Intuitive Web App For Robotics Manufacturer

How crafting a web app with an easy-to-grasp interface and flexible architecture empowered a manufacturer of warehouse driverless forklift systems to unlock new value-creation opportunities and **reach SMB customers.**

Industry:
Logistics, Manufacturing

ENTERPRISE AUTOMATION
UX AND DESIGN
WEB DEVELOPMENT
MVP DEVELOPMENT
SOFTWARE PRODUCT DEVELOPMENT

Business challenge

Under-automated or under-digitized intralogistics inevitably affects a company's profits. However, sometimes the bottlenecks aren't that obvious, and even organizations with warehouse-specific software and robotics in place struggle to achieve the much-coveted outcomes of intelligent automation.

The issue may lie in having **overcomplicated software designed for tech-savvy users** that holds down the company's capabilities in efficient control and optimization of the goods flow within a manufacturing facility, warehouse, or distribution center.

Our client, **Bleichert**, an automation machinery manufacturer, wanted to **address two pain points of their customers** seeking warehouse robotics:

- **'By engineers, for engineers' approach** in designing the software
- **Lack of targeted solutions for small and midsize businesses (SMB)** in the market overloaded with enterprise-grade products

To reach SMB customers and provide them with user-friendly software for warehouse robotics, they decided to broaden their offering with an **innovative model of their driverless forklift systems** that move loads autonomously by following a pre-configured circuit. Bleichert partnered with NODE Robotics and integrated their software modules into the experimental forklift. However, they needed a reliable tech ally to create a **web app with an easy-to-grasp interface for warehouse operators and managers.**

For example, **navigating the driverless forklift systems in small spaces is quite a challenge**, as such robots are usually designed for large warehouses. The equipment can neither go near fire exits nor stop near them by default. However, following these settings is tricky in small spaces with little room for maneuvering. Hence, when the robot has no other way than to make a path near the fire exit, it stops before reaching it. Such cases **require manual human input** to restart the system with an updated route.

Instinctools filled the bill perfectly. Our dedicated team created an **MVP with interactive features for the demo presentation** at LogiMat – the International trade show for intralogistics solutions – in **just seven weeks.**

Solution

When reaching out to *instinctools, the client specified their requirements for the future application.

Architecturally significant requirements (ASRs) included:

- **Accessibility** via common browsers, such as Google Chrome, Safari, Microsoft Edge, etc. with an identical look and feel
- **Flexible architecture** to support and manage multiple robots in the future
- **Real-time** updates of a robot mission

Non-functional requirements (NFRs) encompassed:

- **Quick and easy navigation** with the rule of thumb "max three levels to reach the goal"
- **Less than 0.4 sec** response to the user input
- **Color-coded** interface to simplify operators' work and avoid any possible costly mistakes

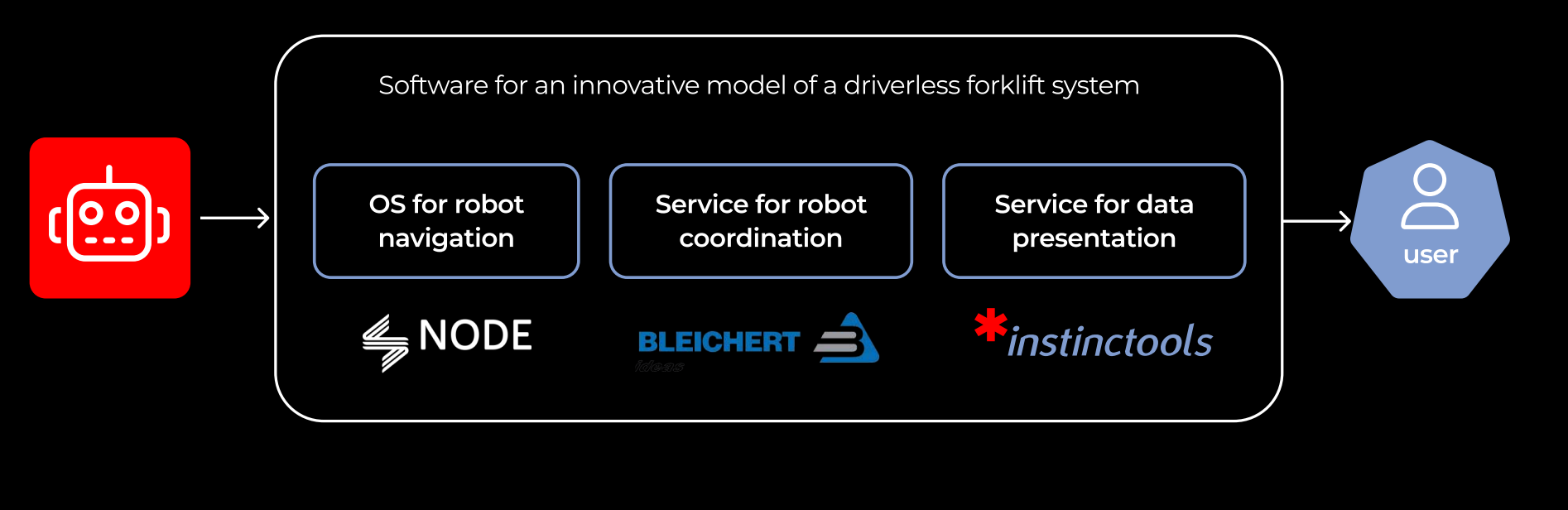
Detailed documentation empowered us to jump into development right away. Our software architects started with establishing a robust tech stack.

01 Deciding on a tech stack

We zeroed in on the client's requirements to shape an **easy-to-maintain tech stack** for the web app development.

- **Seamless compatibility with NODE OS.** Bleichert already installed the autonomy OS for mobile robots on the experimental forklift system to handle map management and interact with the robot's systems and sensors. Since the operational system with related plugins uses **Python**, we suggested sticking to this programming language and **FastAPI framework for server development** for the sake of tech stack consistency.
- **Fast user interface response times.** Besides ensuring the mandatory ease of use for the warehouse operators, we had to take care of the app's interface speed. Our frontend developers suggested going with a single-page application (SPA) using **React JS for user interface development**, as the framework's component-based architecture and virtual DOM allow crafting UI with lightning-fast render times.

Here's the **high-level scheme** of the future web app.


Software for an innovative model of a driverless forklift system
OS for robot navigation
Service for robot coordination
Service for data presentation
NODE
BLEICHERT
instinctools
user

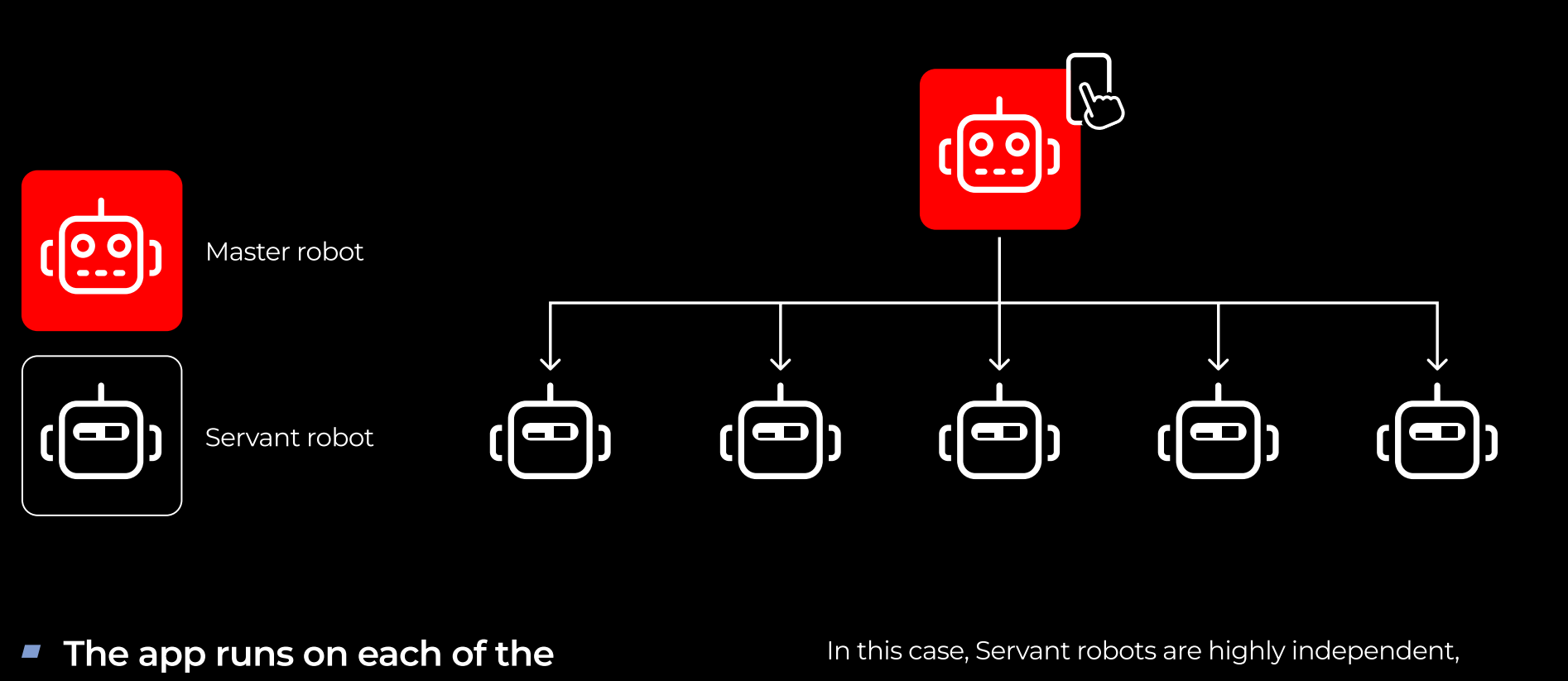
02 Taking into account long-term usage scenarios from the onset

The MVP was tailored to **user interaction with one robot (Master Robot) within a pre-configured environment.** However, outlining the long-term project vision at the early stages is a best practice our team follows to eliminate head-scratchers in the future and enable a hitch-free development process.

Therefore, in our conversations with the client, we discussed **two strategies their customers can choose from in the future** when the app will operate multiple robots (Master Robots and Servant Robots).

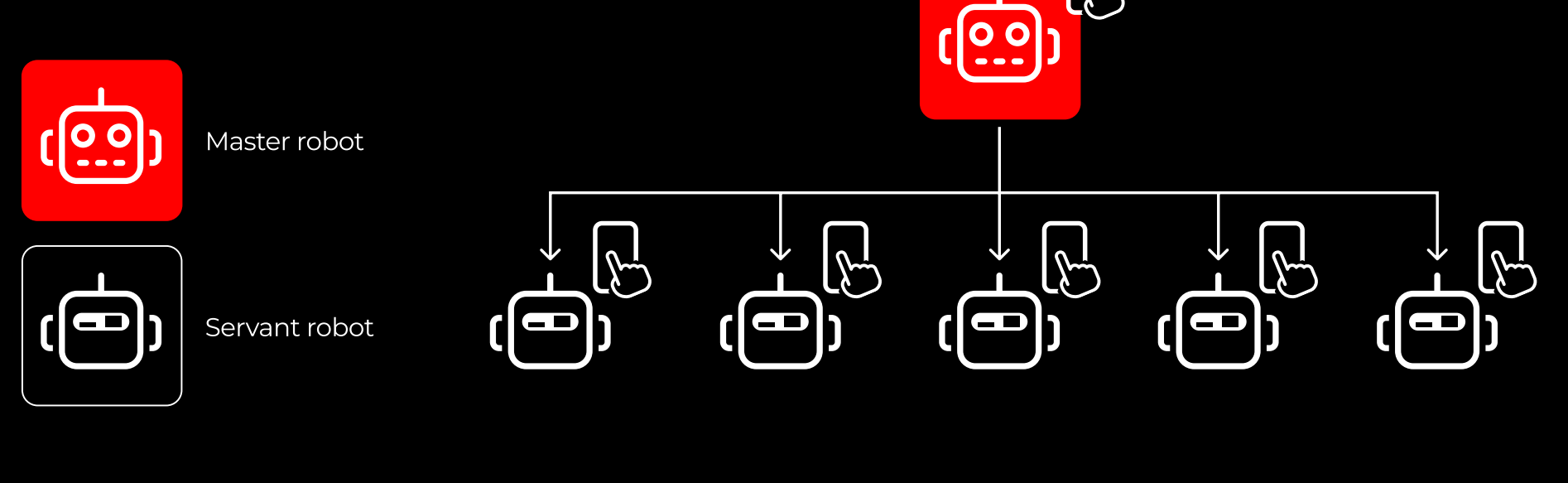
- **Seamless compatibility with NODE OS**

Such an approach provides end users with a single point of control for all forklift systems within the warehouse. However, the main drawback of this method is its high dependence on the Master robot. In case of its malfunctions, Servant robots won't be able to perform their tasks.



- **The app runs on each of the Servant robots with one Master robot for covering managing tasks**

In this case, Servant robots are highly independent, eliminating the risk of a work stoppage due to the issues with the Master robot, but high-level management becomes trickier.



The bone of the **architecture at the heart of the web app is its flexibility**. It can be tailored to either of these two options, depending on the client's customers' feedback on the innovative forklift model.

03 Engineering and designing a user-friendly web app

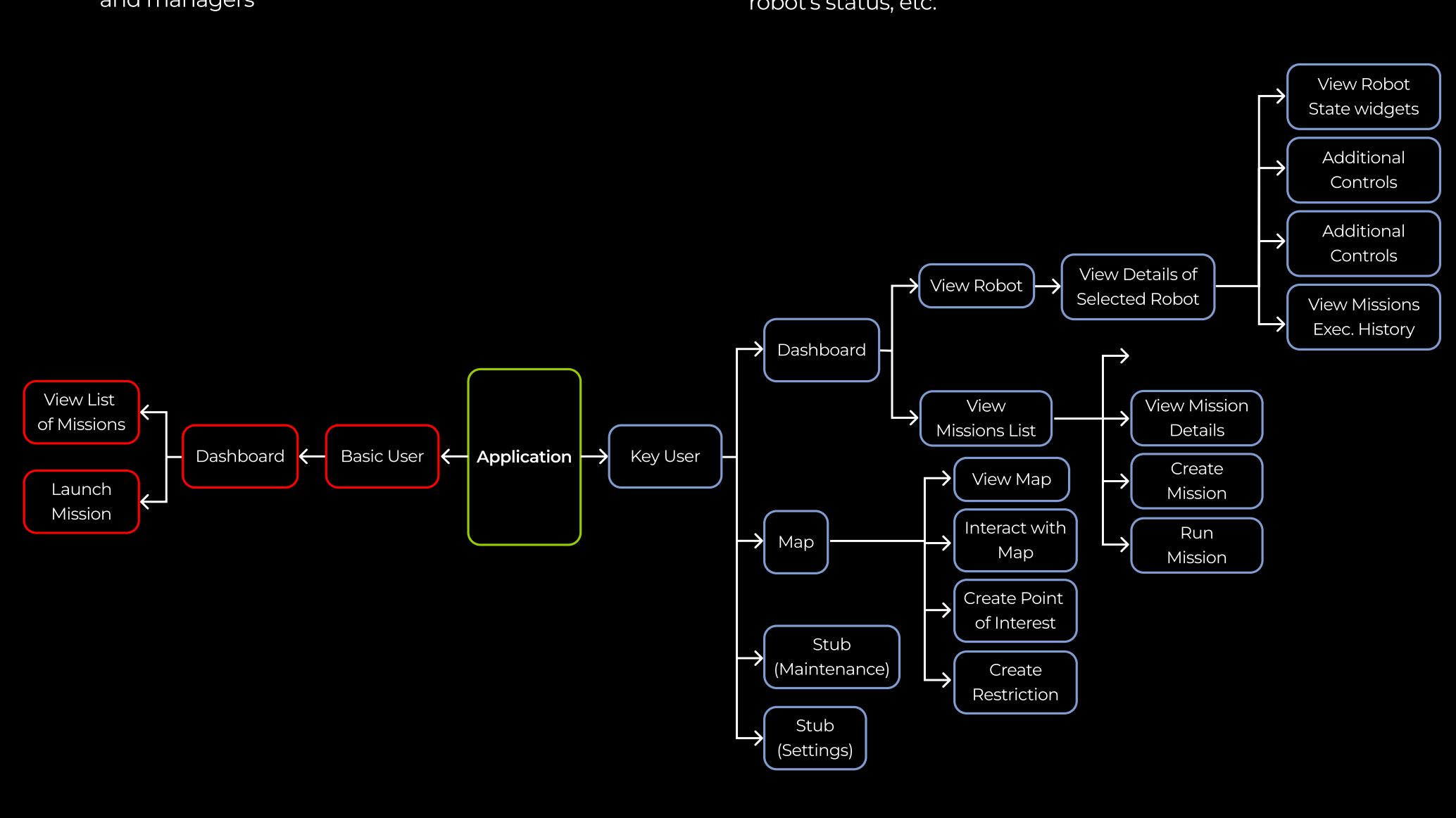
Moving forward according to our Delivery Framework, we proceeded with iterative development, covering user roles, robot management, and more.

User roles within the app

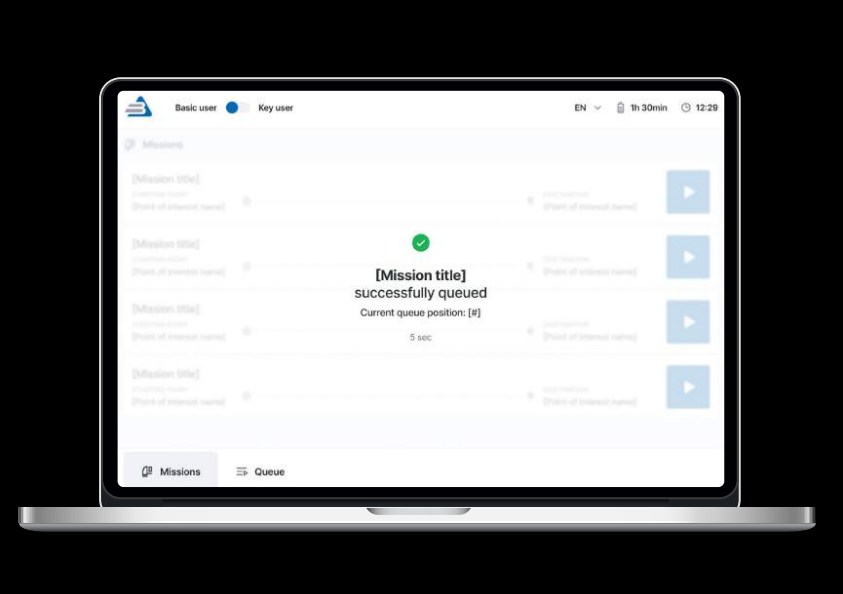
The system has authorization and authentication mechanisms, but they aren't mandatory, and non-authorized users can access the web app. Thus, we provided two modes within the system:

- **Basic user**, designed for the rank-and-file warehouse operators with no authorization required
- **Key user**, accessible only by authorized employees, for high-level warehouse operators and managers

However, the capabilities of authorized and unauthorized users greatly vary. As you can see on a **high-level app vision**, Basic users can only view the list of existing robot missions and launch them, while Key users are allowed to create new routes and missions, add restrictions, view the history, check the robot's status, etc.



Now, let's dive deeper into the robot-related features.



Robot mission management

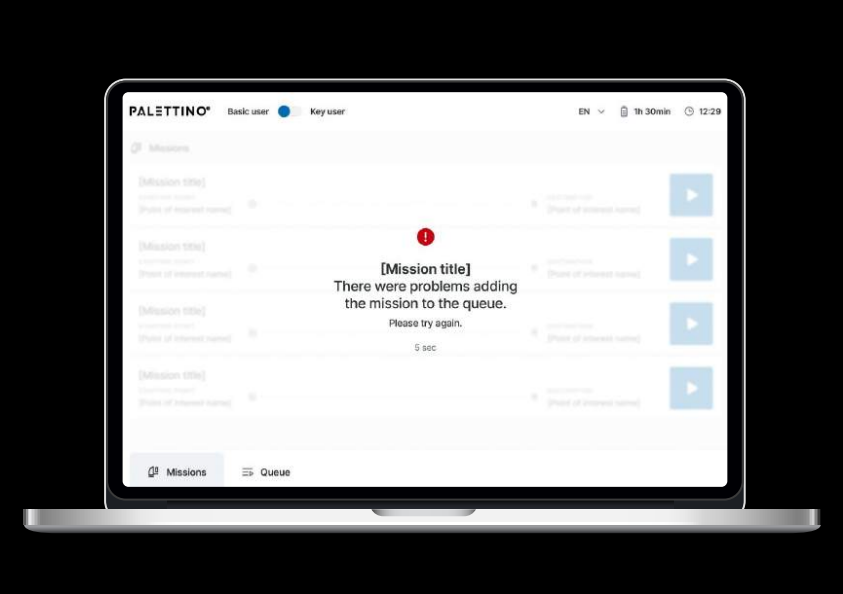
Each mission has the following **attributes:**

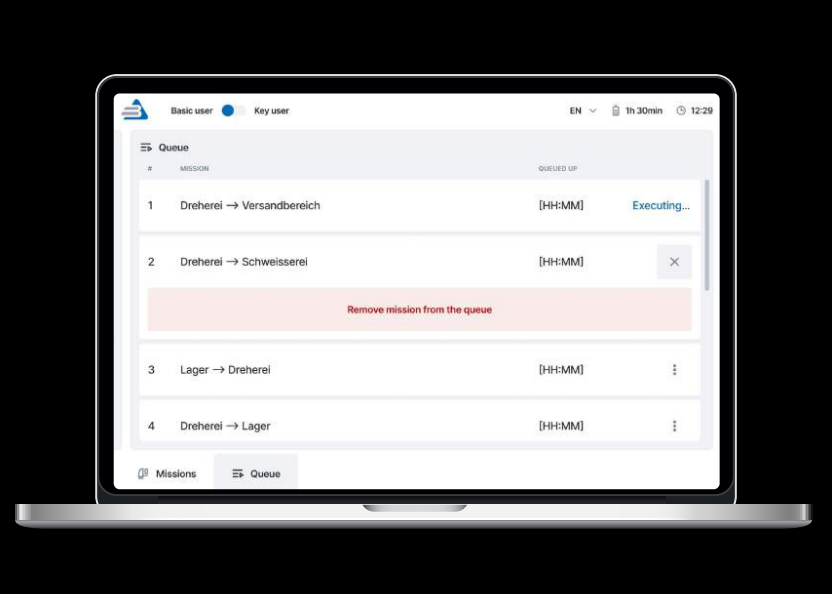
- Name
- Starting point of interest (POI)
- Destination POI

A Key user can **add any mission from the list to the queue** by clicking the "Play" button. The mission will appear at the last position after a five-second pause, during which the list of missions is locked. Then, the operator will be notified whether queuing the mission was successful or failed.

There are four mission statuses:

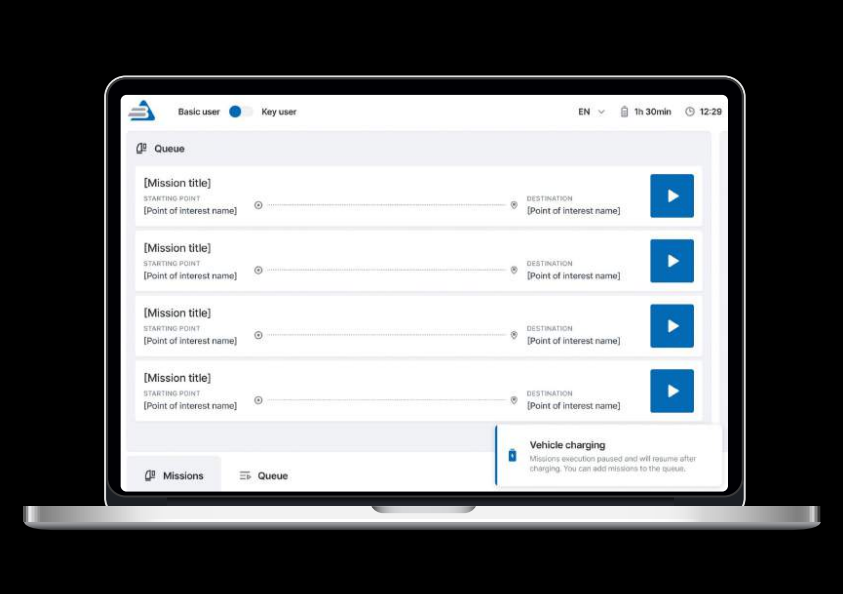
- Active
- Queued
- Completed
- Deleted

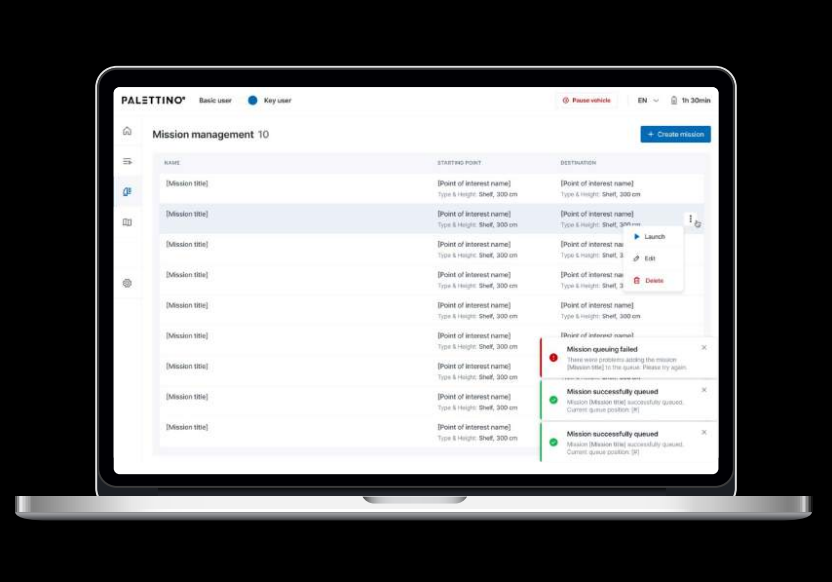




A Key user can **remove any missions in the queue** except the active ones.

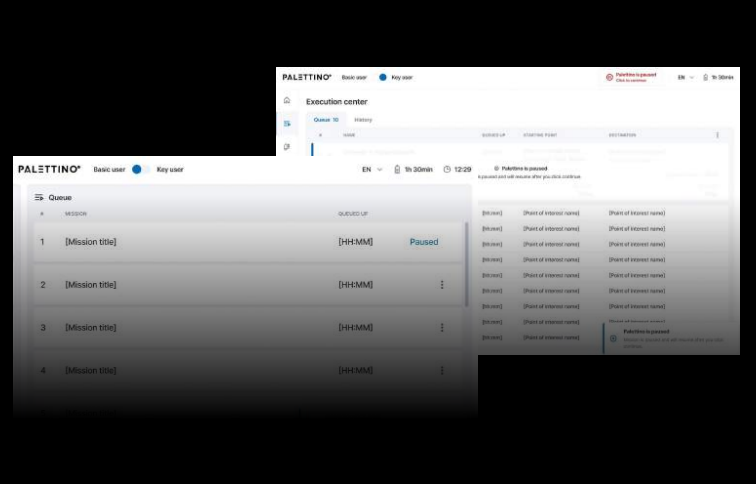
Both Basic and Key users get **toast notifications** about events related to the robot's operation. The alert about the robot charging is shown until the process is completed.

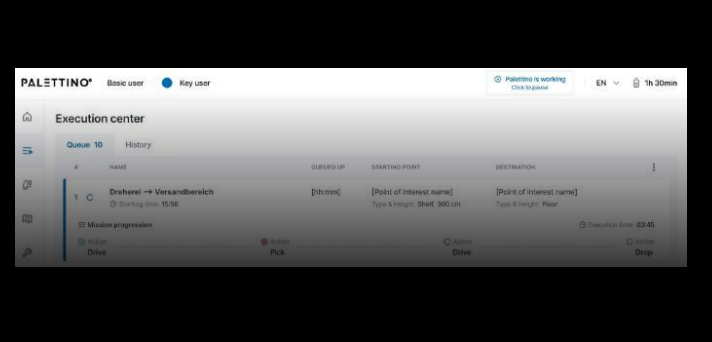




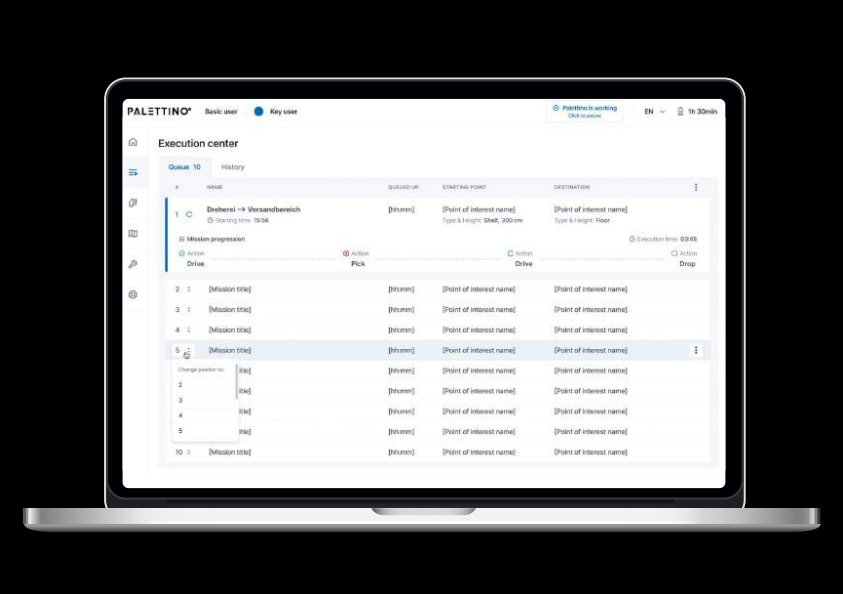
The opportunity to **pause the robot** is available from any application page for both roles.

One of the client's fundamental requirements was tracking a robot mission in real time, so we **set the execution time to update every second.**

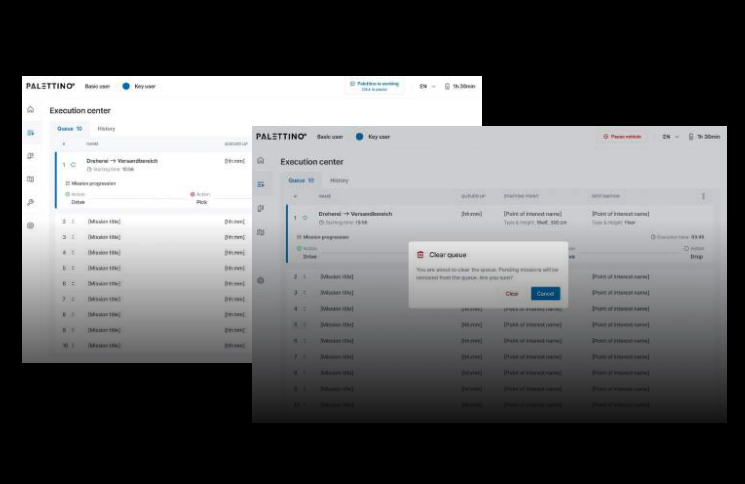




A Key user can **change the order of queued missions.**



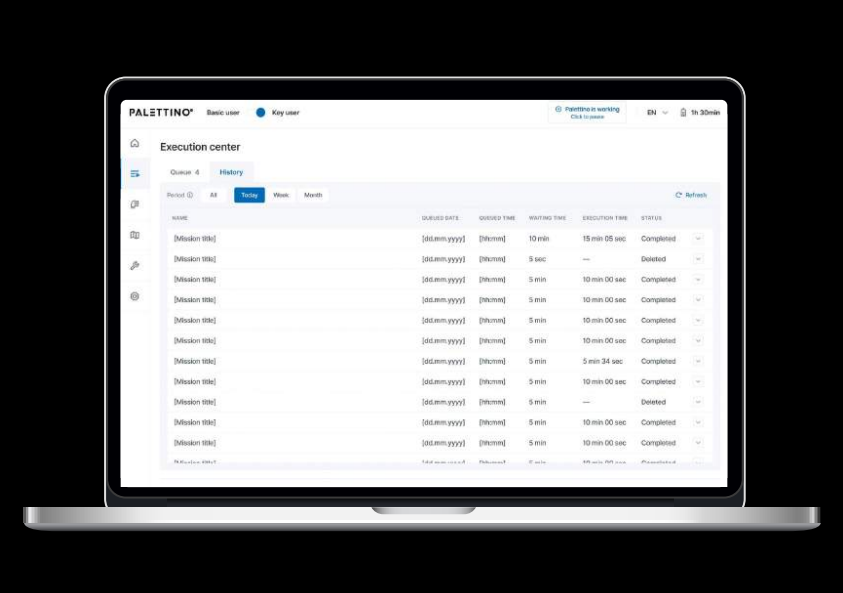
Mission removal is available for a Key user. They need to click the appropriate menu item and follow the additional action confirmation.



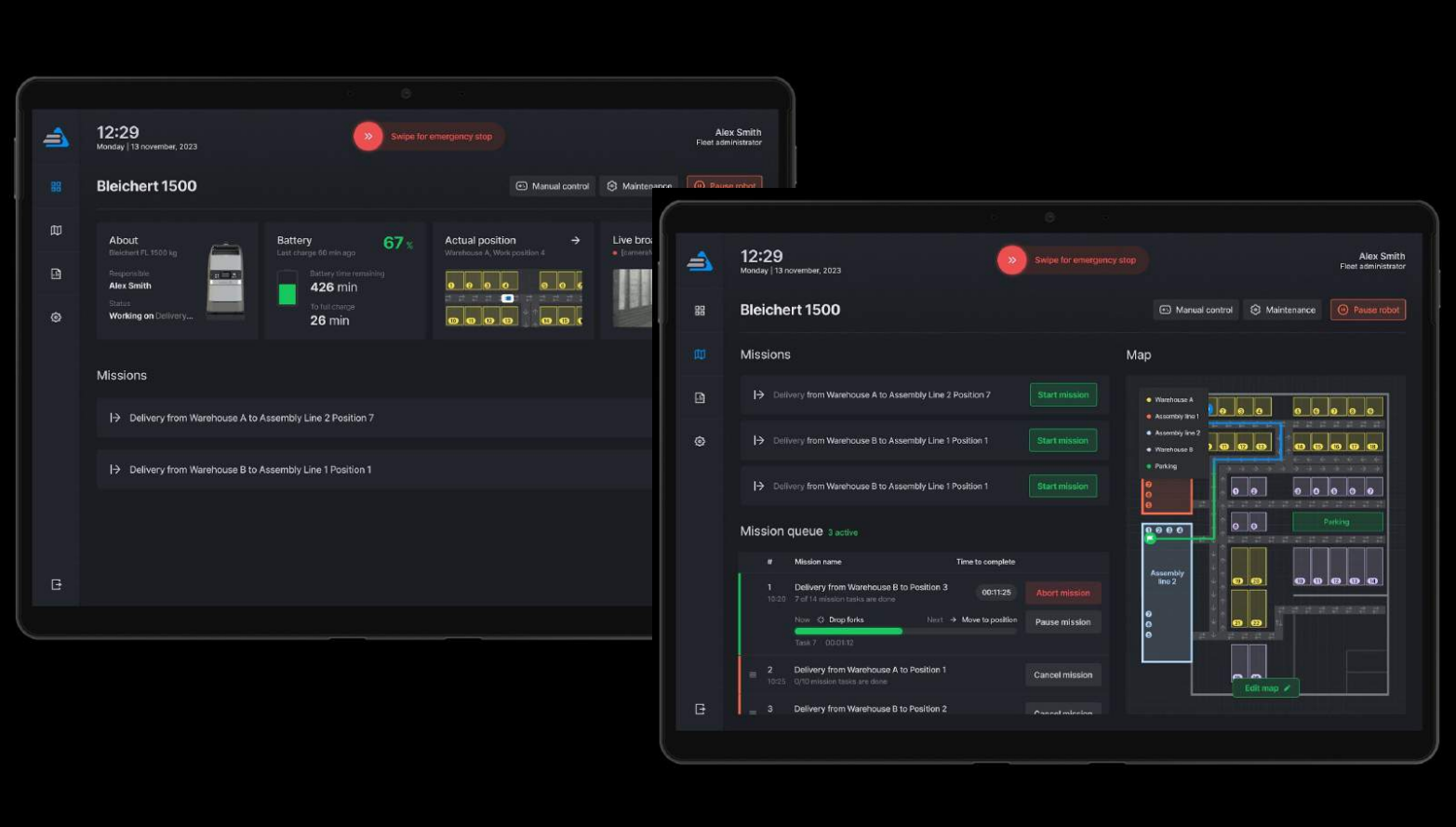
Only a Key user can access the **History of missions and filter them by date:**

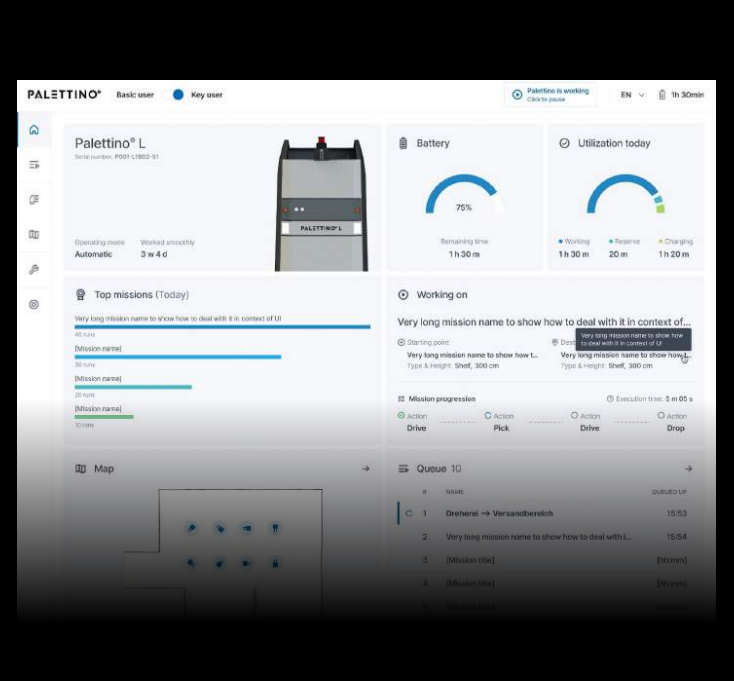
- Today
- Week
- Month

All missions are shown by default.



That's what the **clickable prototype** looks like.

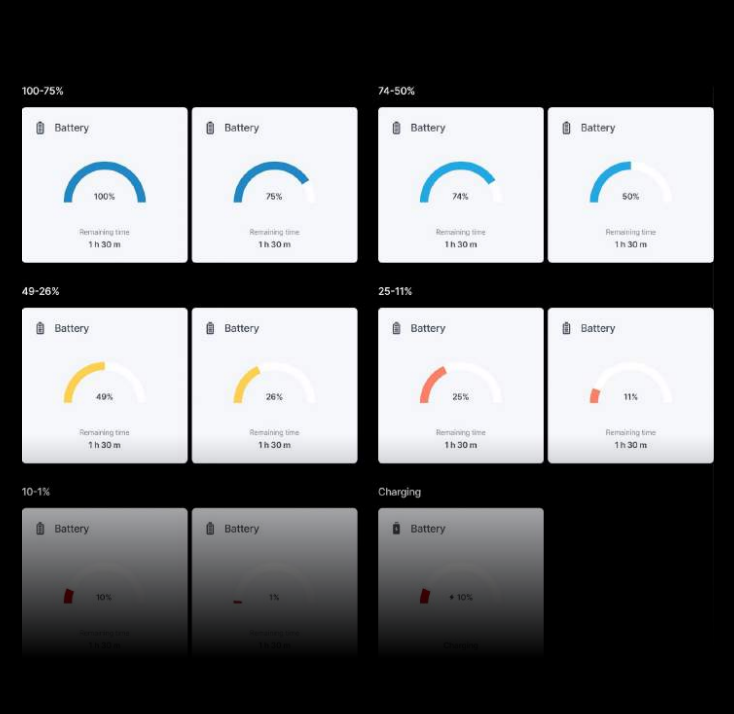




Dashboard for monitoring robot condition

The complex interface wasn't the only reason the client's customers considered previous forklift systems not user-friendly enough. They also strived to have simple visualizations of the robot's status to detect and fix machine maintenance issues before they snowball.

A dashboard with widgets and actual information about the robot's state empowered users to switch to the 'danger foreseen is half avoided' principle and, thus, **decrease machinery maintenance costs.**

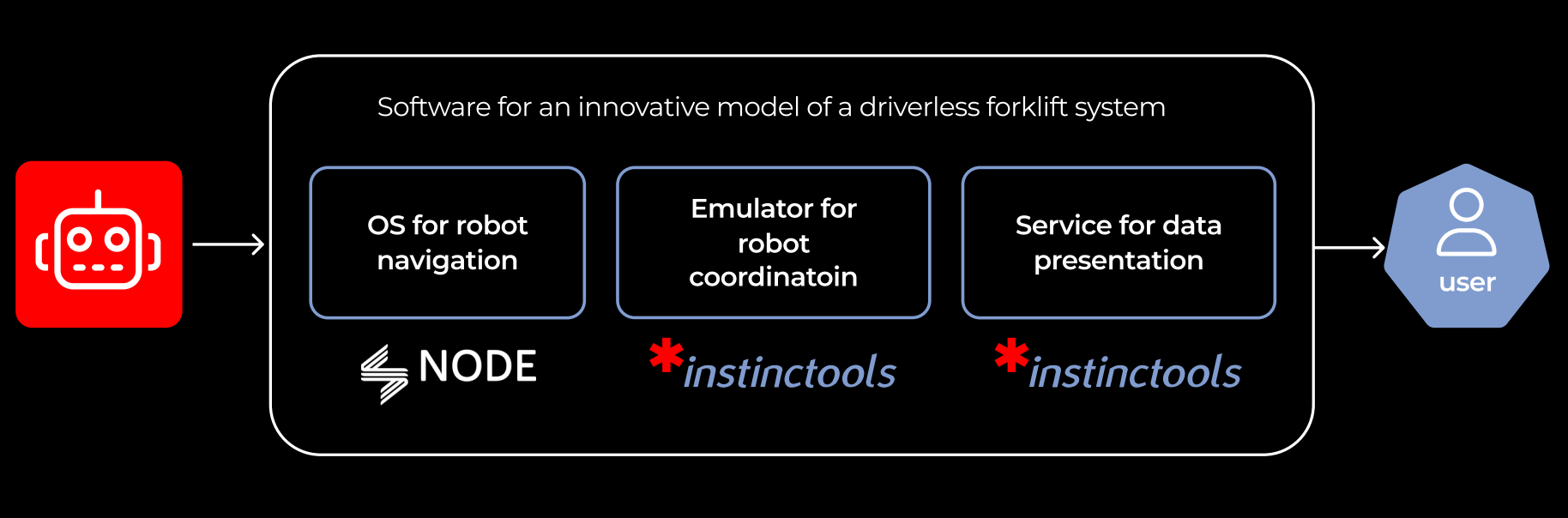


04 Writing a test emulator for the app

The client's web app is an example of an embedded solution that strongly depends on the hardware. In such cases, software testing takes a special place in SDLC. There are two ways to perform it:

- **Running the app directly on the robot.** However, this was a no-go option, as the client wanted to fine-tune the new model's sensors before the trade show. Transporting the forklift back and forth between their headquarters in Osterburken and our development hub in Warsaw would totally mess up the project's timeline.
- **Testing software on an emulator.** The client didn't have an emulated environment that imitates the robot's behavior, so we offered our help and built it.

As we were working on the emulator ourselves, the increased project scope could have put the *instinctools team in a tight spot, given the strict deadline. However, we managed to deliver the robust version of the app on the dot. Moreover, our emulator served as a **temporary substitute for the solution's service for the robot coordination,** making it possible for the client to showcase the trailblazing forklift model at the trade show.


Software for an innovative model of a driverless forklift system
OS for robot navigation
Emulator for robot coordinatoin
Service for data presentation
NODE
instinctools
instinctools
user

05 Shaping the post-MVP backlog

Satisfied with the MVP results, the client decided to proceed with our partnership at the post-MVP stage, during which our team will broaden the functionality related to these three areas:

- User management
- Mission improvements
- Mission assignment

Before

- Complicated app meant for employees with an engineering background
- Lack of visualization
- Partly manual control over a forklift robot on-site

After

- Intuitive app sets new standards in user-friendliness of robotics management software
- Easy-to-grasp dashboards
- Fully remote control of the forklift robots

Business value

- Trailblazing warehouse solution for the **SMB segment**
- **Highly user-friendly UI** suitable for non-technical staff makes the software stand up from the crowd of solutions designed for tech-savvy employees
- **Simplified software support** due to a homogeneous tech stack
- Interface with **no more than three clicks** to reach any feature
- **Less than 0.4 sec response** to the user input
- **Real-time** updates of the robot missions
- **Easy-to-use** dashboards for tracking a robot's state
- **Decreased** robot maintenance cost thanks to detecting malfunctions before they turn into serious issues

Client's testimonial

> *We are glad to find a reliable tech ally in the face of *instinctools. Despite the strict time limits, they met all of our expectations and went the extra mile to guarantee the decent and stable performance of the app at the trade show. As we look back, they were the ones who enabled us to step up warehouse robotic automation and create the next and best version of our Palettino® forklift system.*

Benjamin Kirsten
Head of Development at Bleichert

Do you have a **similar project idea**?

© instinctools