

hackathon@thalia in Münster 2021



Zusammen gegen den Corona Blues und für kreative Kundenlösungen:

Nach einem Jahr Pause haben wir 2021 wieder einen Hackathon veranstaltet, diesmal rein digital. Auch nach vielen Monaten üben und gestalten von neuen Online-Formaten sind diese noch immer spannende Experimente für uns.

Wir haben dieses sehr generische Motto gewählt, um möglichst viele Kolleg:innen anzusprechen und den Raum für innovative Perspektiven weit zu öffnen. Das hat funktioniert:

5 kreative Ideen mit Mehrwert für unsere (internen) Kund:innen konnten durch ihren Pitch überzeugen und wurden an diesem Tag von spontan zusammengefundenen interdisziplinären Teams weiter gedacht.

Nicht nur ITler:innen, auch Expert:innen aus den Fachbereichen waren eingeladen sich zu beteiligen, was durch die unterschiedliche Ausrichtung der Ideen angenehm leicht gemacht wurde. So gab es sowohl eher fachlich als auch eher technisch motivierte Themen zur Auswahl.

5 kreative Ideen

1_Produktentwicklung auf Basis von Kundenfeedback

Worum ging es?

Nicht direkt die offensichtlichste Lösung übernehmen, sondern kreativ sein. Nicht direkt mit der Tür bzw. Lösung ins Haus fallen, sondern sich eingehend mit dem Problem auseinandersetzen. Das Problem verstehen, um die für den Kunden beste Lösung erarbeiten – das war das Ziel dieses Projektes.

Ein spontan zusammengefundenes Team aus verschiedenen Fachbereichen wurde dazu mit einer abgespeckten Variante der Methode [Spark Canvas](#) zunächst auf vorgefiltertes Kundenfeedback losgelassen.



Die App ist super, obwohl ich mir besonders bezüglich der Merkliste mehr Optionen wünschen würde. Ich merke mir viele Bücher mit unterschiedlichen Zwecken. Ich würde mir so etwas wie Ordner/ Pinnwände wünschen, die man benennen kann und in die man die gemerkten Bücher einordnen kann. zB. eine Liste für Geburtstagsgeschenk für einen Freund, eine für klassische Literatur, eine für einen bestimmten Autor, etc. So wäre es einfacher, Ordnung und Überblick zu behalten.

Abbildung: Eine berücksichtigte Rezension

Aus diesen wurde der Use Case „sich einen Überblick über interessante Inhalte bei Thalia verschaffen“ identifiziert und damit verbundene Challenges des Kunden erarbeitet. In einem zweiten Teil haben wir uns zunächst komplett vom Produkt entfernt, um Inspiration an anderer Stelle zu suchen, z.B. bei der App „[KptnCook](#)“. Welche Probleme löst die App besonders gut? Können wir etwas von ihr lernen? Am Ende wurde als die für uns interessante Speciality die Begrenzung der Auswahl, bei KptnCook auf drei Rezepte pro Tag, gewählt.

Ergebnisse

Use Case, Challenges, Inspiration und Speciality wurden am Ende zusammengeführt zu dem Spark „Eine Startseite mit nur drei Büchern“:

Hey Max,
welche Genre
interessieren Dich?

- ☐ Krimi
- ☐ Romane
- ☐ Fantasy

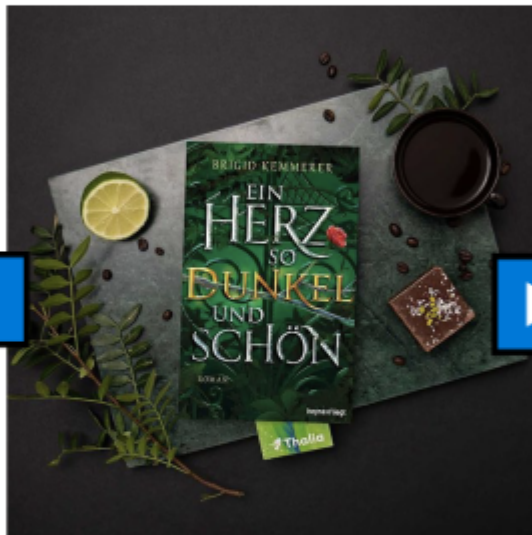
Überspringen

Danke!
Und noch eine
Frage:

- ☐ liest du?
- ☐ hörst du?

Überspringen

Hey Max,
hier deine
Empfehlungen für
deine Woche



bereits gelesen

interessant

Die Geschichte vom Herz
Lorem larem ipsum dad
njidnkwendjwebn

Hey Max,
hier deine
Empfehlungen für
deine Woche



bereits gelesen

interessant

Die Geschichte vom Herz
Lorem larem ipsum dad
njidnkwendjwebn



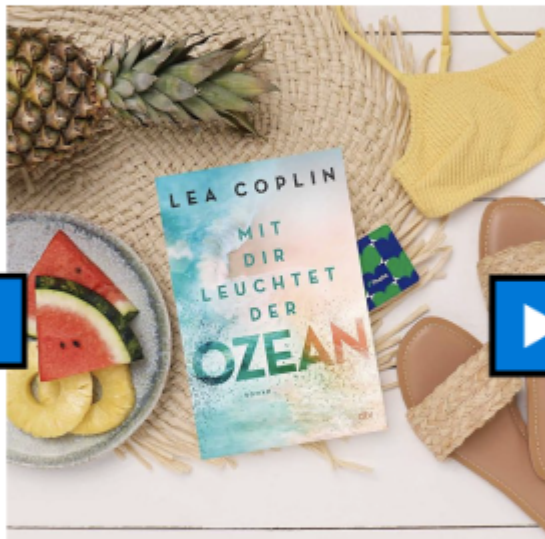
ins Bücherregal



Notizen



Hey Max,
hier deine
Empfehlungen für
deine Woche



bereits gelesen

interessant

Die Geschichte vom Herz
Lorem larem ipsum dad
njidnkwendjwebn

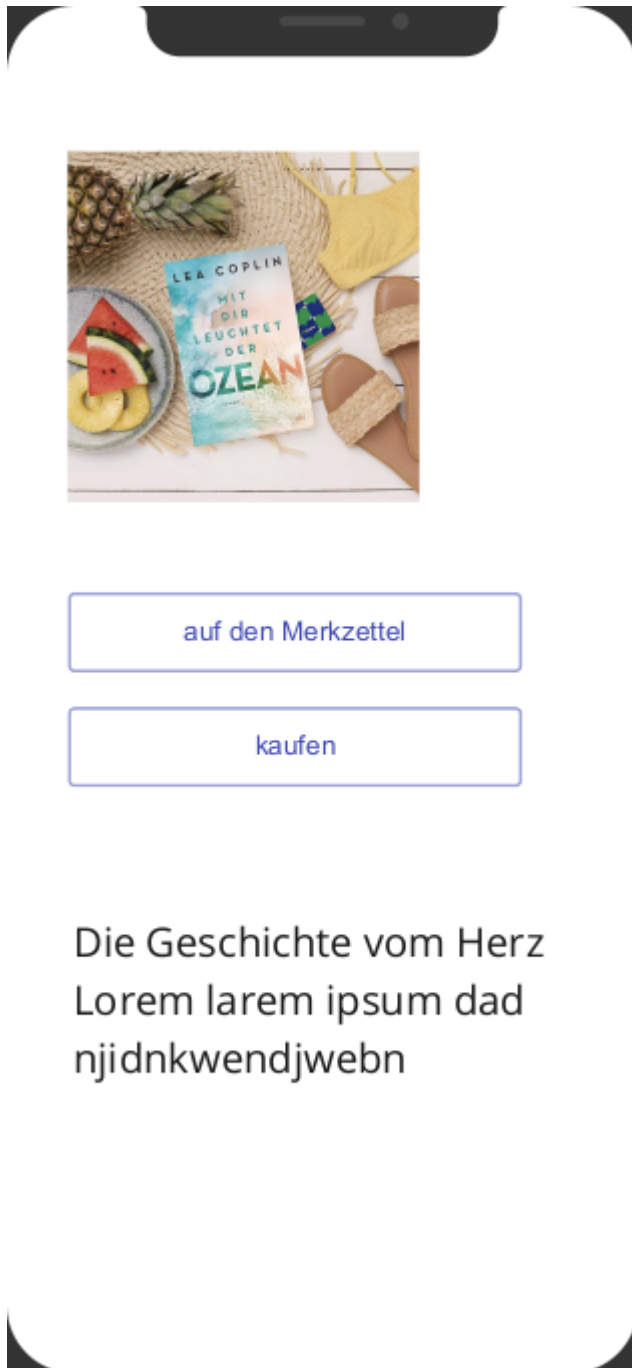


Abbildung: Prototypische Skizze des Sparks

Sich in einem Hackathon mit anderen Themen beschäftigen, oder, in diesem Fall, sich auf eine andere Art mit Themen beschäftigen hat allen viel Spaß gemacht. Es war toll zu sehen, was eine solche kreative Methode aus uns herausholen konnte. Wir sind davon überzeugt, dass dieses Feature unsere Kund:innen gut abholen kann, den Use Case und die identifizierten Challenges berücksichtigt. Er:sie bekommt regelmäßig „Futter“, das unserer Meinung nach gut passt und hat die Möglichkeit, dieses entsprechend einzusortieren.

Mal schauen, ob dieser Spark ein Feature-Feuer entzünden kann...

2_Backend for Frontend mit Apollo GraphQL

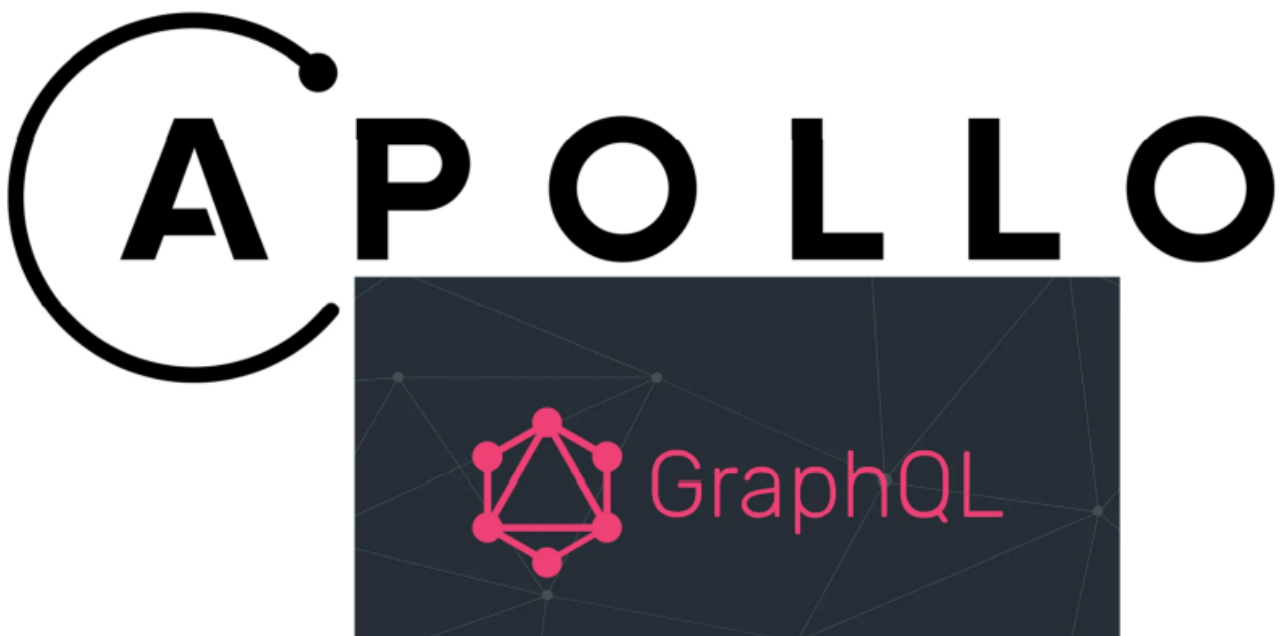
Worum ging es?

Ein [Thema](#), welches uns in den letzten Monaten sehr stark beschäftigt hat. Treiber hierbei ist unser Thalia-App-Team. In der App werden sowohl native Elemente entwickelt als auch Komponenten und Klick-Strecken von anderen Entwicklungsteams eingebunden.

Das Problem am Beispiel beschrieben:

- Team A möchte in einem API-Aufruf alle (für sie) notwendigen Daten von der API von Team B bekommen, um diese in ihrem Frontend darzustellen
- Team B möchte die API nicht erweitern, um Daten „durchzuschleusen“, die nicht zur Domäne des eigenen Teams gehören.

Backend for Frontend mit



Kann diese Problemstellung mit Hilfe von [Apollo](#) und [GraphQL](#) vereinfacht werden?

Im Rahmen des Hackathon sollte versucht werden, ein Setup aufzubauen und

anhand einer einfachen Beispielapplikation die Daten von verschiedenen Thalia-Services, per Backend for Frontend aggregiert, zu konsumieren.

Ergebnisse

Unsere Thalia-App gibt es sowohl in der [Android](#)-Version, wie auch für [iPhone und iPad](#)-Geräte. An folgendem Use Case sollte nun die Verprobung stattfinden:



Ihre persönlichen Empfehlungen



Start



Sortiment



Scanner



Merkzettel



Warenkorb



Das App-Team möchte über eine Schnittstelle nur Artikel inkl. Stammdateninformationen erhalten, für die es eine Empfehlung gibt.

Aktuell werden diese Daten bei uns von 2 unterschiedlichen Teams bereitgestellt

und – nicht überraschend – sind dementsprechend hierfür 2 Microservices verantwortlich:

Der Artikelservice stellt alle Artikelinformationen (EAN, Bilder, Texte, Preise etc.) zur Verfügung.

Der Empfehlungsservice kennt alle Artikel, für die es eine Empfehlung (Personalisierung, PaarKauf-Information, Tipps etc.) gibt.

Das Teams aus Java-Expert:innen, Android- und iOS-Spezialist:innen und Frontend-Entwickler:innen hat eine Umgebung aufgebaut, in der ein Proof of Technology auf beiden App-Versionen lauffähig war.

Per Apollo GraphQL konnten die Apps dann auf kombinierte Daten aus Empfehlungsservice und Artikelservice zugreifen. Der Empfehlungsservice hat in diesem Szenario dann nur ArticleIds (Primärschlüsselinformation) von empfohlenen Artikeln und eben nicht alle Artikeldaten ausgespielt. Der Artikelservice hat dann den Rest der Daten bereitstellt.

Das Team war am Ende des Tages sehr zufrieden: Neben den gesammelten Erfahrungen hat sich die Einschätzung gefestigt, dass hiermit das beschriebene Problem gelöst werden konnte. Eine Fortsetzung im Daily Business ist mehr als wahrscheinlich. Ein prima Ergebnis!

3_Judge a book by its cover

Worum ging es?

Meine Thalia [Lieblingsfiliale](#) in Münster → Rolltreppe in die erste Etage nehmen → dann leicht rechts halten → links hinter dem Info-Point: Mein Lieblingsbüchertisch! Kuratiert von den Kolleg:innen in der Buchhandlung – Neues, Altes, Preisgekröntes, Ungewöhnliches, Mutiges.

Wie schaffen wir es, die Exzellenz der Auswahl in den [Webshop](#), die [App](#), vielleicht sogar den [Tolino](#)-eReader zu bringen? Wir möchten den Kund:innen ein vergleichbares „Stöber-Erlebnis“ bieten, an einem Sonntag oder in einer

Lockdownphase. Oder anders formuliert: Wie können unsere Kund:innen bequem und barrierefrei diese Erfahrung an allen Berührungspunkten zu und bei Thalia nutzen?

Ergebnisse

Agenda

Vision / Ziel

**Dein digitaler Büchertisch
zum Erleben**

10:30 - 11:15 Ideen sammeln

11:15 - 12:00 Zieldefinition

12:00 - 12:45 MITTAGSPAUSE

12:45 - 14:00 Work work work

14:00 - 15:00 Build a prototype

15:00 / 15:15 "Fertig"

15:30 Ergebnispräsentation

Ein bunt gemischtes Team aus den Bereichen Business Development, UX, Softwareentwicklung, Coaching und Produktmanagement hat sich dieser Fragestellung angenommen. Möglichst schnell sollte die Ideenphase in eine Entwicklungsphase übergeben.

Mit Hilfe eines digitalen [Whiteboards](#) (Brainstorming, thematisches Clustern und Voting) und einer straffen Agenda hat die Gruppe, aufgeteilt in 2 Teams, die folgenden 2 Fragestellung bearbeitet:

- *Wie werden eigentlich die Tische zusammengestellt und wie kommen wir an die genaue Zusammenstellung der Artikel? Pro Tisch und pro Filiale?*
- *Wie kann so ein digitaler Tisch oder Regal aussehen?*

Für beide Fragestellungen haben wir Lösungen im eigenen Umfeld bei Thalia und Best Practices auf dem Markt angeschaut. Virtuelle Buchhandlungen, Raumbilder mit klickbaren Elementen & Augmented Reality: Bei der Recherche sind uns viele innovative Lösungen begegnet, die wir uns mit mehr Zeit sicher genauer angeschaut hätten. Als Inspiration für einen ersten MVP aber schon sehr hilfreich.

[Themenwelten](#), Empfehlungen unserer [Lieblingsbuchhändler:innen](#), kuratierte [Sortimentslisten](#) - schon heute hätten wir etliche Möglichkeiten die Artikelzusammenstellung über Schnittstellen bereitzustellen. Verknüpft mit der eindeutigen Filialkennzeichnung, dann in einem weiteren Schritt über die Tischkennzeichnung haben wir für uns festgestellt, das kann funktionieren. Eine gute Nachricht. Mehr konnten und wollten wir an der Stelle nicht erreichen.

Für den Büchertisch selbst hatten wir schnell die notwendigen Artikelattribute definiert. Es sind genau die, die Kund:innen in der Situation sehen: Titel, Autor:in, Klappentext, Preis, Cover-Bild und, um nun den Schwenk auf eCommerce zu schaffen: Bestandsverfügbarkeit in der Filiale (8 Bücher auf dem Stapel), ein Link zur Artikeldetailseite und natürlich die Möglichkeit zu kaufen bzw. zu merken.

Das Ergebnis mit viel UX- und Frontend-Magic kann sich sehen lassen.

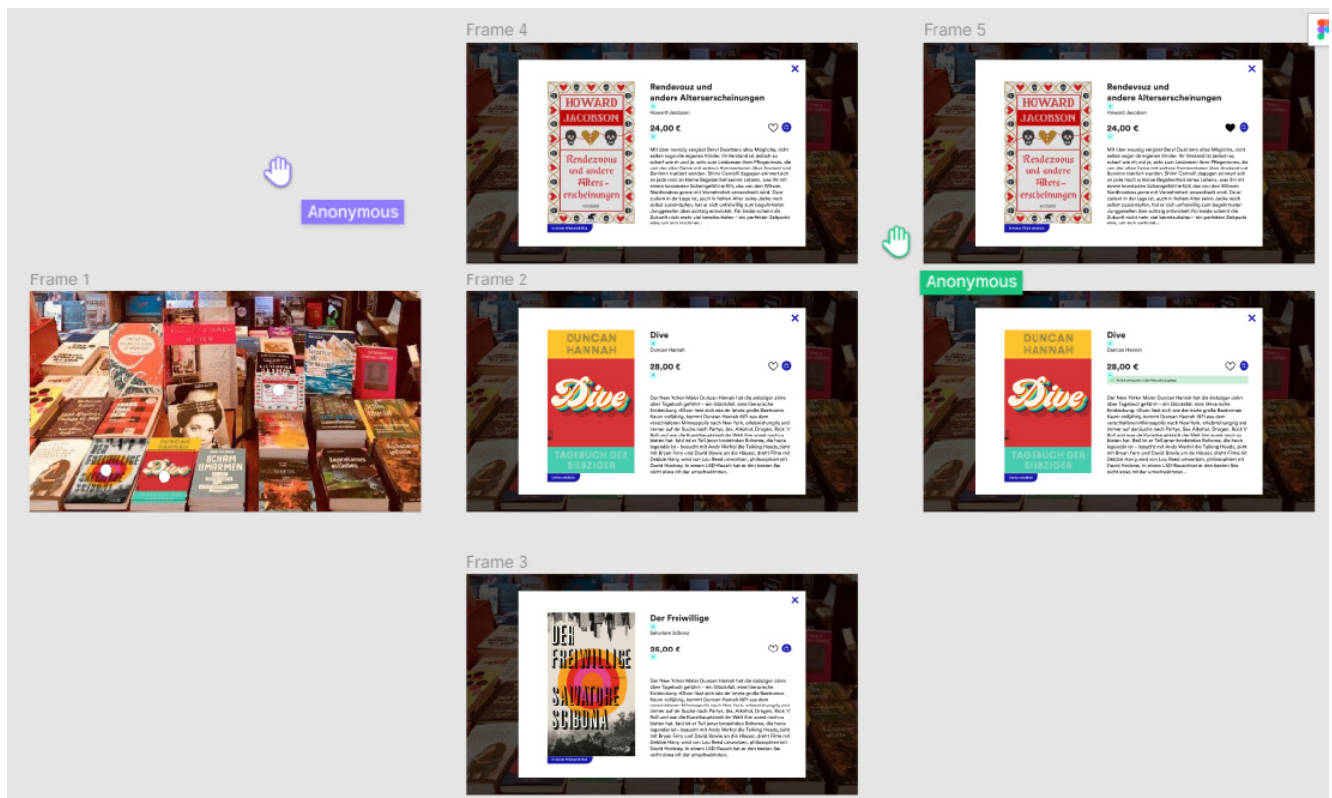


Abbildung: Ergebnispräsentation am Beispiel von 3 Artikeln des Büchertisches.

4_Frequent Pattern Mining FTW!

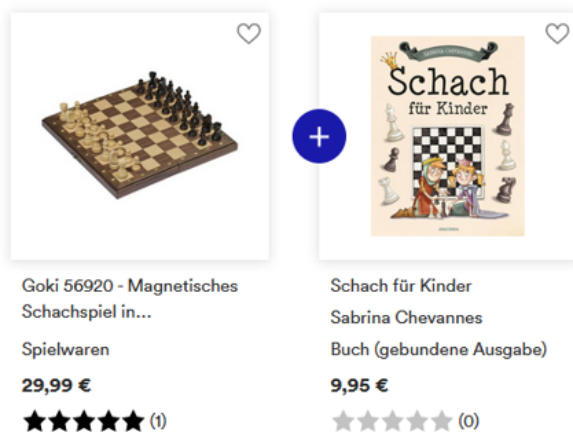
Worum ging es?

Personalisierung mit dem Ziel der perfekten Produkt-Empfehlungen sind für jeden eCommerce-Händler sowohl eine große Herausforderung wie auch eine *signifikante Möglichkeit Kundenzufriedenheit und Umsatz zu steigern.*

Dabei konzentrieren wir uns hier auf den Zusammenhang zwischen gleichzeitig gekauften Produkten. Wie finden wir heraus, dass die Ähnlichkeit zwischen einem Schachspiel und einem Lehrbuch sehr wahrscheinlich größer ist als zwischen einer Schürze und einer DVD?

Oder um es mit den Worten eines Kollegen zu beschreiben: Frequent Pattern Mining ist ein Weg für faule Leute (z.B. Informatiker:innen) diese Zusammenhänge zu finden und nutzbar zu machen, z.B. über Paarkaufempfehlungen.

Wird oft zusammen gekauft



Wird oft zusammen gekauft

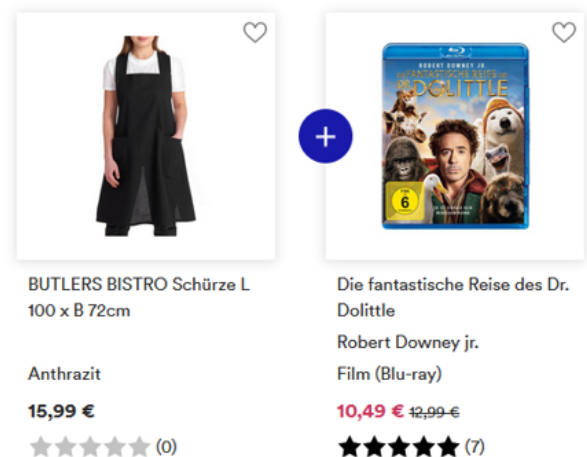


Abbildung: Paarkaufbeispiele auf Thalia.de

Ergebnisse

In einer überaus unterhaltsamen Präsentation wurden spannende Erkenntnisse geteilt:

- Datenanalyse, -bereinigung, -filterung im Vorfeld sind unabdingbar, um gute Ergebnisse zu erhalten. Schlicht 8 Millionen Transaktionen (mit

mehr als 2 Artikeln in einem Kauf) zu nehmen und dann auf ein Wunder zu hoffen, funktioniert nicht.

- Insbesondere die Filterung der richtigen Artikel erhöht signifikant die Ergebnisqualität. So blieben im Testlauf nur ~ 4.000 Artikel übrig, die in mind. 0,01% aller Transaktionen enthalten waren.
- Somit sind neben zu überprüfenden Artikelstammdaten (Stichwort kostenlose eBooks) auch übergreifende Sortimentsinformationen relevant.

Ein kleiner Teaser...

F, A, C, D, G, M, P, I

A, B, C, F, L, M, O

B, F, H, J, O

B, C, K, S, P

A, F, C, E, L, P, M, N

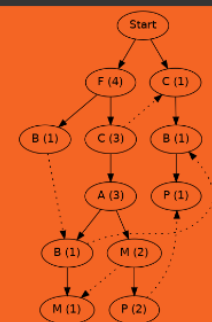
F, C, A, M, P

F, C, A, B, M

F, B

C, B, P

F, C, A, M, P



Alle Transaktionen sind fiktiv, Ähnlichkeiten mit real existierenden Transaktionen sind rein zufällig

Abbildung: Transaktionen und Ähnlichkeiten

Unterschiedliche technische Ansätze im Bereich FPGrowth wurden ausprobiert und bewertet:

- Python #1 (aus Pandas): Schlüssig war diese Implementierung nicht und funktioniert hat sie für unseren Use Case auch nicht. Daher keine Empfehlung, da „irgendetwas“ gemacht wurde, aber nicht FPGrowth.
- Python #2 (github.com/chonyy): Funktions- und lauffähig, aber nicht optimiert auf unsere Anforderung. Auch hier: Viel gelernt, aber keine 5 Sterne.
- [SparkML](#): Ein vielversprechender Ansatz, der ein wenig kompliziert zu implementieren ist. Jedoch überzeugt hier die Möglichkeit der Parallelisierung.

Wir haben gelernt, dass die Berechnung an sich mit einfachen Mitteln machbar ist, die Datenaufbereitung jedoch der Potentialbringer sein wird.

Eine Live-Demo vervollständigte die gelungene Präsentation und hinterließ

weniger fragende Gesichter, als das doch sehr technische Thema im Vorfeld vermuten ließ. Well done!

5_Spring-Cloud-Config: Konfigurationsmanagement für Microservices

Worum ging es?

Konfigurationsmanagement ist häufig eine fehleranfällige und lästige Aufgabe in der Software-Entwicklung. Insbesondere in einer Microservice-Architektur, in der gesamtheitliche Funktionen von mehr als einem Produktteam implementiert, getestet und deployed werden.

Dieses betrifft dann nicht nur mehrere Service-Instanzen, sondern eben auch unterschiedliche Services. Für Integrationstest ist es wichtig, dass auch auf den Stages vor der produktiven Umgebung diese konsistent und effizient gepflegt werden. Auf die Frage, ob dies auch ohne das Durchführen einer Deployment-Pipeline oder Datenbankänderung erfolgen kann, wollte dieses Team eine Antwort geben und ist bereits mit einer konkreten Idee gestartet: [Spring-Cloud-Config](#)

Im Rahmen des Hackathons sollten Erfahrungen mit dieser Implementierung gesammelt werden und eine Machbarkeitsstudie anhand einer konkreten Beispielanwendung erfolgen.

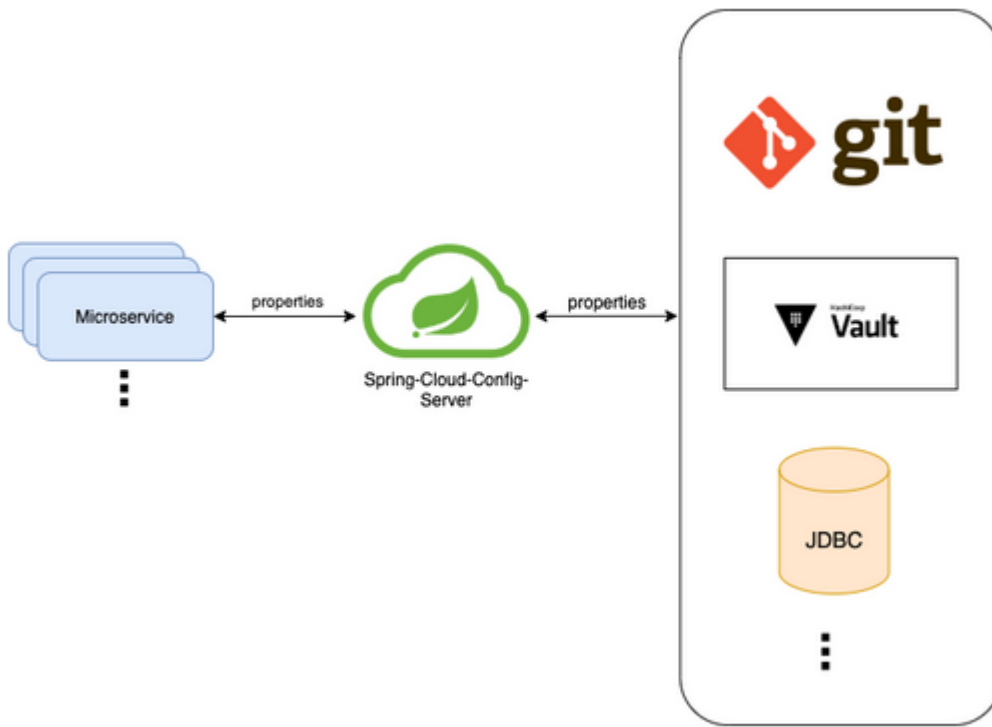


Abbildung: Infrastruktur Idee, s. auch
<https://cloud.spring.io/spring-cloud-config/reference/html/>
Ergebnisse:



Das Team hat sich eine fachliche Konfiguration als Beispiel genommen: Eine generelle Versandkostenfreiheit soll Webshop-spezifisch eingestellt werden.

Als Anmerkung: Wir betreuen auf einer einheitlichen IT-Plattform unterschiedliche Shops, neben unserer Marke Thalia.de auch weitere, wie z.B. bol.de, Thalia.at oder orellfuessli.ch.

Die Versandkosten-Information werden auf der Prüfen&Bestellen-Seite im Rahmen des Checkouts interpretiert. Technisch übernimmt hier der bestelluebersichtsservice die Orchestrierung und ist somit der Hauptakteur des Experiments. Als Messaging-System ist RabbitMQ im Einsatz, die Datenbank ist MySql und das Ganze läuft in einem Docker-Container.

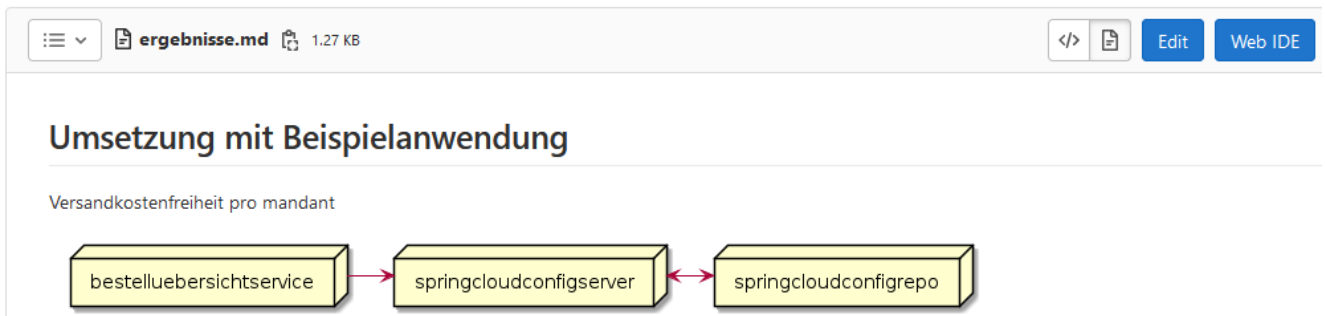


Abbildung: Konkrete Idee – Versandkostenfreiheit pro Webshop

In GitLab wurde ein entsprechendes Projekt aufgesetzt und die einzelnen Schritte dokumentiert.

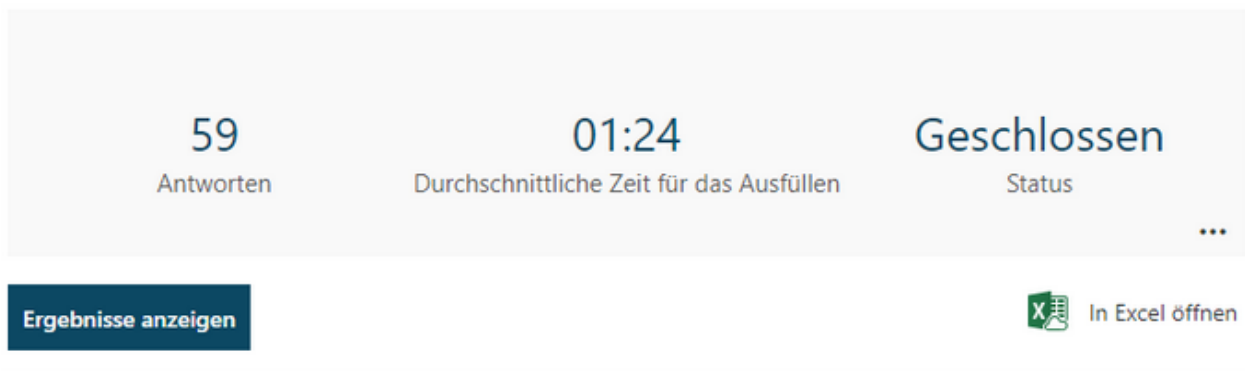
Mit Hilfe von Spring-Cloud-Config-Server konnten die im Use Case benötigten Konfigurationsänderungen zur Laufzeit und ohne Deployments durchgeführt werden. Das in der Vergangenheit bereits mit einem Spring Boot-Service gearbeitet wurde, war von Vorteil und hielt die Aufwände in Grenzen.

Im Experiment wurde auch deutlich, dass insbesondere der Anspruch, zentrale Änderungen service- und teamübergreifend zur Verfügung zu stellen, hier an seine Grenzen kommt. Nicht zwingend technisch, sondern organisatorisch. Wir halten unsere Microservice-Architektur bewusst flexibel, um schnell und effizient Veränderungen in Produktteams durchzuführen. Je mehr Shared-Services wir nutzen, desto höher sind die Kosten so einer Veränderung.

Auch wenn die Zeit etwas knapp wurde: Bewiesen wurde, wie einfach sich ein dynamischer Konfigurationsserver aufsetzen und einbinden lässt. Ob sich dies nun auch für weitere Anwendungsfälle rechnet und wir diese Lösung übergreifend einsetzen? Das Team hat mit diesem Proof of Concept eine gute Entscheidungsvorlage präsentiert.

Zusammenfassung

HACKATHON 2021



1. Welches Team hat aus deiner Sicht den Hackathon 2021 gewonnen?

[Weitere Details](#)

(Buch-)Empfehlungen einsorti...	10
Virtueller Büchertisch – Team "...	24
Backend for Frontend für die ...	10
Empfehlungen auf Basis von g...	7
"versandkostenfrei" konfigurie...	8



Das Experiment hat sich gelohnt. Ein digitaler Hackathon funktioniert. Auf die Frage, ob und wie sich die Thalia-Kolleg:innen einen Hackathon 2022 wünschen war die Antwort jedoch eindeutig: Machen, und zwar analog vor Ort. Wir geben unser Bestes[].

Ach ja, war da nicht noch etwas mit einem Preis?! Ca. 60 Interessierte haben am Ende der Präsentationen ein Voting abgegeben. Unser neuer Wandpokal hat eine erste Signatur:

Die Teilnehmenden des siegreichen Teams Bookcave (# 2_Judge a book by its cover) konnten sich über diesen und ein kleines Überraschungspaket freuen.