



Realtime Publish/Subscribe für Cyber-Physische-Systeme

KSWS / Projekt / NEidl / Projekt CSI

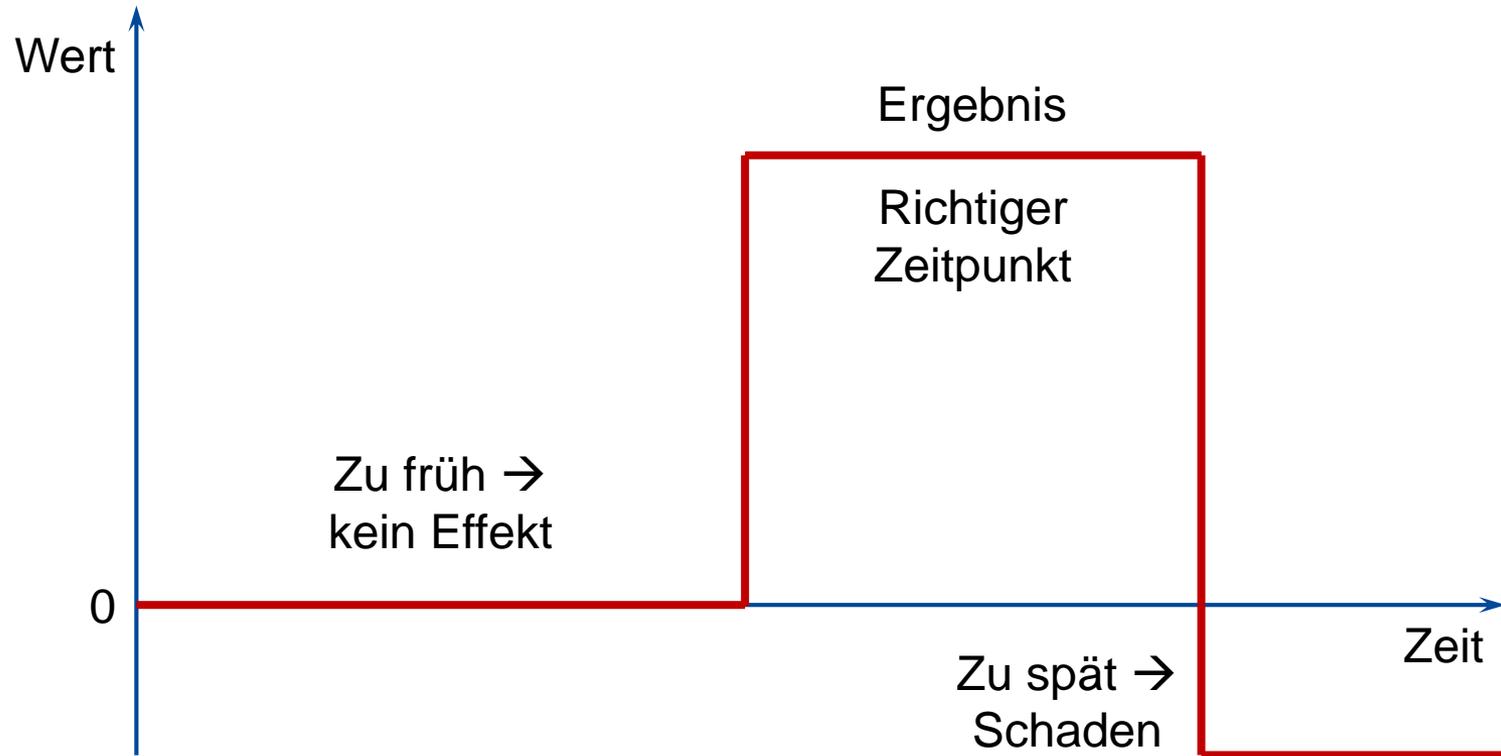
Dr.-Ing. Peter Danielis

Verteiltes Hochleistungrechnen (VHR)

Dr.-Ing. Helge Parzyjgla

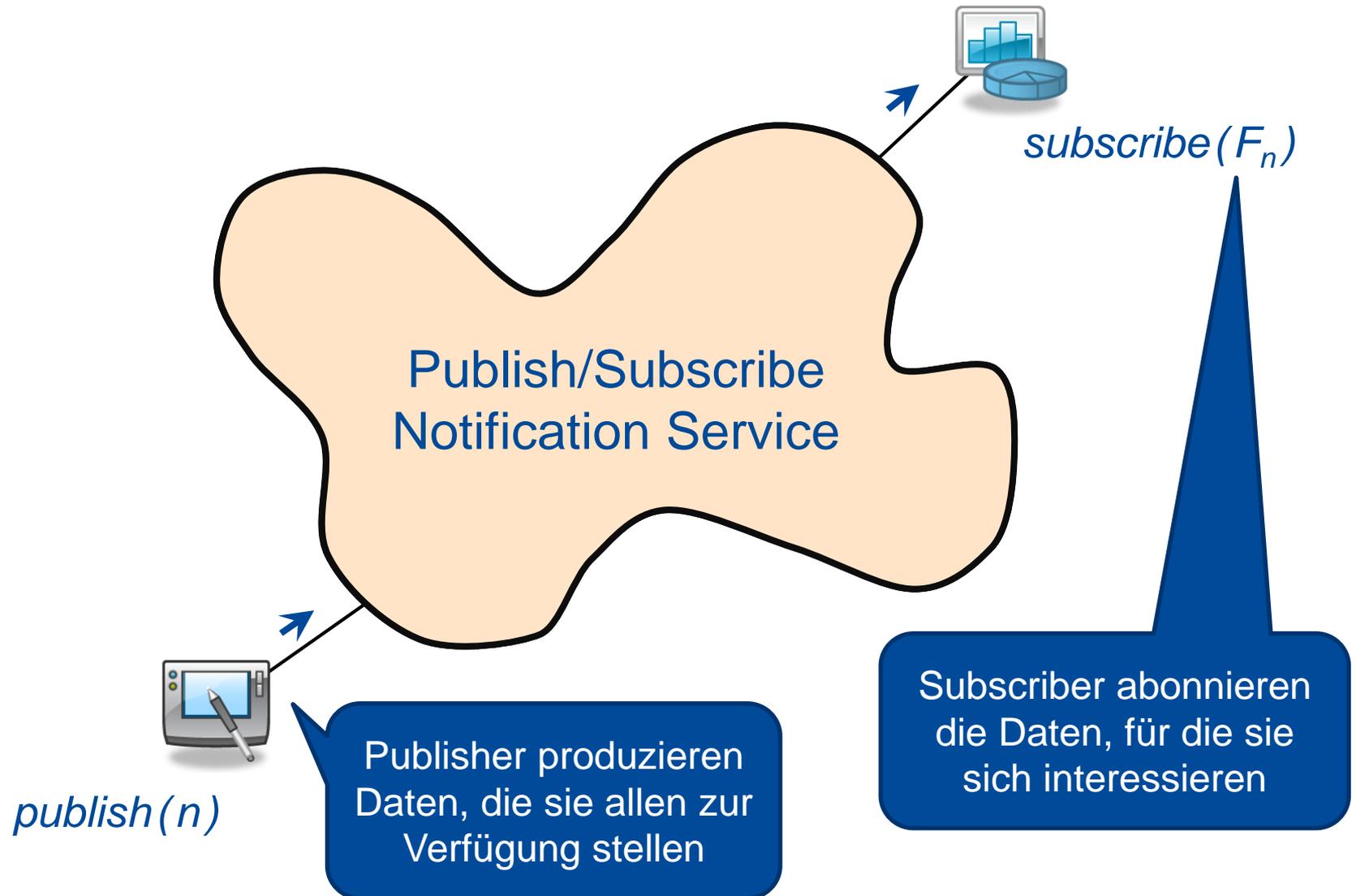
Architektur von Anwendungssystemen (AVA)

Was bedeutet Realtime/Echtzeit?

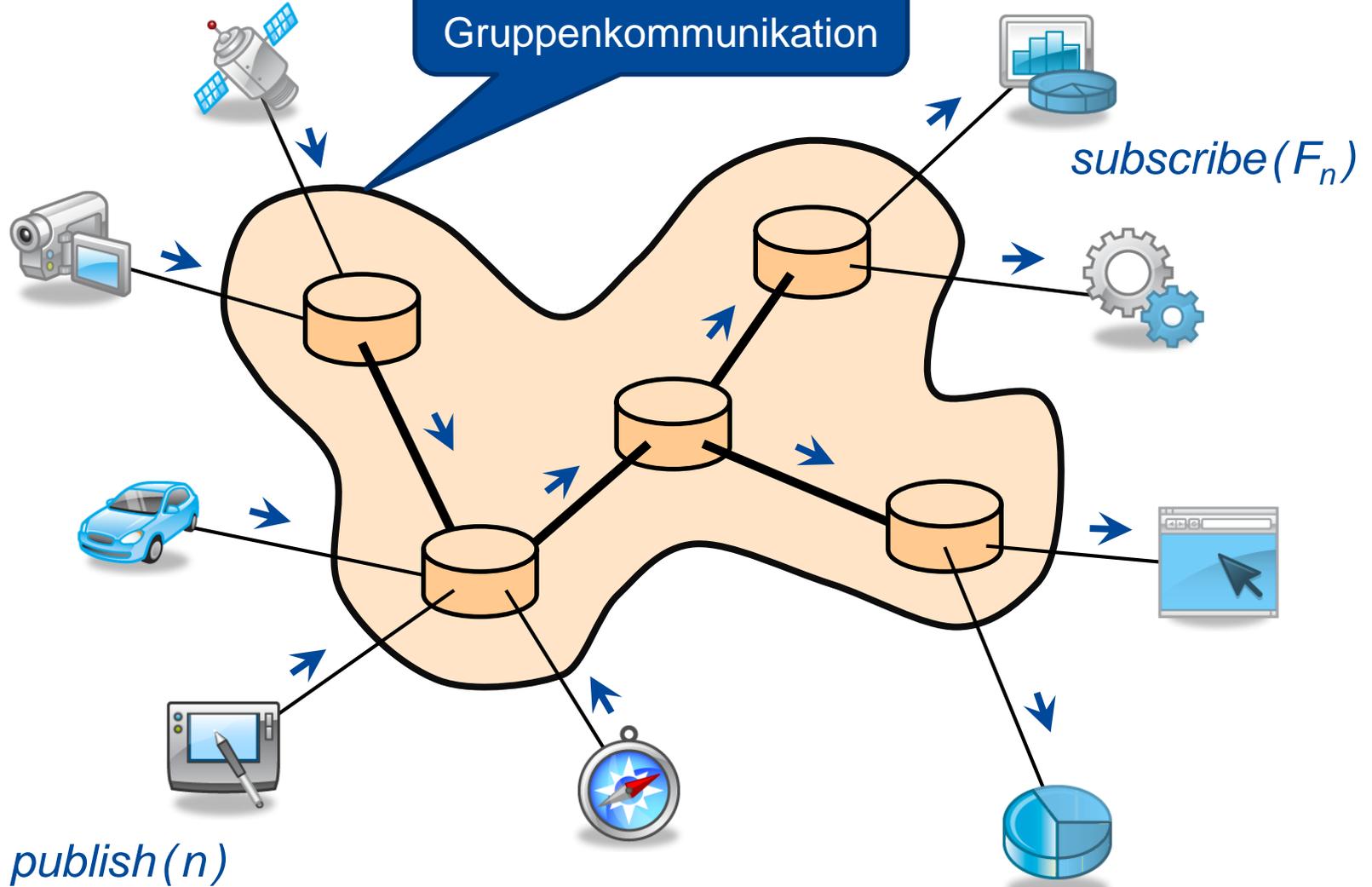


Nicht notwendigerweise schnell, sondern **vorhersagbar!**
→ Das **Richtige** zum **richtigen Zeitpunkt** tun.

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 Factor
 - > Moderne (autonome) Fahrzeuge
 - > Steer/Fly-By-Wire
 - > Autopilotfunktionen jeglicher Art

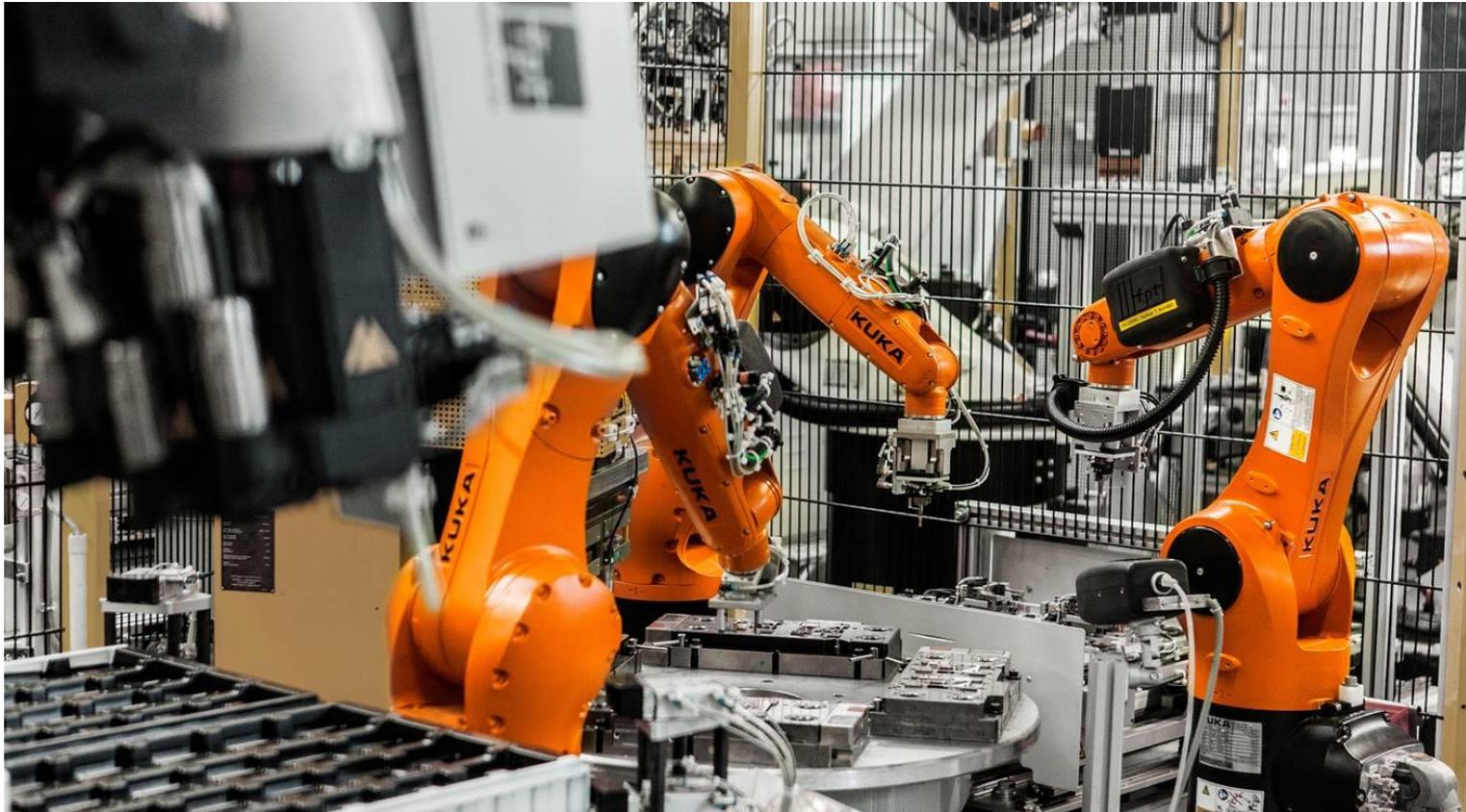
Industrieroboter in der Smart Factory



Fertigungsroboter von Kuka

Zeitkritische Kommunikation bei Übergabe eines Werkstücks.

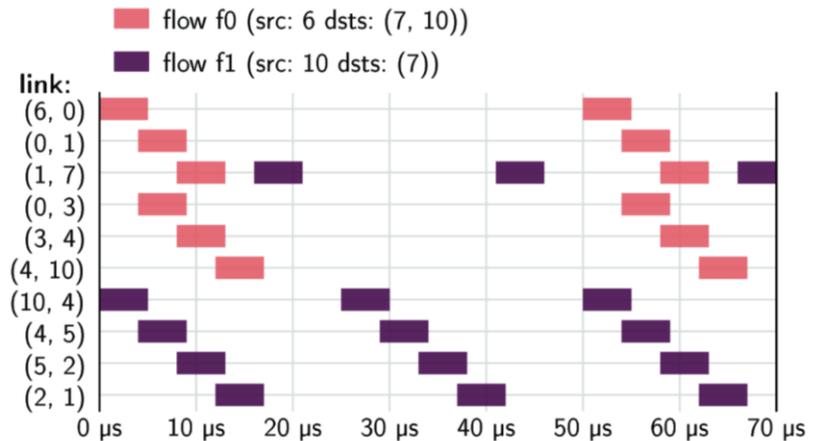
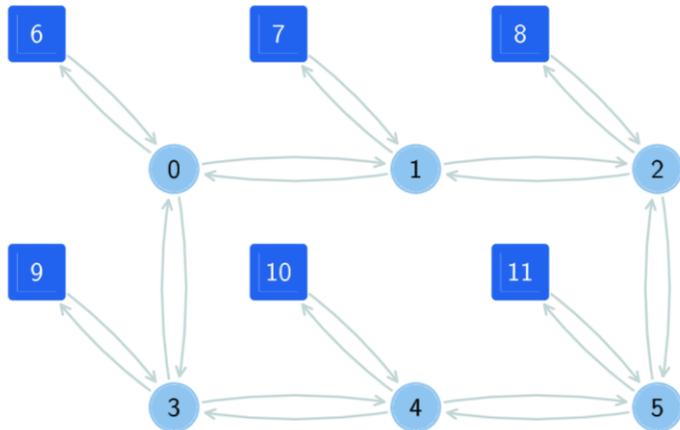
Rekonfigurierbare Produktionszelle



Fertigungsroboter von Kuka

Flexible Kommunikation bei Aufgabenänderung.

Geplante Echtzeitkommunikation



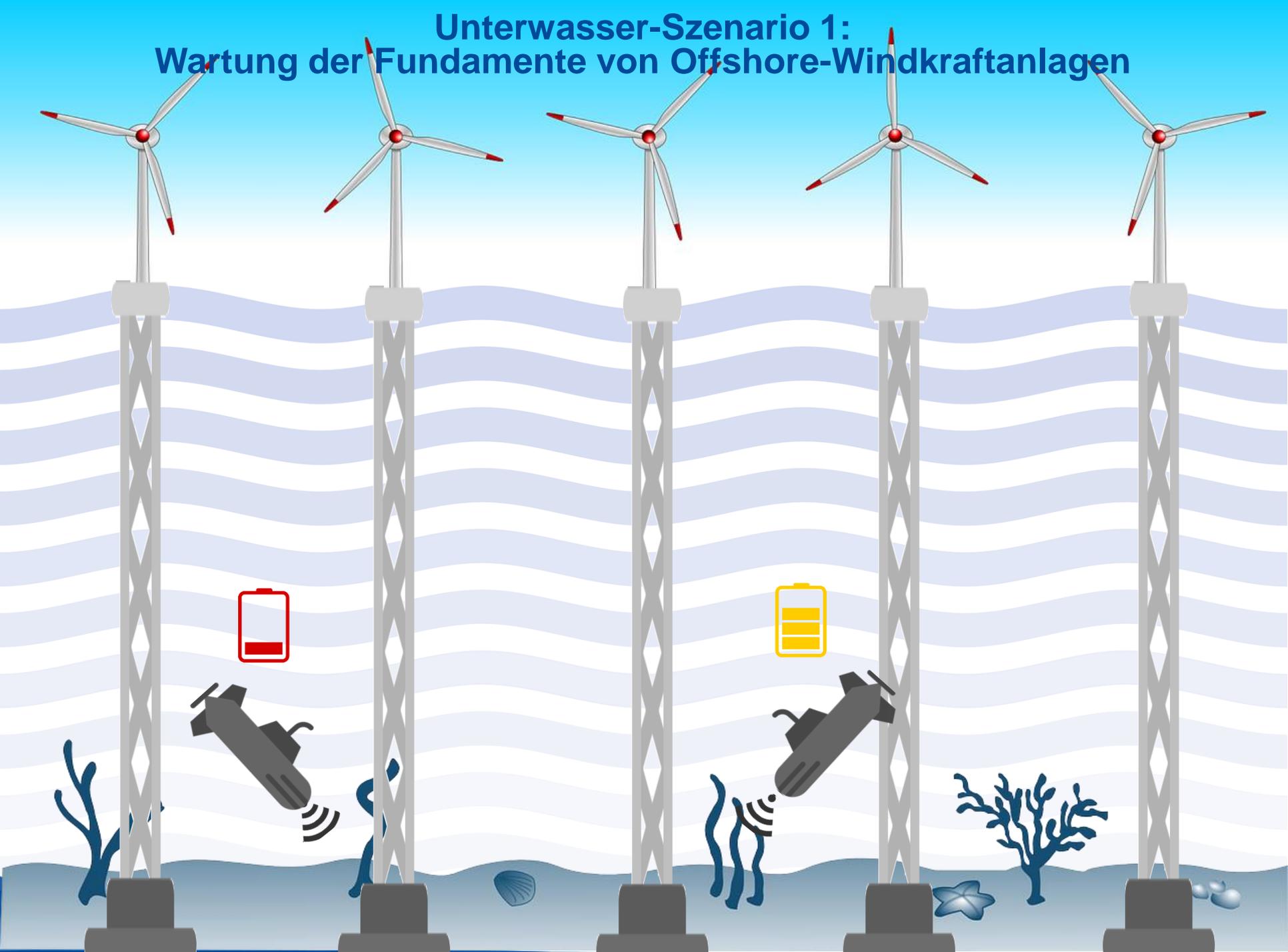
> Streams

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

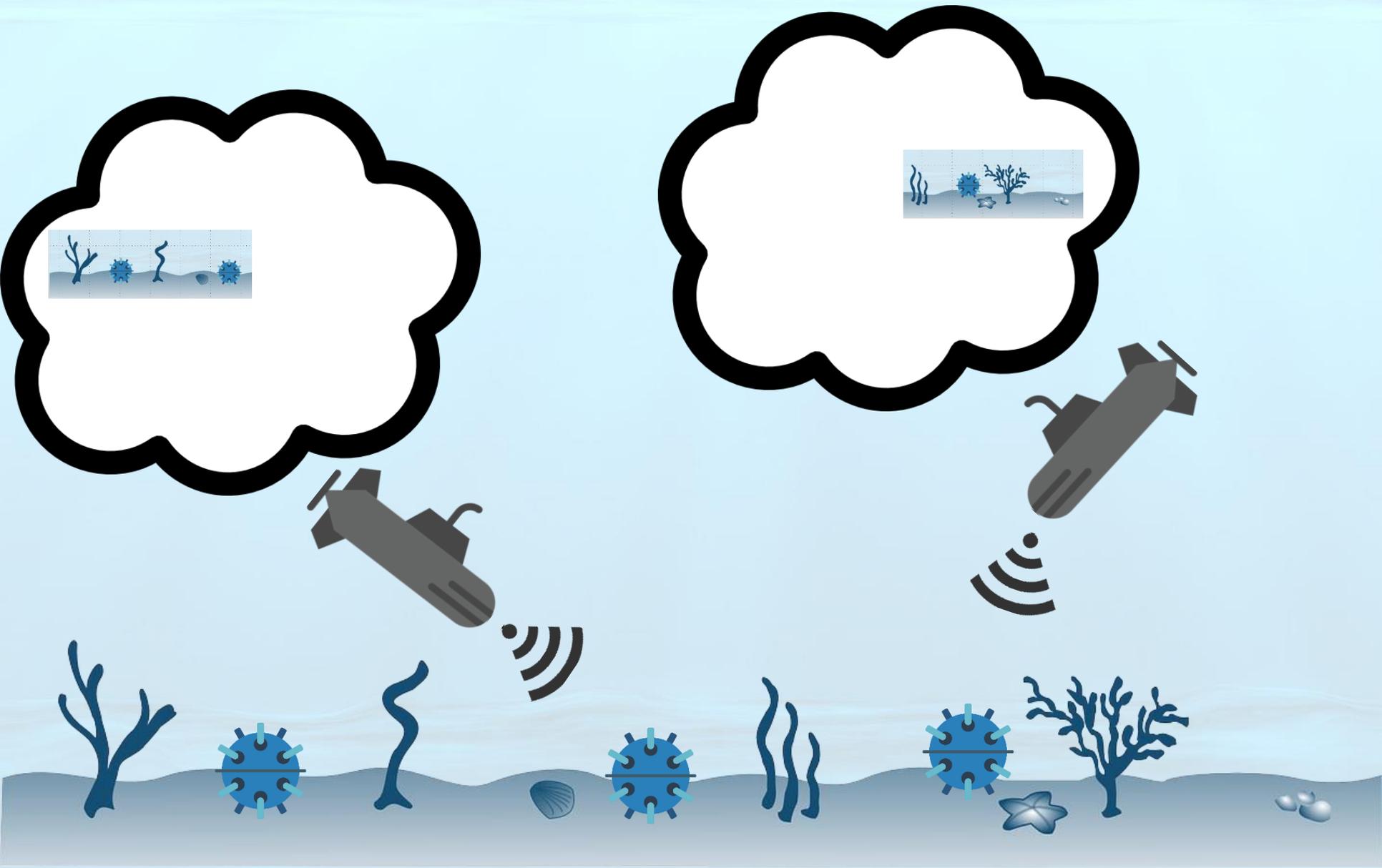
> Ablaufplan (engl. Schedule)

- > 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)
 - > Simulation von gemischt-kritischem Datenverkehr
 - > Konfiguration von zeitkritischen Netzen mit gemischt-kritischem Datenverkehr
 - > Simulationsmodell für Per-Stream Filtering and Policing
 - > Test und Erweiterung neuer TSN-Features von OMNeT++/INET
- > TSN-Scheduler
 - > ILP-Modell für Gurobi bzw. CPLEX (→ Python)
 - > Eigene Heuristiken in diversen Programmiersprachen (C++, Java, Go, Rust, ...)
 - > Input/Output von Constraints bzw. Konfigurationen
 - > Checker für gefundene Lösungen (→ Python)
 - > Benchmarking (Docker Container)

Aufgaben: Autonome Unterwasserfahrzeuge

- > Weiterentwicklung von Bewegungsmodellen für AUVs
 - > Reaktion auf Hindernisse
 - > Autonome Anpassung
- > Integration von Simulationsmodellen für AUVs
 - > Energieverbrauch
 - > Kommunikations mit Akustikmodems
 - > Bewegung
- > 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

- > 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 (Lecture) KSWs: Verteiltes Hochleistungsrechnen
2.  23848 (Integrierte Lehrveranstaltung) Neueste Entwicklungen der Informatik (Verteiltes Hochleistungsrechnen)
3.  23851 (Project) Projekt: Verteiltes Hochleistungsrechnen
4.  23897 (Integrierte Lehrveranstaltung) Projekt Master Computer Science International : AVA

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

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