

Messaging-Applikation WS-ArchitekturSpezifikation.v1

INF202-Projekt

Eren Naci Odabaşı

180501038

TMS-4

e180501038@stud.tau.edu.tr

Nihat Akın

180501008

TMS-4

e180501008@stud.tau.edu.tr



Inhaltsverzeichnis

| | |
|---|----------|
| Inhaltsverzeichnis..... | 2 |
| Architekturüberblick..... | 3 |
| Beschreibung der „Controller“ Klassen..... | 4 |
| Rückverfolgbarkeit der Anforderungen..... | 5 |
| Beschreibung der DB-Zugriffsschicht (Daten-Modelle)..... | 5 |

Architekturüberblick

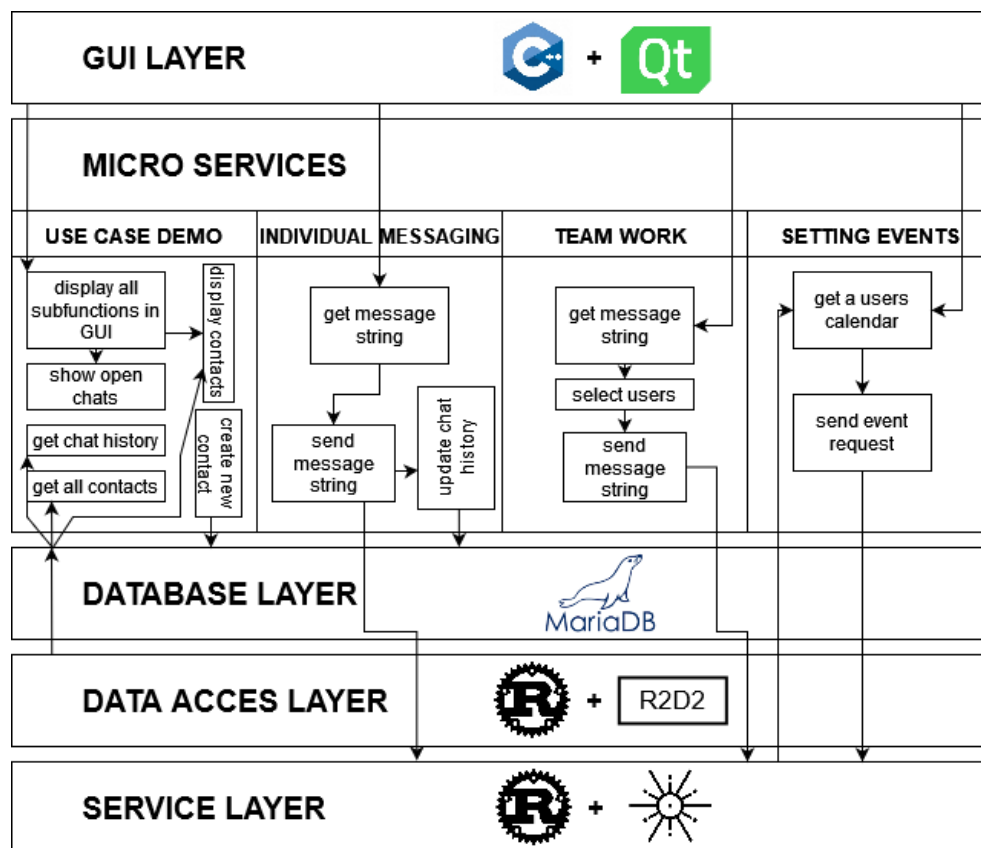


Abb.1: Architekturüberblick

GUI Layer (GUI Schicht): Es ist die Schicht, auf der Benutzer mit der Anwendung interagieren. Diese Schicht wurde mit der C++ Qt-Bibliothek entwickelt. Nachrichtenanfragen von Benutzern kommen hier über Formulare in der Benutzeroberfläche.

Micro Services (Mikrodienste): Es ist die Schicht, die vom Controller, Dienst und Repository von Use-Cases verwendet wird. Die Anwendungslogik für jeden Use-Case wird durch Verknüpfungen mit anderen Schichten demonstriert.

Database Layer (Datenbankschicht): Diese Schicht verwaltet die Datenbank, in der Nachrichten und Benutzerinformationen gespeichert werden. Datenbankverbindungen werden mit der Rust-Bibliothek r2d2 konfiguriert.

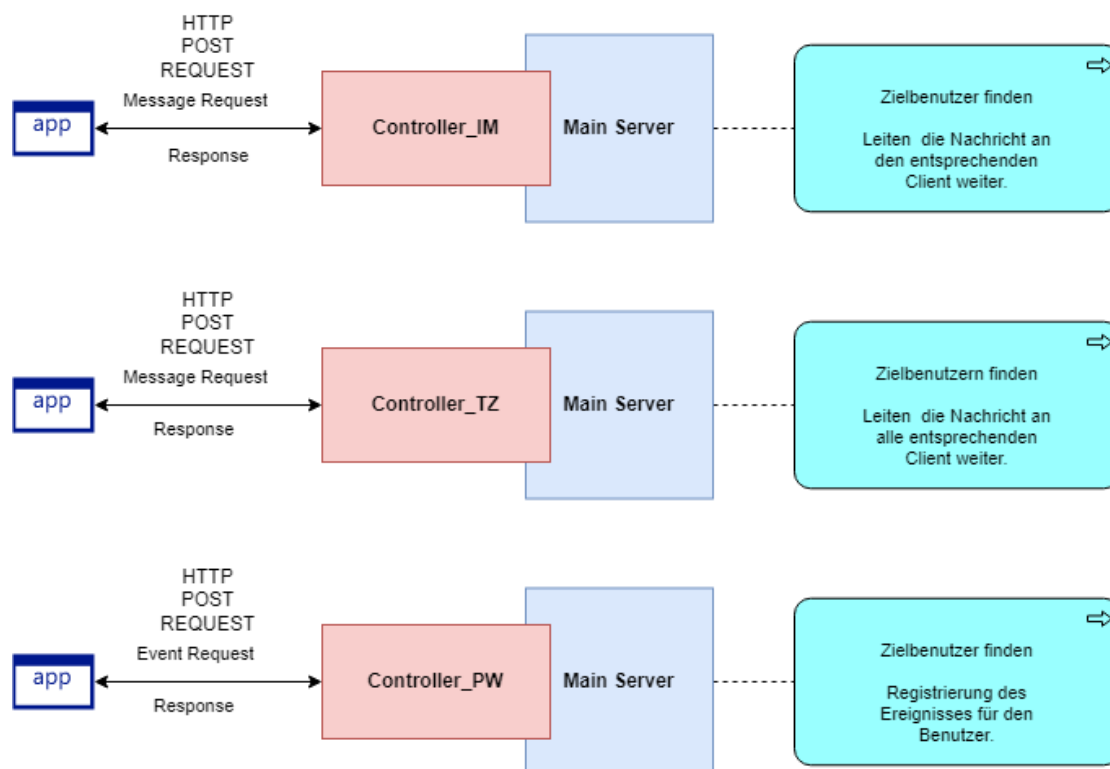
Data Acces Layer (Datenzugriffsschicht): Diese Schicht wird verwendet, um auf die Datenbank zuzugreifen. Operationen wie das Abrufen und Speichern von Daten aus der Datenbank werden in dieser Schicht ausgeführt.

Service Layer (Dienstschicht): Diese Schicht ist die Schicht, in der die Geschäftslogik ausgeführt wird, sie empfängt die Daten von der Datenbankschicht und verarbeitet sie. Darüber hinaus enthält es alle Infrastrukturkomponenten, die für die Entwicklung und den Betrieb der Anwendung erforderlich sind.

Beschreibung der „Controller“ Klassen

Es gibt 3 Hauptcontroller, die Anfragen von Kunden bearbeiten:

1. **Controller_IM**: Dieser Controller ist für die Bearbeitung von Nachrichtenanforderungen zwischen zwei Benutzern zuständig. Der Client des Anfragesenders sendet die Nachrichtendaten (eine Zeichenkette, die entweder in JSON oder im Rohformat gespeichert ist) zusammen mit der eindeutigen Ziel-Benutzer-ID (in unserem Fall eine E-Mail-Adresse). Der Controller prüft dann die Gültigkeit der Ziel-ID. Ist die Prüfung erfolgreich, wird die Nachricht über eine weitere HTTP-POST-Anfrage an den anwendungsseitigen Client des Ziels weitergeleitet.
2. **Controller_TZ**: Wenn eine Nachricht an einen Team-Chat gesendet wird, stellt der Server auf der Anwendungsseite eine HTTP-POST-Anfrage an den Hauptserver; diese Anfrage enthält die Ziel-IDs (mehrere) und die Nachrichtendaten. Der Hauptserver unterscheidet die Arten von Team-Nachrichten nach der Anzahl der Ziele, die die Anfrage hat, und aktiviert seinen Controller. Der Controller_TZ leitet diese Nachricht dann an alle Ziele weiter, wobei er die Gültigkeit der Ziel-IDs überprüft, wiederum über POST-Requests.
3. **Controller_PW**: Ereignisse werden verwendet, um gemeinsame Ziele wie Projekttermine oder Besprechungen zu verfolgen. Die Registrierung von Ereignissen erfolgt, wenn ein Benutzer einen anderen Benutzer (der sich in seiner Kontaktliste befindet) zu einem Ereignis hinzufügt. Dies geschieht ebenfalls durch eine POST-Anfrage. Die Anfrage kann in einem JSON-Format erfolgen.



Rückverfolgbarkeit der Anforderungen

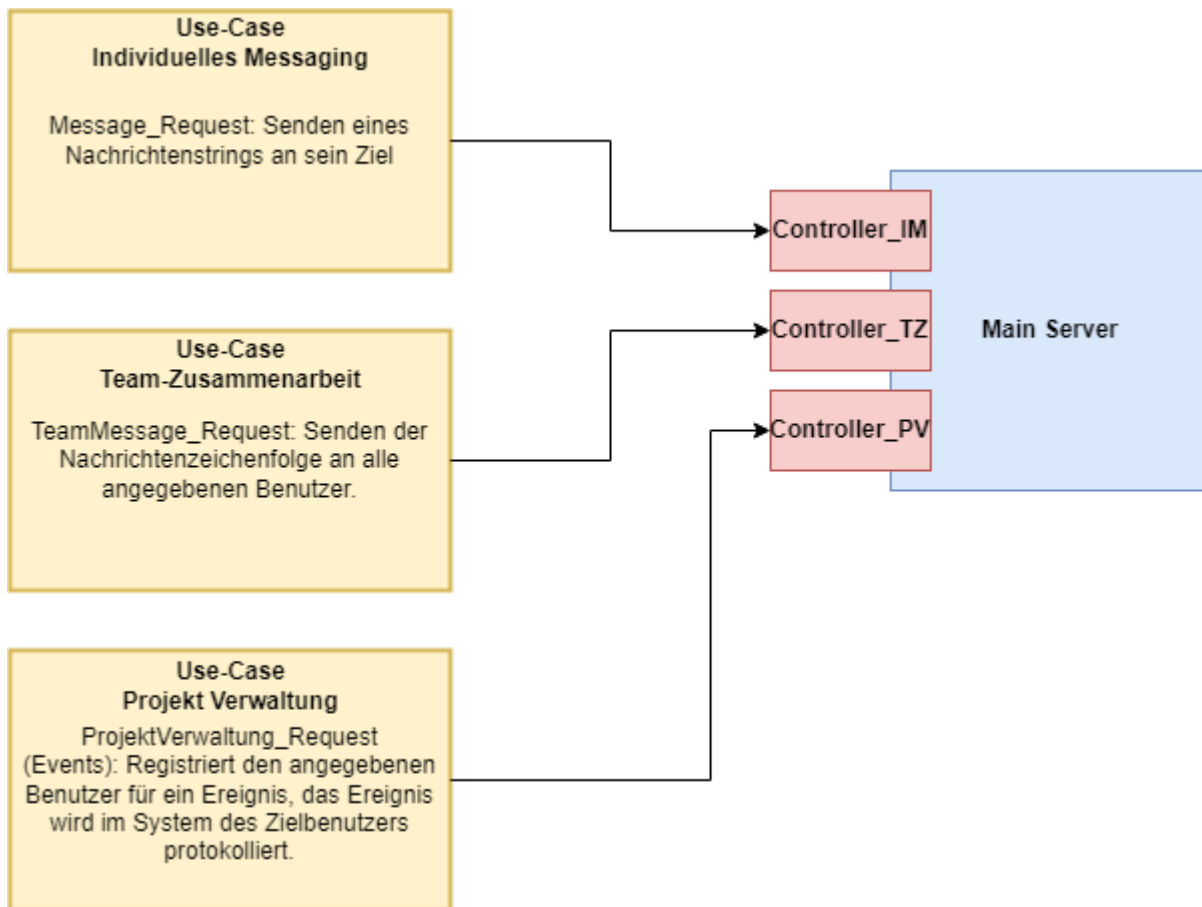


Abb. 3: Rückverfolgbarkeit der Anforderungen

Beschreibung der DB-Zugriffsschicht (Daten-Modelle)

In diesem Projekt gibt es 2 Datenbanken, eine für den Hauptserver und eine für die Desktop-Anwendung.

Die Datenbank der Desktop-Anwendung verwaltet die Startkonfiguration des Benutzers, wie die Kontaktliste, die Team-Chat-Informationen und die Ereignisprotokolle. Für diese Zwecke haben wir uns für die Verwendung einer lokalen MariaDB-Datenbank entschieden, die über einen MySQLDb-Treiber mit der Qt-Anwendung verbunden ist. MariaDB ist eine SQL-Datenbank, so dass die Daten über SQL-Abfragen von der Backend-Seite des Desktops geholt werden.

Die Hauptdatenbank auf der Serverseite kann eine nicht-lokale (wie Firebase) oder eine lokale MongoDB-Datenbank sein. Diese wird vom Hauptserver verwendet, um Informationen über seine Kunden zu speichern, wie z. B. eindeutige Bezeichner und Kommunikationsadressen.

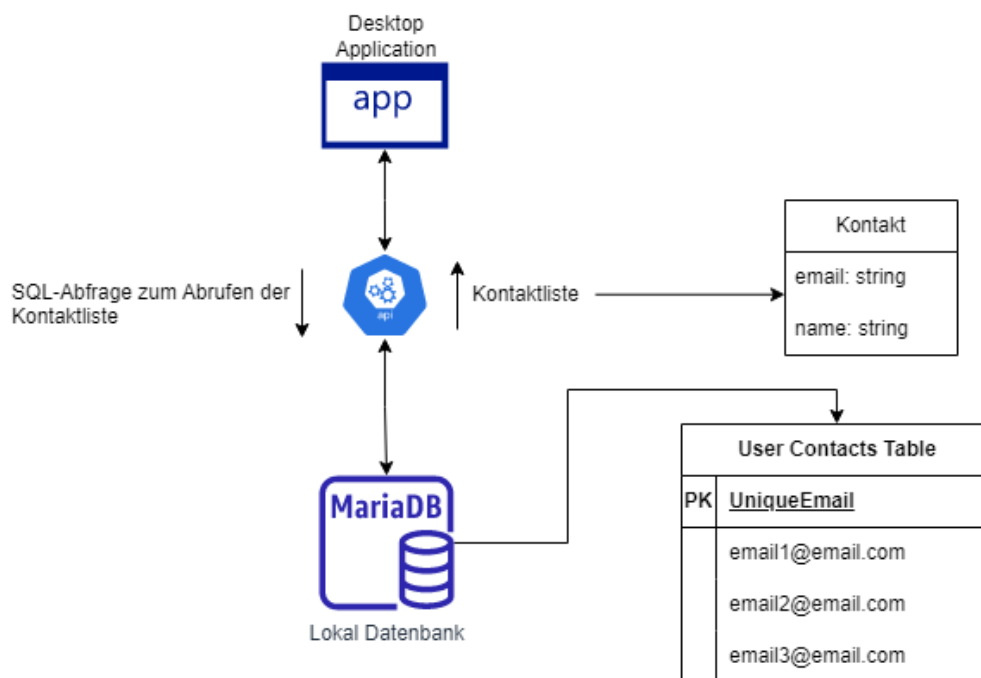


Abb. 4: Zugriffsschicht-Kontakten

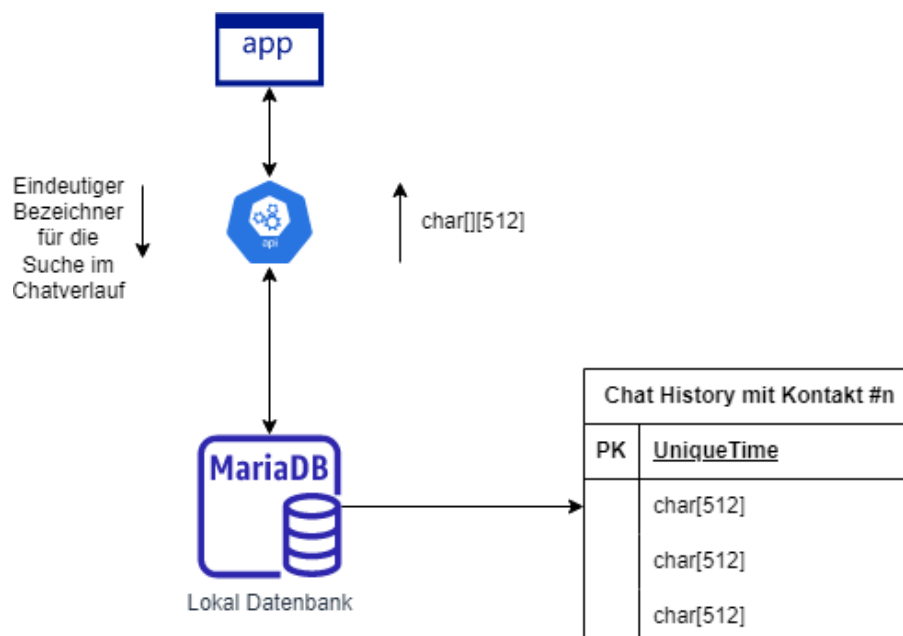


Abb. 5: Zugriffsschicht-Chat Histories