

Product Overview

The aim of this section is to give a functional documentation of the product we are building.

- [Business Case](#)
- [Working hypotheses](#)
- [High-level architecture](#)

Business Case

SBS Applications Visibility aims at bringing insights on applications' users thanks to logs analysis. Logs are enriched with Identity Management information.

It can address several use cases:

Customer visibility

- Better visibility of the most used apps in order to improve the apps and facilitate sharing among users
- Measure users adherence to applications and functionalities detect training needs/new requirements
- Usage and User monitoring allows to be proactive instead of reactive to the issues

Process optimization

- Replace manual queries by automated process
- Support Chromebook adoption process

Portfolio management

- Reduce the number of inactive apps
- Confirm functionality usage
- Confirm Apps usage (lack of knowledge, missing functionalities..)
- Improve licence needs forecast by a more accurate number of users tracking

Security

- Anomaly detection
- Access reports
- Alerts with inactive/exited accounts/offboarding/transfers

Working hypotheses

Standardization

The framework is built in a standardized way. It means that applications comply with the framework, and not the other way around.

Concretely, how is it implemented?

Data tables: all applications share the same data framework (same data schema).

Frequency: all applications logs/access lists are retrieved at the same monthly basis

Dashboard: all applications share a common dashboard.

Why? To make maintenance way easier, and onboard new applications with limited delays scalability and reliability

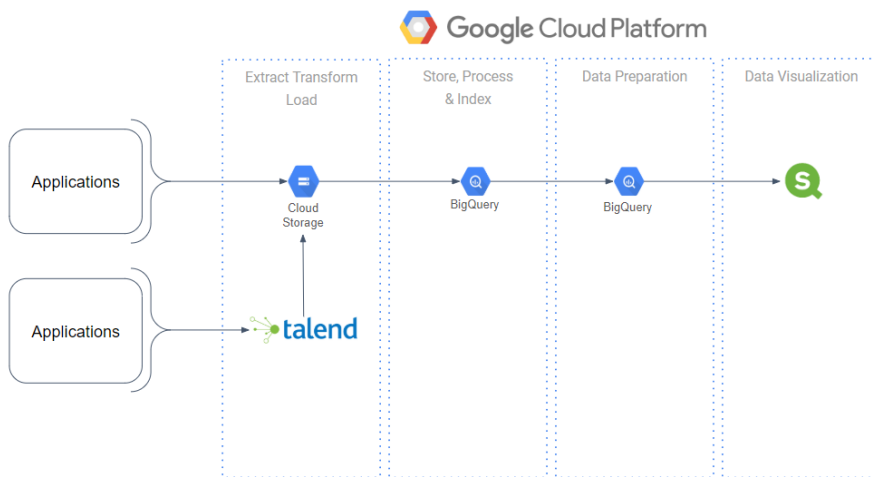
Self-service

The service we target is a self-service application, in which it is possible to onboard a new application with limited intervention of the team, and in which any Application Owner could receive alerts if there is an issue with the data. The dashboard is also provided in a partly self-service mode, with maintained, common indexes, and the possibility to create new screens to customize the visualizations.

Why? To enable fast development and lower the risks maintainability, autonomy and scalability

High-level architecture

SBS Applications Visibility front-end is a dashboard hosted and maintained in QlikSense.



High-level architecture.

1) Data is imported into Google Cloud Storage via an ETL (Talend) or other means. The data ingested can be provided in a shared folder, a Google spreadsheet, a server,

Data is composed of 2 parts: one part is the logs of the app, the other part is the list of users with an access to the app.

2) The data is sent to BigQuery to be stored and indexed.

3) Data Preparation is made to create/update the datasets that will feed the QlikSense

4) Data can be visualized in QlikSense