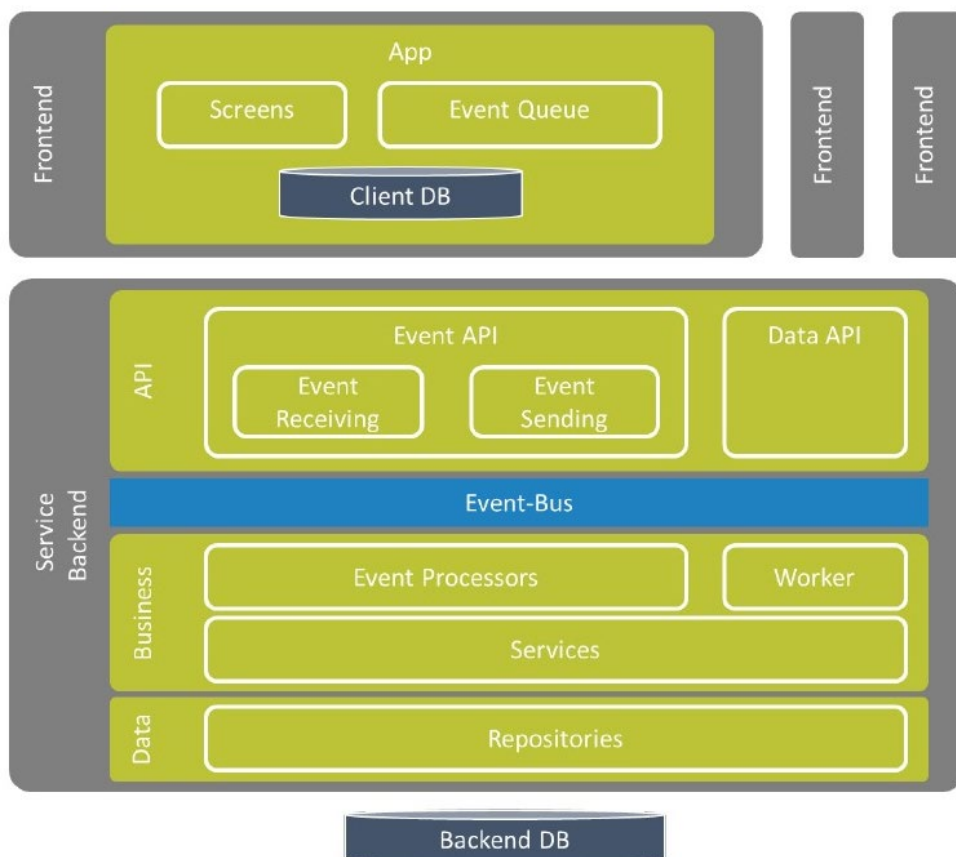


Digitale Quartiersplattform und Energieeffizienz-Applikation

Themenbericht erstellt im Rahmen des Schlussberichts
zum Verbundvorhaben EnStadt:Pfaff
„Implementierung des Reallabors Pfaff-Areal Kaiserslautern“



Digitale Quartiersplattform und Energieeffizienz-Applikation

Themenbericht erstellt im Rahmen des Schlussberichts zum
Verbundvorhaben EnStadt:Pfaff:

„Implementierung des Reallabors Pfaff-Areal Kaiserslautern - Integrierte
Konzepte, innovative Technologien und sozialwissenschaftliche Forschung
im Leuchtturm für klimaneutrale Quartiere“

Teilvorhaben EnStadt:Pfaff: „Fraunhofer EnergieIKT“

Förderndes Ministerium:	Bundesministerium für Wirtschaft und Klimaschutz (BMWK)
Projektträger:	Forschungszentrum Jülich GmbH
Förderkennzeichen:	03SBE112D
Projektlaufzeit:	01.10.2017 – 31.12.2024
Autoren:	Balthasar Weitzel, Patrick Mennig, Phil Stüpfert, Frank Elberzhager, Simon Scherr (IESE)
Ausführende Stelle:	Fraunhofer-Institut für experimentelles Software Engineering IESE Fraunhofer-Platz 1, 67633 Kaiserslautern
Veröffentlicht:	Juli 2025 (Erstveröffentlichung als finaler Meilensteinbericht im November 2022)

Die Verantwortung der Veröffentlichung liegt bei den Autoren.

Das Verbundprojekt EnStadt:Pfaff wurde von 8 Partnern
durchgeführt und als Leuchtturmprojekt gemeinsam gefördert
vom Bundesministerium für Wirtschaft und Klimaschutz (BMWK)
und vom Bundesministerium für Bildung und Forschung (BMBF)

Gefördert durch:



Bundesministerium
für Wirtschaft
und Klimaschutz

aufgrund eines Beschlusses
des Deutschen Bundestages

Gefördert durch:



Bundesministerium
für Bildung
und Forschung

aufgrund eines Beschlusses
des Deutschen Bundestages

Weitere Informationen zu EnStadt:Pfaff: <https://pfaffquartier-klimaneutral.de/>

Konzeptbericht inkl. Erprobungsergebnisse zu AP 2.1.1 im Projekt EnStadt:Pfaff (Meilenstein 3 Bericht)

Digitale Quartiersplattform und Energieeffizienz-Applikation

Autoren:

Balthasar Weitzel
Patrick Mennig
Phil Stüpfert
Frank Elberzhager
Simon Scherr

EnStadt:Pfaff wird gemeinsam gefördert durch



Bundesministerium
für Wirtschaft
und Energie



Bundesministerium
für Bildung
und Forschung

aufgrund eines Beschlusses des Deutschen Bundestags

Version 3.0
30.11.2022

Eine Publikation des Fraunhofer IESE

Zusammenfassung

Das vorliegende Dokument beschreibt die finalen Ergebnisse AP 2.1.1 „Digitale Quartiersplattform und Energieeffizienz-Applikation“. Der Fokus ist dabei auf der Plattform, welche die Grundlage für Dienste für verschiedene Stakeholder darstellt. In dem vorliegenden Bericht wird zunächst ein Überblick über die Plattform gegeben, bevor detailliert die aktuellen Dienste beschrieben werden. Abschließend geben wir eine Übersicht über die Anwendungsentwicklung und beschreiben einige Erprobungsergebnisse. Insbesondere Beispiele für Dienste sind auch im AP 1.1.4 IKT Konzept zu finden, speziell relevant für AP 2.1.1 ist die Energie-App Fish ´n Tipps. Es sei auch auf das Deliverable „Motivationskonzept“ verwiesen, welches im Rahmen von AP 2.1.1 und 2.3.1 entstanden ist.

Schlagworte: EnStadt:Pfaff
Reallabor Pfaff
AP 2.1.1 Abschlussbericht
Quartiersplattform
IKT

Inhaltsverzeichnis

1	Ziele der Plattform.....	11
2	Überblick der Plattform-Architektur.....	13
2.1	Technische Dekomposition der Plattform	14
2.2	Betrieb der Plattform	16
3	Plattform-Dienste	19
3.1	Nutzerverwaltung	19
3.2	Push-Nachrichten-Dienst	19
3.3	Geografische Unterteilung für Anwendungen	20
3.4	Feingranulare Anwendungs-Konfiguration	22
3.5	Anwendungs- und Anwendungs-Versions-Verwaltung	23
3.6	Buchungsverwaltung für Apps durch Städte und Kommunen	24
3.7	Stammdaten-Verwaltung	24
3.8	Blockieren und Melden von Beiträgen	25
3.9	Verwaltung Nutzer-relevanter juristischer Texte	25
3.10	DSGVO-Dienste	25
3.11	Chat-Dienst	27
3.12	Motivations-Dienste	27
3.13	Bilder-Dienst	28
3.14	Leichtgewichtige Statistiken	28
4	Anwendungsentwicklung- und Betrieb auf der Plattform	30
4.1	Entwicklung von Anwendungen	30
4.2	Betrieb von Anwendungen	31
4.3	Nutzung von Anwendungen durch Bürger	32
5	Plattform-Mock.....	34
5.1	Nutzen des Plattform-Mocks	34
5.2	Aufbau des Plattform-Mocks	35
5.3	Verwendung des Plattform-Mocks	36
6	Eine Anwendung am Beispiel der Energieeffizienzapplikation	38
6.1	Kontext und Vorgehensweise	38
6.2	Konzept	39
6.3	Umsetzung	40
7	Evaluation	42

7.1	Feedback-Komponente	42
7.2	Erfahrungen aus der Nutzung des PfaffFunks	43
8	Ausblick	45

1 Ziele der Plattform

Die Digitale Quartiersplattform unterstützt, wie im IKT-Konzept beschrieben, die Erstellung von Anwendungen im Kontext der Digitalisierung des Pfaff-Quartiers. Diese Anwendungen werden zum einen im Projekt entwickelt, sollen aber auch langfristig von externen Partnern zur Verfügung gestellt werden können. Besonders für die Verstetigung der Projektergebnisse nach Ende des Projekts ist die Langlebigkeit der Plattform relevant. Dazu stellt die Plattform Basisdienste zur Verfügung, mit deren Hilfe einfach neue Dienste oder Anwendungen erstellt werden können. Dabei liegt der Fokus stark auf Anwendungen mit direkter Nutzerinteraktion, die beispielsweise als Mobilanwendungen oder Webanwendungen bereitgestellt werden. Aktuell verfügbare Basis-Dienste sind:

- Nutzerverwaltung
 - Anmeldedaten
 - Nutzerprofile
- Push-Nachrichten-Dienst
 - Push-Nachrichten an Mobilgeräte
- Geografische Unterteilung für Anwendungen
 - Gemeinde und Stadtteile Deutschlands
- Feingranulare Anwendungs-Konfiguration
 - Einstellungen für Anwendungen und Anwendungs-Varianten je Gemeinde und Stadtteil
- Anwendungs- und Anwendungs-Versions-Verwaltung
 - Nutzer-Einstellungen für Anwendungen
 - Updates für Anwendungen
- Buchungsverwaltung für Anwendungen durch Städte und Kommunen
 - Verwaltung von Bestellungen von Anwendungen für Nutzer in einem geografischen Bereich
- Stammdaten-Verwaltung
 - Pflege von gemeinsamen, unveränderlichen Daten
- Blockieren und Melden von Beiträgen
 - Blockieren anderer Nutzer auf der Plattform
 - Melden von anstößigen Beiträgen anderer Nutzer
- Verwaltung Nutzer-relevanter juristischer Texte
 - Information der Nutzer
 - Speicherung der Zustimmung
- DSGVO-Dienste
 - Sammeln aller Nutzer-Daten über Anwendungen hinweg
 - Löschen/Anonymisieren aller Nutzer-Daten über Anwendungen hinweg

- Erinnern und Löschen inaktiver Nutzer
- Chat-Dienst
 - Chats für Gruppen von Nutzern
- Motivations-Dienste
 - Umsetzung von Gamification-Konzepten
- Bilder-Dienst
 - Generischer Bildupload
 - Skalierung für verschiedene Geräte und Bandbreiten
- Leichtgewichtige Statistiken
 - Erweiterbare Statistiken über Nutzung von Diensten, aufgeschlüsselt nach geografischem Bereich

Diese Basis-Dienste können von Anwendungen benutzt werden und ermöglichen damit eine Fokussierung auf den eigentlichen Mehrwert des Dienstes. Die Plattform wird in einer Cloud-basierten Umgebung betrieben, die automatisch skaliert und damit eine schwankende Nutzerzahl und Last bedienen kann. Endbenutzer interagieren dabei nicht direkt mit der Plattform, sondern über die Dienste, die auf der Plattform zur Verfügung gestellt werden.

2 Überblick der Plattform-Architektur

Ein Überblick über die konzeptionelle Architektur der Plattform ist in Abbildung 1 dargestellt. Dabei wird grundlegend zwischen Anwendungen auf der Plattform und der Plattform selbst unterschieden. Nutzer interagieren hauptsächlich mit den Anwendungen, nur für querschnittliche Aspekte, wie Anmeldung oder Datenauskunft findet eine direkte Interaktion mit der Plattform statt. Die Anwendungen selbst zerfallen wiederum in einen Frontend- und Backend-Teil. Dabei ist die Aufgabe des Frontends, die direkte Interaktion mit einem einzelnen Nutzer zu ermöglichen. Diese Frontends sind dabei häufig Apps, die auf einem Gerät installiert sind, oder Webanwendungen, die dann im Browser ausgeführt werden. Der Backend-Teil der Anwendungen hingegen hat einen anderen Anspruch. Er sorgt dafür, dass es zu einem geregelten Datenaustausch zwischen den einzelnen Frontends der Anwendung kommt. Am Beispiel einer Chat-Anwendung würden die Frontends der Nutzer die Nachrichten an das Backend schicken, welches dann wiederum diese Nachrichten genau den Frontends zur Verfügung stellt, deren Nutzer darauf Zugriff haben. Zusätzlich sorgen die Backend-Teile der Anwendungen auch für einen Austausch zwischen Anwendungen. Beispielsweise könnte eine Anwendung, die Mitfahrten organisiert, direkt als Bot in einer Chat-Anwendung eingebunden werden.

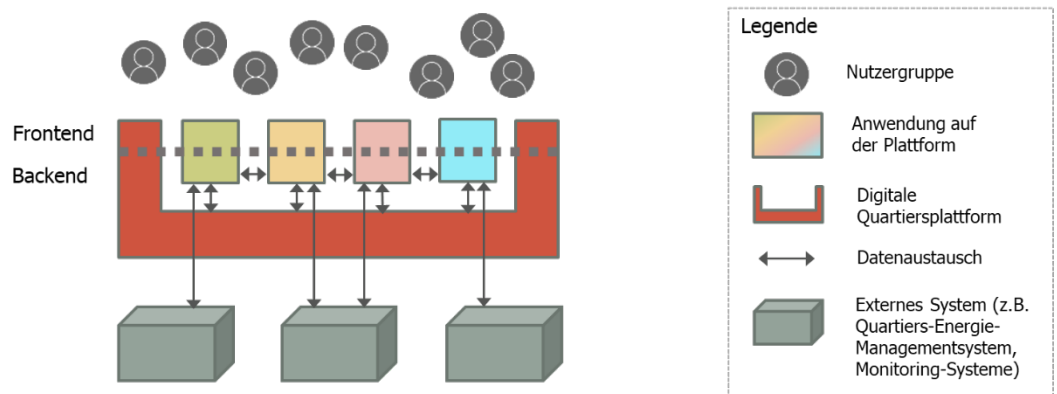


Abbildung 1: Konzeptionelle Architektur der Plattform

Analog zur Unterteilung der Anwendungen gibt es diese auch bei der Plattform. Diese verfügt über eher wenige Frontends, da eine direkte Nutzerinteraktion eher seltener erfolgt, beispielsweise beim Registrieren und Anmelden an der Plattform. Die Hauptaufgabe der Plattform betrifft die Vereinfachung der Backend-Teile der Anwendungen. An dieser Stelle kommen die beschriebenen Dienste der

Plattform zum Einsatz, die dem Anwendungsentwickler einen großen Teil der Komplexität bei der Entwicklung der Anwendung abnehmen.

Die Integration zwischen externen Systemen und Anwendungen auf der Plattform erfolgt bevorzugt über den Backend-Teil der Anwendung selbst. Der Grund dafür ist die individuelle Fachlichkeit der Integration, die nur der Anwendung bekannt ist. Eine generische Integration über die Plattform würde dabei eine Vereinheitlichung erfordern, die es erschwert die konkrete Integration effizient zu gestalten, da alle fachlichen Daten vereinheitlicht werden müssten. Dahin gegen können individuelle Integrationen schneller und effizienter entwickelt werden. Trotzdem gibt es in gewissem Maße eine direkte Interaktion zwischen dem externen System und der Plattform. Diese wird benötigt, um gemeinsame Daten, wie beispielsweise Nutzerinformationen, dem externen System zur Verfügung zu stellen. Auf der anderen Seite muss im Falle einer Anfrage nach einer Datenauskunft oder Datenlöschung durch den Nutzer ebenfalls ein Zugriff von der Plattform aus auf das externe System erfolgen.

2.1 Technische Dekomposition der Plattform

Die Plattform folgt einem ereignisorientierten Datenverarbeitungsparadigma. Dies bedeutet, dass Änderungen an Daten immer als ein Ereignis mit einer bestimmten Semantik gesehen werden. Beispielsweise wäre im Falle einer Logistik-Anwendung die Änderung nicht "Setze Status der Lieferung auf Empfangen", sondern "Empfänger hat Paket erhalten". Dieser Unterschied trägt entscheidend zu einer besseren Verständlichkeit von Anwendungscode bei und ermöglicht ebenfalls eine bedeutend bessere Erweiterbarkeit. Die bessere Verständlichkeit rührt daher, dass die für das Verstehen notwendig Übersetzung des Codes in die Fachlichkeit viel schneller erfolgen kann, wenn diese bereits direkt im Code sichtbar ist, da die Ereignisse immer durch ein fachliches Konzept getrieben sind. Eine Verbesserung der Erweiterbarkeit rührt daher, dass es mit dem Konzept eines Event-Bus, wie er in der Plattform eingesetzt wird, einfach möglich ist, auf das Auftreten bestimmter Ereignisse zu reagieren. Im Beispiel könnte das ein Verschieken einer Benachrichtigung per Email, SMS oder Chat sein, der den Versender über die erfolgreiche Ablieferung der Lieferung informiert. Diese zusätzliche Behandlung des Ereignisses kann dabei komplett ohne Veränderung der ursprünglichen Logistik-Anwendung erfolgen, da nur eine Registrierung am Event-Bus für ein spezielles Event erfolgt. Gerade im Kontext der Digitalisierung des Quartiers, bei dem auch immer wieder neue Konzepte evaluiert und verworfen werden, ist es ein entscheidender Vorteil, dass Erweiterungen so einfach integriert und auch wieder entfernt werden können.

Aus diesen Überlegungen wurde die Plattform wie in Abbildung 2 intern strukturiert. Diese Struktur gilt genauso auch für alle Backend-Teile von Anwendungen auf der Plattform. Frontends orientieren sich passenderweise ebenfalls am

ereignisorientierten Paradigma und behandeln Interaktionen des Nutzers ebenfalls als Ereignisse.

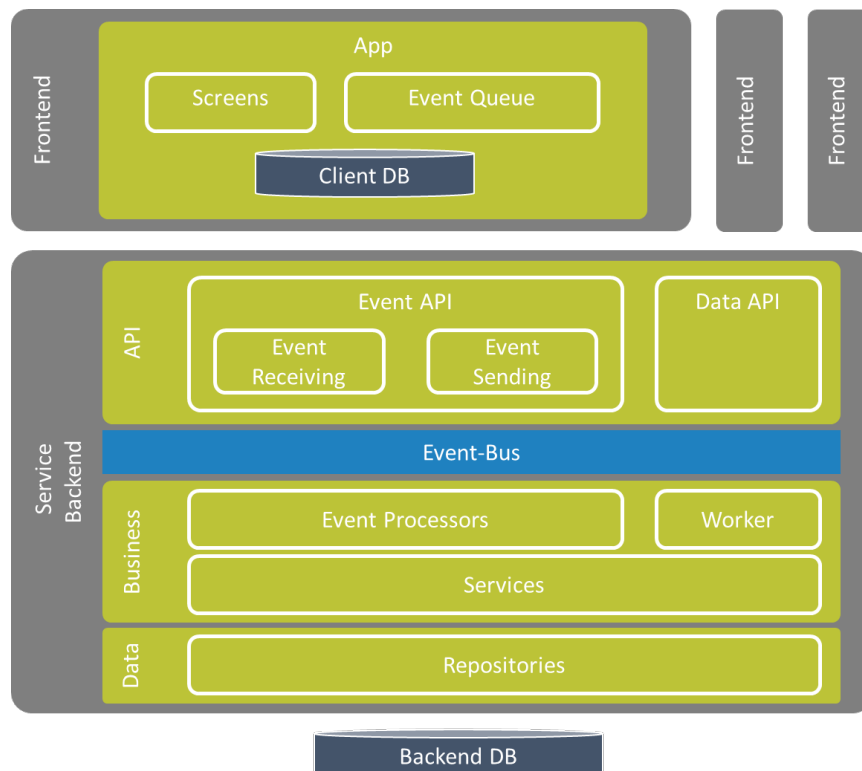


Abbildung 2: Interne Architektur der Plattform

Auf Seite des Backends wird eine REST-API zur Verfügung gestellt, die sich in zwei Teile aufteilt. Zum einen ist das die Event-API, mit der neue Ereignisse angenommen werden. Diese stellt die Authentifizierung und erste Autorisierung sicher und überprüft die Events auf syntaktische Korrektheit. Falls diese erfüllt ist, dann wird das Event auf dem Event-Bus zur Verfügung gestellt.

Alle Event-Prozessoren, die an diesem Typ von Event interessiert sind, werden dann notifiziert und verarbeiten das Ereignis, das heißt sie führen die eigentliche Logik der Anwendung aus. Zur Manipulation von Daten werden wiederum Services verwendet, die für Integrität von zusammenhängenden Datenelementen sorgen. Die Notwendigkeit für diese Services ist, dass häufig mehrere Event-Prozessoren die gleichen Datenelemente manipulieren möchten. Im Beispiel der Logistik-Anwendung wird der Status einer Lieferung durch viele unterschiedliche Ereignisse geändert. Damit nun nicht in jedem beteiligten Event-Prozessor Code geschrieben werden muss, der diese Statusänderung durchführt, wird ein geteilter Service entwickelt, der genau diese Logik zur Änderung des Status enthält.

Der letztendliche Zugriff auf die persistierten Daten erfolgt über Repositories. Diese kapseln die Logik, die zum Ansprechen der Datenbank notwendig ist, und stellen damit klar definierte Schnittstellen zum Abrufen und Speichern von Datenelementen zur Verfügung.

Bestimmte Anwendungsfälle benötigen die periodische Ausführung von Anwendungslogik. Die Plattform selbst hat mehrere solcher Fälle, beispielsweise um der nach DSGVO erforderlichen Pflicht nachzukommen, Nutzerdaten bei Inaktivität zu löschen. Um solche periodischen Ausführungen zu ermöglichen, stellt die Plattform das Konzept von Workern zur Verfügung, die nach einem je nach Worker festgelegten Zeitplan ausgeführt werden. Da die Plattform auf mehreren Maschinen verteilt parallel betrieben wird, ist es notwendig zu garantieren, dass jeder Worker nur einmal ausgeführt wird. Dazu wählt die Plattform eine Maschine als führenden Knoten aus, der dann diese Worker ausführt. Wird diese Maschine dann im Rahmen von Lastskalierung abgeschaltet, so erkennen die anderen Maschinen dies und wählen unter sich einen neuen führenden Knoten aus.

Eine zusätzliche Funktionalität, die auf dem Konzept der Worker aufbaut, ist es, bei Bedarf Ereignisse zu einem bestimmten Zeitpunkt auszulösen. Dazu stellt ein Event-Processor dieses zukünftige Ereignis mit einem gewünschten Zeitpunkt ein. Sobald dann der Zeitpunkt gekommen ist, wird diese Ereignis auf dem Event-Bus zur Verfügung gestellt. Am Beispiel der Logistik-Anwendung könnte das eine Erinnerung sein, dass eine Lieferung abgeholt werden muss.

2.2 Betrieb der Plattform

Die Plattform wird zum Zeitpunkt der Erstellung dieses Dokuments mit Cloud-Diensten betrieben. Dabei werden insbesondere fertig benutzbare Cloud-Dienste eingesetzt, die wenig Aufwand für den Betrieb erfordern. In Abbildung 3 sind die wichtigsten Dienste dargestellt und die Verteilung der technischen Bestandteile der Plattform auf diese Dienste gezeigt. Der Kernbestandteil der Plattform, das Backend, wird auf AWS Elastic Beanstalk betrieben. Dies ist ein Managed Service, der es erlaubt, Anwendungen auf beliebig vielen Rechnern automatisch zu verteilen. Dabei werden einkommende Anfragen automatisch auf alle diese Rechner verteilt. Steigt die Last, so werden automatisch neue Rechner gestartet, sinkt sie, so werden diese wieder heruntergefahren. Zusätzlich ist es damit möglich unterbrechungsfrei eine neue Version der Plattform zur Verfügung zu stellen. Diese wird dann auf einem Rechner installiert, überprüft ob die Anfragen an die neue Version der Plattform erfolgreich sind, und erst dann alle Rechner mit der neuen Version versorgt. Im Fehlerfall wird automatisch die alte Version der Plattform wieder installiert. Dateien wie beispielsweise Bilder, werden auf AWS S3 abgelegt, einem skalierbaren Speicherdienst. Die Datenbanken, auf die die Plattform zugreift, werden ebenfalls über einen Cloud-Dienst zur Verfügung gestellt. Dieser erledigt übliche Aufgaben der Datenbank-Administration wie das Anlegen von Backups und Installieren von neuen Versionen automatisch.

Bestimmte Dienste werden zusätzlich von der Plattform benötigt. Dies betrifft beispielsweise die Authentifizierung der Nutzer: Dazu wird der auf dem OAuth2-Standard basierende Dienst Auth0 benutzt, der die Verwaltung der Authentifizierungs-Informationen der Nutzer übernimmt. Das Senden von Push-Nachrichten an Mobilgeräte muss letztendlich immer über herstellerspezifische Dienste erfolgen. Um hier Komplexität bei der Anbindung dieser unterschiedlichen Dienste zu vermeiden, wird der Cloud-Dienst AWS PinPoint genutzt, der diese spezifischen Dienste wiederum kapselt. In manchen Fällen ist es notwendig, zu einer Adresse ihre geografischen Koordinaten herauszufinden. Dazu greift die Plattform auf die Google Maps API zu. Zum Versand von SMS wird der Dienst Twilio angesprochen. Mails wiederum werden über einen bei Fraunhofer IESE betriebenen Mailserver verschickt. Dort wird auch eine Own-Cloud-Instanz zur internen Ablage von Statistiken und ein Git-Server zur Ablage der Konfigurationsdaten betrieben.

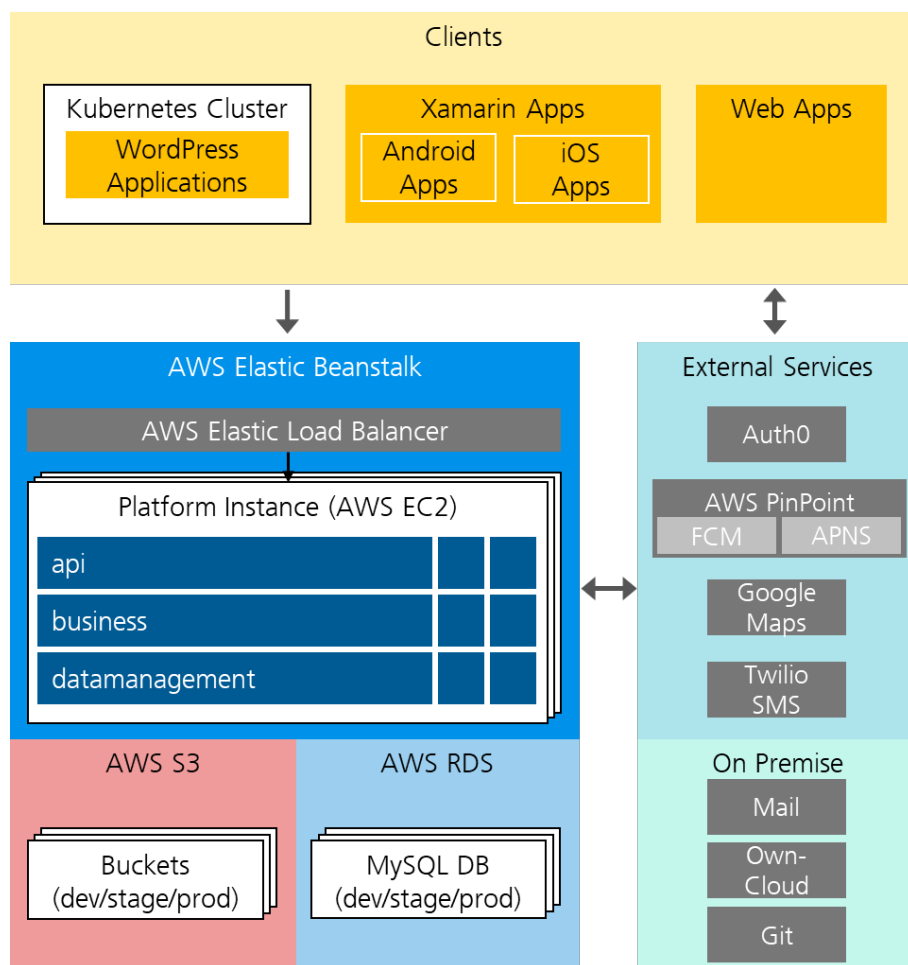


Abbildung 3: Betrieb der Plattform mit Cloud- und On Premise-Diensten

Diese Dienste sind intern immer gekapselt, so dass ein Austausch des Diensteanbieters ohne größere Änderungen an der Plattform möglich ist. Für einige Dienste existieren bereits mehrere Implementierungen, beispielsweise können unterschiedliche Dateiablagen angebunden werden.

Die benutzten Cloud-Dienste werden bevorzugt in Deutschland betrieben; ist diese nicht möglich, so wird zumindest ein Betrieb innerhalb der EU vorgeschrieben.

3 Plattform-Dienste

Die Plattform stellt eine Reihe an Basis-Diensten zur Verfügung, die von Anwendungen auf der Plattform genutzt werden können. Da die Plattform bereits in anderen Projekten erfolgreich produktiv eingesetzt wird, sind die Anforderungen der Basis-Dienste durch konkrete Anwendungsfälle entstanden. Damit ist gewährleistet, dass neue Anwendung, die auf der Plattform entwickelt werden, auf bereits in der Praxis erprobte Dienste zugreifen können, die auch direkt einen Zusatznutzen bringen. Das Ziel dabei ist es, dass sich Anwendung primär auf ihren fachlichen Nutzen konzentrieren können und von technischer Komplexität weitgehend entlastet werden.

3.1 Nutzerverwaltung

Ein wichtiger Aspekt der Plattform ist es, Benutzern einfach zu ermöglichen zwischen Anwendungen auf der Plattform zu wechseln und so deren Synergien zu nutzen. Daher ist eine einheitliche Identifikation der Nutzer notwendig, zusammen mit einem geteilten Benutzerprofil über alle Anwendungen der Plattform hinweg.

Dazu stellt die Plattform einen einheitlichen Registrierungsmechanismus zur Verfügung. Die Plattform bietet dazu aktuell verschiedene Möglichkeiten der Registrierung an: Zum einen kann der Nutzer sich mit Email-Adresse und einem selbst gewählten Passwort registrieren, zum anderen kann ein bestehender Facebook, Google oder Apple-Account verwendet werden. Die in solchen Fällen üblichen Funktionalitäten wie "Passwort vergessen" und "Email-Adresse verifizieren" stellt dabei die Plattform zur Verfügung.

Nach erfolgreicher Registrierung kann ein Nutzer alle Anwendungen auf der Plattform einheitlich benutzen. Die Daten zwischen den Anwendungen haben damit einen eindeutigen Bezug zu dem Nutzer und können somit einfach zwischen den Anwendungen ausgetauscht werden.

3.2 Push-Nachrichten-Dienst

Damit die Nutzer stets über die neusten Chatnachrichten, Veröffentlichungen und weiteren aktuellen Ereignisse innerhalb einer Anwendung informiert werden können, werden diese per Push-Nachrichten-Dienst an die jeweiligen Anwendungen auf dem Gerät des Nutzers gesendet. Dies dient auch dazu alle Anwendungen auf dem aktuellsten Stand zu halten, auch wenn der/die jeweilige Nutzer/in die Anwendung nicht aktiv benutzt und somit die Daten aktiv anfragt. Bei

der nächsten Benutzung muss dann der Inhalt nicht erneut geladen werden, da der Datenbestand bereits auf dem aktualisiert ist.

Pushnachrichten werden nicht immer automatisch als eine Pop-Up-Benachrichtigung angezeigt. Wenn der/die Nutzer/in dies für eine bestimmte Kategorie von Nachrichten nicht möchte, können diese Pop-up-Benachrichtigungen für alle seine/ihre Geräte deaktiviert werden.

Damit die Plattform weiß, welcher Nutzer über welches Endgerät erreichbar ist, meldet sich die jeweilige Anwendung beim ersten Start mit der Nutzer-ID und einer vom Gerät zur Verfügung gestellten Installations-ID an. Dies ermöglicht es auch, dass ein/e Nutzer/in gleichzeitig mit mehreren Geräten angemeldet sein kann und alle Geräte diese Push-Nachrichten erhalten können. Soll eine Nachricht versendet werden, wird auf Seiten der Plattform zunächst entschieden, wer (welche Nutzer-ID) diese Nachricht erhalten soll und in welcher Form (z.B. als Pop-Up-Benachrichtigung) diese angezeigt werden soll. Anschließend wird die Nachricht an alle Geräte für die jeweiligen betroffenen Nutzer in der richtigen Hersteller-spezifischen Form versendet.

3.3 Geografische Unterteilung für Anwendungen

Die Anwendungen, die auf der Plattform den Nutzern zur Verfügung gestellt werden, profitieren stark von der Lokalität ihrer angebotenen Inhalte. Das bedeutet für den Nutzer, dass genau entsprechend seines örtlichen Lebensumfeldes eine Selektion der Dienst-Inhalte vorgenommen wird. Innerhalb der Plattform wird die Gemeindestruktur Deutschlands abgebildet und bis auf Stadtteil-Ebene verfeinert. Diese räumlichen Bereiche werden hier Geo-Areas genannt, die in einer Baumstruktur angeordnet sind.

Jede Geo-Area erhält einen menschenlesbaren Namen, eine passende Typ-Beschreibung, den geographischen Bereich als standardisiertes GeoJSON und die in diesem Bereich vorkommenden Postleitzahlen.

3.3.1 Werkzeug zur Definition der geografischen Bereiche

Zur Erstellung dieser Geo-Areas wurde ein zusätzliches Werkzeug entwickelt. Damit können effizient verschiedene Datenquellen abgefragt werden, aus denen die Geo-Areas dann erstellt werden. Dazu können für neu zu erzeugende Geo-Areas dieser Informationen angegeben:

Regionalschlüssel: Dieser Schlüssel wird vom Statistischen Bundesamt für alle Gemeinden und Städte in Deutschland vergeben und identifiziert diese eindeutig. Die Tiefe dieser Strukturierung ist jedoch je nach Bundesland sehr unterschiedlich. Beispielsweise sind in Rheinland-Pfalz auch recht kleine Gemeinden abgebildet, in Nordrhein-Westfalen sind diese deutlich größer.

WikiData-Identifizier: WikiData ist eine Art maschinenlesbare Version von Wikipedia. Es existieren sehr viele Einträge, die Begriffe und Entitäten durch Attribute beschreiben. Damit kann beispielsweise für eine Gemeinde deren Typ, Postleitzahlen oder geografische Angaben abgefragt werden.

OpenStreetMap Relation oder Weg: OpenStreetMap enthält für sehr viele Gemeinden, Städte und Stadtteile unter anderem die geografischen Umrisse. Dabei wird technisch zwischen Relationen (komplexe zusammengesetzte Strukturen) und Wegen (einfache Strukturen).

Mit Hilfe des Werkzeugs können dann automatisiert verschiedenste Datenquellen abgefragt werden, um so alle relevanten Informationen für die Geo-Areas zu erhalten. Diese Quellen sind:

Statistisches Bundesamt: Das statistische Bundesamt veröffentlicht regelmäßig eine Übersicht aller Länder, Städte und Gemeinden in Deutschland. Diese enthalten eine Zuordnung von Regionalschlüssel zu Namen, Typ und geografischen Mittelpunkt.

Bundesamt für Kartographie und Geodäsie: Von diesem Amt werden regelmäßig die offiziellen geografischen Umrisse aller Länder, Städte und Gemeinden in Deutschland veröffentlicht. Diese stehen als GIS-Export (Shape Files) zur Verfügung.

OpenStreetMap: Über eine spezielle API (Overpass Turbo) kann von OpenStreetMap zu Relationen und Wegen die geografischen Informationen abgerufen werden. Mit diesen werden beispielsweise Umrisse von Gemeinden und Stadtteilen beschrieben.

Geographisches Postleitzahlen-Verzeichnis: Private Anbieter haben aus offiziellen Verzeichnissen der Post die geografischen Umrisse der Postleitzahlen-Bereiche extrahiert und stellen diese als GIS-Export (Shape Files) zur Verfügung.

Google Maps: Anfragen nach Ortsnamen können von Google Maps unter anderem mit der Angabe des geografischen Mittelpunkts beantwortet werden.

WikiData: Je nach Qualität des Eintrags kann zu einem Ort dessen offizieller Name, Typ, geografischer Mittelpunkt und Verweise auf dessen geografische Definition bei OpenStreetMap herausgefunden werden.

Das entwickelte Werkzeug befragt in der passenden Reihenfolge alle genannten Datenquellen, bis es alle Informationen zu der neu zu erstellenden Geo-Area gefunden hat. Damit können effizient neue Städte und Gemeinden zur Plattform hinzugefügt werden.

3.3.2 Benutzung der geografischen Bereiche in Anwendungen

Konzeptionell wird innerhalb der Plattform zwischen Anwendungen und Anwendungs-Varianten unterschieden. Dabei ist eine Anwendung das generische Konstrukt, eine Anwendungs-Variante eine konkrete Instanziierung. Beispielsweise ist eine Anwendung „DorfFunk“, eine Variante davon „PfaffFunk“, der im Rahmen des Projektes erstellt wurde. Diese Unterteilung ist nötig um möglichst effizient Konfigurationen und Schnittstellen zu realisieren.

Somit greift ein Nutzer auf eine Anwendung der digitalen Quartiersplattform über eine konkrete Anwendungs-Variante zu. Diese kann entweder eine App sein, die auf einem Gerät installiert wurde, oder eine Webanwendung. Bei der ersten Nutzung einer Anwendung auf der Plattform bekommt er alle Geo-Areas zur Auswahl und selektiert seinen Heimatort. Bei der ersten Benutzung einer konkreten Anwendungs-Variante bekommt er alle Geo-Areas, die dieser Variante zugeordnet sind, zur Auswahl. Daraus wählt er Geo-Areas als Interessensgebiete für diese Anwendung. Bei der Auswahl der geografischen Gebiete, an denen der Benutzer interessiert ist, werden auch alle in der Hierarchie darüber liegenden Gebiete ausgewählt. Dies ermöglicht es, Inhalte zu verteilen, die für alle Benutzer innerhalb eines größeren geografischen Bereichs relevant sind.

Inhalte, die in der Anwendung erstellt werden, sind bestimmten Geo-Areas zugeordnet. Diese Zuordnung bestimmt so, welche Inhalte welcher Nutzer zu sehen bekommt. Somit bekommt der Nutzer regional zugeschnittene Inhalte, die die Interessen widerspiegeln.

3.4 Feingranulare Anwendungs-Konfiguration

Die Anwendungen und Anwendungs-Varianten müssen oft je nach lokalen Anforderungen im Detail leicht unterschiedliches Verhalten aufweisen. Das reicht von einfachen Namensänderungen hin zu Änderungen in den Funktionen oder komplettes Aktivieren und Deaktivieren von Anwendungsbereichen.

Diese Arten von Änderungen wurden, wie in dem Bereich üblich, als Features gekapselt. Ein Feature hat dabei eine eindeutige Identifikation und kann dabei aktiv oder inaktiv sein und bestimmte Werte enthalten.

Je nach Wunsch der Mandanten können somit die verschiedenen Features der Anwendung oder der Plattform geändert werden. Ein Beispiel ist die Möglichkeit, einen "Sag's uns" Beitrag im DorfFunk zu erstellen oder die im DorfFunk verwendete Farbgebung (beispielsweise für den PfaffFunk). Eine Konfiguration ist damit auf den unterschiedlichsten Ebenen möglich:

- Allgemein für die Plattform
- Allgemein für eine Anwendung und somit für alle ihre Varianten

- Insgesamt für eine Anwendungs-Variante
- Für eine spezifische Geo-Area (Gemeinde, Stadtteil, ...)
- Für einen Hierarchie von Geo-Areas (Bundesland, Landkreis, ...)

Diese Ebenen können auch kombiniert werden, so dass beispielsweise für eine Anwendungsvariante in einer Geo-Area ein bestimmtes Feature geändert werden kann. Damit lassen sich auch effizient feingranulare Konfigurationen abbilden.

Bei ihrem Start fragt jede Anwendung die Plattform nach der aktuellen Feature-Konfiguration für den aktuell angemeldeten Benutzer in dieser Anwendungs-Variante. Dieser Benutzer hat eine bestimmte Heimatgemeinde angegeben. Die entsprechende Konfiguration für diese Geo-Area und Anwendungs-Variante wird zurückgegeben. Dazu wird mittels eines effizienten Algorithmus die gültige Konfiguration interferiert und angewendet.

Auch interne Plattform-Dienste werden zentral über solche Features gesteuert. Damit wird beispielsweise die E-Mail-Verifikation gesteuert. So kann je nach geografischem Bereich diese aktiviert oder deaktiviert werden.

3.5 Anwendungs- und Anwendungs-Versions-Verwaltung

Wie bereits angesprochen wird konzeptionell innerhalb der Plattform zwischen Anwendungen und Anwendungs-Varianten unterschieden. Dabei ist eine Anwendung das generische Konstrukt, eine Anwendungs-Variante eine konkrete Instanziierung. Beispielsweise ist eine Anwendung „DorfFunk“, eine Variante davon „PfaffFunk“, der im Rahmen des Projektes erstellt wurde. Diese Unterteilung ist nötig um möglichst effizient Konfigurationen und Schnittstellen zu realisieren.

Somit finden sich Anwendungs-Varianten beispielsweise in App-Stores, die dann von den Nutzern bezogen werden können. Bei einer Vielzahl von Anwendungen und Anwendungs-Varianten wird deren effizientes Management wichtiger. Ein Aspekt dabei ist Software-Updates. Diese werden immer für Anwendungs-Varianten erstellt und herausgegeben. In dem Fall wird beispielsweise in den App-Stores eine neue Version dieser Anwendungs-Variante zur Verfügung gestellt.

Bei größeren Änderungen kann es vorkommen, dass eine bestehende Version einer Anwendungs-Variante nicht mehr kompatibel zu einer neuen Version der Plattform oder des Anwendungs-Backends ist. Damit ein Nutzer, der noch nicht auf die neueste Version aktualisiert hat, in diesem Fall nicht plötzlich eine fehlerhafte App hat, überprüfen diese regelmäßig, ob ihre jeweilige Version noch unterstützt wird. Ist dies nicht der Fall, so wird der Nutzer darauf hingewiesen, dass ein Update notwendig ist. Um diese Überprüfung zu ermöglichen, stellt die Plattform Funktionalität zur Verfügung, mit der die jeweils unterstützen Versionen der Anwendungs-Varianten konfiguriert und abgerufen werden können.

Zusätzlich hat jede Anwendungs-Variante, die auf der digitalen Quartiersplattform entwickelt wurde, die Möglichkeit für den jeweiligen Nutzer Daten auf der Plattform zu Speichern. Dabei handelt es sich um zweistufige Schlüssel-Wert-Paare, die nur von der Anwendung interpretiert werden. So kann leichtgewichtig Funktionalität in einer Anwendung realisiert werden, ohne dazu direkt einen entsprechenden Backend-Teil entwickeln zu müssen. Diese Daten sind ebenfalls direkt in der Datenauskunft und -Löschung nach DSGVO enthalten.

3.6 Buchungsverwaltung für Apps durch Städte und Kommunen

Innerhalb der digitalen Quartiersplattform müssen verschiedene Organisationseinheiten unterschieden werden können. Beispielsweise könnte die Stadt Kaiserslautern Interesse an der Plattform haben und diese für verschiedene Stadtquartiere einsetzen wollen — gleichzeitig könnte ebenso eine andere Stadt die Plattform nutzen wollen. Anstatt nun eine vollständige Kopie des Systems zu erstellen und beide im Parallelbetrieb laufen zu lassen, können die zwei Städte stattdessen als sogenannte Mandanten angelegt werden. Diese Mandanten buchen Anwendungen auf der Plattform für bestimmte geografische Bereiche.

Die Plattform stellt dazu Funktionalität zur Verfügung, um diese Buchungen abzubilden. Die Anwendungen brauchen dabei keinerlei Wissen über Buchungen zu haben, sie können einfach alle aktuell verfügbaren geografischen Bereiche abrufen. Damit ist es aus Sicht der Anwendung möglich zu entscheiden, ob ein Nutzer eine Anwendung verwenden kann oder nicht. Aus Sicht des Mandanten können alle getätigten Buchungen eingesehen werden. Somit ist die Konfiguration der Plattform und der Anwendungen mit der eigentlichen Buchung verbunden und leichter handhabbar.

3.7 Stammdaten-Verwaltung

Stammdaten sind alle Daten, die nicht direkt durch Nutzerinteraktionen erstellt werden, die aber den Kontext für alle Nutzer setzen. Beispiele sind Mandanten, Gemeinden und Quartiere, Anwendungs-Varianten, Buchungen von Anwendungen oder Feature-Konfigurationen. Diese Daten können auch Bilder beinhalten oder wirken sich auf extern genutzte Dienste aus, wie beispielsweise den Push-Dienst. Um diese Daten auch nachverfolgbar zu verwalten, bietet die Plattform die Möglichkeit, diese in einem stark strukturierten Ordner- und Dateiformat abzulegen. Dabei können Daten in unterschiedliche Bereiche, Umgebungen und Mandanten getrennt definiert werden. Diese werden in ein Git-Repository versioniert und können dann von der Plattform direkt im laufenden Betrieb gelesen werden. Dadurch werden die Stammdaten aktualisiert, ohne dass der Nutzer dabei beeinträchtigt wird. Da die Datenelemente Abhängigkeiten zwischen einander haben, ist es notwendig diese in einer bestimmten Reihenfolge zu aktualisieren oder zu erstellen. Um dies zu ermöglichen werden diese Abhängigkeiten von der Plattform analysiert und für die jeweilige gerade ausgeführte Initialisierung

der Daten eine Ausführungs-Reihenfolge berechnet. Somit können neue Stammdaten-Elemente einfach hinzugefügt werden und es kann auch nur eine Teilmenge der Daten aktualisiert werden.

Um eine klare Nachverfolgbarkeit von Änderungen zu haben wird jedes Einspielen von Stammdaten in die Plattform protokolliert. Dadurch kann genau bestimmt werden welcher Administrator zu welcher Zeit eine Änderung durchgeführt hat.

3.8 Blockieren und Melden von Beiträgen

Nicht immer sind alle Beiträge aller Nutzer frei von problematischen oder verletzenden Äußerungen. Um diese Beiträge schnell aufzufinden und zu entfernen, können alle Nutzer für sie sichtbare Beiträge als anstößig markieren. Der Inhalt wird dann von einem Administrator überprüft und möglicherweise gelöscht, und der Urheber kann gesperrt werden. Markierte Inhalte werden für den Nutzer, der sie markiert hat, ausgeblendet.

Es ist einem Nutzer ebenso möglich, andere Nutzer der Plattform zu blockieren. Deren Inhalte werden dann diesem Nutzer nicht mehr angezeigt, ohne dass der blockierte Nutzer ganz von der Plattform ausgeschlossen wird. Ein Benutzer kann jeden anderen Benutzer blockieren und sieht somit keine Inhalte, die von den blockierten Benutzern verfasst wurden. Dazu gehört auch die Möglichkeit, Chatnachrichten zu schreiben.

3.9 Verwaltung Nutzer-relevanter juristischer Texte

Um die Plattform und darauf aufbauende Dienste zu nutzen, müssen Nutzer bestimmten Vereinbarungen zustimmen. Dazu gehören beispielsweise die Nutzungsbedingungen und Datenschutzerklärung der Plattform und Dienste. Daher muss vor der Erstinutzung der Plattform und bei Änderungen von juristischen Texten, dem/der Nutzer/in zuvor diese zur Verfügung gestellt werden. Erst nachdem der/die Nutzer/in alle notwendigen Texte akzeptiert, darf die Plattform und die Anwendung genutzt werden. Diese Zustimmung wird zentral gespeichert und kann von jeder Anwendung abgerufen werden. Bei jedem Start einer Anwendung wird somit zunächst überprüft, ob der Nutzer den aktuellen relevanten Bestimmungen zugestimmt hat. Dadurch wird sichergestellt, dass keine Nutzung ohne Zustimmung zu den notwendigen Vereinbarungen möglich ist.

3.10 DSGVO-Dienste

Jede/r Nutzer hat das Recht, sämtliche Daten, welche in der Plattform und von den Diensten auf der Plattform von ihr/ihm gespeichert werden, anzufordern. Dabei sollen die Daten sowohl für Menschen verständlich als auch maschinenlesbar geliefert werden. In jeder Anwendung auf der Plattform ist dazu die Möglichkeit eingebaut, diese Anforderung auszulösen. Dabei kann die Anwendung

einfach auf die gemeinsame Funktionalität der Plattform zugreifen. Wird eine Datenanforderung empfangen, werden diese für die/den betreffende/n Benutzer/in zunächst aufbereitet und in eine leserlichere Form gebracht, sodass der Empfänger keine Schwierigkeiten hat, diese zu verstehen. Innerhalb weniger Minuten werden diese Daten dann per E-Mail mit einem Downloadlink zur Verfügung gestellt. Intern kann dabei jeder Dienst, der auf der Plattform realisiert ist, eigene Erweiterungen in diese Datenauskunft integrieren. Da beispielsweise der PfaffFunk die Möglichkeit bietet, Beiträge von Nutzern erstellen zu lassen, hat dieser Dienst eine Erweiterung der Datenauskunft zur Verfügung gestellt, die für einen Nutzer seine selbst geschriebenen Beiträge identifiziert und in verständlicher Art dem Nutzer präsentiert.

Analog zur Datenauskunft hat auch jeder Nutzer die Möglichkeit, eigene Daten löschen zu lassen. Dazu bietet die Plattform ebenfalls eine Funktionalität an, mit der alle Dienste auf der Plattform diese Anforderung erfüllen können. Nach erfolgreicher Löschung wird ein Bericht über die Löschung erstellt. Der innerhalb der Plattform genutzte Mechanismus entspricht dem der Datenauskunft, das heißt Dienst-spezifische Erweiterungen können registriert werden, die dann wieder die Daten des Nutzers entsprechend löschen.

Eine weitere Anforderung aus der DSGVO bezieht sich auf die Lebensdauer von nicht genutzten persönlichen Daten. Diese sollen automatisch nach einer gewissen Zeit entfernt werden. Diese bedeutet konkret, dass Nutzer, die schon eine längere Zeit die Dienste der Plattform nicht mehr genutzt haben, erst notifiziert werden und dann deren Daten gelöscht werden. Ein Dienstanbieter auf der Plattform braucht sich darum nicht zu kümmern, die Plattform erfasst die Nutzung des Dienstes durch den Nutzer und notifiziert sie/ihn automatisch, sobald eine eingestellte Frist der Inaktivität des Nutzers überschritten wird. In dem Fall kann der Nutzer rechtzeitig die Löschung des Kontos verhindern. Falls nicht, dann wird die Löschung der Daten, wie sie oben beschrieben ist, angewandt und der Nutzer darüber informiert.

Auch Anwendungen, die an die Plattform angebunden sind aber extern betrieben werden, sind in diese DSGVO-Prozesse eingebunden. Die Plattform erkennt die Nutzung einer solchen Anwendung durch einen Nutzer und speichert diese Information. Sobald nun entweder eine Datenauskunft oder eine Datenlöschung dieses Nutzers angestoßen wird, so wird auch diese Anwendung über eine definierte Schnittstelle informiert. Diese schickt dann entweder die angeforderten Daten an die Plattform, wo sie in einen Bericht integriert werden, oder löscht alle personenbezogenen Daten dieses Nutzers. Dadurch brauchen sich Anwendungen, die an die Plattform angeschlossen sind, nicht explizit um diese Prozesse zu kümmern.

3.11 Chat-Dienst

Da verschiedene Dienste auf der Plattform die Möglichkeit von Chats zwischen Nutzern anbieten, stellt die Plattform diesen Dienst als generische Lösung zur Verfügung. Anwendungen können dann diese Lösung selbst einsetzen, ohne größeren Implementierungsaufwand zu investieren. Die Grundidee ist ein generischer Dienst, welcher die komplette Funktionalität eines Chats für Anwendungen bereitstellt und somit die direkte Kommunikation zwischen mindestens zwei Personen ermöglicht. Dabei ist der Chat-Dienst unabhängig von der eigentlichen Funktionalität der Anwendung und kann so direkt integriert werden. Der Dienst unterstützt das Versenden von Text-Nachrichten, Bildern, eine „Nachricht-Gelesen-Funktion“, sowie Funktionen zum Löschen und Bearbeiten von Nachrichten. Zusätzlich können anwendungsspezifische Links zu Entitäten innerhalb der Anwendung verschickt werden. Beispielsweise kann im PfaffFunk der Link zu einem Beitrag verschickt werden, so dass der empfangende Nutzer direkt zu diesem Beitrag navigieren kann. Weiterhin ist der Chat-Dienst an die Infrastruktur der Datenauskunft und Löschung angebunden, so dass die Anwendung hier keinerlei zusätzliche Arbeiten erledigen muss.

3.12 Motivations-Dienste

Um Nutzer zu motivieren die Anwendungen auch über einen längeren Zeitraum zu nutzen, haben sich Gamification-Konzepte als eine geeignete Möglichkeit herausgestellt. Diese sind insbesondere dann auch besonders motivierend, wenn sie über mehrere Anwendungen hinweg Verbindungen schaffen. Aus diesem Grund wurden innerhalb der Plattform zwei erweiterbare Gamification-Konzepte umgesetzt.

Zum einen können Nutzer für Handlungen innerhalb einer Anwendung Auszeichnungen (Achievements) bekommen. Diese werden in den Anwendungen durch grafische Symbole dargestellt, die erst sichtbar werden, sobald diese Auszeichnung erreicht wurde. Jede Anwendung kann dabei eigene Auszeichnungen definieren. Dabei werden die grafischen Elemente als auch die Logik, nach denen diese Auszeichnung vergeben wird, festgelegt. Die Plattform integriert dann automatisch alle Auszeichnungen und überprüft wann sie erfüllt wurde.

Das andere umgesetzte Konzept ist eine leichtgewichtige digitale Währung (DigiTaler), die zwischen Nutzern ausgetauscht werden können. Dadurch kann beispielsweise Nachbarschaftshilfe in gewissem Rahmen digital honoriert werden. Auch hier ist der dazugehörige Dienst Anwendungs-unabhängig. Die jeweilige Anwendung realisiert die Logik, nach der die Buchungen erfolgen soll, die Plattform wickelt diese Buchungen ab und informiert die beteiligten Nutzer.

Auch der Motivations-Dienst ist komplett an die Infrastruktur der Datenauskunft und Löschung angebunden.

3.13 Bilder-Dienst

Die verschiedensten Basis- und höherwertige Dienste der digitalen Quartiersplattform erlauben es Nutzern selbst Inhalte zu generieren, beispielsweise Beiträge in Textform. Diese nutzergenerierten Inhalte können auch Bilder sein. Der Bilder-Dienst erlaubt das Speichern und Verwenden von Bildern über die verschiedenen Plattform-Dienste hinweg. Jedes Bild erhält eine eindeutige Identifikationsnummer. Bilder werden dabei nur mit einem einzigen anderen Inhalt verknüpft, so dass beim Löschen des Inhalts auch die Bilder einfach gelöscht werden können.

Der Bilder-Dienst nimmt Bilder der Nutzer entgegen und stellt sie dann innerhalb der Plattform zur Verfügung. Es können beliebige Bilder hochgeladen werden. Da hierbei mitunter größere Datenmengen anfallen dauert der Upload je nach verfügbarer Bandbreite ggf. länger. Dabei besteht das Risiko, dass Bilder nicht vollständig hochgeladen werden, wenn die Verbindung abbricht (z. B. durch ein Funkloch oder beim Verlassen des heimischen WLANs). Deshalb speichert der Bilder-Dienst alle empfangenen Bilder zunächst in einem temporären Nutzerbereich. Wurden alle Bilder erfolgreich übertragen, können sie vom temporären Speicherort an den dauerhaften übertragen werden. Wenn die Bilder erfolgreich hochgeladen wurden, kann der eigentliche Beitrag erstellt werden und die Bilder im temporären Speicherbereich referenzieren. Dann werden sie automatisch von dort in den dauerhaften Speicher verschoben.

Beim Abrufen der Bilder können diese an den anfragenden Client angepasst werden. Der Bilder-Dienst erlaubt es, spezifische Auflösungen anzufragen um den Bilder Download möglichst an die Gegebenheiten des Endgeräts anzupassen. Befindet sich der Nutzer gerade an einem Ort mit schlechter Netzabdeckung oder nutzt ein Gerät mit niedriger Bildschirmauflösung, dann sollten Bilder mit geringer Auflösung geladen werden. Dadurch reduziert sich die benötigte Bandbreite und das Bild wird schneller auf dem Endgerät dargestellt.

Auch dieser Dienst ist komplett an die Infrastruktur der Datenauskunft und Löschung angebunden, die vom Nutzer erstellten Bilder werden dann in einem Zip-Archiv zur Verfügung gestellt.

3.14 Leichtgewichtige Statistiken

Aus Sicht von Mandanten, die Anwendungen auf der Plattform gebucht haben, ist ein Überblick über die Nutzung dieser Anwendungen wichtig. Dabei sind je nach Anwendung unterschiedliche Kennzahlen relevant. Beispielsweise spielen im PfaffFunk die Anzahl der erstellten Beiträge und Kommentare eine große Rolle.

Diese Metriken können durch die Plattform leichtgewichtig zur Verfügung gestellt werden. Dabei werden sie jeweils nachts für den vorherigen Tag erhoben

und in verschiedenen Formaten zur Verfügung gestellt. Für die gesamte Plattform werden dabei Berichte im CSV-Format erstellt, aufgeschlüsselt nach geografischen Bereichen alle Metriken enthalten. Diese werden für verschiedene Zeiträume erstellt (täglich, wöchentlich, monatlich, jährlich), so dass eine einfache Auswertung möglich ist.

Zusätzlich können für eine Auswahl an geografischen Bereichen eigene Berichte im CSV-Format erstellt werden, die dann einfach weiterverarbeitet werden. Dabei kann mit Hilfe des bereits beschriebenen Konfigurationskonzepts definiert werden welche Berichte mit welchen Metriken für welchen Zeitraum erstellt werden.

Zum einfachen Zugriff auf diese Statistiken ist in der Administrationsoberfläche der Abruf der jeweils letzten aktuellen Statistiken möglich. Diese können als CSV-Dateien abgerufen werden, so dass dann mit Hilfe eines Tabellenkalkulationsprogramms Auswertungen gemacht werden können.

4 Anwendungsentwicklung- und Betrieb auf der Plattform

Ein wichtiges Ziel der Plattform ist es, die Entwicklung von Diensten auf der Plattform effizient zu gestalten, so dass eine Fokussierung auf die eigentliche Logik der Anwendung erfolgen kann. Dazu bringt die Plattform die oben beschriebenen Dienste mit, die von den jeweiligen Anwendungen eingesetzt werden können. Damit ist auch die Vernetzung der Anwendungen untereinander leichter möglich, da die Nutzer eine einheitliche Identität über alle Anwendungen auf der Plattform erhalten.

4.1 Entwicklung von Anwendungen

Um eine schnelle prototypische Realisierung von Anwendungen zu ermöglichen, wurde der Plattform-Mock geschaffen, wie bereits im IKT-Konzept aus AP 1.1.4 beschrieben. Der Fokus dieses Plattform-Prototyps ist dabei die Evaluation der Vernetzung von Anwendungen. Damit steht eine deutlich einfachere Version der Plattform zur Verfügung, auf der Anwendung einfach miteinander kommunizieren können. Funktionen, die für einen Produktivbetrieb notwendig sind, aber für eine Evaluation eines Prototyps keine Rolle spielen, wurden hier ausgeklammert. Eine Übersicht dazu wird in Kapitel 5 gegeben.

Bei den Anwendungen auf der Plattform muss zwischen dem Frontend der Anwendung (App oder Webseite) und dem Backend (Verbindende Logik und Datenhaltung zwischen den Nutzern der Anwendung) unterschieden werden. Im Beispiel des PfaffFunk ist das Frontend eine App, die über den App-Store bezogen werden kann und auf dem Gerät des Nutzers installiert wird. Im Gegensatz dazu gibt es den verbindenden Backend-Teil des PfaffFunk, der die Verteilung der Beiträge zwischen den Nutzern überhaupt erst ermöglicht. Dieser existiert nur einmal und hält die Daten aller Nutzer und gibt entsprechend ihrer Berechtigungen darauf Zugriff. Dieser Backend-Teil ist ein separates Modul, was zusammen mit der Plattform auf einem gemeinsamen Server installiert wird.

In der aktuellen Version der Plattform wird der Backend-Teil von Anwendungen als eigene Module entwickelt. Diese Module sind in Java geschrieben und integrieren sich zur Laufzeit in die Plattform. Das hat aus Performanz- und Wartbarkeits-Sicht entscheidende Vorteile, da diese Integration für die Anwendungs-Entwickler einfach zu verstehen ist und eine sehr geringe Latenz bei der Kommunikation zu den Plattform-Diensten auftritt. Da bisher alle Anwendungen auf der Plattform auch von Fraunhofer IESE geschrieben wurden, ist dieser Weg der effizienteste. Nachteile ergeben sich jedoch daraus, dass bei dieser eher direkten

Integration innerhalb der gleichen Java Virtual Machine ein hohes Maß an Vertrauen in den Backend-Code der Anwendung vorausgesetzt wird, da die internen Schnittstellen der Plattform keinen Sicherheitsmechanismus haben. Weiterhin erfordert es eine genaue Abstimmung der Plattform- und Anwendungs-Entwickler, da diese immer gemeinsam in einer neuen Version öffentlich gemacht (deployed) werden. Eine gesonderte Abrechnung der Betriebskosten der Plattform und der einzelnen Anwendungen ist dadurch ebenfalls schwer möglich.

Daher wurde in der Plattform ein Konzept umgesetzt, bei dem die Anwendungen freier von der Plattform entwickelt und veröffentlicht werden können. Dazu wurde ein Teil der bisher internen Schnittstellen der Plattform als REST-API, also eine öffentlich aufrufbare Schnittstelle, zur Verfügung gestellt. Damit kann der Backend-Teil der Anwendung flexibler realisiert werden, auch ohne die Technologie auf Java einzuschränken. Der Betrieb dieser Anwendung kann damit auch unabhängig von der Plattform erfolgen, beispielsweise auf einem eigenen Server des Anwendungs-Entwicklers. Bisher wurde nur ein Teil der internen Schnittstellen als öffentliche Schnittstelle zur Verfügung gestellt, was jedoch ausreicht um einfachere Anwendungen zu realisieren. Dieses Konzept wurde bereits mit einigen Anwendungen erfolgreich evaluiert. Die Anwendungen halten dabei die fachlichen Daten in ihrer eigenen Datenhaltung und beziehen von der Plattform nur ausgewählte Stammdaten.

4.2 Betrieb von Anwendungen

Anwendungen auf der Plattform zerfallen, wie oben bereits erwähnt, häufig in zwei Teile: Die Benutzerschnittstelle, entweder eine Webanwendung oder mobile App, sowie ein dazugehöriges Backend. Der Betrieb und die Verteilung der App ist dabei komplett unabhängig von der Plattform und erfolgt über die üblichen Kanäle, also die Hersteller-spezifischen App-Stores. Änderungen durch die Entwickler an den Anwendungen werden dabei unabhängig von der Plattform direkt bei den jeweiligen App-Stores direkt verteilt.

Bei Webanwendung erfolgt der letztendliche Betrieb im eigentlichen Sinne im Browser des Nutzers, die Verteilung der Webseite erfolgt über ein Hosting für statische Inhalte. Die Plattform schränkt dieses Hosting nicht weiter ein, bietet jedoch die Möglichkeit innerhalb der bestehenden Infrastruktur das Hosting zu übernehmen. Technisch ist dies getrennt von der Plattform, für den Nutzer erscheint es durch die Verwendung einer gemeinsamen URL und des gemeinsamen Logins integriert. Bei einer Änderung der Webanwendung kann diese automatisiert aktualisiert werden.

Für den Backend-Teil einer stark integrierten Anwendung ist der Betrieb aktuell ebenfalls in den Betrieb der Plattform integriert. Anwendungs-Entwickler stellen dazu den Code der Anwendung als Java-Modul zur Verfügung (üblicherweise ein jar), dieses wird dann beim automatischen Zusammenfügen der Plattform

integriert und zu einem größeren Modul gepackt. Dieses Modul wird dann als Ganzes in der Laufzeitumgebung der Plattform installiert, so dass sowohl die Plattform als auch der Backend-Teil der Anwendungen gemeinsam betrieben werden. Dadurch können die Anwendungen einfach durch Funktionsaufrufe mit der Plattform kommunizieren. Eine neue Version dieses Teils der Anwendung kann somit nur gemeinsam mit der Plattform veröffentlicht werden, so dass auf organisatorischer Ebene Abstimmungen notwendig sind. Das Überwachen der Anwendungen ist dabei zusammen mit der Plattform möglich, wobei getrennte Sichten für die Anwendungs-Entwickler ermöglicht werden können, die nur Informationen der jeweiligen Anwendung zeigen.

Für, wie oben beschrieben, nicht stark integrierte in die Plattform integrierte Anwendungen kann der Betrieb des Backend-Teils von Anwendungen auf eine andere Art erfolgen. Die Plattform stellt öffentliche Schnittstellen für Anwendungen bereit, die dann von den Entwicklern benutzt werden können. Die Anwendung kann dann unabhängig von der Plattform veröffentlicht und betrieben werden. Damit wird die organisatorische Abstimmung zwischen Plattform-Entwickler und Anwendungs-Entwickler deutlich vereinfacht. Die Konfiguration der Integration dieser Anwendungen erfolgt über die Plattformadministrationen. Dazu gehören beispielsweise Zugangsdaten, mit denen sich die Anwendungen an der Plattform identifizieren und URLs zu Endpunkten, die die Anwendungen bereitstellen müssen um DSGVO-Prozesse zu unterstützen.

4.3 Nutzung von Anwendungen durch Bürger

Das Ziel der Plattform und der Anwendungen auf der Plattform ist es, den Nutzern ein möglichst gutes Erlebnis bei der Benutzung zu bieten. Dazu gehört es auch, die Anwendung genau auf die Bedürfnisse und die Umgebung des Nutzers zuzuschneiden. Die bisherigen Anwendungen auf der Plattform werden von Verantwortlichen in Gemeinden, Städten, Landkreisen oder Ländern gebucht. Die Administratoren der Plattform bereiten dann die Stammdaten dieser Mandanten vor, das heißt sie erstellen unter anderem auf der Plattform geographische Einheiten, basierend auf den Daten des Statistischen Bundesamtes und zusätzlichen Quellen. Je nach Beauftragung werden diesen Einheiten dann in bestimmten Anwendungen verfügbar gemacht. Damit können dann die Bürger dieser Region die von ihrer Verwaltung beauftragten Anwendungen nutzen. Dieser Prozess ist zu großen Teilen automatisiert, wird jedoch durch Administratoren der Plattform ausgeführt.

Die Einrichtung eines Marktplatzes für Anwendungen auf der Plattform ist eine für die Zukunft der Plattform angedachte Weiterentwicklung. Dieser Marktplatz stellt die in einer Region potentiell verfügbaren Anwendungen der Verwaltung dieser Region übersichtlich dar. Bei Bedarf können diese Anwendungen dann direkt gebucht werden. Die notwendige Konfiguration der Anwendung und der Plattform erfolgt darauf hin automatisch. Diese Art der Bereitstellung wird jedoch

nicht bei allen Anwendungen möglich sein, da oft auch organisatorische Maßnahmen, wie Schulungen und Installationen vor Ort, erfolgen müssen. Daher werden die Buchungen im Marktplatz auch immer manuelle Schritte beim jeweiligen Anwendungs-Entwickler auslösen. Insgesamt erhöht der Marktplatz jedoch stark die Sichtbarkeit der Anwendungen und macht so die Plattform attraktiver für Anwendungs-Entwickler.

5 Plattform-Mock

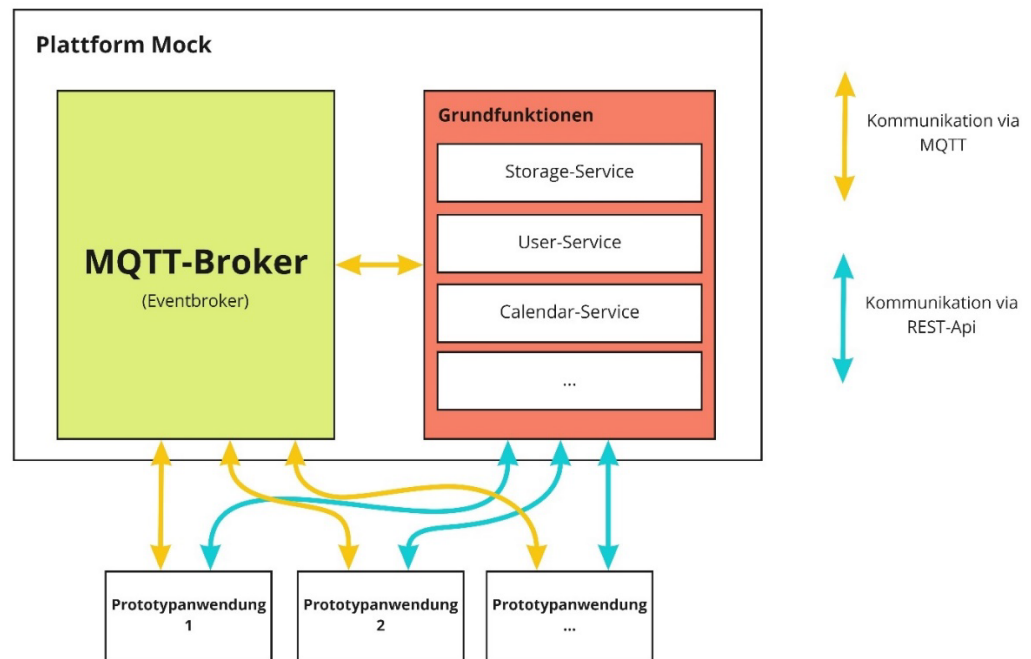
Eine Analyse und Vorstellung des Plattform-Mocks wurde im Zuge der Masterarbeit „Analyse und Erweiterung eines Plattform-Mocks für eine Stadtquartiersplattform“ von Phil Stüpfert angefertigt. Das folgende Kapitel enthält Auszüge dieser Masterarbeit.

5.1 Nutzen des Plattform-Mocks

Der Plattform-Mock bietet eine Prototypumgebung und soll Entwickler dabei unterstützen neue funktionale Prototypen im Plattformkontext schneller und effizienter zu entwickeln. Das bedeutet, dass die Entstehungsdauer eines Prototyps zwischen grundlegender Idee zu Beginn bis zu fertig umgesetzten Prototypanwendung reduziert werden kann. Dies ermöglicht es letztlich, dass neue Anwendungen schneller entwickelt werden können.

Eine wichtige Funktion einer produktiv eingesetzten Plattform ist es die Funktionen unterschiedlicher Dienste zu verknüpfen und den Nutzenden somit einen Mehrwert zu liefern. Zu Beginn einer jeden Entwicklung eines Dienstes steht zunächst eine Idee. Diese muss vor ihrer Realisierung auf ihre Wirksamkeit und Umsetzbarkeit überprüft werden. Aus diesem Grund werden Prototypen angefertigt, um die gesuchten Eigenschaften zu untersuchen. Die realistischsten Ergebnisse erhält man, indem man Prototypen in die reale Plattform integriert. Dadurch entsteht jedoch ein erhöhter Entwicklungsaufwand, da die Plattform hohe Anforderungen, z.B. an die Authentifizierung, Sicherheit oder Kommunikationsstandards, stellt. Um diesen Mehraufwand zu reduzieren stellt der Plattform-Mock alle wesentlichen Plattformfunktionen bereit, welche für die Evaluierung der Prototypen benötigt werden, reduziert jedoch die Komplexität und Kompliziertheit einer produktiv eingesetzten Plattform auf ein Minimum.

5.2 Aufbau des Plattform-Mocks



Aufbau Plattform-Mock

Der Plattform-Mock besteht aus vier Komponenten. Im Kern steht ein Eventbroker. Über diesen können alle Teilnehmer (Dienste) einfach miteinander kommunizieren. Dabei wird das MQTT-Protokoll verwendet, welches in allen gängigen Programmiersprachen sowie IoT-Geräten unterstützt wird.

Die zweite Komponente des Plattform-Mocks sind Services (Dienste), welche Grundfunktionen, welche bei der Prototypentwicklung immer wieder benötigt werden, bereitstellt. Unter anderem wird so z.B. ein Cloudspeicher bereitgestellt, über welchen größere Daten ausgetauscht werden können, oder ein User-Service, mit dessen Hilfe Benutzer angelegt und verwaltet werden können.

Die dritte Komponente ist die Einführung sogenannter „Shared-Topics“. Diese haben durch eine Kommunikationsstandardisierung das Ziel, die Kommunikation zwischen bestehenden und neu hinzugefügten Anwendungen zu vereinfachen. Durch die gewählten Kommunikationsregeln ist es möglich, dass auch bestehenden Anwendungen Nachrichten von neu hinzugefügten Anwendungen erhalten und verarbeiten können, ohne, dass es eine Anpassung an bestehenden Implementierungen bedarf.

Die letzte Komponente ist eine Feedbackkomponente, mit welcher die Kommunikation, welche über den Plattform-Mock durchgeführt wird, analysiert werden

kann. Diese Komponente ermöglicht einen detaillierten Einblick in die Kommunikation eines Prototyps, wie z.B. durchschnittliche Nachrichtengröße, Senderate, Anzahl der Empfänger usw. Die dadurch gewonnenen Informationen können zu Kommunikationsoptimierungen der späteren Anwendung verwendet werden.

5.3 Verwendung des Plattform-Mocks

Der Plattform-Mock steht dauerhaft den Entwicklern auf Servern zur Verfügung, wodurch keine lokalen Instanzen des Plattform-Mocks deployed werden müssen. Sämtliche Funktionen können einfach via MQTT- oder REST-API-Client erreicht und verwendet werden. Um den Einstieg darüber hinaus nochmals zu vereinfachen, stehen den Entwicklern/Entwicklerinnen eine SDK in allen gängigen Programmiersprachen zur Verfügung. Diese bieten alle Grundfunktionalitäten (Kommunikation mit MQTT-Broker, Erstellung von Shared-Topics, Verwendung von Grundfunktionen) per einfachen Aufruf einer Funktion an. Bei der Verwendung eines Prototypen kann dessen Kommunikation stetig analysiert und überwacht werden.

Der Plattform-Mock unterstützt bei der Entwicklung von Prototypen und vereinfacht deren Kommunikation untereinander. Dadurch können Evaluationen von neuen Ideen beschleunigt werden. Dies kann sogar dazu führen, dass mehr Ideen evaluiert werden können und somit potenziell auch mehrere Anwendungen realisiert werden können. Durch den modularen Aufbau des Plattform-Mocks ist es auch möglich nur einzelne Komponenten zu verwenden. Dadurch bietet der Plattform-Mock den Entwickler-Teams volle Flexibilität ohne diese zu restriktieren.

Mit Hilfe des Plattform-Mocks wurden bereits einige Prototypen erstellt. Im Zuge des Projekts suchen wir ebenfalls neue digitale Konzepte, welche in öffentlichen Raum integriert werden und ohne direkte Verwendung von Smartphones funktionieren sollen. So können z.B. digitale Straßenschilder oder Smarte Straßenlaternen auf die aktuelle Verfügbarkeit von Carsharing Angeboten hinweisen. Um diese Konzepte besser visualisieren zu können, haben wir ein Gameboard entwickelt, welches über eine Weboberfläche eine zweidimensionale Karte des Pfaff-Quartiers darstellt. Auf dieser Karte bewegen sich kleine Punkte, welche stellvertretend für Fahrzeuge wie Autos, Busse, Fahrräder usw. aber auch Personen und Tiere stehen können. So mit dieser Visualisierung ist es dem/der Betrachter/in einfach möglich, komplexere Simulationen nachvollziehen zu können. Geschehnisse wie z.B. das Betreten eines Bereichs durch eine Person können Events ausgelöst werden, welche über die Plattform, veröffentlicht werden.

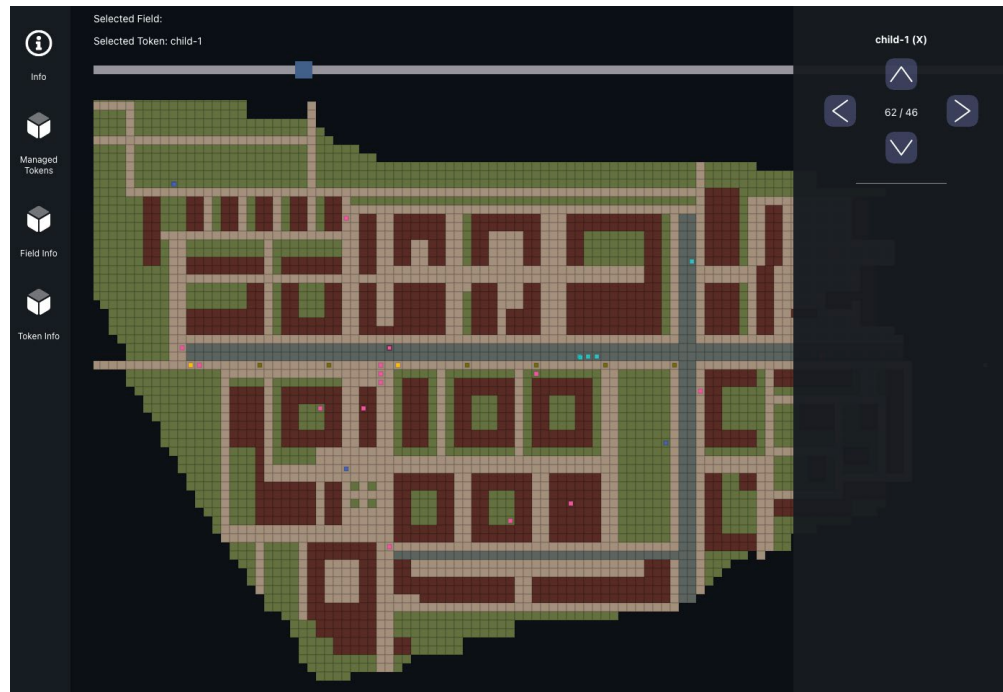


Abbildung 4: Eindruck des Gameboards

Dieses Gameboard wurde zusammen mit dem Plattform-Mock bei dem Hackathon "Pfaffhack 2022" eingesetzt. Die Teilnehmenden hatten die Aufgabe innerhalb von 24 Stunden neue Konzeptideen zu erarbeiten und prototypisch umzusetzen. Der Plattform-Mock half sehr bei der Vernetzung von Anwendungen und das Gameboard diente den Teilnehmenden für ihre Endpräsentation.

6 Eine Anwendung am Beispiel der Energieeffizienzapplikation

Im Rahmen von AP 2.1.1 fanden nicht nur die Konzeption und Entwicklung der Quartiersplattform, sondern auch einer Energieeffizienzapplikation statt. Zielstellung war es, die Infrastruktur der Plattform zu verwenden, um Nutzer*innen ein Nutzungserlebnis zu bieten, das ihren Energieverbrauch sichtbar macht und ihnen hilft, im Alltag Energie zu sparen. Das Konzept der Energieeffizienzapplikation sollte dann prototypisch umgesetzt werden.

6.1 Kontext und Vorgehensweise

Private Haushalte haben einen Anteil von ca. einem Viertel am gesamten Endenergieverbrauch der Bundesrepublik. Gerade in unseren eigenen vier Wänden ergibt sich so ein großes Energieeinsparpotenzial, von dem wir auch direkt profitieren, denn wir sparen Geld. Der Verbrauch ist jedoch nicht offen sichtbar und drückt sich nur auf der Nebenkostenabrechnung eines Haushaltes in Zahlen aus.

Energie ist ein sehr abstrakter Begriff. Einfache Tricks zum Energiesparen sind heute schon recht leicht umzusetzen (z.B. der Verzicht auf Standby-Verbraucher, niedrigere Raumtemperaturen im Winter). Über diese Schritte hinaus wird es jedoch schwierig für private Haushalte. Die richtige Heiztemperatur zum Beispiel ist abhängig von vielen Faktoren wie der Außentemperatur, der Luftfeuchtigkeit, der erwarteten Temperaturentwicklung usw.

Smarte Haustechnik ist ein Baustein, mit dem eine möglichst effiziente Steuerung der Temperatur oder des Energieverbrauchs erreicht werden kann, doch dies muss zum persönlichen Komfortempfinden der Bewohner eines Haushaltes passen. Oft sind Menschen nicht bereit, auf einen hohen Grad an Komfort zu verzichten, wenn sie sich einmal daran gewöhnt haben.

Um eine für die Vision des PFAFF-Quartiers passende Energieeffizienzapplikation zu entwerfen, haben Mitarbeiter*innen des Fraunhofer IESE verschiedene Anforderungen, Randbedingungen und potenzielle Lösungsräume explorativ untersucht. Es zeigt sich, dass die Zielstellung »Energieverbrauch reduzieren«, insbesondere für Privathaushalte, keine eindimensionale Problemstellung ist. Eine reine Visualisierung des Energieverbrauchs geht am Ziel vorbei, denn ohne Hilfsmittel die tatsächlich auch zum Energieeinsparen beitragen und gleichzeitig von Nutzer*innen als adäquat und nicht einschränkend empfunden werden, bleibt die Energieeffizienz privater Haushalte weit hinter ihren Möglichkeiten zurück.

Daher näherten wir uns der Konzeption der Energieeffizienzapplikation mit einem stark strukturierten und ergebnisorientierten Kreativprozess, dem »Design

Sprint«. Ein Design Sprint ist ein zeitlich begrenzter Fünf-Phasen-Prozess, bei dem Design Thinking eingesetzt wird, um das Risiko bei der Markteinführung eines neuen Produkts, einer Dienstleistung oder einer Funktion zu verringern. Der Prozess soll Teams dabei helfen, Ziele klar zu definieren, Annahmen zu validieren und einen Produktfahrplan festzulegen, bevor mit der Entwicklung begonnen wird. Mit Hilfe von interdisziplinärem Fachwissen, Rapid Prototyping und Usability-Tests sollen strategische Fragen geklärt werden. Dieser Designprozess ist vergleichbar mit den Sprints in einem agilen Entwicklungszyklus.

6.2 Konzept

Im Rahmen des Design Sprints entstand das Konzept und ein erster low-fidelity Prototyp der Energieeffizienzapplikation: die »Fish n' Tipps« Anwendung. Das Konzept der Anwendung lässt sich folgendermaßen beschreiben:

»Eigentlich wissen wir doch, was gut und was schlecht für die Umwelt und das Klima ist... und sind trotzdem inkonsequent.« So oder ähnlich lässt sich das aktuelle Handeln der Menschen in Bezug auf Klimafreundlichkeit zusammenfassen. Ein Großteil der Menschen lehnt Klimaschutz und umweltfreundliches Verhalten nicht per se ab, sondern möchte lediglich bequem und ohne größere Anpassungen den Alltag bestreiten. Alternativen sind eventuell gar nicht bekannt. Vielleicht wissen viele Menschen auch gar nicht, wie sich umweltfreundliche Handlungsalternativen leicht umsetzen lassen. Fish n' Tipps bietet persönliche, auf das eigene Leben und die individuelle Erfahrung zugeschnittene Hinweise und Tipps, um das eigene Handeln umwelt- und klimafreundlicher auszurichten.

Jedem Nutzer und jeder Nutzerin wird ein persönliches Haustier in Form eines Fisches zur Seite gestellt. Dieser Fisch repräsentiert das gute Gewissen, die positive Handlung, das klimafreundliche Selbst. Fish n' Tipps ist mit anderen Plattform-Apps verbunden und analysiert die Aktionen, die darin vorgenommen werden, auf ihre Bedeutung für das Klima bzw. die Umwelt (z. B. wenn ein Carsharing-Auto gemietet wird, der Energieverbrauch des eigenen Haushalts über die Smart-Home-App abgefragt wird oder über MeDeTe Werkzeuge ausgeliehen werden). Passend zu diesen Handlungen gibt der eigene Fisch dann individuelle Tipps aus, die dem Nutzer helfen, noch umweltfreundlicher zu handeln oder solches positiv zu unterstützen. Die Tipps wiederum kommen von allen Nutzern, die ihre eigenen Erfahrungen, Ideen, Alternativen und Tipps mit anderen Nutzern teilen können. Besonders engagierte oder erfahrenere Nutzer können so wertvolle Ideen und Tipps mit der gesamten PFAFF-Gemeinschaft teilen, die allen zugutekommen.

Die Energieeffizienzapplikation »Fish n' Tipps« versucht also nicht einzelne Aspekte des Energieverbrauchs von Privathaushalten herauszugreifen und diese sin-

gular zu adressieren. Stattdessen setzt die Anwendung auf die Kraft der Vernetzung einzelner Individuen zu einer »Community von Energiesparern«, welche untereinander praktikable und praxisrelevante Tipps und Hinweise austauschen.

Diese Hinweise wiederum basieren auf bestimmten Ereignissen bzw. »Triggern«, welche über die Quartiersplattform bekannt gemacht werden (siehe 2.1 Technische Dekomposition der Plattform) und fügt sich so nahtlos in das technische Gesamtkonzept der Quartiersplattform ein.

6.3 Umsetzung

Während des fünftägigen Kreativprozesses (Design Sprint) entstand bereits ein erster Prototyp der Energieeffizienzapplikation, welcher dann im Nachgang verfeinert, überarbeitet und mit Hilfe des Plattform-Mock umgesetzt wurde. Die Umsetzung mit dem Plattform-Mock erlaubt es, den Prototypen der Energieeffizienzapplikation schnell im Zusammenspiel mit anderen Anwendungen zu testen, da er ein hochintegriertes, vernetztes Konzept verfolgt.

Es wurde bewusst auf die Umsetzung mit der Quartiersplattform verzichtet und stattdessen auf den Plattform-Mock gesetzt. Durch den langsamen Baufortschritt und somit dem verzögerten Einzug realer Nutzer*innen garantiert die Umsetzung auf dem Plattform-Mock insbesondere die schnelle Weiterentwicklung und Adaption des Prototyps, ohne einer späteren Umsetzung für ein reales Quartier im Weg zu stehen.

Die Anwendung »Fish n' Tipps« wurde als sogenannte Webapp umgesetzt. Eine Webapp ist eine Anwendung, die mit Hilfe von Technologien wie HTML, CSS und JavaScript entwickelt wird und die Engine eines Webbrowsers als Ausführungsumgebung verwendet. In diesem Fall verwendeten wir das Framework »react.js« um die Module zu strukturieren und den Datenfluss durch die Anwendung zu regeln. React.js ist ein weit verbreitetes, bekanntes und gut unterstütztes Framework zur Entwicklung von Web- und nativen Applikationen. Dritte (z. B. Entwickler*innen die an Hackathons teilnehmen) können so schnell den vorhandenen Code begreifen, ihn erweitern und eigene Lösungen darauf aufbauen, wie sich im Rahmen der PFAFF HACKs gezeigt hat (siehe AP 2.1.8 Real-labor-Zentrum: Quartierswerkstatt).

Weitere Erläuterungen zur Quartiersenergieeffizienzapplikation »Fish n' Tipps« finden sich auch in unseren Populärwissenschaftlichen Videos zum Thema: <https://www.youtube.com/watch?v=0l8Sb2Zi-zk> und <https://www.youtube.com/watch?v=Jkt8zLJ5P3E>. Nachfolgende Abbildung 4 zeigt beispielhaft drei Bildschirme, mit denen Nutzer*innen beim Verwenden der Applikation interagieren können.

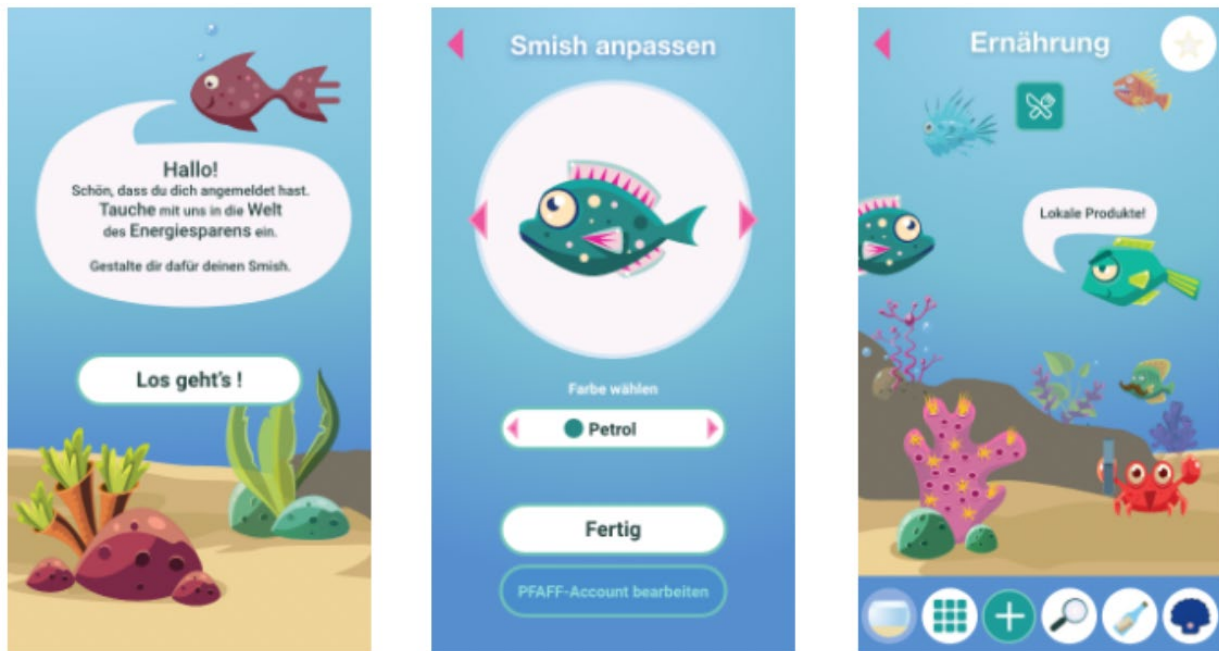


Abbildung 5: Beispielhafte Darstellung der Fish n' Tipps App - die Energieeffizienzapplikation.

7 Evaluation

Im Rahmen des Projektes wurde mit dem PfaffFunk eine eigene Variante des DorfFunks entwickelt, die auf die Kommunikationsbedürfnisse innerhalb des Quartiers und Informationsbedarfe um das Quartier herum fokussiert wurde. Wie in Kapitel 3.5 beschrieben, konnte diese App als eine Variante des DorfFunks durch die Plattform einfach konfiguriert werden, ohne das auf Seite des Backends explizit neuer Code geschrieben werden musste. Die App selbst wurde angepasst, so dass eine Komponente zur Erfassung von Nutzerfeedback integriert werden konnte.

7.1 Feedback-Komponente

Die Entwicklung des Plattform-Mocks erleichtert den Prototyping-Prozess für die Entwickler neuer Dienste. Für bereits im Betrieb befindliche Dienste ist es wichtig im Rahmen einer nachhaltigen Softwareentwicklung eine kontinuierliche Evaluierung vornehmen zu können. Da dedizierte Evaluationsevents aufwändig in ihrer Durchführung sind und nur punktuell Feedback der Nutzer einholen habe wir uns für die Entwicklung einer Feedback-Komponente entschieden. Diese Komponente ist auch im PfaffFunk integriert.

Durch die Einbindung der Komponente sind zwei Arten von Feedback möglich so genanntes „Push-“ oder „Pull-Feedback“. Push-Feedback wird dadurch ermöglicht, dass der Nutzer ein Feedback-Formular zur Verfügung hat, um aktiv seine Meinung zur genutzten Software auszudrücken. So bekommen Entwickler zu einer Vielzahl von Aspekten Feedback, allerdings ist dieses Feedback nicht unbedingt zu Aspekten, die das Entwicklungsteam gerade evaluieren möchte.

Dies übernimmt die zweite Art von Feedback, die die Komponente ermöglicht. Das so genannte „Pull-Feedback“. Hierbei fragt die Komponente nach von den Entwicklern eingestellten Ereignissen in der Anwendung gezielt um Feedback. Da dies den Interaktionsprozess des Nutzers unterbricht, muss dies so minimal invasiv wie möglich gehen.

Der Prozess des Feedbackgebens über den Pull-Mechanismus ist in Abbildung 6 dargestellt. Ein Nutzer drückt den "Herz"-Button, um auszudrücken, dass ihm ein bestimmter Funk gefällt. Danach wird für einige Sekunden das Feedback-Popup angezeigt. Während das Popup angezeigt wird, kann der Nutzer weiter mit der App interagieren. Wenn er kein Feedback geben möchte, wird die Feedback-Anfrage automatisch ausgeblendet. Wenn er ausdrücken möchte, wie ihm die genutzte Funktion gefällt, genügt ein Klick auf die Emojis. Anschließend wird der Nutzer gefragt, ob er weitere Angaben machen möchte. Wie bei der ersten

Anfrage kann der Nutzer weiter mit der App interagieren. Reagiert er nicht auf die Anfrage wird sie einigen Sekunden automatisch geschlossen. Entscheidet sich der Nutzer für weiteres Feedback, wird ein Formular angezeigt, in das er Text und Bilder eingeben kann. Auf diese Weise wird sichergestellt, dass die Abgabe von Feedback eine Interaktion ist, die schnell erledigt werden kann und keine Beeinträchtigung des Benutzererlebnisses darstellt.

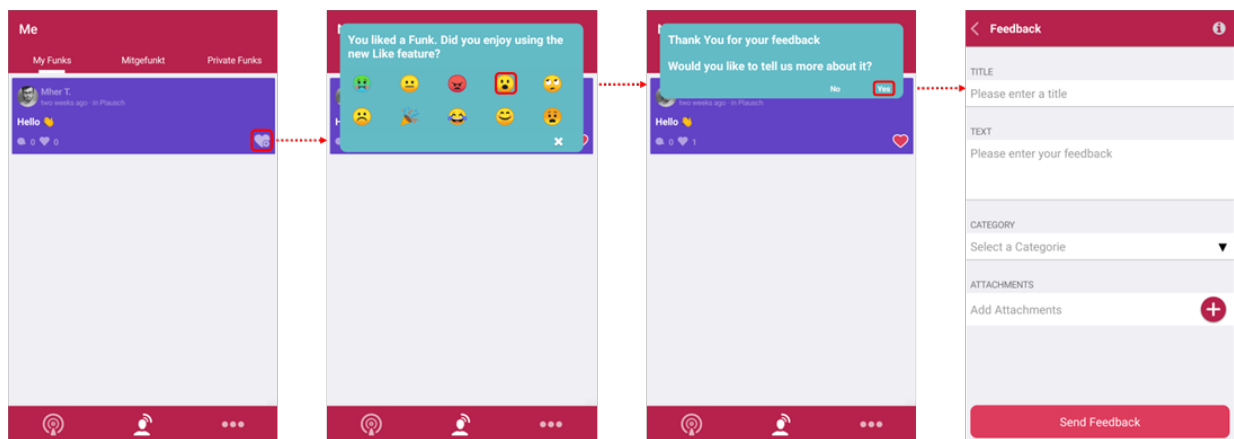


Abbildung 6: Pull-Feedbackmechanismus im PfaffFunk

Die Feedback-Komponente ist so konzipiert, dass sie mit wenigen Schritten in mobile Anwendungen auf Android und iOS sowie auf Windows 10 eingebunden werden kann. Web-Apps können die von unserer Feedback-Plattform angebotene API direkt nutzen. Das gesammelte Feedback ermöglicht es den Entwicklern, neue Ideen, Probleme und verbesserte Funktionen für ihre Produkte zu identifizieren.

Wir haben eine Evaluierung der in PfaffFunk implementierten In-App-Feedback-Komponente durchgeführt, um zu überprüfen, ob unsere zentrale Anforderung an die Komponente - eine einfach zu bedienende und nicht aufdringliche Lösung zur Abgabe von Feedback - erfüllt wurde. Es stellte sich heraus, dass die Teilnehmer die Art und Weise, wie sie Feedback geben konnten, mochten und insbesondere der Emoji-Mechanismus positiv wahrgenommen wurde.

7.2 Erfahrungen aus der Nutzung des PfaffFunks

Während der Nutzung des PfaffFunks wurden regelmäßig durch das Projektteam Beiträge veröffentlicht, die aktuelle Themen des Projekts behandelten. Dadurch wurde ein zusätzlicher Kommunikationskanal an interessierte Bürger etabliert. In Abbildung 7 sind für den Zeitraum von 01.12.2021 – 31.12.2022 genauere Statistiken dargestellt. Diese detaillierten Statistiken wurden erst im Laufe des Projekts entwickelt und noch ergänzt, daher sind sie nicht seit Beginn verfügbar. Im Diagramm wird die Entwicklung verschiedener Kenngrößen gezeigt: Explizite

Nutzer sind solche Nutzer, die als Wohnort das Pfaff-Quartier angegeben haben. Da aktuell noch keine Möglichkeit besteht dauerhaft dort zu wohnen ist diese Anzahl entsprechend gering. Interessierte Nutzer sind solche, die in anderen Gebieten (im Umkreis von maximal 36 km) wohnen, aber sich für das Pfaff-Quartier interessieren. Die Anzahl solcher Nutzer stieg kontinuierlich weiter an. Beiträge sind hier die Beiträge, die vom Projektteam veröffentlicht wurden.

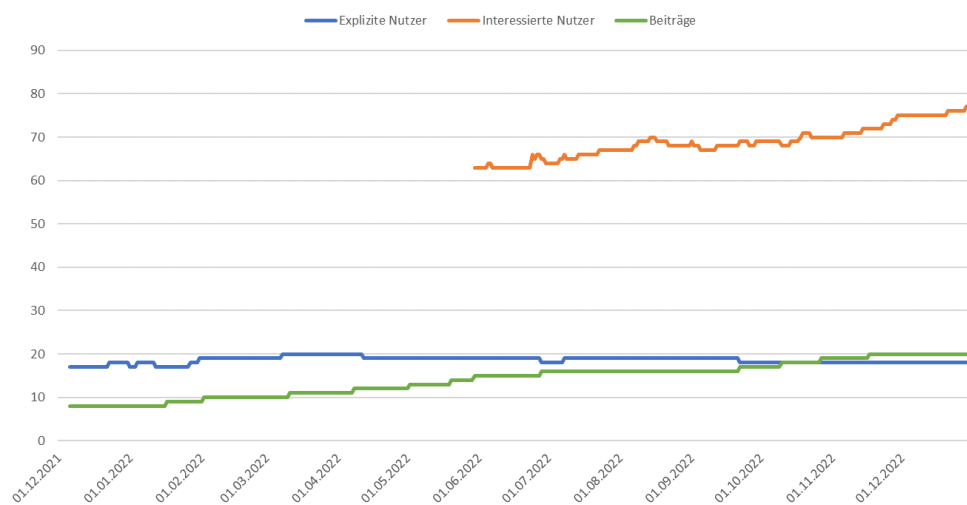


Abbildung 7: Nutzung des PfaffFunk

Insgesamt war die Nutzung des PfaffFunks bedingt durch den Mangel an echten Bewohnern des Pfaff-Quartiers eher gering. Trotzdem zeigen die Nutzungsstatistiken, dass ein echtes Interesse bestehen und dass die App im möglichen Rahmen angenommen wird. Eine weitere Evaluation zu einem späteren Zeitpunkt, wenn Bewohner das Quartier nutzen, steht noch aus.

8 Ausblick

Die Plattform wurde im Verlauf des Projektes intensiv weiterentwickelt. Dabei stand die einfachere Integration von Anwendungen auf der Plattform im Vordergrund, wie auch bereits oben beschrieben. Es wurden die bestehenden Dienste erweitert, wovon sowohl bestehende als auch neue Anwendungen profitierten.

Ein Prototyp der Energie-Effizienz-Anwendung wurde im Rahmen des IKT-Konzept aus AP 1.1.4 bereits erstellt, die "Fish 'n Tipps"-App. Diese wurde im weiteren Verlauf des Projektes noch stärker in den Plattform-Prototyp integriert.

Ein Deployment der Plattform im Pfaffquartier selber sowie das Ausrollen möglicher digitaler Dienste war aufgrund des Baufortschritts auf dem Pfaffgelände leider nicht wie ursprünglich geplant möglich. Dennoch konnte die Plattform selber in anderen Kontexten bereits erfolgreich eingesetzt werden, so dass eine spätere Nutzung prinzipiell möglich ist und dies entsprechend demonstriert werden konnte. Zudem haben wir mit dem Plattform Mock eine Umgebung entwickelt, mit der wir verschiedene digitale Dienste und Ideen dazu sowie prototypische Umsetzungen trotzdem erproben und Feedback sammeln konnten. Damit haben wir unterschiedliche digitale Dienste prototypisch entwickelt, die Teil eines Quartiersökosystems sein könnten.

Die Digitalisierung wird weiterhin eine große Rolle spielen für die Quartiersentwicklung und digitale Dienste können einen großen Mehrwert bieten, wenn es darum geht, klimafreundliches Verhalten auch dort zu unterstützen.