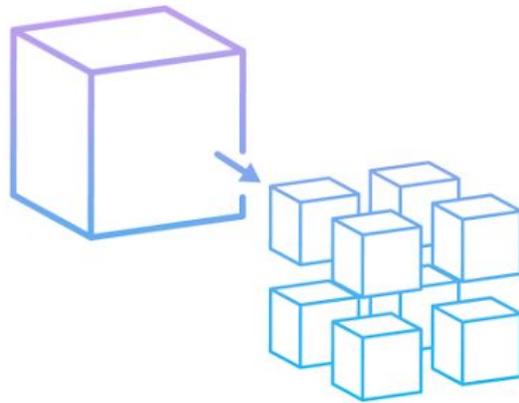
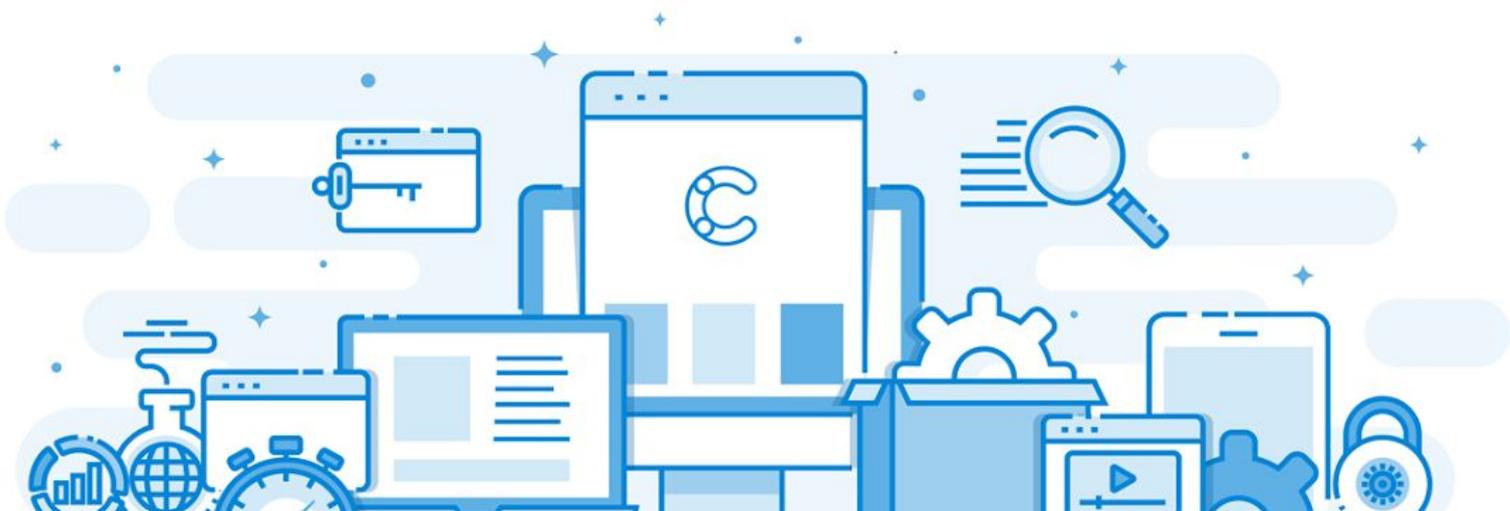


Auswahl und Anpassung von IT-Diensten



Microservice Architektur im Cloud

Rodchenkov Mykhailo · Nikita Ivachshenko
Pouaha Ornella · Tresor Arsene, **WS 2018/19**



Inhaltsverzeichnis

Projektfocus und Problemstellung.....	3
CMMI ARD SP 1.1.....	4
CMMI ARD SP 1.2.....	6
CMMI ARD SP 3.1.....	7
CMMI ARD SP 3.3.....	10
CMMI ARD SP 3.4.....	13
CMMI SSAS SP 1.1.....	15
CMMI SSAS SP 1.2.....	18
CMMI SSAS SP 2.2.....	19
CMMI SSAS SP 2.2.....	21
CMMI SSAS SP 3.2.....	23
Analytic Hierarchy Process.....	26

Projektfokus und Problemstellung

foliage ist ein junges Start Up, welches ein modernes soziales Netzwerk entwickelt. Der Fokus liegt bei Projekten und Aktivitäten, die die Benutzer erstellen und in Teams weiterentwickeln können. Es handelt sich um ein evolutioniertes Ticketsystem, orientiert auf Studenten, Schüler und engagierte aktive Menschen.

Im Rahmen von Auswahl und Anpassung von IT-Diensten beschäftigen wir uns mit dem Problem des Umstiegs von monolithischer Architektur auf die **Microservice Architektur**. Um die Überführung möglichst effizient gestalten zu können, wurde es sofort entschieden, die neuen Services direkt im **Cloud** zu deployen. Für den Umstieg auf Microservices wurde eine Refactoring-Strategie entwickelt, die es ermöglicht, schnell und mit minimalem Aufwand das Refactoring durchzuführen. Die Refactoring-Strategie sorgt für das Umsetzen von neuen Funktionalitäten als unabhängige entkoppelte Services und das Ausbauen von der bestehenden Hauptanwendung. Die Cloud-Umgebung ist perfekt geeignet für die Kollaboration zwischen Entwicklern und die Verteilung von physischen Ressourcen. Das stellt eine störungsfreie Entwicklung sicher.

Als nächstes haben wir uns mit der Auswahl von Cloud-Anbietern beschäftigt. Der Markt bietet momentan eine Menge von unterschiedlichen **Cloud-Diensten**. Wir haben die wichtigsten für uns Anforderungen an Lieferanten erfasst und darauf basierend möglichst objektiv mit Hilfe von AHP die Lieferanten verglichen.

ARD 1.1 Elicit Stakeholder Needs

Mykhailo Rodchenkov

Microservices werden als Allheilmittel bei der Modernisierung von Software-Architekturen gehandelt. Auch wir haben uns für diese Herausforderung entschieden. Laut Best Practices sind die Microservices am besten im Cloud zu pflegen und zu entwickeln. Deswegen wurde die Entscheidung getroffen, die schon implementierte Services und die restliche Anwendung ins Cloud zu überführen und dort zu deployen.

Die Nutzung von Cloud-Diensten bringt eine Menge von Vorteilen. Wir wollen sie alle möglichst effizient und produktiv ausnutzen um hohe Qualität unserer Anwendung erreichen zu können. Wir unterscheiden die Kundenanforderungen in **3 Ebenen**.

Anwendungs- und Benutzerebene. In erster Linie steht die clientseitige Anwendung, die von den Mitarbeitern benutzt wird. Sie muss über einem geeigneten Niveau an Benutzerfreundlichkeit verfügen (Metriken zu Effektivität, Effizienz, Zufriedenheit). Messungen werden quantitativ anhand von Beobachtungen an Benutzern durchgeführt. Außerdem muss das System den internationalen Standards entsprechen, z.B. nach EN ISO 9241-110 "Grundlagen der Dialoggestaltung".

Es muss möglich sein, das System in mehrere unabhängige Umgebungen (Entwicklungs-, Test- und Deployment Umgebung) aufzuteilen. Die Aufteilung muss entweder durch Separation von Hardware erfolgen oder durch entsprechende Software. Die Wartung und Instandhaltung von genannten Umgebungen muss mit dem gleichen Verfahren geschehen, d.h. die Umgebungen müssen sich nur in Anwendungsszenarien unterscheiden und nicht in Bedienung.

Die Voraussetzung für die weitere Entwicklung unserer Anwendung sind die IDEs (Eclipse und/oder IntelliJ Idea für Java-Entwicklung mit etabliertem Framework Spring und Anwendungsserver JBoss), Anwendung für die Administration und Gestaltung von SQL(MySQL) und NoSQL(MongoDB) Datenbanken z.B. IntelliJ

Datagrip oder Plug In für Eclipse, Tools für die Entwicklung von grafischer Oberfläche (Adobe Photoshop ab CS6 und Adobe Illustrator).

Für die interne und externe Kommunikation werden E-Mail-Server und Nachrichtendienst wie Discord oder Slack.

Das System muss nicht unbedingt über all diese Anwendungen verfügen, sondern es muss die Installation und Wartung dessen unterstützen.

Administrationsebene. Wir erwarten vom System die Möglichkeit, jedem Teammitglied eine entsprechende Rolle zuordnen zu können. Mit dieser Rolle würde man die Autorisierungsrechte bekommen und nur auf die für ihn zugelassenen Ressourcen zugreifen können. Es wird vorausgesetzt, dass das System alle internen Änderungen im Log speichert. Dazu gehören alle Login und Logout, geänderte Einstellungen im System und einzelnen Anwendungen sowie die Änderungen an Dateien und Ressourcen. Das System muss rund um die Uhr und von überall verfügbar sein, das heißt dass das Uptime und Response Time eine führende Rolle spielen.

Analytische Ebene. Dazu gehören in erster Linie die etablierten Überwachungs- und Monitoring Tools. Es muss möglich sein, den Status des Systems jederzeit zu prüfen. Dazu zählt das Traffic, Verwendung von physischen Ressourcen und die statistischen Daten über einen gegebenen Zeitraum. Es wird benötigt, eine Liste von allen angemeldeten Benutzern dargestellt zu bekommen.

Fazit. Als endgültige Variante von Anforderungen für das Projekt wurde folgende Liste erfasst und nach Priorität sortiert:

1. Unterstützung von third party Application
2. Kosten
 - a. Einführungskosten
 - b. laufende Kosten
3. Support, Uptime & Response time
4. Benutzerfreundlichkeit & Einstiegsniveau
5. Nützlichkeit der angebotenen Technologien

Zuerst wurde die Priorität einzelner Anforderungen subjektiv geschätzt. Die finale Prioritätenschätzung von Kriterien findet in AHP statt.

ARD 1.2 Develop and Prioritize Custom Requirements

Arsene Tresor

Kundenerwartungen sind in der Sprache des Kunden formulierte Anforderungen an eine durch einen Dienstleister erbrachte Leistung oder an ein durch einen Lieferanten geliefertes Produkt. Ursache von Kundenerwartungen sind die ↑ Kundenbedürfnisse, d.h. das Verlangen der Kunden nach der Behebung eines subjektiv empfundenen Mangels. Typisch für Kundenerwartungen ist, dass sie aus der Perspektive des Kunden und in seiner Sprache formuliert sind. Für die Durchführung eines Projekts ist es deshalb notwendig, aus den Kundenerwartungen zunächst in der Sprache des Lieferanten formulierte ↑ Kundenanforderungen abzuleiten. Aus diesen können dann messbare und überprüfbare ↑ Abnahmekriterien entwickelt werden, die eine reproduzierbare Abnahmeprüfung ermöglichen.

ARD SP 3.1 Establish Operational Concepts and Scenarios

Mykhailo Rodchenkov

Develop operational concepts and scenarios that include operations, installation, maintenance, support and disposal as appropriate

Beim Umbau einer monolithischen Architektur in eine lose gekoppelte Architektur entstehen viele technische und organisatorische Vorteile, wie z.B. höhere Sicherheit, lose Kopplung von Komponenten und als Konsequenz – einfacheres Testen und Entwicklung von Modulen. Die Module werden in Services separiert, sie besitzen eigene Aufgabe und arbeiten unabhängig, aber sie sind mit den anderen Modulen verbunden. Die Kommunikation zwischen den Diensten erfolgt via HTTP Request oder Message Queue. Bei der Einführung von Microservice Architektur wurde eine Systematische Refactoring-Strategie verwendet, um sie möglichst fehlerfrei gestalten zu können.

1. Zuerst werden die neuen Funktionalitäten nicht mehr innerhalb von monolithischer Anwendung entwickelt, sondern werden in komplett eigenständige Microservices umgesetzt. Dieses Service wird unabhängig von der Hauptanwendung entwickelt, bereitgestellt und skaliert.
2. Im nächsten Schritt werden die GUI (Benutzerschnittstelle) und der Anwendungsschicht (Business Logic Layer) separiert. Danach können beide Schichten selbständig entwickelt werden. Ab diesem Schritt kann das Entwicklerteam eventuell aufgeteilt werden, da es nun nicht mehr um eine große Anwendung handelt, sondern um zwei (oder mehr) kleinere Module.
3. Anschließend werden die restlichen Komponenten der monolithischen Anwendung in Microservices umgebaut. Dabei müssen die Entwickler die bestehenden Module extrahieren. Der Komplexitätsgrad kann dabei sehr hoch sein und deswegen müssen die Entwickler zuerst mit dem Modul starten, dessen Extraktion am einfachsten ist oder bei dessen Extraktion der größte Mehrwert entsteht.

Für die Instandhaltung von Microservices bieten die Cloud Anbieter etablierte Docker- und Kubernetes-Technologien. Das erlaubt es, den Remote Service

effektiv sowie effizient in eine neue Generation zu führen und somit die Fernwartung und Instandhaltung zu optimieren. Das Deployment von Anwendung im Cloud wird mittels Anbieter Tools (CodeDeploy bei Amazon, Azure App Service bei Microsoft usw.) und mit "Pipelines" (YAML) durchgeführt.

Define the environment in which the product will operate, including boundaries and constraints

Das Cloud wird in drei Nutzungsszenarien aufgeteilt: Entwicklungs-, Test- und Deployment Umgebung. Dies hat eine Reihe von Vorteilen, wie z.B. Bedürfnisanpassung, Kostenersparnis und schnelle Einrichtung. Der Hauptgrund ist die vollständige Trennung von Test- und Entwicklungsumgebung. Dadurch soll potentieller Schaden für den produktiven Betrieb vermieden werden.

Review operational concepts and scenarios to refine and discover requirements

Das Betriebskonzept wurde überprüft und darauf bezogen wurde eine Liste von Anforderungen erfasst. Dies enthält organisatorische sowie technische Anforderungen. Zu den organisatorischen und finanziellen Anforderungen gehören Schulung, Support, Kosten, Leistungen, Chancen und Risiken; zu den technischen Anforderungen gehören Uptime, Reaktionszeit, Sicherheit, Unterstützung von bekannten Technologien usw.

Develop a detailed operational concept, as candidate solutions are identified and product component solutions are selected by the supplier, that defines the interaction of the product, the end user, and the environment, and that satisfies operational, maintenance, support, and disposal needs

Das Betriebskonzept wird in folgende Aspekte unterteilt:

1. Erklärung des System und ihre Ziele
2. Strategien, Taktiken, Richtlinien und Einschränkungen, die das System beeinflussen
3. Zuständigkeitserklärung und delegierte Befugnisse
4. Prozesse zur Initiierung, Entwicklung und Wartung des Systems

Beim System handelt es sich um eine Cloud-basierte Entwicklungs- und Produktionsumgebung. Das System besitzt mehrere Nutzungsszenarien und wird zum Entwickeln, Testen, Demonstrieren und zur Release Freigabe

verwendet. Es wird eine klare Trennung zwischen den Anwendungsbereichen gefordert um die maximale Sicherheit von Benutzerdaten sowie die Sicherung von Source code und dessen Versionen beschaffen zu können.

Das System muss rund um die Uhr zur Verfügung stehen und es muss möglichst wenig Ausfälle geben. Darüber hinaus muss noch das Backup-System das planmäßige Datensicherung gewährleisten. Die Entwicklungsumgebung wird nur für die Entwickler, Testentwicklung für Tester und Entwickler und Backupsystem für Tech Lead und IT-Administrator zur Verfügung gestellt, um keine technischen Konflikte zu auszulösen.

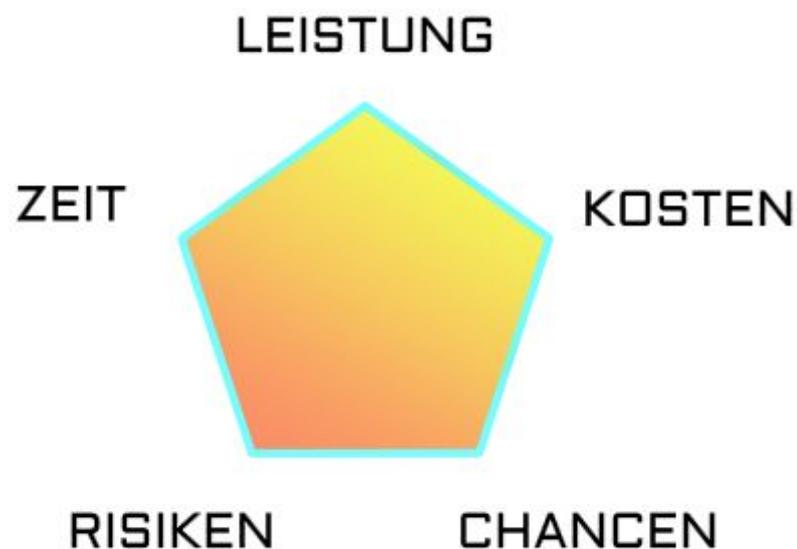
Die Gesamte Verantwortlichkeit für das System trägt der Tech Lead, der IT-Architekt ist für eventuelle Abweichungen im Plan verantwortlich und soll mit dem Tech Lead alle beeinflussten Merkmale besprechen. Allen anderen Beteiligten wird eine exakte Rolle zugeordnet, die dann dem System dabei hilft, die Autorisierungsrechte des Benutzers feststellen zu können.

Das aktuelle System muss schrittweise in die neue Umgebung überführt werden. Dafür wird eine von Tech Lead zu bestimmende Deployment-Strategie verwendet. Diese wird je nach Cloud-Anbieter vor Ort geklärt. Auf der Entwicklungsumgebung muss die Hauptanwendung in einzelne Dienste (Microservices) extrahiert werden. Dafür wurde eine detaillierte Refactoring-Strategie entwickelt. Die Wartung des Systems betrifft insbesondere die Entwickler, dabei handelt es sich um saubere und fehlerfreie Entwicklung. Die Richtlinien für die Entwicklungs werden in Confluence(oder in einer alternativen Umgebung) erfasst. Das Logging, Automatisierung und Integration von einzelnen Modulen ins System wird mit Jenkins (siehe auf [Wikipedia](#)) erfolgen. Damit kann auch der Status jeder Umgebung verfolgt werden. Falls es Konflikte oder Fehler auftreten, ist der Tech Lead für das Fixen verantwortlich. Im weiteren Verlauf soll noch Security Manager eingestellt werden. Er wird die Verantwortung für Sicherheit und das Security Compliance tragen.

Agenda

Anforderungen Analyse um Stakeholder Bedürfnisse zu balancieren und die Einschränkungen zu einstellen.

- Die Fünf wichtige Parameter für unsere Unternehmen sind Plattform-Leistung, Zeit für die Entwicklung, Kosten für diese Entwicklung und dazugehörigen Risiken und Chancen. Für jede Entscheidung/Projekt/Projektteil werden diese Abhängigkeiten und Verhältnisse berücksichtigt.



- Außerdem die Anforderungen werden analysiert, um ein angemessenes Gleichgewicht zwischen Kosten, Zeitplan, Leistung und anderen Faktoren von Stakeholder zu bestimmen. Modelle und Simulationen können verwendet werden, um die Auswirkungen abzuschätzen, die Anforderungen an diesen Faktoren haben. z.b.: Um neue Benutzer zu involvieren, entwickeln wir neue Funktionalität, recherchieren die Anforderungen und ihre Größe, dabei wird die Nützlichkeit berücksichtigt. Also wenn wir 200.000 neue Nutzer engagieren

möchten, und es ist möglich in halbes Jahr zu realisieren, dabei die Gewinn von Werbung und Donations wird größer als Verlust von Entwicklung und Implementierung, dann ist das Projekt realisierbar und unseren Stakeholder werden dabei interessiert.

- Risiken werden sofort ermittelt durch die Beteiligung von Stakeholdern in verschiedenen Phasen des Produktlebenszyklus. Wenn die Risiken als inakzeptabel angesehen werden, können die Anforderungen überarbeitet oder neu priorisiert werden, um das Verhältnis von Kosten, Zeitplan und Leistung zu verbessern. Die Idee ist, dass wir auf jede Stelle mit unsere Beteiligten/Interessengruppen in Kontakt stehen (abhängig von ihrer Unterstützung im Projekt).

- Es ist sinnvoll eine Kosten-Nutzen-Analyse durchzuführen, um die Auswirkungen der Anforderungen auf die Akquisitionsstrategie und die Kosten und Risiken des Akquisitionsprojekts zu bewerten. Da die Kosten und Funktionalität von uns interessierende Software, normalerweise, vordefiniert ist, die Analyse und Konsequenzen unserer Wahl vorhersagbar sind. Es wird auch hilfreich für Lieferanten Auswahl (SSAD SP 2.3)

Bewertung der mit den Anforderungen verbundenen Risiken:

Um effiziente Analyse durchzuführen, wir implementieren 12-Schichtige Risiko-Prioritäts Modell.

Eintrittswahrscheinlichkeit (zum Zeitpunkt der Bewertung):

1. Zweifelhaft <16% (Mu, Lambda)
2. Eher wahrscheinlich <17-50% (Kappa, Iota, Theta, Etha,)
3. Wahrscheinlich <51-84% (Zetha, Epsilon, Delta, Gamma)
4. Unvermeidlich >85% (Beta, Alpha)

Schaden bei Eintritt:

- C. Gering, nicht relevante Prozesse gefährdet.
- B. Mittel, wichtige Prozesse werden gefährdet.
- A. Schwer, gefährlich für gesamtes Projekt.

Beispiele:

2C – Priorität *Lambda*: Innovationen aus der Designabteilung entsprechen nicht zu der Ergonomie und der UX Normen.

2B – Priorität *Etha*: Engagement von Mitarbeitern aus dem Outsourcing, da wir nicht ausreichende Anzahl an Mitarbeitern haben, um rechtzeitig das Projekt zu beenden.

3B – Priorität *Delta*: Die Plattform Update stimmt nicht mit dem geplanten Zeitpunkt überein.

4A – Priorität *Alpha*: Nicht ausreichende Finanzierung und mangelnde Eigenkapital.

Um vollständige Programm zum Erkennung und Analyse von Risiken zu erstellen, können weitere Praktiken aus der Prozessbereich Risikomanagement (RSKM CMMI-ACQ) verwendet werden.

Nach der Festlegung von Lieferantenvereinbarung wird sie jetzt anerkannt und für den Einsatz bereitgestellt. Jedoch vor dem Einsatz werden noch ein paar Punkte analysiert.

1. Risiko ein nicht funktionierendes Produkt am Ende zu bekommen. Hier werden potenzielle Risiken der Produkt Anforderungen analysiert und ermittelt. Im Bezug auf unser Projekt könnte es vorkommen, dass

- Die Produkte einfach zu teuer für unser Budget sind
- Die Software nicht in der gewünschten Version verfügbar oder sehr begrenzt sind
- Trotz Aufwand die Kommunikation im Team nicht funktioniert und das Projekt schief geht

2. Angemessenheit und Vollständigkeit der Anforderungen untersuchen. Dafür erstellen wir ein Szenario mit folgenden Stakeholdern:

- Amazon web Services abgekürzt AWS (einer für uns potenzielle Cloud -Lieferant)

- Uns Mitarbeiter vom Start-up „Foliage“ So sieht das Szenario aus:

- **Foliage:** Wir brauchen eine Plattform mit Datenbanktechnologien Fähigkeit
- **AWS:** Wir können Ihnen AWS Outpost für die Ausführung lokale Infrastruktur und Amazon RDS on VMware für die automatische lokale Datenbankverwaltung anbieten.
- **Foliage:** Super! Diese Komponente sind von enormen Bedeutung für unser Projekt. Dafür können wir 2600\$ ausgeben.
- **AWS:** Der preis würde schon gehen , da die Installation und Bereitstellung circa 2450\$ kostet.
- **Foliage:** Könnten Sie in zwei Wochen das alles schaffen?
- **AWS:** Ja das schaffen wir. Und wie würden Sie bezahlen? In Raten oder auf einmal?
- **Foliage :** Monatlich mit jeweils 204\$
- **AWS:** Alles klar . Wollen wir jetzt den Vertrag abschließen?
- **Foliage:** Ja natürlich.
- **AWS:** Schön, dass alles schnell geklappt hat.
- **Foliage:** Wir freuen uns auch darüber.
- **AWS:** Dann sehen wir uns in zwei Wochen wieder.

3. Bewerten von Produkt- und Produktkomponentenlösungen.

Für unser Microservice Architektur Projekt, da es sich hauptsächlich um Software Produkte handelt, werden wir bei der Bewertung folgende Aspekte betrachten:

- Funktion und Leistung: Die Software-Produkte sollten funktionieren wie erwartet und die Leistung in dem entsprechenden Bereich erbringen
- Benutzerfreundlichkeit : Die Software sollten geschützt aber auch zugänglich einfach anzuwenden sein. Sowohl für unserer Unternehmen als auch für die Kunden, die unser System benutzen.

Develop a list of potential suppliers

1. Amazon Web Services (*aws.amazon.com*)
2. Microsoft Azure (*azure.microsoft.com*)
3. Google App Engine (*appengine.google.com*)
4. Google Cloud Platform (*cloud.google.com*)
5. Joyent (*joyent.com*, by Samsung)
6. IBM Cloud Computing (*ibm.com/cloud*)
7. Apache CloudStack (*cloudstack.apache.com*)
8. Kahu (*lightcrest.com*, by Lightcrest)
9. Platform9 (*platform9.com*)
10. KeyInfo (*keyinfo.com*)

Bei dem Umstieg auf Microservice Architektur wurde entschieden, die Microservices im Cloud zu deployen. Basierend auf den Anforderungen an Cloud-Dienste, wurden aus der Liste vier Cloud Anbieter herausgearbeitet (Amazon Web Services (weiter AWS), Microsoft Azure (weiter Azure), Joyent, Kahu). In der unteren Tabelle sind diese Anbieter und ihre Eigenschaften detailliert dargestellt.

Kriterium	AWS	Azure	Joynet	Khau
Preis	Pay-as-you-go, ab \$2,430.90 (for a scalable app in containers)	Pay-as-you-go, ab VMs \$0.008/h , App Service \$0.013/h , Azure SQL \$0.021/h	Pay-as-you-go, ab \$1,296.13 (for a scalable app in containers)	K1: \$1,265.25 , K2: \$2,536.28 , K3: \$3,619.47 , K4: \$3,162.25 , K5: \$5,349.33 , K6: \$7,033.48
Cloud Typ	hybrid	hybrid	public	hybrid
Unternehmensgrösse	gross	gross	mittel (125 Mitarbeiter)	klein (unter 50 Mitarbeiter)
Weitere Dienste	S3 (storage), EC2 (Virtual Servers), RDS/ DynamoDB/ Aurora für Datenbanken	Machine Learning, SQL Azure, Kubernetes (administration, deployment)	Node.js, Converged Analytics, Multi-Cloud Kubernetes, Container and VMs	DevOps on Demand, Source Control, Full-Stack Engineering, Security Tools
Kunden	RyanAir, Atlassian, Siemens, Adobe, vodafone	HP, Adobe, Bing, Shell, Siemens, Allianz	Dell, Samsung, Iron.io, HashiCorp, opbeat	WonderHowTo, Tweak, Walks of Italy, ATCO Software
Auf dem Markt seit	Anfang 2006	Februar 2010	Ende 2004	2012
Kundenzufriedenheit	96%	98%	keine Daten	92%
Uptime	von 99.00% bis 99.99%	99.99%	bis 100%	keine Daten
Weitere Vorteile	Unterstützung von Ubuntu Server, CentOS, Microsoft Windows Server, Apache, Nginx, Open VPN, SSH	Trainings, dezentralisierte Infrastruktur, Logging und Metrics	Open Source, Gründer von Node.js, Container Native	geeignet für StartUps und kleine Unternehmen

Communicate with potential suppliers concerning the forthcoming solicitation

Wir haben mit jedem der vier ausgewählten Cloud-Anbieter Online-Kontakt aufgenommen und über nähere Informationen kommuniziert. Dies hat uns weitergeholfen, eine mögliche Zusammenarbeit basierend auf den repräsentierten Angeboten vorzustellen. Die letzten zwei Anbieter (Joynet und Kahu) zeigen wesentlich mehr Interesse an der Zusammenarbeit als AWS und Microsoft Azure. Die Anbieter haben jeweils ein Angebot für uns erstellt.

Verify participants who will evaluate supplier proposals

Die vorgestellten Angebote werden zunächst von IT-Abteilung in einer kleinen Runde von allen Entwicklern diskutiert und bewertet. Anschließend wird der Tech Lead die restlichen Aspekte mit Finanzabteilung besprechen und am Ende wird die Schlussfolgerung gezogen. Die Entscheidung wird letztendlich von dem Tech Lead zusammen mit dem Finanzleiter getroffen.

Verify participants in supplier negotiations

Die Verantwortlichkeit für eventuelle Planänderungen und Abweichungen in der Architektur- und Entwicklungskonzept trägt der IT-Architekt. Bei solchen Fällen ist es mit Tech Lead dringend zu kommunizieren. Der Tech Lead trägt die restliche Verantwortung sowohl für technische als auch preisliche Aspekte.

SSAD SP 1.2 Establish a Solicitation Package

Mykhailo Rodchenkov

Unser Anspruch ist der langfristige wirtschaftliche Erfolg und die kontinuierliche Zufriedenheit der Kunden. Wir positionieren uns auf dem Markt als innovatives Start Up im Bereich von sozialen Netzwerken und agilen Projektmanagement-Tools. Hierzu wollen wir neue Technologien und Produkte entwickeln. Dafür brauchen wir qualifizierte und innovative Lieferanten. Für unsere Zwecke möchten wir eine vertragliche und freundliche Beziehung mit den zukünftigen Lieferanten aufbauen. Von den Lieferanten erwarten wir eine dauerhafte und vertrauensvolle Zusammenarbeit, die es uns ermöglicht, unsere Produktqualität auf höchstem Niveau zu halten. Unten sind die wichtigsten für uns Kriterien für eine Partnerschaft:

- Internationale Positionierung, um uns auch im Ausland unterstützen zu können.
- Unterstützung von den neuesten IT-Technologien
- Tag- und Nacht Kundensupport
- Eventuelle Schulungen und Besorgung von up-to-date Dokumentationen
- Verwendung von effektivsten und modernsten Sicherheitstechnik, um Hackerangriffe verhindern zu können
- Interesse an Zusammenarbeit und eventueller Zusammenentwicklung
- Kosten nur für das, was verbraucht wird
- Die Ausfälle sind für uns kritisch und das liegt an den Lieferantenschultern (max. 0.25% oder 20-25 Std. im Jahr ist akzeptabel)
- Leben und handeln nach dem Null-Fehler-Prinzip
- Kontinuierliche Produkt- und Prozessverbesserung

Agenda

Entwicklung und Verfeinerung eines Verhandlungsplans für jeden der in Frage kommenden Lieferanten auf der Grundlage der Bewertung der Lieferanten und ihrer Vorschläge.

- **Rollen und Verantwortlichkeiten der Mitglieder des Verhandlungsteams**
2 Personen:
Technischer expert, der genau die Anforderungen vorliegt. Idealerweise der Leiter von Entwicklungsabteilung.
Sprecher, der unmittelbar die Verhandlung durchführt und mit Unternehmens finanzielle Aspekte bekannt ist.
- **Zu verhandelnde Schlüsselfragen:**
Zeitgrenzen bei Einführung und Implementierung von Lieferanten Software, Wartung und Instandhaltung Verfahren beim Hardware Services, Softwareberater und Trainings bei neuer Software Implementierung.
- **Die Verhandlungsreihenfolge:**
 - Zuerst werden die allgemeinen Anforderungen und Kosten, falls nicht vordefiniert, abgefragt werden bzw. eine Auswahl aus verschiedene Möglichkeiten durchgeführt wird.
 - Zweitens werden nicht entscheidende Elemente präzisiert. Bei der Auswahl aus einigen Diensten/Applikationen wird eine genaue Analyse durchgeführt und eine Entscheidung getroffen.
 - In der letzten Verhandlungsphase wird die Laufbahnplan für Implementierung festgestellt, Auszahlungsplan und Option definiert, Fristen und Meilensteinen festgesetzt.
- **Externe Faktoren**, die die Verhandlungen beeinflussen könnten; (z. B. andere ausstehende Geschäfte und strategische Pläne) sind schwer vorhersagbar, aber können durch ausreichende Unternehmens- bzw. Projektmanagement maximal reduziert werden.
- **Frühere Erfahrungen mit Lieferantenvereinbarungen** zur Ermittlung früherer Positionen und Probleme (und Verhandlungsstile) werden dokumentiert und, *falls da eine ausreichende Anzahl von befähigte Spezialisten und Notwendigkeit in solche Daten gibt*, analysiert

werden, um weitere potentielle Schwierigkeiten zu vermeiden und effiziente Strategie in Verhandlung zu entwickeln.

- **Zeitplan und Reihenfolge** der Lieferantenverhandlungen sollen in Projektmanagement definiert werden und sind möglichst stabil (unverändert) zu planen.
- **Ziele für die einzelne Verhandlungssitzung** werden vor der Verhandlung festgesetzt. Abhängig von der jeweiligen Besprechung wird eine Testgruppe gebildet, die Ziele und Chancen bestimmt, die Risiken begrenzt und schließlich eine ausgewogene Lösung des Problems finden kann.
- **Risiken, Folgen und Milderung Alternativen** werden geplant, wo es möglich und wertvoll ist, um Kosten zu reduzieren, Stabilität des Arbeitsprozesses zu gewährleisten und potentielle Schaden zu vermeiden.

1. Revisions due to negotiations
2. Supplier Selection decision
 - a. Kennen Sie Ihre Bedürfnisse?
 - b. Geringwertige Waren liefern. z.B.: Büromaterial, erkennen was unerlässlich ist.
 - c. verbringen Sie Zeit mit Forschung?
 - d. Potenzielle Lieferanten zur Bonitätsprüfung
 - e. Zuverlässigkeit und Schnelligkeit
 - f. Stimmen Sie den Service -Levels zu, bevor Sie beginnen
 - g. Kaufen Sie nicht bei zu vielen Lieferanten ein
 - h. Nicht einen einzigen Lieferanten haben
3. Evaluation Reports. Am besten geben Sie ihnen eine kurze Zusammenfassung, in der Sie Ihre Anforderungen, Ihre Häufigkeit und Ihr Geschäfts Niveau zusammenfassen.

Fordern Sie ein Angebot an: Es ist sinnvoll, potenzielle Lieferanten zu bitten, Ihnen einen festen Preis für beispielsweise drei Monate zu auch nach Rabatten für langfristige oder volumenstarke Verträge fragen.

Vergleichen Sie mögliche Lieferanten: Wenn Sie das Angebot erhalten haben, vergleichen Sie die potenziellen Lieferanten in Bezug auf das, was Ihnen am wichtigsten ist. Beispielsweise kann die Qualität des Produkts oder der Dienstleistung von größter Bedeutung sein, während deren Standort keine Rolle spielt. Der Preis ist wichtig, aber es sollte nicht der einzige Grund sein, warum Sie einen Lieferanten wählen. Niedrigere Preise spiegeln möglicherweise Waren und Dienstleistungen schlechterer Qualität wider, die auf lange Sicht nicht die kostengünstigste Option darstellen. Seien Sie sicher, dass Ihr Lieferant eine ausreichende Marge zum angegebenen Preis erzielen kann, damit das Unternehmen wirtschaftlich rentabel ist. Vergewissern Sie sich, dass der von Ihnen eingesetzte Lieferant die Arbeit erledigt. Einige Lieferanten können Arbeit an Unterauftragnehmer auslagern. In diesem Fall sollten Sie auch den Unterauftragnehmer untersuchen, um festzustellen, ob Sie mit dieser Vereinbarung zufrieden sind. Wo immer möglich, ist es immer eine gute Idee, einen potenziellen Lieferanten persönlich zu treffen und zu sehen, wie sein Geschäft funktioniert. Wenn Sie wissen, wie Ihr Lieferant funktioniert, können

Sie besser verstehen, wie er Ihrem Unternehmen helfen kann. Denken Sie daran, dass der Ruf Ihres Unternehmens anhand der Arbeitspraktiken Ihrer Lieferanten beurteilt werden kann. Es ist geschäftlich sinnvoll, die ethischen Dimensionen Ihrer Lieferkette zu berücksichtigen.

Verhandeln Sie die Bedingungen. Sobald Sie sich für die Lieferanten entschieden haben, mit denen Sie zusammenarbeiten möchten, können Sie mit dem Aushandeln der Geschäftsbedingungen und der Vertragsgestaltung fortfahren. In unserem Leitfaden erfahren Sie, wie Sie mit Lieferanten über den richtigen Deal verhandeln.

Select a Supplier to be awarded the Supplier agreement: Lieferantenvereinbarungen unterscheiden sich von Lieferant zu Lieferant, es gibt jedoch einige Details, die jedem Vertrag gemeinsam sind. Diese sollten umfassen:

1. Name und Adressen der beiden beteiligten Parteien.
2. Eine Beschreibung der Leistungen und Anforderungen des Lieferanten.
3. Zahlungsbedingungen und Zahlungshäufigkeit.
4. Vertraulichkeitsklauseln.
5. Erstattungen und Entschädigungsklauseln.
6. Unterschriften, Daten und ggf. Zeugen Unterschriften.

Dies ist keinesfalls eine endgültige Liste und es wird weitere Bedingungen und Bedingungen geben, die die Lieferantenvereinbarung ausmachen. Kunden und Lieferanten sollten in der Lage sein, sich hinzusetzen und den Vertrag abzuschließen, und es sollte kein Vertrag unterzeichnet werden, bis alle Bedingungen und Konditionen vereinbart sind. Wenn ein Kunde sich bezüglich der Bestimmungen und Bedingungen unsicher ist, sollte er entweder eine Erklärung verlangen oder die Vereinbarung an einen Experten weiterleiten, der in dieser Angelegenheit beraten kann. Es ist besser zu zeigen, dass die Bedingungen vor der Unterzeichnung nicht klar definiert sind, als auf das Auftreten eines Rechtsstreits zu warten und eine Haftungsklausel später plötzlich zu erheben. Es sollte niemals ein Vertrag oder eine Vereinbarung unterzeichnet werden, wenn der Kunde Zweifel hat.

SSAD SP 3.2 Establish Supplier Agreement

Ornella Pouaha

Hier geht es darum eine Lieferantenvereinbarung nach SSAD festzulegen. Die Vereinbarung sollte folgende Punkte befassen.

1. **Lieferantenvereinbarung festlegen.** Hier ist damit gemeint, alle Regelungen, die zwischen das uns als Unternehmen und die Lieferanten geschlossen werden können. In unserem Projekt geht es um die Microservice Architektur und Deployment im Cloud deswegen suchen wir potenzielle Cloud-Anbieters, die uns Arbeitspakete liefern können. Es wird mit Lieferanten vereinbart:

Leistungsbeschreibung Produkte zu bekommen. Diese Produkte sollten die Beschaffer(Unternehmen) dabei helfen eine monolithische Anwendung in Microservice und Deployment in Cloud umzubauen. Als Produktbeispiele haben wir unter anderen:

- **Eine Plattform.** Für die Umsetzung von Microservice braucht jedes Unternehmen eine Plattform. Die gelieferte Plattform sollte standardisiert sein und folgende Komponente enthalten :Hardware, Service-Orchestrierung, Datenspeicherung, Docker, Datenbanktechnologien.
- Ein anderes Produkt ist **API.** Das von Lieferanten gelieferte API sollte Leistungsstärken im Team erhöhen in dem es eine lose Kopplung und Bereitstellung von Services vereinfacht. Dies sollte auch virtuellen Maschinen verwalten, Deployment automatisieren und die Auswirkungen des neuen Release auf das restliche System im kontrollierten Rollout testen.
- **Entwicklungstools.** Diese sollte die Arbeit im Team effizienter machen und dazu beitragen, dass die Registrierung von neuen Services sofort bei einer zentralen Quelle stattfinden, so dass andere Services sie über ihre exakte Adresse sofort ansprechen können.
- **Monitorings.** Damit können die Engpässe auch in dynamischen verteilten Umgebung im Blick behalten und beim Fehler im System ein Alarm gegeben werden.
- **Zeitplan und Abnahmeprozess.** Es wird in der Liefervereinbarung die Einhaltung der Frist für Lieferung Entgegennahme von Produkten. Dies wird die beide Parteien ermöglichen ziel-und zielorientiert zu arbeiten sowie einen

genauen Arbeitsplan zu führen. Und für unser Projekt besonders eine zeitliche Bereitstellung von Microservice zu planen.

- Ein weiterer wichtiger Punkt, was die Lieferantenvereinbarung enthalten soll ist **die Ermittlung und Kommunikation alle Änderungen der Anforderungen**. Beide Parteien sind verpflichtet bevor Änderungen vorzunehmen erstmal mit dem Partner in Kontakt zu treten und ihn über die kommende Änderungen zu informieren, sodass jeder sich Zeit nehmen kann, um den Mangel zu beseitigen.
- **Dokumentation**, was der Lieferant zur Verfügung gestellt hat und was er vom Erwerber bekommen hat. Diese Dokumentation wird die beide Parteien helfen Kosten und Ausgabe, Materialbestand zu kontrollieren, ein Bilanz oder Jahresabschluss durchzuführen. Dieser Punkt muss auch in der Lieferantenvereinbarung sein. Am wichtigsten wird für unser von Vorteil sein in dem wir im Kenntnis kommen, wann wir die Lizenz verlängern und wann wir Update machen sollten.

2. **Unterzeichnung der Lieferantenvereinbarung von allen Parteien**. Mit dieser Tat erklären alle Parteien, dass sie die Anforderungen verstehen und akzeptieren. Dies wird wie folgt überprüft:

- Von der Lieferantenseite müssen alle gelieferte Produkte alle Erwartungen erfüllen, in der richtigen Menge, am richtigen Ort und richtiger Zeitpunkt geliefert sein und von dem Erwerber gegebene Preis entsprechen
- Von der Unternehmensseite müssen die Materiale entgegengenommen und bezahlt werden, keine plötzliche Stornierung und Lieferantentreue bleiben.

3. **Begrenzung aller Lieferanten, die für die Vereinbarung nicht ausgewählt werden könnten**. Aus der vielfältigen Cloud-Anbietern, die existieren; nicht alle sind dafür fähig Produkte, die unsere Anforderungen erfüllen, zu liefern. Um die Lieferanten genauer auszuwählen sollten wir wissen welche Arten von Cloud existieren, da im Unternehmen Cloud nicht gleich Cloud ist. Es gibt :

- Public Cloud-Lösungen, die Plattformen bieten
- Private Cloud-Lösungen, die der Schutz von Systemen und höher Speicherort ermöglichen, virtualisierte Infrastrukturen bieten.
- Enterprise Cloud-Lösungen, die individuelle Systemarchitektur, Autonomie und Fähigkeit zur Integration in bestehende Systemlandschaften verknüpfen.

Da bei allen Kategorien von Cloud mindestens ein Produkt für unser Projekt wichtig ist, und da nicht alle Produkte nur von einem Cloud-Anbieter geliefert werden können, werden wir nur Cloud-Anbieter auswählen, die uns mindestens ein Produkt zur Verfügung stellen kann und der Rest ignorieren.

4. **Von der Lieferantenvereinbarung nach Bedarf in der Organisation benachrichtigen:** Dies geschieht, wenn ein oder mehrere Aspekte der Vereinbarung nicht eingehalten werden. In diesem Fall wird die unterzeichnete Vereinbarung als Beweis für die Unstimmigkeit dienen

5. **Vereinbarung zwischen Lieferanten und Unternehmen nach Bedarf pflegen:** Hier geht es um alle Methode , um die Beziehung zwischen Parteien in Stand zu halten. Darunter ist die

- Empfehlung der Lieferanten an anderen Unternehmen, die das gleiche Ziel oder etwas ähnliches verfolgen.
- Eine gute Bewertung bei einer erfolgreichen Zusammenarbeit hinterlassen
- Zu den Partnern treu sein
- Regelmäßige telefonische Kontakte und Aktualisierung der Lieferanten oder Unternehmen Daten.

6. **Datensätze wiederverwenden.** Hier werden alle wichtige Daten für eine zukünftige Verwendung gespeichert, verwaltet und kontrolliert. Darunter sind :

- Name und Adresse der Lieferanten
- Bestellte Materiale (Bezeichnung, Anforderung, Datum, Preis)
- Alle beendete Projekte

Analytic Hierarchy Process

Step 1: List Alternatives

1. Amazon Web Services (<https://aws.amazon.com/de/>)
2. Microsoft Azure (<https://azure.microsoft.com/>)
3. Kahu von Lightcrest (<https://www.lightcrest.com/>)
4. Joyent von Samsung (<https://www.joyent.com/>)

Step 2: Define Criteria (hierarchy)

1. Kosten
 - Einführungskosten
 - Laufende Kosten
2. Unterstützung von third party Technologien
3. Nützlichkeit der angebotenen Technologien
4. Support, Uptime & Response time
5. Benutzerfreundlichkeit

Step 3: Weight of Criteria

Erstmal werden die einzelnen Kriterien miteinander verglichen, um die Priorität und die Wichtigkeit davon festzustellen. Die Prioritäten sind bezogen auf kleines Start Up berechnet und können sich von anderen Fällen stark unterscheiden.

Criteria Decision Matrix

	KOS	SUP	TPA	ANG	NIV
KOS	1	5	7	3	7
SUP	1/5	1	5	1	7
TPA	1/7	1/5	1	1/5	3
ANG	1/3	1	1/5	1	7
NIV	1/7	1/7	1/3	1/7	1

Consistency Ratio CR **8.0%**

Abkürzung	Priorities		Rank
KOS	Kosten	51.0%	1
SUP	Support, Uptime & Response time	20.5%	2
TPA	Unterstützung von third party Application	19.2%	3
ANG	Nützlichkeit der angebotenen Technologien	5.8%	4
NIV	Benutzerfreundlichkeit & Einstiegsniveau	3.4%	5

Step 4: Score Alternatives

Zuerst werden die Kosten betrachtet. Da das System nur von einer Zielgruppe (von Entwicklern) betrieben wird, werden die Kosten nur auf Einführungskosten und laufende Kosten aufgeteilt. Nach dem paarweisen Vergleich der beiden Kostenfaktoren wurde die folgende Priorität festgelegt: Laufende Kosten **0.875** und die Einführungskosten **0.125**. Die Einführungskosten können sich je nach Cloud-Anbieter sehr stark variieren, aber sie spielen keine führende Rolle.

Laufende Kosten

	Amazon WS	Microsoft Azure	Kaha	Joyent	Gewicht
Amazon WS	1	1/9	1/7	1/7	0.0367235
Microsoft Azure	3	1	3	5	0.558789
Kaha	7	1/3	1	3	0.264487
Joyent	7	1/5	1/3	1	0.140001

Consistency Ratio CR **9.7%**

Alle Anbieter haben das gleiche Abrechnungsmodell: es wird monatlich bezahlt und es wird dafür bezahlt, was verbraucht wurde (Pay-as-you-go). Amazon Web Services zeigt hier die höchsten Kosten. Für den Vergleich haben wir ein Fall ausgedacht und die Kosten bei allen Anbietern basierend auf genanntem Fall verglichen. Hiermit geht es um eine skalierbare Applikation in Containern. Die niedrigsten Kosten sind merkwürdigerweise bei Microsoft.

Einführungskosten

	Amazon WS	Microsoft Azure	Kaha	Joyent	Gewicht
Amazon WS	1	3	7	7	0.574072
Microsoft Azure	1/3	1	5	5	0.282081
Kaha	1/7	1/5	1	3	0.0915151
Joyent	1/7	1/5	1/3	1	0.0523314

Consistency Ratio CR **8.4%**

Ein der wichtigsten Kriterien ist die Unterstützung von externen Technologien und die Möglichkeit, third party Applikationen nachzuinstallieren. Hier zeigt das AWS den besten Kennwert, denn Amazon bietet die meist verwendeten Betriebssysteme sowohl für den alltäglichen Verbrauch als auch für serverseitige Nutzung an. Microsoft folgt Amazon, da hier die meisten Linux/Unix Distributionen und alle Microsoft Betriebssystemen unterstützt werden. Das heißt, die Nachinstallation und Instandhaltung von third party Apps ist möglich, aber bei Amazon ist es jedoch einfacher und effizienter.

Unterstützung von third party Technologien

	Amazon WS	Microsoft Azure	Kaha	Joyent	Gewicht
Amazon WS	1	3	9	9	0.602129
Microsoft Azure	1/3	1	7	5	0.278082
Kaha	1/9	1/7	1	1/3	0.0417629
Joyent	1/9	1/5	3	1	0.078026

Consistency Ratio CR **6.4%**

Alle Cloud-Lieferanten bieten etablierte Applikationen zur Verfügung an. Das sind beispielsweise Datenbankmanagementsystemen, Storage-Systemen, Dienste für machine Learning und vieles mehr. Joyent ist der Gründer von dem serverseitigen JavaScript Node.JS und bietet es an. Das wäre ein führendes Vorteil, wenn wir unsere Applikation auf JavaScript entwickeln würden, dies ist aber nicht in unseren Plänen. Amazon ist hier auch der Sieger, denn es hat die meisten Dienste, die wir gern hätten. Microsoft hat auch eine Menge an etablierten Applikationen, aber es verfügt nur über Standardpaket und paar weiterer Apps.

Nützlichkeit der angebotenen Application

	Amazon WS	Microsoft Azure	Kaha	Joyent	Gewicht
Amazon WS	1	3	9	7	0.604522
Microsoft Azure	1/3	1	3	7	0.254874
Kaha	1/9	1/3	1	3	0.0925819
Joyent	1/7	1/7	1/3	1	0.0480214

Consistency Ratio CR **7.3%**

Ein führendes Vorteil ist das Response Time. Microsoft hat hier den größten Wert wegen seiner Dezentralisierung. Microsoft hat Cloud-Server in allen Kontinenten und in meisten Ländern, das ermöglicht die schnellste Antwortzeit. Auch was sich auf Uptime bezieht, ist Microsoft der Leader, denn die Server werden sorgfältig gepflegt und haben die kleinste Anzahl an Ausfällen. Amazon und die anderen Anbieter haben dagegen ihre Cloud-Server nur in den USA und die Antwortzeit kann hier sehr hoch werden.

Support, Uptime & Response time

	Amazon WS	Microsoft Azure	Kaha	Joyent	Gewicht
Amazon WS	1	1/5	5	3	0.194
Microsoft Azure	5	1	9	9	0.675091
Kaha	1/5	1/9	1	1/3	0.0447878
Joyent	1/3	1/9	3	1	0.0861206

Das System wird von Entwicklern und IT-Beschäftigten benutzt. Bei der Einstellung von Mitarbeitern wird es vorausgesetzt, dass der/die MitarbeiterIn sich mit Cloud-Technologien auskennt und über ausreichende Kenntnisse verfügt. Aus diesem Grund ist das Einstiegsniveau irrelevant, außer den Situationen, wo die Cloud-Anbieter innovative Software anbieten, die man erst erlernen muss, bevor man damit überhaupt anfangen kann. Hier zeigt Microsoft der höchste Wert, weil Microsoft schon seit über 40 Jahren auf dem Markt ist und jeder, der mal den Rechner benutzt hat, kennt sich schon mit Microsoft-Produkten aus. Auch die neuen Produkten werden intuitiv und nach Usability-Standards entwickelt.

Benutzerfreundlichkeit & Einstiegsniveau

	Amazon WS	Microsoft Azure	Kaha	Joyent	Gewicht
Amazon WS	1	1/3	3	3	0.247607
Microsoft Azure	3	1	5	5	0.549494
Kaha	1/3	1/5	1	3	0.129255
Joyent	1/3	1/5	1/3	1	0.0736443

Consistency Ratio CR **7.3%**

Step 5: Utilities of alternatives in AHP

Als alle Kriterien priorisiert wurden, müssen sie einzelnen Kriterien miteinander verglichen werden. Zuerst sind die Unterkriterien von Kosten (laufende und Einführungskosten) zu berechnen. Die Prioritäten wurden oben berechnet und nun werden sie zusammengesetzt.

	Laufende Kosten(0.875)	Einführungskosten(0,125)	Zusammensetzung
Amazon WS	0,057358	0,574072	0,121947
Microsoft Azure	0,474611	0,282081	0,450545
Kaha	0,324955	0,091515	0,295775
Joyent	0,143076	0,052331	0,131733

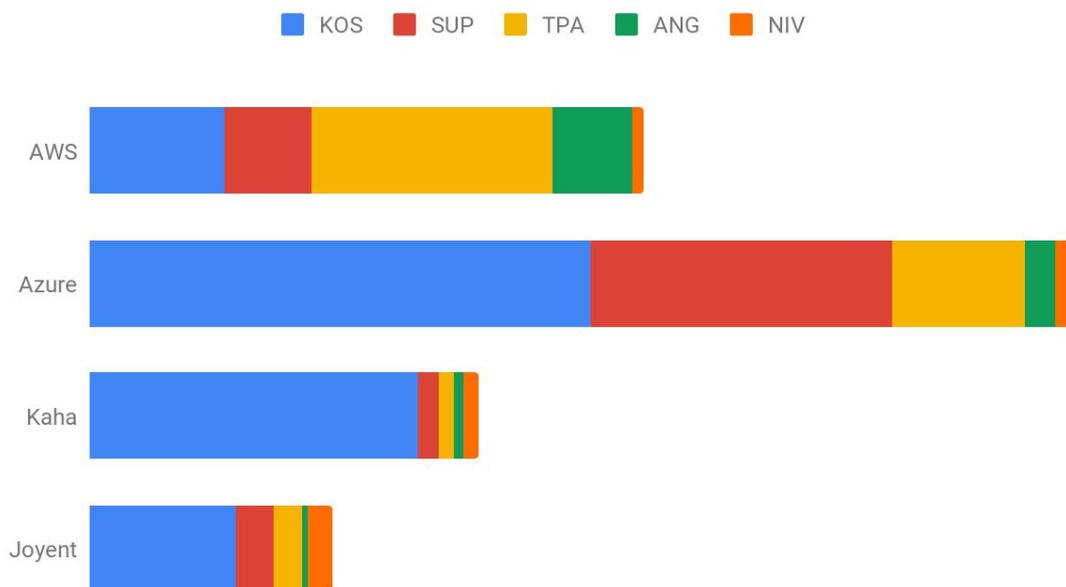
Im weiteren Schritt werden Alle Lieferanten mit entsprechenden Kriterien und mit ihren Prioritäten berechnet. Wie die untere Tabelle zeigt, haben die Kosten die meiste Priorität. Es hängt damit zusammen, dass wir ein junges Start Up sind und wir sind nicht ausreichend zahlungsfähig, um den beliebigen Anbieter auswählen zu können. Das könnte sich eventuell mit der Zeit ändern, indem wir über größeres Kapital verfügen würden. Die anderen Kriterien können sich mit der Zeit

auch stark variieren, indem die Entwicklungs- und Aufschwungsrichtung geändert wird.

	KOS 51.0%	SUP 20.5%	TPA 19.2%	ANG 5.8%	NIV 3.4%	Zusammensetzung
Amazon WS	0,12195	0,57489	0,57489	0,64189	0,15226	0,33283
Microsoft Azure	0,45054	0,31972	0,31972	0,23926	0,28503	0,38027
Kaha	0,29578	0,03962	0,03962	0,07443	0,21031	0,17804
Joyent	0,13173	0,06577	0,06577	0,04443	0,35240	0,10785

(**KOS** - Kosten, **SUP** - Unterstützung von third party Application, **TPA** - Unterstützung von third party Application, **ANG** - Nützlichkeit der angebotenen Technologien, **NIV** - Benutzerfreundlichkeit & Einstiegsniveau)

Der Sieger in der AHP-Berechnung ist Microsoft mit seinem Cloud-Dienst Azure. Es war seit Anfang an unser Favorit, aber die Berechnung wurde objektiv und mit anderen Stakeholdern durchgeführt. Als endgültige Lösung für ein Jahr wählen wir **Microsoft Azure** aus. Das untere Diagramm zeigt den Inhalt von jeweiligen Kriterien in jedem Cloud-Anbieter. Es fällt auf, das Amazon in third party Apps und in angebotenen Technologien besser als Amazon ist, dies ist aber nicht entscheidend in unserem Kontext.



Step 6: Sensitivity Analysis

Die Kosten haben die meiste Priorität (51%). Das ist eine Ausnahme, da es sich um ein kleines Start Up handelt. Wenn die Wichtigkeit der Kosten weggeschoben wird, bleibt Microsoft trotzdem auf dem ersten Platz (mit 0.222 gegen 0.192 bei Amazon).