

THEIA Baustellen App

Eine Lösungsarchitektur für eine mobile Applikation

Masterthesis

Studiengang:	MAS-IT Software Architecture
Autor:	Zürcher Amos Software Architect
Betreuer:	Staub Bill Business Unit Lead "Business Solutions"
Auftraggeberin:	Helion AG
Experte:	Kleiner Helmut Max, Schmidhauser Arno
Datum:	09.03.2025

Management Summary

Ausgangslage

Die Helion AG hat sich auf die Installation von Solar und Energiespeichertechnik spezialisiert. Für das Projektmanagement wird eine Anwendung namens THEIA eingesetzt. THEIA besteht aus einem Desktop Client und einem Backend. Mitarbeiter und Mitarbeiterinnen¹ der Helion AG müssen während Installationsarbeiten auf Projektdaten und Funktionen in THEIA zugreifen können. In der bisherigen Lösung von THEIA setzt dies eine aktive Internetverbindung voraus. Oft haben Mitarbeiter auf der Baustelle keine konstant aktive Internetverbindung. Eine mobile App für Android und iOS Geräte könnte die Projektdaten und Funktionen auch offline verfügbar machen. In so einem Szenario ist der Datentransfer und das Zusammenführen von Daten eine besondere Herausforderung. Aus den Anforderungen der Helion AG wurde ein Pflichtenheft ausgearbeitet, welches als Grundlage für diese Arbeit dient.

Ziel

Das Ziel dieser Arbeit besteht darin, ein geeignetes Entwurfsmuster für die Datensynchronisation zu finden, welches den Integritäts- und Verfügbarkeitsanforderungen der Helion AG entspricht. In einem «Software Architektur Dokument» wird eine Lösungsarchitektur aufgezeigt, welche die Anforderungen der Helion AG erfüllt.

Geplant war eine Umsetzung der Lösung. Auf Grund des zu grossen Umfangs musste dieses Ziel angepasst werden. Umgesetzt wird ein Proof of Concept (PoC), welcher die Funktion der konzipierten Datensynchronisation und das Sicherheitskonzept mit Authentifizierung und Autorisierung aufzeigt. Dies wird erreicht mit einer minimalen Implementation der fachlichen Anforderungen bezüglich Projektdaten.

Vorgehen

Zum Thema «Datensynchronisation in mobilen Applikationen» wurden drei Entwurfsmuster näher betrachtet. In einer Nutzwertanalyse wurde gezeigt, dass das Entwurfsmuster «Timestamp Transfer» die Anforderungen der Helion AG am besten erfüllt.

Basierend auf den Anforderungen der Helion AG und des Ergebnis der Nutzwertanalyse wurde eine Lösungsarchitektur ausgearbeitet.

In einer Implementierung im Umfang eines Proof of Concepts wurde die Funktion der Datensynchronisation und das Sicherheitskonzept verifiziert.

Ergebnisse

In dieser Arbeit wurde eine mögliche Lösungsarchitektur aufgezeigt, um die Anforderungen an die «Baustellen App» gemäss dem Pflichtenheft zu realisieren. In einer Nutzwertanalyse wurde das Entwurfsmuster «Timestamp Transfer» für die Datensynchronisation als geeignet identifiziert. In einem Proof of Concept wurde Datensynchronisation, Authentifizierung und Rollenkonzept umgesetzt. Mit dem PoC konnte gezeigt werden, dass die konzipierte Lösung funktioniert.

¹ Im weiteren Dokument wird die männliche Form stellvertretend für männlich, weiblich und sächlich verwendet.

Inhaltsverzeichnis

Eine Lösungsarchitektur für eine mobile Applikation	1
Management Summary	2
1 Einleitung	5
1.1 Ausgangslage und Motivation	5
1.2 Problemstellung	5
1.3 Ziel der Arbeit	6
1.4 Abgrenzung	6
1.5 Aufbau der Arbeit	6
2 Literaturreview	7
2.1 Grundlagen	7
2.1.1 Set Reconciliation Problem	7
2.1.2 CAP Theorem	7
2.2 Entwurfsmuster für die Datensynchronisation in mobilen Anwendungen	8
2.2.1 Datensynchronisation	9
2.2.2 Datenspeicher und Verfügbarkeit	10
2.2.3 Datenübertragung	11
3 Methodik	14
3.1 Projektmanagement	14
3.2 Masterarbeit Zeitplan	15
3.3 Entwurf der Lösungsarchitektur	16
3.4 Analyse der Datensynchronisation	16
3.5 Umsetzung der «Baustellen App» als PoC	16
4 Ergebnisse	17
4.1 Pflichtenheft	17
4.1.1 Authentifizierung	17
4.1.2 Projektübersicht	17
4.1.3 Projektdetails	18
4.1.4 Notizen zum Projekt	19
4.1.5 Inbetriebnahme	19
4.1.6 Mockups zur Benutzeroberfläche der mobilen Applikation	20
4.1.7 Nicht Funktionale Anforderungen	22
4.2 Anwendungsfälle «Baustellen App»	23
4.3 Lösungsarchitektur	24
4.3.1 Einführung und Ziele	24
4.3.2 Qualitätsziele	25
4.3.3 Stakeholder	25
4.3.4 Randbedingungen	26
4.3.5 Kontextabgrenzung	26
4.3.6 Lösungsstrategie	27
4.3.7 Bausteinsicht	28
4.3.8 Laufzeitsicht	39
4.3.9 Verteilungssicht	44
4.3.10 Querschnittliche Konzepte	46
4.3.11 Architekturentscheidungen	50
4.3.12 Qualitätsanforderungen	59
4.3.13 Risiken	60
4.4 Umsetzung der «Baustellen App» als PoC	61
4.4.1 Verwendete Technologien	61
4.4.2 Applikationsmodel Backend	61
4.4.3 Generelles Applikationsmodel der mobilen Applikation	64
4.4.4 Ansichten für die mobile Applikation	67

4.4.5 Umsetzung der Datensynchronisation	71
5 Diskussion/ Zusammenfassung	71
5.1 Eignung von «Timestamp Transfer» für die Synchronisation der Projektdaten	71
5.2 Eignung der vorgeschlagenen Lösungsarchitektur	72
5.3 Verbesserungspotenzial	72
5.3.1 Einatz von SignalR oder gRPC	72
5.3.2 Weitere Identitätsprovider und externe Registrierung zulassen	72
5.4 Zukünftige Weiterentwicklung	73
5.5 Persönliche Betrachtung und Danksagung	73
6 Abbildungsverzeichnis	74
7 Tabellenverzeichnis	75
8 Glossar	76
Literaturverzeichnis	78
9 Anhang	79
9.1 Sourcecode	79
9.2 «Baustellen App» Installationsanleitung für Android	79
10 Verwendung von KI	80
11 Selbständigkeitserklärung	80

1 Einleitung

1.1 Ausgangslage und Motivation

Für die Versorgung ihrer Mitarbeiter mit projektrelevanten Informationen, hat die Helion AG die isolutions AG angefragt, eine Offerte für eine Mobile Applikation namens «THEIA Baustellen App» zu erstellen. THEIA ist eine Applikation zur Projektverwaltung, bestehend aus einem Desktop Client und einem Backend. THEIA befindet sich bereits im Einsatz innerhalb der Helion AG. Die mobile Lösung «THEIA Baustellen App», später nur noch «Baustellen App» genannt, soll Daten über die bestehende API beziehen und auf mobilen Geräten wie beispielsweise Smartphones verfügbar machen. Eine erwähnenswerte Anforderung an die Applikation ist die Offline Benutzbarkeit. Die Umsetzung soll in mehreren Phasen durchgeführt werden. Der Umfang der ersten Phase wurde als «Minimum Viable Product» (MVP) gemäss [1] definiert und wird in dieser Masterarbeit betrachtet.

Die Helion AG ist eine Expertin für Energiesolutions in der Schweiz und hat sich zum Ziel gesetzt, die Energiewelt voranzutreiben. Helion AG bietet Lösungen im Bereich Photovoltaik, Stromspeicher, Smart Energy, Wärmepumpen und Ladestationen für Elektrofahrzeuge. Die Dienstleistungen umfassen die Beratung, Verkauf, Montage und Wartung und wird von einem Team von über 500 Mitarbeitern ausgeführt².

Heute ist es besonders für die Mitarbeiter auf der Baustelle schwierig, auf die projektrelevanten Informationen zu zugreifen. Um Zugriff zu ermöglichen, muss ein Rechner mit installiertem THEIA Client vorhanden sein. Außerdem gibt es im Projektablauf Prozesse, welche eine Aktion der Mitarbeiter erfordert: Zum Beispiel wird für den Abschluss des Projekts ein Inbetriebnahmeprotokoll mit einer vorgegebenen Checkliste verlangt. Weitere Prozesse sind Materialbestellungen, Informationsaustausch mit dem Projektmanagement oder Backoffice. Heute werden diese Prozesse grösstenteils auf Papier durchgeführt.

Da die Anzahl der Mitarbeiter und das Auftragsvolumen in den letzten Jahren gewachsen ist, stellt sich der Helion AG auch die Herausforderung, ihre Prozesse in der Projektumsetzung zu überarbeiten. Deshalb hat sich die Helion AG dazu entschieden, eine mobile Applikation für Ihre Mitarbeiter umzusetzen.

Im Verlauf der Antragstellung dieser Masterarbeit hat sich die Helion AG dazu entschieden, den Auftrag nicht an isolutions AG zu vergeben. In Absprache mit den Experten und der Studienleitenden der Masterarbeit wurde entschieden, die Masterarbeit trotzdem auf Basis der Anforderungen der Helion AG durchzuführen.

1.2 Problemstellung

Die bestehende Lösung THEIA besteht aus einem Desktop Client und einem Backendsystem. Der Kontext, in dem dieses Client/Server System arbeitet, zeichnet sich dadurch aus, dass eine permanente Netzwerkverbindung vorausgesetzt wird. Daten werden lokal zwischengespeichert aber nicht persistiert, da von einem permanent verfügbaren Backend ausgegangen werden kann.

Mit der Implementierung einer mobilen Applikation werden diese zwei Vorteile der Desktop Lösung negiert. In einer mobilen Applikation kann nicht von einer durchgehenden Netzwerkkonnektivität ausgegangen werden und der Speicher auf dem Gerät ist limitiert und meistens nicht erweiterbar. Tatsächlich ist eine spezifische Anforderung an die mobile Applikation, die offline Fähigkeit, da gemäss Helion AG die konstante Internetverbindung auf der Baustelle oft nicht gewährleistet ist.

Aufgrund dieser Ausgangslage ist klar, dass auf die Verfügbarkeit der Projektdaten, Funktionalität der App ohne Internet, und Konsistenz der Daten ein spezielles Augenmerk gelegt werden muss. Es gibt unterschiedliche Ansätze, Entwurfsmuster und Methoden für die Synchronisierung von Daten,

² <https://www.helion.ch/de/ueber-uns/>

Sicherstellung von Verfügbarkeit und Konsistenz. Es stellt sich die Frage, welches Entwurfsmuster am besten geeignet ist, um die Anforderungen der Helion AG an diese Applikation zu erfüllen.

1.3 Ziel der Arbeit

Das Ziel dieser Arbeit ist es in einem Softwarearchitekturdokument (SAD) eine mögliche Lösungsarchitektur aufzuzeigen. Folgende Lieferobjekte werden erarbeitet, um die Fragen der Problemstellung zu beantworten.

- Lösungsarchitektur für die «Baustellen App» nach arc42 [2].
- Architekturentscheid für die Umsetzung der Datensynchronisation.
- ~~Minimum Viable Product (MVP) der «Baustellen App» umsetzen.~~

In Absprache mit dem Expertengremium wurde am 27.01.2025 entschieden, die Zielsetzung „ein MVP der Baustellen App umzusetzen“ anzupassen. Neu lautet das Ziel:

- Ein PoC der «Baustellen App» umsetzen.

Der PoC muss folgende Anforderungen erfüllen:

- Nachweis der Funktionalität des Entwurfsmusters zur Datensynchronisation.
- Demonstration des Benutzerzugriffs auf Projektdaten.

1.4 Abgrenzung

Die Lösungsarchitektur konzentriert sich auf die Datensynchronisation und den offline Modus. Auf Grund der vorhandenen Infrastruktur beschränkt sich der PoC auf die Plattform Android.

Folgende Punkte sind nicht Teil der Masterarbeit:

- Eine vollständige Implementierung der funktionalen Anforderungen ist nicht Teil dieser Arbeit.
- Eine Integration in die IT-Landschaft der Helion AG ist nicht Teil dieser Masterthesis.

Wie bereits in der Ausgangslage erwähnt, hat sich die Helion AG aus dem Projekt zurückgezogen. Das führt zu einigen Einschränkungen:

- Es steht kein Backend zur Verfügung. Deshalb muss für den MVP ein eigenes Backend erstellt werden.
- Auf Grund der zeitlichen Einschränkung ist Qualitätsicherung kein Bestandteil dieser Arbeit.
Dies beinhaltet folgende Punkte:
 - Komponenten Tests
 - Automatisierte UI Tests
 - Manuelle Testcases
 - Codeanalyse
 - Abnahmetests
- Die Lösungsarchitektur beschreibt den MVP entsprechend den Anforderungen der Helion AG.

In Absprache mit dem Expertengremium wurde am 27.01.2025 die Zielsetzung für die „Umsetzung eines MVP der «Baustellen App» angepasst. Deshalb ergeben sich zusätzliche Abgrenzungen:

- Der PoC hat keine Benutzerverwaltung. Es wird ein statischer Benutzer hinterlegt.
- Der PoC wird nicht den Funktionsumfang enthalten, wie er in der Lösungsarchitektur beschrieben wird. Es werden nur ausgewählte Funktionalitäten umgesetzt.

1.5 Aufbau der Arbeit

Die Masterarbeit ist folgendermassen aufgebaut: Zuerst wird in einer Einführung die Ausgangslage, Motivation, Problemstellung und die Zielsetzung erläutert.

Das zweite Kapitel enthält das Literaturreview zum Thema Datensynchronisation. Es behandelt die Beziehung und Problematik von Partitionstoleranz, Konsistenz und Verfügbarkeit in verteilten Systemen sowie die Datenübertragung, -synchronisation und -speicherung für mobile Applikationen.

Im dritten Kapitel wird die Methodik erläutert. Durch Transparenz in der Methodik soll die Reproduzierbarkeit gewährleistet und die Grundlage für eine kritische Bewertung geschaffen werden.

Im vierten Kapitel ist das Pflichtenheft aus der Anfrage der Helion AG an die isolutions AG aufgeführt. Dieses Pflichtenheft ist die Grundlage für die Lösungsarchitektur.

Im fünften Kapitel werden die Lieferobjekte beschrieben. Im ersten Teil wird die Lösungsarchitektur aufgezeigt. Der zweite Teil befasst sich besonders mit der Datensynchronisation und des gewählten Entwurfsmusters. Die Validierung der Umsetzung des MVPs schliesst dieses Kapitel ab.

2 Literaturreview

Dieses Kapitel beinhaltet die Grundlagen zur Datensynchronisation und Datenkonsistenz.

2.1 Grundlagen

Ausgehend von einem Multiusersystem mit verteilten Komponenten, wie beispielsweise einer Webapplikation mit einer Frontendkomponente als Client und dem Backend als Server, gibt es folgende Probleme. Zum einen den Aspekt der Synchronisation von Daten zwischen den Komponenten und zum anderen die Zusammenführung von Abweichungen der Datensätze. Der Aspekt der Datensynchronisation hat Einfluss auf die Verfügbarkeit der Daten in der Anwendung. Der zweite Aspekt beeinflusst die Konsistenz der Daten, welche für eine Geschäftsapplikation, wie in diesem Fall, von Bedeutung ist.

2.1.1 Set Reconciliation Problem

Das «Set Reconciliation Problem» beschreibt den Umstand, dass in einem System mit verteilten Knoten, ein Datenset sich auf einem Knoten über die Zeit verändert und die Änderungen an einem bestimmten Punkt wieder zusammengeführt werden sollen [3]. Jedes Client/Server System wird mit diesem Problem konfrontiert. Eine einfache Lösung wäre das Überschreiben der Datensets auf den Clients, ausgehend vom Datenstand auf dem Server [3]. Komplexere Lösungen verwenden Algorithmen, um die Differenz der Datensets zu bestimmen und zusammenzuführen [4]. Primär hängt die Wahl der verwendeten Lösung von den Anforderungen an das System, die für die Umsetzung verfügbare Zeit und Budget sowie der verfügbaren Infrastrukturen (Speicher, Netzwerkbandbreite, Rechenleistung) ab [3].

2.1.2 CAP Theorem

Gemäss S. Gilbert und N. Lynch [5] spielt in einem verteilten System das CAP Theorem eine Rolle. Es besagt, dass ein verteiltes System nicht gleichzeitig Konsistenz, Verfügbarkeit und Partitionstoleranz garantieren kann. Das CAP Theorem wurde als erstes von Eric Brewer, im Kontext von geografisch verteilten Webservices, aufgestellt [5]. Dieses Theorem ist auf jedes System anwendbar, welches gemeinsame Datensets in verteilten Konten verwendet [5]. In den nächsten Kapiteln werden die drei Komponenten des CAP Theorems erläutert: «Konsistenz», «Verfügbarkeit» und «Partitionstoleranz».

2.1.2.1 Konsistenz

Consistency bzw. Konsistenz beschreibt die Eigenschaft, dass der Informationsgehalt von Daten für alle Komponenten oder Teilnehmer in einem System zu jedem Zeitpunkt gleich ist [3].

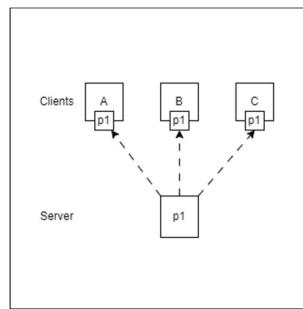


Abbildung 1 Konsistentes System

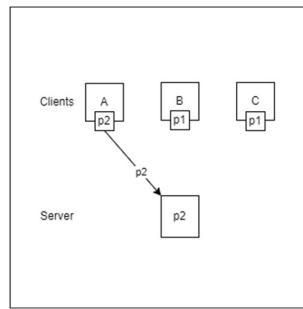


Abbildung 2 Inkonsistentes System

Abbildung 1 zeigt ein verteiltes System mit drei Clients. Alle Clients erhalten vom Server den gleichen Datensatz $\{p_1\}$ und haben somit die identischen Informationen. Das System ist somit konsistent.

Abbildung 2 zeigt ein inkonsistentes System. Dies ist beispielsweise in dem Moment der Fall, wenn der Client A den Datensatz bearbeitet und an den Server sendet, mit der Anweisung den Datensatz auf dem Server anzupassen zu $\{p_2\}$. Der Datensatz ist nicht mehr konsistent da der Client B und C den Datensatz $\{p_1\}$ haben, während Client A und der Server den Datensatz $\{p_2\}$ haben.

Um die Konsistenz wieder herzustellen, muss Client B und C den Datensatz vom Server abfragen, um die Version $\{p_2\}$ zu erhalten.

2.1.2.2 Verfügbarkeit

Als verfügbar bzw. available gilt ein Service, wenn dieser auf eine Anfrage eine Antwort zurücksendet. Je nach Art des Systems ist auch die Zeitspanne von der Anfrage bis zur Antwort von Bedeutung. In einem Echtzeit-System oder einer Benutzeranwendung kann eine Antwort, welche zu lange braucht genau so schlecht wie keine Antwort [5] sein.

2.1.2.3 Partitionstoleranz

Die Partitionstoleranz bzw. Partition Tolerance beschreibt die Fähigkeit eines verteilten Systems, mit Kommunikationsausfällen oder -verzögerungen umzugehen. Ein System ist partitionstolerant, wenn es, trotz Kommunikationsausfall zwischen Knoten im System, weiter arbeiten kann [6].

2.2 Entwurfsmuster für die Datensynchronisation in mobilen Anwendungen

Gemäß Z. McCormick [7] gibt es einige Entwurfsmuster (Design Pattern) für die Synchronisation von Daten im mobilen Applikationsdesign. Diese Entwurfsmuster werden in diesem Abschnitt beschrieben. Sie lassen sich grob in drei Bereiche aufteilen, «Datensynchronisation», «Datenspeicher und Verfügbarkeit» und «Datenübertragung» [8].

2.2.1 Datensynchronisation

Die Datensynchronisation lässt sich in zwei Entwurfsmuster einteilen: synchrone und asynchrone Datensynchronisation. Beide Entwurfsmuster haben Vor- und Nachteile und eignen sich für spezifische Einsatzgebiete.

2.2.1.1 Synchrone Datenübertragung



Abbildung 3 Datensynchronisation als synchroner Fluss [7]

In Abbildung 3 wird ein synchroner Datenfluss in einer mobilen App dargestellt. Eine synchrone Datenübertragung führt in jedem Softwaresystem dazu, dass der Prozess der anfragenden Komponente auf die Antwort wartet. Im Kontext einer mobilen Applikation bedeutet dies, dass die Benutzeroberfläche blockiert wird. Erst wenn die Synchronisation der Daten abgeschlossen ist, kehrt die Applikation wieder in einen benutzbaren Zustand zurück. Die synchrone Datenübertragung kommt in folgenden Fällen zur Anwendung:

- Wenn dringend ein komplettes Datenset für die Funktion der Applikation nötig ist [7].
- Wenn die Antwort nötig ist, um zu bestimmen in welchen Status die Applikation vorrücken soll [7].

Vorteile:

- Das Statusmanagement der Anwendung ist simpel. Da der Statuswechsel strikt sequenziell ist, werden zusätzliche Zustände, wie sie in der asynchronen Datenübertragung verwendet werden, nicht benötigt [7].

Nachteile:

- Wie bereits in der Erklärung beschrieben, ist die Benutzeroberfläche für den Benutzer gesperrt, weil der darunterliegen Prozess auf die Antwort der Datensynchronisation wartet. Entsprechend leidet die Benutzererfahrung [7].

2.2.1.2 Asynchrone Datenübertragung

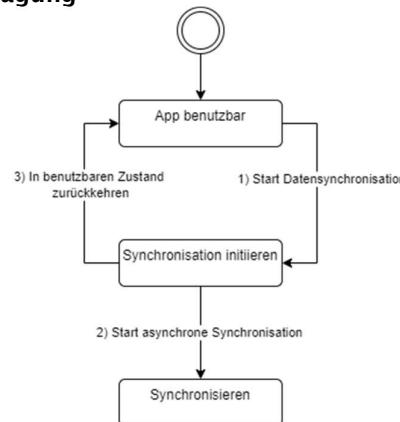


Abbildung 4 Darstellung einer asynchronen Datensynchronisation [7]

In der Abbildung 4 ist der Ablauf einer asynchronen Datensynchronisation zu sehen. Der Unterschied zur synchronen Datensynchronisation ist, dass sich die Applikation bei der Initiierung der Synchronisation in zwei verschiedene Zustände begibt. Die Benutzeroberfläche kehrt in den benutzbaren Zustand zurück, während die Datensynchronisation im Hintergrund läuft [7], dargestellt als Zustand «Synchronisieren».

Vorteile:

- Die Applikation bleibt für den Benutzer benutzbar [7].
- Erlaubt den Datenaustausch, auch wenn die mobile App nicht im Vordergrund ist. [7].

Nachteile:

- Mehrere konkurrierende Anfragen an ein Datenset können auftreten und müssen behandelt werden [7]. Eine mögliche Lösung ist die asynchrone Übertragung von Transaktionen, welche nach der Übertragung synchron angewendet werden [7].
- Die übertragene Datenmenge kann nicht vorausgesagt werden. Das kann einen negativen Einfluss auf die Kosten der Netzwerkbandbreite und deren Verfügbarkeit haben [7].

2.2.2 Datenspeicher und Verfügbarkeit

In diesem Bereich gibt es ebenfalls zwei Entwurfsmuster, welche sich auf die lokale Verfügbarkeit auf dem Gerät auswirken. Bei der Wahl dieser Entwurfsmuster sind besonders der verfügbare Speicherplatz auf dem Gerät und die Netzwerkbandbreite von Bedeutung [7].

2.2.2.1 Partial Storage

Das erste Entwurfsmuster ist «Partial Storage» bzw. Teilspeicher. Dieses Entwurfsmuster reduziert den Speicherbedarf der Daten auf dem Gerät, da mit diesem Muster nur die gerade benötigten Daten vom Backend geladen werden.

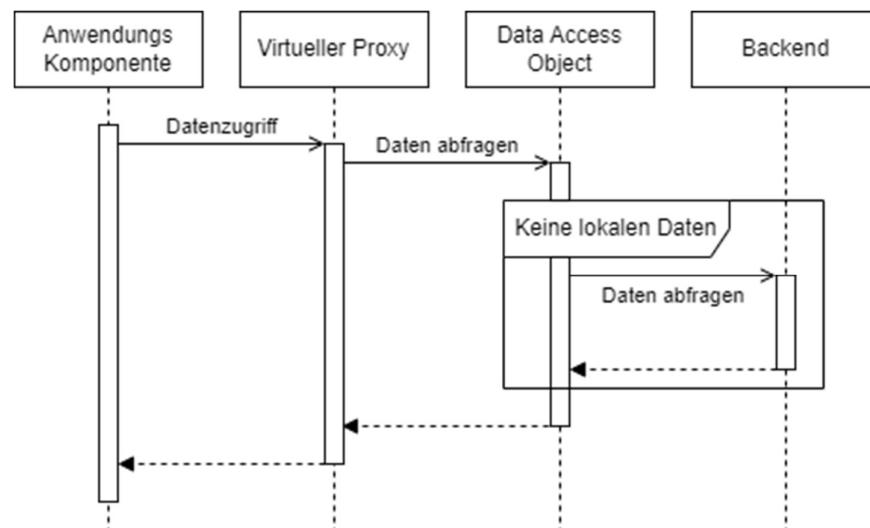


Abbildung 5 Ablauf von «Partial Storage» Datenabfragen [7]

In Abbildung 5 ist der Ablauf und das Prinzip von «Partial Storage» zu sehen. Oft wird das Muster «Partial Storage» mit der Implementierung des «Virtual Proxy» und des «Data Access Object» (DAO) Entwurfsmusters umgesetzt [7]. Der «Virtual Proxy» imitiert das eigentliche Datenobjekt, in dem es das gleiche Interface wie das Datenobjekt implementiert [9]. Der virtuelle Proxy fängt Methoden Calls und Zugriffe auf Objektattribute ab und erlaubt eine gezielte Initialisierung von Feldern in dem Moment, wenn sie gebraucht werden. Dieses Vorgehen nennt sich «Lazy Loading» [7]. Das DAO arbeitet an der Abfrage der Daten, in dem es den Zugriff auf Daten abstrahiert [9]. Sind die angefragten Daten im lokalen Speicher oder einem Cache verfügbar, antwortet das DAO mit einer

Kopie der Daten aus dem lokalen Speicher. Ansonsten initiiert das DAO eine Datenabfrage über das Netzwerk an das Backend [7].

Vorteile:

- Der benötigte Speicher auf den mobilen Gerät ist reduziert, da nur die benötigten Teile eines Datensatzes in den lokalen Speicher geladen werden [7]. Dieser Umstand erlaubt auch die Verwendung von weitaus umfangreicheren Datensätzen [7].
- Dieses Muster erlaubt eine granulare Kontrolle darüber, wann welche Daten synchronisiert werden [7].

Nachteile:

- Die Netzwerkverbindung ist für dieses Muster ein Schwachpunkt. Da das «Lazy-Loading» eine Netzwerkverbindung erfordert, um Daten nachzuladen. Ein Unterbruch des Netzerwerks kann die mobile App unbrauchbar machen [7].

2.2.2.2 Complete Storage

Das zweite Entwurfsmuster ist der «Complete Storage» bzw. Komplettspeicher. Im Gegensatz zum «Partial Storage» wird in diesem Muster der komplette Datensatz auf ein mobiles Gerät geladen. Dieser Prozess wird üblicherweise beim Start der Applikation ausgeführt, damit subsequenter Datenzugriff immer aus dem lokalen Speicher erfolgen kann.

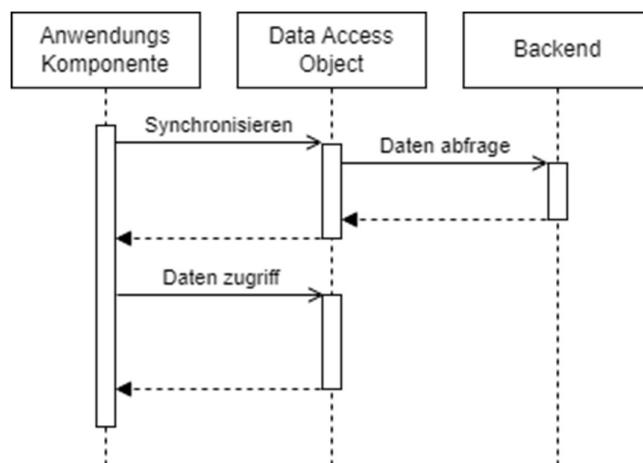


Abbildung 6 Darstellung der «Complete Storage» Datenabfrage [7]

In der Abbildung 6 ist zu erkennen, dass der Datentransfer initial gestartet wird und das DAO alle Datensätze aus dem Backend abfragt. Die Daten werden im lokalen Gerätespeicher abgelegt und subsequente Datenzugriffe erfolgen aus dem lokalen Speicher [7].

Vorteil:

- Die Abhängigkeit zur Netzwerkverfügbarkeit ist gering. Nachdem die Daten initial synchronisiert wurden, kann die Applikation auch ohne Konnektivität verwendet werden [7].

Nachteil:

- Es wird mehr Speicher auf dem mobilen Gerät vorausgesetzt [7].
- Die Datenübertragung ist ineffizient in Bezug auf die Netzwerkbandbreite, da beim Synchronisationsereignis alle Daten gleichzeitig übertragen werden [7].

2.2.3 Datenübertragung

Dieser Abschnitt behandelt das Thema Datenübertragung. Das Ziel ist die, in Bezug auf Netzwerkbandbreite optimierte, Übertragung von Daten zwischen Client und Server. Bei der Wahl des Entwurfsmusters zur Datenübertragung muss primär der Typ der zu übertragenden Daten und die verfügbare Netzwerkbandbreite in Betracht gezogen werden [7].

2.2.3.1 Full Transfer

Für kleine Datensätze oder Daten, welche jeweils als Ganzes verändert werden, beispielsweise ganze Dateien, eignet sich das Entwurfsmuster «Full Transfer». Das Entwurfsmuster zeichnet sich dadurch aus, dass der komplette Datensatz übertragen wird und den aktuellen Datensatz auf dem mobilen Gerät ersetzt.

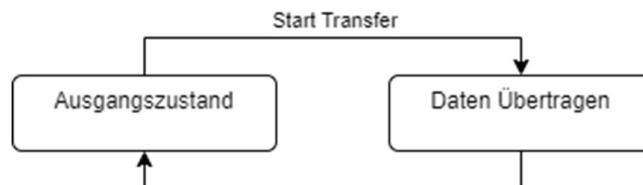


Abbildung 7 Darstellung des «Full Transfer» Übertragungsmuster [7]

In der Abbildung 7 ist zu erkennen, dass der «Full Transfer» ein sehr einfaches Entwurfsmuster ist. Der Transfer wird für das gesamte Datenset initiiert [7]. Dieses Entwurfsmuster eignet sich hingegen nur in Situationen, wo ein klarer Datenmaster definiert werden kann. In einem Szenario mit mehreren gleich berechtigten Clients, in Bezug auf die Editierbarkeit von Daten, kann mit diesem Entwurfsmuster nicht festgestellt werden, welche Änderung angewendet werden soll.

Vorteile:

- «Full Transfer» ist ein einfach zu implementierendes Entwurfsmuster [7].
- Kein Datenverlust in einem Szenario mit einem Datenmaster und Clients, welche nur lesend auf die Daten zugreifen.

Nachteile:

- Die Daten werden redundant gesendet, was zu einer ineffizienten Nutzung der Netzwerkkonfiguration führt [7].
- Eignet sich nur für spezifische Anwendungsszenarien.

2.2.3.2 Timestamp Transfer

Das Muster «Timestamp Transfer» basiert auf dem Vergleich von Zeitstempeln in einem Datensatz.

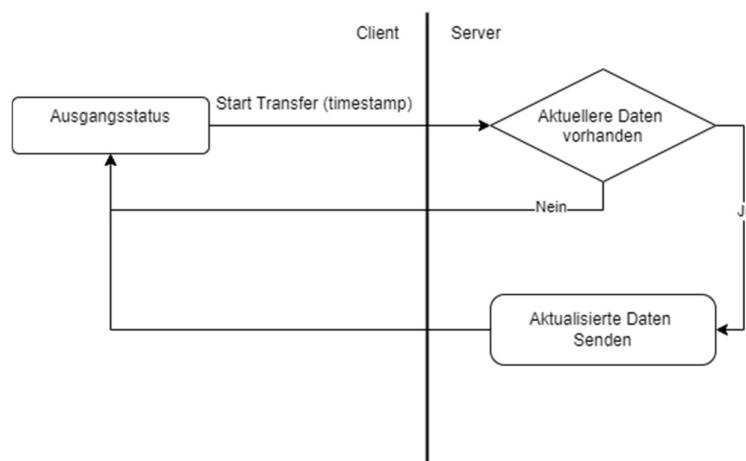


Abbildung 8 Darstellung des Musters «Timestamp Transfer» [7]

Die Abbildung 8 stellt die prinzipielle Logik eines «Timestamp Transfers» dar. Damit dieser Vergleich möglich ist, muss zu jedem Datensatz ein Zeitstempel des Erstellungszeitpunktes und ein Zeitstempel des Zeitpunktes der letzten Änderung gespeichert werden [7]. Die mobile Applikation sendet die Zeitstempel an den Server. Dieser vergleicht die Zeitstempel von der mobilen Applikation mit den eigenen Datensätzen und sendet aktuellere Datensätze an den Client [7]. Es werden jeweils ganze Datensätze übermittelt.

Vorteile:

- Die zu synchronisierenden Daten können anhand ihrer Aktualität gezielt ausgewählt werden, dies sowohl für den Upload wie auch für den Download. Dadurch wird die Menge an zu übertragenden Daten verringert [7].

Nachteile:

- Das Löschen von Datensätzen benötigt zusätzliche Logik, da in diesem Fall der gelöschte Datensatz nicht mehr für den Zeitstempelvergleich zur Verfügung steht [7].
- Die Quelle für den Zeitstempel muss konsistent sein, damit die Zeitstempel vergleichbar bleiben [7].

2.2.3.3 Mathematical Transfer

Das dritte Entwurfsmuster macht sich mathematische Algorithmen zu Nutze, um zu bestimmen welche Teile eines Datensatzes übermittelt werden sollen [7]. Es werden nur Daten ausgetauscht, welche geändert wurden. Die mathematischen Algorithmen sind in der Lage, Änderungen effizienter zusammen zu führen und die Menge an Daten in einer Übertragung auf ein Minimum zu reduzieren [7]. Beispiele für mathematische Methoden sind:

- „Invertible Bloom Lookup Tables“ (IBLTs) [10]
- “Characteristic Polynomial Interpolation” [7]

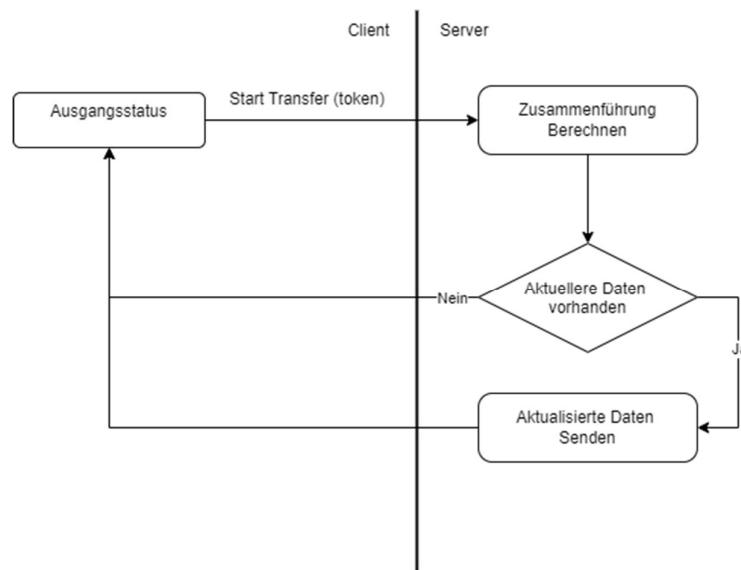


Abbildung 9 Darstellung des Musters «Mathematischer Transfer» [7]

Wie in Abbildung 9 zu sehen ist, weist das Muster «Mathematischer Transfer» Ähnlichkeiten zum «Timestamp Transfer» Muster auf. Der Unterschied ist die mathematische Berechnung der Änderungen [7].

2.2.3.3.1 Invertible Bloom Lookup Tables

«Invertible Bloom Lookup Tables» oder in kurz IBLT ist eine Datenstruktur für die Determinierung von Differenzen zwischen zwei Datensätzen. Ein gern angeführtes Beispiel ist der Vergleich einer Liste zwischen zwei Parteien. Wobei das Ziel ist, dass beide Partien am Ende über die gleiche Liste verfügen. Anstatt das beide Parteien ihre Liste an die jeweilige andere Partei senden, ist es effizienter nur die Differenz der Listen auszutauschen. Vereinfacht gesagt ist eine IBLT eine chronologische Liste von Insert und Delete Operationen eines Datensatzes. IBLTs haben die Eigenschaft, dass sich zwei IBLTs subtrahieren lassen und somit das Ergebnis die Differenz zweier Datenstände ist [11].

2.2.3.3.2 Characteristic Polynomial Interpolation

Stark vereinfacht gesagt, wird sich in diesem Verfahren der Umstand zu Nutzen gemacht, dass beim Vergleich des «Characteristic Polynomial» zweier Datensets, die Gemeinsamkeiten gegenseitig aufheben. Als Resultat bleibt die Differenz zwischen den Datensets [12].

Vorteile:

- Mit diesem Muster kann die benötigte Bandbreite auf ein Minimum reduziert werden [7].
- Geeignet für Szenarien, wenn die effiziente Nutzung der verfügbaren Bandbreite kritisch ist.

Nachteile:

- Mathematische Methoden für die Berechnung sind oft stark abhängig vom Kontext der Daten [7]. Ändert der Datenkontext muss auch der Algorithmus angepasst werden.
- Die Komplexität der mathematischen Methoden ist hoch. Entsprechend wird für die Implementierung mehr Zeit benötigt [7].

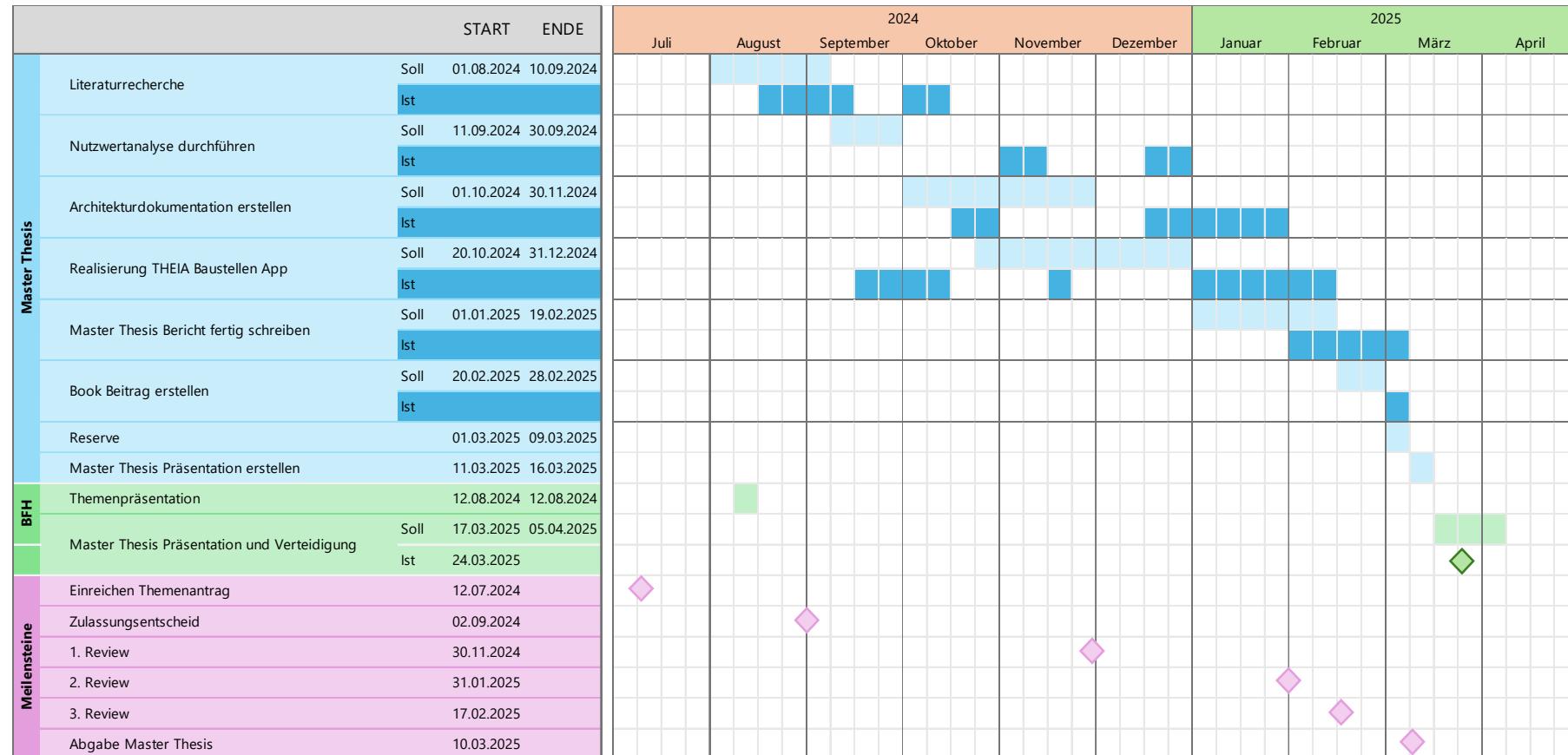
3 Methodik

In diesem Kapitel wird die verwendete Methodik beschrieben. Durch die Transparenz in der Methodik wird nicht nur die Reproduzierbarkeit gewährleistet, sondern auch die Grundlage für eine kritische Bewertung der Arbeit gelegt.

3.1 Projektmanagement

Für die Umsetzung wurde eine agile Vorgehensweise gewählt. Der geplante Fortschritt und der tatsächliche Fortschritt der Masterarbeit wird im folgenden Projektplan dargestellt. Er zeigt die Termine, Meilensteine, geplante und effektive Umsetzungsabschnitte.

3.2 Masterarbeit Zeitplan



Begründung der Abweichung im Zeitplan

Zu Beginn wurde zu viel Zeit in die Einarbeitung in die Technologie verwendet. Außerdem benötigte der Aufbau der Entwicklungsumgebung mehr Zeit als geplant. Zusätzlich beeinträchtigte eine Geburt im November die zur Verfügung stehende Zeit. Dies führte dazu, dass mit der Umsetzung der Lösungsarchitektur und der Applikation erst richtig Mitte Dezember begonnen wurde.

3.3 Entwurf der Lösungsarchitektur

Die Lösungsarchitektur wurde auf Basis der Qualitätsanforderungen erstellt. Als Modell für das «Software Architektur Dokument» diente das arc42-Modell von Dr. Peter Hruschka und Dr. Gernot Starke [13]. Die Lösungsarchitektur beschreibt die statischen Sichtweisen auf die Bausteine, die erwarteten Verhaltensweisen der Anwendungen, Konzepte, Strategien und Architekturentscheidungen. Dies fördert eine klare Kommunikation und ein gemeinsames Verständnis mit den Stakeholdern. Alle Diagramme und visuellen Darstellungen wurden mittels der Software «draw.io» erstellt.

3.4 Analyse der Datensynchronisation

Um die Frage nach der geeignetsten Methode für die Datensynchronisation zu beantworten, wurde eine Nutzwertanalyse durchgeführt. Beurteilt wurden die drei Varianten «Full Transfer», «Timestamp Transfer» und «Mathematical Transfer». Folgende Kriterien wurden analysiert: Komplexität, Speicherbedarf, Rechenleistung, Netzwerkbandbreite und Umgang mit Datenkonflikten.

3.5 Umsetzung der «Baustellen App» als PoC

Ursprünglich wurde die Umsetzung der «Baustellen App» im MVP-Umfang als Ziel der Arbeit gesetzt anhand der Lösungsarchitektur für den MVP aus Kapitel 4.3. In Absprache mit dem Expertengremium wurde am 27.01.2025 entschieden, die ursprüngliche Zielsetzung zur Umsetzung der «Umsetzung des MVPs» auf einen PoC zu reduzieren (Siehe dazu Kapitel 1.3 Ziele der Arbeit)

Der PoC muss folgende Anforderungen erfüllen:

- Er zeigt die Funktion des eingesetzten Entwurfsmusters für die Datensynchronisation.
- Er demonstriert den Zugriff eines Benutzers auf Projektdaten.

Die Backend Services werden mit ASP.NET 9 und C# umgesetzt. Die mobile App wird mit dem multiplattform Framework .NET MAUI umgesetzt. Verwendet wird «Visual Studio 2022» als Entwicklungsumgebung. Für die Quellcodeverwaltung und CI/CD Pipeline werden GitHub Workflows verwendet. Die Plattform für die «Baustellen App» wird der Public Cloud «Azure» gehostet.

4 Ergebnisse

In diesem Kapitel werden die Ergebnisse der Arbeit dargelegt. Zuerst werden die Anforderungen an die «Baustellen App» aufgelistet (Kapitel 4.1). Anschliessend folgt die Lösungsarchitektur für den MVP (Kapitel 4.3) inklusive der Nutzwertanalyse (Kapitel 4.3.11.4). Zum Abschluss wird die Implementierung beschrieben (Kapitel 4.3.13.2).

4.1 Pflichtenheft

Im Pflichtenheft sind die Business relevanten Anforderungen an die Applikation sowie Mockups aufgeführt. Die Mockups visualisieren, wie Anforderungen in die Benutzeroberfläche realisiert werden sollen. Je nach Plattform werden sich die Elemente visuell von den Mockups unterscheiden.

Die Anforderungen und Mockups sind aus dem Angebot «Konzeption THEIA für die Baustelle» für die Helion AG abgeleitet. Das Angebotsdokument wurde von der isolutions AG verfasst. Da das Dokument interne Informationen enthält, wurden die Anforderungen generalisiert.

Eine zentrale Anforderung ist, dass die «Baustellen App» in mehreren Phasen entwickelt wird. Die erste Implementierung wird als MVP ausgelegt, um die Stossrichtung der mobilen Applikation und die Akzeptanz bei den Nutzern zu überprüfen. Nach dieser ersten Version werden weitere Funktionen geplant und implementiert.

4.1.1 Authentifizierung

In diesem Abschnitt sind die Anforderungen an die Authentifizierung aufgelistet.

Tabelle 1 Anforderungen an die Authentifizierung

Nr	Name	Beschreibung
REQ-001	Benutzerzugriff	Die Benutzer müssen sich beim Einstieg in die App als erstes authentifizieren. Erst nach Authentifizierung und Autorisierung können die Benutzer auf die Inhalte zugreifen.
REQ-002	Speicherung der Session	Nach dem Einloggen bleibt die Session des Benutzers in der App gespeichert, damit sich der Benutzer nicht immer wieder neu anmelden muss. Nach einer definierten Zeitdauer wird die Session ungültig und der Benutzer muss sich neu anmelden.
REQ-003	Logout	Der Benutzer muss nach dem Anmelden in der Applikation die Möglichkeit haben, sich wieder abzumelden

4.1.2 Projektübersicht

In diesem Abschnitt sind die Anforderungen an die Projektübersicht aufgelistet.

Tabelle 2 Anforderungen an die Projektübersicht

Nr	Name	Beschreibung
REQ-004	Projektliste	Nach dem Start der Applikation soll als erste Seite eine Liste mit den Projekten dargestellt werden. Vorausgesetzt der Benutzer ist eingeloggt.
REQ-005	Personalisierte Projektliste	In der Projektliste werden nur diejenigen Projekte angezeigt, auf die der Benutzer die Zugriffsberechtigung hat.
REQ-006	Projektinformationen	Jedes Projekt ist mit dem Namen und der Ortschaft der Baustelle aufgelistet.

REQ-007	Projektsuche	Im Suchfeld kann nach Projektnamen und Ortschaften gesucht werden. Die Liste passt sich laufend dem Suchbegriff an. Der Suchbegriff muss wieder gelöscht werden können.
REQ-008	Projektauswahl	Mit einem Klick auf eines der Projekte können die Projektdetails aufgerufen werden.
REQ-009	Offlinefähigkeit	Der Benutzer kann in der Projektliste Projekte markieren, die offline verfügbar sein sollen. Sämtliche Projektdaten zu diesen markierten Projekten sind für den Benutzer offline verfügbar.

4.1.3 Projektdetails

In diesem Abschnitt sind die Anforderungen an die Projektdetails aufgelistet.

Tabelle 3 Anforderungen an die Projektdetails

Nr	Name	Beschreibung
REQ-010	Projektdetails	Es werden die wichtigsten Infos eines Projektes angezeigt. Ist eine Internetverbindung vorhanden, werden die Daten vom Backend geladen. Ohne Internetverbindung werden die Daten aus dem lokalen Speicher geladen.
REQ-011	Adressen in Maps öffnen	Kontakt- und Objektadresse können per Klick in Maps (iOS) oder Maps (Android) geöffnet werden.
REQ-012	Absprung auf externe Ablagen	Wird auf dem Projekt ein Link zu einer externen Ablage mitgeliefert, wird in der Detailansicht ein Link Button auf die externe Ablage angezeigt.
REQ-013	Notizen zum Auftrag	Über einen Button wird der Benutzer zu der Notizenfunktion weitergeleitet.
REQ-014	Inbetriebnahme (IBN)	Über einen Button wird der Benutzer zum Inbetriebnahmeprotokoll weitergeleitet.
REQ-015	Zurück zur Übersicht	Befindet sich der Benutzer ausserhalb der Projektübersicht muss er über den Zurück Button wieder über die Projektübersicht zurücknavigieren.
REQ-016	Offlinefähigkeit	Sämtliche Daten dieses Projekts müssen auch ohne aktive Internetverbindung auf den mobilen Client abrufbar sein. Links auf externe Online-Ablagen funktionieren im Offline-Modus nicht.
REQ-017	Absprung nach MS Teams	Auf einem Projekt kann ein Link auf einen Teams Channel oder Chat hinterlegt werden. In der Projektansicht wird dieser als Link Button angezeigt. Ein Klick auf den Button, leitet den Benutzer weiter zu Teams.
REQ-018	Absprung nach MS SharePoint	In den Projekten kann ein Link auf eine SharePoint Ablage hinterlegt werden. In der Projektansicht wird dieser als Link Button angezeigt. Ein Klick auf den Button leitet den Benutzer weiter auf die SharePoint Ablage.
REQ-019	Logistik Fehler melden	Logistikfehler werden in einer vorhandenen Power App erfasst. In der Projektansicht gibt es einen Link Button, der einen Deeplink zu dieser Power App enthält. Ein Klick führt den Benutzer direkt zur Power App, um Logistikfehler zu melden.

4.1.4 Notizen zum Projekt

In diesem Abschnitt werden die Anforderungen zu den Projektnotizen aufgeführt.

Tabelle 4 Anforderung zu den Notizen

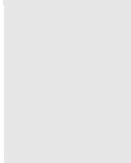
Nr	Name	Beschreibung
REQ-020	Notizen-Übersicht	In einer Liste werden sämtliche bisher erfassten Notizen zum ausgewählten Projekt angezeigt.
REQ-021	Filter nach Kategorie	In einem Dropdown Feld kann die jeweilige Notizen-Kategorie gefiltert werden. Standardmäßig ist kein Filter gesetzt.
REQ-022	Detailansicht Notizen	In einer Detailansicht können die Notizen angezeigt werden. Externe Links (z.B. auf Word-Dokumente, welche in SharePoint abgelegt sind) können angeklickt und im externen Browser (Safari oder Chrome) angezeigt werden, falls die entsprechenden Berechtigungen vorhanden sind. Eingebettete Bilder werden geladen und angezeigt.
REQ-023	Notizen editieren und erstellen.	Notizen können editiert und erstellt werden.
REQ-024	Zurück zur Notizen-Übersicht	Mittels eines Zurück-Buttons muss aus der Detailansicht wieder in die Notizen-Übersicht gewechselt werden können.
REQ-025	Offlinefähigkeit	Sämtliche Notizen der geladenen Projekte müssen auch ohne aktive Internetverbindung aus dem mobilen Client abrufbar sein. Externe Links funktionieren offline nicht.

4.1.5 Inbetriebnahme

In diesem Abschnitt werden Anforderungen zur Inbetriebnahme (IBN) aufgeführt.

Tabelle 5 Anforderungen zu der Inbetriebnahme

Nr	Name	Beschreibung
REQ-026	Inbetriebnahme (IBN)	Für jedes Projekt kann eine Inbetriebnahme erfasst werden. Ist bereits eine IBN da, kann diese editiert werden.
REQ-027	Checkliste anzeigen	Es werden drei Checklisten mit den zu den Checklisten erforderlichen Dokumenten angezeigt. Dazu muss das Dokument «IBN SOP 2.0» angezeigt werden. Die Checkpunkte können je Projekt verschieden sein. Jedes Checklisten-Item ist entweder ein Pflichtfeld oder optional.
REQ-028	Vorhandene Bilder	Sind bereits Bilder vorhanden, können diese via Ansicht-Icon angezeigt werden (aktuell nicht vorhanden im Mockup).
REQ-029	Neue Bilder hinzufügen	Im App kann pro Checklisten-Item eines oder mehrere Bilder hochgeladen werden. Entweder werden diese gleich mit der Kamera aufgenommen, oder sie werden aus der Camera Roll hinzugefügt. Sobald die Fotos hochgeladen wurden, sind sie auch über die Ansicht verfügbar (REQ-028).
REQ-030	Anmerkungen	Es können Anmerkungen in einem Textfeld erfasst oder editiert werden. Dabei ist keine Formatierung vorgesehen.
REQ-031	Speichern und Inbetriebnahme	Der aktuelle Stand der Checkliste kann jederzeit gespeichert oder kombiniert gespeichert und abgeschlossen werden. Ist der IBN einmal abgeschlossen, kann er nur noch angeschaut werden.
REQ-032	Offlinefähigkeit	Sämtliche IBNs der geladenen Projekte (eines je Projekt) müssen auch ohne aktive Internetverbindung aus dem

 mobilen Client abrufbar sein. Auch müssen vorhandene Bilder angezeigt und neue Bilder (Multiupload) hinzugefügt werden können. Sobald eine Onlineverbindung wieder verfügbar ist, werden die Daten über die APIs ins Backend geladen.

4.1.6 Mockups zur Benutzeroberfläche der mobilen Applikation

Mit den Mockups soll das Konzept der Benutzeroberfläche visualisiert werden. Die Mockups wurden mit dem UI/UX Design Tool Figma³ erstellt, wobei Standard-Asset-Bibliotheken verwendet wurden. Das «Look and Feel» der visuellen Elemente entspricht nicht dem finalen Aussehen in der Applikation, sondern dient nur zur Veranschaulichung der Benutzeroberfläche und der User Experience

4.1.6.1 Projektbezogene Ansichten

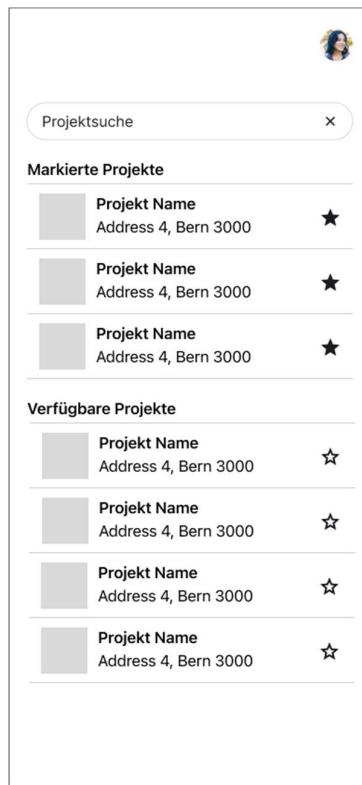


Abbildung 10 Mockups: Darstellung der Projektübersicht

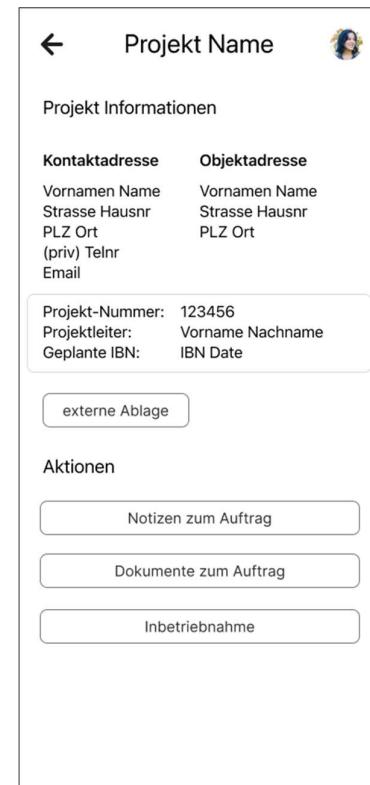


Abbildung 11 Mockups: Detailansicht eines Projektes

³ <https://www.figma.com/>

4.1.6.2 Notizenansichten

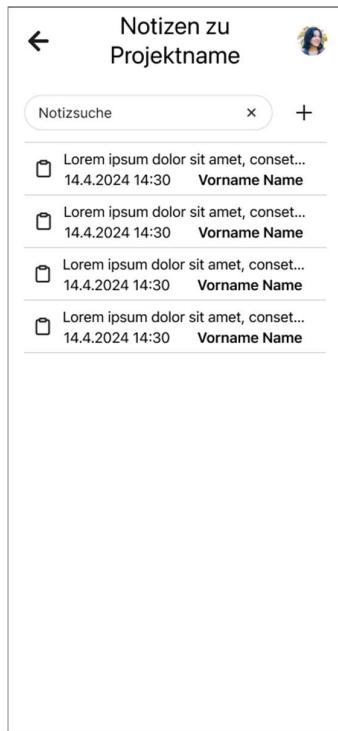


Abbildung 12 Mockups: Übersicht der Notizen

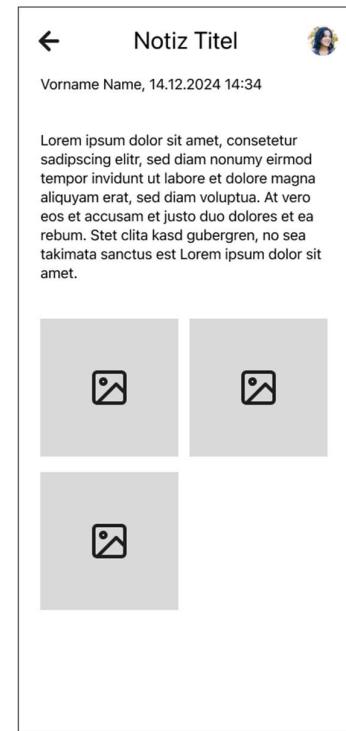


Abbildung 13 Mockups: Detailansicht einer Notiz

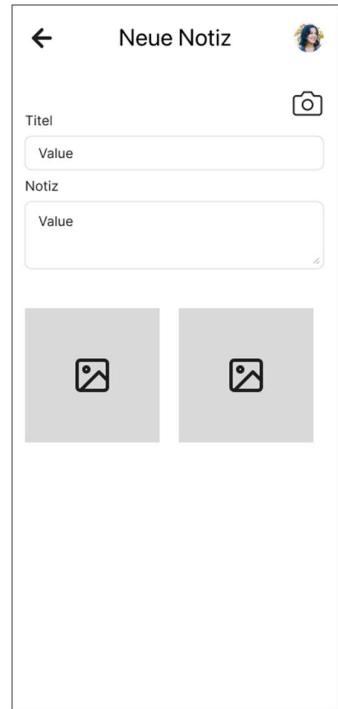


Abbildung 14 Mockups: Notiz erfassen

4.1.7 Nicht Funktionale Anforderungen

In diesem Abschnitt werden die nicht funktionalen Anforderungen aufgelistet. Diese Anforderungen wurden von der Helion AG formuliert.

Tabelle 6 Auflistung der nicht funktionalen Anforderungen

Nr	Name	Beschreibung
REQ-NF-001	Leistung	Die App soll, insbesondere beim Laden von Inhalten und beim Wechseln zwischen den Ansichten, reaktionsschnell sein.
REG-NF-002	Benutzerfreundlichkeit	Die Benutzeroberfläche sollte intuitiv gestaltet sein, um eine nahtlose Experience zu gewährleisten.
REG-NF-003	Sicherheit	Die App sollte sicher sein, sowohl in Bezug auf Datenübertragung wie auch lokale Datenspeicherung. Authentifizierung über Azure AD muss möglich sein.
REG-NF-004	Verteilbarkeit	Die App sollte möglichst einfach auf den Geräten der Mitarbeiter installiert werden können.
REG-NF-005	Wartbarkeit	Die App sollte wartbar sein, neue Versionen sollen möglichst einfach auf die Geräte der Mitarbeiter verteilt werden können.
REG-NF-006	Offlinefähigkeit	Damit die App auch auf der Baustelle und im Keller verwendet werden kann, sollen die Funktionen auch offline verfügbar sein. Nicht geladene Inhalte sollen bei Onlineverbindung heruntergeladen werden können. Für Inhalte die nur im Online-Modus angezeigt werden können, sollte eine passende Fehlermeldung angezeigt werden.
REG-NF-007	Gerätekompatibilität	Die App soll auf Android Version 12 und neuer sowie iOS 16 und neuer lauffähig sein.

4.2 Anwendungsfälle «Baustellen App»

Basierend auf den Anforderungen aus dem Pflichtenheft im Kapitel 4.1, wurden für den MVP der «Baustellen App» Anwendungsfälle abgeleitet. Die Anwendungsfälle dienen zur Visualisierung welche Aktionen das System ausführen kann, welche Akteure mit dem System interagieren sowie in welcher Beziehung die Akteure und Aktionen stehen. [14]

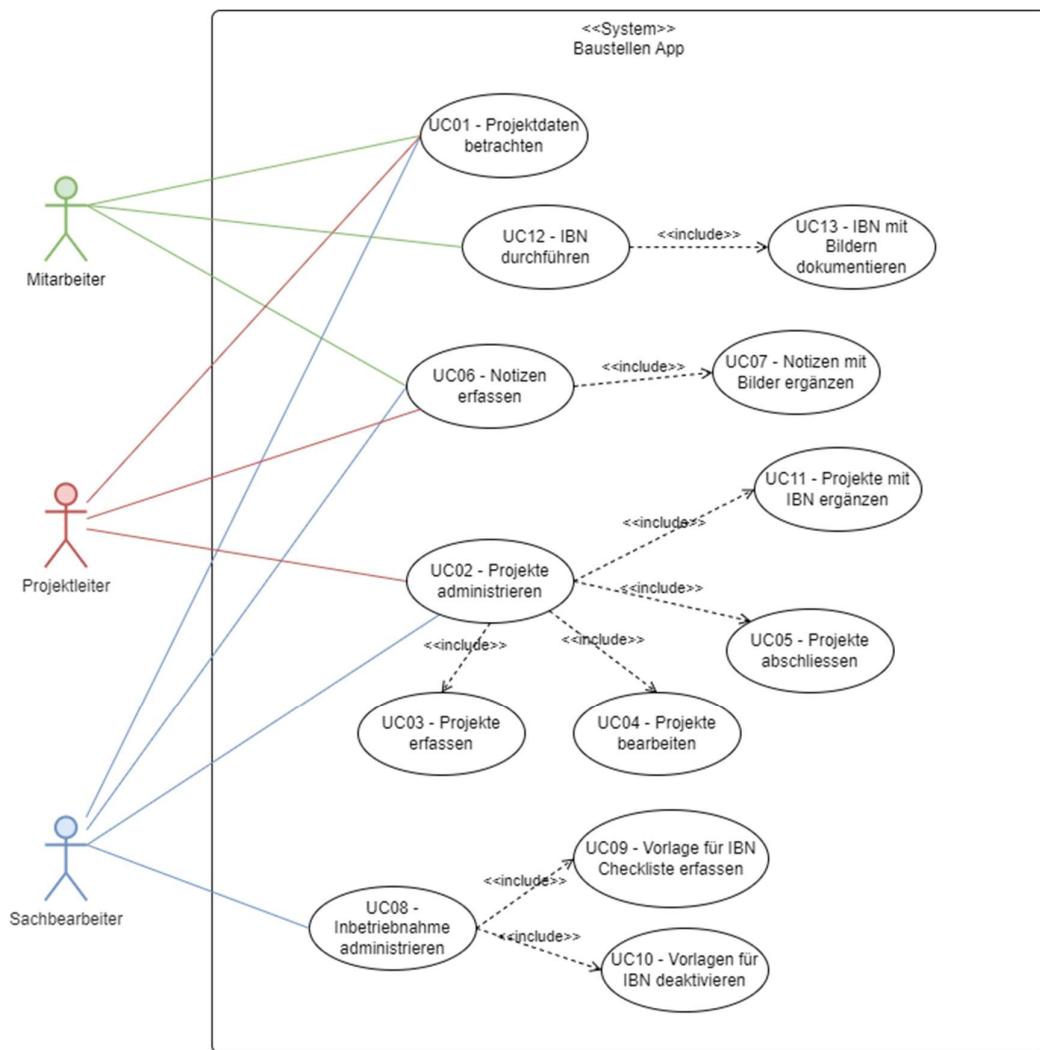


Abbildung 15 Anwendungsfalldiagramm für die Applikation «Baustellen App»

In der nachfolgenden Tabelle sind die Anwendungsfälle beschrieben.

Tabelle 7 Beschreibung der Anwendungsfälle

ID	Name	Kurzbeschreibung
UC01	Projektdaten betrachten	Projektdaten können von allen Akteuren der «Baustellen App» betrachtet werden. Sachbearbeiter betrachten Projekte in der «Baustellen App» via Desktop Client. Projektleiter und Mitarbeiter müssen Projekte mit dem mobilen Client aufrufen können.
UC02	Projekte administrieren	Sachbearbeiter und Projektleiter administrieren die «Baustellen App»-Projekte.
UC03	Projekte erfassen	Sachbearbeiter und Projektleiter eröffnen Projekte.

UC04	Projekte bearbeiten	Sachbearbeiter und Projektleiter bearbeiten Projekte.
UC05	Projekte abschliessen	Sachbearbeiter und Projektleiter schliessen Projekte ab.
UC06	Notizen erfassen	Sachbearbeiter, Projektleiter und Mitarbeiter können Notizen zu einem Projekt erfassen. Diese Notizen erlauben den Austausch von zusätzlichen, projektbezogenen Informationen.
UC07	Notizen mit Bildern ergänzen	Sachbearbeiter, Projektleiter und Mitarbeiter können zu ihren Notizen Bilder hinzufügen. Die Bilder stammen entweder aus der Fotobibliothek des Gerätes oder werden direkt mit der Gerätakamera erstellt.
UC08	Inbetriebnahme (IBN) administrieren	Sachbearbeiter verwalten die Vorlagen für die Checklisten.
UC09	Vorlage für IBN-Checkliste erfassen	Sachbearbeiter erfassen neue Vorlagen für Checklisten.
UC10	Vorlage für IBN deaktivieren	Sachbearbeiter deaktivieren Vorlagen, damit diese nicht weiterverwendet werden können.
UC11	Projekte mit IBN ergänzen	Sachbearbeiter oder Projektleiter wählen aus den Vorlagen der IBN-Checklisten eine aus und fügen diese Vorlage einem Projekt hinzu.
UC12	IBN durchführen	Mitarbeiter führen die IBN auf der Baustelle durch und füllen dabei die Checkliste in der «Baustellen App» aus.
UC13	IBN mit Bildern dokumentieren	Für die Belegbarkeit der durchgeföhrten IBN erfasst der Mitarbeiter Bilder und fügt diese Bilder der IBN hinzu.

4.3 Lösungsarchitektur

In diesem Kapitel wird die Lösungsarchitektur für die «Baustellen App» aufgezeigt. Als Grundlage für die Dokumentation wurde das Template Arc42 von Dr. Peter Hruschka und Gernot Starke verwendet [13]. In der Lösungsarchitektur wird die Umsetzung der «Baustellen App» als MVP beschrieben. Im Rahmen der Masterthesis wird die «Baustellen App» als PoC umgesetzt, um spezifische Anwendungsfälle aufzuzeigen. Die Lösungsarchitektur hingegen beschreibt den kompletten Funktionsumfang des MVPs.

4.3.1 Einführung und Ziele

Die «Baustellen App» ist ein Softwaresystem und soll als Kernanwendung für die Projektabwicklung der Helion AG dienen. Von der Beratung über den Verkauf, Planung und Durchführung der Dienstleistungen der Helion AG, bietet die «Baustellen App» Unterstützung für die durchführenden Personen und dient als zentraler Zugriffspunkt für Projektdaten, Administration und Prozessdurchführung. Die «Baustellen App» soll in einem ersten Schritt als MVP mit einem minimalen Funktionsumfang für eine kleine Benutzergruppe umgesetzt werden.

4.3.2 Qualitätsziele

Die Qualitätsziele an den MVP sind aus den Anforderungen im Pflichtenheft 4.1 der Helion AG abgeleitet. Im Zentrum steht die Prämisse: Die Lösung soll die Akteure: Mitarbeiter, Projektleiter und Sachbearbeiter in ihrer Arbeit unterstützen.

Tabelle 8 Qualitätsanforderungen an die «Baustellen App»

ID	Qualitätsziel	Beschreibung
QZ-001	Zuverlässigkeit - Verfügbarkeit	Ein wichtiges Qualitätsziel ist die Verfügbarkeit der App und der Daten in der App. Es ist dringend notwendig, dass der Mitarbeiter auf der Baustelle Zugang zu den Projektdaten hat und wichtige Arbeitsschritte, wie die Materialbestellung und Inbetriebnahme, durchführen kann. Dies muss er auch tun können, wenn keine Internetverbindung vorhanden ist.
QZ-002	Sicherheit - Integrität	Der zentrale Speicherort für die Daten der Applikation ist das Backend System der «Baustellen App». Da während der Verwendung der mobilen App im offline Modus konkurrierende Datenmanipulationen auftreten können, muss sichergestellt werden, dass die Integrität der Daten gewährleistet ist. Das System muss Konflikte in konkurrierenden Datenbearbeitungen transparent lösen.
QZ-003	Benutzbarkeit - Bedienbarkeit	Das System soll den Mitarbeiter auf der Baustelle unterstützen. Jede Interaktion mit der Applikation muss effizient und einfach erledigt werden können. Systemprozesse sollen den Benutzer nicht blockieren. Wenn das System auf die Antwort eines Prozesses warten MUSS, dann ist dies dem Benutzer transparent darzustellen.

4.3.3 Stakeholder

In der folgenden Tabelle werden die Stakeholder des MVPs aufgelistet und beschrieben.

Tabelle 9 Stakeholder des «Baustellen App» MVPs

Wer	Interesse, Bezug
Mitarbeiter auf der Baustelle	Will schnell und immer auf Projektdaten zugreifen und Notizen sehen und erfassen können. Erwartet Zugriff auf Funktionen und Daten via mobiles Endgerät. <ul style="list-style-type: none"> • Kann alle Projekte betrachten • Kann in allen Projekten Notizen erfassen • Kann Dokumente abfragen • Kann Bilder aufnehmen und in Notizen ablegen
Projektleiter	Muss auf die Projektdaten zugreifen können, sowohl im Backoffice wie auch auf der Baustelle. Will Projekte primär am Desktop administrieren. Will Projektdaten auf der Baustelle mit einem mobilen Endgerät betrachten. <ul style="list-style-type: none"> • Kann Projekte erstellen

	<ul style="list-style-type: none"> • Kann seine Projekte bearbeiten • Kann Notizen erfassen
Sachbearbeiter	<p>Bearbeitet Projektdaten aus dem Backoffice. Verwendet primär einen Desktoprechner.</p> <ul style="list-style-type: none"> • Kann Projekte erstellen • Kann alle Projekte bearbeiten • Kann Notizen erfassen

4.3.4 Randbedingungen

In der folgenden Tabelle sind die relevanten Randbedingungen beschrieben.

Tabelle 10 Randbedingungen

Id	Randbedingung
RB001	Der MVP muss mobile Endgeräte mit Betriebssystem ab iOS Version 16 und Android Version 12 kompatibel sein.
RB002	Für die Umsetzung müssen Microsoft Technologien wie ASP.NET und C# verwendet werden.
RB003	Als Plattform für die «Baustellen App» soll die Public Cloud von Microsoft «Azure» zum Einsatz kommen.

4.3.5 Kontextabgrenzung

Im folgenden Abschnitt wird der Kontext der «Baustellen App» als Blackbox dargestellt und beschrieben.

4.3.5.1 Fachlicher Kontext («Baustellen App» als Blackbox)

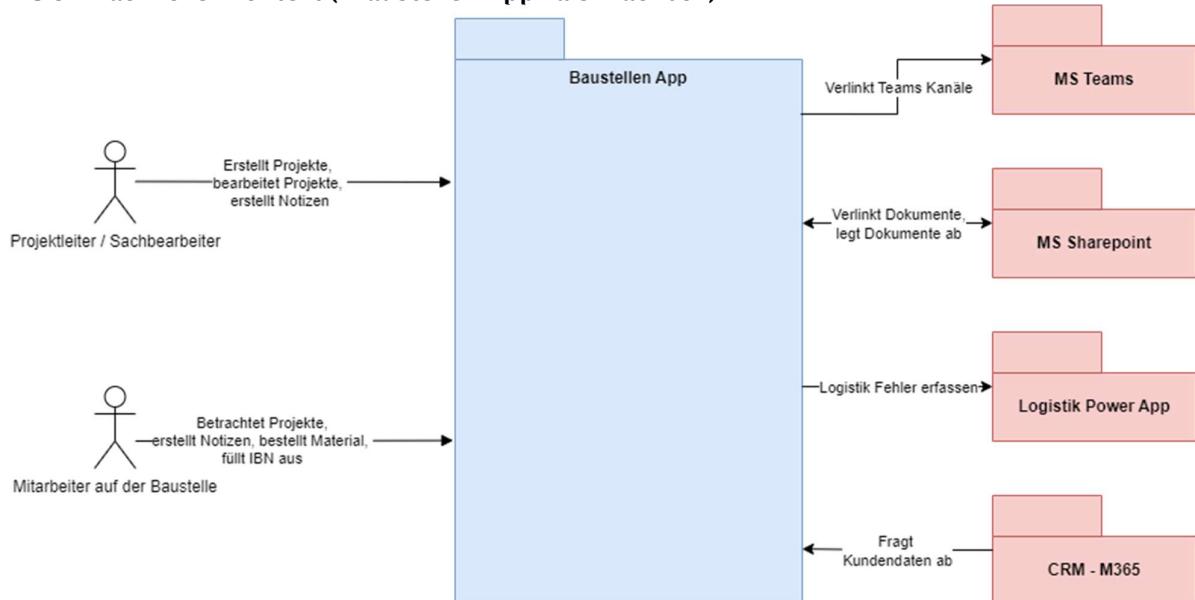


Abbildung 16 Fachlicher Kontext

In der folgenden Tabelle werden die Akteure und externen Systeme beschrieben.

Tabelle 11 Akteure und Systeme im Kontext der «Baustellen App»

Aktor/System	Beschreibung
Projektleiter	Der Projektleiter ist verantwortlich für das Management der Projekte der Helion AG. Er kann die ihm zugewiesenen Projekte bearbeiten.
Sachbearbeiter	Der Sachbearbeiter ist im Backoffice der Helion AG tätig und kann alle Projekte betrachten und bearbeiten. Er mutiert Projektdaten auf Anweisung des Projektleiters und führt im Rahmen der BackOffice Tätigkeit nötige Prozessschritte in der «Baustellen App» aus.
Mitarbeiter auf der Baustelle	Der Mitarbeiter auf der Baustelle ist für die Umsetzung der Projekte auf der Baustelle zuständig. Er erfasst Fehler in der Logistik in der «Logistik Power App» und nimmt Anlagen gemäss Protokoll in Betrieb.
MS Teams	Wird für die Kollaboration von Personal und Kunden verwendet. Für das Projekt wichtige Chats und Kanäle können in der «Baustellen App» verlinkt werden.
MS Sharepoint	Sharepoint wird für die Ablage von Dokumenten verwendet. Die «Baustellen App» verlinkt einerseits auf Dokumente in Sharepoint und legt andererseits Dokumente in Sharepoint ab.
Logistik Power App	In diesem System werden Materialbestellungen für ein Projekt abgewickelt. Ist die Materialbestellung fehlerhaft, kann dies via «Baustellen App» gemeldet werden.
CRM – M365	In diesem System «CRM-M365» werden die Kundendaten der Helion AG verwaltet. Die «Baustellen App» kann Kundendaten direkt aus diesem System beziehen.

4.3.6 Lösungsstrategie

In diesem Abschnitt wird die Lösungsstrategie für die «Baustellen App» aufgezeigt

4.3.6.1 Offline-Verfügbarkeit

Um dem Mitarbeiter auf der Baustelle eine einfache Möglichkeit zu bieten, auf die Projektdaten zuzugreifen, wird eine native, mobile Applikation erstellt. Dieser mobile Client bietet die Möglichkeit, die Applikation offline zu nutzen, Daten zu erfassen sowie Bilder zu speichern. Die Datenmanipulationen werden, sobald die Konnektivität wiederhergestellt wurde, auf dem mobilen Gerät zwischengespeichert und in das Backend synchronisiert. Damit Projektdaten offline verfügbar werden, erhält der Benutzer die Möglichkeit, seine Projekte als «Favoriten» zu markieren. Markierte Projekte werden synchronisiert.

4.3.6.2 Datensynchronisation

Da hier ein Multi-User System mit verteilten Komponenten und offline Fähigkeit realisiert werden soll, wird an die Synchronisation der Daten besondere Anforderungen gestellt. Anhand einer Nutzwertanalyse (Kapitel 4.3.11.4) wurde die geeignete Methode festgestellt. Die gewählte Methode hat die Konsequenz, dass Konflikte anhand der «Last Write Wins» Regel gelöst werden. Diese Entscheidung und mitführende Konsequenz wurden von der Helion AG akzeptiert.

4.3.6.3 Wartbarkeit und Erweiterbarkeit

Die «Baustellen App» wird modular aufgebaut. Die Module sind nach fachlichen Anforderungen getrennt. Ein Modul bündelt zusammenhängende Funktionen. Die Module werden als Microservices aufgebaut. Damit wird die zukünftige Erweiterbarkeit garantiert.

4.3.6.4 Sicherheit

Benutzer müssen sich authentifizieren. Auf Grund der Entscheidung der Helion AG die Azure Cloud als Plattform für die «Baustellen App» zu verwenden, wird für die Authentifizierung Azure Entra ID

eingesetzt. Ein Rollenmodell, welches entlang der fachlichen Akteure aufgebaut ist, gewährt gezielte Rechte für die Anwender.

4.3.6.5 Performance

Um schnelle Antwortzeiten für Anfragen an das Backend zu erreichen, wird das Backend als Microservice Architektur aufgebaut. Die Last wird so auf mehrere Services verteilt, welche einzeln skalierbar sind, wodurch die «Baustellen App» sehr flexibel auf Lastschwankungen reagieren kann.

4.3.6.6 Web Frontend

Für das Management der Projekte wird ein Webportal realisiert. Da diese Arbeit vorwiegend am Desktop geschieht, bietet ein Webportal die bessere Benutzererfahrung für die Administration von Datensätzen als eine mobile App. Das Webportal wird als Single Page Application (SPA) mit dem bewährten Angular Framework realisiert.

4.3.6.7 Mobile App

Für die Anwendung im Feld kommt eine mobile App für iOS und Android zum Einsatz. In diesem Kontext ist Datenverfügbarkeit (auch offline) und Zugriffsgeschwindigkeit von Bedeutung. Die mobile Applikation ist auf diese Qualitätsmerkmale zugeschnitten, damit der Mitarbeiter auf der Baustelle die bestmögliche Effizienz in der Verwendung der «Baustellen App» erhält. Die Mobile Applikation wird mit dem Framework .NET Maui von Microsoft realisiert. Dies ermöglicht den Bau einer mobilen Applikation für iOS und Android aus einer Codebase.

4.3.6.8 Auslieferungsstrategie

Das Backend wird in einer «Public Cloud» auf einer Container Plattform betrieben. Das Webfrontend ist über die gleiche Domäne wie das Backend verfügbar und kann von jedem Rechner mit Verbindung zum Internet aufgerufen werden. Die mobile App soll mit einer Mobile Device Management Lösung (MDM) auf die Geräte der Mitarbeiter ausgeliefert werden.

4.3.7 Bausteinsicht

Im folgenden Kapitel werden die statischen Bausteine in verschiedenen Ebenen dargestellt.

4.3.7.1 Whitebox des Gesamtsystem (Level 0)

Die erste Bausteinsicht wird mit zwei Diagrammen dargestellt.

4.3.7.1.1 Fachlicher Kontext

In Abbildung 17 sind die fachlichen Beziehungen zu den verschiedenen Bausteinen dargestellt.

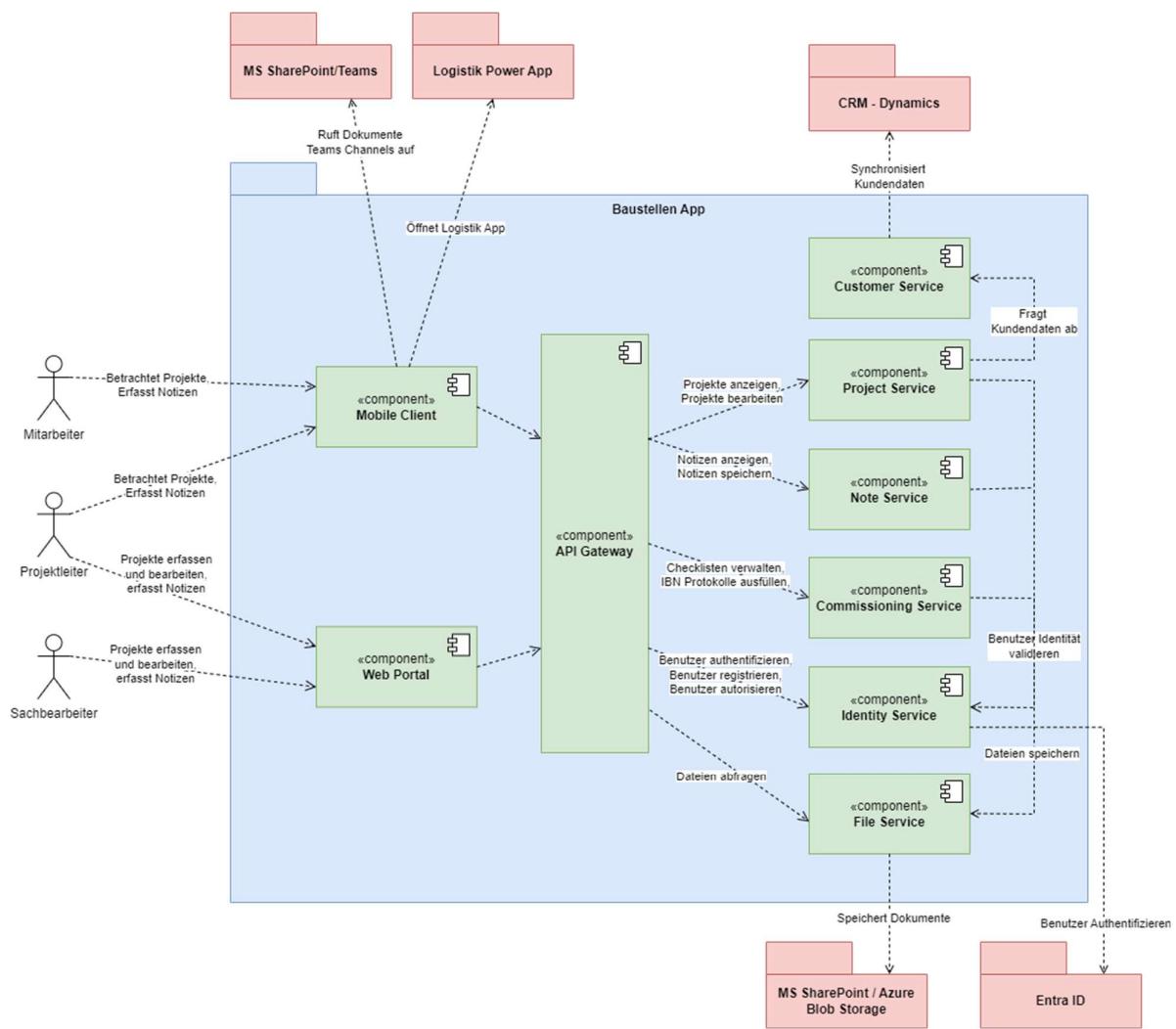


Abbildung 17 Bausteinsicht Level 0 Fachlicher Kontext

4.3.7.1.2 Technischer Kontext

In Abbildung 18 sind die Schnittstellen unter den einzelnen Komponenten dargestellt.

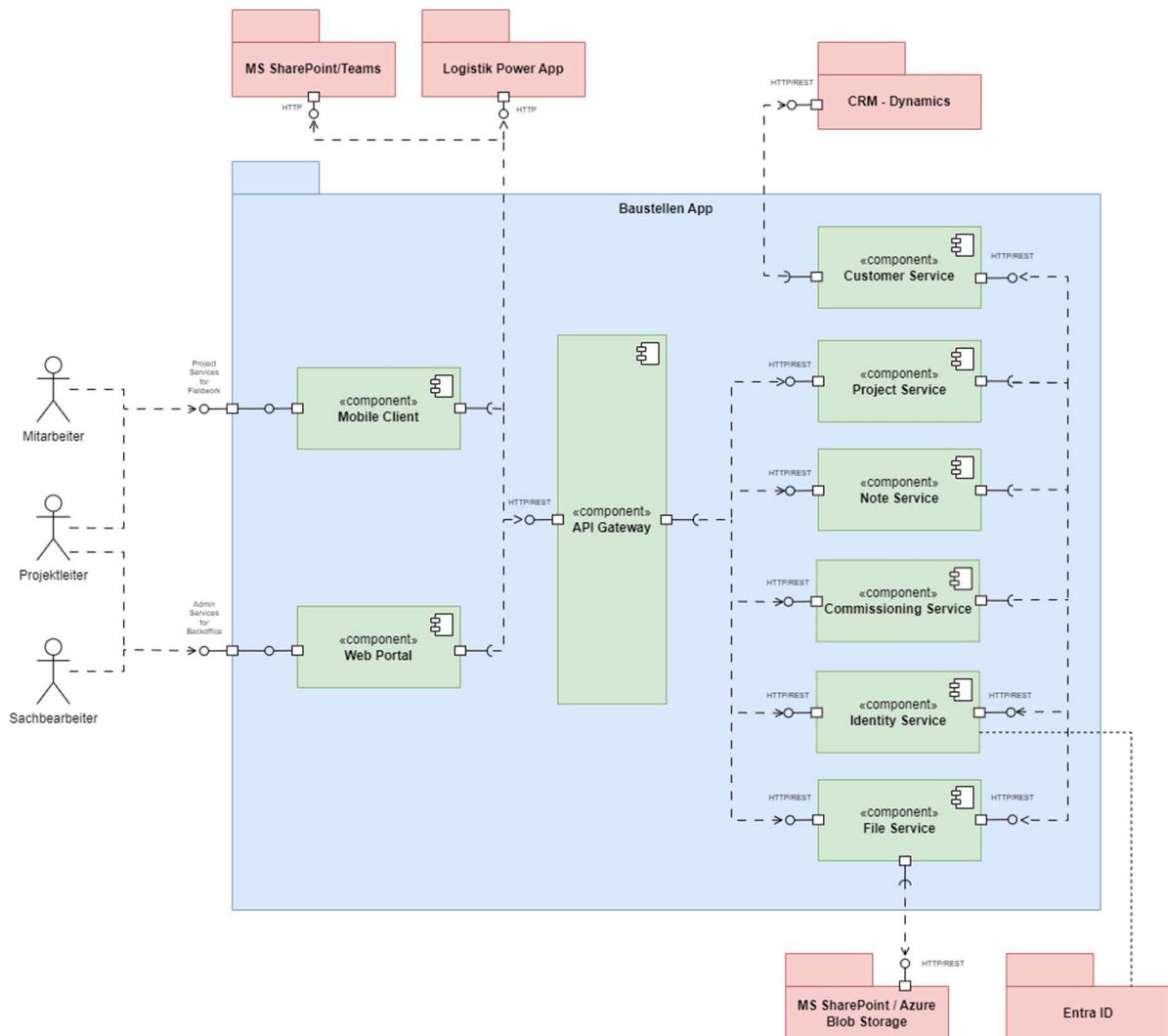


Abbildung 18 Bausteinsicht Level 0 Technischer Kontext

4.3.7.1.3 Begründung

Die «Baustellen App» besteht aus verschiedenen Komponenten, um die Last und Verantwortlichkeiten zu verteilen. Die Business-Logik der gesamten Applikation wird in die Domänen Projekte, Notizen, Inbetriebnahme sowie Identität aufgeteilt und als Microservices aufgebaut. Dadurch lässt sich das System in Zukunft um weitere Domänen erweitern. Schnittstellen zu den Services und dem API-Gateway werden als HTTPS/REST Schnittstellen implementiert. HTTPS/REST ist stabil, bewährt und etabliert. Damit Probleme bezüglich CORS in der Webapplikation verhindert werden können, wird zwischen Services und Web oder Mobile Frontend ein API-Gateway realisiert.

4.3.7.1.4 Enthaltene Bausteine

Tabelle 12 Beschreibung der System Bausteine

Baustein	Beschreibung
Mobile Client	Die mobile App ist der Zugang für Mitarbeiter auf der Baustelle. Der Hauptzweck ist die Informationsbeschaffung auf der Baustelle für die Mitarbeiter und Projektleiter.
Webportal	Im Webportal werden die Projekte administriert. Das Webportal ist für den Zugang von einem Desktop/Laptop Rechner ausgelegt.
API Gateway	Der API Gateway ist ein Reverse Proxy. Er dient als Single Entry Point für das Webportal sowie den mobilen Client und verteilt die Anfragen an die Services im Backend. Außerdem sichert er die Routen, indem er die Weiterleitung für nicht authentifizierte Benutzer blockiert.
Project Service	Der Projekt Service erlaubt die Erfassung und Bearbeitung der Projekte.
Note Service	Der Note Service erlaubt das Erfassen und Bearbeiten von Notizen.
Commissioning Service	Der Inbetriebnahme Service beinhaltet das Erfassen von IBN-Checklisten. Der Service erlaubt das Erstellen von IBN-Protokollen anhand der IBN-Checklisten.
Identity Service	Dieser Service kümmert sich um die Benutzerverwaltung. Benutzer können erfasst und einer Applikationsrolle zugeordnet werden. Dieser Service autorisiert die Benutzer innerhalb der Services.
File Service	Der File Service ist für das Speichern und Ausliefern von Dateien wie Dokumente oder Bilder zuständig.
Customer Service	Der Customer Service synchronisiert Kunden Daten aus dem CRM der Helion AG und stellt sie für die «Baustellen App» zur Verfügung.

4.3.7.1.5 Schnittstellen

In der folgenden Tabelle werden die Schnittstellen zu den externen Systemen beschrieben.

Tabelle 13 Beschreibung der Schnittstellen des Gesamtsystems zu den externen Systemen

Schnittstelle	Risiko	Beschreibung
MS SharePoint/Teams	Für das Projekt relevante Chats und Dokumente können nicht geöffnet werden	Teams und SharePoint sind Standardapplikationen aus dem Hause Microsoft und dienen der Kollaboration (Teams) der Mitarbeiter und Dokumenten Management (SharePoint) im Projekt.
CRM – Dynamics	Aktuelle Kundendaten können nicht in die «Baustellen App» synchronisiert werden	Das CRM enthält die Kundendaten der Helion AG und soll als Masterquelle für Kundendaten in den Projekten eingesetzt werden.
Logistik Power App	Fehler in den Materiallieferungen können nicht erfasst werden.	Die Logistik App ist eine Power App, welche u. a. für Materialbestellungen und die Meldung von Lieferfehlern verwendet wird. Aus dem Projekt soll die Logistik App geöffnet werden können, um Fehler in den Materiallieferungen erfassen zu können.

4.3.7.2 Mobile Applikation (Level 1)

In diesem Abschnitt wird die Komponente «Mobile Client» beschrieben.

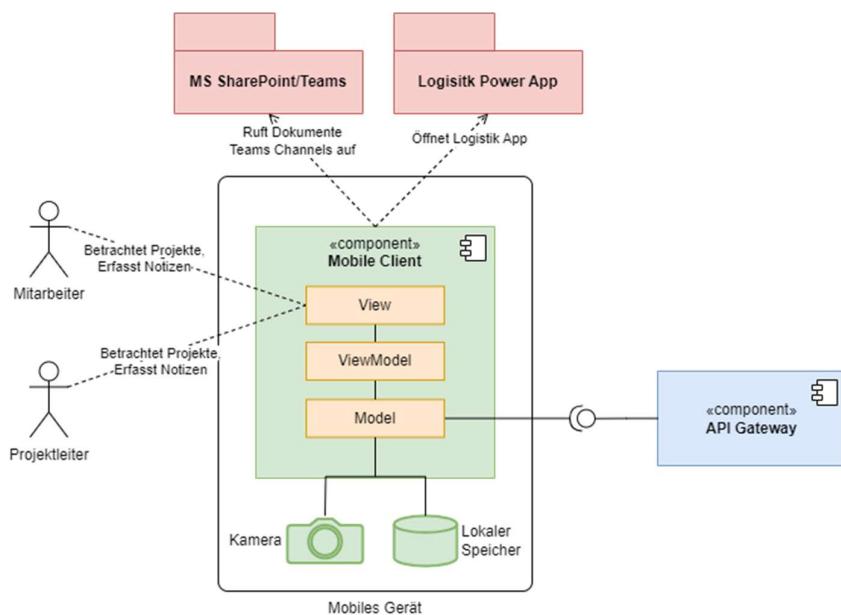


Abbildung 19 Bausteinsicht des «Mobile Client» Level 1

4.3.7.2.1 Zweck/Verantwortung

Die mobile App ist der Client für die Installation auf einem mobilen Gerät. Er dient als Zugang zu den Projekten, Notizen und Inbetriebnahmeprotokollen für die Mitarbeiter auf der Baustelle.

4.3.7.2.2 Schnittstellen

Tabelle 14 Beschreibung der Schnittstellen des «Mobile Client»

Schnittstelle	Beschreibung
MS SharePoint/Teams	Aus der Projektansicht in der «Baustellen App» öffnet der Benutzer ein Dokument (SharePoint) oder Chat (Teams) in der jeweiligen externen App oder in deren Webansicht im Browser. Dokumente oder Chats sind als externe URLs abgelegt.
Logistik Power App	Der Benutzer verwendet die Logistik Power App, um Fehler in der Materialbestellung zu melden. Im Projekt wird eine externe URL auf das Meldeformular in der Logistik Power App hinterlegt. Das Formular öffnet sich in einer externen App oder dem Browser.
API Gateway	Über den API Gateway findet die Kommunikation mit den Backend Services statt.
Lokaler Speicher	Wird zum lokalen Speichern der Daten verwendet, damit die mobile App auch offline verwendet werden kann.
Kamera	Wird verwendet, um Bilder für die IBN-Protokolle oder die Notizen zu machen.

4.3.7.2.3 Qualitäts-/Leistungsmerkmale

- Die mobile App unterstützt Betriebssysteme ab Android 12 und iOS 16.
- Die App ist für die Verwendung im Portraitmodus optimiert.
- Die App speichert ausgewählte Daten lokal, damit die Applikation offline verwendet werden kann.
- Das Software-Design wird nach dem Entwurfsmuster MVVM umgesetzt (Siehe Kapitel 4.3.11.5).

4.3.7.3 Web Portal (Level 1)

In diesem Abschnitt wird die Komponente «Web Portal» beschrieben.

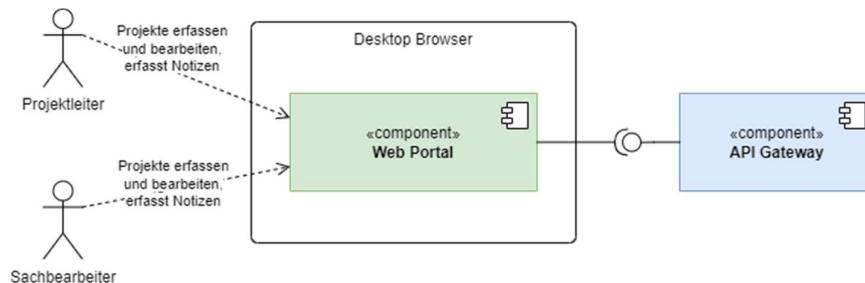


Abbildung 20 Bausteinsicht «Web Portal» Level 1

4.3.7.3.1 Zweck/Verantwortung

Das Web Portal wird zur Administration der Daten durch Sachbearbeiter und Projektleiter verwendet.

4.3.7.3.2 Schnittstellen

Tabelle 15 Beschreibung der Schnittstellen des «Web Portals»

Schnittstelle	Beschreibung
API Gateway	Über den API Gateway findet die Kommunikation mit den Backend Services statt.

4.3.7.3.3 Qualitäts-/Leistungsmerkmale

- Ist für die Verwendung im Desktop Browser optimiert.

4.3.7.4 API Gateway (Level 1)

In diesem Abschnitt wird die Komponente «API Gateway» beschrieben.

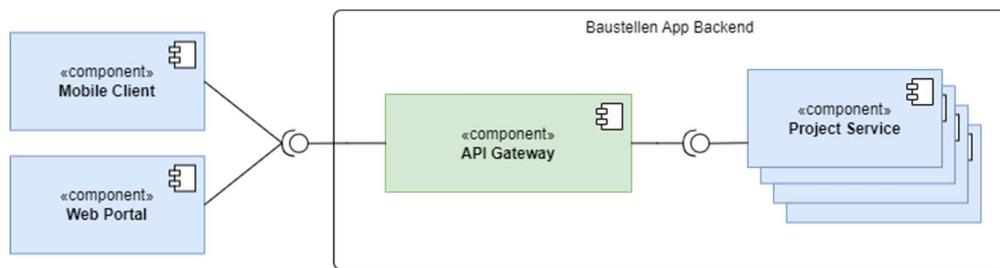


Abbildung 21 Bausteinsicht «API Gateway» Level 1

4.3.7.4.1 Zweck/Verantwortung

Der API Gateway ist für das Routing von Anfragen an das «Baustellen App»-Backend zuständig. Die Komponente ist ein Reverse Proxy, welcher die Anfragen der mobilen App und des Web Portals an die

Microservices im Backend weiterleitet. Die Hauptaufgabe des API Gateways ist die Kapselung der verschiedenen Service Domänen hinter einer öffentlichen Domäne. Der Gateway sichert die Routen zu den Backendservices vor unautorisiertem Zugriff. Zusätzlich ermöglicht der Gateway die Einhaltung der CORS Regeln für das Web Portal.

4.3.7.4.2 Schnittstellen

Tabelle 16 Beschreibung der Schnittstellen des «API Gateways»

Schnittstelle	Beschreibung
Ankommende Anfragen von «Mobile App» und «Web Portal»	Die Schnittstelle für die Clients ist gegenüber dem Internet exponiert und über eine öffentliche Domäne aufrufbar. Der API Gateway nimmt an dieser Schnittstelle die Anfragen der Clients entgegen.
Backend Services	Anhand von URI Pfadregeln leitet der Reverse Proxy die Anfragen über die Backend Schnittstellen an die Backend Services weiter.

4.3.7.4.3 Qualitäts-/Leistungsmerkmale

- Sichert Backend Services vor unautorisiertem Zugriff.
- Kapselt Anfragen an Backend Services für die «Baustellen App» Clients.

4.3.7.5 Customer Service (Level 1)

In diesem Abschnitt wird die Komponente «Customer Service» beschrieben.

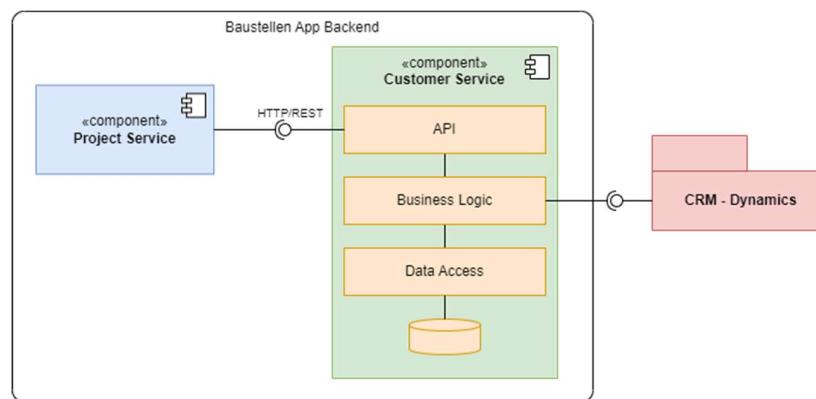


Abbildung 22 Bausteinsicht «Customer Service» Level 1

4.3.7.5.1 Zweck/Verantwortung

Der «Customer Service» ist für die Ablage und Synchronisation von Kundendaten aus dem Backend verantwortlich. Die Kundendaten werden über die API des «Customer Service» verfügbar gemacht.

4.3.7.5.2 Schnittstellen

Tabelle 17 Beschreibung der "Schnittstellen des «Customer Service»

Schnittstelle	Beschreibung
CRM - Dynamics	Schnittstelle zum CRM von Helion AG für die Synchronisation von Kundendaten in die «Baustellen App».

Project Service	Fragt Kundendaten vom «Customer Service» für die Verwendung in einem Projekt ab.
-----------------	--

4.3.7.5.3 Qualitäts-/Leistungsmerkmale

- Importiert und normalisiert Kundendaten für die Verwendung in der «Baustellen App».
- Stellt Kundendaten innerhalb der «Baustellen App» zur Verfügung.

4.3.7.6 Project Service (Level 1)

In diesem Abschnitt wird die Komponente «Project Service» beschrieben.

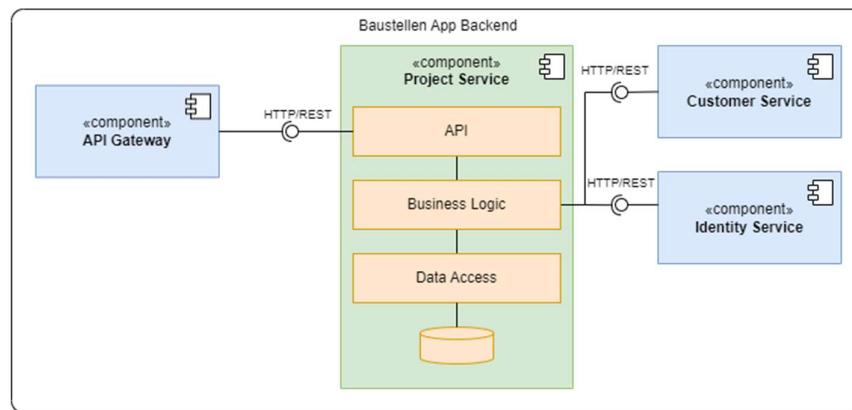


Abbildung 23 Bausteinsicht des «Project Service» Level 1

4.3.7.6.1 Zweck/Verantwortung

Der «Project Service» ist für die Projektdaten verantwortlich. Der Service bietet Standard-Datenmanipulationen für Projekte. Projekte können erstellt, verändert und abgeschlossen werden. Eine Such API erlaubt die Implementierung von Projektsuchen in Clients. Der Project Service bietet ausserdem eine API für die offline Synchronisation an.

4.3.7.6.2 Schnittstellen

Tabelle 18 Beschreibung der «Projekt Service» Schnittstellen

Schnittstelle	Beschreibung
API Gateway	Über diese Schnittstelle werden Anfragen von den Clients an den «Project Service» gestellt.
Customer Service	Der «Project Service» fragt über diese Schnittstelle Kundendaten zu den Projekten ab.
Identity Service	Autorisierter Zugriff auf die API Endpoints des «Project Service» und stellt Benutzerdaten zu den Projekten zur Verfügung.

4.3.7.6.3 Qualitäts-/Leistungsmerkmale

- Der Service ermöglicht die Synchronisation von Projektdaten für die offline Verwendung.
- Aktionen sind autorisiert und der Zugriff auf Projektdaten ist gesichert.

4.3.7.7 Note Service (Level 1)

In diesem Abschnitt wird die Komponente «Note Service» beschrieben.

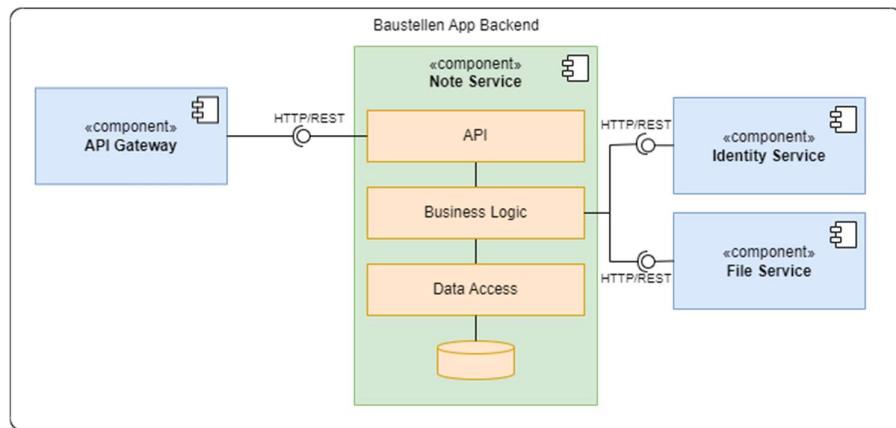


Abbildung 24 Bausteinsicht des «Note Service» Level 1

4.3.7.7.1 Zweck/Verantwortung

Der Note Service ist für die Erstellung und Bearbeitung von Notizen zuständig. Enthält eine Notiz Bilder, werden die Bilddaten an den «File Service» zur Ablage übergeben.

4.3.7.7.2 Schnittstellen

Tabelle 19 Beschreibung der «Note Service» Schnittstellen

Schnittstelle	Beschreibung
Identity Service	Autorisierter Zugriff auf die API Endpoints des «Note Service» und stellt Benutzerdaten zu den Notizen zur Verfügung.
File Service	Schnittstelle für die Ablage von Bildern zu Notizen.
API Gateway	Über diese Schnittstelle werden Anfragen von den Clients an den «Note Service» gestellt.

4.3.7.7.3 Qualitäts-/Leistungsmerkmale

- Zugriff auf die Notizen ist abgesichert.
- Notizen können für die offline Verwendung synchronisiert werden.

4.3.7.8 Commissioning Service

In diesem Abschnitt wird die Komponente «Commissioning Service» beschrieben.

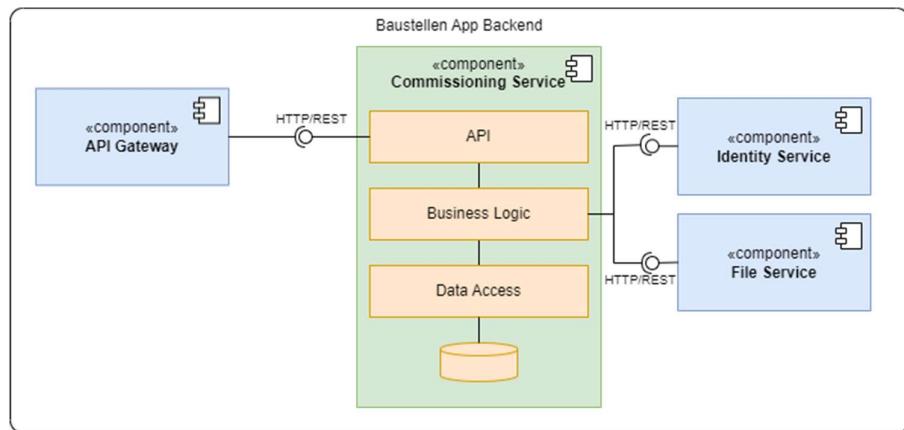


Abbildung 25 Bausteinsicht «Commissioning Service» Level 1

4.3.7.8.1 Zweck/Verantwortung

Der «Commissioning Service» verwaltet die Inbetriebnahmen. Checklisten können als Vorlagen angelegt und einer Inbetriebnahme zugeordnet werden. Wird eine Inbetriebnahme durchgeführt, wird die ausgefüllte Checkliste als Protokoll abgelegt.

4.3.7.8.2 Schnittstellen

Tabelle 20 Beschreibung der «Commissioning Service» Schnittstellen

Schnittstelle	Beschreibung
Identity Service	Autorisierter Zugriff auf die API Endpunkts des «Commissioning Service» und stellt Benutzerdaten für die Abnahmeprotokolle zur Verfügung.
File Service	Schnittstelle für die Ablage der Abnahmeprotokolle.
API Gateway	Über diese Schnittstelle werden Anfragen von den Clients an den «Note Service» gestellt.

4.3.7.8.3 Qualitäts-/Leistungsmerkmale

- Zugriff auf die Inbetriebnahme-Daten sind gesichert.
- Aktionen sind autorisiert.
- Daten für die Inbetriebnahme können für die offline Verwendung synchronisiert werden.

4.3.7.9 Identity Service (Level 1)

In diesem Abschnitt wird die Komponente «Identity Service» beschrieben.

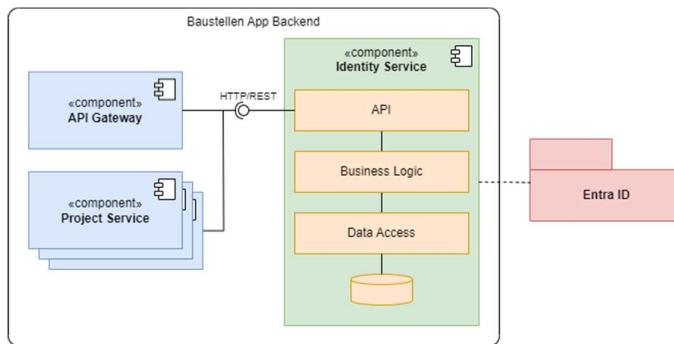


Abbildung 26 Bausteinsicht «Identity Service» Level 1

4.3.7.9.1 Zweck/Verantwortung

Der «Identity Service» stellt Benutzerdaten zur Verfügung und erlaubt die Registrierung von neuen Benutzern. Der Service stellt das Rollenmanagement zur Verfügung und autorisiert Benutzer anhand ihrer Rolle.

4.3.7.9.2 Schnittstellen

Tabelle 21 Beschreibung der «Identity Service» Schnittstellen

Schnittstelle	Beschreibung
API Gateway	Über diese Schnittstelle werden Anfragen von den Clients an den «Note Service» gestellt. Beispielsweise werden Benutzerdaten angefragt oder die Clients triggern die Login Challenge.
Business Services (Projekte, Notizen, IBN)	Buisness Services fragen Benutzerdaten ab und autorisieren Aktionen.
Entra ID	Authentifiziert den Benutzer und gibt ein Access Token zurück.

4.3.7.9.3 Qualitäts-/Leistungsmerkmale

- Kapselt die Authentifizierung in Microsoft Entra ID.

4.3.7.10 File Service (Level 1)

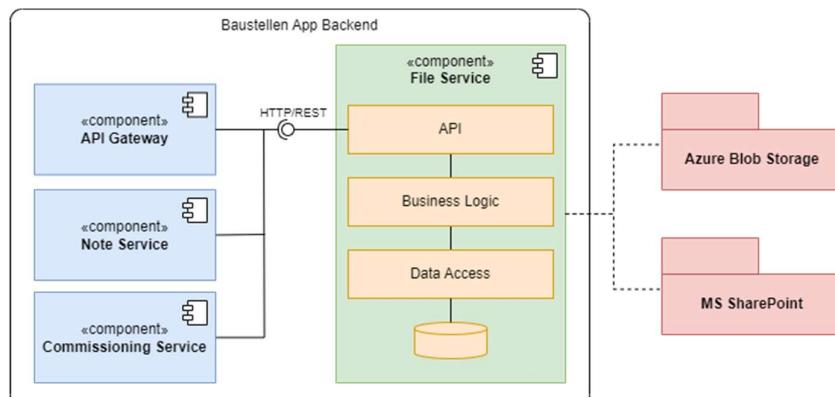


Abbildung 27 Bausteinsicht «File Service» Level 1

4.3.7.10.1 Zweck/Verantwortung

Der «File Service» kapselt den Zugriff auf «MS SharePoint» und «Azure Blob Storage». Der Service erlaubt die Ablage und Abfrage von Dokumenten und Bildern. Bilder werden durch den Service auf dem «Azure Blob Storage» abgelegt. Dokumente speichert der «File Service» in «MS SharePoint».

4.3.7.10.2 Schnittstellen

Tabelle 22 Beschreibung der «File Service» Schnittstellen

Schnittstelle	Beschreibung
API Gateway	Über diese Schnittstelle werden Anfragen von den Clients an den «File Service» gestellt. Primär wird die Schnittstelle genutzt für die Abfrage von Bildern.
Services (Note, Commissioning)	Diese Schnittstelle wird genutzt, um Bilder oder Dokumente abzulegen. Außerdem veranlassen Services die Löschung von Bildern oder Dokumenten über diese Schnittstelle.
Azure Blob Storage	Über diese Schnittstelle werden Bilder gespeichert, abgefragt oder gelöscht.
MS SharePoint	Über einen SharePoint Konnektor speichert der «File Service» Dokumente in SharePoint ab. Über den Konnektor kann der Service Dokumente aus SharePoint abfragen und ausgeben.

4.3.7.10.3 Qualitäts-/Leistungsmerkmale

- Ermöglicht autorisierten Zugriff auf Bilder und Dokumente
- Abstrahiert Zugriff auf physische Ablagen

4.3.8 Laufzeitsicht

In diesem Kapitel werden die Laufzeitsichten für die «Baustellen App»-Aktivitätsszenarien dargestellt. Als erstes wird der Ablauf der Datensynchronisation dargestellt. Diese Sequenz ist Bestandteil von darauffolgenden Laufzeitsichten und wird in diesen als Blackbox dargestellt.

4.3.8.1 Datensynchronisation zwischen Mobile Client und Backend

In diesem Abschnitt wird die Datensynchronisation im Detail beschrieben. Zuerst wird der Vergleichsalgorithmus im Backend und im Frontend aufgezeigt. Danach wird die Sequenz der Datensynchronisation von und zum «Mobilien Client» und den verschiedenen Backend Services dargestellt.

4.3.8.1.1 Ablauf der Datensynchronisation im Mobile Client

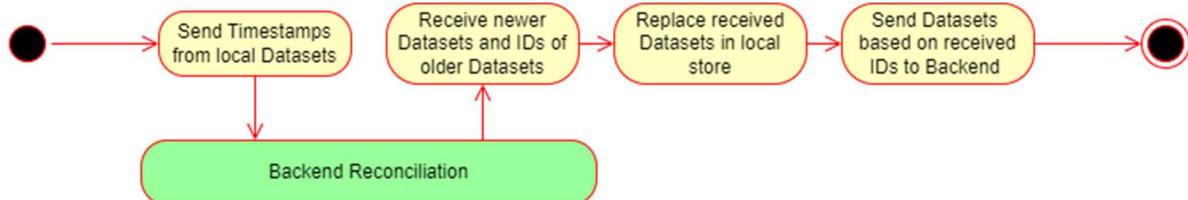


Abbildung 28 Schematischer Ablauf der Datensynchronisation im «Mobilen Client»

Die Abbildung 28 zeigt die Schritte einer Datensynchronisation im «Mobilen Client» auf. Es ist zu beachten, dass nach dem ersten Schritt, dem Senden der Zeitstempel der lokalen Datensätze, die Sequenz auf die Antwort vom involvierten Backend Service wartet. Dieser wird im folgenden Abschnitt 4.3.8.1.2 beschrieben. Der «Mobile Client» erwartet in der Antwort aus dem Backend eine Liste mit aktuelleren Datensätzen und eine Liste mit IDs zu älteren Datensätzen. Die empfangenen Datensätze

überschreiben die lokal gespeicherten Datensätze. Anschliessend sendet der «Mobile Client» alle Datensätze, zu denen eine ID als Antwort vom Backend erhalten wurde, aus dem lokalen Speicher an das Backend.

4.3.8.1.2 Ablauf der Zusammenführung der Datensätze im Backend

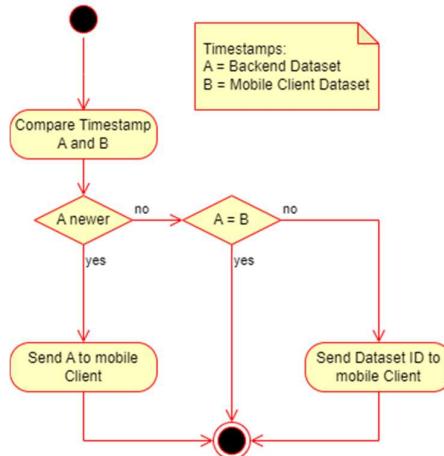


Abbildung 29 Schematischer Ablauf der Datensynchronisation in den Backend Services

In der Abbildung 29 ist zu sehen wie die Zusammenführung der Datensätze, anhand des Zeitstempels, im Backend behandelt wird. Die Antwort an den «Mobilen Client» enthält eine Liste mit den Datensätzen, welche im Backend einen aktuelleren Zeitstempel hatten und eine Liste mit IDs von Datensätzen, zu denen der Zeitstempel aus dem «Mobilen Client» aktueller war.

4.3.8.1.3 Synchronisation der supportenden Services⁴⁵

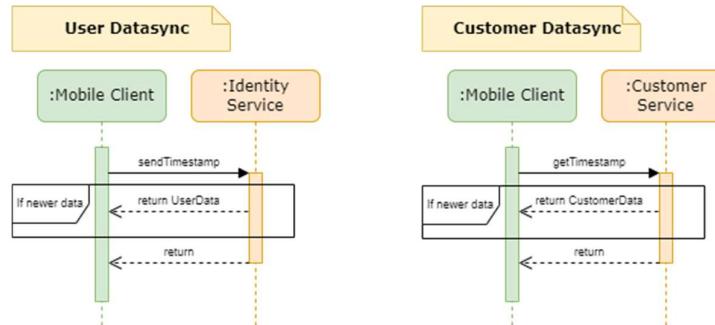


Abbildung 30 Laufzeitansicht zur Datensynchronisation der support backend Services

Daten aus «Identity Service» und «Customer Service» werden bei der Erstellung und Bearbeitung von Projekten benötigt. Auf dem «Mobilen Client» müssen deshalb die Datensets beider Services als Ganzes synchronisiert werden. Der Zeitstempel gilt deshalb in beiden Services für das gesamte Dataset.

⁴ <https://github.com/zuercheram/bauestellen-app/blob/dev/src/Client/Services/SyncingService.cs>

⁵ <https://github.com/zuercheram/bauestellen-app/blob/dev/src/Identity.Api/Program.cs>

4.3.8.1.4 Synchronisation von Projektdaten⁶⁷

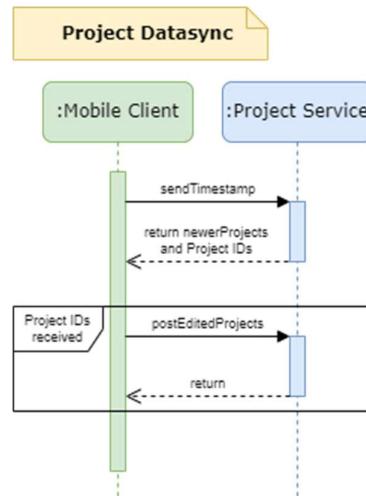


Abbildung 31 Laufzeitansicht zur Datensynchronisation Mobile Client zu Project Service

Wie in Abschnitt 4.3.8.1.1 und 4.3.8.1.2 ausgeführt, werden Projektdatensätze anhand ihres Zeitstempels, des Zeitpunkts der letzten Manipulation, zusammengeführt. Abbildung 31 zeigt den Ablauf der Synchronisation von Projektdaten auf.

4.3.8.1.5 Synchronisation von Notizdaten

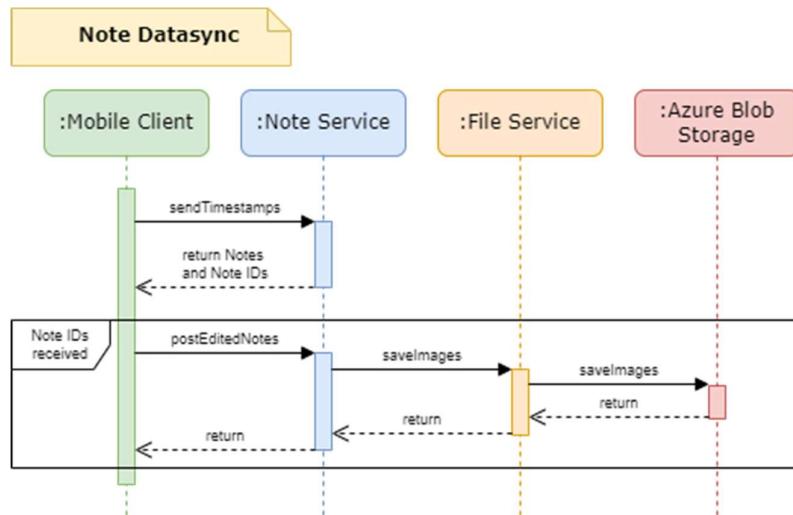


Abbildung 32 Laufzeitansicht zur Datensynchronisation von «Mobile Client» zu «Note Service»

Wie in Abschnitt 4.3.8.1.1 und 4.3.8.1.2 ausgeführt, werden Notizdatensätze anhand ihres Zeitstempels, des Zeitpunkts der letzten Manipulation, zusammengeführt. Enthält ein Notizdatensatz Bilder, werden diese mittels «File Service» im Azure Blob Storage abgelegt.

⁶ <https://github.com/zuercheram/bauestellen-app/blob/dev/src/Client/Services/SyncingService.cs>

⁷ <https://github.com/zuercheram/bauestellen-app/blob/dev/src/Projects.Api/Services/ProjectService.cs>

4.3.8.1.6 Synchronisation von Inbetriebnahmedaten

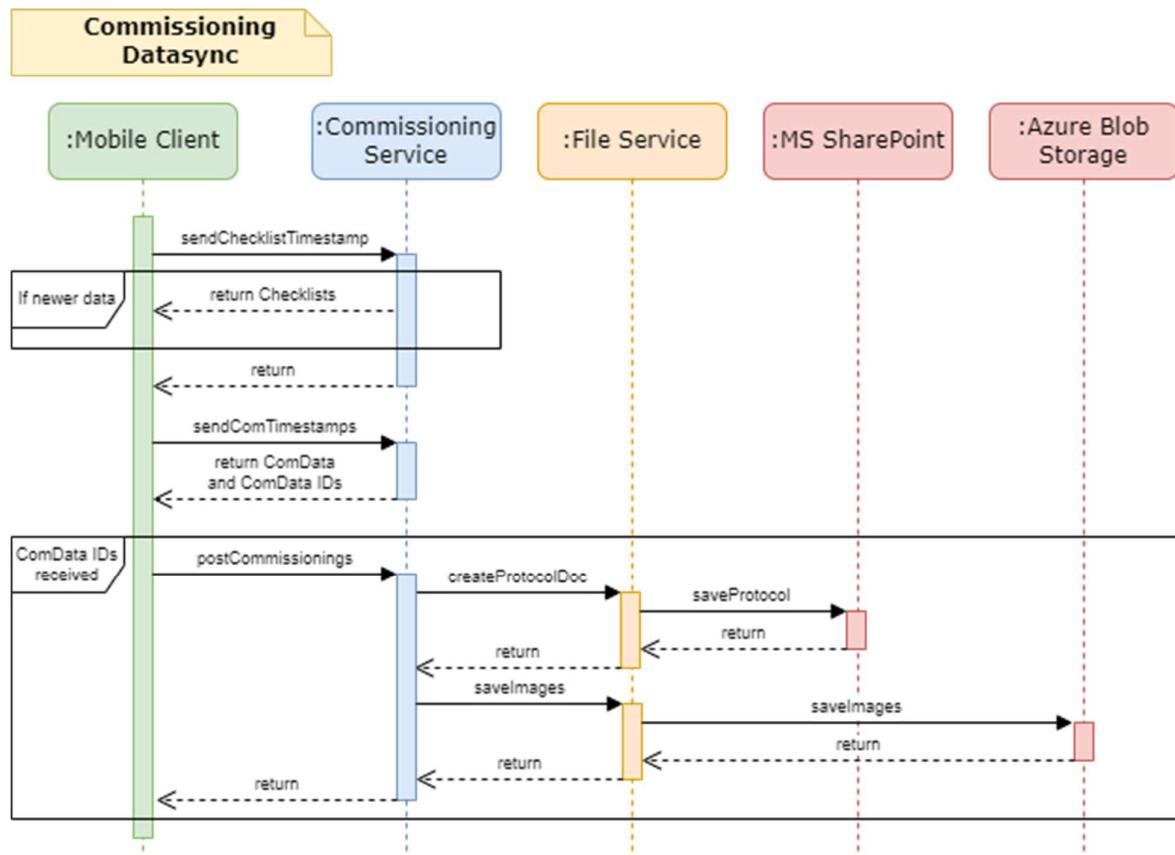


Abbildung 33 Laufzeitansicht zur Datensynchronisation von «Mobile Client» zu «Commissioning Service»

Die Inbetriebnahme Daten bestehen aus zwei Datensatztypen. Die Vorlagen für die IBN Checklisten werden wie die Benutzer und Kundendaten als komplettes Datenset in den «Mobile Client» geladen. Die Inbetriebnahme Protokolle mit Bildern und Checklisten werden als individuelle Datensätze, wie in Abschnitt 4.3.8.1.1 und 4.3.8.1.2 beschrieben, synchronisiert. Die Inbetriebnahme Datensätze werden als Protokolle via «File Service» in SharePoint abgelegt. Enthält ein IBN Datensatz Bilder, werden diese ebenfalls via «File Service» im «Azure Blob Storage» abgelegt.

4.3.8.2 Projektadministration über das Web Portal

In diesem Abschnitt geht es darum, die Abläufe von Projektadministrationsszenarios aus dem Webportal heraus aufzuzeigen. Aus dem «Web Portal» werden Datenänderungen über Standard CRUD Operationen durchgeführt. Da das «Web Portal» für die Verwendung in einem Desktop Browser vorgesehen ist, wird davon ausgegangen, dass zu jeder Zeit Konnektivität von «Web Portal» zum Backend besteht.

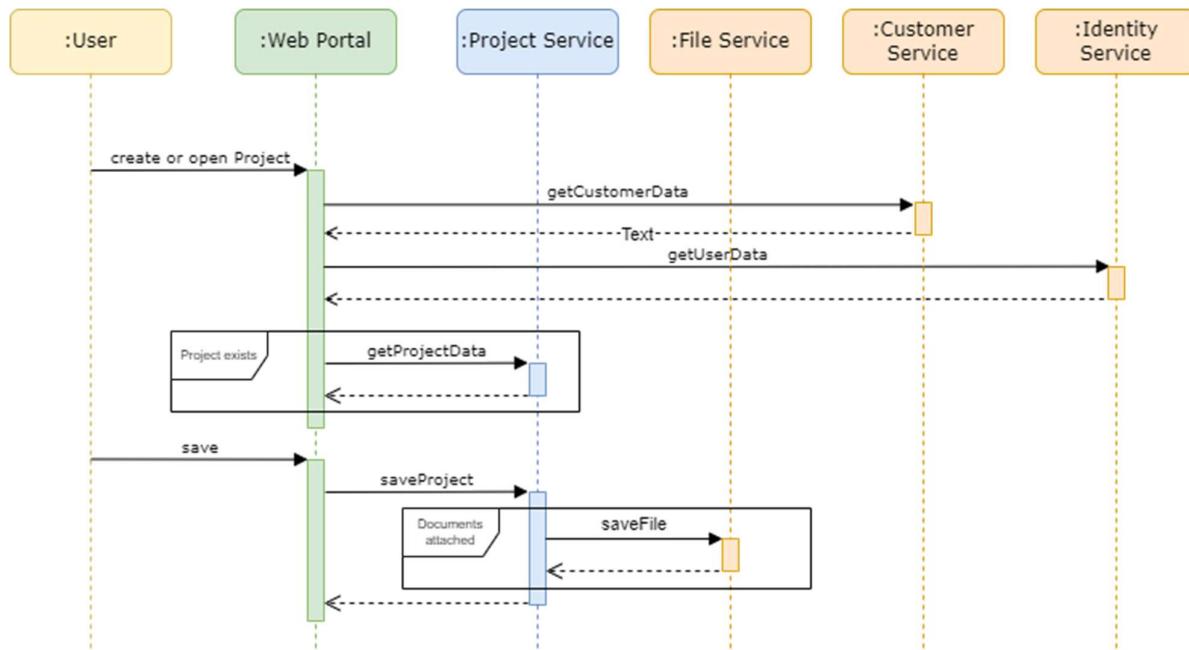


Abbildung 34 Laufzeitansicht zur Projektadministration im «Web Portal»

4.3.8.3 Notizen erfassen im Web Portal

Im «Web Portal» können Notizen erfasst und mit Bildern ergänzt werden. Im folgenden Laufzeitdiagramm ist der Ablauf für das Erfassen einer Notiz im «Web Portal» aufgezeigt.

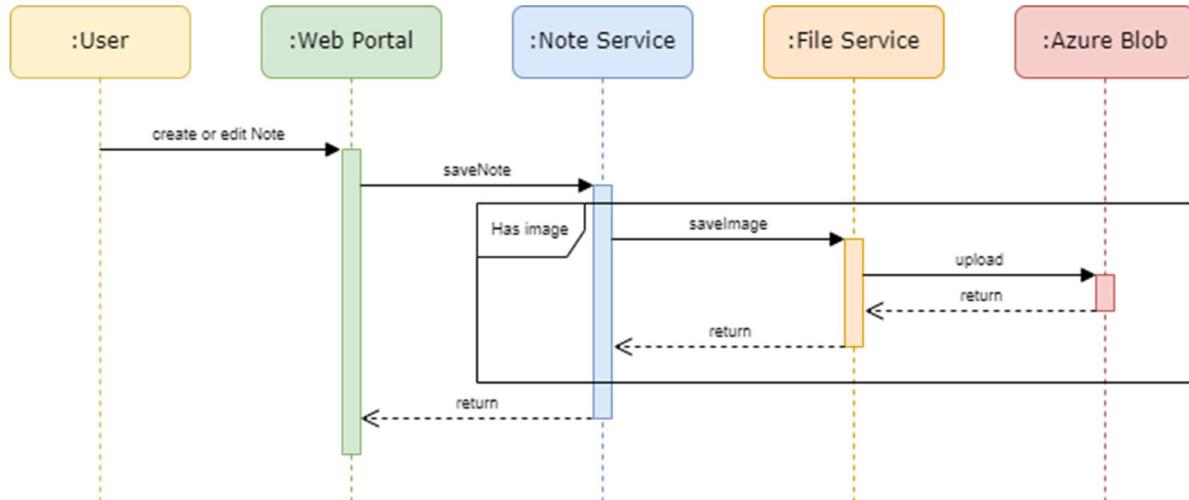


Abbildung 35 Laufzeitsicht zur Erfassung und Bearbeitung von Notizen im Web Portal

4.3.8.4 Inbetriebnahme Checklisten administrieren im Web Portal

Bezüglich Inbetriebnahme wird das «Web Portal» für die Administrierung von Checklisten für die Inbetriebnahmeprotokolle verwendet.

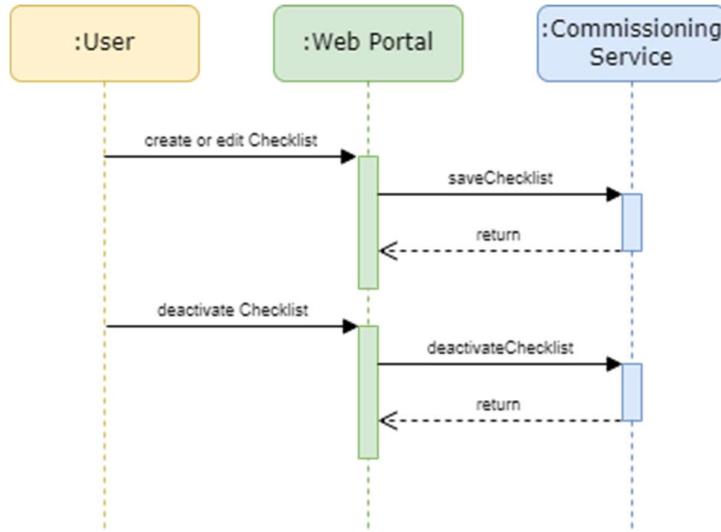


Abbildung 36 Laufzeitsicht zur Administration von Checklisten

4.3.9 Verteilungssicht

In diesem Kapitel wird die Infrastruktur für die «Baustellen App» beschrieben.

4.3.9.1 Infrastruktur Ebene 1

Die «Baustellen App» setzt auf Plattform-As-A-Service (PaaS) in der Azure Cloud. Zum Einsatz kommt ein «Container App Environment». Die «Container App» Plattform ist eine Abstraktion der Kubernetes Plattform (K8s). Microsoft beschreibt «Container App» als serverlose Plattform für Microservice Applikationen. «Container App» übernimmt die Konfiguration und Orchestrierung der Container [15].

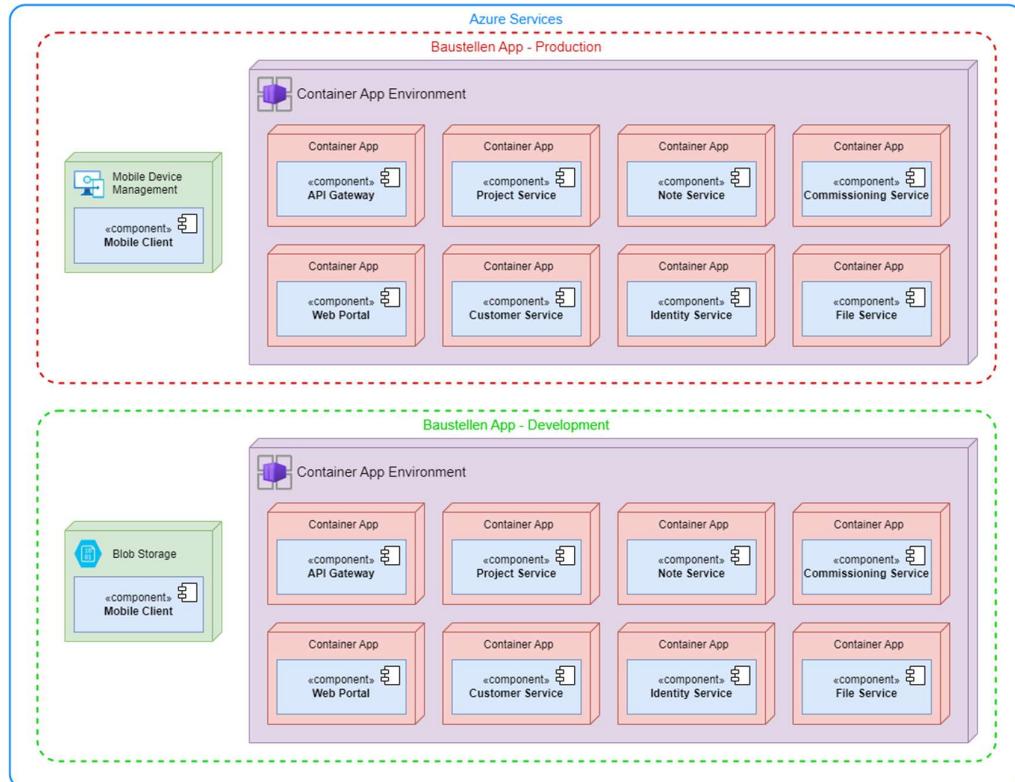


Abbildung 37 Verteilung der Anwendungskomponenten auf Infrastrukturkomponenten

Es werden zwei strukturell gleiche Umgebungen aufgebaut. Eine Produktion für die Anwendung im Tagesgeschäft der Helion AG. Die zweite Umgebung dient der Integration und Testing von neuen Funktionen. In der Abbildung 37 fällt ein Unterschied zwischen Produktion und Development für die Verteilung des «Mobilen Clients» auf: In der Produktion wird die App via «Mobile Device Management» verteilt, während für die Development Umgebung das Anwendungsartefakt für den «Mobile Client» in einem Blob Storage abgelegt wird. Testanwender laden und installieren die Applikation aus dem Blob Storage.

4.3.9.2 Infrastruktur Ebene 2

In diesem Abschnitt werden einzelne Infrastrukturkomponenten und die Kommunikation dazwischen dargestellt. In der Abbildung 38 sind zwei «Ressource Groups» (RG) zu erkennen.

Infrastrukturkomponenten für die Produktivumgebung (Production) und Entwicklungsumgebung (Development) werden in diesen «Ressource Groups» zusammengefasst. Dargestellt ist nur die RG für die Entwicklungsumgebung (d1). Die Produktion ist nur angedeutet anhand der «baupp-p1-resource-group». Die Ressourcen innerhalb der RG sind für beide Umgebungen genau gleich.

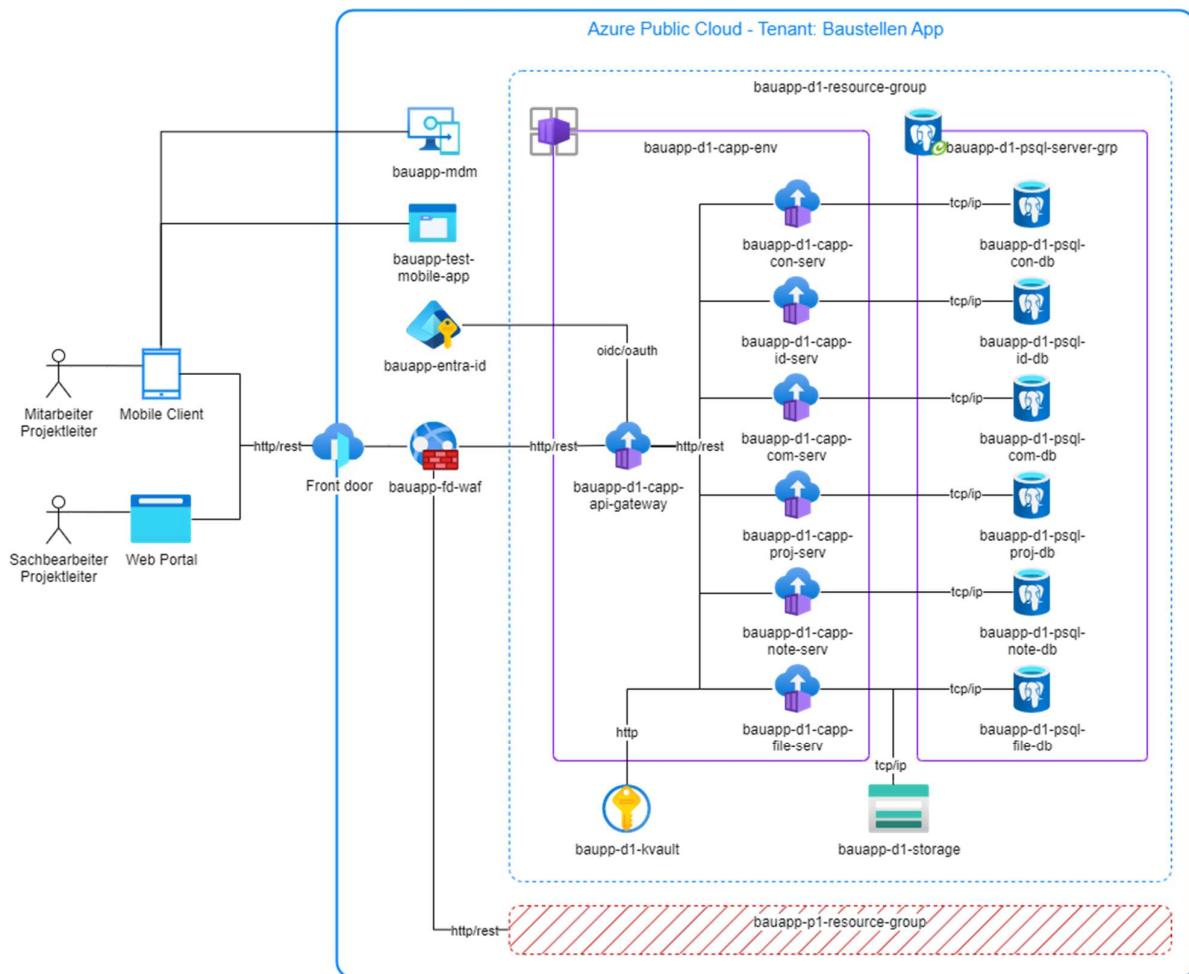


Abbildung 38 Netzwerkansicht über die Entwicklungsumgebung der «Baustellen App»

4.3.10 Querschnittliche Konzepte

In diesem Kapitel werden die wichtigsten, querschnittlichen Konzepte der «Baustellen App» beschrieben.

4.3.10.1 Sicherheit

Dieser Abschnitt beschreibt die Sicherheitskonzepte für die «Baustellen App».

4.3.10.1.1 Benutzer authentifizieren

Benutzer müssen sich in der «Baustellen App» authentifizieren, damit ihnen Zugang zu der Applikation gewährt wird. Die Identität der Benutzer wird durch den Dienst «Entra ID» von Microsoft verwaltet. Benutzer werden in Entra ID registriert. Danach authentifizieren sie sich gegenüber «Entra ID». Die eingesetzte Technologie für die Authentifizierung ist «OpenId Connect», welche auf «OAuth 2.0» basiert [16]. Authentifizierte Benutzer erhalten ein «JWT Token». Dieses Token wird in den Clients gespeichert. Anfragen an das Backend enthalten dieses Token im «http header». Die Backend Services werten dieses Token aus und überprüfen dessen Gültigkeit gegenüber «Entra ID». Dadurch wird sichergestellt, dass der Zugriff auf das Backend nur von authentifizierten Benutzern ausgeführt werden kann.

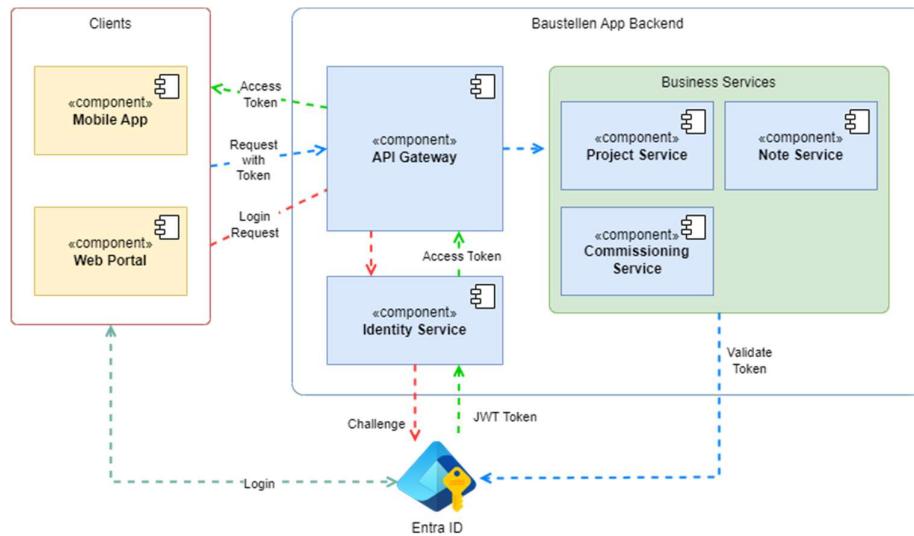


Abbildung 39 Authentifizierung mit Azure Entra ID

In der Abbildung 39 wird dargestellt, wie die Komponenten für die Authentifizierung zusammenarbeiten. In Abbildung 40 ist der Ablauf der Authentifizierung und der Token Validierung dargestellt.

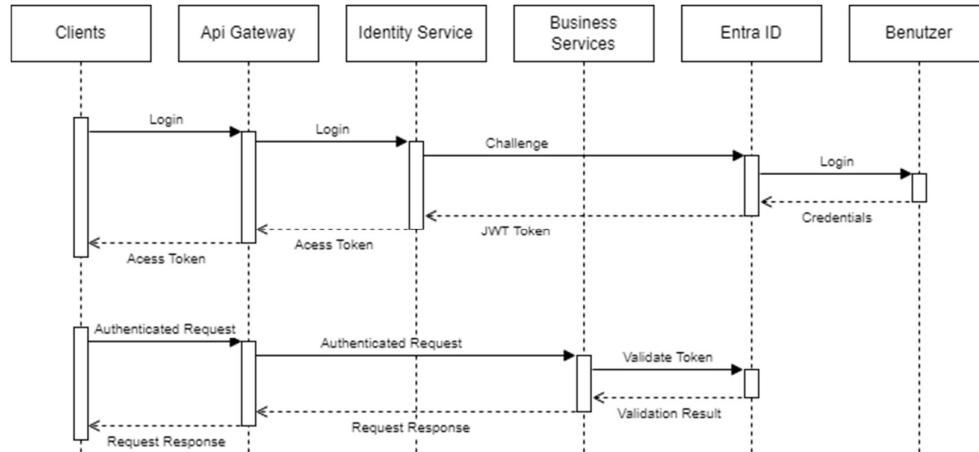


Abbildung 40 Ablauf der Authentifizierung und Token Validierung

4.3.10.2 Deployment

Die «Baustellen App» hat drei verschiedene Deployment-Ziele.

- Mobile App
- Web Portal
- Backend Services

Zuständig für das Deployment sind GitHub Workflows. Damit lässt sich das Deployment automatisieren und in Pipelines aufbauen. Entsprechend sind drei Pipelines für jeweils ein Deployment-Ziel geplant. Für das Deployment wird «Continuous Integration / Continuous Deployment» (CI/CD) als Strategie eingesetzt. Diese Strategie ermöglicht unterbruchfreies Integrieren und Ausrollen der Services.

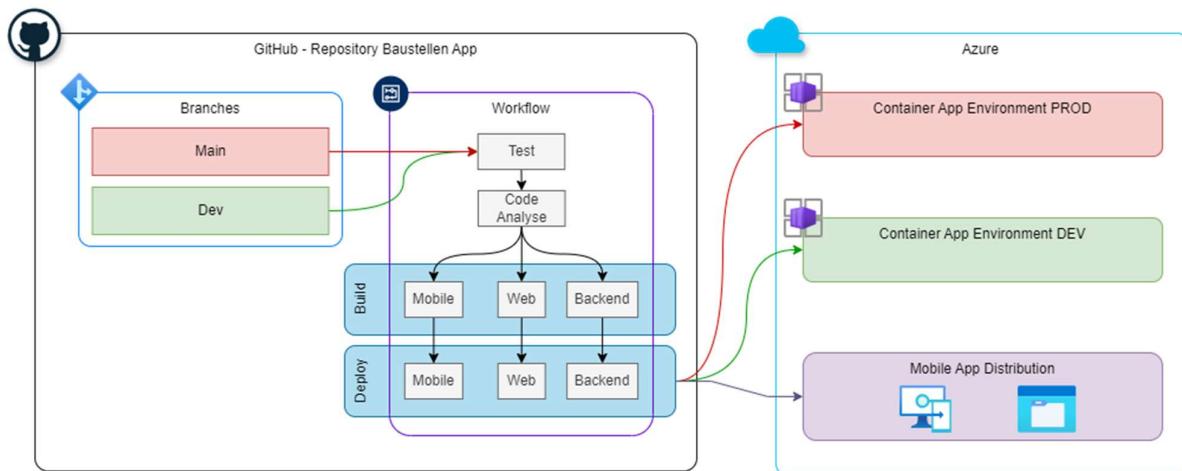


Abbildung 41 Darstellung des Build und Deployment Flows

In der Abbildung 41 ist dargestellt, wie der Quellcode der «Baustellen App» verwaltet, kompiliert und verteilt wird. Die CI/CD Pipeline wird mit GitHub Workflows realisiert. Im ersten Schritt werden Komponenten-Tests, Automatisierte UI-tests und Integrations-Tests ausgeführt. Anschliessend wird der Code durch ein Code Analyse Tool analysiert. Werden diese Qualitätsschritte erfolgreich abgeschlossen, werden für den «Mobile Client», das «Web Portal» und alle Backend Services je ein Build ausgeführt. Die Build Artefakte werden in der Deployment Pipeline auf die jeweiligen Plattformen verteilt. Backend und «Web Portal» werden als Container in das «Container App Environment» auf Azure kopiert. Für die Verteilung auf den mobilen Geräten wird der «Mobile Client» in die Mobile Device Management Anwendung gepusht. Die Umgebungen sind zweistufig aufgebaut. Eine

Development Umgebung (DEV) für das Testen von neuen Integrationen und eine Produktive Umgebung (PROD) für den Live-Einsatz.

4.3.10.3 Qualitätssicherung

In diesem Abschnitt wird die geplante Qualitätssicherung von der Entwicklung bis zur Integration in die Produktive Umgebung beschrieben.

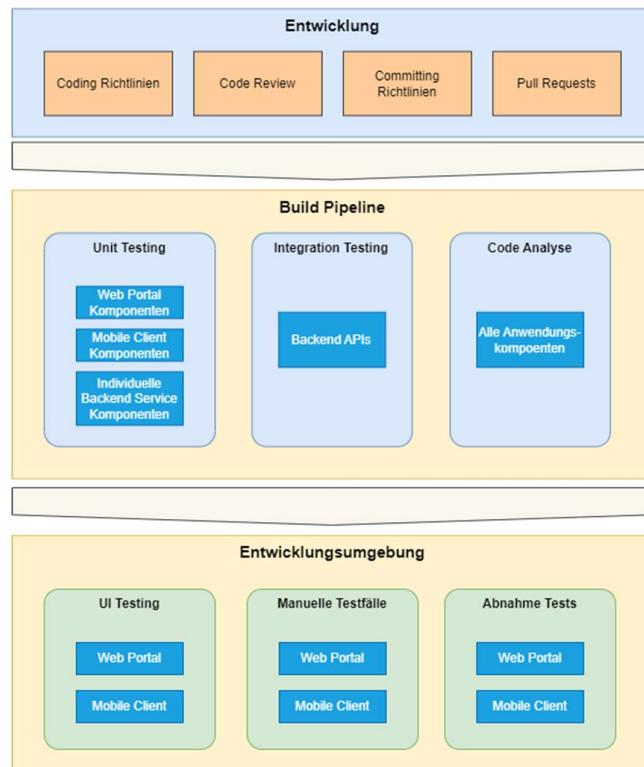


Abbildung 42 Darstellung der Qualitätssicherungslevels mit den betroffenen Komponenten

Die Qualitätssicherung wird für die «Baustellen App» in mehrere Levels aufgeilt. Nachfolgend werden die einzelnen Levels beschrieben.

4.3.10.3.1 Stufe Entwicklung

Tabelle 23 Qualitätsmaßnahmen während der Entwicklung

Massnahme	Beschreibung
Coding Richtlinien	Das Ziel dieser Massnahme ist eine Vereinheitlichung des Quellcodes. Damit wird die Lesbarkeit des Codes und damit die Wartbarkeit erhöht. Die Coding Richtlinien werden im Entwickler Team erarbeitet, verabschiedet und niedergeschrieben. Es werden grundsätzlich Best Practices und die programmiersprachspezifische Standardformatierung angewendet.
Code Review	Während der Entwicklung gilt das 4-Augen Prinzip. Entwickler reviewen ihren Code gegenseitig. Sie honorieren kluge und saubere Lösungen und kritisieren Verletzungen der Coding Richtlinien. Das Code Review soll auch technische Fehler oder mögliche Bugs aufdecken.
Committing Richtlinien	Die Committing Richtlinien sind Vorgaben, die sicherstellen sollen, dass Änderungen am Code in einem Versionsverwaltungssystem, wie zum Beispiel Git, kontrolliert und von hoher Qualität sind. Es ist eine

	Massnahme zur Qualitätssteigerung und verhindern eine Übermittlung von unkomplizierbarem Code in das remote Repository
Pull Requests	Wie in Abbildung 41 dargestellt. Gibt es ein Main und ein Dev Branch, welche die entsprechende CI/CD Pipeline versorgen. Für die Implementierung von neuen Funktionen, wird auf sogenannten «Feature Branches» gearbeitet. Diese Branches können nur via Pull Request (PR) in den Dev Branch zusammengeführt werden. Damit der Pull Request ausgeführt werden kann, muss ein anderer Entwickler den Code Testen, Reviewen und den PR bestätigen. Damit wird sichergestellt, dass der Code auf Dev immer lauffähig ist, sowohl technisch als auch fachlich.

4.3.10.3.2 Stufe «Build Pipeline»

Auf dieser Ebene werden drei Massnahmen für die Qualitätssicherung beschrieben. Das erfolgreiche Passieren dieser Massnahmen ist Bedingung für die weitere Ausführung der Pipeline.

Tabelle 24 Qualitätsmassnahmen auf der Stufe Build Pipeline

Massnahme	Beschreibung
Unit Testing	<p>Unit Tests werden während der Entwicklung implementiert. Algorithmen und Business Logik Bausteine werden mittels Unit Tests auf die korrekte Funktion getestet. Unit Tests kommen im «Web Portal», im «Mobile Client» und für die «Backend Services» zum Einsatz. Die Unit Tests sollen funktionskritische Logik testen. Es wird keine spezifische, prozentuale Testabdeckung im Code angestrebt. Folgende Technologien kommen zum Einsatz:</p> <ul style="list-style-type: none"> • C# code wird mittels NUnit getestet (Backend und Mobile Client) • Angular App/TypeScript wird mit Karma/Jasmine getestet (Web Portal)
Integration Testing	<p>Auf der Ebene der Backend APIs werden Integrationstests verwendet. Damit wird der Service von der API über alle Ebenen bis zur Datenbank getestet. Schnittstellen zu externen Systemen werden während des Testings als Mocks implementiert. Integrationstests werden für die folgenden Services angewendet:</p> <ul style="list-style-type: none"> • Identity Service • File Service • Customer Service • Project Service • Note Service • Commissioning Service <p>Für das Integrationstesting wird während des Testsetups ein WebServer und eine InMemory Datenbank hochgefahren. Via HTTP Client werden API-Anfragen an die temporäre Umgebung abgesetzt und die Antwort der API ausgewertet. Diese Testumgebung ist volatil und wird nach Abschluss des Tests wieder abgebaut.</p>
Code Analyse	<p>Für die Code Analyse kommt ein externes Analyse Tool zum Einsatz. Verwendet wird der kostenpflichtige Service von SonarCloud. SonarCloud bietet gründliche Code Analysen für alle gängigen Programmiersprachen. Die Analyse Regeln werden laufend erweitert und verbessert. Deshalb ist SonarCloud ein verlässliches Werkzeug für</p>

	die Aufdeckung von Fehlern oder unsauberer Programmierung im Code.
--	--

4.3.10.3.3 Stufe Entwicklungsumgebung

Wurde die Anwendung in die Entwicklungsumgebung ausgerollt, werden folgende Qualitätsmaßnahmen durchgeführt, bevor auf die Produktive Umgebung ausgerollt werden kann.

Tabelle 25 Beschreibung der Massnahmen auf der Entwicklungsumgebung

Massnahme	Beschreibung
UI Testing	<p>Die Benutzeroberflächen müssen in zwei Komponenten aufgeteilt werden.</p> <p>Web Portal: Das «Web Portal» wird mittels Selenium automatisch getestet. Selenium emuliert einen Browser. Die Tests in Selenium werden mittels Code programmiert. Im Fall der «Baustellen App» wird dies C# sein. Der Code beschreibt, was Selenium mit der zu testenden App tun soll. Beispielsweise gehe auf Adresse XY und klicke auf den Button mit der ID «html-element-id». Danach wird das Ergebnis mit einem erwarteten Ergebnis verglichen.</p> <p>Mobile Client: Für das Automatisierte Testing des UI des Mobile Clients gibt es spezialisierte Anbieter von Testumgebungen mit realen Geräten. Diese Umgebungen sind sehr kostspielig. Deshalb wird in Absprache mit dem Kunden vorerst auf die automatischen UI Testing für den «Mobile Client» verzichtet. Als alternative kommen manuelle Testfälle zum Einsatz.</p>
Manuelle Testfälle	<p>Manuelle Testfälle werden primär für das Testing des «Mobilen Clients» eingesetzt. Die Testfälle werden schriftlich festgehalten. Jeder Testfall ist wie ein Drehbuch und beschreibt:</p> <ul style="list-style-type: none"> • Wie der Test vorbereitet werden muss • Schritt für Schritt wie der Test durchgeführt werden muss • Welches Ergebnis nach der Durchführung des Tests erwartet wird. <p>Ein manueller Test soll eine isolierte Funktion der Anwendung testen.</p>
Abnahme Tests	Abnahmetests sind erweiterte manuelle Testfälle, welche umfangreichere Szenarien testen sollen. Der Kunde ist für die Durchführung der Abnahmetests verantwortlich. Er Verifiziert das Ergebnis und gibt auf Basis der Abnahmetests das GO für das Deployment auf die Produktive Umgebung.

4.3.11 Architekturentscheidungen

Im folgenden Abschnitt werden die wichtigsten Architekturentscheidungen für die Umsetzung der «Baustellen App» dokumentiert.

4.3.11.1 Aufteilung des Backends in «Micro-Services»

Tabelle 26 Entwurfsentscheidung: Aufbau des Backends als Micro Service Architecture

ID	ADR-001
Titel	Aufbau des Backends als Micro Service Architecture (MSA)
Kontext	Für den Aufbau des Backends stehen folgende Architekturen zur Auswahl:

	<p>1. Micro Service Architecture Eine MSA hat einige wichtige Charakteristiken. MSA gilt als gut Wart- und Erweiterbar, weil die Fachliche Logik der Anwendung auf voneinander unabhängige Services verteilt wird [17]. Insbesondere für komplexere Systeme eignet sich eine MSA [18] und auf längere Sicht sollen auch die Kosten für eine MSA geringer ausfallen [18].</p> <p>2. Monolith Eine monolithische Architektur zeichnet sich dadurch aus, dass die Funktionen der Anwendung in einem einzigen ausführbaren Software System gekapselt sind und dieses System aus einer gemeinsamen Codebasis besteht [18]. Für ein simples Software System bietet sich diese Architektur an, da die Time-To-Market für ein einfaches Software System generell kurz ist [18].</p> <p>3. Serverless Architecture Eine serverlose Architektur macht sich das PaaS [19] Angebot von Cloud Anbietern zu nutze. Eine serverless Architecture bietet eine sehr hohe Skalierbarkeit. Die Verantwortung über die Verfügbarkeit und Sicherheit der Services wird an den Plattform Betreiber übergeben [19].</p>
Entscheidung	Das Backend der «Baustellen App» wird als Micro Service Architektur umgesetzt.
Begründung	Die «Baustellen App» wird in einem ersten Schritt als MVP entwickelt und soll in Zukunft weiter ausgebaut werden. Erweiterbarkeit ist deshalb von zentraler Bedeutung. MSA bietet eine gute Grundlage, um eine verfügbare, performante und erweiterbare Applikation zu erstellen.
Konsequenz	Initial ist der Aufwand für die Umsetzung einer MSA höher, weil der zusätzliche Aufwand, insbesondere im Bereich Kommunikation unter den Services, hoch ist. Der Aufwand zahlt sich auf lange Sicht aus.
Alternativen	Monolith

4.3.11.2 Einsatz von «asynchroner Datensynchronisation»

Wie im Kapitel 2.2.1 beschrieben stehen für die Datensynchronisation zwei Muster zur Verfügung. In diesem Abschnitt ist der Entscheid für das asynchrone Entwurfsmuster beschrieben.

Tabelle 27 Entwurfsentscheidung: Einsatz von Asynchroner Datensynchronisation

ID	ADR-002
Titel	Verwendung von asynchroner Datensynchronisation
Kontext	<p>Zach McCormick und Douglas C. Schmidt beschreiben zwei Entwurfsmuster für die Datensynchronisation [7].</p> <p>1. Synchrone Datenübertragung Die synchrone Datenübertragung ist im Kapitel 2.2.1.1 beschrieben. Die Anwendung wartet nach der Initialisierung von Datensynchronisation auf die Antwort. Während des Synchronisationsvorgangs ist die Benutzerinteraktion blockiert.</p> <p>2. Asynchrone Datenübertragung Die asynchrone Datenübertragung ist im Kapitel 2.2.1.2 beschrieben. Die Synchronisation von Daten wird in einem Subprozess initiiert, während die Benutzeroberfläche für den Benutzenden verwendbar bleibt.</p>
Entscheidung	Für die Datensynchronisation der «Baustellen App» wird die asynchrone Datensynchronisation verwendet.

Begründung	Die Helion AG setzt, wie in den Qualitätszielen beschrieben (QZ-003), setzt eine bedienerfreundliche Benutzeroberfläche und dass der Benutzer effizient arbeiten kann voraus. Deshalb wird auf das Entwurfsmuster «asynchrone Datensynchronisation» in der Umsetzung der «Baustellen App» gesetzt.
Konsequenz	Das State-Management der Applikation wird etwas komplexer. Sollte es sich zeigen, dass gewisse Module der «Baustellen App» auf komplett Datensätze oder dringend auf die Antwort einer Schnittstelle angewiesen sind, muss das im State-Management berücksichtigt werden.
Alternativen	Synchrone Datenübertragung

4.3.11.3 Einsatz von «Complete Storage» für den mobilen Client

Tabelle 28 Entwurfsentscheidung: «Complete Storage» für den mobilen Client

ID	ADR-003
Titel	Anwendung des Complete Storage Pattern
Kontext	Zach McCormick und Douglas C. Schmidt beschreiben zwei Entwurfsmuster für die Datenspeicherung [7]. <ul style="list-style-type: none"> 1. Partial Storage Das Entwurfsmuster Partial Storage ist im Kapitel 2.2.2.1 beschrieben. Prinzipiell werden Daten nur geladen, wenn sie benötigt werden. Dadurch wird der benötigte Speicher optimiert. 2. Complete Storage Das Entwurfsmuster Complete Storage ist im Kapitel 2.2.2.2 beschrieben. Dieses erhöht die Verfügbarkeit der Daten, auch wenn die Netzwerk Konnektivität unterbrochen ist.
Entscheidung	Für den mobilen Client wird das Entwurfsmuster Complete Storage angewendet.
Begründung	Die wichtigste Anforderung an die mobile App ist die Offline-Fähigkeit (QZ-001). Ein von Helion AG bestätigter Fakt ist, dass die Internetkonnektivität der Geräte auf der Baustelle oft unterbrochen ist. Deshalb kommt für den mobilen Client nur das Muster «Complete Storage» in Frage. Damit kann die Anforderung an die Verfügbarkeit von Projektdaten im offline Betrieb erfüllt werden. <p>Der Einsatz von «Complete Storage» bezieht sich auf offline synchronisierte Datensätze auf dem Projektlevel. Das bedeutet, dass jeweils ganze Projekte und zum Projekt gehörende Daten komplett synchronisiert werden. Alle Projekte zu synchronisieren, wäre aus speichertechnischen Gründen nicht möglich. Der Benutzer muss das zu synchronisierende Projekte auswählen (Favoritenfunktion).</p>
Konsequenz	Der Speicherbedarf auf den mobilen Endgeräten ist begrenzt. Das muss in der Umsetzung berücksichtigt werden. Entsprechend muss fehlender Speicherplatz in der mobilen App behandelt werden.
Alternativen	-

4.3.11.4 Einsatz von «Timestamp Transfer» für den «Mobile Client»

Der «Mobile Client» von der «Baustellen App» hat die bedeutende Eigenschaft, dass der Client für unbestimmte Zeit offline sein kann. Während dieser Zeit können Datensätze auf dem «Mobile Client» und im Backend auseinander divergieren. Deshalb ist die Art der Datenübertragung und Zusammenführung zwischen dem «Mobile Client» und dem Backend von grosser Bedeutung.

Als Basis für den Datentransfer wurden drei Verfahren in Betracht gezogen:

Tabelle 29 Varianten für die Datenübertragung zwischen Mobile Client und Backend

ID	Variante	Beschreibung
V1	«Full Transfer»	Siehe Kapitel 2.2.3.1
V2	«Timestamp Transfer»	Siehe Kapitel 2.2.3.2
V3	«Mathematical Transfer»	Siehe Kapitel 2.2.3.3

Um diese Varianten objektiv zu vergleichen und zu bewerten, wurde eine Nutzwertanalyse durchgeführt. In der folgenden Tabelle werden die relevanten Kriterien für die Auswahl aufgezeigt.

Tabelle 30 Bewertungskriterien bei der Auswahl des Datentransfermusters

ID	Bewertungskriterium	Beschreibung
K1	Komplexität	<p>Die betrachteten Lösungsvarianten unterscheiden sich in Bezug auf ihre Komplexität. Das Bewertungskriterium ist aus folgenden Gründen bedeutsam:</p> <ul style="list-style-type: none"> 1. Entwicklungszeit und -kosten Die Entwicklung komplexer Software erfordert oft viel Zeit und Ressourcen. Es wird berücksichtigt, wie sich die Wahl der Variante auf die Entwicklungskosten und den Aufwand auswirken würde. 2. Wartbarkeit Je komplexer eine Software ist, umso schwieriger ist es, Fehler zu finden und Anpassungen sowie Wartungen vorzunehmen. 3. Fehleranfälligkeit Eine komplexe Software enthält tendenziell mehr Punkte an denen Fehler entstehen können. Einfache Architekturen sind oft robuster und weniger fehleranfällig. <p>Bewertungsskala: 1 = hohe Komplexität 2 = mittlere Komplexität 3 = geringe Komplexität</p>
K2	Speicherbedarf	<p>Die Betrachteten Lösungsvarianten unterscheiden sich in Bezug auf den zusätzlichen Speicherbedarf. Dieses Kriterium ist aus folgenden Gründen wichtig:</p> <ul style="list-style-type: none"> 1. Zusätzlicher Speicher Gewisse Algorithmen benötigen für ihre Funktion zusätzlichen Speicherplatz. Der benötigte Speicherplatz kann sich mit dem Typ des Datensatzes stark verändern. 2. Arbeitsspeicher (RAM)

		<p>Es hängt davon ab, ob der Algorithmus seine Daten während der Verarbeitung im Arbeitsspeicher halten kann oder ob die Datenmenge eine Auslagerung in eine Datenbank erfordert. Dies beeinflusst die Effizienz des Algorithmus.</p> <p>Bewertungsskala: 1 = hoher zusätzlicher Speicherbedarf (RAM und Datenbank) 2 = mittlerer zusätzlicher Speicherbedarf (RAM oder Datenbank) 3 = kein zusätzlicher Speicherbedarf</p>
K3	Rechenleistung	<p>Die betrachteten Lösungsvarianten unterscheiden sich in Bezug auf die Rechenleistung. Aus folgenden Gründen ist dieses Kriterium von Bedeutung:</p> <ul style="list-style-type: none"> 1. Auswirkung auf die Verfügbarkeit Dauert die Berechnung für die Zusammenführung lange, hat dies Auswirkungen auf die Antwortzeiten. 2. Einfluss von grosser Divergenz Die Zusammenführung von Datensets kann mit zunehmender Divergenz rechenintensiver werden. 3. Einfluss auf die Skalierung Besonders beim Wiedererlangen der Konnektivität wird der Datentransfer und damit das Zusammenführen von Datensätzen durchgeführt. Dies kann zu einem Performance Peak führen. <p>Bewertungsskala: 1 = hohe Rechenleistung 2 = mittlere Rechenleistung 3 = geringe Rechenleistung</p>
K4	Netzwerkbandbreite	<p>Die betrachteten Lösungsvarianten unterscheiden sich in Bezug auf die benötigte Netzwerkbandbreite. Aus folgenden Gründen ist dieses Kriterium von Bedeutung:</p> <ul style="list-style-type: none"> 1. Kosten Für mobile Geräte entstehen für die Benutzung von Netzwerkbandbreite oft Kosten. 2. Verfügbare Bandbreite Die tatsächlich verwendbare Bandbreite in einem mobilen Netz kann stark variieren. <p>Bewertungskriterien: 1 = hoher Bandbreitenbedarf 2 = mittlerer Bandbreitenbedarf 3 = geringer Bandbreitenbedarf</p>
K5	Geeignetes Konfliktmanagement	<p>Die betrachteten Lösungsvarianten unterscheiden sich im Konfliktmanagement und in dessen Eignung für den mobile Client. Das Kriterium wird aus den folgenden Gründen bewertet:</p> <p>1. Konkurrierende Änderungen</p>

		<p>Konkurrierende Änderungen sind im Anwendungsszenario der «Baustellen App» wahrscheinlich. Die Datenübertragung muss mit konkurrierenden Änderungen umgehen können.</p> <p>2. Potenzieller Datenverlust Änderungen am selben Datensatz von unterschiedlichen Quellen kann zum Verlust von Daten führen. Die Akzeptanz und der Umfang von Datenverlust müssen mit den Anforderungen der Helion AG übereinstimmen.</p> <p>Bewertungskriterien: 1 = ungeeignetes Konfliktmanagement 2 = eingeschränkt geeignetes Konfliktmanagement 3 = geeignetes Konfliktmanagement</p>
--	--	--

In den folgenden Tabellen wird die Bewertung der Varianten im Bezug auf die oben beschriebenen Kriterien aufgezeigt.

Tabelle 31 Bewertung der Komplexität der Varianten

Variante	Komplexität	Begründung
«Full Transfer»	Geringe Komplexität	Das Muster «Full Transfer» überträgt alle Daten vollständig, ohne Algorithmen zur Bestimmung der zusammenzuführenden Daten zu verwenden [7]. Dadurch bleibt die Implementierung einfach und die Fehleranfälligkeit gering.
«Timestamp Transfer»	Mittlere Komplexität	Für die Implementierung wird ein Algorithmus benötigt, welcher Zeitstempel vergleicht [7]. Außerdem werden Zusätzliche Endpoints für den Austausch der Zeitstempel benötigt.
«Mathematical Transfer»	Hohe Komplexität	Diese Variante verwendet komplexe mathematische Algorithmen [4] für die Berechnung der Zusammenführung von Datensätzen. Dies führt zu hohem Entwicklungsaufwand und insgesamt zu einem komplexeren System [7].

Tabelle 32 Bewertung des Speicherbedarfs der Varianten

Variante	Speicherbedarf	Begründung
«Full Transfer»	Kein zusätzlicher Speicherbedarf	Diese Variante muss keine zusätzlichen Daten für die Datenübertragung abspeichern, weil der Datensatz jeweils als Ganzes übermittelt wird [7].
«Timestamp Transfer»	Mittlerer zusätzlicher Speicherbedarf	Damit der Zeitstempel verglichen werden kann, muss jeder Datensatz ein «Erstellt am» und «Modifiziert am» Datumsfeld enthalten [7]. Dadurch ist ein zusätzlicher Speicherbedarf pro Datensatz nötig.

«Mathematical Transfer»	Hoher zusätzlicher Speicherbedarf	Die Berechnung der Zusammenführung der Datensätzen benötigt oft zusätzlichen Speicher oder sogar separate Tabellen in der Datenbank [10]. Der Bedarf an Speicher kann mit der Zunahme der Differenz oder der Menge an Datensätze steigen [10].
-------------------------	-----------------------------------	--

Tabelle 33 Bewertung der Performance der Varianten

Variante	Rechenleistung	Begründung
«Full Transfer»	Geringe Rechenleistung	Es wird keine Rechenleistung für die Ausführung einer Zusammenführung von Datensätzen benötigt [7].
«Timestamp Transfer»	Geringe Rechenleistung	Der Vergleich der Timestamps benötigt keinen nennenswerten Mehraufwand der Rechenleistung [7].
«Mathematical Transfer»	Hohe Rechenleistung	Die Berechnung für die Zusammenfügung kann je nach Algorithmus höher ausfallen oder mit der Zunahme von Datensätzen steigen [20].

Tabelle 34 Bewertung der Netzwerkbandbreite der Varianten

Variante	Netzwerkbandbreite	Begründung
«Full Transfer»	Hoher Bandbreitenbedarf	Gemäss [7] hat das «Full Transfer» Muster einen hohen Bedarf an Netzwerkbandbreite. Dies beruht auf der Tatsache, dass die kompletten Datensätze übertragen werden.
«Timestamp Transfer»	Mittlerer Bandbreitenbedarf	Gemäss [7] benötigt dieses Muster weniger Bandbreite als «Full Transfer». In diesem Muster werden nur die Datensätze übertragen, welche sich verändert haben oder neu dazugekommen sind [7].
«Mathematical Transfer»	Geringer Bandbreitenbedarf	Dieses Muster hat das Potenzial den geringsten Bandbreitenbedarf aufzuweisen [7]. Durch mathematische Algorithmen kann die Entscheidung, welche Daten übertragen werden, sehr granular ausfallen [7].

Tabelle 35 Bewertung der Eignung des Konfliktmanagements der Varianten

Variante	Konfliktmanagent	Begründung
«Full Transfer»	Ungeeignet	Der «Full Transfer» überschreibt vorhandene Datensätze im Moment der Übertragung [7]. Im Fall der «Baustellen App» würde dies zu unvorhersehbarem Datenverlust führen. Mit dieser Variante kann zum Zeitpunkt der Synchronisierung

		nicht bestimmt werden, welche Änderung, die aktuell gültige ist.
«Timestamp Transfer»	Geeignet	Mit dem «Timestamp Transfer» wird bei einem Konflikt der zuletzt geänderte Datensatz angewendet («Last Write Wins») [7]. Aus den Anforderungen der Helion AG kann dieses Verhalten als das gewünschte Verhalten bezeichnet werden, da keine Interaktion durch die Anwender nötig ist.
«Mathematical Transfer»	Eingeschränkt geeignet	«Mathematical Transfer» hat das Potenzial Konflikte sehr granular aufzulösen [7]. Konkurrierende Änderungen sind auch für mathematische Algorithmen nicht immer zu 100% auflösbar und erfordern entweder eine Regel oder manuelles Eingreifen zur Auflösung.

In der folgenden Tabelle wird die gewählte Gewichtung der Kriterien aufgelistet und begründet.

Tabelle 36 Auflistung und Begründung der Gewichtung der Kriterien

Kriterium	Gewichtung	Begründung
Komplexität	20%	Da sich die Komplexität auf Aufwand und Kosten der Implementierung und die zukünftige Erweiterbarkeit auswirkt, hat dieses Kriterium eine höhere Gewichtung.
Speicherbedarf	5%	Der Speicherbedarf wirkt sich auf die Betriebskosten der Plattform aus. Deshalb wird dieses Kriterium betrachtet, hat jedoch eine geringe Gewichtung.
Rechenleistung	5%	Die benötigte Rechenleistung wirkt sich auf die Betriebskosten der Plattform aus. Deshalb wird dieses Kriterium berücksichtigt, hat jedoch eine geringe Gewichtung.
Bandbreite	20%	Die verfügbare Bandbreite ist eine gegebene Größe. Deshalb ist die effiziente Nutzung der verfügbaren Bandbreite eine wichtige Voraussetzung und wird entsprechend höher gewichtet.
Konfliktmanagement	50%	Das Konfliktmanagement ist für die Helion AG ein zentraler Punkt. Es sollte möglichst keine Benutzerinteraktion erforderlich sein. Trotzdem muss die unumgängliche Divergenz zwischen Datensätzen gehandhabt werden. Deshalb ist dieses Kriterium sehr hoch gewichtet.

Basierend auf der Bewertung der Kriterien ergibt sich folgendes Gesamtergebnis:

Tabelle 37 Ergebnis der Nutzwertanalyse

Kriterium	Gewicht	«Full Transfer»		«Timestamp Transfer»		«Mathematical Transfer»	
		Punkte	Gewichtet	Punkte	Gewichtet	Punkte	Gewichtet
Komplexität	20%	3	0.6	2	0.4	1	0.2
Speicherbedarf	5%	3	0.15	2	0.1	1	0.05

Rechenleistung	5%	3	0.15	3	0.15	1	0.05
Bandbreite	20%	1	0.2	2	0.4	3	0.6
Konfliktmanagement	50%	1	0.5	3	1.5	2	1
Summe	100 %	11	1.6	12	2.55	8	1.9

Die Nutzwertanalyse zeigt, dass die Variante «Timestamp Transfer» für die Implementierung in die «Baustellen App» am besten geeignet ist. Aus dieser Analyse ergibt sich folgende Architekturentscheidung:

Tabelle 38 Entwurfsentscheidung: Einsatz von Timestamp Transfer für den Datentransfer

ID	ADR-004
Titel	Anwendung des «Timestamp Transfer» Entwurfsmuster
Kontext	Der Datentransfer ist von zentraler Bedeutung für die Datensynchronisation zwischen der mobilen App und dem Backend der «Baustellen App».
Entscheidung	Für den Datentransfer wird das Entwurfsmuster «Timestamp Transfer» angewendet.
Begründung	Die Wahl des Datentransfer-Entwurfsmusters wurde auf Grund der oben aufgeführten Nutzwertanalyse getroffen.
Konsequenz	Der Datentransfer erfolgt gemäss dem «Timestamp Transfer»-Entwurfsmusters. Für die Zusammenführung von Datensätzen wird die «Last Write Wins» Regel auf Datensatzlevel angewendet.
Alternativen	Folgende Varianten wurden betrachtet: <ul style="list-style-type: none"> • «Full Transfer» • «Timestamp Transfer» • «Mathematical Transfer»

4.3.11.5 Einsatz des MVVM-Entwurfsmusters in der mobilen App

Die Randbedingung RB002 gibt vor, dass Microsoft Technologien für die Umsetzung eingesetzt werden sollen, deshalb kommt für die Umsetzung der mobilen App das Framework .Net MAUI zum Einsatz. Für die folgende Architekturentscheidung ist dieser Umstand von Bedeutung.

Tabelle 39 Entwurfsentscheidung: Einsatz von MVVM für die mobile App

ID	ADR-005
Titel	Anwendung des «MVVM»-Entwurfsmusters für die mobile App
Kontext	Das Softwaredesign der mobilen App ist von Bedeutung für die Umsetzung des Datentransfers und der Speicherung. <ol style="list-style-type: none"> 1. MVVM Das Model-View-ViewModel Muster trennt zwischen der Darstellung von Daten (View), Aufbereitung von Daten für die View und Empfangen von Datenmanipulationen (ViewModel) sowie dem Senden und Empfangen von Daten (Model). Der Einsatz des Model-View-ViewModel (MVVM) Musters ist in .NET MAUI-Anwendungen weit verbreitet [21]. MVVM hat sich als gängige Praxis etabliert, um die Trennung von Darstellung und Geschäftslogik zu gewährleisten und die Wartbarkeit und Testbarkeit von Anwendungen zu verbessern. 2. Data Access Object (DAO) Auch das DAO-Pattern kapselt Datenabfragen ab. Für die Präsentationslogik wird die Quelle der Daten dadurch irrelevant [22].
Entscheidung	Die mobile App wird mit dem Entwurfsmuster MVVM umgesetzt.
Begründung	Sowohl mit dem Muster MVVM und DAO lassen sich Datentransfer und -speicherung realisieren. Ausschlaggebend ist die Wahl des Frameworks .NET Maui und dem von Microsoft empfohlenen Einsatz von MVVM.

Konsequenz	Die mobile Applikation trennt Präsentation und Datenabfrage gemäss dem MVVM-Muster auf.
Alternativen	Data Access Object

4.3.12 Qualitätsanforderungen

Das folgende Kapitel beschreibt weitere Qualitätsszenarien. Die wichtigsten Qualitätsziele sind in Kapitel 4.3.2 dokumentiert. Hier werden weitere Qualitätsanforderungen aufgeführt und in entsprechenden Qualitätsszenarien beschrieben.

4.3.12.1 Qualitätsszenario Offline Synchronisation

Tabelle 40 Qualitätsanforderung Offline Synchronisation

ID	QSZ-001
Quelle des Auslösers	Projektleiter oder Mitarbeiter auf der Baustelle
Auslöser	Markiert ein Projekt als Favorit in der Projektübersicht.
Systembestandteil	Mobile App
Antwort	Projektdaten und dazugehörige Daten werden an das mobile Gerät geliefert und dort abgespeichert.
Antwortmetrik	Synchronisationsstatus
Anforderung	Der Synchronisationsstatus wird dem Benutzer angezeigt, damit dieser den aktuellen Status kennt.
	Status: <ul style="list-style-type: none"> • Nicht synchronisiert • Am Synchronisieren • Synchronisiert

4.3.12.2 Qualitätsszenario Datensynchronisation

ID	QSZ-002
Quelle des Auslösers	Mobile App
Auslöser	Wechselt vom offline in den online Modus
Systembestandteil	Mobile App
Antwort	Sendet lokale Datenmanipulationen an das Backend und empfängt aktuellere Daten vom Backend.
Antwortmetrik	
Anforderung	<ul style="list-style-type: none"> • Die Synchronisation der Daten wird automatisch ausgelöst, wenn der mobile Client wieder Internetkonnektivität besitzt. • Trifft für «Favorisierte» Projekte zu.

4.3.13 Risiken

In diesem Kapitel werden die wichtigsten Risiken aufgeführt, bewertet und entsprechende Massnahmen beschrieben. Identifiziert als Risiko wurde die Datenintegrität und die Verfügbarkeit von Daten im mobilen Client.

4.3.13.1 Risiko Datensicherheit - Integrität

Tabelle 41 Risikobewertung: Datenverlust während Datenzusammenführung

ID	R001
Risiko	Die gewählte Zusammenführungsstrategie «Last Write Wins» führt zu Datenverlust und verletzt das Qualitätsziel QZ-002 «Sicherheit - Datenintegrität».
Ursachen	Wenn offline synchronisierte Daten sowohl in einem oder mehreren offline Client und/oder im WebPortal manipuliert werden, müssen diese Änderungen während der nächsten Synchronisation zusammengeführt werden.
Mögliche Auswirkungen	Die letzte Manipulation gemäss Zeitstempel wird angewendet. Alle früheren Manipulationen werden verworfen und gehen verloren.
Eintrittswahrscheinlichkeit	Hoch
Bewertung	Mittel
Mögliche Massnahmen	<ol style="list-style-type: none"> Zusammenführung von Daten auf dem Attributlevel. Dadurch wird die Eintrittswahrscheinlichkeit gesenkt. Einsatz des «Mathematical Transfer» Musters. Mitarbeiter können nur lesend auf Projektdaten zugreifen. Notizen und IBN-Protokolle können durch Mitarbeiter nur erstellt, nicht bearbeitet werden.
Entscheidung	Dieses Risiko wurde mit dem Kunden bewertet und die Massnahmen abgesprochen. Hier wurde zwischen Aufwand und Nutzen abgewägt. Sowohl «Mathematical Transfer» als auch Zusammenführung auf Attributlevel führt zu einer Erhöhung der Komplexität, bedeutet Mehraufwand und -kosten und schmälert im Gegenzug die Wartbarkeit. Deshalb wurde entschieden, auf Datensatzebene auf «Timestamp Transfer» zu setzen. Wie in der 3. Massnahme beschrieben, sollen Mitarbeiter nur lesend auf Projektdaten zugreifen. Projektleiter und Sachbearbeiter können Daten bearbeiten.

4.3.13.2 Risiko Zuverlässigkeit - Verfügbarkeit

Tabelle 42 Risikobewertung Zuverlässigkeit der «Baustellen App»

ID	R002
Risiko	Die Qualitätsanforderung QZ-001 «Verfügbarkeit» kann nicht eingehalten werden.
Ursachen	<ul style="list-style-type: none"> Das Projekt wurde vor dem Verbindungsunterbruch nicht mit der mobilen App synchronisiert.
Mögliche Auswirkungen	Besteht keine Verbindung zum Backend (Offline Modus), können keine Projektdaten aus dem Backend nachgeladen werden. Deshalb ist ein nicht synchronisiertes Projekt im Offlinemodus nicht einsehbar. Damit Projektdaten geladen werden können, muss sich der Benutzer an einen Ort mit Verbindung in das Internet begeben.
Eintrittswahrscheinlichkeit	Mittel
Bewertung	Niedrig
Mögliche Massnahmen	<ol style="list-style-type: none"> Hinweis in der mobilen App einblenden. Benutzer schulen, dass vorgängig Projekte synchronisiert werden.

	<ol style="list-style-type: none"> 3. Projektsynchronisation in den Prozess der Arbeitsvorbereitung für den Einsatz auf der Baustelle einbinden. 4. Alle Projekte auf den «Mobile Client» synchronisieren. 5. Die zuletzt geöffneten Projekte werden synchronisiert.
Entscheidung	Mit dem Kunden wurde entschieden, dass die Benutzer Projekte als «Offline Verfügbar» markieren können. Diese Projekte werden in der mobilen App für die offline Verwendung synchronisiert.

4.4 Umsetzung der «Baustellen App» als PoC

Im folgenden Abschnitt wird die Umsetzung der «Baustellen App» als PoC beschrieben. Wie bereits in der Einleitung 1.1 dargestellt, hat sich die Auftraggeberin «Helion AG» aus dem Projekt zurückgezogen. Dies hat zur Folge, dass sich der Fokus von der Entwicklung einer mobilen Applikation für das bestehende THEIA Backend, hin zu einer Full-Stack Entwicklung verschoben hat. In der Lösungsarchitektur wurde deshalb das System als Ganzes gemäss den Anforderungen des Kunden konzipiert. Die Lösungsarchitektur wäre jedoch für eine Umsetzung im Rahmen dieser Arbeit zu umfangreich. Deshalb wurde in Absprache mit den Experten der Umfang reduziert. Im Zentrum steht nun die Demonstration des Datensynchronisationskonzeptes aus der Lösungsarchitektur, inklusive Authentifizierung und Autorisierung von Anwendern.

4.4.1 Verwendete Technologien

Für die Umsetzung der mobilen Applikation wurde das Multi-Plattform Framework .Net Maui⁸ von Microsoft verwendet. Die Applikationslogik wird mit der Programmiersprache C# implementiert. Die Benutzeroberfläche wird mit der Markup Sprache XAML deklariert. Maui unterstützt unter anderem Android und iOS, welche als Zielplattformen von der Helion AG spezifiziert wurden. Im Backend kommt das Framework ASP.NET⁹ für die Entwicklung der APIs zum Einsatz. Da alle diese Frameworks auf C# basieren, lassen sich sowohl im Backend wie auch im Frontend Algorithmen und Datenobjekte wieder verwenden. Außerdem wird im Backend .Net Aspire¹⁰ für die Service Discovery und Container Orchestrierung eingesetzt. Ausgerollt wird das Backend in eine Azure Container Apps¹¹ Umgebung.

4.4.2 Applikationsmodel Backend

In den nächsten Abschnitten wird das Klassenmodell der umgesetzten Komponenten aufgezeigt. Klassen und Schnittstellen aus Bibliotheken oder welche in einem früheren Diagramm eingeführt wurden, werden in nachfolgenden Diagrammen als Black-Box dargestellt.

⁸ <https://learn.microsoft.com/de-de/dotnet/maui/what-is-maui?view=net-maui-9.0>

⁹ <https://dotnet.microsoft.com/en-us/apps/aspnet>

¹⁰ <https://learn.microsoft.com/en-us/dotnet/aspire/get-started/aspire-overview>

¹¹ <https://azure.microsoft.com/de-de/products/container-apps>

4.4.2.1 Geteilte Objekte (Shared Objects)¹²

Wie bereits in 4.4.1 beschrieben verwenden alle Applikationskomponenten C# für die Implementierung. Deshalb können einige Teile für andere Komponenten wieder verwendet werden. In diesem Abschnitt sind alle geteilten Datenobjekte beschreiben.

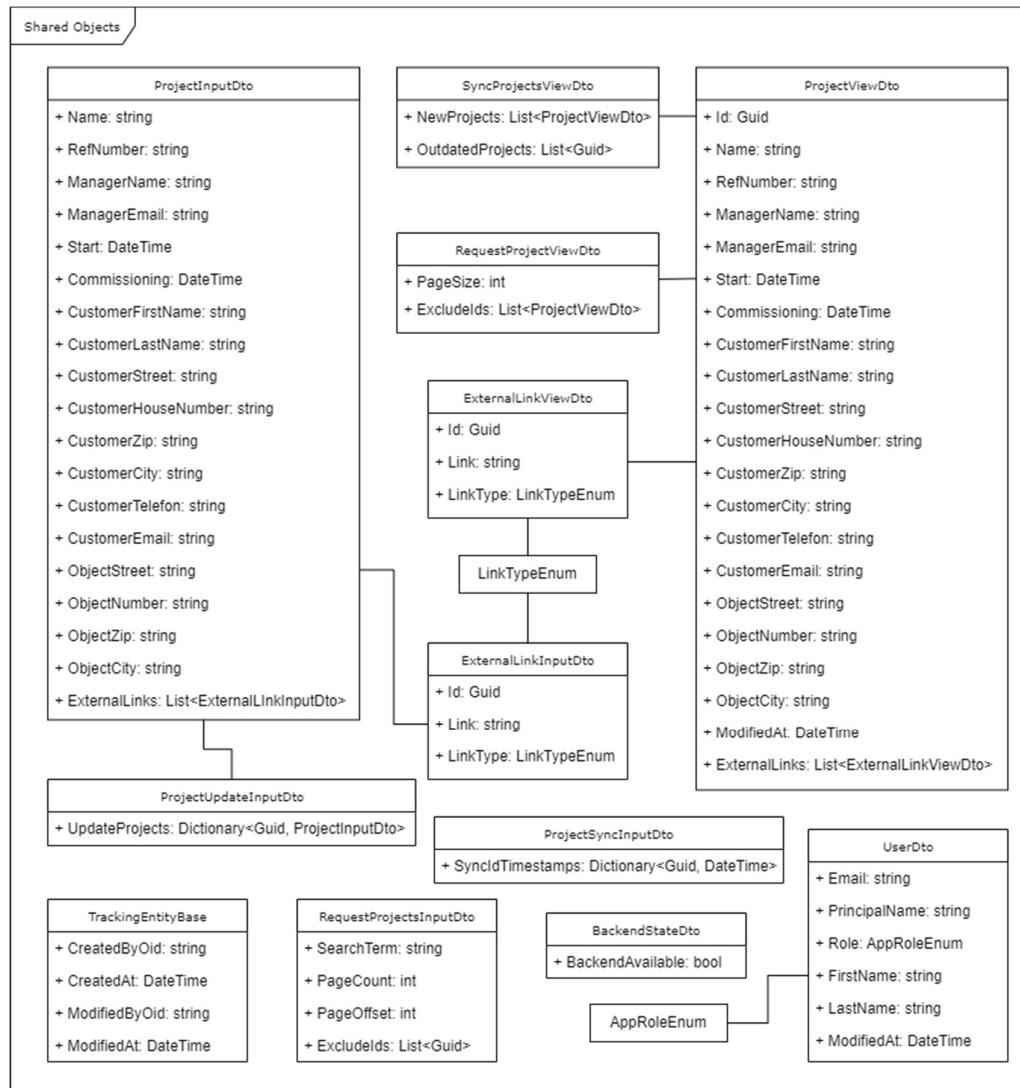


Abbildung 43 Geteilte Objekte für das gesamte Softwaresystem

Die «Datentransfer Objekte» (Dto) in der Abbildung 43 werden sowohl in den Backend Services als auch in der Mobilen Applikation verwendet. Ausserdem werden sie zur Serialisierung in JSON verwendet für die Übertragung via HTTP.

¹² <https://github.com/zuercheram/baustellen-app/tree/dev/src/Shared>

4.4.2.2 Projekt API

Die Komponente «Project.Api» ist verantwortlich für das Persistieren, Abfragen und Synchronisieren von Projektdaten.

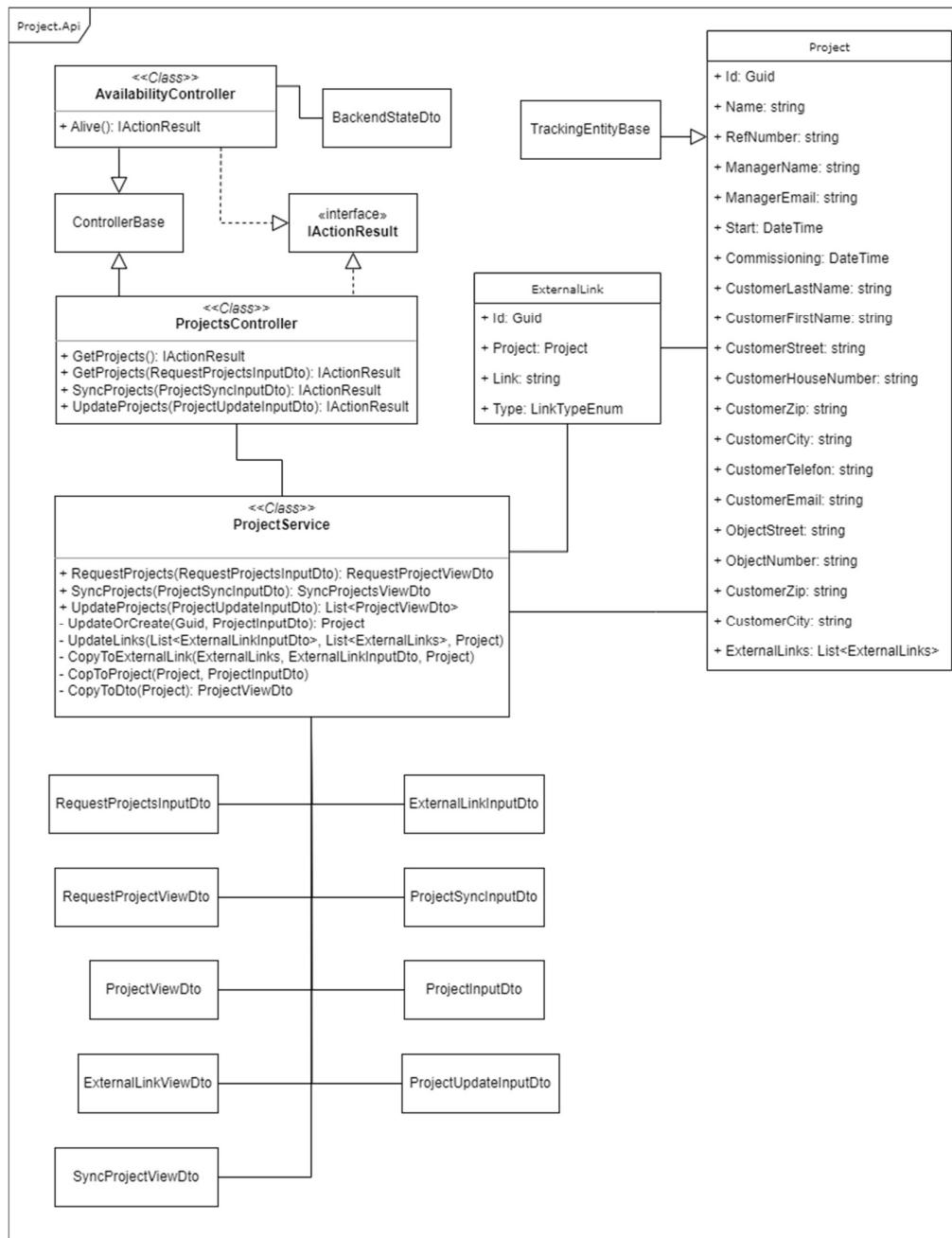


Abbildung 44 Klassendiagramm zum Backendservice Project.Api

t

4.4.2.3 Identity Api

Applikationsbenutzer werden durch die «Identity.Api» verwaltet. Ein wichtiger Aspekt in diesem Service ist die Rollenverwaltung. Benutzerrollen werden durch andere Services oder die mobile Applikation abgefragt, um den authentifizierten Benutzer zu autorisieren.

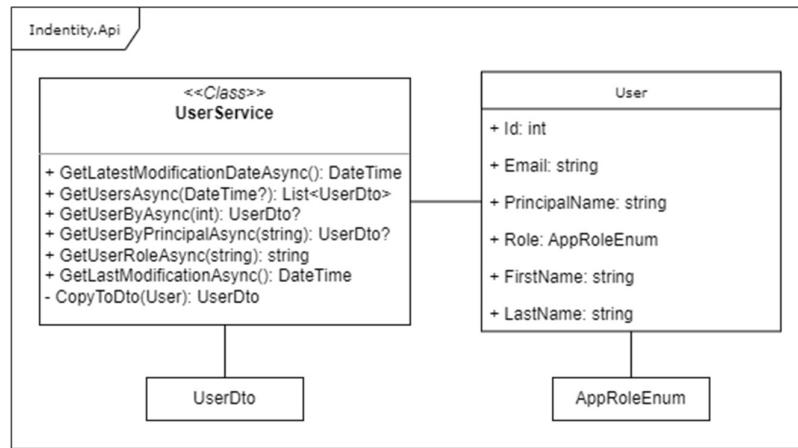


Abbildung 45 Klassendiagramm zum Backendservice Identity.Api

4.4.3 Generelles Applikationsmodell der mobilen Applikation

Das Applikationsmodell ausgenommen für die mobile Applikation wird im nachfolgenden Abschnitt beschreiben. Ausgenommen sind die Modelle für die Ansichten. Diese werden in einem separaten Abschnitt beschrieben. Zur Verbesserung der Übersicht, wurden die Diagramme nach Verantwortlichkeiten aufgeteilt.

4.4.3.1 Basisklassen für die Wiederverwendung von Funktionen¹³

In diesem Abschnitt werden einige Basisklassen beschrieben, welche in der mobilen Applikation verwendet werden.

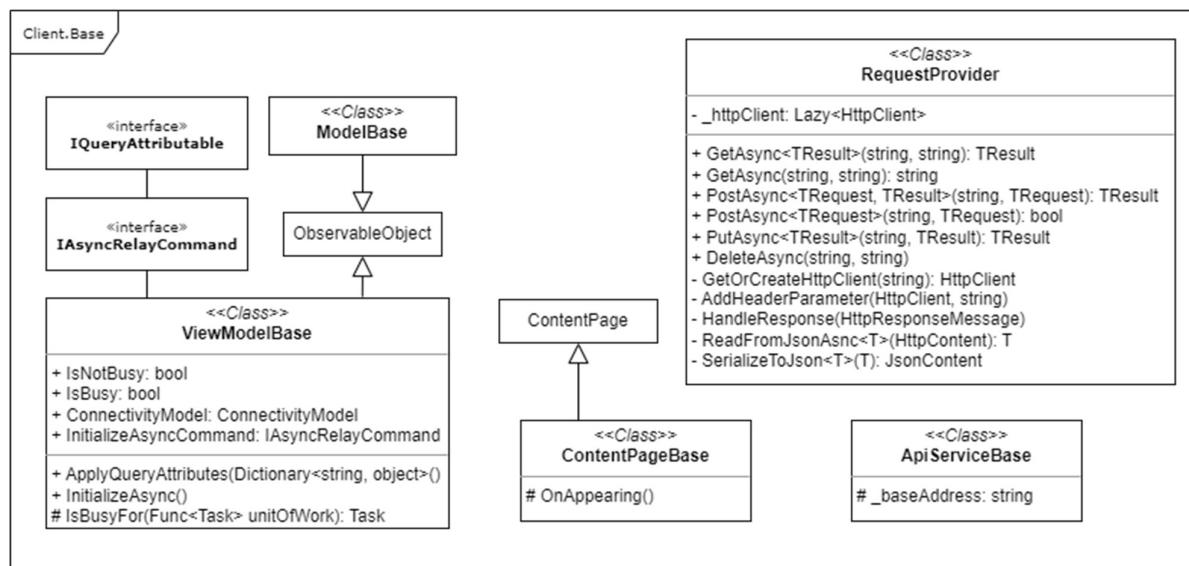


Abbildung 46 Klassendiagramm zu verschiedenen Basisklassen für die mobile Applikation

Insbesondere die Maui-Views, -Pages, ViewModels und Models verwenden Basisklassen, welche generelle Funktionen und Eigenschaften abstrahieren und verfügbar machen. Beispielsweise enthält

¹³ <https://github.com/zuercheram/baustellen-app/tree/dev/src/Client>

die «ViewModelBase» Funktionalität, um dem Benutzer anzuzeigen, wenn Hintergrundberechnungen stattfinden. Hierfür wird die Eigenschaft «IsBusy» verwendet. Diese Eigenschaft wird von allen ViewModels verwendet und ist deshalb in der Basisklasse zu finden.

4.4.3.2 Serviceklassen für den Zugriff auf das Backend¹⁴

Für die Abfrage von Daten aus dem Backend werden Klassen mit dem Suffix «Service» implementiert. Sie enthalten die Logik, um Daten vorzubereiten, damit sie an das Backend gesendet werden können oder nimmt Daten aus einer Antwort des Servers entgegen und deserialisiert den Inhalt in ein konkretes Objekt.

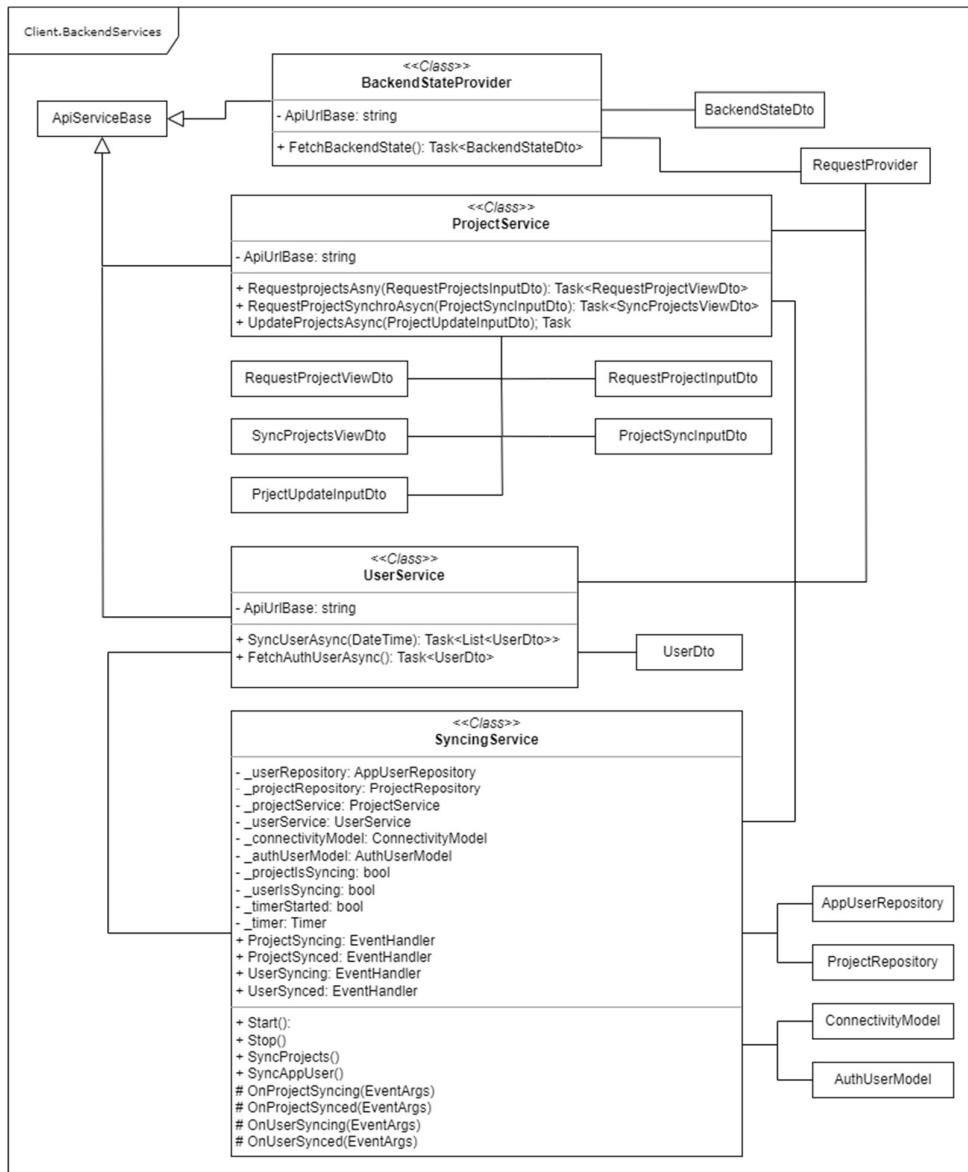


Abbildung 47 Klassendiagramm zu den Serviceklassen für die Kommunikation mit dem Backend

¹⁴ <https://github.com/zuercheram/baustellen-app/tree/dev/src/Client/Services>

4.4.3.3 Lokale Datenablage und abfrage¹⁵

Für den offline Zugriff werden Projekt- und Benutzerdaten lokal auf dem mobilen Gerät gespeichert. Im folgenden Diagramm werden die Datenklassen und Abfragerklassen für lokale Daten dargestellt.

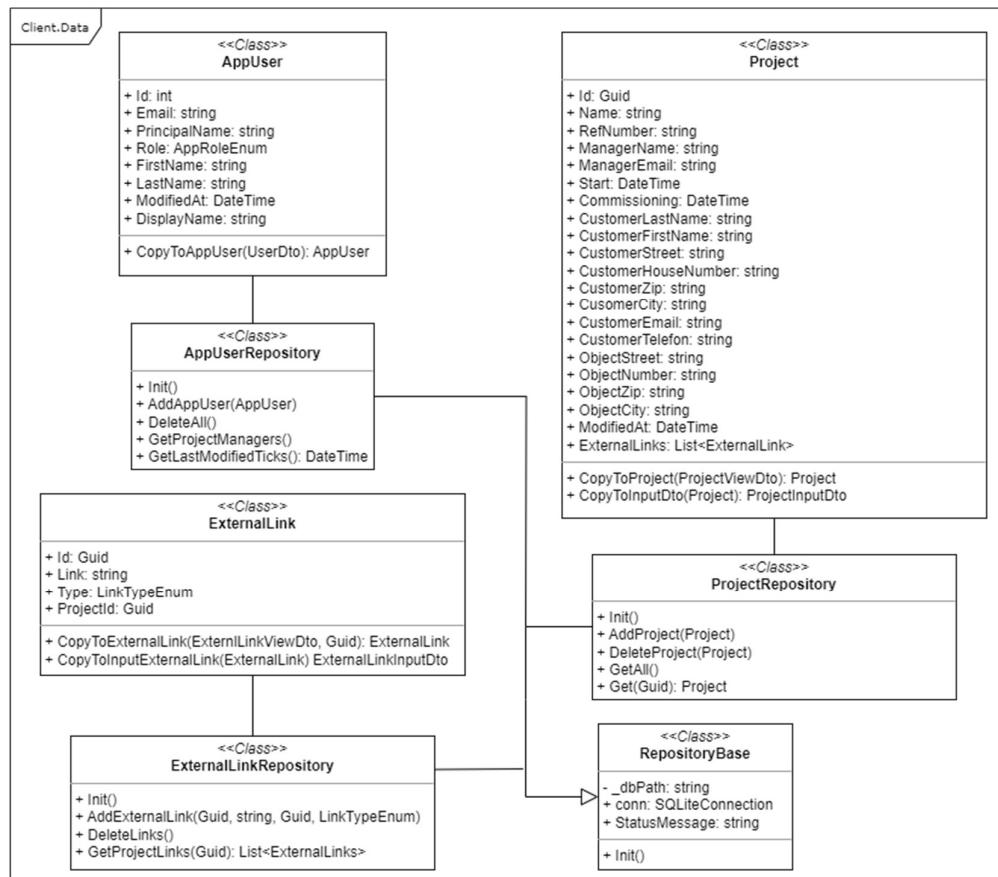


Abbildung 48 Klassendiagramm für das Persistieren von Daten auf einem mobilen Gerät

Daten werden auf dem mobilen Gerät in einer SQLite Datenbank persistiert. SQLite bietet den Vorteil einer sehr schlanken Implementierung auf einem mobilen Gerät. Für den Betrieb der Datenbank sind keine zusätzlichen Services oder Prozesse nötig. Das Datenbankfile wird auf dem für die App reservierten Bereich des Filesystems gespeichert und der Zugriff erfolgt im Applikationsprozess. Dazu wurden die zu speichernden Daten als Klassen modelliert. Via Repositoryklassen werden die Datenobjekte in die SQLite Datenbank gespeichert oder abgefragt.

¹⁵ <https://github.com/zuercheram/baustellen-app/tree/dev/src/Client/Data>

4.4.4 Ansichten für die mobile Applikation

In diesem Abschnitt werden die Modelle für die Ansichten der Applikation beschreiben. Die Ansichten sind nach dem MVVM Prinzip aufgebaut.

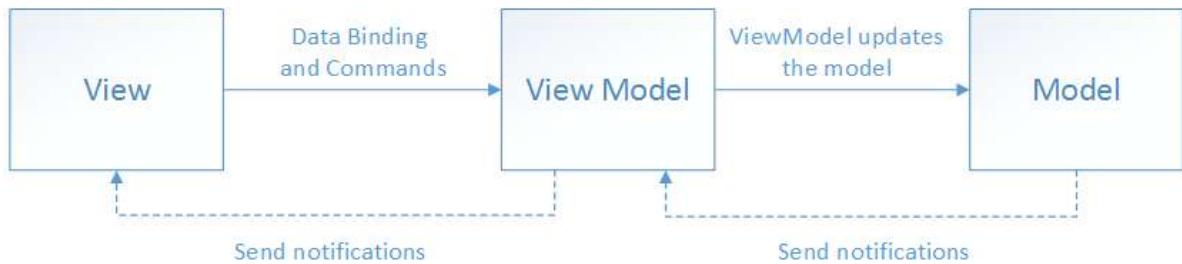


Abbildung 49 Darstellung des Model View ViewModel Entwurfsmusters¹⁶

In der View ist die Benutzeroberfläche mittels XAML definiert. Das ViewModel enthält View spezifische Logik und empfängt beispielsweise Benutzerinteraktionen und reagiert darauf. Außerdem stellt das ViewModel die Datenfelder für die Anzeige auf der View bereit. Das Model enthält datenspezifische Logik. Es initialisiert beispielsweise die Abfrage von Daten aus dem Backend. Das Model referenziert deshalb oftmals die Services für die Kommunikation mit dem Backend und die Datenrepositories für den Zugriff auf die lokalen Daten.

4.4.4.1 Anwendungsansichten: Hauptseite¹⁷

Die Hauptseite ist der Einstieg in die mobile Applikation. Nach der Initialisierung wird die Hauptseite geladen und auf dem Bildschirm des Gerätes dargestellt. In der «Baustellen App» enthält die Hauptseite die offline und die online Projekte in zwei separaten Listen. Hat die Applikation eine Verbindung zum Backend werden die online Projekte angezeigt. Andernfalls wird der gesamte Abschnitt für die online Projekte ausgeblendet.

¹⁶ <https://learn.microsoft.com/de-de/dotnet/architecture/maui/media/mvvm-pattern.png>

¹⁷ <https://github.com/zuercheram/baustellen-app/tree/dev/src/Client/Views>

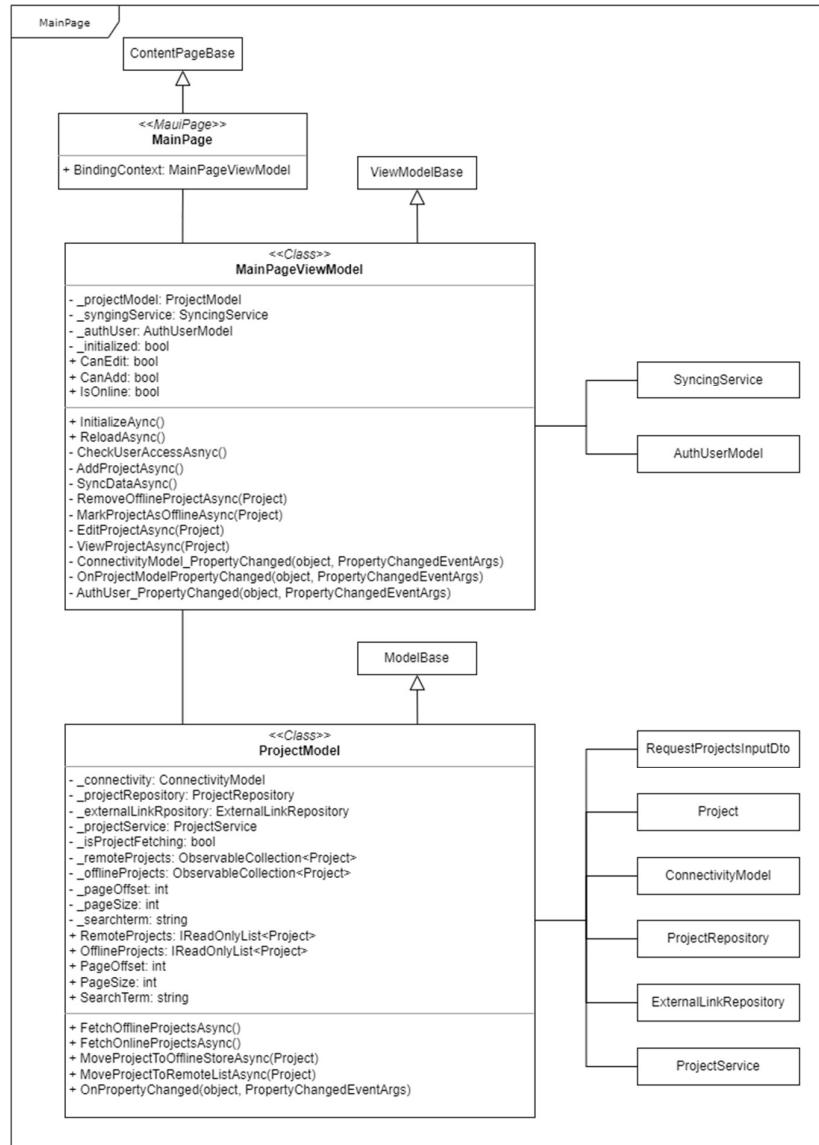


Abbildung 50 Klassendiagramm der Hauptseite

4.4.4.2 Anwendungsansichten: Projekt bearbeiten¹⁸

Die Bearbeitung der Projekte wird über diese Ansicht in der Applikation bereitgestellt.

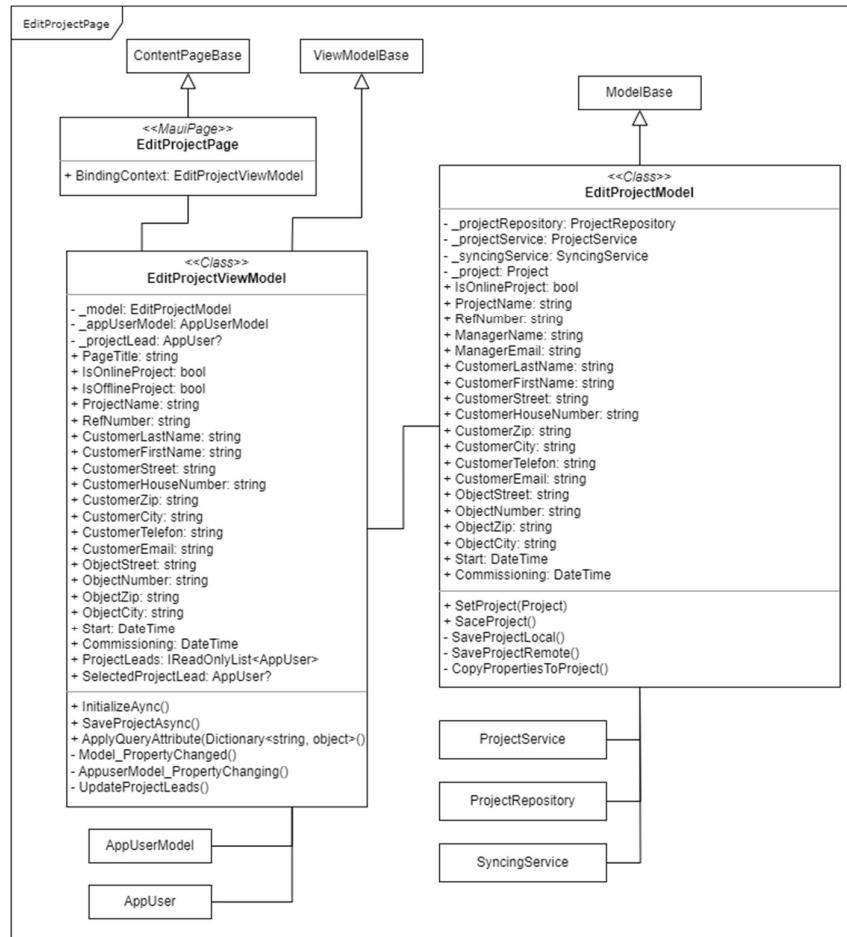


Abbildung 51 Klassendiagramm der Projektbearbeitungsansicht

¹⁸ <https://github.com/zuercheram/bauestellen-app/tree/dev/src/Client/Views>

4.4.4.3 Anwendungsansichten: Projektdaten betrachten¹⁹

Damit auf die Projektdaten ohne Bearbeitungsfunktion zugegriffen werden kann, stellt die mobile Applikation eine Projektansicht zur Verfügung.

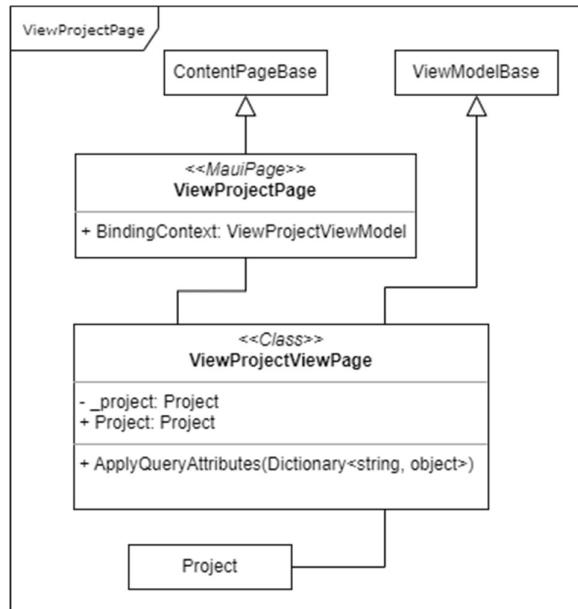


Abbildung 52 Klassendiagramm der Projektansicht

4.4.4.4 Anwendungsansichten: Benutzerprofil und Authentifizierung²⁰

Die Benutzerauthentifizierung wird über die Benutzerprofilansicht abgewickelt. Anwender haben dort die Möglichkeit, sich an- bzw. abzumelden. Außerdem wird dort das Benutzerobjekt des authentifizierten Benutzers angezeigt. Daraus kann beispielsweise die zugeteilte Applikationsrolle gelesen werden.

¹⁹ <https://github.com/zuercheram/bauestellen-app/tree/dev/src/Client/Views>

²⁰ <https://github.com/zuercheram/bauestellen-app/tree/dev/src/Client/Views>

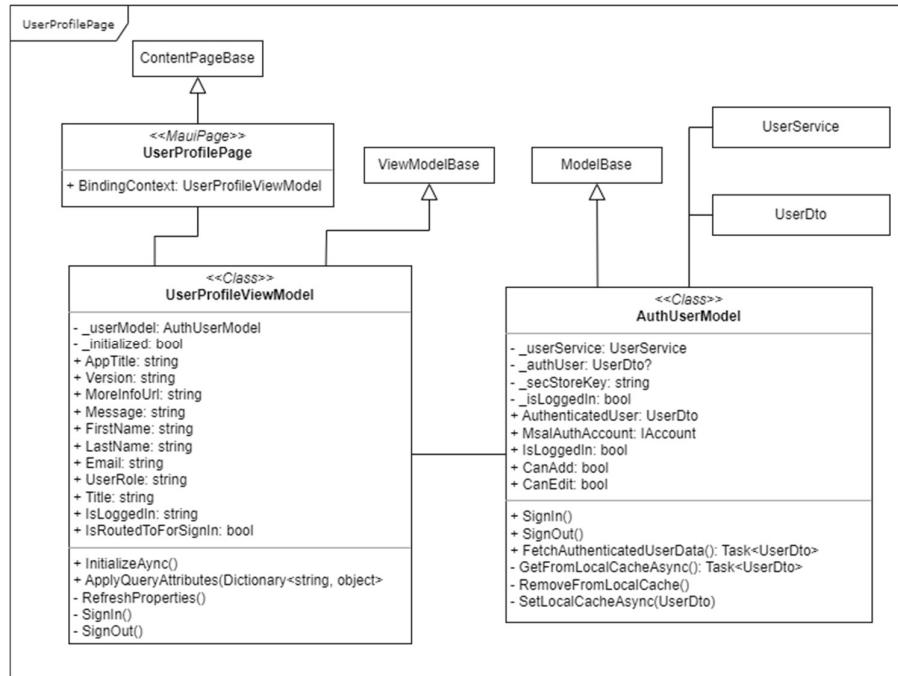


Abbildung 53 Klassendiagramm der Benutzeroauthentifizierung

4.4.5 Umsetzung der Datensynchronisation

Im PoC wurde die Datensynchronisation gemäss dem Konzept aus Kapitel 4.3.8.1 umgesetzt. Der Ablauf der Synchronisation ist in der Abbildung 31 ersichtlich. Die umgesetzte Synchronisation entspricht diesem Ablauf. Die Synchronisation wird im PoC immer von der mobilen Applikation ausgelöst. Im PoC wird die Synchronisation in folgenden Fällen ausgelöst:

- Die mobile App registriert den Wiederaufbau einer Verbindung zum Backend.
- Anwender können die Synchronisation manuell auslösen.
- Ein Hintergrundservice löst regelmässig eine Synchronisation aus.

Für die Synchronisation wird eine aktive Verbindung zum Backend benötigt. Der Hintergrundservice überprüft die Verbindung und unterricht die regelmässige Synchronisation, wenn die Verbindung unterbrochen ist und nimmt diese wieder auf, sobald eine neue Verbindung vorhanden ist.

5 Diskussion/ Zusammenfassung

In diesem Kapitel werden die Ergebnisse der Masterthesis diskutiert. Dabei wird zunächst auf die Analyse des Datentransfer Entwurfsmusters (Kapitel 4.3.11.4) eingegangen. Anschliessend wird die Lösungsarchitektur (Kapitel 4.3) und die Umsetzung des PoC (Kapitel 4.4) kritisch hinterfragt. Das Kapitel schliesst mit einer Darstellung von Verbesserungspotenzial und der zukünftigen Weiterentwicklung ab.

5.1 Eignung von «Timestamp Transfer» für die Synchronisation der Projektdaten

Im Rahmen der Arbeit wurden folgende Entwurfsmuster genauer betrachtet:

- «Full Transfer»
- «Timestamp Tranfer»
- «Mathematical Transfer»

Es hat sich herausgestellt, dass das Muster «Full Transfer» auf Grund der benötigten Bandbreite und das Muster «Mathematical Transfer» auf Grund der Komplexität für die Implementierung nicht in Frage kommen. Obwohl im «Baustellen App» Kontext der «Timestamp Transfer» ein gewisses Risiko

zu Datenverlust besteht, ist dieses Muster im Verhältnis von Aufwand und Ertrag am geeignetsten. Ausserdem kann das Risiko des Datenverlustes mit folgenden Massnahmen verringert werden:

- Auf der Mobilen App können Daten nur lesend betrachtet werden.
- Auf der Mobilen App können Daten nur hinzugefügt werden (Notizen und Inbetriebnahme Protokolle)
- Änderungen sind nur durch Sachbearbeiter und Projektleiter via Webportal möglich.
- Detailliertes Konzept, welche Felder wann und durch wen bearbeitet werden können.

Ein detailliertes Zugriffskonzept bedeutet mehr Aufwand für die Umsetzung, bietet jedoch eine gute Minimierung des Datenverlustrisikos.

Die Zusammenführung von Daten mittels «Mathematical Transfer» würde das Datenverlustrisiko auch minimieren. Der Aufwand für die Implementierung wäre jedoch höher. In Fall der «Baustellen App» nimmt der Kunde Helion AG explizit das Risiko von geringem Datenverlust in Kauf, da dies durch das Rollen- und Zugriffskonzept minimiert werden kann.

5.2 Eignung der vorgeschlagenen Lösungsarchitektur

Im Rahmen dieser Arbeit wurde die «Baustellen App» gemäss den Anforderungen aus dem Pflichtenheft der Helion AG entworfen. Der Umfang wurde von der Helion AG als MVP definiert. Die Lösungsarchitektur beschreibt das komplette Softwaresystem, welches für die Umsetzung der Anforderungen nötig wäre. Da eine komplette Umsetzung des MVPs im Rahmen dieser Masterarbeit nicht möglich war, wurde ein Proof of Concept umgesetzt. Der PoC implementiert die Datensynchronisation, Authentifizierung und die Autorisierung. Damit die Funktionen getestet werden können, wurde eine einfache Benutzeroberfläche Android und iOS implementiert. Die Benutzeroberfläche stellt Projektdaten aus dem Backend dar und erlaubt diese für die offline Synchronisierung zu markieren. Die Datensätze können erstellt oder bearbeitet werden. Der Proof of Concept zeigt auf, dass die konzipierte Lösung für die Datensynchronisation funktioniert und eine Umsetzung im Rahmen des MVPs durchgeführt werden kann.

5.3 Verbesserungspotenzial

Im folgenden Abschnitt wird auf mögliches Verbesserungspotenzial eingegangen.

5.3.1 Einatz von SignalR oder gRPC

In der Lösungsarchitektur wurde für die Kommunikation mit dem Backend auf REST-Apis gesetzt. Im Umfeld von mobilen Applikationen würde sich jedoch der Einsatz von alternativen Protokollen anbieten, beispielsweise gRPC oder SignalR. Beide Protokolle setzen auf Remote Procedure Call und Web Sockets. Im Gegensatz zu http bieten diese Protokolle eine schnellere Datenübertragung und hätten dadurch das Potenzial, die Performance der «Baustellen App» zu verbessern. Ausserdem ermöglichen diese Protokolle die aktive Kommunikation von dem Backend in das mobile Frontend. In der aktuellen Lösungsarchitektur muss das Frontend das Laden von Daten aus dem Backend initialisieren. Damit die mobile App Änderungen an Daten mitbekommt, muss das Backend regelmäßig nach Anpassungen abgefragt werden. Dies wird auch als Polling bezeichnet und hat den Nachteil Netzwerkbänder zu beanspruchen. Mit SignalR oder gRPC könnte dieses Polling verhindert und dadurch Netzwerkbänder eingespart werden.

5.3.2 Weitere Identitätsprovider und externe Registrierung zulassen

Im PoC wird für die Authentifizierung Microsoft Entra ID verwendet. Dies setzt allerdings voraus, dass die Anwender in Entra ID verwaltet werden. Die Helion AG verwendet Entra ID für die Verwaltung ihrer Mitarbeiter. Allerdings haben externe und temporäre Mitarbeiter keinen Zugriff auf die «Baustellen App». Andere Identity Provider wie Google und Apple Authentifikation würden den Benutzerkreis erweitern. Eine zusätzliche Rolle für externe Identitäten könnte den Anwendern ausserhalb der Helion AG eine Entra ID automatisch vergeben. Entra ID könnte ebenfalls für die Verwendung von externen Mitarbeitern konfiguriert werden. Eine sogenannte B2C Funktionalität würde es externen Mitarbeitern ermöglichen, sich im Helion B2C Tenant zu registrieren und so Zugriff auf die «Baustellen App» zu erhalten.

5.4 Zukünftige Weiterentwicklung

Das Projekt zur Umsetzung der «Baustellen App» wurde von der Helion AG nicht weiterverfolgt. Deshalb wird auch das Ergebnis dieser Arbeit nicht weiterentwickelt. Der Erkenntnisgewinn bezüglich Datensynchronisation und mobiler Applikationsentwicklung mittels .Net Maui wird innerhalb der isolutions AG jedoch dringend für zukünftige Projekte benötigt. Die Arbeit bietet somit einen wertvollen Beitrag zur Erweiterung der Expertise innerhalb der isolutions AG bei.

5.5 Persönliche Betrachtung und Danksagung

Die Masterarbeit hatte zum Ziel, eine Lösungsarchitektur und eine Nutzwertanalyse für den Einsatz eines Entwurfsmusters durchzuführen und die beschriebene Lösung umzusetzen. Die Arbeit sollte im Rahmen eines Projektes für die Helion AG umgesetzt werden. Die Helion AG hätte das Backend zur Verfügung gestellt. Somit war der ursprüngliche Fokus, die Entwicklung der mobilen Applikation. Da sich die Helion AG aus dem Projekt zurückgezogen hatte, verschob sich auch der Fokus der Lösungsarchitektur und betrachtete ebenfalls die Backend Services. Insgesamt hat sich dadurch auch der Umfang der Arbeit verändert und die Lösung konnte nicht mehr als gesamtes System umgesetzt werden. Deshalb wurde die Umsetzung im Verlauf der Arbeit zu einem Proof of Concept geändert. Dieser PoC sollte die Implementierung des «Timestamp Transfer» Entwurfsmusters, und die Authentifizierung demonstrieren. Der Erkenntnisgewinn im Bereich Datentransfer und mobiler Applikationsentwicklung ist für den Autor und die Isolutions AG von grosser Bedeutung.

Der Autor dankt den Experten Max Helmut Kleiner und Arno Schmidhauser für die Begleitung und Beurteilung der Arbeit, Regina Traber und Anna Zürcher für das Gegenlesen und der Überprüfung der Rechtschreibung.

6 Abbildungsverzeichnis

Abbildung 1 Konsistentes System	8
Abbildung 2 Inkonsistentes System.....	8
Abbildung 3 Datensynchronisation als synchroner Fluss [7]	9
Abbildung 4 Darstellung einer asynchronen Datensynchronisation [7]	9
Abbildung 5 Ablauf von «Partial Storage» Datenabfragen [7]	10
Abbildung 6 Darstellung der «Complete Storage» Datenabfrage [7]	11
Abbildung 7 Darstellung des «Full Transfer» Übertragungsmuster [7]	12
Abbildung 8 Darstellung des Musters «Timestamp Transfer» [7]	12
Abbildung 9 Darstellung des Musters «Mathematischer Transfer» [7].....	13
Abbildung 11 Mockups: Darstellung der Projektübersicht	20
Abbildung 12 Mockups: Detailansicht eines Projektes	20
Abbildung 13 Mockups: Übersicht der Notizen	21
Abbildung 14 Mockups: Detailansicht einer Notiz	21
Abbildung 15 Mockups: Notiz erfassen	21
Abbildung 17 Anwendungsfalldiagramm für die Applikation «Baustellen App».....	23
Abbildung 18 Fachlicher Kontext	26
Abbildung 19 Bausteinsicht Level 0 Fachlicher Kontext	29
Abbildung 20 Bausteinsicht Level 0 Technischer Kontext.....	30
Abbildung 21 Bausteinsicht des «Mobile Client» Level 1	32
Abbildung 22 Bausteinsicht «Web Portal» Level 1	33
Abbildung 23 Bausteinsicht «API Gateway» Level 1	33
Abbildung 24 Bausteinsicht «Customer Service» Level 1	34
Abbildung 25 Bausteinsicht des «Project Service» Level 1	35
Abbildung 26 Bausteinsicht des «Note Service» Level 1	36
Abbildung 27 Bausteinsicht «Commissioning Service» Level 1	37
Abbildung 28 Bausteinsicht «Identity Service» Level 1	38
Abbildung 29 Bausteinsicht «File Service» Level 1	38
Abbildung 30 Schematischer Ablauf der Datensynchronisation im «Mobilien Client».....	39
Abbildung 31 Schematischer Ablauf der Datensynchronisation in den Backend Services	40
Abbildung 32 Laufzeitansicht zur Datensynchronisation der support backend Services.....	40
Abbildung 33 Laufzeitansicht zur Datensynchronisation Mobile Client zu Project Service	41
Abbildung 34 Laufzeitansicht zur Datensynchronisation von «Mobile Client» zu «Note Service»	41
Abbildung 35 Laufzeitansicht zur Datensynchronisation von «Mobile Client» zu «Comissioning Service»	42
Abbildung 36 Laufzeitansicht zur Projektadministration im «Web Portal».....	43
Abbildung 37 Laufzeitsicht zur Erfassung und Bearbeitung von Notizen im Web Portal	43
Abbildung 38 Laufzeitsicht zur Administration von Checklisten	44
Abbildung 39 Verteilung der Anwendungskomponenten auf Infrastrukturkomponenten	44
Abbildung 40 Netzwerkansicht über die Entwicklungsumgebung der «Baustellen App».....	45
Abbildung 41 Authentifizierung mit Azure Entra ID.....	46
Abbildung 42 Ablauf der Authentifizierung und Token Validierung.....	47
Abbildung 43 Darstellung des Build und Deployment Flows	47
Abbildung 44 Darstellung der Qualitätssicherungslevels mit den betroffenen Komponenten.....	48
Abbildung 45 Geteilte Objekte für das gesamte Softwaresystem	62

Abbildung 46 Klassendiagramm zum Backendservice Project.Api.....	63
Abbildung 47 Klassendiagramm zum Backendservice Identity.Api.....	64
Abbildung 48 Klassendiagramm zu verschiedenen Basisklassen für die mobile Applikation	64
Abbildung 49 Klassendiagramm zu den Serviceklassen für die Kommunikation mit dem Backend.....	65
Abbildung 50 Klassendiagramm für das Persistieren von Daten auf einem mobilen Gerät	66
Abbildung 51 Darstellung des Model View ViewModel Entwurfsmusters	67
Abbildung 52 Klassendiagramm der Hauptseite	68
Abbildung 53 Klassendiagramm der Projektbearbeitungsansicht	69
Abbildung 54 Klassendiagramm der Projektansicht.....	70
Abbildung 55 Klassendiagramm der Benutzeroauthentifizierung	71

7 Tabellenverzeichnis

Tabelle 1 Anforderungen an die Authentifizierung	17
Tabelle 2 Anforderungen an die Projektübersicht	17
Tabelle 3 Anforderungen an die Projektdetails	18
Tabelle 4 Anforderung zu den Notizen	19
Tabelle 5 Anforderungen zu der Inbetriebnahme	19
Tabelle 6 Auflistung der nicht funktionalen Anforderungen	22
Tabelle 7 Beschreibung der Anwendungsfälle	23
Tabelle 8 Qualitätsanforderungen an die «Baustellen App»	25
Tabelle 9 Stakeholder des «Baustellen App» MVPs	25
Tabelle 10 Randbedingungen	26
Tabelle 11 Akteure und Systeme im Kontext der «Baustellen App»	27
Tabelle 12 Beschreibung der System Bausteine	31
Tabelle 13 Beschreibung der Schnittstellen des Gesamtsystems zu den externen Systemen	31
Tabelle 14 Beschreibung der Schnittstellen des «Mobile Client»	32
Tabelle 15 Beschreibung der Schnittstellen des «Web Portals»	33
Tabelle 16 Beschreibung der Schnittstellen des «API Gateways»	34
Tabelle 17 Beschreibung der "Schnittstellen des «Customer Service»	34
Tabelle 18 Beschreibung der «Projekt Service» Schnittstellen	35
Tabelle 19 Beschreibung der «Note Service» Schnittstellen	36
Tabelle 20 Beschreibung der «Commissioning Service» Schnittstellen	37
Tabelle 21 Beschreibung der «Identity Service» Schnittstellen	38
Tabelle 22 Beschreibung der «File Service» Schnittstellen	39
Tabelle 23 Qualitätmaßnahmen während der Entwicklung	48
Tabelle 24 Qualitätmaßnahmen auf der Stufe Build Pipeline	49
Tabelle 25 Beschreibung der Massnahmen auf der Entwicklungsumgebung	50
Tabelle 26 Entwurfsentscheidung: Aufbau des Backends als Micro Service Architecture	50
Tabelle 27 Entwurfsentscheidung: Einsatz von Asynchroner Datensynchronisation	51
Tabelle 28 Entwurfsentscheidung: «Complete Storage» für den mobilen Client	52
Tabelle 29 Varianten für die Datenübertragung zwischen Mobile Client und Backend	53
Tabelle 30 Bewertungskriterien bei der Auswahl des Datentransfermusters	53
Tabelle 31 Bewertung der Komplexität der Varianten	55
Tabelle 32 Bewertung des Speicherbedarfs der Varianten	55

Tabelle 33 Bewertung der Performance der Varianten	56
Tabelle 34 Bewertung der Netzwerkbandbreite der Varianten	56
Tabelle 35 Bewertung der Eignung des Konfliktmanagements der Varianten	56
Tabelle 36 Auflistung und Begründung der Gewichtung der Kriterien	57
Tabelle 37 Ergebnis der Nutzwertanalyse	57
Tabelle 38 Entwurfsentscheidung: Einsatz von Timestamp Transfer für den Datentransfer	58
Tabelle 39 Entwurfsentscheidung: Einsatz von MVVM für die mobile App	58
Tabelle 40 Qualitätsanforderung Offline Synchronisation	59
Tabelle 41 Risikobewertung: Datenverlust während Datenzusammenführung	60
Tabelle 42 Risikobewertung Zuverlässigkeit der «Baustellen App»	60

8 Glossar

Backend

Als backend wird die Softwarekomponente eines verteilten Systems bezeichnet, welche auf einem Server betrieben wird und über ein privates Netzwerk oder das Internet erreichbar ist. Üblicherweise ist das Backend für das Persistieren von Daten zuständig und enthält unter Umständen Geschäftslogik, welche die ankommenden oder ausgehenden Daten entsprechend verarbeitet.

Build Artefakt

Ist das Ergebnis der Code Kompilierung (des Builds). Das Artefakt ist ein lauffähiges Bündel Software, welches auf ein System ausgerollt und dort ausgeführt werden kann.

CI/CD

CI steht für «Continous Integration» und beschreibt das kontinuierliche Integrieren von neuem Code in das zentrale Repository. CD steht für «Continous Deployment» und beschreibt die Strategie, neuen Code oder neue Funktionen kontinuierlich auf die Produktion auszurollen. Diese Strategie wird oft im DevOps Bereich eingesetzt und setzt eine robuste Qualitätssicherung durch Testing und Codeanalyse voraus.

Client/Server System

Ein Client/Server System beschreibt ein Softwaresystem bestehend aus einer Komponente, welche als Backend auf einem Server in der Cloud oder einem Rechenzentrum ausgeführt wird. Die zweite Komponente, der Client, wird verwendet, um Benutzern Zugriff auf die Backendfunktionalität zu ermöglichen. Der Client wird entweder auf einem Rechner installiert oder in einem Browser angezeigt.

CORS

Cross-Origin Resource Sharing ist ein auf HTTP-Headern basierender Mechanismus zur Einschränkung von Sideloaden von Ressourcen. Der Webserver konfiguriert die CORS-Regeln im http Header und teilt damit dem Browser mit, von welchen anderen Domänen Inhalte geladen werden dürfen. Werden keine CORS-Regeln definiert, wenden die meisten Browser eine Standartregel an, welche besagt, dass nur Inhalte von der Originaldomain geladen werden dürfen.

CRUD

CRUD ist ein Akronym für Create, Read, Update, Delete und beschreibt die Basis-Operationen für Datensätze in einer Software-Anwendung oder Datenbank.

Design Patter

Auf Deutsch Entwurfsmuster, ist ein Begriff aus der Softwareentwicklung. Ein Entwurfsmuster ist wie der Name sagt ein Muster, um ein bestimmtes Problem zu lösen. Das Design Pattern beschreibt die Lösung ohne konkrete Implementierungen. Dadurch kann das Entwurfsmuster unabhängig von der Plattform eingesetzt werden.

Domäne/Domain

Eine Domäne oder im englischen «Domain» ist ein eindeutiger Name einer Internet-Ressource. Domänennamen werden verwendet, um Services oder Webseiten im Internet aufrufbar zu machen. Der Domänenname wird, wie in einem Telefonbuch, auf eine Adresse eines bestimmten Webservers aufgelöst.

Frontend

Das Frontend ist der Teil eines Softwaresystems, welches für die Interaktion mit den Anwendern zuständig ist. Es stellt eine Benutzeroberfläche zur Verfügung.

FQDN

Dies ist eine Abkürzung für «Fully Qualified Domain Name». Der Begriff beschreibt den vollständigen und eindeutigen Domänennamen für eine Ressource im Internet.

Micro Service Architektur

In einer Micro Service Architektur wird das System nach Domänen oder Verantwortlichkeiten geschnitten. Jede Domäne oder Verantwortlichkeitsbereich wird als eigenständiger Service, mit eigener Businesslogik und Datenhaltung umgesetzt. Grundsätzlich sollte Jeder Service unabhängig von anderen Services funktionieren.

MVP

Minimum Viable Product ist die englische Beschreibung eines Produkts mit dem minimal nötigen Funktionsumfang, um von Kunden oder Benutzern verwendet werden zu können und Rückmeldung für die zukünftige Funktionsentwicklung zu gewinnen. Dieser Term wird üblicherweise in der Softwareentwicklung verwendet.

Proof of Concept

Ein Proof of Concept oder in kurz PoC ist eine experimentelle Umsetzung eines Konzeptes, um die Machbarkeit des Konzeptes zu beweisen.

Reverse Proxy

Ein Reverse Proxy wird vor die Webserver bzw. im Fall der «Baustellen App» vor die Microservices gesetzt. Der Reverse Proxy nimmt Anfragen von Clients entgegen und leitet sie an die dahinter liegenden Services weiter.

SAD

Das Software Architektur Dokument ist das Ergebnis der Arbeit des Software-Architekten. Das Dokument gibt darüber Auskunft, wie die Qualitätsanforderungen an die Applikation umgesetzt werden sollen. Es dokumentiert die relevanten Architekturentscheidungen, analysiert Risiken, definiert Massnahmen zur Risikominderung und enthält weitere Querschnittsthemen, welche aus Sicht des Architekten für die Applikation von Bedeutung sind.

UI/UX

UI ist die Abkürzung für User Interface. Auf Deutsch: Benutzeroberfläche. UX steht für User Experience und beschreibt die Benutzererfahrung. Beide Begriffe zusammen behandeln das Aussehen und Verhalten von grafischen Oberflächen von Softwaresystemen, welche von menschlichen Anwendern verwendet werden.

Use Cases

Auf Deutsch: Anwendungsfälle. Ein Anwendungsfall beschreibt eine Funktionalität eines Systems, welcher Aktor oder welches System diese Funktionalität verwendet

Literaturverzeichnis

- [1] «What is minimum viable product?», O'Reilly Online Learning. Zugegriffen: 8. März 2025. [Online]. Verfügbar unter: <https://learning.oreilly.com/videos/what-is-minimum/9781492036999/>
- [2] D. G. Starke, «arc42 Template Overview», arc42. Zugegriffen: 12. Juli 2024. [Online]. Verfügbar unter: <https://arc42.org/overview>
- [3] «Data Syncing in Core Data Based iOS Apps | DENIVIP Media». Zugegriffen: 19. Oktober 2024. [Online]. Verfügbar unter: <http://blog.denivip.ru/index.php/2014/04/data-syncing-in-core-data-based-ios-apps/?lang=en>
- [4] A. Selvan, «Set reconciliation in multi-node environment», in *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, Juli 2013, S. 1–4. doi: 10.1109/ICCCNT.2013.6726730.
- [5] S. Gilbert und N. Lynch, «Perspectives on the CAP Theorem», *Computer*, Bd. 45, Nr. 2, S. 30–36, Feb. 2012, doi: 10.1109/MC.2011.389.
- [6] «Was ist das CAP-Theorem? | IBM». Zugegriffen: 25. Januar 2025. [Online]. Verfügbar unter: <https://www.ibm.com/de-de/topics/cap-theorem>
- [7] Z. McCormick, «Data Synchronization Patterns in Mobile Application Design».
- [8] D. Hasanenko, «Data Synchronization Patterns», Medium. Zugegriffen: 1. Juli 2024. [Online]. Verfügbar unter: <https://hasanenko.medium.com/data-synchronization-patterns-c222bd749f99>
- [9] E. Freeman *u. a.*, *Head first design patterns: building extensible & maintainable object-oriented software*, Second edition. Sebastopol, CA: O'Reilly Media, 2020.
- [10] M. Mitzenmacher und R. Pagh, «Simple multi-party set reconciliation», *Distrib. Comput.*, Bd. 31, Nr. 6, S. 441–453, Nov. 2018, doi: 10.1007/s00446-017-0316-0.
- [11] M. T. Goodrich und M. Mitzenmacher, «Invertible Bloom Lookup Tables», 4. Oktober 2015, arXiv: arXiv:1101.2245. doi: 10.48550/arXiv.1101.2245.
- [12] A. Boral und M. Mitzenmacher, «Multi-Party Set Reconciliation Using Characteristic Polynomials», 9. Oktober 2014, arXiv: arXiv:1410.2645. doi: 10.48550/arXiv.1410.2645.
- [13] D. G. Starke, «arc42 Template Overview», arc42. Zugegriffen: 27. Januar 2025. [Online]. Verfügbar unter: <https://arc42.org/overview>
- [14] «Use case diagrams are UML diagrams describing units of useful functionality (use cases) performed by a system in collaboration with external users (actors).» Zugegriffen: 31. Dezember 2024. [Online]. Verfügbar unter: <https://www.uml-diagrams.org/use-case-diagrams.html>
- [15] craigshoemaker, «Schnellstart: Bereitstellen Ihrer ersten Container-App über das Azure-Portal». Zugegriffen: 12. Februar 2025. [Online]. Verfügbar unter: <https://learn.microsoft.com/de-de/azure/container-apps/quickstart-portal>
- [16] «How OpenID Connect Works - OpenID Foundation». Zugegriffen: 10. Februar 2025. [Online]. Verfügbar unter: <https://openid.net/developers/how-connect-works/>
- [17] A. Razzaq und S. A. K. Ghayyur, «A systematic mapping study: The new age of software architecture from monolithic to microservice architecture—awareness and challenges», *Comput. Appl. Eng. Educ.*, Bd. 31, Nr. 2, S. 421–451, 2023, doi: 10.1002/cae.22586.
- [18] «Monolithisch vs. Microservices - Unterschied zwischen Softwareentwicklungsarchitekturen - AWS», Amazon Web Services, Inc. Zugegriffen: 29. Januar 2025. [Online]. Verfügbar unter: <https://aws.amazon.com/de/compare/the-difference-between-monolithic-and-microservices-architecture/>
- [19] «Was ist Serverless Computing? – Serverless Computing erklärt – AWS», Amazon Web Services, Inc. Zugegriffen: 29. Januar 2025. [Online]. Verfügbar unter: <https://aws.amazon.com/de/what-is/serverless-computing/>
- [20] M. Mitzenmacher und G. Varghese, «The complexity of object reconciliation, and open problems related to set difference and coding», in *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Okt. 2012, S. 1126–1132. doi: 10.1109/Allerton.2012.6483345.
- [21] michaelstonis, «Model View ViewModel - .NET». Zugegriffen: 7. Februar 2025. [Online]. Verfügbar unter: <https://learn.microsoft.com/de-de/dotnet/architecture/maui/mvvm>
- [22] «Data Access Object», Wikipedia. 31. Juli 2021. Zugegriffen: 7. Februar 2025. [Online]. Verfügbar unter: https://de.wikipedia.org/w/index.php?title=Data_Access_Object&oldid=214388984

9 Anhang

9.1 Sourcecode

Der Sourcecode zum PoC der «Baustellen App» befindet sich in folgendem Git Repository

<https://github.com/zuercheram/bauestellen-app>

9.2 «Baustellen App» Installationsanleitung für Android



1. Mit dem Android Gerät den QR Code scannen oder folgenden Link auf dem Android Gerät öffnen: <https://accstoragedev.blob.core.windows.net/dev-app-distro/com.isolutions.baustellen.app.client.apk>
2. Auf «Link öffnen» klicken.
3. Android warnt, dass die Datei schädlich sein könnte. Auf «Trotzdem öffnen» klicken.
4. Wenn die Datei heruntergeladen wurde, auf «öffnen» klicken oder die Datei in den Downloads finden. Der Dateiname lautet: «com.isolutions.baustellen.app.apk»
5. Android verhindert eventuell das Installieren von Apps, welche nicht aus dem App Store stammen. In den Einstellungen muss das Installieren von App Store fremden Apps erlaubt werden. Unter Einstellungen > Sicherheit und Datenschutz > Automatische Sperre ausschalten.
6. THEIA Baustellen App installieren.
7. Eventuell muss die App vom Google Play Protect auf Viren gescannt werden.
8. Danach wird die App installiert und kann geöffnet werden.

Folgende Testbenutzer sind angelegt:

Mitarbeiter

Benutzername: field.user@zuercheram1.onmicrosoft.com, Password: Fozu166923

Projektleiter

Benutzername: pl.user@zuercheram1.onmicrosoft.com, Password: Waku534958

Sachbearbeiter

Benutzername: backoffice.user@zuercheram1.onmicrosoft.com, Password: Fozu166923

Im Benutzer Profile Tab auf «Anmelden» klicken und eine der oben genannten Authentifizierungsdaten eingeben.

10 Verwendung von KI

In dieser Arbeit wurde auf die Verwendung von KI verzichtet.

11 Selbständigkeitserklärung

Ich bestätige, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der im Literaturverzeichnis angegebenen Quellen und Hilfsmittel angefertigt habe. Sämtliche Textstellen, die nicht von mir stammen, sind als Zitate gekennzeichnet und mit dem genauen Hinweis auf ihre Herkunft versehen.

Ich bestätige weiterhin, dass ich bei der Erstellung dieser Studienarbeit durchgehend steuernd gearbeitet habe und von einer KI erzeugte Texte bzw. Textfragmente nicht unreflektiert übernommen habe.

Ort, Datum: 3628 Uttigen, 09.03.2025

DocuSigned by:

FAFECDAA4DCDB426...
Amos Zuercher