

# Kundenbesprechung: 31. Oktober 2018

## Anwesende Kunden

- Kerstin Öchsner (MaibornWolff GmbH) - Lastenheft Fachliche und organisatorische Ansprechpartnerin, bei MaibornWolff seit 1 Jahr, Usability Engineer
- Luisa Lucius (BMW AG) - Lastenheft: Fachliche Ansprechpartnerin. Connected Drive seit 3 Jahren, Projektleiterin
- ... Name? (MaibornWolff GmbH) - Anforderungen, Priorisierung etc, seit 3 Jahren IT Consultant, Wirtschaftswissenschaftler, dann bei IBM

## Organisatorisches

- Kommunikation (Fragen etc.): Jahrgangsverteiler und ST Projektleiter auf CC, Mails gerne an Kerstin Öchsner. Wegen Zeitramen lieber früher als später anfragen.
- 15.11. ist Termin, 7.12. Architektur-Workshop, nächster dann nach Weihnachten
- 3 Teams, gleiche Lösung, am Ende wird ein Sieger ausgewählt. Technische Rahmenbedingungen? Alle Use Cases abgedeckt? Benutzfreundlich? Passt von Look and feel zum Bestehendem? Gibt es zusätzliche Features?

## Übersicht

- Ziel: Flottenmanager sollen nicht immer die Flottenübersichtsseite besuchen müssen, sondern Nachrichten über Events etc. bekommen
- Schreiben eines Microservice für Nachrichten, Frontend
- Fuhrpark Größe: 70.000 Autos; Testdaten: 200 Autos
- Festes Regelwerk, aus denen man sich Sachen zurechtclicken kann: Wie oft man Nachrichten bekommen möchte, Von welchem Service, etc.
- Dokumentation Englisch
- Wir arbeiten nicht auf echten Daten, sondern auf Testdaten die von BMW bereitgestellt werden
- Alltag des Flottenmanagers: Computer-Person, aber sitzt nicht nur vor dem Rechner

## Abnahme

- Soll abgegeben werden als Docker-Container der auf OpenShift läuft
- Müssen erstmal nicht manuell als Docker-Container auf Heroku pushen, umstellen ist etwas für danach

## Eckdaten

- Wenn Benachrichtigungen nicht ankommen? Nicht darauf reagiert wird? - Loggen, in der Oberfläche sehen, wenn E-Mail nicht ankommt
- Flottenadmin wählt einmal aus über welchem Weg er immer kontaktiert werden will
- Einmalige Benachrichtigungen vs. periodisch neu versendet wenn sich nichts ändert
- 2 Use Cases: 1. Außerplanmäßige Events 2. Regelmäßiger Abstand (jeden Montag 10 Uhr welche Fahrzeuge laufen aus, letzter Freitag im Monat Benachrichtigung über alle Fahrzeuge die im nächsten Monat in den Service müssen)
- Pro Regel Adressaten festlegen. Pro Regel mehrere E-Mail-Adressen und Handy-Nummern angeben, für Flottenadmin reicht einmaliges Hinterlegen einer Präferenz, dieser ist eh immer dabei?
- Wir sollen von einem einzelnen Admin des Systems für jetzt ausgehen
- Eine Liste von Regeln pro User (-> Jeder Manager hat seine eigenen Regeln, die auch der Admin nicht sehen muss)
- Flottenadmin darf nicht sehen was seine Flottenmanager machen
- Es gibt Flottenadmins- und manager
- Wir müssen nicht verhindern dass von verschiedenen Usern gleiche Nachrichten an einzelne Leute versendet werden
- Mehrere Flottenmanager pro Flotte, nicht mehr als 2 realistischerweise
- Ein mehrere Flottenmanager kann mehrere Flotten managen, separate Konfiguration pro Flotte -> Beziehung Flotte <-> Manager ist N-zu-N, Ein Regelwerk bezieht sich auf Paar (Flotte, Manager)
- Für jede Regel einstellen für welche Flotte
- Prioritätsstufen von Nachrichten: Kritisch und nicht-kritisch
- Regelwerk für Nachrichten -> Es gibt am Anfang ein festes / initiales, aber welches geändert werden kann
- Sollen Nachrichten nur an bestimmte Personen oder auch zB an Fahrer gesendet werden? - Nicht an Fahrer, dieser ist unbekannt
- Zweiter Adressat Bsp Use Case: Mit Kollegen zusammen Arbeit teilen, Urlaubsvertretung usw -> Es gibt Flottenadministrator (super-admin) und Flottenmanager (für einzelne Flotten). D.h unterschiedliche Nachrichtentypen auf Regeln festlegen können, wer jeweils benachrichtigt wird, und wer z.B auf CC sein soll
- Konkrete Einstellung oder Learning von Nachrichten: Flottenmanager soll konkret und explizit einstellen, welche Benachrichtigungen er bekommen soll
- Daten die "Fahrzeugzustand" definieren? -> Erechnet sich aus 7 verschiedenen Werten: Motoröl, Brems..., ...

## Anbindung

- Microservice Architektur
- Daten kann man pullen, Test-Schnittstelle unklar ob pushen oder pullen oder was auch immer
- Dynamische Daten Updates; momentan in der Nacht immer daily updates
- In welchen Perioden sollen Daten angefordert werden? - ca 1x pro Stunde. Zum Testen evtl öfter
- Es gibt von BMW Messaging Service, über diesen können wir die Sachen verschicken haben aber noch keine Schnittstellenspezifikation
- Integration mit existierenden Services wird von uns gemocked (z.B. von BMW Messaging-Service, andere Microservices), mit Beispieldaten und Specs der Services
- Tokens nur weitergeben im Backend, Frontend ist schon alles gemacht
- Logging? -> Wichtig. Aber keine genauen Anforderungen. Wann wurde was versendet. -> Logging Service? -> Nein, wird gerade erst konzipiert
- Authentication mit OAuth+JWT
- Alle Daten kommen übers Backend nur: Service-Termin fällig wird erst aktualisiert wenn der Service im Backend angekommen ist, momentan nur über Notizen
- SMS Service Provider -> Twilio? -> Nein, durch BMW Internen Service (E-mail und SMS)

## Interface

- Regel-Editor von Outlook als Bsp anschauen; Vorschläge als optionaler Zusatz?
- Benachrichtigungen anzeigen - keine Priorität aber anzeigen + als erledigt markieren usw Nice2Have, keinerlei Priorität
- Zugriff auf das Frontend: Wir bekommen eine Kopie des Frontends
- Styleguides bekommen wir

## Nicht Funktionale Anforderungen

- Performance und Usability: an ISO-Norm orientieren, soll an Look und Feel anschließen von bestehendem Frontend -> An größe der Flotten (bis 70.000) orientieren für Performanceeinschätzung

## Unsere Fragen

- ☒ Kotlin Support? -> Ja ist von MaibornWolff verwendet und kann unterstützt werden
- ☒ GitHub Organisation? -> Nein

- ☒ Heroku/ Travis Premium? -> Nein
- ☒ Open Source Libraries -> kein Copy Left, d.h. nichts was kommerzielle Nutzung einschränkt
- ☒ Bekommen wir Folien mit Personas etc.? -> Ja
- ☒ Only production-grade libraries? (Exposed ORM!) -> Wir verwenden Hibernate.
- ☒ Benachrichtigungen erstmal nur als Mails (SMS wurde auch genannt)? -> Modular aufbauen, Service von ihnen nutzen von denen wir noch keine Schnittstellenspezifikation haben
- ☒ Was existieren für Vorstellungen wie die Nachrichten ausschaue sollen? Nur direkt Benachrichtigungen weiterleiten? -> Betreff von Email sollte wichtigste Infos beinhalten (Bsp: CBS Wert stimmt nicht, welches Fahrzeug, welche Floote) Innerhalb der E-Mail wichtigste Infos, .csv Anhang mit weiteren Infos zB Fahrzeugliste welche Fahrzeuge alle betroffen sind -> Frage: Link zur Aufbereitung im Portal? -> Link ja, warum nicht, aber .csv nicht schlecht
- ☒ Seite soll eingebunden werden können in bestehendes Frontend, aber wird erstmal extra entwickelt
- ☒ Gibt es React Component Package? -> Ja, bekommen wir wohl
- ☒ Welche Browserversionen werden unterstützt -> Firefox, Chrome, Safari und Edge, jeweils die letzten drei. Fokus auf Chrome und Firefox.
- ☒ Datenschutzerfordernisse für uns erstmal egal
- ☒ Responsiveness: Es soll auch auf dem Tablet nutzbar sein. min screen size = 768px
- ☒ Anforderungen an E-Mails -> Müssen wir uns keine Gedanken machen, es ist davon auszugehen das BMW-Service das macht (an Anforder)

## Auf später zurückgestellte Fragen

- ☐ Frameworkliste? -> Später abklären erst z.B. beim Architekturworkshop
- ☐ Other Frontend Technologies? Redux state mngmnt? TypeScript? -> Abklären bei Architekturworkshop
- ☐ “Um eine Weiterentwicklung und Anpassungen zu ermöglichen muss der Service auf einen kontinuierlichen Integrations-Prozess ausgelegt sein. Dies betrifft v.a. die Migration des Datenbank Schemas” -> Also zwingend DB Schema Migration Tool? -> Das Schema Migration Tool wird vorgegeben, Information welches gibt es später, wahrscheinlich beim
- ☐ Frontend Components als package verfügbar? -> Konnte nicht beantwortet werden

## Weitere Notizen

- Idee: Als “Wow” feature: OAuth anbindung von 3rd party apps, wie if-this-then-that für benachrichtigungen