Introduction of a data visualization service: Reflect.

Lu Han

1 Why we need data visualization service

Data visualization is important since we understand and retain information better when we can visualize our data. Nowadays, more and more companies start incorporating data analytics into their application. For example, for b2c (business to customer) web application, a customized, dynamic interactive data visualization and analytics tool would help business owners make their business better. However, many companies spend millions of dollars and hire many engineers to build their similar infrastructures for visualization of their own data. To prevent repetitive work of the same type, we need a data visualization service.

2 What is reflect

Reflect is a data visualizations and analytics platform for developers based in Portland, Oregon. Just like Sendgrid, Twilio and Strip, Reflect was developed with the idea of giving developers discrete modular tools that they could directly plug into their applications without worrying about the code.

3 How Reflect works

Reflect is composed three major parts: Reflect agent, the Reflect API and Reflect.js.

To use this, Step 1 is to connect Reflect agent with your data warehouse. Step 2 is to design your data model and design some data visualizations for your project with a browser-based drag-and-drop user interface provided by the Reflect API. Step 3, user can then embed the stunning visualization into your application with a few lines of code.

In sum, Reflect agent is the middle layer between Reflect API and your data and it automatically connect to Reflect API. It translates requests from embedded visualizations in your application to the native query language of your database. The agent fetches data defined by data model and embedded visualization and brings it into Reflect, where you align the columns in your database to the dimensions and metrics you want to visualize.

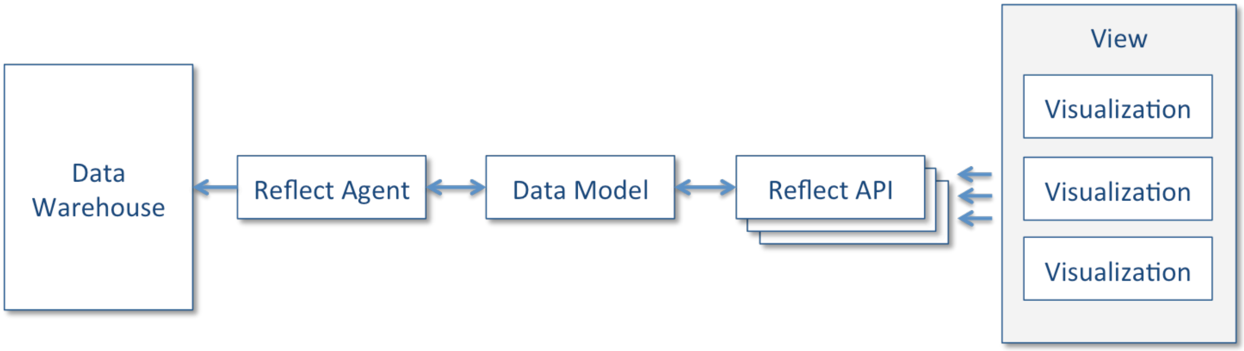

Figure 1. Reflect workflow diagram.

The functionality of each component of Reflect is described in more detail below.

3.1 Reflect Agent

Reflect agent is the middle layer between your data warehouse and reflect API. It is “a highly scalable, secure, and fault-tolerant way of delegating access to databases in your infrastructure to Reflect”. The agent only need read access to your database and it will never write to your database. You can either give Reflect’s software agent the access to your database or, if you are worried about security, you can host your own copy of the agent (Support Linux operating system). Only reflect services should be able to talk to your agent. Thus, a two-way SSL is used. Agent response and clients authentication are both encrypted via SSL for security.

3.2 Reflect API

Reflect API provides a browser-based application, which enables easy data-model configuration and data visualization design.

Security: Authentication is required for Reflect API access.For security, all traffic through Reflect is encrypted using TLSv1.2.

Projects: Everything about the visualization you embed into your application will be contained in a project object. These information includes all the views you will embed and the data model.

Data model: In general, data model informs determines what Reflect agent will fetch and how to use these data to generate reports. More specifically, a data model can contains the following parts:

a. Field: A field is made up from a column in a table, a calculation or a SQL expression.

b. Filters: A filter defines a field that user can use to filter their reports.

c. Data set definition: This definition defines where does some data lives in your database and stores the credentials for accessing that data set. It also defines how data sets relate to each other.

Views: Views are made of multiple view components. The configuration of each components and view are stored in the configurations object. There are five major view components: Timeseries, Donut, KPI, Bars, and DataGrid.

3.3 Reflect.js – the view embedding library

Reflect.js provides two view-embedding methods. It also has libraries for popular JavaScript frameworks, which enables customization of views to fit your web site style.

4 Sample web application

As shown down below with simple embedding, you can have many customized beautiful interactive charts on your application.

Customizable style: with CSS, you can change the style of your charts to fit your application style.

Interactive: when you move your mouse on top of certain parts of the chart, related data secures of data points will be highlighted.

Customized: User can create filters to filter out unwanted data points. User can also change data secures that gets displayed. User can tune the granularity of data. For example, one can change the Top geos donut chart to display by states instead of by regions for more detailed information.

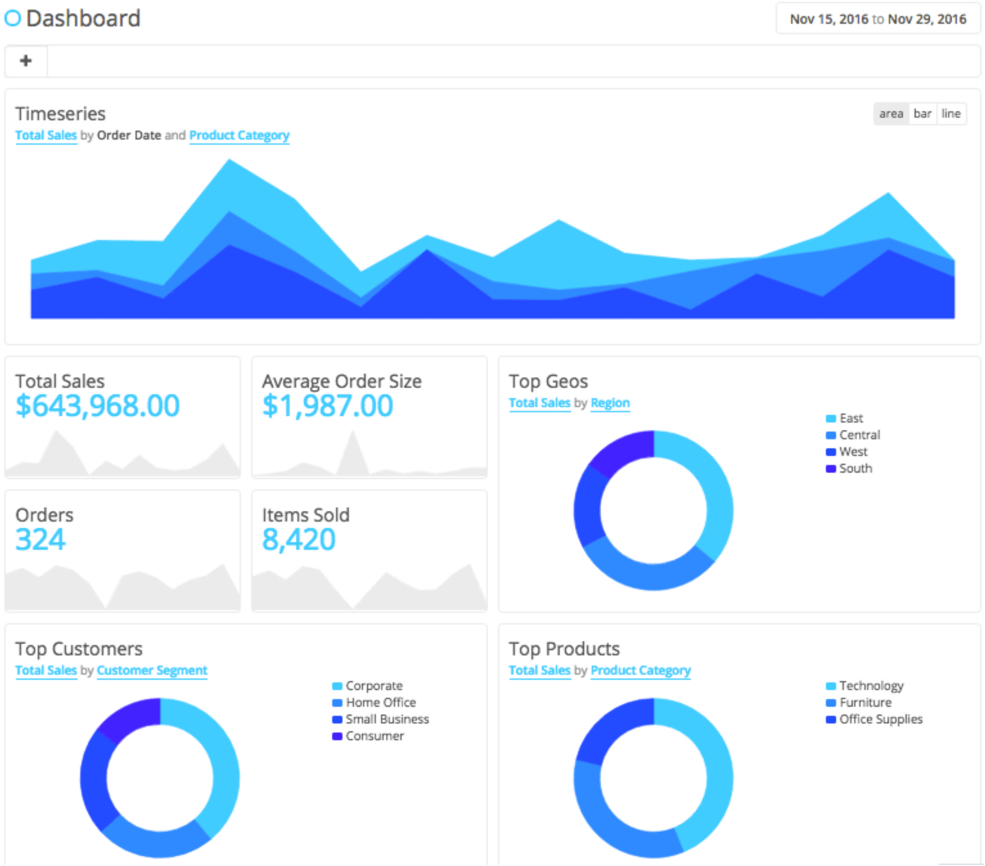

Figure 2. Sample embedded data visualizations in web application

5 Discussion

The Reflect platform has the advantages such as easy addition of data visualization models, shortening development time, saving developing resources.

On the other hand, it also contains several limitations. First, as software developed by a small startup, the functionality of this platform are limited and not fully developed. For example, there're only five visualization-component types, which limits the potential use-cases of this platform. Secondly, this platform is not open source and it has a small user community. If you encounter a problem when using the platform, it may be hard or takes a long time to troubleshoot or to fix. Third, this is a non-free commercial service. It would be a discouraging factor for users with a small budget.

6 Summary

Reflect is a data visualization service developed by a small start-up. It provides easy way of incorporating data visualization dashboards into any web or mobile application within minutes. Developers only need to care about what data they want to display and how to visualize them.

Being a fairly new product, it certainly has its limitations and possible problems. However, I think it is a promising product or at least idea for it provides a discrete tool or service with high demand is great for that it saves resources and prevents unnecessary repetitive work.

7 References

Reflect Agent: https://reflect.io/docs/reference/agent-documentation.html
Reflect API: https://reflect.io/docs/reference/api-documentation.html
Reflect.js: https://reflect.io/docs/reference/reflect-js-documentation.html
Reports about Reflect:
https://techcrunch.com/2016/07/26/data-visualization-service-reflect-raises-2-5m-seed-round-led-by-dfj/
http://www.computerworld.com/article/3099119/application-development/reflect-raises-cash-to-be-the-visualization-engine-for-the-modern-world.html
http://www.creativebloq.com/design/seven-dirty-secrets-data-visualisation-2132787