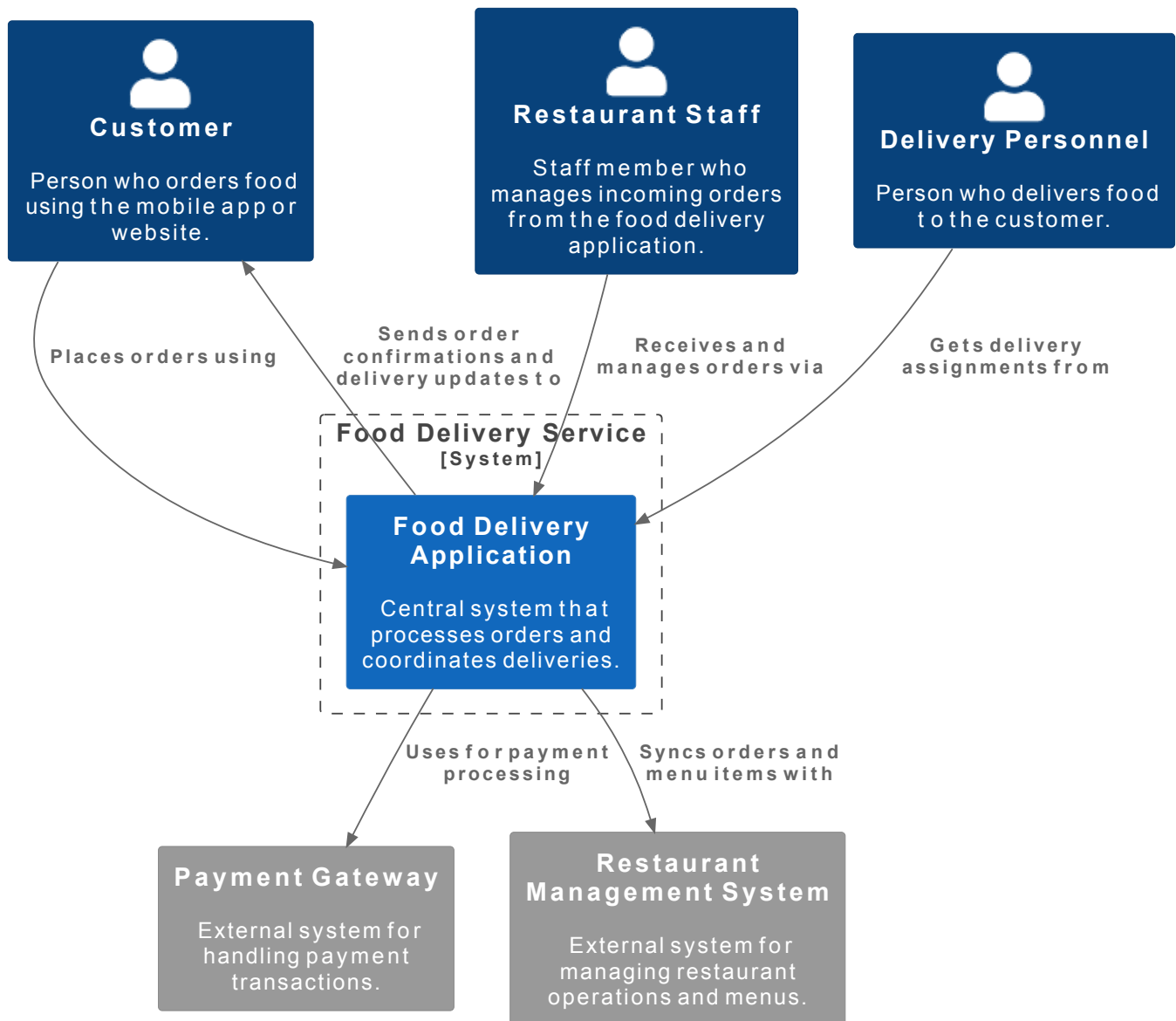


SSE

- Overview
 - 2 - Food Delivery Service
 - 3 - API Application
 - 4 - Order Code

Overview

System Context diagram for Food Delivery Service



Legend

| |
|-----------------|
| person |
| system |
| external person |
| external system |

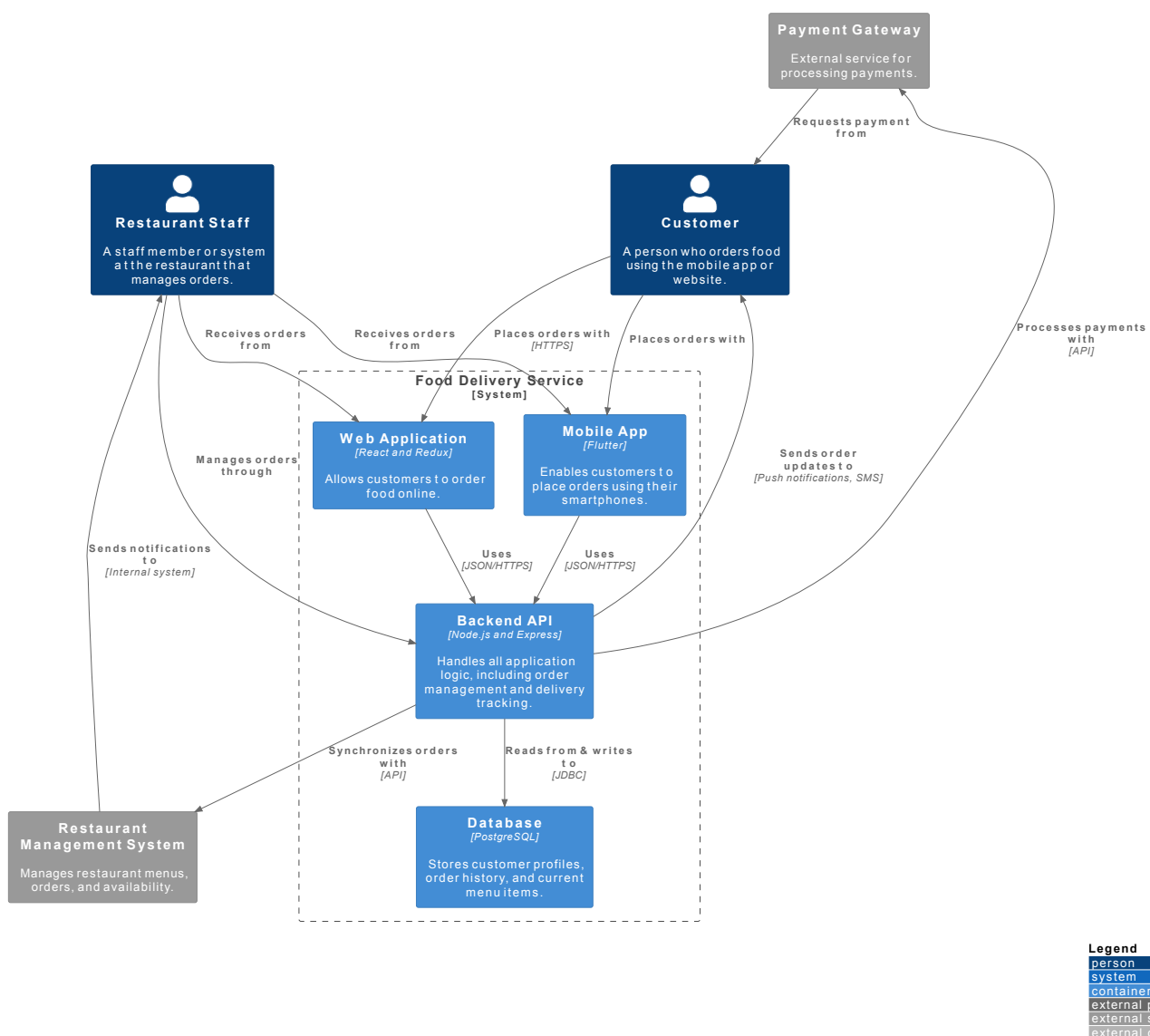
Das system context diagram zeigt einen Gesamtüberblick über eine einzige Softwarestruktur. In der Mitte steht unser DeliverySystem, umgeben von allen Akteuren die es beeinflussen.

Technische Details sind hierbei nicht abgebildet sondern nur das "Big picture", welches auch für nicht techaffine Menschen verständlich sein soll.

Haupt Element: Unser Food-Delivery-Service. **Unterstützende Elemente:** Personen (z.B.: customer, Restaurantmitarbeiter, etc.) und Softwaresysteme die direkt verbunden sind mit dem Hauptelement. **Intended audience:** Für alle die sich mit unserem System beschäftigen wollen, unabhängig davon wie sehr sie sich mit dem Team oder Technischen details auskennen.

2 - Food Delivery Service

/2 - Food Delivery Service



Level 2: Container diagram

Ist ein "zoomed-in" Blick in die Systemgrenzen. Das Container diagram verschafft einen Überblick auf die software Architektur, ohne dabei genau ins Detail zu gehen. Jedoch weist es auf die wesentlichen Technology entscheidungen hin. Es wird auch verdeutlicht wie die Einzelnen Komponenten miteinander verknüpft sind.

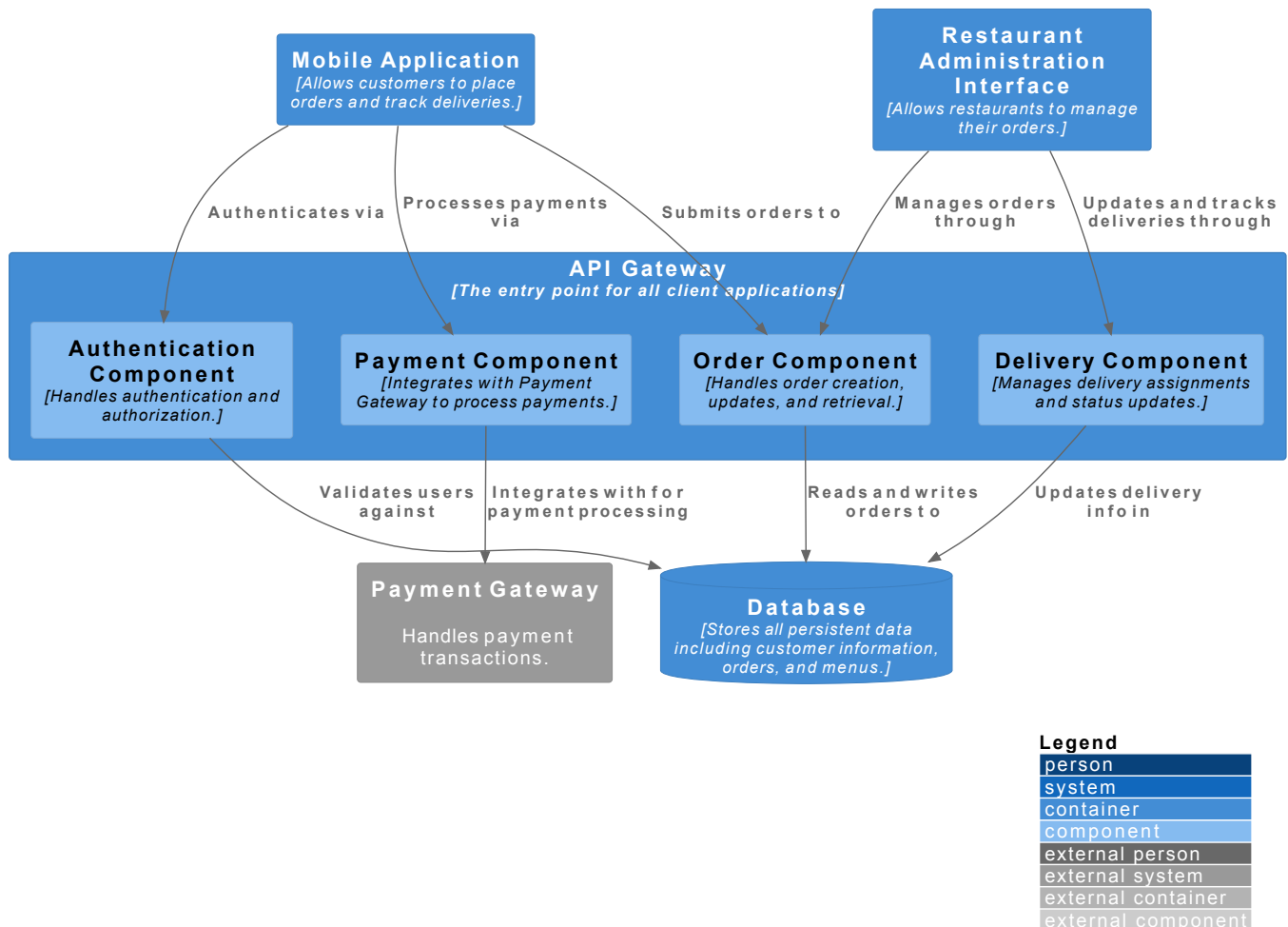
Ein "Container" ist hier eine separat laufende Einheit die Aufgaben, wie code-execution oder das Speichern von Daten, übernehmen kann. Beispiele für Container: Web application, mobile app, database, etc.

Haupt Elemente: Container within the software system. **Unterstützende Elemente:** Personen und Systeme die direkt mit den Containern verbunden sind. **Intended audience:** Technikaffine Personen im und um das Softwaredevelopment Team (z.B.: software architects, developer, tester, etc.)

3 - API Application

/2 - Food Delivery Service/3 - API Application

Component diagram for the API Gateway - Food Delivery Service

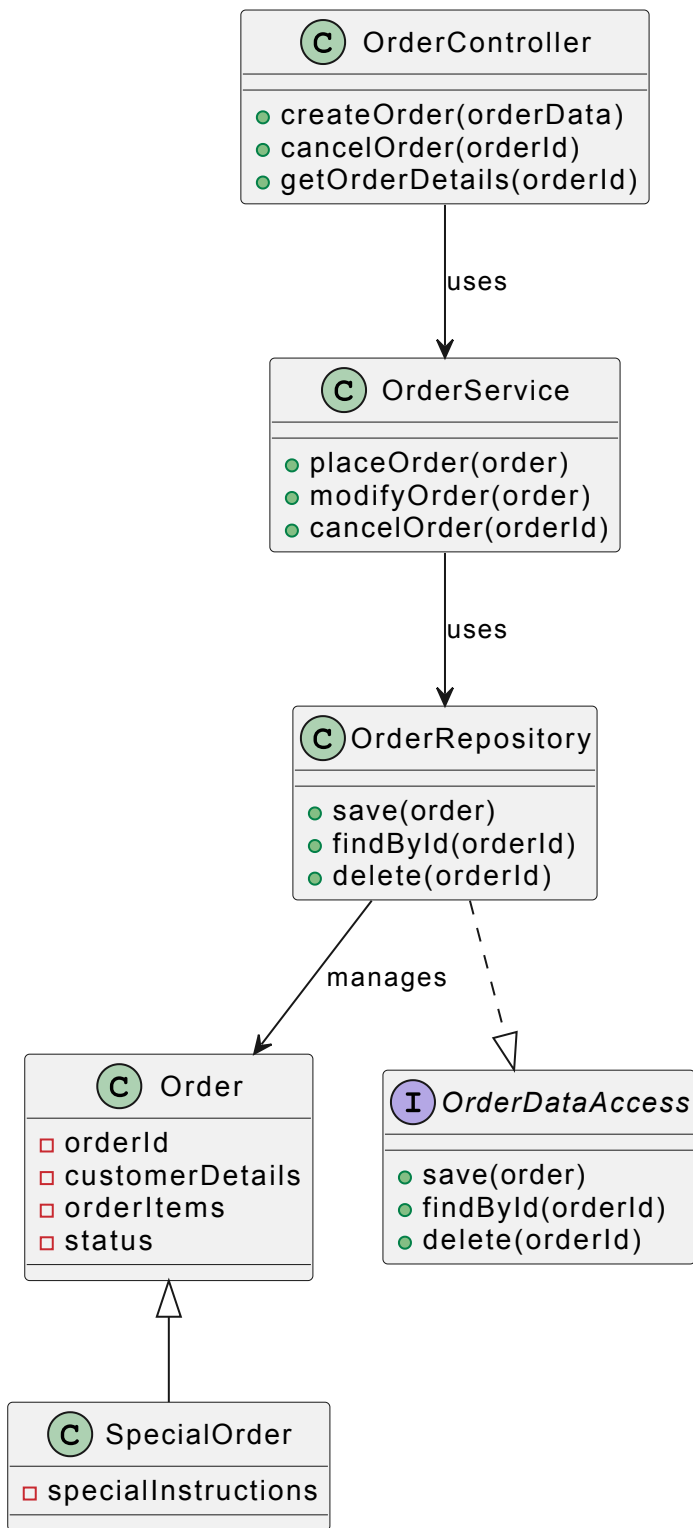


Level 3: Component diagram

Man zoomed in jedem Container und Dekonstruiert aus was er gemacht ist, um seine wesentlichen Bestandteile zu analysieren. Die besagten Bestandteile nennt man "Components". Sie werden zuerst identifiziert und dann beschreibt man ihre Aufgaben und Implementationsdetails.

Haupt Element: Components in einem einzigen Container. **Unterstützende Elemente:** Personen und andere Container oder Systeme die direkt mit dem Container verbunden sind. **Intended audience:** Softwarearchitekten und developer.

4 - Order Code



Level 4: Dynamic diagram

Ist nützlich um zu zeigen wie Elemente eines Statischen Models, dynamisch erzeugt und verändert werden können sobald das system läuft. Es wird meistens mit UML dargestellt, da es sich sehr gut dafür eignet.

Haupt Element: Je nach diagram scope können hier ein oder mehrere systeme Abbgebildet werden.