Team Ratula

Members

Maluki Muthusi Maluki malukimuthusi@gmail.com

FrankLine Bosire frankbosire2017@gmail.com

Some Important Links

- Our app is at https://equity.riviatechs.com
- Our API server is at https://mt940-server-s47opgtmgq-uc.a.run.app The domain name will be https://server.equity.riviatechs.com
- Server code is at https://github.com/riviatechs/mt940_server
- App code is at https://github.com/riviatechs/Adjustable-Widget-Equity-Hackathon
- Link to this report https://github.com/riviatechs/equity-report

Background Information

Many Corporates receive a lot of daily transactions through their accounts. The people managing those accounts find themselves many times trying to search the transactions to find certain transactions when need arises.

Sending the normal MT940 statements to these clients, does not fulfill their special use cases.

They want to get an updated view of their transactions at any time. They might choose ask for some specific data in the history of their transactions.

Problem Statement II, Configurable MT940 Statements

MT940 format is an electronic bank statement file used by the SWIFT network to send and receive end-of-day bank account statements and transactions reporting.

The bank can send MT940 statements to its users for the transactions that have happened.

The bank wants to have a way to send only statements that are of interest and relevance to the users.

Objectives

1. Develop a configurable widget to download MT940s in various formats e.g xls, pdf, or free formats that allows selection and formatting of columns(column layout, content, etc)
2. A widget which the bank would then consume from within equity platform via API calls.
3. Users to be able to select the columns and layout from the raw format via a simple UI
4. Easily download the selected MT940 data and corresponding columns in various file types

Justification of Solution

1. The solution will offer flexibility to the users when requesting data from the bank
2. Through our solution equity bank users will be able to get statements of their transactions in the formats they want.
3. They can select what they want to be included in the statements reports and also be capable of filtering the statements to what their want.
4. It makes it easy to keep track of their financial transactions.
5. By offering users to select what they want, it helps to keep the customers happy, and keep using the banks online.
6. Our solution has the look and feel of the Equity products and it is easy to integrate with it.

Benefits

1. Improve on customer experience with the product.
2. This solution can be used with Equity's various products, like mobiles apps, web apps and even USSD
3. This solution helps to support users with low bandwidth, by sending only the exact data the users want
4. Corporate users can use this feature to confirm their transactions up to nuisance granularity
5. This feature helps to implement other features that can be different to implement when using a different architecture

Detailed Description of the solution

We have developed a configurable widget that can support user queries up to the very detailed granularity of what they want.

At the core of our widget is a query schema, which have built using open source technologies, Golang and Graphql.

Users can select to include or not any include the following fields:-

1. Date of the transaction
2. Transaction Reference number
3. Transaction type credit/debit
4. Account Number in which money was debited or credited to
5. Account Name of the account in which money was credited or debited from
6. Amount that was credited or debited
7. Narrative, which explains the transaction was about, for example reversal, pay electricity bill e.t.c

Users can select to filter the transactions with combinations of the following criteria

1. Amount Range, specify to get transaction with certain range
2. Exact Amount, Specify to get transactions that equal a given amount
3. Period, Specify to get transactions that happened during a particular period
4. Credit/Debit, Specify to get transactions of the type money in or money out to your account, or get both type of transactions

5. Currency, specify transactions of a particular currency
6. Download, Specify to download the transactions and get a file format of either pdf, csv or xls

Example of a pdf report generated by the user

EQUITY

Statements History

Opening Balance	200000.00
Closing Balance	20000000.00

Account Number	01234567891	**Date From**	04/12/2021
Account Name	Maluki Muthusi	**Date To**	04/02/2022

Transaction No.	Date	Account Number	Account Name	Credit/Debit	Amount
554528	1/2/2022	01879115772	Lindsay Gotts	Debit	85074850
370681	1/2/2022	01435844452	Aubrey Squires	Credit	42778240
554154	30/1/2022	01275894337	Brandy Watson	Credit	53130704
912284	30/1/2022	01768211854	Raymond Cruse	Debit	56759740
31328	29/1/2022	01268572624	Peggy Benton	Debit	33214874
604342	29/1/2022	01836073415	Murray Fitzsimons	Credit	43184010
731797	28/1/2022	01814424579	Nora Hill	Credit	43363852

Example of a CSV/XLS report generated by the user

F4 Credit

	A	B	C	D	E	F
1	Date	Transaction REF	Account Name	Account Number	Amount	Debit/Credit
2	2/1/2022	554528	Lindsay Gotts	1879115772	85074850	Debit
3	2/1/2022	370681	Aubrey Squires	1435844452	42778240	Credit
4	1/30/2022	554154	Brandy Watson	1275894337	53130704	Credit
5	1/30/2022	912284	Raymond Cruse	1768211854	56759740	Debit
6	1/29/2022	31328	Peggy Benton	1268572624	33214874	Debit
7	1/29/2022	604342	Murray Fitzsimons	1836073415	43184010	Credit
8	1/28/2022	731797	Nora Hill	1814424579	43363852	Credit
9	1/27/2022	631408	Leanne Harold	1226272715	43247052	Debit
10	1/27/2022	34533	Deanne Hart	1317458738	28615230	Debit
11	1/27/2022	844283	Lucia Carrington	1102031458	92097140	Credit
12	1/26/2022	457780	Alec Thring	1679314634	73562390	Credit
13	1/25/2022	97485	Curtis Abbot	1031600616	63886260	Credit
14	1/25/2022	311568	Joanne Fisher	1875816402	22163404	Debit
15	1/25/2022	506650	Pearl Aikin	1715892824	95228970	Credit
16	1/25/2022	274679	Melissa Matterson	1845077655	25315600	Debit
17	1/23/2022	466357	Clara Bedser	1533615843	61263244	Debit
18	1/22/2022	73907	Hugh Dane	1617750358	85264100	Debit
19	1/21/2022	556284	Johnson Smith	1059628014	88811490	Credit
20	1/21/2022	921541	Maximilian Begley	1003117390	46308150	Debit
21	1/20/2022	23529	Melvin Peverett	1383277862	73731470	Debit
22	1/20/2022	660530	Nicholas Kirkland	1792473738	6181664	Debit
23	1/19/2022	654592	Corrie Hayes	1013879121	7521701.5	Debit
24	1/19/2022	186503	Wayne Wattis	1533848812	79257800	Debit
25	1/19/2022	630709	Arthur Ashley-Cooper	1452127251	90636700	Credit
26	1/19/2022	512183	Linda Laughton	1385472492	9992493	Debit
27	1/18/2022	945343	Herbert Axtell	1480602428	82576580	Credit
28	1/17/2022	755145	Marilyn Firestone	1637172091	46804868	Debit
29	1/17/2022	81347	Eve Dobb	1782236467	31231082	Credit
30	1/16/2022	44131	Frank Deeks	1804112488	5825406	Credit

2e4c5a1c-f42c-4f45-90e8-27b8840

Ready

Assumptions and Limitations

1. We assumed users already have equity bank accounts and they are authenticated through Equity's authentication schemes.

2. We are using test data to simulate a corporate account that receives a lot of transactions.

Conclusion

We have designed and implemented a solution that caters for the requested needs

1. We have implemented a system that enables users to select the fields they want and apply filters of their choice and request to download the data in various formats for their needs. This meets objective 1 of the problem.

2. We have provided an API which the bank can consume in its various products, mobile apps, web apps, USSD e.t.c. We have built the API using open standards, REST and Graphql.

3. Through the system users can select the fields they want and apply filters to view specif data.

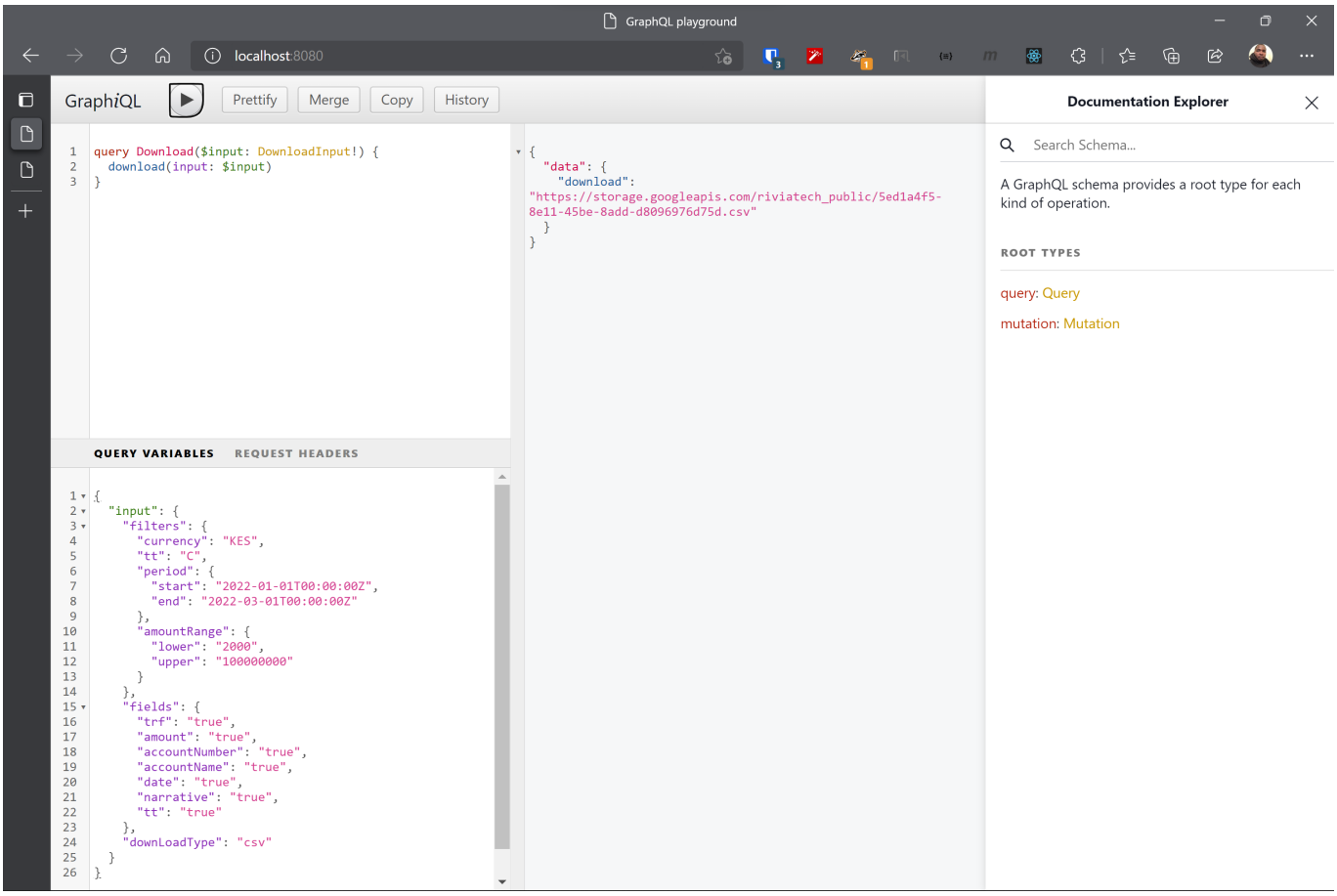
4. Our solution offers a means to easily download data various formats, PDF/CSV and XLS.

Technical Documentation

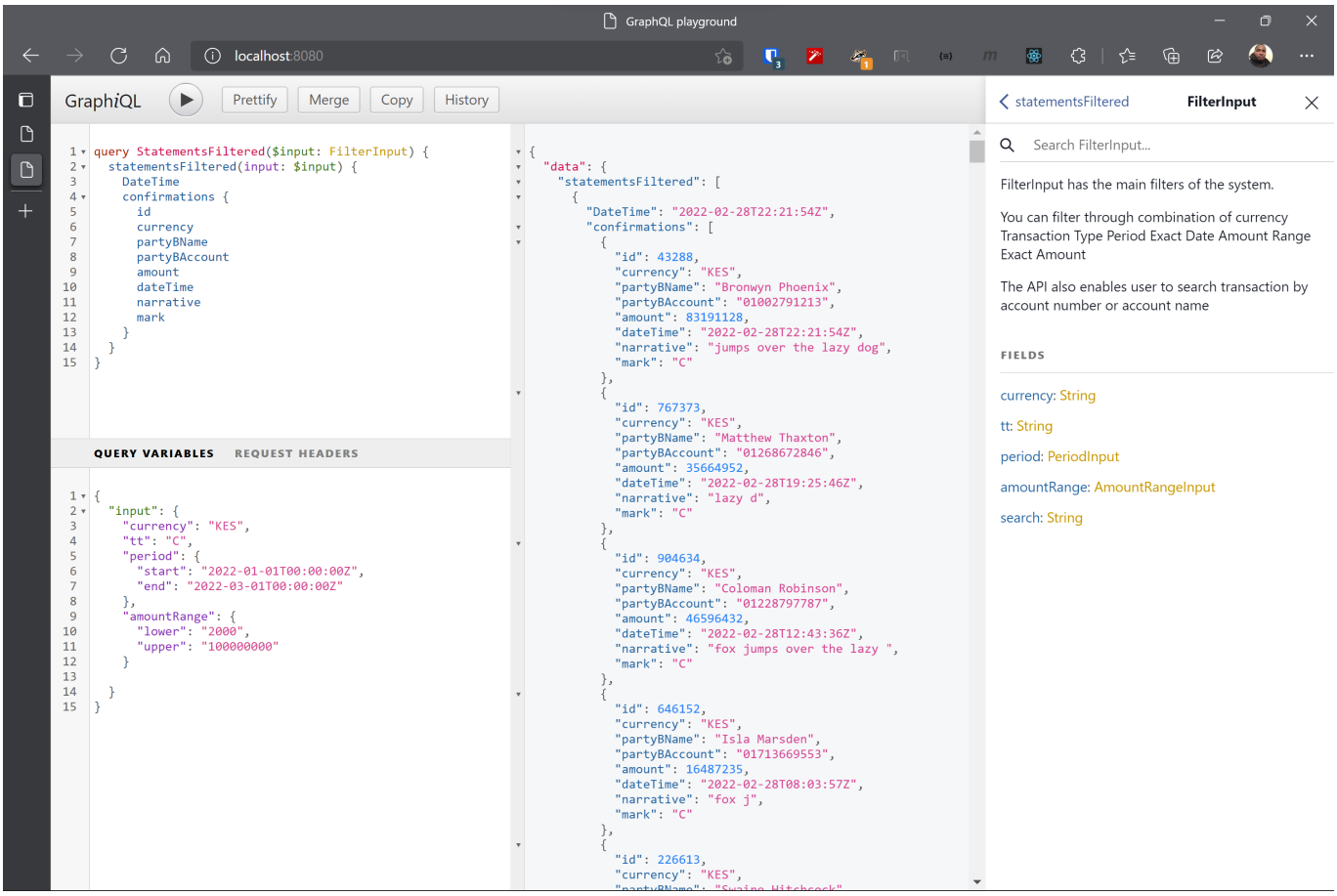
API Level

At the very high level, we have build an API that offers flexibility in what you want.

For example, you want to view all the transaction that happened between 01/01/2019 to 02/02/2020 and for those transactions, you only want those that were of type credit to your account. You also want to get only the date and Amount and dowload that data in excel format or a pdf report.

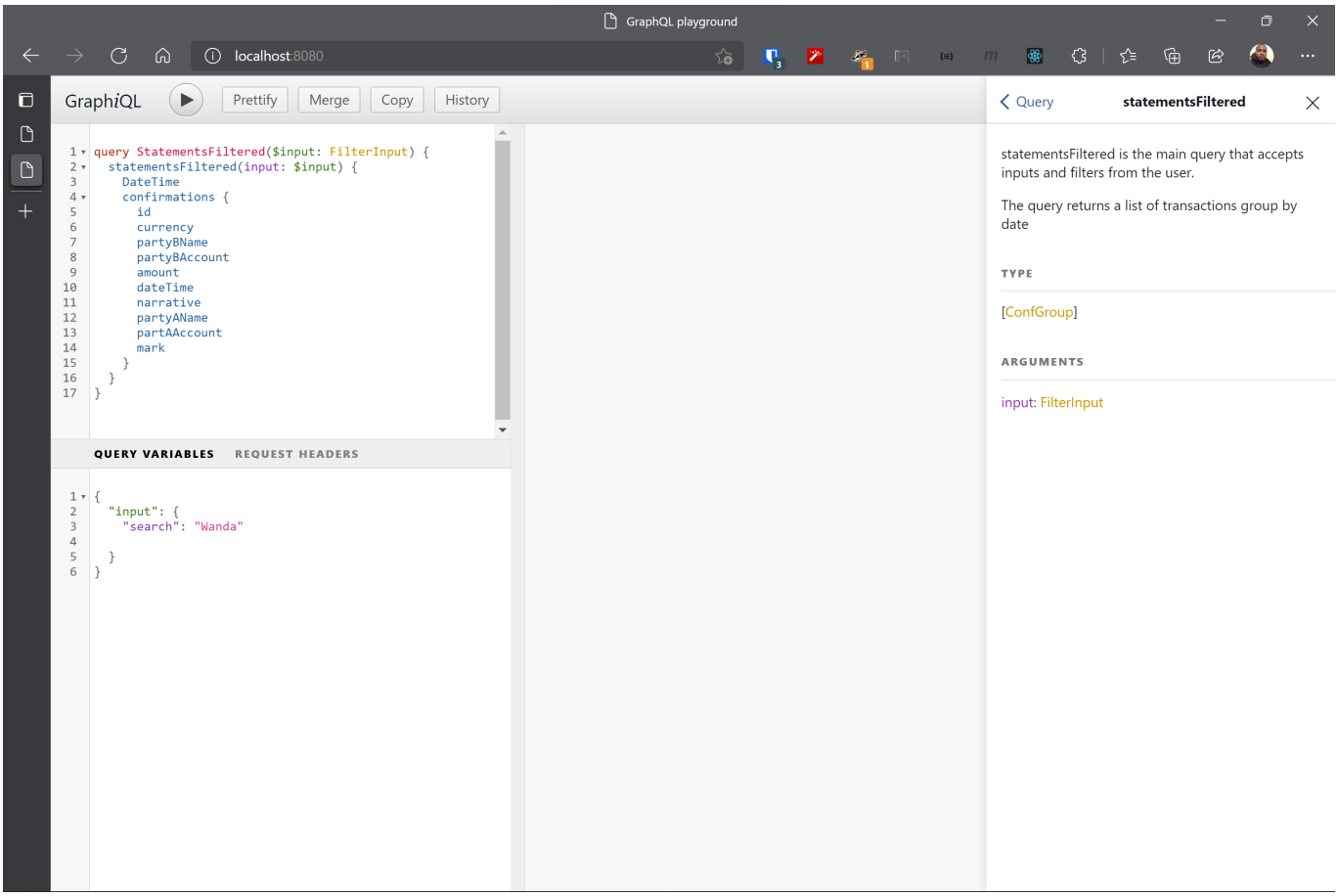


Our query system supports those kind of use cases and other complicated ones.



We have been able to achieve these through the use of data graph solution. We have build our API schema using graphql, an open source query language.

User can download pdf or csv or xls document of the fields that they have indicated they want, with filters they have specified.

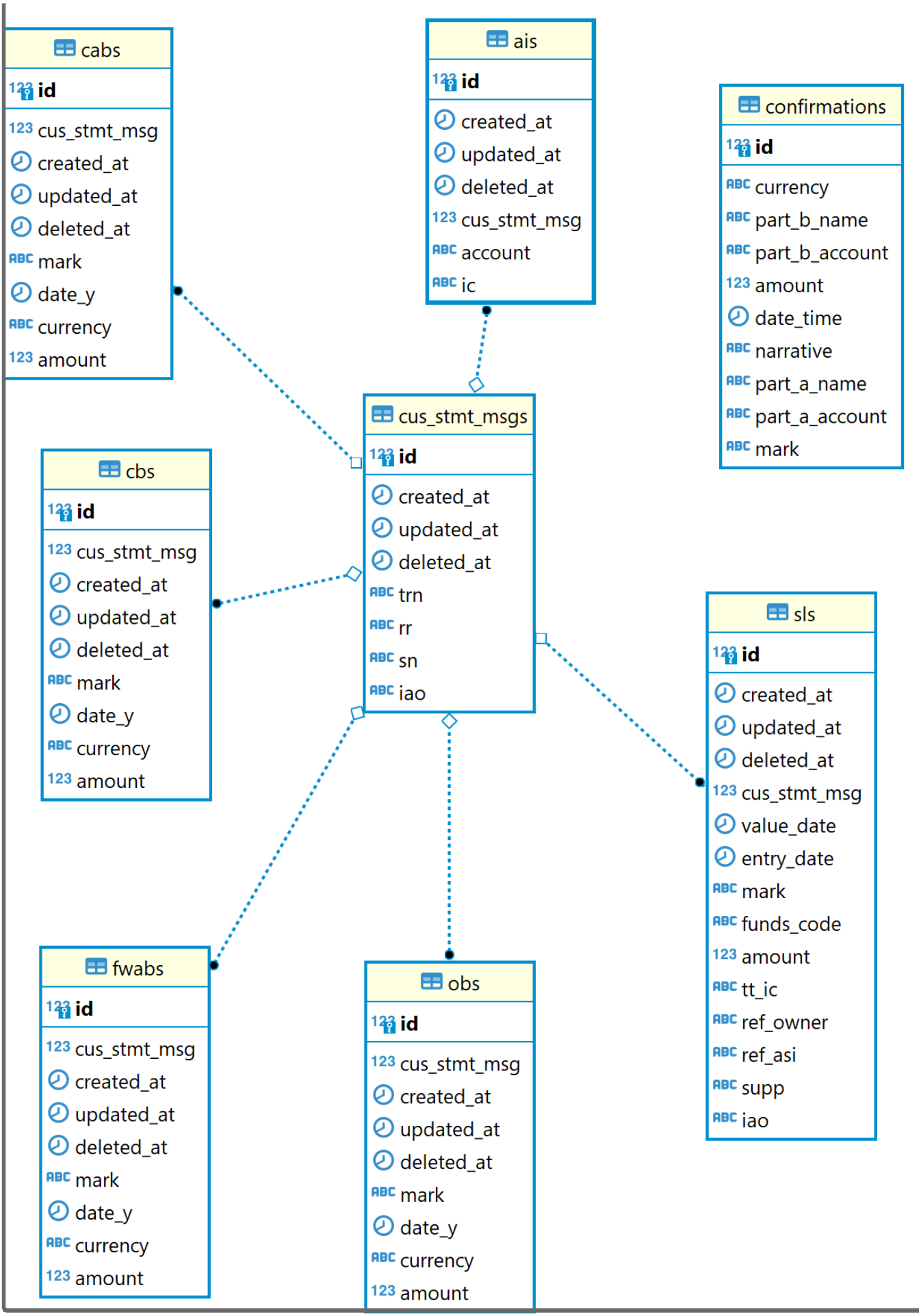


Low level

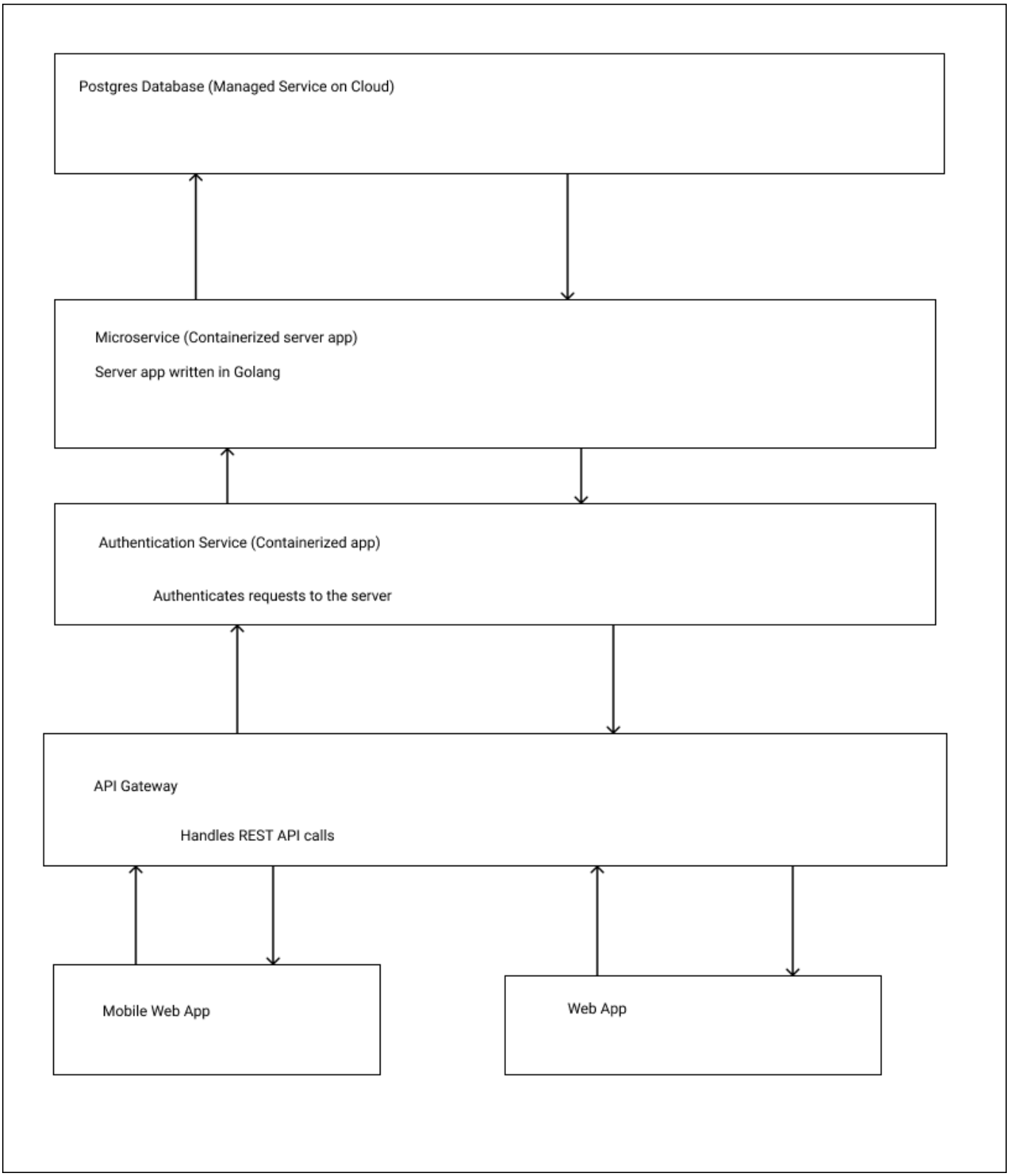
We have a parser that we can use to receive MT900 (Confirmation of Debit) and MT910 (Confirmation of Credit) and MT940 (Customer Statement Message).

We then parse that data and save it into our database

ERD



Architecture



Programming Languages and Tools

- The mobile and web app is built using ReactJS. In particular we are using https://nextjs.org, because it offers server side rendering (SSR) and Client Side Rendering (CSR)
- The API server is build using Golang https://go.dev
- For authentication we have deployed our own Single Sign On (SSO) service, using and open source solution https://casdoor.org

- We have built our API using graphql query language. Graphql is an open source solution that uses REST APIs.
- We are using docker to containerize our applications so that they can be deployed on any cloud service provider.
- We have deployed our containers to Cloud Run, equivalent to AZURE Container Instances.
- We are using continuos deployment of our code. Each git pull triggers a new build of the code. If the build is successful the code is deployed to the staging environment.
- We test the app on our devices and then trigger a build to our production environment.
- We have used postgres for storage of data.
- We are using a loosely coupled architecture and each layer can be scaled independently to meet the number of requests.

Solution Access

- Our app is at https://equity.riviatechs.com
- Our API server is at https://mt940-server-s47opgtmgq-uc.a.run.app The domain name will be https://server.equity.riviatechs.com
- Server code is at https://github.com/riviatechs/mt940_server
- App code is at https://github.com/riviatechs/Adjustable-Widget-Equity-Hackathon
- Link to this report https://github.com/riviatechs/equity-report