

Universität
Rostock



Traditio et Innovatio

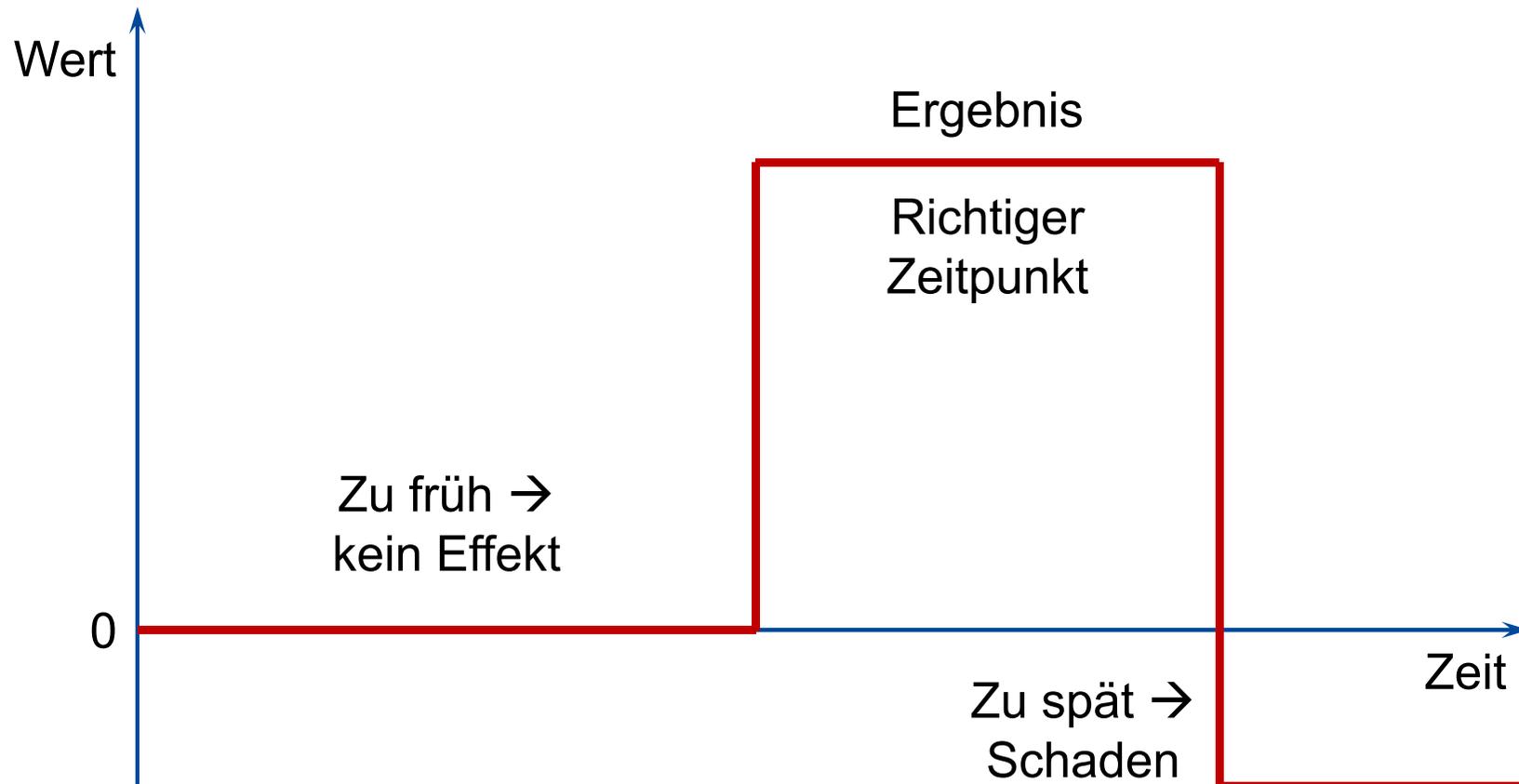
Realtime Publish/Subscribe für Cyber-Physische-Systeme

KSWS / Projekt

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

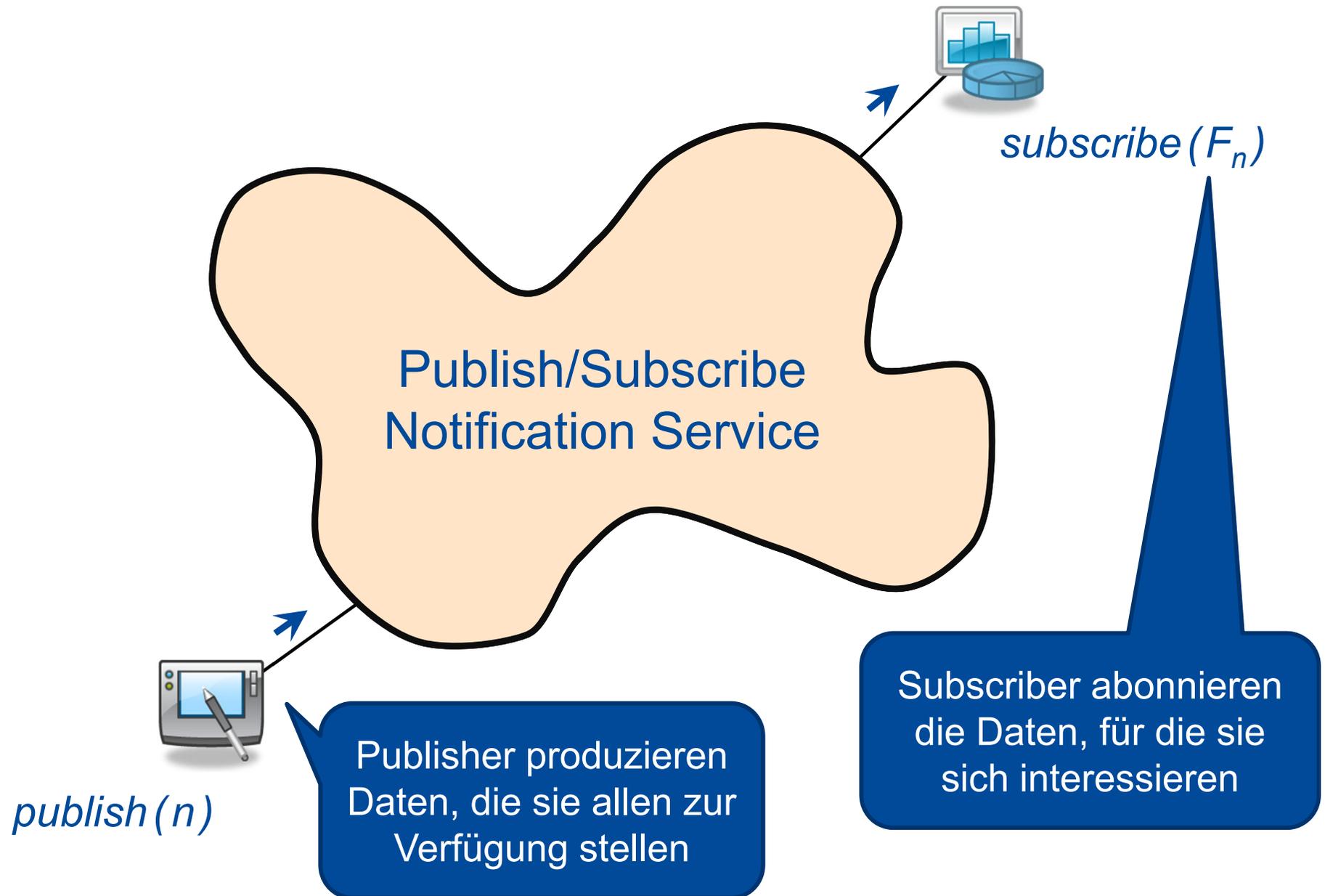
Dr.-Ing. Helge Parzyjgla
Architektur von Anwendungssystemen (AVA)

Was bedeutet Realtime/Echtzeit?

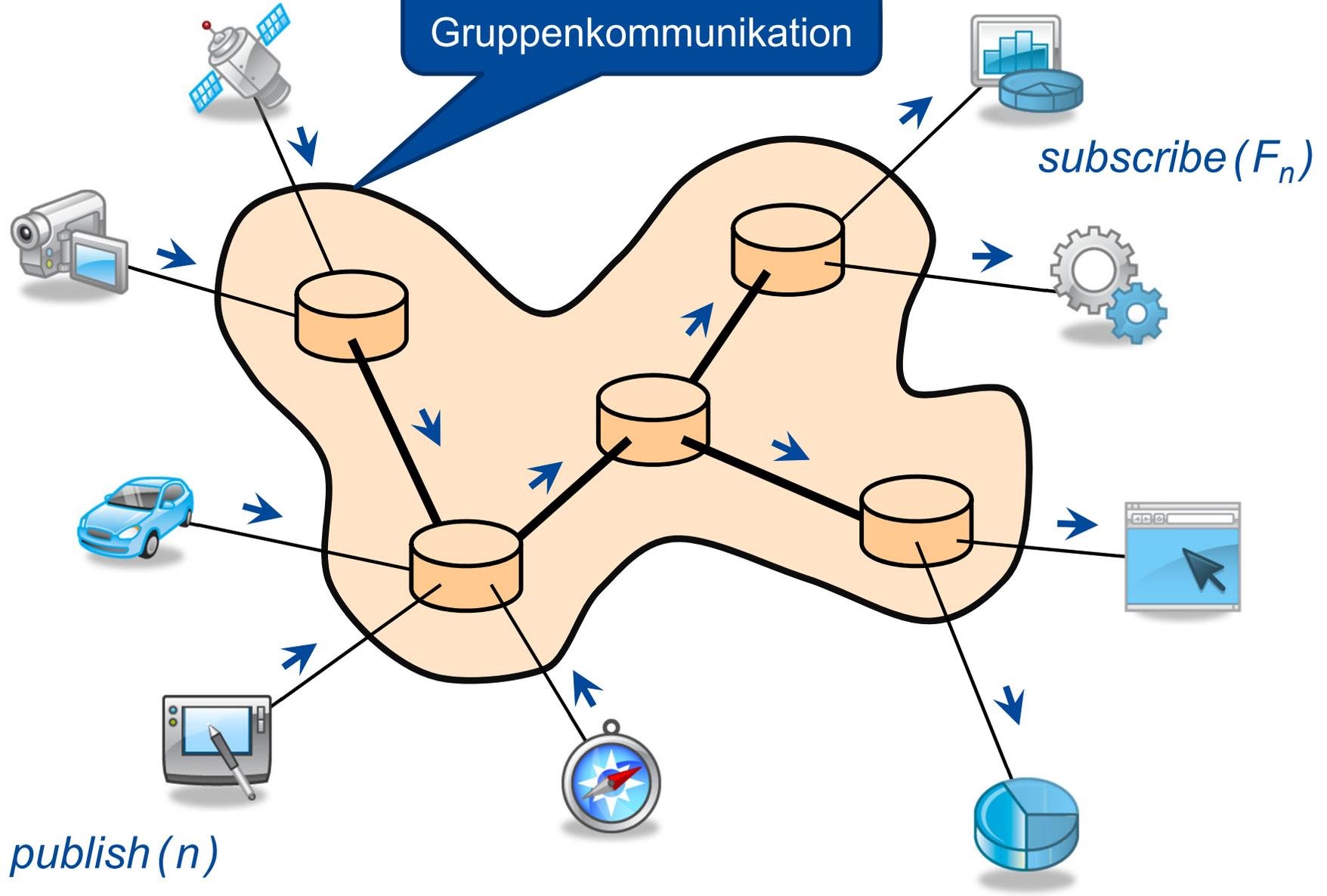


Nicht notwendigerweise schnell, sondern **vorhersagbar!**
→ Das **Richtige** zum **richtigen Zeitpunkt** tun.

Was ist Publish/Subscribe?



Skalierbare $m:n$ -
Gruppenkommunikation



Was sind Cyber-Physische Systeme?

- > Systeme bestehend aus Software-Komponenten und mechanischen bzw. elektronischen Teilen verbunden über ein Kommunikationsnetz
- > Wirken auf die reale, physische Welt ein
 - unterliegen physikalischen Gesetzen
 - haben zeitliche Anforderungen (Echtzeit)
- > Beispiele
 - > Industrieroboter
 - > Fertigungsstraße in der Smart Factory
 - > Rekonfigurierbare Produktionszelle einer Smart Factory
 - > Moderne (autonome) Fahrzeuge
 - > Steer/Fly-By-Wire
 - > Autopilotfunktionen jeglicher Art

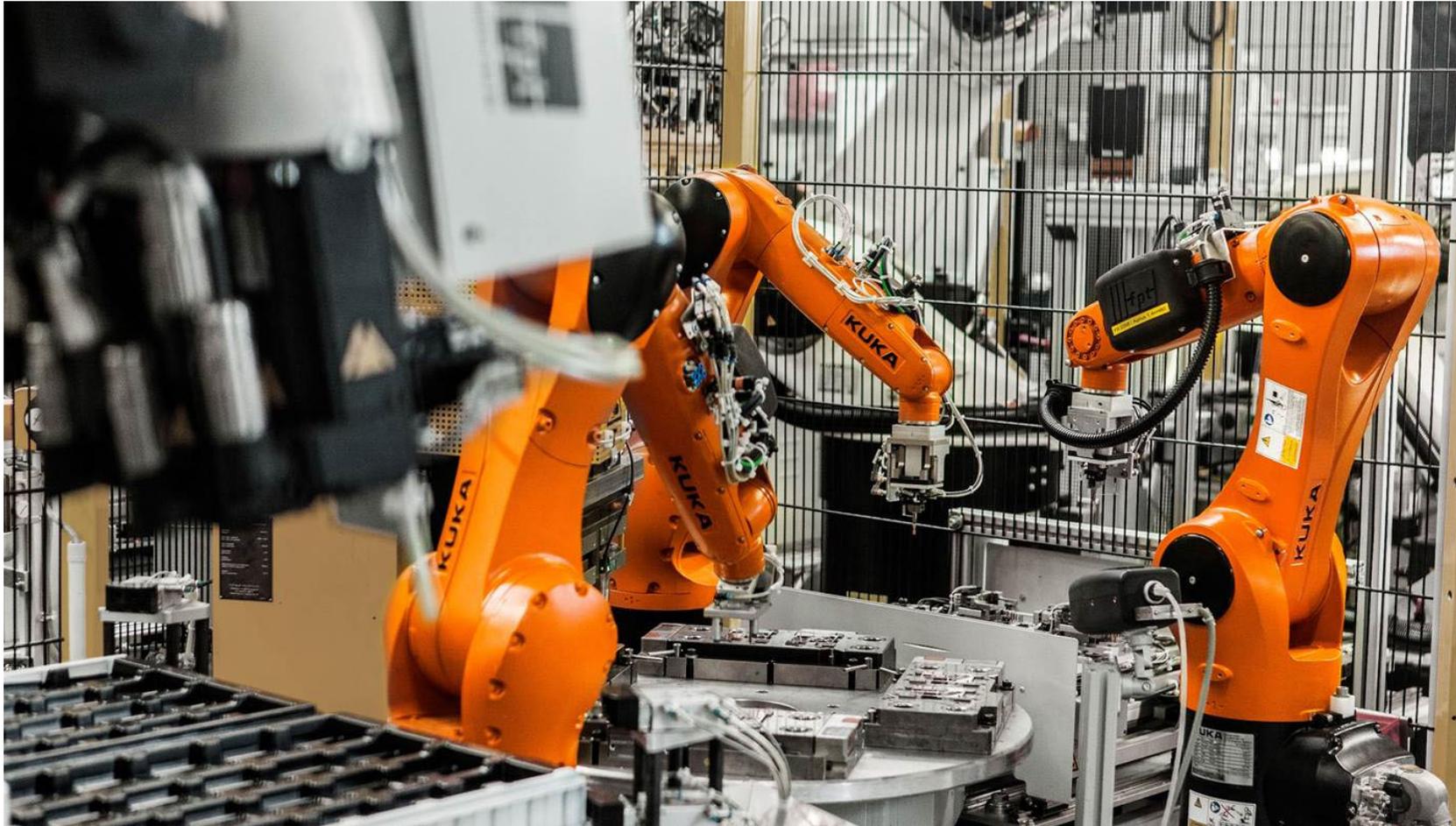
Industrieroboter in der Smart Factory



Fertigungsroboter von Kuka

Zeitkritische Kommunikation bei Übergabe eines Werkstücks.

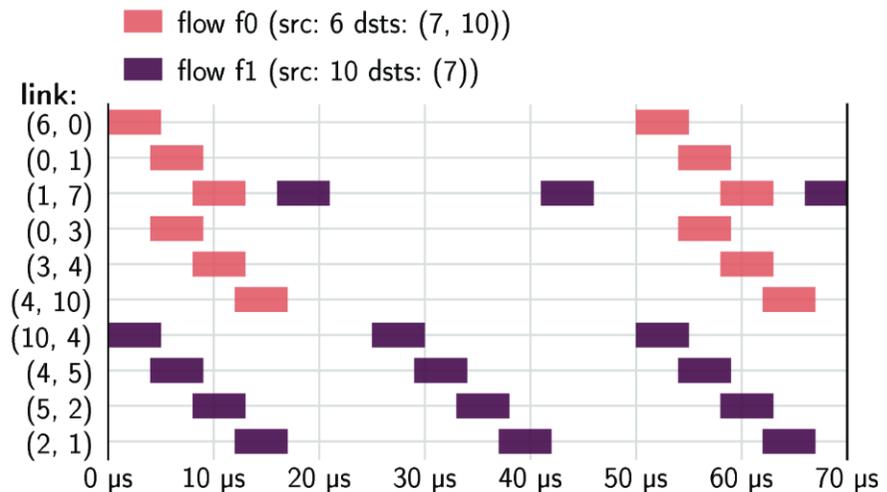
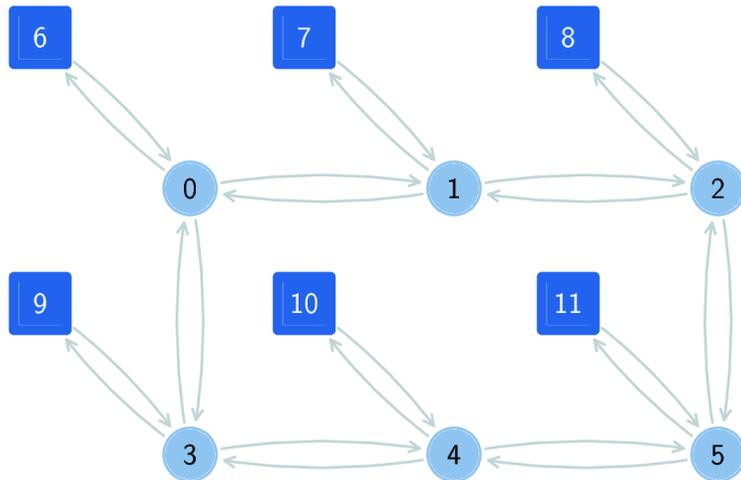
Rekonfigurierbare Produktionszelle



Fertigungsroboter von Kuka

Flexible Kommunikation bei Aufgabenänderung.

Geplante Echtzeitkommunikation



> Streams

- > Von Knoten 6 zu Knoten 7 und 10 (Multicast)
- > Von Knoten 10 zu Knoten 7

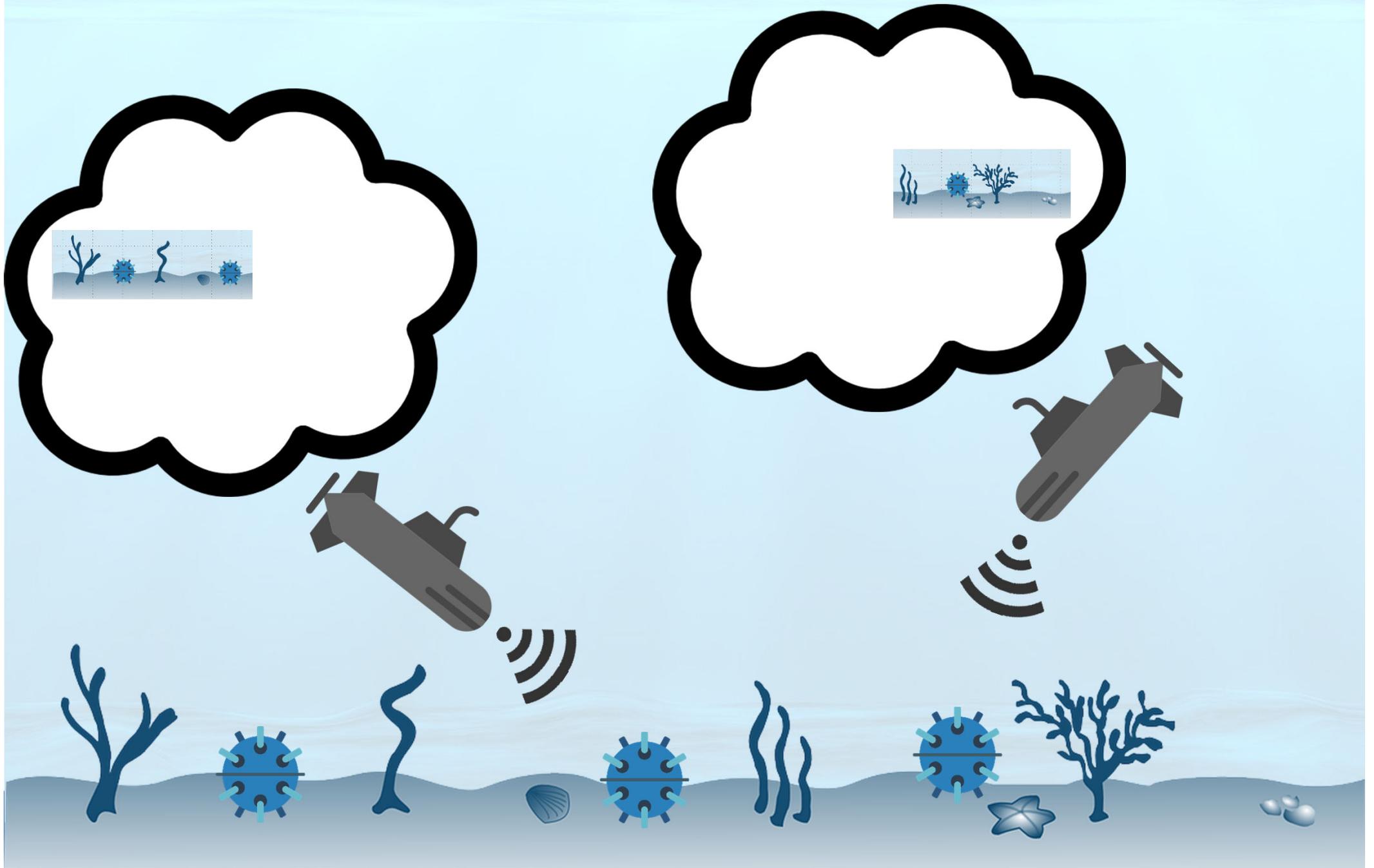
> Ablaufplan (engl. Schedule)

- > Bestimmt, wann welches Paket über welchen Link gesendet wird
- > Stets ohne Konflikte → nachweisbar korrekt
- > Anpassung bei Änderungen des Kommunikationsmusters
- > Zusätzlicher, weniger wichtiger Datenverkehr möglich

Unterwasser-Szenario 1: Wartung der Fundamente von Offshore-Windkraftanlagen



Unterwasser-Szenario 2: Räumung von Altlasten aus dem 2. Weltkrieg (Blindgänger, Unexploded Ordnance - UXO)



Projekte und Kooperationen

- > Echtzeitfähige Publish/Subscribe-Kommunikation
 - > Teil eines DFG-Projektes
 - > Planung flexibler Kommunikationsmuster und Reservierung notwendiger Zeitslots auf den Kommunikationsverbindungen
 - > Abschätzung der Worst-Case-Laufzeit einer Publikation und deren (ggf. inhaltsbasierter) Filterung und Auslieferung
 - > Einsatzgebiet in der Smart Factory
- > Autonome Unterwasserfahrzeuge (AUVs)
 - > Zusammenarbeit mit dem Institut für den Schutz maritimer Infrastrukturen, Abteilung Resilienz maritimer Systeme, Deutsches Zentrum für Luft- und Raumfahrt (DLR) Bremerhaven
 - > Kooperative Navigation mehrerer AUVs
 - > Begrenzte Energie limitiert Sensoren und Bewegung
 - > Opportunistische Kommunikation durch Akustikmodems

Aufgaben: Echtzeit-Publish/Subscribe

- > Simulationsmodelle für Echtzeitkommunikation (TSN-Standards)
 - > Test und Erweiterung neuer TSN-Features von OMNeT++/INET
 - > Simulation von gemischt-kritischem Datenverkehr
 - > Konfiguration von zeitkritischen Netzen mit gemischt-kritischem Datenverkehr
 - > Simulationsmodell für Per-Stream Filtering and Policing
- > Publish/Subscribe SDN-Controller
 - > Erweiterung des ONOS SDN-Controllers um Publish/Subscribe
 - > Open Network Operating System (ONOS)
 - > Modulare, erweiterbare und verteilte Controller-Infrastruktur in Java
 - > Bereitgestellt als Linux-VM
 - > Northbound-Anwendung zum Management basierend auf Intents
 - > REST-Schnittstelle für technische Nutzer (z.B. Anwendungen)
 - > Webanwendung für menschliche Nutzer (z.B. Admins)
 - > Southbound-Schnittstelle als alternatives (Inband-) Management-Interface
 - > Emulation des Netzwerks via Mininet und Open vSwitch
 - > Vergleich (z.B. Evaluation) zwischen Management via Northbound- und Southbound-Schnittstelle

Aufgaben: Autonome Unterwasserfahrzeuge

- > Entwicklung einer integrierten Simulationsarchitektur
- > Portierung bestehender Modelle in OMNeT++ 6.0/INET 4.4
- > Weiterentwicklung von Bewegungsmodellen für AUVs
 - > Reaktion auf Hindernisse, autonome Anpassung
- > Integration von Simulationsmodellen für AUVs
 - > Energieverbrauch, Kommunikations mit Akustikmodems, Bewegung, Sensorik
- > Simulation von kooperativen Missionen
 - > Formation mehrerer AUVs
 - > Kartierung des Meeresbodens
 - > Kooperative Jagd
- > Implementierung in dem Simulator OMNet++ in C++
 - > Python etc. zur Auswertung der Simulationsergebnisse

Organisatorisches

- > Wöchentliche Treffen **donnerstags** um **13:00 Uhr** in **SR 014 (AE26)**
- > Bis zu zwei Teams
 - > Team A: Echtzeit-Publish/Subscribe
(wahrscheinlich feingliedrigere Aufgabenaufteilung)
 - > Team B: Autonome Unterwasserfahrzeuge
- > Entwicklungsmethodik
 - > Agile Entwicklung
 - > Drei Meilensteine bzgl. Entwurf, Implementierung, Bericht

**Art und Umfang der Aufgaben nach Anzahl
und Interessen der Teilnehmer!**

Anmeldung und Kontakt

> Eintrag in die richtige Stud.IP-Veranstaltung

1.  23850 (Vorlesung) KSWS: Verteiltes Hochleistungsrechnen
2.  23851 (Projekt) Projekt: Verteiltes Hochleistungsrechnen

> Fragen an Peter Danielis und Helge Parzyjegla per E-Mail

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