



Praxisprojekt Anwendungssysteme – Daimler Future Van

Domenic Bosin, Alexander Dittmann, Sebastian Lienau, Marius Möck, Philipp Ratz, Antonio Schürer, Christoph Witzko

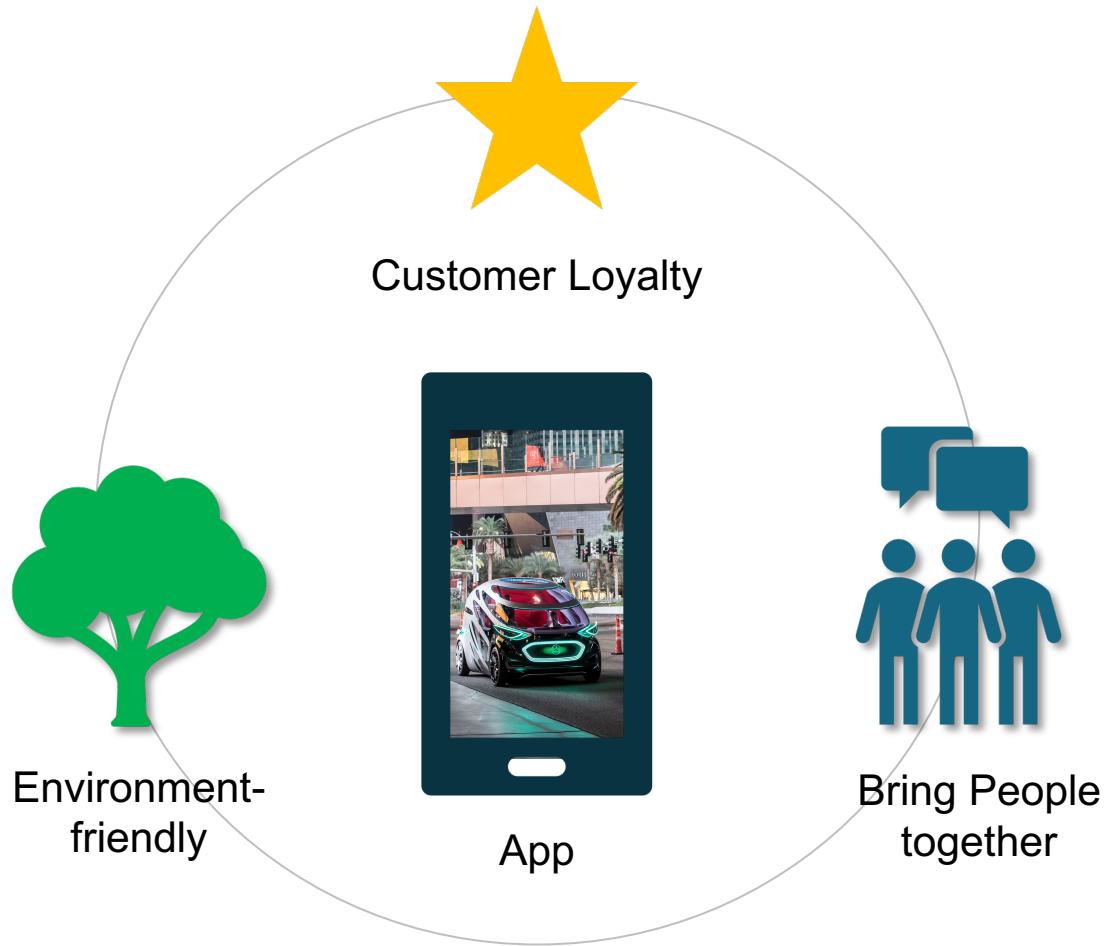
Daimler Future Van Concept



Domenic Bosin, Alexander Dittmann, Sebastian Lienau, Marius Möck, Philipp Ratz, Antonio Schürer, Christoph Witzko



Motivation





Agenda

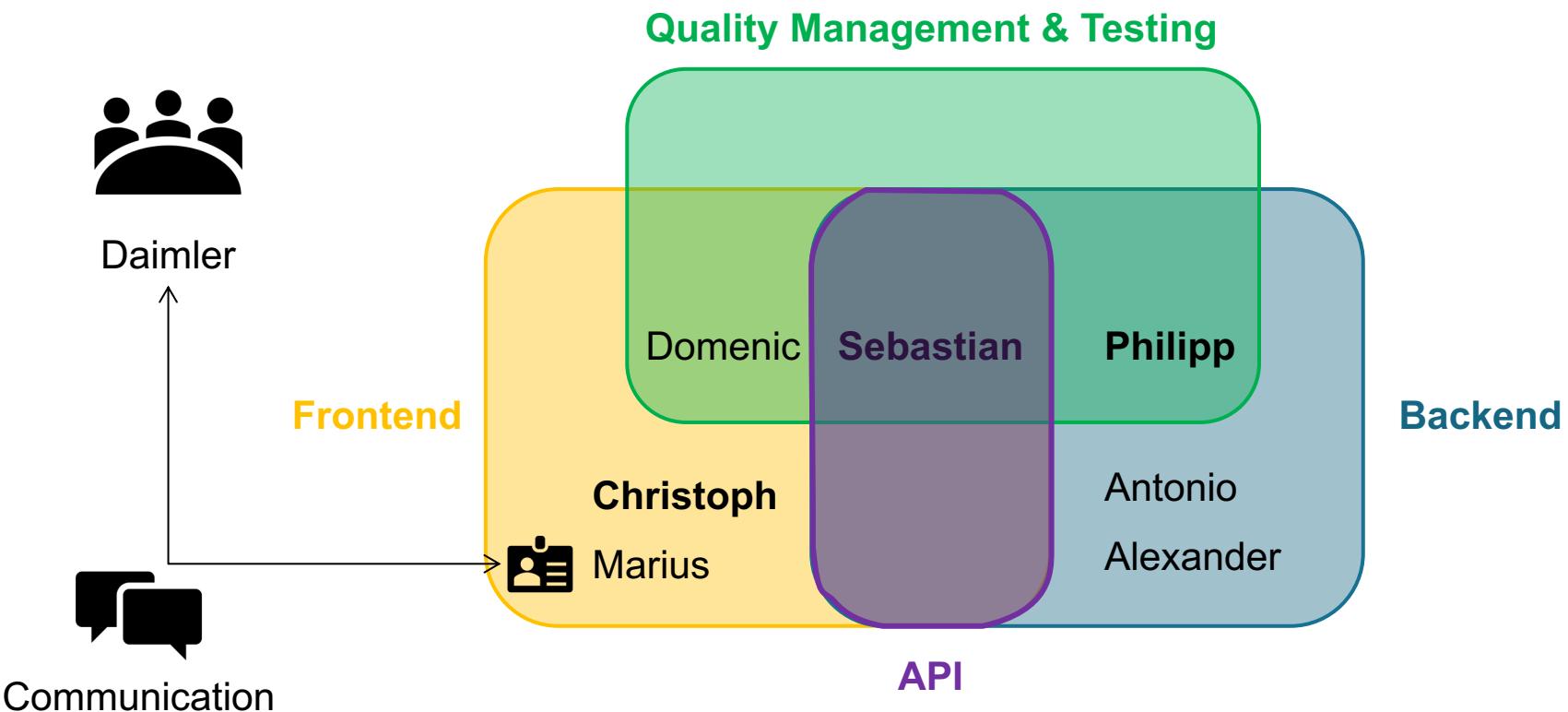
1. Organisation
2. Product Overview
3. Live Demo
4. Ride Sharing
5. Summary



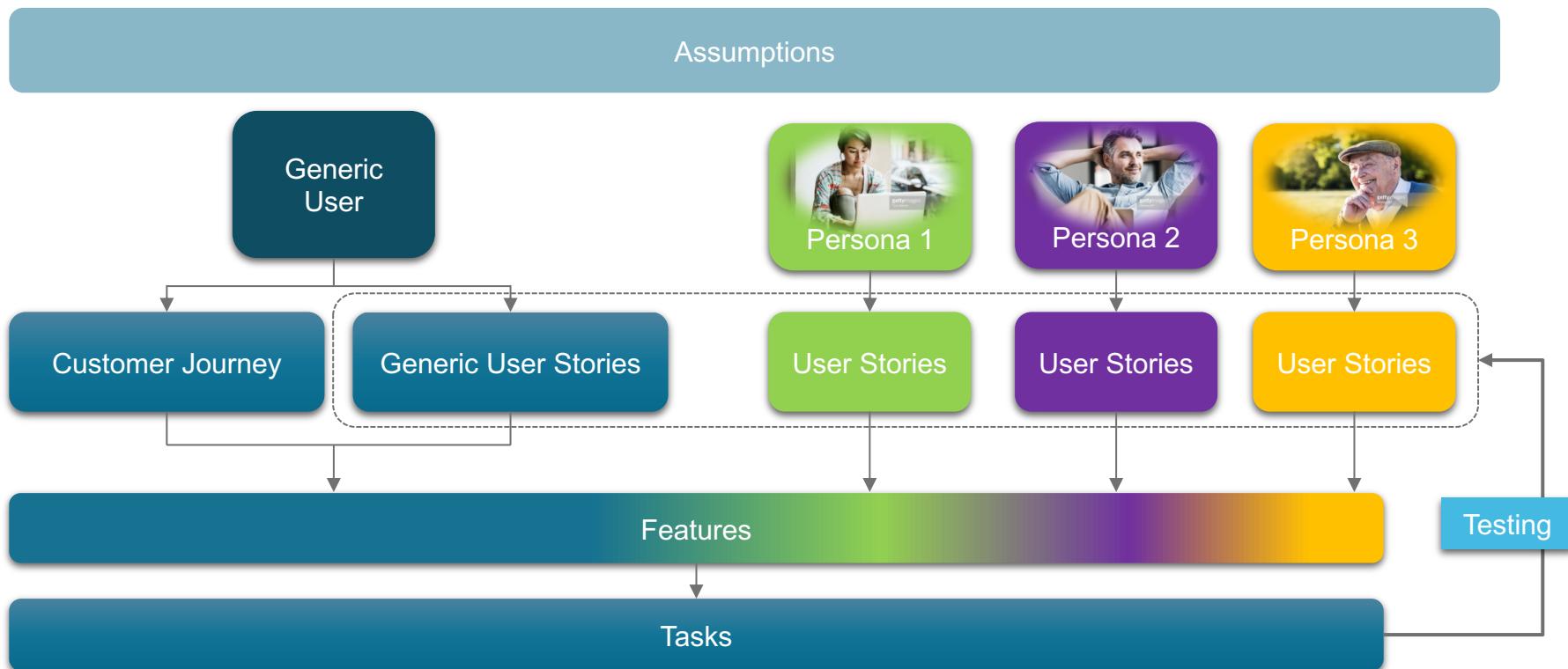
Agenda

1. Organisation
2. Product Overview
3. Live Demo
4. Ride Sharing
5. Summary

Team Organization - SCRUM

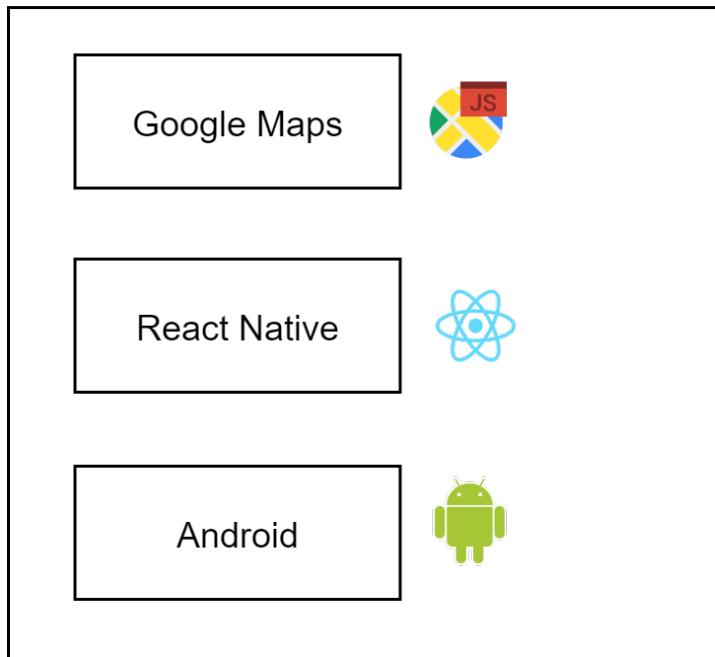


Requirement Engineering Process

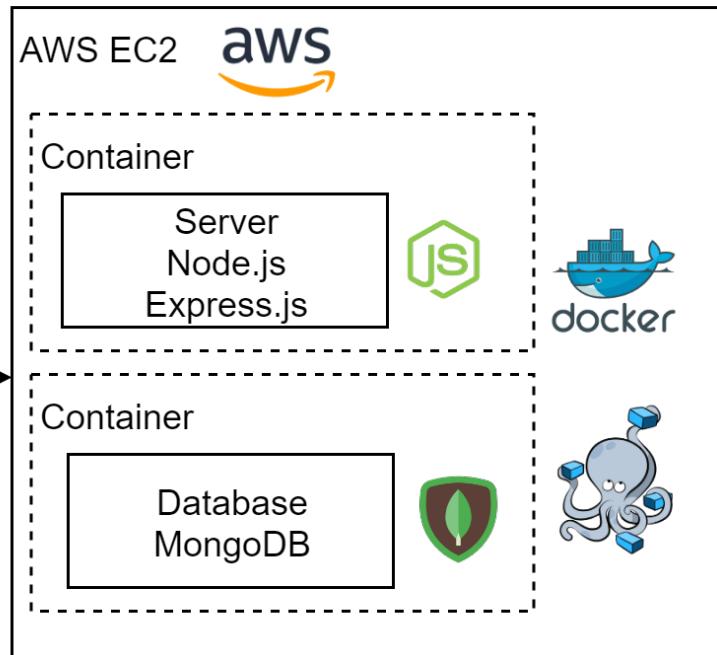


Architecture

Frontend



Backend



Quality Assurance & Testing

Quality

Constant User Story mirroring & checking

Bi-weekly Review Meetings to adjust User Stories

Weekly Call with Project Owner

Impediment List with clear Responsibilities & Dates

Testing

Functional Tests for all Processes & Scenarios

CI: Automated Pre-Commit & Pre-Deploy Tests

Van Management System incl. Simulator & Interface

Self-developed Browser Tool to view Vans & Routes



Agenda

1. Organisation
2. Product Overview
3. Live Demo
4. Ride Sharing
5. Summary

Goals

ADD VALUE

- Traffic Reduction
- Ecofriendly
- Time savings



USER-FRIENDLY

- Intuitive, easy-to-use app interface
- Simple ordering & riding process
- Flexibility (rides can be canceled until user has boarded the van)



FUN

- Ride Sharing enables social encounters
- Gamification Features (Loyalty Status & Leaderboard)



Agenda

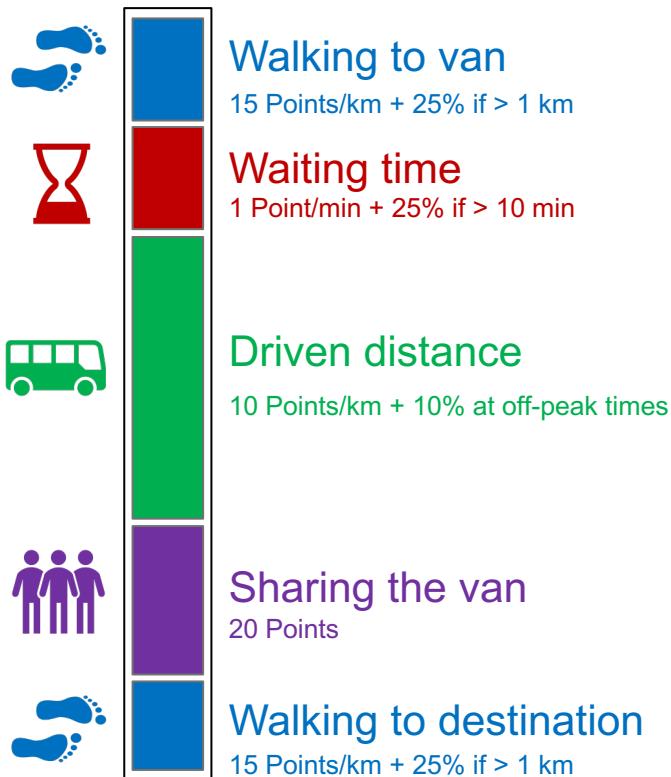
1. Organisation
2. Product Overview
3. Live Demo
4. Ride Sharing
5. Summary

Live Demo - Scenario

- Alex overslept and is late for his presentation at TU Berlin
- He is currently at Bahnhof Zoo and instead of walking 15 minutes he decides to book a Daimler Van that will take him right to the main building of the University



Loyalty Program



Achieve a loyalty status



Loyalty leaderboard



CO2 savings

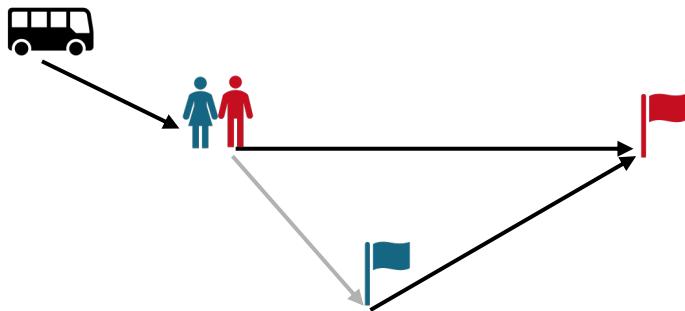
(based on EU emission limits = 0,13 kg/km)

Agenda

1. Organisation
2. Product Overview
3. Live Demo
4. Ride Sharing
5. Summary

Ride sharing with same Start

- Two passengers can share their ride and ride in the same van if the start location of both passengers coincides
- Ride sharing will only happen if none of the passengers faces a delay of more than 10 minutes compared to their direct routes



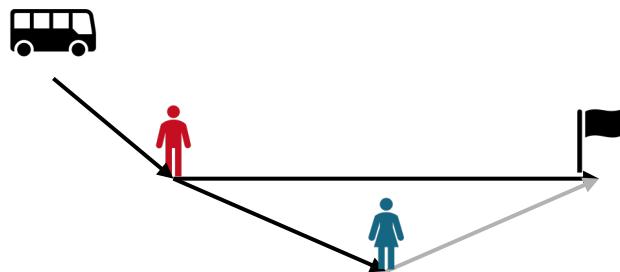
Legend

→	Route after ride sharing
→	Planned van's route



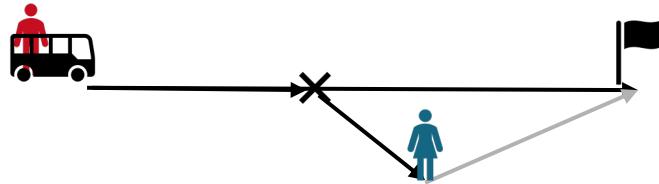
Ride sharing with same Destination

Ride sharing before Red is picked up



- Two passengers can also share their ride if the destination of both passengers coincides
- Again, ride sharing will only happen if none of the passengers faces a delay of more than 10 minutes compared to their direct routes

Ride sharing with Red already in the Van



Legend
—→ Route after ride sharing
—→ Planned van's route

Ride Sharing – Demo-Video

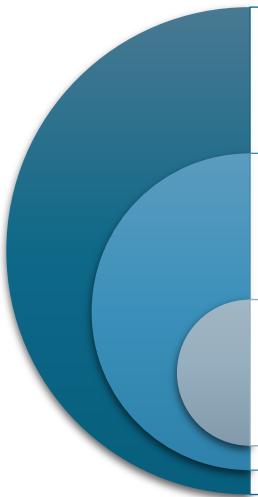


Agenda

1. Organisation
2. Product Overview
3. Live Demo
4. Ride Sharing
5. Summary

Summary

Development Focus



Loyalty Program

Ride Sharing

MVP

Future

Productivity

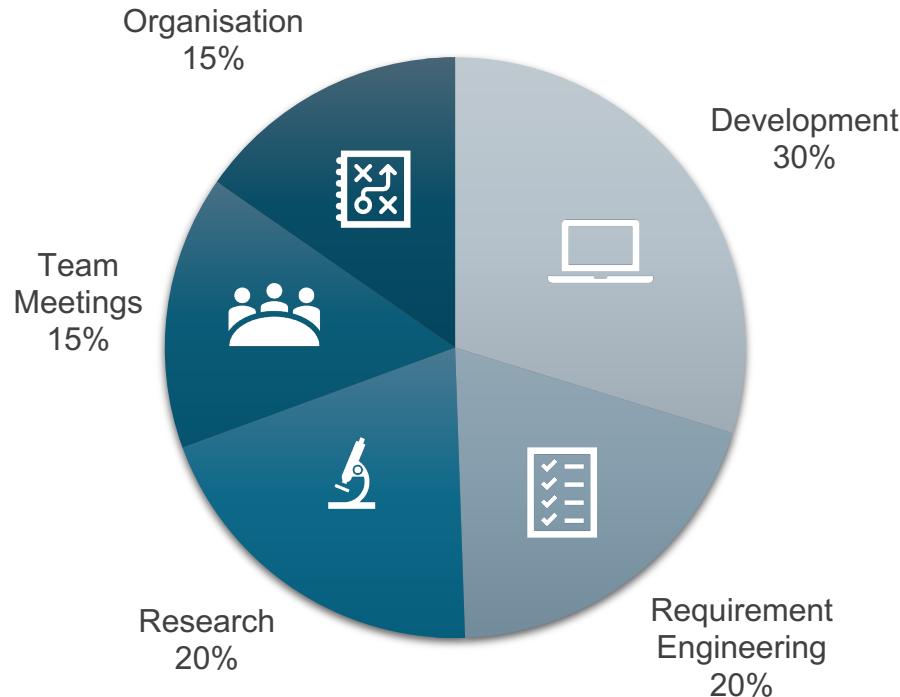
Customization

Social

Augmented
Reality

Project Statistics

2292 Total Hours



Tasks
176



Lines of Code
7320



Backup

Uncovered User Stories - Categories

Productivity

- Electronic Receipt (16)
- Calender Conjunction (19)
- Pre-Order Van (22)
- Instructive Tour (31)
- Route Suggestions (33)

Social

- Share Ride on Social Media (23)
- Social Media Conjunction (27)
- Fellow Passanger Info (28)
- Games (34)
- Game Results (35)

Customizability

- Notification Settings (20)
- Route Shortcut (21)
- Privacy Settings (29)
- Route Preferneces (30)

Uncovered User Stories - Justification

Productivity

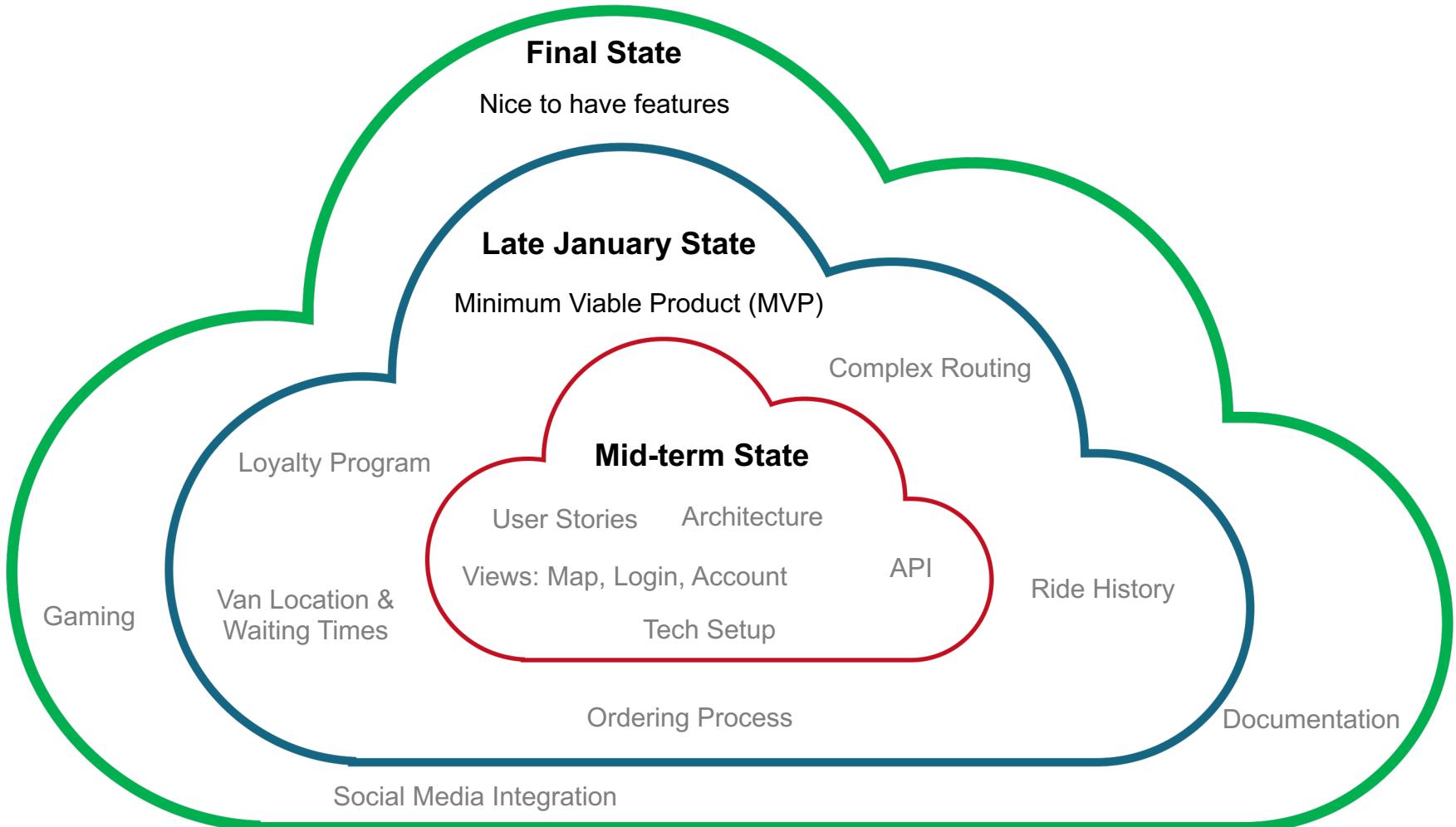
- Only useful features, when solid base is existing

Social

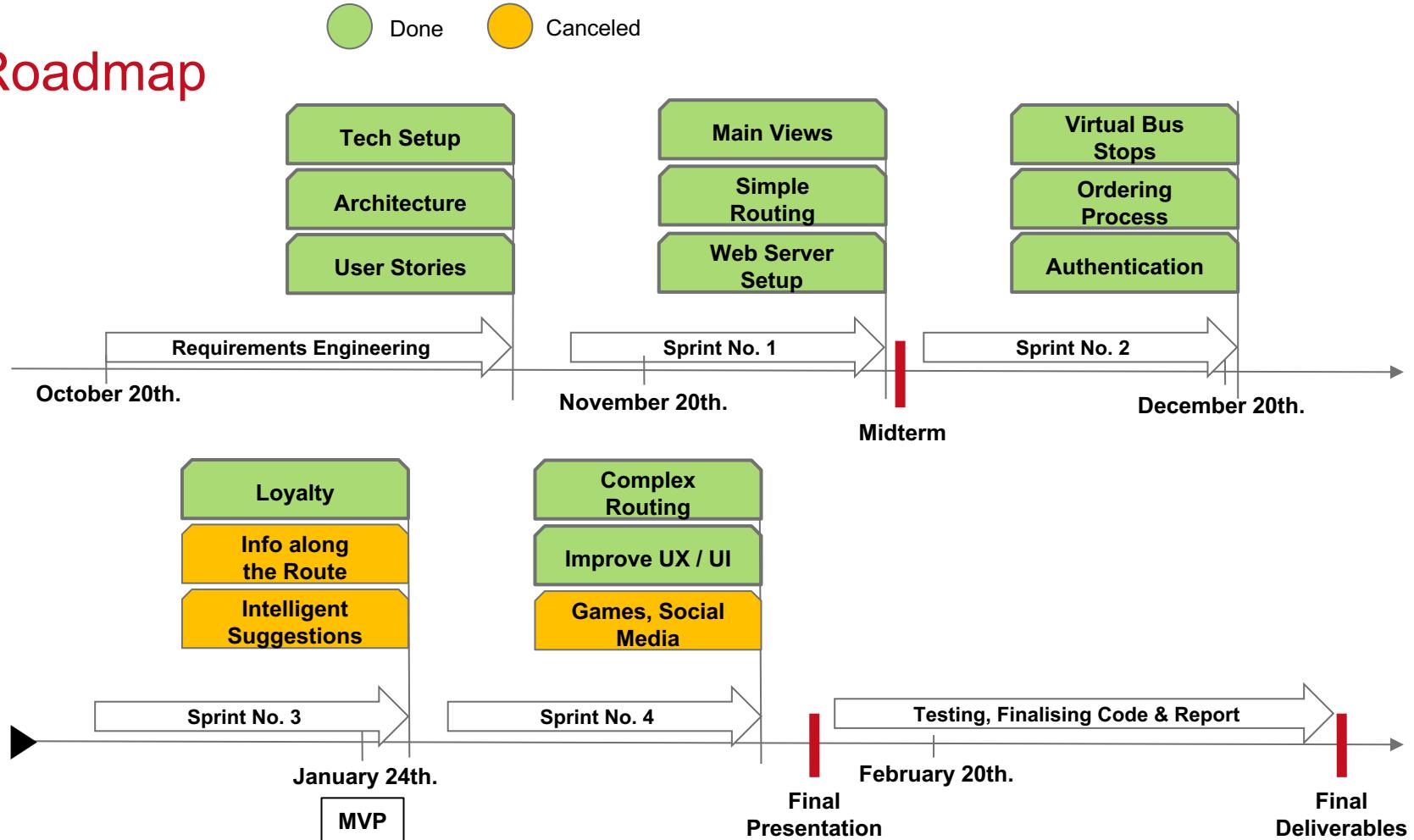
- Long-term motivation through loyalty program

Customizability

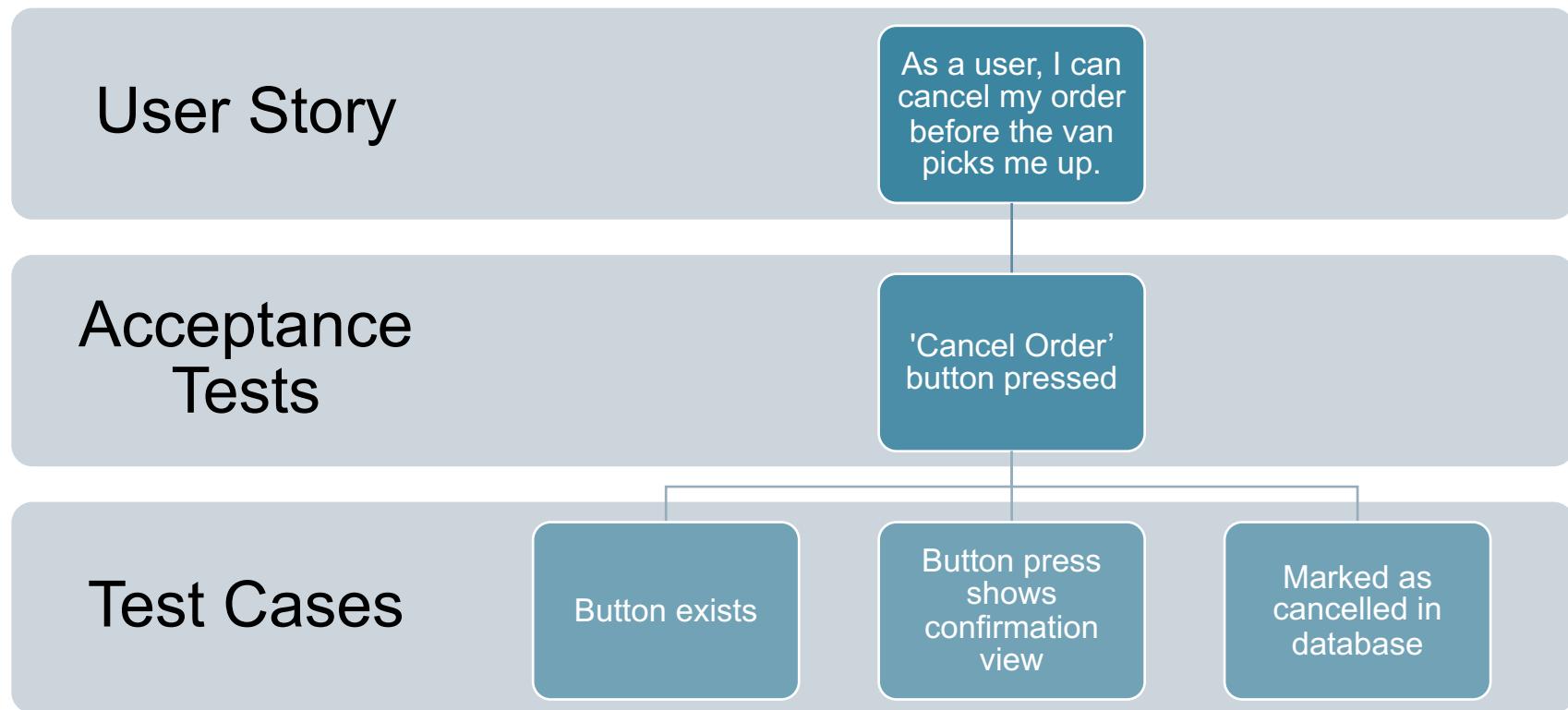
- No core focus
- Lowest priority



Roadmap



Quality Management & Testing



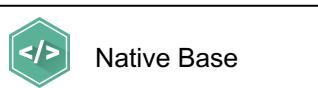
Quality Management & Testing

- Testing user stories
 - Breaking each user story in one or more acceptance tests
 - acceptance tests cover a high level test scenario and follow the happy path
 - Each acceptance test will be broken down in multiple test cases
- Example
 - User story
 - As a user, I can cancel my order before the van picks me up.
 - Acceptance Test
 - Given that I am logged in as a customer and have ordered a van, then clicking the button 'Cancel Order' takes me to a view showing that my order was successfully canceled.
 - Test cases
 - After logging in and booking a van the 'Cancel Order' Button exists
 - Pressing the 'Cancel Order' Button goes to the order cancel confirmation view
 - The order should be marked as canceled in the database

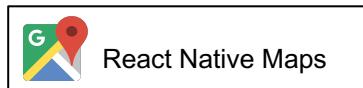
Quality Management & Testing

- Code quality
 - Every team member is encouraged to write legible code
 - We enforce a unified coding standard with a linter
 - Every change on the code base is peer reviewed
- Integration tests
 - We plan to implement automated integration tests to ensure compatibility between app and backend
- Automated Testing
 - Every committed change is automatically tested on a continuous integration (CI) platform
- Release
 - Semantic versioning to prevent incompatibility between frontend/app and backend

Technology Stack - Frontend



Native Base



React Native Maps



Redux



React Native



Android SDK



Java

Comparison - Native App Development

	React Native	Android	iOS
Programming Language	JavaScript	Java	Objective C
Supports crossplatform development	✓	✗	✗
Operating System (Development)	Mac / Windows / Linux	Mac / Windows / Linux	Mac
Market Share	-	~ 70%	~ 20%

Statcounter: Mobile OS Market Share: <http://gs.statcounter.com/os-market-share/mobile/worldwide> (accessed on 04.12.2018)

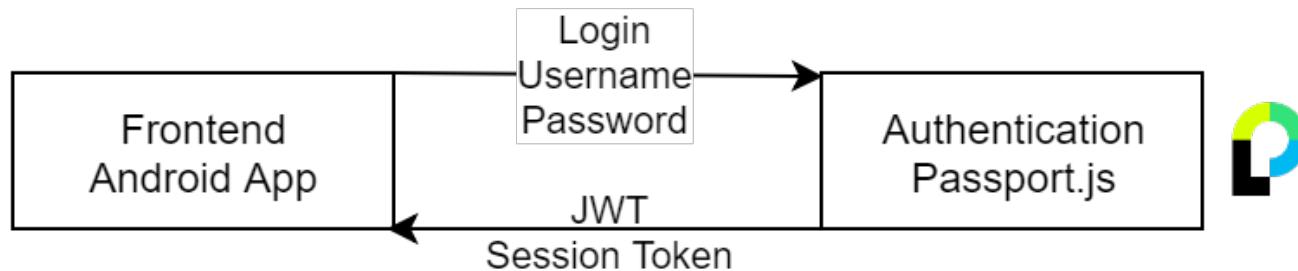
Comparison – Map Services

Feature / Maps	Google	OpenStreet	MapBox	HERE
very detailed API documentation	✓	✓	✓	
Location Tracking	✓	(-)	✗	
Tour by turn navigation	✓	✗	✓	
Public transport	✓			(-)
Gated areas	✗			
Free	(-)			✓
Customization Options	✓	✓	✓	✓

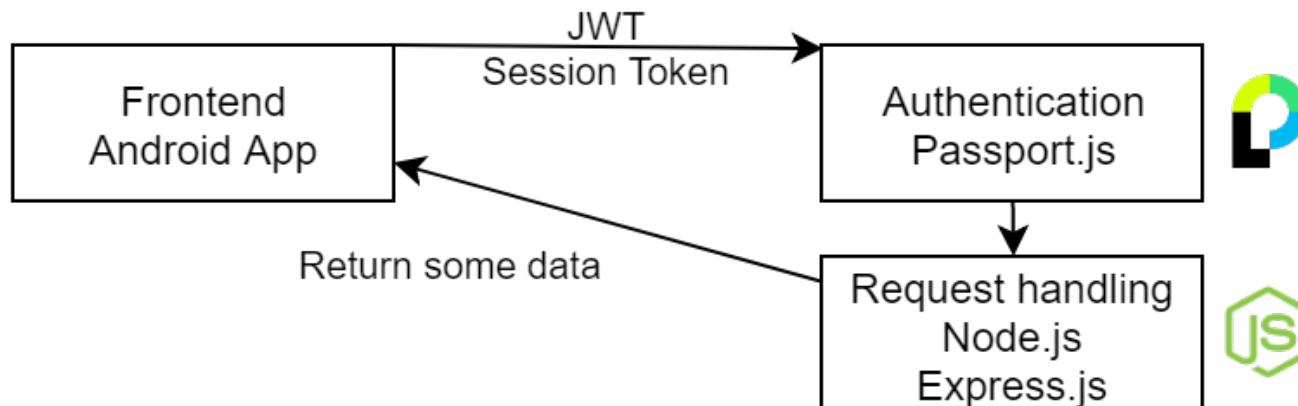
(-) Limited

Authentication

1. Login - request a JWT Session Token



2. Requests authenticated with JWT Session Token



Git Workflow



Organisation Tools



Slack main communication tool



Trello project management



GitHub code management



OneDrive meeting documents and presentations



Overleaf LaTeX project report & documentation



Clockify time tracking

Management System

Management System

A (mocked) system, that provides us with all resources we need for our purposes that usually real vans would provide us with.

Simulator	Handler	Assignment Alg.
<ul style="list-style-type: none">• Provides live van locations• Simulates van rides• Resets waiting vans	<ul style="list-style-type: none">• Responsible for maintaining the van• E.g.: (Re)calculate routes, let passengers exit or enter the van, start or end rides	<ul style="list-style-type: none">• Calculates an optimal route for a request• Picks optimal van• Implements the pooling algorithm

Project experience

- React Native
- Working with AWS / Azure
- Next time using a relational database with json support rather than a NoSQL database
- Great teamwork (Everyone had a task and mastered it)
- Learn how we work with and integrate map services

The app's language

1. **Journey:** All of the user's impressions & interactions when using the service to travel from A to B
2. **Route:** A potential geographic track from the user's start to their destination
3. **Virtual Bus Stop (VBS):** A set geo-point where users can enter or exit the vehicle
4. **Ride:** The part of the journey between two VBSs that involves interaction with the vehicle
5. **Van:** A self-driving, 8-person minivan

