

Universität
Rostock



Traditio et Innovatio

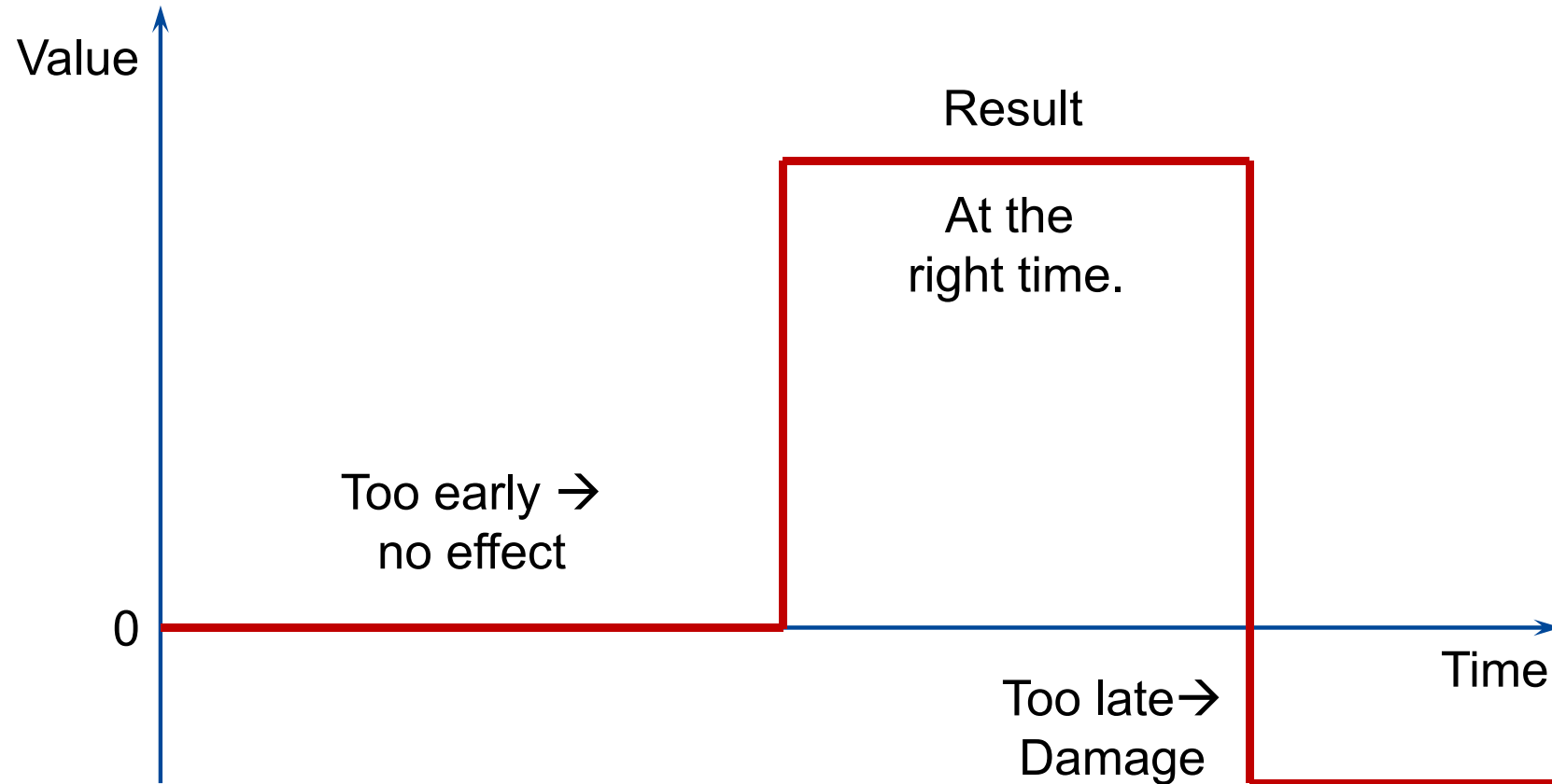
Realtime Publish/Subscribe for Cyber-Physical Systems

NEidl / Projekt CSI

PD Dr.-Ing. habil. Peter Danielis
Privatdozent für Parallele Systeme (ParSys)

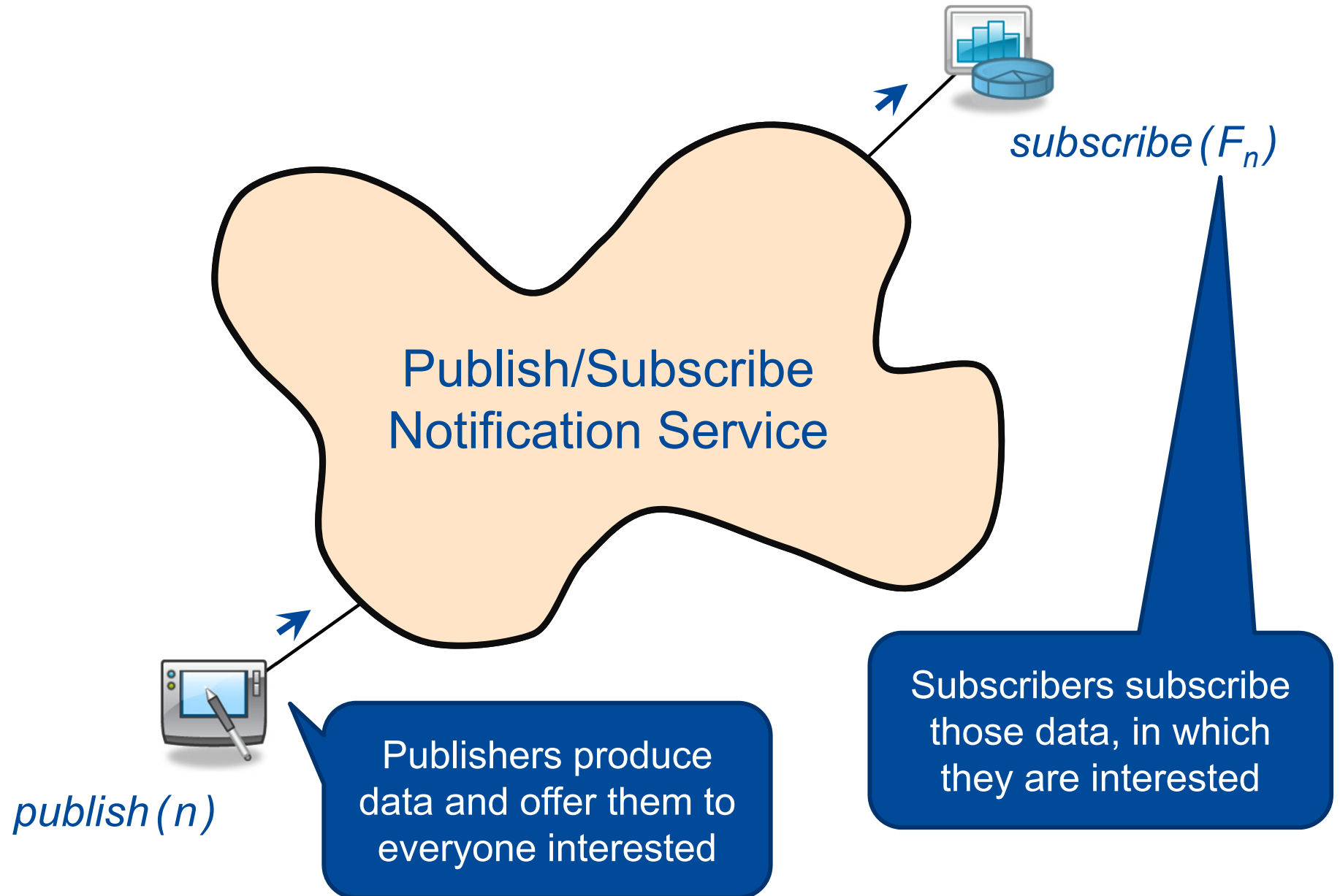
Dr.-Ing. Helge Parzyjegla
Architektur von Anwendungssystemen (AVA)

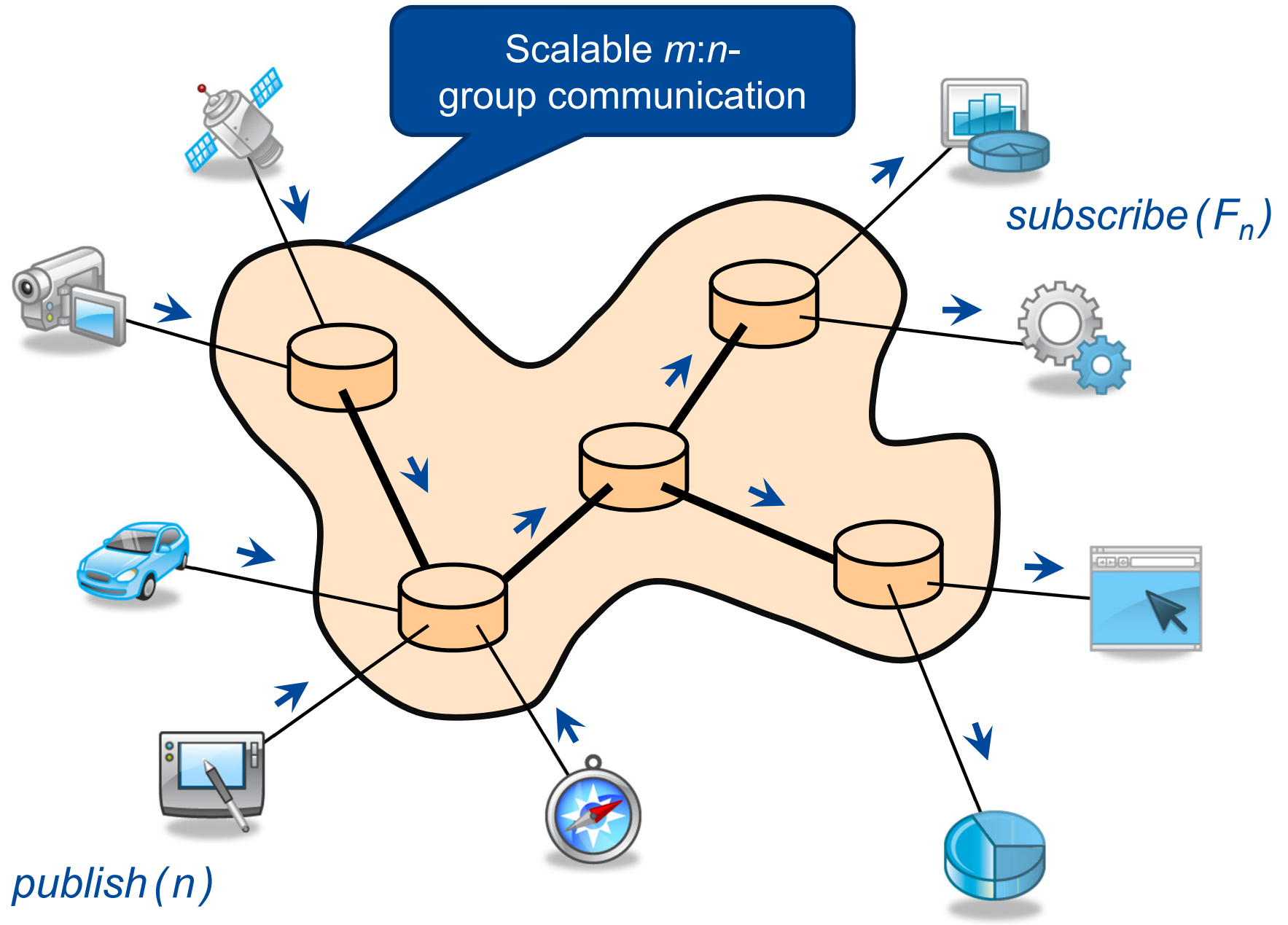
What is Realtime (Echtzeit)?



Not necessarily fast, but **predictable!**
→ Do the **right thing** at the **right time**.

What is Publish/Subscribe?





What are Cyber-Physical Systems?

- > Systems containing software components and mechanical or electronic parts that are interconnected via network
- > Interact with the real, physical world
 - are subject to physical laws
 - have requirements w.r.t (real) time
- > Examples
 - > Industry robots
 - > Production line in the smart factory
 - > Reconfigurable production cell of a smart factory
 - > Modern (autonomous) vehicles
 - > Steer/fly by wire
 - > Autopilots of any kind

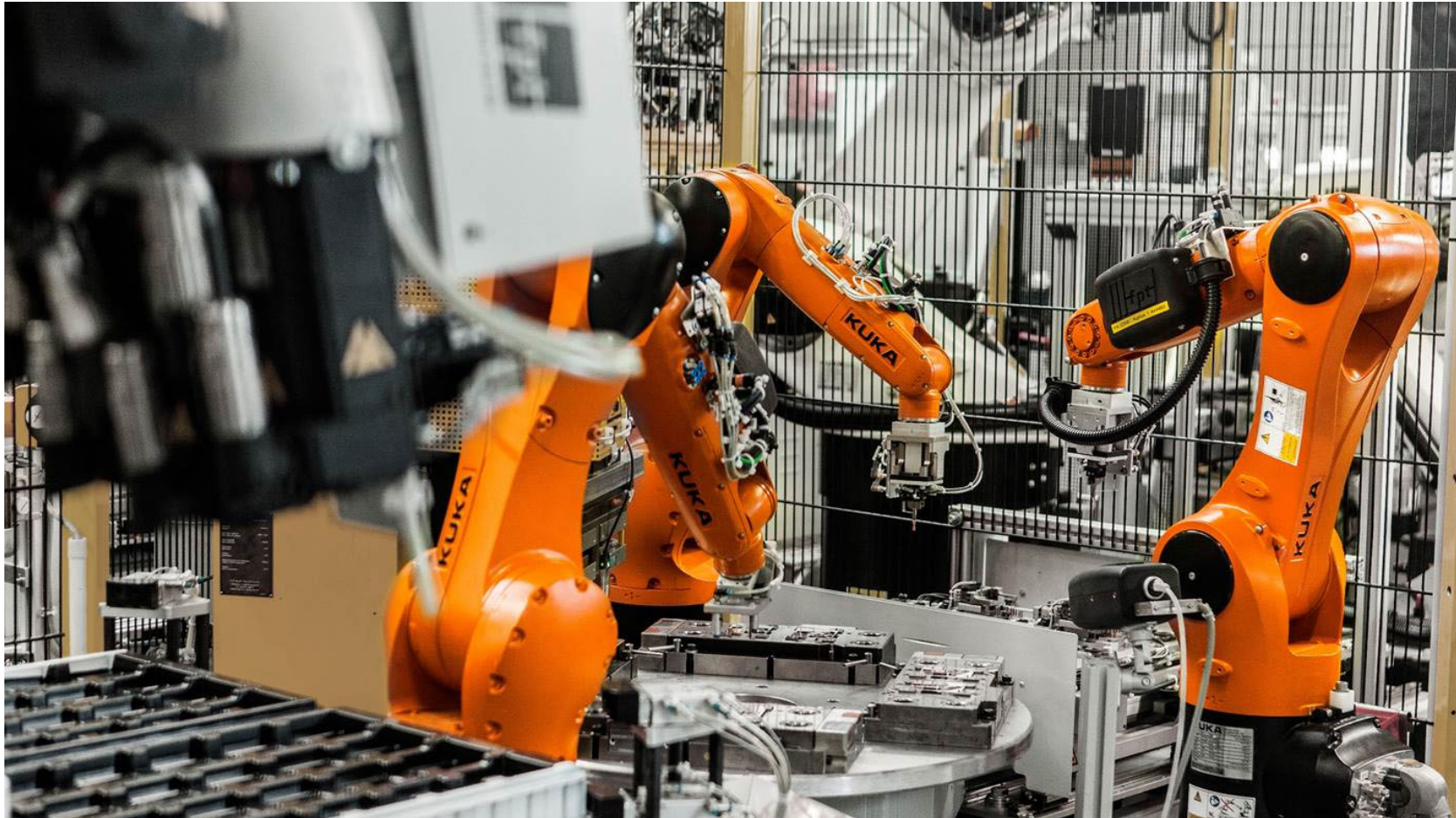
Industry Robots in a Smart Factory



Industry robots made by Kuka

Time-critical communication when handing over work pieces.

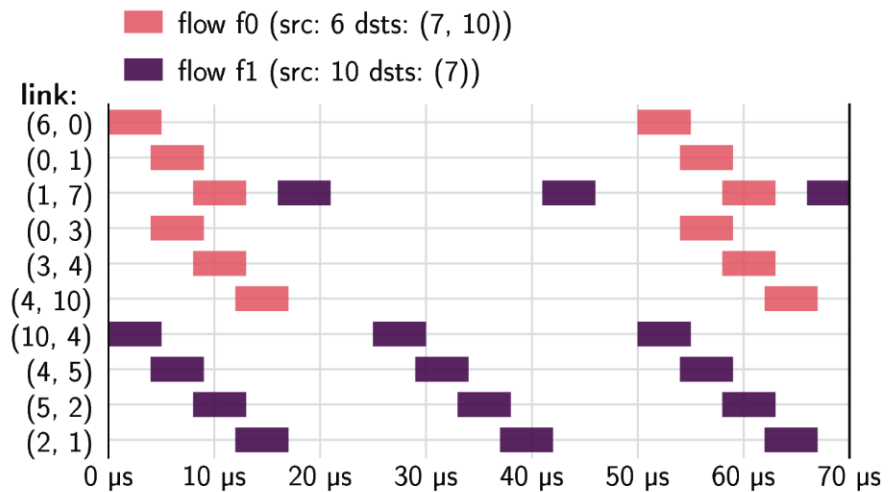
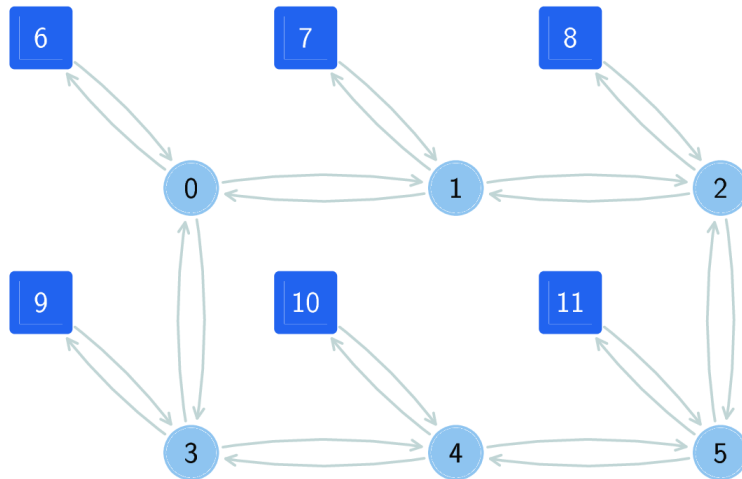
Reconfigurable Production Cell



Industry robots made by Kuka

Flexible communication in case of task changes.

Communication Schedule



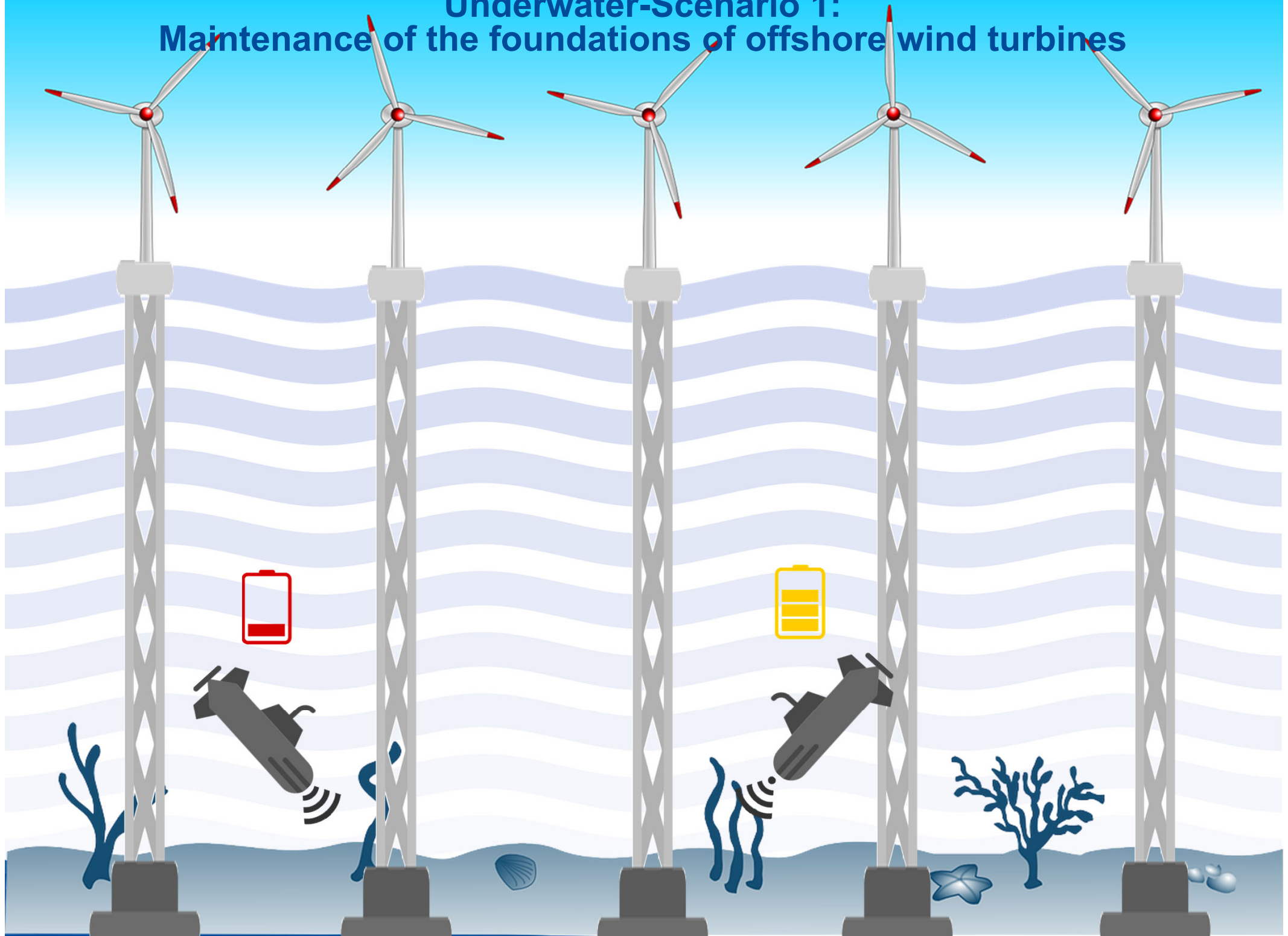
> Streams

- > From node 6 to nodes 7 and 10 (multicast)
- > From node 10 to node 7

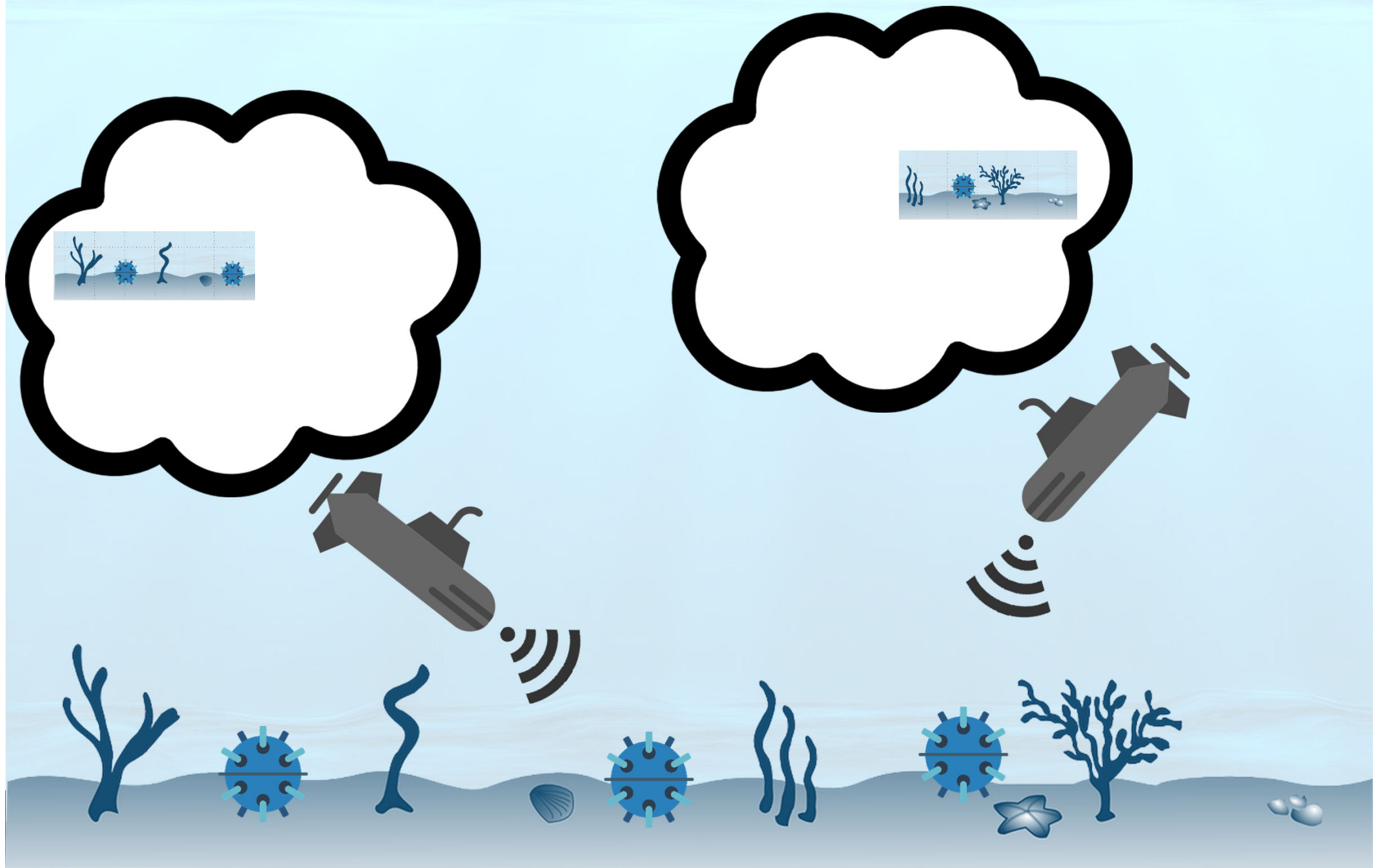
> Schedule

- > Determines exactly when which packet is sent over which link
- > Has to be always without conflicts → provable correct
- > Needs to be adapted whenever communication pattern changes
- > Additional traffic of lesser importance is possible

Underwater-Scenario 1: Maintenance of the foundations of offshore wind turbines



Underwater-Scenario 2: Clearance of Unexploded Ordnance (UXO) from World War II



Projects and Collaborations

- > Realtime publish/subscribe communication
 - > Part of a DFG project
 - > Planning of flexible communication patterns and reservation of required time slots on communication links
 - > Formal models and methods for scheduling
 - > Estimation of the worst case runtime for publishing and filtering (content-based if necessary) a notification
 - > Application scenario within a smart factory
- > Autonomous Underwater Vehicles (AUVs)
 - > Cooperation with the Institute for the Protection of Maritime Infrastructures, Resilience Department of Maritime Systems, German Aerospace Center (DLR) Bremerhaven
 - > Cooperative navigation of several AUVs
 - > Limited Energy restricts movement and usage of sensors
 - > Opportunistic communication via acoustic modems

Tasks: Realtime Publish/Subscribe

- > Simulation models for realtime communication (TSN standards)
 - > Test and extension of new TSN features of OMNeT++/INET
 - > Simulation of mixed-critical data traffic
 - > Configuration of time-critical networks with mixed-critical traffic
 - > Simulation model for per-stream filtering and policing
- > Publish/Subscribe SDN-Controller
 - > Extension of the ONOS-SDN-Controller for Publish/Subscribe
 - > Open Network Operating System (ONOS)
 - > Modular, extensible, and distributed controller architecture written in Java
 - > Available as a Linux VM
 - > Northbound application for management based on intents
 - > REST interface for technical users (e.g., applications)
 - > Web application for human users (e.g., administrators)
 - > Southbound interface as alternative (inband) management interface
 - > Network emulation via Mininet and Open vSwitch
 - > Comparison (e.g., evaluation) of management via northbound and southbound interface

Tasks: Autonomous Underwater Vehicles

- > Development of an integrated simulation architecture
- > Porting of existing models to OMNeT++ 6.0/INET 4.4
- > Further development of motion models for AUVs
 - > Reaction to obstacles, autonomous adaptation
- > Integration of simulation models for AUVs
 - > Energy consumption, communication with acoustic modems, motion, sensors
- > Simulation of cooperative missions
 - > Formation of multiple AUVs
 - > Mapping of the seafloor
 - > Cooperative hunting
- > Implementations using simulator OMNeT++ and C++
 - > Python for scripting and evaluation of simulation results



Organizational Matters

- > Weekly meeting on **Thursdays** at **11:00 am** in **SR 014 (AE26)**
- > Up to two teams
 - > Team A: Realtime publish/subscribe
(probably more fine-grained distribution of tasks)
 - > Team B: Autonomous Underwater Vehicles (AUVs)
- > Design methodology
 - > Agile development
 - > Three milestones w.r.t. design, implementation, documentation

**Type and size/scale of tasks depends on number
and interests of participants!**

Registration and Contact

> Enrolement in respective Stud.IP course

1.  23848 (Integrierte Lehrveranstaltung) Neueste Entwicklungen der Informatik (Verteiltes Hochleistungsrechnen)
2.  23897 (Integrierte Lehrveranstaltung) Projekt Master Computer Science International : AVA

> Questions via email to Peter Danielis and Helge Parzyjegla

- > peter.danielis@uni-rostock.de
- > helge.parzyjegla@uni-rostock.de