0.1 Architecture Overview

The infrastructure was designed to follow a *Service Orientated Architecture* where each of the components were responsible for a different area of responsibility.

The user entry point to the application is served by the *interface* service. This provides a web based interface to the various backend components and rendering of graphs as well as acting as a reverse prosy to the various APIs behind the application.

The *observations* service is exposed via a mostly JSON based HTTP API. It's primary responsibility is the accessing of raw observation files and events, as well as downsampling for rendering in a web browser.

The *SAX* service is again exposed via a JSON HTTP API. Its purpose is to perform PAA and SAX operations on a given event and return data for rendering visualisations as well as the produced string from the SAX calculation.

[Suffix Trees]

The *worker* service exists to perform long running or compute heavy tasks asynchronously from the interface. It is exposed via a message queue running on Redis (an in-memory key-value database) where the interface or other services can queue tasks.

Additionally the application relies on a PostgreSQL RDBMS instance and Minio (a distributed object store similar to Amazons S3).