

Realtime Publish/Subscribe für Cyber-Physische-Systeme

KSWS VHR / Projekt VHR / NEidI VHR
Underwater Vehicles meet Smart Factory

Dr.-Ing. Peter Danielis

Verteiltes Hochleistungrechnen (VHR)

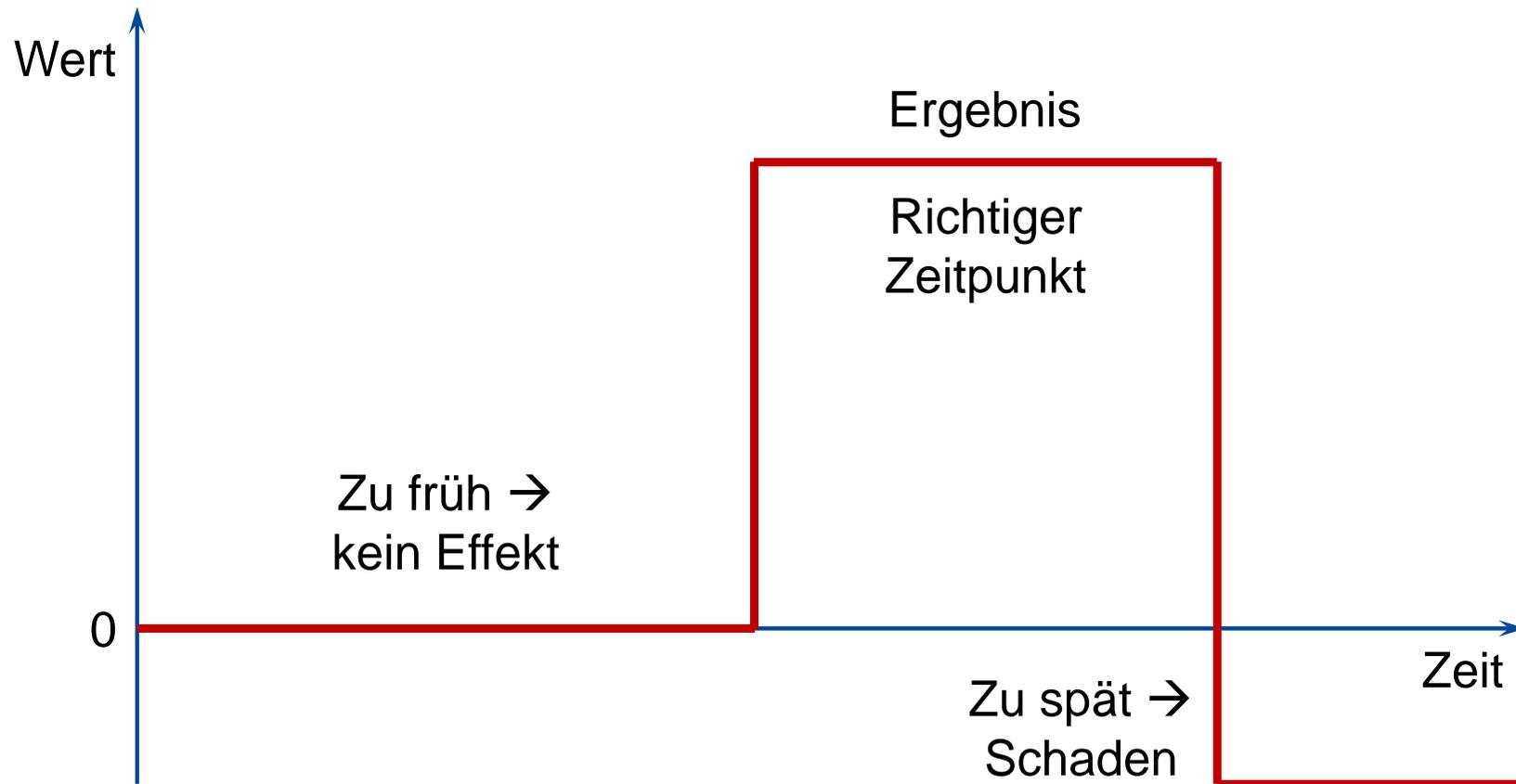
Dr.-Ing. Helge Parzyjegla

Architektur von Anwendungssystemen (AVA)

M.Sc. Eike Björn Schweißguth

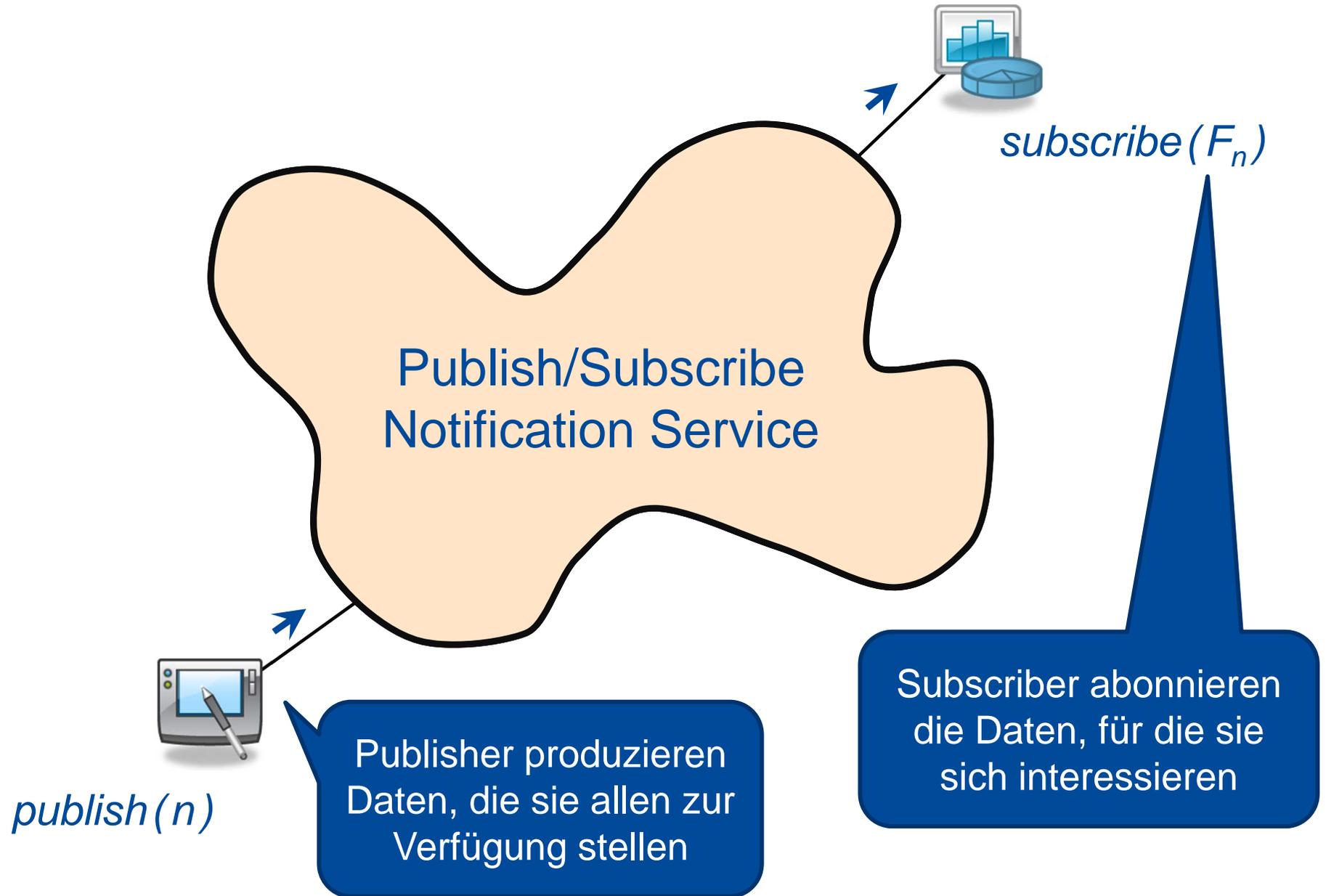
Institut für Angewandte Mikroelektronik und Datentechnik (IMD)

Was bedeutet Realtime/Echtzeit?

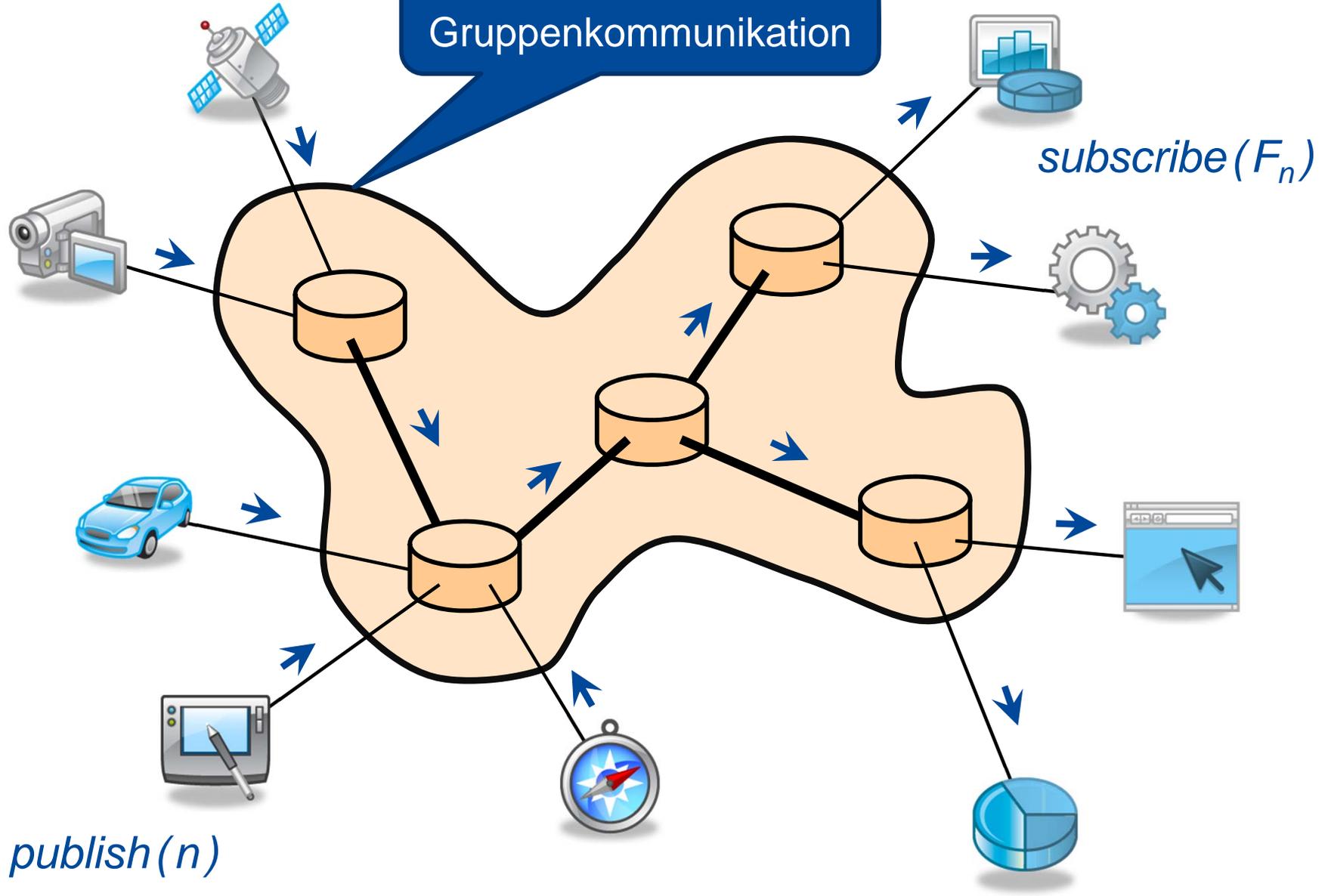


Nicht notwendigerweise schnell, sondern **vorhersagbar!**
→ Das **Richtige** zum **richtigen Zeitpunkt** tun.

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
 - > Schwarm von Autonomen Unterwasserfahrzeugen (AUVs) zur
 - > Wartung der Fundamente von Offshore-Windkraftanlagen
 - > Räumung von Altlasten (Blindgängern) auf Meeresboden
 - > Industrieroboter
 - > Fertigungsstraße in der Smart Factory
 - > Rekonfigurierbare Produktionszelle einer Smart Factory

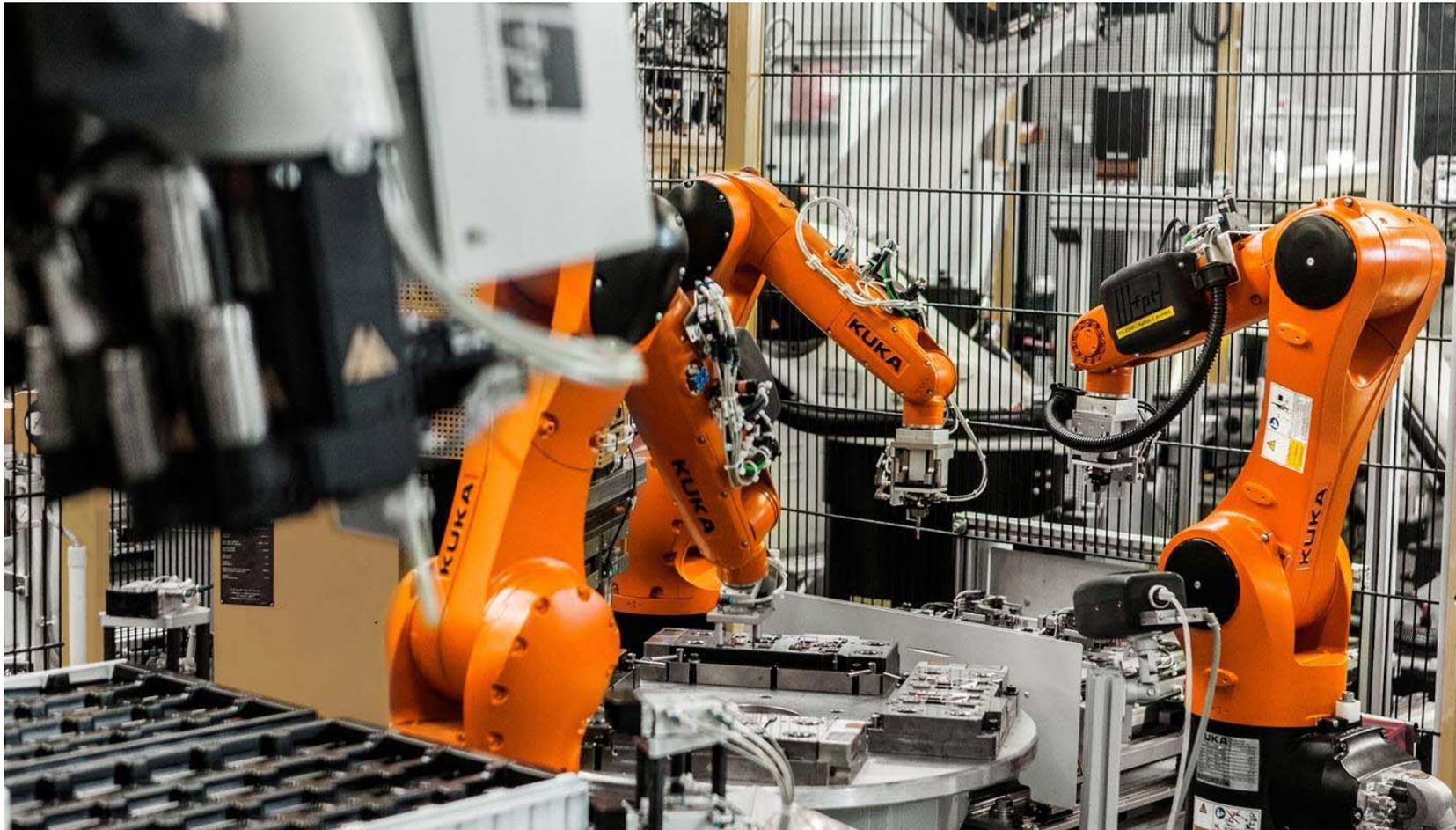
Industrieroboter in der Smart Factory



Fertigungsroboter von Kuka

Zeitkritische Kommunikation bei Übergabe eines Werkstücks.

Rekonfigurierbare Produktionszelle



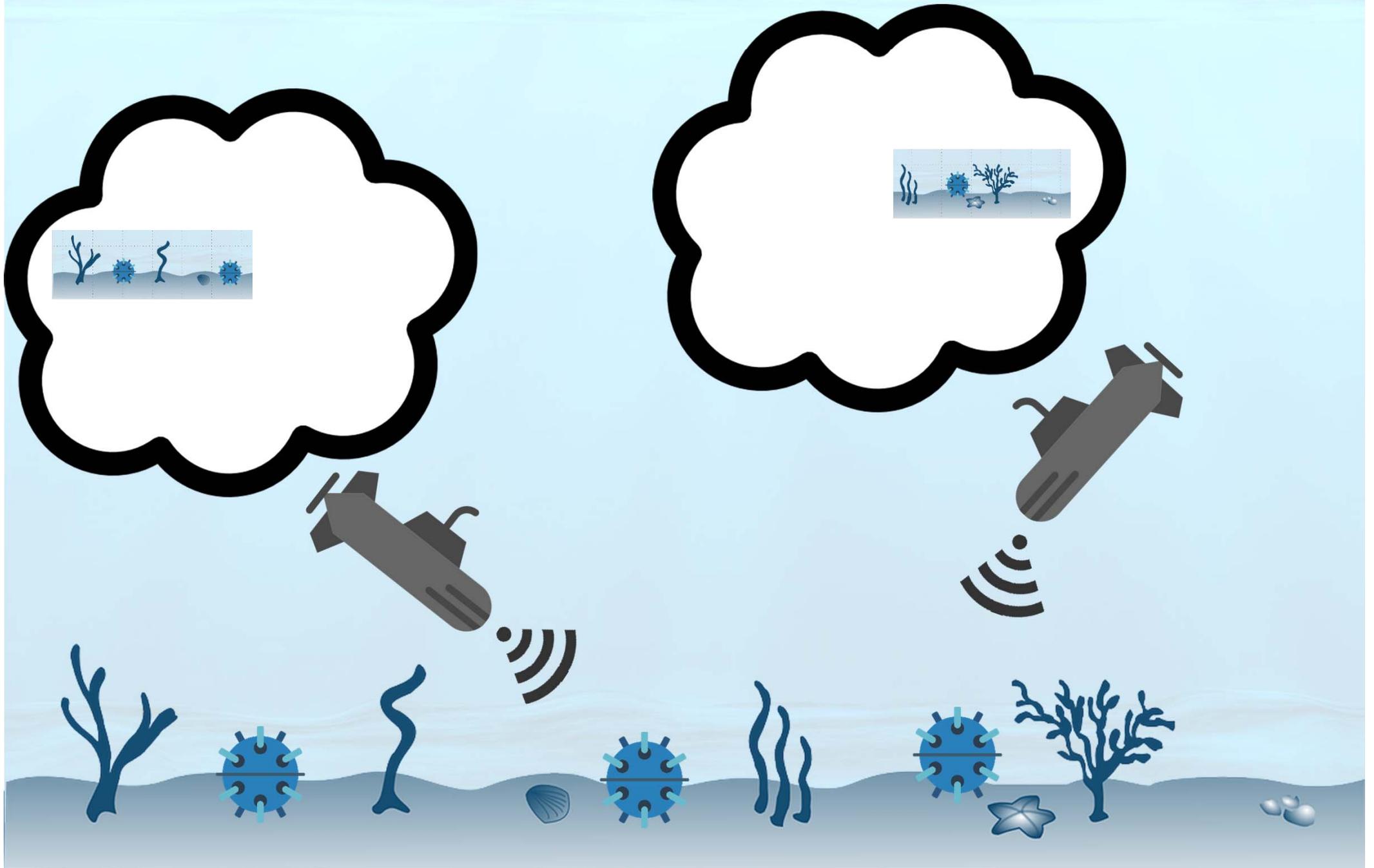
Fertigungsroboter von Kuka

Flexible Kommunikation bei Aufgabenänderung.

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



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



AUVs und Echtzeit-Kommunikation

- > 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
- > Echtzeitfähige Publish/Subscribe-Kommunikation
 - > Teil eines DFG-Projektes (VHR, AVA und IMD)
 - > 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

Aufgaben: Autonome Unterwasserfahrzeuge

- > Kooperative Navigation mehrerer AUVs
 - > Implementierung von Bewegungsmodellen
 - > Implementierung von Lokalisierungsalgorithmen
- > Energieverbrauch durch aktivierte Sensoren und Bewegung
 - > Implementierung von Energieverbrauchsmodellen für Sensorik und Bewegung
 - > Implementierung von Energieverbrauchsmodellen für Bildverarbeitungsalgorithmen
- > Opportunistische Kommunikation durch Akustikmodems
 - > Implementierung realistischer Unterwasserkommunikation
- > Implementierung in Simulator OMNet++ in C++
 - > Python zur Auswertung der Simulationsergebnisse
- > Echtzeitfähige Plattform für Prototypen
 - > Untersuchung der Eignung von Linux mit Preempt-RT-Patch bzw. von RTAI-Linux mit User-Space-Erweiterung

Aufgaben: Echtzeit-Publish/Subscribe

- > Datenmodelle und Filtermodelle
 - > Entwurf von Datenmodellen
 - > Entwurf zugehöriger Filtermodelle
 - > Prototypische Implementierung wahrscheinlich in C
- > Echtzeit-Eigenschaften
 - > Abschätzung der Worst-Case-Execution-Time (WCET)
 - > Analyse der Skalierbarkeit (#Filter, #Daten, #Subscriber)
 - > Testen/Messen der Implementierung
- > Echtzeitfähige Plattform für Prototypen
 - > Untersuchung der Eignung von Linux mit Preempt-RT-Patch bzw. von RTAI-Linux mit User-Space-Erweiterung

Organisatorisches

- > Zwei Teams mit jeweils bis zu 5 Personen
 - > Team A: Autonome Unterwasserfahrzeuge
 - > Team B: Echtzeit-Publish/Subscribe
- > 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.  23848 (Vorlesung) Neueste Entwicklungen der Informatik (Verteiltes Hochleistungsrechnen)
3.  23851 (Projekt) Projekt: Verteiltes Hochleistungsrechnen

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

> helge.parzyjegla@uni-rostock.de

> peter.danielis@uni-rostock.de