# Systemarchitektur

Ein Bild, das Text, Diagramm, Zahl, Schrift enthält.

Automatisch generierte Beschreibung

Clientseite:

Auf dem Client werden folgende Komponenten betrieben:

* Die Weboberfläche, die dem Benutzer die Interaktion mit dem System ermöglicht.
* Die Frontend-Logik, die die Benutzerinteraktion steuert. Dazu gehört die Verarbeitung von Aktionen wie das Zusammenfassen von Gerichten im Warenkorb zu einer Bestellung und deren Übermittlung als JSON an den Server. Ebenso wird die Verwaltung von Benutzereinstellungen und die Anforderung von Bestellhistorien hier abgewickelt.

Die Entscheidung, die Weboberfläche und die Frontend-Logik auf dem Client zu betreiben, bietet den Benutzern eine reaktionsschnelle und benutzerfreundliche Benutzererfahrung.

Serverseite:

Die Serverkomponenten sind in Docker-Containern organisiert, um eine einfache Austauschbarkeit und Bereitstellung zu gewährleisten. Dies ermöglicht die Ausführung der Container auf verschiedenen Arten von Servern. Außerdem erleichtert es dem Entwicklerteam die agile Entwicklung der Anwendung.

Kommunikation zwischen Containern: Die genaue Kommunikationsstrategie zwischen den Containern wird noch definiert. Eine potenzielle Technologie zur Verwaltung des Container-Netzwerks und zur Kontrolle der Container ist Docker Compose. Hiermit können Container orchestriert und verwaltet werden.

Serveranwendungen: Auf dem Server laufen folgende Anwendungen in separaten Containern:

* Datenbank-Container: Enthält die PostgreSQL-Datenbank und ihre Daten.
* Backend-Logik-Container: Hier befindet sich die gesamte Backend-Logik, die für die Verarbeitung von Anfragen vom Client und die Bereitstellung von Daten zuständig ist.
* Webserver-Frontend-Container: Ein Container, der einen Webserver ausführt auf welchem Angular gehostet wird. Dies ist typischerweise ein Nginx, Node.js oder Apache-Server. Der Webbrowser des Clients erstellt beim Zugriff dann eine HTTP-Anfrage an diesen Webserver, um die Ressource (HTML-Datei, CSS-Datei, JavaScript-Datei usw.) anzufordern, die für die angezeigte Webseite benötigt wird. Der Webserver soll dann dem Client die angeforderten Ressourcen bereitstellen. Der Webbrowser des Clients empfängt die HTTP-Antwort des Servers und rendert den Inhalt entsprechend.

REST-API:

Die Kommunikation zwischen Client und Server erfolgt über eine REST-API.

Die REST-API wurde gewählt, da sie eine einfache, plattformunabhängige und skalierbare Möglichkeit bietet, Daten zwischen Client und Server auszutauschen. Durch die Verwendung von standardisierten HTTP-Methoden. Die REST-API ermöglicht es dem Client, mit dem Server zu kommunizieren, Anfragen zu senden und Daten zu empfangen.

Zum Austausch von Daten wie Bestellungen oder Zusammenfassung soll das Datenaustauschformat JSON verwendet werden. Durch die Verwendung von JSON als Datenaustauschformat wird eine effiziente Übertragung von Informationen ermöglicht, was zu einer verbesserten Leistung und Interoperabilität führt.