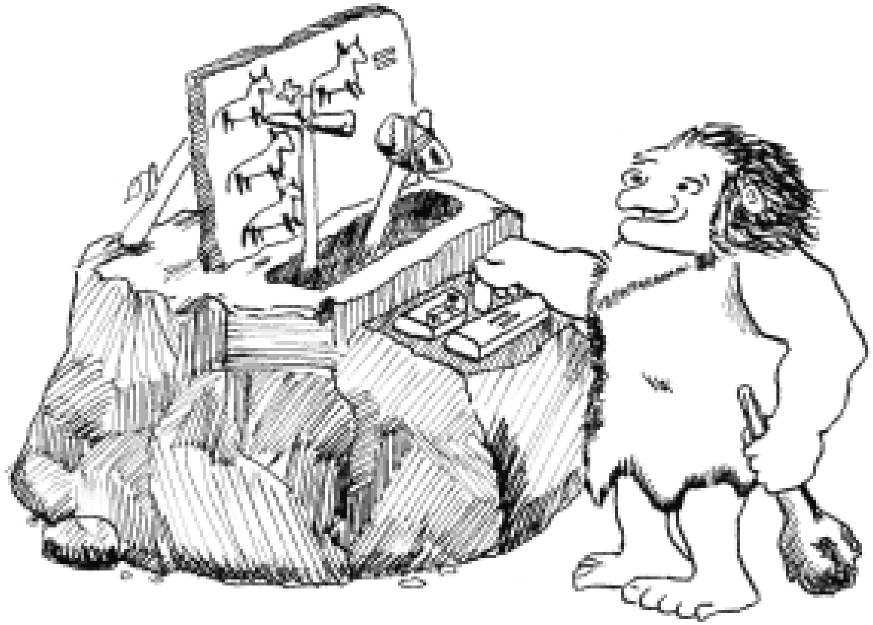
Physical architecture and the computational platform

Lecture 6
8 May 2025

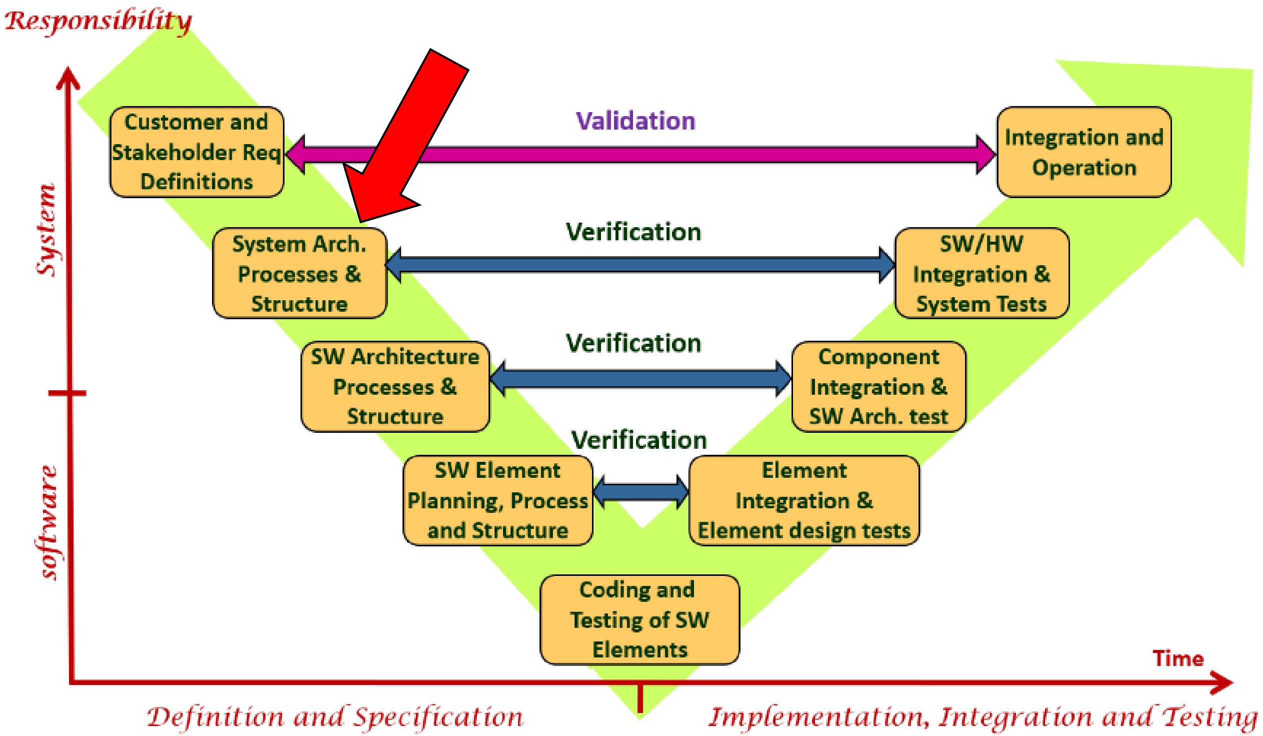
Slides created by
Prof Amir Tomer
tomera@cs.technion.ac.il



Planning and Documenting Physical Architecture

[System architecture: Structure]

- Our goal: Plan and document the system’s physical architecture and computational platform
- Inputs:
  - Hardware constraints (HC) from the requirements
  - System technical description
  - Plan/design for the system and hardware from systems engineering
- Outputs:
  - Physical architecture model
  - Trackback from HC to architecture



Responsibility
System
software
Customer and Stakeholder Req Definitions
Validation
Integration and Operation
System Arch. Processes & Structure
Verification
SW/HW Integration & System Tests
SW Architecture Processes & Structure
Verification
Component Integration & SW Arch. test
Verification
SW Element Planning, Process and Structure
Element Integration & Element design tests
Coding and Testing of SW Elements
Time
Definition and Specification
Implementation, Integration and Testing

Typical levels of scope in development of SIS

Reminder

Organization/Business Level

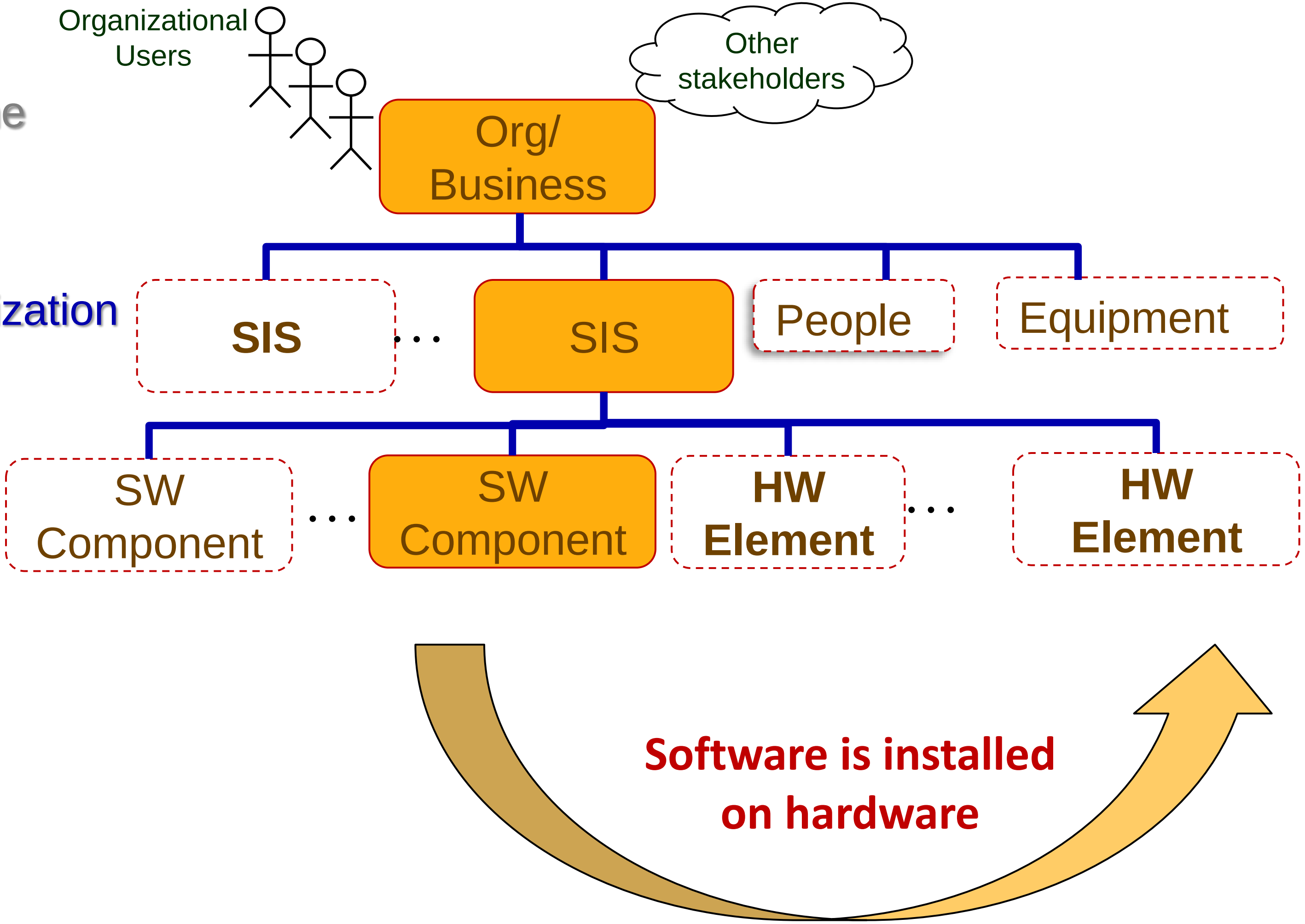
The operational framework in which the system will be installed and serve

Computer System Level

HW/SW system that serves the organization for some purpose

Software Component Level

Software is installed on the HW and users/other systems work with it



8 May 2025

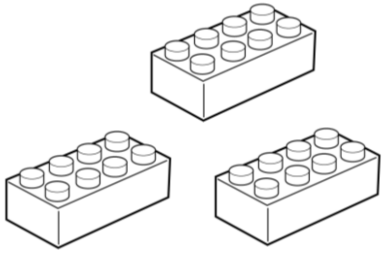
We are still at the system level

What do we have so far?

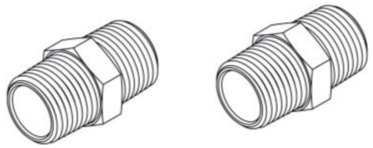
- Environment: Given
- Goals: Will be achieved via the defined system processes (i.e. use cases)

What do we need next?

- Ingredients
  - Hardware components
  - Software components



- Organization/Structure (internal and external connections)
  - Physical connections
  - Logical connections

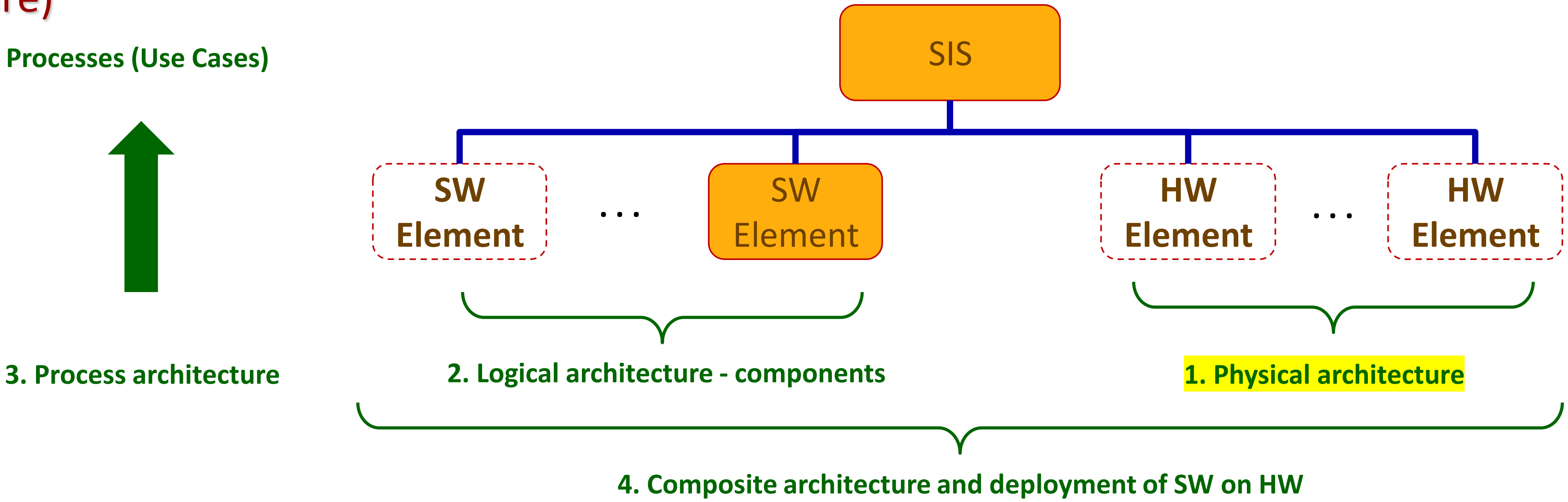


- Interaction and Behavior
  - How do the components work together to implement the defined processes?

Image source: https://cliparting.com/wp-content/uploads/2016/10/Lego-blocks-black-and-white-clipart-free-clip-art-images-image-2-3.gif, https://www.tompkinsind.ca/products/brass-adapters-and-fittings/brass-pipe-adapters/3325

System Architecture: SIS architecture includes

1. Physical architecture: Hardware components and physical connections – static model (structure)
2. Logical architecture: Software components and logical connections – static model (structure)
3. Process architecture: Implementation of processes via interaction between components - dynamic model (behavior)
4. Composite architecture: Implementation of logical connections via physical connections – static model (structure)



Processes (Use Cases)

SIS

SW Element … SW Element

HW Element … HW Element

3. Process architecture

2. Logical architecture - components

1. Physical architecture

4. Composite architecture and deployment of SW on HW

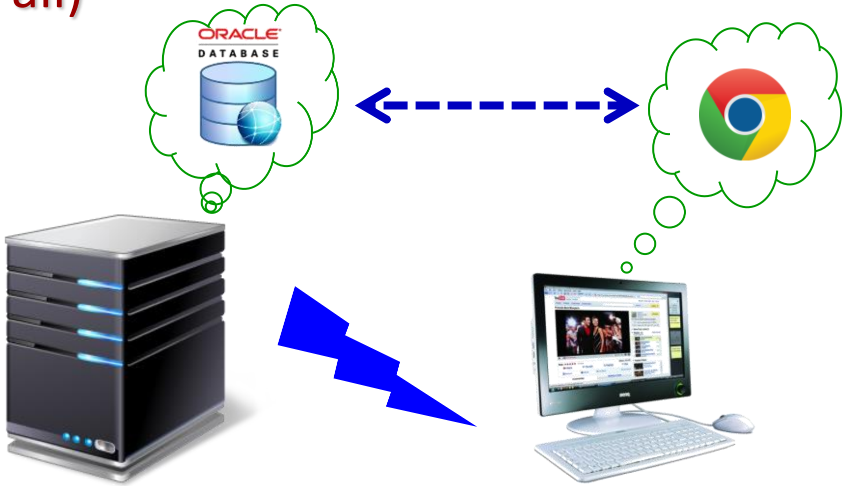
Sample Architectures

- Physical architecture
  - Linux server, Windows client endpoint, network connectivity, TCP/IP
  - Dedicated computer, no communication

- Logical architecture
  - Exchange server, Outlook client, SMTP protocol
  - eBay website, Oracle DB, Chrome browser, HTTP
  - Navigation service, Waze app, HTTP

- Composite architecture
  - Backend service on server, frontend on client endpoint
  - Frontend and backend on standalone computers

- Supported processes (for all)
  - Sending & receiving email
  - E-commerce
  - Navigation



8 May 2025

Where do we start?

Case 1: Physical architecture already exists/predefined

- Commonly occurs when:
  - Developing an embedded multi-functional system (SW, HW, mechanical, …)
  - Upgrading legacy system
  - (Simple) Web applications
- Work procedure
  - Document physical architecture
  - Build appropriate logical architecture

Case 2: Physical architecture not yet defined

- Commonly occurs when:
  - Information systems
  - Startup
  - Building libraries or generic software
- Work procedure
  - Build the logical architecture
  - Consider alternatives for physical architecture

Computational Platform: Where the SW runs

- Computational platform includes

HW Environment (Physical Architecture)

- Physical components
  - HW boxes
    - Processors, storage devices, communication devices
- Physical interfaces
  - Connections between physical components to transfer information
    - Connectors, cables, electromagnetic radiation, internet
- Physical protocols
  - How the physical components communicate
    - RS-232, Bluetooth, HTTP, TCP/IP

Software Environment

- Execution environment
  - Environment that allows the SW to run on the HW
    - OS, DBMS, Interpreter
- Additional elements
  - Other programs installed on the HW that help the SW under development
    - Config files, database, DLLs

Common computational platforms (HW)

Computers

- Servers
- Endpoint computers
- Microprocessors

Storage devices

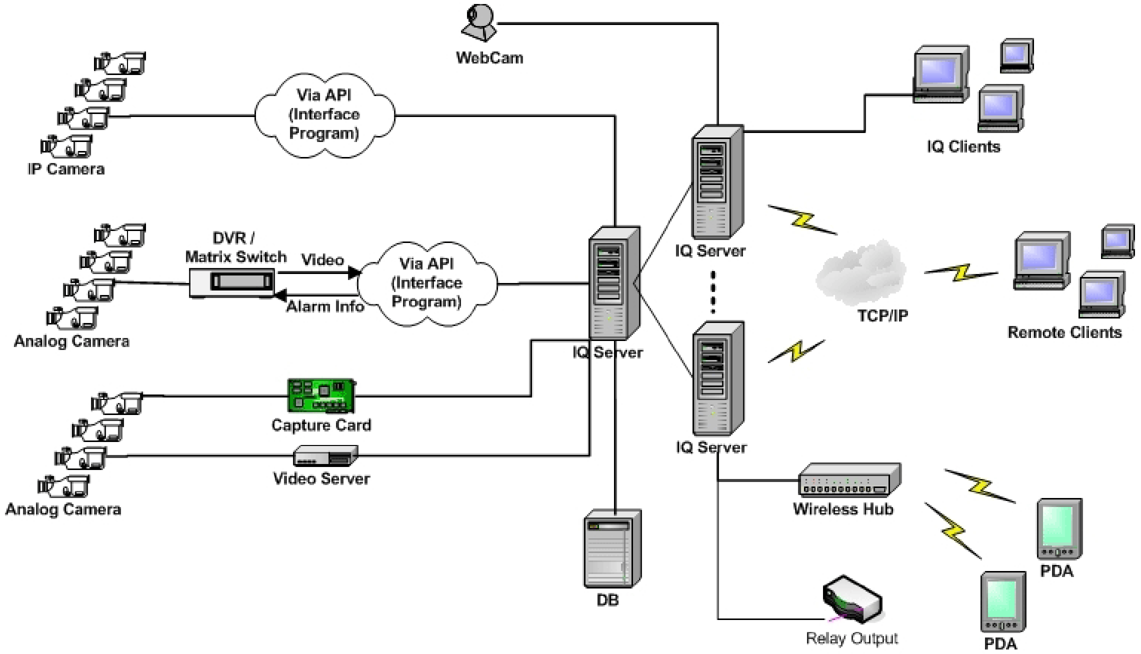
- Disks
- External storage devices
- Network attached storage (NAS)

Communication devices

- Modems
- Routers, switches
- Antennas, Wi-Fi/Cellular

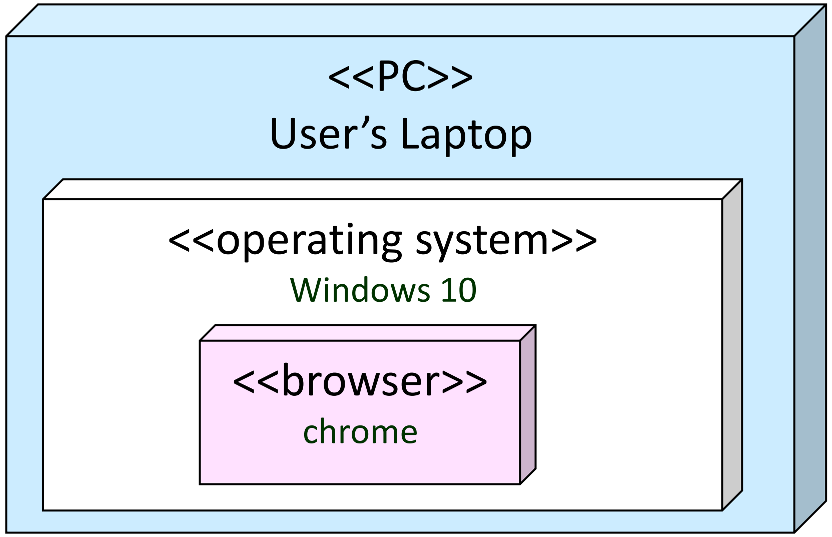
Peripherals

- Printers
- Scanners
- Display devices
- Sensors, cameras



UML Physical architecture: Nodes

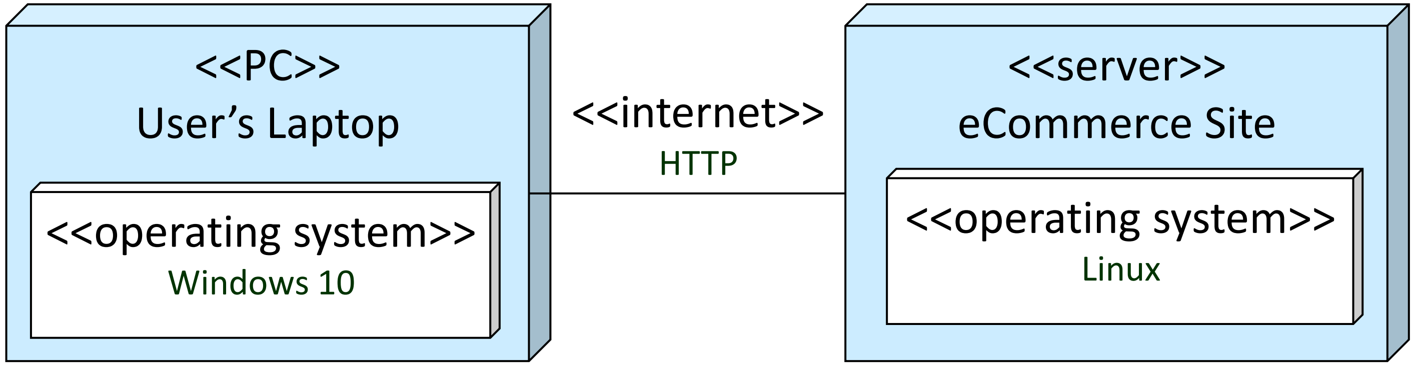
- Node: A active physical computational object that normally has memory and processing ability
  - Common stereotypes: <<server>>, <<device>>, <<smartphone>>
  - Common understanding: <<device>> is a non-programmable element or one with built-in SW
- Execution environment
  - Common stereotypes: <<operating system>>, <<web server>>, <<cloud>>
- Nodes can be nested



For simplicity: from here on, we'll normally consider all layers as a single entity

UML Physical architecture: Connections

- Communication paths
  - Physical connection between nodes
    - Normally non-directional (full duplex)
    - Define medium and protocol
  - Common practice: Stereotype is medium, name is the protocol



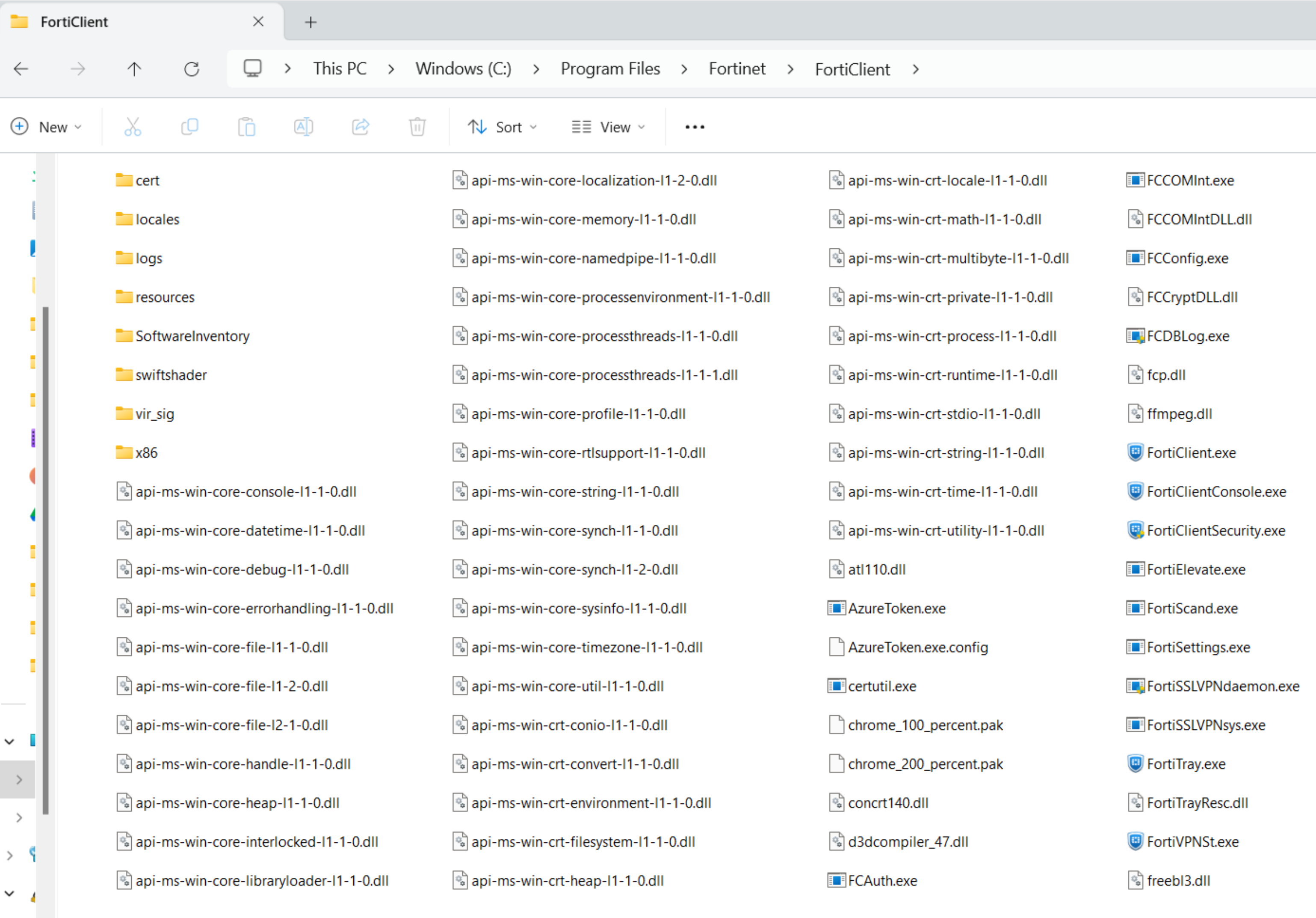
Common software items to find on the physical platform

- Platform software
  - OS, communications
  - Standard apps (browser)
- Executables
  - .exe, .jar, .py
  - DLLs, Drivers

- Configuration
  - Installation files
  - Registry
  - Format files
- Data items
  - Data files
  - Databases

- Media items
  - Images
  - Audio, video
- Information items
  - Help files
  - Online manuals

Common software items to find on the physical platform

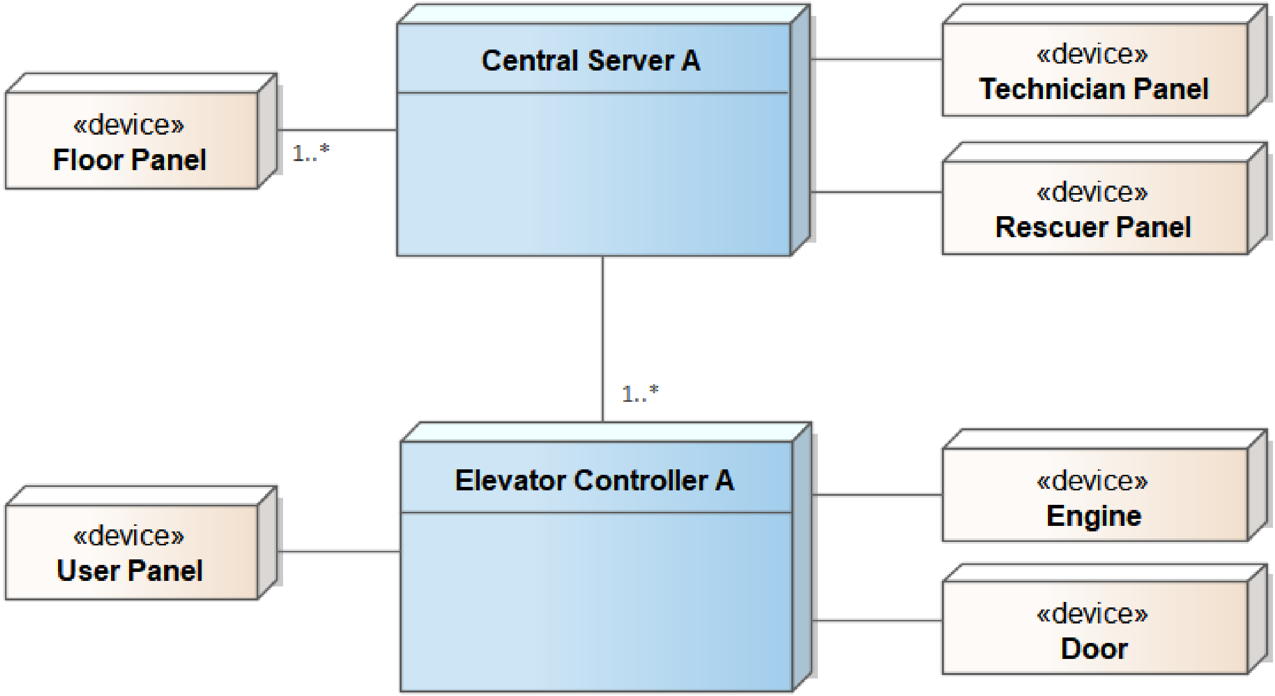


This PC > Windows (C:) > Program Files > Fortinet > FortiClient >

cert
locales
logs
resources
SoftwareInventory
swiftshader
vir_sig
x86
api-ms-win-core-console-l1-1-0.dll
api-ms-win-core-datetime-l1-1-0.dll
api-ms-win-core-debug-l1-1-0.dll
api-ms-win-core-errorhandling-l1-1-0.dll
api-ms-win-core-file-l1-1-0.dll
api-ms-win-core-file-l1-2-0.dll
api-ms-win-core-file-l2-1-0.dll
api-ms-win-core-handle-l1-1-0.dll
api-ms-win-core-heap-l1-1-0.dll
api-ms-win-core-interlocked-l1-1-0.dll
api-ms-win-core-libraryloader-l1-1-0.dll
api-ms-win-core-localization-l1-2-0.dll
api-ms-win-core-memory-l1-1-0.dll
api-ms-win-core-namedpipe-l1-1-0.dll
api-ms-win-core-processenvironment-l1-1-0.dll
api-ms-win-core-processthreads-l1-1-0.dll
api-ms-win-core-processthreads-l1-1-1.dll
api-ms-win-core-profile-l1-1-0.dll
api-ms-win-core-rtlsupport-l1-1-0.dll
api-ms-win-core-string-l1-1-0.dll
api-ms-win-core-synch-l1-1-0.dll
api-ms-win-core-synch-l1-2-0.dll
api-ms-win-core-sysinfo-l1-1-0.dll
api-ms-win-core-timezone-l1-1-0.dll
api-ms-win-core-util-l1-1-0.dll
api-ms-win-crt-conio-l1-1-0.dll
api-ms-win-crt-convert-l1-1-0.dll
api-ms-win-crt-environment-l1-1-0.dll
api-ms-win-crt-filesystem-l1-1-0.dll
api-ms-win-crt-heap-l1-1-0.dll
api-ms-win-crt-locale-l1-1-0.dll
api-ms-win-crt-math-l1-1-0.dll
api-ms-win-crt-multibyte-l1-1-0.dll
api-ms-win-crt-private-l1-1-0.dll
api-ms-win-crt-process-l1-1-0.dll
api-ms-win-crt-runtime-l1-1-0.dll
api-ms-win-crt-stdio-l1-1-0.dll
api-ms-win-crt-string-l1-1-0.dll
api-ms-win-crt-time-l1-1-0.dll
api-ms-win-crt-utility-l1-1-0.dll
atl110.dll
AzureToken.exe
AzureToken.exe.config
certutil.exe
chrome_100_percent.pak
chrome_200_percent.pak
concrt140.dll
d3dcompiler_47.dll
FCAuth.exe
FCCOMInt.exe
FCCOMIntDLL.dll
FCConfig.exe
FCCryptDLL.dll
FCDBLog.exe
fcp.dll
ffmpeg.dll
FortiClient.exe
FortiClientConsole.exe
FortiClientSecurity.exe
FortiElevate.exe
FortiScand.exe
FortiSettings.exe
FortiSSLVPNdaemon.exe
FortiSSLVPNsys.exe
FortiTray.exe
FortiTrayResc.dll
FortiVPNSt.exe
freebl3.dll

Elevator System Physical Architecture (v1)

- Distributed architecture
  - Every elevator is an independent node with local computational services for its riders
  - A central server manages and controls the whole system and gives central services (maintenance, rescue)
  - Direct connection between elevators and the server (default: cabling, protocol up to the hardware engineer)



Documenting the Physical Architecture

Document all known architectural elements and details

- Write details textually or using the modeling tool

Computers, including

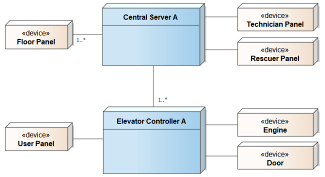
- Computing hardware
- Operating systems
- Execution environments (if applicable)
- Software artifacts installed (or that will be)

Physical interfaces:

- Medium
- Protocol

Unknowns can be filled in during development

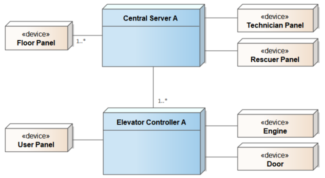
Example architectural documentation



Computers

Computer	Element	Diagram ID	Type	Role	Req. Trackback
Central Server	Hardware	Central Server	Server *TBD*	Central server for all elevators	
	Operating System	*TBD*	*TBD*	OS for the central server	
	Execution environment	None	-	-	
	Software artifact 1	Server SW	-	Manages the whole system, including maintenance and rescue	
Elevator controller	Hardware	Elevator controller	Controller *TBD*	Computer that manages the elevator car	
	Operating System	*TBD*	*TBD*		
	Execution environment	None	-	-	
	Software artifact 1	Elevator SW		Manages the car’s operations	

Example architectural documentation



Devices

Device	Diagram ID	Type	Role	Req. Trackback
Floor panel	Floor panel	Light-up buttons	Per floor interface to call elevator car	
Technician panel	Technician Panel	Control panel	Manage system tests by technicians	
Rescuer panel	Rescuer panel	Control panel	Manage rescue operations	
User panel	User panel	Light-up buttons	User interface inside the elevator	
Engine	Engine	Engine with control board	Move and stop the elevator	
Door	Door	Door with control board	Open and close the elevator entry	

In class assignment: Physical architecture for ePark

- Based on the customer story and considering the requirements list (primarily the hardware constraints HC), offer an architecture for the ePark system by making a deployment diagram
  - Choose hardware nodes
  - Define the connections between them with the proper multiplicity labels
  - Try to offer execution environments/software elements (that you know about now) for the architecture you chose

Non-Functional Requirements: Quality of the soluti

Reminder

Specify additional aspects of the solution that must be met while meeting the functional requirements

Performance Requirement (PR)

- Parameters that measure the speed of actions
- Response time, data size, processor utilization

Quality Attributes (QA)

General aspects of the solution

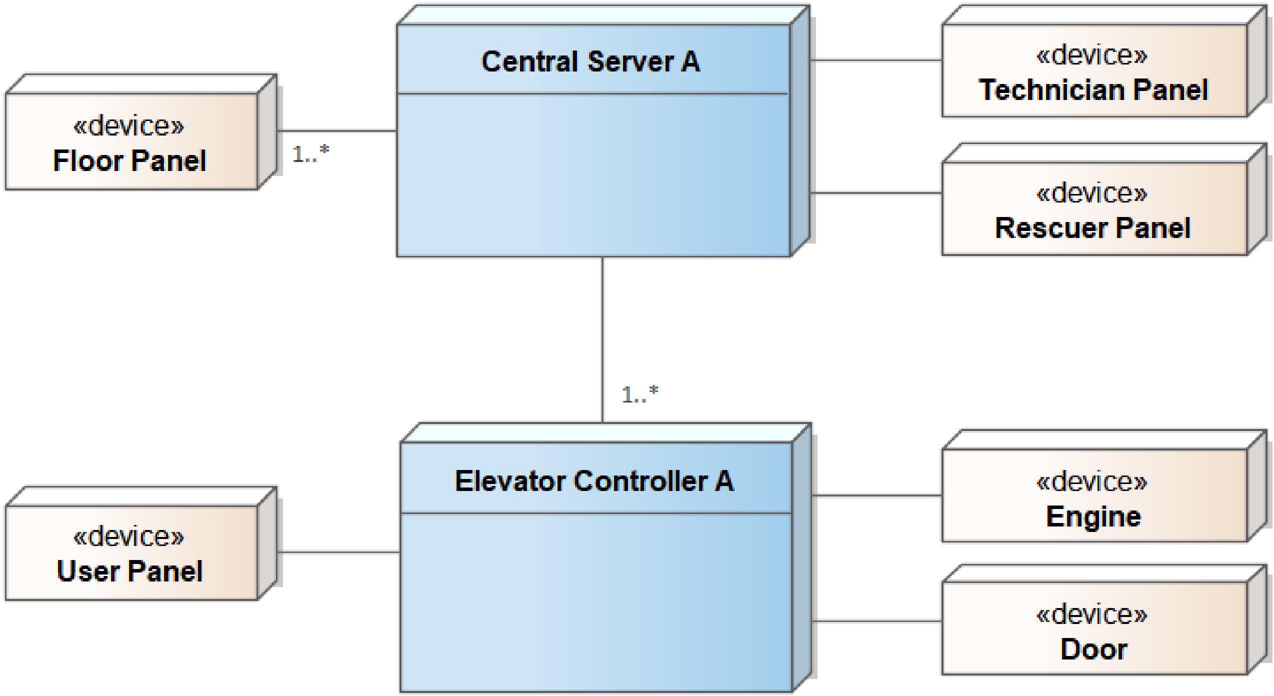
- Reliability: Works without errors for a certain amount of time
- Availability: Continuous service, fast recovery from errors
- Safety: Protects users and the environment from the system
- Security: Protects the system from users
- Testability: Ability to test and verify the systems actions (also after the fact)
- Maintainability: Ability to easily change and repair the product
- Usability: Effectiveness and efficiency that the system gives users in performing their tasks and reaching their goals

For QA, ensure the requirement is measurable and verifiable!

Elevator System Physical Architecture (v1)

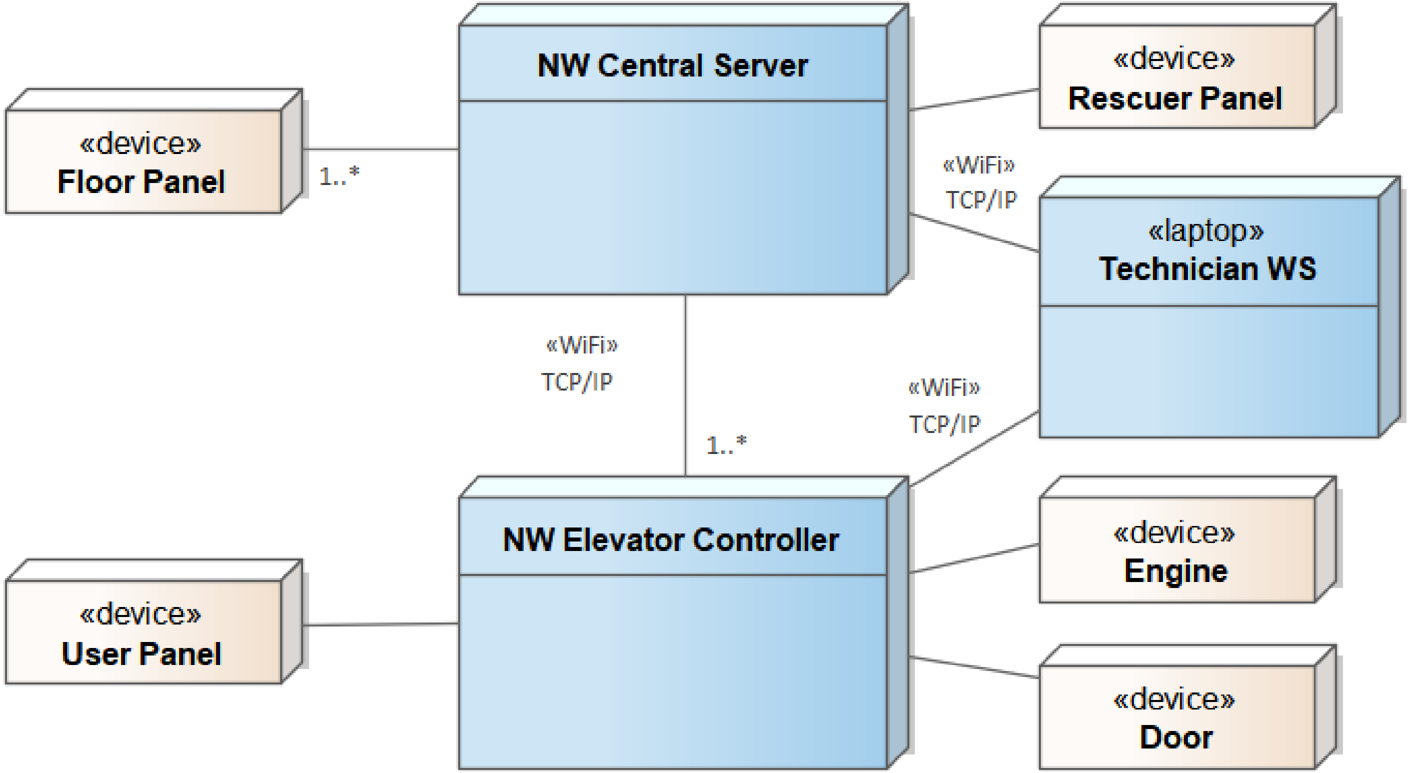
Reminder

- Distributed architecture
  - Every elevator is an independent node with local computational services for its riders
  - A central server manages and controls the whole system and gives central services (maintenance, rescue)
  - Direct connection between elevators and the server (default: cabling, protocol up to the hardware engineer)



Elevator System Physical Architecture (v2)

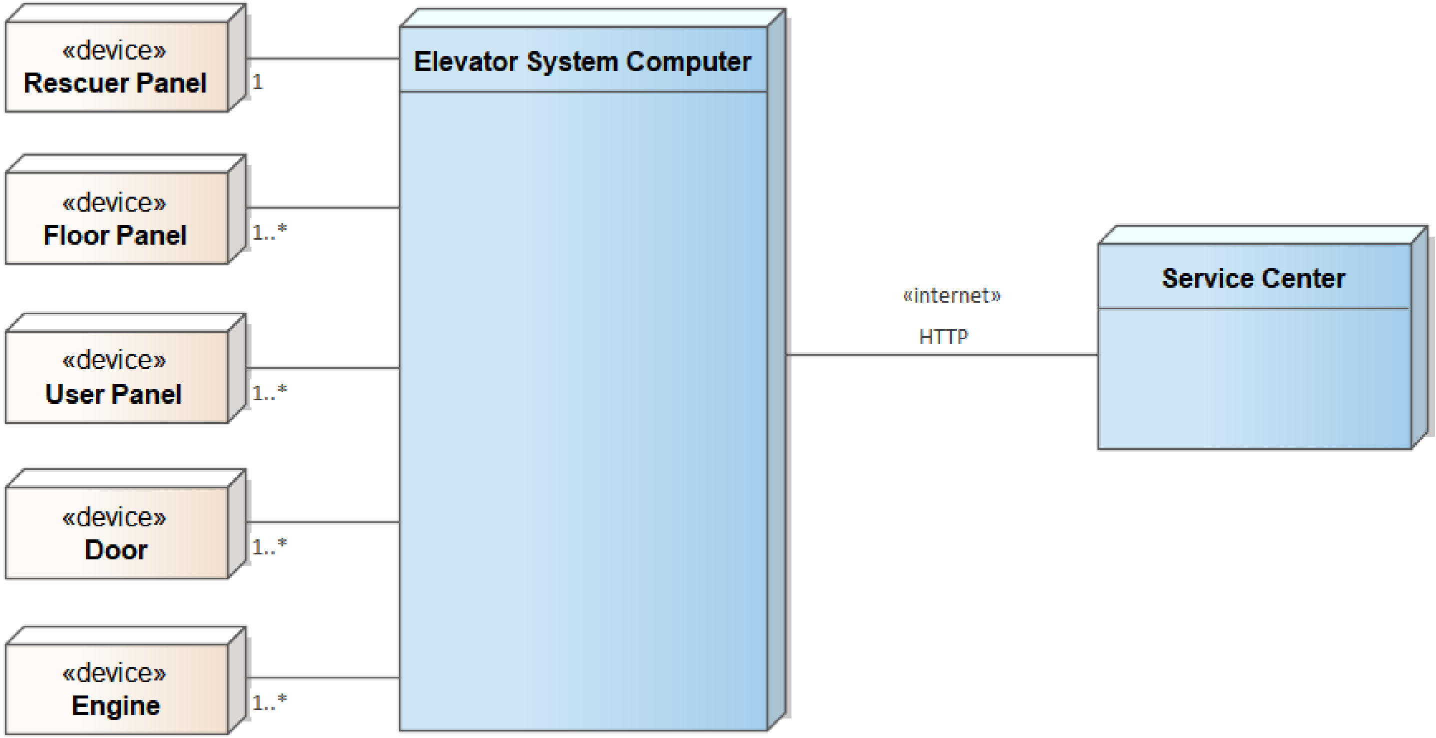
- Network architecture
  - Central server and elevators connected via a wireless network
  - Technician comes with a laptop and connects to the system via the network



8 May 2025

Elevator System Physical Architecture (v3)

- Centralized architecture
  - The whole system is controlled and operated by a single computer with IoT connections to all devices
  - External services (operation and control) are offered remotely via the internet



Evaluating alternatives based on quality attributes

- The architecture chosen affects the QA of the system
  - The most influential requirements on the architecture are the non-functional ones
  - All architectures suggested can meet the functional requirements

Architecture Quality Attribute	Distributed (v1)	Network (v2)	Centralized (v3)
Performance	H	M	M
Availability	H	M	L
Security	H	L	H
Maintainability	L	M	H
Cost	H	M	L

Conclusion

- Physical architecture