

Backend as a Service

Xiaoxiao Wang, University of Illinois at Urbana-Champaign

Building backend for modern applications can be a challenging task. The time and financial cost of maintaining their own infrastructure is usually a bottleneck for small or medium size business. Thanks to the quick evolution of cloud computing, Backend as a Service becomes an emerging trend in mobile and web application development. Being part of the cloud computing industry, BaaS helps developers manage server maintenance and complex infrastructure. This allows them to be able to concentrate on creating great user experience. By providing features such as user management, push notifications, scaling and securing, global deployment etc. BaaS has made the process of delivering applications faster than ever. We review the recent development of the BaaS industry and compare features of three BaaS providers (Parse, Firebase, AWS Mobile Hub) and their prospect in this paper.

1. INTRODUCTION

Building your own backend for your application has its pros and cons. On the positive side, you can choose almost any language or framework. With this full control, highly customizable and flexible features can be incorporated into your apps. On the negative side, building a backend can be a daunting task without adequate experience with server side programming and maintenance. A custom backend takes a lot of time to build, and afterwards require regular maintenance. Besides, security and data access protection is also a crucial concern for you. All these costs, however, may not be worth the benefit.

Ever since 2011, mobile backend as a service (MBaaS), also known as “backend as a service” (BaaS) [Monroe 2013a][Monroe 2013b][Lane 2012], becomes a rising trend in web and mobile application development. This service provides a way to link your applications to backend cloud storage and APIs exposed by back end applications. Instead of building a whole custom backend, BaaS providers enable developers to store their data and interact with other critical services in the manner that is predefined. Without the headaches and cost of maintaining their own infrastructure, developers can focus more on user interface and user experience design. Although this industry in cloud computing is fairly new, it has been attracting mainstream interest with enterprise consumers [Boyd 2014]. According to the report “Cloud/Mobile Backend As A Service (BAAS) Market by Service Type (Data and Application Integration, Identity and Access Management, Usage Analytics, Professional Service, and Support and Maintenance Service) - Global Forecast to 2020”, the global BaaS market size is estimated to grow from USD 1.32 Billion in 2015 to USD 28.10 Billion by 2020, at a Compound Annual Growth Rate (CAGR) of 84.2% [Mar 2015].

In addition to cloud-based storage, which is just the beginning of a BaaS offer, BaaS providers often enable common features required for web and mobile applications such as user management, push notifications, integration with social networking services, custom business logic and more [Lane 2013][Parse 2016]. These services are provided via the use of custom software development kits and application programming interfaces. In general, what is possible with a BaaS varies depending on the platform provider. Thus it is important to understand the need of your app before selecting a BaaS offering. In this paper, we will compare some popular options of BaaS from many perspectives.

Our paper is organized as follows. First we briefly introduce the concept of cloud computing as background, and compare the differences of BaaS from IaaS and PaaS in Section 2. Section 3 about three

BaaS providers in details, namely Parse [Parse 2016], Firebase [Firebase 2016], AWS Mobile Hub [AWS 2016]. Following this, we compare the difference among them and some other providers. Finally Section 4 summarizes our paper.

2. CLOUD COMPUTING

Last ten years has witnessed ever increasing growth of cloud computing. Cloud computing, also on-demand computing, according to IBM.com, “is the delivery of on-demand computing resources-everything from applications to data centers-over the Internet on a pay-for-use basis” [IBM 2016]. Typical computing resources include network, servers, storage, applications, and services etc. Compared with having local servers or personal devices to handle applications, the benefits of cloud are prominent.

The advantages of cloud can be summarized in the following aspects:

- Costs: Free from the upfront costs for purchasing, deploying and maintaining an infrastructure (including the software, server and storage), cloud computing allows a company to focus on projects that differentiate its businesses. Enterprise can get their applications up and running faster, with improved manageability and less maintenance.
- Elasticity: An organization is able to quickly scale up and down its services to meet fluctuating and unpredictable business demand. The common pay-as-you-go subscription model enables easily adding or removing services and you will only pay for what you do use.
- Backups and high accessibility as well as availability: Most cloud providers have high availability and SLA of 99.999% [Spolala 2015].

For small and medium size business who is often limited in time and financial resources for infrastructure, cloud computing provides an appealing shortcut. This shortcut does not only facilitate the process of developing apps, but also more importantly, it lower the cost of failure, which is not uncommon in the world of consumer software.

Notable cloud computing vendors include Amazon EC2 and Web Services, Google App Engine, IBM Smart Cloud, Microsoft Private Cloud and Windows Azure [Berry 2013]. Some of them are experiencing growth rates of 50% per year over the last decade [FSN 2016].

Cloud computing solutions are grouped in different service models depending on how the resources are exposed to final users. Fig. 1 [Remde 2011] illustrates the idea of three main categories. They are Infrastructure as a service (IaaS), Platform as a service (PaaS) and Software as a service (SaaS). The main differences among them lie in which resources are managed by users and which are managed by cloud providers.

2.1 IaaS - Infrastructure as a Service

IaaS is the most basic cloud-service model. It provides access to computers as either physical or virtual machines (the latter is more common), and other resources such as storage, firewalls etc. Basically users of IaaS will receive virtual servers that they will manage. To deploy their applications, users need to install and maintain operating-system images and their application software on the cloud infrastructure.

2.2 PaaS - Platform as a Service

PaaS is a form of cloud computing that offers a platform and environment to allow developers to build and deliver applications. Cloud providers provide a computing platform, typically including operating system, programming language execution environment, database, and web server. Some PaaS offers like Microsoft Azure and Google App Engine allocate resources for users automatically. Users have

Separation of Responsibilities

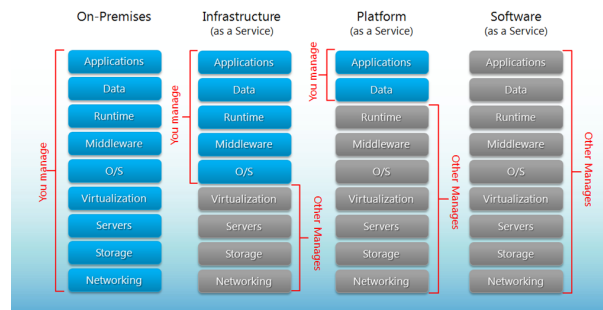


Fig. 1. Comparison between IaaS, PaaS and SaaS [Remde 2011]

control over the deployed application and configuration settings for the application hosting environment.

2.3 SaaS - Software as a Service

SaaS is a business model where software is licensed and delivered in the cloud. SaaS can be seen as on-demand software for end users. Most notable SaaS are Gmail, Facebook, Evernote, and so on.

2.4 BaaS - Backend as a Service

BaaS provides a backend solution for applications (mostly mobile). It is a fairly recent development in cloud computing industry as mentioned before. Most BaaS startups dated from 2011 or even later. Its arising is closely related to the unprecedented growth in the number of mobile apps on multiple platforms such as iOS, Android, Blackberry, Windows Phone and HTML5 etc. on all kinds of devices like mobiles, tablets and desktops.

In developing a simple new mobile application, BaaS avoids the frustration of deployment of IaaS platforms. It is somehow similar to SaaS, but BaaS is mostly targeted at developers instead of end users. Unlike PaaS, which is not designed specifically for mobile developers, BaaS targets exactly what developers need to build the next generation of mobile apps.

3. BAAS PROVIDERS

Building modern apps requires a wide range of skill-sets. Apart from front end UI/UX design, it can call for configuring servers, managing databases, writing APIs, managing security and authenticating users, networking and serialization, scaling, building deployment processes, and so on. With BaaS doing this heavy lifting, developers are able to focus more on creating a great user experience and making their apps unique from others.

Despite of some most common services provided, each BaaS provider offers a slightly different set of backend tools and resources. Choosing the right one can be difficult. In this section, we will discuss three main Providers. They are Parse from Facebook [Parse 2016], Firebase from Google [Firebase 2016], and AWS Mobile Hub from Amazon [AWS 2016] [Cowart 2014][Manglani 2016][Heller 2014].

3.1 Parse

Parse was founded in 2011 by Tikhon Bernstam, Ilya Sukhar, James Yu, and Kevin Lackner [Crunch-Base 2016a]. It was already well known before it was acquired by Facebook in April, 2013 for 85 million dollars [RUSLI 2013], and became even more popular thereafter (see Fig. 2.).

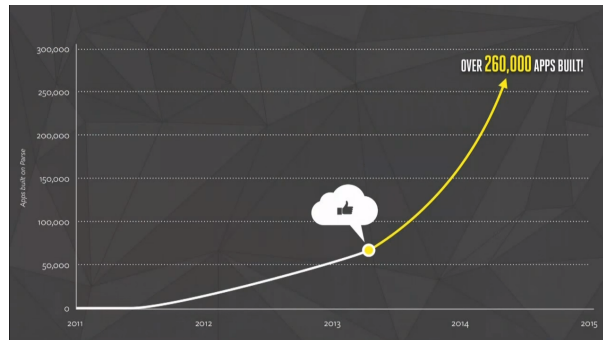


Fig. 2. Number of apps based on Parse.

With an impressive array of client SDKs, people can easily create apps for all devices from desktop to mobiles using Parse. Officially, Parse provides native SDKs for iOS, Android, Windows 8, Windows Phone 8, Mac OS X, JavaScript and Unity (the cross-platform game engine). It also offers the option of utilizing the HTTP REST API. Third party developers have created client libraries for .NET, ActionScript, Clojure, Java, Qt, Ruby, PHP, Python, WebOS and more. Its JavaScript SDK is originally based on Backbone.js framework, but it provides flexible APIs that allow it to be paired with other JS libraries.

Parse has three main features. They are Parse core, Parse push and Parse analytics. These features are independent of each other [FacebookDevelopers 2014].

3.1.1.1 *Parse Core.* Parse core is comprised of three main functionalities.

- Its data storage service allows developers to save data to and fetch data from it.
- Developers have the ability to run custom app code on Parse’s servers.
- It provides ways to manage users so that developers can easily create and manage user accounts and sessions.

3.1.1.1.1 *Data Storage.* Storing data on Parse is built around Parse.Object. Each Parse.Object contains key-value pairs. You can do CRUD for simple data. Below is sample code for save, retrieve and delete objects in JavaScript.

```

1 Parse.initialize("app ID", "js ID");
2 var BeerObject = Parse.Object.extend("BeerObject");
3
4 var beer = new BeerObject();
5 beer.set("name", "FooBeer");
6 beer.set("brewer", "Me, Myself, and I");
7 beer.set("abv", 9.2);
8
9 beer.save(null, {
10   success: function(ob) {
11     //do something
12   },
13   error: function(ob, err) {
14     //oh no! woe is me!
15   }
16 });

```

Listing 1 Parse save data

The screenshot shows the Parse Data Browser interface. At the top, there's a navigation bar with 'Parse' logo and links like 'Dashboard', 'Quickstart', 'Tutorials', 'Documentation', 'Downloads', 'Help', 'Pricing', and 'Products'. Below this is a 'Data Browser' tab. The main area displays a table of data. The table has columns: 'objectId', 'comments', 'howdangerous', 'location', 'numcows', 'createdAt', and 'updatedAt'. The table contains 165 rows of data. On the left, there's a sidebar with 'Classes' and 'Installation' sections. The 'Classes' section shows a list of classes with their respective counts. The 'Installation' section shows a list of installations with their respective counts. The table is currently displaying 20 rows, with a pagination bar at the bottom showing '20 / 165 rows'.

Fig. 3. Parse data browser

```

1 Parse.initialize("app ID", "js ID");
2 var BeerObject = Parse.Object.extend("BeerObject");
3
4 var q = new Parse.Query(BeerObject);
5 q.get("someid", {
6   success:function(ob) {
7     console.log("I got beer " + ob.get("name"));
8   },
9   error:function(ob, err) {
10    //oh no! woe is me!
11  }
12 });
13
14 //Assume b is a beer object we got...

```

Listing 2 Parse retrieve data

```

1 //NOW DESTROY IT! MUHAHAHAHAH!
2
3 b.destroy({
4   success:function(ob) {
5
6   },
7   error:function(ob, err) {
8
9   }
10 });

```

Listing 3 Parse delete data

What's more, Parse also provides a web UI for data browsing and simple manipulation (see Fig. 3.).

Parse supports all kind of relationships, including one-to-one, one-to-many, and many-to-many. Parse.Query enables retrieving data with conditions such as constraints, limits and starting indexes, sorting, counting etc. Parse supports storing photos with Parse.File. Adding a Parse.GeoPoint to a Parse.Object allows associating real-world latitude and longitude coordinates with an object. This makes queries like finding out nearby users straightforward.

3.1.1.2 Cloud Code. Developers are enabled the ability to run custom app code written in JavaScript in Parse's cloud. This is called Cloud Code. They can run code when objects are saved or deleted, create custom functions and web hooks, or schedule long-running jobs with background jobs. Cloud modules from 3rd parties can also be integrated so that developers can send emails, take payments, add SMS and so on.

3.1.1.3 User Management. Parse handles user signing up, logging in and adding 3rd party authenticator. It can manage user sessions, create roles for users with different access, protect user securities etc. The code below in JavaScript illustrates how to sign up a user.

```

1 var user = new Parse.User();
2 user.set("username", "my name");
3 user.set("password", "my pass");
4 user.set("email", "email@example.com");
5
6 // other fields can be set just like with Parse.Object
7 user.set("phone", "415-392-0202");
8
9 user.signUp(null, {
10   success: function(user) {
11     // Hooray! Let them use the app now.
12   },
13   error: function(user, error) {
14     // Show the error message somewhere and let the user try again.
15     alert("Error: " + error.code + " " + error.message);
16   }
17 });

```

Listing 4 Parse create user

3.1.2 Parse Push. Push notifications are a great way that Parse offers to keep app users engaged and informed. Parse has defined an Installation object that pairs an installation of the application to a device. Push notifications are sent to installations.

There are two ways to send push notifications in Parse: Channels and advanced targeting. Channels are for stable interest groups who subscribe to one or more channels. Advanced targeting allows you to send pushes to changing groups by pushing to queries.

Sending notifications can be done from the Parse.com push console (see Fig. 4.), the REST API or from the SDKs used in Cloud Code (See Code Example below).

```

1 Parse.Push.send({
2   channels: [ "Giants", "Mets" ],
3   data: {
4     alert: "The Giants won against the Mets 2-3."
5   }
6 }, {
7   success: function() {
8     // Push was successful
9   },
10  error: function(error) {
11    // Handle error
12  }
13 });

```

Listing 5 Parse push notifications

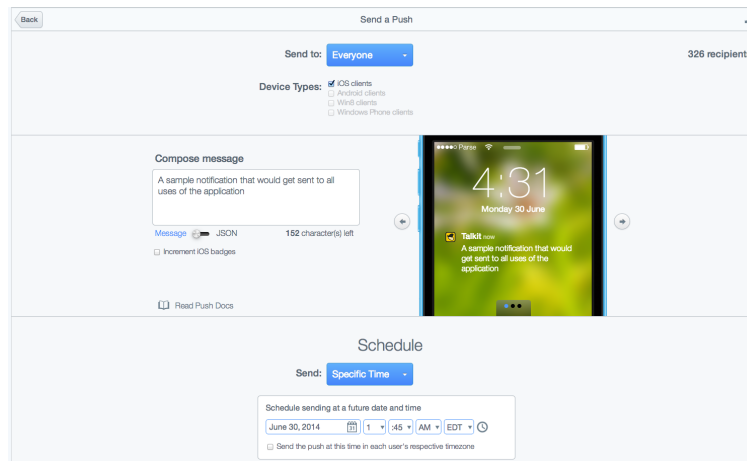


Fig. 4. Parse push notifications

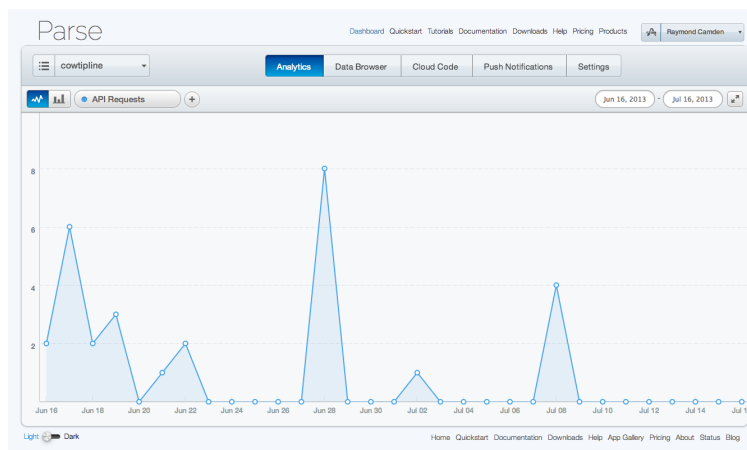


Fig. 5. Parse Analytics

3.1.3 Parse Analytics. Parse Analytics is another great feature that Parse provides. It can measure app usage data such as API calls, burst rates and users' response to push notifications.

Parse provides basic real-time graphs and breakdowns (by device type, Parse class name, or REST verb). You can also track events in the form that you defined, to test the popularity of new functionality you add to your app, or the difficulty of one game level. Similarly, this is accessible from both SDKs and from the Web UI they provide (see Fig. 5.).

3.1.4 Parse Shutting Down. Despite of its great features and popularity among developers, on Jan 28, 2016, Parse announced that they are shutting down their service [Lacker 2016]. Developers will have one year to move their products over to other platforms. It is reported that this decision is going to affect 600,000 apps up to the point when they made the announcement [Lardinois and Constine 2016].

Facebook decides that they need to focus their resources elsewhere. This decision quickly becomes a massive surprise to the developer community and has shaken developers trust in Backend as a Service

industry [Lardinois and Constine 2016]. Even Facebook cannot guarantee their services, who else can we rely on?

3.2 Firebase

Firebase is one of the most popular replacements for Parse. It was founded in 2011 by Andrew Lee and James Tamplin. And then was acquired by Google in October 2014 with an undisclosed price [Crunch-Base 2016b]. As a real-time cloud data service who is designed for building real-time collaborative applications such as chat or multi-player games, its primary product is a real-time database which provides an API that allows developers to store and sync data across multiple clients.

Firebase offers Android, IOS, and JavaScript SDKs. Developers can build cross-platform native mobile and web applications with its SDKs. It also allows connection to existing backend via its server side libraries or REST API.

We will introduce three main features of Firebase: real-time database, authentication, and hosting.

3.2.1 Real-time database. Firebase provides a real-time database. Data in Firebase is standard JSON and it is synchronized in real-time to every connected client. All clients of an application on different platforms share one database and automatically receive updates with the newest data. Developers can use either client libraries or the REST API to access the data. Its Latency is within milliseconds when changes to data are made through a client library [Lee 2013]. Official libraries support Geo Location, AngularJS, Ember, ReactJS, Ionic, UI bindings. Some third party libraries also exist. The REST API uses the Server-Sent Events protocol, which is an API for creating HTTP connections for receiving push notifications from a server [Darrow 2012].

Firebase also provides a highly flexible expression-based security API that allows developers to easily define how the data should be structured and which user has what kind of access to what data. These rules are automatically enforced all the time.

Below shows sample JavaScript code for saving and retrieving data.

```

1 // Create a connection to your Firebase database
2 var ref = new Firebase("https://<YOUR-FIREBASE-APP>.firebaseio.com");
3 // Save data
4 ref.set({ name: "Alex Wolfe" });
5 // Listen for realtime changes
6 ref.on("value", function(data) {
7     var name = data.val().name;
8     alert("My name is " + name);
9 });

```

Listing 6 Firebase sample code

3.2.2 Authentication. Firebase has built-in functionality for authenticating users with email & password, Facebook, Twitter, GitHub, Google, and single-session anonymous login. Developers can choose to authenticate user login entirely with its built-in client-side code. They can also integrate authentication with their existing backend servers using custom auth tokens. Once a user authenticates to the application, Firebase manage their session, ensuring that the user is remembered across browser or application restarts.

```

1 var ref = new Firebase("https://<YOUR-FIREBASE-APP>.firebaseio.com");
2 ref.createUser({
3     email    : "bobtony@firebase.com",
4     password : "correcthorsebatterystaple"
5 }, function(error, userData) {

```



```

6   if (error) {
7       console.log("Error creating user:", error);
8   } else {
9       console.log("Successfully created user account with uid:", userData.uid);
10  }
11  });

```

Listing 7 Firebase create user

3.2.3 Hosting. Hosting was launched on May 13, 2014 [Raynor 2014]. It is a static asset web hosting service which supports hosting static files such as CSS, HTML, JavaScript and other files that do not change dynamically. The service delivers files through HTTP Secure (HTTPS) and Secure Sockets Layer encryption (SSL) from their global CDN. Deploy can be done using Firebase command line tools. Every app gets its own firebaseapp.com domain, and paid firebase plans can host content on a custom domain.

Compared with Parse, who is a relatively complete technology stack for building mobile apps, Firebase is more like a real-time database in the cloud. One of the important things that Firebase is missing is the push notification features. There exists some third party services like Batch that fix this problem. Besides, Firebase doesn't have a built-in way to add Cloud Code as in Parse. You have to add additional logic on your own server to accomplish this. However, on the other side, Parse does not have real time capabilities.

3.3 AWS Mobile Hub

Being one of the earliest and most successful cloud computing platforms, Amazon Web Service is spawning new features, services, and options faster than ever. It launched its Mobile Hub for accessing cloud services to build and monitor mobile apps in Oct 8, 2015 [Krill 2015].

Mobile Hub is an integrated console that combines many AWS services such as DynamoDB [Sivasubramanian 2012], Lambda, API Gateway, Cognito etc. It provides features very similar to that of Parse. Right now it supports both iOS and Android mobile platforms. Based on the features you choose, AWS Mobile Hub will provision and configure AWS resources on your behalf.

In AWS Mobile Hub, user authentication is handled by Amazon Cognito Identity. Push notifications are sent by Amazon Simple Notification Service (Amazon SNS). Amazon S3 enables content delivery such as audios or videos, and caching of downloaded files is possible. Amazon S3 bucket is used to store user data and maintain access permissions. Mobile Hub use Amazon Mobile Analytics to configure event collections, which enables app analytics. Cloud logic is powered by AWS Lambda, which allows developers to write code without worrying about managing frameworks and scaling back-end infrastructure. Cloud functions can be written in JavaScript, Java, or Python. All these different components within AWS are configured to play nice with each other.

Apart from the above features, AWS's generous limits and reasonable pricing makes it one of the best alternatives for Parse. Being a well-respected platform that supports many huge companies like Netflix and Yelp, AWS is supporting over 1 million active business customers right now. [Lunden 2015].

4. CONCLUSION

In this paper, we compare the pros and cons of building backend for applications by yourself and choosing Backend as a Service. Despite of the great features that BaaS provides, taking short cuts means limited control over infrastructure. And one additional risk to consider before choosing a provider is its longevity. The shutting down of Parse has greatly shaken the trust of the development community on Backend as a Service industry. Another example is StackMob, which is BaaS provider acquired by

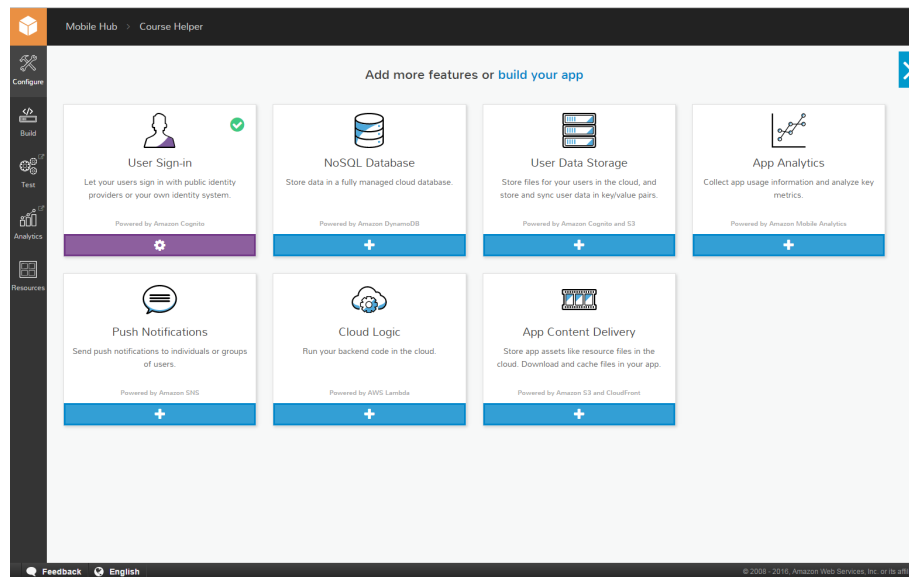


Fig. 6. AWS Mobile Hub console

Paypal in 2013 and was closed two months later [NOVET 2014]. Once the service you chose is shutting down, it is usually non trivial to find alternatives and migrate your applications.

We also reviewed three BaaS: Parse from Facebook, which is shutting down soon, Firebase from Google who has said that they will continue to support its service indefinitely which you never know, and AWS Mobile Hub which is a relatively new service provided by Amazon. Among them, Parse and AWS Mobile Hub provides complete package of features for mobile app development, while Firebase focuses more on real-time data representation and performance. Compared with the other two well-established providers, AWS Mobile Hub has only released SDKs for iOS and Android.

REFERENCES

2015. Backend as a Service (BaaS) Market worth 28.10 Billion USD By 2020. <http://www.marketsandmarkets.com/>. (2015).
- AWS. 2016. <https://aws.amazon.com/mobile/>. (2016).
- Megan Berry. 2013. The economy is flat so why are financials Cloud vendors growing at more than 90 percent per annum? <http://www.fsn.co.uk/>. (March 2013).
- Mark Boyd. 2014. built.io Is Building an Enterprise MBaaS Platform for IoT. <http://www.programmableweb.com/news/>. (March 2014).
- Jim Cowart. 2014. Finding the Right BaaS. <http://www.developereconomics.com/>. (March 2014).
- CrunchBase. 2016a. <https://www.crunchbase.com/>. (2016).
- CrunchBase. 2016b. <https://www.crunchbase.com/>. (2016).
- Barb Darrow. 2012. Firebase secures its real-time back-end service. <https://gigaom.com/>. (Dec. 2012).
- FacebookDevelopers. 2014. Build: Getting Started with Parse. <https://www.youtube.com/>. (May 2014).
- Firebase. 2016. <https://www.firebase.com/>. (2016).
- FSN. 2016. Major Cloud Computing Vendors. <http://www.itmanagerdaily.com/>. (2016).
- Martin Heller. 2014. AnyPresence, Appcelerator, FeedHenry, Kinvey, and Parse share plenty of common ground, but two stand apart. <http://www.infoworld.com/>. (Nov. 2014).
- IBM. 2016. <https://www.ibm.com/cloud-computing/>. (2016).
- Paul Krill. 2015. Amazon introduces Mobile Hub for easier, faster app delivery and adds Python support to AWS Lambda. <http://www.infoworld.com/>. (Oct. 2015).

- Kevin Lacker. 2016. Moving On. <http://blog.parse.com/announcements/moving-on/>. (Jan. 2016).
- Kin Lane. 2012. Rise of Mobile Backend as a Service API Stacks. <http://apievangelist.com/>. (June 2012).
- Kin Lane. 2013. Overview Of The Backend as a Service (BaaS) Space. <https://s3.amazonaws.com/>. (May 2013).
- Frederic Lardinois and Josh Constine. 2016. Facebook Shuttters Its Parse Developer Platform. <http://techcrunch.com/>. (Jan. 2016).
- Andrew Lee. 2013. Cloud Services: How does Firebase differ from Parse? <https://www.quora.com/>. (Jan. 2013).
- Ingrid Lunden. 2015. Amazon's AWS Is Now A 7.3B Business As It Passes 1M Active Enterprise Customers. <http://techcrunch.com/>. (Oct. 2015).
- Kishin Manglani. 2016. Top 5 Parse Alternatives. <https://www.raywenderlich.com/>. (Feb. 2016).
- Martin Monroe. 2013a. The Gospel of MBaaS (part 1 of 2). <http://www.infoq.com/news/>. (May 2013).
- Martin Monroe. 2013b. The Gospel of MBaaS (part 2 of 2). <http://www.infoq.com/news/>. (May 2013).
- JORDAN NOVET. 2014. PayPal closing down backend service StackMob mere months after buying it. <http://venturebeat.com/>. (Feb. 2014).
- Parse. 2016. <https://parse.com/>. (2016).
- Chris Raynor. 2014. Introducing Firebase Hosting. <https://www.firebase.com/blog/>. (May 2014).
- Kevin Remde. 2011. SaaS, PaaS, and IaaS. Oh my! ("Cloudy April" C Part 3). <https://blogs.technet.microsoft.com/>. (2011).
- EVELYN M. RUSLI. 2013. Facebook to Buy Mobile Startup Parse in Cash-and-Stock Deal. <http://www.wsj.com/news/>. (April 2013).
- Swaminathan Sivasubramanian. 2012. Amazon dynamoDB: a seamlessly scalable non-relational database service. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. ACM, 729–730.
- Cristian Spolala. 2015. Cloud offering: Comparison between IaaS, PaaS, SaaS, BaaS. <https://assist-software.net/blog/>. (April 2015).