Blockchain

For Enterprises

Completed by: Ishan Singh Gill (203444P)

Quick Recap for our Project

- Storing Clients Information on the Private Ethereum Blockchain

- Retrieve Clients Information automatically from the Policy Smart Contract

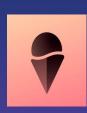- Schedule the sending of Premium Payment Reminders to multiple Clients

Quick Recap for Sprint 1 and 2

- Set up Besu Blockchain

- Created the Smart Contract Factory to deploy the Policy Smart Contract for the Client

- Deployed an Automated Smart Contract that emits an event every 25 seconds

Centralized Database

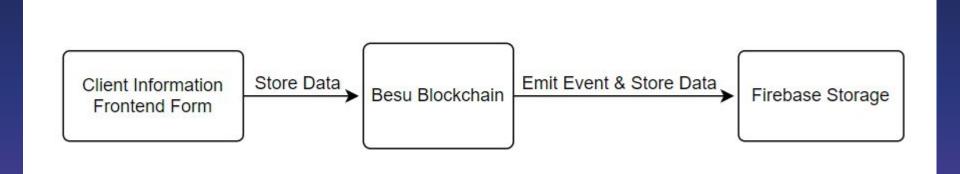- Store the Client's Smart Contract Address (document id)

- Within a Collection with the format DDMM of the Client's Start Date of Policy

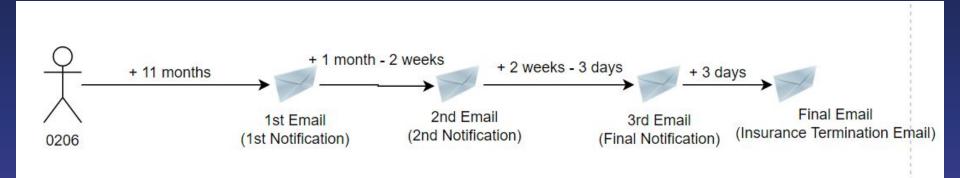- Retrieve the Smart Contract Addresses from the specific collection based of the current date

How do know when to Store into Firebase?

For Storing:
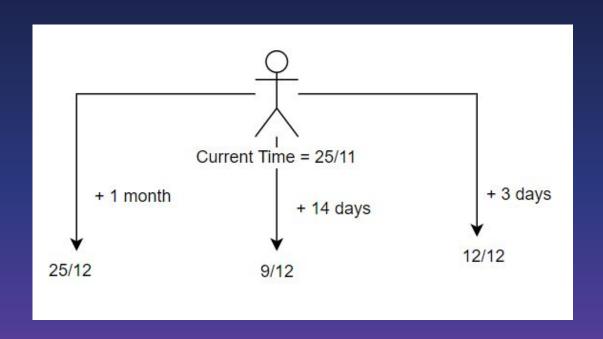
How do know when to Retrieve from Firebase?

Notification Process:
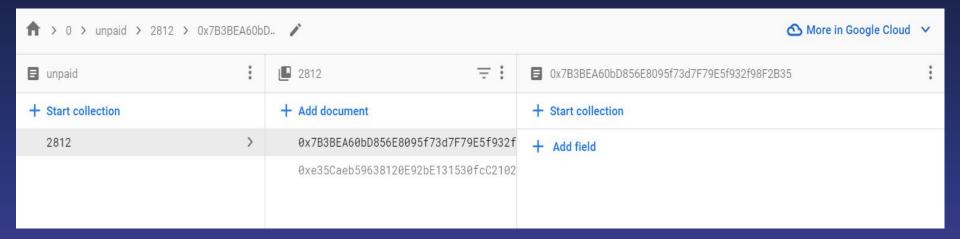
How do know when to Retrieve from Firebase?

Strategy:

Database Structure

☁️ More in Google Cloud ⌄

📄 unpaid	⋮	📖 2812	⚏ ⋮	📄 0x7B3BEA60bD856E8095f73d7F79E5f932f98F2B35	⋮
➕ **Start collection**		➕ **Add document**		➕ **Start collection**	
2812	›	0x7B3BEA60bD856E8095f73d7F79E5f932f		➕ **Add field**	
		0xe35Caeb59638120E92bE131530fcC2102			

Benefits of such a Database Structure?

- No Paid Firebase Functions/Extensions needed
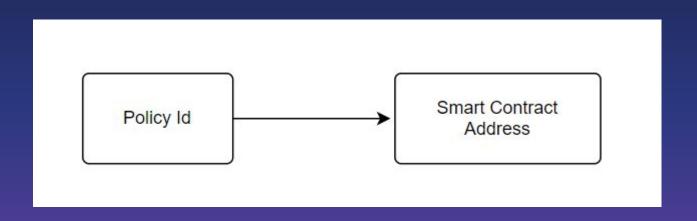
- No Queries/Filters Needed

- Less Documents to read and the backend is able to update on the fly

How do we identify the Client's Policy Smart Contract Address?

Create a Mapping Smart Contract which maps:

Add a Payment UI for the Client
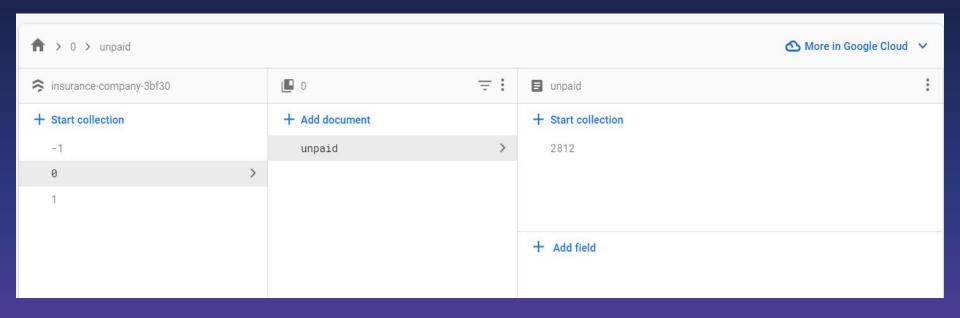
- Client is prompted to input their Policy Id

- Retrieve Clients Policy Price based of their Smart Contract Address and prompt Client to make Payment
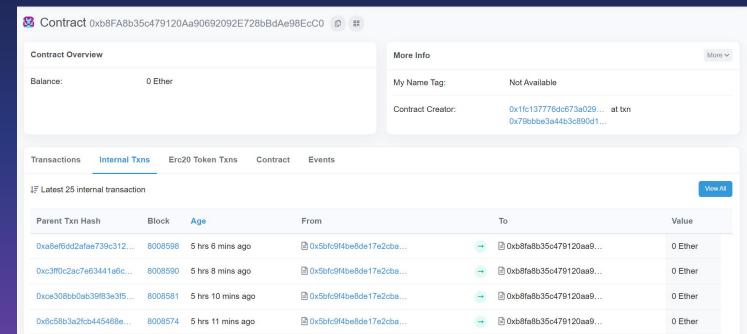
- Upon Payment, Fires an Event to the rpa code to update the payment status for the specific client based of their date of policy

Modify the Database Structure
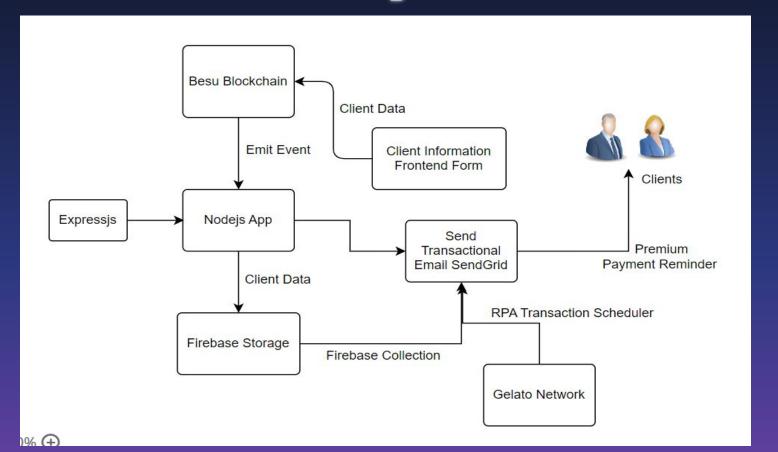
Integrate with the Gelato Subscription

Each Event Subscription fires automatically every 25 seconds, hence we can input our backend code into the subscription. (including hardhat, date objects and firebase code)
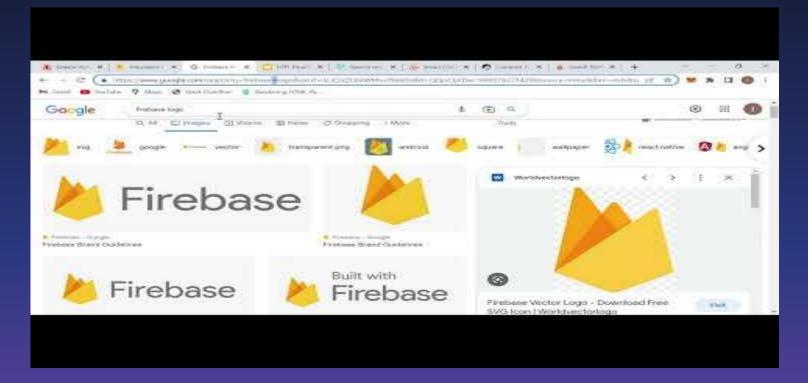
Overview of the Project

Demo (rpaDaily.js)

Demo (rpaDaily.js)

Suggestions & Ideas on how to continue on with the Project

- Integrate Node.js Backend with React Frontend

- Send SMS Notification to the Client via Twilo and the Clients Phone Number

- Improve on the UI and Form Inputs with React

React.js + Node.js

Thank You!