



Architecture Component Descriptions

ACD1: Angular Web Application

Die Angular Web Application wird dem Nutzer über einen nginx Server zur Verfügung gestellt. Die UI wird dabei auf der Client-Side gerendert. Die Daten dafür werden in Form von JSON-Objekten per HTTP API vom Gateway Microservice aufgerufen. Die Funktionen umfassen eine Auflistung der Räume, eine Buchungsansicht mit Bild und Zeitraum für die einzelnen Räume, eine Auflistung der Events, eine Buchungsansicht mit Bild und Zeitraum für die einzelnen Events, eine Galerie mit allen Medien und eine Wetteranzeige.

ACD2: Gateway Microservice

Der Gateway Microservice ist eine Node.js Express Anwendung und bietet HTTP Endpunkte, um Daten aus den anderen Microservices zu erhalten. Ebenso bietet dieser Microservice die Möglichkeit, Endnutzer an einer zentralen Stelle zu authentifizieren.

ACD3: Rooms Microservice

Der Rooms Microservice ist eine Node.js Express Anwendung und bietet einen HTTP Endpunkt, um ein Array an allen Room-Objekten, ein bestimmtes Room-Objekt oder alle vorhandenen Bilder zu erhalten. Die Daten werden dabei aus einer MySQL-Database gelesen. Ebenso wird ein MinIO Speichersystem für die Lagerung von Bildern zu den Room-Objekten verwendet, welches über die Node.js Express Anwendung entsprechend den MySQL-Daten vorausgefüllt wird. Die Bilder werden bei Bedarf direkt von der Angular Web Application ausgelesen. Der zugehörige Link ist in den einzelnen Objekten der MySQL-Daten zu finden.

ACD4: Events Microservice

Der Events Microservice ist eine Node.js Express Anwendung und bietet einen HTTP Endpunkt, um ein Array an allen Event-Objekten, ein bestimmtes Event-Objekt oder alle vorhandenen Bilder zu erhalten. Die Daten werden dabei aus einer MySQL-Database gelesen. Ebenso wird ein MinIO Speichersystem für die Lagerung von Bildern zu den Event-Objekten verwendet, welches über die Node.js Express Anwendung entsprechend den MySQL-Daten vorausgefüllt wird. Die Bilder werden bei Bedarf direkt von der Angular Web Application ausgelesen. Der zugehörige Link ist in den einzelnen Objekten der MySQL-Daten zu finden.

ACD5: Weather Microservice

Der Weather Microservice ist eine Node.js Express Anwendung und bietet einen HTTP Endpunkt, um die aktuellen Wetterdaten zu erhalten. Aktuell wird eine Liste an Weather-Objekten aus einer MySQL-Database gelesen. Zurückgegeben wird ein zufällig ausgewähltes Weather-Objekt aus der Liste.

ACD6: Videos Microservice

Der Videos Microservice ist eine Node.js Express Anwendung und bietet einen HTTP Endpunkt, um ein Array an allen Video-Objekten zu erhalten. Die Daten werden dabei aus einer MySQL-Database gelesen.

ACD7: Sales Microservice

Der Sales Microservice ist eine Node.js Express Anwendung und bietet einen HTTP Endpunkt, um ein Order-Objekt hochzuladen. Die Daten werden dabei in eine MySQL-Database geschrieben.

Architecture Decision Records

ADR1: Microservice-Architektur

Das System wurde gemäß des Microservice-Architekturstils entworfen, da es zahlreiche Funktionen realisiert, die eine klare, eindeutige Modularisierung ermöglichen, wobei die entstehenden Module eine starke interne funktionale Kohäsion aufweisen. Da es sich um ein Web-basiertes System handelt, bietet sich dafür der Microservice Architekturstil an.

ADR2: Client-Side UI Rendering

Da heutzutage nahezu alle Endgeräte über genügend Leistung verfügen, wird die UI auf der Client-Side gerendert, um die benötigten Ressourcen auf dem Server zu reduzieren. Ebenfalls wird die UI stellenweise dynamisch gerendert. Zudem wird der Datenverkehr auf ein Minimum reduziert. Die zugehörigen Dateien einer Website werden von modernen Browsern im Cache gespeichert. Somit besteht der Datenverkehr überwiegend aus JSON-Objekten.

ADR3: Elastic Horizontal Scaling und Load Balancer im Business Logic Tier

Da die Komponenten "Gateway Microservice", "Rooms Microservice", "Events Microservice", "Weather Microservice", "Videos Microservice" und "Sales Microservice" einem unvorhersehbaren Workload ausgesetzt sind, müssen diese elastisch horizontal skaliert werden. Die elastische horizontale Skalierung wird dabei vom Cloud Provider durch einen Managed Hosting Service in Form von PaaS automatisiert ausgeführt.

ADR4: Relationale MySQL Datenbank für diverse Microservices

Die Microservices verwalten Informationen in relationalen MySQL Datenbanken. Die Entscheidung für relationale Datenbanken resultiert aus der eindeutigen Struktur der Daten

und der Notwendigkeit, komplexe Abfragen über die einzelnen Attribute der Datensätze auszuführen.

ADR5: Database per Microservice

Jeder Microservice mit Datenbank-Zugriff verwaltet Informationen in seiner eigenen Datenbank. Die Datenbanken sind gemäß des „Database per Service“ Patterns ausschließlich dem jeweiligen Microservice zugänglich. Daraus folgt, dass alle anderen Komponenten des Systems Informationen nur über die HTTP API des jeweiligen Microservice abfragen können. Diese Entscheidung wurde getroffen, da es im System keinerlei Funktionen in anderen Komponenten gibt, die Datenabfragen zu Datenbanken außerhalb des eigenen Microservices erfordern. Sofern es notwendig ist, werden Daten in der Angular Web Application, über den Gateway Microservice, zusammengeführt (z. B. werden Medien für die Galerie von mehreren Quellen zusammengetragen).

Deployment Decision Records

DDR1: Microservices auf AWS Elastic Beanstalk

Gemäß ADR3 müssen die Komponenten „Gateway Microservice“, „Rooms Microservice“, „Events Microservice“, „Weather Microservice“, „Videos Microservice“ und „Sales Microservice“ elastisch durch einen PaaS horizontal skaliert werden. Daher werden die Komponenten auf AWS Elastic Beanstalk deployed, da es sich dabei um einen Full-Managed Hosting Service auf der Ebene PaaS handelt, der automatisierte elastische Skalierung ermöglicht. Ebenso lassen sich Umgebungsvariablen flexibler einstellen als in anderen Produkten wie dem AWS App Runner.

DDR2: Angular Web Application auf AWS App Runner

Der nginx Container mit der Angular Web Anwendung wird auf dem AWS App Runner gehostet, da dieser Service speziell für Webanwendungen entworfen wurde. Dieser Service sorgt für einen verschlüsselten Lastausgleich, skaliert den Datenverkehr und ermöglicht die Konfiguration des Zugriffs auf Services und die Kommunikation mit anderen AWS-Anwendungen in einem privaten Amazon VPC. Ebenso können Domainnamen per AWS App Runner zugewiesen werden und es werden automatisch benutzerdefinierte Domain-Zertifikate für die Anwendung bereitgestellt und automatisch erneuert, bevor diese ablaufen.