Motivation & Inhaltsübersicht

Vorlesung Verteilte Systeme
Wintersemester 2004/05

Fachbereich Informatik
cs.uni-salzburg.at

O.Univ.-Prof. Dipl.-Ing. Dr.techn. Wolfgang Pree

UNIVERSITÄT SALZBURG

Context

- Charakteristika verteilter Programmierung
- Konzeption der LVA
  Verteilte Systeme/Distributed Systems

Charakteristika verteilter Programmierung

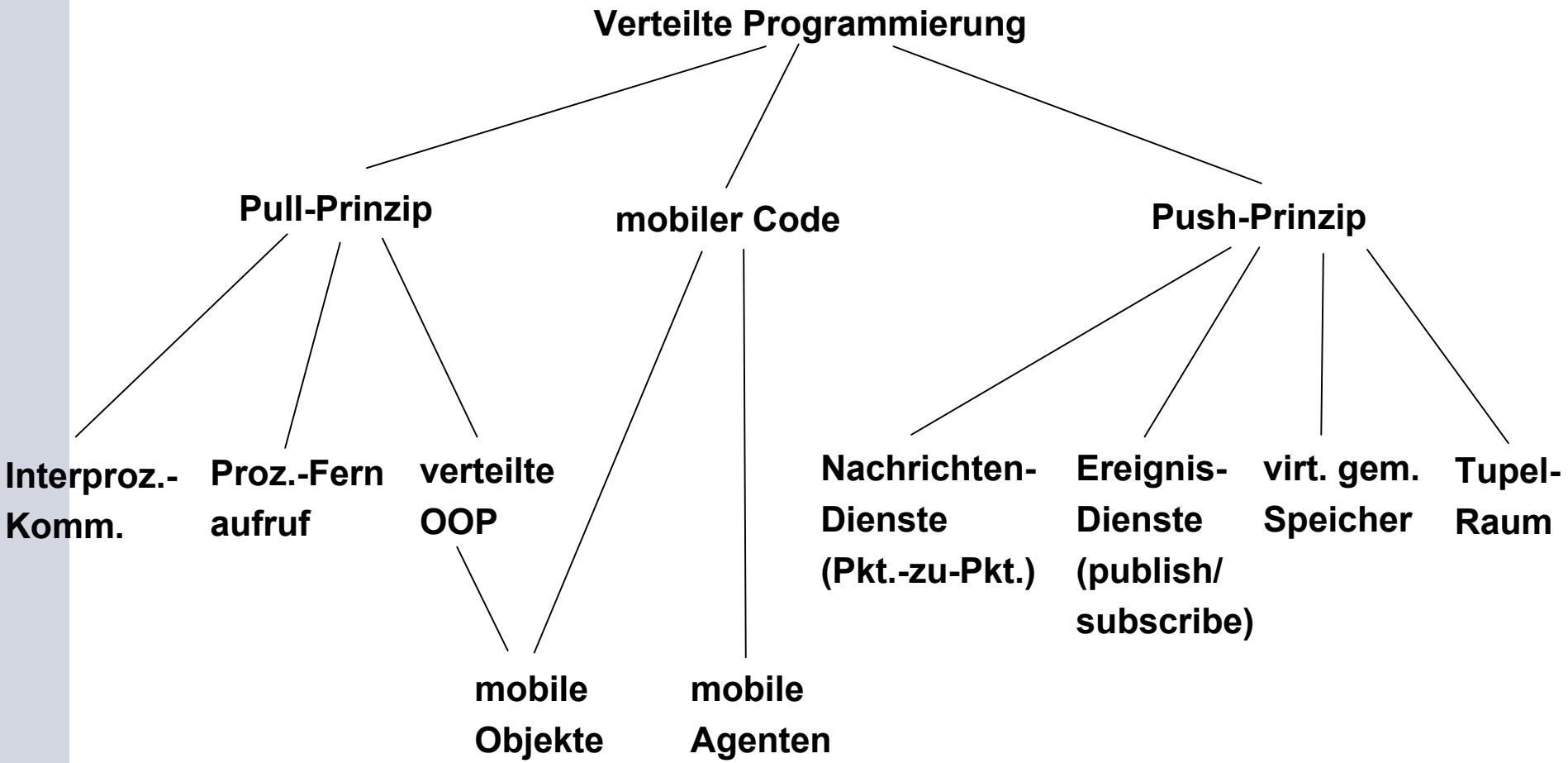
Definition

- Ein verteiltes (Datenverarbeitungs-)System besteht aus mehreren autonomen Prozessor-Speicher-Systemen, die mittels Datenaustausch kooperieren (Mühlhäuser 2002; Informatik-Handbuch, Hanser)
  - häufig sind die Prozessor-Speicher-Systeme vollständige Rechner
  - Mehrprozessorsysteme mit gemeinsamem Speicher sind KEINE verteilten Systeme
  - ebenso eng gekoppelte Mehrrechnersysteme, die die durch zentrale oder verteilte Koordination die Autonomie der Teilsysteme einschränken

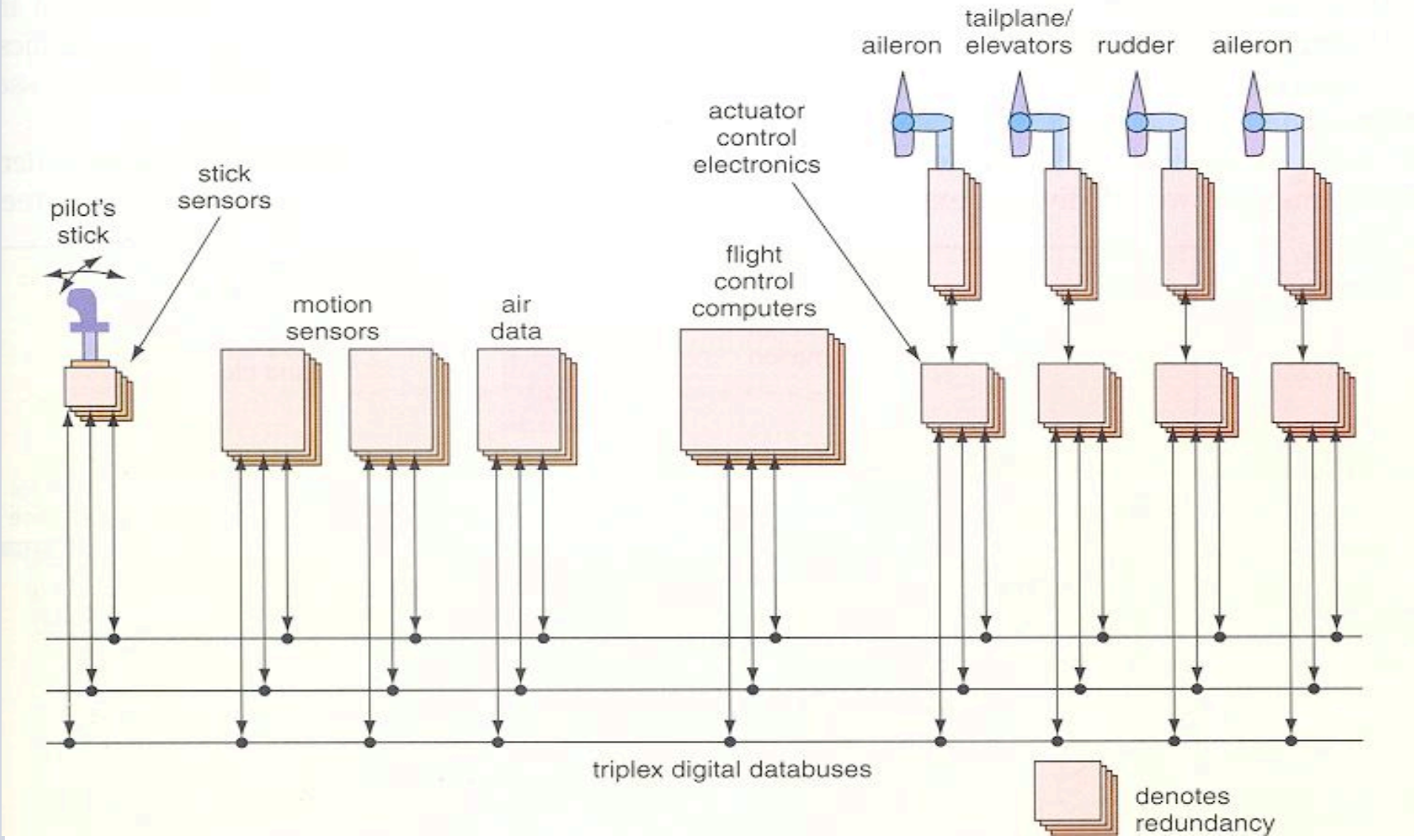
Charakteristika

- Fehlen einer aktuellen konsistenten Sicht auf das Gesamtsystem durch Autonomie, Parallelität und räumliche Entfernung der Knoten
- Heterogenität von Rechnerarchitekturen, Leistungskenngrößen, Kommunikationsprotokollen, Betriebssystemen und Benutzerschnittstellen

Ausprägungen verteilter Programmierung



Beispiel für ein paralleles, verteiltes System: Avionics Control



Kooperationspartner in Forschungsprojekten

- AVL List, Graz (avl.com)

distributed Formula-1 engine measurement & control application

- MagnaSteyr Engineering (magnasteyr.com) reengineering of current single-ECU system to a distributed one

Konzeption der Lehrveranstaltung Verteilte Systeme

Ziele

- Konzepte und Begriffe kennen und verstehen
- ingenieurmäßiges Herangehen an verteilte Software-Entwicklung lernen

key technologies

- high-level protocols
- distributed state sharing

Example isn‘t another way to teach, it is the only way to teach

Albert Einstein

topics

assumption: communication protocols are already well known

- object/service-oriented distributed system design (component standards; Web services)
- distributed shared memory
- consistency models
- distribution and consistency protocols
- distributed naming & directory services
- time-triggered synchronisation
- scheduling
- fault tolerance
- security issues (access control, security management, etc.)