



# 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 von den Microservices aufgerufen. Die Funktionen umfassen eine Auflistung der Autos, eine Detailansicht mit Bild, Videoimpressionen, Konfigurationsmöglichkeiten und Kaufmöglichkeiten für die einzelnen Autos, die Möglichkeit ein Super-Auto auf einer Rennstrecke zu buchen, ein Super-Auto auf einer Karte anzuzeigen und einen Merchandise Shop inklusive Shopping Cart.

## ACD2: Product Information Microservice

Der Product Information Microservice ist eine Node.js Express Anwendung und bietet einen HTTP Endpunkt, um ein Array an allen Merchandise-Objekten zu erhalten. Die Daten werden dabei aus einer MySQL-Database gelesen. Ebenso wird ein MinIO Speichersystem für die Lagerung von Bildern zu den Merchandise-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.

## ACD3: Shopping Cart Microservice

Der Shopping Cart Microservice ist eine Node.js Express Anwendung und bietet einen HTTP Endpunkt, um den letzten Stand des Einkaufswagens zu erhalten und einen anderen HTTP Endpunkt, um den Einkaufswagen zu aktualisieren. Die zugehörigen Daten werden dabei aus einem Redis Cache gelesen bzw. darin geschrieben.

## ACD4: Cars Information Microservice

Der Cars Information Microservice ist eine Node.js Express Anwendung und bietet einen HTTP Endpunkt, um ein Array an allen Auto-Objekten zu erhalten, einen HTTP Endpunkt, um Infos für ein einzelnes Auto zu erhalten und einen HTTP Endpunkt, um eine Liste an Super-Auto-Objekten zu erhalten. Die Daten werden dabei aus einer MySQL-Database gelesen. Ebenso wird ein MinIO Speichersystem für die Lagerung von Bildern zu den Auto-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: Track Information Microservice

Der Track Information Microservice ist eine Node.js Express Anwendung und bietet einen HTTP Endpunkt, um ein Array an allen Rennstrecken-Objekten zu erhalten. Die Daten werden dabei aus einer MySQL-Database gelesen. Ebenso wird ein MinIO Speichersystem für die Lagerung von Bildern zu den Rennstrecken-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.

## ACD6: Localization Microservice

Der Localization Microservice ist eine Node.js Express Anwendung und bietet einen HTTP Endpunkt, um die Koordinaten eines Super-Autos zu erhalten. Aktuell wird eine Liste an Super-Auto-Positionen aus einer MySQL-Database gelesen. Zurückgegeben wird eine zufällig ausgewählte Position aus der Liste.

## ACD7: Sales Microservice

Der Sales Microservice ist eine Node.js Express Anwendung und bietet einen HTTP Endpunkt, um ein Bestellinformationen-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 "Product Information Microservice", "Shopping Cart Microservice", "Car Information Microservice", "Track Information Microservice", "Localization Microservice"

und "Sales Service" einem Unpredictable 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 (Produkt-)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 sowie unterschiedliche Datensätze in Abfragen zu kombinieren.

## 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 zusammengeführt. (Z. B. Auf der Seite zur Buchung einer Testfahrt auf einer Rennstrecke wird die Liste an Super-Autos von dem Cars Information Microservice angefragt und die Liste an Rennstrecken vom Track Information Microservice.

# Deployment Decision Records

## DDR1: Microservices auf AWS Elastic Beanstalk

Gemäß ADR3 müssen die Komponenten "Product Information Microservice", "Shopping Cart Microservice", "Car Information Microservice", "Track Information Microservice", "Localization Microservice" und "Sales Service" 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.

## 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.