



# Architektur von Anwendungssystemen

Prof. Dr.-Ing. Stefan Tai, Sebastian Werner, Karl Wolf

# Agenda

- Begrüßung und Kurzvorstellung des Fachgebiets ISE
- Organisatorisches
- Kahoot-Quiz
- Einführung Architektur von Anwendungssystemen

# Das Fachgebiet “Information Systems Engineering”

...ist eines der ca. 70 Fachgebiete der Fakultät IV Elektrotechnik und Informatik,

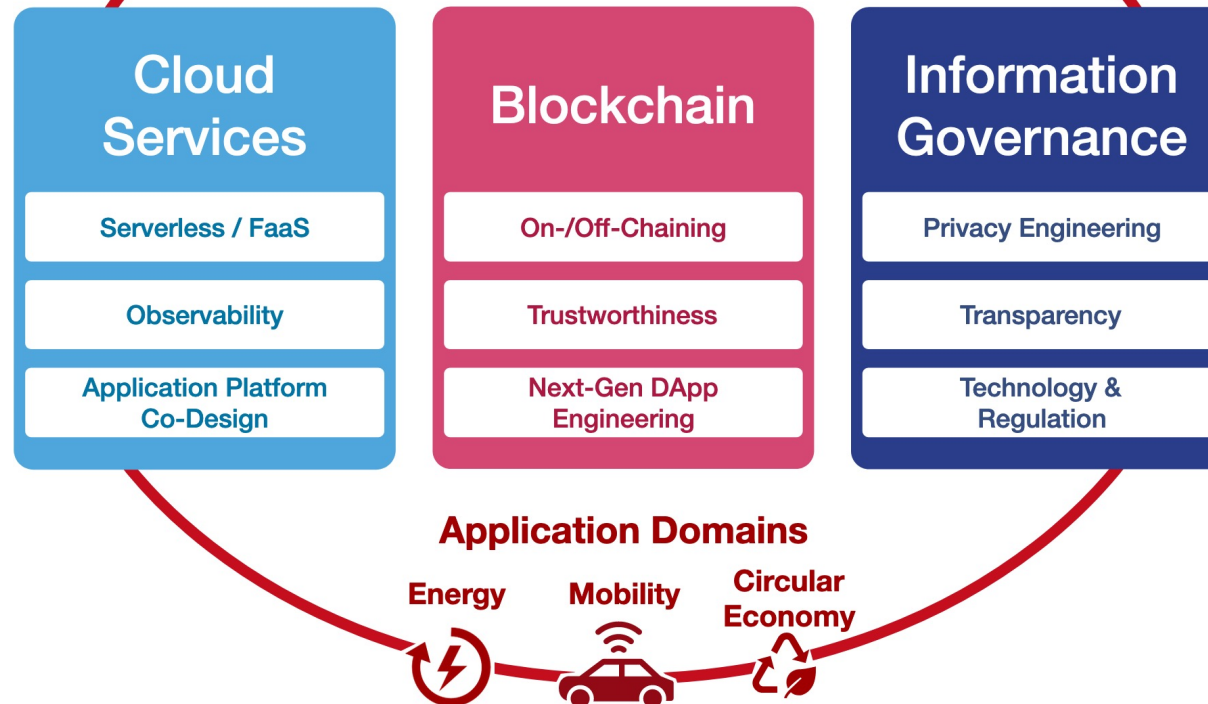
lehrt und forscht unter der Leitung von Prof. Dr.-Ing. Stefan Tai im Bereich des **qualitätsgetriebenen, kontinuierlichen Engineerings zukunftsweisender verteilter Softwaresysteme**, in nationalen und internationalen „Leuchtturm-“ Forschungsprojekten und Kooperationen mit Partnern in Wirtschaft und Wissenschaft.

Aktuelle Schwerpunkte sind

- Cloud-native Plattformen und –Anwendungen
- Blockchain-Technologien und –Innovationen
- Privacy Engineering

Das Fachgebiet ISE ist im Sommersemester 2022 *letztmalig* verantwortlich für das Modul Anwendungssysteme.

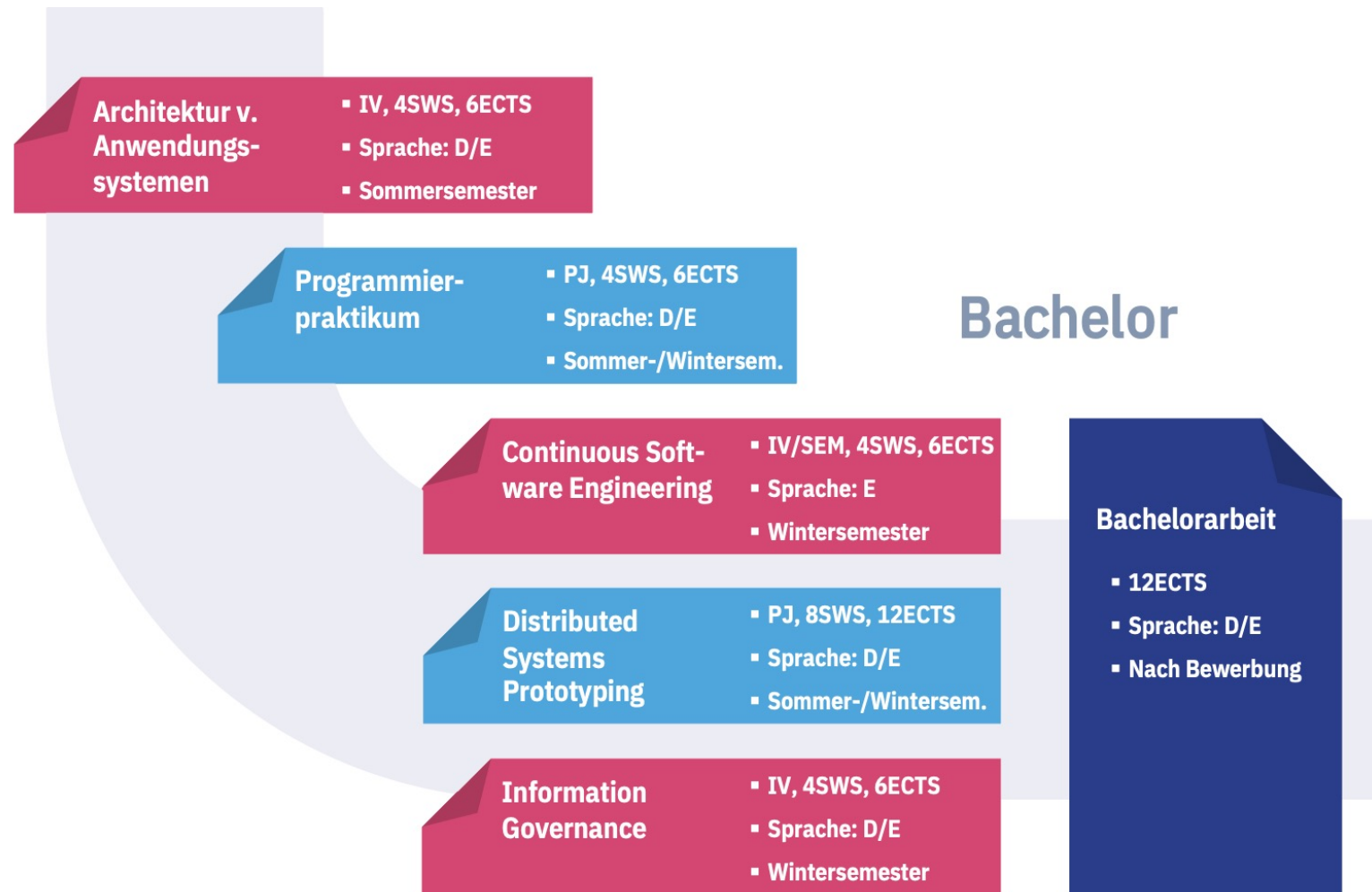
# Quality-driven, Continuous Software Systems Engineering



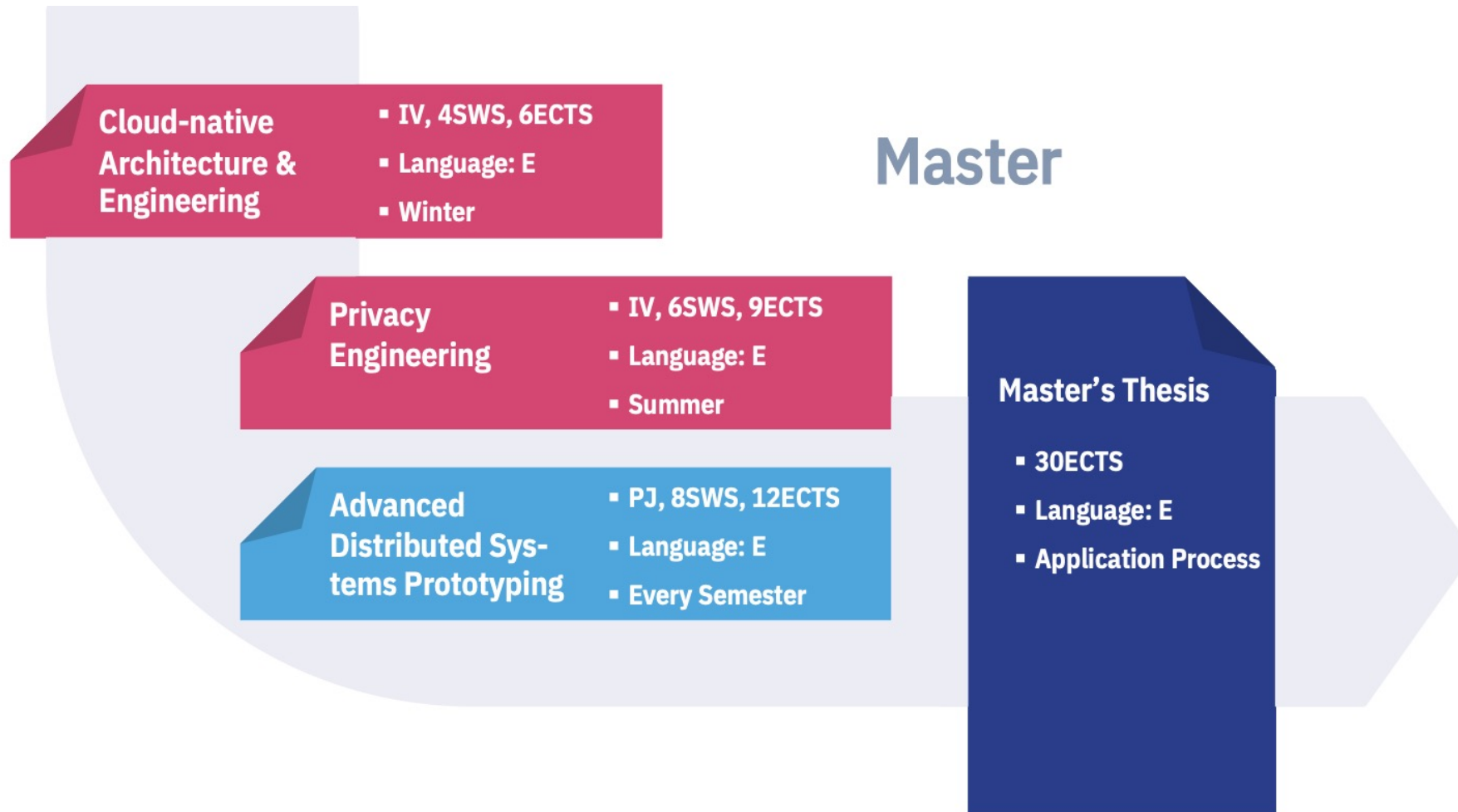
[www.ise.tu-berlin.de](http://www.ise.tu-berlin.de), demnächst: [www.tu.berlin/ise](http://www.tu.berlin/ise)



# ISE-Lehrangebot im Bachelor (Stand 04/22)



# ISE Courses for Master Students (as of 04/22)



# Das Team für „Architektur von Anwendungssystemen“ im Sommersemester 2022



Sebastian Werner



Karl Wolf



Prof. Dr. Stefan Tai



Alexander Schmidt

# Organisatorisches



# Das Modul “Architektur von Anwendungssystemen”

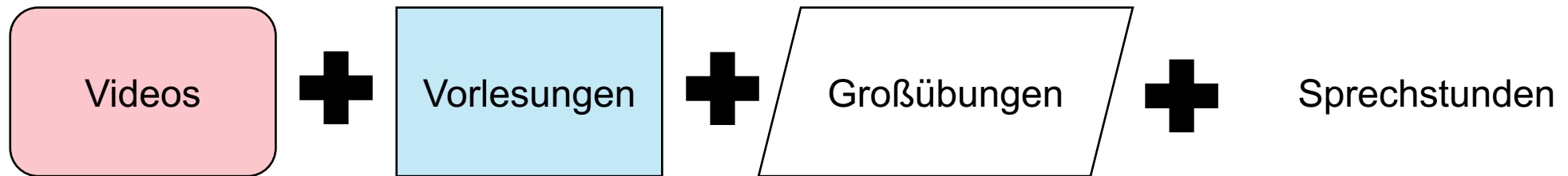
- Pflichtmodul im Bachelor Wirtschaftsinformatik
- Pflichtmodul für die Vertiefungsrichtung IKS im Bachelor Wirtschaftsingenieurwesen
- **Lernergebnisse [MOSES]:**  
Studierende kennen nach dem erfolgreichen Abschluss dieses Moduls unterschiedliche Architekturen verteilter Anwendungssysteme und ausgewählte, diesen zugrunde liegenden, Middleware- und Web-Technologien. Sie sind in der Lage, Anwendungsarchitekturen unter dem Einsatz dieser Technologien zu gestalten und zu bewerten.
- **Lehrinhalte [MOSES]:**  
Das Modul vermittelt Wissen über verteilte Anwendungssysteme aus Anwendungs-, Technologie- und Unternehmensorganisationsperspektive. Der Schwerpunkt liegt auf der Gestaltung und Bewertung von Software-Architekturen, dabei insbesondere der Auswahl und dem Einsatz unterschiedlicher Middleware- und Web-Technologien sowie der Entwicklung verschiedener Software-Komponenten unter Berücksichtigung kritischer nicht-funktionaler Eigenschaften. Fundamentale Paradigmen zur verteilten Anwendungsentwicklung, wie beispielsweise asynchrone Kommunikation und Service-orientierte Architekturen, werden dazu eingeführt. Die Themen der Vorlesung werden in der Übung mittels praktischer Programmieraufgaben vertieft. Die Übungsaufgaben werden je nach Teilnehmerzahl in Einzelarbeit oder Kleingruppen durchgeführt.

# Aufbau im SoSe2022 inkl. Ausnahmeregelung

**Prüfungsform:** Schriftliche Prüfung (§ 53, AlgStuPo), Notenschlüssel 2

**Präsenztermin:** Jeden Dienstag, 12-14 Uhr [H 1058](#)

**Ablauf:**

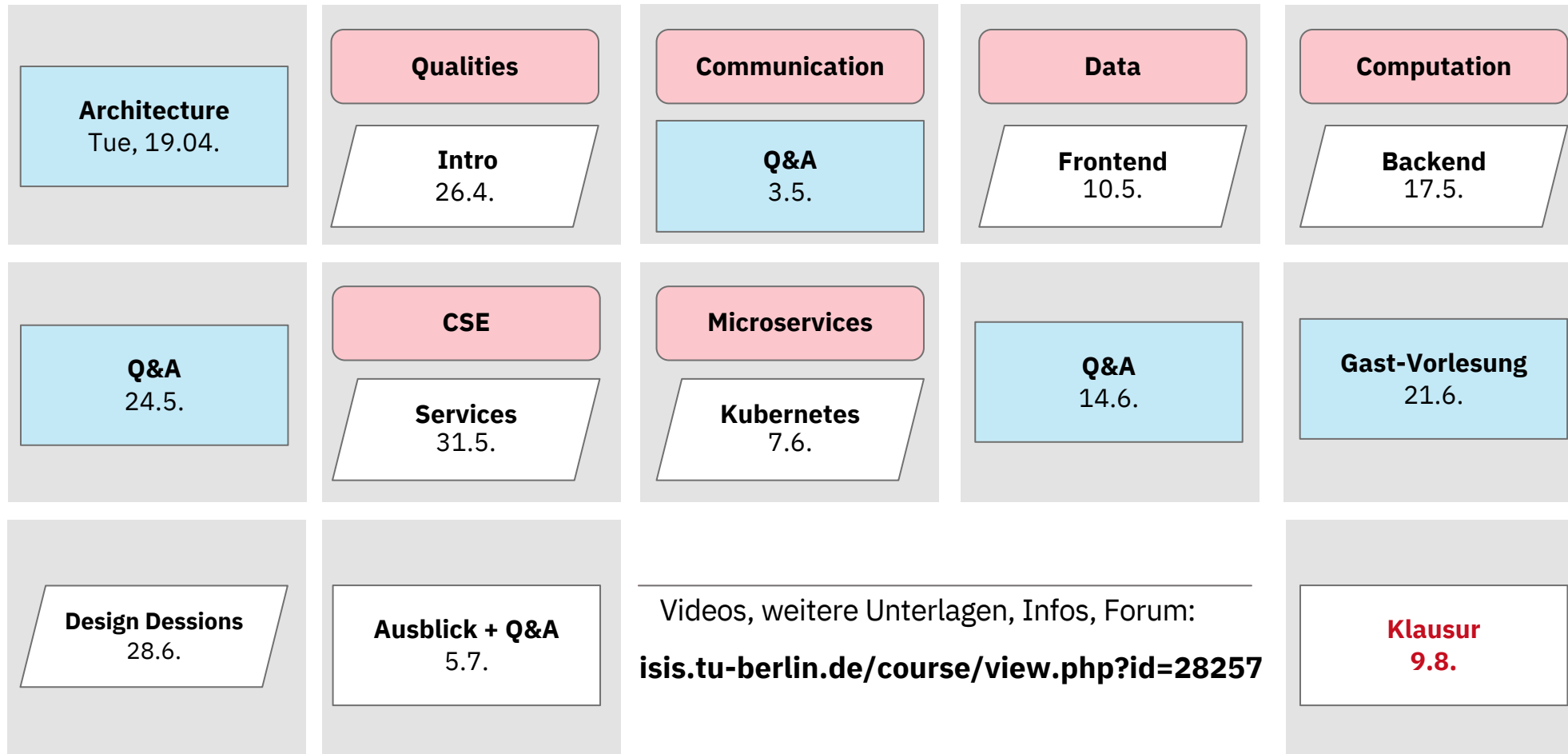


**Schriftlicher Test:**


- Dienstag, 9.8.2022

Punkte	$\geq 95$	$\geq 90$	$\geq 85$	$\geq 80$	$\geq 75$	$\geq 70$	$\geq 65$	$\geq 60$	$\geq 55$	$\geq 50$	$< 50$
Note	1.0	1.3	1.7	2.0	2.3	2.7	3.0	3.3	3.7	4.0	5,0

# Verlaufsplan



One Week

 Vorlesung

 Videos

 Großübung

**Präsenztermine:**  
dienstags, 12-14 Uhr  
[H 1058](#)

Für Ankündigungen und sämtliche Unterlagen zur Veranstaltung,  
treten Sie bitte folgendem ISIS-Kurs bei:

<https://isis.tu-berlin.de/course/view.php?id=28257>

# Wichtiges – Zusammenfassung

- ISIS-Kurs: <https://isis.tu-berlin.de/course/view.php?id=28257>
- **Präsenztermin:** Dienstag, 12.15 Uhr, Raum H 1058
- Keine Tutorien (Ausnahme im SoSe22 aufgrund fehlender Tutor:innen)
- **Prüfungsform:** Schriftliche Prüfung
- **Prüfungstermin: 9.8.2022**
- **Anmelde- und Rücktrittsfrist für die Modulprüfung: ab Mitte des Semesters**

# Kahoot Quiz

Besuchen Sie

**<https://kahoot.it/>**



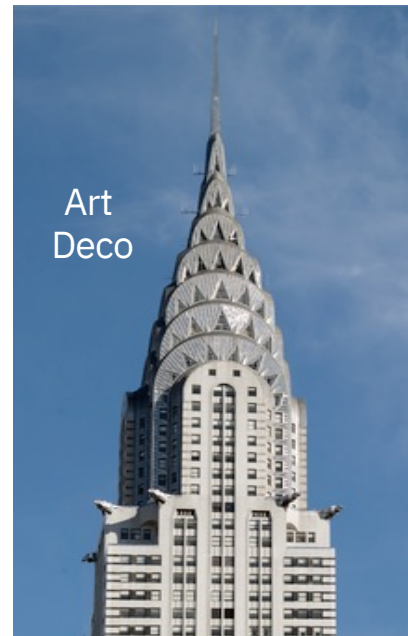


# Architecture

# Architecture



# Architectural Styles

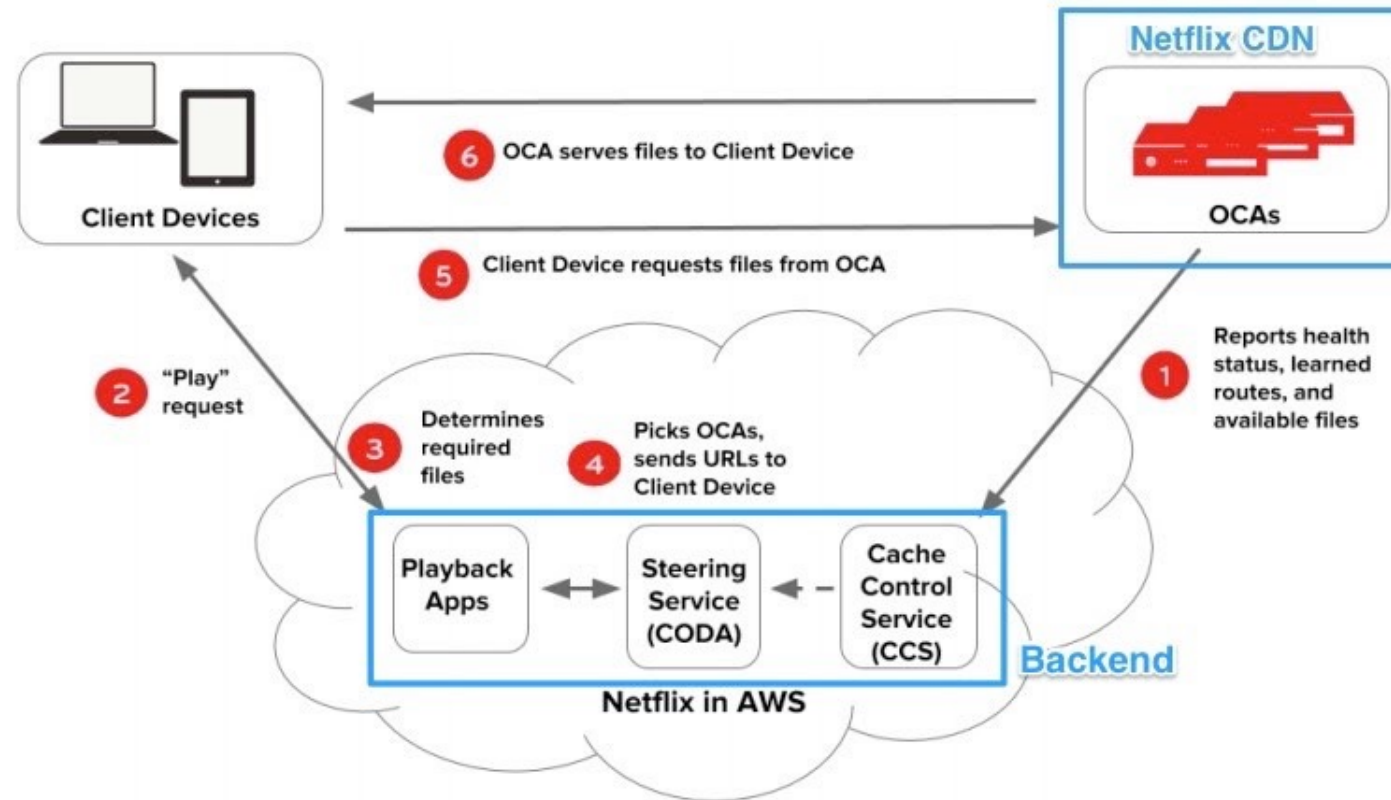




# Buildings and Infrastructures

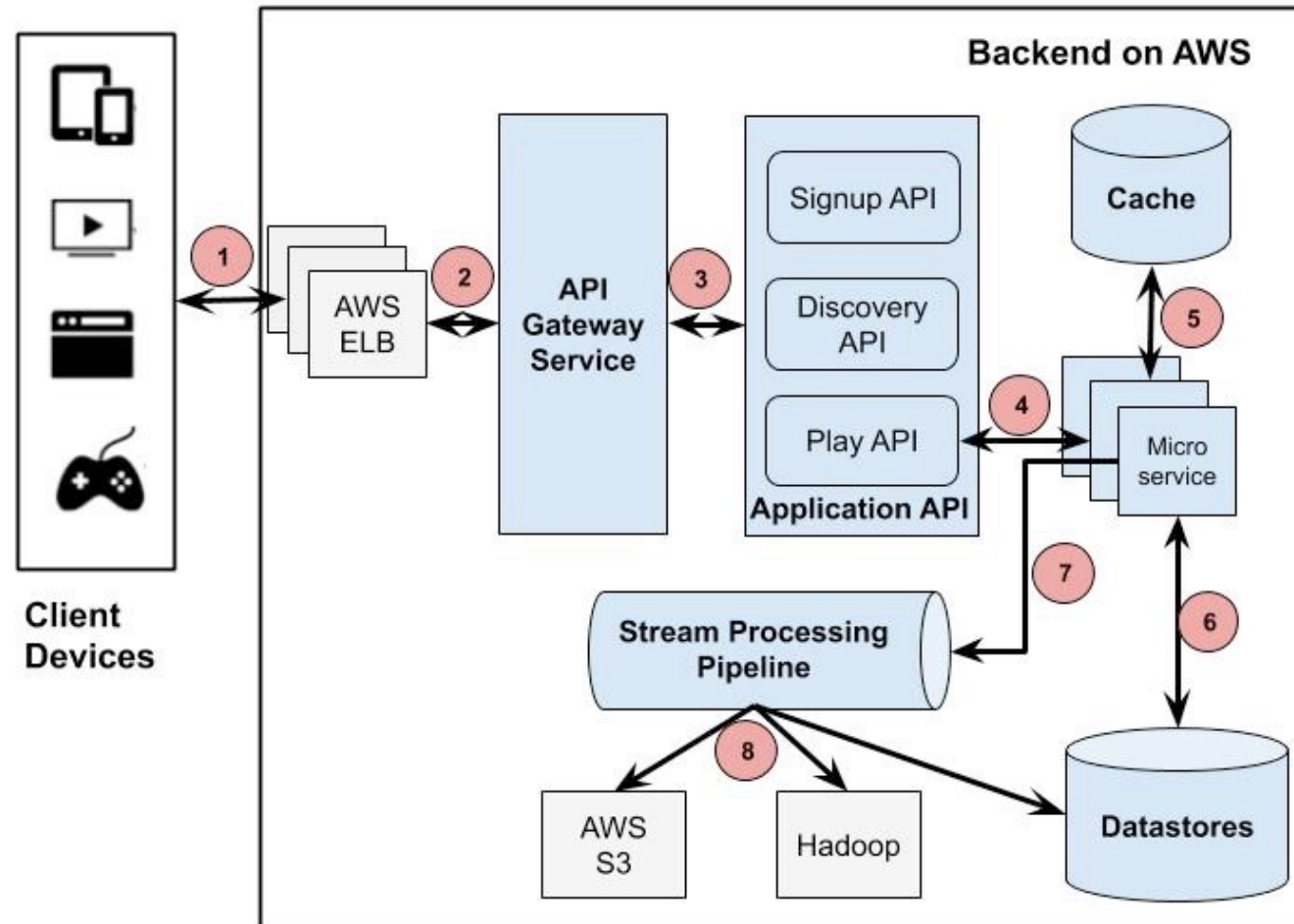


# A “Typical” Architecture Diagram – Netflix’ Playback Architecture for Streaming Videos



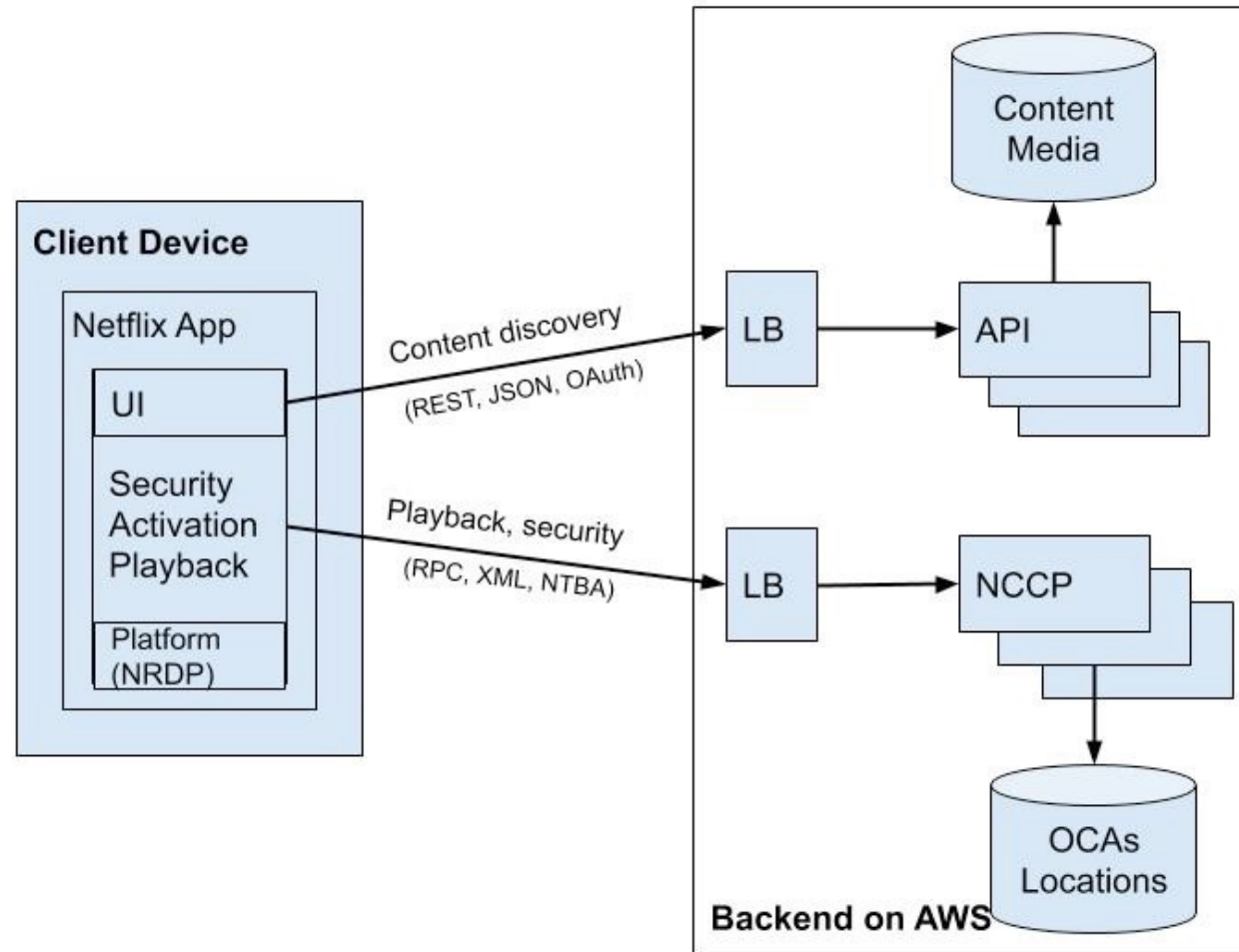
Source: <https://medium.com/swlh/a-design-analysis-of-cloud-based-microservices-architecture-at-netflix-98836b2da45f>

# Zooming into the Backend on AWS

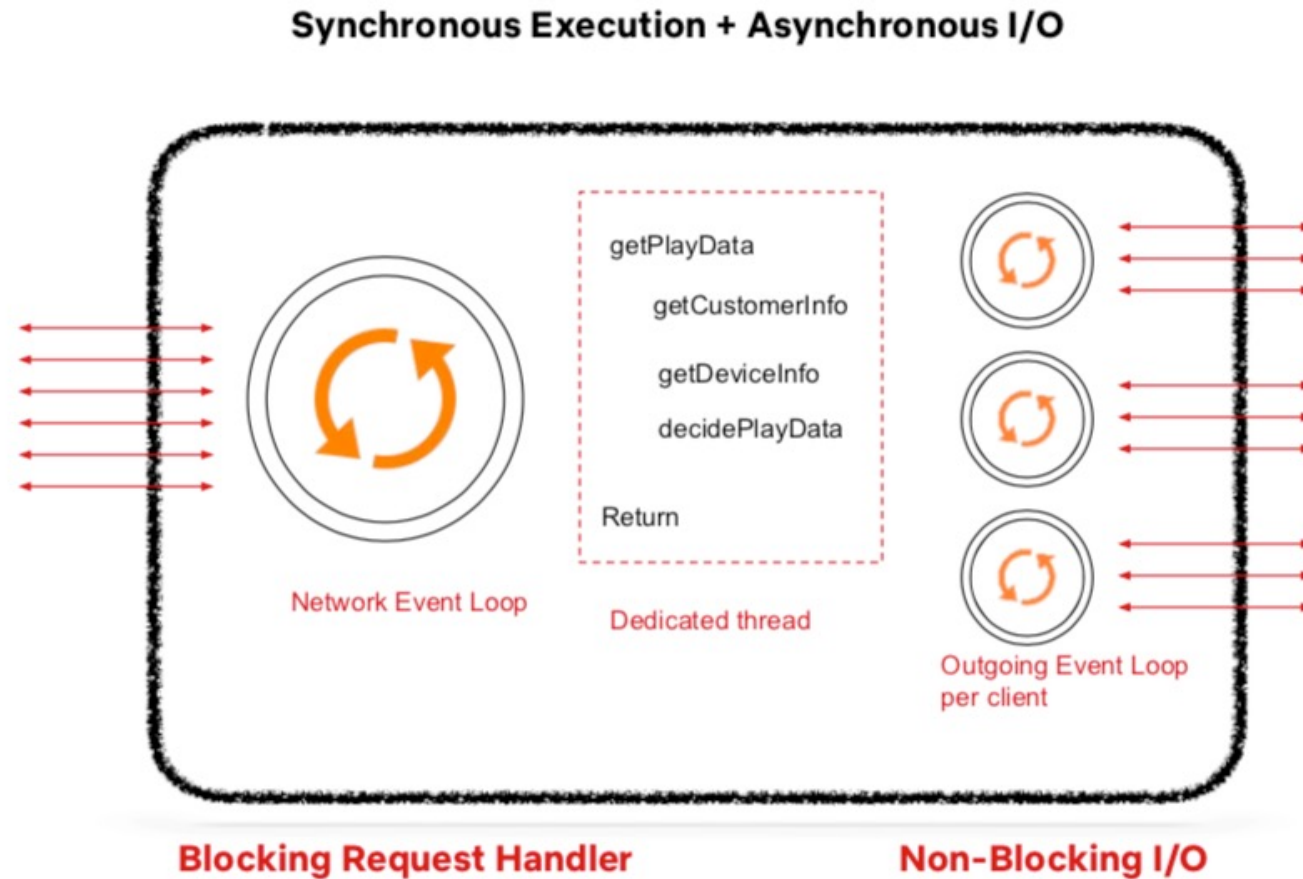




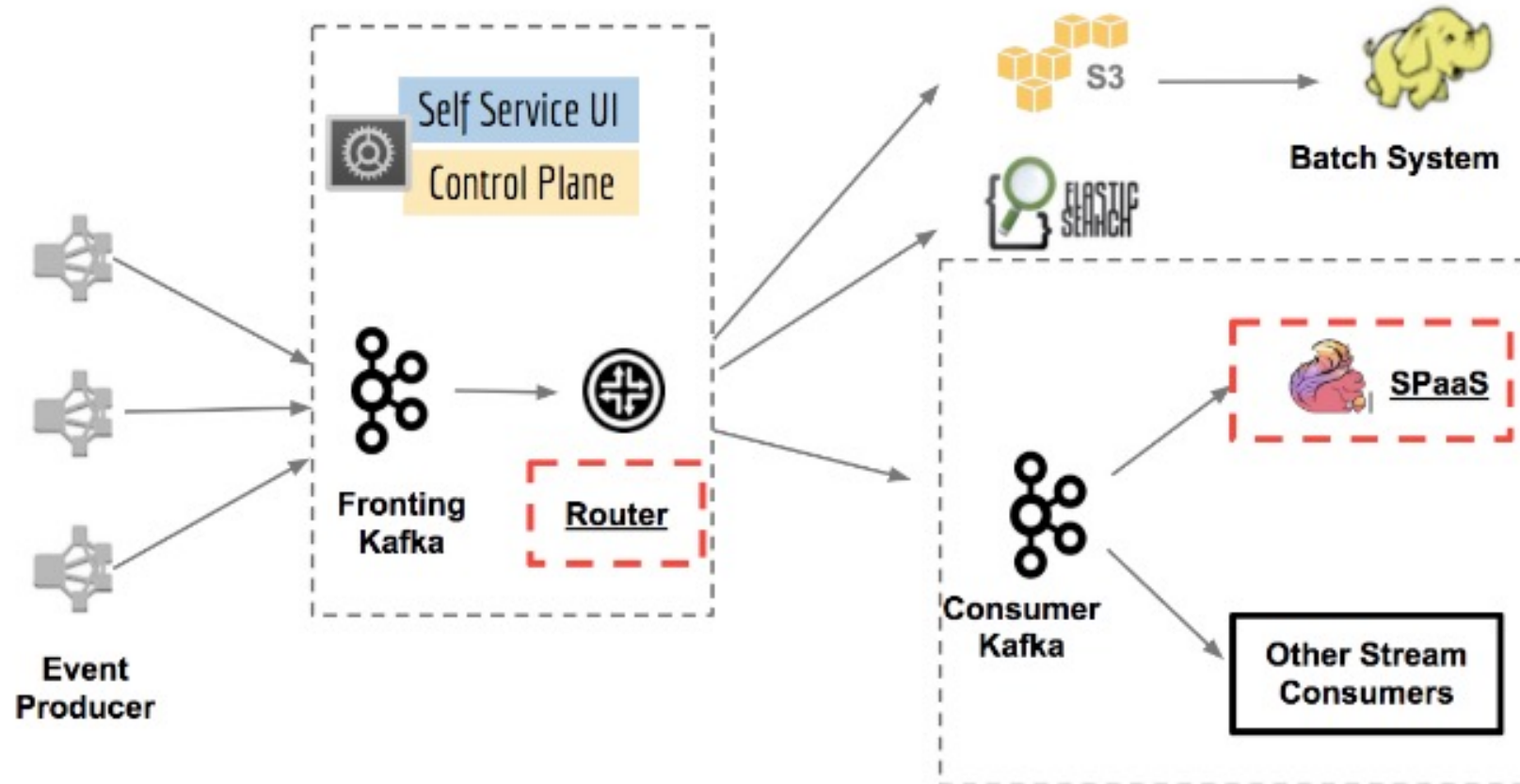
# Zooming into the Client-Backend Connection



# Zooming into I/O of Application APIs



# Zooming into the Stream Processing Pipeline for Business Analytics



# Getting Lost?





# “Anwendungssysteme 2022” to your rescue!



# What is Software Architecture, then?

*Most generally:*

*“The software architecture of a system is the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both.”*

Source: Bass et al, Software Architecture in Practice n(Fourth Ed), Addison-Wesley, 2021



# Impact of Software Architecture

- Inhibits or enables a system's quality attributes
  - Performance, Modifiability, Security, Scalability, Reusability
  - Allows prediction of some qualities
- Implications for change management
  - Scope of changes: local (single component), non-local (multiple components), architectural (interaction of components)
  - Goal: most common changes are local
- Enables communication among stakeholders
  - Different stakeholders are concerned with different characteristics
  - Architecture (to some extent) should be understandable by non-technical people
- Carrier of fundamental design decisions and constraints
  - Determines lifecycle of the whole system
  - Implicit constraints on an implementation

# What Makes a “Good” Architecture?

- There is no such thing as an inherently good or bad architecture.
- Architectures are either more or less fit for some purpose.
- Architectures can be evaluated but only in the context of specifically stated goals.
- Every system, however, has a software architecture.  
(But this architecture may be documented and disseminated, or it may not be.)

# Videos to study

Woche	Block	Themenblöcke	Präsenztermin (Di, 12-14 Uhr)
19. Apr	1	Intro + Orga & "Architecture": Components, Services, APIs	Vorlesung
26. Apr		"1 Qualities": <b>Non-Functional Requirements, Middleware, Cross-Cutting Concerns</b>	GÜ: Projekt & Tooling
03. May	2	"2 Communication": <b>Network, Messaging, RPC, GRPC</b>	Q&A - Block 1
10. May		"3 Data": SQL, NoSQL, <b>ACID, BASE</b>	GÜ: "Frontend" HTML/JSON /Caching
17. May		"4 Computation": Bare-metal, Virtual/Cloud, Container, Kubernetes, Serverless	GÜ: Backend
24. May			Q&A - Block 2
31. May	3	"5 CSE": Agile, <b>DevOps</b> , IaC	GÜ: Services
07. Jun		"6 <b>Microservices</b> ": Bringing it all together	GÜ: Kubernetes
14. Jun		Guest Lecture: Microservices+Docker in Practice	Vorlesung
21. Jun			Q&A - Block 3
28. Jun			GÜ: Design Decisions
05. Jul		Klausurvorbereitung	Vorlesung

## Videos

Zur Wiederholung und Vertiefung gibt es aus vergangenen Semester noch Videos zu vielen Themen:

### 01 Qualities

- **Non-Functional Requirements**
- **Middleware & Remote Procedure Calls (RPCs)**
- **Cross-Cutting Concerns**

# Wichtiges – Zusammenfassung

- ISIS-Kurs: <https://isis.tu-berlin.de/course/view.php?id=28257>
- **Präsenztermin:** Dienstag, 12.15 Uhr, Raum H 1058
- Keine Tutorien
- **Prüfungsform:** Schriftliche Prüfungen
- **Prüfungstermin: 9.8.2022**
- **Anmelde- und Rücktrittsfrist Modulprüfung: ab Mitte des Semesters**
- Nächste Woche 1. Großübung: **Wenn vorhanden Laptop mitbringen.**