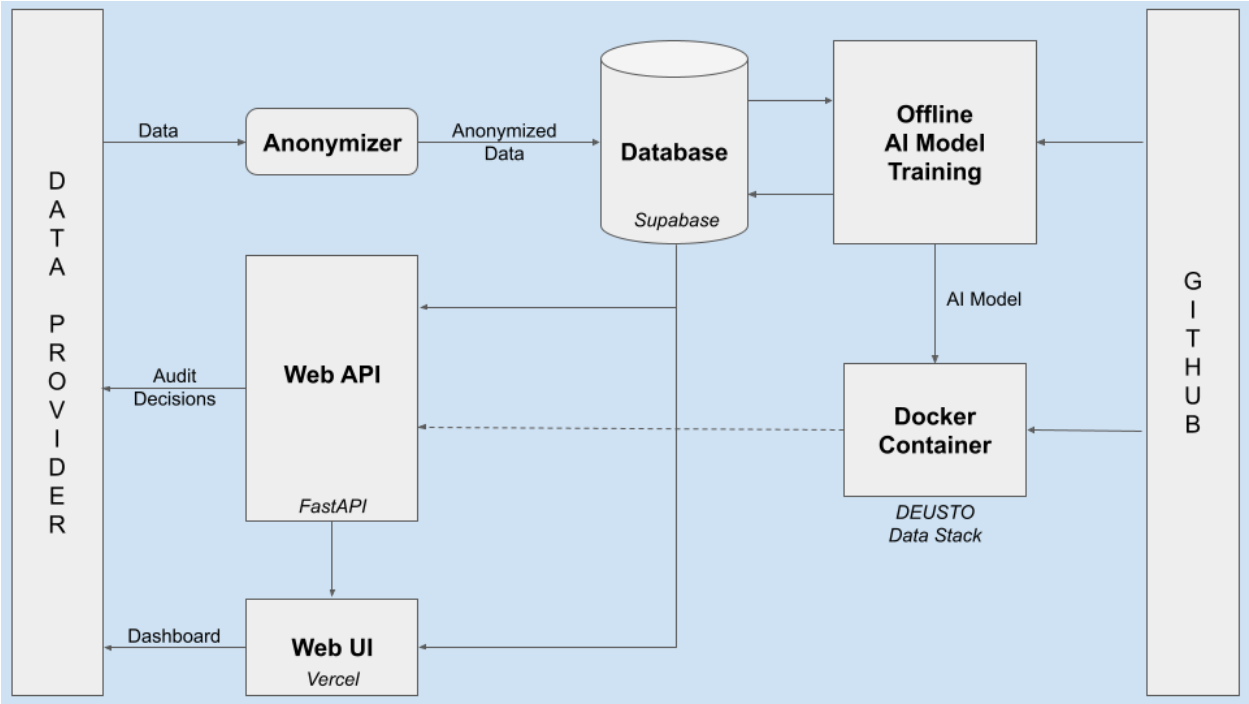
CartGuard EXPERIMENT Phase Technical Specification
20.09.2023

Technical Scope

Our solution **minimizes the fraud in physical retail with self checkout processes** by optimizing the audit decisions. We train an AI model using the consumer, store, cart, transaction and physical cart audit data and provide the **optimized audit decisions** through an API and **analytics** and **admin tools** through a dashboard.



Initially, the whole data is anonymized by using **Anonymizer** from the REACH Toolbox and then stored in our **Postgres** database provided by **Supabase**. On a weekly basis, an **AI model** is trained over this data in an offline manner and the trained AI model is stored. After each training or when a new **GitHub** commit occurs, the deployment process of the application is triggered. A **RESTful API** powered by **FastAPI** is built into a **Docker** container with the previously saved AI model and then deployed to serve as a real-time **Web API** that can be used directly by the data provider for their audit decision queries. Additionally, we also provide a **Web UI** that has a **dashboard** through a graphical interface deployed on **Vercel**.

With this architecture, our solution not only **perfectly tackles the challenge** of the data provider by serving them audit decisions through an API but also provides a dashboard to see the **analytics** about the audits and **an admin platform** to manage them.

Algorithms and Tools

First of all, we performed a thorough **exploratory data analysis** and **data preparation** on the final data set provided by the data provider. Later, we generated **engineered features** for consumers, stores and products to reveal hidden dynamics of the data. Also, due to the imbalanced nature of the data, we employed a **data augmentation** scheme using **SMOTE** to

increase the chance of discovering fraudulent instances. Finally, we introduced **new metrics** like total fraud detected, unnecessary audit cost etc. and by using them we created a **customized objective function** instead of only using accuracy in order to capture the essence of the real-world problem.

Then, we trained a **LightGBM** algorithm with **hyperparameter optimization** and a **Deep Learning** algorithm with **model selection**. Conclusively, we used an **additive ensemble** of these algorithms to match the audit frequency request of the data provider.

Scalability and Flexibility

As we employ an **offline AI model training** scheme, the real-time performance of our solution is not affected by the size of the data. Additionally, as our solution is built into a **Docker** container, it can be run in any environment that can run containerized apps and with the **Kubernetes** technology it can be **scaled up automatically** on demand.

Our approach is adaptable to different data providers as we employ a **data configuration structure** to implement the inputs and outputs of our AI model training scheme. This also enables us to even change the internal AI model without disturbing the whole process.

During the EXPERIMENT Phase, we have deployed our solution to **DEUSTO Data Stack** from the REACH Toolbox but our solution is **cloud-agnostic** and can later be deployed to any cloud environment.

We serve our solution through an API, hence it **can easily be integrated** into any related **Data Value Chain**. Actually, we had already partnered up with **Rovimatica** and **Phasmatic** to integrate their objection detection solution in the physical retail stores during the EXPLORE Phase. In the EXPERIMENT Phase, we extended these collaborations with **Amplify Analytix** to integrate their retail sales forecasting algorithm into our solution in order to increase our accuracy in fraud detection.

Data Governance and Legal Compliance

Our solution does not collect any personal information such as name, email, phone number etc. and it is compatible with **GDPR**. Additionally, we also anonymize all the user-related data from the data provider by using **Anonymizer** from REACH Toolbox before storing in our database system in order to further provide **k-anonymity** among the anonymized users.

Finally, we deploy our solution on an **enterprise cloud platform** where the security is guaranteed by the service provider and our access is realized with **2-factor authentication**. We then serve our solution through a secure Web API with **HTTPS** protocol using **TLS certificates**. Furthermore, we also have **Row Level Security** and **JWT authentication** in our database that only allows users to access the data that they are authenticated for.

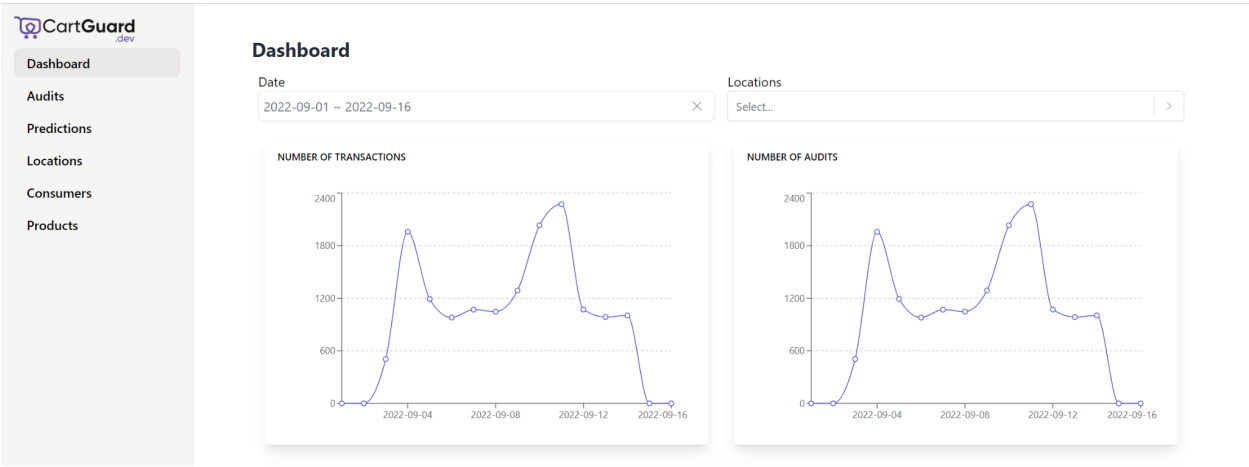
Quality Assurance and Risk Management

The quality assurance and risk management are covered by well established processes:
Project management: PMI standards with certified PMP program manager
Data quality management: Exploratory data analysis, data cleaning and data validation.
Development process management: Version control with GitHub, unit and integration tests, CI/CD, automated deployment.
Algorithm precision management: Daily training of multiple candidate algorithms.
Environment management: Development, staging and production environments.

Annex 1. Means for accessing the MVP

Dashboard

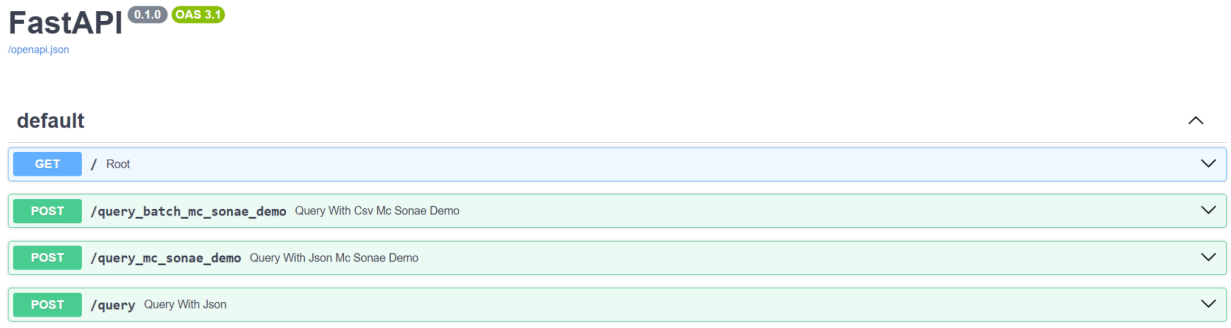
URL: https://app.cartguard.dev/
Email: orhan@cartguard.dev
Password: Reach1234



Due to the NDA signed between the Data Provider MC Sonae and CartGuard, only the publicly available sample data of the corresponding REACH Challenge is accessible through the dashboard to review. Thus, be aware that the data is only available for 09.2022 and with some missing features.

Prediction API

The automatically generated documentation page provided by FastAPI from where you can also generate API calls, can be accessed through the following link:
https://cartguard.apps.deustotech.eu/docs/



Also, here is an example Post request to access the API using Curl:

```
curl -X 'POST' \
  'http://cartguard-ai.apps.deustotech.eu/query_batch_mc_sonae_demo' \
  -H 'accept: application/json' \
  -H 'Content-Type: multipart/form-data' \
  -F 'file=@audits_sample.csv;type=text/csv'
```