

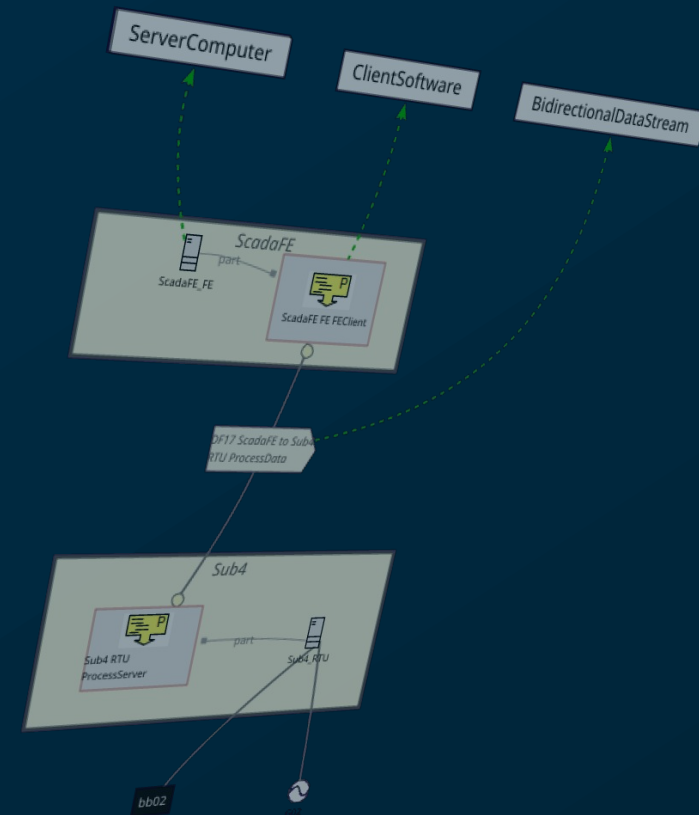


HÖGSKOLAN
I SKÖVDE

Model-Based Cybersecurity Analysis: Extending Enterprise Modeling to Critical Infrastructure Cybersecurity

A Tutorial

MANFRED JEUSFELD
WITH HELP FROM YUNING JIANG



READINGS

E.D. Knapp: Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems, ISBN 978-0443137372, Syngress, Third Edition, 2024.
(textbook)

Gupta, B.B., Akhtar, T.: *A survey on smart power grid: frameworks, tools, security issues, and solutions*. Ann. Telecommun. 72, 517–549 (2017). <https://doi.org/10.1007/s12243-017-0605-4>

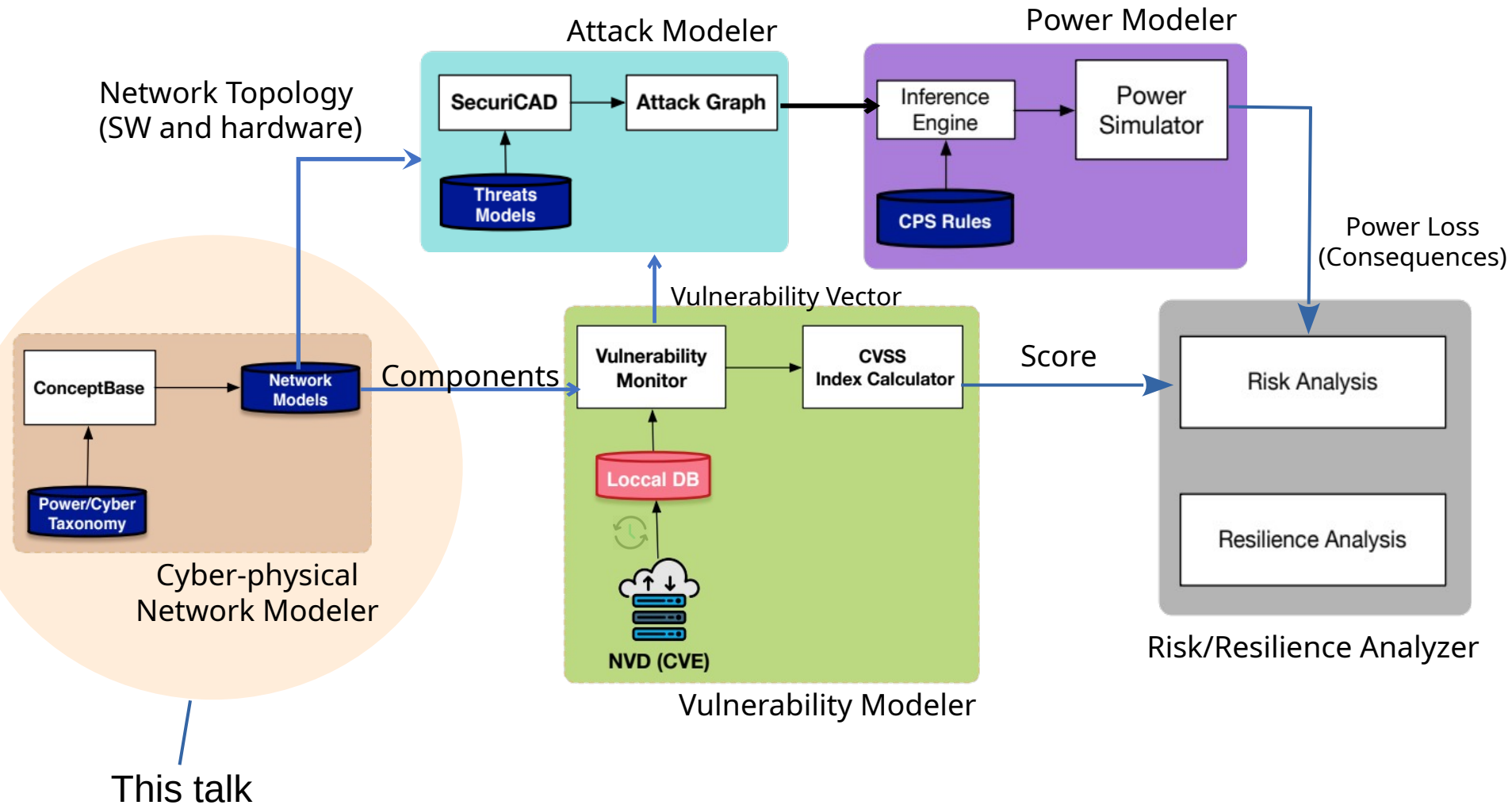
Jingcheng Gao Jing Liu Bharat Rajan Rahul Nori Bo Fu Yang Xiao Wei Liang C. L. Philip Chen: *SCADA communication and security issues*. Security and Communication Networks, Volume 7, Issue 1, <https://onlinelibrary.wiley.com/doi/full/10.1002/sec.698>

P. Chopade and M. Bikdash: *Critical infrastructure interdependency modeling: Using graph models to assess the vulnerability of smart power grid and SCADA networks*. 2011 8th International Conference & Expo on Emerging Technologies for a Smarter World, New York, NY, 2011, pp. 1-6. <https://ieeexplore.ieee.org/abstract/document/6135885>

Y. Jiang, M.A. Jeusfeld, J. Ding, E. Sandahl:
Model-Based Cybersecurity Analysis - Extending Enterprise Modeling to Critical Infrastructure Cybersecurity. Bus. Inf. Syst. Eng. 65(6): 643-676 (2023). <https://doi.org/10.1007/s12599-023-00811-0>

The accompanying web page for this tutorial is at
<https://conceptbase.sourceforge.net/elvira-edoc2026>

ELVIRA PROJECT CONTEXT



Project site:his.se/elvira

SOFTWARE FOR THIS TUTORIAL



This tutorial uses the open-source ConceptBase.cc modeling system:
<https://conceptbase.sourceforge.net/>

Instructions to download and install for Linux, MacOS and Windows are at
<https://conceptbase.sourceforge.net/CB-Download.html>

The user manual is available at
<https://conceptbase.sourceforge.net/userManual85/>

Source code of the models and graph files of examples are available from
<https://conceptbase.sourceforge.net/elvira-edoc2026>

Some model examples in the 2nd part of the tutorial some with web links like [fig1.gel]. These links directly open the model with ConceptBase if you installed it on your computer.

OUTLINE

- Managed systems, real-time systems
- Network architectures
- Control systems components and their management
- A taxonomy of cyber-physical system components
- Modeling cyber-physical systems with ConceptBase

SYSTEMS

”A system is a set of components (sometimes being sub-systems) that are related to each other”

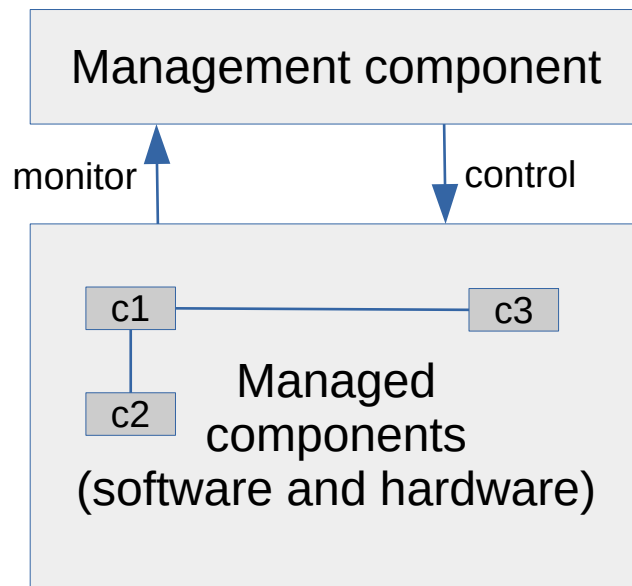
Examples:

- our solar system consisting of the sun, the planets, and millions of asteroids comets bound together by gravitation
- the system of rivers and lakes in Sweden
- the railway system of Sweden
- the transportation system of Sweden (including the railway system)
- the European railway system
- the power grid system of Skövde
- the district heating system of Skövde
- the logistics system of company XYZ
- the production line system of company ABC

MANAGED SYSTEMS

”A system with a management component that monitors and controls its operation”

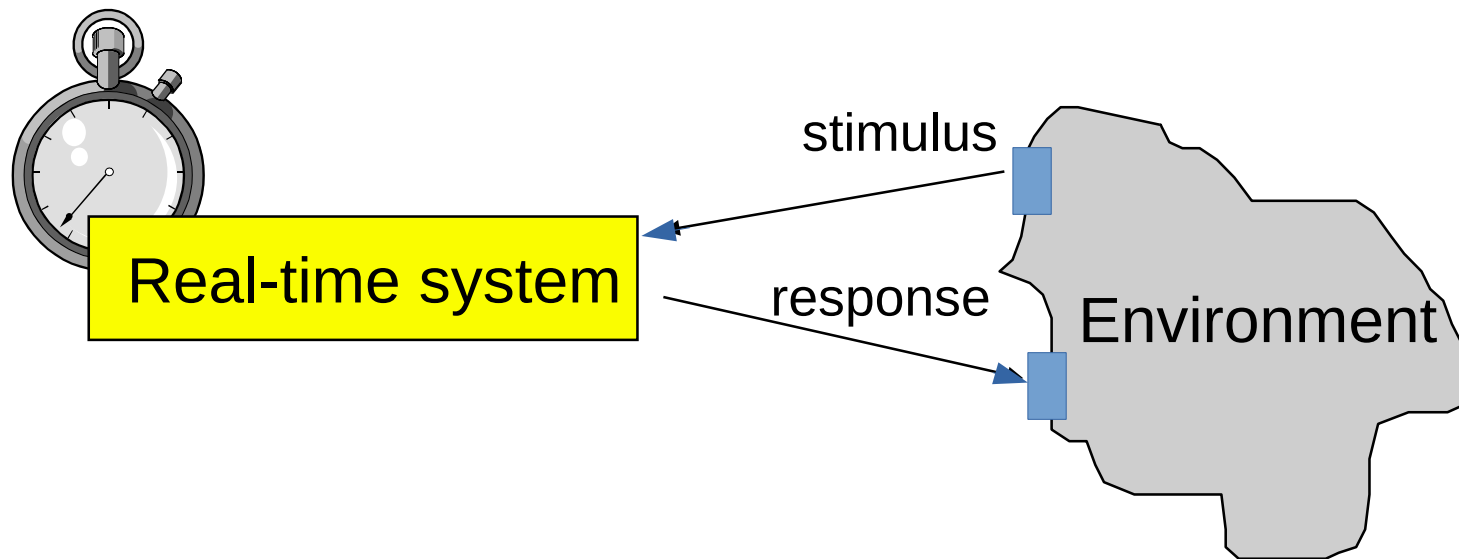
Examples:



- operating system (OS) managed processes, memory, network adapters, and other devices of a computer
- the power grid of Skövde is managed by dedicated programs that balance the power generation with the power consumption
- the management board of a company monitors its operation and intervenes by decisions

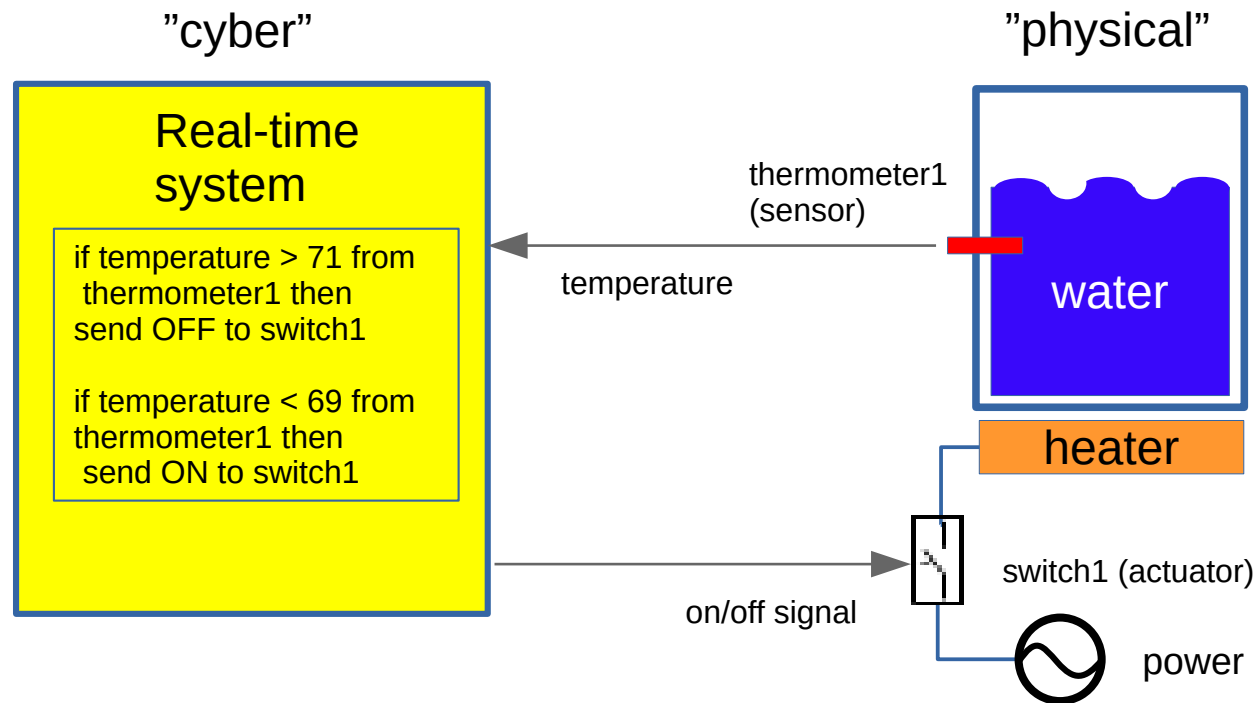
REAL-TIME SYSTEMS

"A managed system where actuator control react *sufficiently* quickly to react to sensor readings."



Examples: process control system
in an oil refinery, patient monitoring system

REAL-TIME SYSTEMS: EXAMPLE



Note that timing is important. We may need to read the temperature every second or just every minute depending on **how quickly it can change** and **how quickly we need to react**. Note that powering up the heater changes the temperature only with a time delay depending on the volume of the water and other parameters.

We may also need more sensors and actuator, e.g. to keep the water level in a certain range.

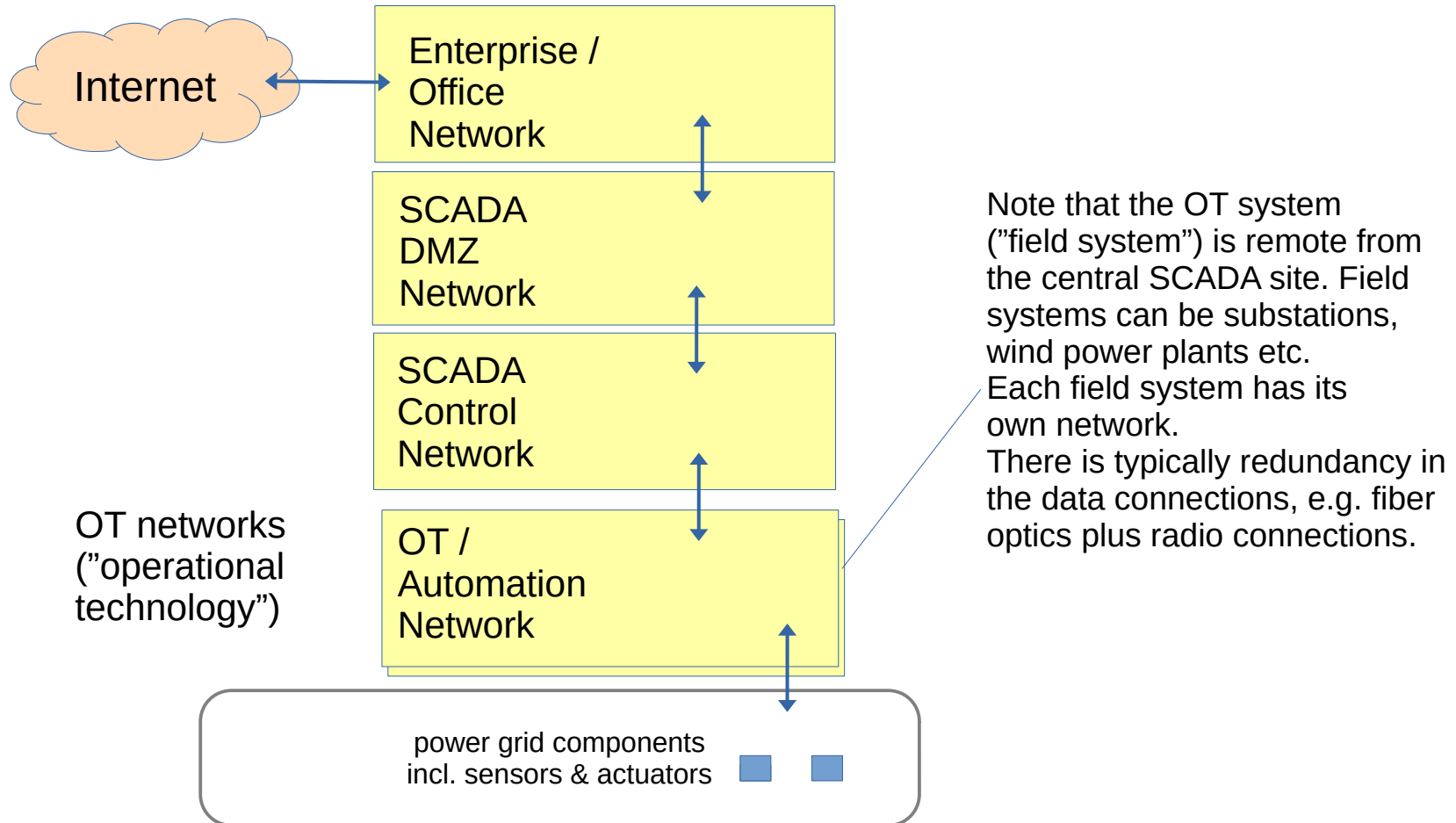
THE POWER GRID: TYPES OF COMPONENTS

Software	ERP, office, Scada control system,...	cyber
Information Technology (IT)	computer, server, laptop, smartphone,...	cyber-physical
Communication Network	WAN, LAN, Router, ...	cyber-physical
Operational Technology (OT)	RTU, MTU, IED,...	cyber-physical
Physical production and distribution	generator, power line, transformer, ...	physical

A municipal power company may manage ~100-1000 of power grid components such as transformers, generators, switches. In addition, each household has smart meters and possibly solar panels etc. Thus, there are thousands of sensors and actuators that need to be managed.

District heating, water supply, etc. are similar in nature to power grids. They are also about managing the distribution of matter/energy.

SUPERVISORY CONTROL AND DATA AQUISITION (SCADA)



This is a simplified figure. There is typically a network called DMZ ("demilitarized zone") that hosts some less critical servers such as the historian. Network components such as routers/switches/firewalls are left out here as well. Firewalls and other tools such as encryption&authentication and intrusion detection systems are used to mitigate risks.

THE CONTROL CENTER



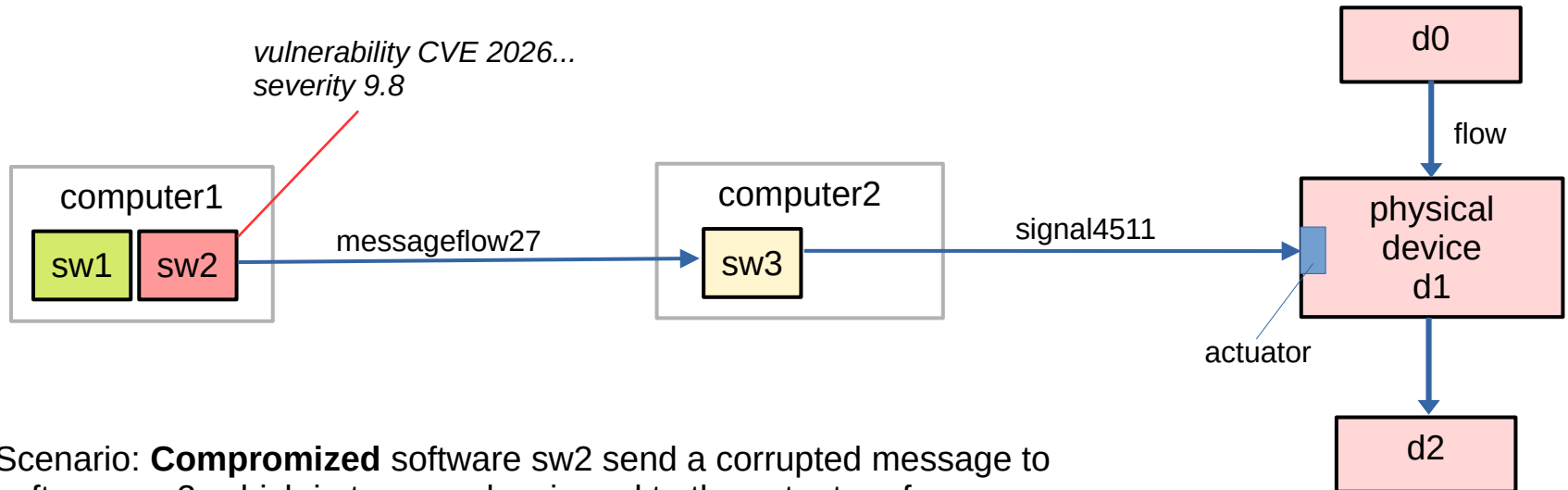
image: Wikipedia

The status of the physical process is displayed with diagrams. While the physical process runs largely with automated control, operators can intervene.

PROTECT THE PHYSICAL PROCESS

In order to **protect the assets**, we need to be able to identify them.

For example, if a certain software component sw2 has a vulnerability, we need to know where it is installed and what is its role in our process! A compromised component may have cascading effects.



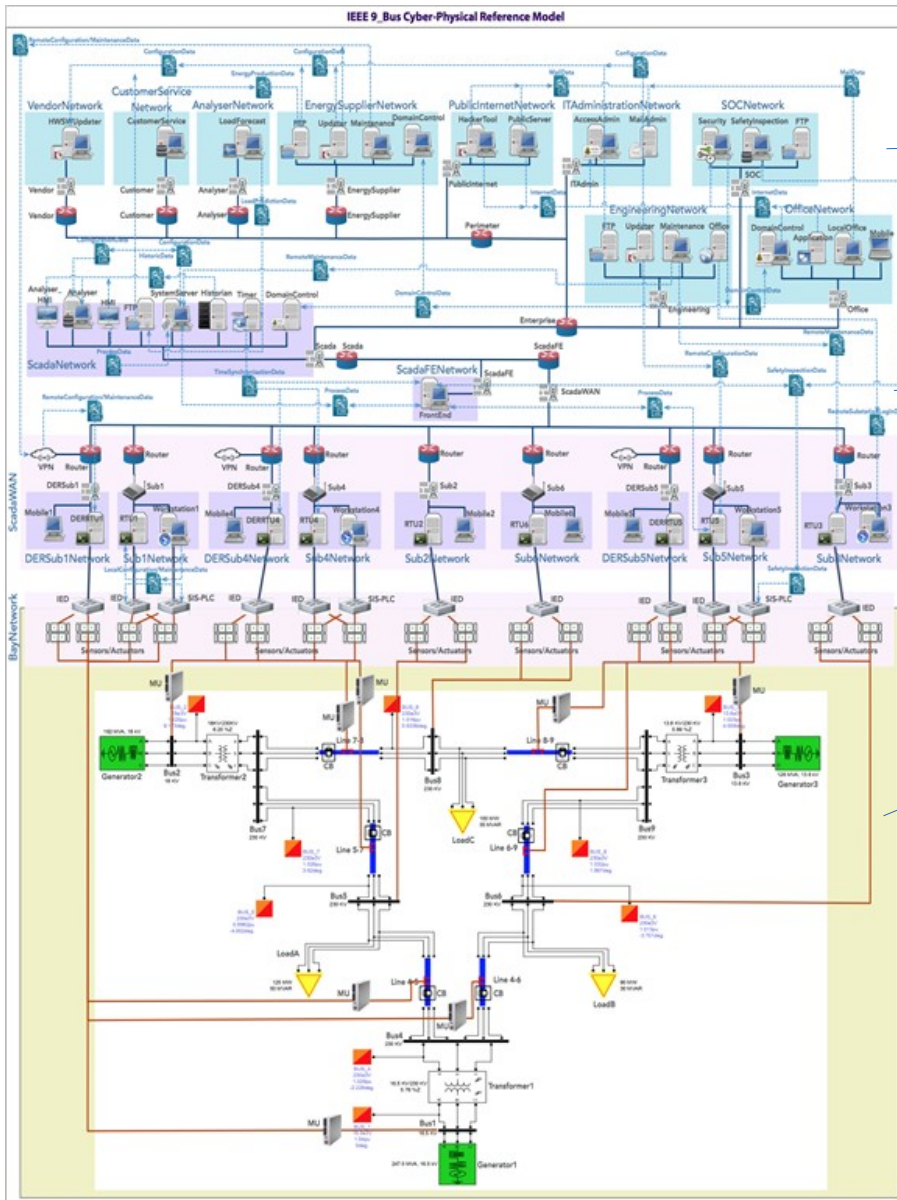
Scenario: **Compromized** software sw2 send a corrupted message to software sw3, which in turn send a signal to the actuator of a physical component d1, e.g. to turn it off.

This can then disrupt the **flow of matter/energy** in the physical process.

We call components like sw2 **critical** because they can affect the operation of physical components and consequently the physical flow.

There can be further cascading effects, e.g. on devices depending of d2.

CYBERPHYSICAL REFERENCE MODEL NINEBUSSC



Business networks, vendor networks, engineering networks, ...

SCADA network (computers plus their application programs)

control inside the 6 field systems
i.e. six networks at the power/substations
plus some test networks

Original NineBus physical system
(power grid with 3 generators,
3 substations, nine "busbars")

==> about 800 components plus
their connections

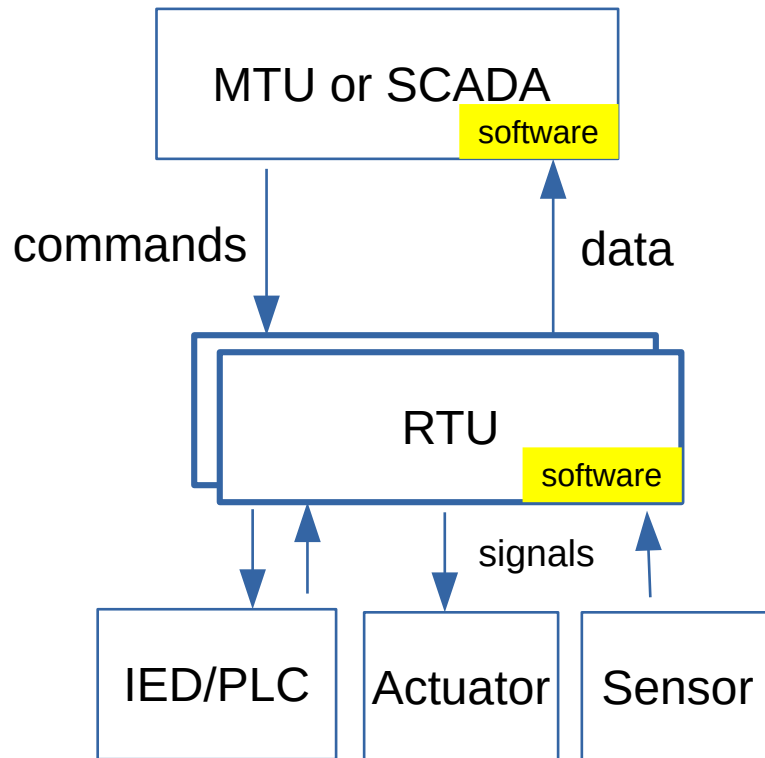
(developed by Yuning Jiang from IEEE NineBus)

THE ASSETS/COMPONENTS OF A CYBER-PHYSICAL SYSTEM

- **IT components** (computers, routers, data lines,...)
 - these are hardware components for information processing
- **OT components** ("operational technology")
 - hardware components to directly influence some physical processes such as energy production and distribution
- **Physical components**
 - these are the components that execute the physical processes such as energy production (e.g. generators, transformers, ...)
- **Cyber components**
 - software, operating system, data stored on a computer, data transfers, ...

RTU: REMOTE TERMINAL UNIT

An OT component that can receive signals from sensors and send signals to other OT components such as IEDs or directly to sensors/actuators attached to physical components. It receives commands from other components such as MTU (master terminal unit) and returns the (aggregated) sensor signals to the upper level. RTUs are designed for **communication** with a central control system, e.g. SCADA or MTU. Programming: C, C++...



*This is a essentially
a specialized
computer*



image: Wikipedia

MTU: MASTER TERMINAL UNIT

An OT component that manages several RTUs or IEDs or PLCs. It can serve as point of access to control all physical components of a substation (or an area of a factory,...). It can perform certain **aggregations** of data received from RTUs and can also filter out some data errors (e.g. impossible values).

Note that while an MTU is a hardware component, it does contains software to process the data and command streams from and to it.

While there are MTUs that communicate via Internet protocols such as TCP/IP, they traditionally use specialized communication protocols such as MODBUS.

PLC: PROGRAMMABLE LOGIC CONTROLLER

A specialized OT computer that runs programs to automate certain tasks such as keeping the temperature in a container constant using actuators and based on sensor reading.

PLCs have no dedicated operating system and the programming language is very simple.

Fast reaction times to allow managing real-time processes. Can operate under harsh conditions such as **factories**.

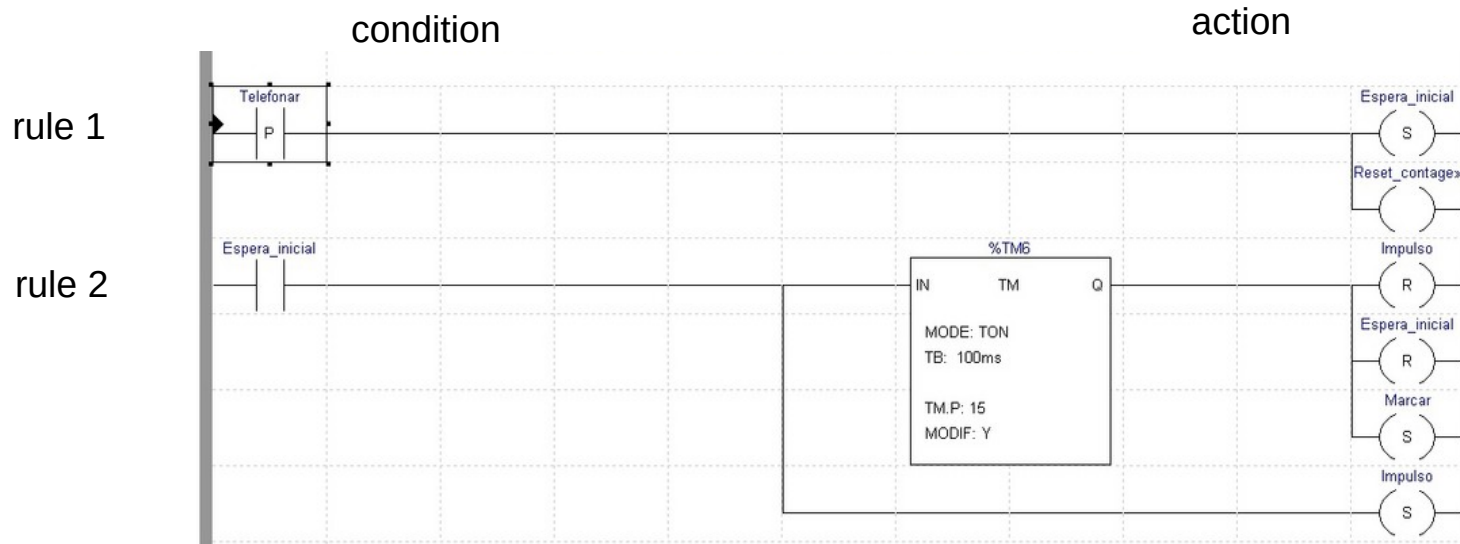
Predecessors were relays, e.g. relays that react on temperature.

Programming: "ladder logic"; customer provides the programming

Protocols: Modbus, Profinet, Internet protocol (TCP/IP), ...

LADDER LOGIC IN PLCs

PLCs contain software to manage the physical system at the micro-level, e.g. to keep the water level in a tank between defined low and high point.



source: [Wikipedia](#)

"Ladder logic": a simple programming language to manage sensor readings and to send signals to actuators.

Each rule is in principle able to "fire" independently of the others.

The PLC has an infinite loop where one if executed after the other in sequence.

IED: INTELLIGENT ELECTRONIC DEVICE

An OT component mainly used in the electric power industry to control power grid components such as circuit breakers and transformers.

They are connected to sensors and actuators and can perform certain actuations automatically, such as activating a circuit breaker when the voltage is too high.

There is a trend to make these devices more and more intelligent to automate whole substations.

The software running on IEDs is called "firmware" by manufacturers. The firmware can be updated, e.g. to fix bugs or add capabilities.

IEDs are frequently used in power grid control. Very fast reaction times. Specialized protocols such as IEC 61850, DNP3, ...

Programming: vendor specific

SBC: SINGLE-BOARD COMPUTER

A fully equipped computer system with a standard operating system such as Linux on a single board. Devices like a Raspberry Pi are now also used in the industry for control tasks.

GPIO (general purpose input/output) pins allow to read sensor data and to signal actuators. Hence can serve as OT component.

Cannot rival with IEDs/PLCs on speed and may be more vulnerable to cyber attacks.

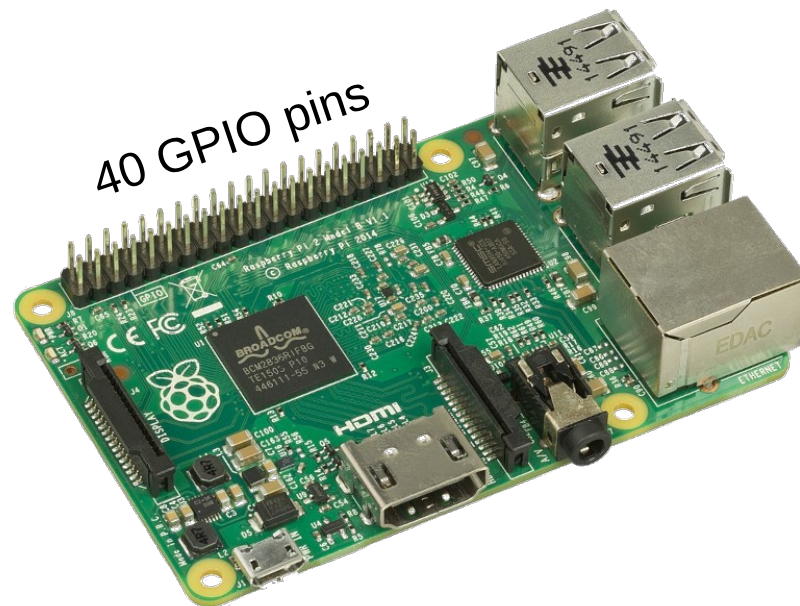


image: Wikipedia

HMI: HUMAN-MACHINE INTERFACE

A control panel that allows human operators to interact with PLCs, RTUs and IEDs.

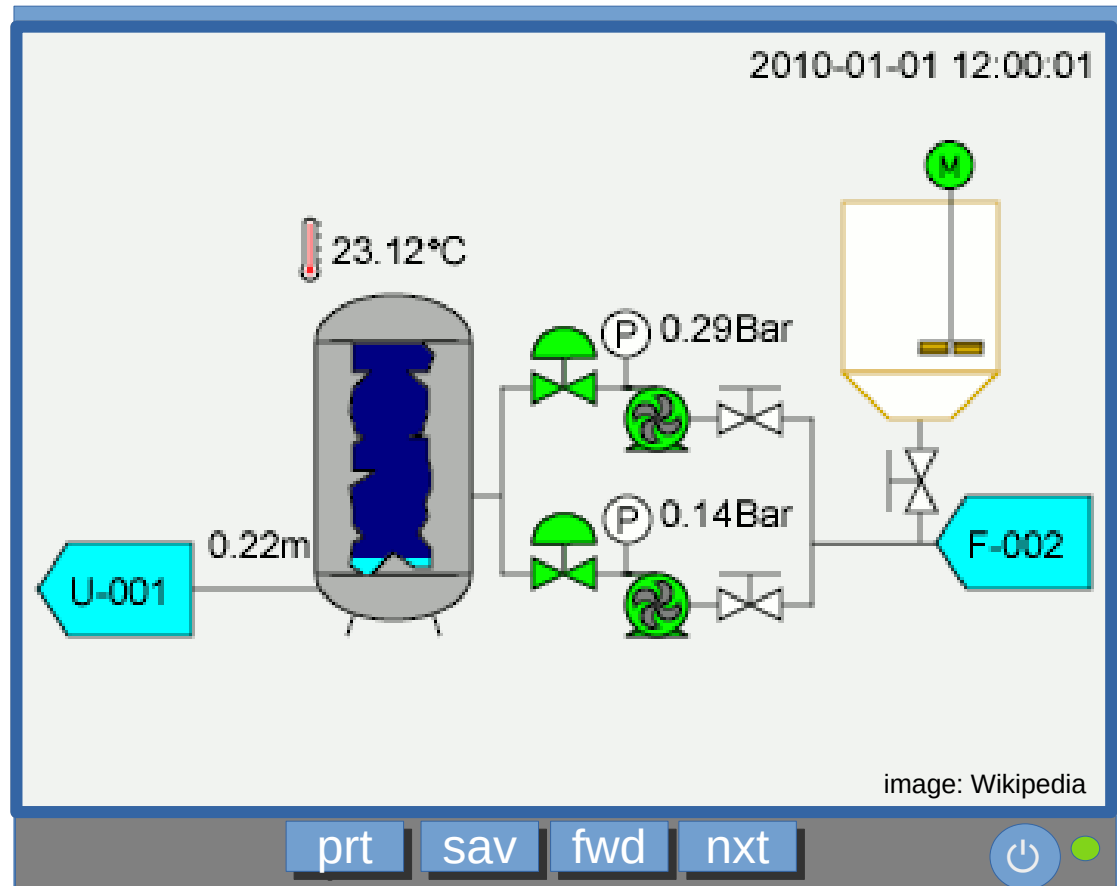
By this, the operators can for example send a command to an RTU to activate a circuit breaker, or to check and change the data points where IEDs react to sensor signals.

HMIs can visualize data graphically so that operators can quickly assess the state of the physical process.

HMIs are operated by software. They can be located in substations but also in control centers to intervene directly in the physical process.

Via VPN channels, HMIs can also run on remote computers and tablets.

HMI VISUALIZATION



The operator sees the state of the physical system in a diagrammatic way on a monitor.

SUPERVISORY WORKSTATIONS

A computer dedicated to collect information from the physical system (e.g. the power grid) and to visualize it for the human operator.

Normally, a supervisory station is a "read-only" device, i.e. it only monitors the physical processes. It may also be regarded as a HMI in read-only mode.

To display historical data (how the state of the physical system was in the past), it can be linked to a data historian.

Note that the function of a supervisory workstation is realized by software running on the supervisory workstation.

Due to the read-only mode, a supervisory workstation is typically in the so-called SCADA DMZ network (de-militarized zone) or even in the business network.

DATA HISTORIAN

A computer for managing a database of time-tagged data points about the physical system, e.g. storing the voltage at various locations in the powergrid.

It may also contain production targets and similar human-generated data points.

The information in the data historian is needed for various purposes, in particular for forecasting changes in the physical system like imminent increases in the loads in a power grid. The SCADA system may need these forecasts to send appropriate commands to power generators sufficiently early to cope with an increased load.

The function is realized by software. The data historian needs appropriate protection due to its role in managing the physical system.

Data historian systems have interfaces to import data from various sources.

BUSINESS INFORMATION CONSOLES & DASHBOARDS

These can be seen as extensions of supervisory workstations by providing more analytical functions. They can be used for example for trading the output of the physical system (e.g. the electric power).

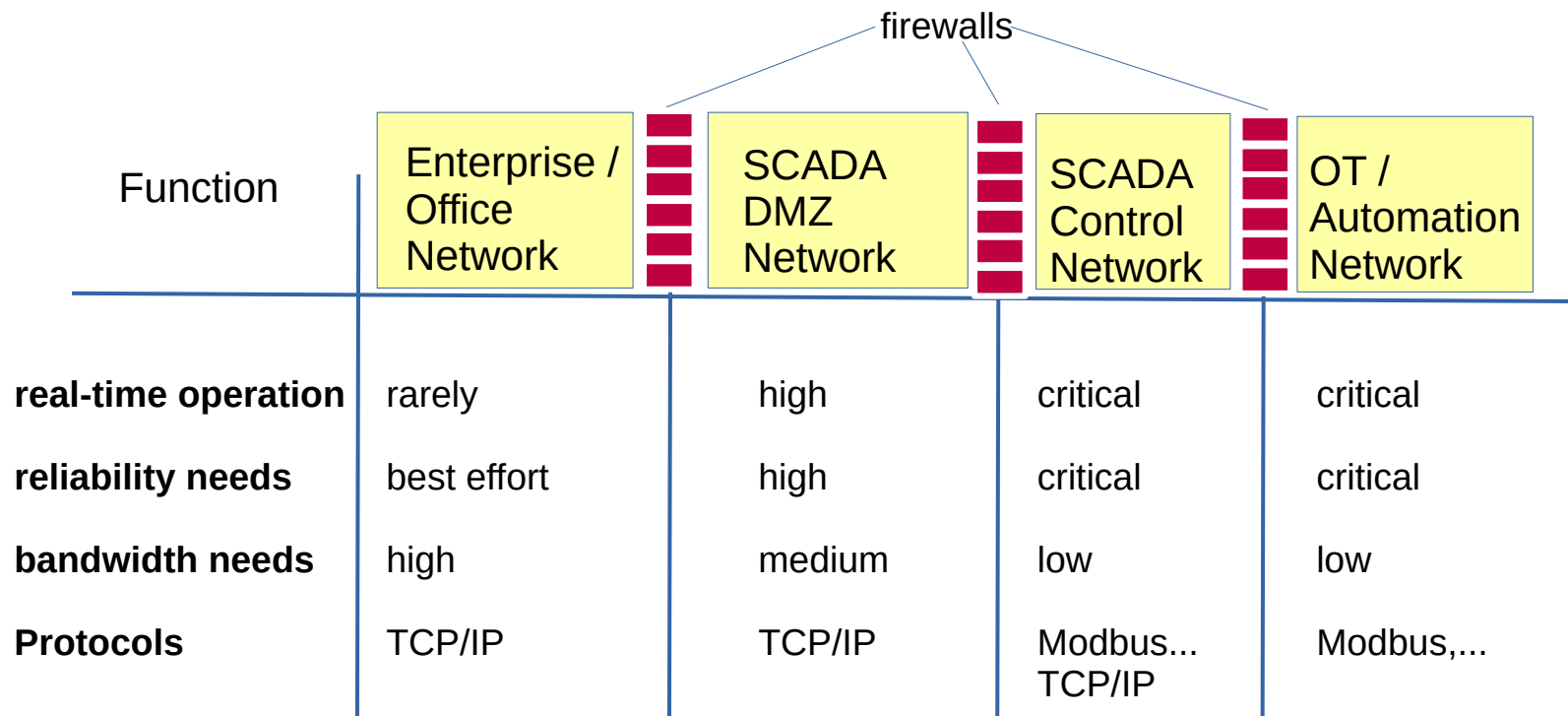
The computer is typically located in the SCADA DMZ, though the display may be directed to offices outside the network using secure connections.

Sometimes, the function is realized by exporting data from the data historian and processing it outside of the DMZ, e.g. as a spreadsheet. In the case of the power grid, the traders of a power grid company need forecasts of the power consumption in the next year to make deals with power providers.

Note the potential security risk with creating such copies!

NETWORK CONSIDERATIONS

The communication networks to link the IT and OT components may have different architectures (ring, bus, star,...) and different parts of the network use different communication protocols (TCP/IP, Modbus, Profibus, DNP3, ...). Note the geographic distribution of the physical system (substations,...). The closer to the physical system, the higher the demands on security and reliability.



SOME NETWORK PROTOCOLS IN OT

Protocol	Primary Use	Strengths	Weaknesses
TCP/IP	Standard transport layer backbone for OT/IT network integration.	High throughput, scalable, universal, standardized	Unencrypted, unauthenticated, spoofing, eavesdropping
Modbus (serial) Modbus over TCP/IP	Simple controller-to-device and sensor communication. Widely used in RTUs (serial lines). Modbus over TCP/IP wraps Modbus.	Simple, vendor-neutral, lightweight, ubiquitous	Cleartext, unauthenticated, unverified, hijackable Very old but still widely used
OPC UA	Cross-platform IT/OT data integration, SCADA-to-cloud/HMI.	Encrypted, authenticated, object-oriented, cross-platform	Heavy overhead, implementation-dependent, complex management
DNP3	Utility SCADA (electric grid, water, oil & gas wide-area telemetry).	Resilient links, timestamped, event-buffered, unsolicited-reporting Can use secure authentication (DNP3-SA). Encryption via TLS.	Cleartext by default, legacy unauthenticated, spoofing prone

source: Google, Gemini

NETWORKING COMPONENTS

These include routers (directing incoming data packages to the next router or IT component), firewalls (blocking network traffic that does not fulfill the specified rules of allowed data packages), modems, switches, and data lines (ethernet cables, fiber cables, wireless connections,...).

Some networking components are combining functions of separate individual networking components, e.g. a firewall included in a router.

And sometimes the networking component is implemented in software only or virtualized as a virtual device on a hypervisor, e.g. "virtual routers".

POWER GRID COMPONENTS

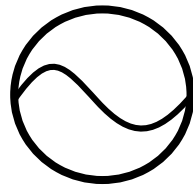
In this course, we use the power grid as example infrastructure. The powergrid has a central position in a country's economy. In particular, other infrastructures (water supply, food supply, telecommunications, transportations,..) depend on the power grid.

About 50% of all cyber attacks on infrastructures currently target the powergrid.

Subsequently, we discuss the major components of a power grid as a physical system.

From a control perspective, the power grid needs at all times produce (generators) the exact same electric energy that is consumed (loads).

GENERATOR



A component that produces electric energy, e.g. a nuclear power plant, a wind energy turbine, a water power plant.

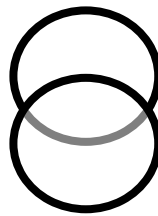
A generator usually produces AC current with a fixed frequency such as 50Hz.

The frequency must be kept close to 50Hz throughout the power grid.
Deviations indicate an imbalance between the generated and the consumed power.



images: Wikipedia

TRANSFORMER



A component that transforms an input voltage to an output voltage.

We distinguish step-up transformers (increase the voltage) and step-down transformers (decrease the voltage). A typical voltage of a generator is 16500 Volt. This is usually stepped-up to a very high voltage (400.000 Volt) to transmit the power.

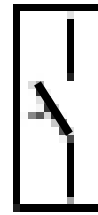
For distributing the power, this high voltage need to be stepped down (e.g. to 110000 Volt) and then for consumers to for example 380 Volts. This is why you see big transformers in sub-stations.

Note that such transformers need to be cooled and constantly monitored!



image: Wikipedia

CIRCUIT BREAKER



A component to close or break a power connection.

Some circuit breakers are operated only manually, others are controlled locally (e.g. by IEDs), and others are controlled by the SCADA system.

Consumer homes also have circuit breakers to manually or automatically cut off the power, e.g. when there is a short circuit.

If the voltage is very high, the circuit breakers are very complex systems that use certain gases (Argon) to avoid that electricity causes flashes when the breakers are activated. Note that the gas pressure needs to be monitored and controlled!



low
power
circuit
breaker



high
power
circuit
breaker

images: Wikipedia

A bar of highly conductive material (e.g. copper), to which various power grid components can be connected using power connections (also consisting of highly conductive material).

Busbars are very common in substations and power stations to connect generators, transformers and power lines (transmission or distribution).



image: Wikipedia

POWERLINE



A set of highly conductive cables that connect power stations and substations, and substations with consumer end points.

A powerline cable has a certain diameter and length. This, together with the material properties determine how much energy is lost by heat dissipation when power flows through the line. If too much power flows through the line, it has to be shut down via circuit breakers or fuses to prevent that it overheats and then trips.

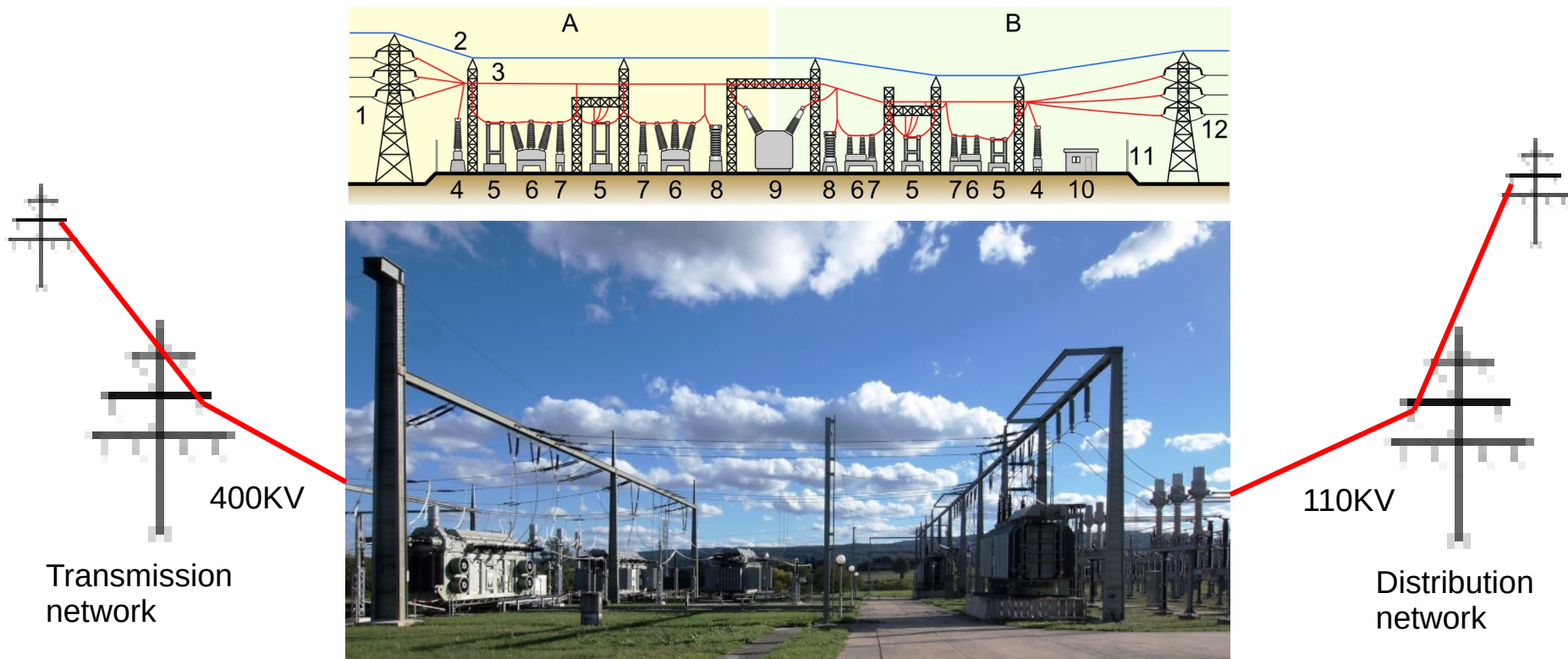
The power grid shall be designed such that it can cope with high loads and occasional failures of power lines.



image: Wikipedia

SUBSTATION

An area or building with incoming and outgoing powerlines, transformers, busbars, circuit breakers etc with the purpose to **transform** the voltage of the incoming powerlines to the voltage of outgoing powerlines.



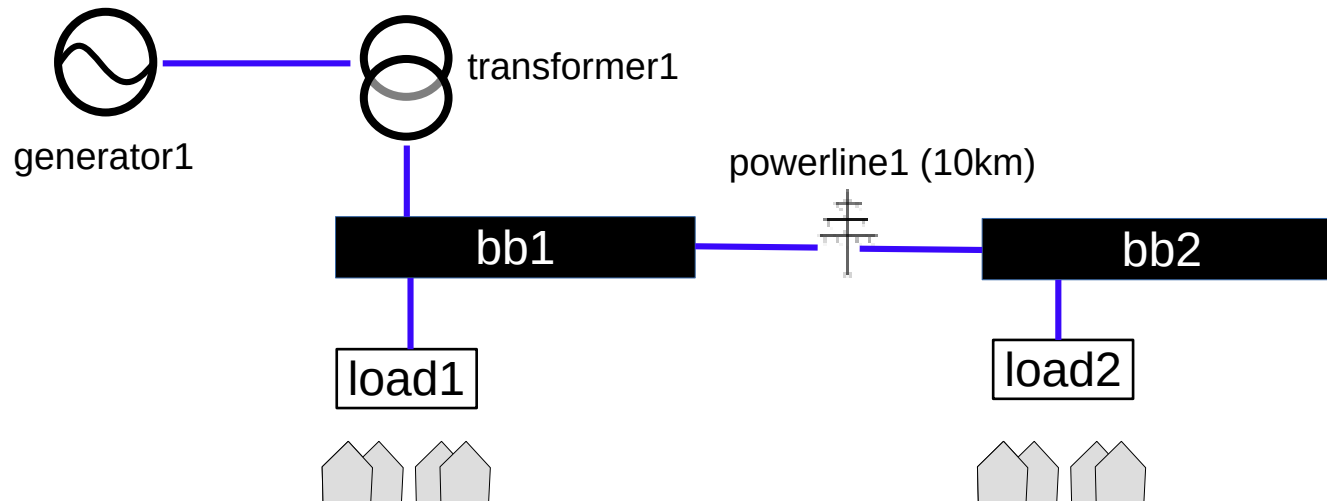
images: Wikipedia

LOAD

An abstract component standing for a set of consumer end points that are receiving electric power.

To manage the power grid, loads can also be created by large batteries or water pump plants (e.g. in Norway, Switzerland, Austria).

If there is too much energy generated, these loads can consume the excess energy.



TAXONOMY OF COMPONENTS

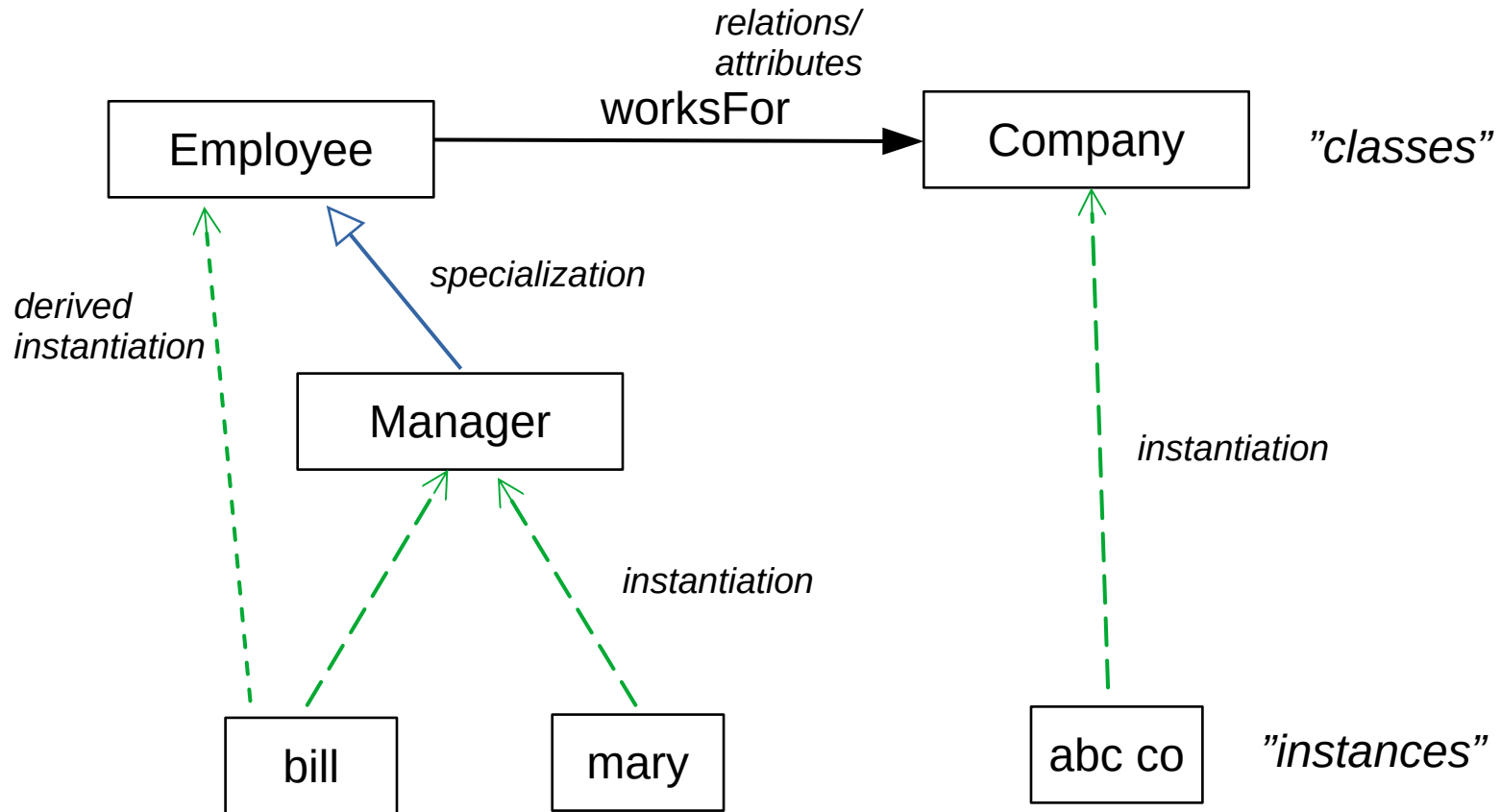
Subsequently, we present an ontology/taxonomy of components and their relations to describe cyber-physical systems.

We shall use the ConceptBase.cc system to describe the ontology. It allows to represent the **component types (or classes)** and also their examples.

The advantage of this approach is that we can define (and **extend**) the set of allowed component types and at the same time we can define example cyber-physical systems in terms of the ontology!

The taxonomy was developed in the ELVIRA project (HIS + Combitech AB).

THE TELOS LANGUAGE OF CONCEPTBASE



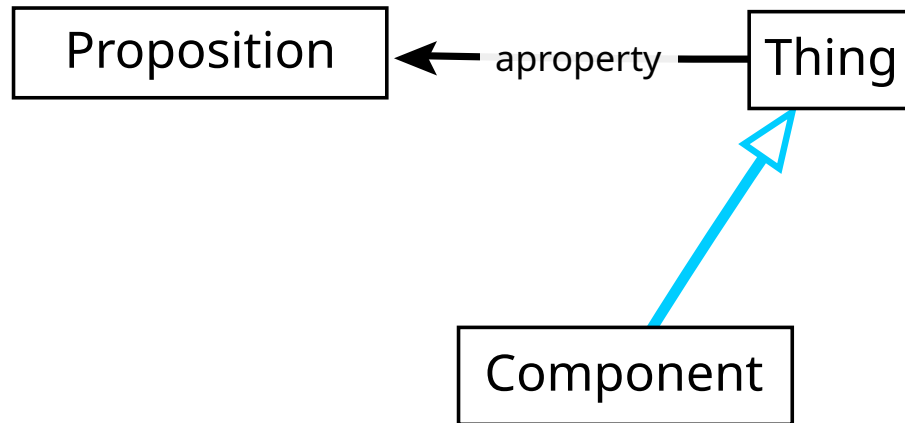
[employee-example.gel]

In additions, Telos offers deductive rules, integrity constraints, queries and functions. See also the tutorials on Canvas and documentation at <https://conceptbase.sourceforge.net/>

NOTES ON THE SUBSEQUENT GRAPHS

- The subsequent graphs are also made available as so-called graph files (fig*.gel) via CANVAS. You can directly start them by the command
 `cbgraph figX.gel`
under Linux. Also check
 <http://conceptbase.sourceforge.net/CB-Mime.html>
- Graph files (*.gel) contain all definitions ("Telos frames"). The tutorial web page at <https://conceptbase.sourceforge.net/elvira-edoc2026/GELs/> contains the plenty of example graph files for this tutorial. Start them from the web browser or download them to a local directory and start them via a terminal comment like
 `cbgraph figX.gel`

TOP-LEVEL OF THE ONTOLOGY



[[fig1.gel](#)]

See directory subdirectory GELs; you can start them with ConceptBase CBGraph to inspect the definitions.

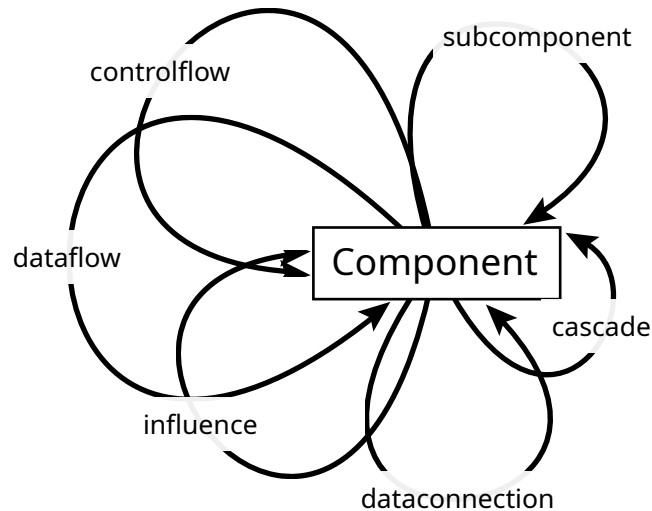
```
Thing with
  attribute
    aproperty: Proposition
end

Component isA Thing end
```

Telos frames of object/class definitions. 'Proposition' is a pre-defined class that has all objects/classes as instances. In Telos, classes are also objects. Even explicit instantiations, specializations, attributes and relations are objects. Values like 123 are also objects in Telos!

'Thing' is the most generic concept of the ontology defining that it can have describing properties, e.g. nominal voltage or serial number of a transformer. 'Component' is a subclass of Thing. We shall use it to describe cyber-physical systems.

RELATIONS OF COMPONENT

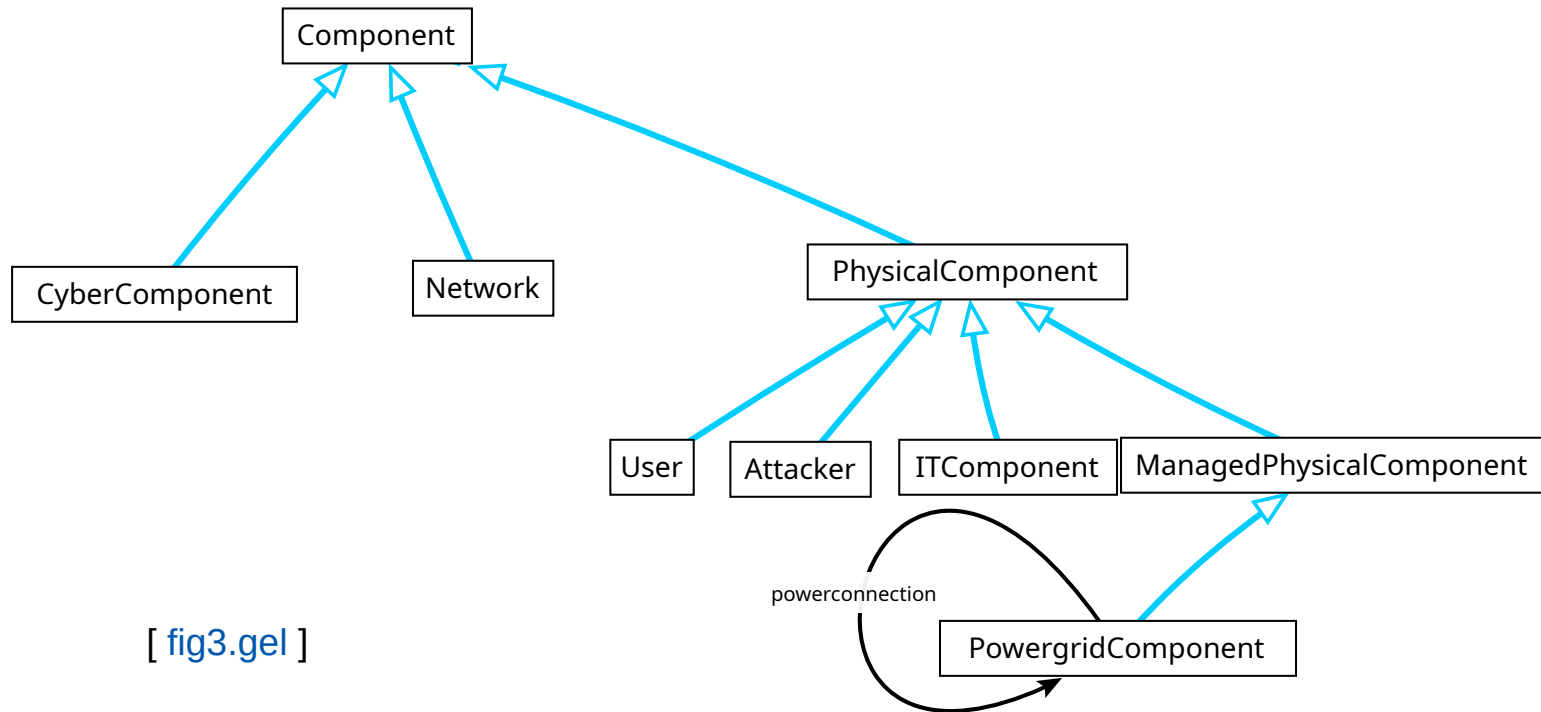


[[fig2.gel](#)]

There can be data and control flows between components.
Components can influence each other.
Failures of components can cascade to other components.
Components can have subcomponents.
Components can be connected by data connections (such as wires).

Lab1: Only uses subcomponent and dataconnection.

SUBCLASSES OF COMPONENT

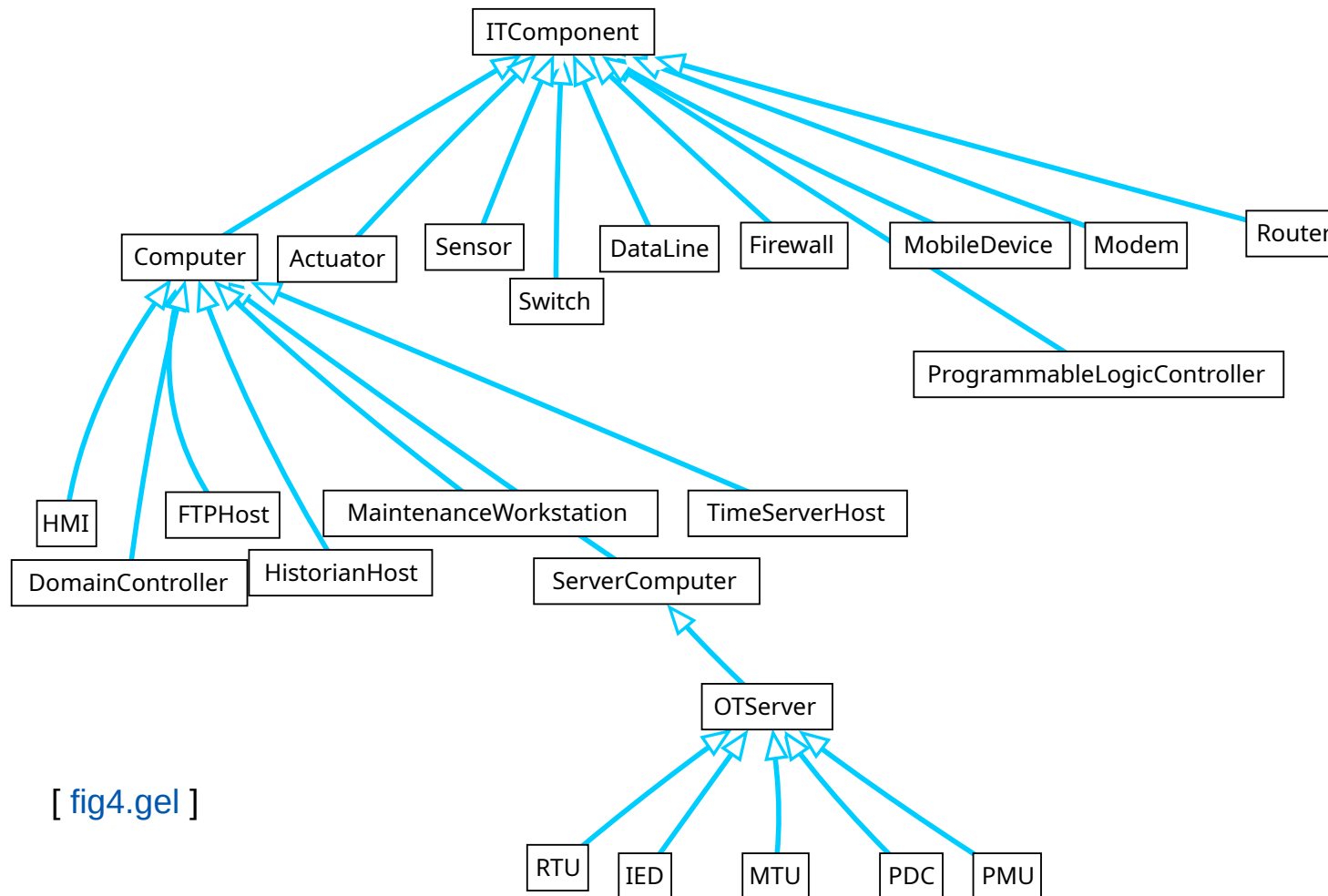


Physical components are further sub-divided. Powergrid components are special managed physical components.

Note that powergrid components can have powerconnections between them.

Next we look at the ITComponents ...

SUBCLASSES OF ITCOMPONENT



[fig4.gel]

This hierarchy is extensible and may not list all known types of IT components. Note that we regard IT components as hardware, i.e. they are physical objects.

IT COMPONENTS FOR NETWORKS

Router: a networking component that forwards data packets in a computer network. A router has data lines connecting it to routers of other networks. The packets are directed to neighbor routers based on their "address".



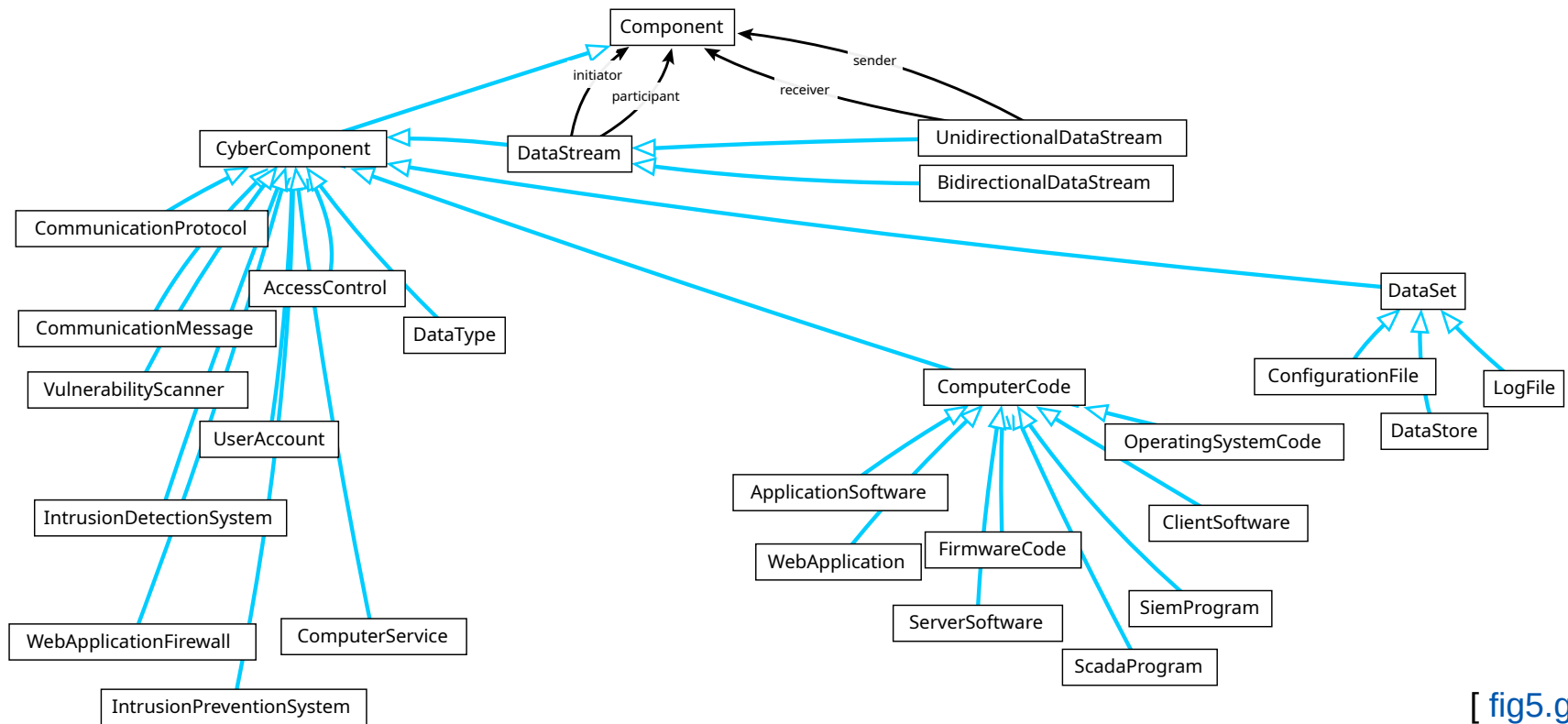
Switch (network switch): a component that connects several IT components via data lines.



Firewall: security components for networking that monitors incoming and outgoing data packets.



CYBERCOMPONENT



[[fig5.gel](#)]

- Note the definition of (logical) data streams. They go from components to other components, e.g. from a SCADA system to a program running on an RTU.
- The hierarchy of cyber components can be extended at any time.

FROM ONTOLOGY TO NETWORK MODELS

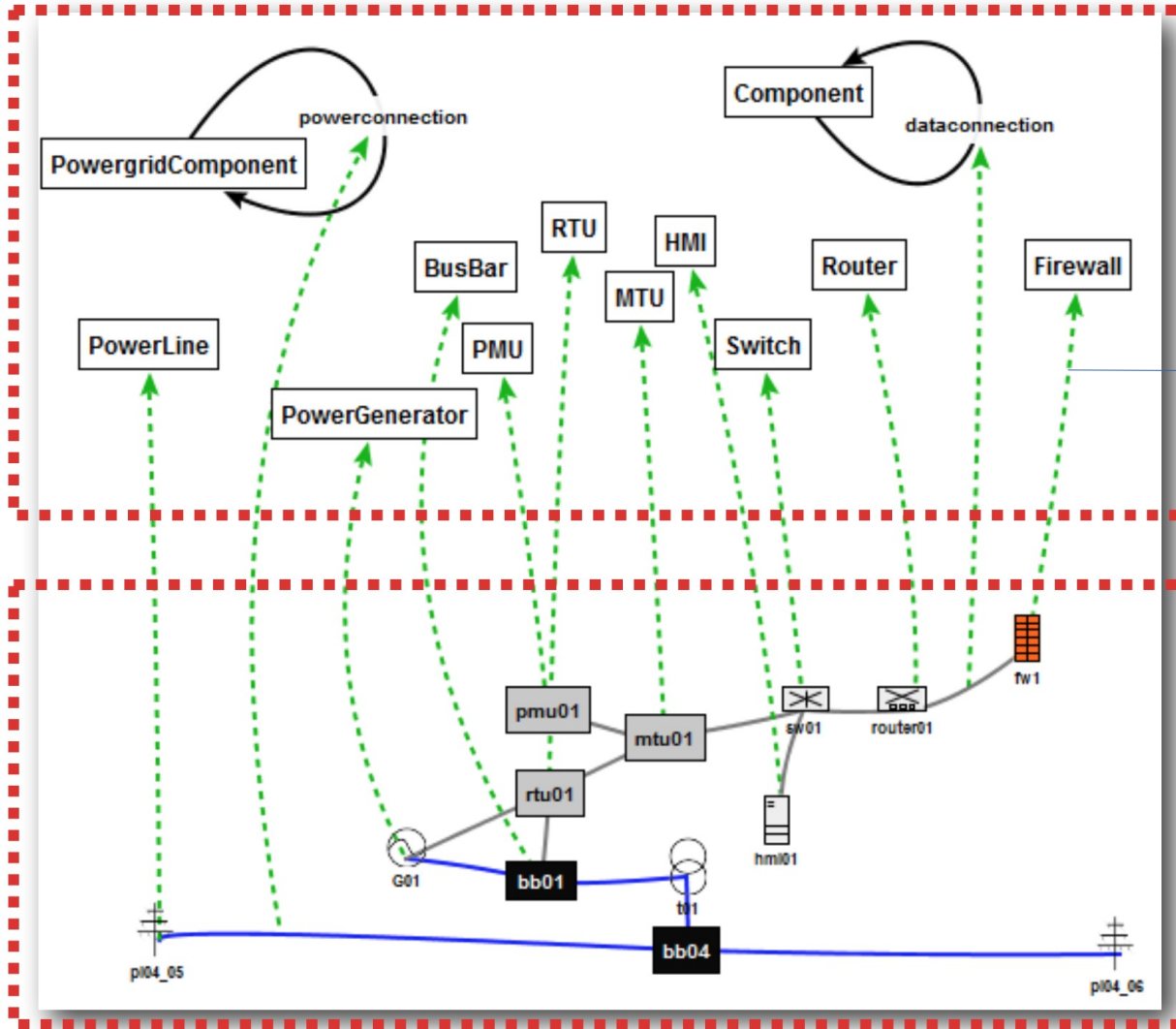
So far, we defined classes of CPS components and their relationships in a module called "ElviraTaxonomy".

In ConceptBase, we can instantiate the classes to model actual CPS networks, e.g. a power grid together with the IT/OT components and the software that manages the power grid.

It is useful to create a sub-module for the network models, so that we can have several of them in parallel.

In the lecture, we use the pre-defined network model NineBusSC.

DEFINING POWERGRIDS



Taxonomy

"=classes"

Instantiation

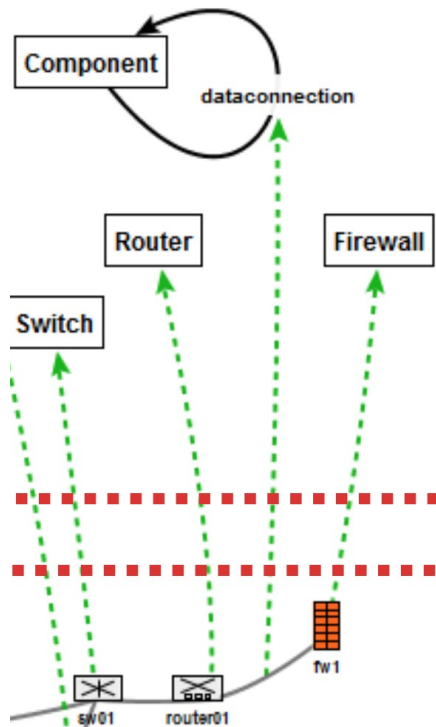
Physical
Components

"=instances"



DEFINING POWERGRIDS

Textual definition in ConceptBase



```
fw1 in Firewall with  
  dataconnection  
    con1: router01  
end  
  
router01 in Router  
end  
  
...
```

The textual definitions are made with the 'cbiva' tool of ConceptBase.

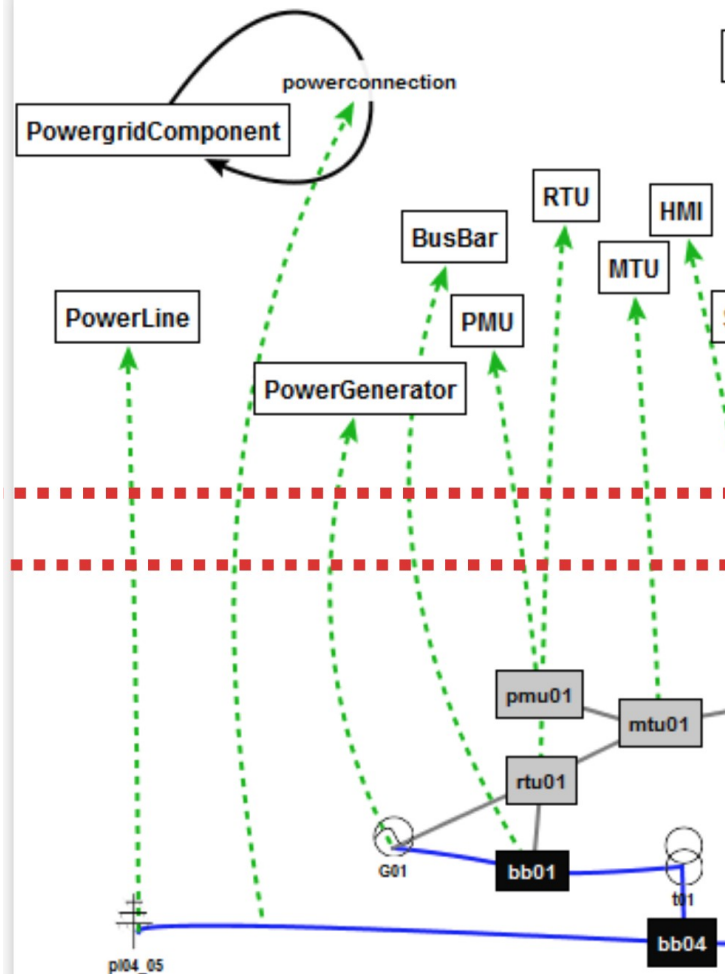
You can later visualize them with the tool 'cbgraph' of ConceptBase.

Keep the textual definitions in plain text files on your local computer!

The keyword "in" means "instance of"!

DEFINING POWERGRIDS

Textual definition in ConceptBase
(excerpt in Telos)



```

bb04 in BusBar with
  powerconnection
    con1: pl04_05;
    con2: t01
end

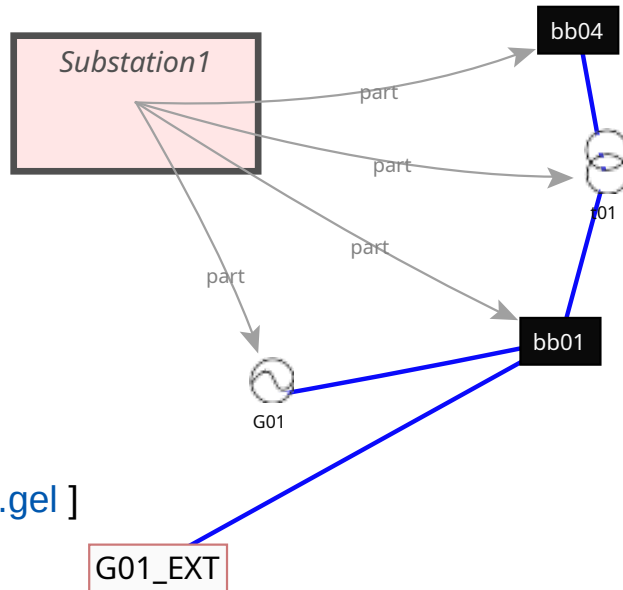
pl04_05 in PowerLine end

t01 in StepUpTransformer with
  powerconnection
    con1: bb01
end
...

rtu01 in RTU with
  dataconnection
    con1: bb01;
    con2: G01
end

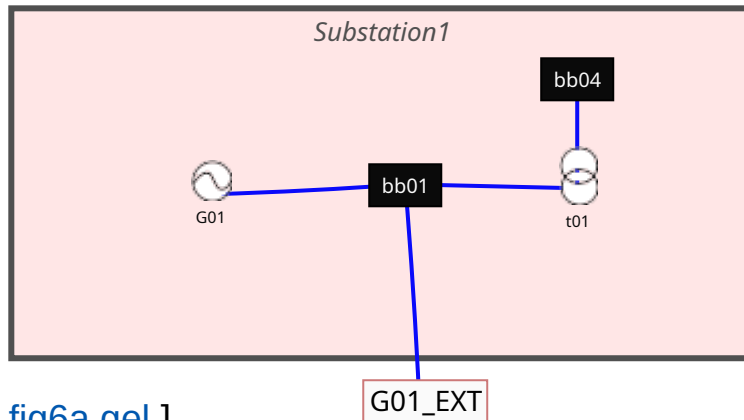
...
    
```

DEFINING SUBSTATIONS



[fig6.gel]

```
Substation1 in PowerStation with
subcomponent
  c1 : G01;
  c2 : bb01;
  c3 : t01;
  c4 : bb04
end
```

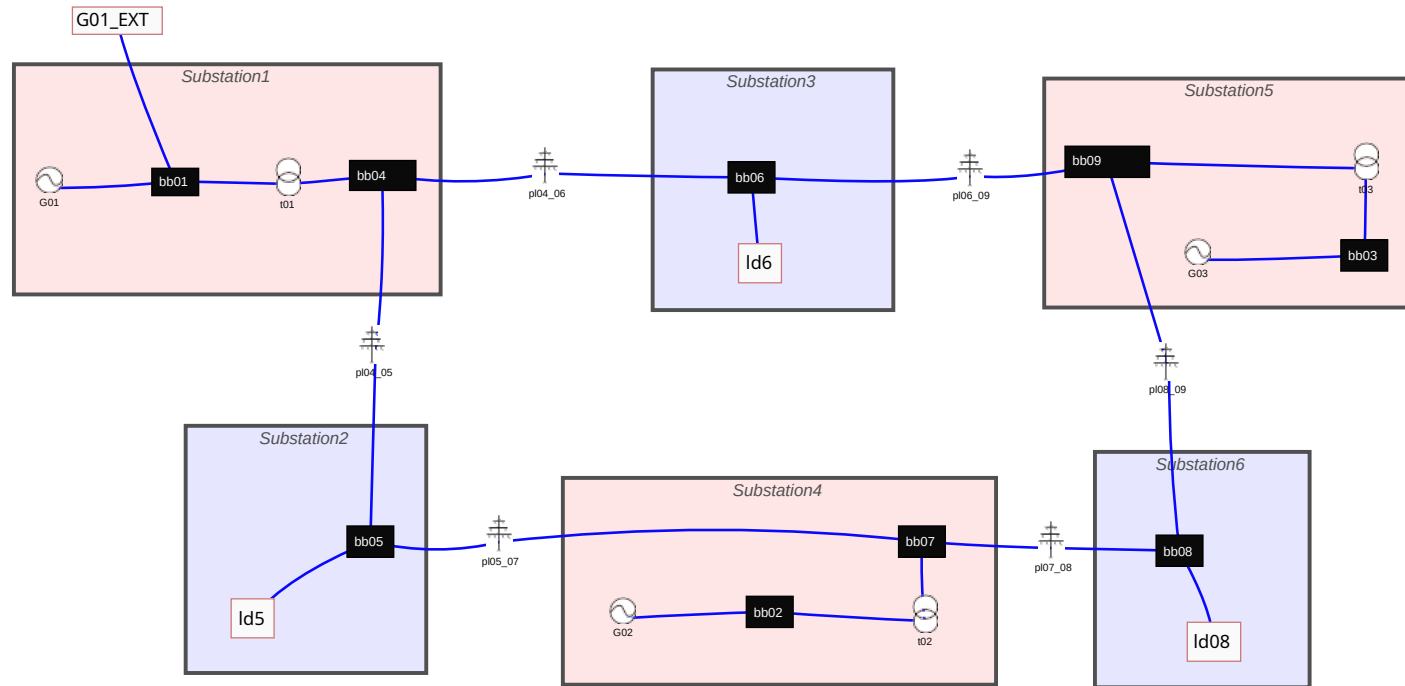


[fig6a.gel]

Preferred visualization of 'subcomponent' shows them inside the Substation1 rectangle.

G01_EXT is the grid outside of the substation, e.g. the regional powergrid feeding into the municipal powergrid at Substation1.

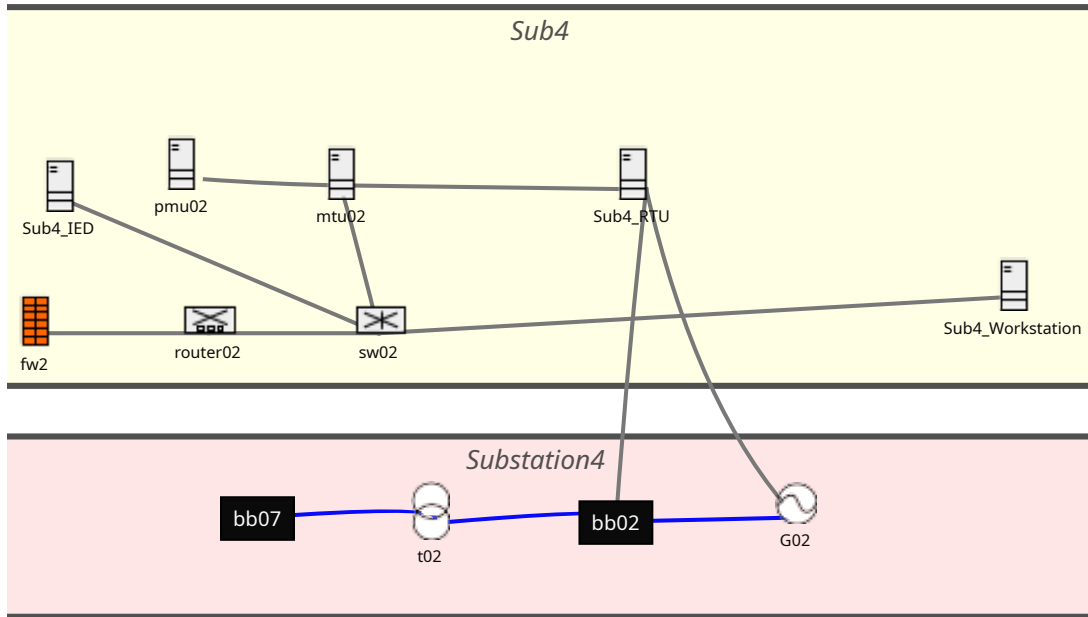
THE IEEE NINEBUS MODEL



[[fig11.gel](#)]

This rather small powergrid model has 3 power stations (with generators), 3 substations (with loads), and powerlines connecting the 6 stations. The model is named after the number of busbars. The powergrid components have electrical properties stored with the model. **G01_EXT** represents the connection to the external powergrid.

DEFINING NETWORKS



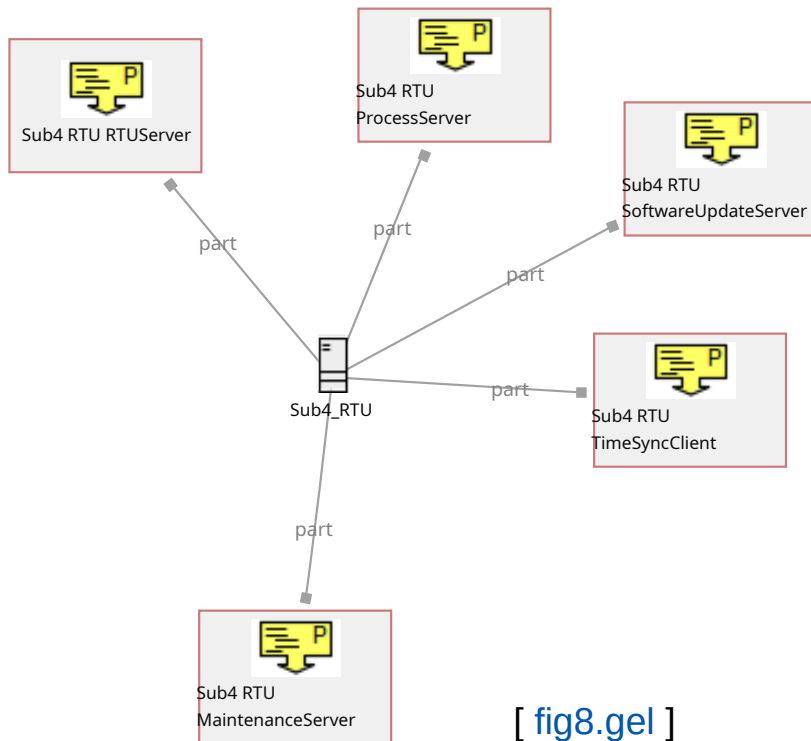
```
Sub4 in Network with
subcomponent
  sub1 : Sub4_RTU;
  sub2 :
    Sub4_Workstation;
  sub3 : Sub4_IED;
  ...
end
```

[[fig7.gel](#)]

The 'subcomponent' relation is used in all situations where a bigger component includes part components:

- substations contain powergrid components
- networks contain IT and OT components
- control centers contain servers, HMIs and other IT components
- computers contain operation systems and application software
- ...

DEFINING SOFTWARE IN IT/OT



[fig8.gel]

Sub4_RTU in RTU with
subcomponent

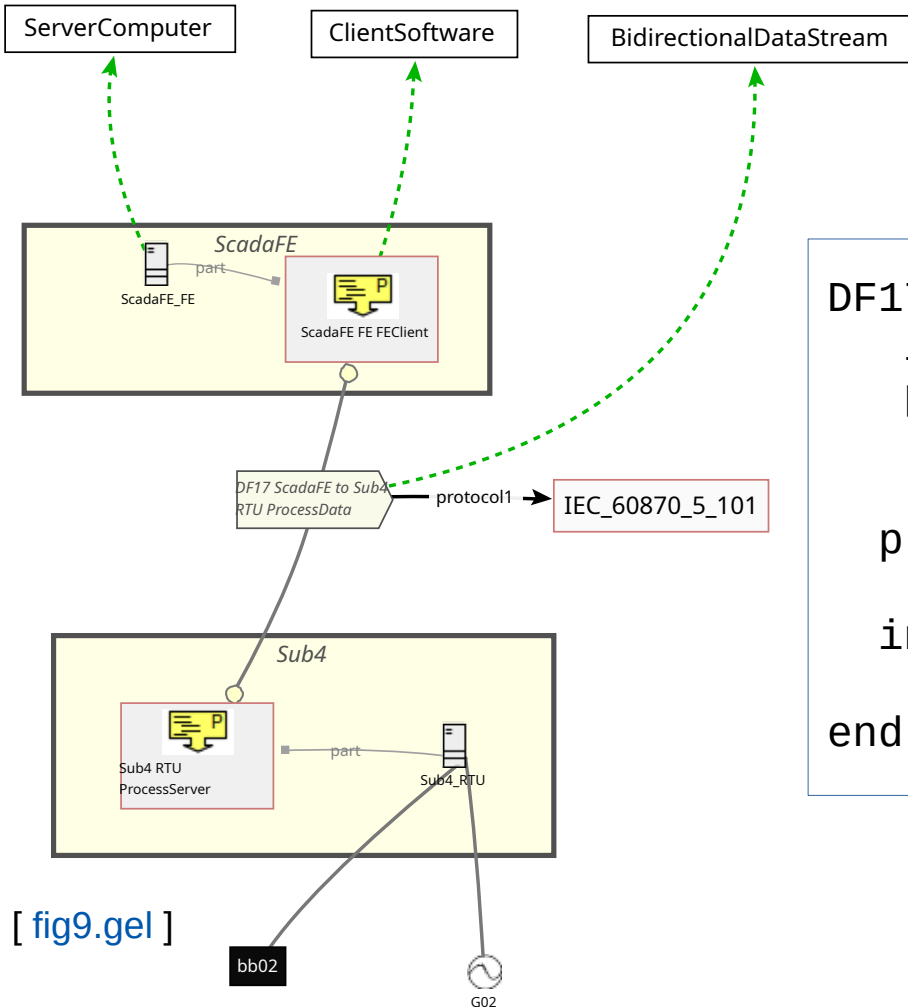
```
sub1 : Sub4_RTU_MaintenanceServer;  
sub2 : Sub4_RTU_RTU Server;  
sub3 : Sub4_RTU_ProcessServer;  
sub4 : Sub4_RTU_SoftwareUpdateServer;  
sub5 : Sub4_RTU_TimeSyncClient
```

end

This is an extensive description of Sub4_RTU. The subcomponents are all software.

Of particular interest is Sub4_RTU_ProcessServer. It receives commands from a SCADA system to control powergrid components.

DATASTREAMS



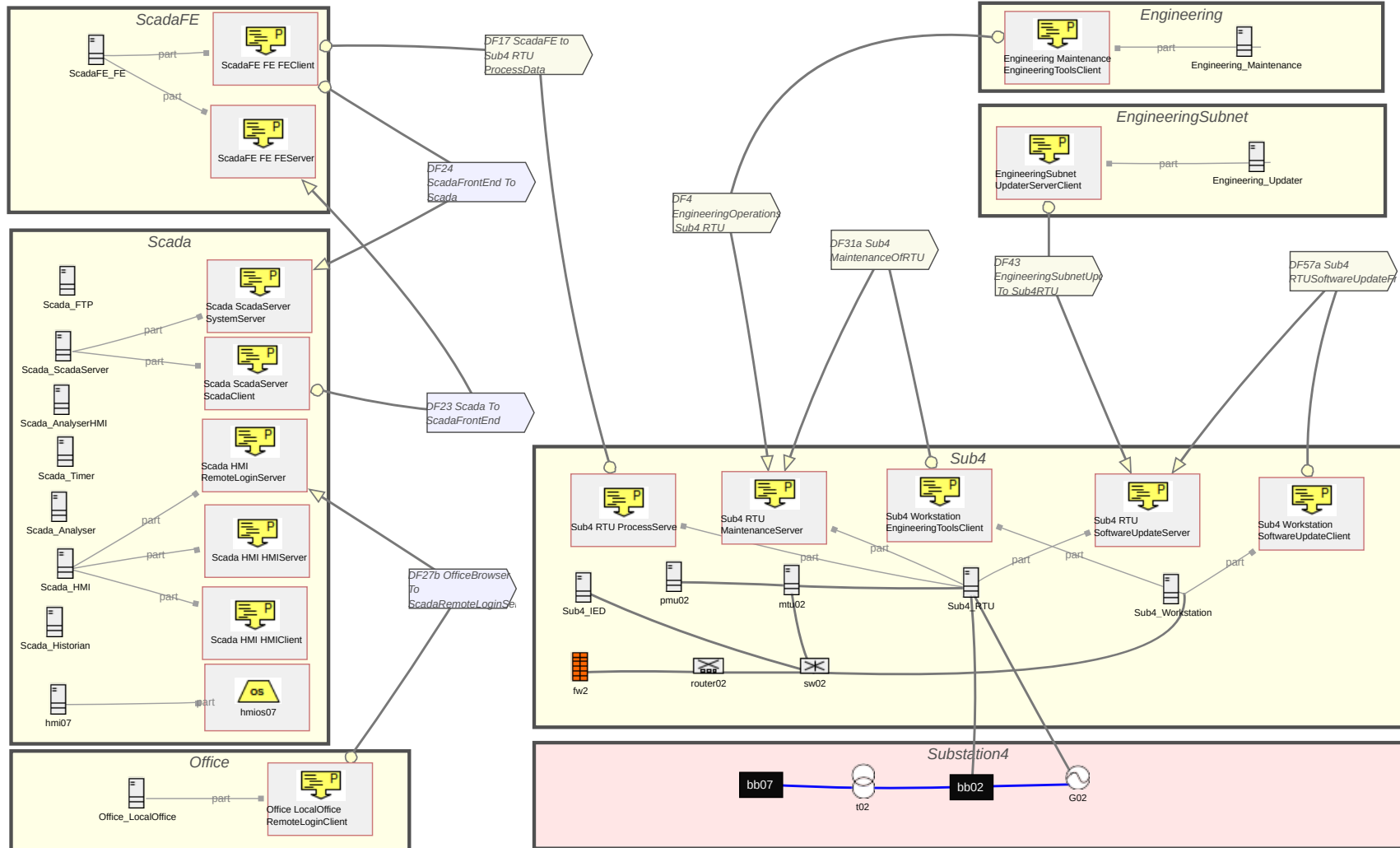
```

DF17_ScadaFE_to_Sub4_RTU_ProcessData
in BidirectionalDataStream with
participant
    sender1 : ScadaFE_FE_FEClient;
    receiver1 : Sub4_RTU_ProcessServer
protocol
    protocol1 : IEC_60870_5_101
initiator
    initiator1 : ScadaFE_FE_FEClient
end
    
```

[fig9.gel]

The datastream definitions establish a logical dependency between the components.

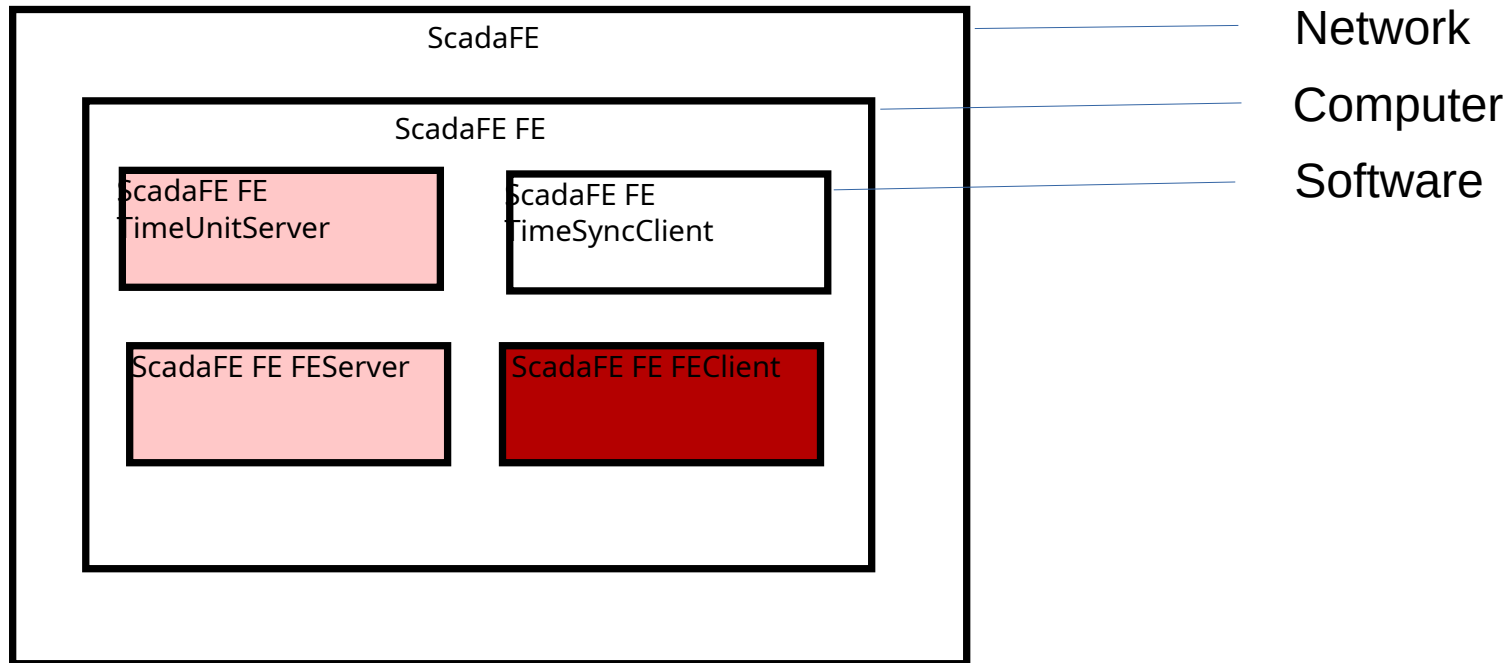
CYBER-PHYSICAL MODEL NINEBUSSC



[[elvira-referencemodel.gel](#)]

Datastream definitions establish a logical dependency between the components.

VISUALIZING CRITICALITY & VULNERABILITY



Criticality (C): number of command flows of some software to software on an OT component that controls a power grid Component

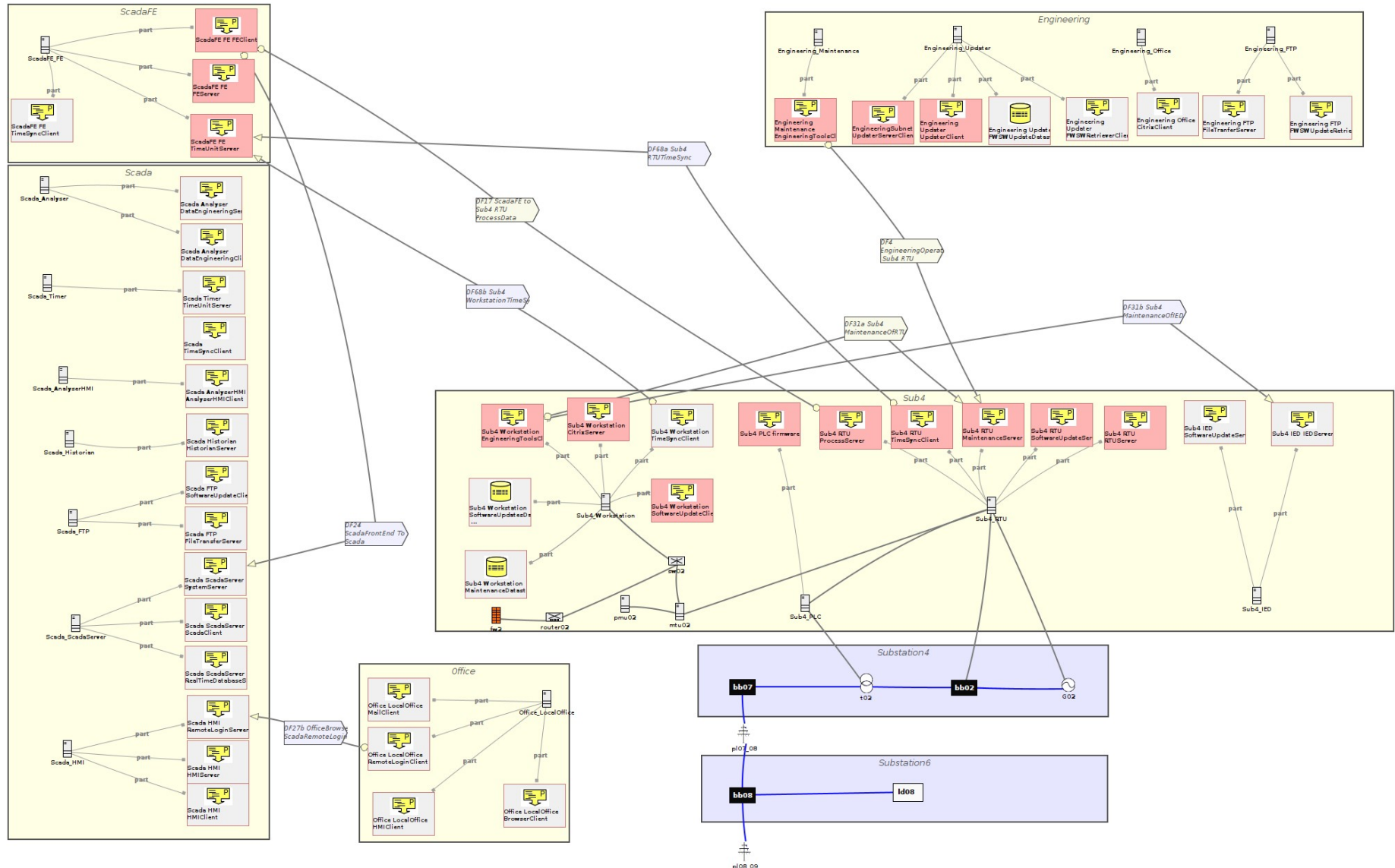
Vulnerability (V): aggregated CVSS score of CVE vulnerabilities for a software component

Total score = $C * V$

HEATMAP OF CRITICAL SOFTWARE COMPONENTS IN THE NINEBUS SC REFERENCE MODEL (EXCERPT)



HIGHLIGHT CRITICAL COMPONENTS OF NINEBUSSC



ANALYZE THE MODEL

- the network model is defined in terms of the ontology/taxonomy
- relations: powerconnection, dataconnection, subcomponent
- data streams (bidirectional or unidirectional) establish dependencies between the cyber part and the physical parts of the model

Now:

How can we analyze the model?

Here:

Use the query language of ConceptBase!

<https://conceptbase.sourceforge.net/userManual85/cbm003.html#sec28>

CBIVA: COUNT INSTANCES OF A CLASS

The screenshot shows the CBIVA - ConceptBase.cc User Interface in Java. The main window has a menu bar (File, Edit, Browse, Options, History, Windows, Help) and a toolbar. The toolbar contains several icons, with the 'ASK' button (a blue square with a white question mark) highlighted by a red box. An arrow points from the text 'press the ASK button' to this button. Below the toolbar is the 'Telos Editor' window, which contains the query: `1 COUNT(Component)`. An arrow points from the text 'formulate the query call, here a function that returns a single value' to this query. Below the Telos Editor is the 'History Window', which shows a list of queries. The first entry is '6: ASK' with a query icon. Below this, the query 'Query: COUNT(Component)' is shown, followed by the result 'Result: 526'. An arrow points from the text 'see the result (will also show up in the Telos Editor window)' to this result. At the bottom of the window, there is a status bar. The left part of the status bar is green and says 'Connected'. The right part says 'Version: Now -- 2020-03-24 12:22 (UTC)' and 'Module: NineBusSC'. An arrow points from the text 'note the module, here NineBusSC!' to the 'Module: NineBusSC' text.

formulate the query call, here a function that returns a single value

press the ASK button

see the result (will also show up in the Telos Editor window)

note the module, here NineBusSC!

EXTERNALLY CONTROLLED RTU

List those RTUs whose firmware (ComputerCode) is receiving messages (UnidirectionalDataStream) from programs running on another computer ("controller").

ExternallyControlledRTU in QueryClass

isA RTU with
constraint

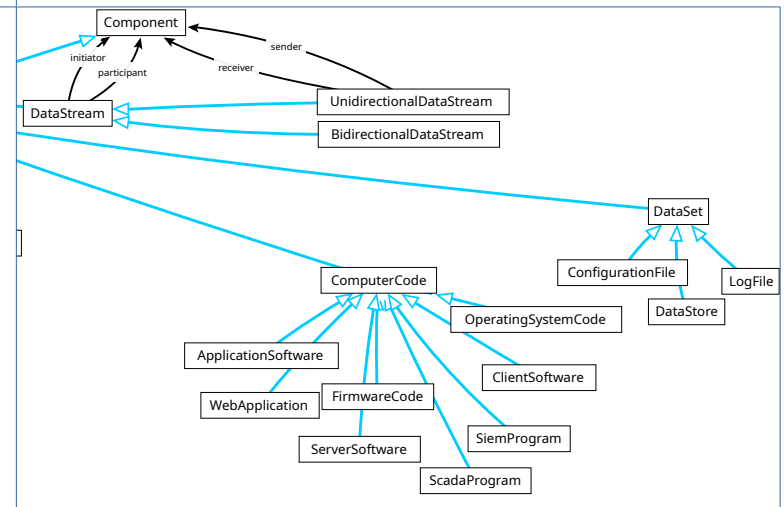
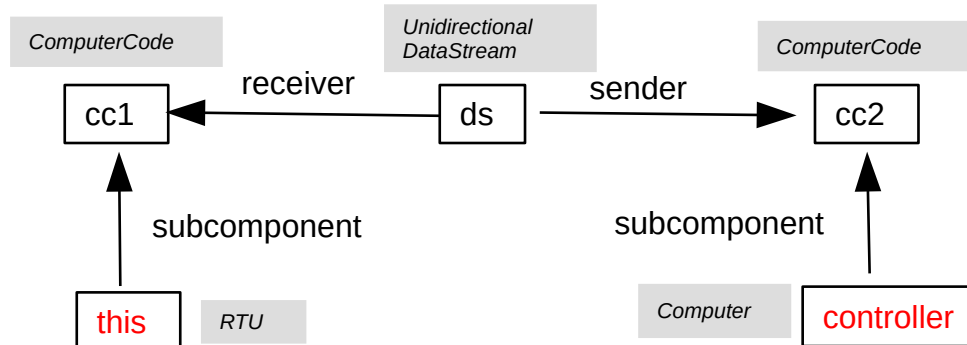
c1: \$ exists cc1,cc2/ComputerCode
ds/UnidirectionalDataStream
controller/Computer
(**this** subcomponent cc1) and
(**controller** subcomponent cc2) and
(ds sender cc2) and
(ds receiver cc1) \$

end

range of answer variable '**this**'

local variables with their classes (after "/")

condition



See also taxonomies in slides 41 and 43!

QUERY WITH RETURN VALUES

```
RTUcontrolledFrom in QueryClass
  isA RTU with
    computed_attribute
      controller: Computer
    constraint
      c1: $ exists cc1,cc2/ComputerCode
          ds/UnidirectionalDataStream
          (this subcomponent cc1) and
          (~controller subcomponent cc2) and
          (ds sender cc2) and
          (ds receiver cc1) $
end
```

variable to become
return value

The ~ character is just to indicate
that it is a variable. It can also be
omitted

This query returns all RTUs controlled by 'other' computers and attached the name of the other computer in the answer:

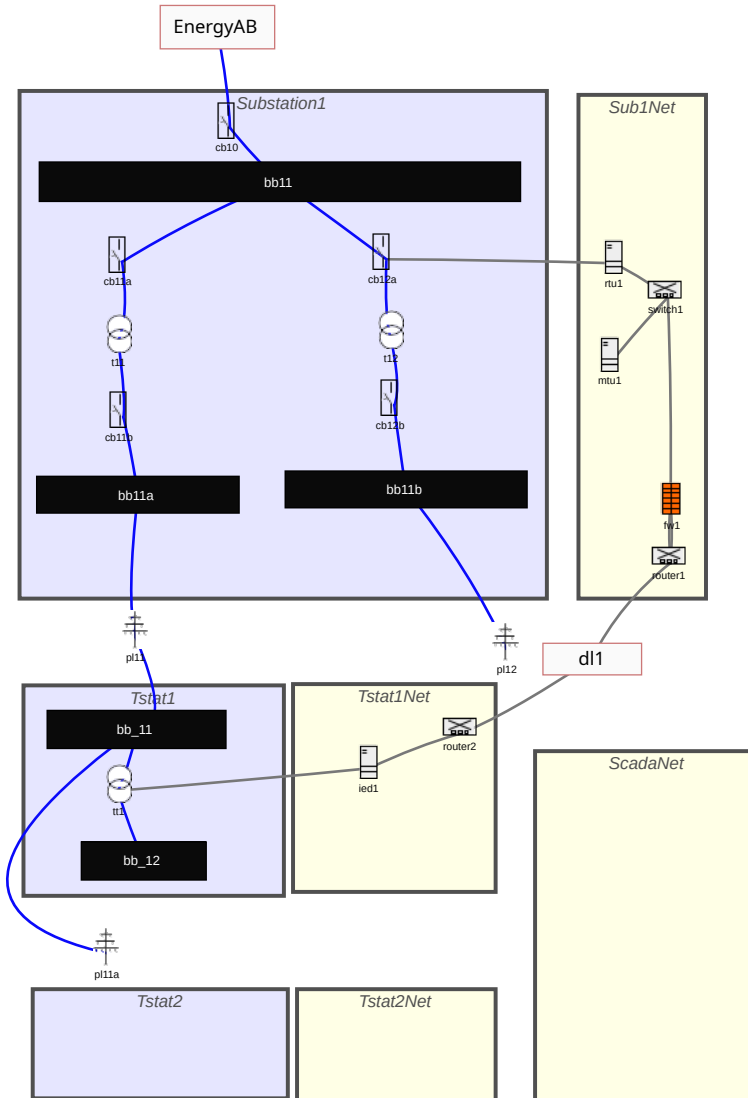
```
Sub1_RTU in RTUcontrolledFrom with
  controller
    COMPUTED_other_id_8457 : Engineering_Maintenance;
    COMPUTED_other_id_8563 : Sub1_Workstation;
    COMPUTED_other_id_8665 : Engineering_Updater
end
...
```

SUMMARY

- Ontology/taxonomy of cyber-physical components
- Actual CPS networks are defined as instances of the ontology classes
- Logical dependencies are established by data streams between components
- Use queries to analyze the model

PRACTICAL EXERCISE (MODELING)

Consider the following fragment of a municipal powergrid



Task 1: Add powergrid elements to Tstat2 analogous to the elements in Tstat1. Link the incoming busbar of Tstat2 to powerline p11a.

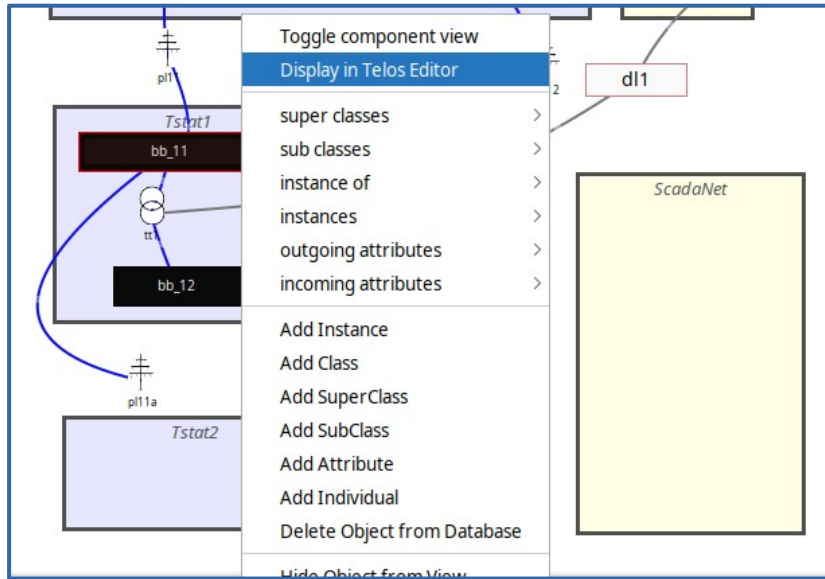
Task 2: Add a network Tstat2Net analogous to Tstat1Net.

Task 3: Create data connections between r tu1 and the transformer t12 in Substation1. Create a 2nd RTU r tu2 that controls the left side of the substation similar to r tu1.

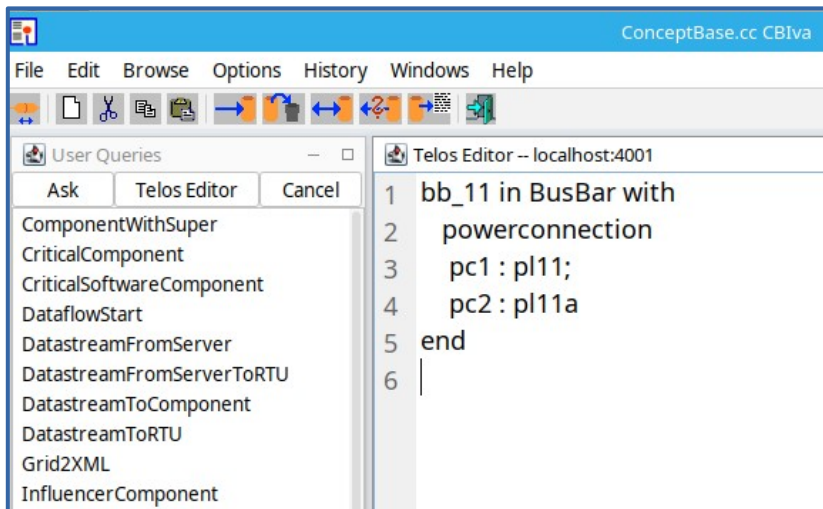
Task 4: Define a Scada network with a computer "scada_server" running a ScadaProgram called "microscada1". Add the usual network components like router, switch and firewall to the the Scada network. Link the router of the Scada network to the router in Sub1Net via a dataline (taxonomy type DataLine). Add a "processserver67" (taxonomy type FirmwareCode) as sub-component to mtu1.

Task 5: Create a data stream (taxonomy type UnidirectionalDataStream) between "microscada1" (sender) and "processserver67".

INSPECT AND MODIFY A MODEL



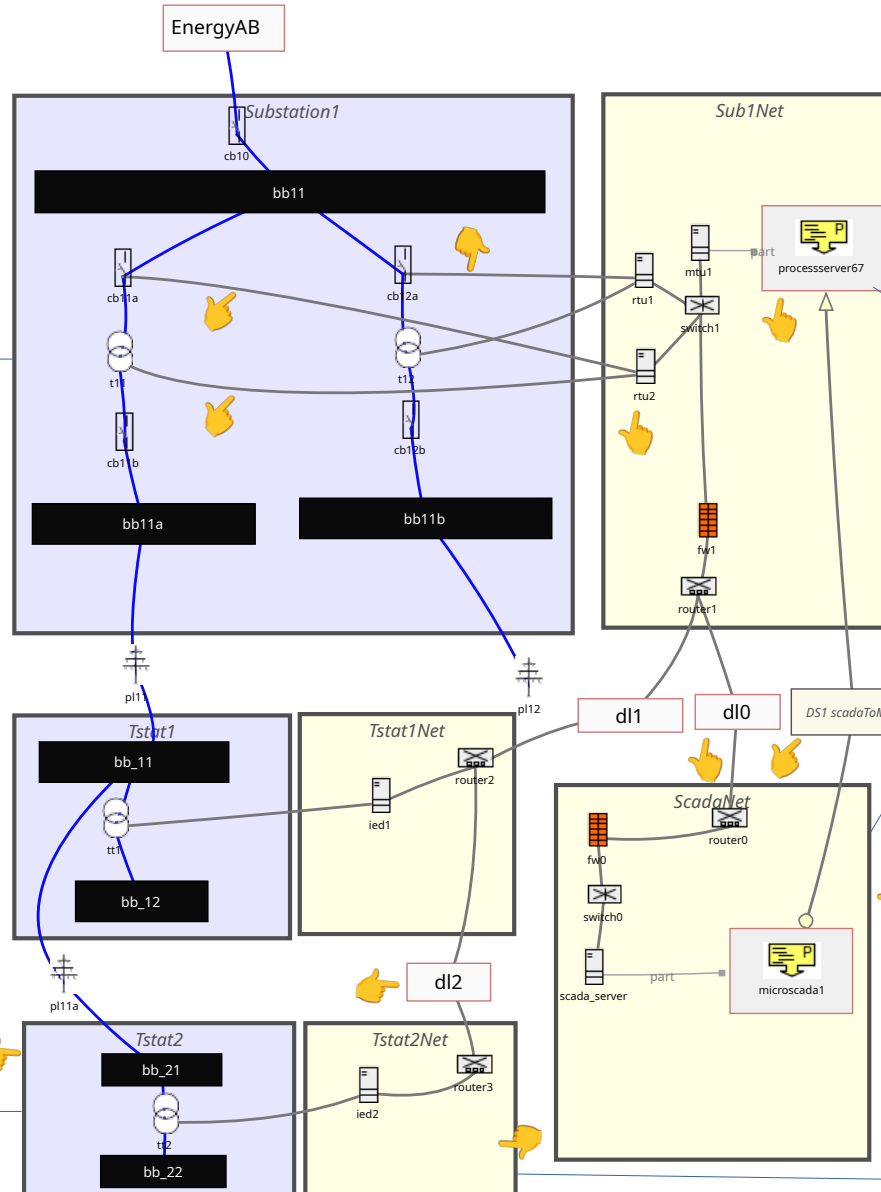
1. Right-click on an element like bb_11
2. Select Display in Telos Editor



You can also right-click an element in the Telos Editor and the select "Load Object" or use other options there.

After modify the definitions, you can press the Tell button to store it in the database.

THE TARGET MODEL



Task 3: Create data connections between rtu1 and the transformer t12 in Substation1. Create a 2nd RTU rtu2 that controls the left side of the substation similar to rtu1.

Task 4: Define a Scada network with a computer "scada_server" running a ScadaProgram called "microscada1". Add the usual network components like router, switch and firewall to the the Scada network. Link the router of the Scada network to the router in Sub1Net via a dataline (taxonomy concept DataLine). Add a "processserver67" (taxonomy type FirmwareCode) as sub-component to mtu1.

Task 5: Create a data stream (taxonomy UnidirectionalDataStream) between "microscada1" (sender) and "processserver67".

Task 1: Add powergrid elements to Tstat2 analogous to the elements in Tstat1. Link the incoming busbar of Tstat2 to powerline p11a

Task 2: Add a network Tstat2Net analogous to Tstat1Net.

Sources and examples for this tutorial and the practical exercises are at

conceptbase.sourceforge.net/elvira-edoc2026

THANK YOU

manfred.jeusfeld@acm.org
conceptbase.cc/mjf