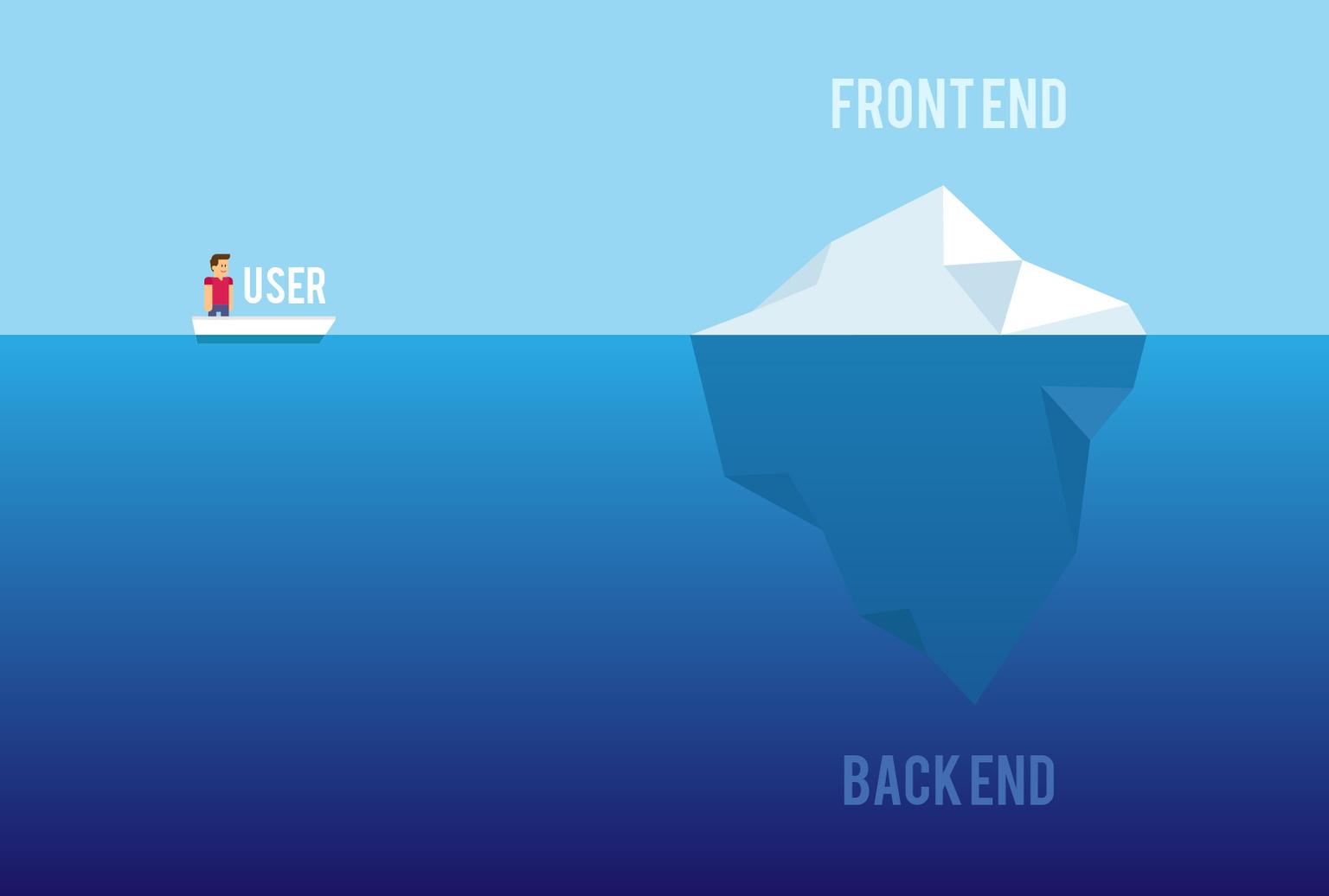
**Research on Back-end Systems e.g. Firebase**

**2nd October, 2018**

**Made by Karina Karushkova**

**What is a Back-end System?**

****For my research, for which I mainly used the web, I found many websites in which Back-end is explained. Some of them are:

* “…any system that supports back-office applications. These systems are used as part of corporate management and they work by obtaining user input and gathering input from other systems to provide responsive output.” (Techopedia.com, n.d.)
* “…deal with databases and data processing components, so the purpose of the back-end system is to launch the operating system's programs in response to front-end system requests and operations” (Techopedia.com, n.d.)
* “… back-end system implements responses to what the front end has initiated.” (Techopedia.com, n.d.)
* “BaaS providers have a broad focus, providing SDKs and APIs that work for app development on multiple platforms, such as iOS, Android, Blackberry, Windows Phone, HTML5, and others.” (En.wikipedia.org, n.d.)
* “Each BaaS provider offers a slightly different set of backend tools and resources. Among the most common services provided are push notifications, file storage and sharing, integration with social networks such as Facebook and Twitter, location services, database persistence and queries, messaging and chat functions, user management, running business logic, and usage analysis tools.” (En.wikipedia.org, n.d.)

**List of providers for Mobile Back-end services:** (En.wikipedia.org, n.d.)

* Azure
* Backendless
* CloudBoost
* Firebase (which we are going to explore more)
* Oracle Cloud
* Red Hat

To be able to build a mobile project, there will be a choice made between a local application and an application connected to a cloud. The difference between those two types is the connection to a server/cloud and interaction. The local application doesn’t connect to a data server or the client; everything is stored on the user’s device. The Connected to a server application, on the other hand, has complete interaction with the app server and user /and between users as well. The choice the developer makes will be important as this will affect the next steps of development and end solutions of the app. (Apps Panel, 2018)

****But as we are talking about building apps, let’s look at the one of the most popular services: the Firebase.

**What is a Firebase?**

* Firebase is a service founded in 2011 and acquired by Google in 2014.
* The Firebase Real-time Database is a cloud-hosted database
* “Data is stored as JSON and synchronized in real-time to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of your clients share one Real-time Database instance and automatically receive updates with the newest data” (Firebase, n.d.) - firebase sends the new data as soon as it’s updated
* “When you connect your app to Firebase, you’re not connecting through normal HTTP. You’re connecting through a Web Socket.” (Esplin, 2016)
* “Firebase uses a NoSQL; JSON database engine.” (Firebase, n.d.)
* “Communication between database and client application is handled via web sockets. That’s actually one of the tricks Firebase uses to provide the real time functionality to developers and end users. Of course, such approach forces you to handle your mobile application code more efficiently.” (Hałabuda, 2016)

**Good and bad parts of this service: (Esplin, 2016)**

**Pros**

* Email & password, Google, Facebook, and Github authentication
* Realtime data
* Ready-made api
* Built in security at the data node level
* File storage backed by Google Cloud Storage
* Static file hosting
* Treat data as streams to build highly scalable applications
* Don’t worry about your infrastructure!

**Cons**

* Limited query abilities due to Firebase’s data stream model
* Traditional relational data models are not applicable to NoSQL; therefore, your SQL chops will not transfer
* No on-premise installation
* **“Vendor lock-in**, which prevents you from making your app portable. Besides that, you cannot access your data as it is hosted on the Firebase server.” (Clark, 2017)

And the **vendor lock-in** leads to: (Clark, 2017)

* You cannot optimize your backend according to your app’s needs;
* You’ll lose the ability to shop around and the power to negotiate;
* You’ll not be aware of the new technologies that are used by other vendors;
* You are always locked by Google updates, and considering the nature of these updates, it is impossible for you to customize your app on server levels;

**When you should and when you shouldn’t use Firebase:**

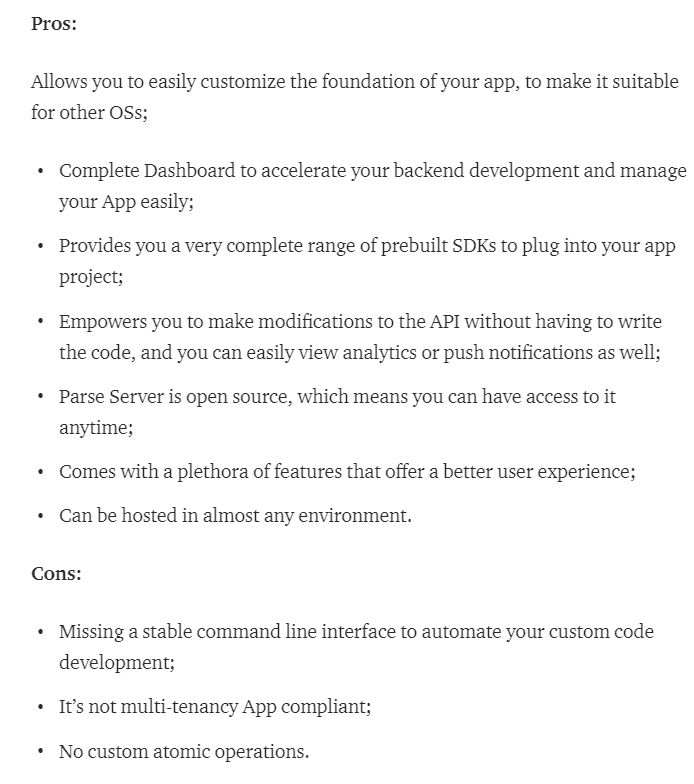
“If all you need is to react to the addition/update of items in a collection or object, Firebase is great. If you need extensive queries or have complex relational data, Firebase would be a poor choice for your main database. Still, it could still be an excellent component of a well balanced serverside infrastructure.” (Croos, 2018)

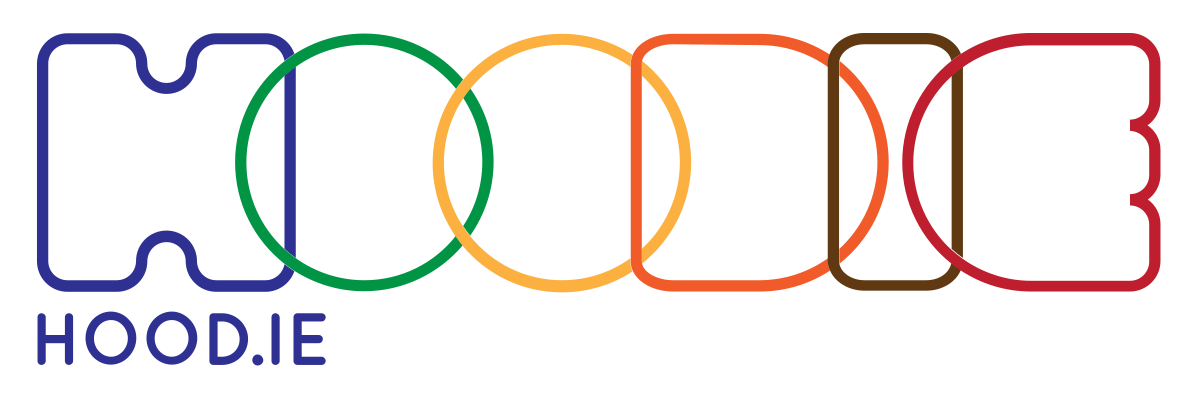
**Are there other open-sourse back-end systems?**

Yes, there are. For my research I found three more back-end systems that can be alternatives for Firebase. The website from which I found all the information below (about the three systems) is “[Medium.com”](https://medium.com/@brenda.clark/firebase-alternative-3-open-source-ways-to-follow-e45d9347bc8c), written article by Brenda Clark.

 The first example of a system is:

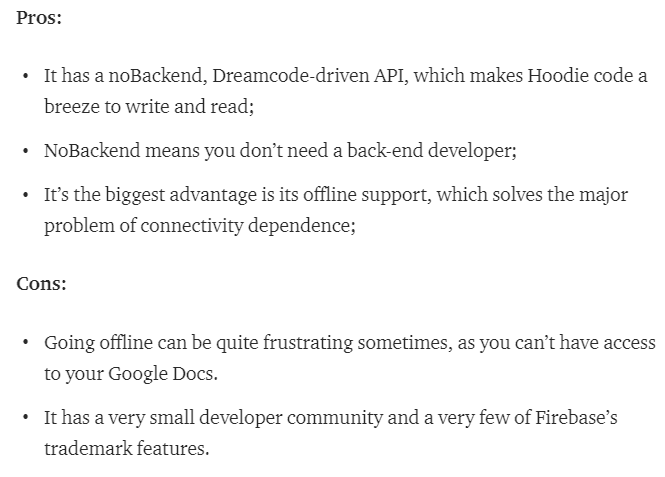
**Parse Server** – “was created by facebook (and the open source community) with the same power and a huge and important differential: to be Open Source.”





The second one is:

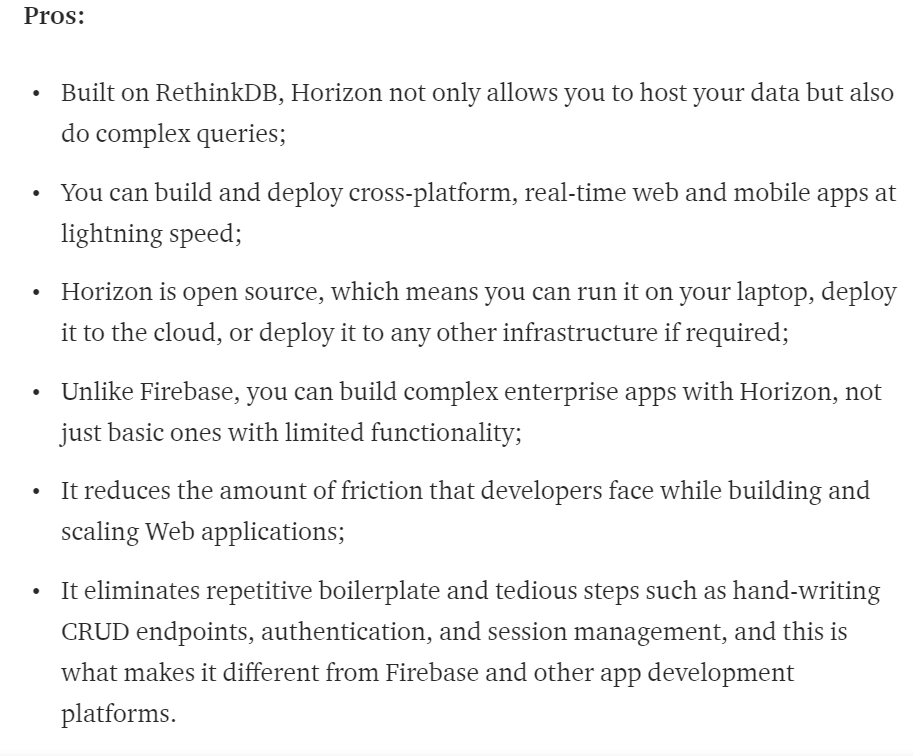
**Hoodie – “**It the First Offline JavaScript Backend that makes web application development very fast, easy, and accessible. It is a simple, self-hosted, and open source platform that allows developers to build robust and intuitive applications for both web and iOS.”; “Unlike other backend platforms, Hoodie uses a community-driven approach to API designing, known as Dreamcode.”

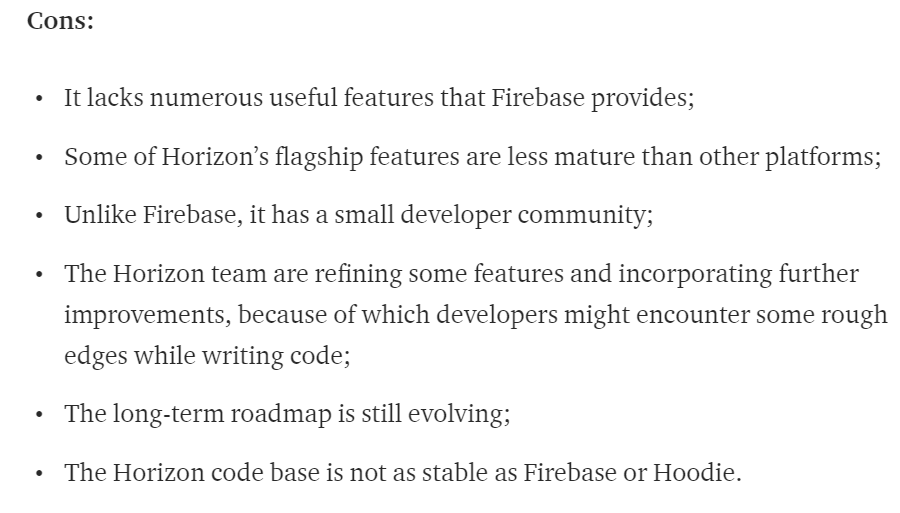


And the third one:



**Horizon – “**a real-time, open-source backend platform that allows you to rapidly build and deploy web and mobile apps using a simple JavaScript API. Built by the RethinkDB team and an open-source community, Horizon lets you scale your app to millions of users without any backend code. It is a standalone server for browser-based JavaScript apps. “



****

**References:**

1. Apps Panel. (2018). What is the Mobile Backend ?. [online] Available at: http://www.appspanel.com/en/what-is-the-mobile-backend/ [Accessed 2 Oct. 2018].

2. Techopedia.com. (n.d.). What is a Back End System? - Definition from Techopedia. [online] Available at: https://www.techopedia.com/definition/1405/back-end-system [Accessed 2 Oct. 2018].

3. En.wikipedia.org. (n.d.). Mobile backend as a service. [online] Available at: https://en.wikipedia.org/wiki/Mobile\_backend\_as\_a\_service [Accessed 2 Oct. 2018].

4. Firebase. (n.d.). Firebase Realtime Database | Firebase Realtime Database | Firebase. [online] Available at: https://firebase.google.com/docs/database/ [Accessed 2 Oct. 2018].

5. Hałabuda, P. (2016). How To Use Firebase In Mobile Apps Development. [online] Whalla Labs. Available at: http://whallalabs.com/how-to-use-firebase-in-mobile-applications/ [Accessed 2 Oct. 2018].

6. Esplin, C. (2016). What is Firebase? – How To Firebase. [online] How To Firebase. Available at: https://howtofirebase.com/what-is-firebase-fcb8614ba442 [Accessed 2 Oct. 2018].

7. Croos, P. (2018). When You Should (and Shouldn't) Use Firebase | Codementor. [online] Codementor.io. Available at: https://www.codementor.io/cultofmetatron/when-you-should-and-shouldn-t-use-firebase-f62bo3gxv [Accessed 2 Oct. 2018].

8. Clark, B. (2017). Firebase Alternative: 3 Open-Source ways to follow – Brenda Clark – Medium. [online] Medium. Available at: https://medium.com/@brenda.clark/firebase-alternative-3-open-source-ways-to-follow-e45d9347bc8c [Accessed 2 Oct. 2018].