

# Verteilte Systeme

Prof. Dr. Martin Becke

CaDS - HAW Hamburg

Version 0.9

# Contents

## 1 Vorstellung

- Person
- Person Aufgaben
- CaDS AG

## 2 Veranstaltung VS

- Anforderungen
- Ressourcen
- Prüfung
- Praktikum
- Praktikum - Best Practice
- Praktikumsversionen
- PVL Kriterien

# Verteilte Systeme

## Personen

- ▶ Ansprechpartner: Prof. Dr. Martin Becke (scimbe),  
Frank Matthiessen
- ▶ Einstiegspunkt: <http://www.becke.net>
- ▶ Gruppe: Kommunikation und verteilte Systeme

# Aktuelle Aufgaben

## Ämter/Gremien/Mitgliedschaften

- ▶ Studiengangskoordinator Informatik Technischer Systeme
- ▶ OE Beauftragter TI Informatik
- ▶ Mitglied (Stellvertreter) für Departmentrat
- ▶ Mitglied der Masterauswahlkommission
- ▶ Vorstand Förderverein Elektrotechnik und Informatik der HAW Hamburg e.V.
- ▶ Kontakt: Gerne Teams oder Mail:  
Martin.Becke@HAW-Hamburg.de

# CaDS

## Communication and Distributed Systems

- ▶ Schwerpunkte: Rechnernetze, Verteilte Systeme, Moderne Browser Kommunikation, Transport Protokolle, Diskrete Eventsimulation
- ▶ „Hobby“: Menschenrechte und Internetprotokolle (GW)
- ▶ Faziniert von „Open Source Projekten“
- ▶ Sekundäre Arbeitsgruppe: autosys research lab, FTZ Smart Systems

# Contents

## 1 Vorstellung

- Person
- Person Aufgaben
- CaDS AG

## 2 Veranstaltung VS

- Anforderungen
- Ressourcen
- Prüfung
- Praktikum
- Praktikum - Best Practice
- Praktikumsversionen
- PVL Kriterien

# Anforderungen für Einstieg VS

Was sollte ich mitbringen?

- ▶ Interesse daran, wie verteilte Anwendungen funktionieren
- ▶ Ein wenig Englisch ist hilfreich
- ▶ Ausreichende Kenntnisse der Programmiersprachen Java oder C++
- ▶ Kenntnisse der Module: AD, DB, Programmier\*, SE, BS, RN

# Ressourcen

Wo finde ich was?

- ▶ Zentraler Punkt: MS Teams (Code: 1mnf6rv)
- ▶ Zentrales Script: NEW Version 0.9 (Stabiler Inhalt, Format und Sprache werden weiter optimiert)
- ▶ Referenzliteratur und Basis früherer Vorlesungen <https://www.distributed-systems.net/index.php/books/ds4/>
- ▶ Folien mit der Struktur der Vorlesung



# Prüfungsform

Abhängig der Teilnehmer

- ▶ Default: Klausur
- ▶ 90 Minuten
- ▶ Alles Klausurrelevant (Unterlagen, Vorlesung, Praktikum)
- ▶ Aufgaben stark mit Fokus auf Transfer

# Praktikumsinformationen

## Entwicklungszyklus in VS - Empfehlung

- ▶ Mit Standalone Applikation Design starten
- ▶ Trennen in Client/-Server mit Middleware
- ▶ Orchestrieren mit RPC-Architektur
- ▶ Optimieren mit verteilten Algorithmus

# Praktikumshinweise

Was sollte ich beachten?

- ▶ Maximale Gruppengröße 4 (auch 2 oder 1 möglich).
- ▶ Praktikum ist zeitaufwendig
- ▶ Praktikum ist ein iterativer Prozess
- ▶ Praktikum hat keine Anwesenheitspflicht, nur verpflichtende (2x) 1:1 Gespräche
- ▶ Thema muss bis 27.10 gewählt sein

# Freies Praktikum

## Anforderungen

- ▶ Maximale Gruppengröße 4 (auch 2 oder 1 möglich).
- ▶ Eine API Anbindung an einen externen Dienst
- ▶ Loadsharing
- ▶ Service Orchestrierung über RPC
- ▶ AI Komponente
- ▶ Auf der ICC ausführen (Wenn vorhanden)
- ▶ Muss vom Lehrenden akzeptiert werden

# Relatives freies Praktikum

## Anforderungen

- ▶ Maximale Gruppengröße 2 - 3
- ▶ Visualisierung eines Konsens Algorithmus
- ▶ Flooding oder Gossip
- ▶ Container-Lösung (Keine Simulation)
- ▶ Theoreisch beliebig skalieren
- ▶ Messungen zur Verdeutlichung der Leistungsfähigkeit
- ▶ Muss vom Lehrenden akzeptiert werden

# Default Praktikum

## Anforderungen

- ▶ Maximale Gruppengröße 2 - 3
- ▶ Umsetzung Dining Philosophers
- ▶ Kein gemeinsamer Speicher
- ▶ Kein (zentraler) Koordinator
- ▶ Eigene RPC Lösung
- ▶ Abstimmung über Nachrichten (Kooperatives Verfahren)
- ▶ Performance Messungen
- ▶ Anzahl der Speisungen mit CFRD Counter

# PVL

## Anforderungen

- ▶ Vollständige Dokumentation (Vorschlag ARC42) - Dokumentation muss zum Code passen und Code zur Dokumentation
- ▶ Dokumentation sollte mindestens Komponenten, Klassen und Interaktionen umfassen (Vorschlag UML 2.5)
- ▶ Code Base, mit Test (Martin Becke und Frank Matthiesen müssen eingetragene Developer sein)
- ▶ Funktionale Umsetzung der beschriebenen Use Cases
- ▶ Es sollten alle VS Standards nach Tanenbaum/van Steen eingehalten werden, oder wie sie auch im Skript beschrieben sind
- ▶ Layer und Tiers