

Carbon-ML.org Tech Stack

Tech List

Frontend:

- **Gatsby** - It's an open-source framework that combines functionality from **React**, **GraphQL** and **Webpack** into a single tool for building static websites and apps. We use it for CarbonML website and example use cases for GS1 and Steel. Free.
- **Gatsby Cloud** - Is a cloud platform to deploy Gatsby websites. We use it for CarbonML website and other static sites we use as the GS1 and Steel use cases. Free for use cases, \$42.5 p/mo for the website.
- **Next.js** (React) - It's an open-source framework that allows to build server-rendered Javascript applications through the **React Library**. Free.
- **Typescript** - It's a free and open-source programming language super set of JavaScript that adds static typing to the code to improve security and development experience. We use it for the dashboard application, which uses Next.js.
- **Google Cloud Api** - It's a complete cloud infrastructure that allows to extend functionality on programming workflows. We use it for Maps and OAuth. Maps costs \$0.002 USD per each request when they are below 100,000.
- **Tailwind CSS** - Flexible Open Source CSS framework that provides utility-first CSS classes to compose custom components. Free.
- **Flowbite Pro** (Developer Edition) - An ecosystem built on top of Tailwind CSS that includes component library, block sections, templates and a Figma design system. A one-time payment of \$269.

Backend:

- **Node.js** - It's an open-source, cross-platform Javascript runtime environment. We use it for backend development, communication with databases and cloud services like Lambda functions. Free.
- **Postgres** - It's an advanced, enterprise class open source relational database that supports both SQL (relational) and JSON (non-relational) querying. Free.
- **AWS RDS** - It's a web service that makes it easier to set up, operate, and scale a relational databases in the Amazon Web Services. Used mainly as a UI interface to Postgres and we are connecting to RDS through Hasura.
- **Hasura** - Is a blazing-fast Open Source **GraphQL** server engine that gives instant, realtime GraphQL and REST Apis over databases such as Postgres or MySQL. It's Free.
- **Hasura Cloud** - Gives scalable, highly available, globally distributed, secure GraphQL Api over data sources. We're using AWS RDS configured with a special VPC Security Group permissions to connect securely to Hasura Cloud. Pricing that scales, free or \$99/mo for 20GB data pass-through.

DevOps:

- **GitHub Actions** - Makes it easy to automate software workflows and includes a **CI/CD**. We use it to automate deployments and notify repository updates. Pricing is included with GitHub.
- **Docker** - It's a suite of software development tools for creating, sharing and running individual containers. \$9 per user/mo on annual Team option.
- **AWS EKS** - It's a managed container service to run and scale **Kubernetes** applications in the cloud or on-premises. \$0.10 per hour per cluster.

Design:

- **QR Code Generator** - It's an online service to generate QR codes through a simple user interface. \$15.99 p/mo for teams billed annually.
- **Figma** - Collaborative web application to design user interfaces to review and deliver better products. \$45 per editor/mo.
- **123RF** - It's a stock photo agency selling cheap stock photos on demand. 10 images for \$45 in pack.

Project Management:

- **GitHub** - It's a hosting service for software development and version control using Git. We use it for our Open Source projects, as well for private working and prototyping. \$44 per user/year for the first 12 months.
- **Prismic** - Headless CMS used to generate dynamic content required on multiple ends. We use it to store the content data that Gatsby uses to generate the static sites.
- **ClickUp** - Web application that provides a service to plan, track and management projects of any kind. \$5 per member/mo.
- **Slack** - It's an instant message program to communicate and share files with the team. 7.25 USD/mo per active user in the Pro version.
- **Google Docs** - It's an online text editor used to create and collaborate text documents online. We use it mainly for the White Paper. Free.
- **Microsoft 365** - It's a product ecosystem of collaboration and cloud based services made for teams. We use it through GoDaddy for email, calendar and hosting.
- **1Password** - It's a password manager and secure digital wallet. We use it store all important passwords, keys and credentials for the services that CarbonML uses.
- **Userback** - It's a customer feedback software for SaaS companies to collect product insights and scale outstanding customer experiences. \$59 p/mo for 10 users.

Other Options:

- **AWS Elastic Beanstalk** - As alternative for EKS, we might use Elastic Beanstalk, which is an orchestration service offered by Amazon Web Services for deploying applications which orchestrates various AWS services, including **EC2**, **S3**, **SNS** (Simple Notification Service), **CloudWatch**, autoscaling, and **Elastic Load Balancers**.

To Be Retired:

- **AWS Amplify** - Provides a simple, fast, and modern UI-driven approach to building mobile and web applications. It was useful to create prototypes, as the PoC, but as the project grows we'll migrate from it to a more robust and flexible architecture. Free.
- **Wordpress** - It's an open-source content management system (CMS) that simplifies the creation and management of simple websites.
- **MUI**

Architecture & Security

1. On Frontend we use **Next.JS** with **Typescript**, in a way to provide confidence on the security and adaptability on the project.
2. We use **Hasura Engine** as a bridge to communicate to our database from **GraphQL** and **RESTful** operations.
3. We use **Hasura Cloud** as the service to host our **Hasura Engine**.
4. We're using **AWS RDS** configured with a special **VPC Security Group** permissions to connect securely to **Hasura Cloud**.
5. For initial tables at our database we have accounts, sessions, users and verification_tokens. All of them related to Authentication and Session.
6. To authenticate, we use **Next-Auth** on Frontend to create a better consistency around **Next** ecosystem. **Next-Auth** has multiple service providers to help users login as they want to.
7. For secure requests we use a combination of **JWT** (JSON Web Tokens), a **Hasura** secret and a **Next-Auth** secret. Each provider used uses its own secret as well (like Google, Facebook, GitHub, etc).
8. **JWT** gets encoded and decoded with a **HS256** algorithm as strategy to communicate with **Hasura**.
9. On the auth callbacks we send to **Hasura** additional variables like allowed roles, roles and user ids if needed following the **JWT** strategy.
10. Each **API endpoint** can be created manually with **Next.JS**, which therefor uses the **GraphQL** queries to retrieve or mutate data at the databases. This simplifies the use of resources at multiple external services.

Microservices (for reference):

- Hasura Engine - APIs (RESTful and GraphQL endpoints)
- Hasura Cloud
- AWS RDS
- AWS EKS
- AWS EC2
- AWS S3
- AWS SNS
- AWS CloudWatch
- AWS ELB
- AWS Lambda
- Google OAuth 2.0
- NPM
- Next Auth
- Nginx
- GitHub Actions (some)
- Prismic