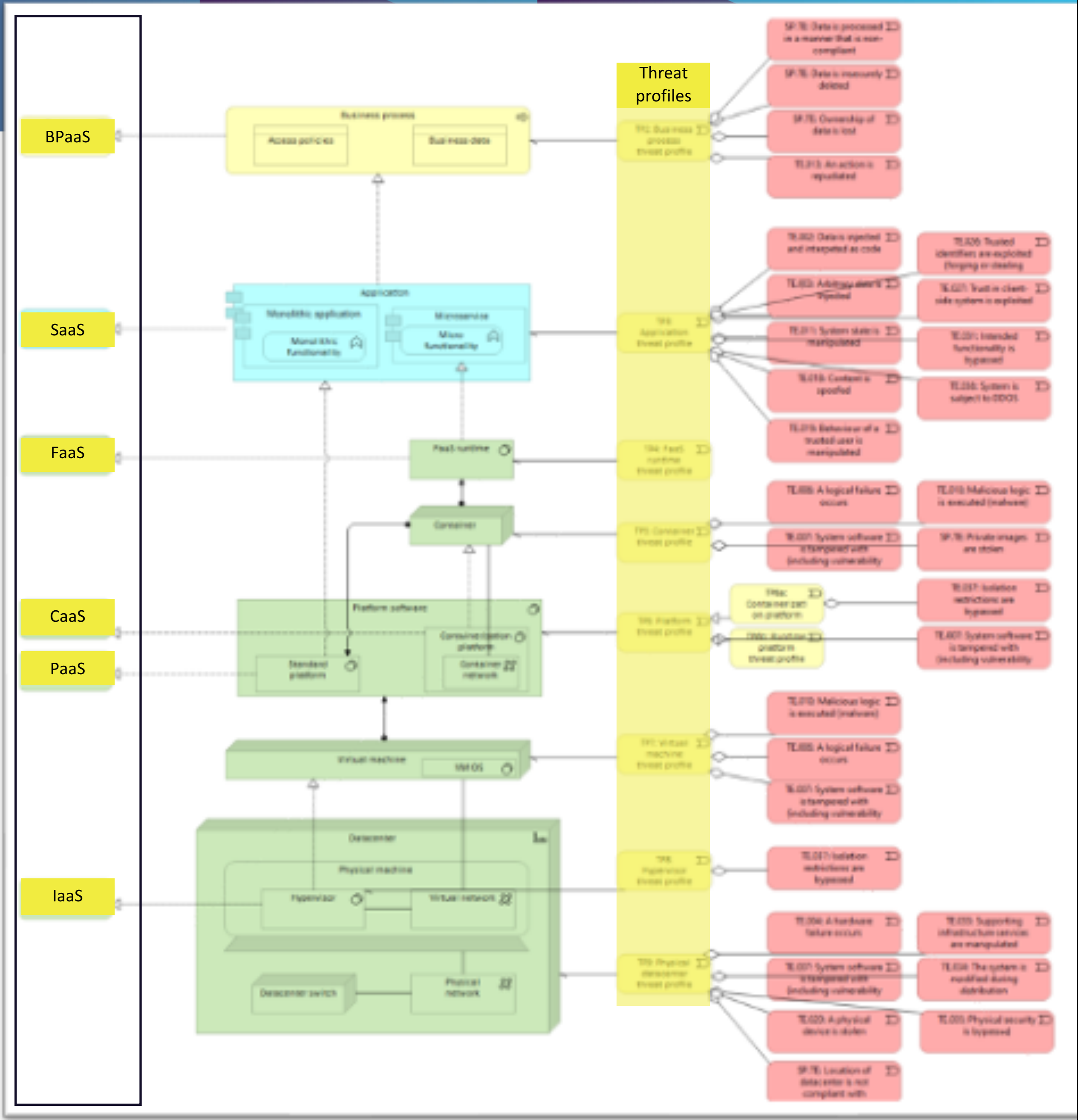
https://www.opensecurityarchitecture.org/cms/library/threat_catalogue

Step 2b: you need to identify threat events

- Use the CAPEC mechanisms of attack list as starting point
- Optionally cross-reference with CAWE catalog
- Analyze the threat in relation to the context model and add if applicable


System threat profile
TP1: Any system threat profile
TE.001: The resources of the system are exhausted
TE.005: The system is misconfigured
TE.012: Loss or compromise of logs
TE.014: The system operates in a manner that is non-compliant
TE.015: A threat laterally moves from an already compromised system to
TE.017: Identities are spoofed (e.g. via a stolen password or key)
TE.021: Traffic is intercepted
TE.023: Privileges are escalated
TE.024: Access control is bypassed
TE.025: User access rights are abused
TE.030: Input and/or output data is manipulated
TE.036: The system is brute forced
TE.039: High-privilege access rights are abused
TE.040: Insecure exposed interfaces are misused

Threats applicable to all systems



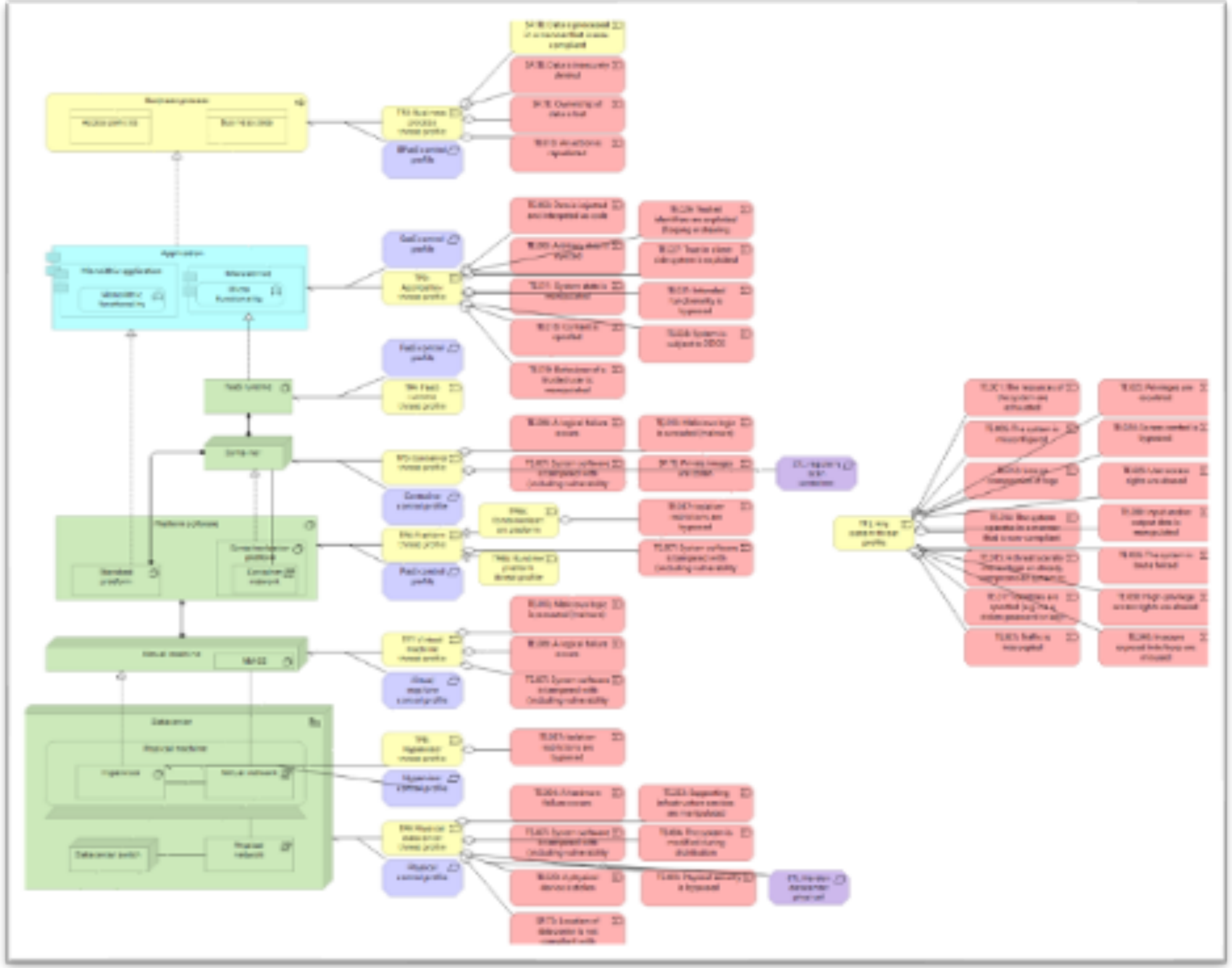
Threat model for the generic cloud architecture

Step 3: you need to identify controls

We use the following process for threat identification

- Controls can be bundled in control profiles
- Each threat profile can be linked to a control profile
- Depending on the service model chosen, **either you or the service provider is responsible for these controls** (and thus must be part of the contract)
- Threat actors in this exercise shift depending on the cloud service model chosen

Not all controls and control profiles have been added in this example model.



Example threat model with controls and control profiles

Demo using Archi: how to do this in practice

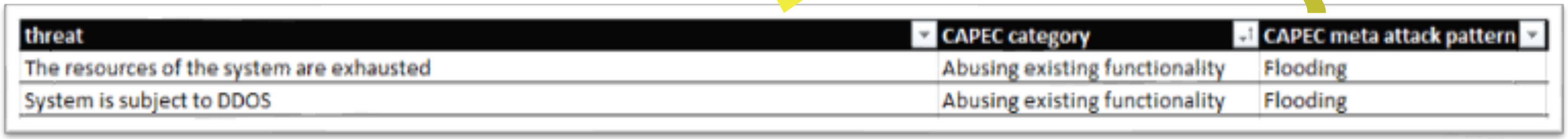
Demo: start from CAPEC mechanisms

- Browse the mechanisms of attack. This list contains:
  - Categories: this is a collection of attack patterns based on a common effect or a common attacker’s intent. It is not an actionable attack on its own.
  - Meta patterns: this is an abstract characterization of a specific methodology or technique used in an attack. A meta-attack is often void of a specific technology or implementation and is meant to provide an understanding of a high-level approach. **Meta level attack patterns are particularly useful for architecture and design level threat modeling exercises.**
  - Standard attack patterns: this is focused on a specific methodology or technique used in an attack.

Well, this is what we need.

These are very useful in solution threat modeling

We usually like to translate the meta patterns to organization-specific threats.

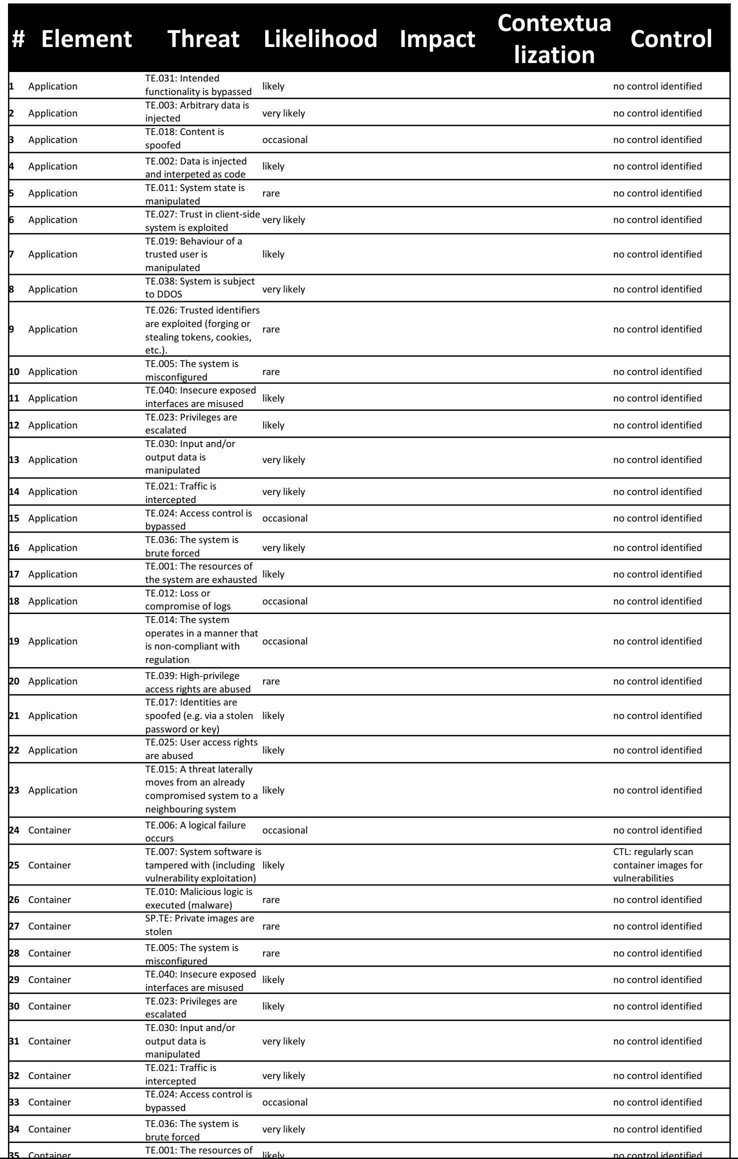


threat	CAPEC category	CAPEC meta attack pattern
The resources of the system are exhausted	Abusing existing functionality	Flooding
System is subject to DDOS	Abusing existing functionality	Flooding

Demo: using Archi as support tool

- Create a view for each step
- Drag and drop threat events and threat actors
- Automatically generate traceability matrix

Traceability matrix



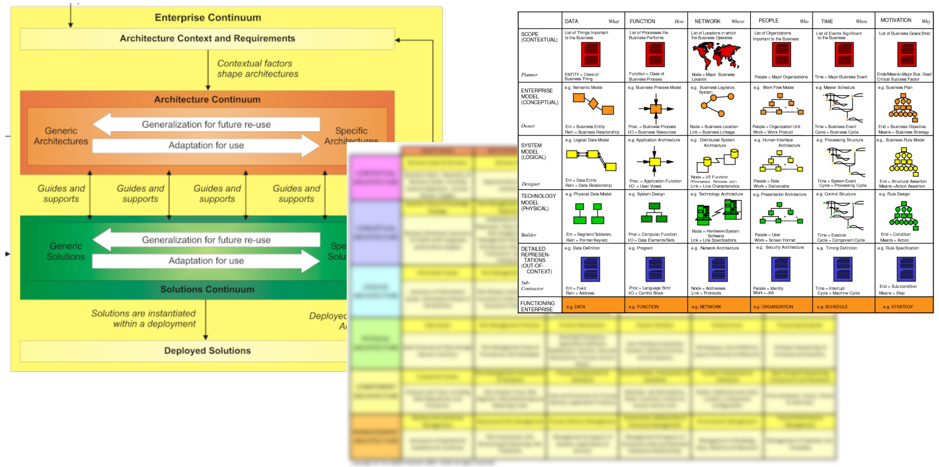
#	Element	Threat	Likelihood	Impact	Contextualization	Control
1	Application	TE.031: Intended functionality is bypassed	likely			no control identified
2	Application	TE.003: Arbitrary data is injected	very likely			no control identified
3	Application	TE.018: Content is spoofed	occasional			no control identified
4	Application	TE.002: Data is injected and interpeted as code	likely			no control identified
5	Application	TE.011: System state is manipulated	rare			no control identified
6	Application	TE.027: Trust in client-side system is exploited	very likely			no control identified
7	Application	TE.019: Behaviour of a trusted user is manipulated	likely			no control identified
8	Application	TE.038: System is subject to DDOS	very likely			no control identified
9	Application	TE.026: Trusted identifiers are exploited (forging or stealing tokens, cookies, etc.).	rare			no control identified
10	Application	TE.005: The system is misconfigured	rare			no control identified
11	Application	TE.040: Insecure exposed interfaces are misused	likely			no control identified
12	Application	TE.023: Privileges are escalated	likely			no control identified
13	Application	TE.030: Input and/or output data is manipulated	very likely			no control identified
14	Application	TE.021: Traffic is intercepted	very likely			no control identified
15	Application	TE.024: Access control is bypassed	occasional			no control identified
16	Application	TE.036: The system is brute forced	very likely			no control identified
17	Application	TE.001: The resources of the system are exhausted	likely			no control identified
18	Application	TE.012: Loss or compromise of logs	occasional			no control identified
19	Application	TE.014: The system operates in a manner that is non-compliant with regulation	occasional			no control identified
20	Application	TE.039: High-privilege access rights are abused	rare			no control identified
21	Application	TE.017: Identities are spoofed (e.g. via a stolen password or key)	likely			no control identified
22	Application	TE.025: User access rights are abused	likely			no control identified
23	Application	TE.015: A threat laterally moves from an already compromised system to a neighbouring system	likely			no control identified
24	Container	TE.006: A logical failure occurs	occasional			no control identified
25	Container	TE.007: System software is tampered with (including vulnerability exploitation)	likely			CTL: regularly scan container images for vulnerabilities
26	Container	TE.010: Malicious logic is executed (malware)	rare			no control identified
27	Container	SP.TE: Private images are stolen	rare			no control identified
28	Container	TE.005: The system is misconfigured	rare			no control identified
29	Container	TE.040: Insecure exposed interfaces are misused	likely			no control identified
30	Container	TE.023: Privileges are escalated	likely			no control identified
31	Container	TE.030: Input and/or output data is manipulated	very likely			no control identified
32	Container	TE.021: Traffic is intercepted	very likely			no control identified
33	Container	TE.024: Access control is bypassed	occasional			no control identified
34	Container	TE.036: The system is brute forced	very likely			no control identified
35	Container	TE.001: The resources of	likely			no control identified

OWASP 2022
VIRTUAL
APPSEC
JUN 6-10

Conclusions & pitfalls

Main conclusions

1. Layering



In this talk we focused on the architectural layer.

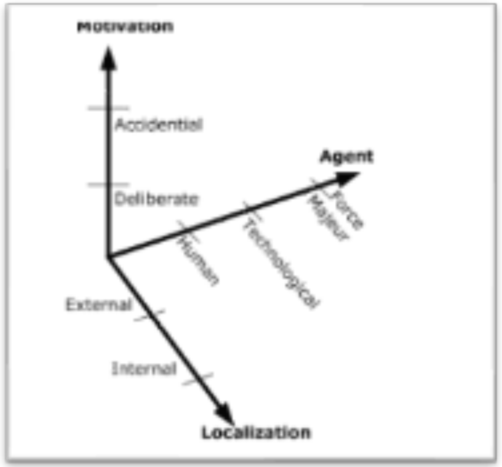
2. Comparison with solution threat modeling – the same, but different

Similar methodology (but stricter)	Context modeling	Threat identification	Managing controls
Different techniques	ArchiMate & TOGAF	OSA and MITRE CAPEC	Future work: integrate with OSCAL
Different scope	Enterprise architecture deliverables – not detailed designs.		
Different goals	Security principles & objectives. Traceability to security requirements.		

Facilitating threat modeling for Enterprise Security Architects.

Common pitfalls to avoid

- Overlapping threats: threats within the same catalog or across catalogs may overlap, leading to duplicates. Avoid by tracking related threats.
- Missing generalizations: many threats are based on very detailed attacks. As Enterprise Security Architect you must attempt to generalize (e.g., not ‘XSS’ but rather ‘Input/output manipulation’).
- Missing threats: MITRE CAPEC mainly lists human threats. You may miss technology threats (e.g. growing complexity) and force majeure threats (e.g. earthquakes). Avoid by adding these threats to your default threat catalog up front – they are usually limited in number.
- Bad prioritization: prioritization of threats is key. At architectural level, risk prioritization techniques can be reused (e.g., FAIR).
- Paralysis by analysis: security experts generally have a deep understanding of technology and tend to become paralyzed by analysis. Avoid by communicating with a business minded person.
- Overly focus on differences between solution threat modeling and architectural threat modeling. You will see it when you need it (reference architectures, patterns, etc.).



Threat classification method

https://www.opensecurityarchitecture.org/cms/library/threat_catalogue

OWASP 2022 VIRTUAL APPSEC

JUN 6-10

THANK YOU!