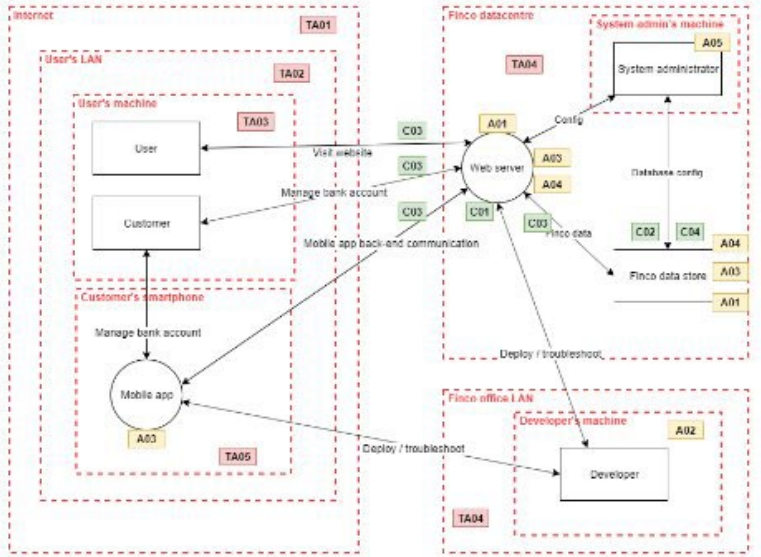

THREAT MODELING
CONNECT
POWERED BY
IRIUSRISK
THREAT
MODCON
2024
LISBON
ADVANCING THREAT MODELING CAPABILITIES TOGETHER

Layered Threat Modeling

By Roos Hubrechtsen & Michael Boeynaems

Solution threat modeling

- Created during the design or build phase of the SSDLC
- Focused on a **single** solution.
- Notations: DFDs, UML diagrams, …
- Techniques: STRIDE, LINDDUN, …



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

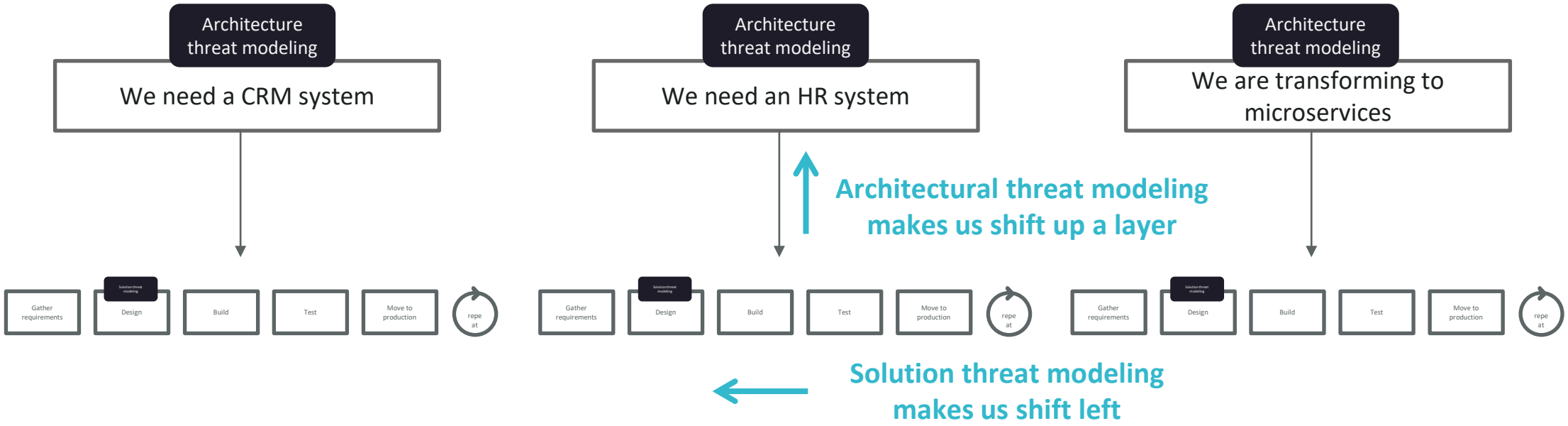
Threat Actors	
ID	Description
TA01	Unauthenticated external user (Internet attacker)
TA02	Unauthenticated internal user (LAN attacker)
TA03	Malicious customer
TA04	Malicious employee
TA05	Attacker with jail-broken device

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

Gather requirements → Design (Solution threat modeling) → Build → Test → Move to production → repeat

Architectural threat modeling

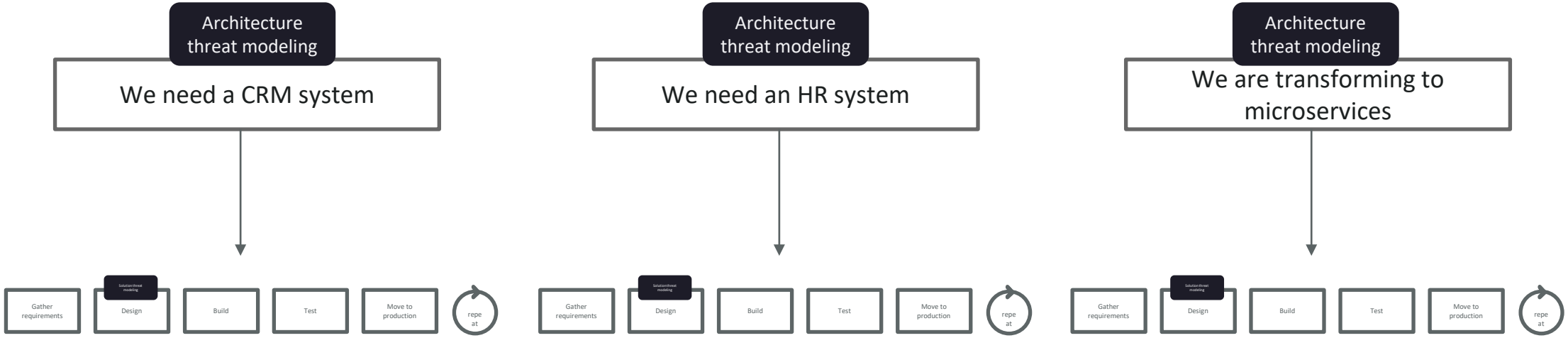
- Created before an SSDLC even starts;
- Higher level than solution threat modeling;
- Focused on concepts, not solutions.



Architectural threat modeling - **WHY**

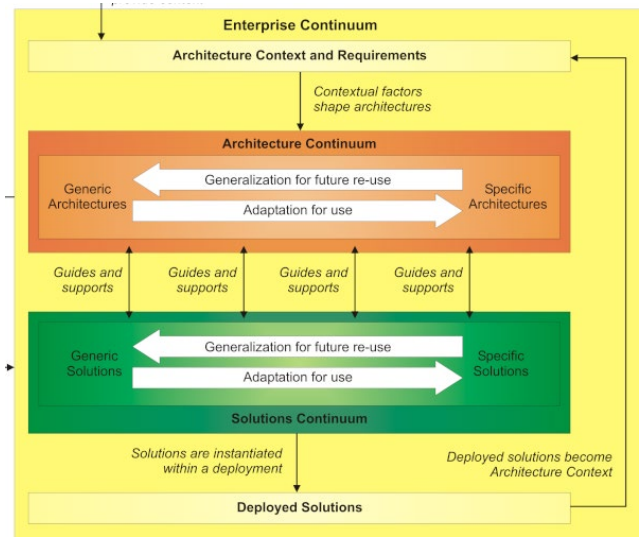
- Identify transversal threats;
- Identify threats before a project even starts;
- Identify threats that impact multiple solutions;

We argue you also should threat model the (enterprise) architecture layer and we will show an example of how it is done.

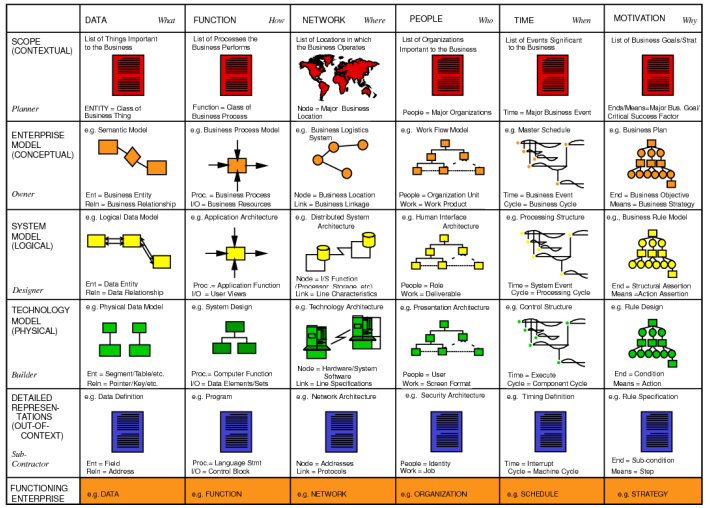


Layers

The notion of layering is widely accepted by different frameworks.

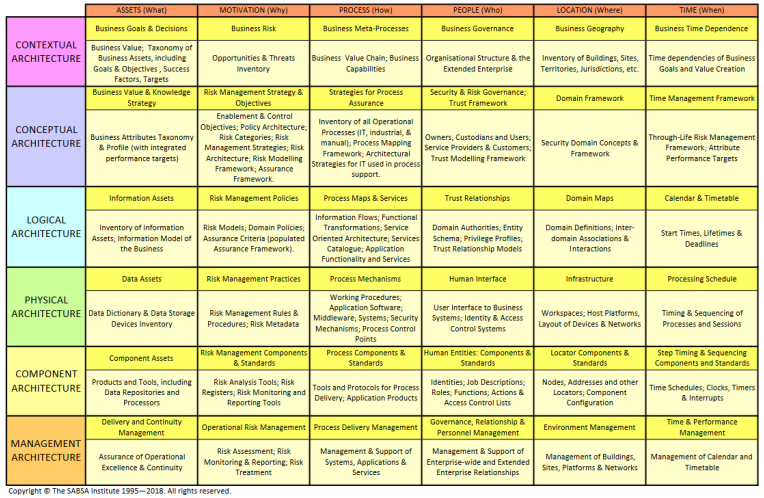


TOGAF



	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 Rein = 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 Rein = 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. Rein = 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 Rein = 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



	ASSETS (What)	MOTIVATION (Why)	PROCESS (How)	PEOPLE (Who)	LOCATION (Where)	TIME (When)
CONTEXTUAL ARCHITECTURE	Business Goals & Decisions	Business Risk	Business Meta-Processes	Business Governance	Business Geography	Business Time Dependence
	Business Value; Taxonomy of Business Assets, including Goals & Objectives , Success Factors, Targets	Opportunities & Threats Inventory	Business Value Chain; Business Capabilities	Organisational Structure & the Extended Enterprise	Inventory of Buildings, Sites, Territories, Jurisdictions, etc.	Time dependencies of Business Goals and Value Creation
CONCEPTUAL ARCHITECTURE	Business Value & Knowledge Strategy	Risk Management Strategy & Objectives	Strategies for Process Assurance	Security & Risk Governance; Trust Framework	Domain Framework	Time Management Framework
	Business Attributes Taxonomy & Profile (with integrated performance targets)	Enablement & Control Objectives; Policy Architecture; Risk Categories; Risk Management Strategies; Risk Architecture; Risk Modelling Framework; Assurance Framework.	Inventory of all Operational Processes (IT, industrial, & manual); Process Mapping Framework; Architectural Strategies for IT used in process support.	Owners, Custodians and Users; Service Providers & Customers; Trust Modelling Framework	Security Domain Concepts & Framework	Through-Life Risk Management Framework; Attribute Performance Targets
LOGICAL ARCHITECTURE	Information Assets	Risk Management Policies	Process Maps & Services	Trust Relationships	Domain Maps	Calendar & Timetable
	Inventory of Information Assets; Information Model of the Business	Risk Models; Domain Policies; Assurance Criteria (populated Assurance Framework).	Information Flows; Functional Transformations; Service Oriented Architecture; Services Catalogue; Application Functionality and Services	Domain Authorities; Entity Schema; Privilege Profiles; Trust Relationship Models	Domain Definitions; Inter-domain Associations & Interactions	Start Times, Lifetimes & Deadlines
PHYSICAL ARCHITECTURE	Data Assets	Risk Management Practices	Process Mechanisms	Human Interface	Infrastructure	Processing Schedule
	Data Dictionary & Data Storage Devices Inventory	Risk Management Rules & Procedures; Risk Metadata	Working Procedures; Application Software; Middleware; Systems; Security Mechanisms; Process Control Points	User Interface to Business Systems; Identity & Access Control Systems	Workspaces; Host Platforms, Layout of Devices & Networks	Timing & Sequencing of Processes and Sessions
COMPONENT ARCHITECTURE	Component Assets	Risk Management Components & Standards	Process Components & Standards	Human Entities: Components & Standards	Locator Components & Standards	Step Timing & Sequencing Components and Standards
	Products and Tools, including Data Repositories and Processors	Risk Analysis Tools; Risk Registers; Risk Monitoring and Reporting Tools	Tools and Protocols for Process Delivery; Application Products	Identities; Job Descriptions; Roles; Functions; Actions & Access Control Lists	Nodes, Addresses and other Locators; Component Configuration	Time Schedules; Clocks, Timers & Interrupts
MANAGEMENT ARCHITECTURE	Delivery and Continuity Management	Operational Risk Management	Process Delivery Management	Governance, Relationship & Personnel Management	Environment Management	Time & Performance Management
	Assurance of Operational Excellence & Continuity	Risk Assessment; Risk Monitoring & Reporting; Risk Treatment	Management & Support of Systems, Applications & Services	Management & Support of Enterprise-wide and Extended Enterprise Relationships	Management of Buildings, Sites, Platforms & Networks	Management of Calendar and Timetable

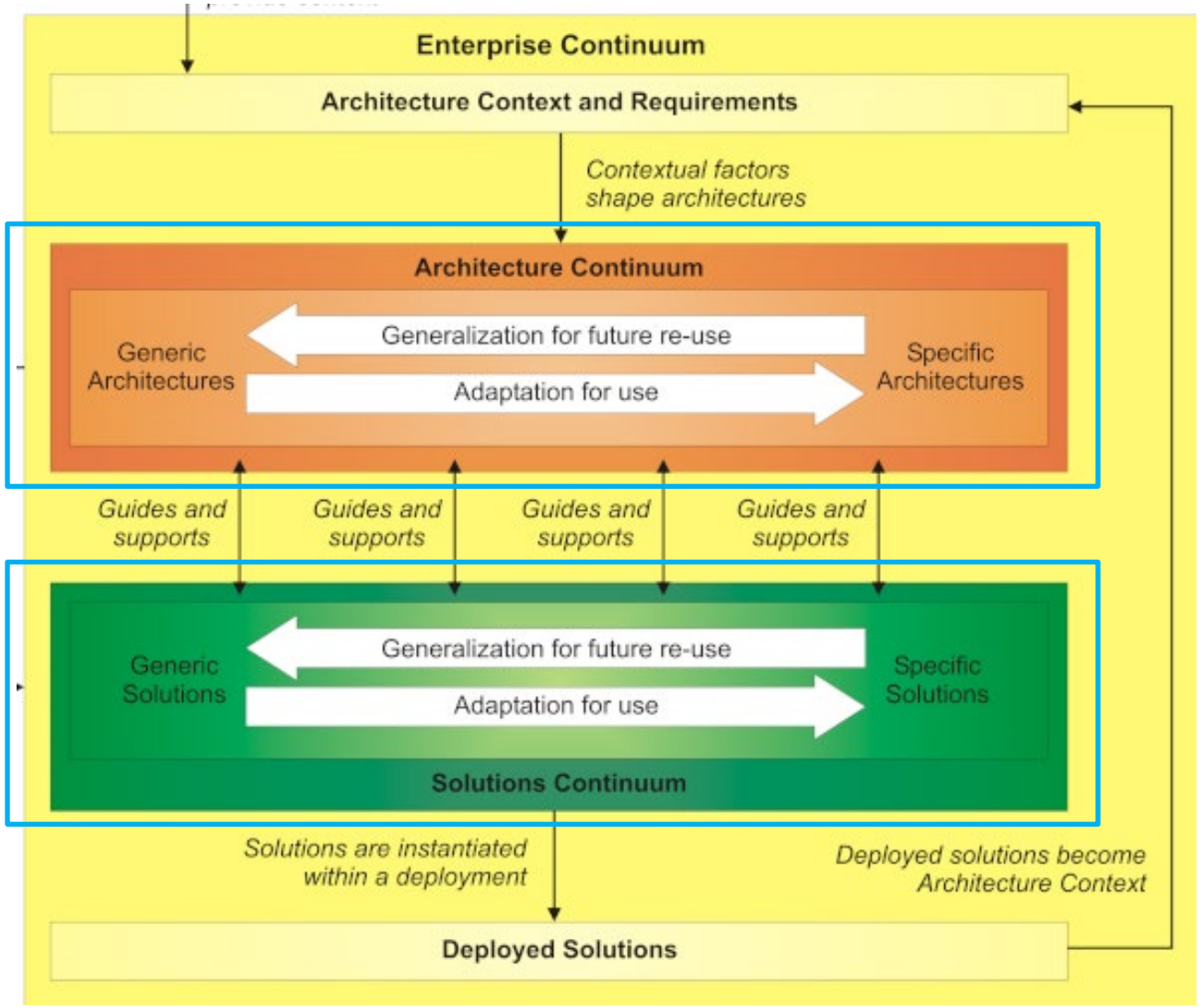
SABSA

Architecture layer

Example: a rocket
Example: BIAN

Solutions layer

Example: the Starship Megarocket
Example: Santander's solution architecture

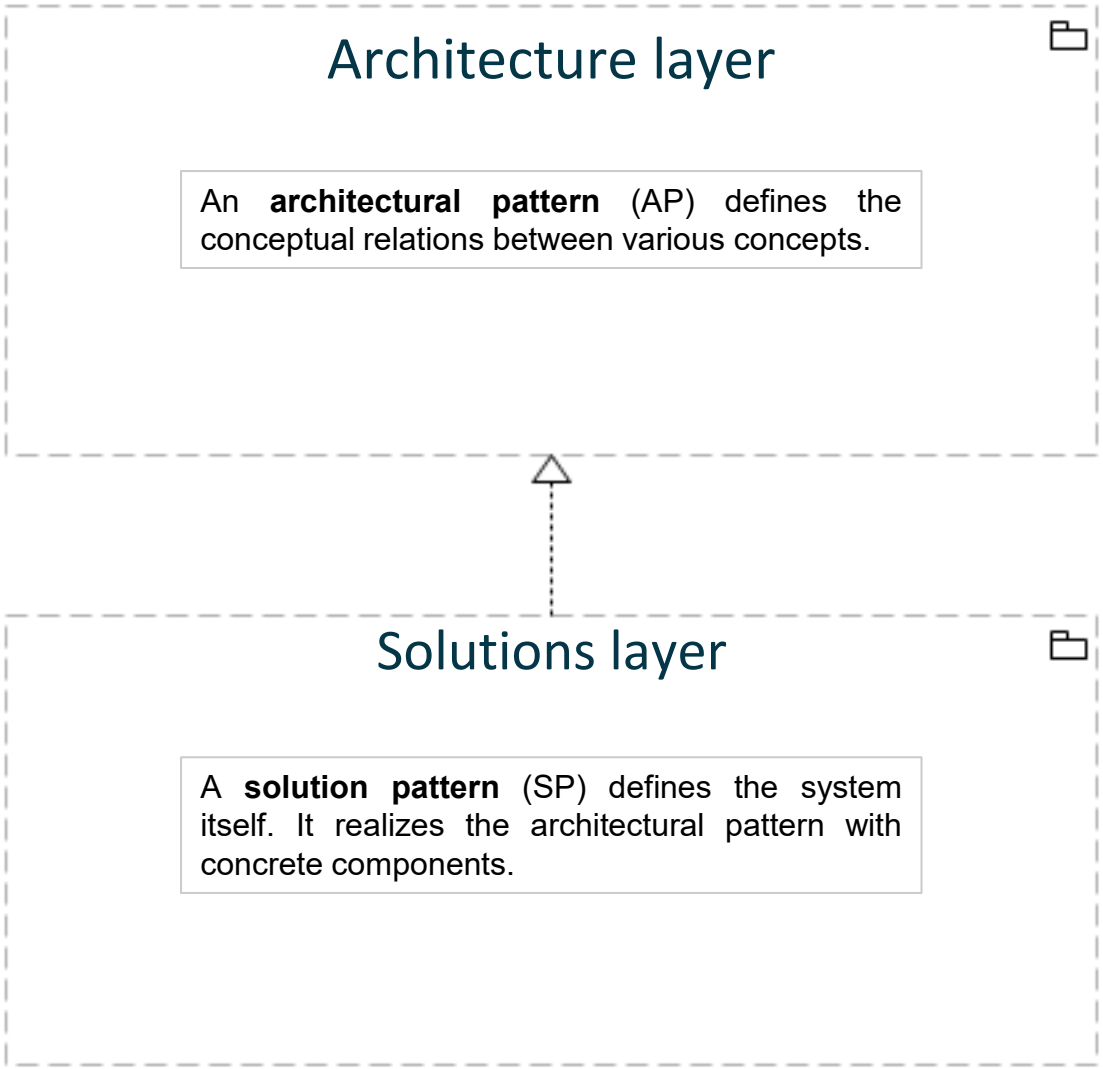


Demonstration.

Demonstration: threat modeling a pattern

Pattern (noun)

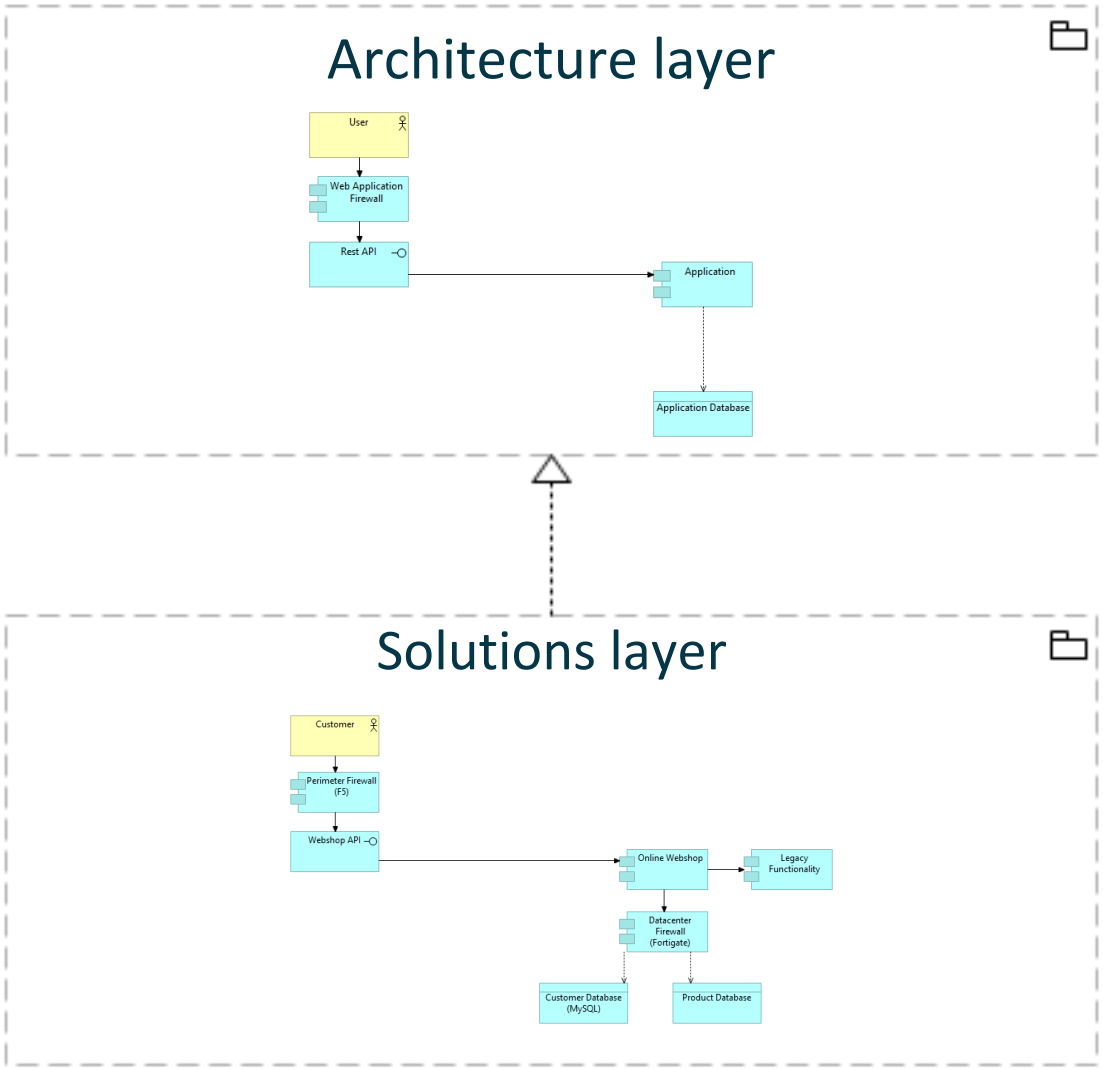
A template describing a generic solution to a problem that occurs frequently in a given context (TOGAF 9).

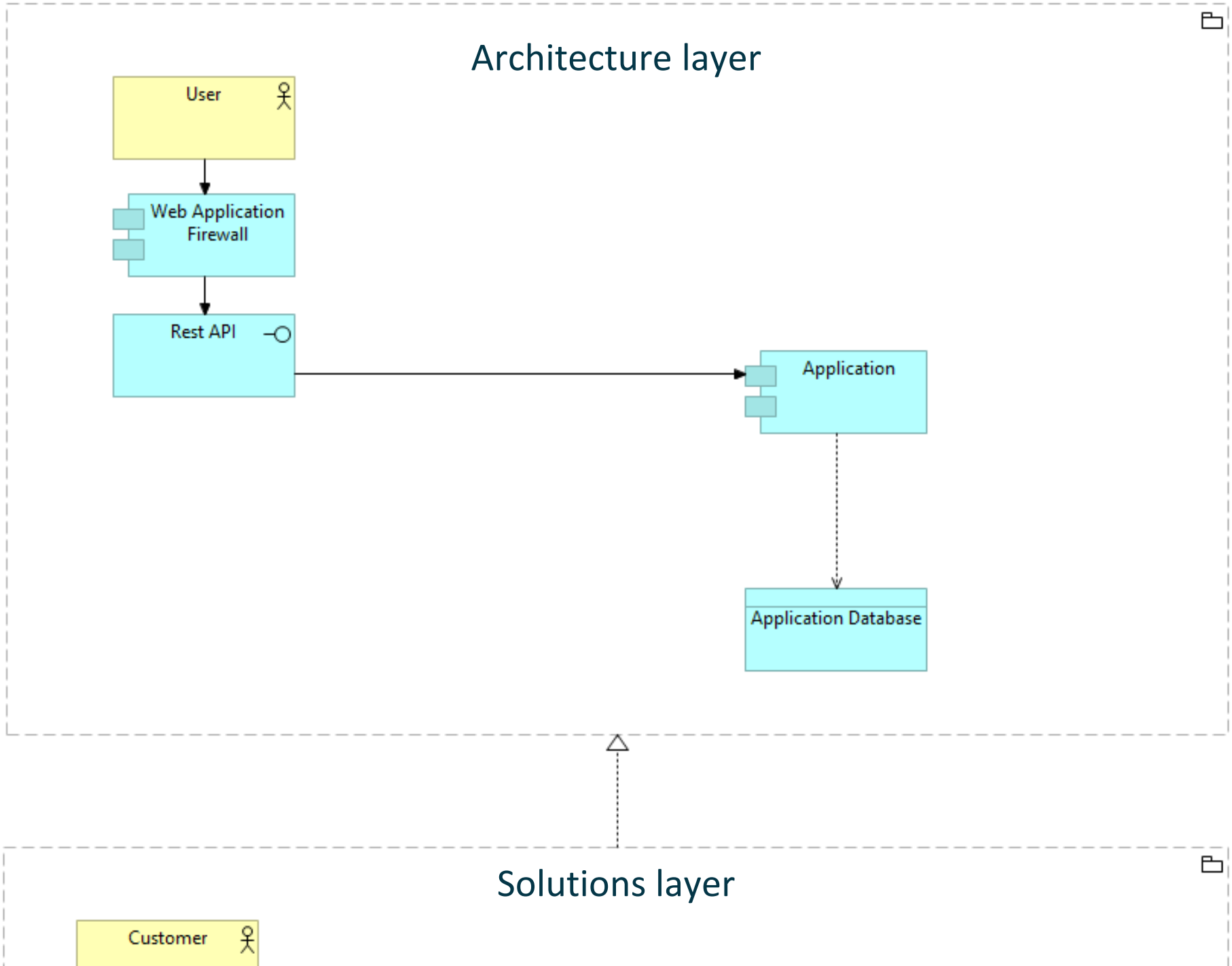


Demonstration: threat modeling a pattern

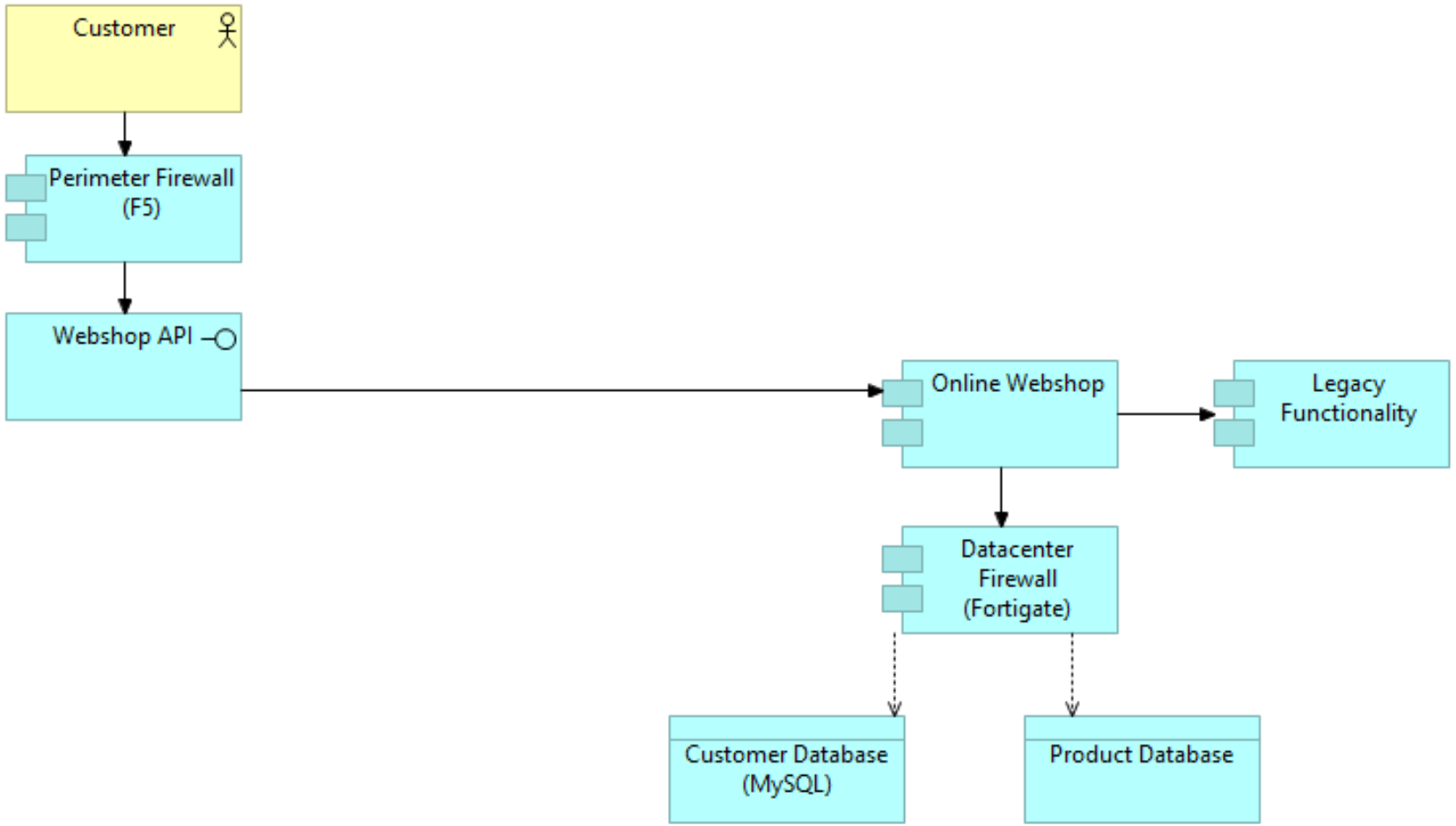
Pattern (noun)

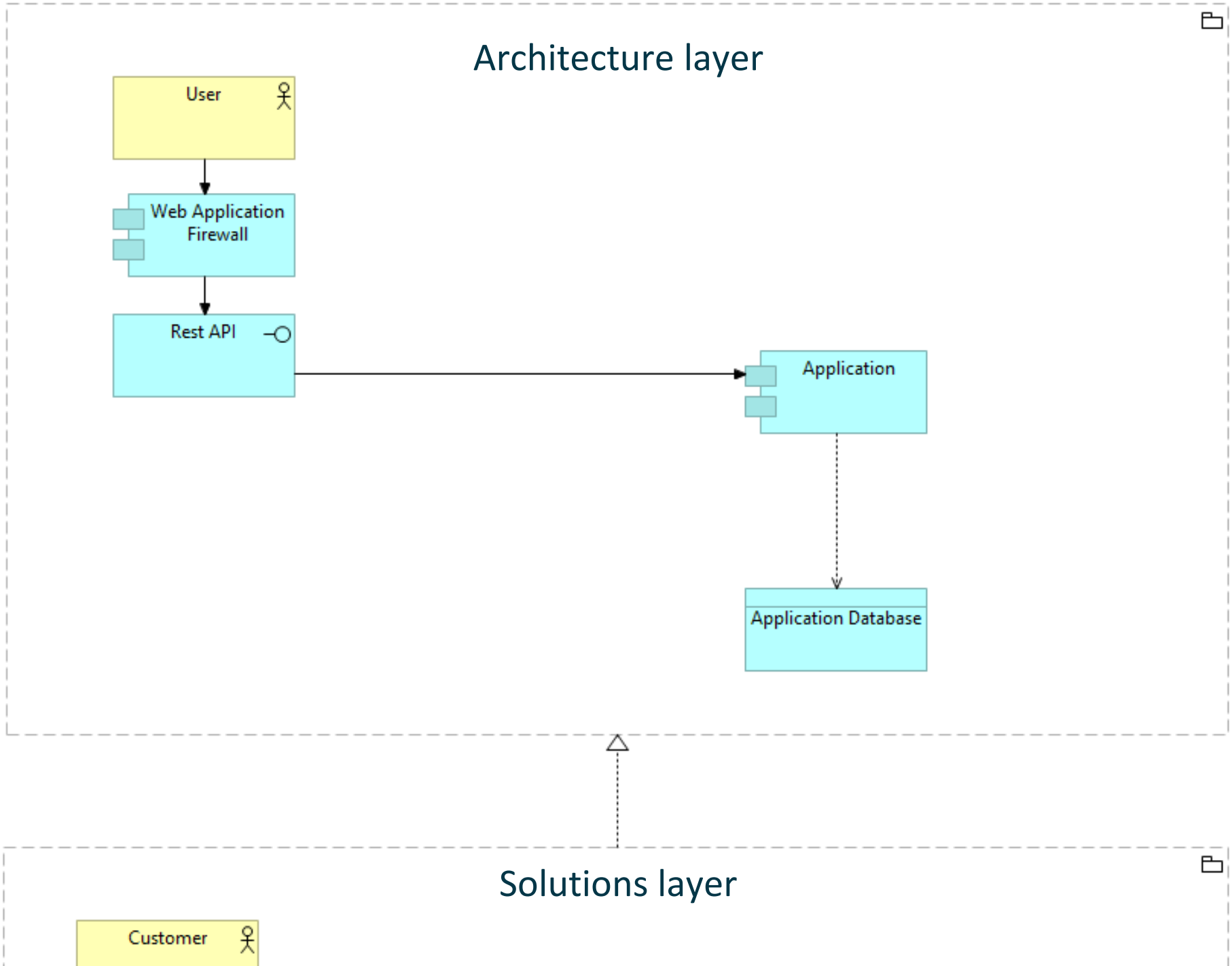
A template describing a generic solution to a problem that occurs frequently in a given context (TOGAF 9).

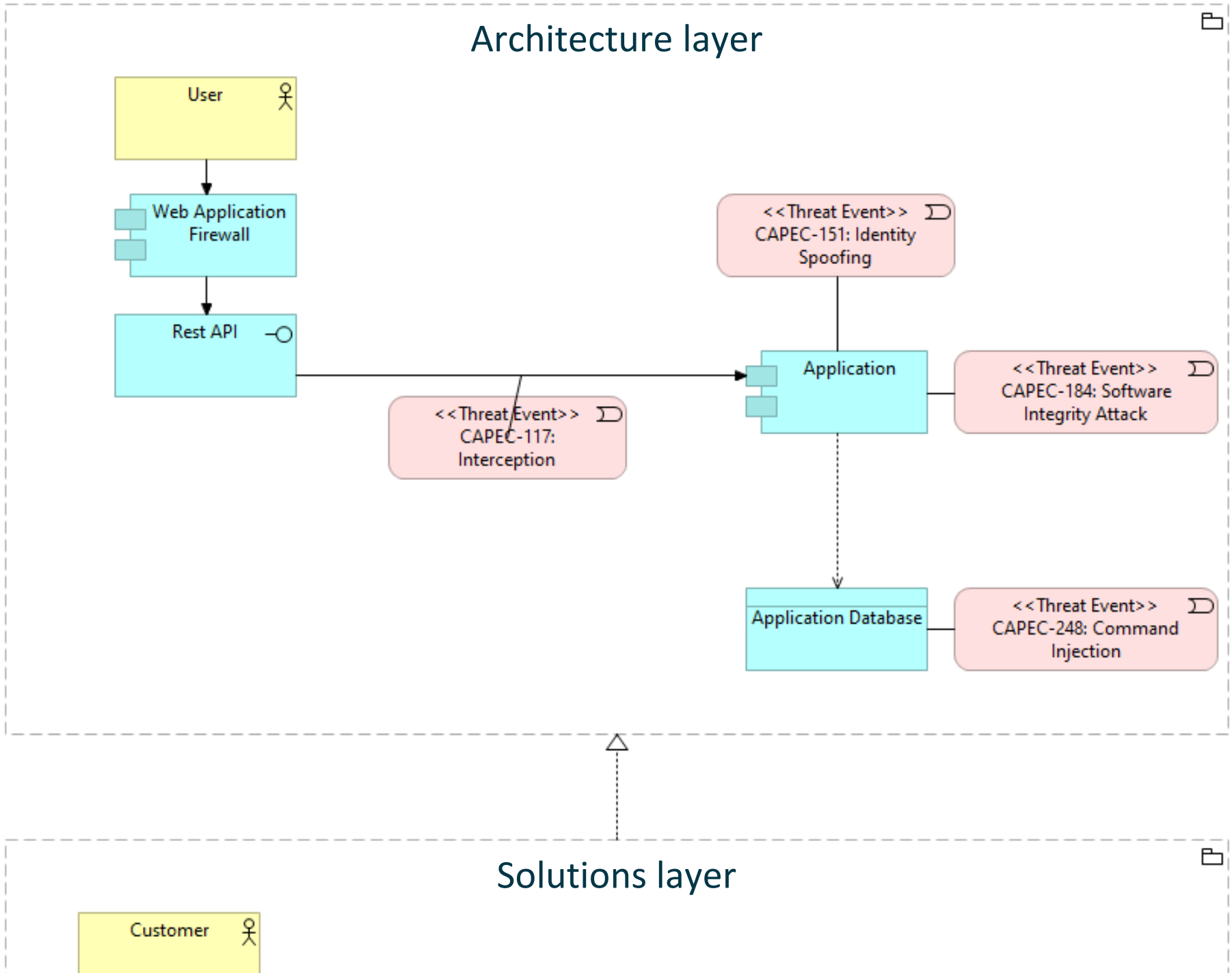


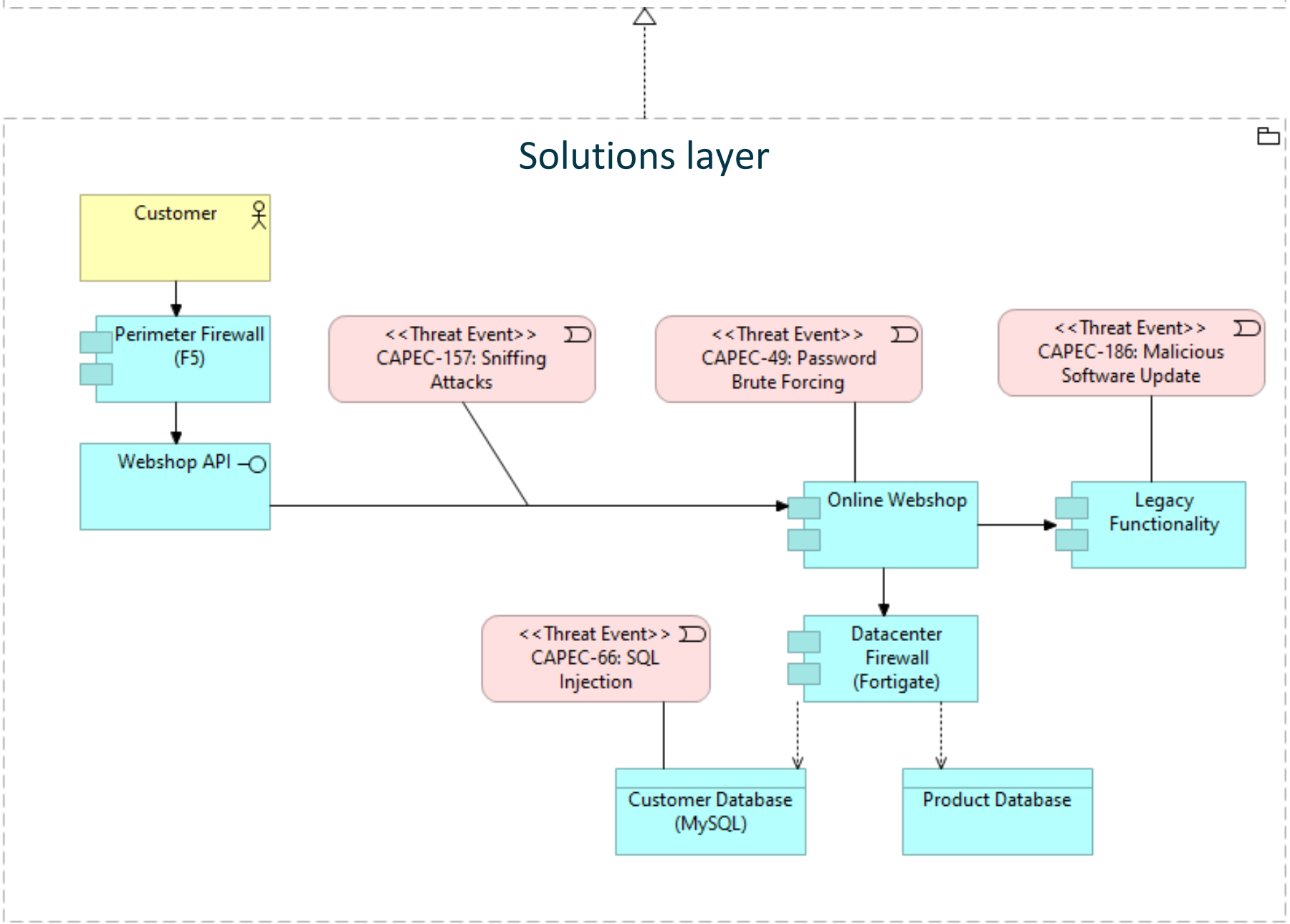

Architecture layer
User
Web Application Firewall
Rest API
Application
Application Database
Solutions layer
Customer

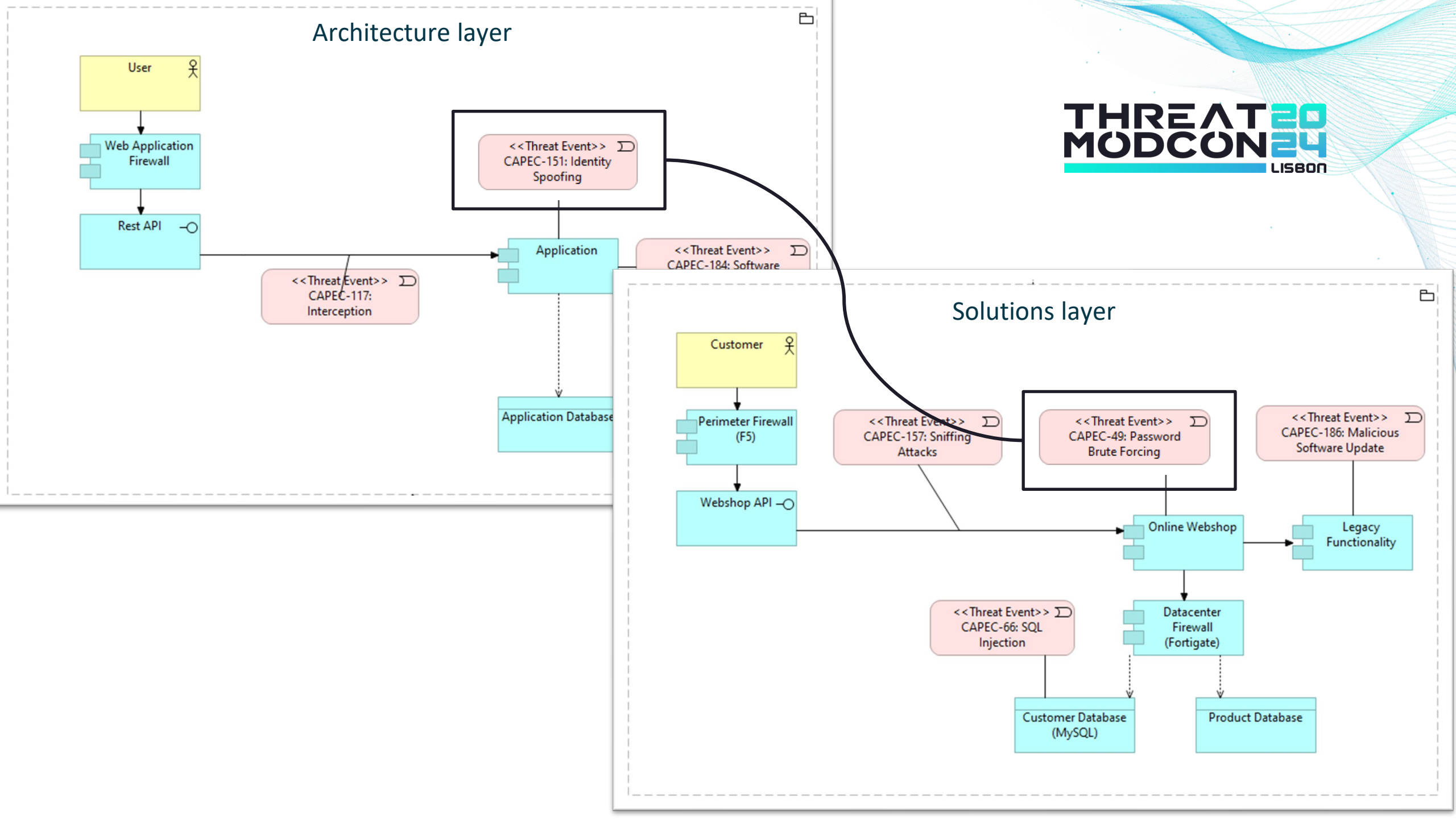
Solutions layer




Architecture layer
User
Web Application Firewall
Rest API
Application
Application Database
Solutions layer
Customer


Architecture layer
User
Web Application Firewall
Rest API
<<Threat Event>>
CAPEC-151: Identity Spoofing
<<Threat Event>>
CAPEC-117: Interception
Application
<<Threat Event>>
CAPEC-184: Software Integrity Attack
Application Database
<<Threat Event>>
CAPEC-248: Command Injection
Solutions layer
Customer


Injection
Solutions layer
Customer
Perimeter Firewall (F5)
Webshop API
<<Threat Event>> CAPEC-157: Sniffing Attacks
<<Threat Event>> CAPEC-49: Password Brute Forcing
<<Threat Event>> CAPEC-186: Malicious Software Update
Online Webshop
Legacy Functionality
<<Threat Event>> CAPEC-66: SQL Injection
Datacenter Firewall (Fortigate)
Customer Database (MySQL)
Product Database


Architecture layer
User
Web Application Firewall
Rest API
<<Threat Event>>
CAPEC-151: Identity Spoofing
Application
<<Threat Event>>
CAPEC-184: Software
<<Threat Event>>
CAPEC-117: Interception
Application Database
Solutions layer
Customer
Perimeter Firewall (F5)
Webshop API
<<Threat Event>>
CAPEC-157: Sniffing Attacks
<<Threat Event>>
CAPEC-49: Password Brute Forcing
<<Threat Event>>
CAPEC-186: Malicious Software Update
Online Webshop
Legacy Functionality
<<Threat Event>>
CAPEC-66: SQL Injection
Datacenter Firewall (Fortigate)
Customer Database (MySQL)
Product Database

Architectural vs Solution threats

CAPEC-151: Identity Spoofing

Attack Pattern ID: 151
Abstraction: Meta

View customized information: Conceptual | Operational | Mapping-Friendly | **Complete**

Description

Identity Spoofing refers to the action of assuming (i.e., taking on) the identity of some other entity (human or non-human) and then using that identity to accomplish a goal. An adversary may craft messages that appear to come from a different principle or use stolen / spoofed authentication credentials.

Extended Description

Alternatively, an adversary may intercept a message from a legitimate sender and attempt to make it look like the message comes from them without changing its

Architectural threat

wishes to change what the message says. In an Identity Spoofing attack, the adversary is attempting to change the identity of the content.

- Likelihood Of Attack
- Typical Severity
- Relationships
- Prerequisites
- Resources Required
- Consequences
- Mitigations
- Related Weaknesses
- Content History

CAPEC-49: Password Brute Forcing

Attack Pattern ID: 49
Abstraction: Standard

View customized information: Conceptual | Operational | Mapping-Friendly | **Complete**

Description

An adversary tries every possible value for a password until they succeed. A brute force attack, if feasible computationally, will always be successful because it will essentially go through all possible passwords given the alphabet used (lower case letters, upper case letters, numbers, symbols, etc.) and the maximum length of the password.

Extended Description

Solutions threat

passwords must be of a certain level, there is no need to check smaller candidates.

- Likelihood Of Attack
- Typical Severity
- Relationships
- Execution Flow
- Prerequisites
- Skills Required
- Resources Required
- Indicators
- Consequences
- Mitigations

Architectural vs Solution threats

THREAT MODCON 2024 LISBON

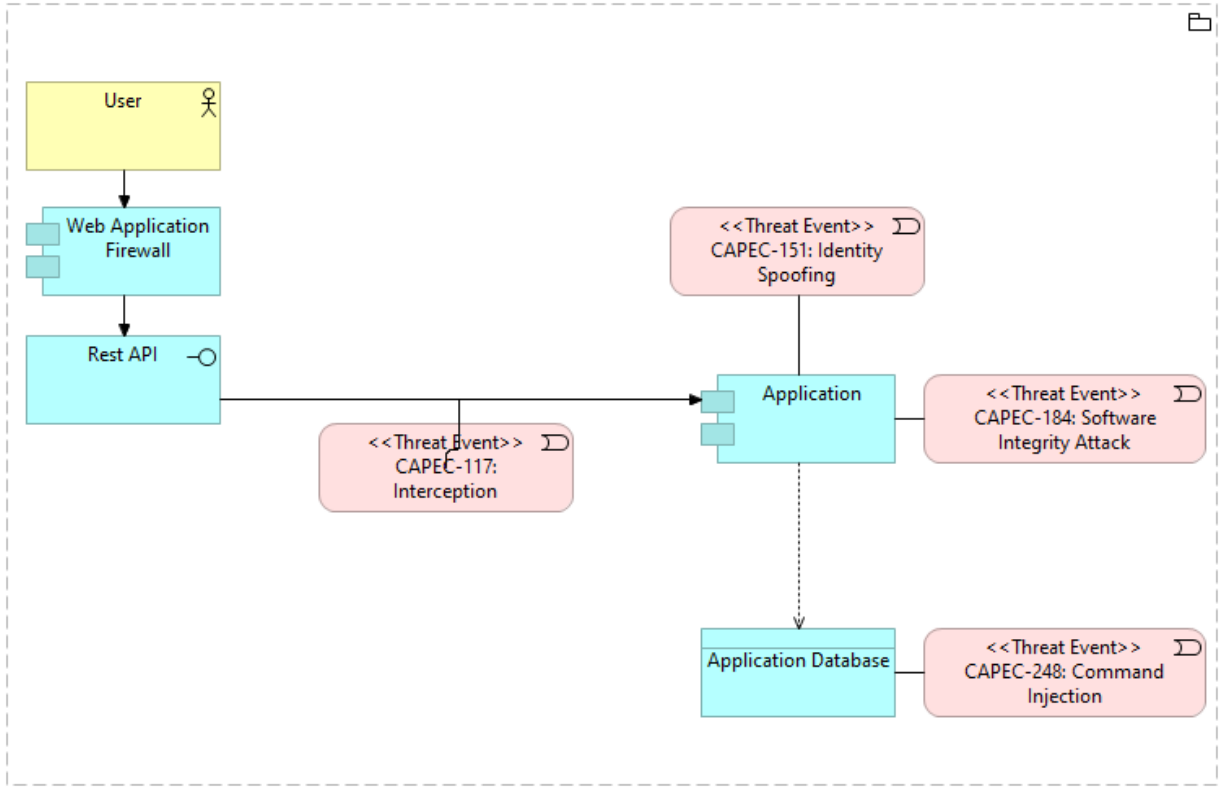
Architectural threat

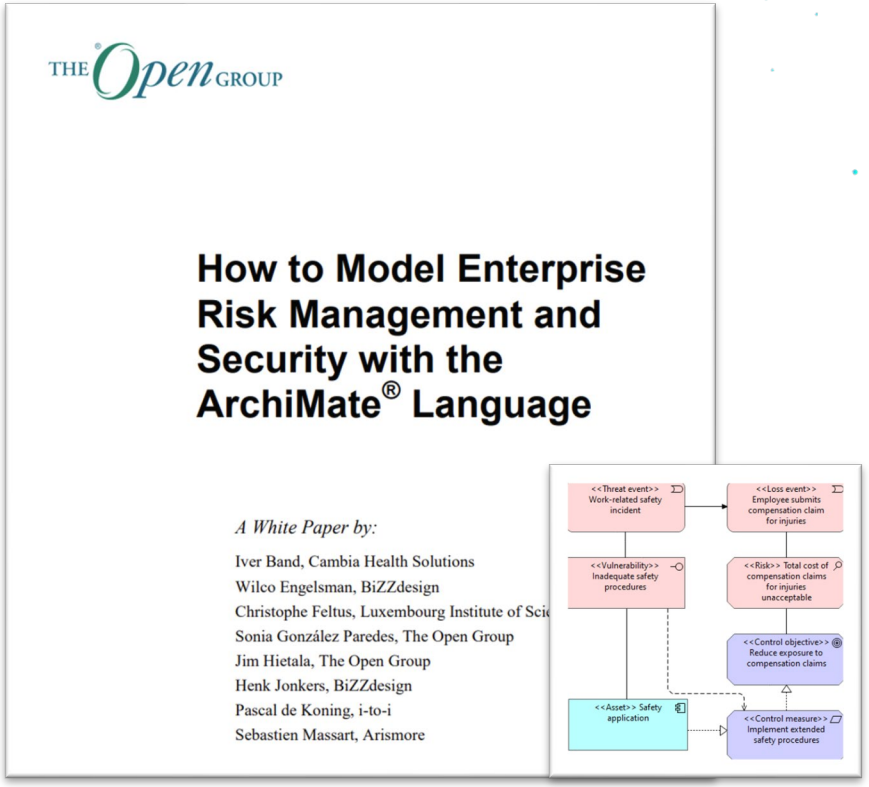
- CAPEC Meta Attack patterns
- BSI Elementary threats
- STRIDE
- ‘Architectural Risk Assessment’
- …

Solutions threat

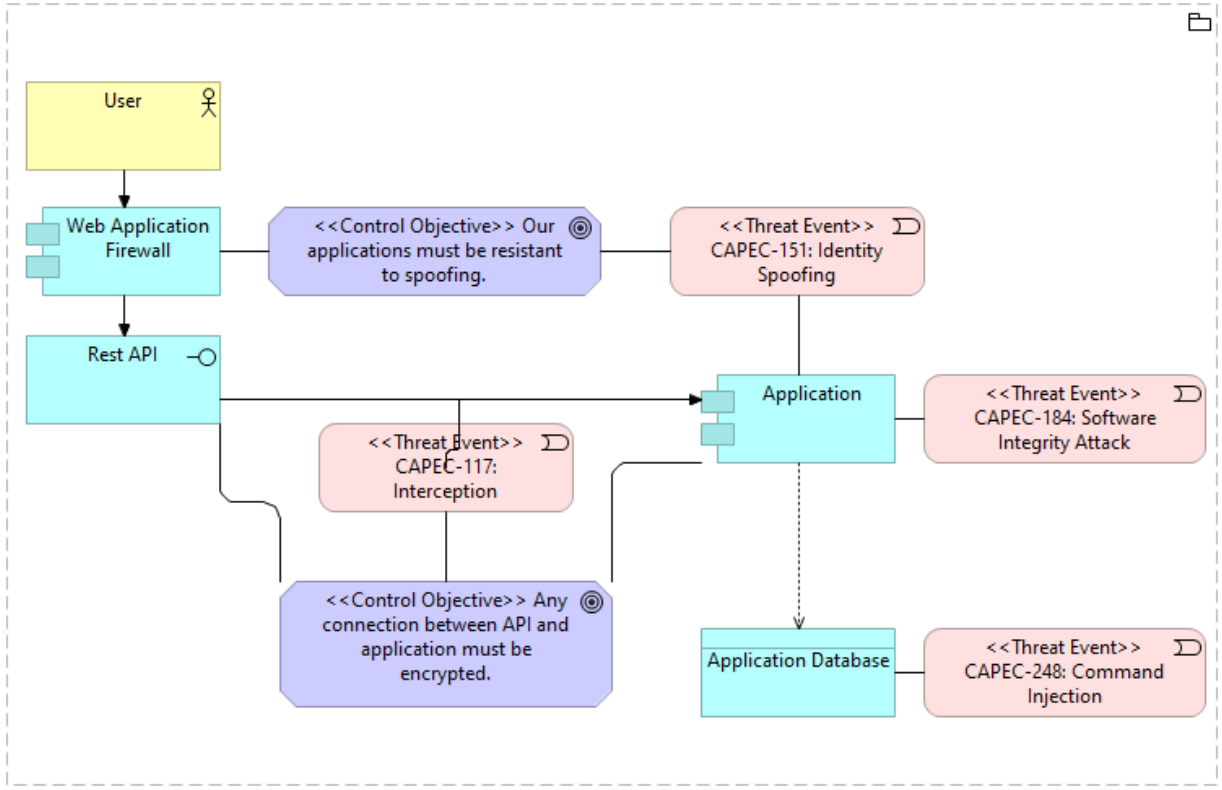
- CAPEC Standard Attack patterns
- STRIDE
- OWASP Cornucopia
- ‘Solution Threat Model’
- …

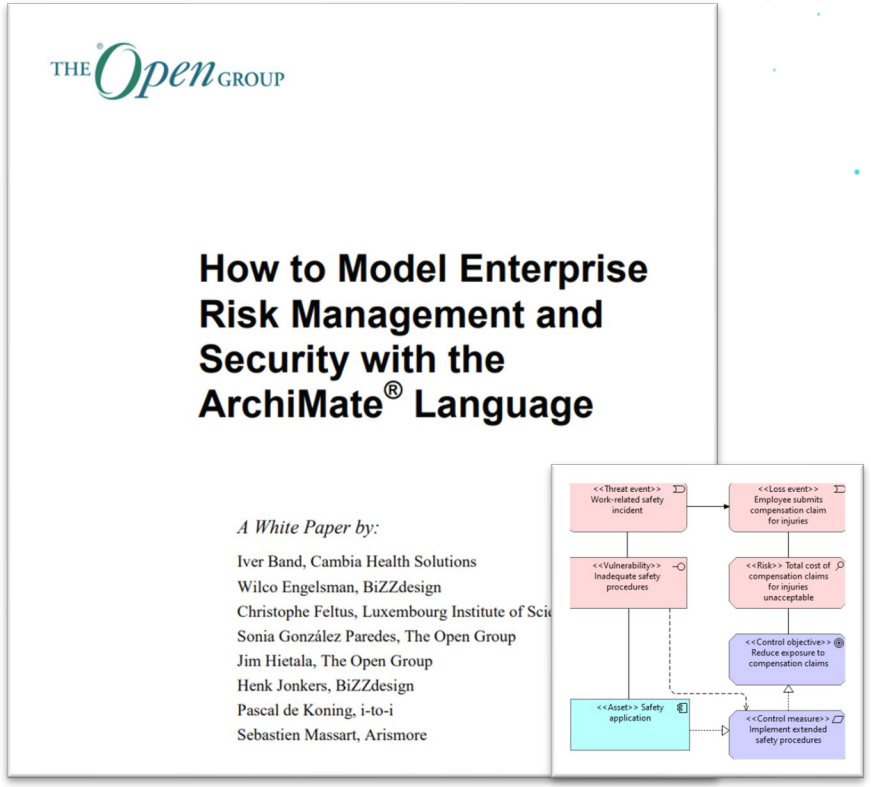
We adopt the ArchiMate risk overlay





We adopt the ArchiMate risk overlay





Thank you

- Threat model at architecture layer
- Use more generic threats at architecture layer
- Reuse notation, tooling, and frameworks

Time for Q&A