

3.5. Ergebnisse der Interviews

3.5.1. Frage 1

Haben Sie das Gefühl, dass es Bedingungen gibt, die PluraPolit erfüllen sollte, bevor es ihre Softwarearchitektur zu einer Microservicearchitektur umstellt und wenn ja, welche Bedingungen empfinden Sie als wichtig?

Alle drei Experten haben diese Frage mit Ja beantwortet und weiterführend Bedingungen genannt (siehe Anhang A.1, S. I, Anhang A.2, S. VI und Anhang A.3, S. XII).

Sowohl Sebastian Schlaak, als auch Alexander Troppman haben auf die Frage geantwortet, dass sich erst Microservices in betracht ziehen, wenn inhaltlich unterschiedliche Anwendungen vorliegen. So beschreibt Herr Schlaak in seinem Interview: *„ich glaube, dass wäre eine Bedingung, wenn man sagt: [...] ich habe etwas [eine neue Funktion], was [...] ein ganz anderen Zweck erfüllt [...]“* (siehe Anhang A.3, S. XII). Herr Troppman fasst es in seinem Interview wie folgt zusammen: *„Also wie gesagt, ich brauche eine technische Trennung [...]“* und verwies auf die Trennung zwischen unterschiedlichen logischen Abläufen (siehe Anhang A.2, S. VII).

Als zweite Bedingung wurde die Wirtschaftlichkeit genannt. Sie wurde einmal von Christoph Rahles und Alexander Troppman hervorgehoben. Folglich führte Herr Rahles in seinem Interview an, dass es jemanden geben muss, *„der [...] sich die Frage stellt: Ist es wirtschaftlich, Ja oder Nein?“* (siehe Anhang A.1, S. I). Herr Troppman hielt es wie folgt fest: *„[...] ich muss ein Business Case haben, dass sich das auch lohnt.“* (siehe Anhang A.2, S. VII). Demnach lässt zu schlussfolgern, dass der wirtschaftliche Nutzen ein wesentliche Voraussetzung für den Einsatz von Microservices ist.

Allein Herr Rahles verwies bei dieser Frage darauf, dass die Entscheidung für Microservices vom “Reifegrad des Geschäftsmodells” abhängt (siehe Anhang A.1, S. I).

3.5.2. Frage 2

Microservices ermöglicht es Team unabhängig von einander an unterschiedlichen Services zu arbeiten. Welche Rahmen Bedingungen sehen Sie als notwendig, dass Teams separat voneinander Arbeiten können?

Alle drei Kandidaten bestätigten die Aussage, dass Teams unabhängig voneinander arbeiten. Herr Rahles führte jedoch den vermerk hinzu, dass die Teams stets zu einem Unternehmen gehören und nie ganz unabhängig sind.¹⁷

¹⁷ „Auch diese Teams gehören zu einem Unternehmen [...], das heißt unabhängig voneinander sind sie nie.“ (siehe Anhang A.1, S. III)

Herr Schlaak und Herr Troppman führten an, dass insbesondere Schnittstellen, definiert und beschrieben sein müssen.¹⁸

Weiterführend wurde von Herr Troppman eine Struktur hervorgehoben, die Abläufe genauer beschreibt.¹⁹ Diesem Ansatz schloss sich Herr Rahles an, in dem er die Notwendigkeit von Projektmanagement ansprach.²⁰ Daraus lässt sich folgern, dass Struktur notwendig ist, dass Teams separat voneinander Arbeiten können.

3.5.3. Frage 3

Gibt es in Ihren Augen irgendwelche technischen Anforderungen die PluraPolit erfüllen sollte?

Als wohl wichtigste technische Anforderung stellt sich das Monitoring heraus. Demnach beschrieben alle drei Experten, die Notwendigkeit die Kommunikation der Services zu Monitoren (siehe Anhang A.2, S. XI und Anhang A.3, S. XIV). Besonders Herr Rahles verwies auf diese Bedingung hin.²¹

Auch Sicherheit wurde von Herrn Rahles als Grundvoraussetzung gesehen. Dabei sah er es jedoch als Grundvoraussetzung für jegliche Infrastruktur und nicht nur explizit für Microservices.²²

Herr Schlaak und Herr Rahles hoben beide ein gut implementiertes Fehlermanagement hervor. So beschrieb Herr Schlaak und Herr Rahles, dass Services „vernünftige Fehlermeldungen“ zurück geben sollten (siehe Anhang A.3, S. XIV und Anhang A.1, S. V).

Die technischen Anforderungen Monitoring, Sicherheit und Fehlermanagement, lassen schlussfolgern, dass Kenntnisse in der Implementierung und Managen vorausgesetzt ist. Eine Aussage, die von Herrn Troppman noch einmal bestätigt wurde.²³

Herr Rahles verstärkt ebenfalls diese Schlussfolgerung, indem er die „heutzutage geltenden Best Practices als Anforderung“ sieht (siehe Anhang A.1, S. IV). Er führt diese Bedingung jedoch noch weiter aus und zählt automatische Tests, sowie ein automatischer Integration-

¹⁸ „Ganz wichtig ist, dass die Schnittstellen der Services entsprechend gut beschrieben sind [...]“ (siehe Anhang A.3, S. XIII); „[...] die Teams müssen sich einig sein über welche Schnittstelle die Services kommunizieren.“ (siehe Anhang A.2, S. IX); „Schnittstellen nach außen müssen geklärt sein.“ (siehe Anhang A.2, S. X)

¹⁹ „[...] zumindest benötigt man Strukturen, sodass man weiß, was wie abläuft.“ (siehe Anhang A.2, S. X)

²⁰ „[...] es ist einfach sinnvoll zu gucken, dass [...] die Teams entsprechend des Projektmanagements richtig aufgestellt sind.“ (siehe Anhang A.1, S. IV)

²¹ „Monitoring ist das A und O. Das heißt mit Anstieg der Komplexität, muss ich sicher sein, dass ich auch diese Komplexität hinreichend überblicken kann.“ (siehe Anhang A.1, S. V)

²² „Sicherheit gehört für mich zu den Grundvoraussetzungen, egal über welche Architektur wir reden.“ (siehe Anhang A.1, S. V)

²³ „[...] es [braucht] einen Entwickler, der sich mit dem Aufbau der Infrastruktur auskennt.“ (siehe Anhang A.2, S. XI)

und Deployment-Prozess als erforderlich, um nicht Zeit beim Aussetzen des Systems zu verlieren (siehe Anhang A.1, S. IV).

3.5.4. Frage 4

Ein Start-up zeichnet sich dadurch aus, dass es insbesondere in der Anfangsphase zu vielen Veränderungen in der ursprünglichen Geschäftsidee kommt. Microservices auf der anderen Seite zeichnen sich durch aus, dass sie feste Schnittstellen und Kontextgrenzen besitzen. Meinen Sie, dass trotzdem Microservices in einem dynamischen Umfeld eingesetzt werden sollten?

Christoph Rahles antwortete auf diese Frage, „*Da gibt es glaube ich kein Richtig oder Falsch.*“ und verdeutlichte, dass die Frage immer im Kontext der wirtschaftlichen und technologischen Lage des Start-up getroffen werden muss (siehe Anhang A.1, S. II).

Alexander Troppman vertrat die Meinung, dass Microservices grade im dynamischen Umfeld eingesetzt werden sollten. So gab er an, dass Microservices auf Grund des einfachen austauschen von Services im dynamischen Umfeld Sinn ergeben. „*Ich kenne viele Start-ups, die von Anfang an Microservices eingesetzt haben, weil grade Microservices so sind, dass man verschiedene Services austauschen kann. Also grade im dynamischen Umfeld machen Microservices Sinn.*“ (siehe Anhang A.2, S. VII)

Sebastian Schlaak antwortete, dass er mit einem Monolithen anfangen würde, da man dadurch deutlich schneller ist und führte weiter fort, dass die Vorteile von Microservices „*erst in der späteren Skalierungsphase [...] so richtig zum tragen kommen*“ (siehe Anhang A.3, S. XIV).

Die Aussage, dass man mit einem Monolithen anfängt, wurde von Herr Rahles bekräftigt. So äußerte er sich, „*dass man [meistens] mit einem Monolithen anfängt*“ und später, wenn „*es sehr drückend wird*“, umstellt (siehe Anhang A.1, S. II).

Herr Troppman bestärkt die Aussage, dass die Vorteile von Microservices erst später zum tragen kommen, indem er in seinem Interview sagte, dass eine Umstellung zu Microservices „*sich erst mittel bis langfristig [lohnt]*“ (siehe Anhang A.2, S. IX).

Sowohl Herr Rahles, als auch Herr Troppman antworteten, dass die Unternehmensziele die Entscheidung zur Umstellung beeinflussen. So verdeutlichte Herr Rahles, dass die Entscheidung auch von der „*Strategie der Firma*“ abhängt (siehe Anhang A.1, S. II).

Nach Herrn Schlaak sollte besonders, dann über eine Microservicearchitektur nachgedacht werden, wenn neue „*Funktionen [...] nichts mehr mit dem Kerngeschäft zu tun haben [...]*“ (siehe Anhang A.3, S. XV). Diese Aussage wird von Christoph Rahles geteilt. „*Meistens ist es so, dass man mit einem Monolithen anfängt und irgendwann an ein*

Punkt kommt, [...] wo man sagt, es macht Sinn Dinge auszulagern.“ (siehe Anhang A.1, S. II).

3.5.5. Frage 5

Mit Ihrem aktuellen Wissensstand, welche Softwarearchitektur empfehlen Sie PluraPolit?

Alle drei Experten haben eine Monolith empfohlen (siehe Anhang A.2, S. XI, Anhang A.2, S. XI und Anhang A.3, S. XVI).

Alexander Troppman und Sebastian Schlaak haben weiterhin empfohlen das Backend und das Frontend voneinander zu trennen. So hat Sebastian Schlaak empfohlen im Frontend ein „*modernes JavaScript Framework*“ zu verwenden und das „*Backoffice*“, der Mitarbeiter von der Darstellung der Endkunden zu trennen (siehe Anhang A.3, S. XVI). Beide haben diesen Ansatz empfohlen, um die Flexibilität zu haben das Frontend bzw. das Backend auszutauschen.

Sowohl Herr Rahles, als auch Herr Troppman haben angegeben, dass sie einen Monolithen empfehlen, da es keine separate Geschäftsdomäne gibt, die eine Aufteilung in Services rechtfertigt.²⁴

Des Weiteren hat Herr Rahles ein Monolithen empfohlen, da nicht genügend Entwickler vorhanden sind (siehe Anhang A.1, S. VI).

²⁴„*Ich glaube, dass es keine separate Business Domain gibt, wo man sagt, die muss zwangsläufig ausgelagert werden.*“ (siehe Anhang A.1, S. VI); „*Ich sehe da vom fachlichen her nichts, was eine Microservicearchitektur rechtfertigt.*“ (siehe Anhang A.2, S. XI)