The Power of Together

Main objectives

- Develop a LFE approach for a specific security risk analysis, needed security measures and an architecture for a use cases of LFE.
- We will 'test' this approach for an important use case, the compas module. Therefore, the 'test' is serious, and it should deliver more or less serious a security advise for the project compas module itself.

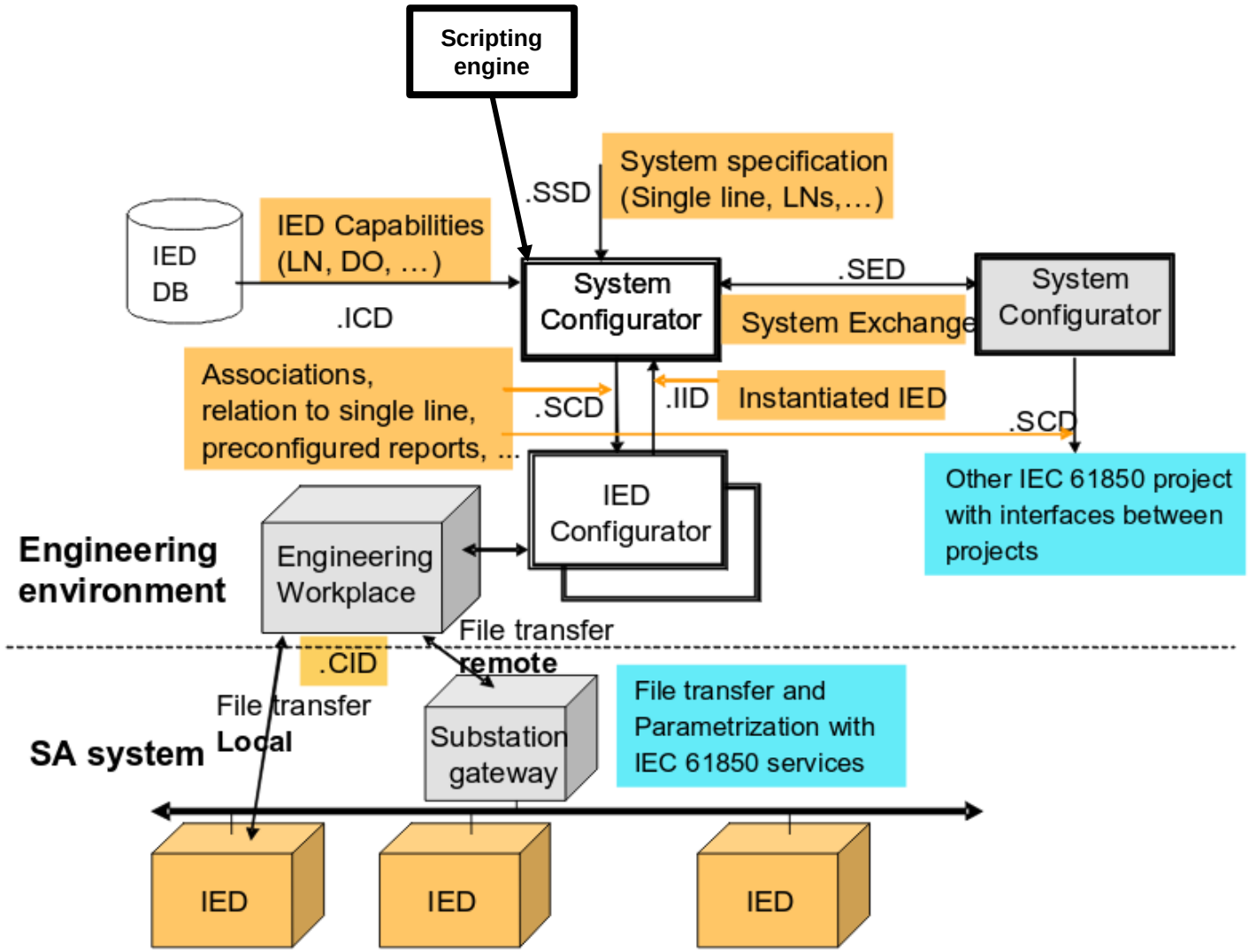
Agenda

- Welcome
- Part 1 – Re-cap scope of the use case
- Part 2 – Security risk analysis of the use case
  - Security impact assessment with Stride Approach
  - Threat Modeling with BowTies
- Part 3 – Security measures for the use case
- Part 4 – Security architectural principles for the use case
- Discussion
  - David Wheeler
  - Sander Jansen
- Next steps

Welcome

- Introduction participants

Part 1 – Recap scope of the use case



The main functional blocks:

- System configuration: “System Specification Description (SSD)” to “Substation Configuration Description (SCD)” conversion, PACS policy registry (scripts?), API to vendor specific IED configurators
- IEC61850 profile management: logical device/function builder, library of common profiles for usual functions, versioning, definition of reusable user profile of IEC 61850 data model (potentially continue/restart ENTSO-E profiling tool)
- Conformity verification of System Configuration description Language (SCL) files
- System specification: profile to “System Specification Description (SSD)” conversion, PACS policy registry (scripts?), API to vendor specific “IED Capability Description (ICD)” tools, ICD conformity check, ICD compatibility management, ICD versioning / repository
- Availability of Substation PACS data at enterprise level (Functions & settings, operational process data)

Part 2 – Security risk analysis

Security risk assessment

Agenda:

- **Impact assessment:** analyze what information is processed by the application and what the impact is if the confidentiality, integrity or availability is compromised
- **Threat assessment:** analyze how the data could be compromised

Impact assessment

Information assets

Information asset	Compromise of confidentiality	Compromise of integrity	Compromise of availability
Configuration (by means of the Compas Module) for setting the protection relays.		Protection relay will not be active (if the configuration by means of the Compas Module is not integer), when there is a primary eletrotechnical fault. The result clould be physical damage substation. When it comes to the threat assessment a distinction should be made between: - Processing of the data. - The database (DB) where the data are retrieved.	

Information assets

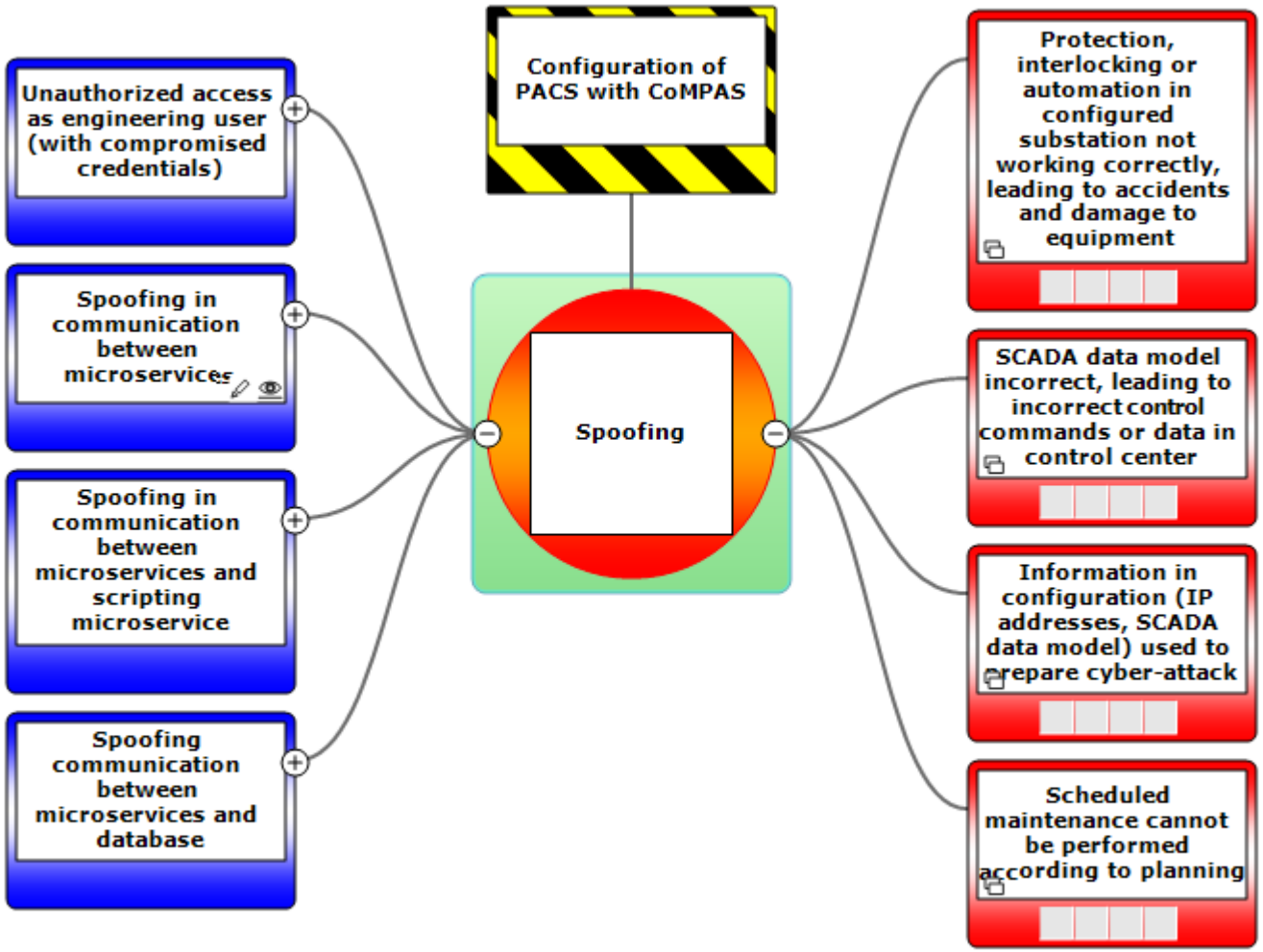
Information asset	Compromise of confidentiality	Compromise of integrity	Compromise of availability
Configuration setting by means of the Compas Module for maintenane	If the data are stolen, the threat actor has a lot knowledge of the specific configuration and architecture of a substation, which gives a basic knowledge for a threat scenario which could effect even the cross border high voltage grid. For instance, the SCD file contains all the IP-adresses. It is a XML-file. There are no passwords, but a lot of information of the substation, its position in the whole grid and its connections (hence, big power plant).	The impact could be rework. When there are tests, probably the issues with integrity will be identified. If the test gives the wrong result, there is impact (rework) .	The use of the Compas Module is for big projects. How critical the Compas Module is, is dependent on the way of working of the TSO/DSO. The impact is probably limited, because you could copy versions of other substations, and proceed with this data for maintenance for other substations. However, if the availability is longer, the impact is higher.

Information assets

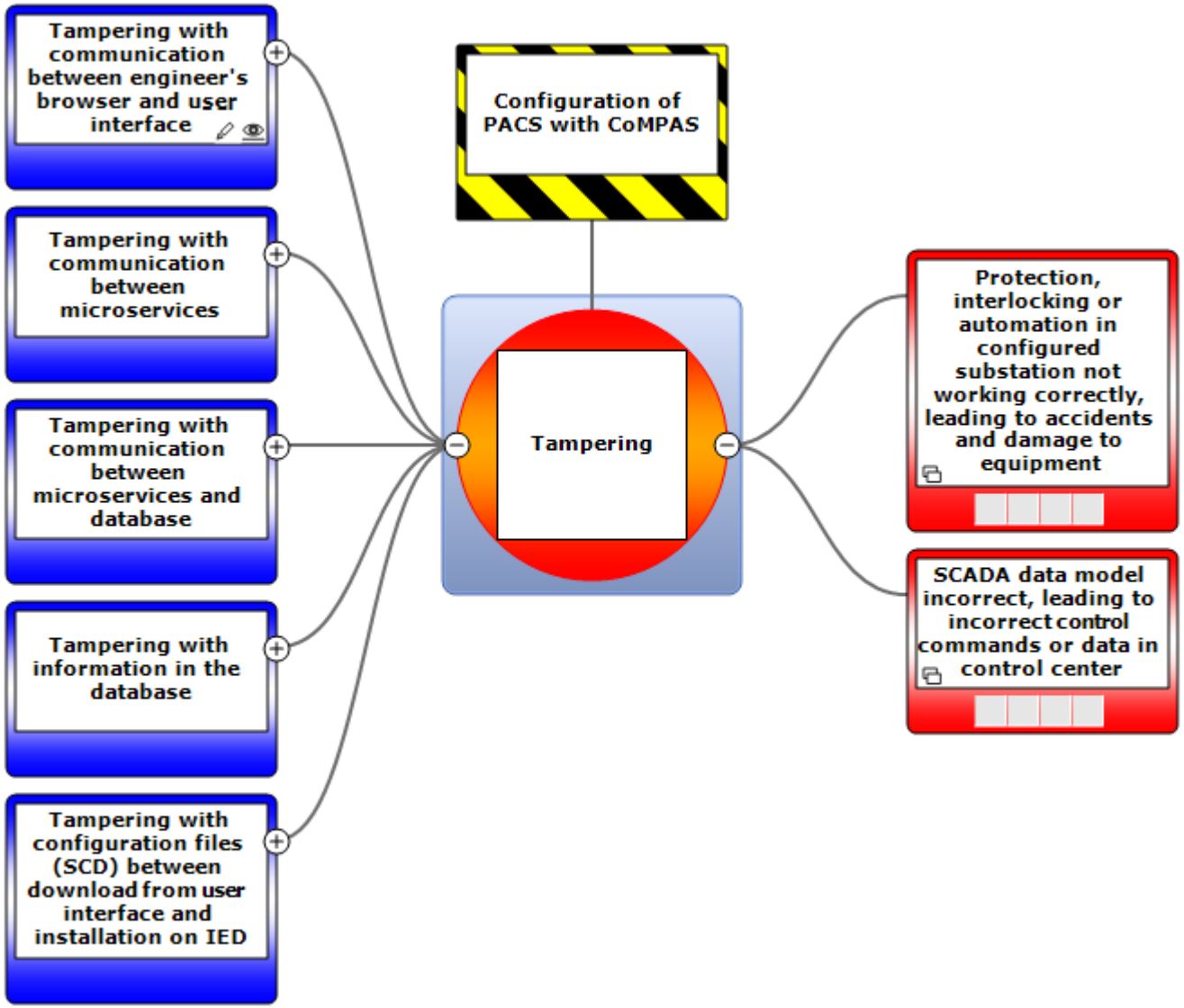
Information asset	Compromise of confidentiality	Compromise of integrity	Compromise of availability
Substation is present in the SCADA by means of the Compas Module ('SCADA modeling')	With this data, the hacker can understand how the status data, the measurement data etc. are made (based on the configuration)		Limited
Substation can be switched in the SCADA by means of the Compas Module ('SCADA modeling')	If this data is stolen, the hacker could make its own switch command, inject this message in the network (to the dispatching centre and/or substations) and the hacker is able to create significant impact on the grid.	If the command is corrupted, it is visible (no switch)	Limited

Threat assessment

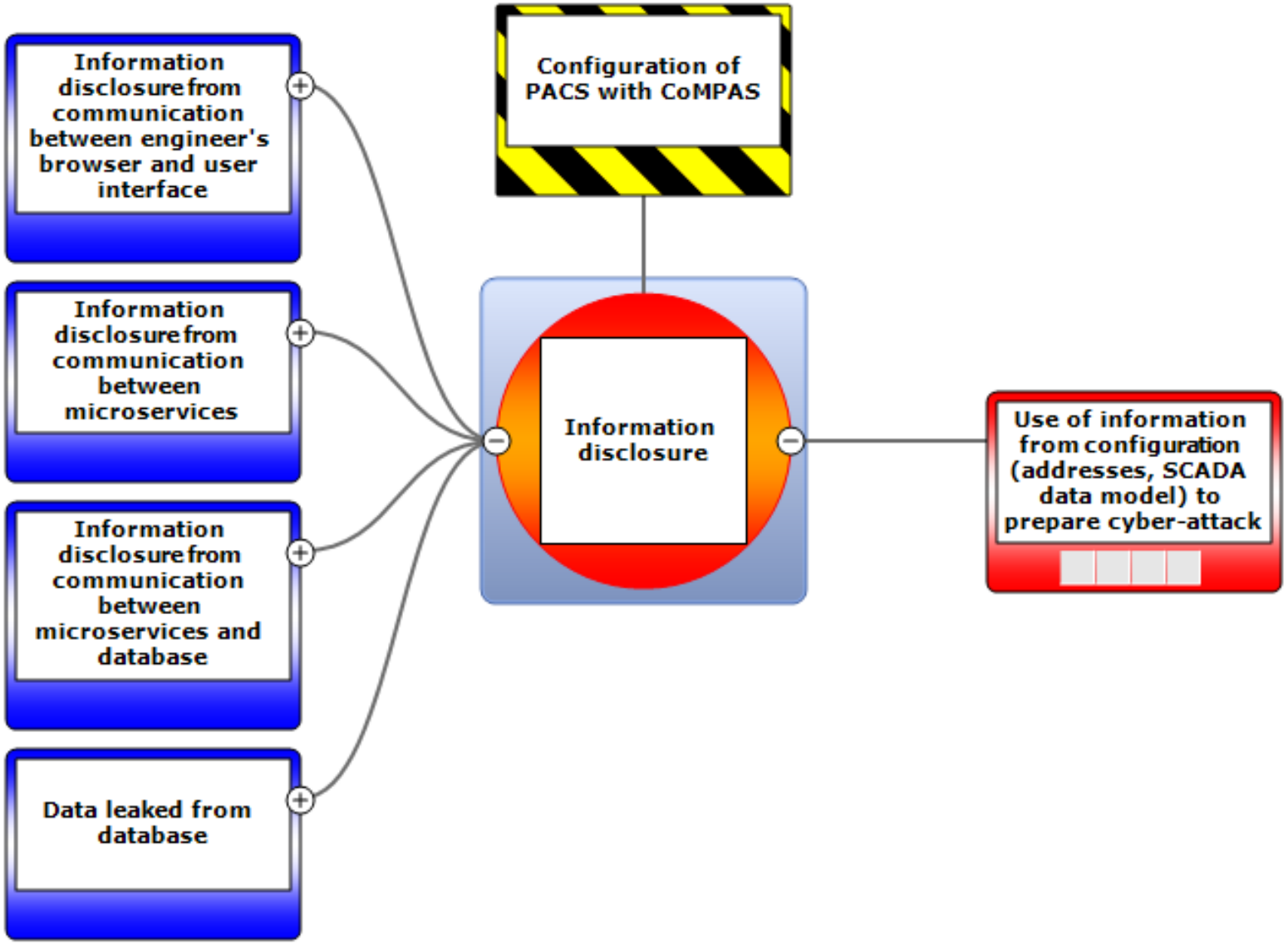
Spoofing


Unauthorized access as engineering user (with compromised credentials)
Spoofing in communication between microservices
Spoofing in communication between microservices and scripting microservice
Spoofing communication between microservices and database
Configuration of PACS with CoMPAS
Spoofing
Protection, interlocking or automation in configured substation not working correctly, leading to accidents and damage to equipment
SCADA data model incorrect, leading to incorrect control commands or data in control center
Information in configuration (IP addresses, SCADA data model) used to prepare cyber-attack
Scheduled maintenance cannot be performed according to planning

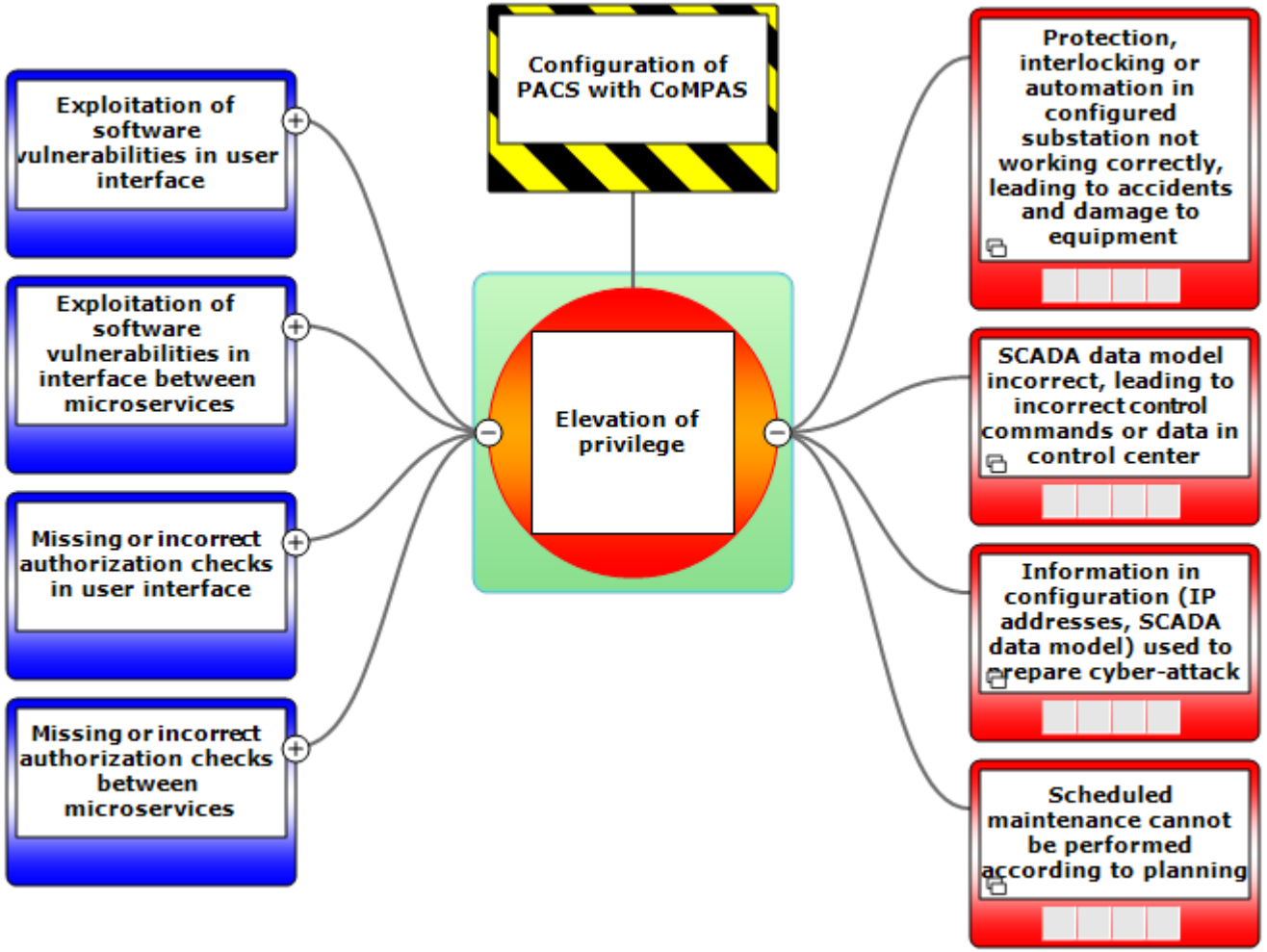
Tampering


Tampering with communication between engineer's browser and user interface
Tampering with communication between microservices
Tampering with communication between microservices and database
Tampering with information in the database
Tampering with configuration files (SCD) between download from user interface and installation on IED
Configuration of PACS with CoMPAS
Tampering
Protection, interlocking or automation in configured substation not working correctly, leading to accidents and damage to equipment
SCADA data model incorrect, leading to incorrect control commands or data in control center

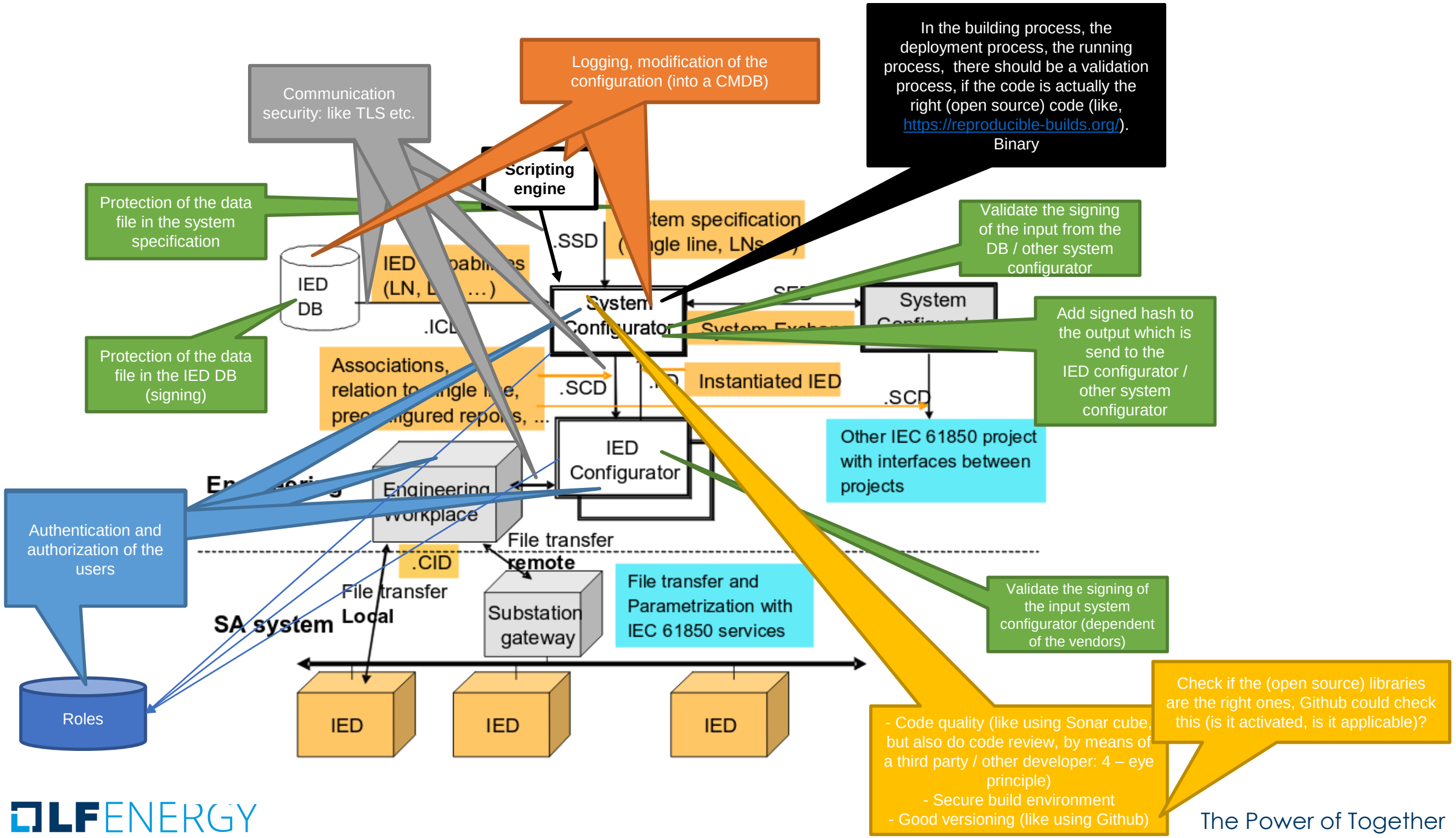
Information disclosure



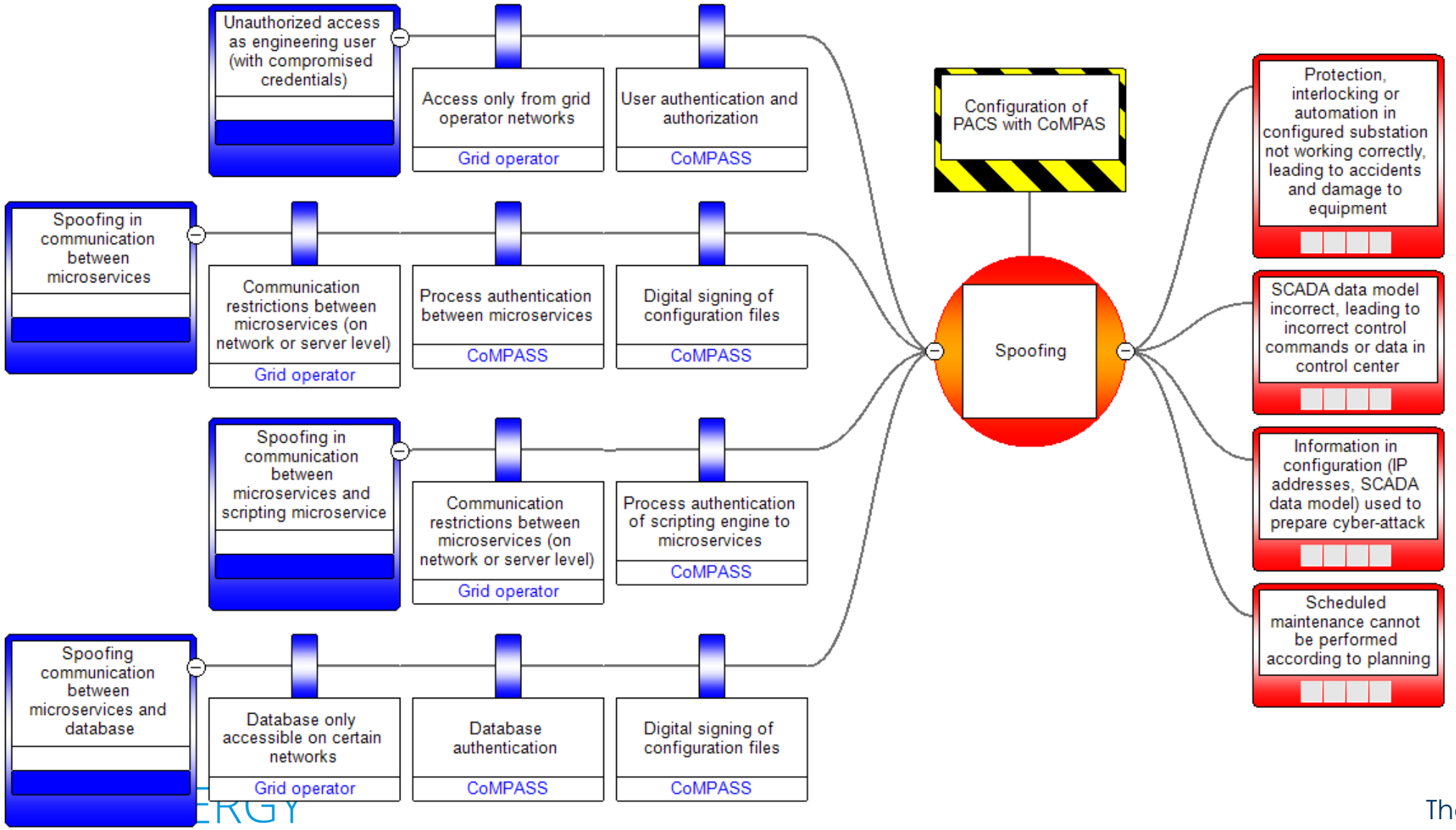
Elevation of privilege



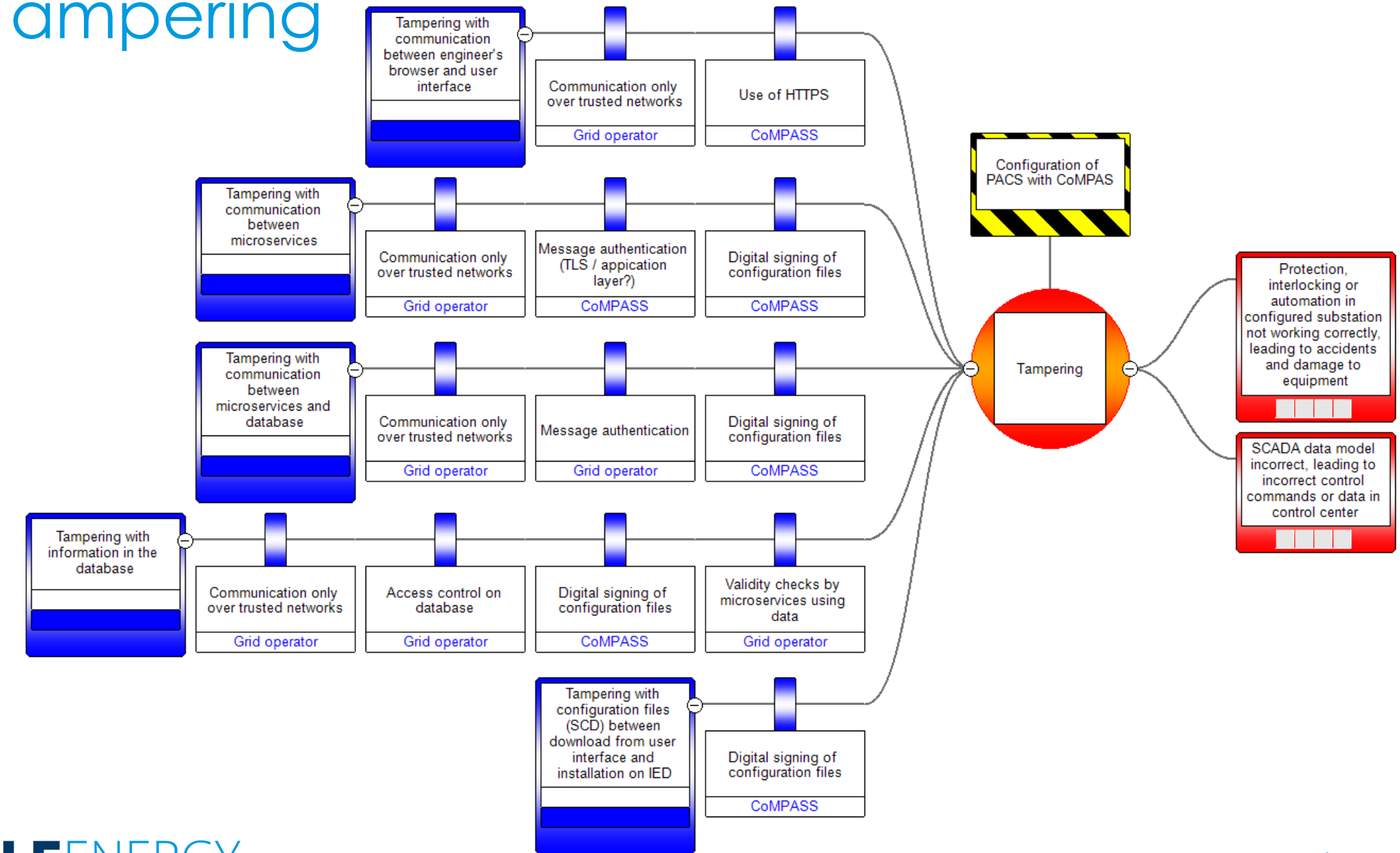
Part 3 – Security measures for the use case


In the building process, the deployment process, the running process, there should be a validation process, if the code is actually the right (open source) code (like, https://reproducible-builds.org/). Binary
Logging, modification of the configuration (into a CMDB)
Communication security: like TLS etc.
Protection of the data file in the system specification
Protection of the data file in the IED DB (signing)
Validate the signing of the input from the DB / other system configurator
Add signed hash to the output which is send to the IED configurator / other system configurator
Scripting engine
.SSD
IED DB
.ICD
System Configurator
System
Instantiated IED
.SCD
Associations, relation to single line, preconfigured reports, ...
IED Configurator
Other IEC 61850 project with interfaces between projects
Engineering Workplace
Authentication and authorization of the users
File transfer
remote
.CID
File transfer
Local
SA system
Substation gateway
File transfer and Parametrization with IEC 61850 services
Validate the signing of the input system configurator (dependent of the vendors)
Roles
IED
IED
IED
- Code quality (like using Sonar cube, but also do code review, by means of a third party / other developer: 4 – eye principle)
- Secure build environment
- Good versioning (like using Github)
Check if the (open source) libraries are the right ones, Github could check this (is it activated, is it applicable)?

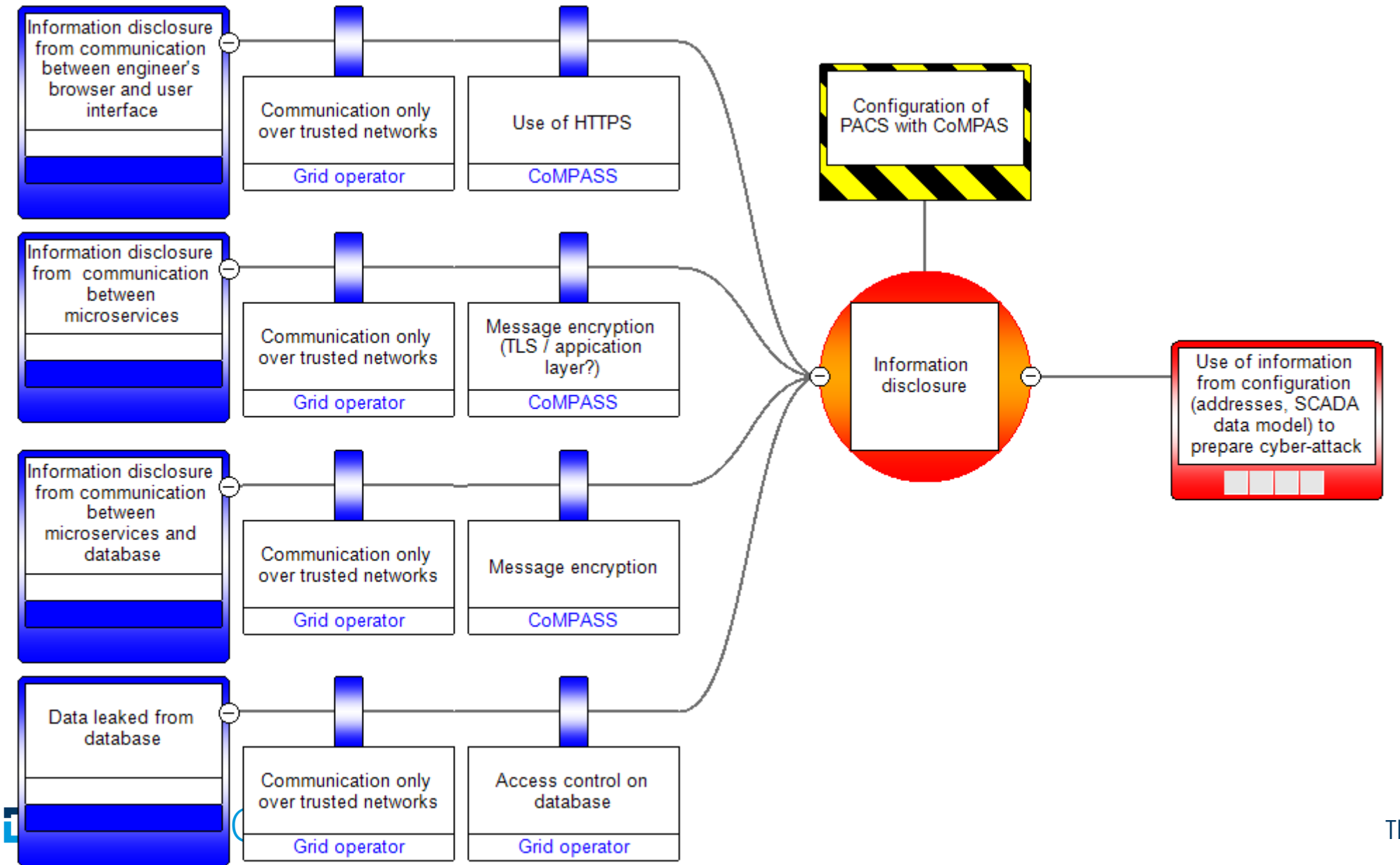
Spoofing


Unauthorized access as engineering user (with compromised credentials)
Access only from grid operator networks
Grid operator
User authentication and authorization
CoMPASS
Spoofing in communication between microservices
Communication restrictions between microservices (on network or server level)
Grid operator
Process authentication between microservices
CoMPASS
Digital signing of configuration files
CoMPASS
Spoofing in communication between microservices and scripting microservice
Communication restrictions between microservices (on network or server level)
Grid operator
Process authentication of scripting engine to microservices
CoMPASS
Spoofing communication between microservices and database
Database only accessible on certain networks
Grid operator
Database authentication
CoMPASS
Digital signing of configuration files
CoMPASS
Configuration of PACS with CoMPAS
Spoofing
Protection, interlocking or automation in configured substation not working correctly, leading to accidents and damage to equipment
SCADA data model incorrect, leading to incorrect control commands or data in control center
Information in configuration (IP addresses, SCADA data model) used to prepare cyber-attack
Scheduled maintenance cannot be performed according to planning

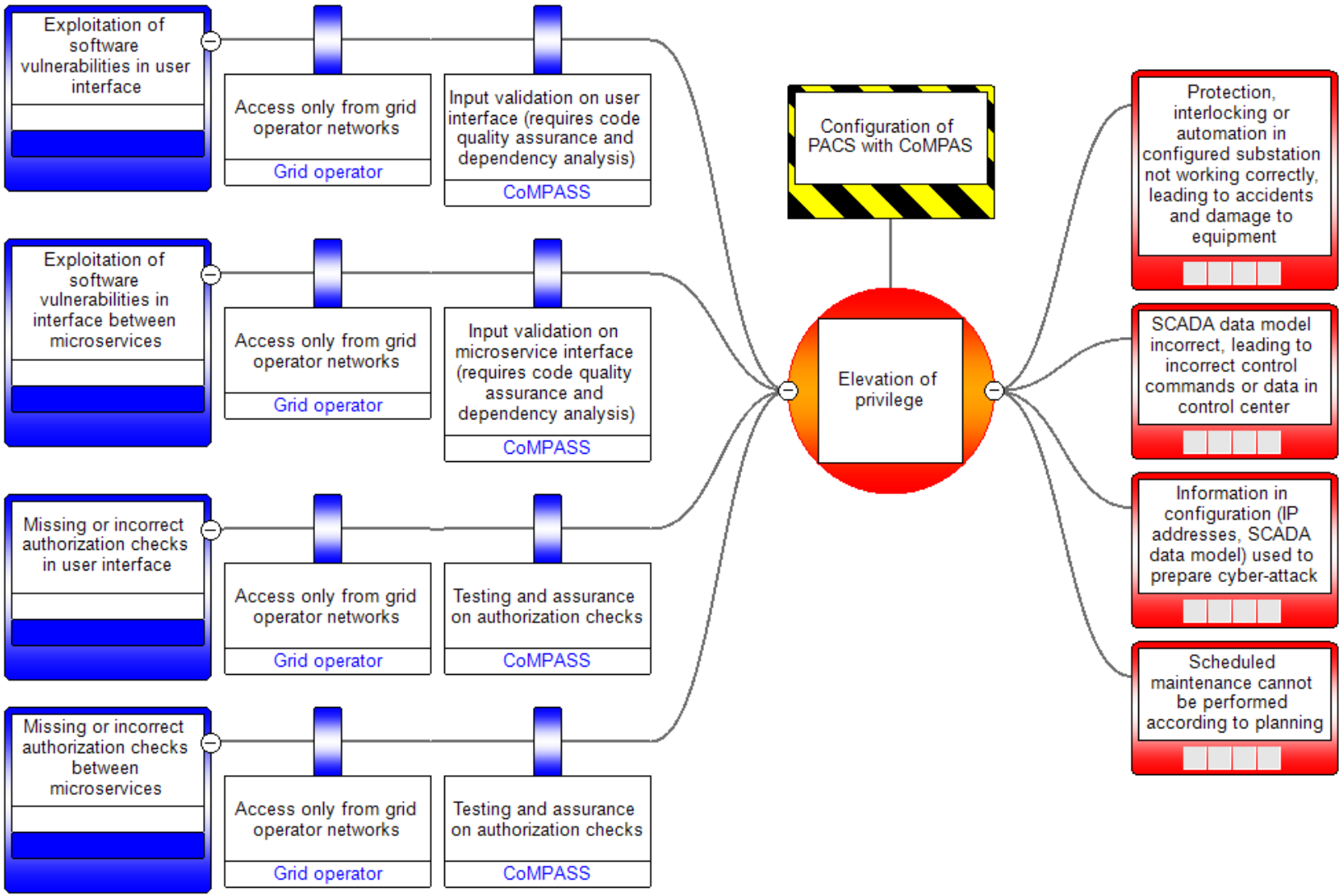
Tampering


Tampering with communication between engineer's browser and user interface
Communication only over trusted networks
Grid operator
Use of HTTPS
CoMPASS
Tampering with communication between microservices
Communication only over trusted networks
Grid operator
Message authentication (TLS / appication layer?)
CoMPASS
Digital signing of configuration files
CoMPASS
Tampering with communication between microservices and database
Communication only over trusted networks
Grid operator
Message authentication
Grid operator
Digital signing of configuration files
CoMPASS
Tampering with information in the database
Communication only over trusted networks
Grid operator
Access control on database
Grid operator
Digital signing of configuration files
CoMPASS
Validity checks by microservices using data
Grid operator
Tampering with configuration files (SCD) between download from user interface and installation on IED
Digital signing of configuration files
CoMPASS
Configuration of PACS with CoMPAS
Tampering
Protection, interlocking or automation in configured substation not working correctly, leading to accidents and damage to equipment
SCADA data model incorrect, leading to incorrect control commands or data in control center

Information disclosure



Elevation of privilege

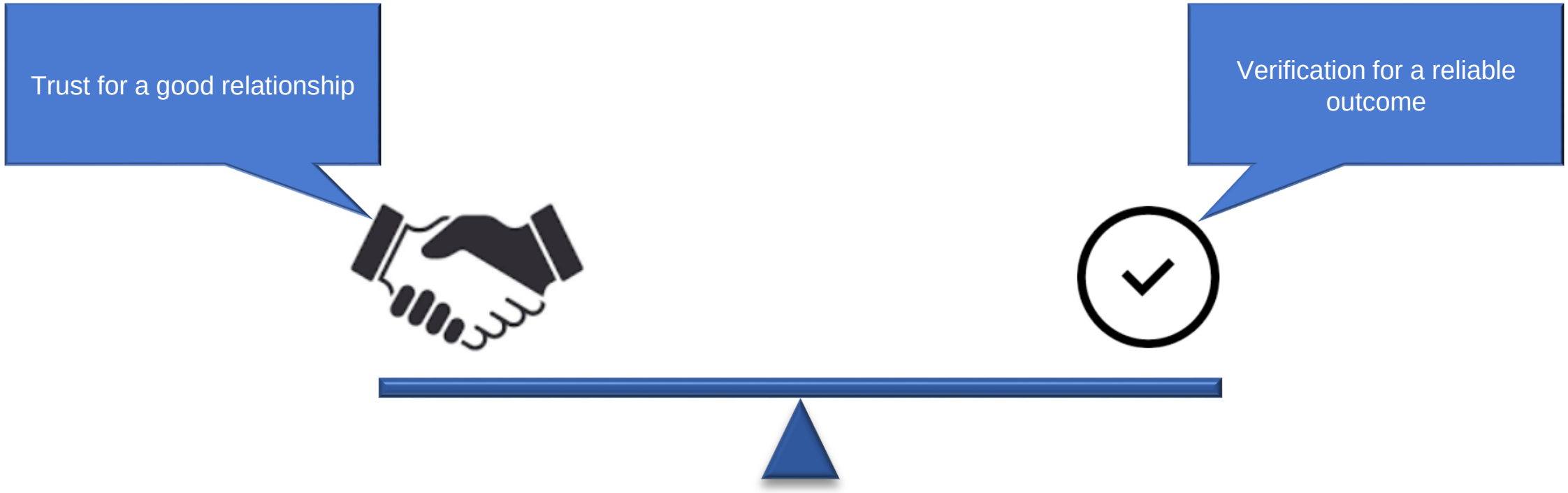


Part 4 – Security architectural principles for the use case

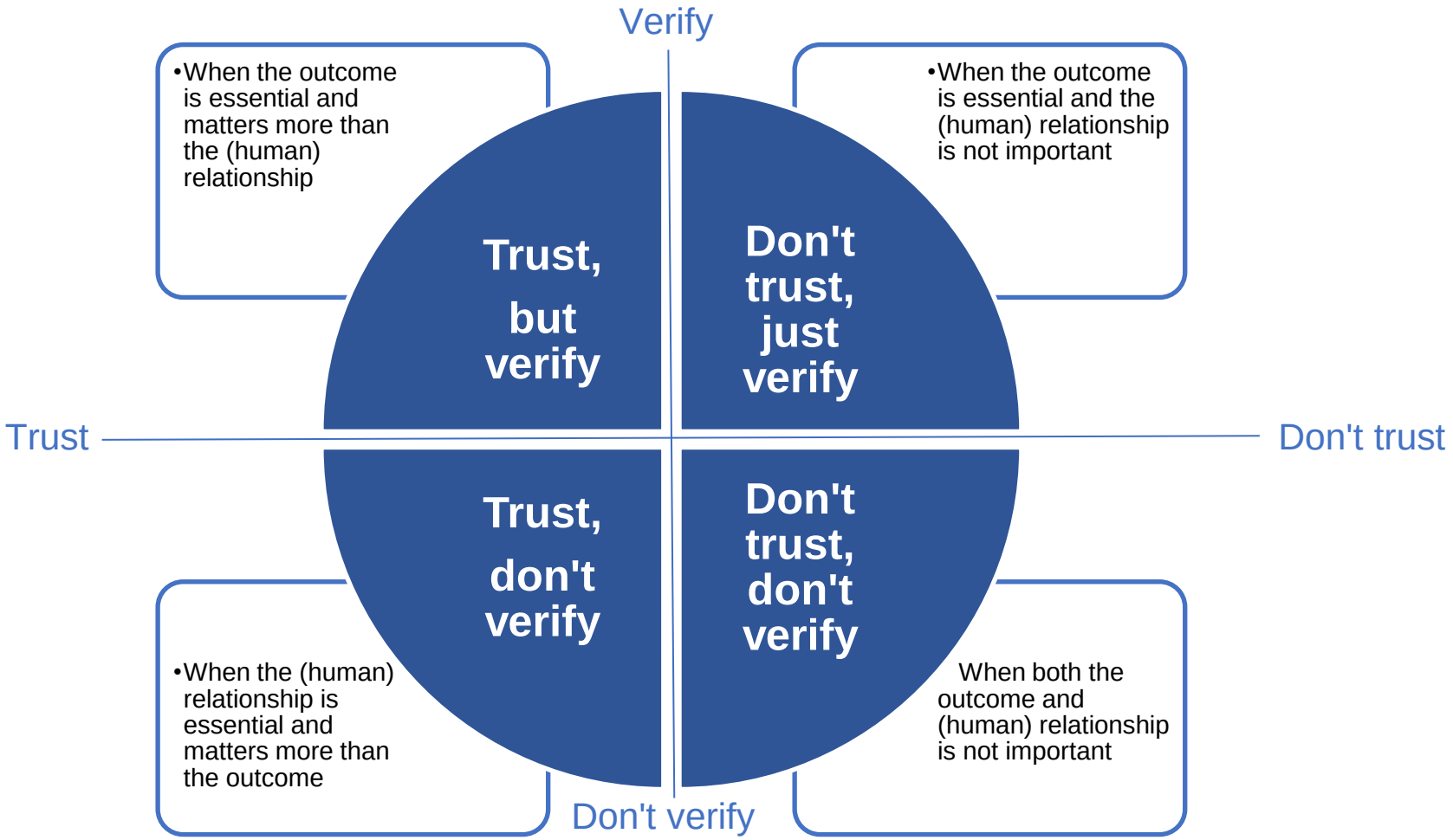
The differences

Closed systems	Open platforms
Limited amount of (internal) stakeholders	Lots of (external) stakeholders
Limited set of internal interests	Lots of external 3rd party interests
Total control in our own hands	More control in the marked, out of our hands
Unidirectional information flows from high to lower secure zones	Bidirectional information flows
Exclusive use of own information sources	Use of external information sources for automated decision-making
Information for safety and efficiency	Information for safety, efficiency and money (Congestion Management)
	Information becomes 'money'
	More APT's active in energy sector

The balance between trust and verification



Security architectural principles for the use case



Discussion

David Wheeler

- I strongly encourage using a dependency analyzer (aka software composition analyzer aka origin analyzer) to look for included software with known vulnerabilities. They won't catch all publicly-known vulnerabilities in your software system, but they're a big help.
- So if you're on GitHub, enable it. GitLab also provides this service (via Gymnasium).
- If you can, I suggest using at least two different ones; none are perfect, and in particular they differ on what components they spot & the databases they use. LF projects will also be able to get this service via CommunityBridge (that one is based on Synk).
- These tools are *much* more effective if you work *with* them. In particular, use a package manager where possible, so that the tool can simply examine the package manager's database to determine what's being reuse. Some tools can look line-by-line, but they're more expensive & that will be less accurate (because the task they're trying to do is FAR more difficult).

Sander Jansen

- This is the Github feature for dependancies:
- https://help.github.com/en/github/managing-security-vulnerabilities/about-alerts-for-vulnerable-dependencies
- It is already used in the GXF project:
- https://github.com/OSGP/open-smart-grid-platform/pulls
- Github makes a disclaimer:
- Note: GitHub's security features do not claim to catch all vulnerabilities. Though we are always trying to update our vulnerability database and alert you with our most up-to-date information, we will not be able to catch everything or alert you to known vulnerabilities within a guaranteed time frame. These features are not substitutes for human review of each dependency for potential vulnerabilities or any other issues, and we recommend consulting with a security service or conducting a thorough vulnerability review when necessary.

Next Steps