

Apps für den ländlichen Raum – Sicherstellung der Qualität bei stark heterogener Konfiguration

Frank Elberzhager, Balthasar Weitzel
Fraunhofer IESE, Fraunhofer-Platz 1, 67663 Kaiserslautern
{frank.elberzhager, balthasar.weitzel}@iese.fraunhofer.de

Zusammenfassung:

Heutige Software-Systeme werden immer komplexer. Der Trend darüber hinaus hin zu Ökosystemen erhöht die Komplexität weiter und macht auch die Qualitätssicherung schwieriger. Oftmals erfordert das neue bzw. erweiterte Strategien für die Qualitätssicherung. Im Rahmen des Projektes Digitale Dörfer entwickeln wir seit vier Jahren unterschiedliche Dienste für den ländlichen Raum. In unterschiedlichen Kommunen sind in der Regel die Anforderungen an die Dienste verschieden, was zu angepassten App-Varianten und App-Features führt. Das Management dieser verschiedenen App-Varianten und Features für die Qualitätssicherung ist ebenso komplex. In diesem Beitrag wollen wir Mechanismen aufzeigen, mit denen wir die geschilderte Komplexität besser beherrschbar machen. Damit möchten wir Impulse setzen für solche, die mit ebensolchen Herausforderungen zu tun haben.

Schlüsselworte: Qualitätssicherung, Konfigurationskomplexität, App-Entwicklung

1. Einleitung und Motivation

Die Welt, in der wir heute leben, wird zunehmend vernetzt. Vernetzte Autos, Haushalte und Fabriken, intelligente und mobile Geräte bis hin zu Kombinationen davon sind nur einige Beispiele. Dieser Trend zieht sich sowohl durch den privaten als auch den geschäftlichen Bereich und wird heute oft als digitale Transformation bezeichnet. Für Unternehmen ergeben sich neue Möglichkeiten, ihre Produkte und Dienstleistungen in solchen vernetzten Systemen anzubieten, und die Chancen für innovative Produkte und neue Geschäfte steigen. In diesem Zusammenhang haben sich Begriffe wie Internet der Dinge (IoT), digitale Ökosysteme oder cyber physical systems etabliert.

Während solche vernetzten Systeme einerseits enorme Chancen für Unternehmen bieten, gibt es andererseits viele Herausforderungen bei der Entwicklung und dem Betrieb solcher Systeme. Die Entwicklung von

eingebetteten Systemen oder Informationssystemen als solche ist oft schon durch hohe Komplexität gekennzeichnet, aber die Integration in vermehrt vernetzte Systeme erhöht diese Komplexität noch einmal erheblich. Herausforderungen sind z.B. die hohe Heterogenität, organisatorische Fragen wie entgegengesetzte Motivation der Akteure, Evolutionsaspekte oder widersprüchliche Qualitätsanforderungen. Neben diesen Herausforderungen nimmt der Druck, schnell am Markt zu sein, zu, d.h. ein Ziel für Unternehmen ist es oftmals, eine kurze Time-to-Market zu erreichen, um wettbewerbsfähig zu bleiben. Darüber hinaus spielt die Qualität eine entscheidende Rolle bei der Nutzerakzeptanz, aber auch bei der Innovationskraft und damit der Zukunftsfähigkeit solcher Systeme.

In diesem Beitrag wollen wir einige Erfahrungen aus dem Digitale Dörfer Projekt teilen. Seit 2015 entwickeln wir eine Plattform und darauf laufende Dienste für den ländlichen Raum und haben dazu mit unterschiedlichen Herausforderungen zu tun, welche Entwicklung, Qualitätssicherung und Betrieb umfassen. Ziel des Projekts ist es, den Alltag der Menschen im ländlichen Raum durch digitale Innovationen zu unterstützen. Dabei wollen wir in diesem Beitrag den Fokus auf die hohe Konfigurationskomplexität legen und aufzeigen, welche Mechanismen wir nutzen, um diese beherrschbar zu halten, so dass eine hohe Qualität sichergestellt wird.

2. Hintergrund und Herausforderung

Aktuell forschen wir an der Frage, wie die Attraktivität des ländlichen Raums in Deutschland durch digitale Dienste gesteigert werden kann. Dazu haben wir, u.a. im Projekt Digitale Dörfer¹, in den letzten Jahren eine Plattform sowie unterschiedliche Dienste entwickelt und in Form von Apps zur Verfügung gestellt. Diese Apps werden in den teilnehmenden Gemeinden in Deutschland eingesetzt. Solche unterschiedlichen Apps, wie beispielsweise der DorfFunk oder die LieferBar, wurden darüber hinaus in spezifischen Varianten für Bürger verfügbar gemacht. Da die Gemeinden eher heterogen sind, werden nicht alle Funktionalitäten der jeweiligen Apps in allen

¹ <https://www.digitale-doerfer.de/>

Gemeinden angeboten. Insbesondere in der Kommunikationslösung ist auch die hierarchische Gemeindestruktur in Deutschland relevant, um Bürgern die Möglichkeit zu geben, ihre Kommunikation feingranular zu steuern und beispielsweise einzelne Ortschaften aus Verbandsgemeinden zu adressieren.

Das Management von App-Varianten und App-Features für verschiedene Nutzergruppen stellt Entwicklungsbeteiligte immer wieder vor Herausforderungen. Aus der geschilderten Situation ergibt sich ein großer Raum an Möglichkeiten, was in der Qualitätssicherung zu berücksichtigen ist: Apps, App-Varianten, Mandanten (Gemeinden, Verbandsgemeinden, Landkreise) und Features je Mandant.

3. Ansätze zur Beherrschung der Konfigurationskomplexität

Wir nutzen vier technische Mechanismen, um die geschilderte Konfigurationskomplexität besser zu beherrschen. Dadurch sind wir in der Lage, die verschiedenen Konfigurationen abzubilden und im laufenden Betrieb unterbrechungsfrei anzupassen. Dies erstreckt sich von der Pflege der Konfiguration über die Anwendung und Vorhaltung im Backend sowie die Anwendung in den jeweiligen Apps.

Das gesamte System folgt einer ereignisorientierten Architektur, mit deren Hilfe eine einfache Erweiterbarkeit und leichtgewichtige Clients realisiert wurden. Diese Art der Kommunikation ermöglicht uns den beschriebenen übergreifenden Aspekt konsistent in allen Apps umzusetzen.

Die vier Kernkonzepte sind:

- Mandantenmanagement
- App-Variantenmanagement
- Featuremanagement
- Konfigurationsmanagement

3.1 Mandantenmanagement

Mandanten fassen Nutzergruppen zusammen, für die die gleichen Anforderungen gelten. Im Sinne der Vernetzung der Bürger untereinander sind diese Mandanten nicht hart voneinander getrennt, wie es in anderen Systemen der Fall ist. Die Mandanten bestehen wiederum aus einer Menge von hierarchisch organisierten Regionen und Gemeinden. Bürger haben dabei eine Heimatgemeinde und eine Menge von anderen Gemeinden, für die sie sich interessieren. Im Falle der Kommunikationsanwendung werden für diese Gemeinden neue Beiträge angezeigt.

3.2 App-Variantenmanagement

Um möglichst unterschiedliche App-Varianten zu ermöglichen, haben wir konzeptionell zwischen Apps und deren Varianten unterschieden. Eine App ist in diesem Sinne die Obermenge aller Funktionalitäten der daraus abgeleiteten App-Varianten. Dadurch kann im Backendteil der Plattform auf einheitliche Art mit allen App-Varianten kommuniziert werden. Dies ist insbesondere beim Verschicken von Push-Nachrichten relevant.

App-Varianten werden dann für einzelne Mandanten freigeschaltet.

3.3 Featuremanagement

Features fassen logisch zusammengehörige Funktionalitäten zusammen. Diese können entweder komplett aktiviert oder deaktiviert sein, oder in vordefiniertem Rahmen angepasst werden. Diese Anpassung betreffen bestimmte Werte, die von den Funktionen des Features verwendet werden.

Um Features zu steuern bietet die Plattform die Möglichkeit je Mandant und App-Variante die Features unterschiedlich zu konfigurieren. Das bedeutet konkret, dass je nach Wohnort des Nutzers andere Funktionalitäten in der gleichen App-Variante zur Verfügung stehen.

Dazu werden sowohl in den Apps, als auch im Backend-Teil der Plattform diese Feature-Konfigurationen für jeden Nutzer ausgewertet. Da bei vielen Anfragen solch eine Auswertung stattfindet, müssen diese sehr performant innerhalb des Backends zur Verfügung stehen.

3.4 Konfigurationsmanagement

Die drei genannten Konzepte fließen im Konfigurationsmanagement zusammen. Um die Vielzahl an Konfigurationen zu beherrschen, wurden die Einstellmöglichkeiten in unterschiedliche Dateien aufgetrennt, die jeweils für einen einzelnen Aspekt zuständig sind, beispielsweise die Konfiguration eines Mandanten, eines Features oder einer App-Variante. Für jeden Mandanten sind dabei separate Dateien angelegt, so dass Änderungen einfach zu überblicken sind. Mittels einer Erweiterung für die im Projekt verwendete Entwicklungsumgebung wird die Verwaltung dieser Dateien erleichtert. Diese Konfigurationsdateien können dann im laufenden Betrieb unterbrechungsfrei angewendet werden.

4. Fazit und Ausblick

Durch die Auftrennung der Konfiguration in verschiedene Aspekte und deren revisions sichere Speicherung kann zu jeder Zeit genau bestimmt werden, welches Verhalten für den jeweiligen Nutzer in der jeweiligen

App zu erwarten ist. Da die unterschiedlichen Konfigurationen nur Ausprägungen von generischen Merkmalen sind, können diese trotz der Komplexität einfach getestet werden. Eine Herausforderung ist dabei immer noch das Zusammenspiel der Konfigurationen auf Anwendungsebene. Dabei kann trotz korrektem Systemverhalten eine Konfiguration geschaffen werden, die den Nutzer in ungewünschte Situationen bringen kann, die durch das Zusammenspiel oder das Abschalten von Features entstehen, die so nicht zusammen intendiert waren.

Eine Überprüfung der Konfiguration auf solche Fehler findet bisher nur auf niedriger Ebene statt. Sobald den Verantwortlichen in den jeweiligen Mandanten die Möglichkeit zur Konfiguration gegeben wird, muss diese jedoch auch auf Anwendungsebene erfolgen.

Unser Backend ist auf AWS deployed, so dass hier mittelfristig auch für die Qualitätssicherung Mehrwerte erwartet werden können. Insgesamt gehen wir davon aus, dass die Komplexität weiter steigen wird, da neue Mandaten stetig hinzukommen. Mit den beschriebenen Mechanismen ist es uns aber möglich, entsprechend zu skalieren und die Qualitätssicherung so durchzuführen, dass die Heterogenität beherrschbar bleibt.

Danksagung

Dieser Beitrag wurde im Kontext des Projekts EnStadt: Pfaff (Förderkennzeichen: 03SBE112D und 03SBE112G) erstellt, gefördert vom Bundesministerium für Wirtschaft und Energie (BMWi) sowie vom Bundesministerium für Bildung und Forschung (BMBF).