Team Kaufen auf dem Weg zur Omnichannel Plattform – Let’s get it started

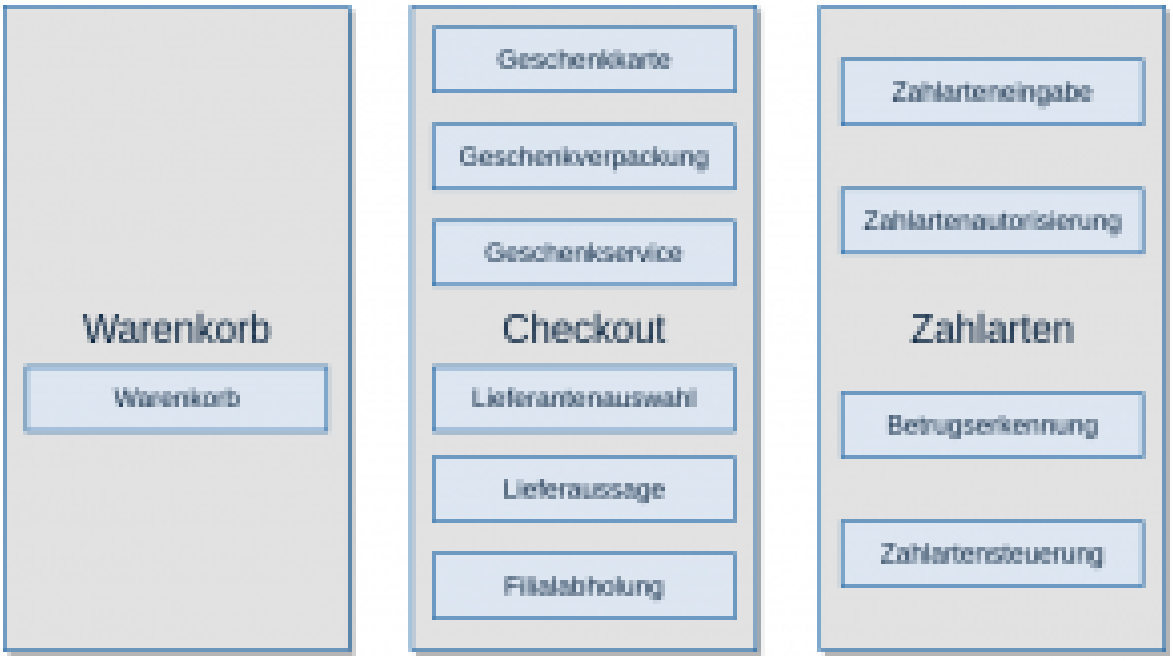
Als Thalia sind wir dabei, unsere bisherige Systemlandschaft im eCommerce, mit einem Webshop-Monolithen als Kern, auf neue Füße zu stellen. Ziel ist dabei eine zukunftsfähige **Omnichannel Plattform**. Dieser Beitrag beschreibt die Startphase des **Produktteams Kaufen**. Er erläutert, welche Themen zu unserem Produkt gehören und wie wir uns die zugehörige Systemlandschaft vorstellen.

Die Reise beginnt

Die Reise vom Team Kaufen beginnt mit einer Sammlung von Themen, die zu unserem Produkt gehören. Diese drehen sich im Wesentlichen um den Prozess des Kaufens, den Checkout.

Ausgehend von dieser Idee und den verschiedenen Themen rund um den Checkout beginnt der Prozess, die notwendigen Systeme und Services zu definieren. Nach langen Diskussionen entsteht der Plan, unsere Zuständigkeiten auf drei Systeme aufzuteilen. Die folgende Darstellung stellt dabei das Zielbild

dar.



Der **Warenkorb** soll im Kern eine Artikelliste abbilden, der nur artikelrelevante Informationen verarbeitet und nach Möglichkeit keine Kundeninformation. Dadurch erhoffen wir uns, die zugehörige Logik einfacher abbilden zu können.

Zudem ergibt sich eine klare Trennung vom **Checkout**, bei dem Kundeninformationen zum Tragen kommen. Der Kunde muss sich somit beim Übergang vom Warenkorb zum Checkout anmelden. Zum System Checkout gehören dann auch Themen wie etwa Thalia Geschenkkarten, Versandoptionen und Geschenkoptionen.

Das Thema **Zahlarten** hat zwar auch Bezug zum Checkout, ist aber so umfangreich, dass wir es als eigenes System abbilden. Dazu gehören das Speichern von Zahlarten zum Kunden, die Steuerung der verfügbaren Zahlarten im Checkout und die Autorisierung von Zahlungen.

Welche Grundsätze liegen den Systemen zugrunde?

Alle drei Säulen sollen eigenständig funktionieren und als Self-Contained System umgesetzt werden. Eine Integration von zwei Säulen, also zwei Systemen, soll

somit nicht über synchrone Aufrufe im Backend erfolgen, sondern nach Möglichkeit über das Frontend bzw. durch Datenreplikation.

Es soll **in jedem System einen führenden Service** geben, der die Darstellung im Frontend steuert und auch die Schnittstellen nach außen anbietet. Sollte ein synchroner Aufruf zwischen zwei Systemen im Backend nicht vermeidbar sein, soll die Kommunikation dann über diese Schnittstellen erfolgen. Der führende Service fungiert somit als eine Art Gateway für die eingehende Kommunikation in ein System.

In unserem Zielbild kann **ein System aus 1-n Services** bestehen. Die Integration verschiedener Services innerhalb eines Systems kann dabei sowohl über das Frontend als auch das Backend erfolgen. Schon jetzt haben wir bestimmte Services, die perspektivisch kein Frontend ausliefern werden, und somit reine Backend Services bleiben sollen.

Welche technischen Herausforderungen erwarten wir unterwegs?

Zusätzlich zu den fachlichen Themen erwarten wir zahlreiche technische Herausforderungen während der Neugestaltung der Systeme. Dazu zählen unter anderem:

- Technik Stack
- Konfigurationsmanagement
- Datenreplikation
- Session Management
- Monitoring

Unser Technik Stack setzt sich seit langem und auch weiterhin aus bewährten Frameworks und Tools zusammen, wie etwa Spring Boot, Maven, Handlebars, Tomcat, Jenkins, Git, Gerrit, Puppet, Ansible, Graylog, Grafana, InfluxDB, RabbitMQ etc.

Der Punkt **Konfigurationsmanagement** hat uns dabei sicherlich am meisten

beschäftigt. Bisher haben wir für neue Services auf Konfiguration über YAML-Dateien gesetzt. Diese sind gut strukturierbar und mit Spring automatisch in eine gut nutzbare Objektstruktur zu überführen. Je nach Umfang eines Services stoßen wir hier jedoch an Grenzen, da eine hohe Anzahl an Konfigurationen die Übersichtlichkeit erschwert. Insbesondere in einem mandantenfähigen System, in dem es viele mandantenspezifische Konfigurationen zu verwalten gibt. Eine Lösung hierfür, welche die aktuellen und in Zukunft absehbaren Anforderungen erfüllen kann, planen wir in Zukunft nach und nach umzusetzen.

Datenreplikation ist bei der Verwendung von verteilten Systemen, die wir bei Thalia für die Omnichannel Plattform umsetzen, notwendig, um die ständige synchrone Kommunikation zwischen den Systemen vermeiden zu können. Dabei sind für uns zwei Aspekte die zentralen Diskussionspunkte.

> Zum einen: **Soll jeder einzelne Service seine Replikation haben oder kann ein System für alle dazugehörigen Services die Replikation zentral halten?** Die erste Option unterstützt zwar die lose Kopplung der einzelnen Services, führt unter Umständen aber auch zu einer großen replizierten Datenmenge und zur Implementierung der Replikationslogik an vielen Stellen. Bei der zweiten Option verhält es sich genau anders herum. Zudem bleibt dabei zu klären, wie die einzelnen Services eines Systems sich die Daten teilen.
>
> Zum anderen: **Können wir überall mit Eventual Consistency leben oder brauchen wir irgendwo konsistente Daten über mehrere Service hinweg?** An vielen Stellen können wir in einem verteilten System wie unserem mit Eventual Consistency leben, die wir uns mit der Datenreplikation zwischen den Services einkaufen. Insbesondere beim Checkout kann es jedoch auch notwendig sein, auf konsistenten Daten zu arbeiten. Bspw. sollten während des Kaufprozesses die gleichen Artikelpreise zugrunde gelegt werden. Da wir **nicht mehrere Services mit Zugriff auf ein und dieselbe Datenbank** haben wollen und die **Datenkonsistenz** eine Rolle spielen kann, verfolgen wir aktuell die Idee, dass der führende Service eines Systems die Daten repliziert und den anderen Services in der API übergibt. Ob dieser favorisierte Weg tatsächlich so funktioniert wie gedacht wird der Prozess der Umsetzung der einzelnen Services zeigen müssen.

Session Management war bisher in unseren Legacyanwendungen ein wichtiges

Thema. Unsere neuen Services sollen eigentlich **stateless** werden, also auch ohne Session arbeiten können. Aus heutiger Sicht scheint das je nach System aber nicht immer ein sinnvoller Weg und die Verwendung von einer verteilten In-Memory-Lösung wie Hazelcast auch eine Option zu sein. Wie wir das umsetzen werden, werden wir sehen, sobald die Notwendigkeit dafür besteht.

Ein zentraler Aspekt bei verteilten Systemen ist das **Monitoring** der einzelnen Services. Hierfür nutzen wir sowohl proprietäre Lösungen als auch Tools wie Graylog und Grafana. Hier sind wir technisch ganz gut aufgestellt und dabei, unsere Nutzung der vorhandenen Tools kontinuierlich zu verbessern.

Fazit

Die Reise des Teams Kaufen zu einer zukunftsfähigen Omnichannel Plattform beginnt mit den beschriebenen Rahmenbedingungen und Konzepten. Unterwegs werden wir diese sicher kontinuierlich hinterfragen und bei Bedarf anpassen müssen. Jedoch fühlen wir uns durch die zahlreichen Workshops, in denen wir ausgehend von einer Sammlung der zu verantwortenden Themen und einer Idee der Zielarchitektur diese Grundlage schaffen konnten, sehr gut für den weiteren Weg vorbereitet.