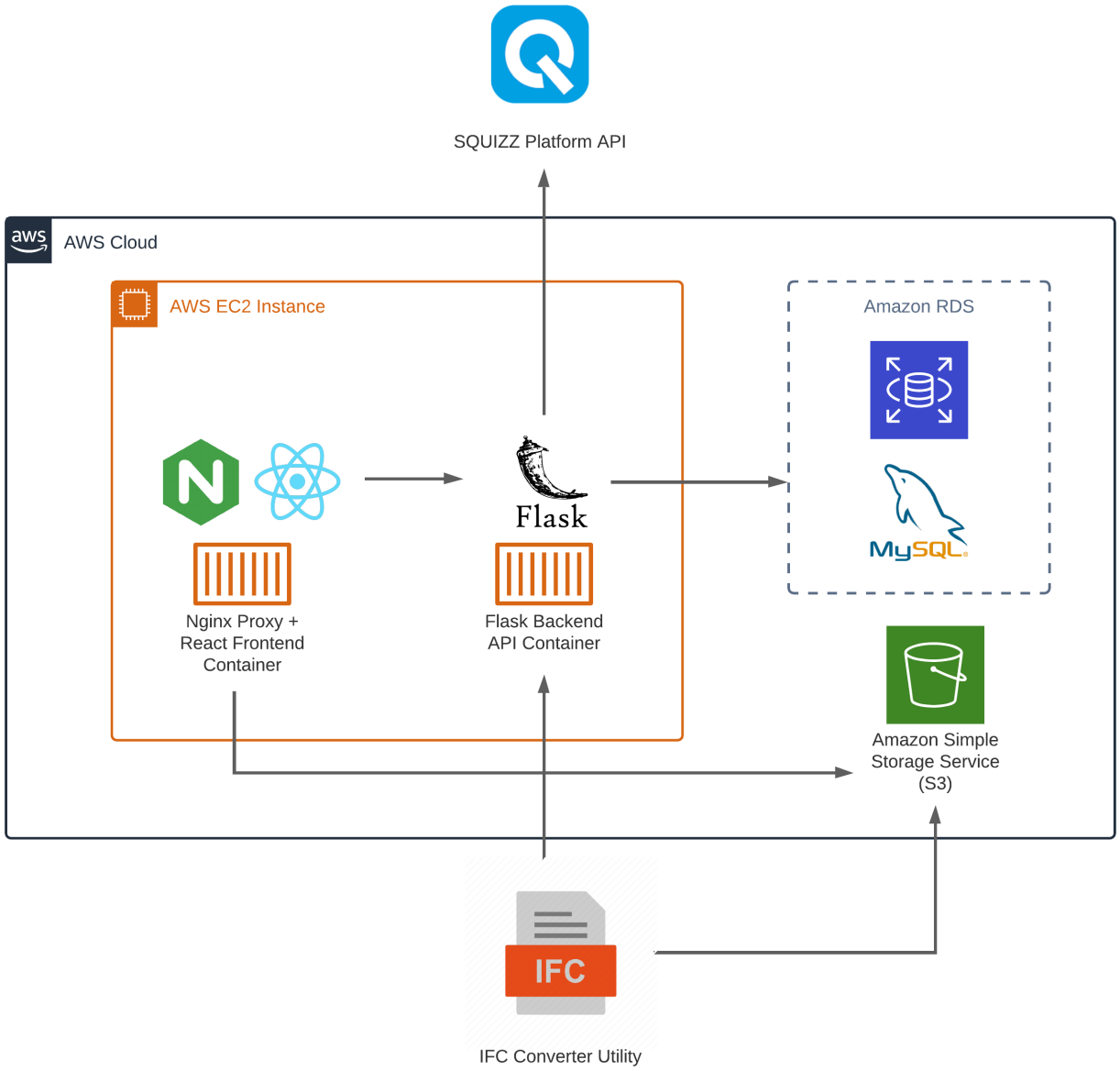
# Architecture Diagram

## Versions:

| Version ID | Date | Changes |
|---|---|---|
| 1.0 | 23/09/2020 | Initialized the document with the architecture diagram |
| 1.1 | 26/11/2020 | Updated the High-Level Architecture |

## High Level Architecture




### Frontend

The frontend uses React.js as our library of choice, in addition to Webpack as a module bundler, and React Ant Design as a UI component library.

The latter allows both teams to not waste time writing vanilla HTML/CSS to create UI elements, and instead focus on completing the key features required by the client.

## Backend

The backend uses Flask as a micro-framework to establish a RESTful web API that allows the frontend client to interact with it.

The Flask backend interacts with a MySQL database to retrieve products and relevant data (including prices and image URIs etc) and return them to to our React frontend to display the data.

## Database

The MySQL database stores all business and Enterprise Resource Planning related data, including products, product prices, and category trees.

It also stores organisations, users, and credentials.

## SQUIZZ Platform API

The SQUIZZ API enables our web application backend to retrieve organisation data, and subsequently import it into our database.

It also allows us to procure purchases for products and submit them within the SQUIZZ platform.

## IFC to 3D Image Utility

This converter utility is developed using C# and .Net Framework 4.7 and is responsible for batch conversion of IFC files to 3D model images and converting the text files that contain the model parameter data into the JSON file.
The Utility is also responsible for sending this data to the Server and upload the Models to the S3 Bucket on the AWS.