

Projektanleitung

Inhalt

1 Projektbeschreibung.....	2
2 Übersicht zu Anforderungen und Aufgaben	3
3 Termine und Projektablauf	4
4 Bewertungskriterien.....	5

Achtung: Lesen Sie diese Anleitungen und Informationen sehr gewissenhaft.

1 Projektbeschreibung

Im Projekt Web-basierte Anwendungssysteme sollen Sie im Team eine Anwendung aus verteilten Web Services erstellen. Ziel ist also ein Anwendungssystem aus **Distributed Web-based Services** zu erstellen. Die folgende Abbildung soll Ihnen hierzu einen ersten Eindruck vermitteln.

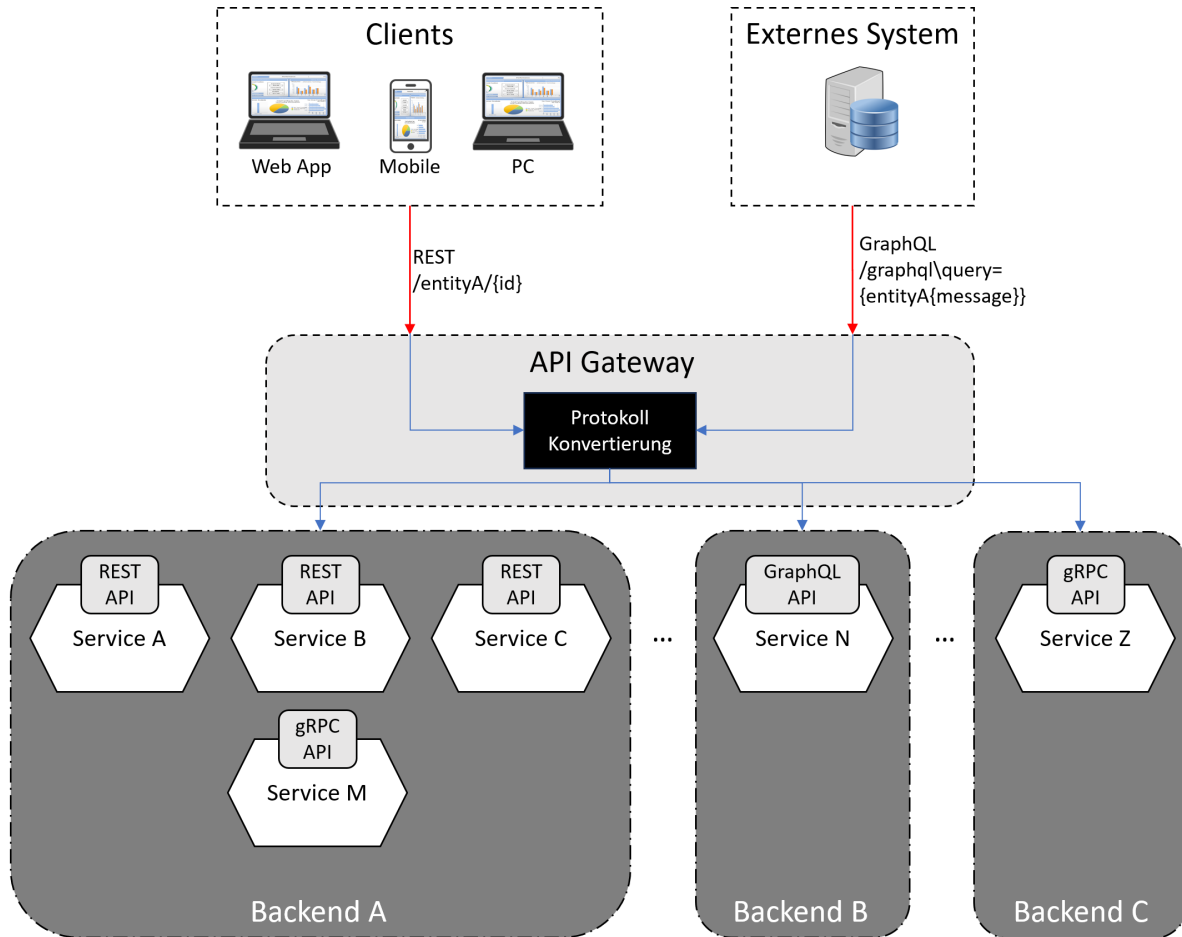


Bild 1: Beispielfhafte Übersicht zu Distributed Web-based Services

Zentral ist ein sogenanntes API Gateway dargestellt, welches die eintreffenden Anfragen (Requests) an die zugehörigen einzelnen verteilten Services weiterreicht und übersetzt bzw. anpasst. Die einzelnen nachgelagerten Services bieten jeweils spezialisierte Funktionalitäten, wie z.B. Authentifizierung, Bezahlung, Verwaltung von Sitzungen oder ein Repository zur Aufbewahrung von geordneten Informationen wie z.B. Dokumenten.

Zum Verständnis können Sie sich gerne folgendes Video ansehen: [What is API Gateway?](#)

- Die spezialisierten Funktionalitäten, die für Ihre Anwendung benötigt werden, müssen Sie im Team identifizieren und während der Planungsphase festlegen.
- Sie entscheiden im Team welchen Umfang Ihre Anwendung und verteilten Web Services haben.
- Ein API Gateway ist zwingend zu realisieren.
- Es müssen mind. zwei unterschiedliche API Technologien (REST, GraphQL, gRPC) eingesetzt werden.
- Es müssen mind. zwei unterschiedliche Möglichkeiten der Anwendungsnutzung clientseitig bzw. externes System realisiert werden, z.B. Konsole, SPA, PWA, etc.

2 Übersicht zu Anforderungen und Aufgaben

Die folgende Liste fasst die Anforderungen und Projektaufgaben kurz zusammen:

1. Projektplanung über den gesamten Projektverlauf muss zu Beginn (**innerhalb der ersten Woche**) der Projektlaufzeit feststehen und in GitHub erstellt werden, dazu gehört:
 - *Meilensteinplanung* - jedes Teammitglied muss Meilensteine definieren und in GitHub erstellen
 - *Arbeitspaketplanung* (inkl. zeitliche Übersicht und Zuordnung zu Bearbeitern und Meilenstein) - jedes Teammitglied muss Arbeitspakete definieren und in GitHub erstellen
 - muss in GitHub Projects als Roadmap direkt erkennbar sein und übersichtlich mit zeitlichen Zuordnungen und Fortschritten der Meilensteine und Arbeitspakete erstellt werden
2. Entwurf/Architektur des Anwendungssystems mit allen Schnittstellen und APIs
 - grafische Darstellung inkl. Erläuterung muss in GitHub als Markdown-Datei erstellt werden
 - jeder einzelne Web Service muss in einem eigenen Markdown-Dokument definiert und beschrieben sein. Dazu zählen zwingend Funktionsbeschreibung und Schnittstellendefinition. Es können gerne Tabellen und User Stories zur Erläuterung eingesetzt werden
 - ein Markdown-Dokument, das die Funktionalität, Schnittstellendefinitionen und Kovertierungsvorschriften des API-Gateways enthält
3. Dokumentation des Projekts in einer Markdown-Datei
 - Festlegung der Anforderungen an selbst-definiertes Anwendungssystem. Beantworten Sie dazu z.B. folgende Fragen: Was soll das System tun? Welche Aufgabe(n) soll das System lösen bzw. anbieten?
 - Beschreibung und Übersicht des Anwendungssystems
 - Gliederung der Dokumentation in Kapitel
 - Wenn Fremdquellen verwendet wurden sind Quellen anzugeben!
4. Präsentation aller beteiligten des Teams der Anwendung mit Videoabgabe in Panopto
 - Folien als PDF in GitHub hinterlegt
 - Präsentationsdauer inkl. Demonstration maximal 20 Minuten, mind. 15 Minuten
5. Struktur des GitHub Repository und Quellcode der Anwendung
 - Anwendungssoftware muss sauber gegliedert
 - Übersichtlich
 - Kommentiert
 - Anwendung soll eine gute Codequalität haben
 - Formatierung des Codes (z.B. Einrückungen, Code-Konventionen)
 - Code sollte gut verständlich/lesbar sein (Bezeichner für Variablen, Klassen und Methoden)
 - Programmiersprache ist immer Englisch!
 - Entwicklung der Backend-Anwendung und des API Gateway muss mit Spring Boot erfolgen
 - Clientseitige Software ist frei zu realisieren
 - GitHub Repository muss eine gute Struktur in Ordnern und Dateien gegliedert haben

3 Termine und Projektablauf

Projektlaufzeit beträgt exclusive Winter- und Weihnachtsferien ca. 8 Wochen.

Projektbeginn: **06.12.2024**

Projektende: **14.02.2025 23:59 Uhr**

Alle Abgaben haben bis spätestens zu erfolgen: **14.02.2025 23:59 Uhr**

Präsentation:

- Erstellen von Folien (auch in GitHub abzulegen)
- Demonstration der laufenden Anwendung (evtl. nur in Auszügen)
- Abgabe in CampUAS über Panopto
 - denken Sie an die Zeit (Präsentation inkl. Demonstration 15 - 20 Minuten)
 - gleichmäßig auf alle Gruppenmitglieder verteilt
 - klare Gliederung
 - Demonstration erkennbar für den Betrachter (im Video erkennbar, nicht zu klein)
 - Was möchten Sie vermitteln?

4 Bewertungskriterien

Dem Projekt und der abschließenden Bewertung liegen folgende Kriterien zugrunde:

1. Projektkontinuität (kontinuierliche Arbeitsweise) 20%
 - Nur erkennbare kontinuierliche Leistung kann bewertet werden
 - Ein realistischer Arbeitsplan ist unerlässlich
 - Realistische Struktur der Arbeitslast verteilt auf das Team
 - Individuell bewertbar (Commits, Milestones, Issues, aber keine Comments)
 - Regelmäßiges Arbeiten an
 - der Projektplanung
 - der Dokumentation
 - dem Code (Gateway und Services)
2. Projektplanung 25%
 - Kontinuität (siehe Punkt 1), Issues und Milestones (keine Comments)
 - Während der gesamten Projektlaufzeit (Anfang bis Ende)
3. Projektumsetzung 25%
 - Kontinuität (siehe Punkt 1), insbesondere Commits und deren Qualität
 - Wer hat was und wie oft beigetragen (Commits und Kontinuität)
 - Qualität des Codes (z.B. Kommentare/Code-Konventionen/verständlich/lesbar)
 - Aufbau der Anwendung (Architektur?)
 - Komplexität/Umfang der Anwendung
 - Wurden gängige Prinzipien befolgt?
 - Strukturierung in GitHub
 - Es werden keine Branches außer dem main-Branch bewertet!
4. Projektpräsentation 15%
 - Einhaltung der Zeit?
 - Gleichverteilt auf Gruppenmitglieder?
 - Aufbau und Stil
 - Inhalt (Wurden wichtige Aspekte erläutert?)
5. Projektdemonstration 15%
 - Stil
 - Aufbau
 - Integration in Präsentation?

Achtung: Es hilft nicht viele einzelne kleine (vorallem) Commits Issues und Milestones zu erstellen, nur damit die Anzahl steigt. Der Inhalt und die Qualität, insbesondere der Commits, wird bewertet und gewichtet.