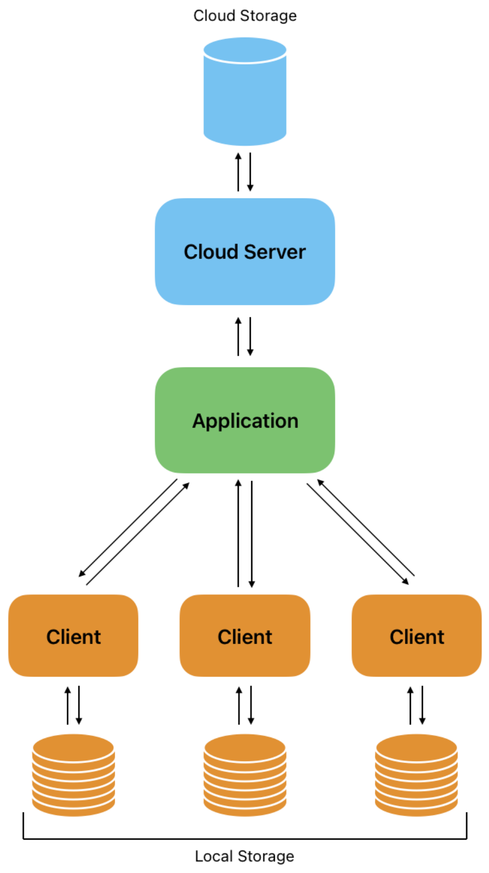
U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Shopping Lists Application

Large Scale Distributed Systems - M.EIC 2023/24

1. Domain Description and High-Level Design

The project aims to develop a local-first shopping list application, that offers a seamless user experience with the ability to create and manage shopping lists, while ensuring data persistence and high availability. The application operates on a two-tier architecture: the user's device and a cloud component. Users can create shopping lists with a unique identifier (e.g., URL) and share these lists with others. The application allows multiple users to collaboratively modify lists concurrently, and it provides features like adding/removing items and setting target quantities.



The design architecture for the application is a critical aspect of this project. It needs to provide a robust and scalable system while ensuring data integrity, availability, and collaborative functionality. To achieve these goals, the architecture will follow a two-tier approach:

1. **Local Device Component**

This component is responsible for providing an intuitive user interface for creating, managing, and collaborating on shopping lists. It will employ a local database for data storage and will support offline access to shopping lists. Users will be able to create and make modifications to the lists locally, and when they intend, they can add their modifications to the cloud.

2. **Cloud Component:**

The cloud component will serve as the central hub for sharing and backing up shopping lists. It will ensure data consistency and availability, especially in cases where multiple users are making concurrent changes to the same list. Data replication will be a fundamental component of the cloud architecture to guarantee high availability and fault tolerance. It will handle data synchronization and conflict resolution. For achieving data consistency, the architecture will implement Conflict-free Replicated Data Types (CRDTs) for improved data synchronization in a distributed environment.

For the purpose of this university project, we will be using JSON files to store the replicas of the shopping lists, instead of local databases and a cloud. More on this in section 3.

2. Chosen Technologies

In this project, Node.js will serve as the backend runtime, providing a robust and efficient server environment. It will handle data management, synchronization, and communication with the cloud component. On the frontend, HTML, CSS, and JavaScript will be used to create a responsive and interactive user interface. HTML will structure the content, CSS will style the interface for a visually appealing look, and JavaScript will enable dynamic and real-time interactions with shopping lists. These technologies combined will facilitate a seamless and user-friendly shopping list application that operates on both web and mobile platforms, catering to a diverse user base. For data storage and replication, we will use JSON files and we explain this choice in the next section.

3. CRDT's and Storage

To store our shopping lists, we are going to implement CRDT's. With these data types we can automatically merge the versions of each client in a single one, resolving conflicts, ensuring data consistency and reliability across the system. For the shopping items, we are going to implement an Add-wins set (AWSet), that favors additions over removals. We think it is the best choice, because in the context of a shopping list it is best to have more items than missing ones.

Additionally, we are going to store two quantities per item, one for the desired quantity and one for the acquired quantity. For these parameters we are going to implement PN-Counters, with restrictions so that our quantities can not be less than zero. In principle, we will implement State-Based CRDT's, as they require less communication effort, facilitating message checks.

Using JSON files for data storage is a straightforward approach. In our collaborative shopping list system, the JSON files are going to represent replicas of the shopping lists, for both clients and the cloud. This simplicity can be advantageous, especially for a university project, as it reduces the complexity of setting up and managing databases and cloud services. Another advantage is that JSON doesn't have a binary format and because of that we can determine the value of a CRDT.

Group Elements:

- Diogo Alexandre da Costa Melo Moreira da Fonte - up202004175@edu.fe.up.pt
- Domingos José Silva Moreira dos Santos - up201906680@edu.fe.up.pt
- Duarte Filipe Campos Barbosa Lopes - up202006408@edu.fe.up.pt
- José Pedro Teixeira Ramos - 202005460@edu.fe.up.pt