Security Analysis of Large-Scale Computer Systems (EP2790) / Cyber Security Analysis (EP279V)

Lecture slides

Agenda, lecture 1

- Background and motivation
  - System architectures and their managment
  - Risk assessment
  - Threat modeling
  - Yet another cybersecurity risk assessment method (Yacraf)

- Course design
  - Material and philosophy

Overall goal of the course

(Learn a methodology, *threat modeling*, to)
assess cybersecurity risk of (large-scale) computer systems.

Brave new digital world...

...what could possibly go wrong?

Cybersecurity risk

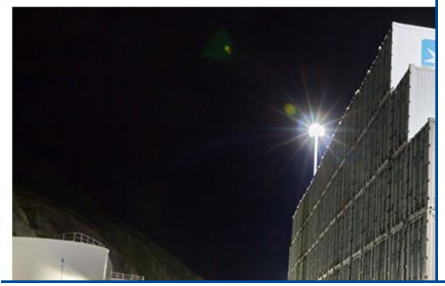
- Cyber crime is estimated to cost 30 Billion SEK in Sweden 2021. (Stockholms Handelskammare)
- 28% of surveyed Swedish companies were exposed to a cyberattack latest 12 months (2023) (Svenskt näringsliv)

When the screens went black: How NotPetya taught Maersk to rely on resilience – not luck – to mitigate future cyber-attacks

Adam Bannister 09 December 2019 at 12:09 UTC
Updated: 09 December 2019 at 13:06 UTC

Ransomware | Cyber-attacks | Maritime

Serendipity intervened to rescue world's largest shipping conglor



KIM ZETTER 11.03.14 06:30 AM

An Unprecedented Look at Stuxnet, the World's First Digital Weapon

Ukraine power cut 'was cyber-attack'

11 January 2017

Bangladesh Bank Attackers Hacked SWIFT Software

Attackers Used Malware to Steal $81 Million, BAE Systems Says

Mathew J. Schwartz (euroinfosec) • April 25, 2016

Twitter | Facebook | LinkedIn | Credit Eligible | Get Permission

CYBERSECURITY DIVE Sign up

Salt Typhoon telecom hacks one of the most consequential campaigns against US ever, expert says

A prominent former member of a recently shuttered cyber-incide
said the board should be reconstituted with independent authority

Published May 1, 2025

David Jones
Reporter

BBC

JLR hack is costliest cyber attack in UK history, say analysts

8 hours ago Share Save

Joe Tidy
Cyber correspondent, BBC World Service

The CMC estimated the damage to be in the range of £1.6bn and £2.1bn but predicted the most likely cost will be £1.9bn.

million fro
he bank's SV

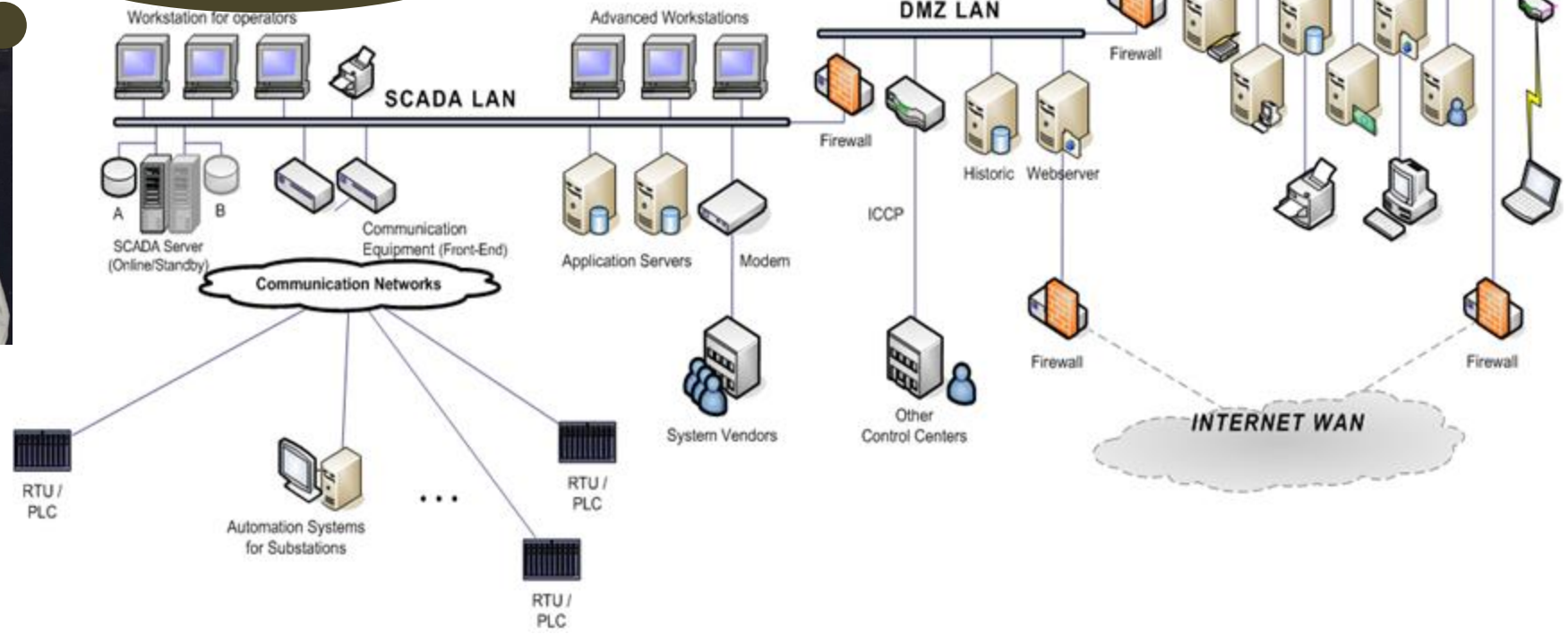
'Shocking' hack of psychotherapy records in Finland affects thousands

Distressed patients flood support services after hack of private firm Vastaamo

Cybersecurity management is difficult!

- Is my IT system secure?
- ..And what exactly makes the system as a whole (in)secure?

security architect (etc.)



Hey.., actually, we know what to do!

- Firewalls
- Operational intrusion detection
- Patch management
- Security awareness
- Logging (forensics)
- Hardening
- Access control
- Encryption
- Multi factor authentication
- ….

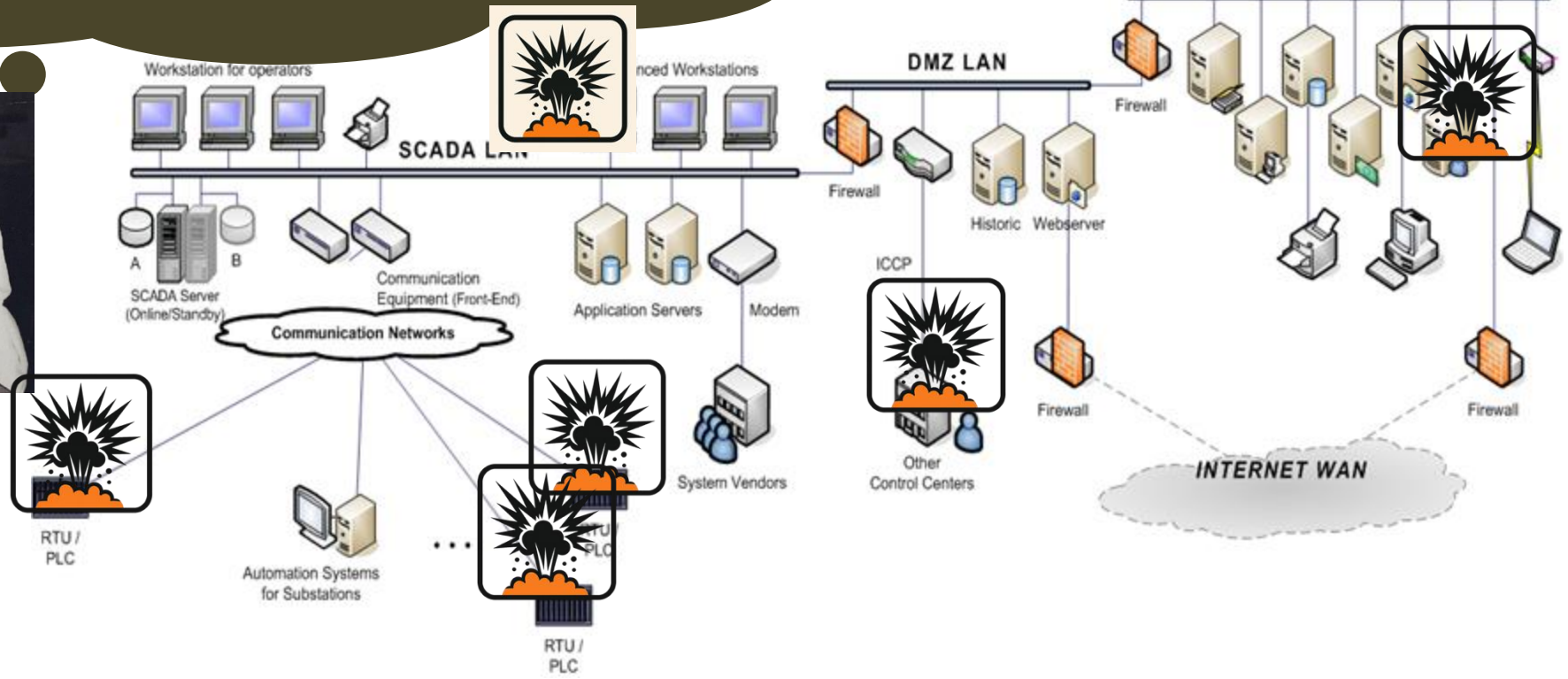
But we have limited resources…

→ ***What is the most resource effective combination of security mechanisms?***

Cybersecurity management is difficult!

- What happens if I am breached!?

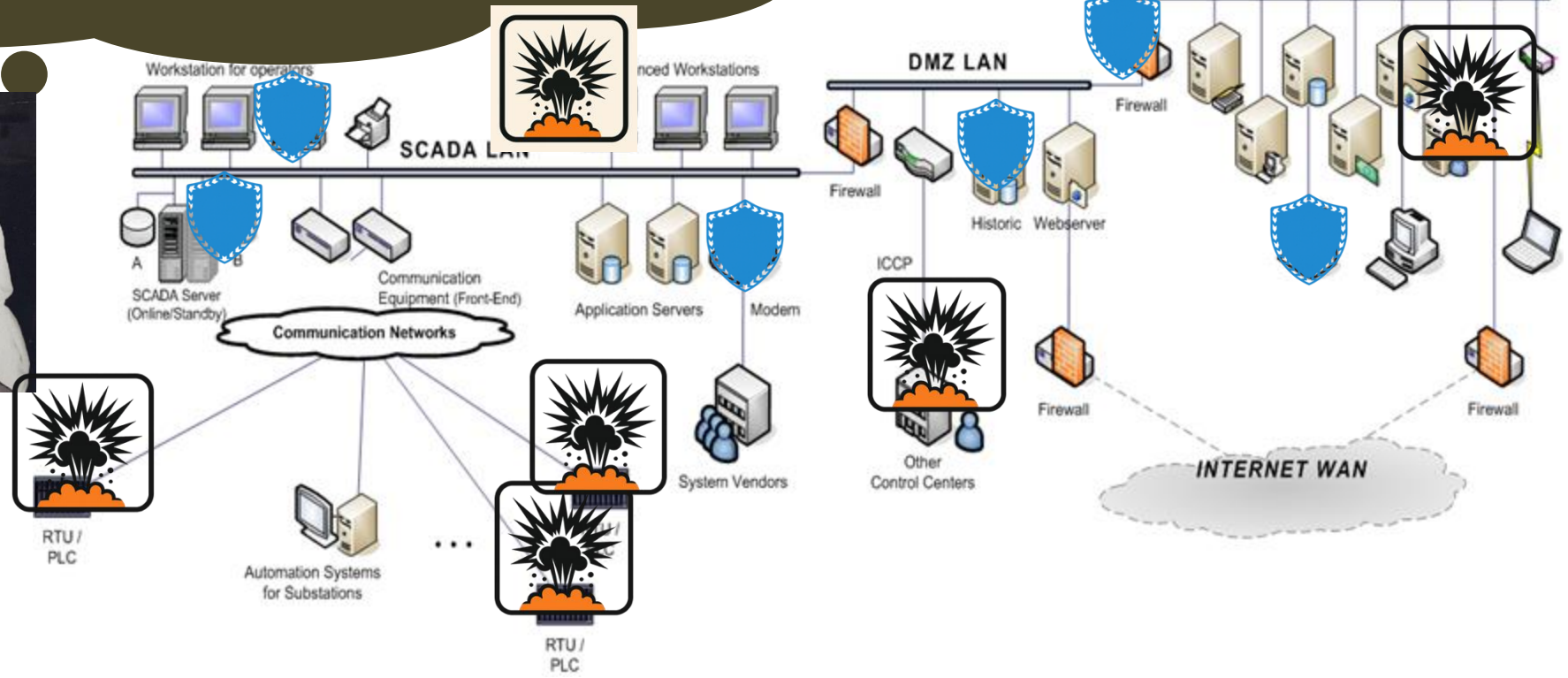
security architect (etc.)



Cybersecurity management is difficult!

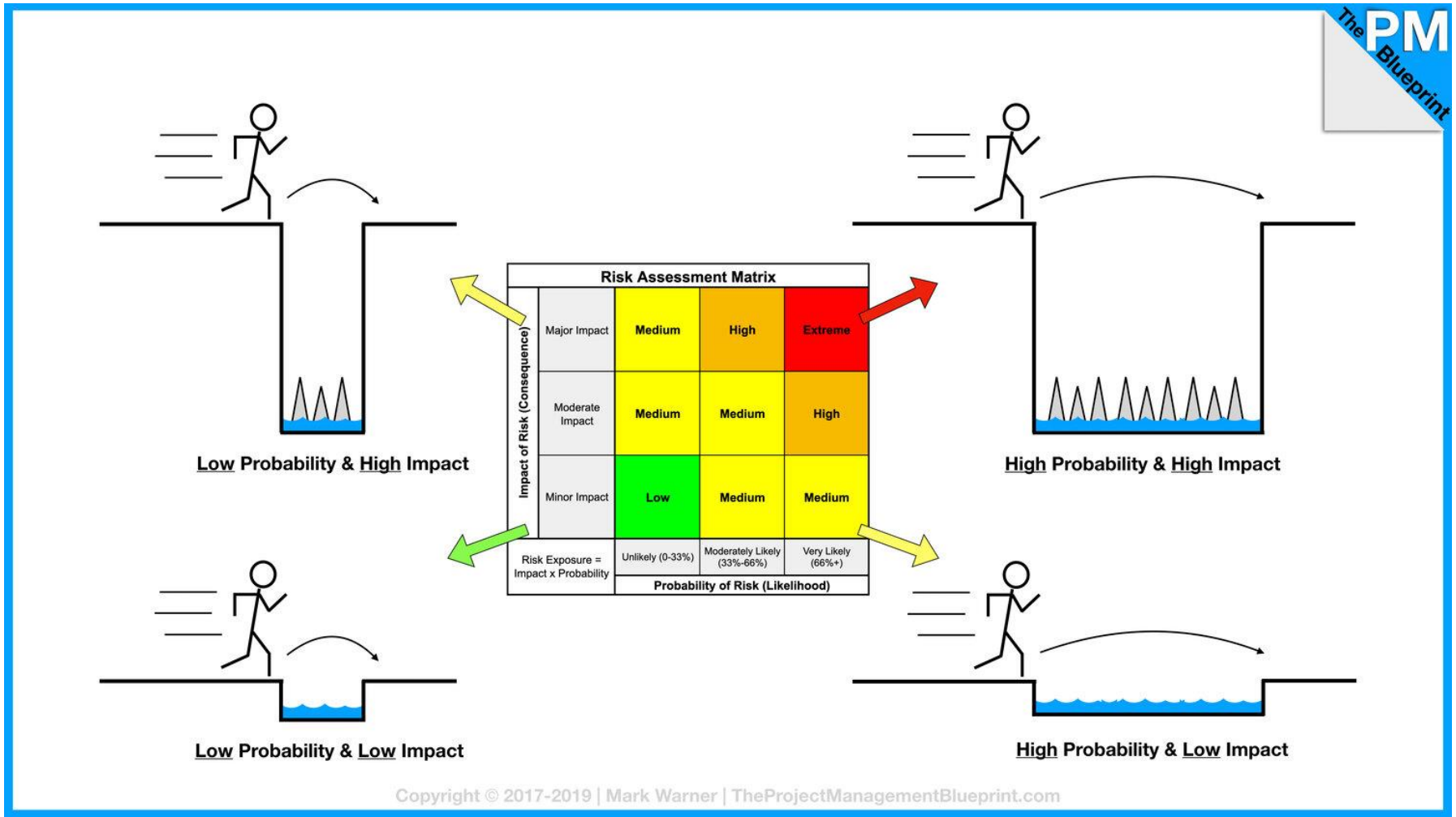
- How can I make *my* IT system securer/secure enough?

security architect (etc.)



Risk assessment (Life on the gray scale..)

The standard approach to risk: Risk = Probability * Impact



Risk in cybersecurity

Risk = f(Threat, System Resilience) * Impact

Threat encompass attacker properties, e.g.:

- Attacker skills
- Attacker resources
- Compromise reward for attacker
- Ability to repudiate

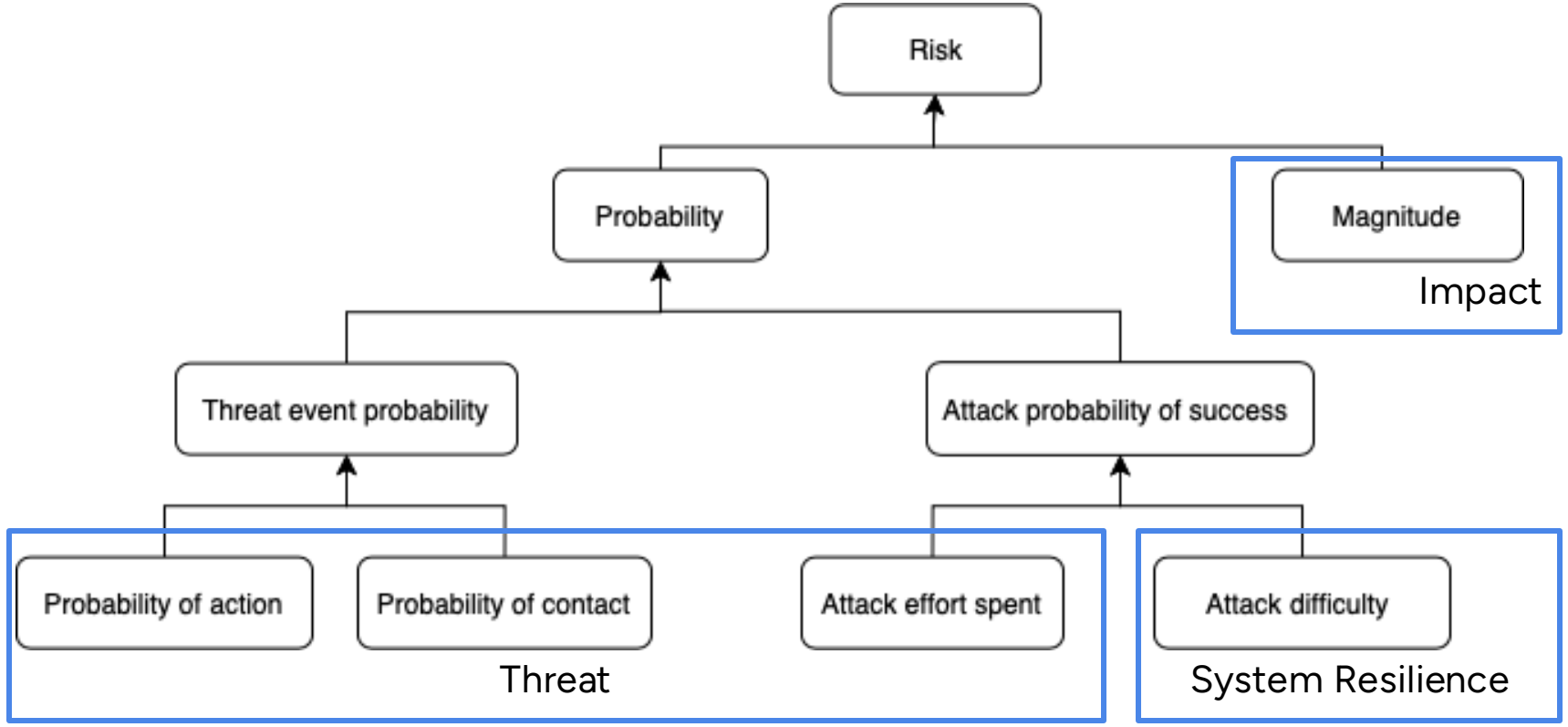
System Resilience relates to the attacked system properties, e.g.:

- software vulnerabilities
- employed protection mechanisms
- system structure (defense-in-depth, moving-target-defense, …)

Impact relates to any consequences of a successful attack(s) for the targeted organization (but also possibly to society at large), e.g.:

- Production, income, lost customers, ..
- Reputation, safety, privacy suffering, lawsuits, ..
- Restoration costs

Risk assessment framework in this course



Inspired from FAIR (Factor Analysis of Information Risk) approach.

(Freund & Jones + Open Group material)

A note on security risk vs. general risk

Our scope, cybersecurity risk:

- Malicious intent
- Where computer systems are a key element

Outside course scope, general risk:

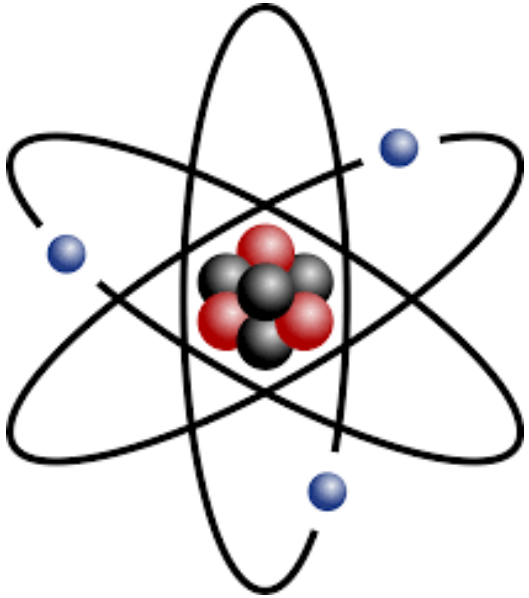
- Targeted and non-targeted
- Malicious and non-malicious
- People, nature, society. Whomever!
- Inherent uncertain processes

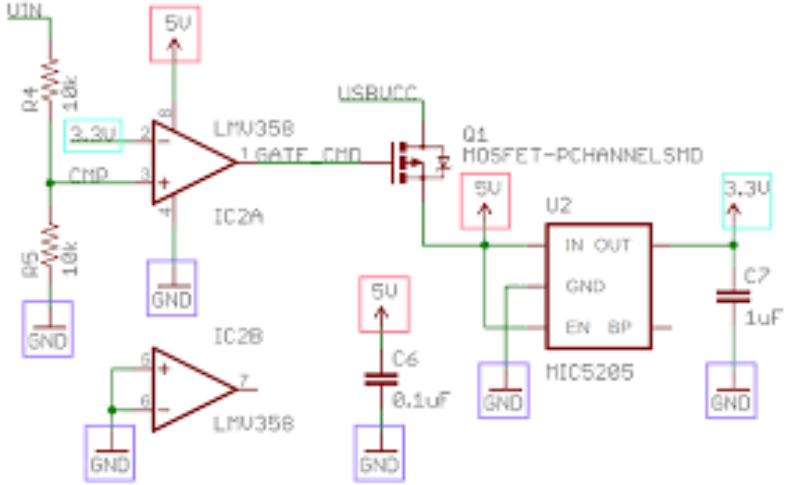
Potential confusion when “risk” meaning either threat, resilience, or impact.

- E.g., the impact of a general risk can be weakened cybersecurity resilience..

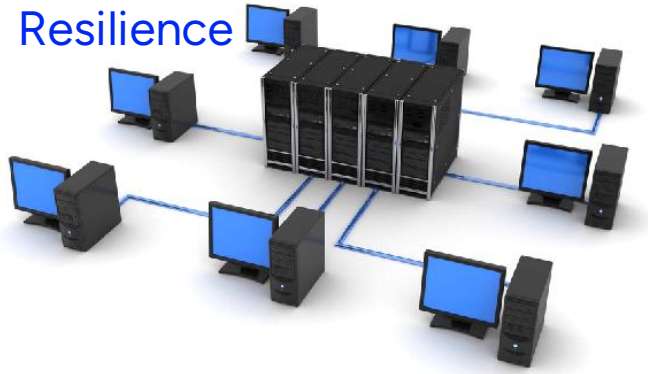
All types of engineers uses models for their analysis and design

Models are tailored for a purpose!



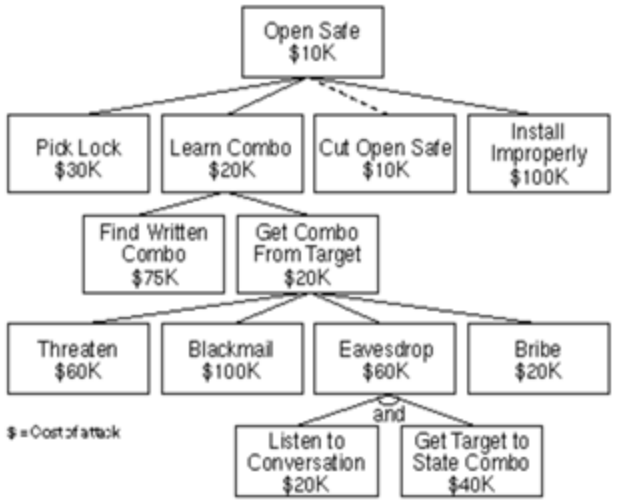


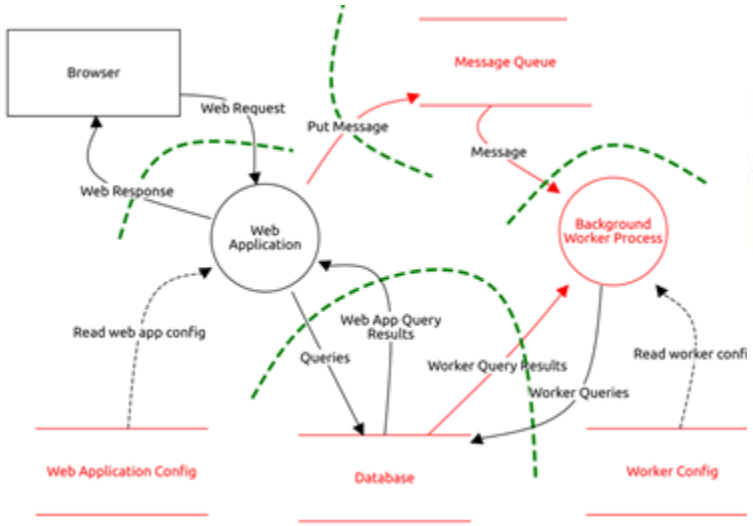
Enter: threat model(s)!

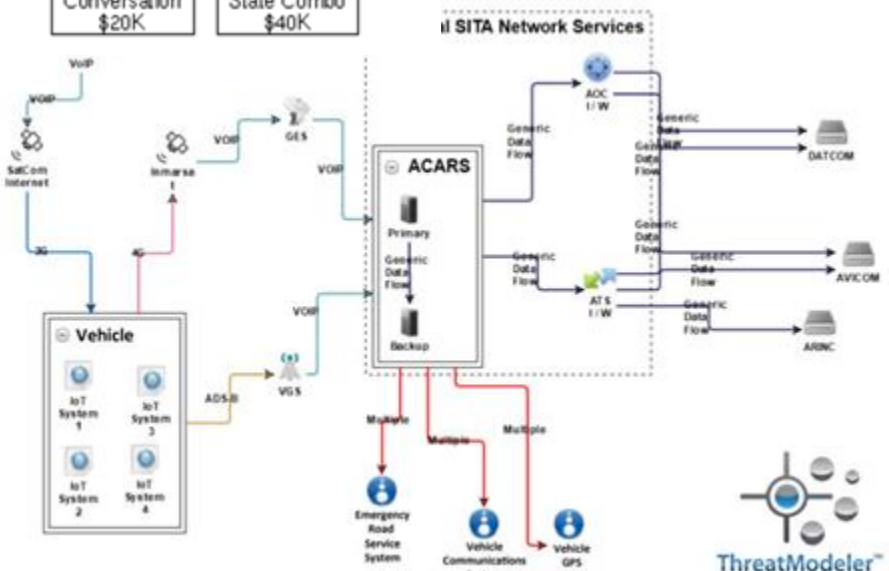


Reality

Models







Threat modeling as an approach to cybersecurity assessment

- There are many ways of assessing cyber security, e.g.: pentesting, code analysis, and law/standard compliance audits.
- Threat modeling aims to make security assessment of (large and complex) systems in a **holistic** and context-aware manner.
- Threat modeling is the best approach for this available today. However, unfortunately, it is a fairly immature field…

Threat Modeling Method in this course

Yet Another Cybersecurity Risk Assessment Framework (Yacraf)

Combining risk calculation framework with threat models

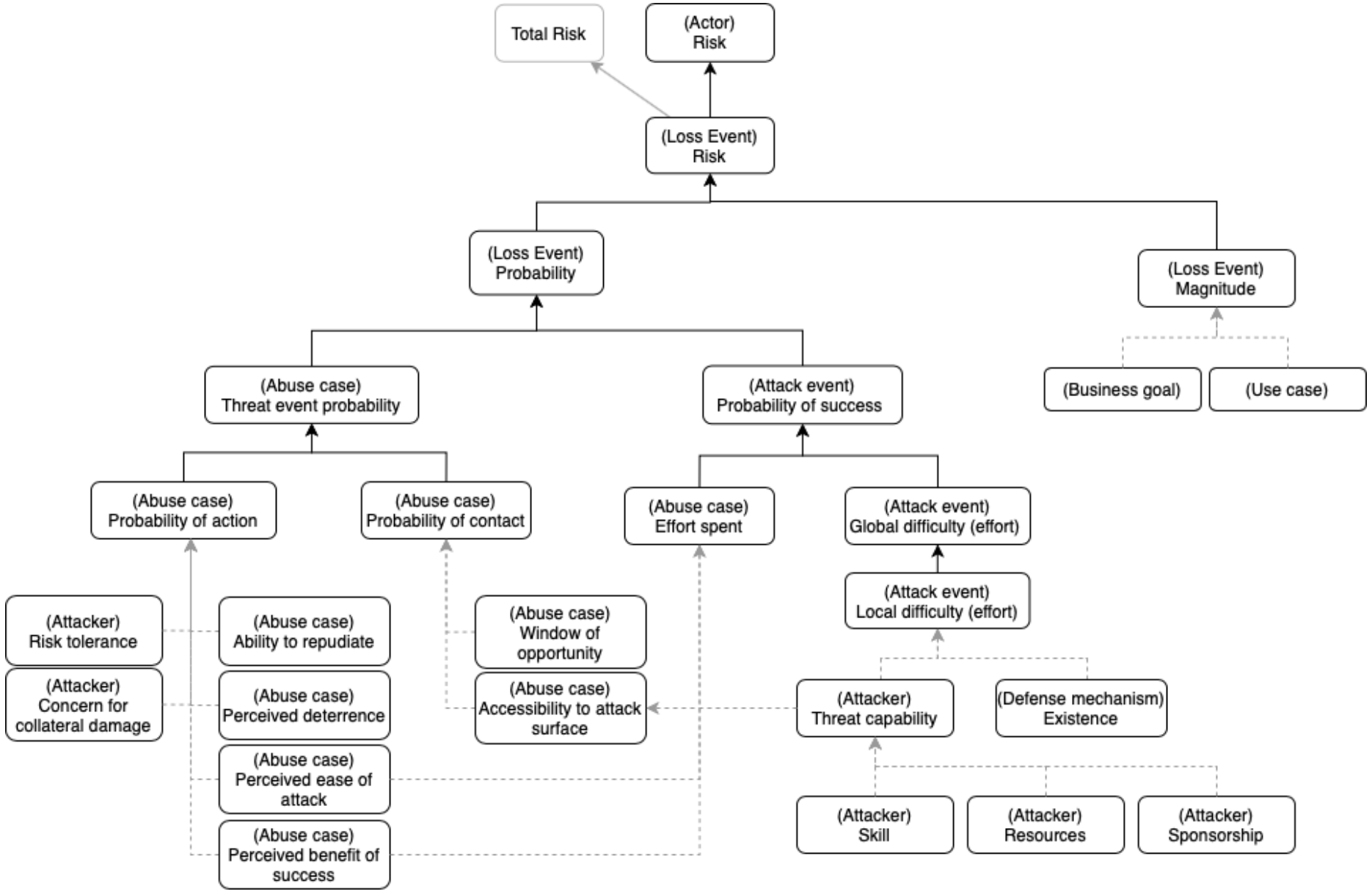
Contains 5 (+1) phases/parts:

0. Scope and Delimitations
1. Business Analysis
2. System Definition and Decomposition
3. Threat Analysis
4. Attack and Resilience Analysis
5. Risk Assessment and Recommendations

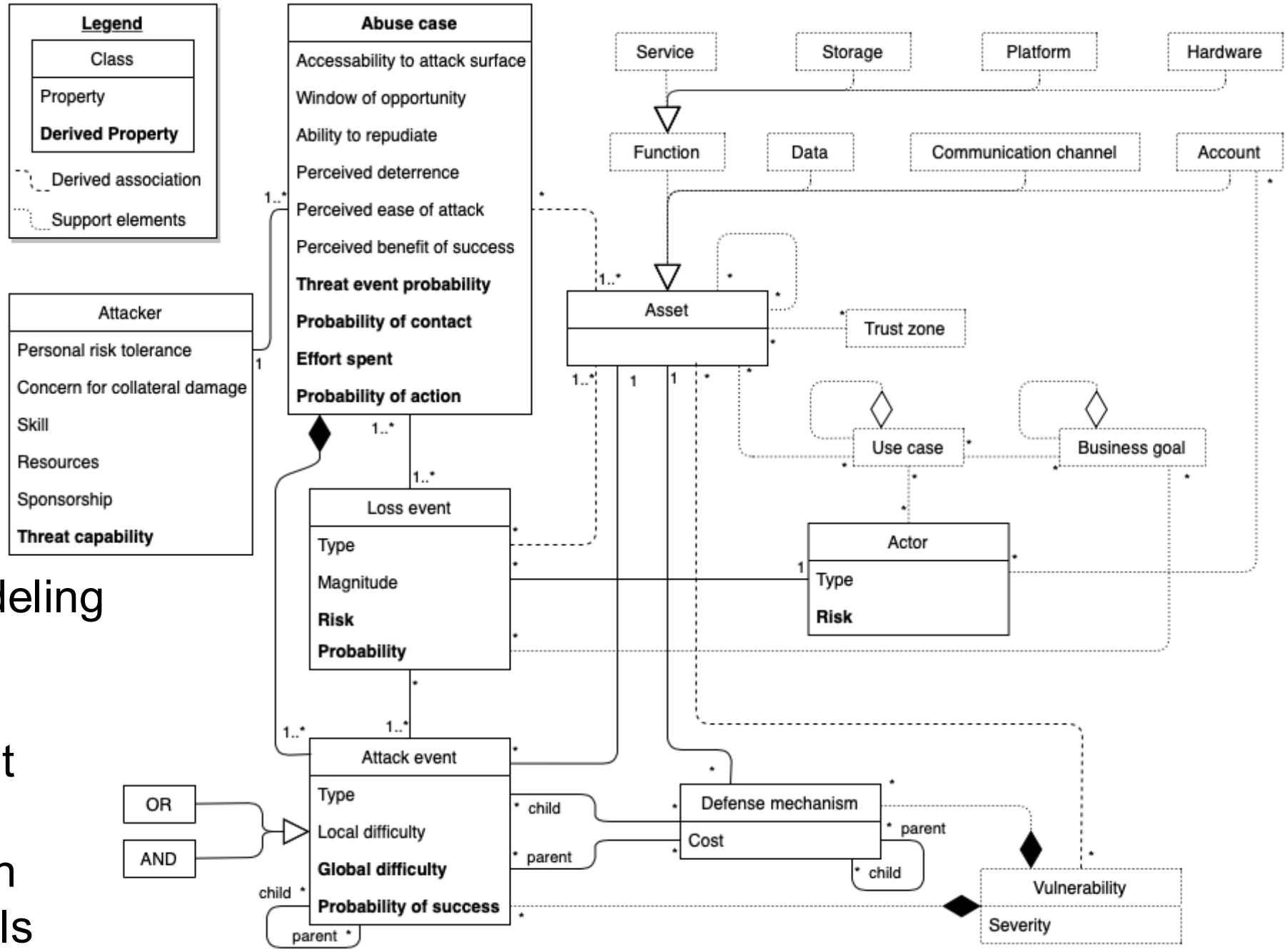
Developed at KTH

- No-one uses this methodology as-is the way we do it in the course
  - **WAIT, WHAT!? -** Why teach something that is not used..?!?
  - All of the different parts of Yacraf are needed to do a full cyber risk analysis – but other commonly used methods all lack (or are weak in) some of these parts
- Yacraf is a combination of the best parts of a number of other methods
  - It is more a reference framework that can be tailored in the end
  - You need to know what parts you are simplifying when doing this in real projects
- …With good motivation you can deviate from it also in the course

Yacraf risk calculation framework



Yacraf threat modeling language

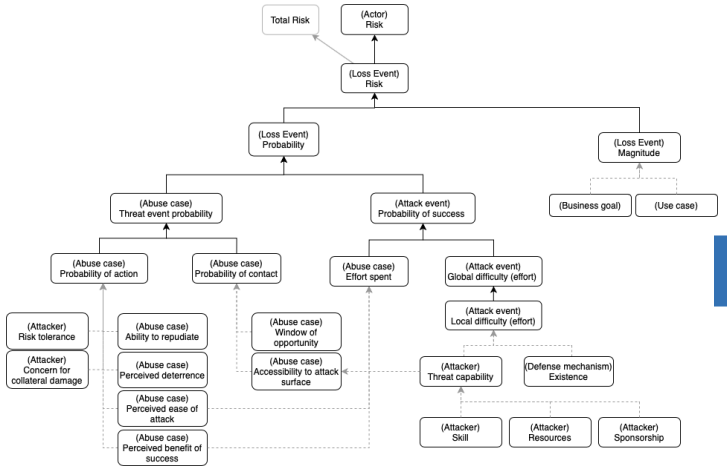


Why do we have a modeling language?

– To ensure we have a coherent and consistent approach

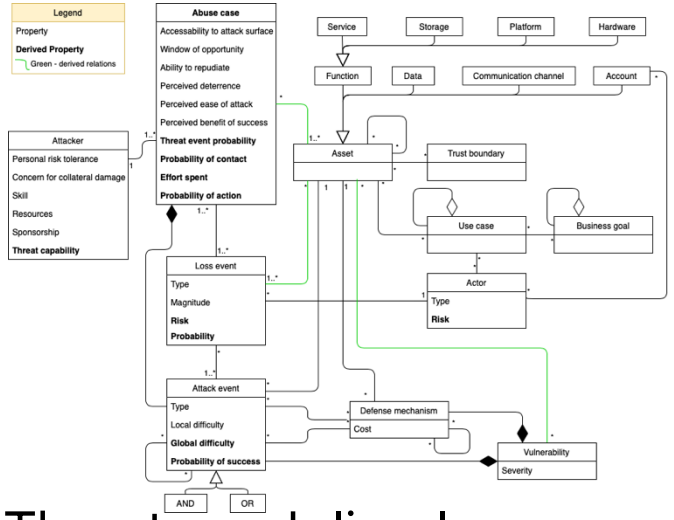
– In a language you can express instance models

Threat model-based risk analysis



Risk calculation framework

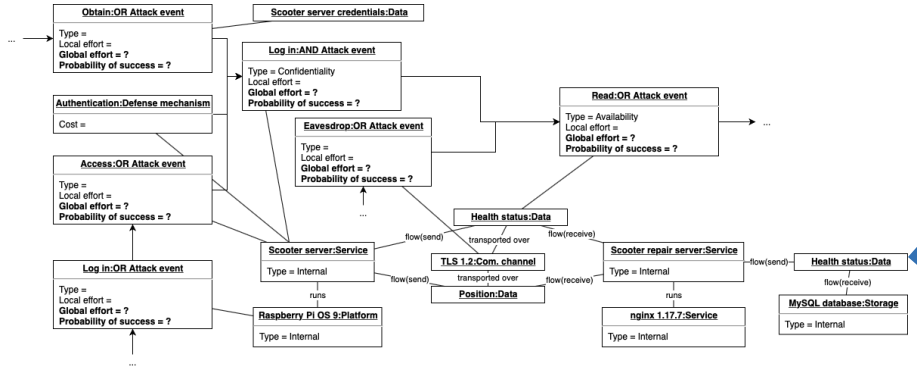
Encoded in



Threat modeling language

governs

Provides



Threat model instance

Risk assessment

Framework vs method vs modeling rules vs …

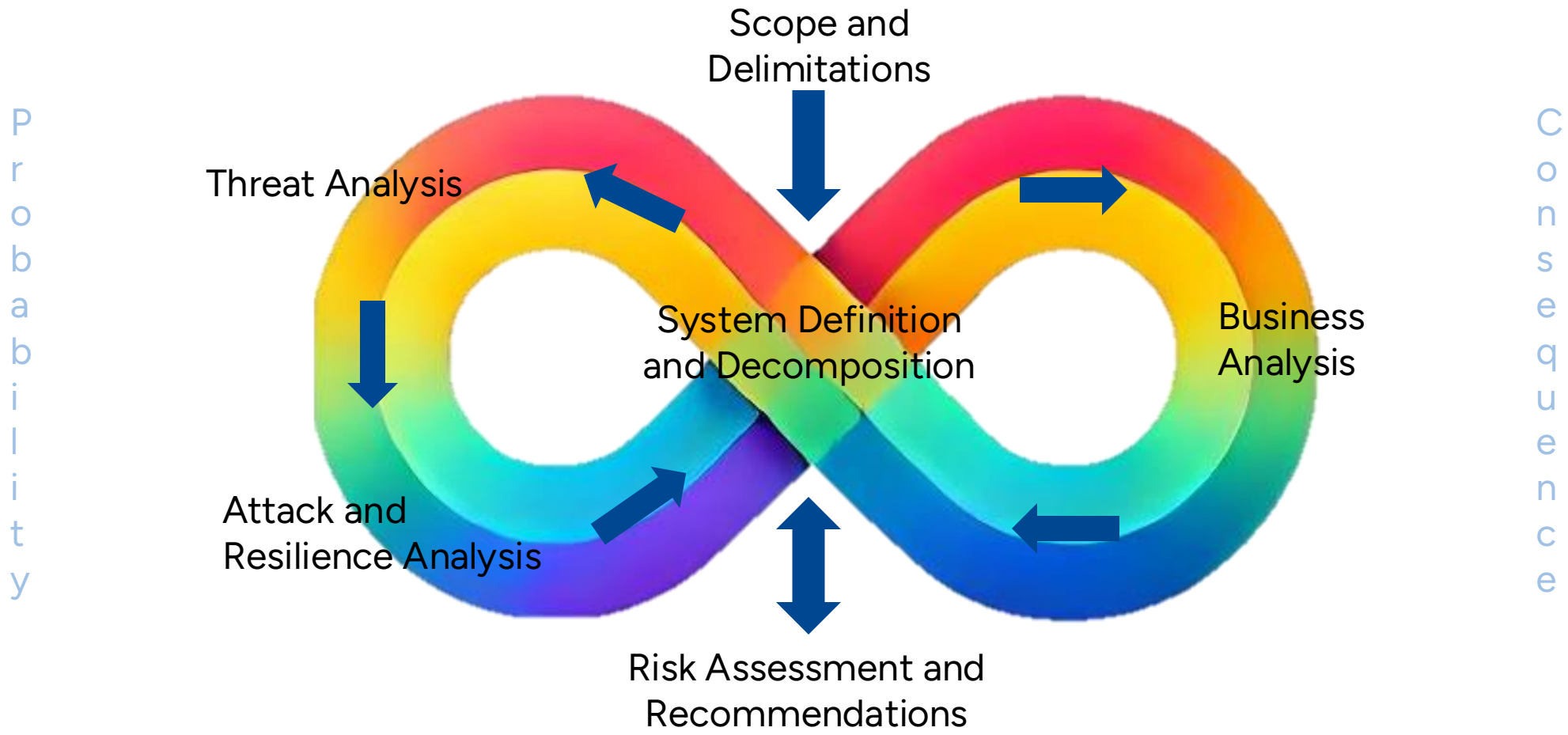
Yacraf contains 5(+1) conceptual parts/phases:

0. Scope and Delimitations
1. Business Analysis
2. System Definition and Decomposition
3. Threat Analysis
4. Attack and Resilience Analysis
5. Risk Assessment and Recommendations

- Looks linear …but:
  - Has a conceptual start and end point (scope and recommendations, respectively)
  - As a method - preferably iterated
  - As a report – preferably linearly presented

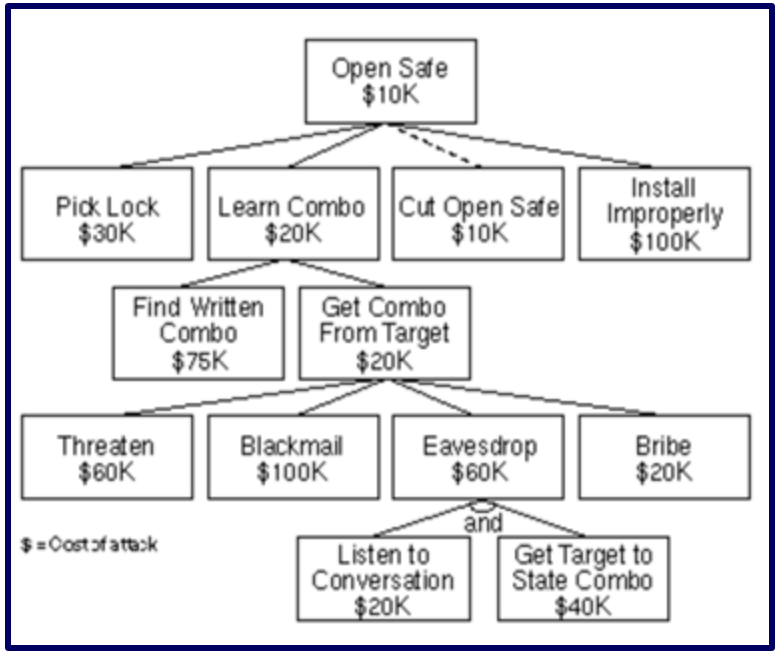
(Suggested) Way-of-working

IT-system is our centerpoint (the cyber threat modelling best practice)

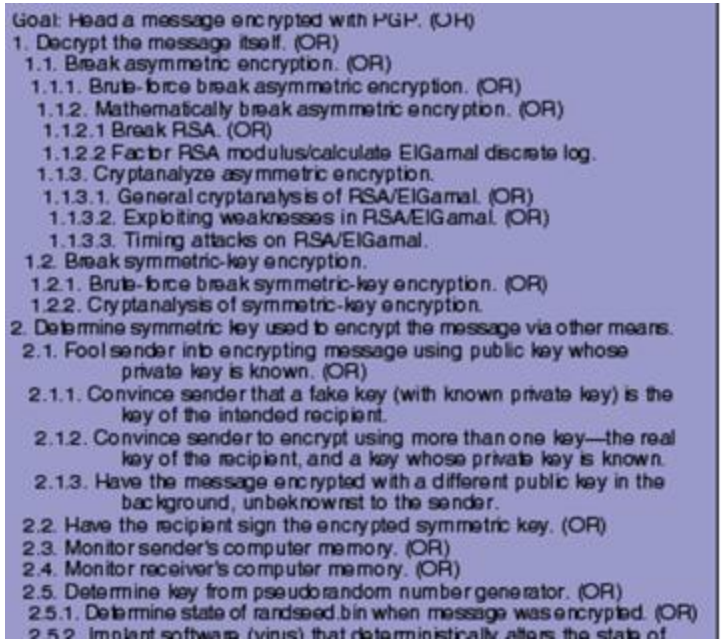


Graphical models

Threat modeling can be more or less focused on graphical models. In general, we prefer graphical models in this course. They are good for conceptually organizing things, get an overview, and to assist consistency. However, graphical models should not be used when they become a burden…



vs.



A note on terminology

- Risk terminology is not consistent and also sometimes unambiguous
  - E.g. again, the impact of a general risk can be weakened cybersecurity resilience..
- Talking to others may lead to misunderstandings
- References used as supplementary material in this course, including the books, is not always self-consistent, and certainly not consistent between each other
- Guest lecturers might use terms differently

Notes on course theory and literature

- This course is an amalgamation of a number of references and previous work, refined to make a consistent whole for our intended scope.
- The single reference that is covering the course scope (threat modeling-based risk analysis) is the PASTA method by Uceda-Velez & Morana
- For the risk calculations the YACRAF is heavily inspired by the FAIR method by Freund & Jones. Also the threat analysis draws much from FAIR, but also PASTA and the Software engineering concept of Abuse cases.
- For the business modelling we follow a significantly reduced mix of software engineering and enterprise architecture approaches.
- For systems modeling we mainly follow data-flow diagrams and their later evolution into “threat models”
- For attack modeling we are based on the attack tree method as introduced by Schneier with a few add-ons simplified from previous research at KTH (with the so called Meta Attack Language) and attack-defense trees by Kordy et al.
- The most generally renowned threat modeling method is STRIDE and the book by Shostack. This work complement our method, in particular with an approach to identify vulnerabilities.

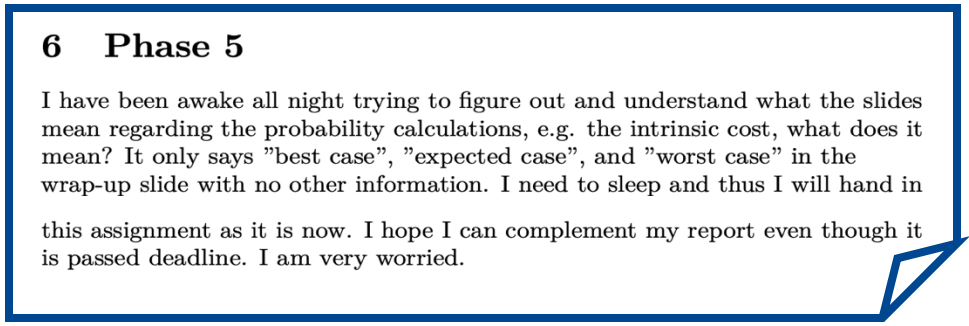
Course design

Main course building blocks

- Main course assignement
  - Conduct a cyber risk analysis of a fictitious organization and suggest effective measures to improve security.
- Course material
  - Method - main
    - Yacraf paper, recorded lectures, examples
  - Systems, security and risk
    - Some pointers, self study, reference look-up.
  - Supplementarty (for broader and deeper understanding of the topic)
- Activities
  - ***Self study!***
  - Case lectures
  - Q&A sessions
  - Draft hand-ins for feedback
  - Peer review
  - Study groups
  - Guest lectures (mandatory in EP2790)

You need to plan your course activities and learning - We are here to support with this during the course

- The course is to ***do a project*** – model, analyze, report.
  - We suggest iterations and prototyping – not waterfall execution. And make the first iteration ASAP.
  - We want to avoid →
  - The assignment is open-ended!



6 Phase 5

I have been awake all night trying to figure out and understand what the slides mean regarding the probability calculations, e.g. the intrinsic cost, what does it mean? It only says ”best case”, ”expected case”, and ”worst case” in the wrap-up slide with no other information. I need to sleep and thus I will hand in this assignment as it is now. I hope I can complement my report even though it is passed deadline. I am very worried.

- Most sessions are Q&A – you need to have the Qs!
  - Reflect and ask. E.g. avoid the feeling of “The example diagrams and tables shown in the examples are much more complex compared to what is described in the lectures. It is difficult to know what is required and makes me feel that I am always lacking some knowledge to proceed.”
  - 1:1 discussions, but public. You can come with your specific questions relating to your assignment, but everyone is allowed to listen.
  - Designed for physical. Zoom at best effort.
- “Not having physical lectures makes it so that the course becomes less appealing since there is not a lot of communication with the teachers. This demotivates a lot when it comes to working on the report.” → **Come talk to us!**

Use of large language models

- Encouraged tool. Use them as much as you find it useful. No restrictions. – Your personal teacher.
- BUT, transparent reporting of usage.
  - Ethical approach, EECS code of conduct
    - In particular: “Rule 2: In any assessment, every student shall honestly disclose any help received and sources used”
  - We (students and teachers) want and need to learn how LLM should and should not be used.

- You are still responsible for what you hand in
  - You need to quality check
    - LLMs are not specifically trained in the course…
  - “ChatGPT says …” is not a good motivation per se.

- Hand in reflection of tool usage

The (quite new) Yacraf tool

- In order to avoid repetitious calcualtion work for students we have developed a tool to support modeling, analysis, and calculation according to Yacraf.
- Developed by a previous student of the course. It is an by-engineers-for-engineers type of tool, not a polished enduser product.
- No warranty, no support.
  - Well, perhaps just a little. We want the tool to become useful so we will be spending some effort on maintenece and development. Please let us know if you find bugs or if you have ideas for improvement. We do not guarantee we address issues though.
  - Known bugs collected in GitHub Issues
- Free and open source
  - Let us know if you want to contribute (perhaps as a TA later on)

”What advice would you like to give to future participants?”

- Results from previous course surveys

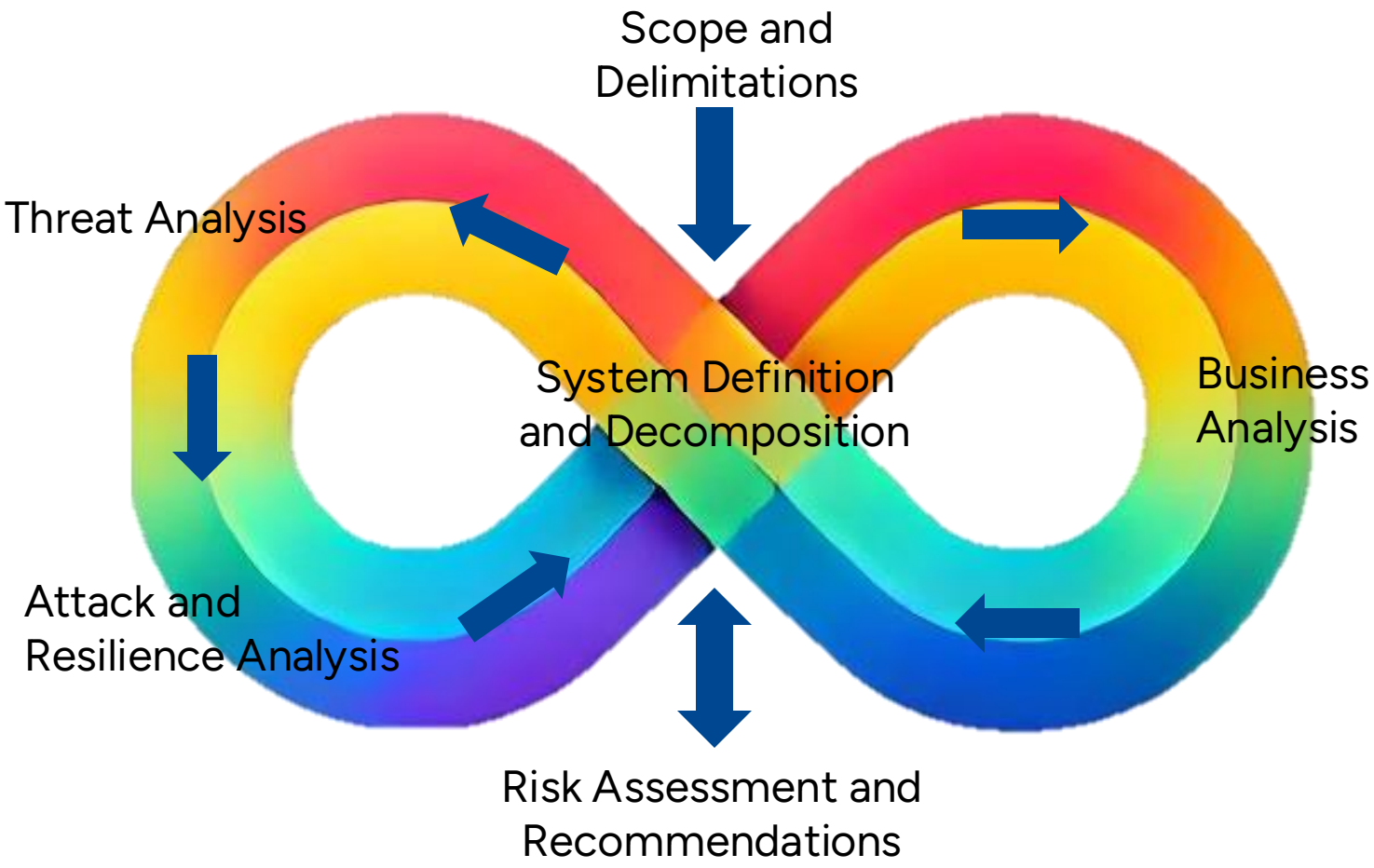
- Start and finish the first iteration fast, ask if have questions, have an understanding of the calculation math.
- Choose a system that is interesting to you and the course will be more fun and you will learn more!
- Start early on, it takes a bit of time in the beginning before you fully understand what you are doing. Additionally, focus on an iterative approach instead of spending too much time on each phase in the beginning. There is a risk that you will have to change a lot as you get a better understanding of how things should be done later on.
- At least for me I feel that it would have been easier to get started in full speed earlier if some of the concepts would have been done earlier in the course.
- To really work with iterations from the beginning and not take any shortcuts. The first iteration will feel overwhelming, but once you understand how the parts of this threat-model actually relate to each other, things become much more clear.
- Even though the freedom given was nice, I feel like it was too much in the beginning, and it was a bit overwhelming to get started. I know this is intended but I feel like if there was more guidance in the beginning it would have been easier
- I felt it was a fair amount of work to do each week, especially if you started early on.

(Suggested) Way-of-learning

Go through all material on a high level – then iterate deeper into the material

E.g.

- Start with understanding and modelling structure
- Add simplistic data/figures
- Add more details in the models
- Add more realism/motivation/references to the models and data
- Add uncertainty



Grading

- The assignment will be evaluated according to a number of grading criteria listed in on the grading page.
  - Understand these criteria ASAP, cross-examine your work regularly.

- …But how many pages should my report be..? – No limits, ***number of pages is not a grading criteria***!
  - However, some stats to provide some general feeling (pre Yacraf tool):
    - Average pages (total): A-reports: 38, E-reports: 23
    - Average pages (business analysis): A-reports: 5, E-reports: 6
    - Average pages (system definition and decomposition):A-reports: 4, E-reports: 3
    - Average pages (threat analysis): A-reports: 6, E-reports: 3
    - Average pages (attack analysis): A-reports: 10, E-reports: 5
    - Average pages (risk assessment):A-reports: 9, E-reports: 3
    - Average abuse cases: A-reports: 7, E-reports: 4
    - Average threat actors: A-reports: 3 E-reports: 2
    - Average tables: A-reports: 15, E-reports: 5
    - Average figures: A-reports: 17, E-reports: 12
    - Average references: A-reports: 25, E-reports: 9

Some new and changed slides

- All slides are gathered in this document.
- Some new complementing slides have been developed along the way. Recorded main lectures are older.
  - Some slides have been slightly updated as compared to recorded material.
  - Some slides are new
    - Some new slides are also presented in short videos.
  - (New slide template)

- All updates are marked
  - (Except for some language corrections etc.)

Material walk-through

- Canvas and GitHub

Method Overview

Yet Another Cybersecurity Risk Assessment Framework (Yacraf)

Yacraf

Contains 5(+1) conceptual parts/phases:

0. Scope and Delimitations
1. Business Analysis
2. System Definition and Decomposition
3. Threat Analysis
4. Attack and Resilience Analysis
5. Risk Assessment and Recommendations

- Looks linear …but:
  - Has a conceptual start and end point (scope and recommendations, respectively)
  - ***As a method - preferably iterated***
  - As a report – preferably linearly presented

Phase 0
Scope and definition

Phase 0 - Scope and Delimitations

Describe the system under evaluation on a high level; what is it and what does it do?

- What are the main technical components?
- What is the system trying to achieve, what is its purpose?
- What components are considered outside the scope of the analysis?

In principle only done once. But maybe the scope change during the analysis.

Describe in text, perhaps with some simple figure(s) as illustration. Does not follow the modeling language strictly.

- In this course: 1-3 pages.

Phase 1
Business value and attack consequences

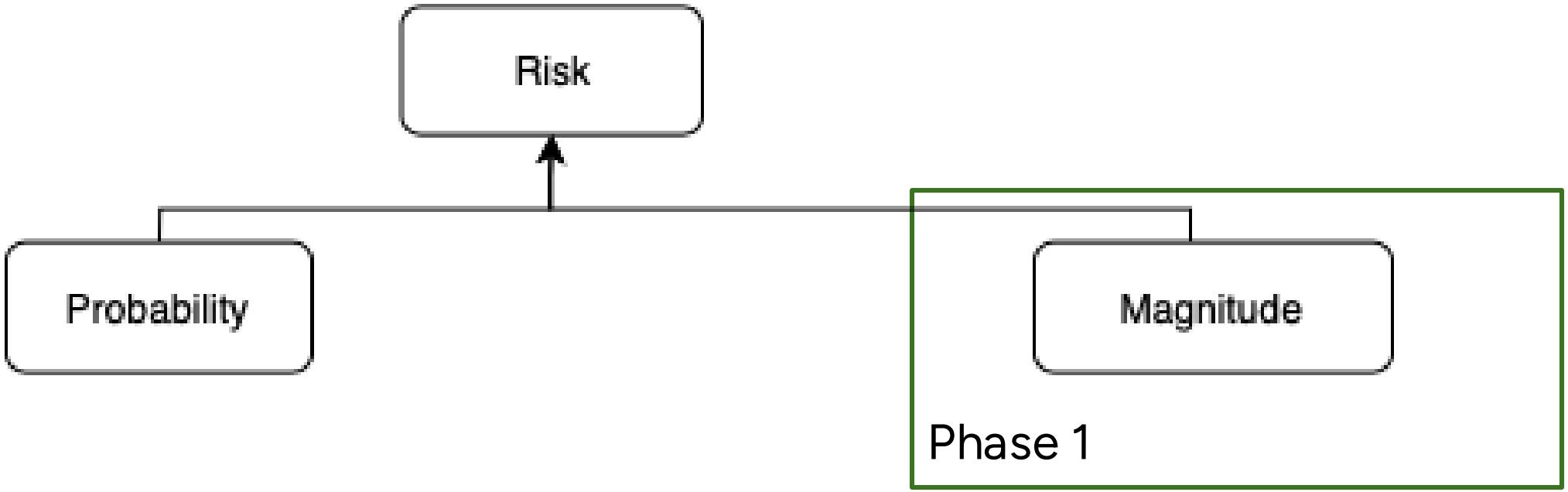
Phase 1 - Business Value of System

We start with describing what is the system used and good for, and the critical question of what can go wrong and how bad that would be?

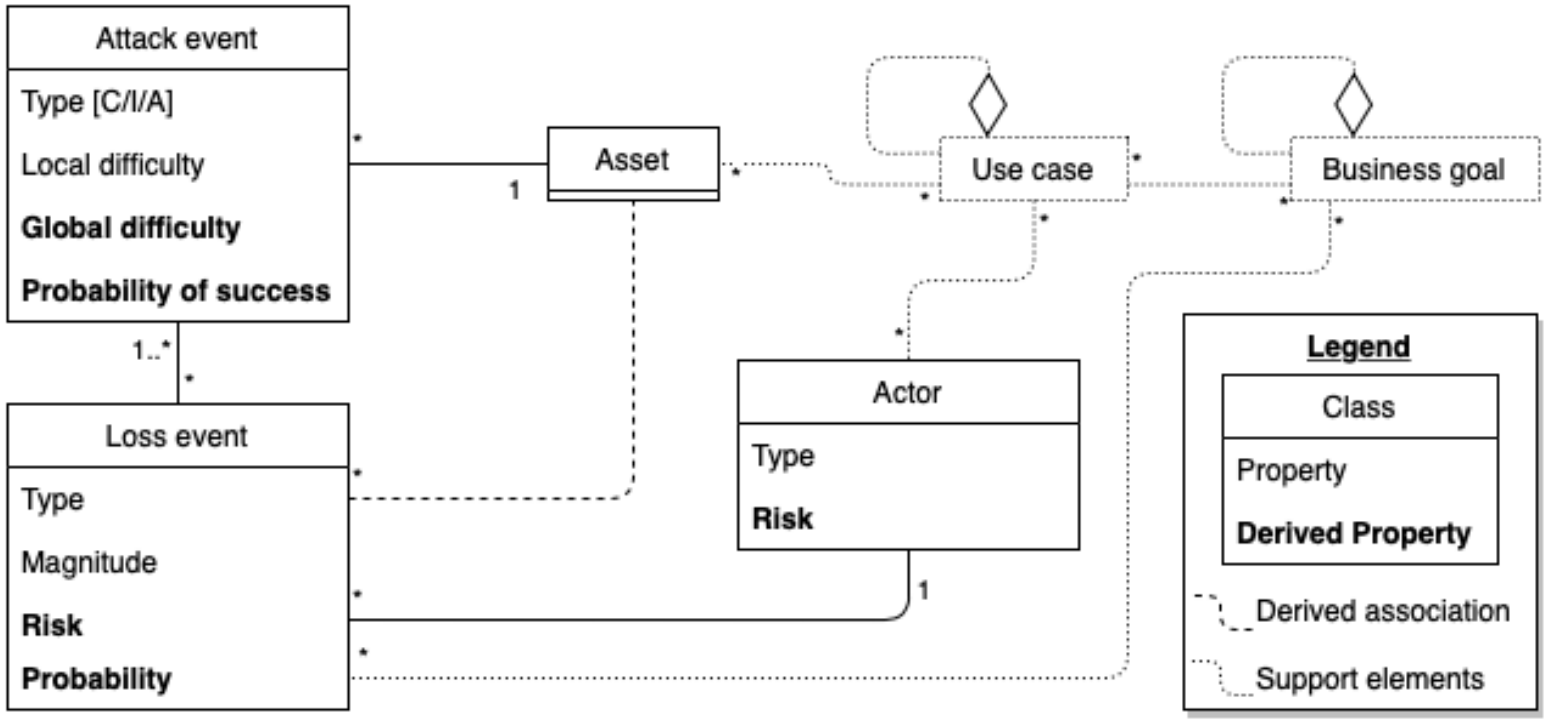
Phase steps

1. Describe business goals
2. Describe business architecture
3. Describe negative business impact of breach

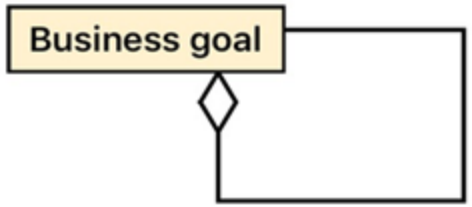
Phase 1 purpose:
Loss magnitude assessment



Language overview

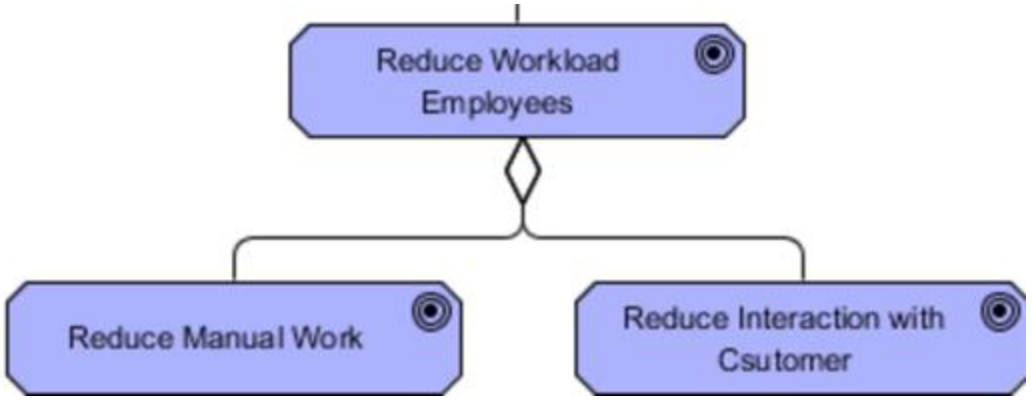


Step 1:
Describe business goals



- Describe what the goal/purpose/utility of the system is.
- This activity is simply performing goal-break down. It is easy to capture the overall high-level goals of organizations, such as
  - Maximize profit, Maximize market share, Maintain happy and healthy staff, and Minimize CO2 emissions.
- However, this does not really tell us much about what the organization do, so we want to break down these high-level goal into something more organization-specific related to the specific operations. E.g. In order to Maximize profit an online retail company might want to:
  - Minimize downtime of their web store, Minimize incorrect shipments, and Keep a slim stock.
- We might refine goals in several layers of subgoals, and subgoals might support several super goals. Hence the aggregation relation in our language.
- To keep it simple, in this course we use business goals to represent several motivation-related phenomena such as requirements, strategies, and objectives.

Generic goal model example

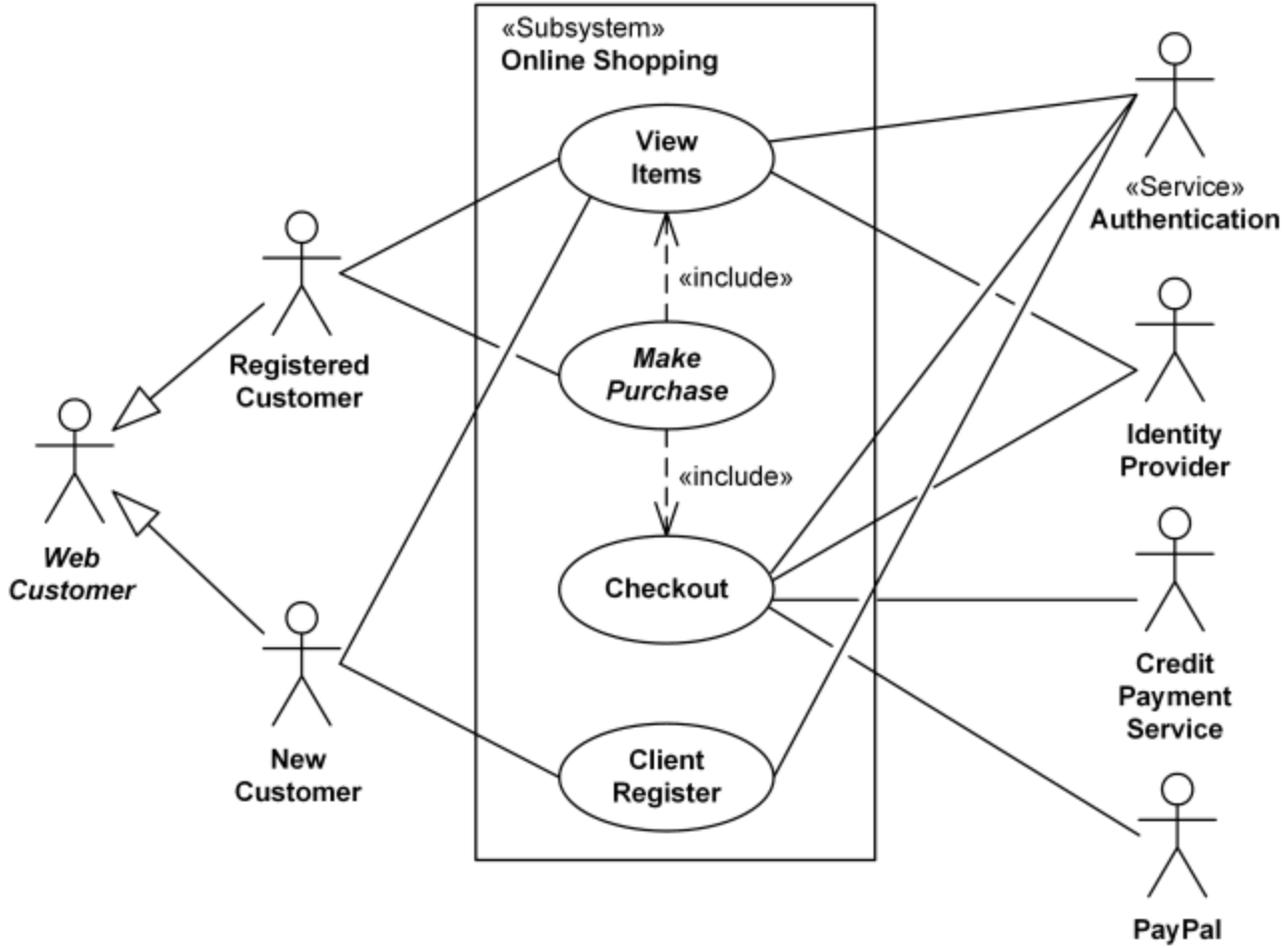


Step 2:
Describe business architecture

- How are the goals achieved? The business architecture is describing what the organization does and how it interacts with the IT-systems. There exist a number of advanced and specific approaches for business models. We will disregard all of those and simply employ **use cases** to to describe what the business does and how it works. Sometimes this means that the use cases become a little skewed, this we just have to live with.
- We can **organize use cases in hierarchies of groups/sub use cases**. The low-level use cases should (normally) describe how systems are interacted with (i.e. traditional use cases). Normally this is focused on how humans are interacting with systems, but sometimes also system-to-system use cases are developed. “Higher up” use cases can be more similar to business processes, describing pure business activities.

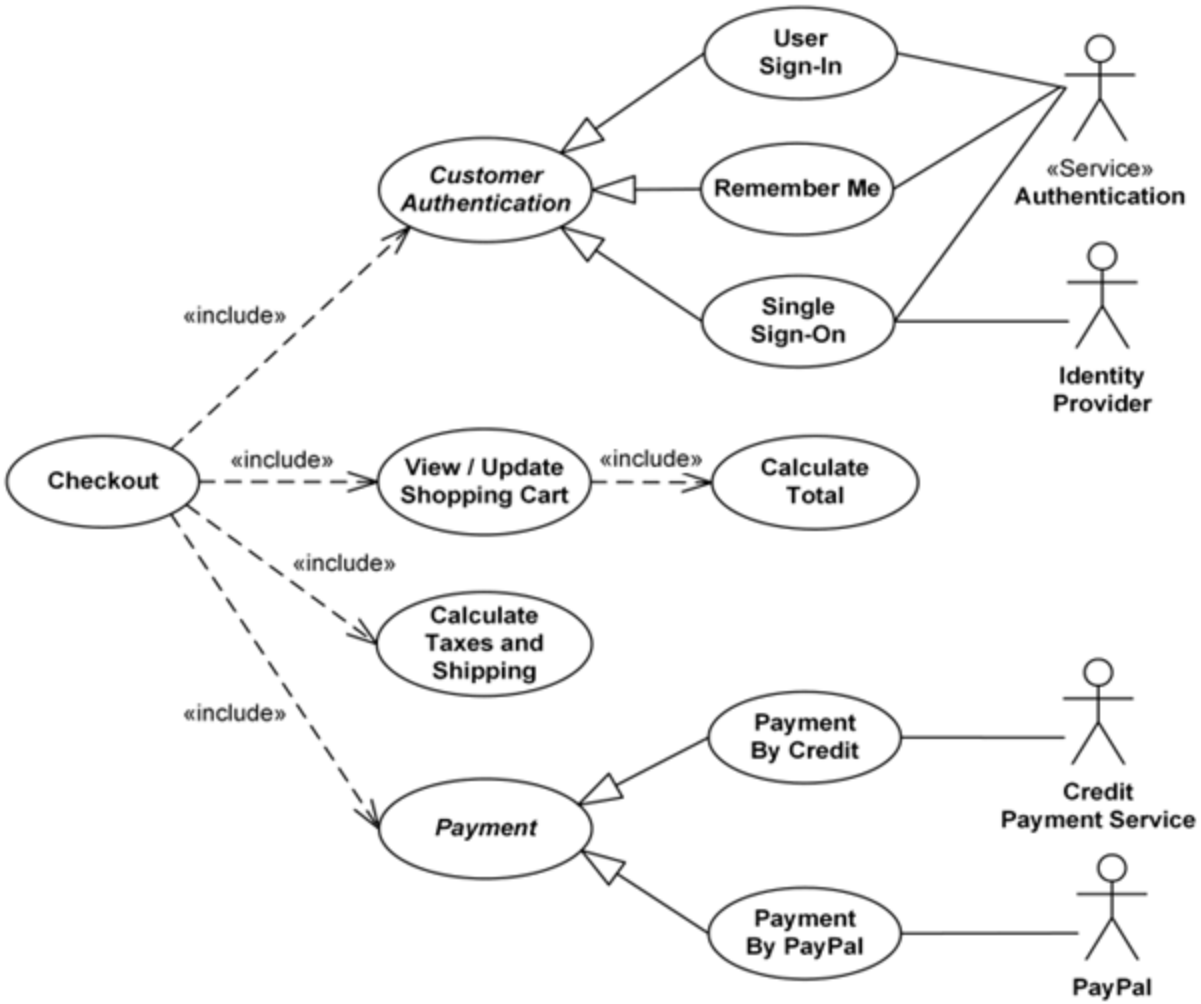
- Note! use cases and business goals are not strictly needed to make the risk assessment. Outlining them is a means to understand where and how losses may occur (next step).

Generic use case example (UML*)



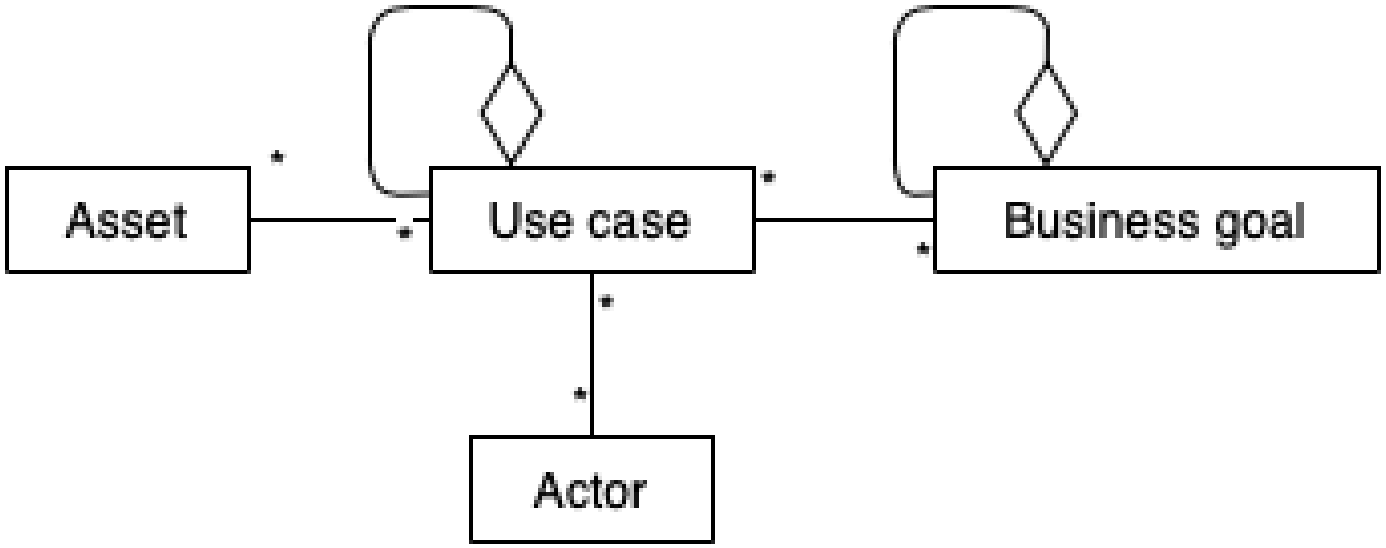
* The Unified Modeling Language (UML) is the most well renowned flavor of use cases.

Generic use case example (UML)



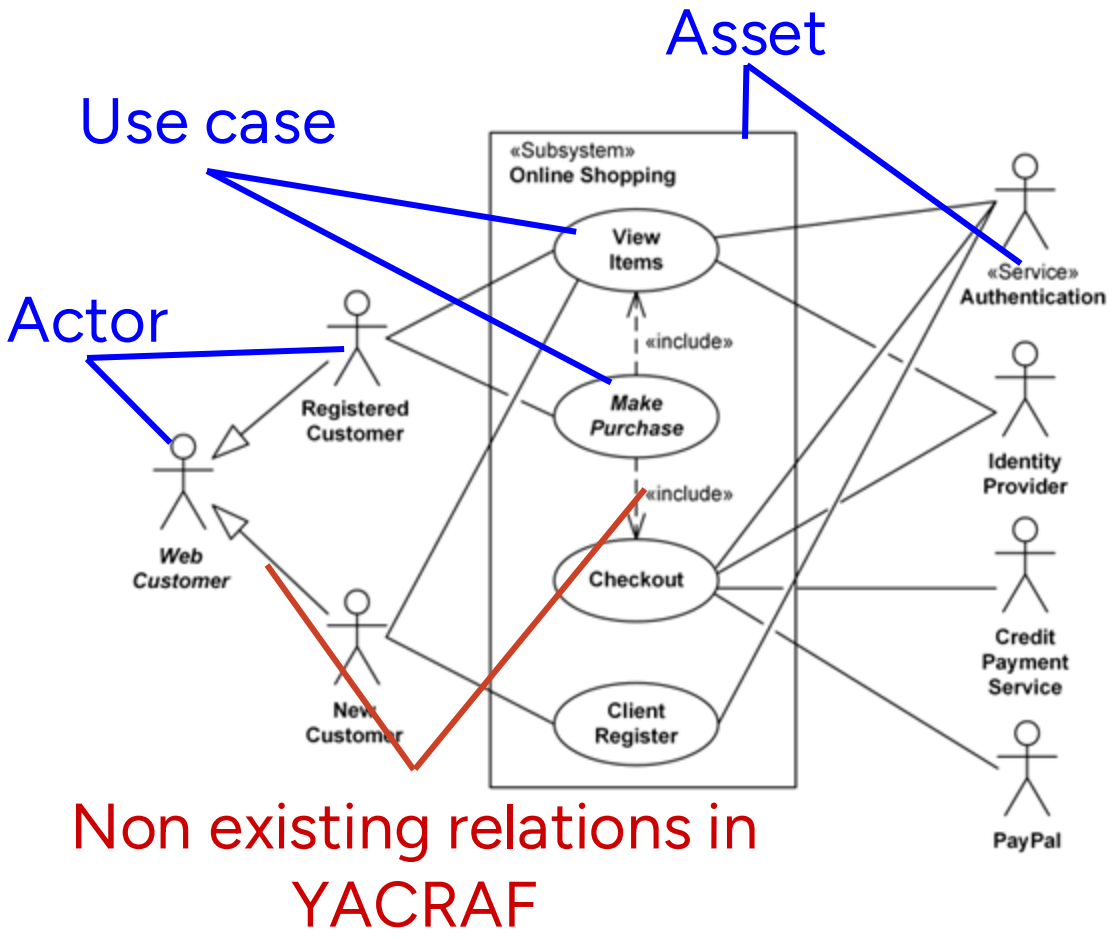
Our use case language

A simplified language for use cases



(You may if you want use a more advanced language both a more extensive use cases and business architecture. However, make sure it has a clear interface towards the system assets.)

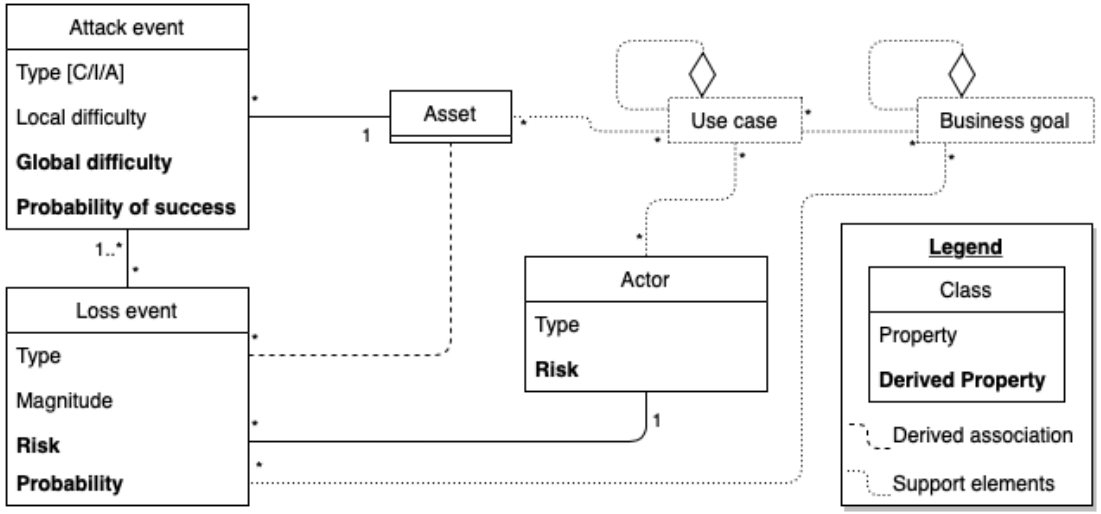
Mapping to UML



We try to keep our language simple. But if there is a need for an extension this could be added if it not messes up the rest of the language (e.g. kind-of Actors could be useful for some cases)

Step 3: Describe negative business impact of breach

- In this step we want to understand what kind of negative consequences a cyber attack would have.
- Given the identified Assets and the role they play in the business we now start examining (speculating, simulating, using historical data, etc.) what kind of events would have what type of impact for whom.

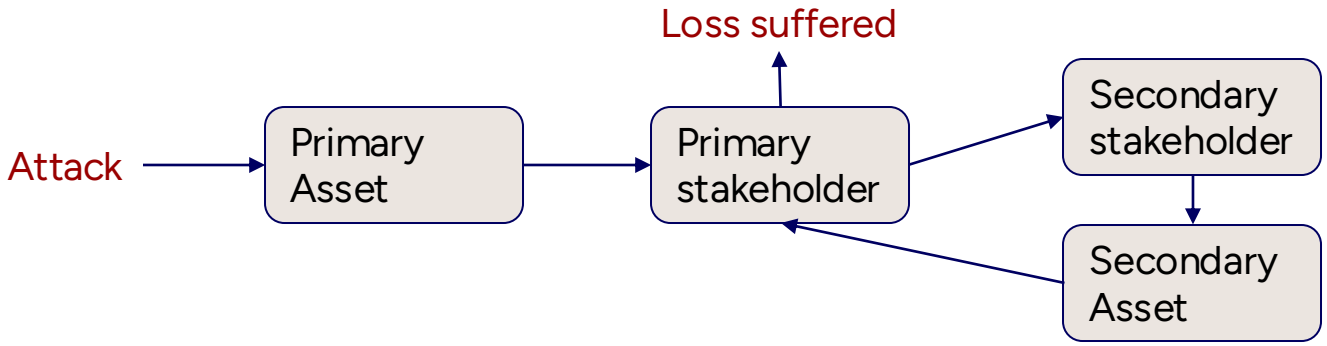


Step 3: Loss Events - Types of Loss (FAIR prelude)

- **Primary**, related directly to the observed organization, e.g. (p.37 Freund&Jones):
  - revenue loss from operational outage
  - extra wages due to operational outage
  - replacement of tangible assets
  - restoration costs
- **Secondary**, secondary stakeholder reactions, e.g. (p.40 Freund&Jones):
  - law and contractual fines, notification costs, credit monitoring, covering secondary stakeholder monetary loss, PR cost, legal defense cost, sanctions, lost market share, stock price down, increased cost of capital
  - all of these are secondary but still related to the organization. FAIR only look at the organization. In addition, **loss for society and external stakeholders** could/should also be included.
  - Secondary stakeholders are most accurately seen as secondary threats.

Loss flow (FAIR prelude)

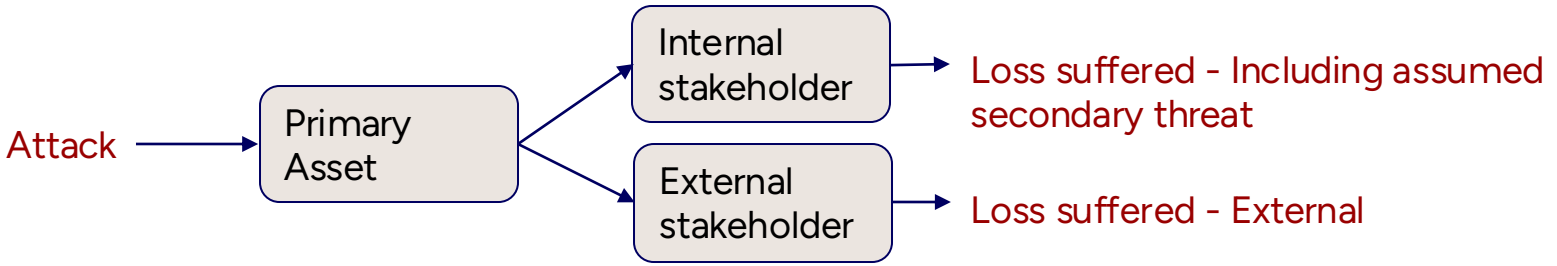
- Impacts are quite complicated… (and it is domain specific.)
- The FAIR model:

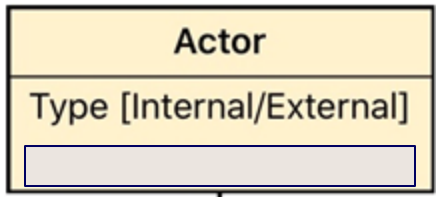


- E.g. a bank suffers from an attack and many customers loose money → reputation goes down → market share goes down → new market campaigns
  → regulators will audit extra carefully → maybe licenses will be revoked
  → insurance cost will go up and maybe coverage will be limited

Our loss flow model

- Simplified *and* extended model used in this course

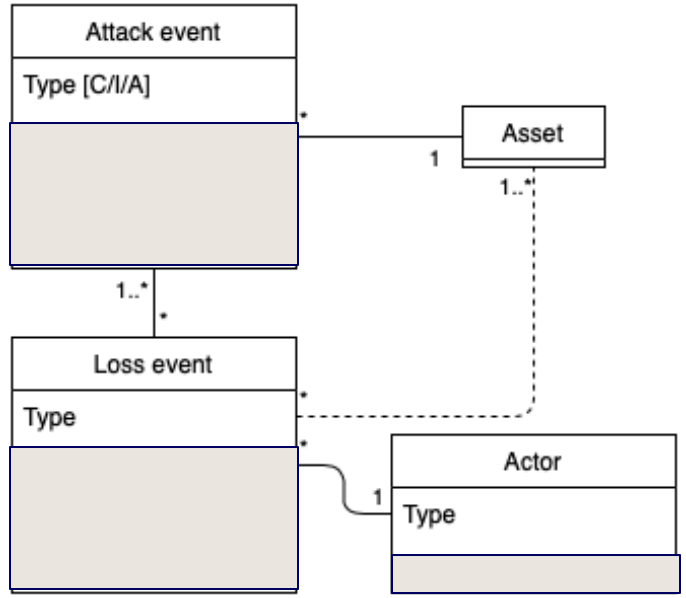




- Actors here are different groups or individuals that for one reason or another are suitable to "host" loss, given the purpose and focus of the threat analysis.
- *If no-one is suffering the loss - there is no loss!*

Loss Events

- Loss events are the completion, including aftermath, of successful attacks.
- Condensed type categories of loss according to FAIR (p.66 Freund&Jones) to be used as inspiration/checklist for identifying loss - categories are only a help to structure the model and analysis, they do not impact the analysis per se:
  - Productivity (primary)
    - Losses of lost value delivery to customers (lost or only delayed?), wasted fixed costs.
  - Response (primary & secondary)
    - Cost related to incident management.
      E.g. incident response teams and customer management.
  - Replacement (primary)
    - Intrinsic cost/value of assets. You need to buy new
  - Competitive advantage (secondary)
    - E.g. Intellectual property, business secrets, market info,..
  - Fines and judgements (secondary)
  - Reputation (secondary)
    - *Effects* of reputation. E.g. stock price, market share, recruitment
  - Other specific categories
    - E.g. injury/death, environmental impact (e.g. CO_2 emission), ...

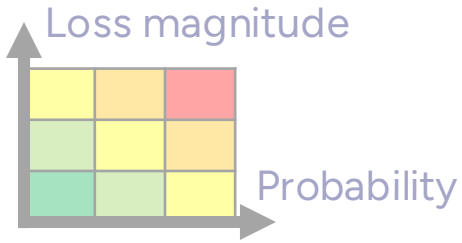


- A breach, i.e. a successful attack, can lead to multiple Loss events.
  An Asset can have many breaches.
- A Loss event must be related to exactly one Actor - the one that experience the “suffering”. (And consequently a single breach could hit multiple actors through multiple, possibly similar, losses.)
- For now we can think of attacks as breaches of Confidentiality, Integrity, and Availability (Attack type) of some Asset (if the type at all matters for the Loss Event).

Loss magnitude

Loss event
Type
Magnitude
Risk
Probability

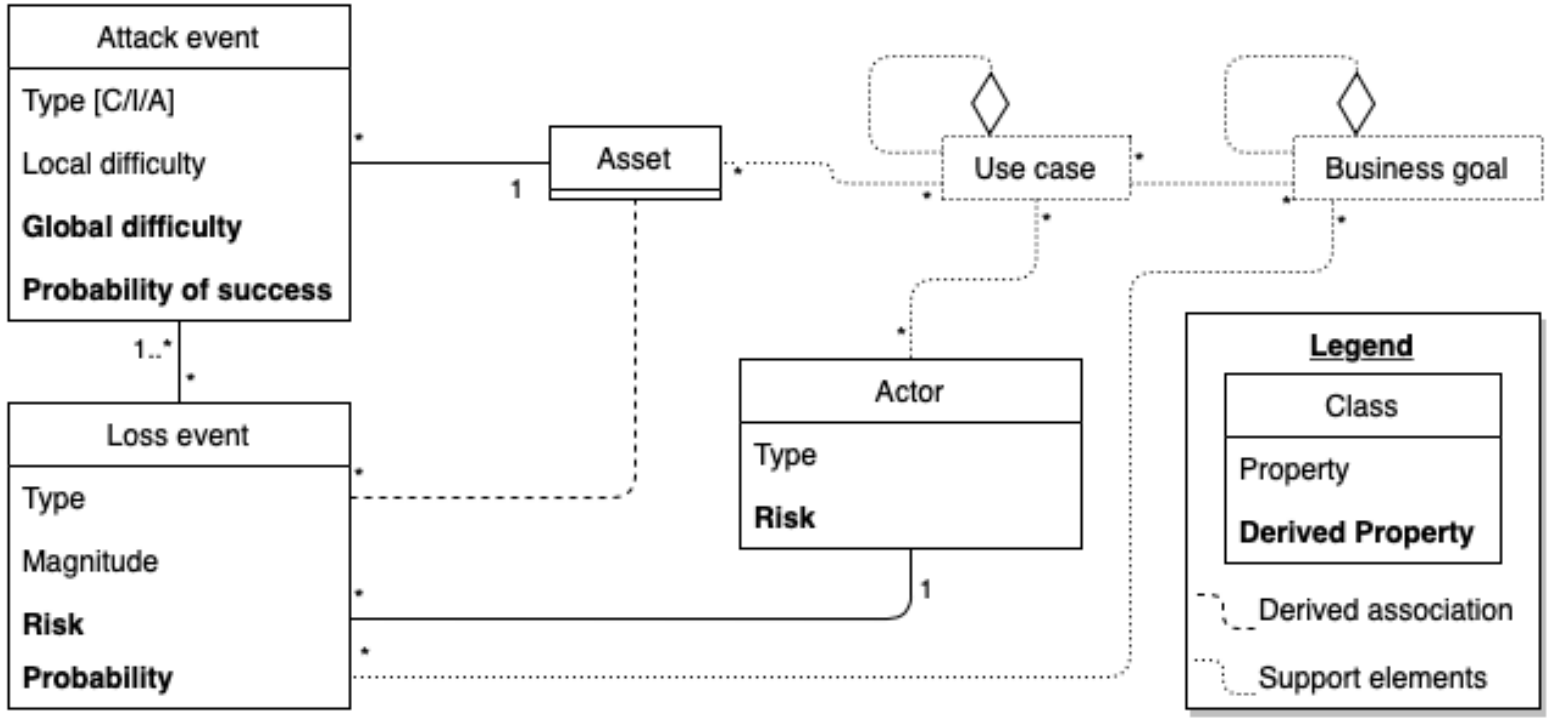
- It is normally difficult quantify loss. But we aim to be as quantitative as possible.
- Possible scales:
  - Monetary - preferred if possible
  - Qualitative scale (1-10, 1-5, low-high…)
  - Appropriate scales for specific loss types (customer satisfaction index, # injury/death, CO2 emission, …)
  - Not all scales can be multiplied with a probability to generate a risk score
    -> Risk matrix:



- Possibly multiple loss magnitude parameters needs to be used.

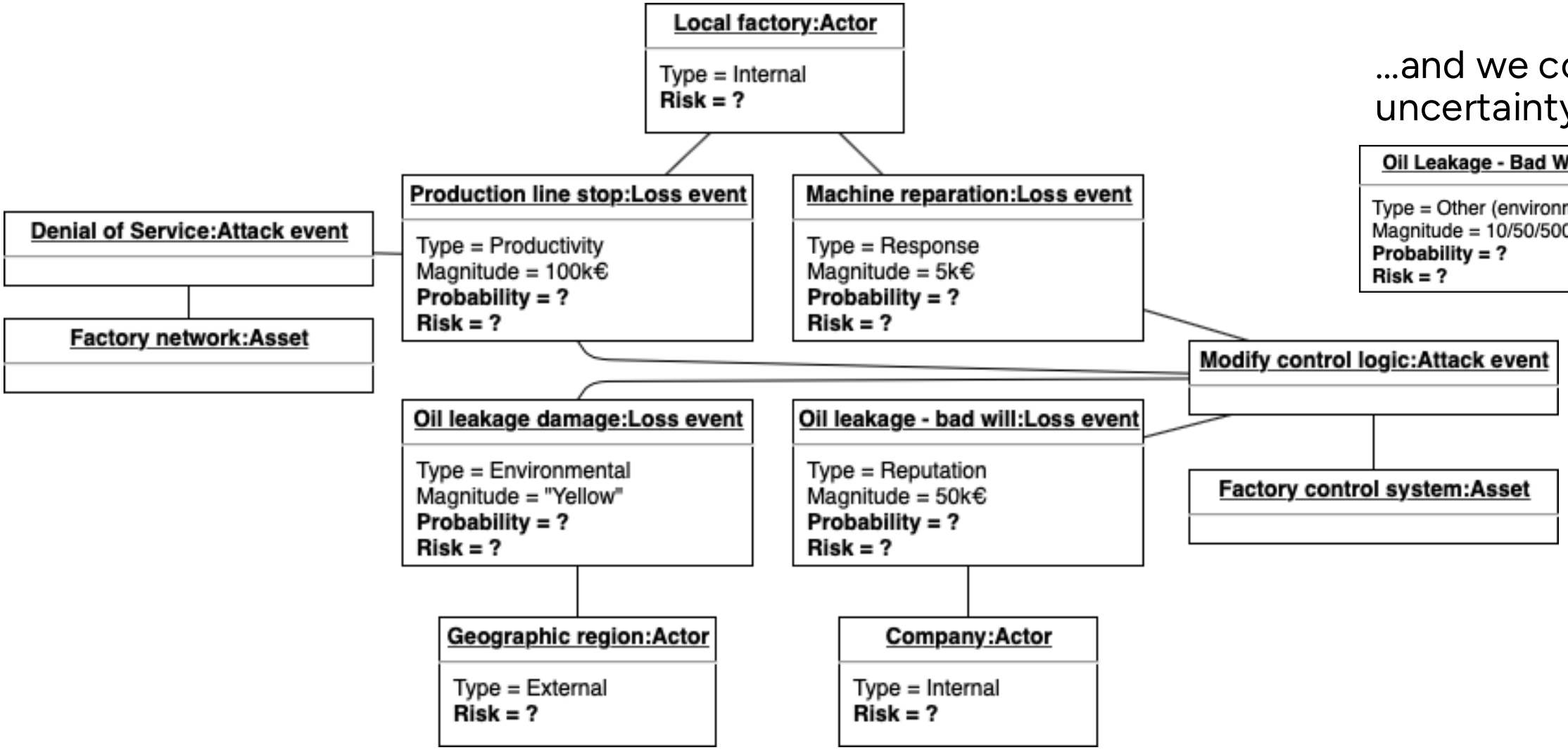
- All estimates have uncertainty. Not always is it expressed. For a threat model of high ambition, it should. Parameters are then expressed probabilistically
  - often three-point; 5% (low)/ 50% (expected)/ 95% (high) or minimum/ expected/ maximum
  - …otherwise any probability distribution.

Language overview



In phase 1 we do not necessarily care about the attacks per se. We might want to relate Loss events directly to Assets. This is however a simplification to ease this step. This relation is *actually* dependent on some attack targeting the Asset.

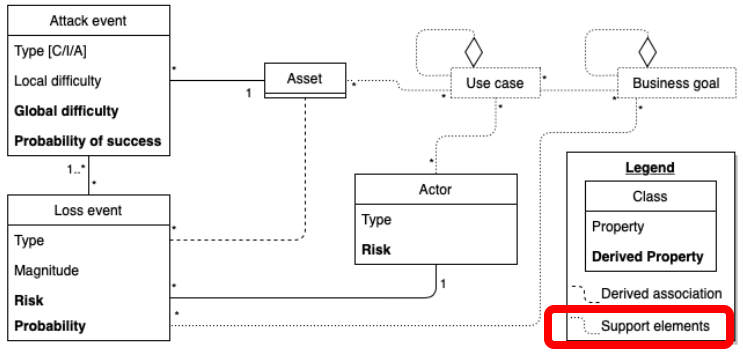
Example: Destroy factory machinery



...and we could add uncertainty ranges:

Oil Leakage - Bad Will:Loss Event

Type = Other (environmental damage)
Magnitude = 10/50/500 k€
Probability = ?
Risk = ?



A note on "support elements"

- As we have seen, for the risk analysis we need losses and their magnitude – and this we capture in the Loss event objects.
- Why bother with the business analysis, i.e. use cases and business goals?
  - They are only there to support you to identify and motivate the Loss event
  - The quality of your risk analysis, how correct it is and how much one can trust the figures in it, is determined by how well the loss events are explained by the business model.
    - If your loss event does not counteract any goal – does anyone, or more precise, what exact actor is suffering from the loss? E.g., if you goal is not to earn money (or maintain it) how can it be a loss to lose it, or if your goal is to keep your online store open during weekdays why would you care about a DoS attack Sunday night?
    - Another reflection is that if we turn the perspective around, in this phase we have now devised a requirements specification over what assets we want to protect against what type of attacks. So, if we forget some Loss event, caused by some type of attack, this will fall out of scope in our consecutive risk analysis
  - In real life business analysis and business performance rule. Your work as a cybersecurity risk analyst would include to understand and decipher what someone else - the business people – value and convert this into potential (cyber induced) loss events. Sometimes the impact chain/causality between a loss event and a business goal could be long and complicated, and sometimes it might be trivial.
    - We want this mapping to be transparent. For the business people, and for us.

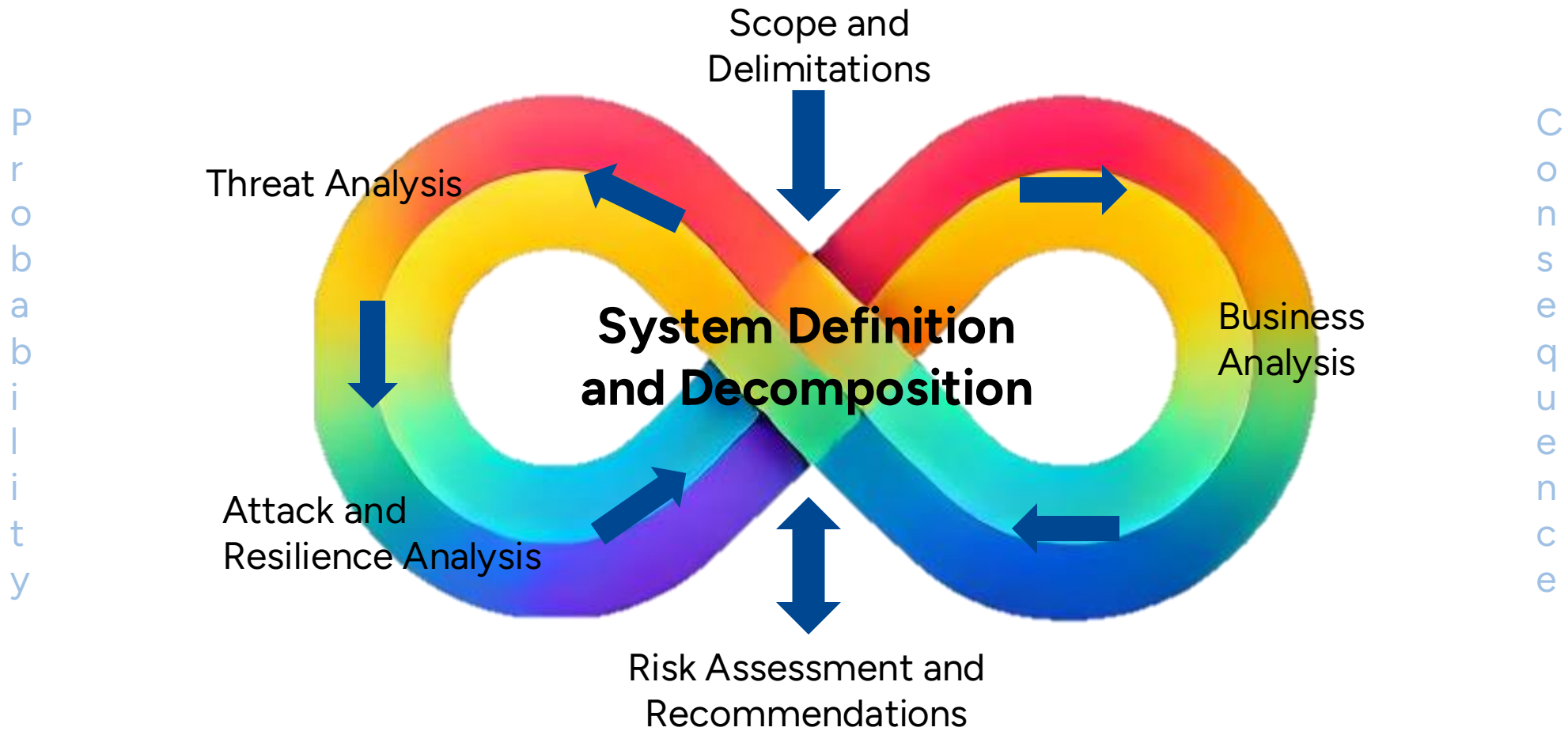
→ ***Be diligent and clear with the use of the business support elements!***

Phase 2
System Definition and Decomposition

Security Analysis of Large-Scale Computer Systems (EP2790) /
Cyber Security Analysis (EP279V)

(Suggested) Way-of-working

IT-system is our centerpoint (the cyber threat modelling best practice)



Phase 2 - System Definition and Decomposition

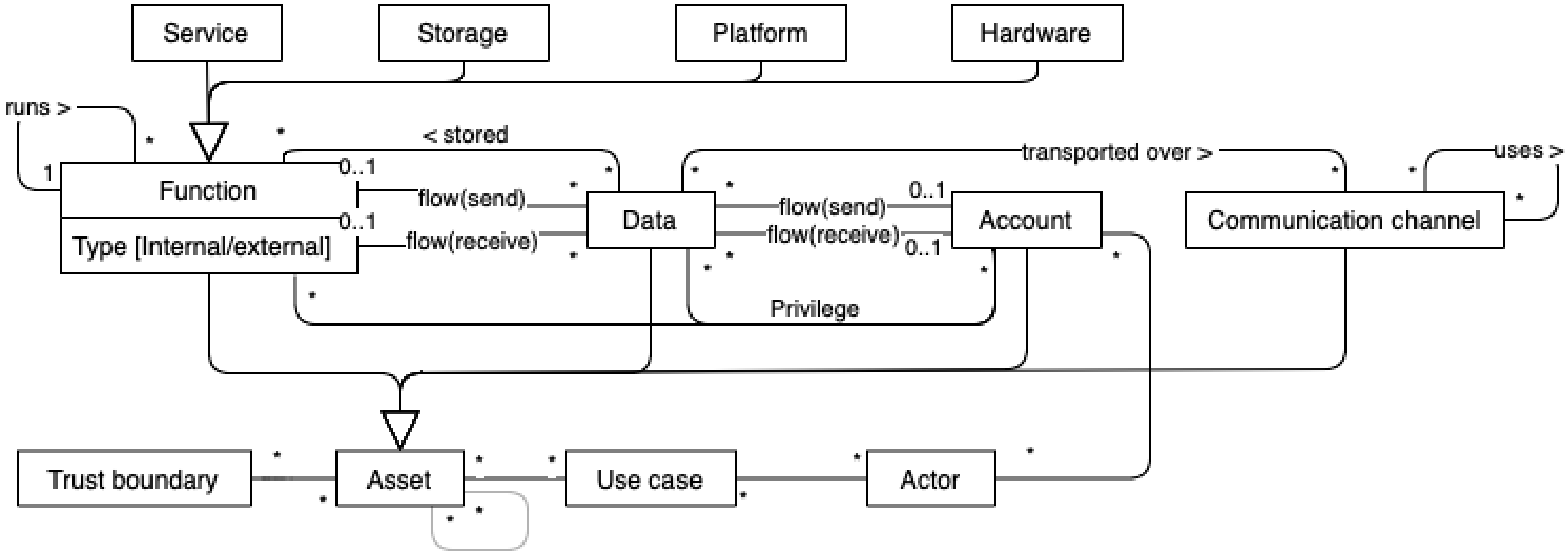
Given the decided scope (which might change with iterations), devise a full technical system specification and architecture.

Phase steps:

1. List (and refine) system assets/components
2. Develop data flow diagrams and system architecture

Possibly iterate these steps for several sub-systems if the overall system is large/complex.

Language overview

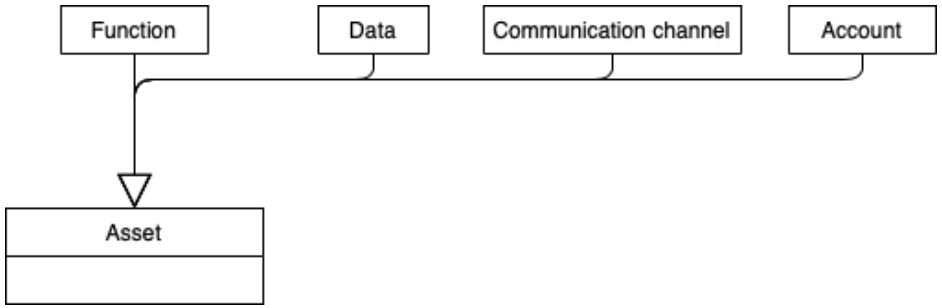


Step 1: List and refine system assets/components

Possible Input: Assets from Scoping or previous iterations.

Assets here refer to the constituents that make up the IT-system that is under evaluation, software and hardware. Since IT-systems are large and complex, it is common to use different types of assets when developing IT-system models. For threat modeling purposes (and in this course) we use the following types of assets:

- Functions
- Data
- Communication channel
- Accounts



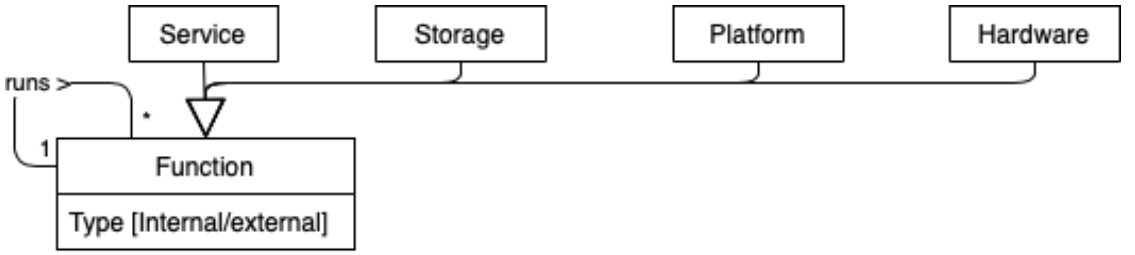
The concept of "Asset" can thus be seen as an abstract class. All Assets are of some type when looking close enough. When we do our system specification it might however be convenient to not always specify exactly what type an asset is. (For various reasons, e.g. that it is unknown at the time of modelling or because we in fact speak of a set of detailed assets jointly.)

This step does not necessarily lead to results presented a report in the end, It is more to be seen as a step producing working material for step 2.

Step 1: List and refine system assets/components

Functions. These are components that feature some kind of *behavior*. We use four types:

- *Services*; that are end user facing applications and features that per se deliver value to users
- *Platforms*; that are non-end user facing and constitutes some sort of infrastructure to services, e.g. OS, middleware, and web servers
- *Storage*; a function that keeps data, e.g. DB, file, cloud bucket/blob.
- *Hardware*; execution infrastructure for platforms and services



Even though our system is built up by all of these functions, we might not have access and ownership to all of them. To take a common example; we might develop a set of services to which are used by our customers, and these services are run in a cloud environment. The cloud environment we do not see the inner workings of, we only use it as a service, and the hardware we do not even see.

→ Functions can be internal or external

Moreover, some functions can execute other functions, e.g. hardware can run a platform that can run services.

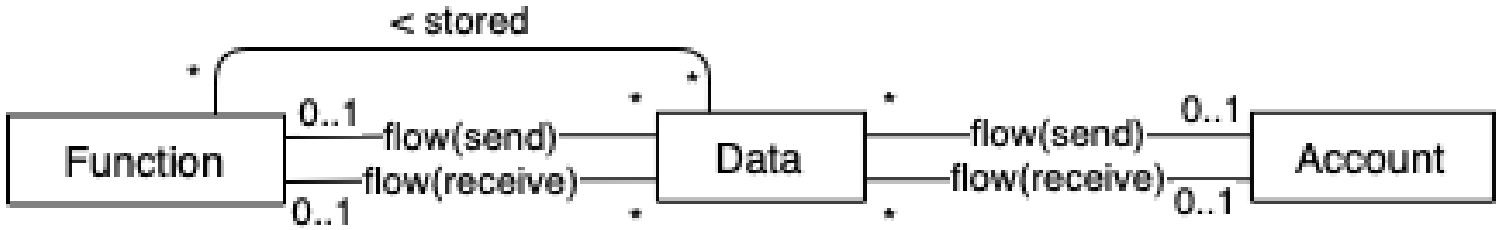
Step 1: List and refine system assets/components

Data. This represents *passive* components constituting some *information* in the system. It could be e.g. customer records, credentials, configuration data, financial transactions.

Data can be both at rest (in a storage) and in transit sent in messages between different functions.

Often data is the end target of cyber attacks.

A note: the data asset sometimes becomes a bit confusing in terms of differentiating between the data and the information in terms of the content or meaning of the data. The same information, say a specific customer record, might be stored in multiple data (in different DBs) and it may be sent around in transit between multiple functions. In general, we try to stay out of this complexity here by just avoiding to differentiate these things.



Step 1: List and refine system assets/components

Communication channel. This is any means that connect functions and and transmit data, ranging from the internet to a Bluetooth connection.

Most commonly we think of *networks* as communication channels. It is often convenient to think of networks as the communication protocol used (e.g. https). Looking at it this way the protocol represents a communication channel.

Furthermore, networks are built up of several layers of protocols (http/tcp/ip/ethernet) over some physical medium (cat5 cable). Normally we are not interested in all these layers, instead we simplify this stack by mentioning the parts that (we believe) are most important for the threat analysis (ftp over internet) and where (open wifi connected to internet).

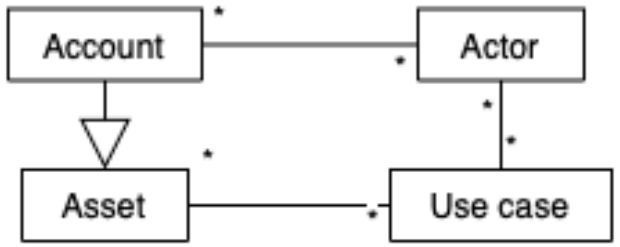
On top of all networks there is data transmitted. But where to "draw the line" and consider something as *data* rather than *communication channel* is a bit arbitrary and depend on what is most suitable for the case. (E.g. does SQL request represent communication infrastructure or all data in a database..?) Sometimes we want to differentiate between and analyze different specific data, and sometimes we can lump it all together and consider it as a grey mass (e.g. all web browser traffic over https). This depends on the purpose of the threat model. This in turn is largely decided in the attack analysis (phase 4), so iterations are encouraged.

Step 1: List and refine system assets/components

Actors and accounts.

Possible Input: Use cases and Actors from Phase 1.

We want to list who the actors are in the system. Actors may be both humans (assuming roles) and other systems. In order to interact with the system assets, the actors uses accounts. Accounts maybe have different names in different technical ecosystems they might also be more or less advanced in terms of refined structures with accounts being part of different Roles or Groups. Here we do not go into describing all of these variants. The important thing is that we want to be able to capture the different **privileges** for the different Accounts. We thus run into a challenge of choosing abstraction level of the model. For instance, we might end up representing a group of technical accounts as a single modeled account with a certain set of privileges (for convenience, if we do not care about individual accounts in the analysis.)



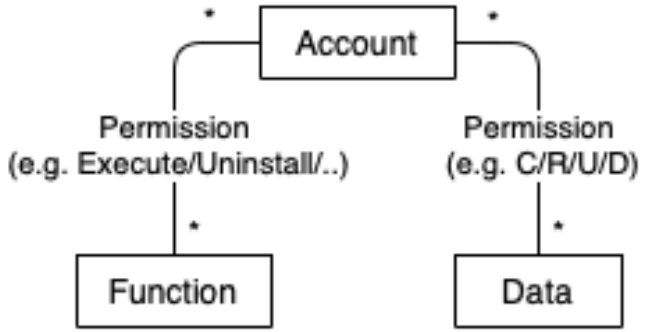
Step 1: List and refine system assets/components

Account privileges.

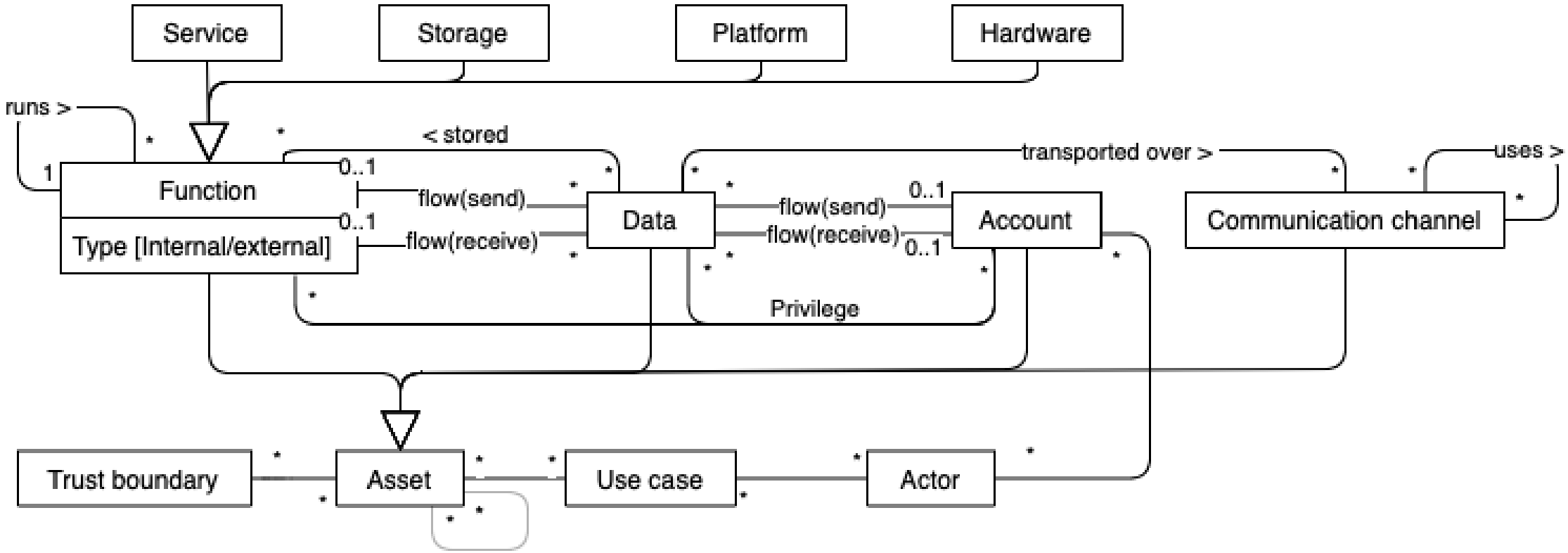
The accounts have permissions to do different things with functions and data in the system. Depending on the system environment the existing types of privileges may vary greatly. For data a common base privilege framework is Create, Read, Update, Delete (CRUD). (Other privileges could include operations like backup, restore, lock, share.) For functions common privileges include Start, Stop, (/Execute), Install, Uninstall, Configure. (Other privileges could include operations related to monitoring, execution scaling, permission setting).

We can capture these permissions by means of relations in the system model.

Privileges, accounts as well as authorization to them are normally encapsulated in an *Identity and Access Management (IAM) system* of a platform.



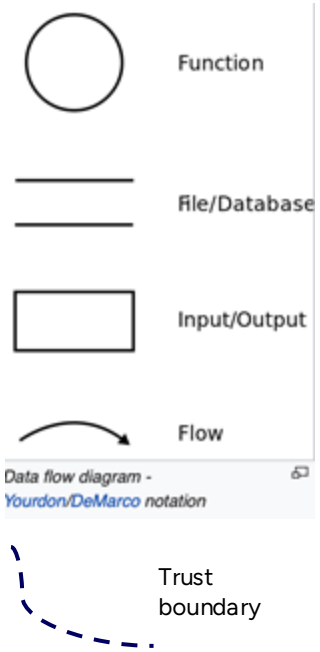
Language overview



Step 2: Develop data flow diagrams

Data flow diagrams (DFDs) are traditional models from software engineering. As the name implies it describes how data is flowing between parts of a software program (or system at large in our case). There are four model constructs in the original DFD;

- Data flow
- Data-at-rest
- Function
- Input/Output

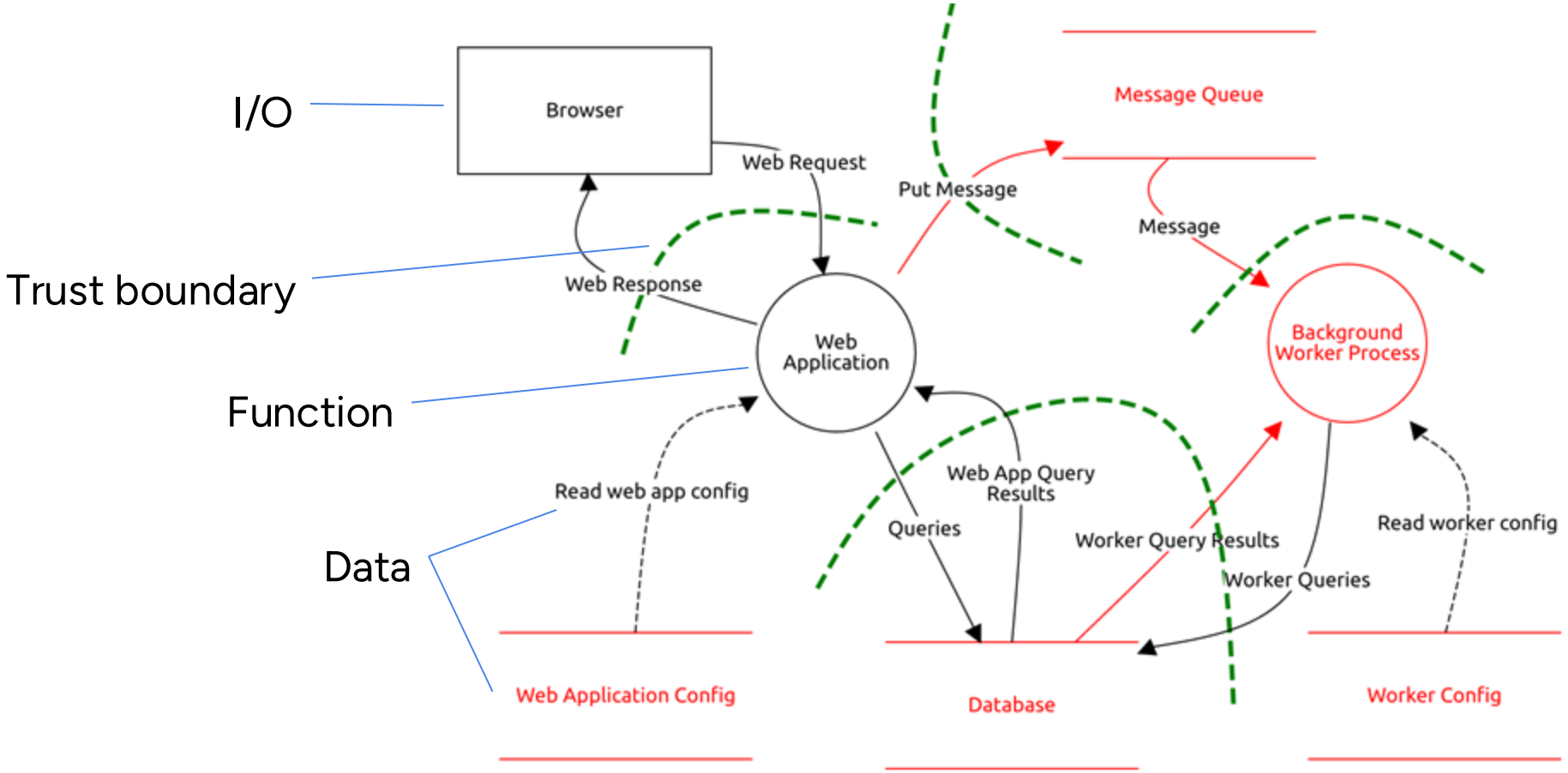


Later, for threat modeling, also *Trust Boundary* has been added. Typically, this is called the “Threat Model”.

Trust boundaries

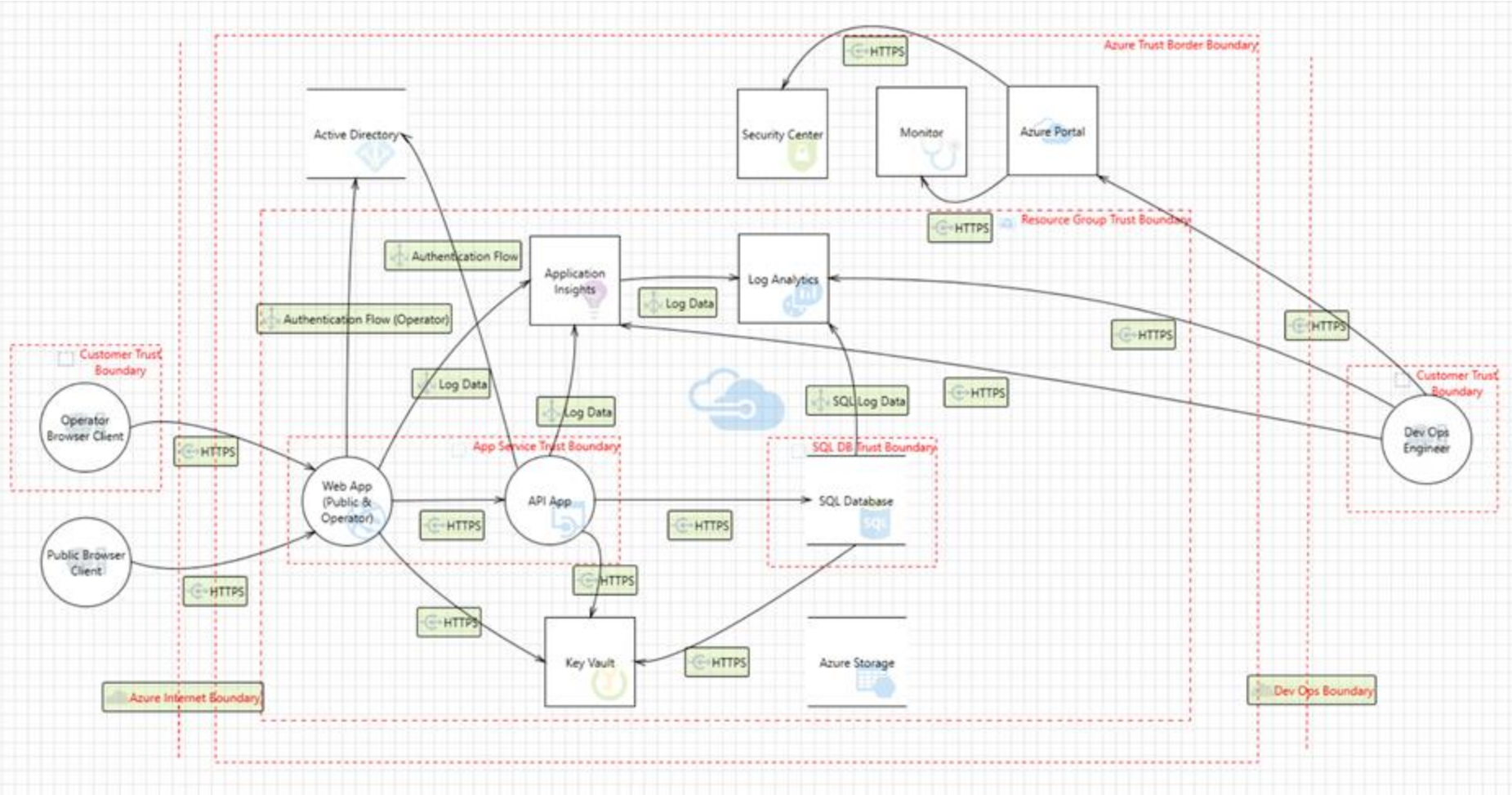
- A trust boundary essentially represent a place where additional precaution needs to be taken when data is traversed over the boundary. Within a boundary (a zone) we assume that there is little or no need to control correctness or legitimacy of requests and responses.
- The reason to add a trust boundary could be several, e.g.:
  - the system is not in our domain of control
  - the code base is different
  - actors belong to different types of groups (development/test/sales)
  - the level of testing/quality assurance performed differs
  - …
- Since the term is not strictly defined, we should primarily think of trust boundaries as a concept to support design rather than something that objectively exist (in the way that we normally think of data and functions to objectively exist).

DFD / threat model example



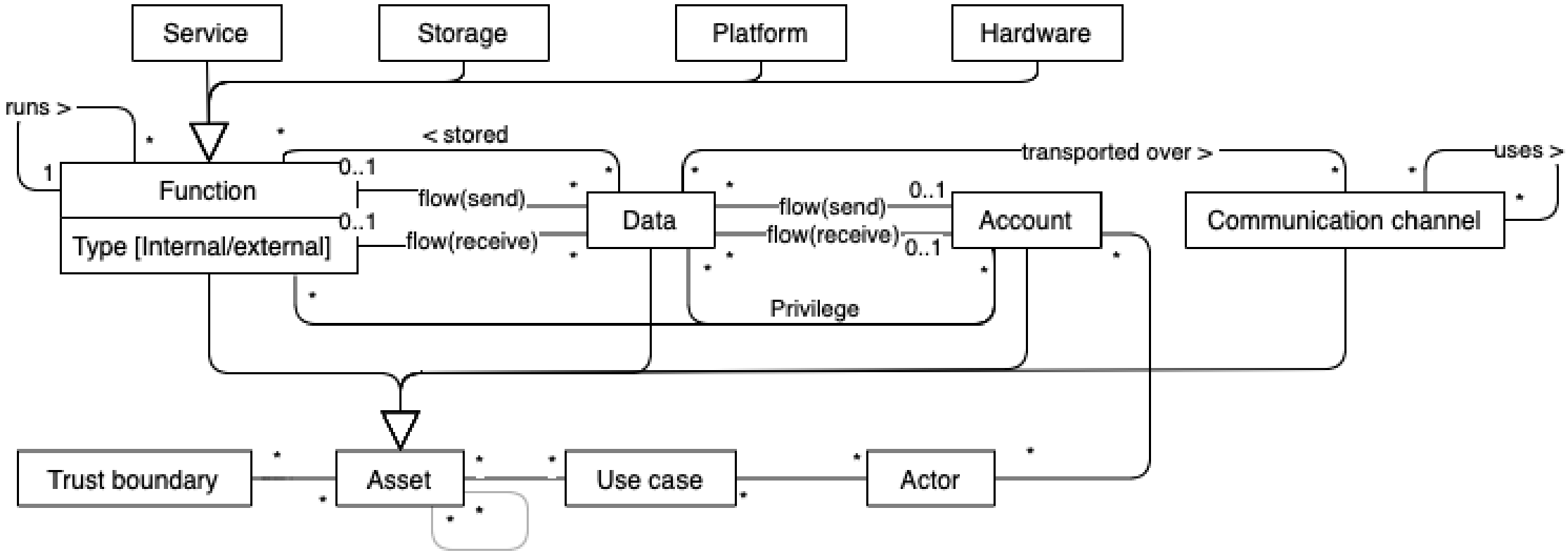
OWASP Threat Dragon tool - free online

DFD / threat model example



Microsoft Threat Modeling Tool - free download

Our language

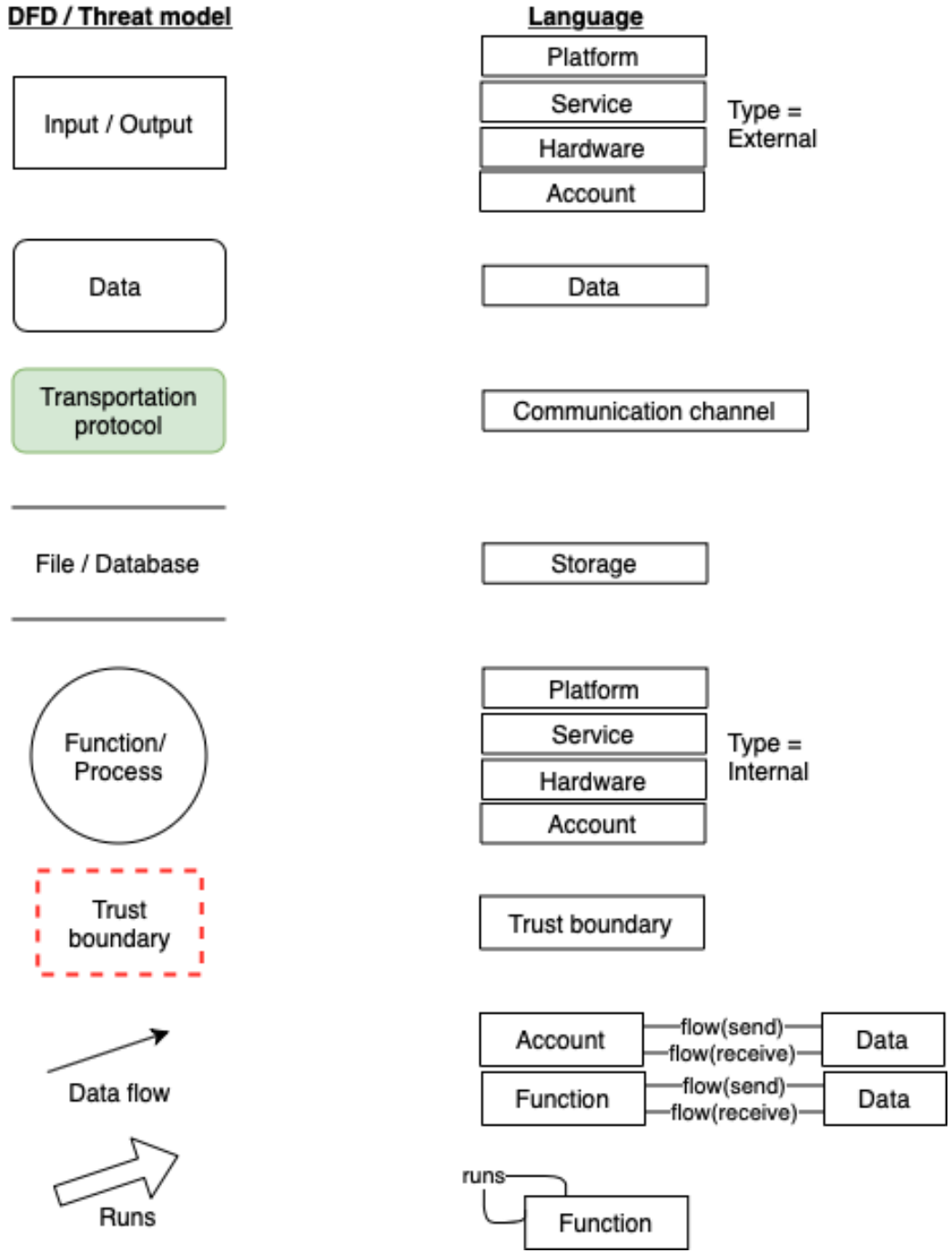


Mapping DFD/threat models to Yacraf language

Comparing legends →

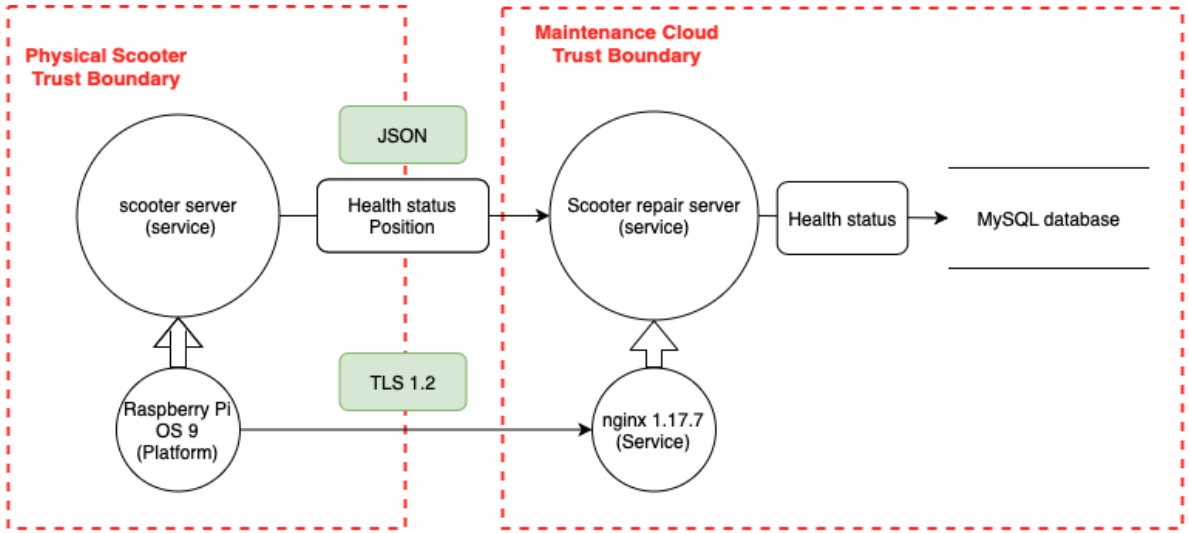
(Threat models are however normally not explicit with what underlying language they use…)

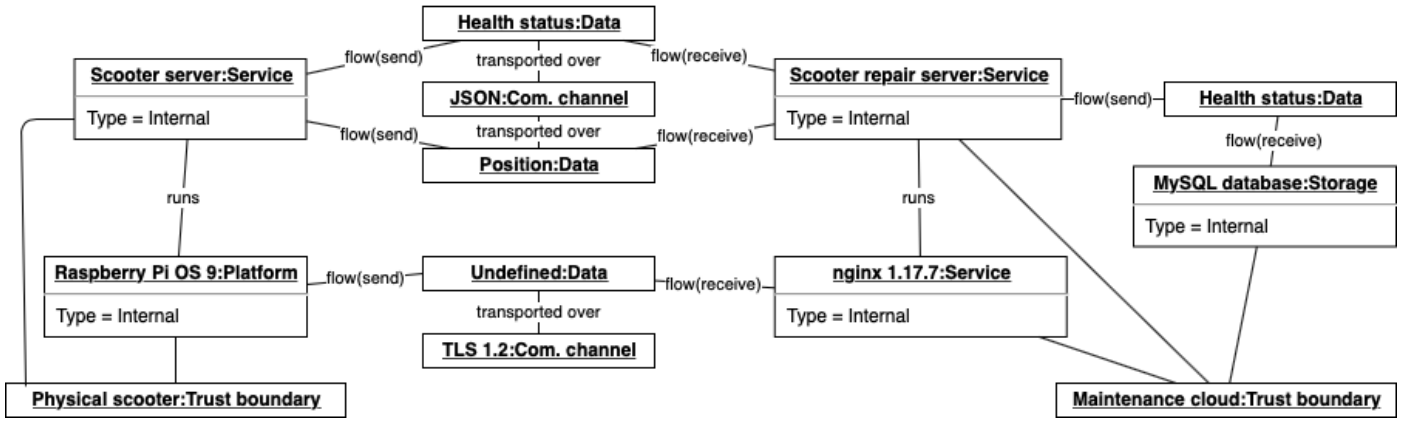
Privileges are not clearly modeled in threat models…



Mapping DFD/threat models to Yacraf language

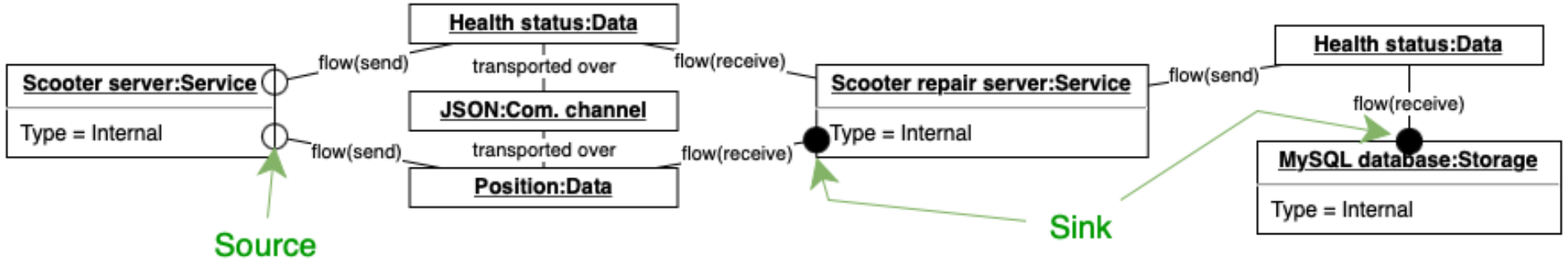
An example model:





The flow in DFDs

If we want to clarify the dataflow aspect of DFDs one common trick is to highlight the system-wide start point/source and end point/sink of each data flow.
(This helps the subsequent security analysis.)



Where is the system border?

In general we are interested in modeling the IT-system architecture. Assets are then digital components. However, our overall goal is to do security risk assessments. Depending on the observed system it might have important “analogue” parts of the system that might invalidate the analysis at large unless included. In such cases we need to also incorporate these dimensions. (Say that pieces of paper with QR-codes act as an important part of the system architecture, then we should not disregard that data just because it is printed.) Moreover, human behavior is also a key ingredient in many systems.

Now, this is not to say that we should not scope our studied system at all and always take everything into account. ***The focus is still digital systems.***

How deep and wide do you go?

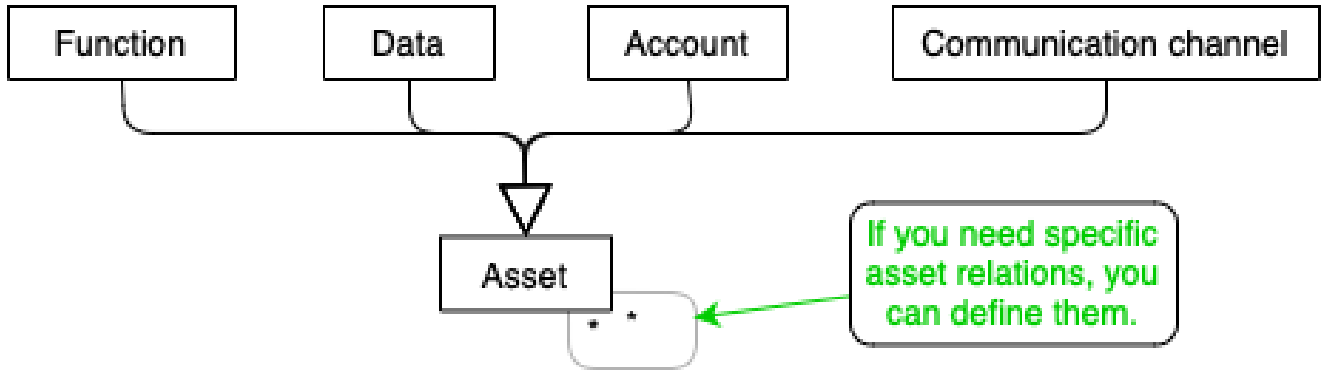
... As much as time allows!

We want to have sufficient detail about the system where we believe that there might be vulnerabilities, or vulnerabilities that we consider interesting to analyze. E.g. if we fully trust an end-to-end crypto solution there is little point in detailing the the network the encrypted data is traversing. Instead, we might be more interested in the quality of the end points and the key distribution process. To motivate what you trust and why is thus important to the over all analysis.

Of course, there is a hen-and-egg problem with the above; detailing the system is a means to figure out where there might exist vulnerabilities. Another principle is to look closer at assets close to loss events of high impact magnitude. However, neither this is a fully working strategy as risk depend on all parts of the threat model (the impact, the threat actor, the system resilience). So, if we detail around the largest impacts, we might miss important details about the system vulnerabilities that might drive the largest risks in the end.
→ ***Ideally, we should thus iterate the phases many times.***

Some general notes on system modeling

- How to model IT-systems is a topic of much discussion and research. There exist a large variety of suggested modeling languages all abstracting the IT-system slightly different and all modeling slightly different things.
- At the end of the day, what and how to model is determined by the purpose of modeling. For threat modeling languages we want to capture the most important things related to security.
- Also, for all modeling languages we want to keep them as simple as possible while still capturing the essentials (i.e. in our case security).



Phase 3
Threat Analysis

Phase 3 - Threat Analysis

“Threat” (in this course) covers the attacker and its properties. What attackers are we facing threat from? How eager are they to attack? How good are they at attacking?

Output: assumed attacker profiles and activities.

Phase steps:

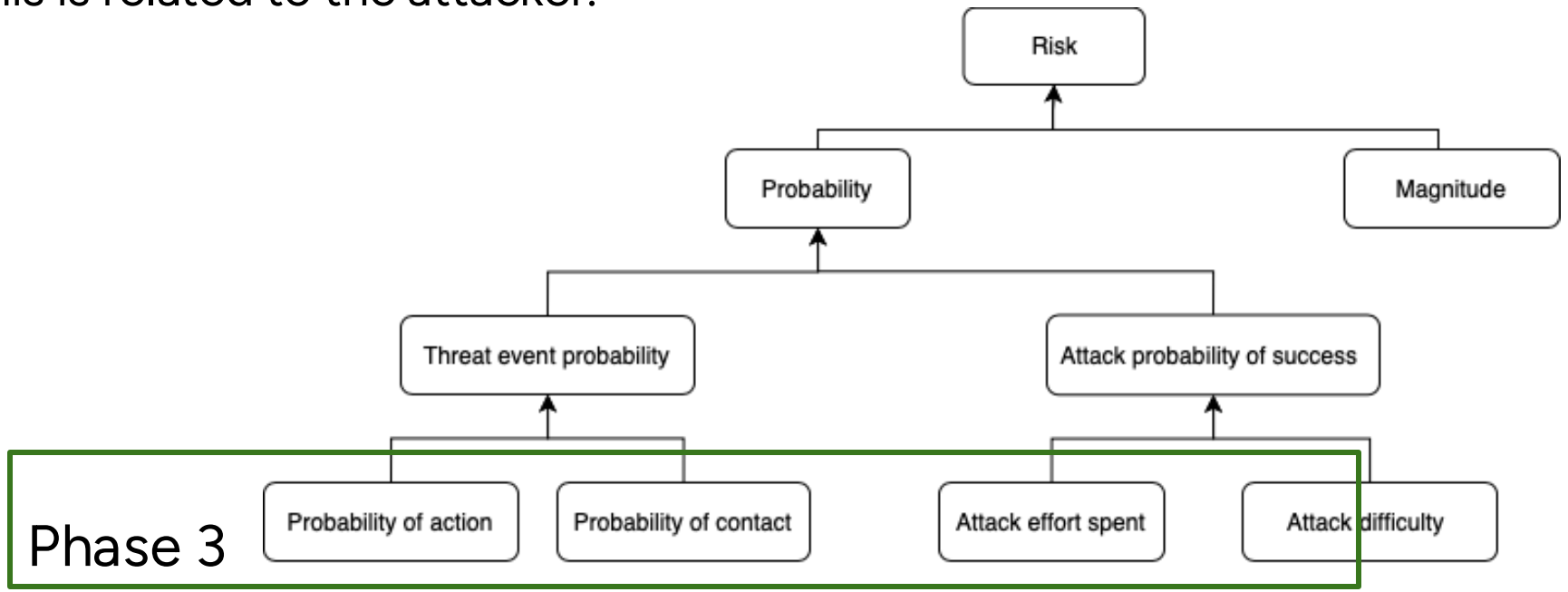
1. Develop attacker profiles
2. Develop abuse cases

Phase 3 purpose

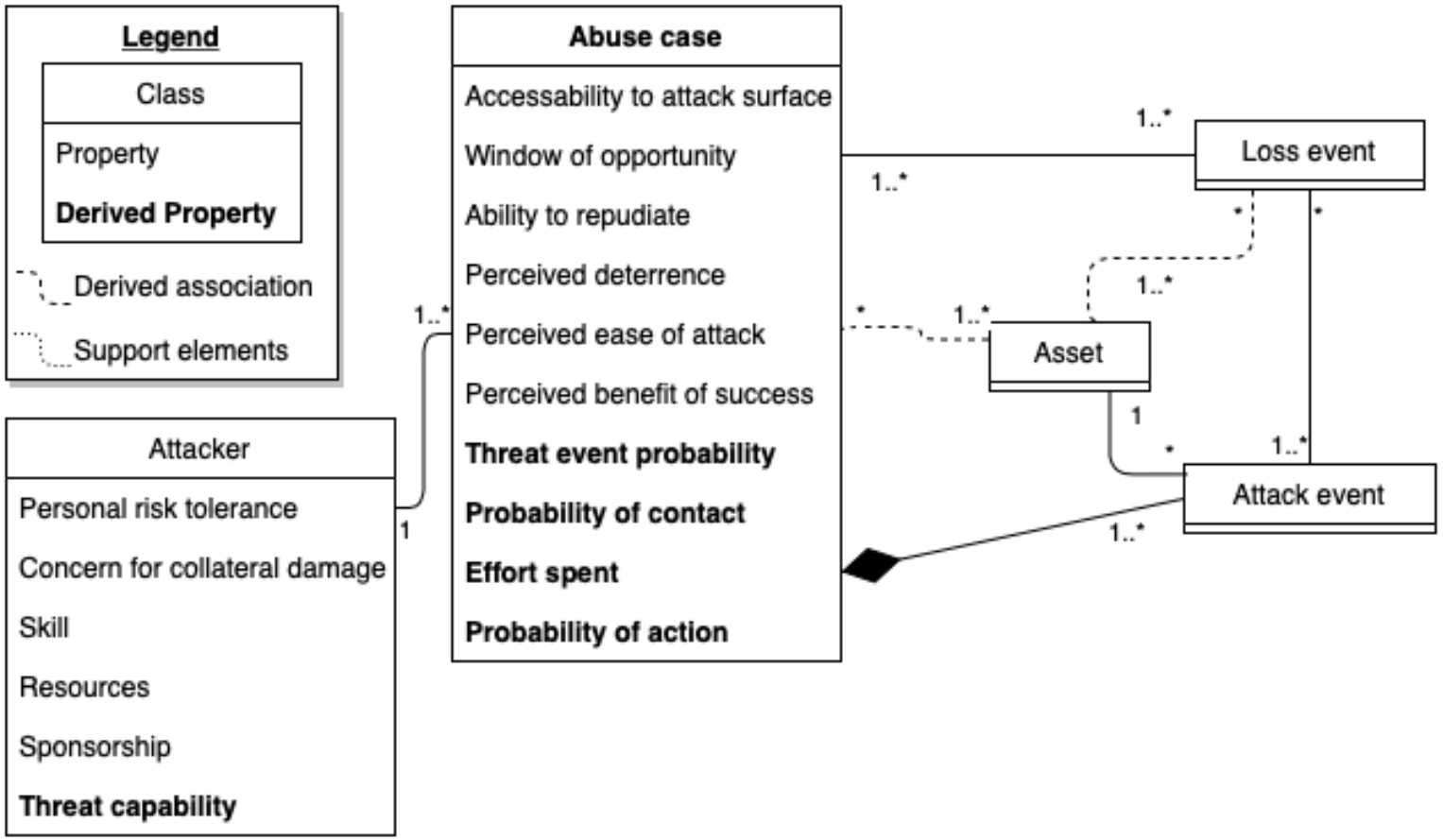
We want to come up with estimates on the

- Contact probability
- Probability of action
- Attack effort spent

This is related to the attacker.



Language overview



Step 1:
Develop attacker profiles

Who are our antagonists? - This is typically very difficult to know!

We need cyber threat intelligence on attackers and/or relevant attacks.
(Some potential sources: https://attack.mitre.org/groups/, https://kb.cert.org/, https://www.us-cert.gov, https://www.sans.org/reading-room/whitepapers/threats/creating-threat-profile-organization-35492)

- Decide on what attackers to include in the threat analysis.
- Estimate *Threat Capability* for these attackers respectively.

Nation states

Frequently reported on: USA, UK, (Five Eyes), EU, Russia, China, North Korea, Iran, …

Complicated internal organizations. E.g. Russia (https://www.cisa.gov/uscert/ncas/alerts/aa22-110a)

- **The Russian Federal Security Service (FSB) (a.k.a. Berserk Bear, Dragonfly, Energetic Bear, …)**

… has conducted malicious cyber operations targeting the Energy Sector, including UK and U.S. energy companies, U.S. aviation organizations, U.S. government and military personnel, private organizations, cybersecurity companies, and journalists. FSB has been known to task criminal hackers for espionage-focused cyber activity; these same hackers have separately been responsible for disruptive ransomware and phishing campaigns.

- **Russian Foreign Intelligence Service (SVR) (a.k.a. Cozy Bear, Nobelium, Yttrium, …)**

… has targeted multiple critical infrastructure organizations. Responsible for the *SolarWinds Orion* supply chain compromise and the associated campaign that affected U.S. government agencies, critical infrastructure entities, and private sector organizations. *2016 US elections*.

- **Russian General Staff Main Intelligence Directorate (GRU), 85th Main Special Service Center (GTsSS) (a.k.a. Fancy Bear, Sofacy, Strontium, …)**

…primarily targets government organizations, travel and hospitality entities, research institutions, and non-governmental organizations, in addition to other critical infrastructure organizations. E.g., *2016 US elections*

- **GRU's Main Center for Special Technologies (GTsST) (a.k.a. Sandworm, Voodoo Bear, Electrum, …)**

…has targeted a variety of critical infrastructure organizations, including those in the Energy, Transportation Systems, and Financial Services Sectors. Cyber espionage as well as destructive and disruptive operations against NATO member states. E.g. *2015&2016 power blackouts in Ukraine, NotPetya "ransomeware", 2018 winter olympics/paralympics*

- **Ministry of Defense, Central Scientific Institute of Chemistry and Mechanics (TsNIIKhM) (a.k.a. Temp.Veles, XENOTIME)**

Actors associated with TsNIIKhM have developed destructive control systems malware. *E.g. Triton/Trisis*.

The 9-to-5 criminals

E.g. Interview with **BlackMatter** - A ransomware operator

(https://therecord.media/an-interview-with-blackmatter-a-new-ransomware-group-thats-learning-from-the-mistakes-of-darkside-and-revil/)

Q: […] How long ago did you start developing [BlackMatter]?

A: Before starting the project, we studied the following products in detail: LockBit (…has a good codebase, but a skimpy and non-functional panel [..]. If you compare it to a car, you can say that this is a Japanese car production line with good engines but an empty and non-functional interior. You can ride one, but with little pleasure.), REvil (…), Darkside (…). The executable itself has incorporated the ideas of LockBit, REvil, and partly DarkSide.

Q: LockBit 2.0 is considered the fastest locker at the moment. What is the encryption/decryption speed of your variant?

A: This is not true. After reading the question – we decided to prepare ourselves by downloading the latest publicly available version of LockBit (end 06.21) and conducting tests, we can state the following: BlackMatter: 2.22 LockBit: 2.59. The tests were carried out under the same conditions. Moreover, LockBit encrypts the first 256 kb of the file (which is pretty bad from the point of view of cryptographic strength). We, on the other hand, encrypt 1 MB. Essentially, that's the secret to their speed.

A: "We created a project and brought it to the market exactly at a time when the niche is vacant and the project fully meets the market demands, therefore its success is inevitable."

Q: Most recently, the largest groups—DarkSide, REvil, Avaddon, BABUK—have disappeared from the scene.

A: Yes, we believe that to a large extent their exit from the market was associated with the geopolitical situation on the world stage. […] (*Note: At this time Biden was messaging to Putin that he needed to take better control over cyber criminal groups.*)

Q: Obviously, there are many talented professionals on your team. Why is it that this talent is aimed at destructive activities? Have you tried legal penetration testing?

A: We do not deny that business is destructive, but if we look deeper—as a result of these problems new technologies are developed and created. If everything was good everywhere there would be no room for new development. There is one life and we take everything from it, our business does not harm individuals and is aimed only at companies, and the company always has the ability to pay funds and restore all its data. We have not been involved in legal pentesting and we believe that this could not bring the proper material reward.

Q: What do you think about the attacks carried out against Colonial Pipeline's infrastructure or JBS? Does it make sense to attack such large networks?

A: We think that this was a key factor for the closure of REvil and DarkSide, we have forbidden that type of targeting and we see no sense in attacking them.

Q: Tell me a secret.

A: There are no secrets, but we believe in our motherland, we love our families, and we earn money for our children.

The cowboy youngsters

E.g. Reporting on **LAPSUS$** from leaked Telegram chats

(https://krebsonsecurity.com/2022/04/leaked-chats-show-lapsus-stole-t-mobile-source-code/)

LAPSUS$ is known for **stealing data and then demanding a ransom** not to publish or sell it.

…But the leaked chats indicate this mercenary activity was of little interest to the tyrannical teenage leader of LAPSUS$, whose obsession with **stealing** and leaking proprietary computer **source code** from the world's largest tech companies…

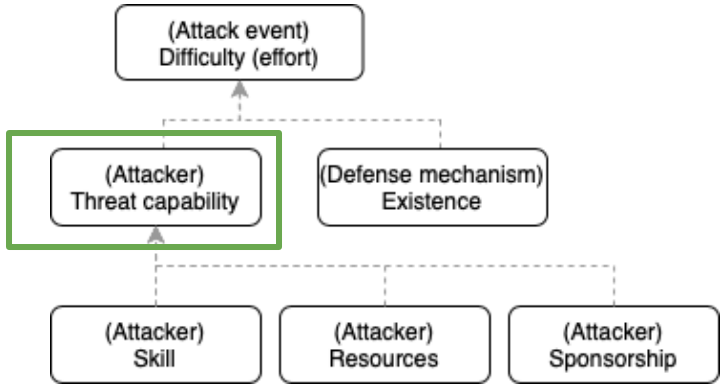
In early April (2022), multiple news outlets reported that U.K. police had arrested **seven people aged 15-21** in connection with the LAPSUS$ investigation.

In one chat, the **LAPSUS$ leader** — a 17-year-old from the U.K. who goes by the nicknames "**White**" […] is sharing his screen with another LAPSUS$ **member** who used the handles "**Amtrak**" […]. […] The vast majority of noteworthy activity documented in these private chats takes place between White and Amtrak, but **it doesn't seem that White counted Amtrak or any of his fellow LAPSUS$ members as friends or confidants**. On the contrary, White generally behaved horribly toward everyone in the group, and he particularly seemed to enjoy abusing Amtrak.

E.g.: White also appeared unconcerned when Amtrak admits that the City of London police found LAPSUS$ Telegram chat conversations on his mobile phone. Perhaps to demonstrate his indifference (or maybe just to screw with Amtrak), White responds by leaking Amtrak's real name and phone number to the group's public Telegram channel

E.g.: "Mox", one of the LAPSUS$ members who shows up throughout these leaked chats […]. At one point, Mox's first name briefly shows up in a video he made and shared with the group, and Mox mentions that he lives in the United States. White then begins trying to find and leak Mox's real-life identity.

Attacker capability



How difficult it will be to succeed with an attack depend on the attacker´s abilities. We summarize this in the property Threat capability.
FAIR methodology (Freund&Jones) thinks of this as a percentage:

"*Threat capability* is simply a percentile scale 1-100 which represents the comprehensive range of capabilities for a population of threat agents. The least capable agent represents the first %, and most the capable the last %"

Attacker
Personal risk tolerance
Concern for collateral damage
Skill
Resources
Sponsorship
Threat capability

Depend on (*e.g.*):

- Skill (quality, domains of expertise, ..)
- Resources (time, head count, tools, ...)
- Sponsorship

Estimate Threat capability percentage based on qualitative assessment and combination of these underlying properties.

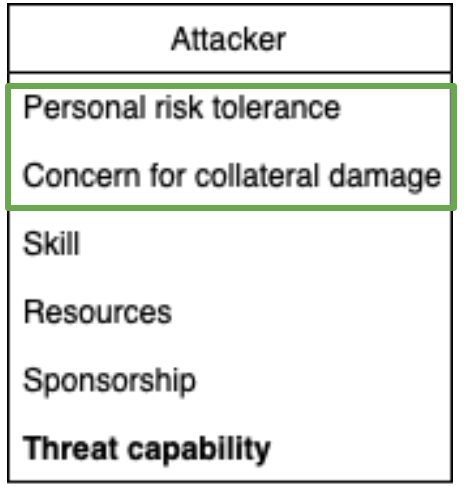
The Threat capability as an explicit number is not strictly needed for our further risk calculations but it can be convenient to estimate as part of the risk assessment process

Attacker attitude

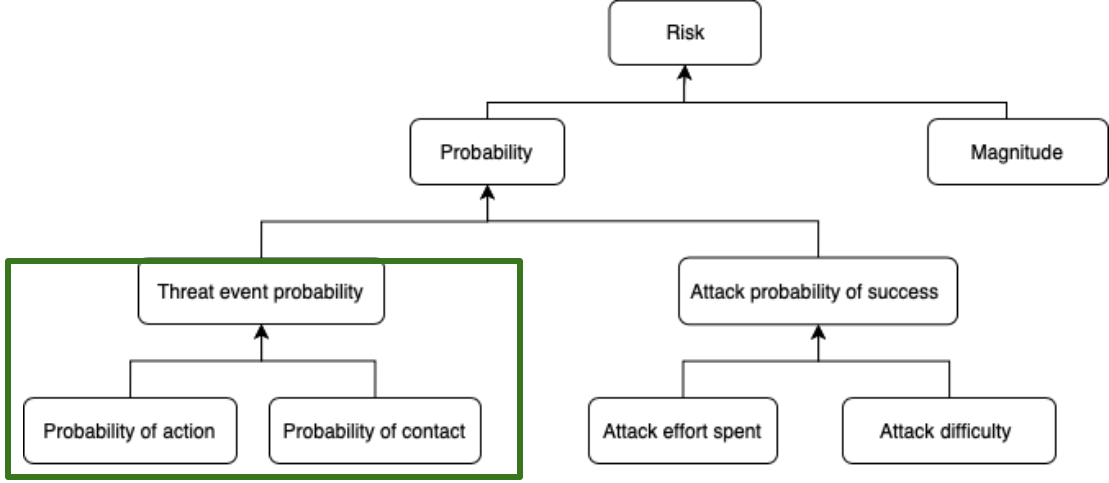
The attacker attitude will also impact the risk analysis. This is influenced by many factors. Freund & Jones discuss two that we also consider:

- Risk tolerance (how much are they concerned with getting caught and willing to self sacrifice)
- Concern for collateral damage

Essentially these factors impact what type of attack activities the attacker is willing to perform the abuse cases



Step 2:
Develop abuse cases



Develop abuse cases and their respective *Threat event probability*.

- *"An abuse case is a type of complete interaction between a system and one or more actors, where the results of the interaction are harmful to the system, one of the actors, or one of the stakeholders in the system."* Like a use case, but malicious.

- Here we think of abuse cases as representing potential goals of actions and intent of some specific attacker. (So this relates/explains the term threat. Threat is an action that *could* happen.) An abuse case encodes a threat.

Abuse case - context

- Abuse cases are close to Loss events:
  Loss events represents bad things that can happen (and there we want to capture how much it hurts if it happens) and Abuse cases are ways of doing these bad things.
- We assign Abuse cases to exactly one Attacker (Abuse case attributes relate to individual Attackers). Abuse cases thus encodes what we assume the attacker will do.
- Bottom line, an Abuse case is a number of/chains of Attack events; attack scenarios. (This is elaborated in the next phase.)
- Abuse cases essentially specifies the start and end of the attack chains. The starting point is thought of as the attack surface, and the end is the when a successful breach is achieved that will lead to the Loss event.
- In this phase we do not go into details about the attacks. Instead we relate the abuse cases to the Assets in the system specification.
- We need to name (or otherwise annotate) the Abuse case so that it illustrates the approximate attack chain (to the extent it is important).

Abuse case (intrinsic) properties

The abuse case has a number of properties related to our beliefs about the threat/abuse case being executed.

- Accessibility to attack surface (attack surface is not necessarily the asset that the loss event impact relates to)
- Window of opportunity (This is mainly intended to represent the system asset design. E.g. a targeted communication channel is only up two hours a year.)
- Ability to repudiate (is it possible to deny an attack)
- Perceived deterrence (consequence of being caught)
- Perceived ease of attack
- Perceived benefit of success
- (Again, possibly there are more factors of relevance, this covers the the big picture..)

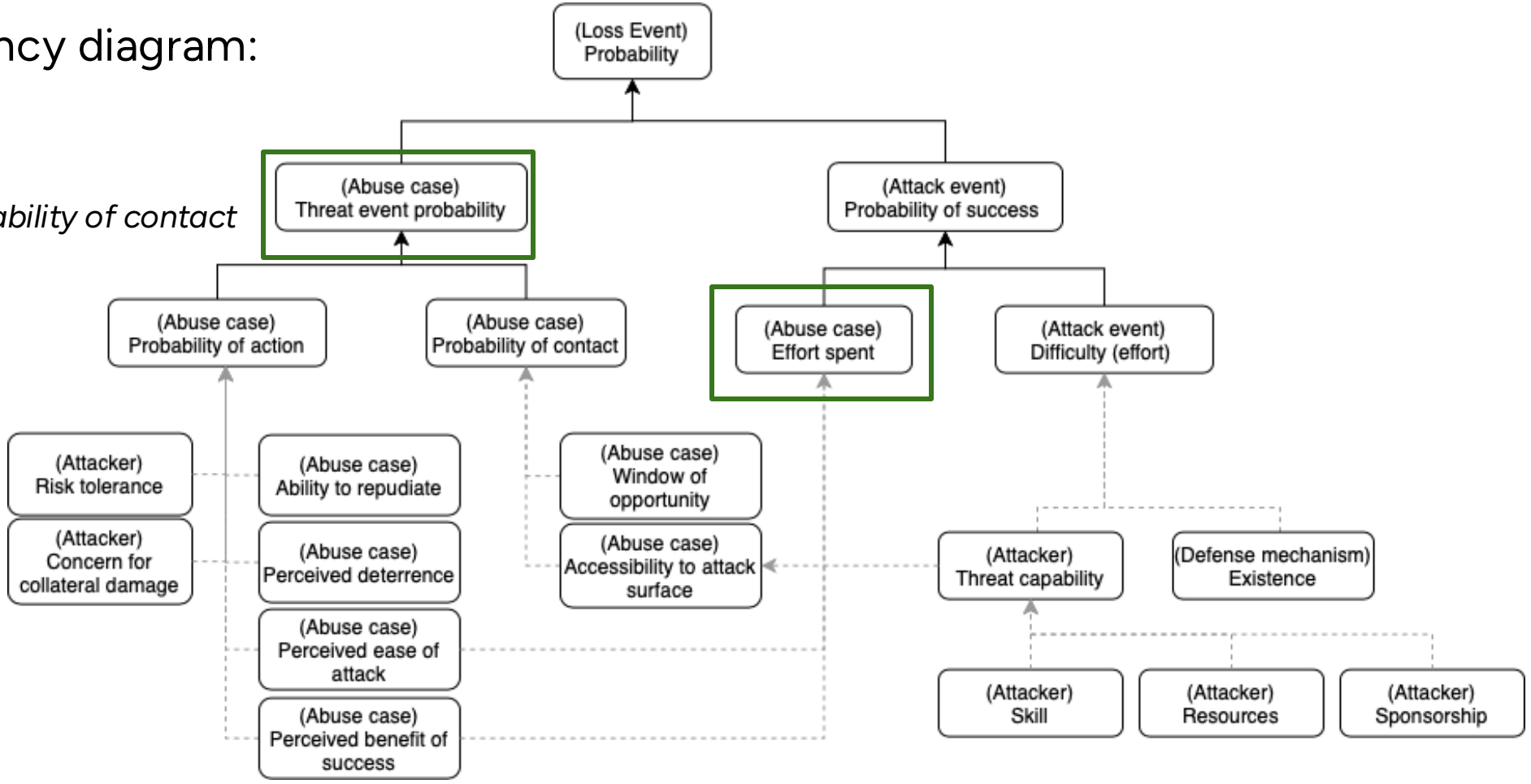
Deriving abuse case properties

We have two derived abuse case properties Threat event probability and Effort spent, essentially how likely is it that the attackers will execute and how much effort will they spend on it.

Dependency diagram:

*Threat event probability = Probability of action * Probability of contact*

Dashed lines - qualitative assessment and combination of underlying properties.



Assessing and deriving attributes

- Assessing threat (to know your enemy) is inherently difficult.
- In practice you can elicit knowledge and opinions from various experts in the organization and from various cyber threat intelligence (CTI) resources.
  In this course you have:
  - yourselves (make as good guesses as as you can)
  - the Internet, e.g. open CTI (back up as much of your guesses as you can/is reasonable given the time with references to others. The more experts they are the better!)
- In general make assessments as formal as possible. For threat/attacker assessments you need to make many qualitative assumptions and aggregations (i.e. all the dashed lines in the dependency diagram)

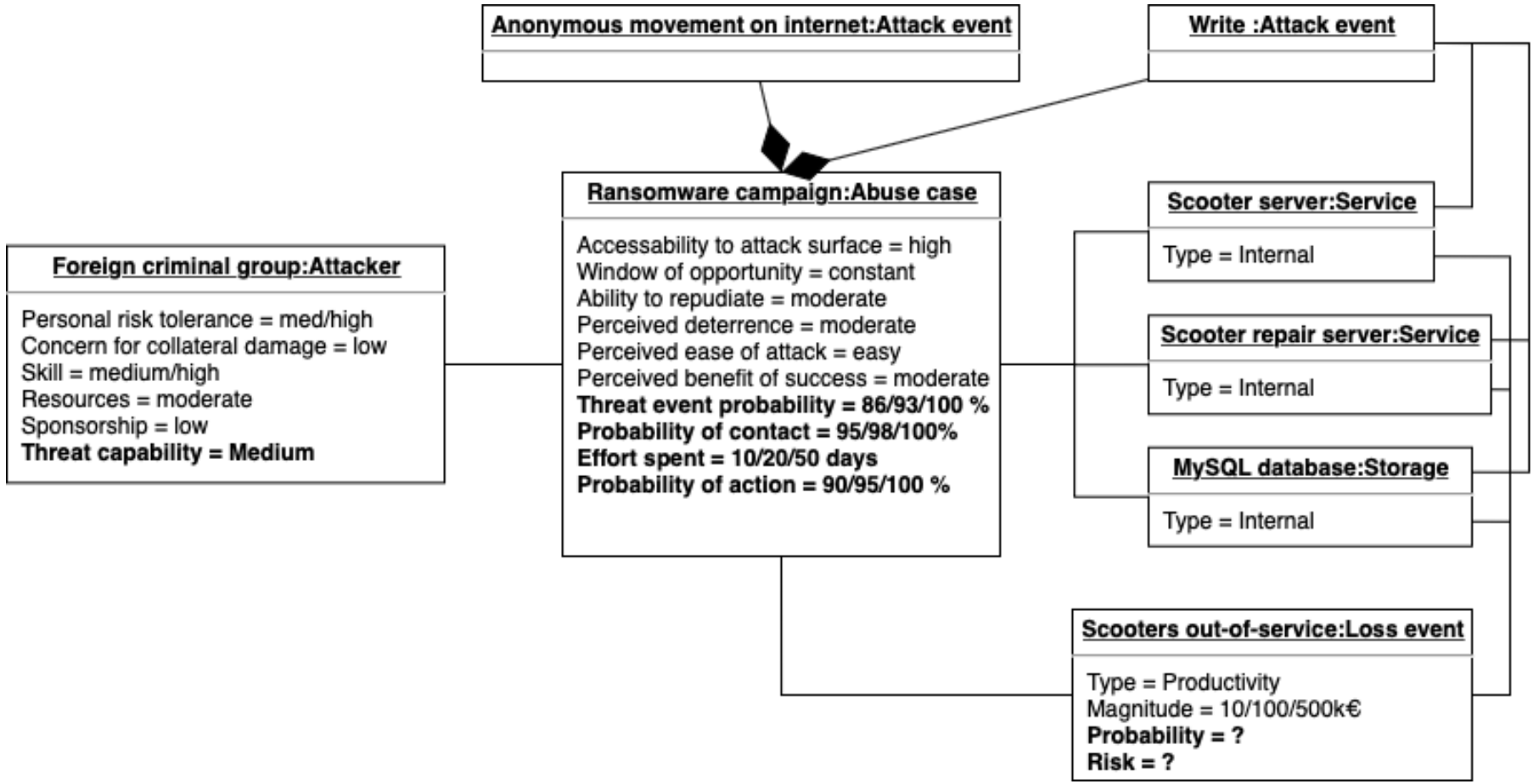
  ⇒ ***Be transparent in assessments and derivations.***

Probabilistic assessment

All estimates have uncertainty. Not always is it expressed. For a threat model of high ambition it should. Parameters are then expressed probabilistically

- often three-point; 5% (low)/ 50% (expected)/ 95% (high)
  or worst/ expected/ best case
- ..otherwise any probability distribution.

A simple model example



Phase 4
Attack and Resilience Analysis

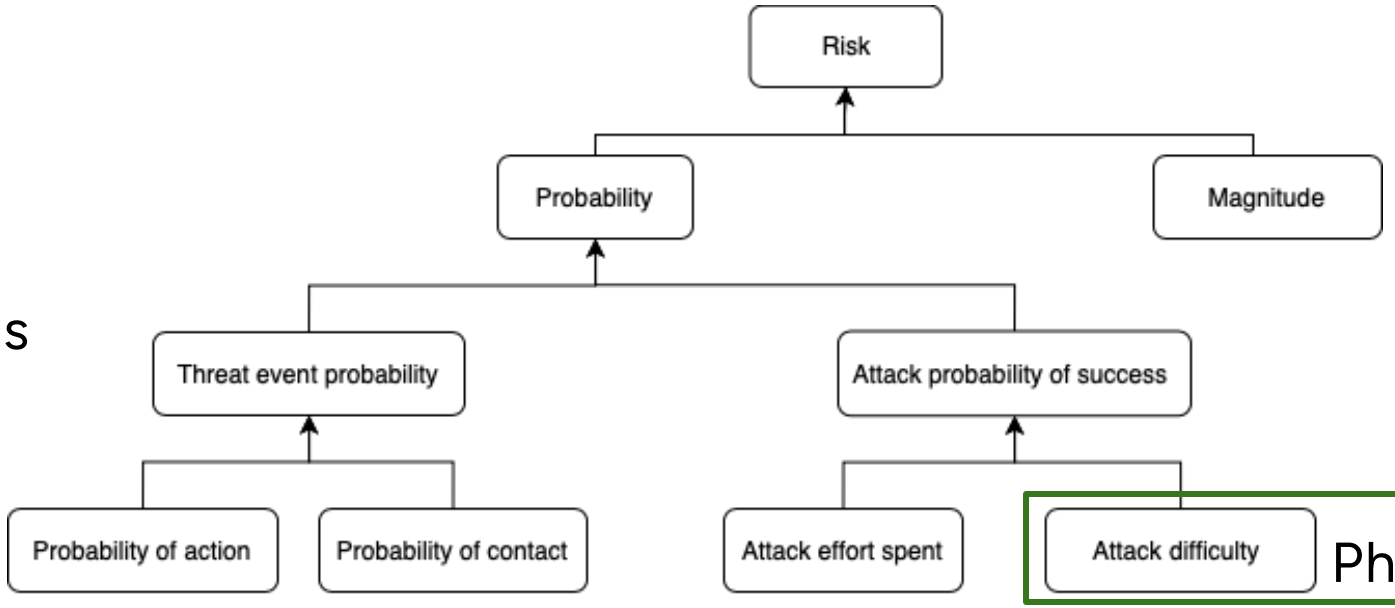
Phase 4 - Purpose

“Attacks are the realization of some specific threat impacting the confidentiality, integrity, accountability, or availability of a computational resource.” (NISP 800-28v2)

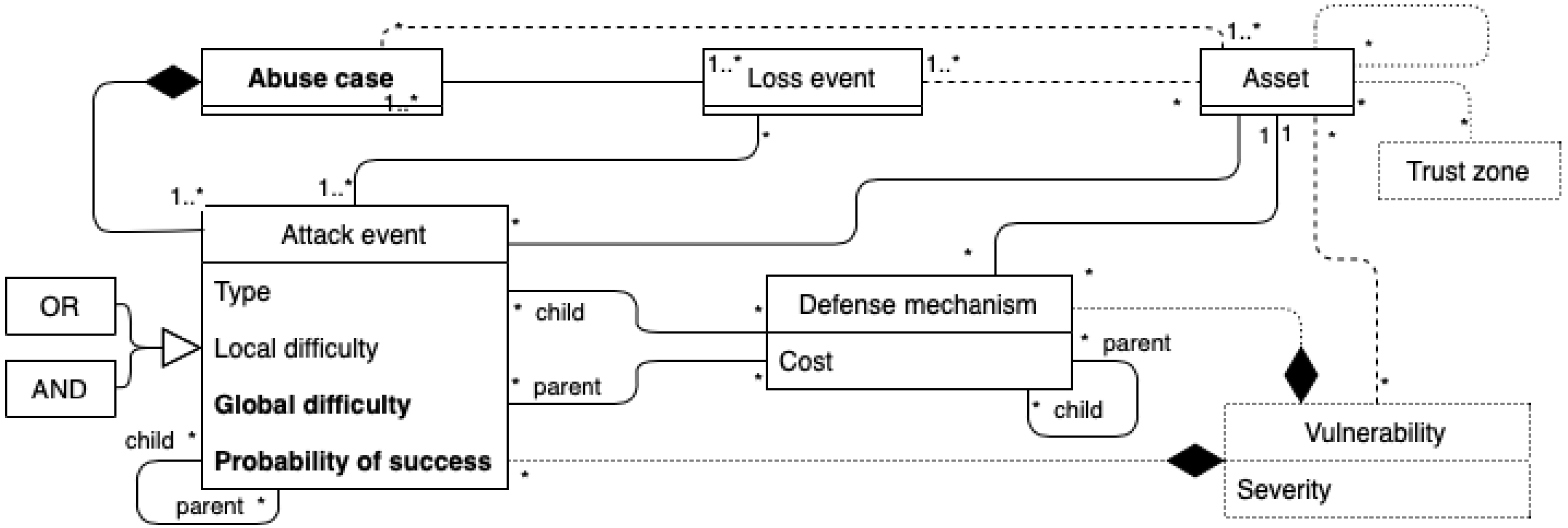
Assessing the inherent resilience of the system. Given that someone attempts to attack our system, how difficult will it be? How much protection mechanisms do we have in place?

Phase steps:

1. List vulnerabilities
2. Develop attack trees/graphs



Language overview



Step 1:
List vulnerabilities

Some background:

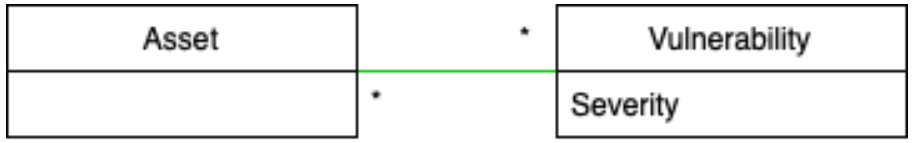
- There are well established meaning of “weakness” and “vulnerability” through the MITRE Corp.’s Common Weakness Enumeration (CWE) and Common Vulnerability and Exposures (CVE).
- CVEs are vulnerability instances found in specific software. Known vulnerabilities are numbered, e.g. CVE-2017-0144 and are disclosed in public databases (e.g. NVD).
- With a real system to examine at hand we can search for vulnerabilities (e.g. through scanning or pentesting). In the case of system design we cannot. (And in this course they have to be imagined..)
- CWEs are types/categories of vulnerabilities. (This is generally more suited for the course.)
- In the course we will not differentiate between vulnerabilities and weaknesses from a modeling and analysis point of view. So, we might also use the term vulnerability meaning weakness in the MITRE sense. (From a threat modeling point of view, the main difference between the two concepts lies in how certain you are about the vulnerability existence and its properties.)

https://cve.mitre.org + https://cwe.mitre.org/

Step 1: List vulnerabilities - Types of vulnerabilities

- Product implementation bugs → CVE/CWE
- Configuration vulnerabilities
  - Low entropy password
  - Single factor authentication
  - Weak encryption
  - …
- Structural / architectural vulnerabilities
  - It is good security practice to structure communication networks in various segments to prevent attacks from spreading.
  - Lack of logging
  - Zero trust models is good practice. (Always verify/authenticate requests from clients.)
  - …

Step 1:
List vulnerabilities



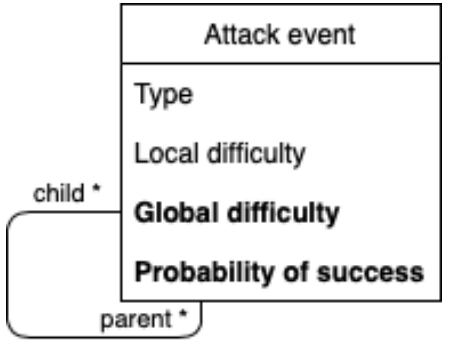
- Identify vulnerabilities and argue for why they are likely/relevant.
  - Software can be old, can have known vulnerabilities, some parts of the system might be less “known” where there are more uncertainty around vulnerability existence, etc.
  - (In the course there is also an element of “planting” vulnerabilities just to practice the threat modeling and suggesting improvements in the end…)
- Map vulnerabilities to assets.
  - This is just an intermediary step. In the next step vulnerabilities will be converted into attack/defense graphs.
  - (BTW, this is essentially “threats” in the OWASP Threat dragon tool.)
- Estimate their severity.
  - CVEs are by many vulnerability databases assessed with respect to their severity in a standardized way; the Common Vulnerability Scoring System (CVSS) ranging from 0-10. (Also there is a CWSS but that seem to lack momentum.)
  - In the course you make your own severity assessment. E.g. on a scale 0-10.

https://www.first.org/cvss/

Step 2:
Devise attack graphs

- Abuse cases indicates the goals and starting point of the attackers, but how do they move in-between? This is illustrated by the attack graph. Attack vectors are enabled by vulnerabilities in the system. So by exploiting vulnerabilities scattered over the system the attacker is closing in on the goal, step by step. Describe these vectors from the attacker entry points to the goals. I.e. vulnerabilities/weaknesses are here transformed into attack events/steps and (missing) defense mechanisms.
- Start working with the vulnerabilities/weaknesses that were identified with the highest severity.
- Assign “local difficulty” per attack event/step. I.e. assign cost for (successfully) conducting the attack steps. This indicates the resilience of the system.

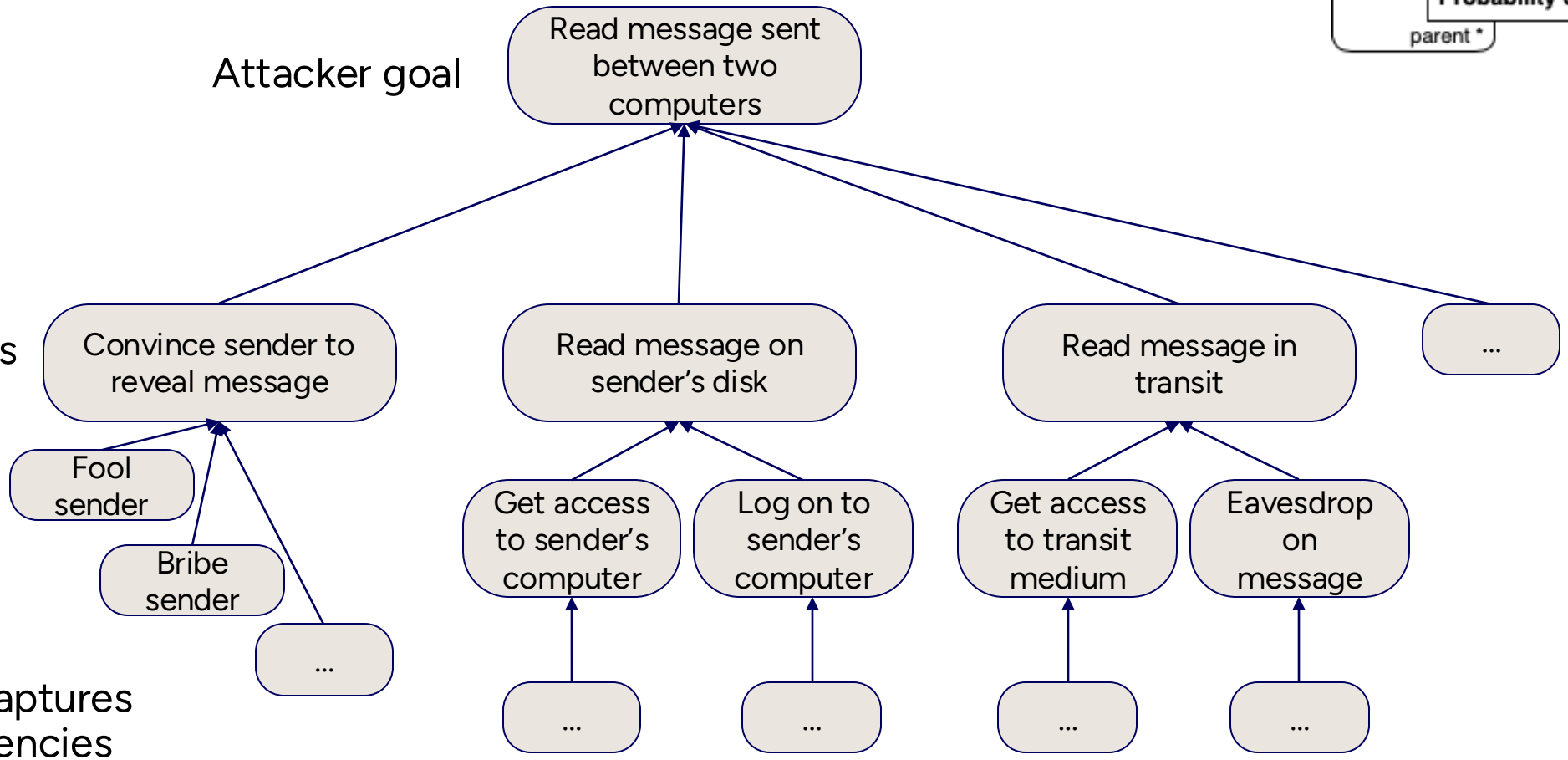
Attack trees



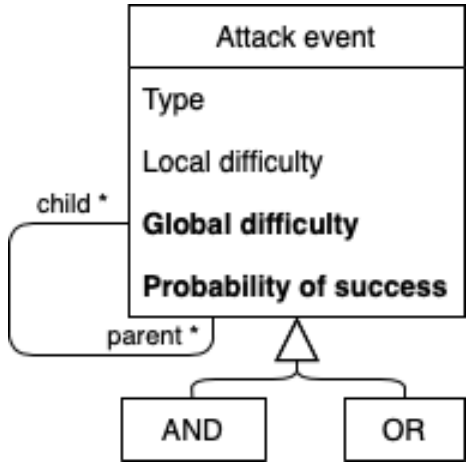
Attacker goal

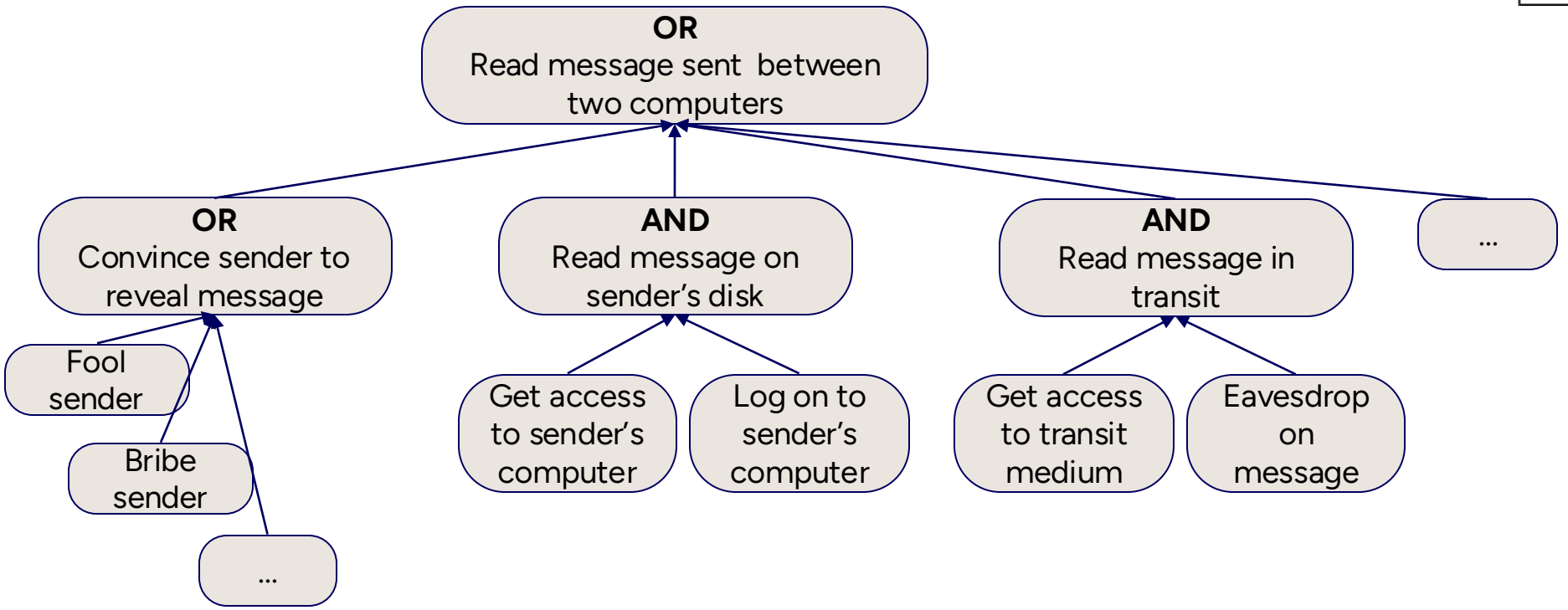
Attacker means

Associations captures causal dependencies

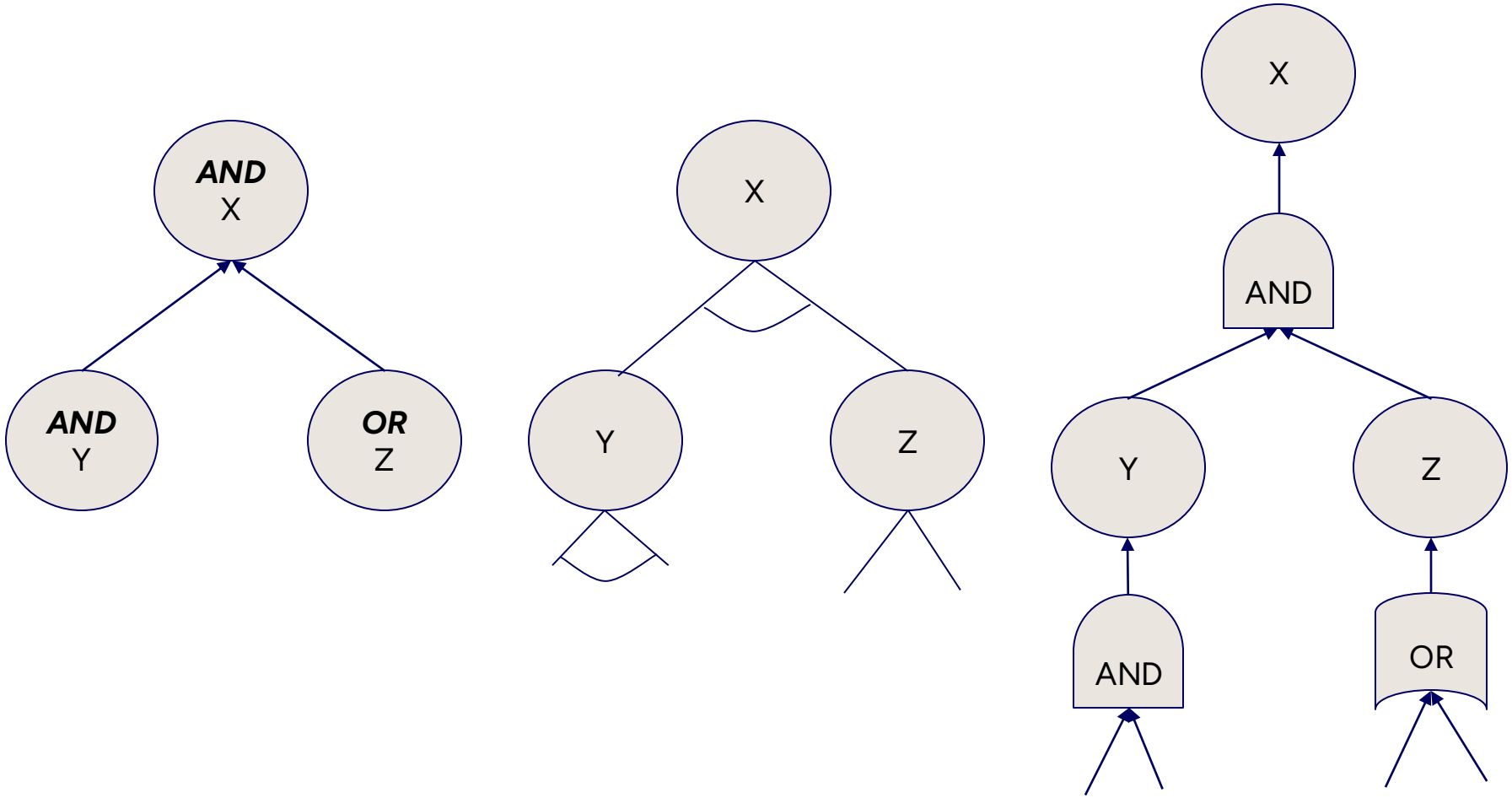


Attack trees - AND / OR gates

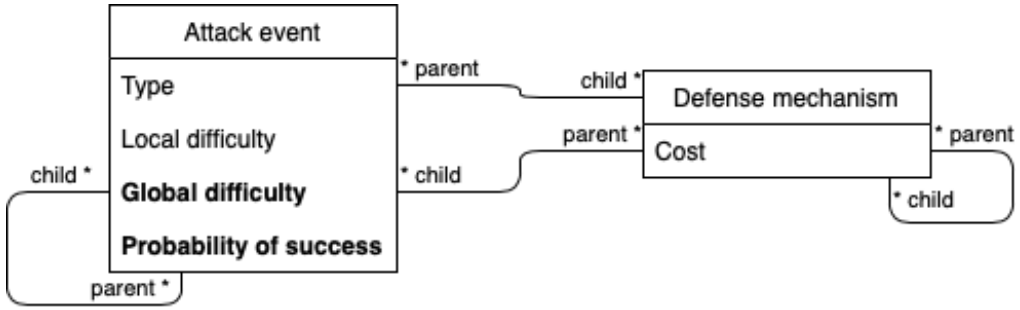




Alternative attack tree syntax



Attack/defense trees



OR
Read message sent between two computers

AND
Read message on sender's disk

AND
Read message in transit

...

OR
Get access to sender's computer

OR
Log on to sender's computer

OR
Get access to transit medium

OR
Eavesdrop on message

Authentication mechanism

Message encryption

OR
Obtain credentials

OR
Break encryption

...

OR
Crack credentials

OR
Fool user to reveal credentials

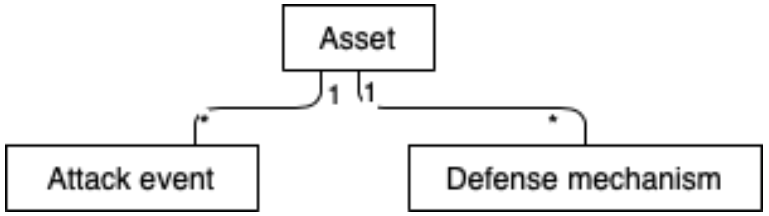
OR
Obtain decryption key

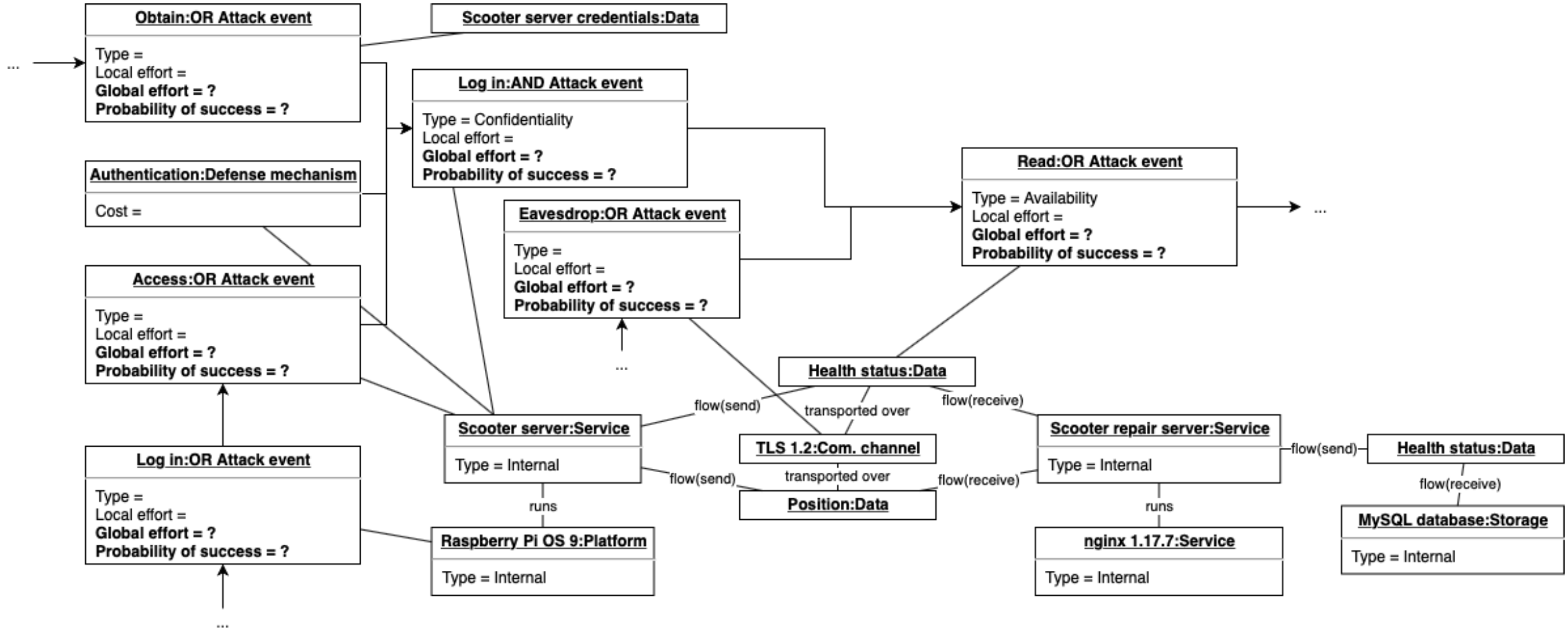
OR
Break crypto system

High Entropy Policy

No credential reuse

Attacks and defenses relate to assets





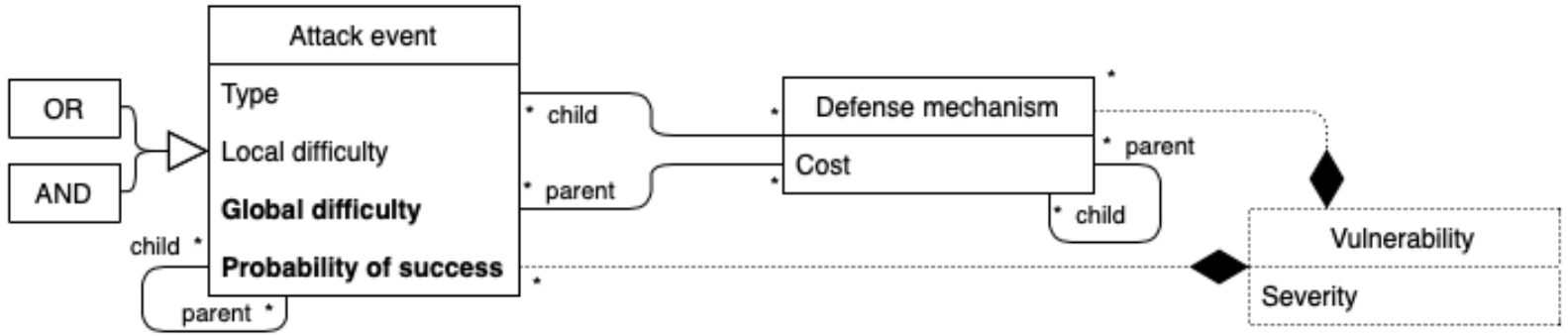
Transforming vulnerabilities into attack/defense trees

Normally vulnerabilities are the inverse of an attack. We can generically think “exploit vulnerability” as an attack event. This is clearly the case for software vulnerabilities; there is exploit code tailored for a specific software vulnerability. But the same logic applies for many other things; password reuse (vulnerability) can be exploited by a rainbow table attack, a short cryptographic key (vulnerability) can be exploited by a brute force attack, a security unaware person (vulnerability) can be exploited with a phishing email attack, and so on.

Sometimes it is however easier to think of vulnerabilities as missing defenses; a two factor authentication solution (defense) would have stopped the account login (attack).

Either way, vulnerabilities are thus conceptually strange creatures in our context. They are only placeholder or abstract concepts that are waiting for further refinement in the threat modeling process.

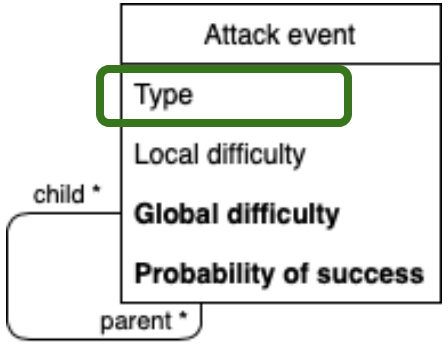
*Vulnerabilities are not used or needed **per se** in the security risk assessment. They are (very important) **means** to develop the attack/defense tree.*



Type of breach

•It is common to speak of three types of security breaches, CIA:

- Confidentiality (attacker should not be able to read)
- Integrity (attacker should not be able to write/manipulate)
- Availability (the users should be able to access as intended)



•Sometimes we also add:

- Non-repudiation (attacker should not be able to deny breach)

•There are also other attack/breach taxonomies, e.g. STRIDE

- **S**poofing of user identity
- **T**ampering (integrity)
- **R**epudiation
- **I**nformation disclosure (confidentiality)
- **D**enial of service (availability)
- **E**levation of privilege

•For attack/threat modelling it is not strictly needed to use these types, but often it is convenient. Both to be able to talk about and logically group attacks on a high level, but also as method to identify possible loss events (as described in Phase 1).

Attack difficulty - Putting values to attack events

How difficult is it for the attacker succeed with the attack?

We need to quantify attack difficulty. But how? E.g.:

- Possible/impossible
- special skill/no special skill
- attacker cost
- attacker time
- probability of success

Unfortunately, there exist no single standard for this quantification. In Yacraf we suggest to use ***attacker cost*** as the metric.

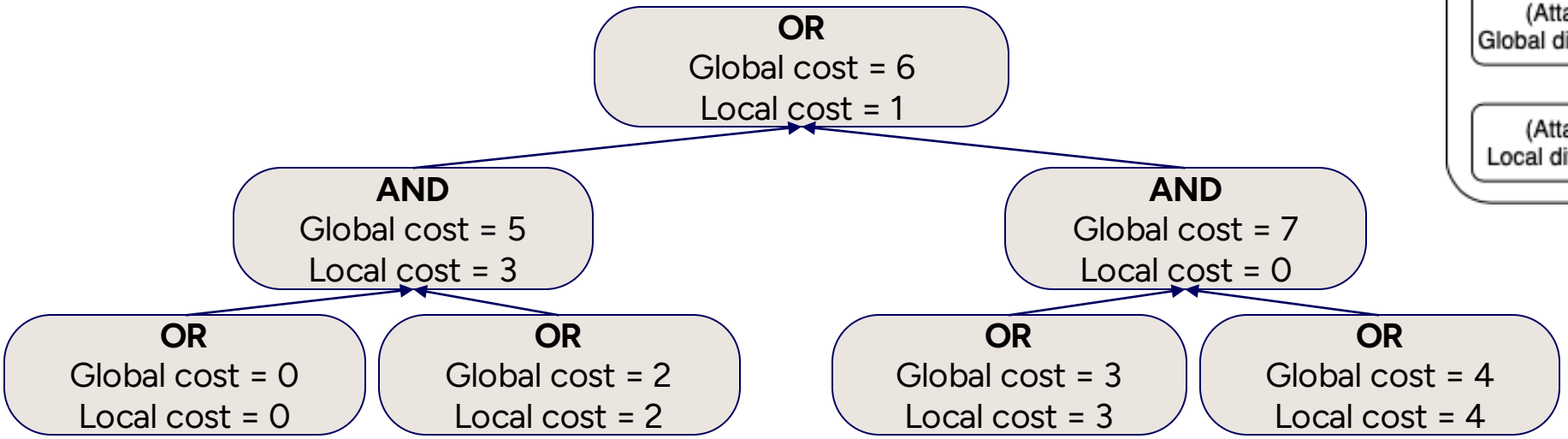
Aggregating attack difficulty

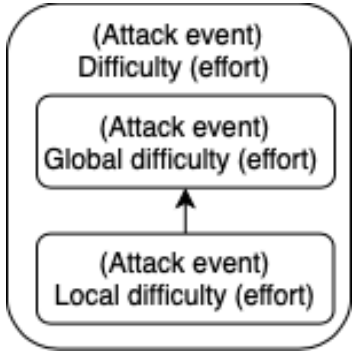
We use two metrics:

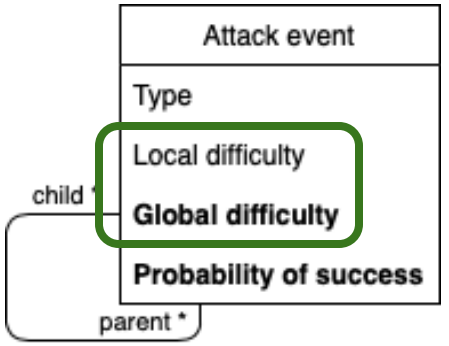
- Local values per event (how difficult is it to succeed with this event)
- Global (aggregated) values per event (how difficult is it to reach to this point)

The aggregated value per attack step is simply calculated by:

- for OR nodes: adding the local value to the minimum global value of the parents
- for AND nodes: adding the local value to the sum the global values of all the parents

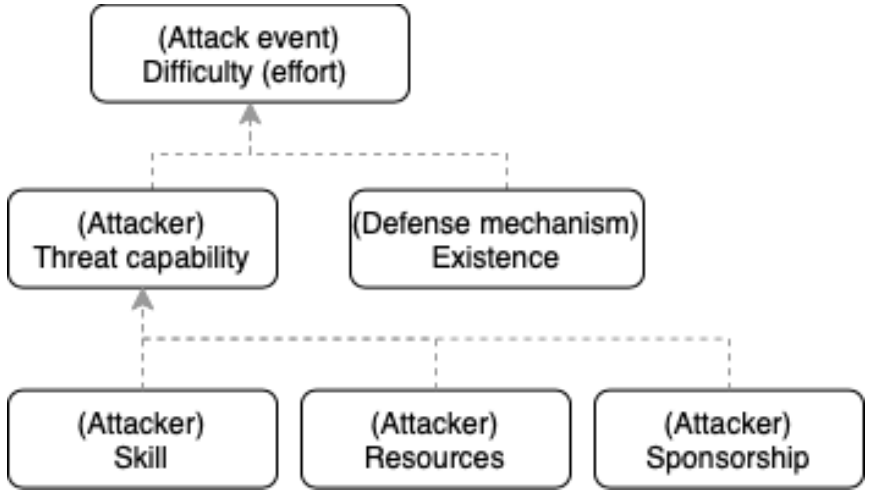






Attack difficulty dependencies

Attacker cost then “represents” attacker capabilities (skill, resources, and sponsorship) as well as defense mechanisms, vulnerabilities, and system properties impacting the inherent system resilience



The pure existence of vulnerabilities and defense mechanisms we consider to be completely independent of the attacker. E.g. the existence of a software vulnerability (a CVE), a password with poor entropy, and the use of anti-virus software are part of the system design and configuration, something that only we as defenders can change (at least in principle). With the absence of vulnerabilities and existence of defenses attack difficulty will increase.

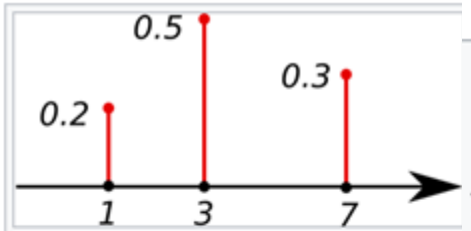
But with good tooling and competence, attacking generally becomes faster/cheaper. A high entropy but reused and previously leaked and present in some dump is easy to bypass, only if the attacker is actually using this dump. With much resources we perhaps assume the attacker has access to the dump but not otherwise. Similarly, we might want to assume that some attackers are not capable of developing an exploit to, or finding, a zeroday vulnerability, while others are.

→ **Attack trees need to be developed individually for different attackers (abuse cases). Partly they will often be similar for various attackers, though.**

Uncertainty and probabilistic modeling

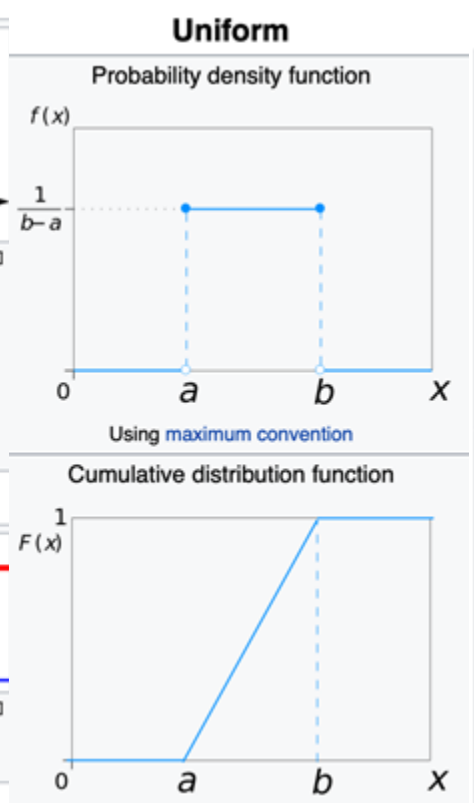
Attack step difficulty variables (cost) are often probabilistic either because we lack knowledge or because it varies due to non modeled parameters. E.g. cost of developing an exploit or social engineer a user, or our knowledge about if vulnerabilities exist in the system or not.

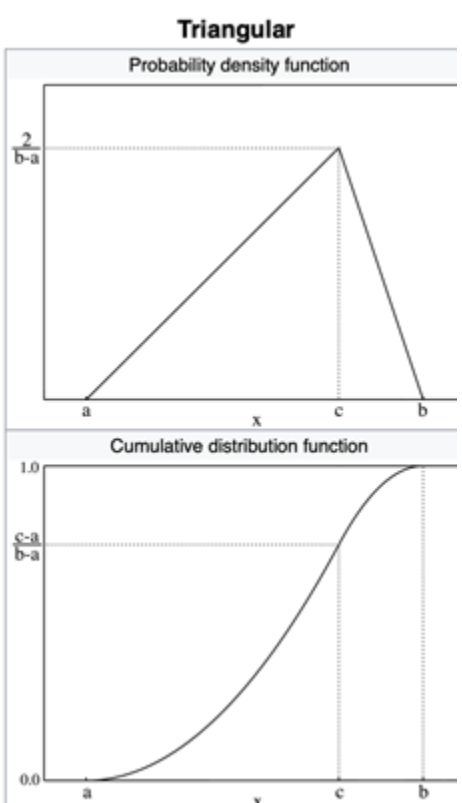
This probability can be captured with probability density functions and cumulative distribution functions.

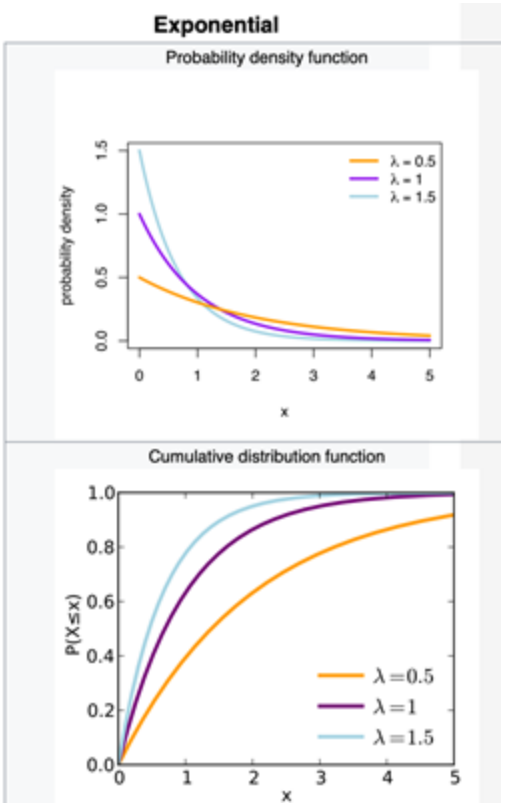


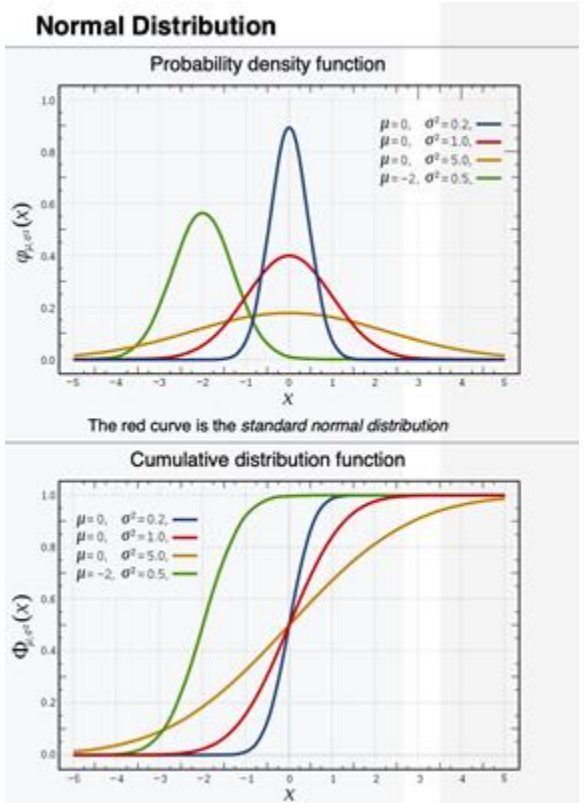
The probability mass function of a discrete probability distribution. The probabilities of the singletons {1}, {3}, and {7} are respectively 0.2, 0.5, 0.3. A set not containing any of these points has probability zero.

The cdf of a discrete probability distribution, ...

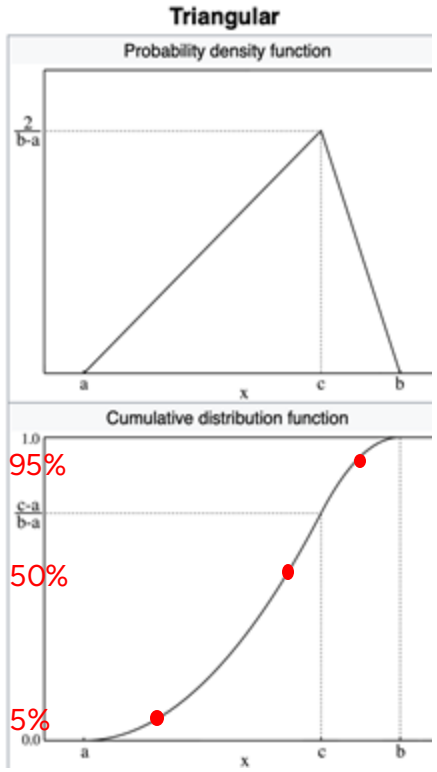








Deriving global attack difficulty



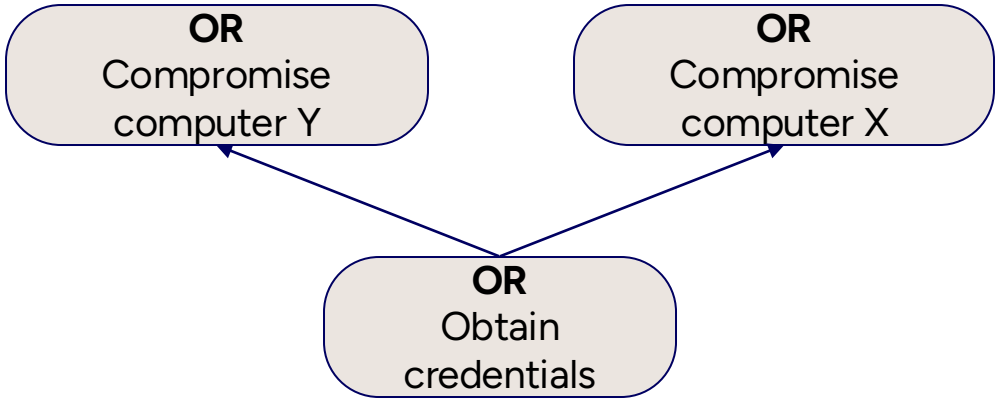
- We could use Monte Carlo simulations and sample the probability distributions for all attack steps and add the results. This would provide a full cumulative probability distribution for the global resilience values.
- This is quite ambitious for this course (since we don't have a ready to use simulation tool). Instead we can use single deterministic values as a base.
- More ambitious analysis can e.g. use best, expected and worst case calculations. Even more ambitious could be to let the best case correspond to 5% probability of attack success, expected 50%, and worst case 95%. Intrinsic resilience would then have a three (five point) distribution function.
- *Probabilistic analysis is a criteria for grading.*

Examples of more elaborate probabilistic representation of cost

(possibly too overambitious to use here though…)

- Buying e.g. a Pineapple rouge wifi hotspot would be a fixed cost with a 100% probability of success above that cost and 0% probability of success below the cost. Building a rogue hotspot could perhaps have cost distribution that is uniform between two costs a and b and maybe a remaining 10% chance of being infinite. I.e. if the attacker spend more cost (time) than b we believe that the project will never be successful. (Then there are additional probability distributions for getting targeted users to use the hotspot..)
- There is a 1% chance that the user left his/her credentials on a post-it note under the keyboard, and then it takes 0 cost to obtain the credentials, but there is a 99% chance the the credentials cannot be retrieved this way so then the cost will be infinite (and some other attack needs to be conducted).

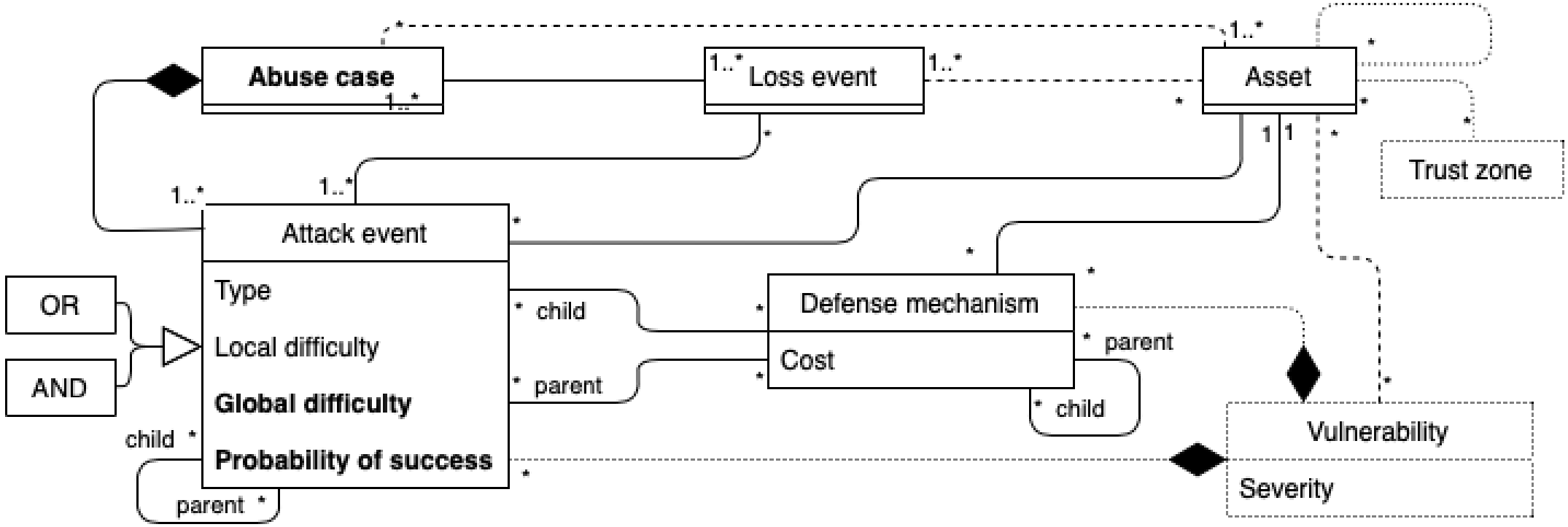
Sometimes attack steps enables several other steps
→ Attack/defense *graphs*



It is straight forward to make a tree from a graph; parent nodes are just duplicated. (For large graphs this can get messy; there might be many nodes and it might be difficult to keep track of which are copies of each other.)

For graphs (or trees with duplicates) we ideally want to avoid “double counting” attack difficulty cost when aggregating attack vectors, but this can be difficult with large graphs. For automated calculations approximative algorithms, such as shortest path, have to be used.

Language overview



How to develop attack trees?

From Schneier, Attack trees, 1999 (the (almost) first article on the topic):

“How do you create an attack tree like this? First, you identify the possible attack goals. Each goal forms a separate tree, although they might share subtrees and nodes. Then, try to think of all attacks against each goal. Add them to the tree. Repeat this process down the tree until you are done. Give the tree to someone else and have him think about the process and add any nodes he thinks of. Repeat as necessary, possibly over the course of several months. Of course there's always the chance that you forgot about an attack, but you'll get better with time. Like any security analysis, creating attack trees requires a certain mindset and takes practice.

Once you have the attack tree and have researched all the node values (these values will change over time, both as attacks become easier and as you get more exact information on the values), you can use the attack tree to make security decisions. You can look at the values of the root node to see if the system's goal is vulnerable to attack. You can determine if the system is vulnerable to a particular kind of attack; password guessing, for instance. You can use the attack tree to list the security assumptions of a system; for example, the security of PGP might assume that no one could successfully bribe the programmers. You can determine the impact of a system modification or a new vulnerability discovery: Recalculate the nodes based on the new information and see how the goal node is affected. And you can compare and rank attacks -- which is cheaper, which is more likely to succeed, and the like.”

“Repeat this process down the tree until you are done” ...Well, when is that?

- When we leave attack nodes as leaves (without parents) we assume that they are the first hurdle for the attacker. So, this is the starting point for the attacker, before that the attacker has no costs (or costs that we do not want to include in the analysis). Essentially this is determined by the Abuse case “Accessibility to attack surface” and perhaps “Window of opportunity”.
- The most common starting point is the internet and the street. But for insider threat actors for instance the starting point is normally a corporate network and the company premises.

Phase 4
Attack and Resilience Analysis

Complementary update

ATT&CK

How to develop attack trees?

From Schneier, Attack trees, 1999 (the (almost) first article on the topic):

“How do you create an attack tree like this? First, you identify the possible attack goals. Each goal forms a separate tree, although they might share subtrees and nodes.

Then, try to think of all attacks against each goal.

Add them to the tree. Repeat this process down the tree until you are done. Give the tree to someone else and have him think about the process and add any nodes he thinks of. Repeat as necessary, possibly over the course of several months. Of course there's always the chance that you forgot about an attack, but you'll get better with time. Like any security analysis, creating attack trees requires a certain mindset and takes practice.

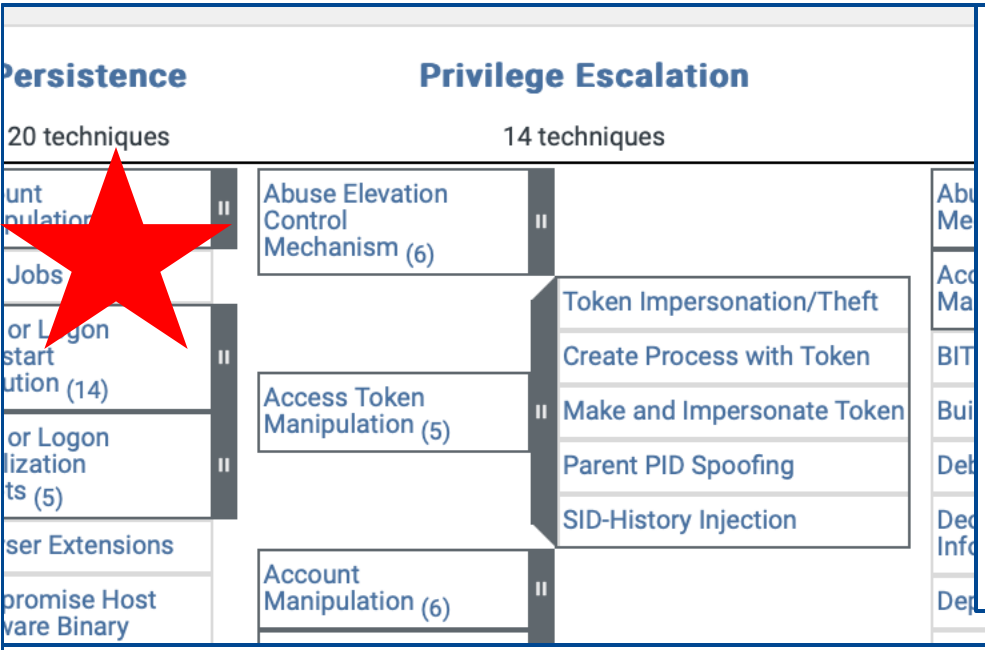
Once you have the attack tree, and have researched all the node values (these values will change over time, both as attacks become easier and as you get more exact information on the values), you can use the attack tree to make security decisions. You can look at the values of the root node to see if the system's goal is vulnerable to attack. You can determine if the system is vulnerable to a particular kind of attack; password guessing, for instance. You can use the attack tree to list the security assumptions of a system; for example, the security of PGP might assume that no one could successfully bribe the programmers. You can determine the impact of a system modification or a new vulnerability discovery: Recalculate the nodes based on the new information and see how the goal node is affected. And you can compare and rank attacks -- which is cheaper, which is more likely to succeed, and the like.”

MITRE ATT&CK

- "[…] knowledge base of adversary tactics and techniques based on real-world observations."
- Organizes attack actions in Tactics that contains Techniques. Gathered in Matrices. "Enterprise Matrix" being the big common one. The Enterprise Matrix can also be sorted on platforms.

https://attack.mitre.org + https://attack.mitre.org/matrices/enterprise/

ATT&CK Technique example



Access Token Manipulation: Create Process with Token

Other sub-techniques of Access Token Manipulation (5)

Adversaries may create a new process with an existing token to escalate privileges and bypass access controls. Processes can be created with the token and resulting security context of another user using features such as `CreateProcessWithTokenW` and `runas`.[1]

Creating processes with a token not associated with the current user may require the credentials of the target user, specific privileges to impersonate that user, or access to the token to be used. For example, the token could be duplicated via Token Impersonation/Theft or created via Make and Impersonate Token before being used to create a process.

While this tech
a token is dupl

ID: T1134.002
Sub-technique of: T1134
Tactics: Defense Evasion, Privilege Escalation
Platforms: Windows
Defense Bypassed: File system access controls, System access controls, Windows User Account Control

Procedure Examples

ID	Name	Description
S0456	Aria-body	Aria-body has the ability to execute a process using `runas`.[2]
		call WTSQueryUserToken and CreateProcessAsUser to start a new process with local system privileges.[3]
		abs a user token using WTSQueryUserToken and then creates a process by impersonating a logged-on user.[4]
		se `Invoke-RunAs` to make tokens.[5]

Mitigations

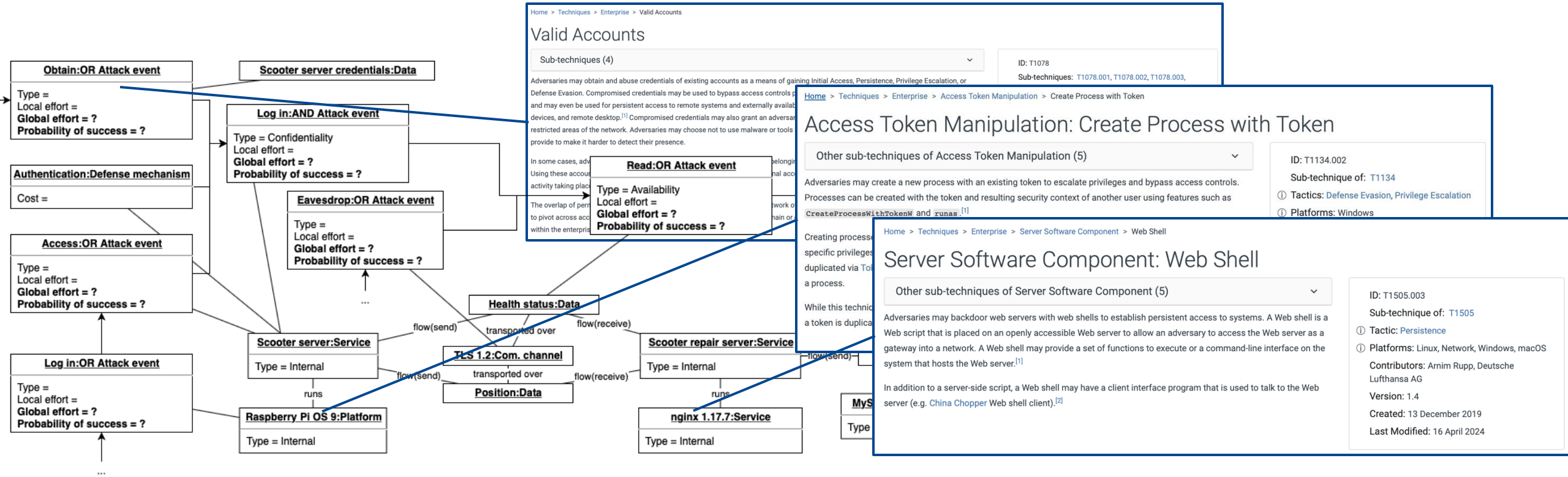
ID	Mitigation	Description
M1026	Privileged Account Management	Limit permissions so that users and user groups cannot create tokens. This setting should be defined for the local system account only. GPO: Computer Configuration > [Policies] > Windows Settings > Security Settings > L... [16] Also define who can create a process level token to only the local and netw... Windows Settings > Security Settings > Local Policies > User Rights Assignmer... Administrators should log in as a standard user but run their tools with adminis... command `runas`.[18]
M1018	User Account Management	An adversary must already have administrator level access on the local system... accounts to the least privileges they require.

Detection

ID	Data Source	Data Component	Detects
DS0017	Command	Command Execution	Monitor executed commands and arguments to detect token manipulation by auditing command-line activity. Specifically, analysts should look for use of the runas command or similar artifacts. Detailed command-line logging is not enabled by default in Windows.[19]
DS0009	Process	OS API Execution	Monitor for API calls associated with detecting token manipulation only through careful analysis of user activity, examination of running processes, and correlation with other endpoint and network behavior. Analysts can also monitor for use of Windows APIs such as `CreateProcessWithTokenW` and correlate activity with other suspicious behavior to reduce false positives that may be due to normal benign use by users and administrators.

Relate attack techniques/events to YOUR system

- Not all attack events in your attack graphs needs to be 1:1 mapped to ATT&CK – it is a good source of inspiration and confirmation.
  - Similar types of attacks
  - Mapping abstraction-level may vary – tactics/technique/sub-technique + multiple attack events per technique
  - Not all of your attack events may have a clear mapping



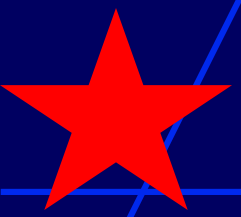
Assets and vulnerabilities are support elements…

- The goal of the attack and resilience analysis (Phase 4) is to produce the possible/probable attack vectors through our system.
- The top node(s) in the attack graph(s) represent the difficulty dimension in the overall risk analysis (more on this in phase 5). Thus, we strictly do not need assets nor vulnerabilities to make the calculations.

BUT

It is the system properties and architecture (including trust zones) and its vulnerabilities that enables/disables the attacks, so they are key to understand and validate the attack vectors. It would be very difficult/impossible to imagine attack difficulty without a detailed understanding of the system.

→ ***Be careful and diligent in mapping the attack events to the system assets.***



Phase 4
Attack and Resilience Analysis

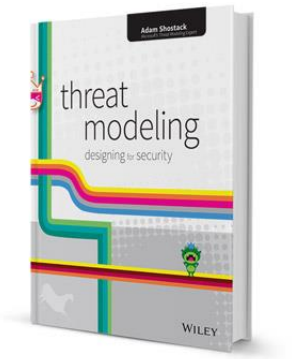
Complementary update

STRIDE

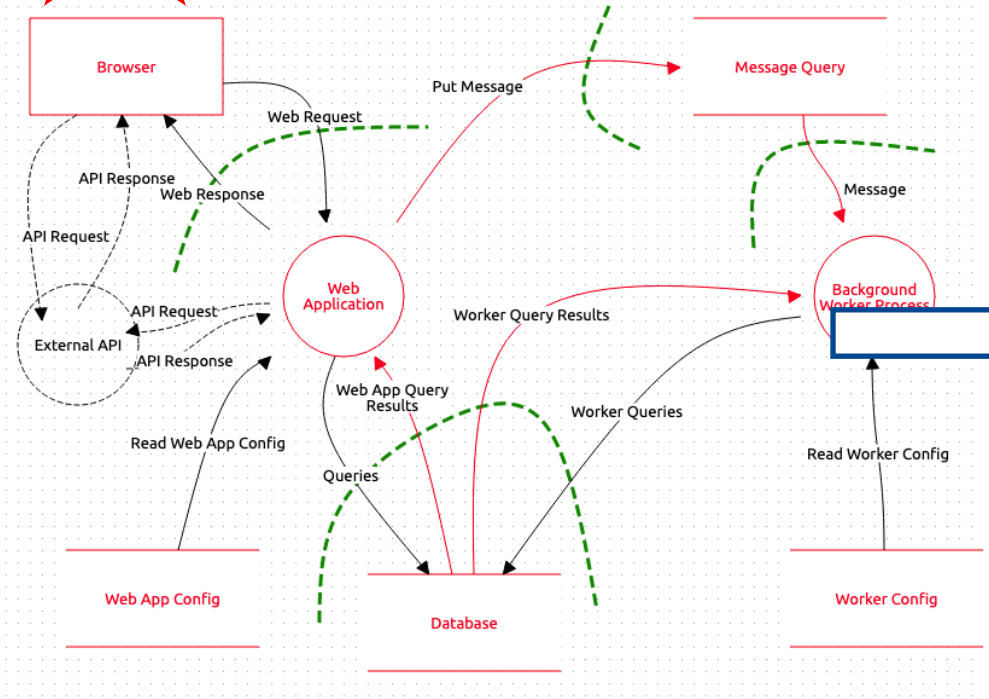
STRIDE framework

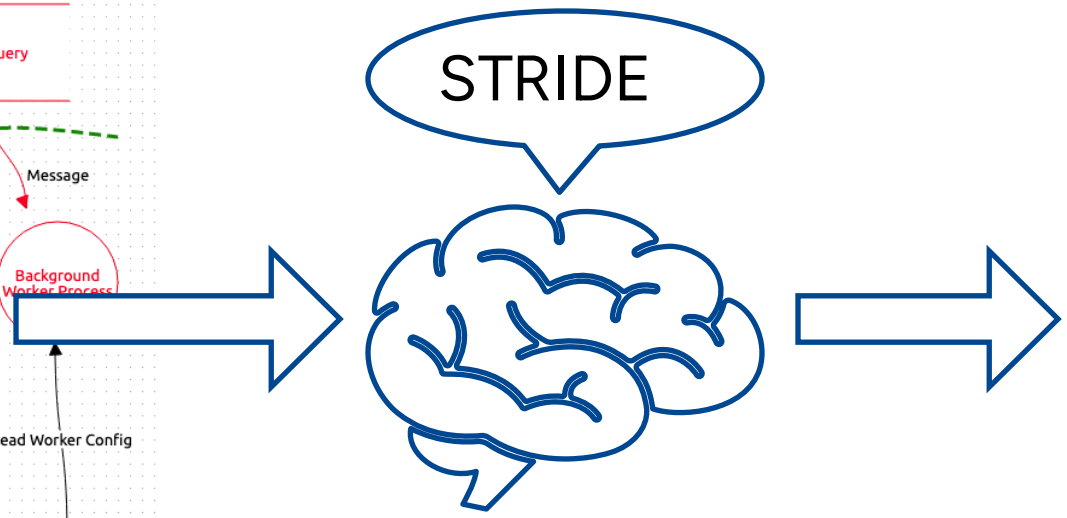
	Threat	Property Violated	Threat Definition
S	Spoofing identify	Authentication	Pretending to be something or someone other than yourself
T	Tampering with data	Integrity	Modifying something on disk, network, memory, or elsewhere
R	Repudiation	Non-repudiation	Claiming that you didn't do something or were not responsible; can be honest or false
I	Information disclosure	Confidentiality	Providing information to someone not authorized to access it
D	Denial of service	Availability	Exhausting resources needed to provide service
E	Elevation of privilege	Authorization	Allowing someone to do something they are not authorized to do

- Developed at Microsoft
- Popularized/refined by Adam Shostack



STRIDE analysis





- Vulnerability 1
- Vulnerability 2
- Vulnerability 3
- ...

- From a development perspective, identifying vulnerabilities are often the end goal.
- Our method has a larger scope, however.

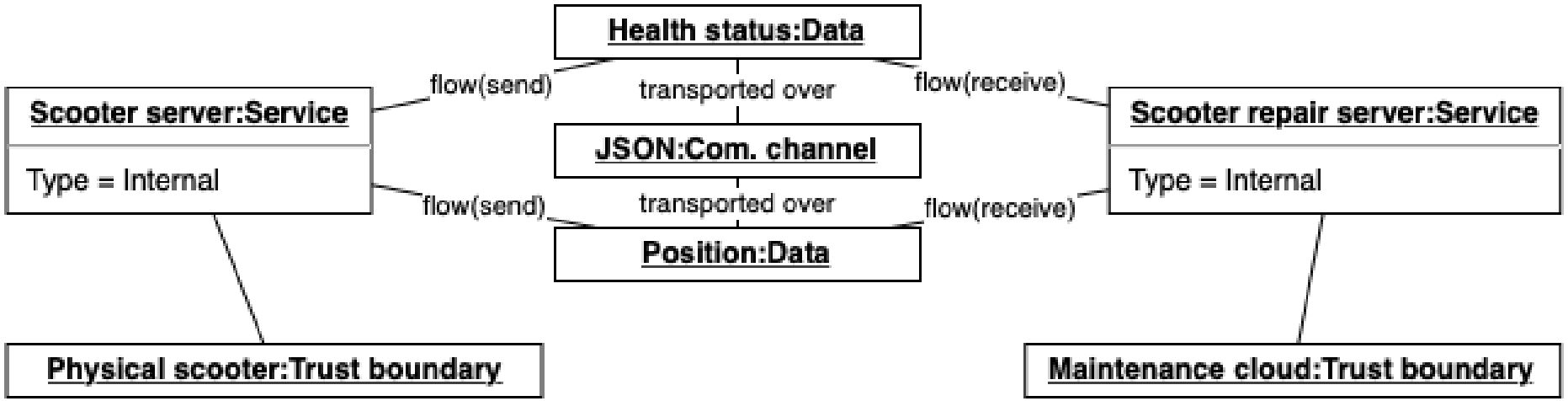
STRIDE analysis

- STRIDE-per-element
  - Retrive elements from DFD
  - Apply threat-per-element Table
- STRIDE-per-interaction
  - Retrieve data flows “source-to-sink” in DFD
  - Apply threats per boundary crossing

	S	T	R	I	D	E
External Entity	✔		✔			
Process	✔	✔	✔	✔	✔	✔
Data Store		✔	?	✔	✔	
Data Flow		✔		✔	✔	

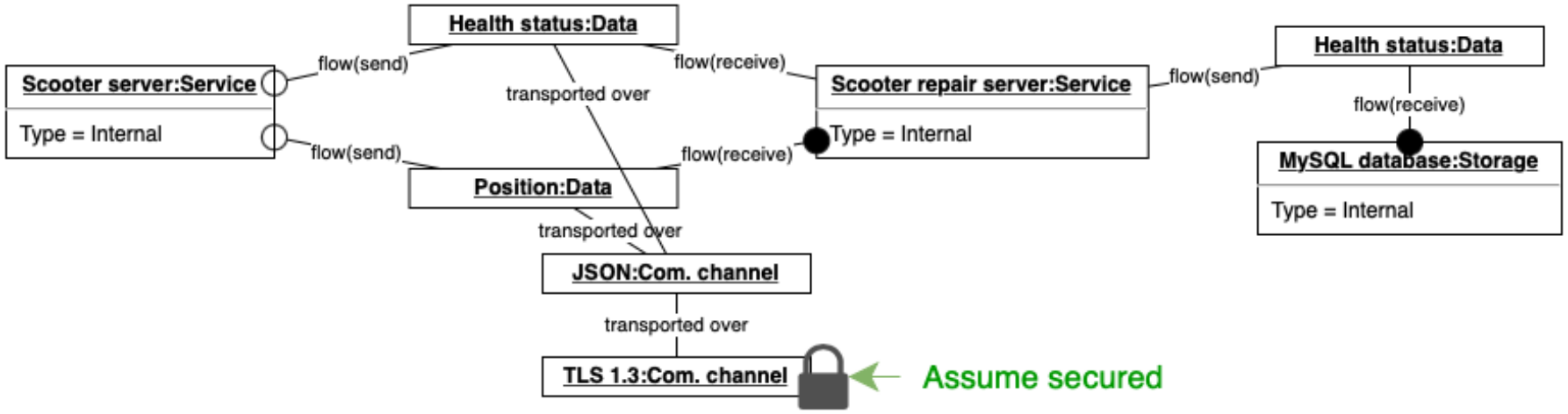
A STRIDE-per-interaction example

S	T	R	I	D	E
X	X	X	X	(X)	



Security assumptions

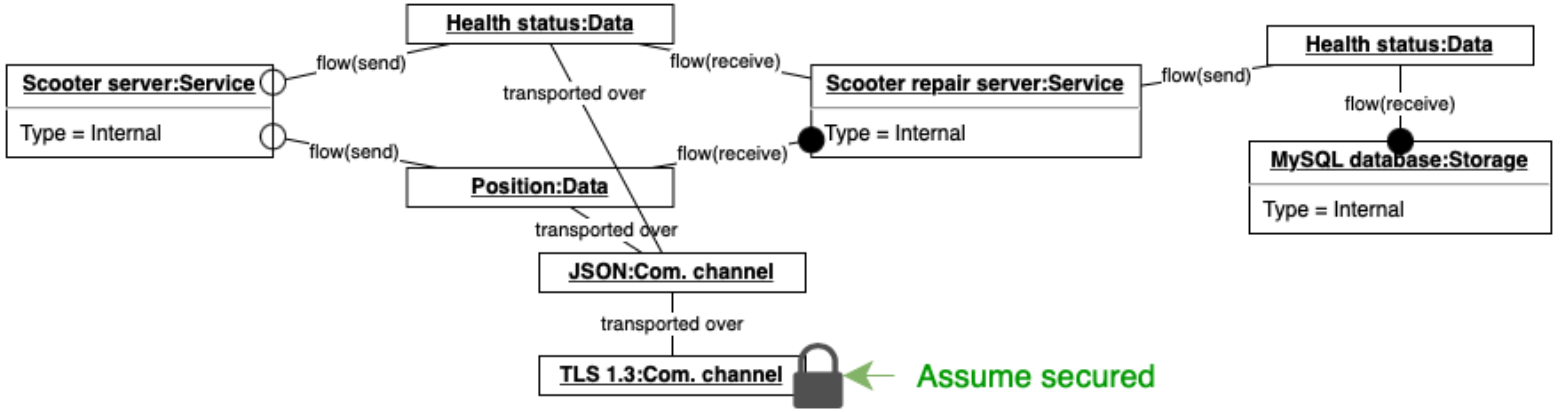
In order to delimit the analysis space we can (/must) make assumptions about elements we believe are secure by definition. It is generally preferred to be explicit with such assumptions. (E.g. by marking objects with a pad lock or similar.)

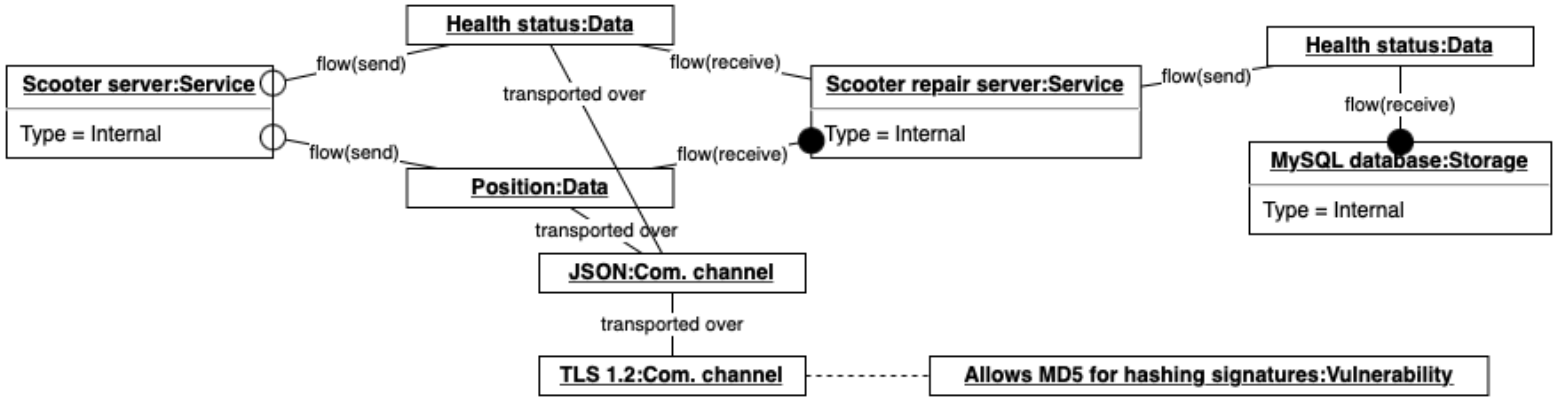


Security assumptions

Make reasonable assumptions. There are no clear rights or wrongs, instead it depends on the case. E.g.

- Generally considered difficult to exploit TLS 1.2
- There are vulnerable implementations of TLS1.3



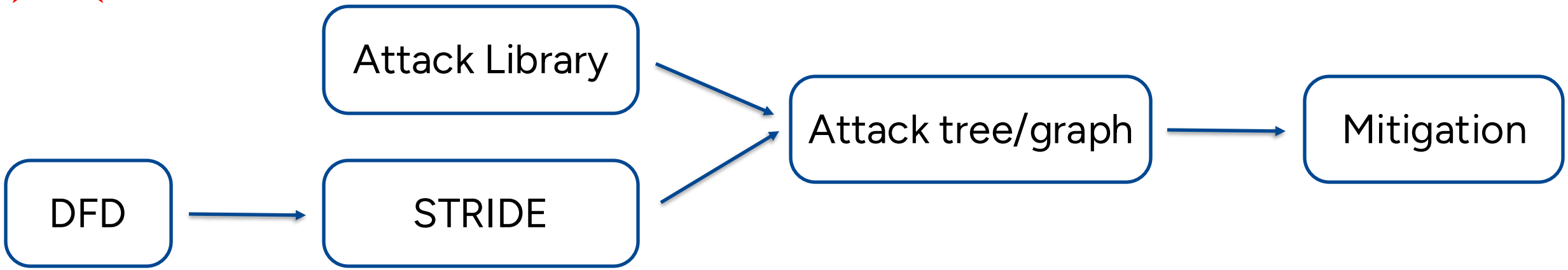


Security properties

Sometimes we might want to be upfront with what limitations we have in terms of design space. We can then attach an explicit statement about system/security properties.

- E.g. some dataflow cannot be encrypted (due to performance requirements perhaps), some machine can only be patched with low frequency (perhaps the extensive functional testing needed afterwards)

Advanced threat modeling with STRIDE

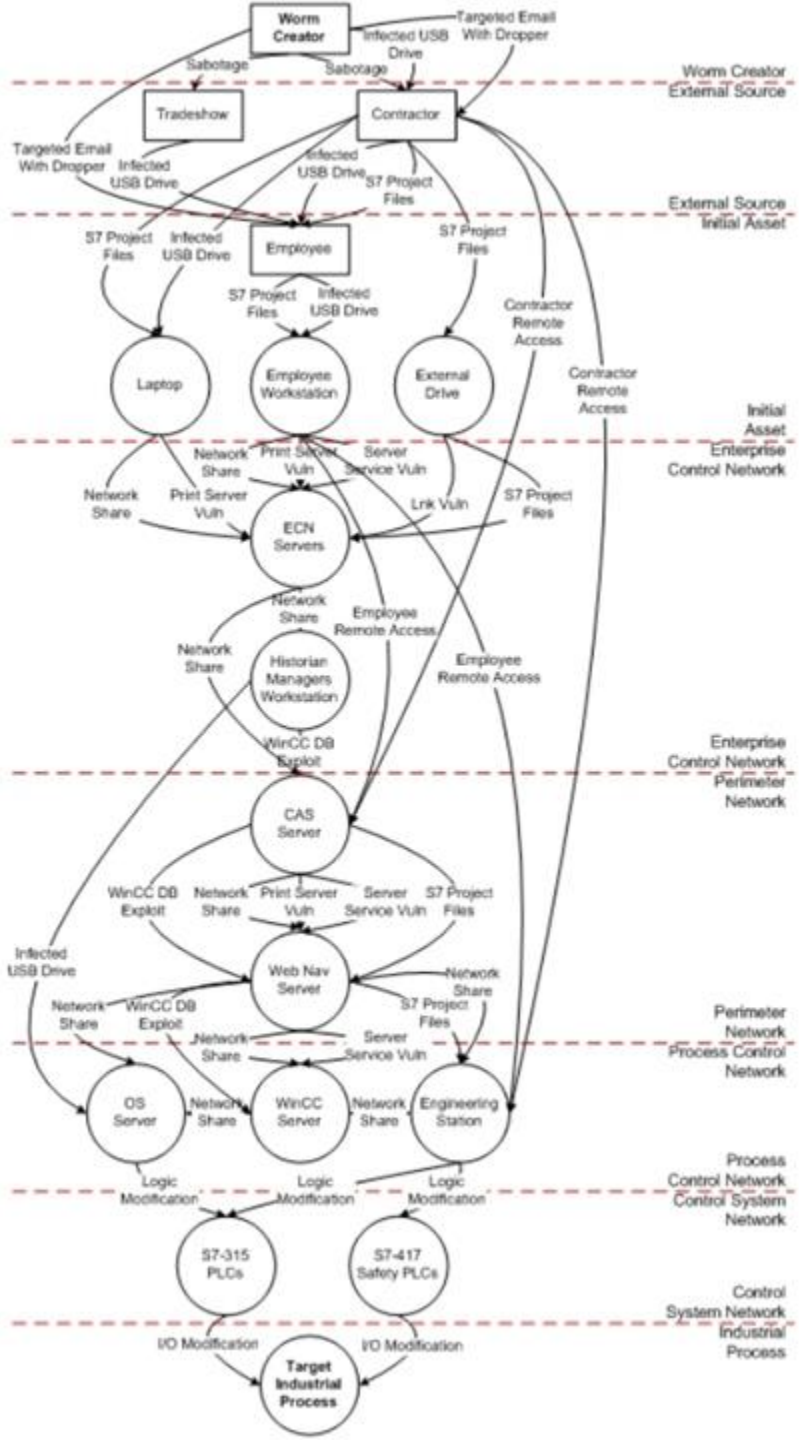


This is similar to our approach.

So, STRIDE can be used as supporting method to identify vulnerabilities, flaws and exploit chains in our more encompassing approach.

An interesting attack graph

- The Natanz hack / Stuxnet
- (In the wrong format according to the course language though…)
- This and other examples can be used as inspiration for the course assignment.



Phase 5
Risk Assessment and Recommendations

Phase 5 - Risk assessment and recommendations

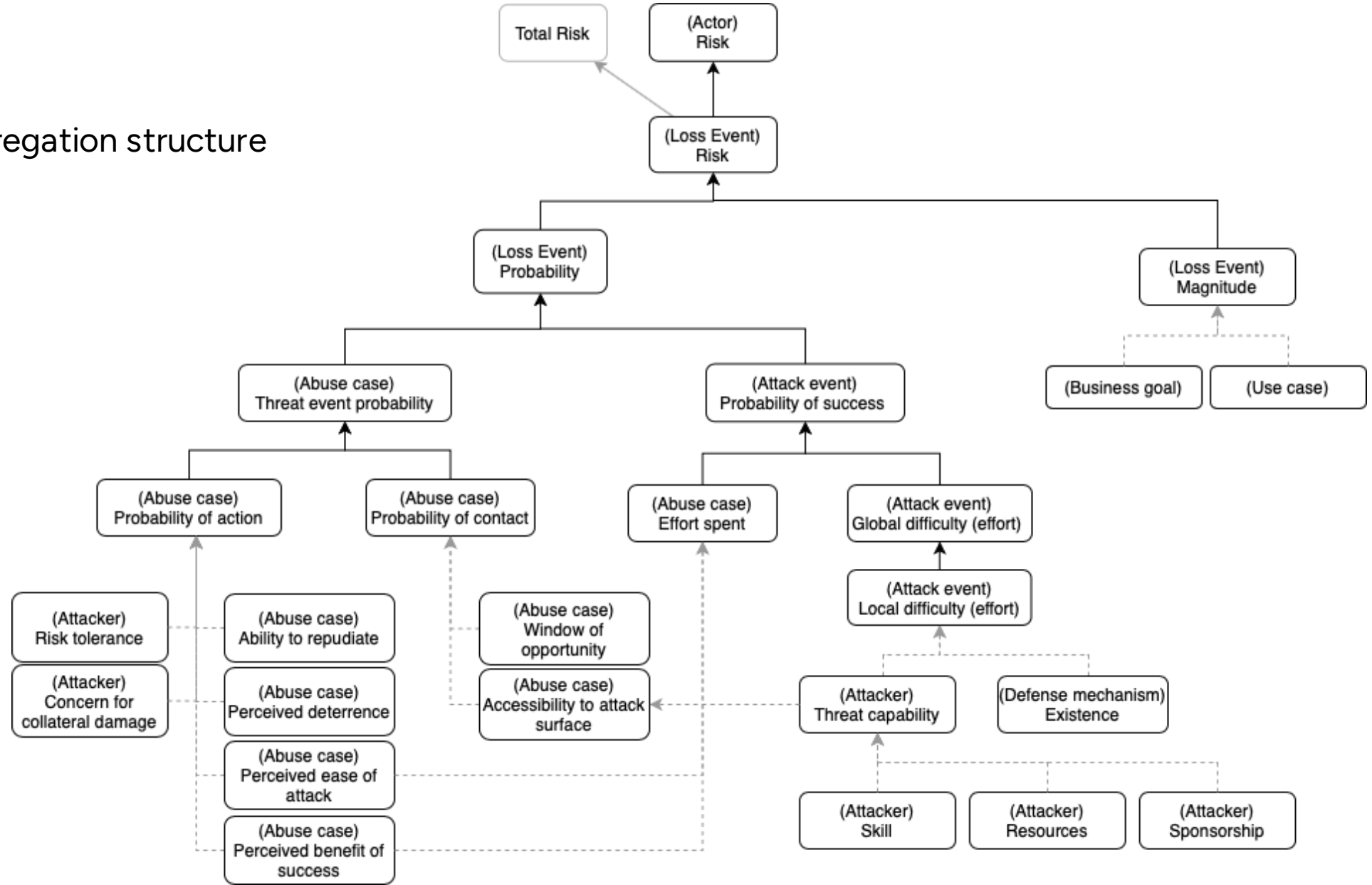
In this phase the result of all previous phases are combined into an overall result.

Phase steps:

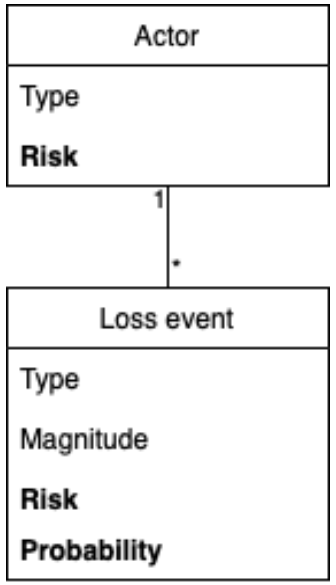
1. Perform overall risk assessment
2. Evaluate protection scenarios (reiterate phases 2-5.1)
3. Recommend course of action(s)

Step 1: Perform overall risk assessment

Overall aggregation structure



Aggregating risk

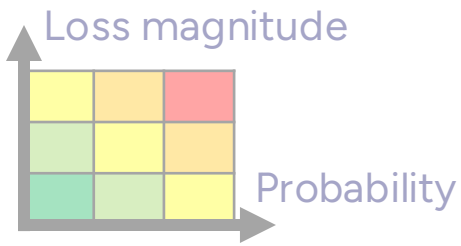


..And we could summarize risk by aggregating/filtering in various ways, e.g.:

- Actor.Type (e.g. all external risk)
- Loss event.Type (e.g. all productivity risk)
- Attack event.Type (e.g. all risk from confidentiality breach)
- Asset (e.g. all risk related to a certain asset)

Aggregating risk - scales and probability

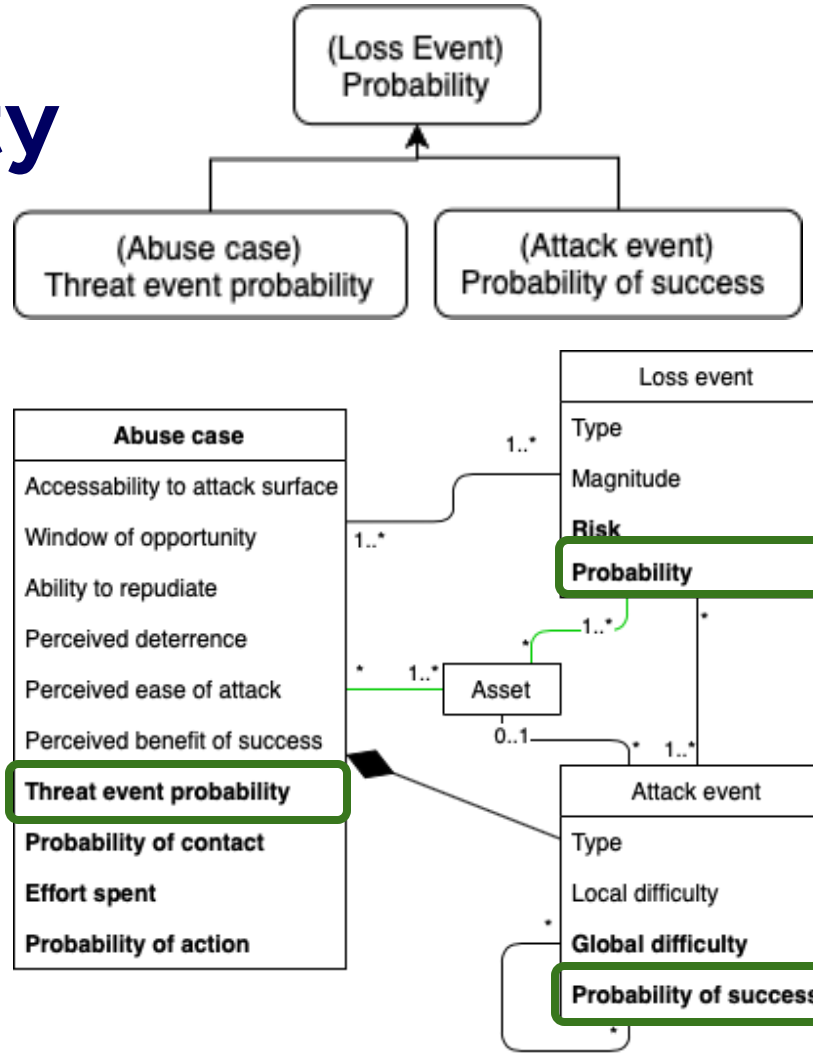
- It is normally difficult quantify loss. But we aim to be as quantitative as possible.
- Possible scales:
  - Monetary - preferred if possible
  - Qualitative scale (1-10, 1-5, low-high…)
  - Appropriate scales for specific loss types (customer satisfaction index, # injury/death, CO2 emission, …)
  - Not all scales can be multiplied with a probability to generate a risk score
    -> Risk matrix:



- Possibly multiple loss magnitude parameters needs to be used.

- *Uncertainties can be aggregated. Preferably through normal statistical approaches and Monte Carlo simulations. Or in our simplified case manually, with e.g. three-point estimates where one calculation is done separately for worst/expected/best case values.*

Aggregating Loss event Probability



Which probabilities? Only those related to the specific Loss event:
Loss event.Attack event ∩
Loss event.Abuse case.Attack event

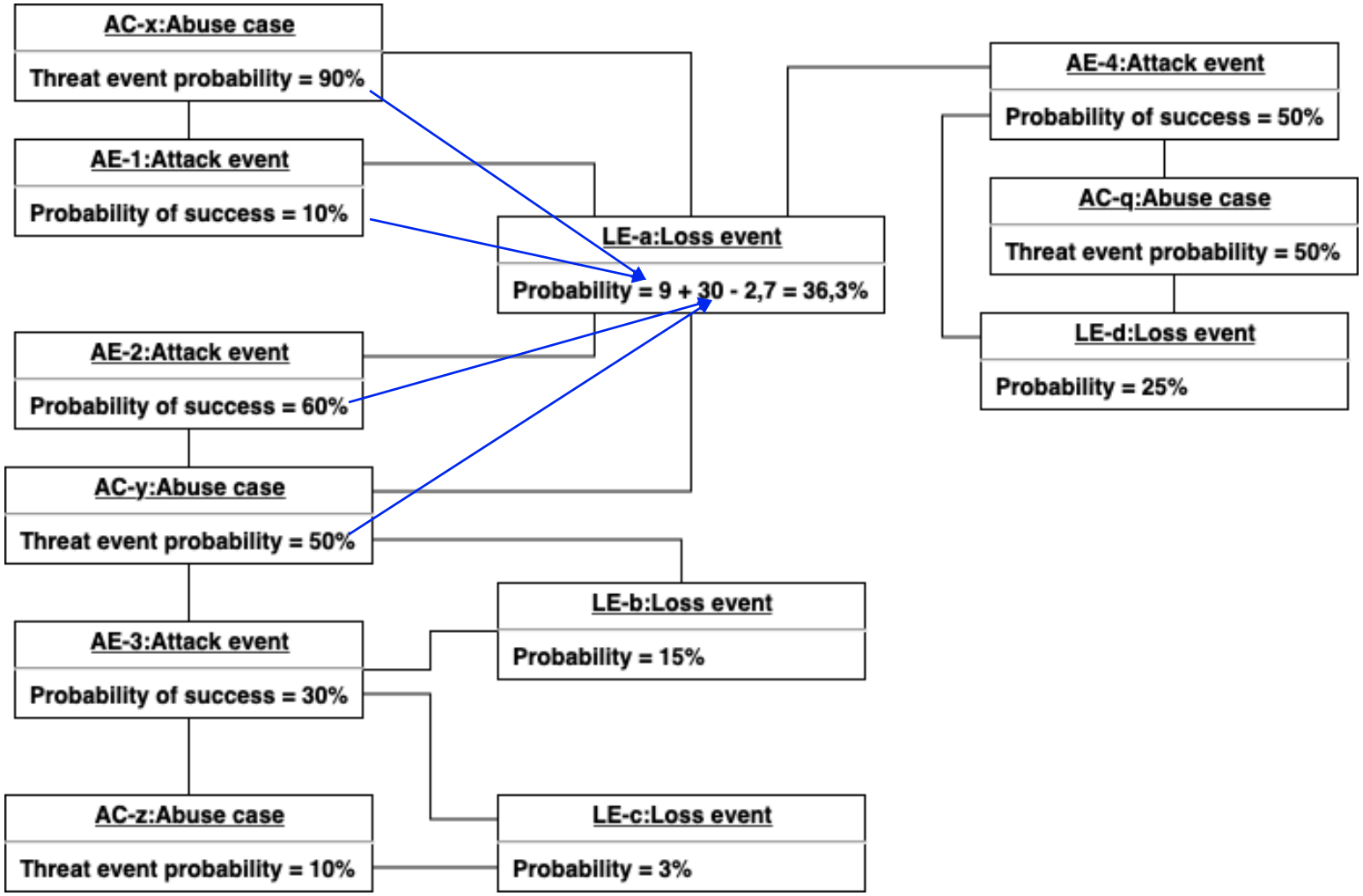
We assume independence between:
- Abuse case.Threat event probability
- Attack event.Probability of success
→ P(Loss event.Probability) = *P(Tep AND Pos) = P(Tep ∩ Pos) = P(Tep)*P(Pos)*

Add all Abuse cases
→ *Union of NOT mutually exclusive events:*
$P(\text{Loss event.Probability}_{\text{Abuse case1}} \cup \text{Loss event.Probability}_{\text{Abuse case2}})$

$$P(A \cup B) = P(A) + P(B) - P(A \cap B) = P(A) + P(B) - (P(A) * P(B))$$

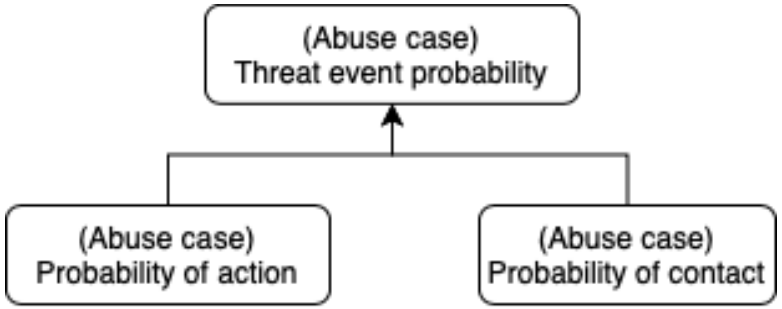
And so on: $P(A \cup B \cup C) = P(A) + P(B) + P(C) - P(A \cap B) - P(A \cap C) - P(B \cap C) + P(A \cap B \cap C)$ etc.

A simple example



Aggregating Abuse case Threat event probability

We assume independence between:
- Probability of action
- Probability of contact
→ Threat event probability =
*P(Poa AND Poc) = P(Poa ∩ Poc) = P(Poa)*P(Poc)*



Abuse case
Accessability to attack surface
Window of opportunity
Ability to repudiate
Perceived deterrence
Perceived ease of attack
Perceived benefit of success
Threat event probability
Probability of contact
Effort spent
Probability of action

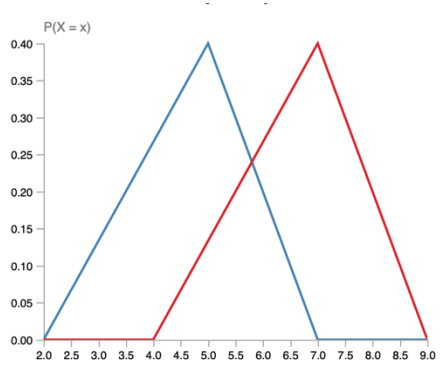
Aggregating Attack event Probability of success

We assume independence between:
- Attack event.Global difficulty (representing how costly it is to succeed with an attack for the attacker)
- Abuse case.Effort spent (representing the cost the attacker is willing to spend)

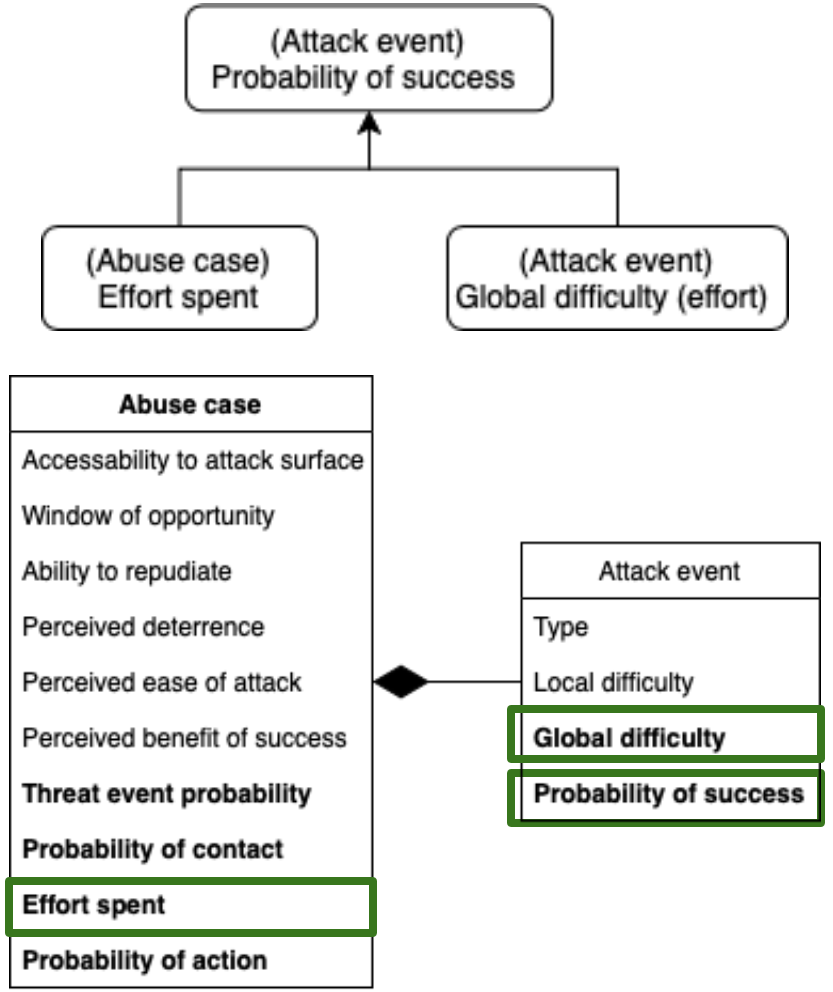
Considering these parameters as probability distributions we can sample these distributions to consider an individual case. And for every case when the sample value of the Effort spent is larger than the global difficulty the attack becomes successful. If we run several samples the fraction of successful samples is our probability.

An example:
- Global difficulty (min/med/max) = 2/5/7
- Effort spent (min/med/max) = 4/7/9

→ Probability of success ~= 9%



For deterministic calculations this probability has to be qualitatively estimated based on the input about the underlying attacker and the threat scenario properties (discussed previously).



Aggregating Attack event Probability of success

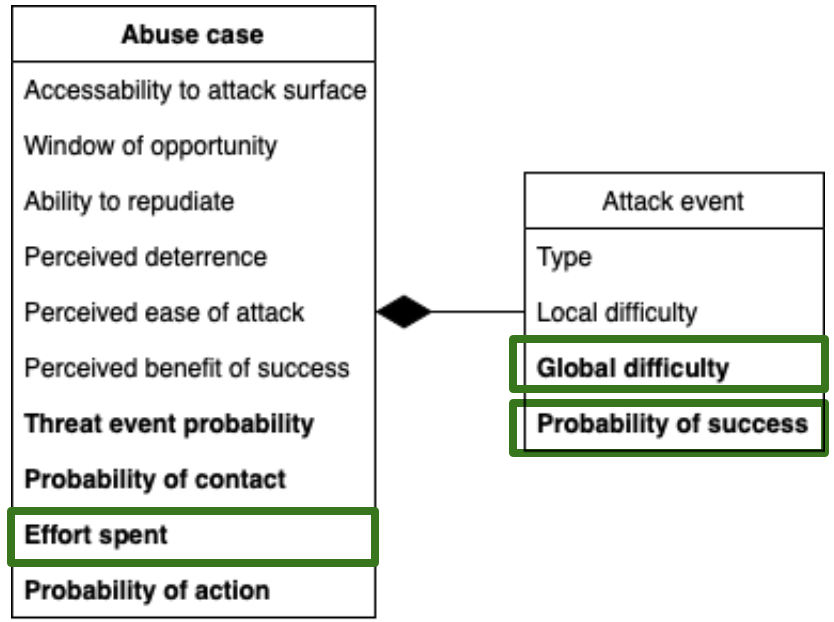
We simplify and assume that the Effort spent is spent on all individual Attack events belonging to the Abuse case. (Theoretically we could think of an Abuse case that is composed of two isolated Attack event chains, the Effort spent would then be applied to both chains which could be considered double counting. Practically though, designing Abuse cases with such independent Attack event chains normally doesn't make sense.)

Moreover, Attack events can be composed in multiple Abuse cases. We treat the Effort spent in the different Abuse cases as independent and thus combine these efforts by taking the *Union of NOT mutually exclusive* Abuse cases:

$P(\text{Attack event.Probability of success}_{\text{Abuse case1}} \cup \text{Attack event.Probability of success}_{\text{Abuse case2}})$

Remember $P(A \cup B) = P(A) + P(B) - P(A \cap B) = P(A) + P(B) - (P(A) * P(B))$

(Again, practically we perhaps normally do not assign Attack events to multiple Abuse cases.)



Summarizing risk

- So, risk can be aggregated into a single indicator.
- As we have seen we might want to nuance the risk into various types or actor exposure, but we might also want to treat different threats differently. E.g. perhaps we have different risk tolerance for insider attackers, APTs and criminals.
- Here there is no right and wrong, either case just has to be motivated.

Step 2 - Evaluate protection scenarios

Up until here we have analyzed the base case, normally the as-is situation of the organization. Now the question is what should we do (to improve security)?

As this method is intended as support for IT staff the “design space” is limited to the IT domain, where we can:
– add defense mechanisms
– remove vulnerabilities
– restructure the the system architecture (while still maintaining business services)

Essentially this means reiteration of phases 2-5.1 but where most changes will happen in phase 2 and 4. For some design suggestions a lot of the analysis of the abuse cases from base case can be reused. For other suggestions new analyses might be needed.

Of course, there is an infinite number of potential change scenarios that could be made. Making thorough threat analysis according to our method of them all is not viable. Instead, we want to threat model and analyze a few alternatives that all look most promising at glance, e.g. implementing security best practices.

Each scenario would lead to (hopefully) a reduced risk as compared to the base case.

Step 3 - Recommend course of action(s)

Once we have a number of scenarios analyzed with respect to the risk the question is how do we value their overall effectiveness in fulfilling our overall goals with respect to cyber security management.

Of course, the main indicator is the risk value we calculated. But all scenarios come with a cost in terms of buying, building, and operating the components of the scenario. Management deciding on implementation are thus interested in having the alternatives ordered according to “Return on Security Investment (RoSI)”. Thus, the scenarios have to be complemented with cost estimates (one or several depending how detailed you want to be). Practically there is normally also a limited budget available that might disqualify some scenarios. In this course however we do not bother with that.

Thus far we can think “best RoSI, best scenario”, however there might other factors complicating the overall evaluation. One scenario might move the risk from one actor to another, we might have good insurance for a particular type of attack, one scenario might include solutions that will irritate employees and users, one scenario perhaps also impacts the overall business effectivity, etc. Generally speaking we try to normalize such additional factors, and in this course they are not the main focus so they need only to be briefly discussed.

In summary, order your change scenarios and briefly summarize the motivation (coming from the threat modeling and analysis) of your suggested course of action.

A general comment on scope and level of detail

A general question is how detailed and wide the analysis should be. At a high level this is decided by the defined scope and available time; cover the full scope as detailed as time allows! However, bottom line we want to have the most details where there is most risk. As we know risk depend on all parts of the threat model (the impact, the threat actor, the system resilience) so we cannot work strictly top-down. I.e. if we detail around the largest impacts, we might miss important details about the system vulnerabilities that might drive the largest risks. Ideally, we should iterate the phases many times. That said, the choice of widening and detailing the model should always be guided by available information and the model in its as-is state. E.g. in the first iteration it is better to let the high-impact assets guide the detailing generally speaking, but not completely.

Misc

Our risk calculation model – comparison to FAIR

Overall, the risk aggregation method in this course largely follows the FAIR method. Biggest change is the the way system resilience is estimated. Moreover, we have replaced frequencies with probabilities.