Technische Hochschule Ingolstadt

Fakultät für Elektrotechnik und Informatik

Zukunft in Bewegung

Prototype einer Prozesssicht

Architektur- und Entwurfsmuster der Softwaretechnik

Prof. Dr. Bernd Hafenrichter

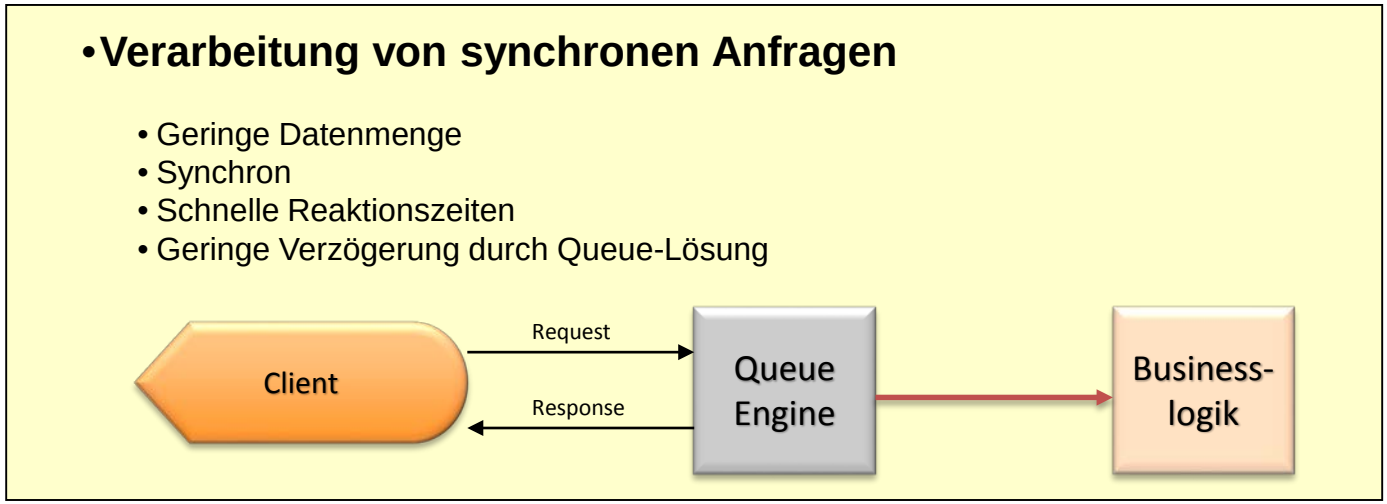
Aufgabenstellung

- Entwickeln Sie ein Konzept für eine Komponente „QueueEngine“ die auf dem Prinzip eins auftragsbasierten Servers arbeitet. Definieren Sie hierzu die statische Sichtweise (UML-Klassendiagramm) sowie die dynamische Sichtweise (Sequence-Diagramm) welches die Details der Verarbeitung zeigt.
- Falls möglich: Implementieren Sie das Konzept prototypisch

- **Anforderung 1:**
  - Die Komponenten „QueueEngine“ soll eine möglichste gute Wiederverwendbarkeit aufweisen
  - Hierzu soll dass Prinzip „Trennung von A und T“ Software angewendet werden.
  - Die QueueEngine soll nichts über Art und weise Wissen wie eingehende Anfragen übermittelt verarbeitet werden.
  - Die QueueEngine soll ein Schnittstelle bereitstellen über welche Aufträge übergeben werden können.

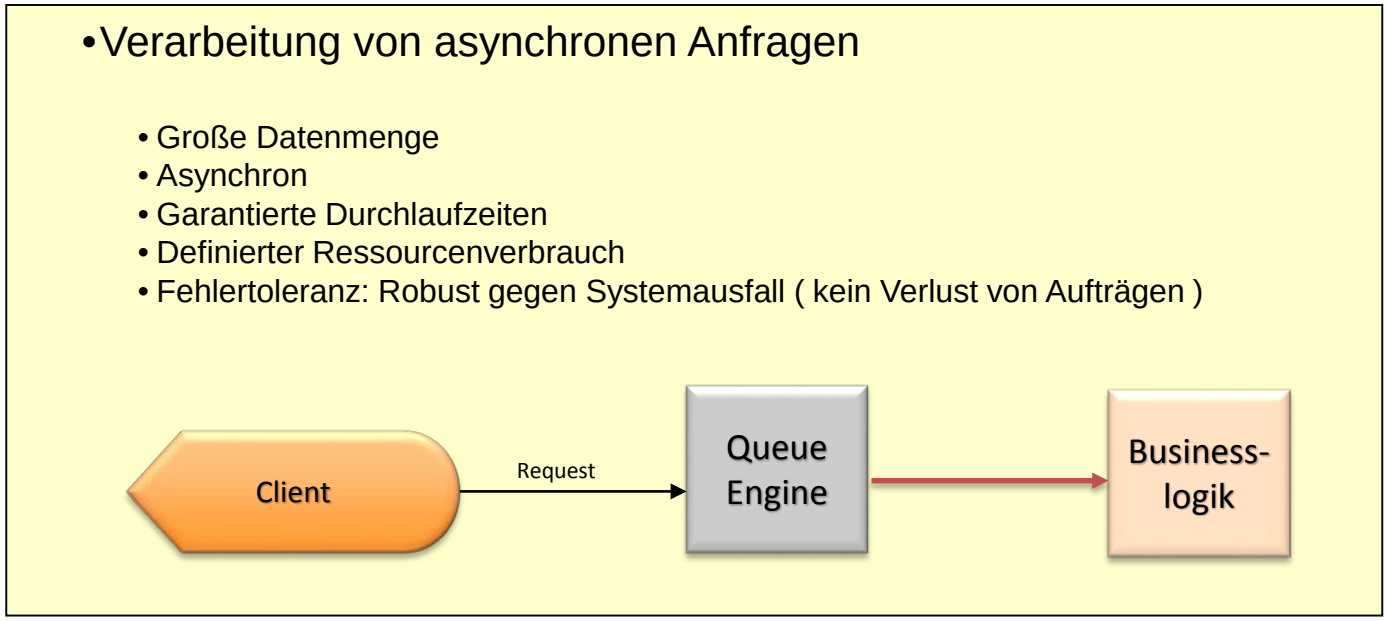
Aufgabenstellung

- **Anforderung 2 - Verarbeitung von synchronen Aufträgen**
  - Die Komponenten „QueueEngine“ muss in der Lage sein synchrone Anforderungen zu verarbeiten. D.h. der Client warte bis das Ergebnis der Verarbeitung vorliegt.



Aufgabenstellung

- **Anforderung 3 - Verarbeitung von asynchronen Aufträgen:**
  - Die Komponenten „QueueEngine“ muss in der Lage sein asynchrone Anforderungen zu verarbeiten. D.h. der Client übergibt einen Auftrag zur Verarbeitung ohne auf das Ergebnis zu warten.



Aufgabenstellung

- **Anforderung 4 - Verarbeitungsreihenfolge:**

  Die QueueEngine soll folgende Verarbeitungsvariante für eingehende Aufträge anbieten: First-In-First-Out und parallel Verarbeitung.

  - **First-In-First-Out:** Eingehende Aufträge sollen in der Reihenfolge in der die Aufträge eintreffen verarbeitet werden. Eine parallele Ausführung ist zu verhindern.
  - **Parallel:** Eingehende Aufträge können gleichzeitig verarbeitet werden können.
  - **Achtung**: Der Client entscheidet darüber welcher Quality-Of-Service benötigt wird. Alle Kombinationen sollen gleichzeitig möglich sein.

Aufgabenstellung

- **Anforderung 4 - Skalierbarkeit:**
  - Das System soll mit steigender Last skalieren und die Ressourcen der Maschine optimal ausnutzen

- **Anforderung 5 - Robustheit:**
  - Das System soll in der Lage sein bei Überlast den Dienst weiterhin Aufrecht zu erhalten.
  - Das System soll robust gegenüber Systemausfällen sein

- **Anforderung 6 - Latenz:**
  - Die durch das System erzeugten Latenzzeit sollen möglichst gering gehalten werden.