AV Lecture in Summer Term 2018

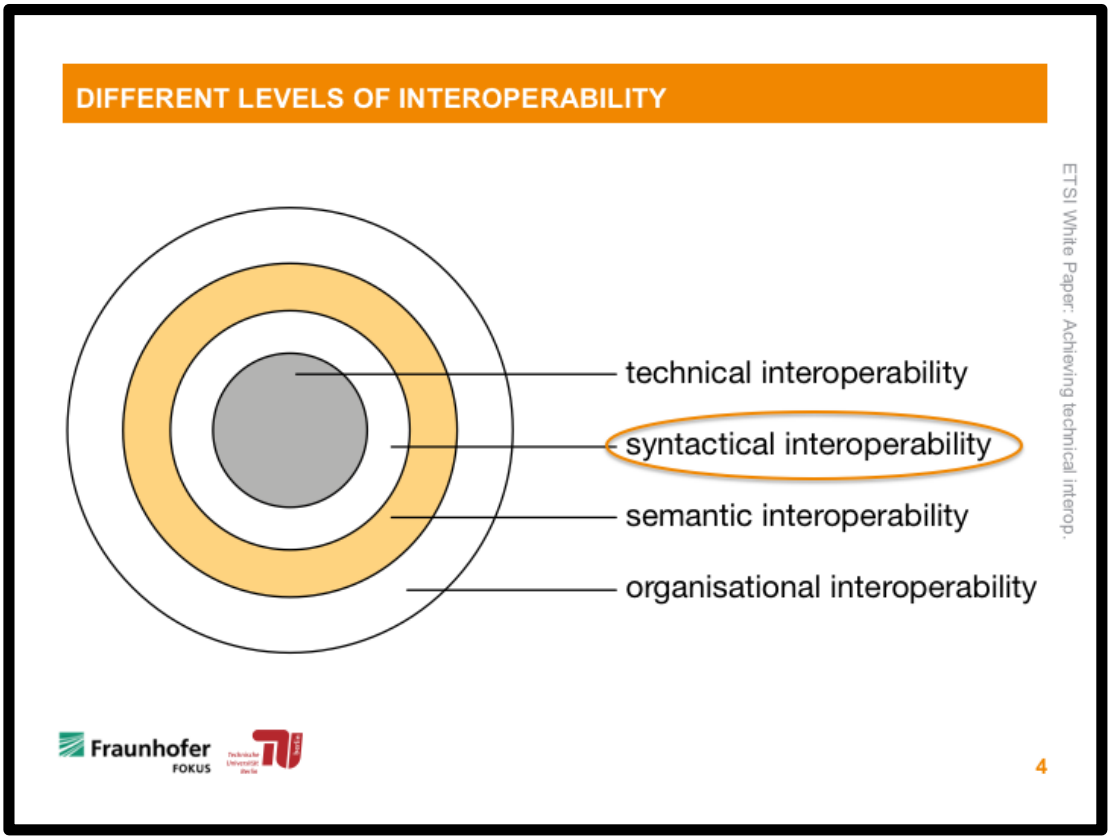
Dr.-Ing. Alexander **Willner**, Mathias **de Brito**, Varun **Gowtham**, Ronald **Steinke**

Fraunhofer FOKUS

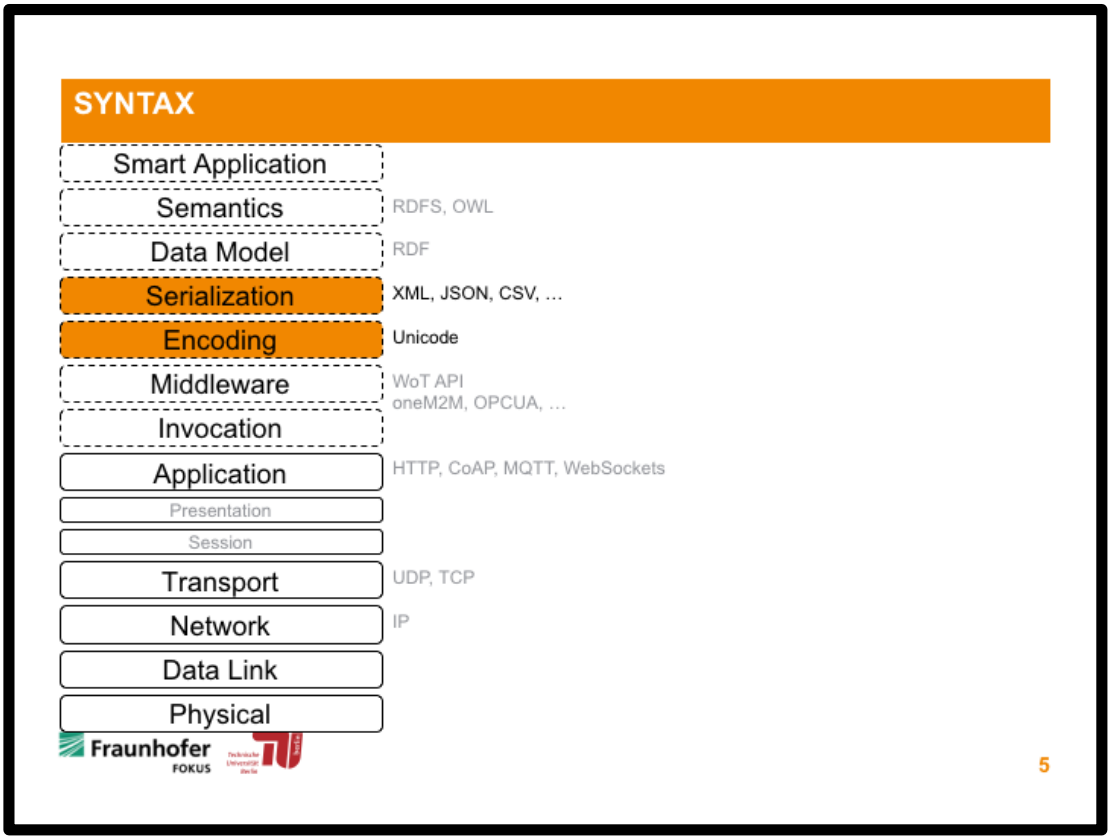
THE LAST LECTURE

3 minutes

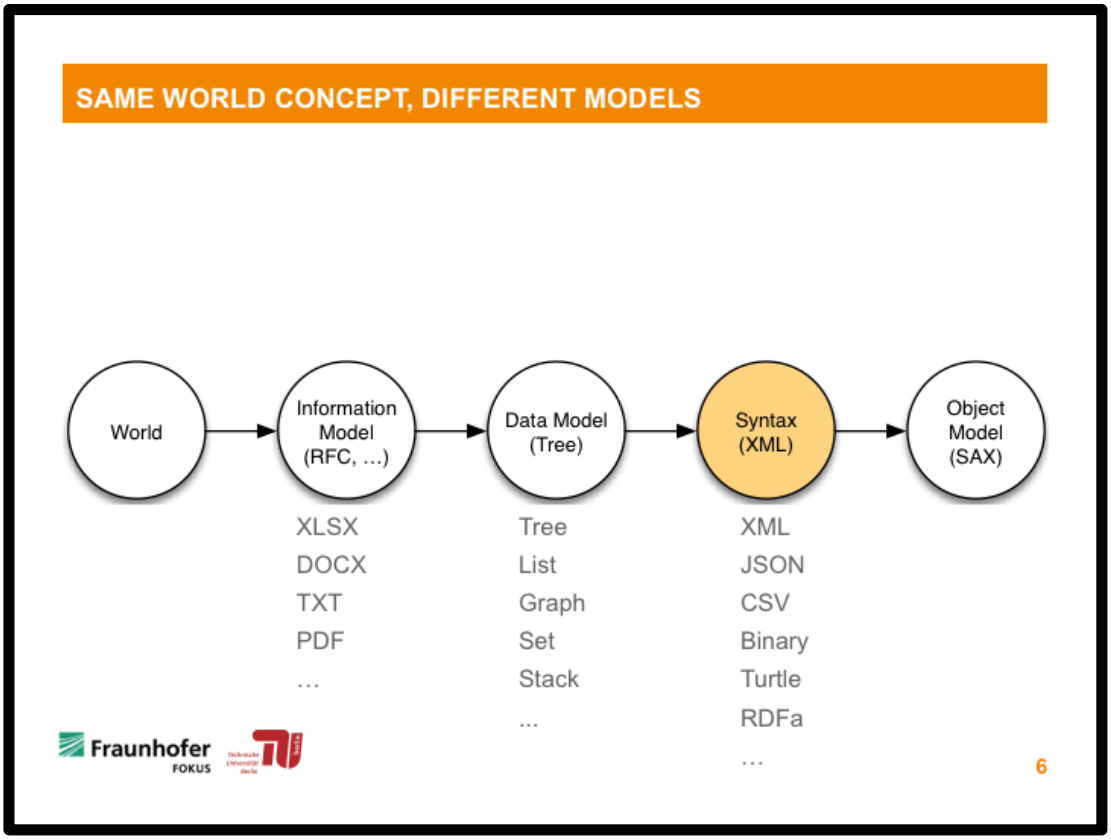
- Last lecture we talked about aspect 3



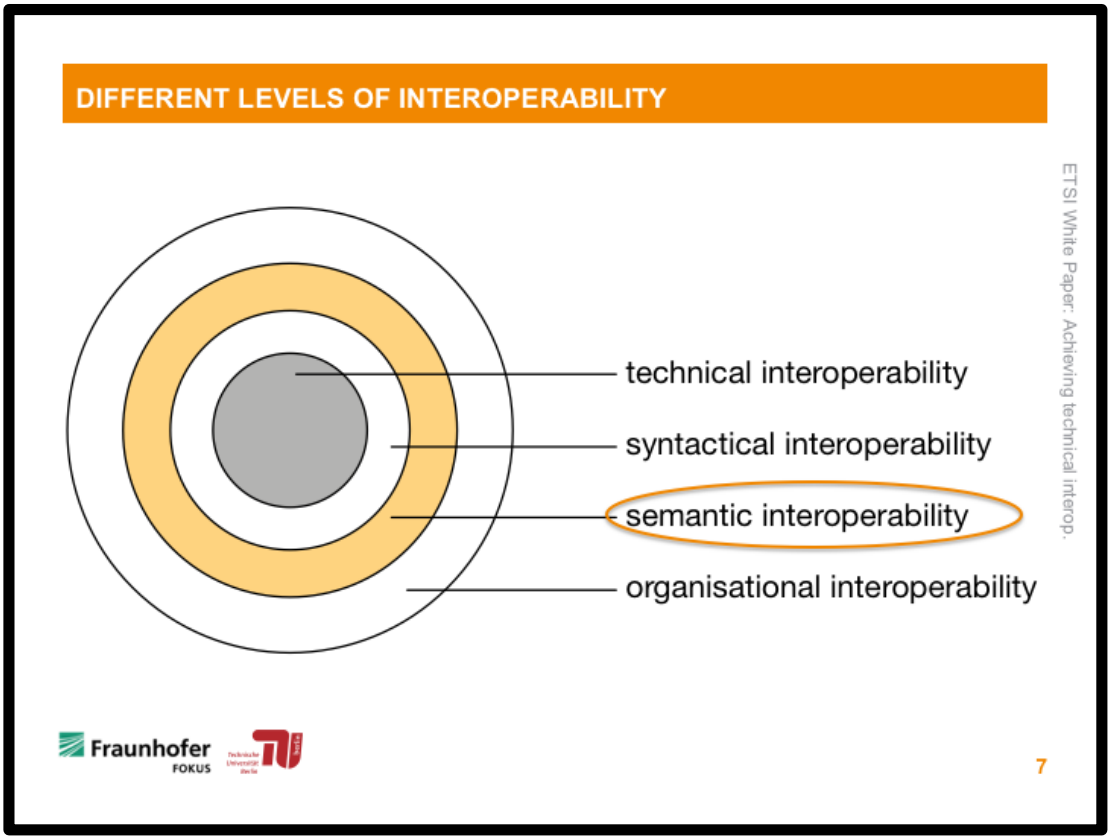
- We talked about the syntactic layer



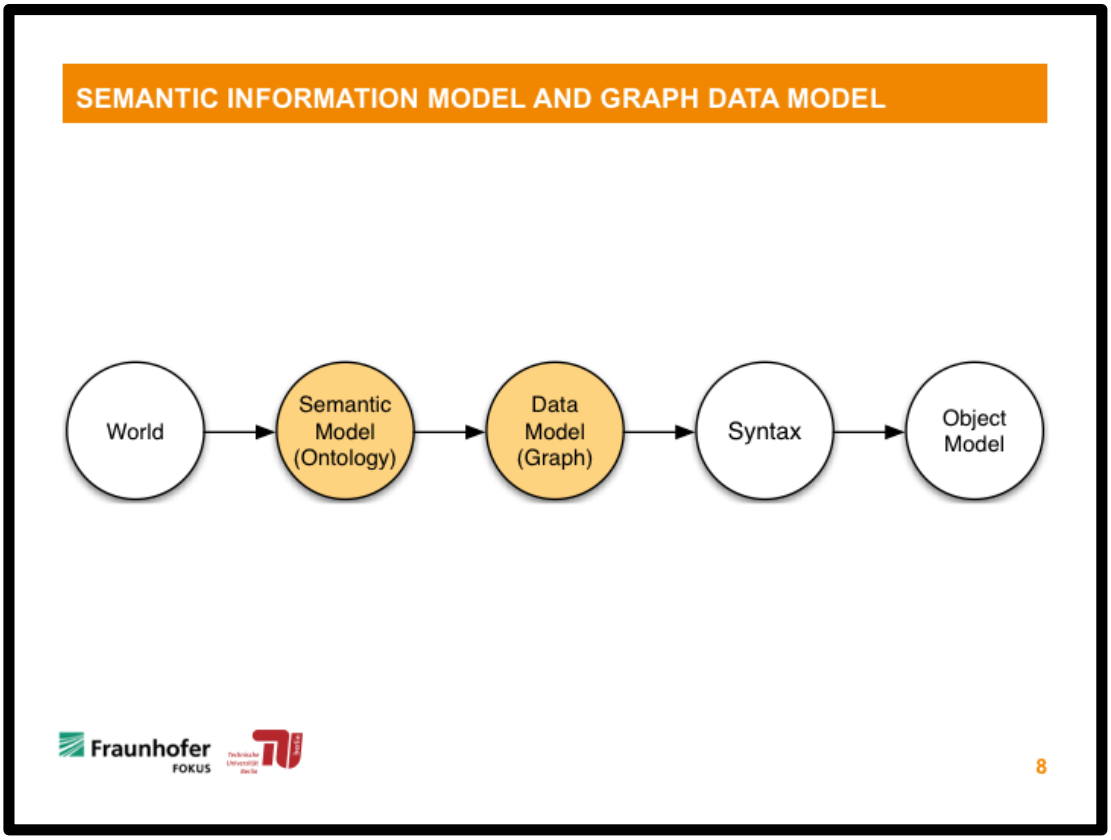
- Two main aspects



- An information model is an abstraction of a real word concept
- Usually human readable (from humans for humans)
- Examples below

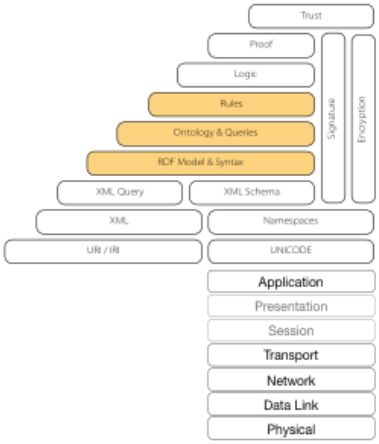


- Next, we then talked about the semantic layer



- Moving from human readable information models to formal information models
- Moving from lists or trees to graph data models

THE SEMANTIC WEB IS ONE POSSIBLE IMPLEMENTATION



Fraunhofer FOKUS

9

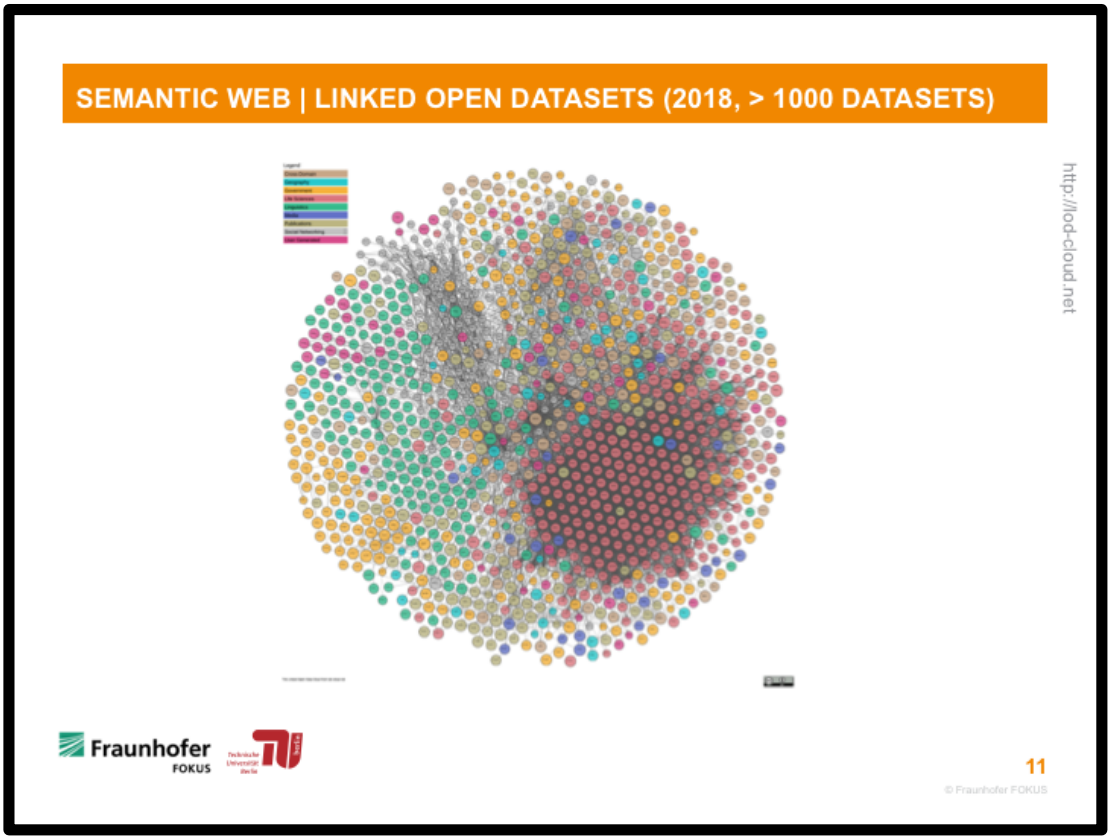
- The Semantic Web is ONE possible implementation of the related subsymbolic KI mechanisms

5 STAR LINKED OPEN DATA

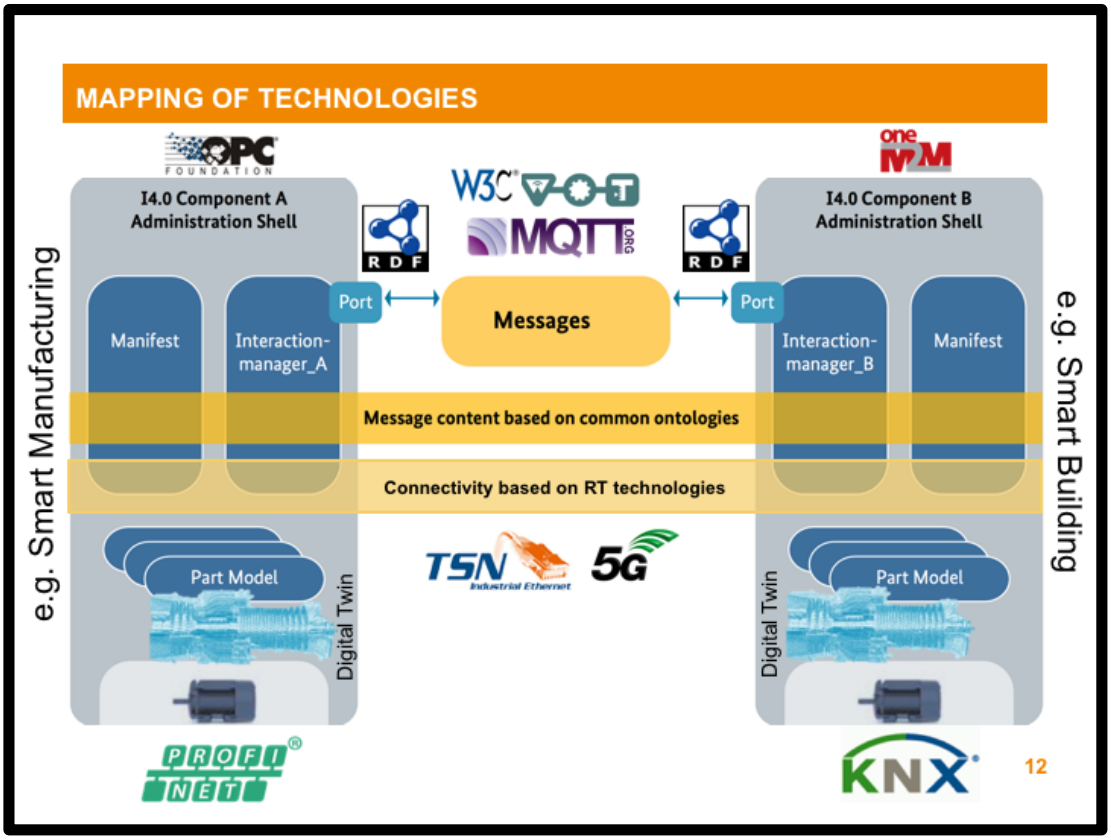
Tim Burners Lee

Fraunhofer FOKUS

- We talked about Linked Open Data (LOD)



- Using LOD principles it is possible to create large interconnected knowledge graphs



- We mapped the technologies we already learned about to an overview

- We finished with this question

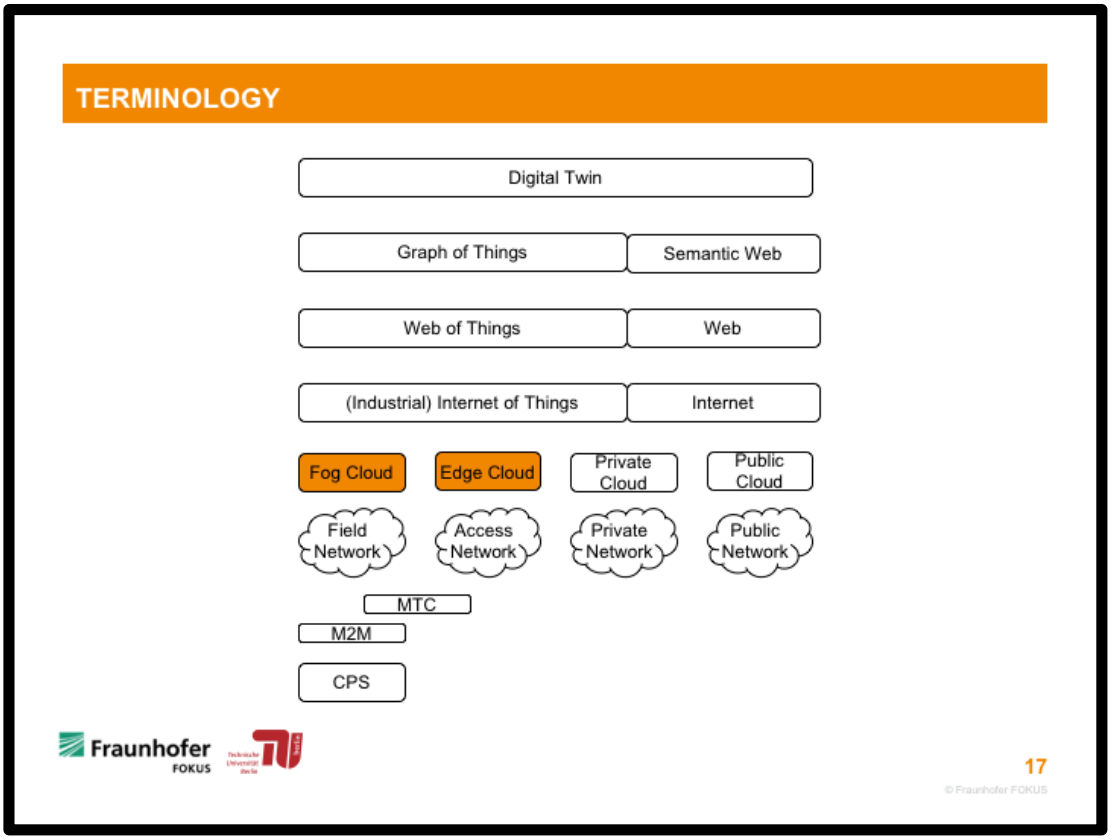
THE LAST LECTURE

Questions?

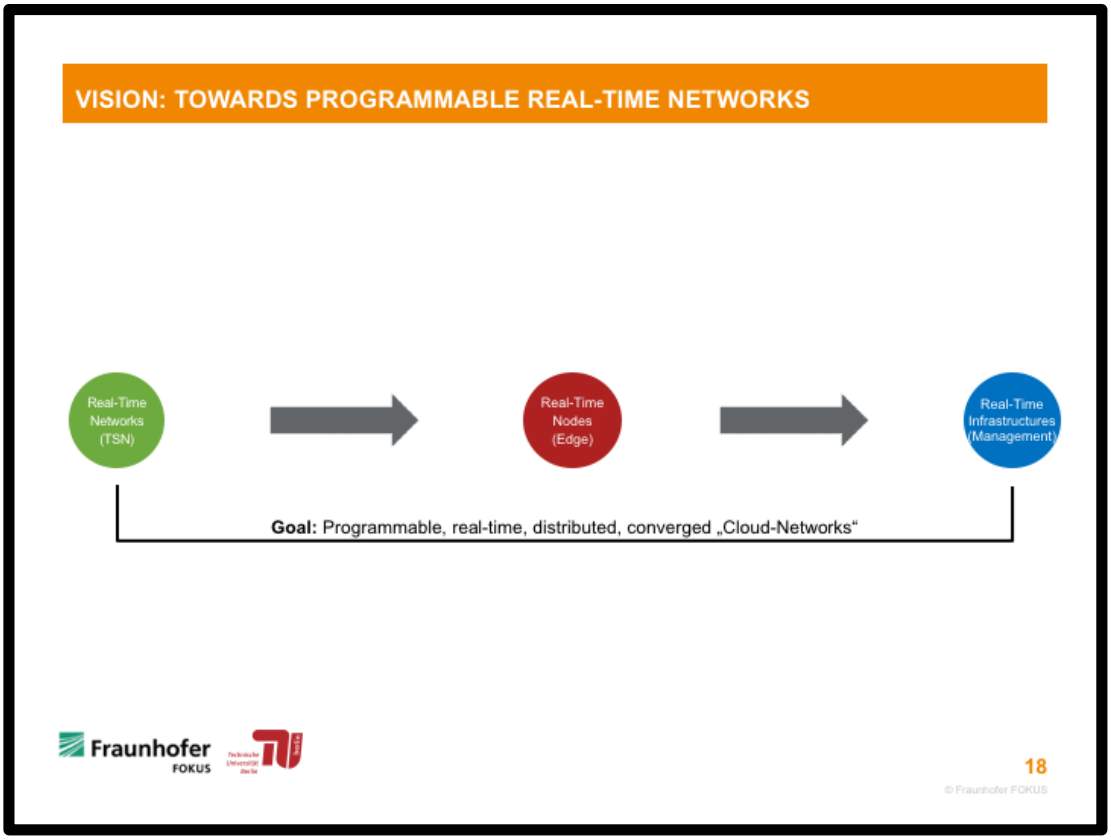
PLAN FOR TODAY

4 minutes

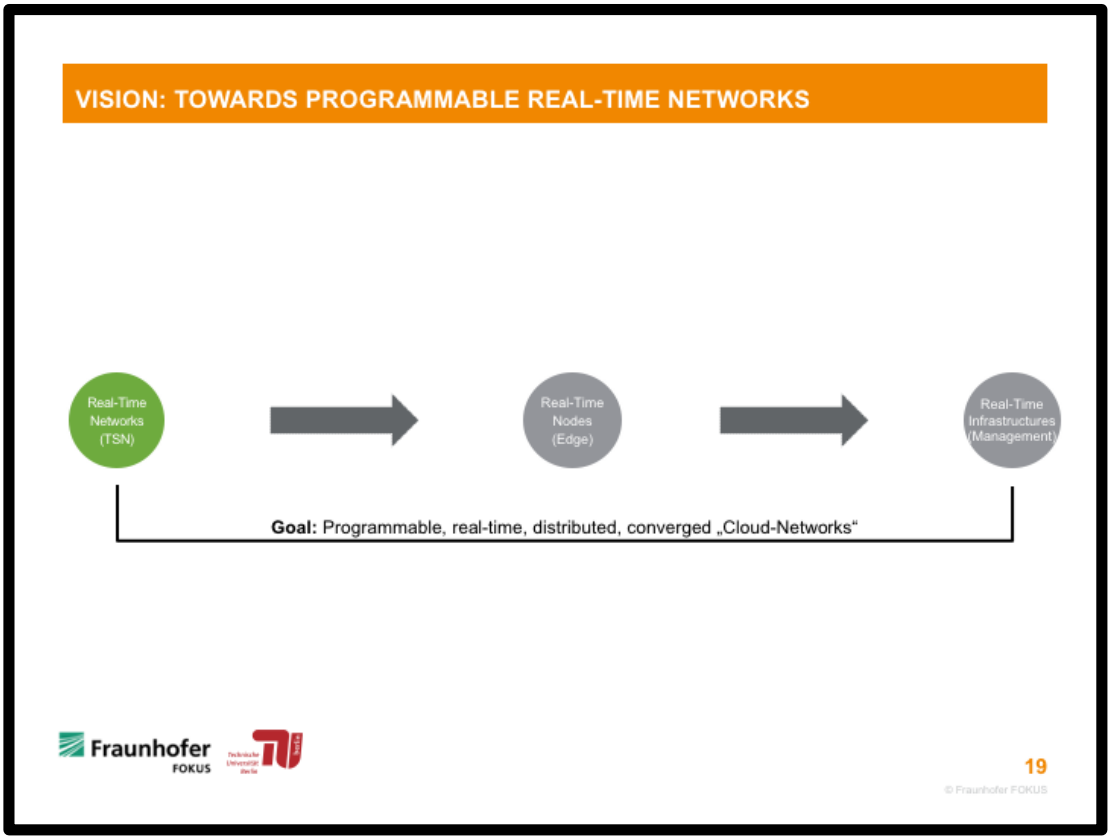
- Finally, the last part of this course



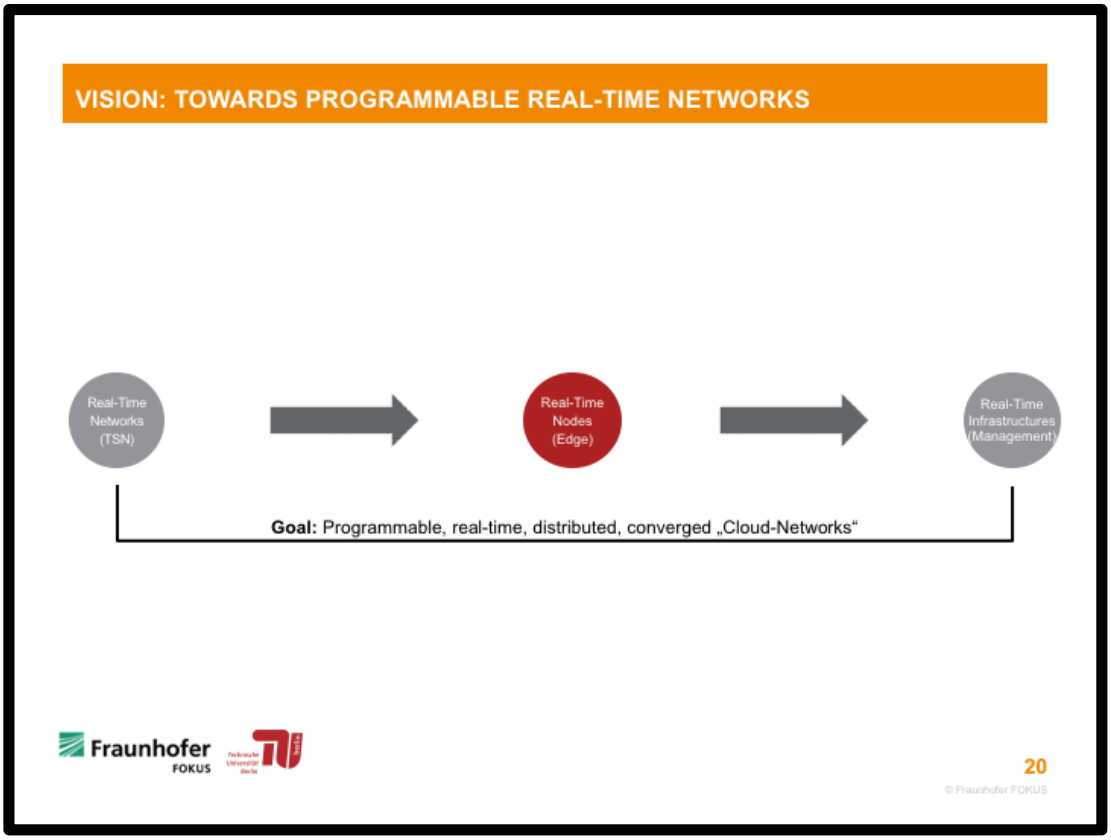
- We're back down at the field and access network
- Why?



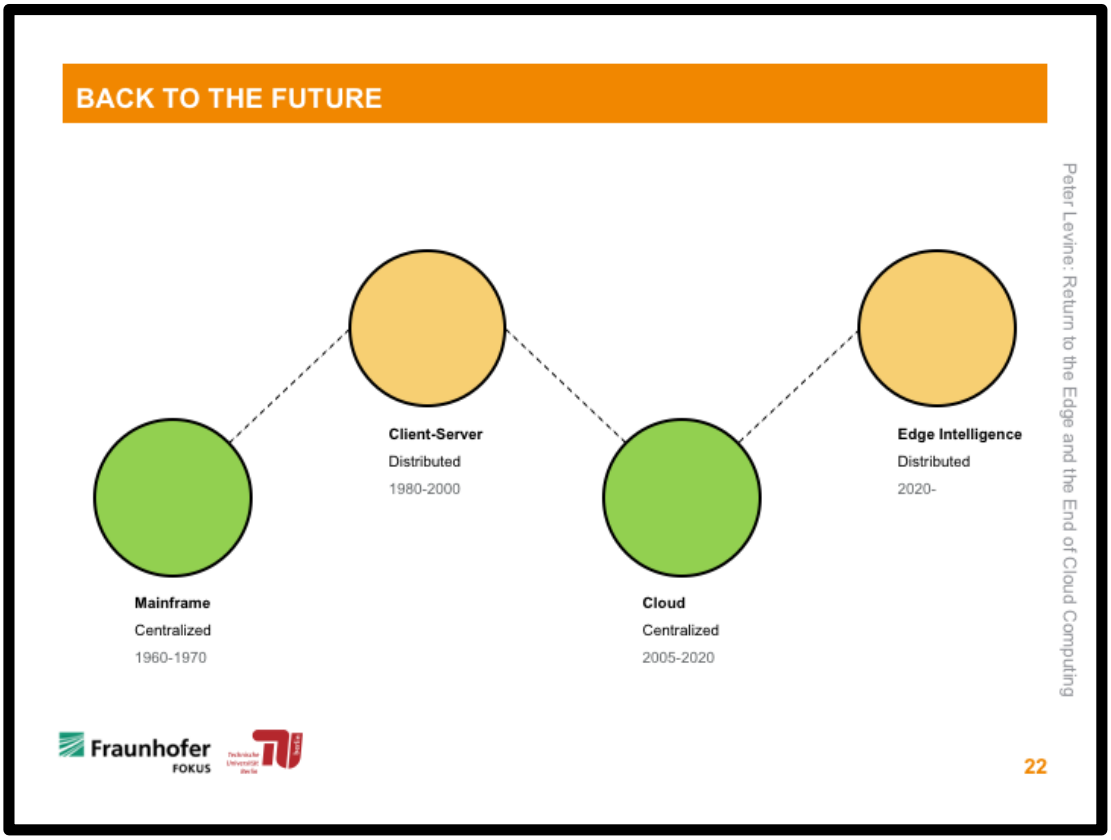
- Every component, the devices, gateways, switches, … have compute capabilities
- Everything can be programmed/managed, including the network
- Key aspect in many industrial use cases is real-time communication



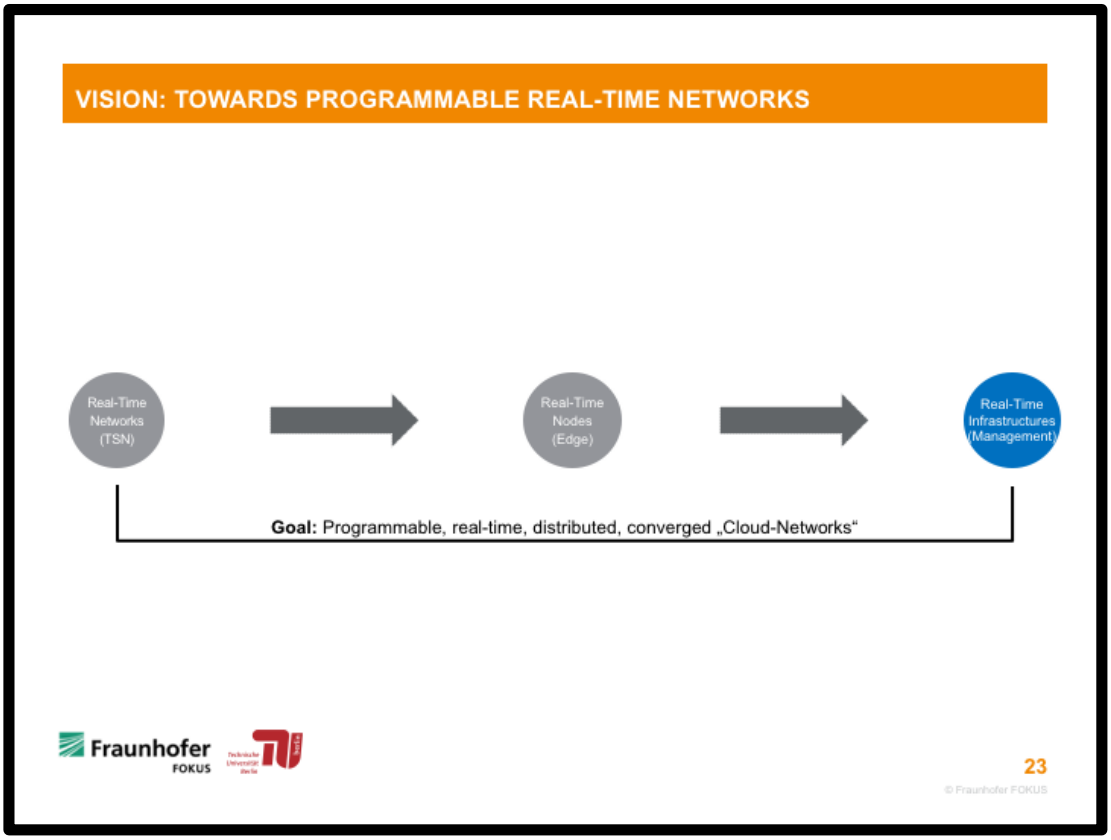
- We have already talked about real-time networks (e.g. TSN)



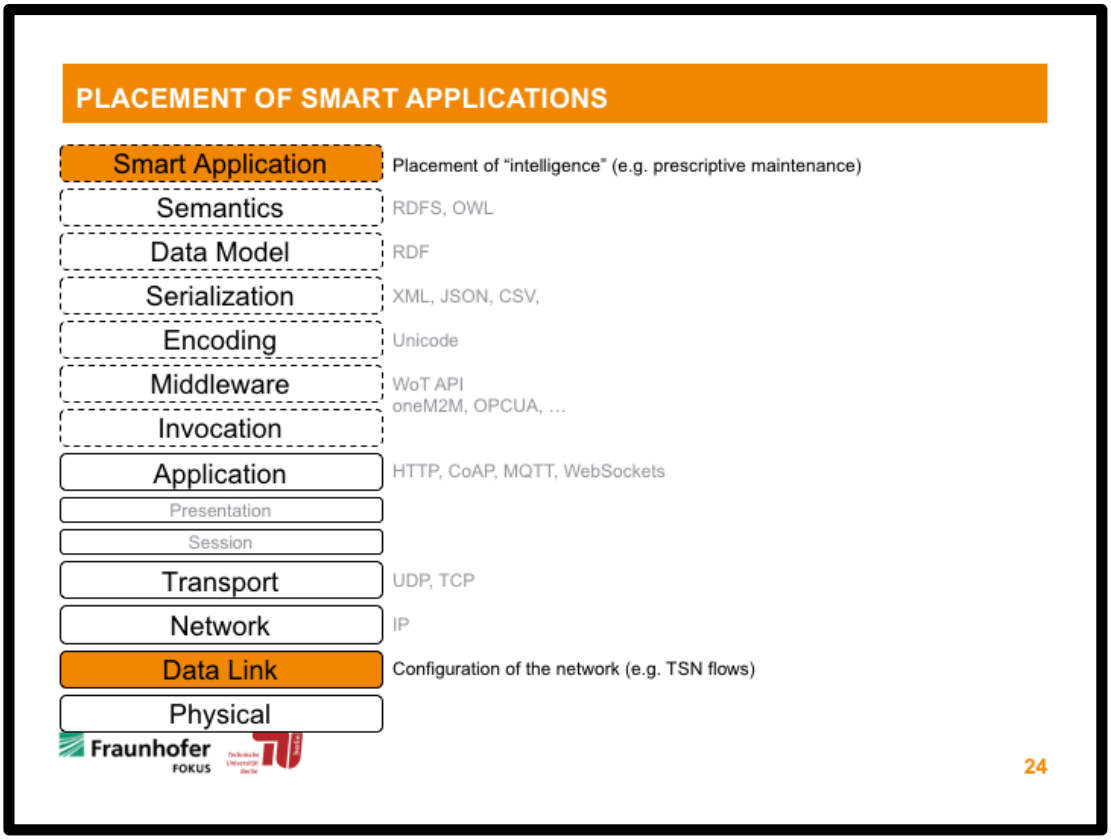
- Next we’ll talk about Real-Time Edge Computing



- Old concept and the next evolution: distributed clouds



- Then we'll talk about real-time infrastructures, i.e. the management/orchestration of functionalities within a network to allow real-time applications



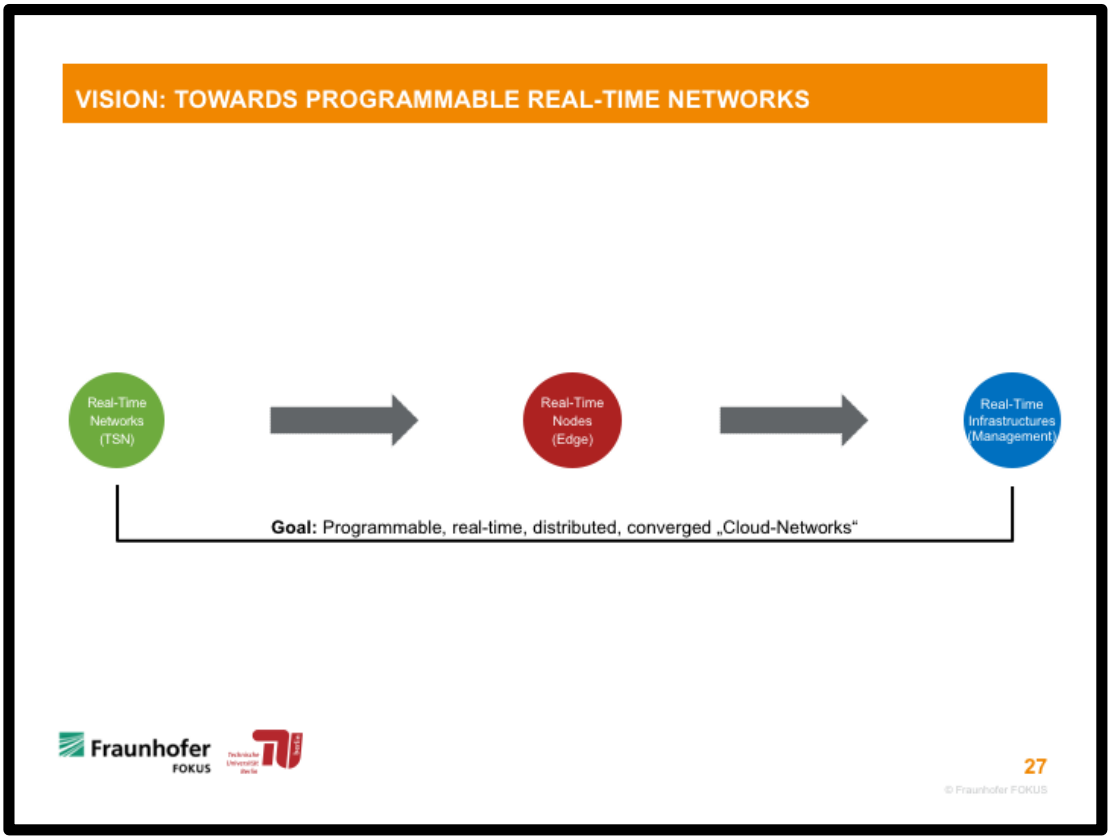
- Orchestration of functionality across the network and configuration of the network is key to enable real-time applications

PLAN FOR TODAY

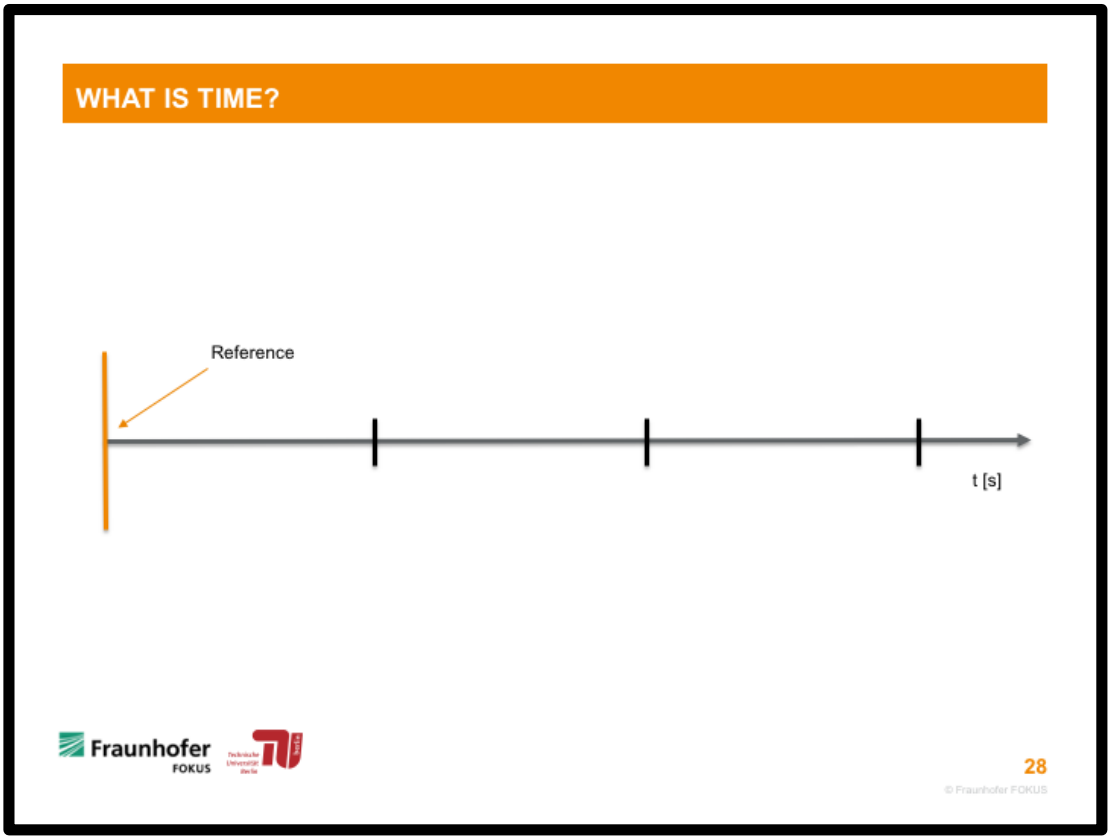
Questions?

REAL-TIME

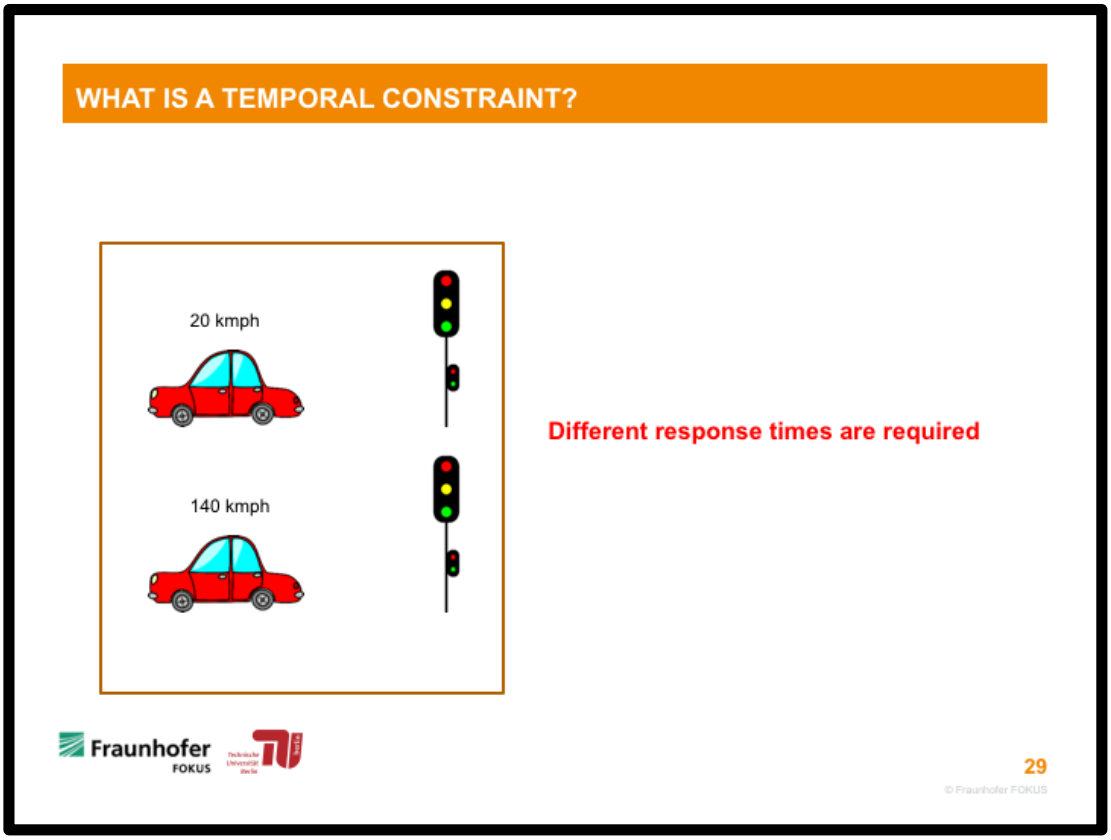
8 minutes



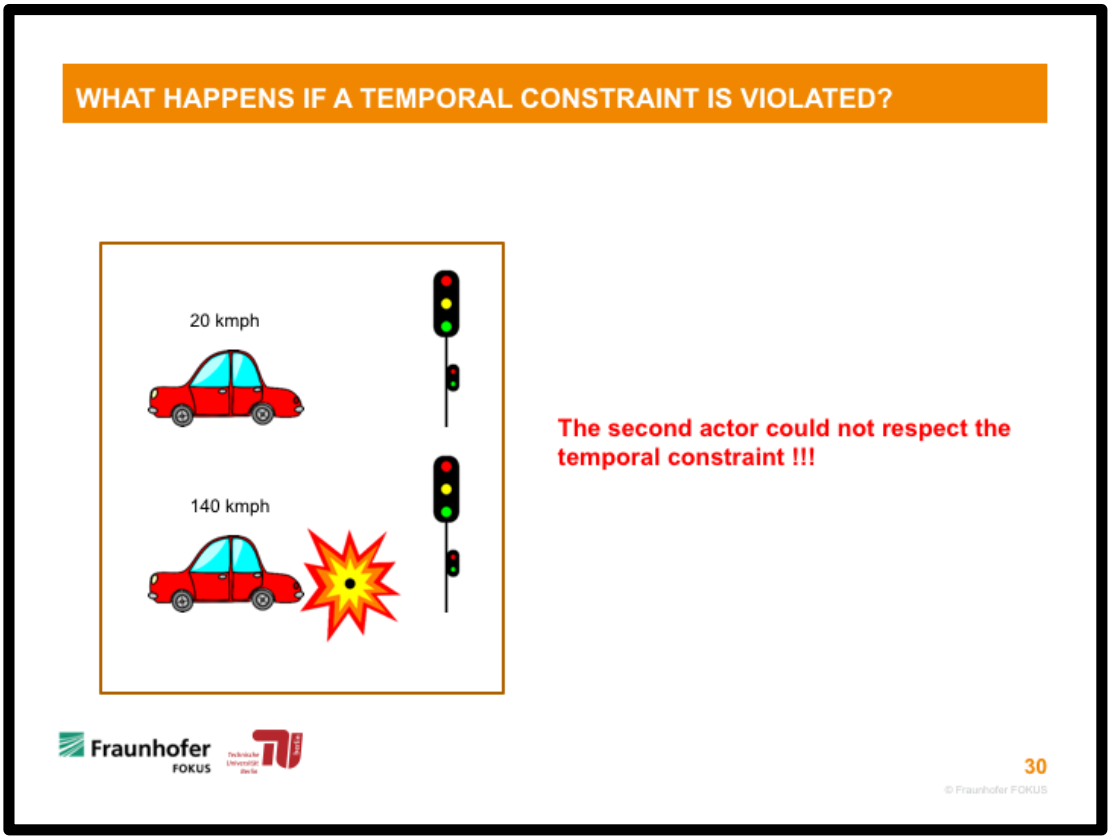
- Every component, the devices, gateways, switches, … have compute capabilities
- Everything can be programmed/managed, including the network
- Key aspect in many industrial use cases is real-time communication
- So let us talk about time



- This might be a more philosophical question.
- Time is: a basic physical quantity, continuous, asymmetric (i.e., moves only in one direction), is immutable (the past is immutable), measured in seconds (s)
- Two types: absolute time (Global reference: E.g. Time since the Big bang, UTC, 16.06.2018), relative time (local reference: E.g. Tomorrow, personal reference)



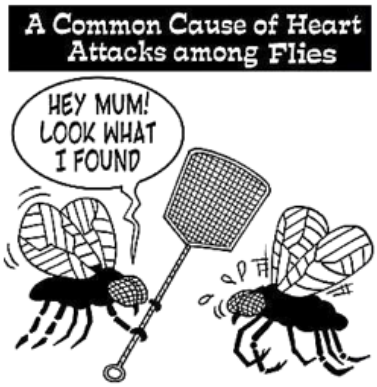
- Constraints obtained due the temporal relationship between two actors.
- The car and the traffic light are actors. Being same actors, the temporal constraints for both situations are different. The lower car is travelling at much higher speed than the top car.
- Considering the drivers, see the traffic light at a known distance and need to stop at the intersection, the response time of the driver to apply brake is much lesser than the one in top car to avoid hitting the intersection.
- This is the temporal constraint introduced as a result of speed of the cars.



- Classic case of violation of temporal constraint.
- The lower car could not react in time or it was too late to come to a safe stop at the intersection. Thus the lower car has violated the temporal constraint and this has led to a catastrophe.

WHAT IS REAL-TIME?

- Faster response time ? NO
- Report an event as it happens ? PROBABLY
- Respect a deadline relevant to the temporal constraint ? YES



Fraunhofer FOKUS

31

- The definition of real time is NOT being fast.
- Real time systems are mandated to complete a task within a deadline, doesn't matter if the execution was finished much earlier than deadline (fast systems).
- Example 1: the tortoise and rabbit side by side are still real time because they are abiding by the temporal constraints set for them.
- Example 2 : Swatting a fly, disturbs the temporal constraints the fly was designed for. However, the fly is quick to respond to a human trying to swat with bare hand, it easily escapes.

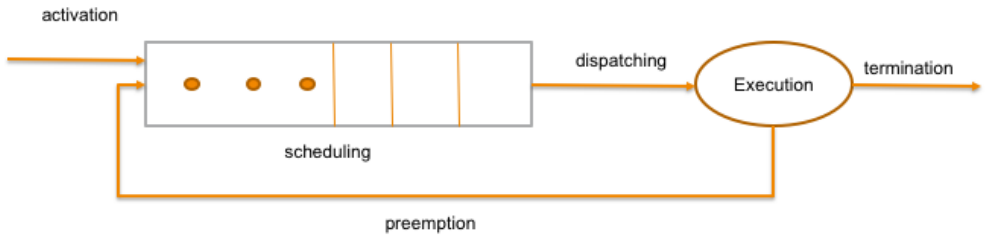
REAL-TIME

Questions?

SCHEDULING

13 minutes

TASK EXECUTION ON AN UNIPROCESSOR (UP) COMPUTING NODE

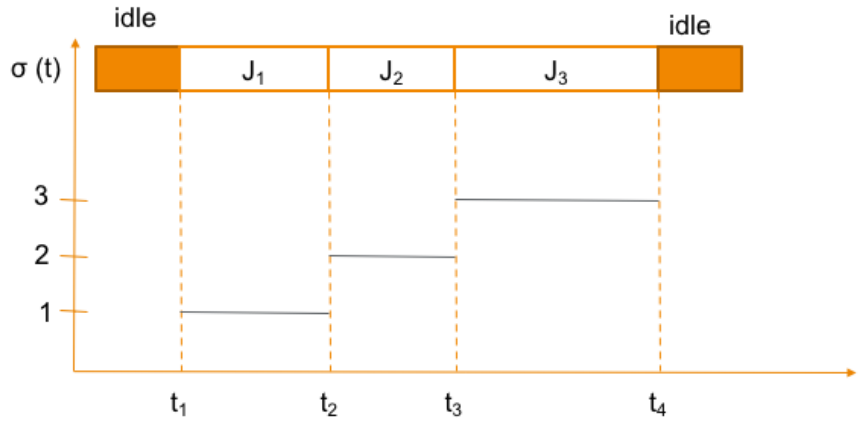


Fraunhofer FOKUS

34

- The task execution in a uniprocessor system. When a task is sent for execution it is in ACTIVATION state and queued. The scheduler is a program which chooses a task from the queue and DISPATCHES it for EXECUTION.
- During the execution of the task, it could be PREEMPTED by the scheduler with another task and sent to the queue to be recalled later by the scheduler. The task after finishing it's execution, it is TERMINATED.

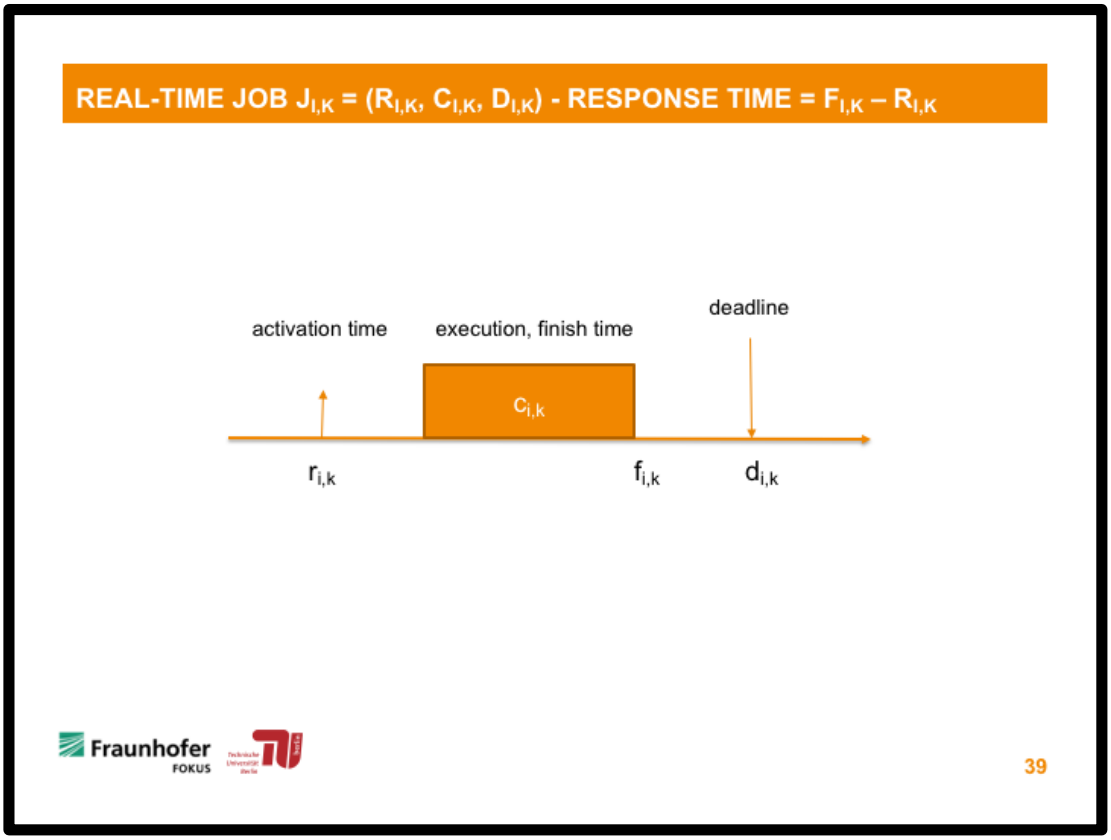
TASK EXECUTION ON AN UNIPROCESSOR (UP) COMPUTING NODE



Fraunhofer FOKUS

35

- σ = sigma
- σ (t) is an algorithm which maps the task number to the time.
- This is the operation of the scheduler.



Response time of job $J_{i,k} = (r_{i,k}, c_{i,k}, d_{i,k})$

- Arrives at time r_i (activation time)
  - Activated by a timer event or an interrupt
- Executes for C_i
- Finishes at time f_i
- Should finish within an absolute deadline d_i

WHAT APPLICATIONS ARE BOUND BY DEADLINES?

- **Hard**: A deadline miss causes a catastrophe and is not affordable to miss. E.g. Mission critical operation.

- **Firm:** A deadline miss does not cause a catastrophe but could damage the system. E.g. robot identifying an obstruction.

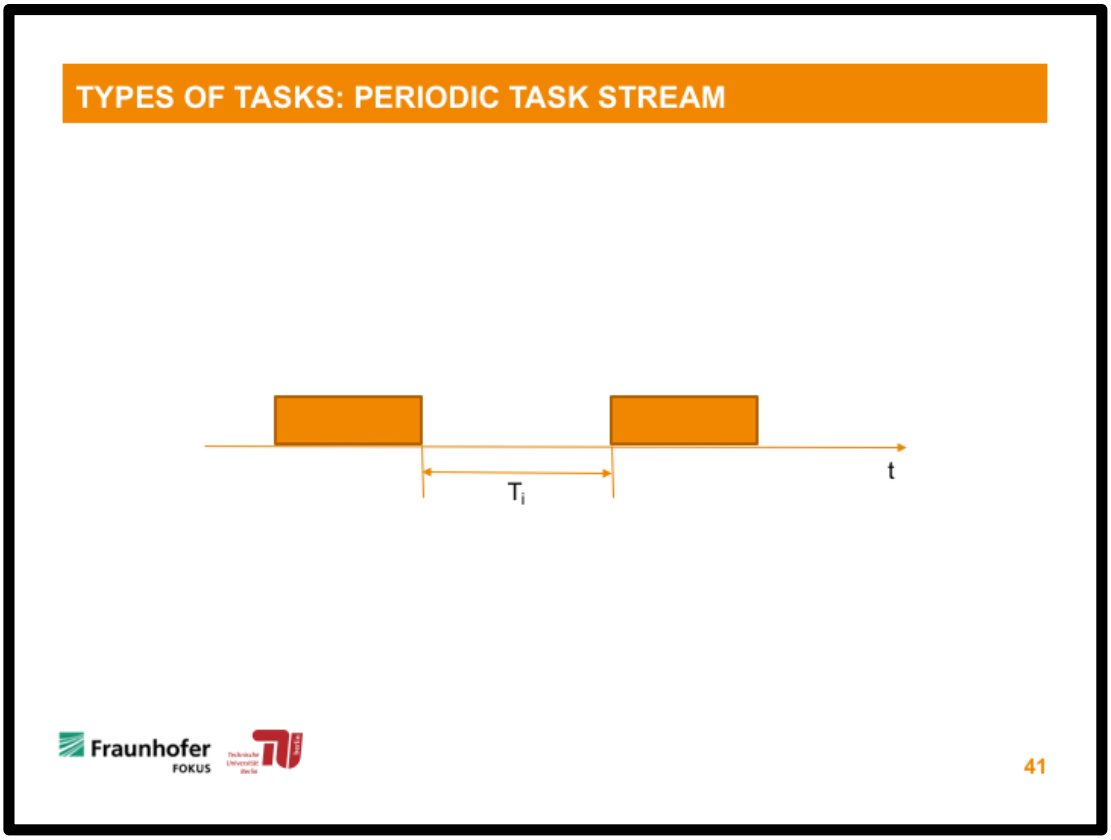
- **Soft:** A deadline miss is tolerated to a certain extent (jitter). E.g. web video.

Fraunhofer FOKUS

40

Depends on the task respecting it‘s deadline:

1. Hard : There is no opportunity to miss even a single deadline, if it misses it leads to a catastrophe. Eg. Shuttle
2. Firm : There is no opportunity to miss even a single deadline, if it misses it may lead to system damage but not a catastrophe. Eg. Robot
3. Soft : It is ok to miss some deadlines in a given interval of time but not too much. Too many missed deadlines can be detectable. Eg. Skype missing some frames in a video.

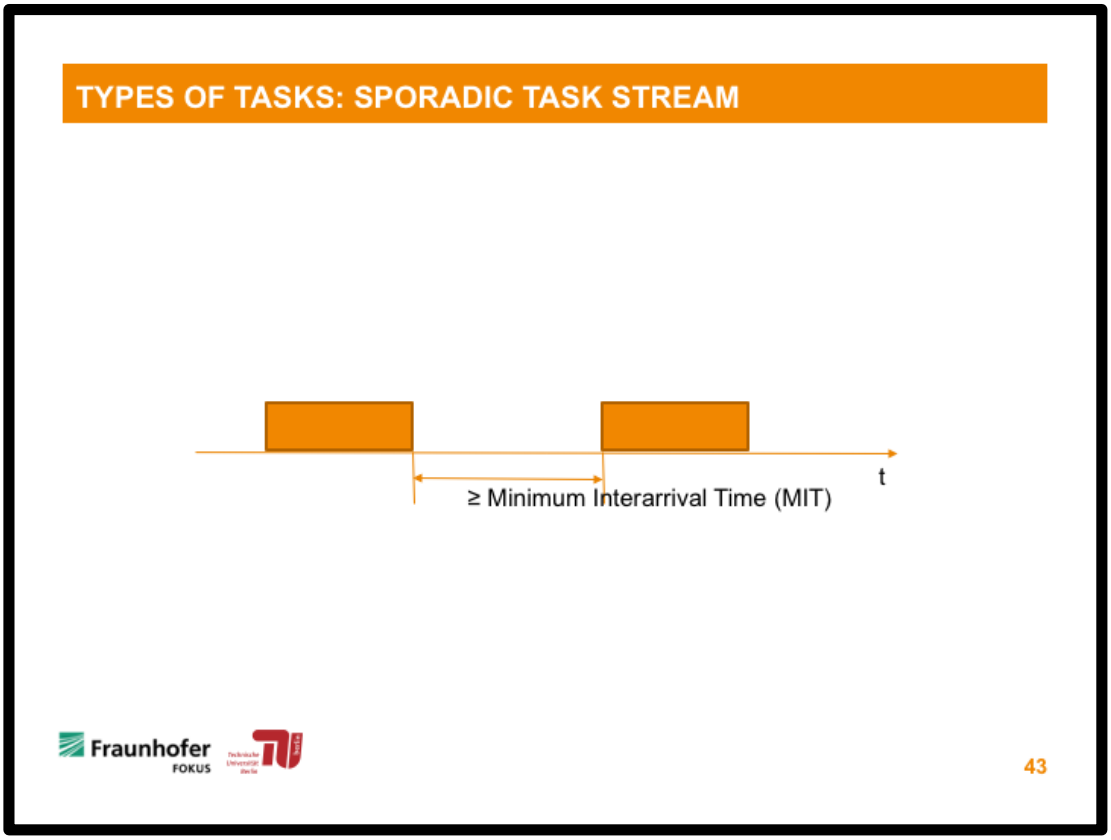


Tasks are not single entities. The same task can occur periodically.

$\tau_i = (C_i, D_i, T_i)$

- $r_{i,k+1} = r_{i,k} + T_i$
- $d_{i,k} = r_{i,k} + D_i$
- $C_i = \max_k \{c_{i,k}\}$

- T_i is task period, D_i is task relative deadline, C_i task **Worst Case Execution Time (WCET)**
- $R_i = \max_k \{\rho_{i,k}\} = \max_k \{ f_{i,k} - r_{i,k} \}$
  - For task to be correctly scheduled, $R_i \leq D_i$



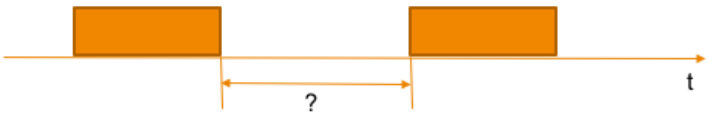
Similar to periodic tasks, however, there is an MIT between the tasks.

$\tau_i = (C_i, D_i, T_i)$

- $r_{i,k+1} \geq r_{i,k} + T_i$
- $d_{i,k} = r_{i,k} + D_i$
- $C_i = \max_k \{c_{i,k}\}$

- T_i is Minimum Inter-Arrival Time (MIT), D_i is task relative deadline,
  C_i task **Worst Case Execution Time (WCET)**
- $R_i = \max_k \{\rho_{i,k}\} = \max_k \{ f_{i,k} - r_{i,k} \}$
  - For task to be correctly scheduled, $R_i \leq D_i$

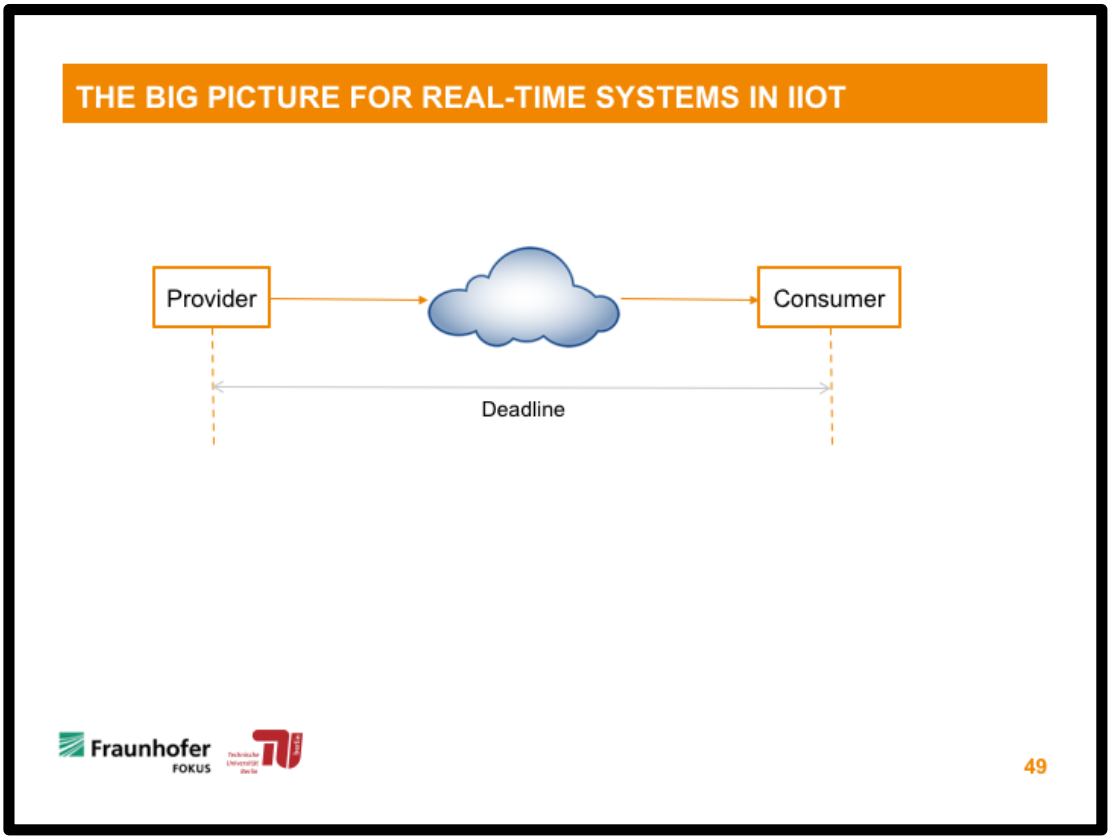
TYPES OF TASKS: APERIODIC TASK STREAM



Fraunhofer FOKUS

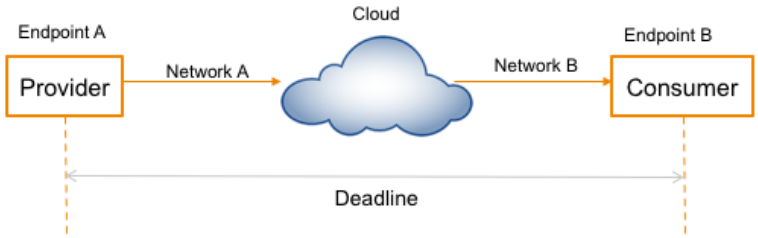
44

- No specific model
- Tasks might arrive in burst
- Equip the system to handle such loads by providing margins



- The big picture on an industry shop floor
- The provider and consumer of information communicate over a cloud
- The final goal is to fix a deadline for the communication from provider to consumer.

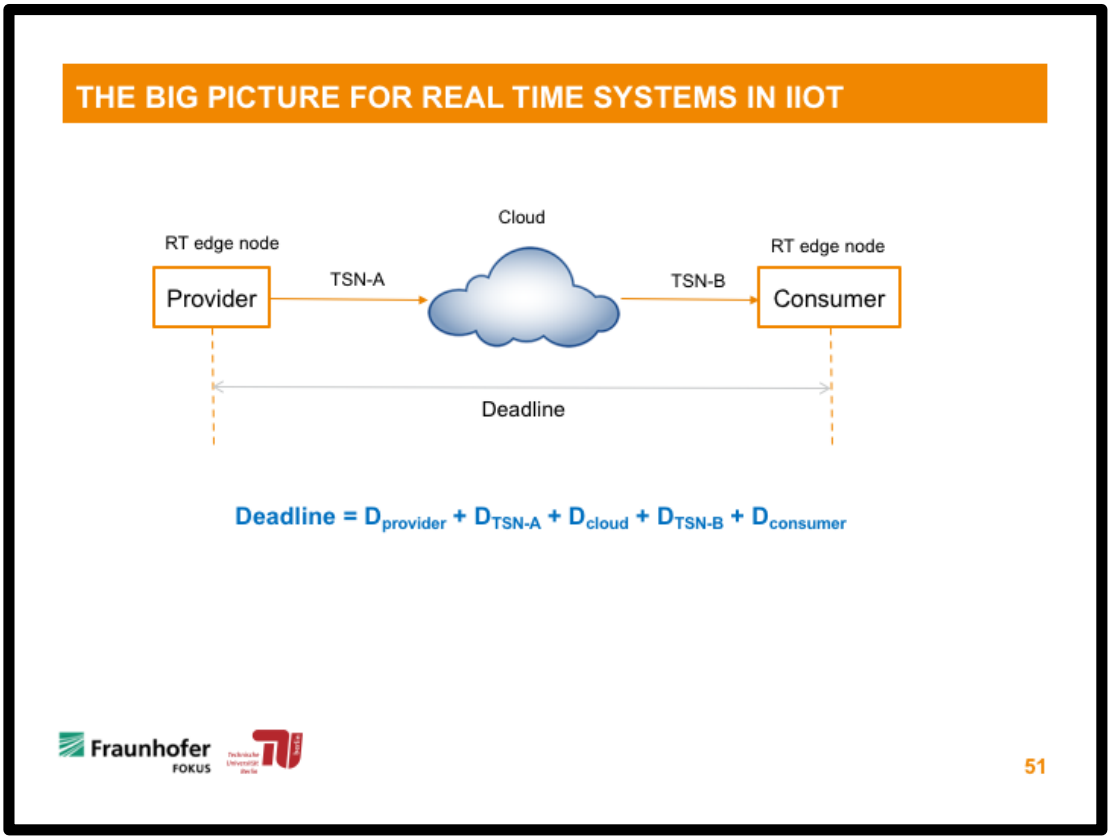
THE BIG PICTURE FOR REAL TIME SYSTEMS IN IIOT



Fraunhofer FOKUS

50

- The final deadline is sum of all the deadlines of the internal components.



- TSN switches: real time systems where the tasks forward incoming network packets to the right port
- Edge nodes: deadlines are associated with tasks such as linking control with sensor information (control loop)

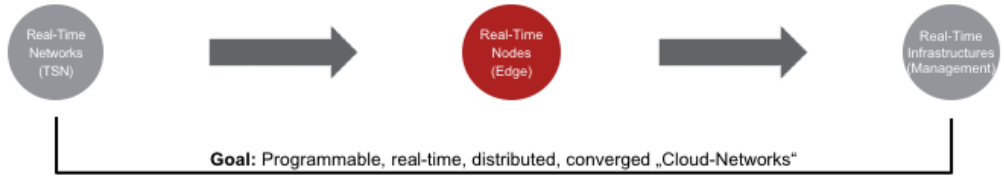
References:

1. Lecture notes from Luca Abeni, University of Trento, Italy: http://www.dit.unitn.it/~abeni/RTOS/
2. G. C. Buttazzo, Hard Real Time Computing Systems: Predictable Scheduling Algorithms and Applications. 2008.

EDGE COMPUTING

14 minutes (incl. questions)

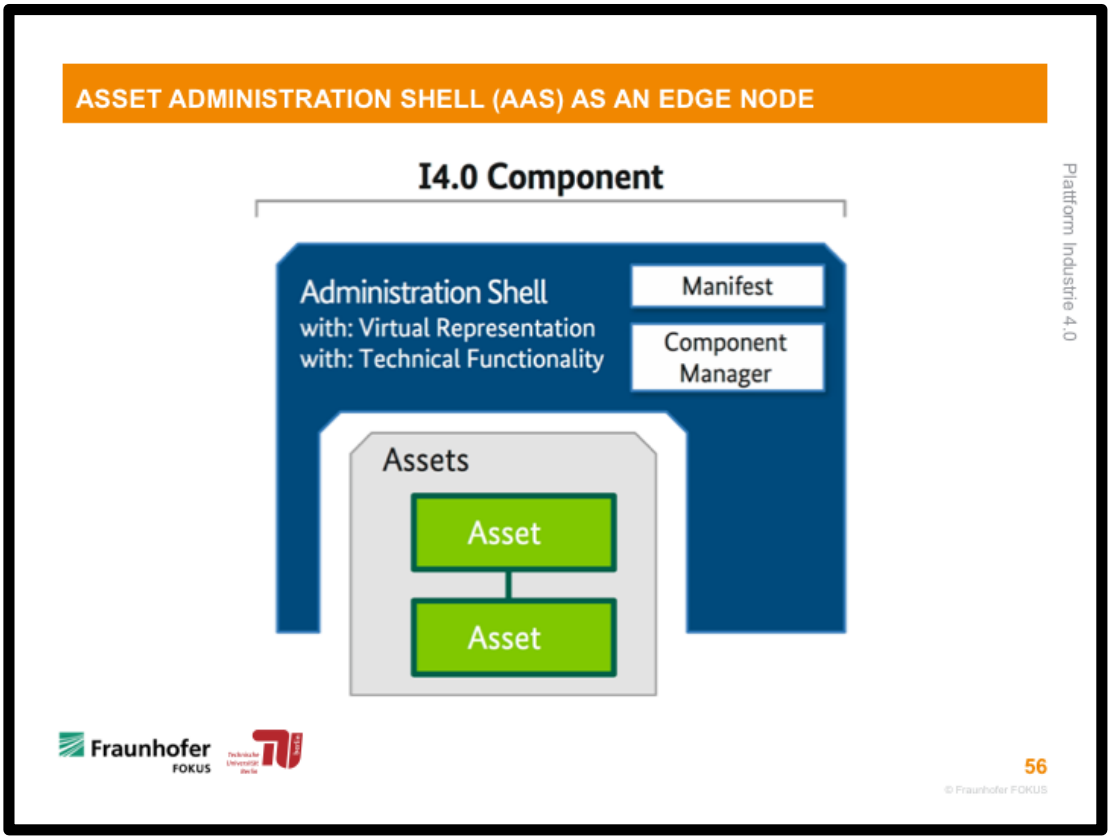
VISION: TOWARDS PROGRAMMABLE REAL-TIME NETWORKS



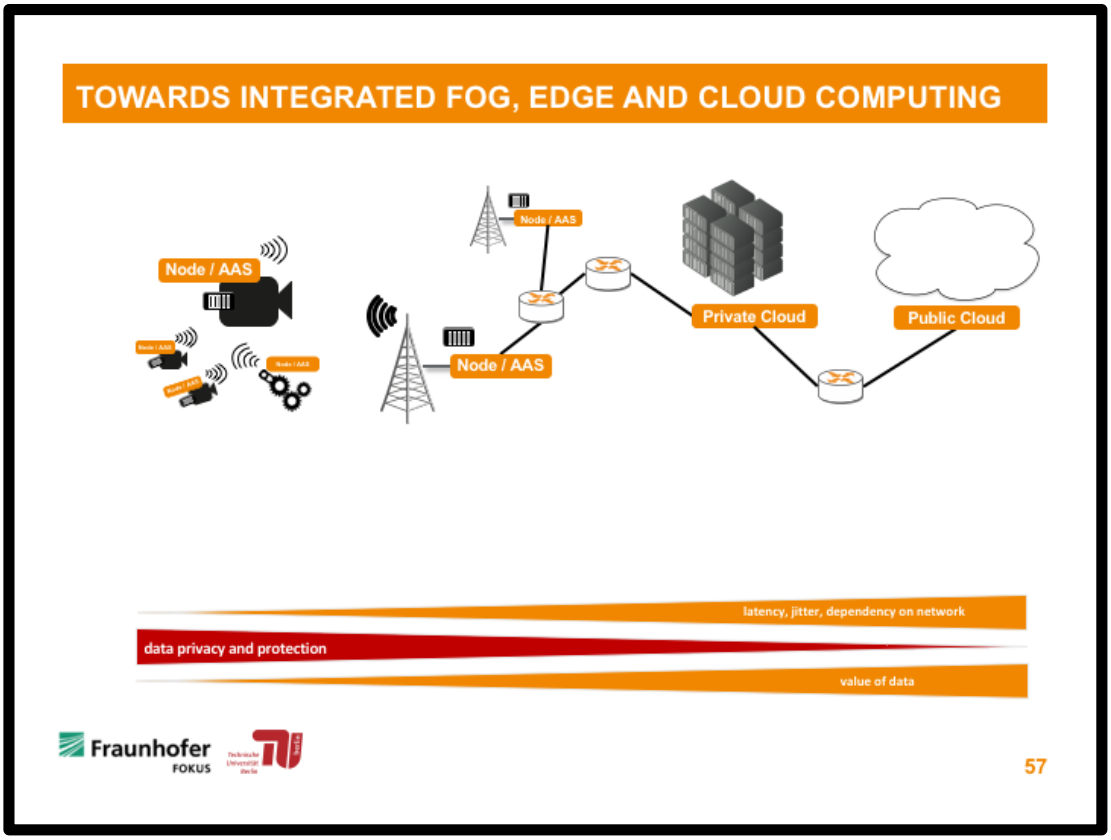
Goal: Programmable, real-time, distributed, converged „Cloud-Networks“

Fraunhofer FOKUS

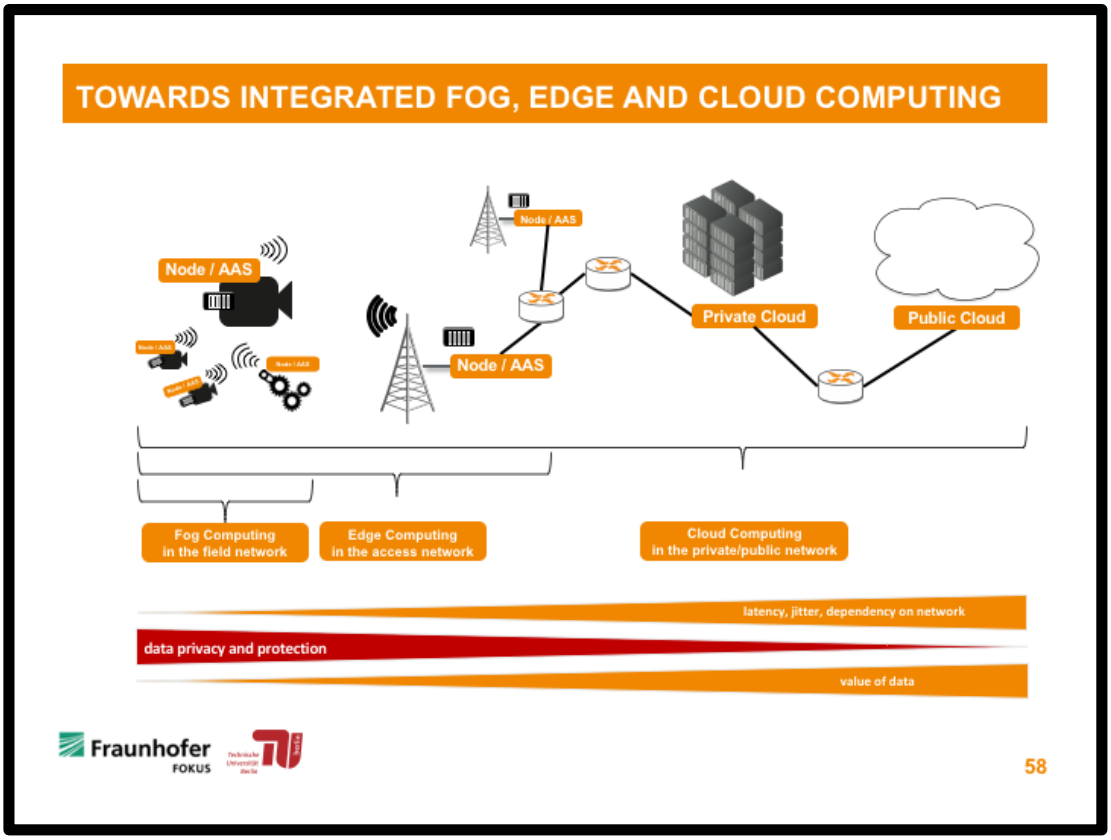
54



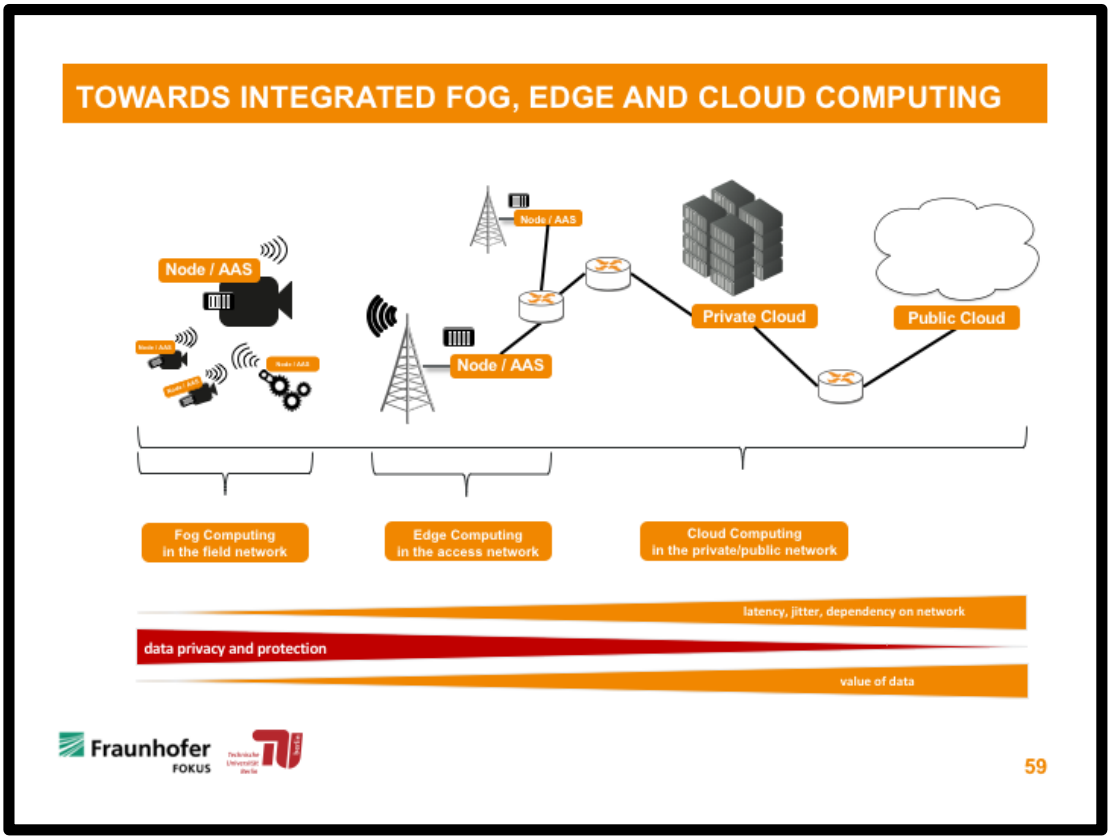
- The AAS can be seen as an Edge node that runs applications



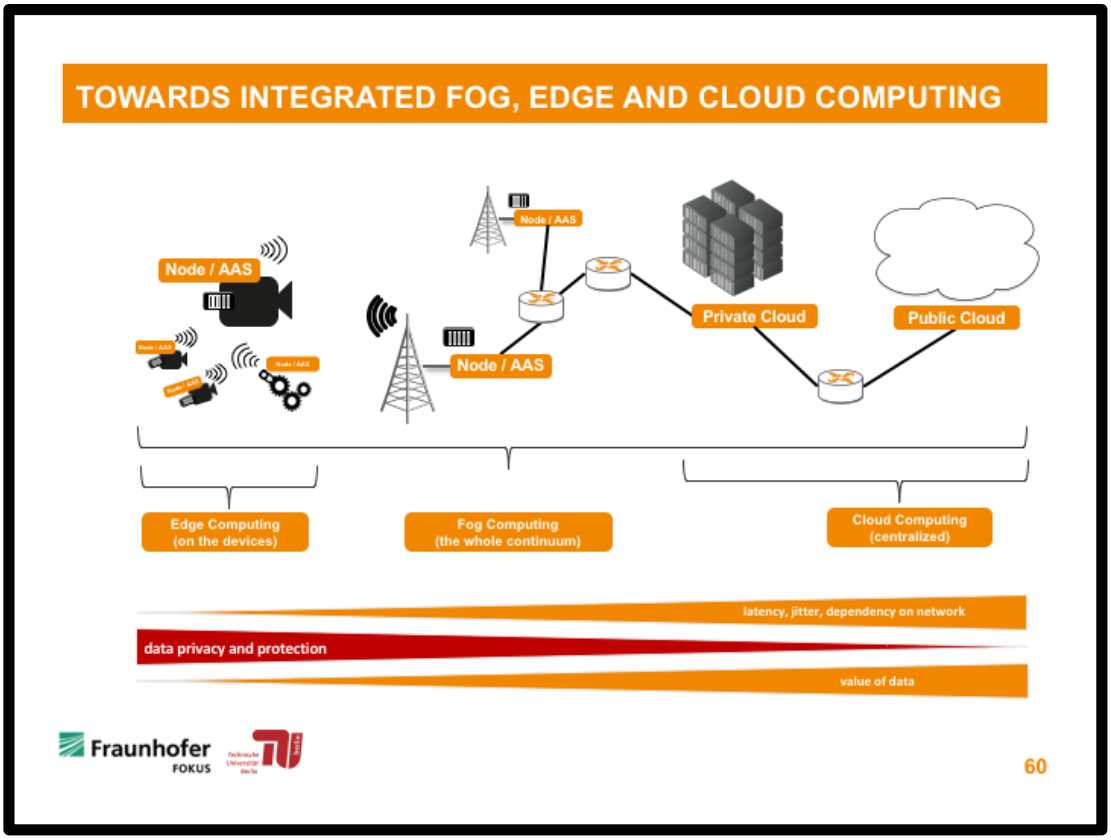
- Main advantages: low latency/jitter/network dependency, higher data privacy/protection, value of data depends



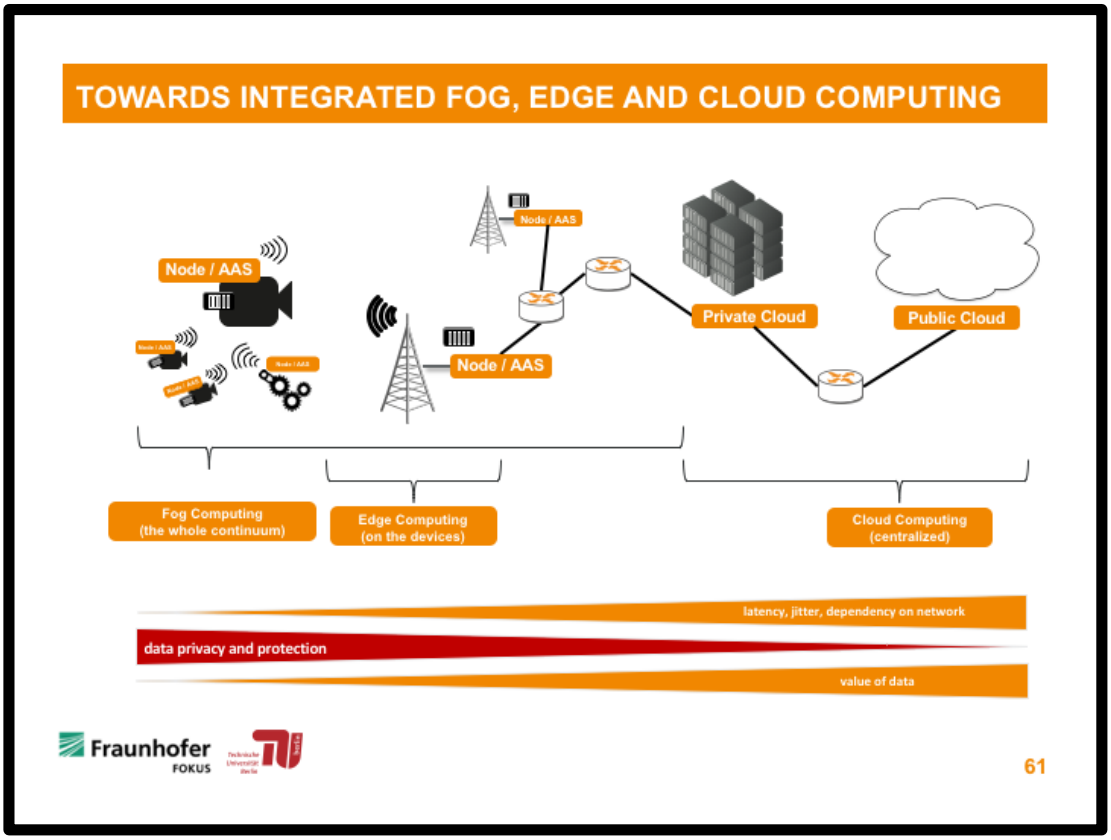
- The terminology differs widely!



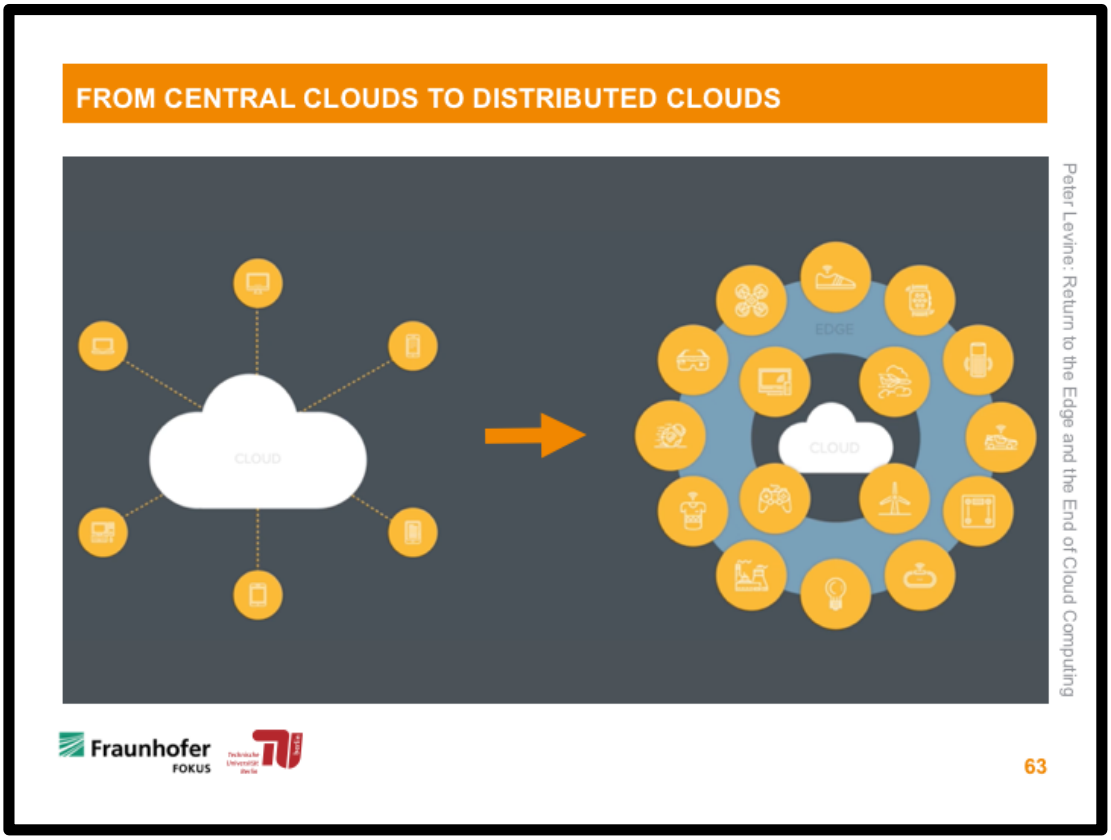
- The terminology differs widely!



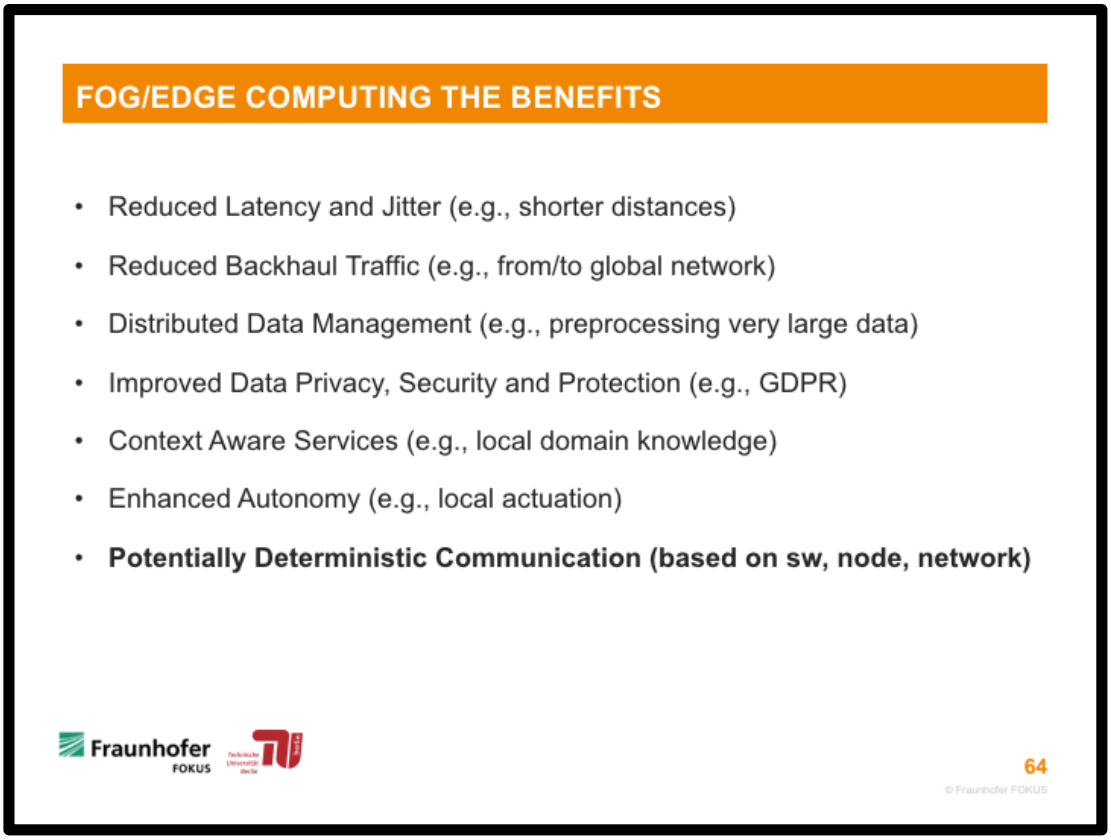
- The terminology differs widely!



- The terminology differs widely!



- The specific terminology doesn't matter, the core idea is to distribute computing/storage/intelligence/…



- GDPR = General Data Protection Regulation
- By bringing computing power to the surrounds of those consuming services and applications, Edge Computing can bring advantages not possible to get by using solely Cloud.
- For example: deterministic communication

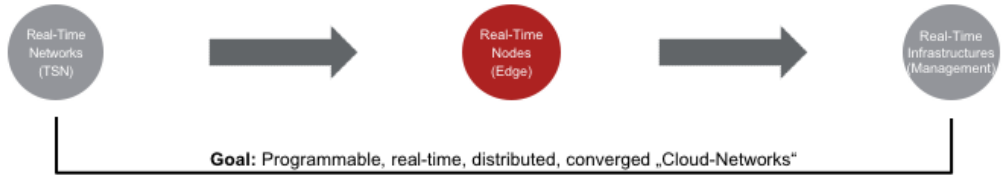
EDGE COMPUTING

Questions?

REAL-TIME EDGE COMPUTING

7 minutes

VISION: TOWARDS PROGRAMMABLE REAL-TIME NETWORKS

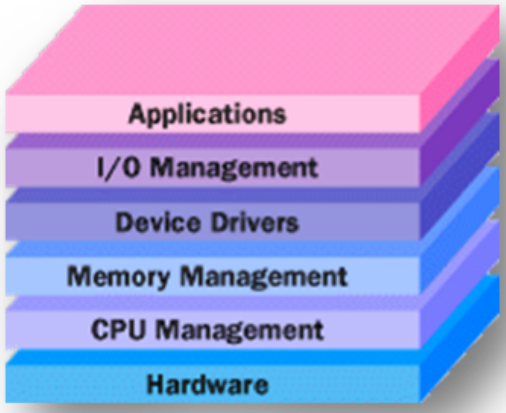


Goal: Programmable, real-time, distributed, converged „Cloud-Networks“

Fraunhofer FOKUS

70

OPERATING SYSTEM: RESOURCE (INCL. TASK) MANAGEMENT

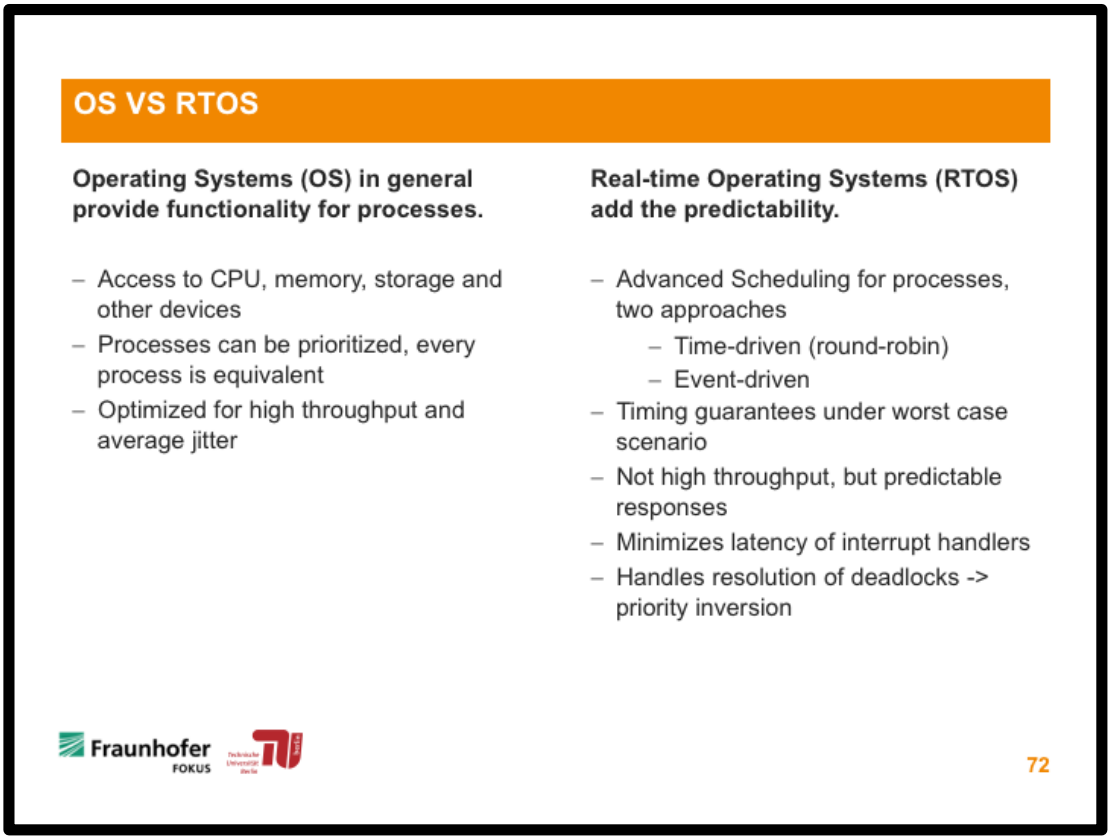


https://ils.unc.edu/courses/2014_spring/inls261_001/sessions.01.03a.servers.UNIX.html

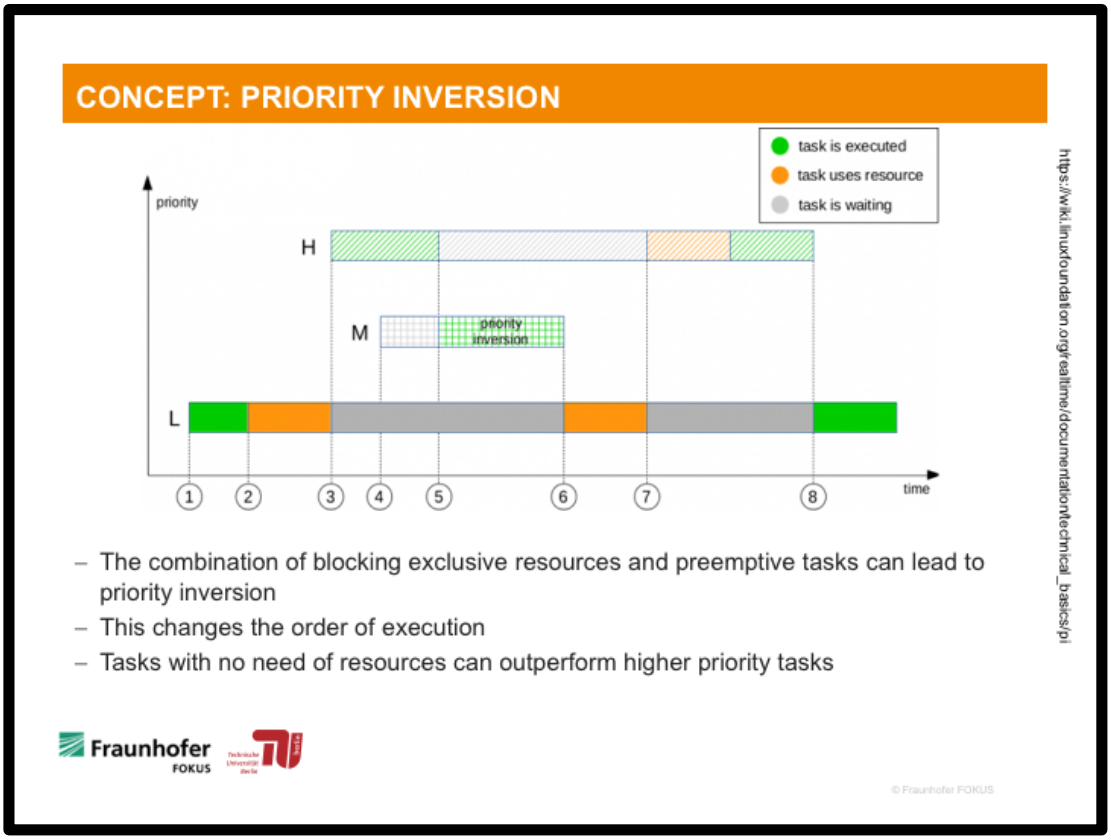
Fraunhofer FOKUS

71

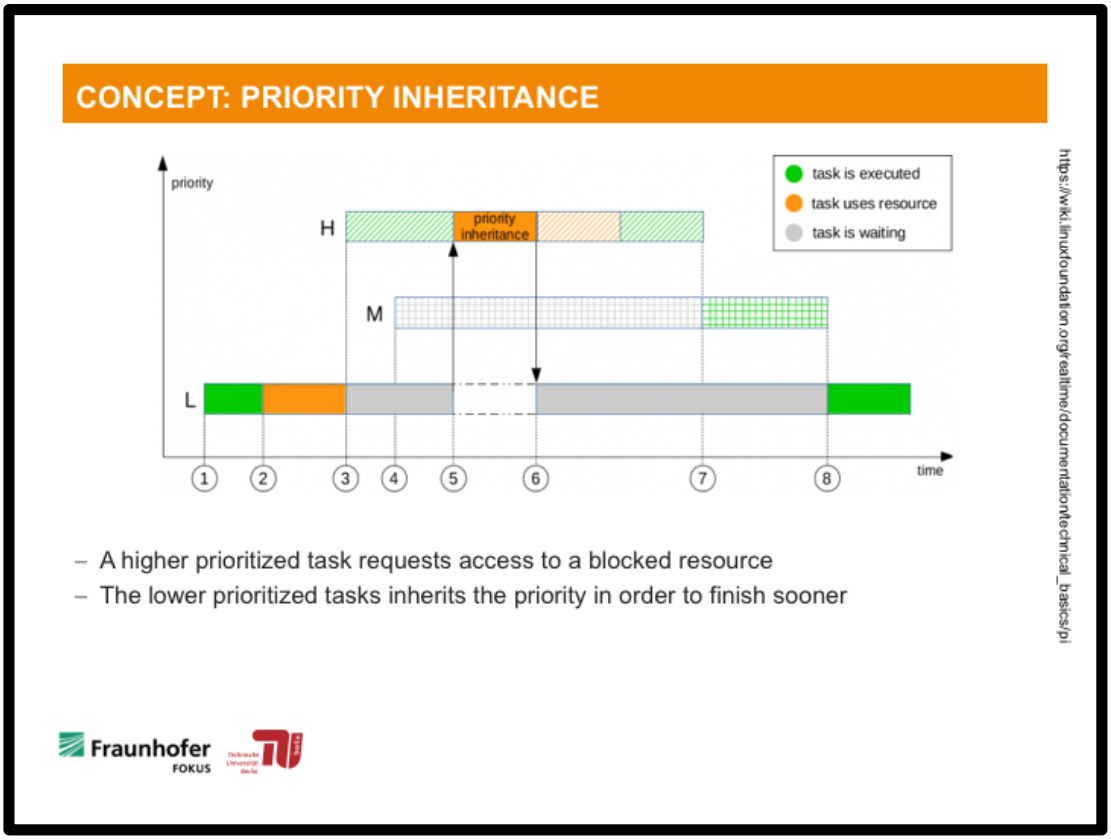
- Operating Systems manage how tasks on a node a scheduled



- Basically two types scheduling mechanisms: RT and non-RT
- Time-Driven will switch every task after a specific time, even when no switch was necessary
- Event-driven will switch, when an event of higher priority occurs or the task has finished



- A task with low priority L becomes runnable and starts executing (1).
- It acquires a mutually exclusive resource (2).
- Now a task with a higher priority H becomes runnable and preempts L while L is holding the resource (3).
- A third task M with priority between Hs and Ls priority (and with no need for the resource) becomes runnable, but it has to wait because H with higher priority is running (4).
- H needs the resource still held by L (5), and so H stops running to wait until the resource is released.
- The runnable task M prevents L from executing because of its higher priority. This leads to the priority inversion because H has to wait until M has finished (6) so that L can release the resource (7).

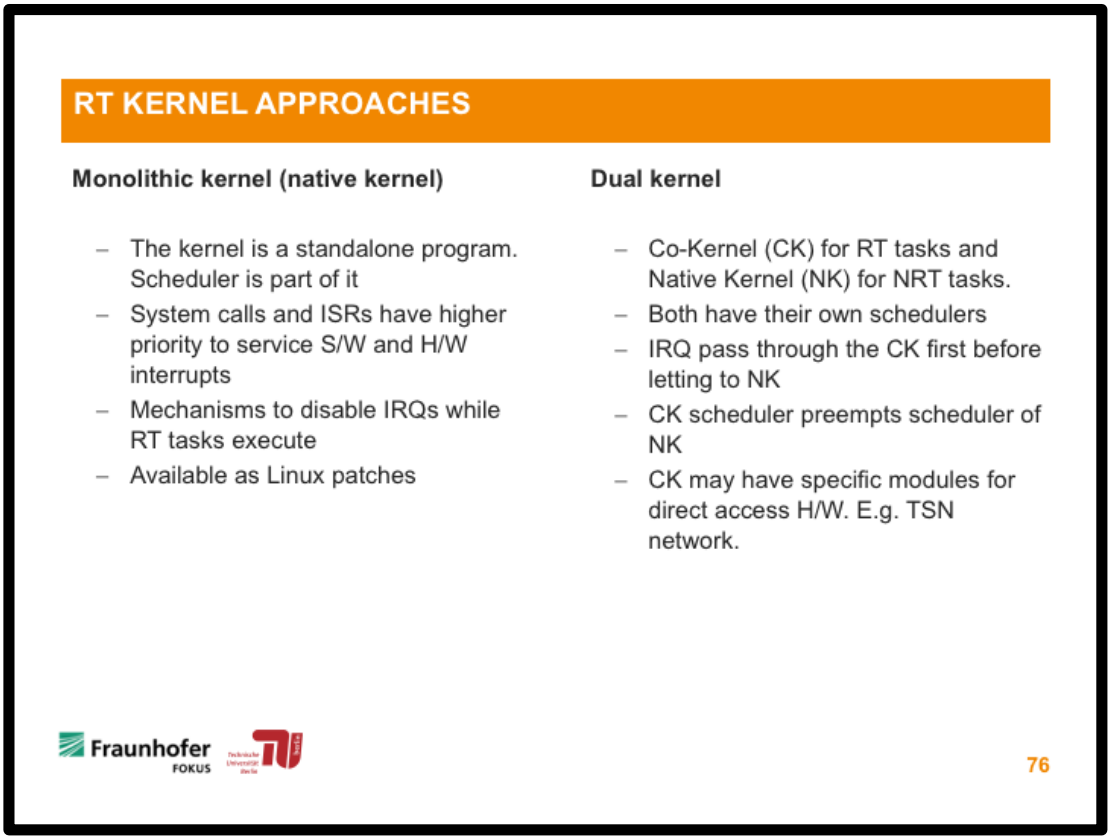


- This means than when H needs the resource held by L, L inherits H's priority (5) in order to release the resource sooner (6). When M gets ready to run, M has to wait until the currently higher prioritized task L releases the resource and H finishes running (7).
- Other solutions for the inversion:
  - Disabling all interrupts -> critical sections can not be interrupted
  - Priority ceiling -> processes in critical sections get the highest priority temporally
  - Random boosting -> tasks that hold resources get randomly boosted in their priority
  - Avoid blocking

CHALLENGE: APPLICATION DEVELOPMENT

- Enabling real-time for the application -> start Real-Time Scheduler thread
- Not using dynamic memory allocation -> plan used data and data types in advance
- Using the correct ways to use resources provided by the OS -> mutex, semaphore, etc.
- Specify Worst Case Execution Time of sub tasks -> for correct scheduling
- What happens in error cases -> error handling
- How many applications can be handled by the system -> load in systems for aperiodic task streams

Fraunhofer FOKUS



- ISR is Interrupt Service Routine used to service Interrupt Request (IRQ). IRQ can be raised by S/W or H/W. S/W raised interrupts are called sort-interrupts H/W raised interrupts are called interrupts in general.
- Example for soft interrupt is user raised System Call. Example for hardware interrupts are the one raised by NIC cards to service the block of messages it may have received.
- Dual and Micro-kernels (User chooses the only needed functionalities) provide a freedom and flexibility to work with H/W and S/W to plan and manage RT task execution.

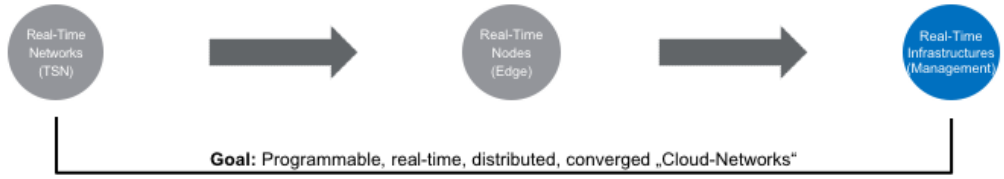
REAL-TIME EDGE COMPUTING

Questions?

REAL-TIME ORCHESTRATION

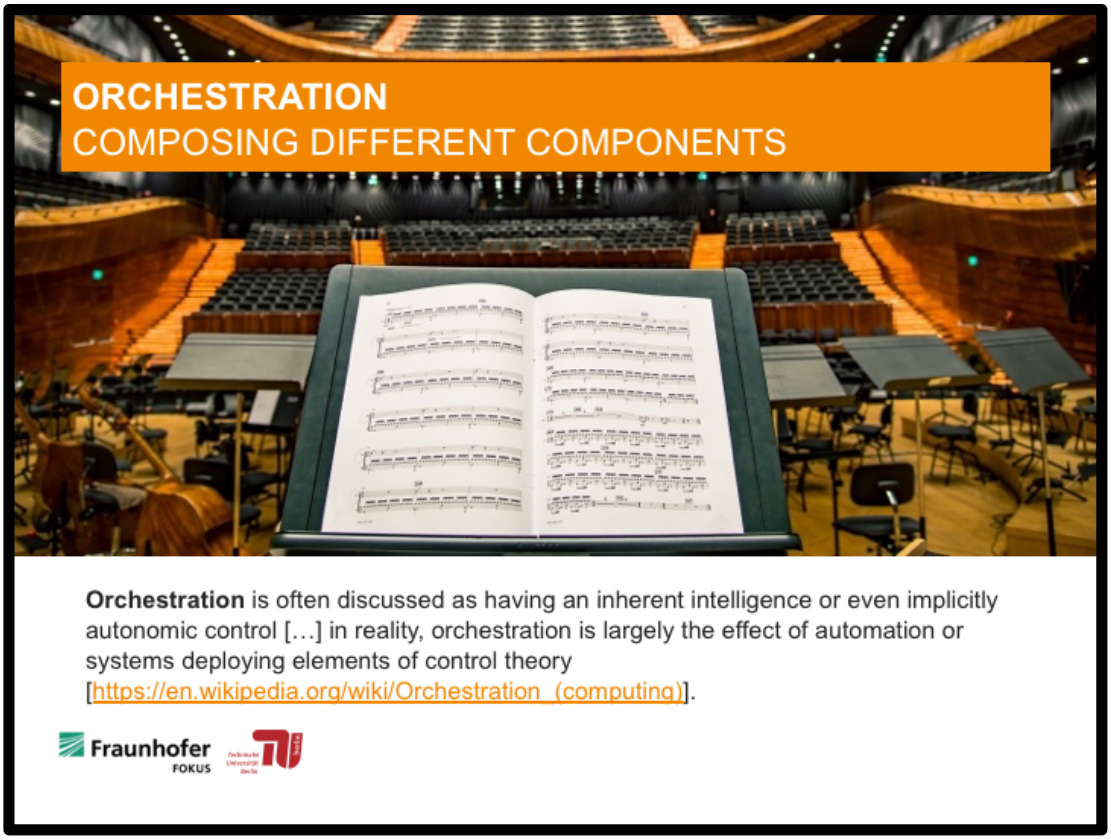
15 minutes

VISION: TOWARDS PROGRAMMABLE REAL-TIME NETWORKS



Goal: Programmable, real-time, distributed, converged „Cloud-Networks“

Fraunhofer FOKUS



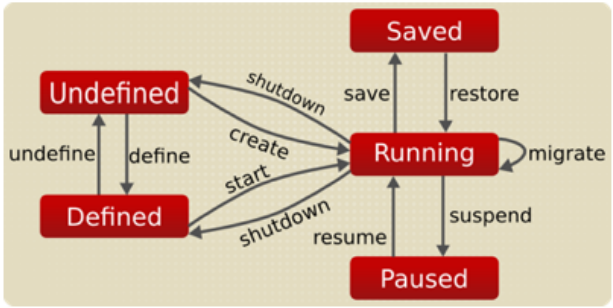
ORCHESTRATION
COMPOSING DIFFERENT COMPONENTS

Orchestration is often discussed as having an inherent intelligence or even implicitly autonomic control […] in reality, orchestration is largely the effect of automation or systems deploying elements of control theory [https://en.wikipedia.org/wiki/Orchestration_(computing)].

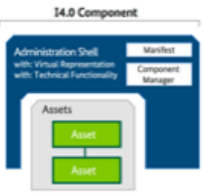
Fraunhofer FOKUS

LIFE CYCLE MANAGEMENT

- **Status**
  - Create
  - Start
  - Save
  - Migrate
  - Restore
  - Suspend
  - Shutdown
  - etc.
- **Type**
  - Virtual Machine (VM)
  - Container
  - RT Application
  - TSN Stream
  - …



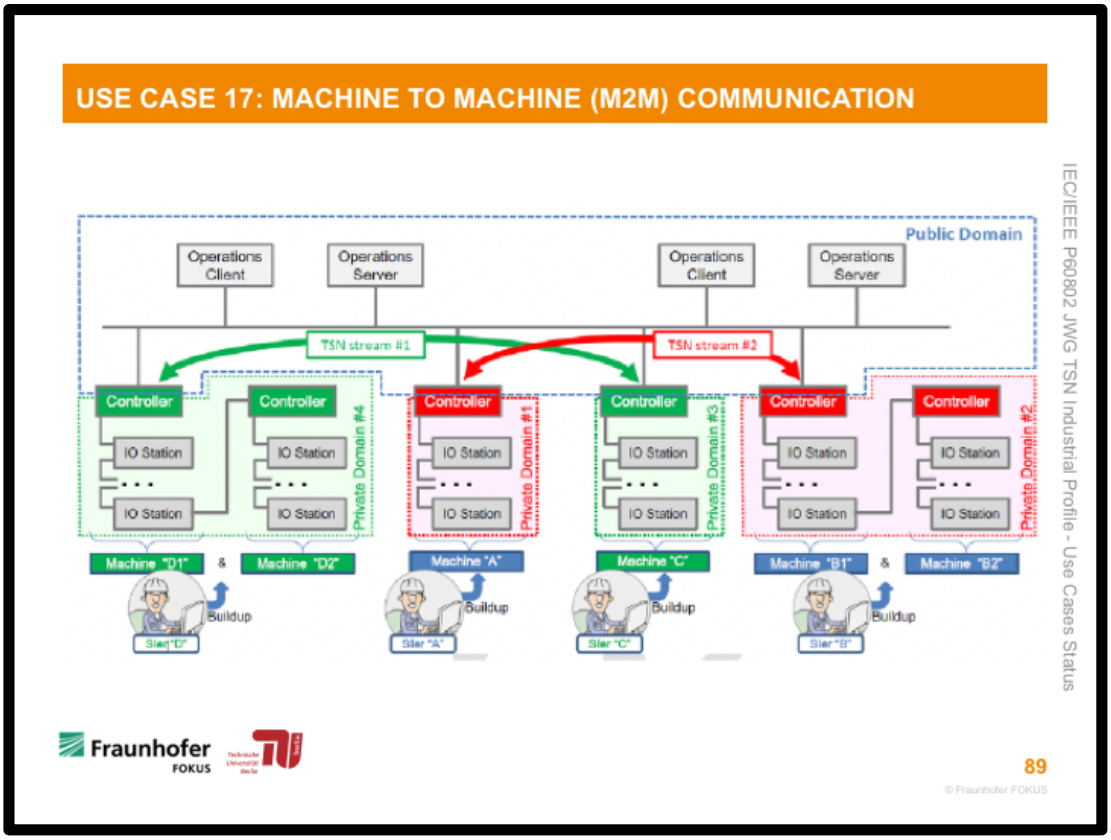
Source: http://wiki.libvirt.org/page/VM_lifecycle



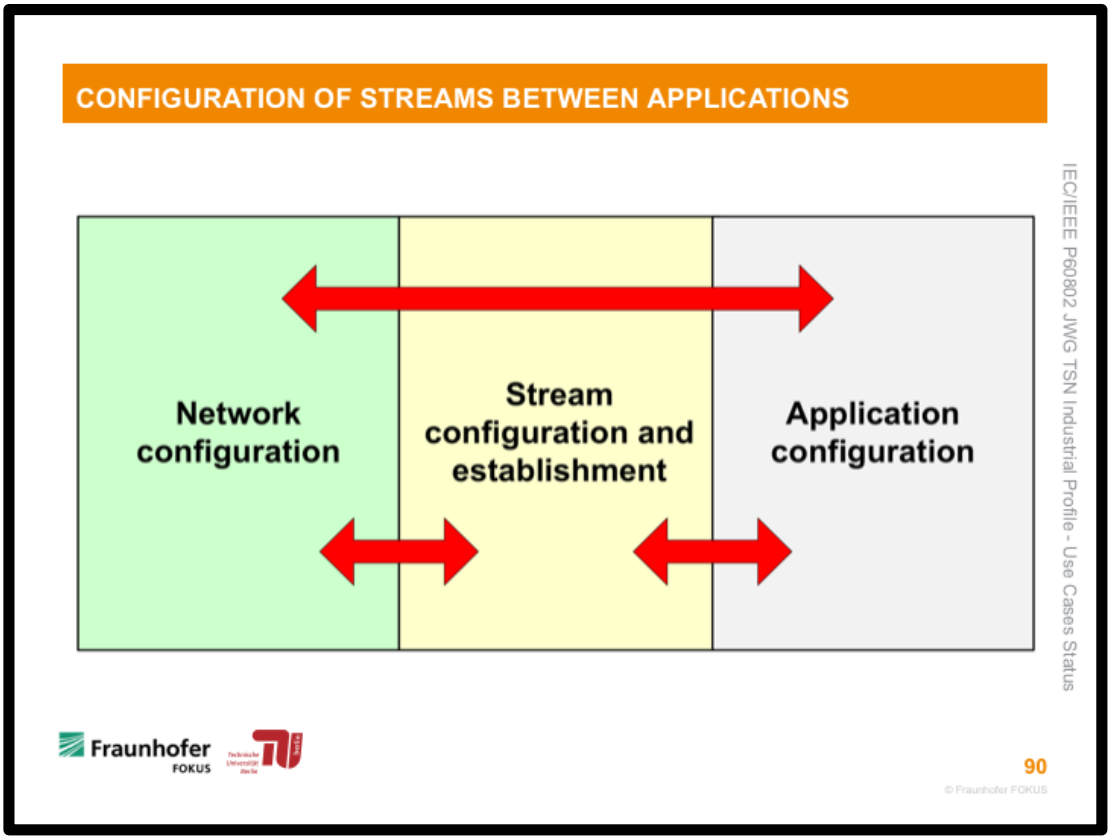
Fraunhofer FOKUS

88

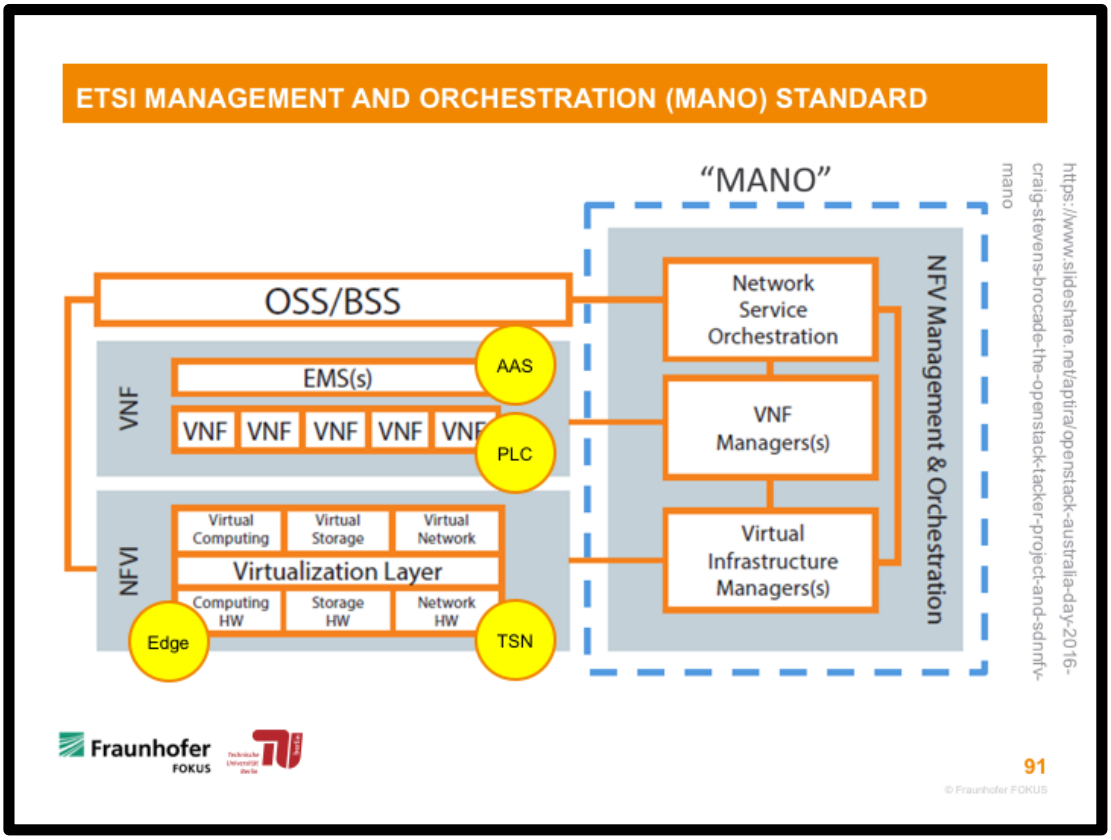
- Orchestrated components can be created, started, stopped, …
- Orchestrated components can be VMs, Containers, …



- Example from the „IEC/IEEE P60802 JWG TSN Industrial Profile - Use Cases Status" document
- IEC = International Electrotechnical Commission
- IEEE = Institute of Electrical and Electronics Engineers
- JWG = Joint Working Group
- P60802 = Standard for Local and Metropolitan Area Networks – Time-Sensitive Networking Profile for Industrial Automation
- Dedicated machine interfaces can decouple machine internal information and communication as private domain from the public upper layer networks of production cells or plants.
- Current Controller = PLCs,
- Future Controllers = RT Edge Nodes?



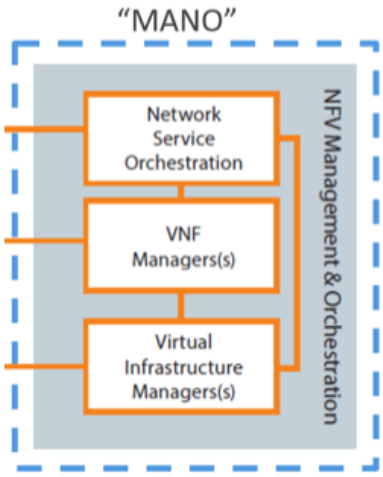
- Overall, it's a combination of network (e.g. TSN) and application (e.g. Edge) configuration to allow real-time communication



- One possible standard to adopt in this context is ETSI MANO
- Highlighted here: the mapping towards the IIoT context: edge, tsn, aas, plc
- ETSI = European Telecommunications Standards Institute
- OSS = Operational Support System / BSS = Business Support System
- EMS = Element Management System (e.g. an AAS)
- VNF = Virtualized Network Function (e.g. a PLC service)
- NFVI = Network Function Virtualization Infrastructure (e.g. Edge Node and TSN Switches)

THE NETWORK SERVICE ORCHESTRATION (NFVO)

- The NFVO must build a deployment plan and manage the lifecycle of the Network Services (NS)
- This is based on a Network Service Descriptor (NSD) with one or more VNF descriptors and their relationship (remember RDF?).
- This means, requiring the resources from the VIM, and the instantiation of the necessary VNFs to the VNF Manager.
- **Challenge:** Placement of functionalities with timing demands (**real-time**) on specific nodes (**Edge**) and configuring streams between them using specific networks (**TSN**).



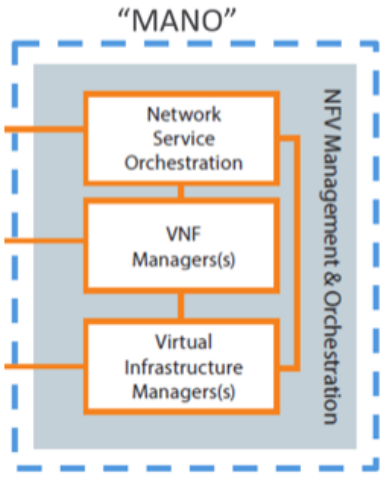
Fraunhofer FOKUS

92

- RDF as a good (better) candidate to describe dependency graphs
- Important challenge: placing rt applications to the correct edge nodes and configure the network

THE VNF MANAGER

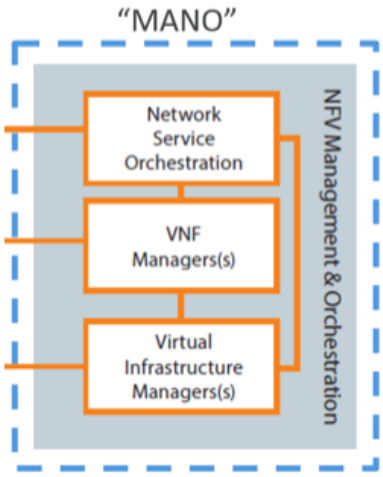
- The VNF Manager will take care of the lifecycle of VNFs, the building blocks of a Network Service (see slides before).
- Bellow some of the functionalities of the VNF Manager:
  - Check VNF instantiation feasibility
  - Update
  - Modification
  - Scale out/in and up/down
  - Termination
- VNFs are defined by the VNF Descriptor (VNFD)



Fraunhofer FOKUS

THE VIRTUALIZED INFRASTRUCTURE MANAGER (VIM)

- The VIM interfaces with the infrastructure (e.g. TSN Switches or Edge Nodes) that will provide the necessary resources to run the VNFs.
- Once allocated, the resources are then made available to the Orchestrator.



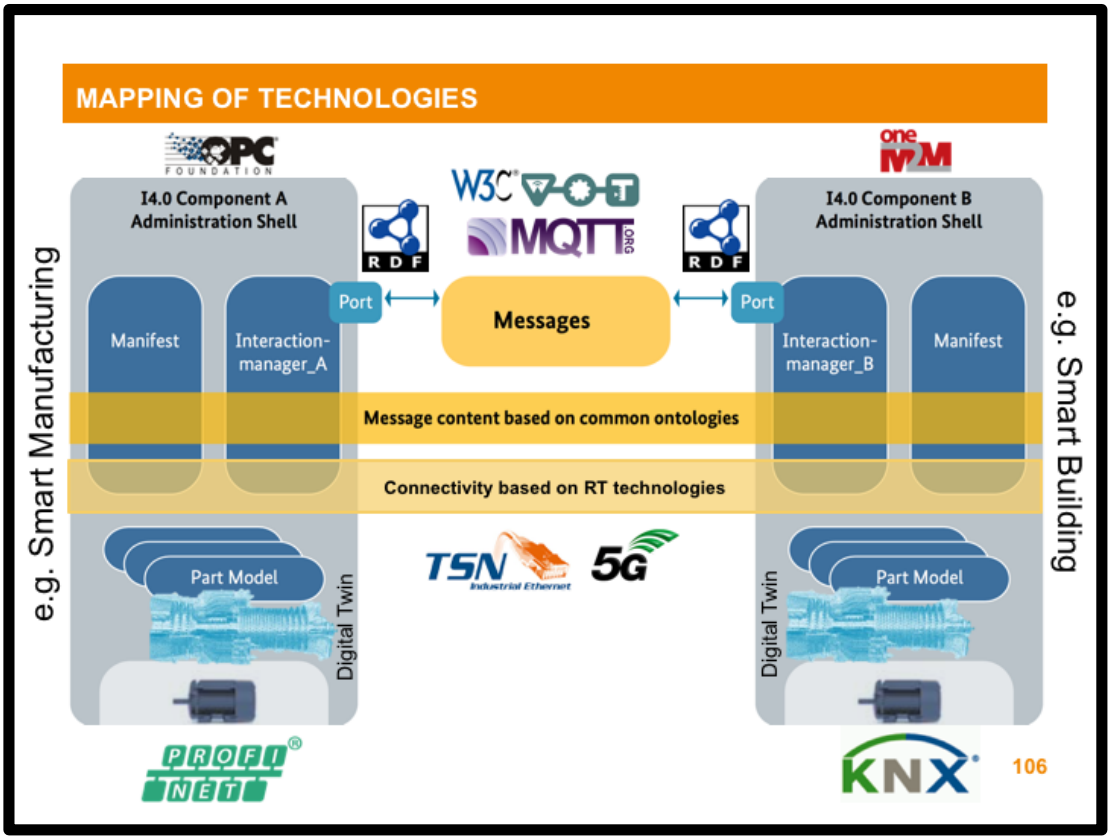
Fraunhofer FOKUS

REAL-TIME ORCHESTRATION

Questions?

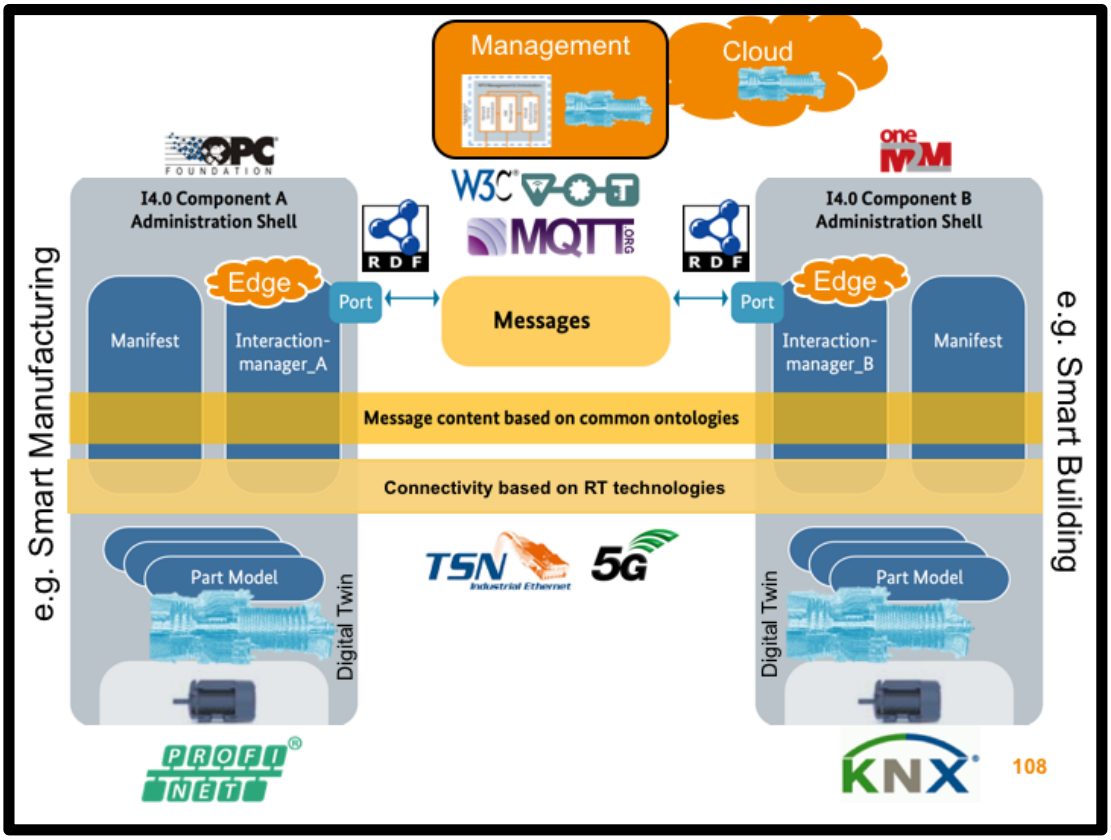
THE END

2 minutes



- We mapped the technologies we already learned about to an overview

WHERE IS THE CLOUD IN THIS CONTEXT?



- And this, depicted using one of the most ugliest figures available in the slide deck, is how all lectures in this course relate to each other.
- Example: left a robot from a manufacturing context - right a door of the building containing the shop floor.
- Each AAS is an edge node.
- A centralized (and potentially distributed) management system configures the network and nodes, holds a digital twin of the infrastructure.
- Edge-to-Cloud cooperation needed for many applications, also the cloud contains (preprocessed) parts of the distributed, linked digital twin.