

Service-Governance-Plattform

★ Kurzüberblick

- **Problem:** Werkzeuge und Informationen lagen verteilt vor – es fehlte eine zentrale, konsistente Steuerungsoberfläche.
- **Beitrag:** Aufbau einer **rollenbasierten Governance-Konsole**, die **Monitoring, Konfiguration** und **Plattform-Infos** in einer UI bündelt.
- **Wirkung:** Vereinheitlichte Developer-Workflows, schnellere Diagnosepfade und deutlich weniger Abstimmungsaufwand.

🕒 Hintergrund

Mit dem Wachstum der Microservice-Landschaft entstand die Notwendigkeit einer zentralen, produktisierten Governance-Plattform, die alle entwicklungsrelevanten Plattformfunktionen in einer Benutzeroberfläche bündelt und so Transparenz, Standardisierung und Steuerbarkeit der gesamten Architektur verbessert.

Sie sollte insbesondere die verstreuten Werkzeuge und Informationen zusammenführen und dadurch die Entwicklungs- und Diagnoseprozesse deutlich vereinfachen.

🔧 Technische Umsetzung

Die technische Umsetzung folgte einem modularen Ansatz. Die Plattform integrierte bestehende Unternehmenskomponenten wie der **Identity-Provider**, das **API-Gateway**, die **Konfigurationsverwaltung** sowie die Oberflächen der **Messaging-** und **Storage-Services**.

Die wichtigsten Funktionen:

- **Zentrale UI** für alle Services und Instanzen – **rollenbasiert** gefiltert
- **Monitoring** auf Service- und Instanzebene (**OS, JVM, Applikationsmetriken**)
- **Konfigurationsmanagement** (z. B. dynamische Log-Level & Service-Konfigurationsdateien)
- **Service-Authentifizierung** und **-Autorisierung**
- Operationen für integrierte **Cache-, Messaging- und Storage-Komponenten**
- **Einheitliche Oberfläche** zur Unterstützung aller entwicklungsrelevanten Prozesse

Die Plattform wurde als **modulare Webanwendung** mit klar getrennten Verantwortlichkeiten umgesetzt. Frontend und Backend bildeten eigenständige Schichten, während das Backend über eine standardisierte REST-API funktionierte.

Die technische Umsetzung folgte drei übergeordneten Prinzipien:

- **Integration in den bestehenden Microservice-Verbund:**
Die Plattform wurde als regulärer Service betrieben und nutzte sämtliche vorhandenen Plattformmechanismen.
- **Konsistente Nutzung vorhandener Plattformdienste:**
Persistenz, Caching und Messaging wurden vollständig über bestehende Unternehmenskomponenten realisiert, um Redundanzen zu vermeiden und die Entwicklungsgeschwindigkeit zu erhöhen.
- **Kommunikation über etablierte Schnittstellen und Events:**
Die Interaktion erfolgte ausschließlich über REST-APIs und etablierte Event-Mechanismen. Dadurch blieb die Architektur modular, erweiterbar und unabhängig von konkreten Infrastrukturdetails.

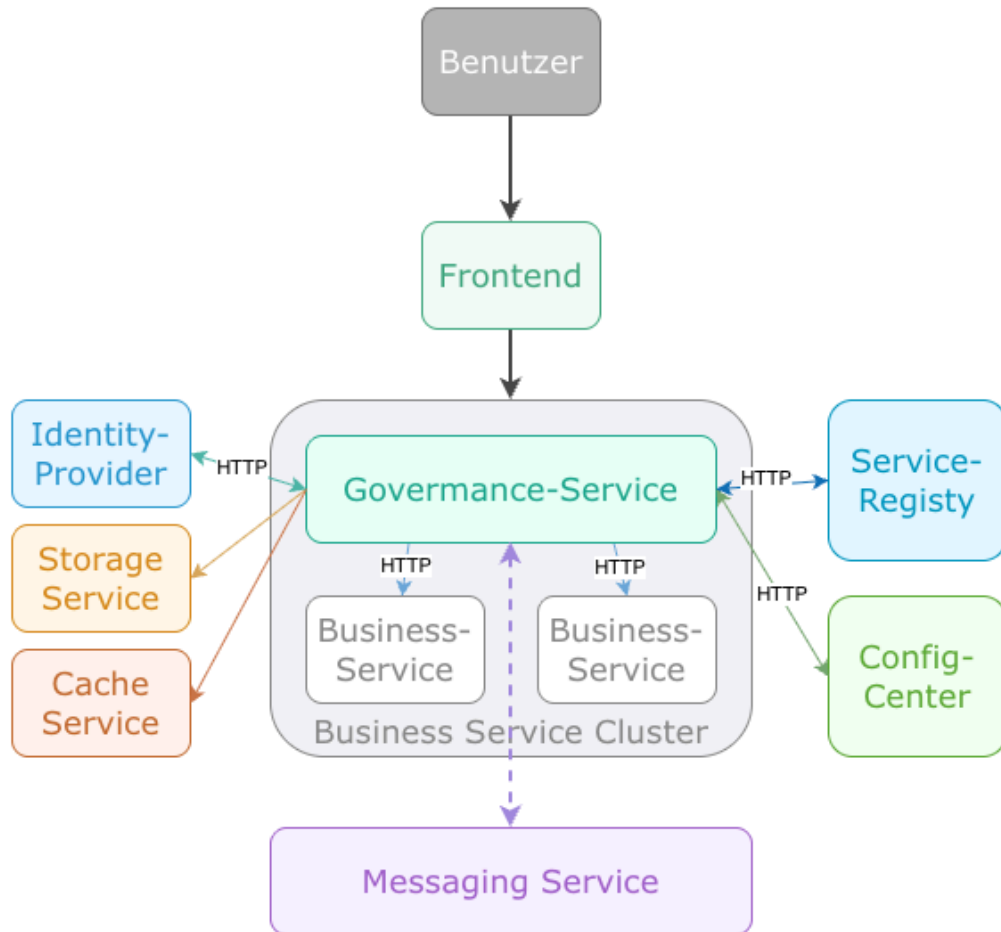


Abbildung 1: Abstrahierte Architektur der Governance-Plattform-Architektur

Ergebnisse

- **Zentrale Sichtbarkeit** über alle Services, Instanzen und Plattformkomponenten
- **Reduzierter Abstimmungs- und Betriebsaufwand** durch konsolidierte Informationen
- **Verbesserte Steuerbarkeit** durch rollenbasierte Kontrollen und standardisierte Konfiguration
- **Höhere Plattformstabilität** durch klare, einheitliche Governance-Mechanismen
- **Schnellere Entscheidungen**, da technische und organisatorische Daten in einer Oberfläche zusammenlaufen