

Das Datenvogelhäuschen

Projektbericht zum VDE Rhein-Ruhr e. V. Technikpreis 2021/2022



Ein Projekt von Schülern des Don-Bosco-Gymnasiums

Über uns:

Anschrift der Schule
Don-Bosco-Gymnasium Theodor-Hartz-Straße 15 45355 Essen

Liste der Beteiligten:

Name	Adresse	Geb. Datum
Wodarczak, Marian	[REDACTED]	07.06.2005
Tillenburg, Mathis	[REDACTED]	20.04.2005
Engelhardt, Ben	[REDACTED]	21.04.2005
Lorenz, Max	[REDACTED]	07.02.2005
Labas, Samir	[REDACTED]	07.10.2005
Bohnhorst, Joshua	[REDACTED]	18.11.2004

Betreuender Lehrer:

Lübbering, Christian (Technik, Physik, Erdkunde)



Projektkurzbeschreibung:

Vogelschutz ist in Deutschland ein wichtiges Thema. Durch jahrelange Vernachlässigung von Naturschutz ist der Vogelbestand alleine in den letzten 20 Jahren um etwa die Hälfte gesunken. Hauptprobleme? Nahrungsknappheit durch Insektensterben, klimatische Veränderung und das Nichtbestehen von sicheren Nistmöglichkeiten. Wir wollen einen Beitrag dazu leisten, diese Probleme zu lösen. Es entstand: Das Datenvogelhäuschen.



Abbildung 1:
Ausgestattetes
Vogelhäuschen am Baum
(Prototyp)

Das Datenvogelhäuschen ist ein unscheinbarer, ca. 8 x 8 x 4 cm großer Plastikkasten, welcher problemlos in bestehende oder neue Nistkästen installiert werden kann. Dieser ist dann in der Lage, Daten zum Vogel und zu dessen Umwelt zu erfassen, aufzuzeichnen und zentral zur Auswertung zur Verfügung zu stellen.

Was trocken und wissenschaftlich klingt hat einen viel größeren Nutzen, als nur ein Diagramm zu füllen. Auch der Besitzer des Datenvogelhäuschens selber profitiert: So kann man über das eingebaute Webinterface einen Blick ins Vogelhaus werfen oder sich die aktuellen Wetterdaten (darunter: Temperatur, Luftfeuchtigkeit, Luftdruck, CO₂- und TVOC-Dichte) ansehen. Diese Daten kann der Nutzer dann als eine Art „Spende“ an einen Zentralserver schicken, welcher diese Daten in einer riesigen Datenbank indiziert und Umweltorganisationen eine Möglichkeit gibt, Statistik über das Brutverhalten von Vögeln in verschiedenen Regionen Deutschlands zu führen. Auch die direkte Kontrolle einzelner Nistkästen ist durch die eingebaute IR-Kamera möglich, sofern der Nutzer es erlaubt.

Im Gegenzug zu dieser „Datenspende“ erhält der Nutzer Zugriff auf Datenvogelhäuschen DASH; einem Online-Service, welcher es ermöglicht, von überall aus auf die Streams seines Datenvogelhäuschens zuzugreifen und diese, sofern man wünscht, auf z. B. den eigenen Social-Media-Kanälen oder einer eigenen Website zu teilen. Auch ist es über DASH gegen eine Spende möglich, Big-Data-Querys über unsere SQL-Datenbank laufen zu lassen um (anonym) zur Verfügung gestellte Daten der Geräte selber auszuwerten und für eigene Zwecke zu nutzen. Diese Zwecke können z. B. Verwendung im Unterricht, private Forschung und weitere sein. Umweltorganisationen bekommen auf Anfrage kostenfreien Zugriff.

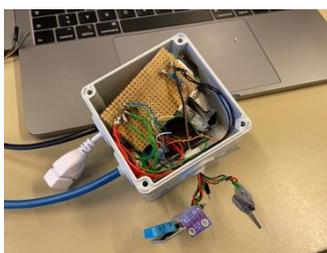


Abbildung 2: Technik des
Prototyps

Ziel des Datenvogelhäuschen-Projekts ist es, ein kostengünstiges System zu schaffen, welches problemlos von technisch unerfahrenen Anwendern installiert und genutzt werden kann. Eine simple Installation in Form von Verschrauben der Hardware im Vogelhäuschen und dem Anstecken des mitgelieferten Access Points – oder, bei mehr technischer Erfahrung: dem Erstellen eines WLAN-Netzwerks mit der SSID „Vogelhaus“ – sollen dies als einzige Installationsschritte ermöglichen. Eine abschließende Konfiguration erfolgt über das Webinterface, welches unter <http://datenvogelhauschen/> im Netzwerk erreichbar ist. Dieses wurde weitestgehend erreicht.

Ausführliche Projektbeschreibung:

Wie bereits in der Zusammenfassung des Projektes erwähnt, hat das Datenvogelhäuschen-Projekt das Ziel, ein in sich geschlossenes System zu erschaffen, welches ein Nutzer in ein Vogelhaus installieren kann, mit Strom versorgt und anschließend nutzen kann – ohne großes technisches „Knowhow“. Zentrale Bestandteile dessen sind dementsprechend die Planung und Umsetzung von Hardware, Firmware und Web-Service, welche im Folgenden separat thematisiert werden:

Hardware des Datenvogelhäuschens:

Das technische System des Datenvogelhäuschens besteht aus verschiedensten Komponenten: einem Mikrokontroller, einer Kamera, mehreren Sensoren, einer Status-LED, IR-Beleuchtung und einer PV-Anlage, welche bei uns als Energieversorgung eingesetzt wird (der Einsatz des Datenvogelhäuschens ist selbstverständlich auch ohne PV-Anlage möglich, insofern der Nutzer Strom in seinem Garten – oder Wald – hat). Das Datenvogelhäuschen in sich ist nach folgendem Schema verschaltet:

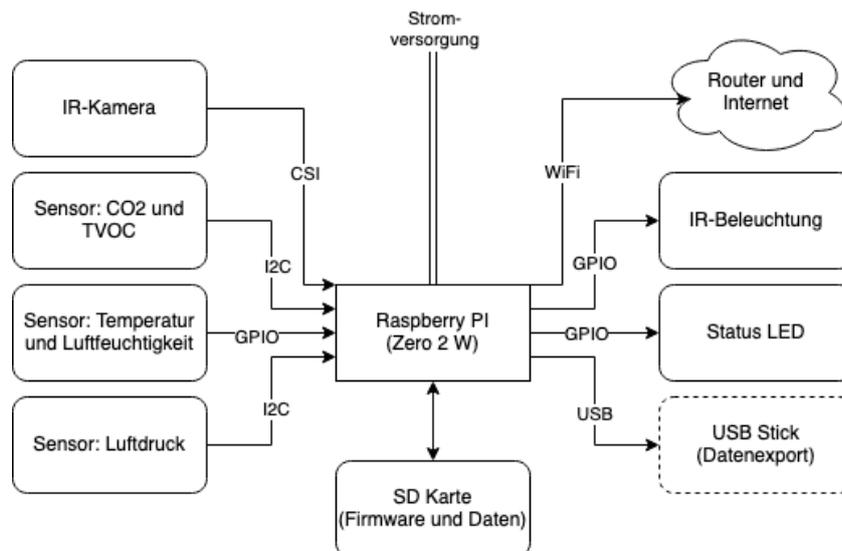


Abbildung 3: Verschaltung der Komponenten im Datenvogelhäuschen.
Auf den Pfeilen: Verbindungstyp; Gestrichelter Kasten: Optional präsent

Um eine Stromversorgung sicherzustellen ist in unserem Falle eine PV-Anlage nötig. Da das Datenvogelhäuschen über einen standardisierten USB 2 (5V; bis zu 3A) Anschluss versorgt werden kann, ist es problemlos möglich, dieses mit vielen fertigen PV-Systemen zu verwenden: So taten wir es auch. Unsere PV-Anlage ist nach folgendem Schema verschaltet (im oberen Diagramm als „Stromversorgung“ dargestellt):

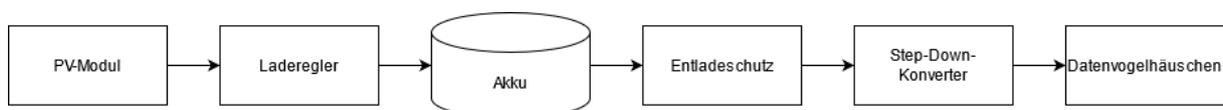


Abbildung 4: Aufbau unserer PV-Anlage

Diagramme sind zwar informativ, meist jedoch auch langweilig. Dementsprechend hier einige Bilder von unserem echten Aufbau:



Abbildung 5: Inhalt der wasserdichten Box für die PV-Anlage. Hier zu sehen: Laderegler und Entladungsschutz



Abbildung 6: PV-Modul auf dem Dach unserer Schule. Da keine Befestigungsmöglichkeit vorhanden war, nutzten wir eine Palette zum Beschweren.

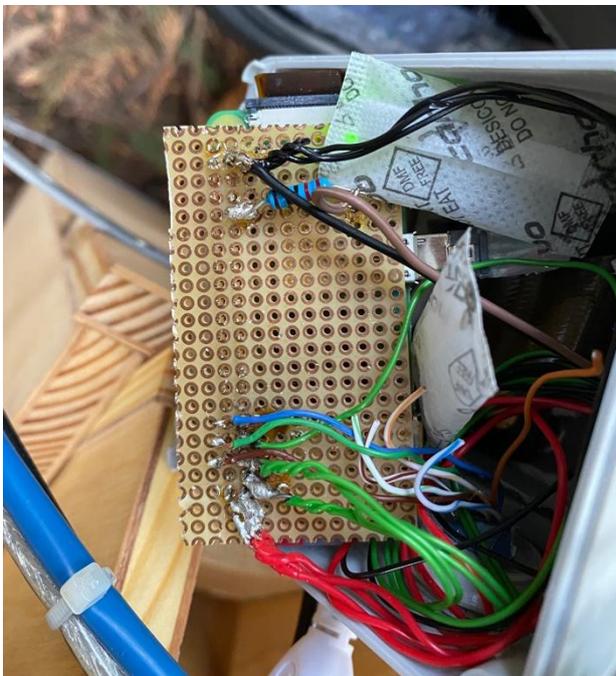


Abbildung 7: Inhalt des Prototyps. Silica-Gel-Beutel halten die Technik auch im Notfall trocken.

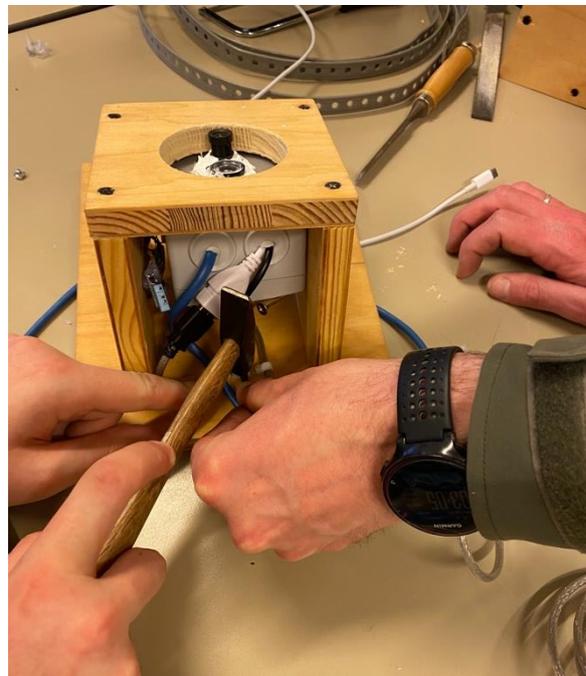


Abbildung 8: Der Kasten des Prototyps wird in die Decke eines Vogelhäuschens installiert.

Bei dem auf den Bildern abgebildeten Aufbau handelt es sich noch um unseren Prototypen. Alle Komponenten, sind zwar, wie sich auch im fertigen Produkt verbaut werden, verbunden, sind jedoch nur (unordentlich) in eine Verteilerbox eingeklebt. Für das fertige Produkt gibt es eine spezielle 3D-gedruckte Hülle, welche um einiges

kleiner ist und somit eine Installation in alle Vogelhäuser ermöglicht – nicht nur in eigenes dafür angefertigte, wie beim gezeigten Prototypen. Für diesen war es nötig, ein spezielles Vogelhaus mit doppeltem Dach zu bauen, welches Platz für die bis dahin noch klobige Technik bot:



Abbildung 9: Das Dach des Datenvogelhäuschen Prototypen bietet Platz für Elektronik



Abbildung 10: Das Dach kann einfach auf den Nistkasten gesetzt werden

Unsere finale Hülle unterscheidet sich wesentlich von der vorher genutzten Installationsmethode:

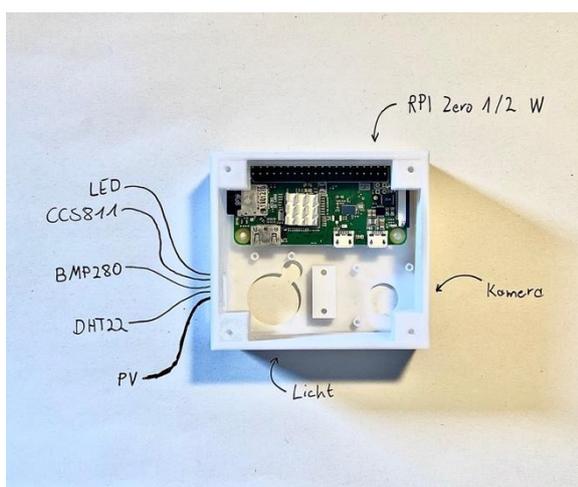


Abbildung 11: 3D-gedrucktes Gehäuse mit angedeuteter Verkabelung/Bestückung. Sensoren hängen wie beim Prototypen außerhalb des Gehäuses und können nach Möglichkeit aus dem Vogelhaus geführt werden, um genauere Messwerte zu ermitteln.



Abbildung 12: Geschlossene (unverschraubte) Box mit Datenvogelhäuschen.de-Schriftzug und Seriennummer (nötig zur Registrierung)

Die finale Version des Datenvogelhäuschens kann sich nun sehen lassen. Ein schlichter, stabiler weißer Kasten – einfach zu installieren und fast wartungsfrei im Betrieb. Dieser hat jedoch ohne die richtige Firmware keinen Zweck, entsprechend wird diese nun beschreiben.

Firmware des Datenvogelhäuschens:

Die Firmware des Datenvogelhäuschens basiert auf Raspberry Pi OS (zuvor: Raspbian), welches eine Debian-basierte Linux Distribution ist, welche speziell für Raspberry Pis entwickelt wurde. Wenn auch zunächst geplant war, Arch Linux Embedded zu verwenden um den Stromverbrauch zu senken, stellte sich das offizielle Raspberry Pi OS nach einigen Tests als wesentlich stabiler heraus – was in einer „Production“-Umgebung mehr Priorität als ein geringerer Stromverbrauch hat. Ziel ist es, eine Software zu schaffen, welche monatelang ohne (manuellen) Neustart laufen kann, ohne sich selbst zu beeinträchtigen. Die Grundlagen waren mit Raspberry Pi OS gewährleistet.

Die eigentliche Software wurde in NodeJS mithilfe von TypeScript entwickelt. Warum? NodeJS ist eine recht moderne und trotzdem erprobte Programmiersprache, welche durch ihre Non-Blocking-Architektur eine ideale Nutzung des „schwachen“ ARM-Prozessors im Raspberry Pi Zero gewährleistet. Zudem handelt es sich um eine weit verbreitete Web-Programmiersprache: Pakete wie „express“ machen es möglich, einfach schnelle Web-Services auf Node-Basis zu entwickeln. Das war wichtig für uns, schließlich muss man ja auch irgendwie in das Vogelhaus hineinsehen können. Das ist über das integrierte Web-Interface möglich:

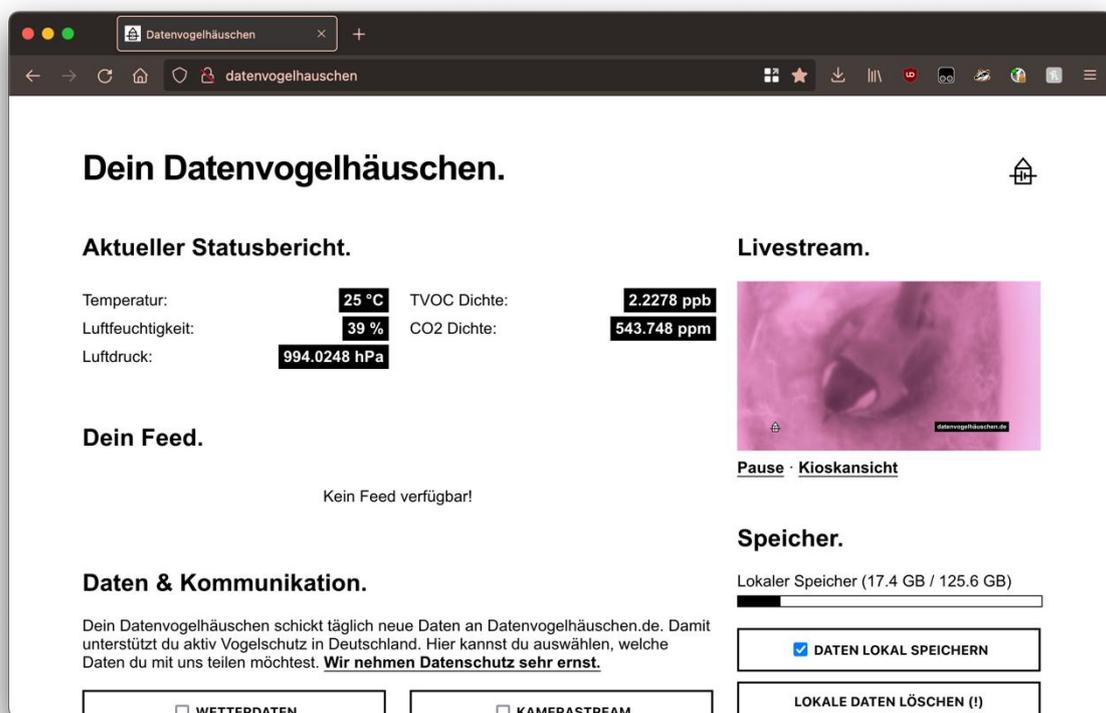


Abbildung 11: Lokales Web-Interface des Datenvogelhäuschens. Nur im lokalen Netzwerk erreichbar. Von hieraus kann u. A. verwaltet werden, welche Daten an unseren Zentralserver geschickt werden.

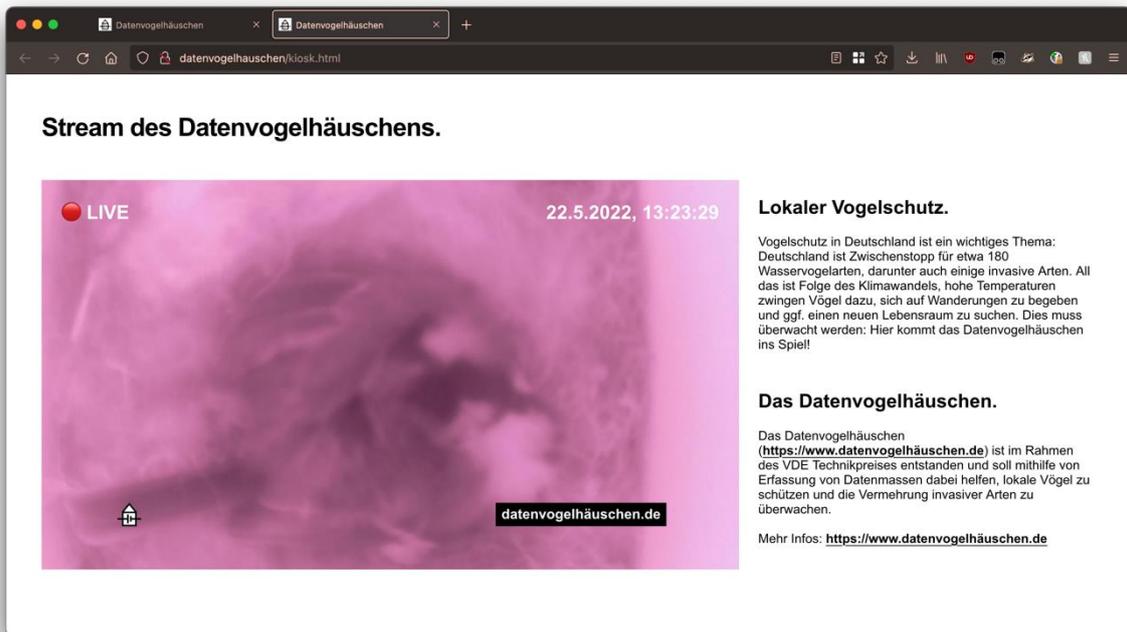


Abbildung 12: Die Kiosk-Ansicht des Datenvogelhäuschens ermöglicht das Ausstellen des Streams des eigenen Vogelhäuschens. Diese kann z. B. in einem Cafe genutzt werden, um den Gästen einen Einblick in das eigene Vogelhäuschen zu gewähren.

Neben dem Webinterface des Datenvogelhäuschens steht auch eine netzwerkinterne JSON-HTTP-API zur Verfügung welche es ermöglicht, Sensorwerte auszulesen, den Kamera-Stream weiterzuleiten und den allgemeinen Status des Datenvogelhäuschens zu überprüfen. Auch Einstellungen können über diese verändert werden.

Der gesamte Code des Datenvogelhäuschens ist in einem öffentlichen Git Repository auf GitHub zu finden: <https://github.com/johnst/datenvogelhaeuschen-software>. Damit ist auch der eigene Nachbau des Datenvogelhäuschens möglich, ohne sich an uns zu wenden. Schließlich wollen wir etwas verändern – kein Geld verdienen!

```

1  import {ESensorType} from "./ESensorType";
2
3  /**
4   * Sensor base class
5   */
6  export class Sensor {
7      static READ_INTERVAL: number = 15000;
8      static LAST_SENSOR_DELAY: number = 0;
9
10     identifier: string = "unidentified_sensor";
11     readableName: string = "Unidentified Sensor";
12     unitChar: string = "UU";
13     unitFull: string = "Unknown Unit";
14
15     sensorType: ESensorType = undefined;
16     pin: number = -1
17     bus: number = -1
18
19     value: number = 0;
20
21     /**
22     * To be called if a sensor has to be read in a cycle.
23     * This will execute this.read every _ ms.
24     */
25     startReadCycle = () => {
26         setTimeout(() => {
27             setInterval(this.read, Sensor.READ_INTERVAL);
28         }, Sensor.LAST_SENSOR_DELAY);
29     }
30     Sensor.LAST_SENSOR_DELAY += 2000;
31 }
32
33 /**
34 * Used to refresh sensor value
35 */
36 read = () => {

```

Abbildung 13: Screenshot eines Teils des Codes des Datenvogelhäuschens

Web-Service für das Datenvogelhäuschen:

Um das Datenvogelhäuschen erst im vollen Umfang nutzen zu können, ist ein Web-Service nötig. Dieser trägt Daten zusammen und stellt diese zur Verfügung. Die Architektur des Web-Services sieht wie folgt aus:

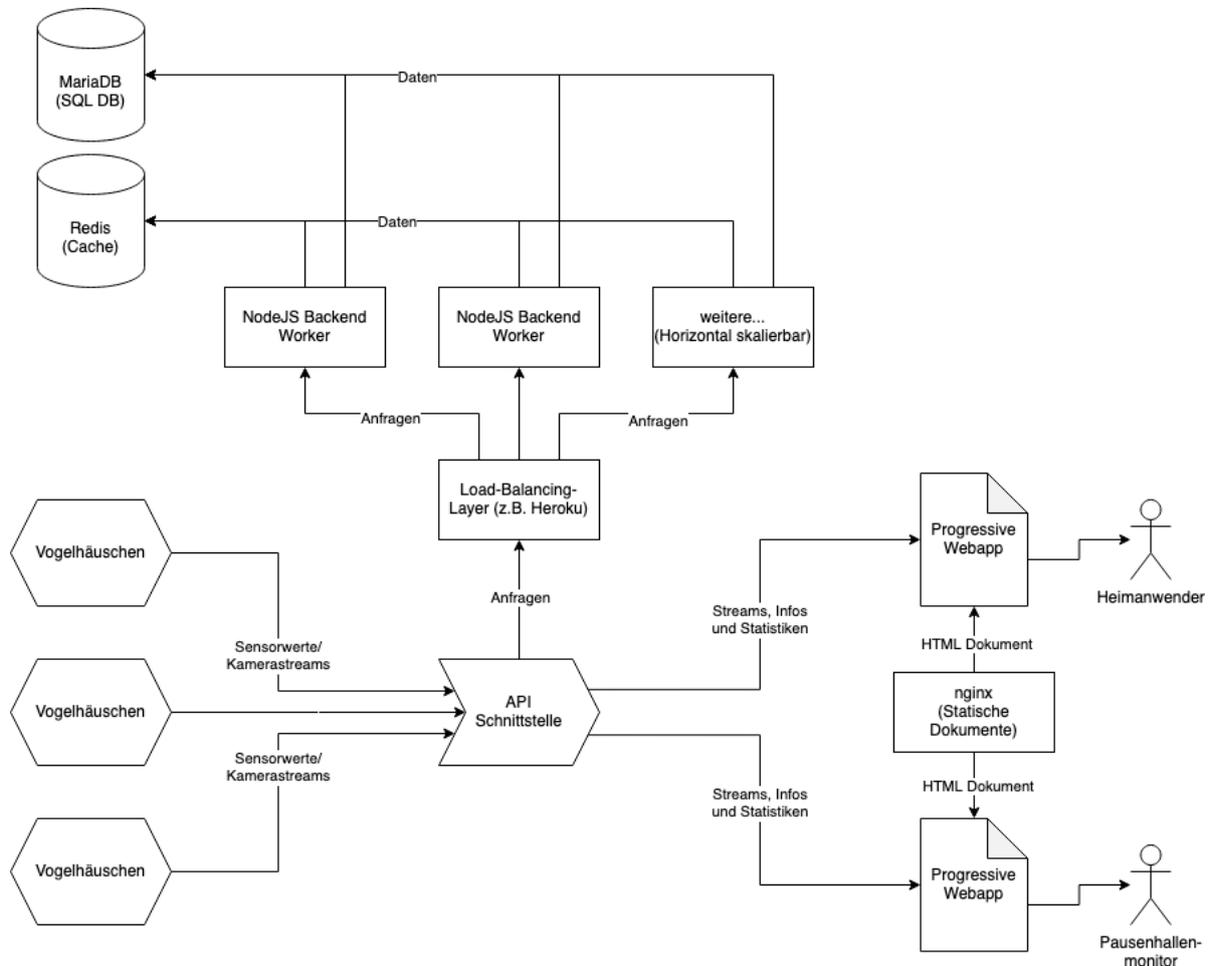


Abbildung 14: Architektur des Web-Services

Wie auf dem Diagramm zu erkennen, nutzen wir aktuelle Web-Technologien um das bestmögliche Nutzererlebnis zu ermöglichen. Nutzer bekommen Statische Dokumente von einem nginx-Server zugeschickt, welche im Browser Cache gesichert werden. Diese Dokumente, welche eine Vue 3 App enthalten, laden dann Daten über unsere API nach. Unsere API-Backend-Worker werden über einen Load-Balancer angesteuert und minimieren mithilfe von Redis-Caching die Auslastung der Datenbank, was einen kosteneffizienten und trotzdem schnellen Betrieb des Services ermöglicht.

Als Datenbank nutzen wir bewusst eine „traditionelle“ SQL-Datenbank, statt einer moderneren NoSQL-Datenbank wie z.B. MongoDB. Dies bietet nämlich die Möglichkeit direkte ausführen von Queries auf der Datenbank durch Umweltorganisationen zu erlauben, was über MongoDB zwar möglich, jedoch um einiges komplizierter wäre – je nachdem, was ausgewertet werden soll.

Sensorrauschen zu filtern wäre beispielsweise bei NoSQL-Datenbanken auf Server-Ebene unmöglich. Unsere Datenbank speichert nach folgendem Schema Daten:

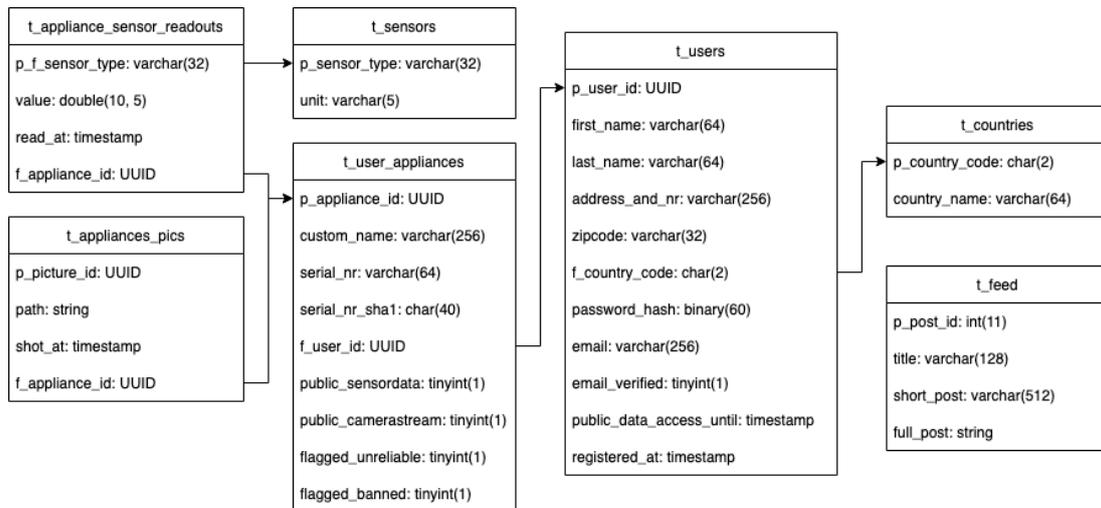


Abbildung 15: Datenbank-Schema

Diese werden dann von Workern genutzt um den Service zur Verfügung zu stellen oder können von Umweltorganisationen anonymisiert gelesen werden um Statistiken zu erstellen. Durch die Nutzung einer SQL-Datenbank lassen sich die Daten auch einfach für die Visualisierung auf Browserebene vorbereiten, was besonders in unserem Gebiet sinnvoll ist. Auch der Export als beispielsweise .CSV ist kein Problem. Damit wird es möglich, große Datenmengen über dafür vorgesehene Data-Science-Tools darzustellen und zu untersuchen:

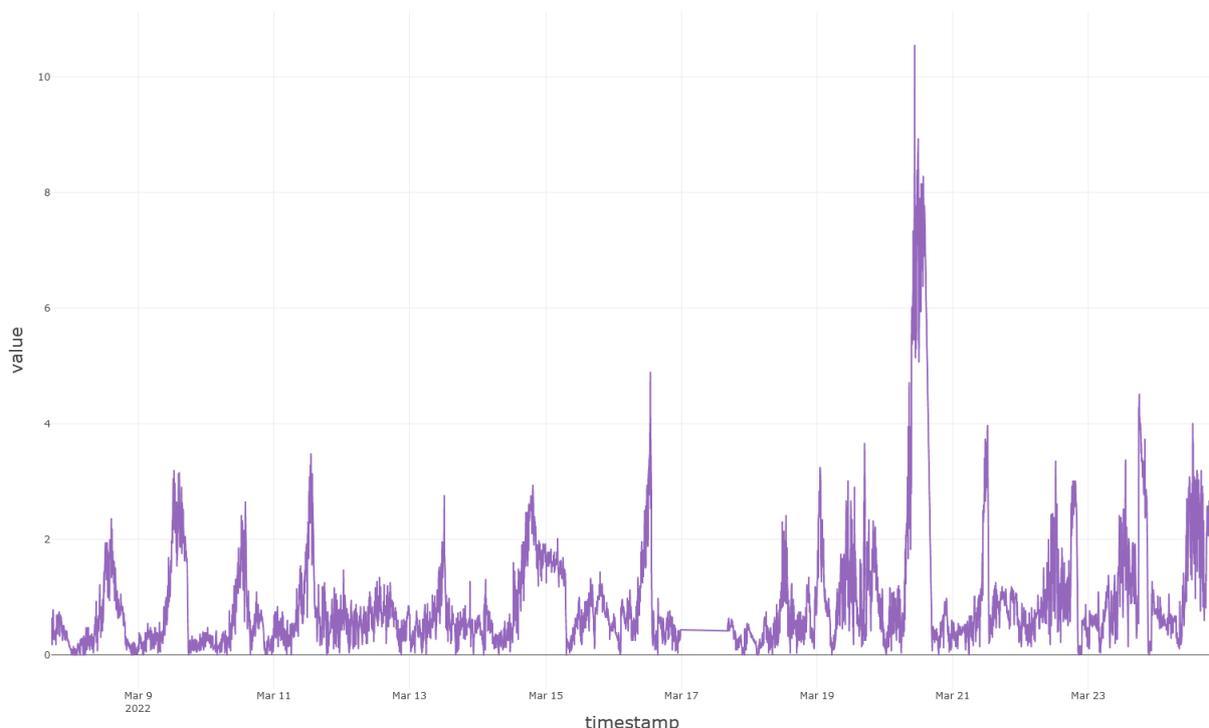


Abbildung 16: Mithilfe von CSVPlot erstelltes Diagramm der TVOC-Dichte (in ppb) nach Zeit

Die Datenvogelhäuschen DASH Webapp ist frontendseitig bislang noch in Entwicklung, was es unmöglich macht, dieses in diesem Bericht zu zeigen. Es ist noch nicht klar, ob das momentane Design final sein wird. Bis zur Vorstellung sollte sie allerdings fertig gestellt sein. Ihr Aufbau wird dem des lokalen Webinterfaces ähneln, mit der Erweiterung, dass es möglich sein wird, größere Datenmengen anzuzeigen und auszuwerten.

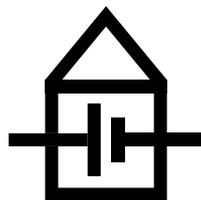
Die Webapp wird, sobald verfügbar, unter <https://dash.datenvogelhäuschen.de/> zu finden sein.

Unser Logo und unsere Design-Guidelines:

Neben dem Vogelschutz geht es bei uns auch um den Menschen. Deshalb sind jegliche Designs – darunter auch dieses Dokument – möglichst simpel gehalten um das Lesen mit Spezialhardware und die Nutzung von OCR-Software möglichst problemlos zu ermöglichen. Alle unsere Websites nutzen Bootstrap-Grid, was, als Webstandard, garantiert, dass Websites auch nicht-optisch mit richtigem Layout dargestellt werden können (beispielsweise durch Text-to-Speech Software).

Für grundlegende Barrierefreiheit bei nur leicht eingeschränkter Nutzungsfähigkeit unserer Websites sollen unsere Design-Guidelines helfen. Wir nutzen, wo möglich, Schwarz und Weiß als Kontrastfarben und setzen auf einfache Gestaltung von Logo und interaktiven Webkomponenten, wie Buttons.

Unser Logo in sich beschreibt dieses Prinzip perfekt:



Dicke, schwarze Balken bilden mithilfe von nur 2 Grundformen und 4 Linien ein einfach zu erkennendes Logo, welches immer noch als dieses – trotz Simplizität – erkannt werden kann. Nach diesem Prinzip entstehen jegliche unserer Designs.

Jeder Mensch sollte die Möglichkeit haben einen Unterschied zu machen – bei unserem Projekt durch Vogelschutz.