

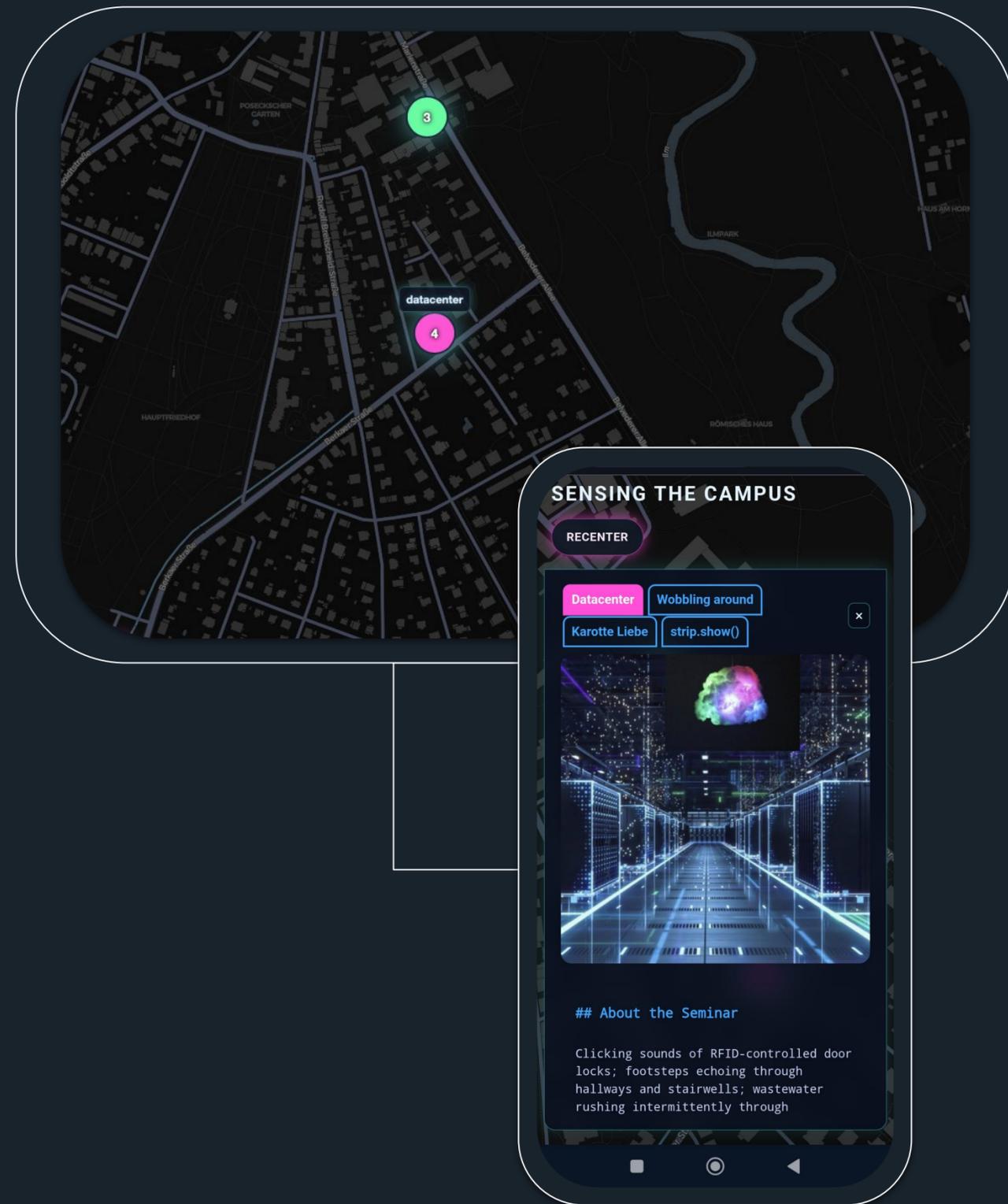
Datacenter.

Sensing the Campus, WiSe 25/26

Mirjam Nowotny, 127300, mirjam.luisa.nowotny@uni-weimar.de

Project Overview

The **Datacenter Website** functions as the public interface of *Sensing the Campus*. It **maps the distributed sensor nodes** installed across the campus and invites visitors to open each node to **discover its individual story**. **Live and historical sensor readings** are visualized through **data graphs**, while a real-time backend streams incoming values and preserves the archive for retrospective exploration. Designed for navigation on **desktop and mobile devices**, the interface uses **interactive clusters and cards** to **reveal the campus-wide sensing network** as a coherent system.



Project Overview

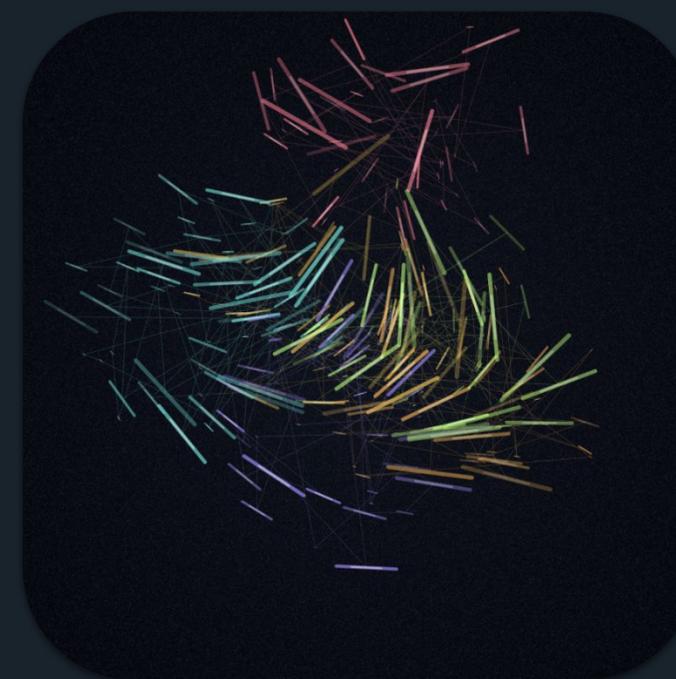
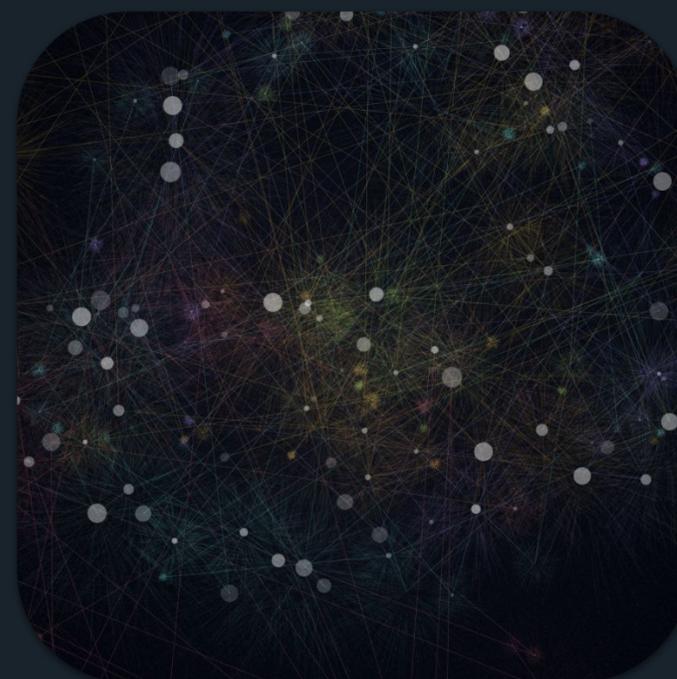
During the **Winterwerkschau 2026**, the website also operated as an on-site entry point. QR codes attached to sensor locations across campus **linked directly to the platform**, where people were guided to the exhibition venue. In this way, the Datacenter Website not only **visualizes the hidden rhythms of the building infrastructure** but also **extends them into a navigable public experience**.

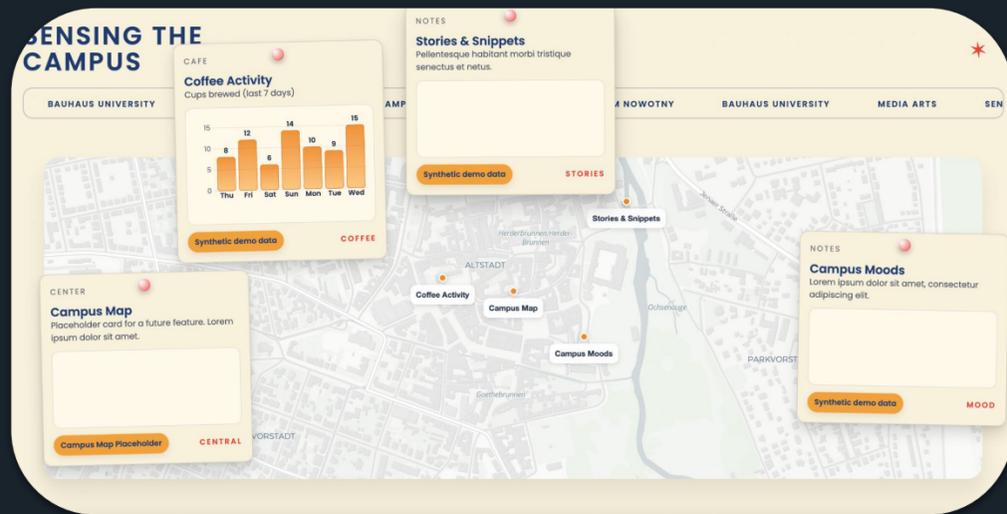


Concept and Intent

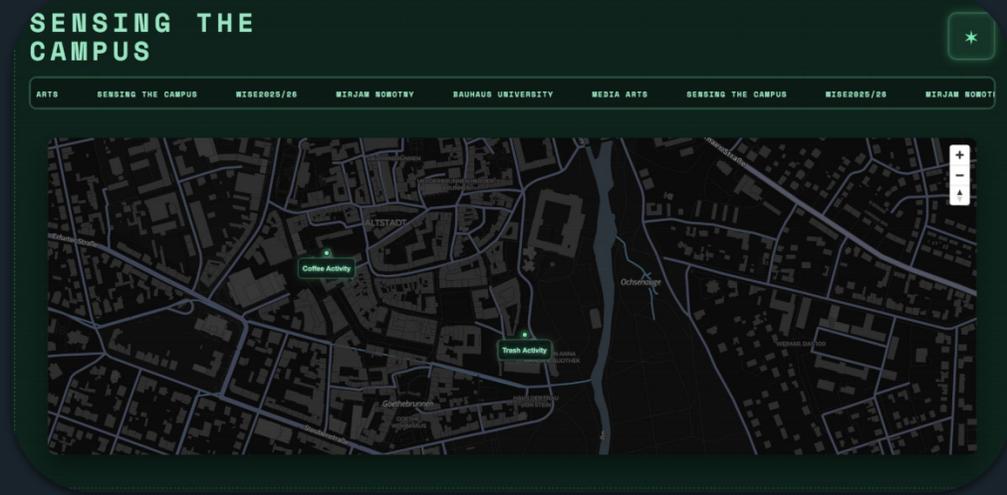
From an early interest in data visualization, the idea of **transforming sensor input into a digital artwork** was present from the beginning. Given my background as a software engineer, the technical feasibility felt realistic.

**On this slide you can see my first tests on data art that I made in the beginning of the semester.*





Concept and Intent



When the ideas of the other StC participants emerged the idea of creating a **central point for all the data incoming** became more appealing. The project would not only become an artwork on itself but also **connect all the other projects into one**, no matter how different they are.

**On this slide you can see my first drafts of a datacenter-website*

Concept and Intent

At the same time, I did not want to abandon the original idea of making the **data itself visually expressive**. The goal therefore evolved beyond simply marking sensor locations on the website: each node would also expose its **live readings** through graphs. In this way, the platform not only indexes the distributed sensing infrastructure but also **renders its continuous activity aesthetically**.

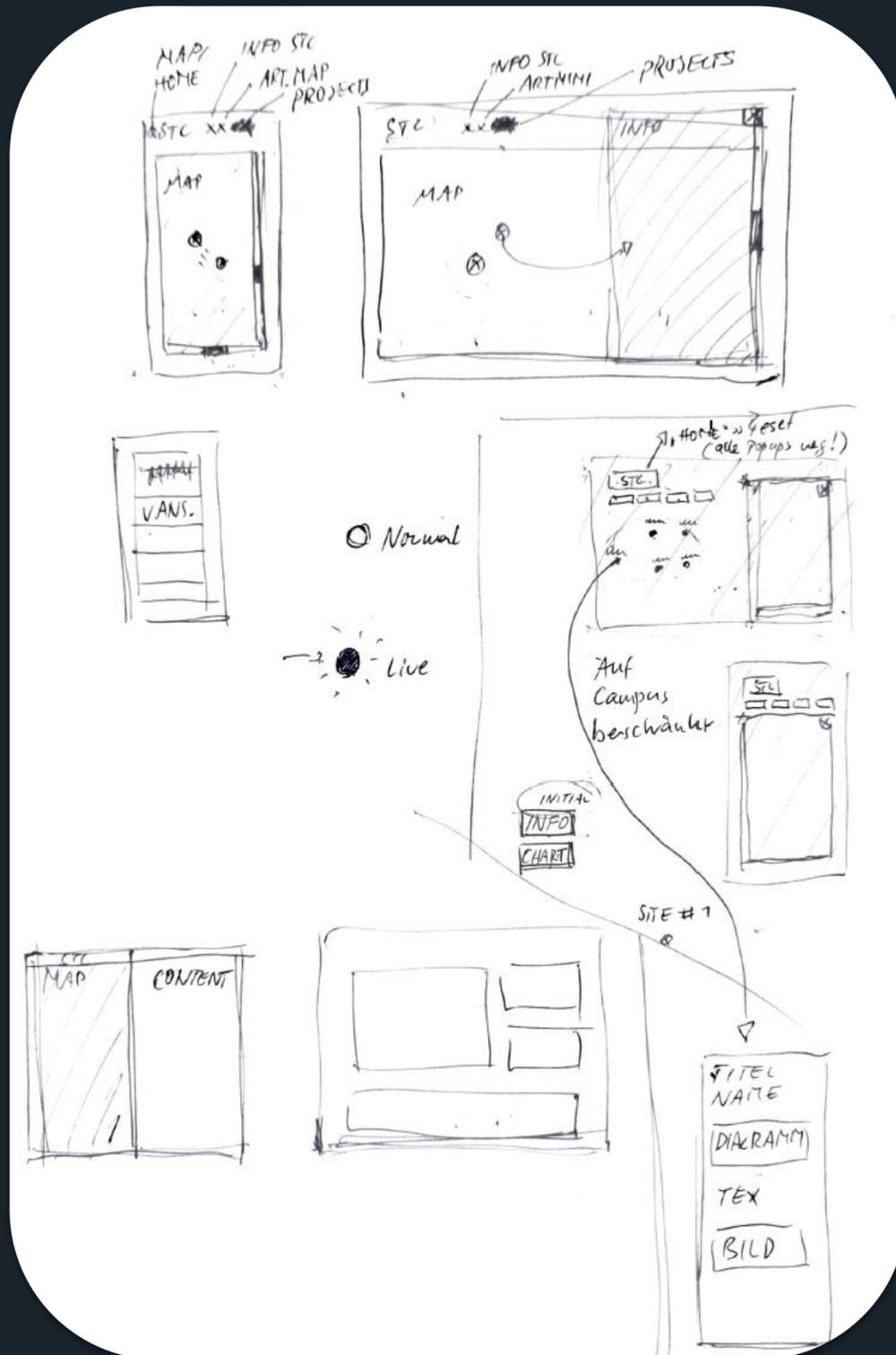
**On this slide you can see the sensor data of different projects that is also visible on the website*



Concept and Intent

After deciding to build a datacenter website to unify all projects in one central hub, Christian Doeller and I developed **both a technical concept and the initial design vision for mobile and web**. From that point, I iteratively designed and implemented the frontend.

**This slide shows a sketch of the design and programming foundation Christian and I created.*





The **physical layer** consists of 7 IoT sensors, each connected to an ESP32.

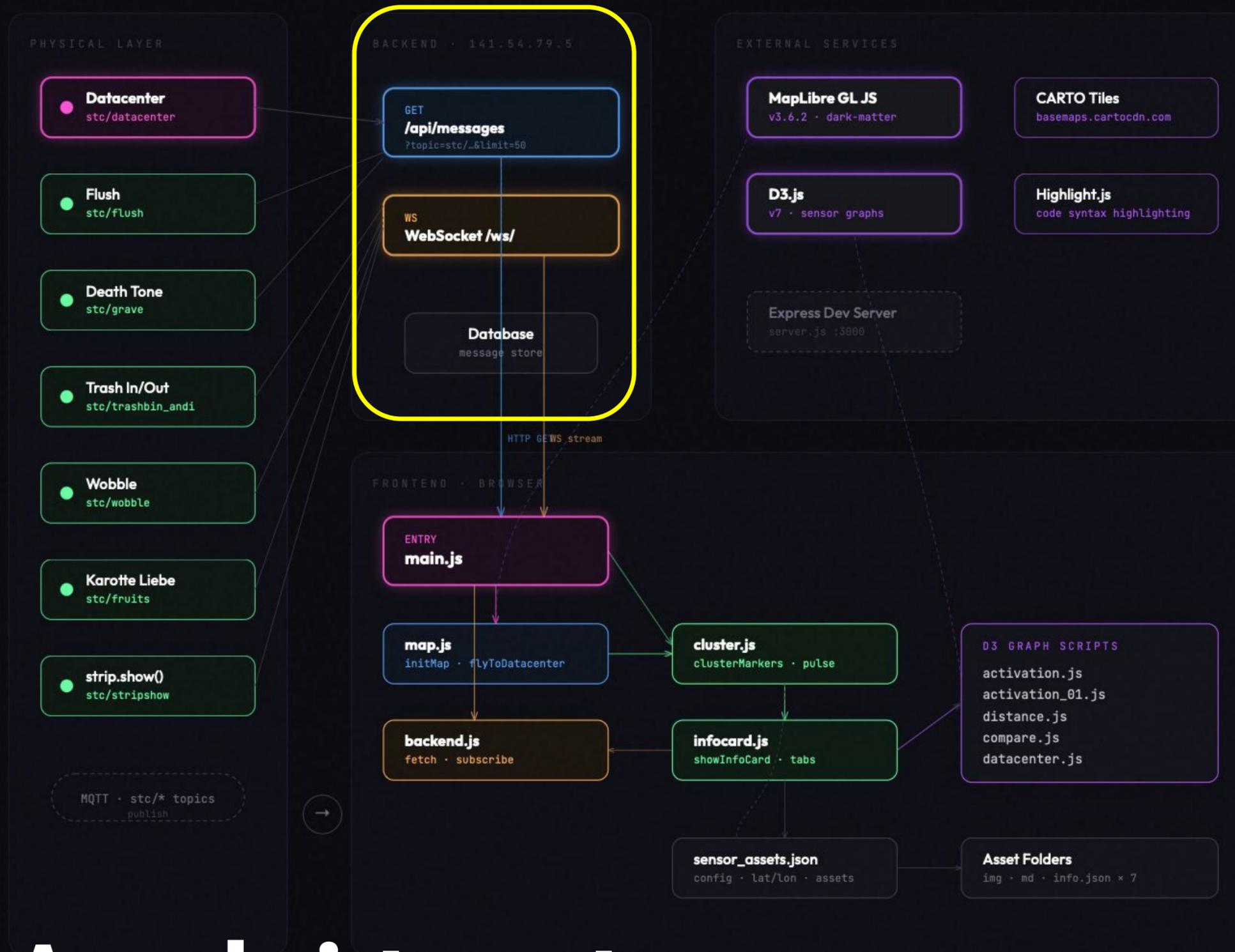
They publish data **over MQTT** to topics under the stc/namespace

The Datacenter node acts as the **central hub**

Each sensor has its own asset folder with an image, markdown description, and a specific D3 graph type assigned to visualize its data

**Technical diagram fully visible (including small explanations) in attached architecture.html)*

Architecture

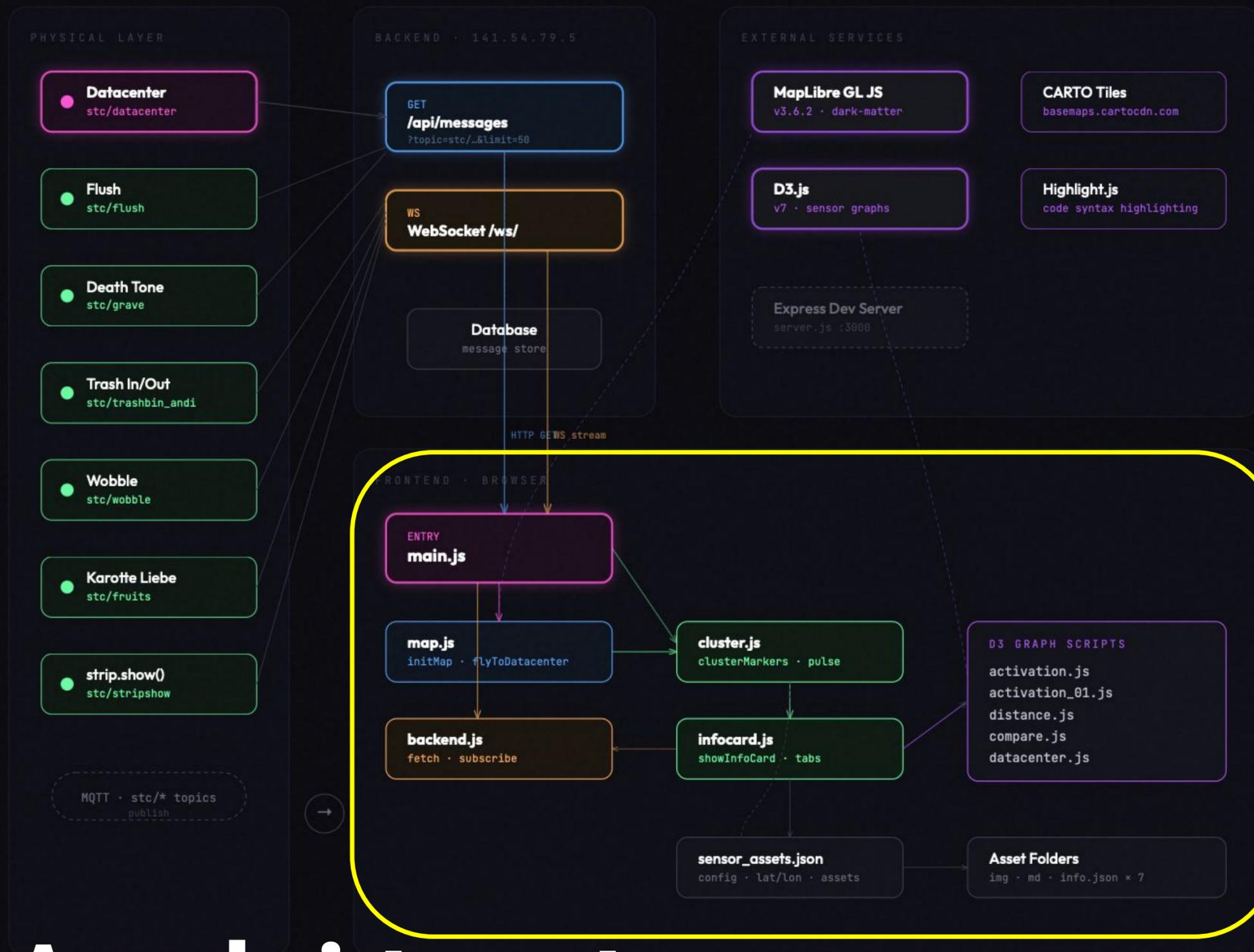


The backend runs on a single and does two things: it stores all incoming MQTT messages in a database, and it exposes that data to the browser through two channels (developed by Christian Doeller)

A REST API at /api/messages lets the frontend fetch historical readings with optional ?topic and ?limit filters, **while a WebSocket endpoint at /ws/ pushes live sensor updates.**

**Technical diagram fully visible (including small explanations) in attached architecture.html)*

Architecture



The frontend is a **vanilla JS** app that runs entirely in the browser. **main.js** is the entry point.

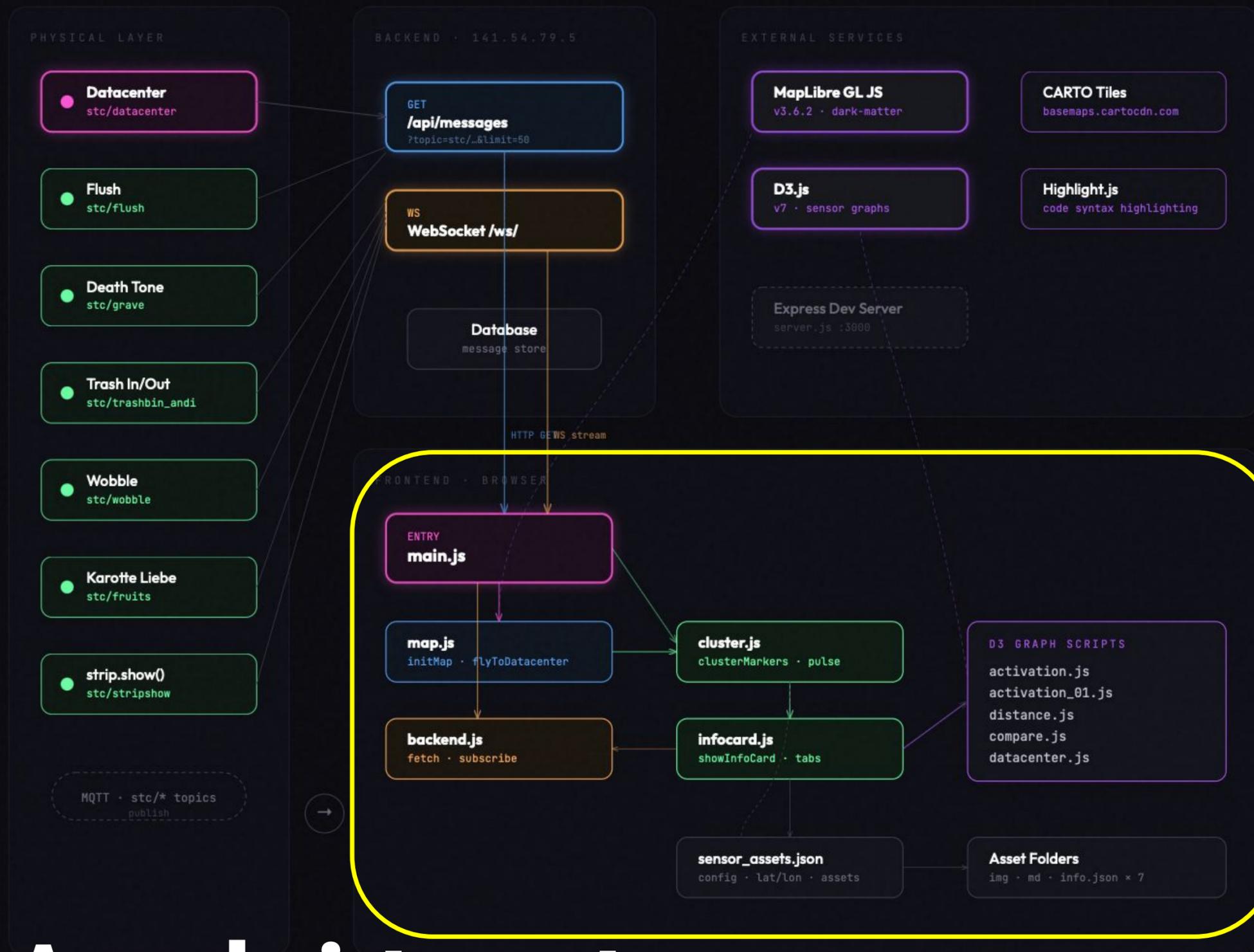
It initializes the map, fetches historical data via the **REST API**, and opens a **WebSocket connection for live updates**.

The map layer uses **MapLibre GL with CARTO dark-matter tiles**, bounded to the **Weimar campus area**.

cluster.js handles **grouping** nearby sensor markers into numbered dots based on pixel distance, and re-clusters on every pan or zoom.

**Technical diagram fully visible (including small explanations) in attached architecture.html*

Architecture



The frontend When live data arrives, it pulses the relevant cluster marker. Clicking a marker opens the info card, which is the main UI panel managed by `infocard.js`. It loads the sensor's image, renders its markdown description, and dynamically imports the appropriate D3 graph script (one of five visualization types depending on the sensor). Everything is configured through `sensor_assets.json`, which maps each sensor key to its coordinates, display title, image, markdown file, and graph script.

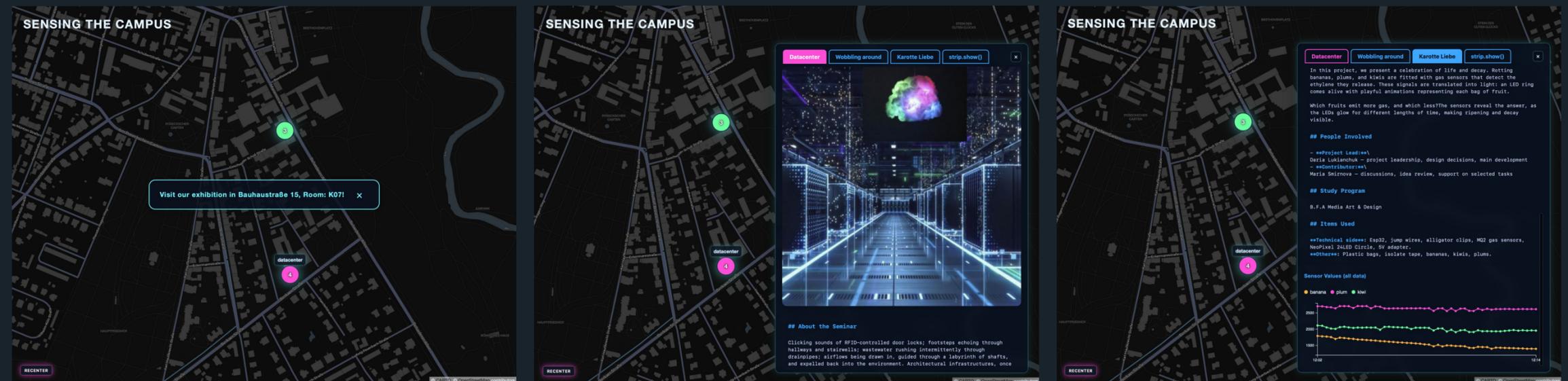
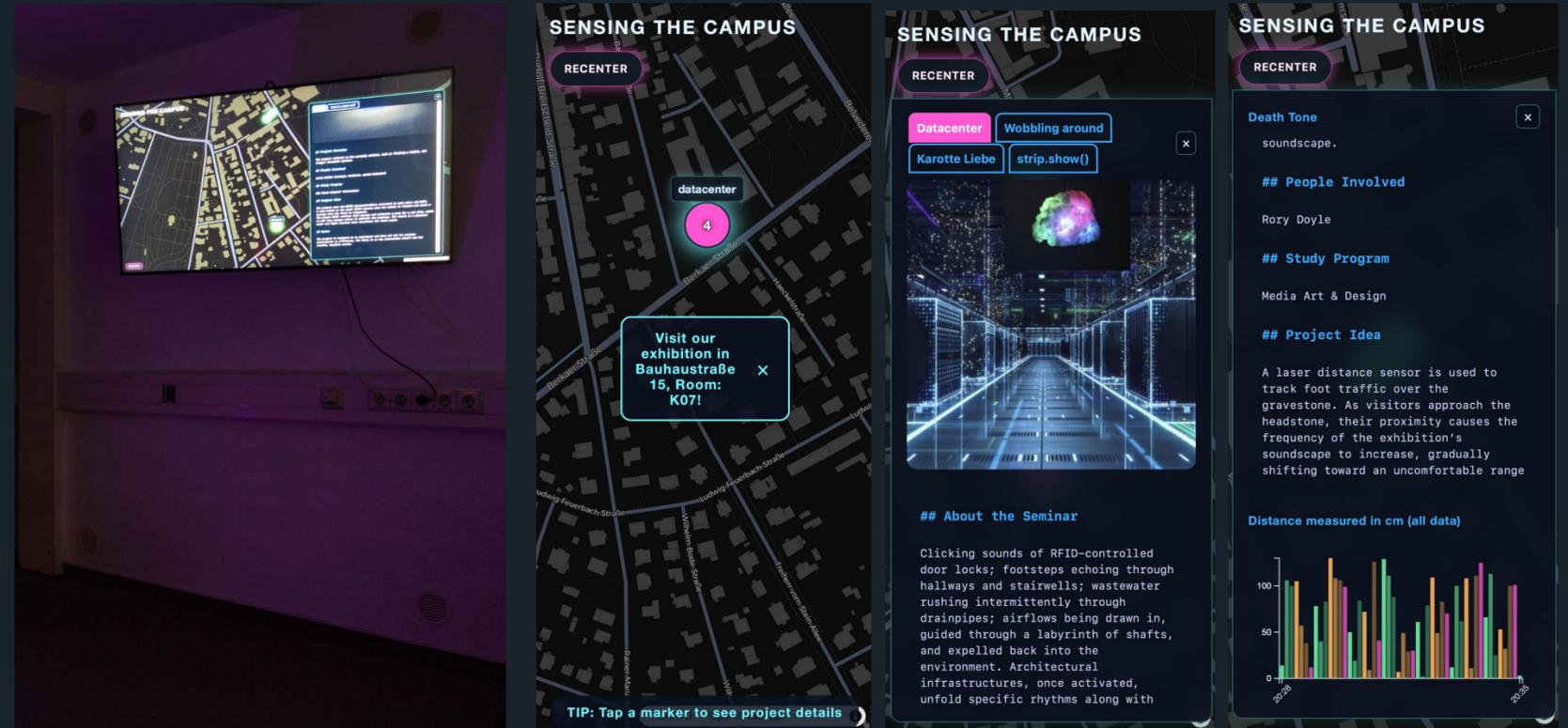
**Technical diagram fully visible (including small explanations) in attached architecture.html)*

Architecture

The Datacenter

**Thanks to everyone who participated in StC and contributed their project data.*

**Special thanks as well to Christian Doeller for providing the backend support.*



Sensing the Campus, WiSe 25/26

Mirjam Nowotny, 127300, mirjam.luisa.nowotny@uni-weimar.de