

1 Einleitung

Für ein junge Unternehmen ist es wichtig schnell Ideen umzusetzen und erstes Feedback zu erhalten. So lassen sich aus Einem dadurch ersten Annahmen bestätigt und zum Anderen leichter Investoren finden.

Diese schnelle Entwicklung führt jedoch zu einigen Abstrichen hinsichtlich der Qualität. So wird zu Beginn oftmals auf einen automatischen Prozess zur Bereitstellung der Applikation verzichtet und leicht umsetzbare Lösungen vor langfristigen bevorzugt. Auch hinsichtlich der Auswahl der Architektur wird anfängliche Performance als oberstes Auswahlkriterium bestimmt. In der Regel endet jedoch der Weg für ein Start-Up nicht nach wenigen Monaten und die anfänglichen Entscheidungen müssen hinterfragt werden.

Genau an diesem Punkt ist das junge Unternehmen PluraPolit, welches sich erst in der Mitte des letzten Jahrs gegründet hat und Anfang dieses Jahr innerhalb von wenigen Monaten ein fertiges Produkt entwickelte. Als Frontendentwickler bin ich seit Anfang Januar zu diesem Projekt angestellt. Gemeinsam mit einem der drei Gründer stellen wir zu Zweit die technische Abteilung des Unternehmens dar und kümmern uns um die ständige Weiterentwicklung der Plattform. PluraPolit hat sich zur Aufgabe gestellt eine Bildungsplattform für Jung- und Erstwähler zu entwickeln und bei der Meinungsbildung zu unterstützen. Gefördert wird das Projekt von der Zentrale für Politische Bildung und ist politisch Unabhängig. Des Weiteren handelt es sich bei PluraPolit auch um ein gemeinnütziges Unternehmen und verfolgt keine Absicht Gewinne auszuzahlen.

Der Content der Plattform wird von den zwei Anderen Gründern gepflegt und eingeholt. Es handelt sich dabei um neun verschiedene Tonaufnahmen zu einer Frage. Die Audioaufzeichnung kommen in der Regel ausschließlich von Politikern und beziehen sich auf ein politisches Thema, welches aktuell diskutiert wird. So gibt es zum Beispiel das Thema: Sollte der öffentliche-rechtliche Rundfunk abgeschafft werden. Die jeweiligen Politiker werden direkt für ein Statement angefragt und anschließend ohne Inhaltliche Veränderung auf die Seite gebracht. Angefragt werden immer alle Parteien die im Bundestag vertreten sind. Neben den von ausschließlich Politikern diskutierten Fragen, kommen auch Fragen auf die Plattform, die von jeweiligen Experten/innen beantwortet werden auf die Plattform. So gab es bei der oben genannten Frage Vertreter des Rundfunkbetreibers ARD. Im Gegensatz zu anderen Webseiten, die Neuigkeiten und Nachrichten vermitteln, gibt es auf PluraPolit ausschliesslich

Sprachnachrichten von Experten/innen. Es wurde sich exklusiv für dieses Format entscheiden, um die junge Zielgruppe anzusprechen und die einzelnen Beiträge wie ein Podcast hören zu können.

1.1 Problemstellung

Umgesetzt wurde die Plattform in Ruby on Rail im Backend und React im Frontend. Es wurde sich dafür entscheiden die Datenspeicherung separat von der Informationsdarstellung zu handhaben. Dieses Aufteilen wurde soweit durchgezogen, dass Beide einen unterschiedlichen Bereitstellungsprozess haben und in unterschiedlichen Ordnern gespeichert werden. Inhaltlich gibt es jedoch keine Trennung. So bezieht sich das Frontend ausschließlich und komplett auf das Backend.

[Amazon Web Services wird verwendet]

Mit wachender Codebase erhöht sich die Zeit, die notwendig ist um neue Funktionen zu entwickeln und zu implementieren. Dies liegt zum einen daran das die Anzahl der Klassen und Funktionen deutlich höher ist als am Anfang und sich dadurch viele Abhängigkeiten entwickelt haben. Des Weiteren steigt der Aufwand sich in den Quellcode einzuarbeiten. Vereinfacht wird dieser Prozess indem im Frontend die Funktionen und Klassen in logisch getrennte Bausteine geteilt werden. Diese sogenannten Komponenten werden an mehreren Stellen verwendet und ermöglichen ein schnelles entwickeln. Jedoch führt die Wiederverwendung von Komponenten dazu, dass bei kleinen Veränderungen dieser Änderungen an mehreren Stellen entsteht. Diese Abhängigkeiten macht es mit steigender Menge an Klassen und Funktionen immer komplexer weitere Funktionen umzusetzen, ohne bestehende Logik zu verändern. Hinzukommt, dass neben den eigenen Funktionalitäten externe Logik genutzt wird, welche im Frontend durch npm Packages installiert wird. Diese werden nur in Teilen der Applikation verwendet, stehen jedoch der gesamten Webseite zur Verfügung und müssen daher auch einheitlich deployed werden. Die externen Pakete verlangsamen jedoch die Bereitstellung der Applikation, da diese während des Prozesses installiert werden müssen. Für eine schnelle Entwicklung ist es somit wichtig schnelle Deploymentprozesse zu entwickeln und die Zahl der externen Pakete auf das nötigste zu begrenzen.

Um in Zukunft eine schnelle Weiterentwicklung der Applikation sicherzustellen, hat Plura-Polit entschlossen den aktuellen Aufbau in eine Mircoservices Architektur zu ändern und die

gesamte Plattform in inhaltlich getrennte Module zu teilen.

1.2 Zielsetzung

Schon im Jahr 2005 hat Peter Rodgers auf der Web Services Edge Conference über Micro-Web Services referiert. Er kombinierte die Konzepte der Service-Orientierten-Architektur mit den der Unix-Philosophie und sprach von verbundenen REST-Services. Er versprach sich dadurch eine Verbesserung der Flexibilität und der damalige Service-Orientierten-Architektur. Erstmals 2011 wurde dieser Ansatz als Microservices bezeichnet. Seit dem gibt es einiges über Microservices zu lesen auch einige Fachliteratur widmen sich diesem Thema. Insbesondere die Abgrenzung zwischen SOA und Microservices ist gut beschrieben. Somit ist die Definition und die Charakteristiken von Microservices gut beleuchtet. Des Weiteren gibt es einige Beispiele von bekannten Unternehmen, wie Netflix oder Amazon, die die Herausforderungen der Überführung ihres Systems zu eine Microservices Architektur beschreiben.

Trotz der aktuellen Informationslage ist jedoch noch relative unbekannt, ob auch junge Unternehmen Microservices umsetzen sollen und welche Bedingungen dafür erforderlich sind. Daher gibt es kaum Erfahrungen, die es PluraPolit ermöglicht abzuschätzen, ob sich eine Umstellung zum aktuellen Zeitpunkt lohnt. Auch gibt es keine Einschätzung darüber, welche Eigenschaften ein Unternehmen erfüllen muss, um Microservices umzusetzen.

Aus diesem Grund ist es Ziel dieser Arbeit für PluraPolit die Bedingungen zu ermittelt, die für die Umstellung erforderlich sind und eine klare Bewertung für diese zu geben. Insbesondere das ausarbeiten der Bedingungen an ein Unternehmen soll PluraPolit und anderen Start-Ups helfen zu bewerten, ob sich eine Umstellung lohnt.

1.3 Vorgehen

Die Arbeit teilt sich in drei Bereiche auf: Theoretischer Rahmen, Methodik und Auswertung. Die einzelnen Kapitel sind aufbauend und beziehen sich auf einander. So wird im ersten Abschnitt die theoretische Grundlage für Microservices gebildet. Es werden einzelne wichtige Merkmale von Microservices beleuchtet und beschrieben. Außerdem wird ein Vergleich zwischen der aktuellen Software-Architektur und Microservices erstellt. Anschließend werden aus den

Merkmale und dem Vergleich wichtige Bedingungen für Microservices abgeleitet, welche die Grundlage für die Einschätzung sind.

[Begründung für die Methode des Experteninterviews]

Im nächsten Abschnitt werden diese Bedingungen im Rahmen einer qualitativen Befragung von Experten eingeschätzt und bewertet. Hierfür werden Interviews durchgeführt. Es wird beschrieben welche Experten ausgewählt werden und welche Expertise sie mitbringen. Des Weiteren werden die einzelnen Interviewfragen beschrieben und dessen Zusammenhang zur Zielsetzung erklärt. Dadurch wird klar welchen Einfluss die Aussagen der Experten auf die Einschätzung für PluraPolit hat.

Abschließend werden die Aussagen aus den Befragungen mit der theoretischen Ausarbeitung verglichen und auf PluraPolit bezogen. Hierfür wird die Umsetzbarkeit für das junge Unternehmen hinterfragt und die Auswertung diskutiert. Neben Microservices wird auch SOA als alternative Lösungen vorgestellt und die Umstellung in einem weiteren Kontext betrachtet. Beendet wird dieser Abschnitt mit einer Einschätzung für PluraPolit, ob es ihre Software-Architektur in der aktuellen Situation zu einer Microservices Architektur ändern sollte.