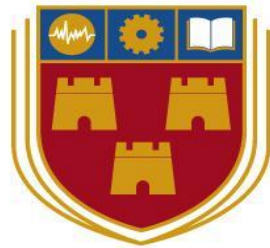


5th April 2017

Research document

Institiúid Teicneolaíochta Cheatharlach



INSTITUTE *of*
TECHNOLOGY

CARLOW

At the heart of South Leinster

Department of Computing and Networking

Software Development Degree

Project name: MaaP (Message as a Platform)

Student: Chihabeddine Ahmed

Student Number: C00210496

Supervisor: Joseph Kehoe

Table of Contents

| | |
|-------------------------------------|----|
| 1. Introduction | 1 |
| 2. Messaging..... | 2 |
| 2.1 History of messaging | 2 |
| 2.2 Type of messaging..... | 3 |
| 3. Existing Applications | 6 |
| 3.1 Matthermost..... | 6 |
| 3.2 WhatsApp..... | 7 |
| 4. Hosting solutions | 8 |
| 4.1 Self-hosted vs Web-Hosted | 8 |
| 4.2 Architecture | 10 |
| 5. Development Tools | 11 |
| 5.1 Eclipse | 11 |
| 5.2 Android Studio..... | 11 |
| 6. Platforms | 12 |
| 6.1 Android..... | 12 |
| 6.2 iOS | 13 |
| 6.3 Cordova..... | 13 |
| 7. Technologies..... | 14 |
| 7.1 Backend technologies..... | 14 |
| 7.1.1 Node.js..... | 14 |
| 7.2 Database Systems | 14 |
| 7.2.1 Mongo | 15 |
| 7.2.2 MySQL..... | 15 |
| 8. Other technologies | 17 |
| 8.1 Java | 17 |
| 9. Messaging protocols..... | 18 |
| 9.1 Websockets..... | 18 |
| 9.2 XMPP..... | 19 |
| 10. Conclusion | 20 |
| 10. References | 21 |

1. Introduction

The purpose of this document is to present the research carried out as a part of the fourth year project. To develop the MaaP (Message as a platform) app, it's useful to understand the basic concepts behind MaaP as well as to obtain some information about MaaP architecture and technologies that are used to develop this application. As this project focuses on delivering a self-hosted platform solution to different type users, we can look at what information required to complete this research document.

This document will be divided into ten sections, each describing a different part of the discussed research. The first section will focus on messaging and precisely will focus on instant messaging, their history, how do they work etc. The second chapter will focus on existing messaging platforms and how are they different from the app that will be delivered. Also, the document will discuss the difference between this existed app, their strengths, weakness, and variety of platforms they support. The third section of this document will focus more on hosted and self-hosted solutions and how they are different. The fourth and the fifth chapter of the document focuses on the development tools and the different type of platforms the application will be developed for. The sixth section will list and describe the frontend and backend technologies that will be used to develop the platform. This includes the programming languages and the most common type of database system which will be considered for the development of this project.

Other technologies that are used to develop the front end will be mention in the seventh chapter. Section eight of this document will provide the reader with knowledge about the most commonly used instant messaging protocols.

The last section will conclude the research and provide a brief overview of selected technologies and such technologies were selected for this project and to include any reference to third party information.

2. Messaging

Messaging is a method of exchanging messaging between two or more people using a real-time software. This involves a messaging system and a messaging client. A messaging system is a peer-to-peer facility: A messaging client can send messages to, and receive messages from, any other client.

2.1 History of messaging

The internet has transformed and simplified how people communicate between each other around the world. This done via instant messaging, email and other communication tools, while instant messaging has played a leading role in bringing people together. We all have used an application like WhatsApp, Google chat etc. But the three is a history behind these applications.

But where did all begins? Messaging has been around for a long time, so it's somewhat difficult to pinpoint its origin.

But we can look at the important advances of instant messaging made over the last fifty years. While the internet becomes so popular, many people were already online.

Services like America Online (AOL), Prodigy and CompuServe were the only application that allows people to get together online.

They provide an actual interface so that the people can use to connect with other people, which creates UX (user experience) for users. It was as early as the 1990s when people began to spend most of their time on the internet. They started to create software such designing chat-rooms and setting up some servers for them.

A chat room it's like physical rooms where people get together and start chatting. The only difference is the message is typed and its instant and it's seen by everyone in the room.

At first, the chat room is only capable of handling two people using instant messaging. And in November 1996 the phrase "instant messages" is exploded on the internet and that's when Mirabilis introduced ICQ, (I seek you). ICQ is a free software application, called the client that can be installed on your computer. It communicates with ICQ server as long as you are online and the client is running. By1997, Services like America Online (AOL), Prodigy and CompuServe were the only applications that allow the users the ability to communicate via chat rooms using instant messages. By June 1998, AOL acquired Mirabilis and ICQ. The ICQ model is the basis for most instant-messaging utilities on the market today [1].

2.2 Type of messaging

Instant messaging (IM)

Instant Messaging allows you to quickly send a message and transfer files instantly to one or more of your friends over the internet as you continue chatting you see a full transcript of your conversation from in the chat window. A message takes a little time to reach the other person and it doesn't matter whether that person is next door to you or around the world! Nowadays there is many IM that you can use. Example. Skype, WhatsApp, Viber, hangouts etc. a typical IM will look like this.

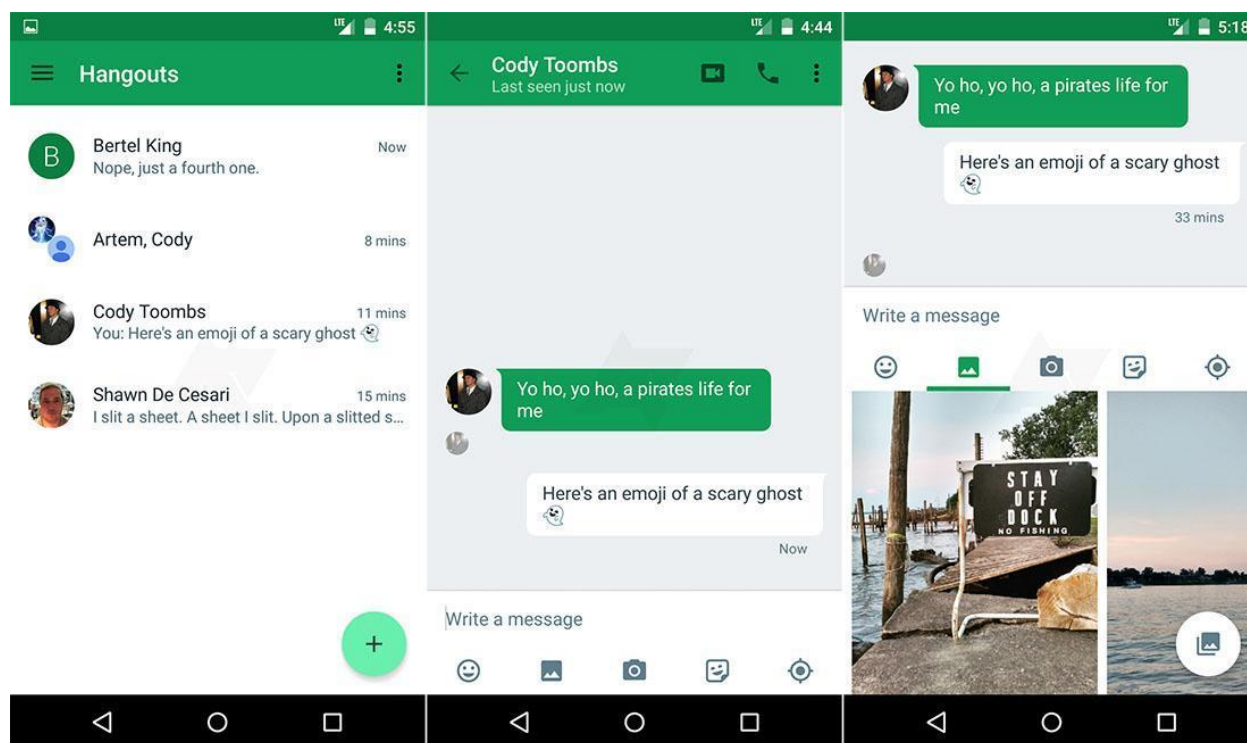


Figure 1: A simple hangouts chat

<https://hangouts.google.com/>

How does IM work?

IM works by first allowing the user to Authenticate using username and password, but most of the mobile application they authenticate using phone number as the identity of the user that would allow him or her to connect or sends information to and from other people. An example of that is, WhatsApp. WhatsApp uses the Phone number as an identifier.

When you start up an IM program, called an IM client, it tries to contact to a particular company's IM server including, Google or Yahoo.

The IM server verifies your username and password and then logs you in. At this point, you're connected. Now you'll see a list of your buddies that are currently online. As your friends sign in and out you'll see them appear and disappear from your contact list. You'll notice the statuses of friends automatically change from "Online" to "Away", "Busy" or "Offline" based on whether that person is logged in and using his/her computer. These statues are often indicated by a simple status icon to tell user's status and when you send a message to your friend it travels from your IM client to the IM server to which you're connected. The message then travels directly to your buddy and a window pops up on his/her machine. As you continue your conversation the messages go back and forth between the clients and servers with little delay.

Now let's suppose you want to send a file to your friend. Your IM client essentially asks the IM server where your friend is located on the internet and then helps your computer make a direct connection to your friend's computer. The reason a direct connection is made is because more efficient than trying to send every file through the IM servers and means the IM companies don't have to use as much internet bandwidth. Instant Messages are tiny while files like movies and audio can be quite large. On average, a whole month of Instant Messaging internet traffic adds up to under 1 megabyte, a tiny amount of space. Consequently, it's no wonder Instant Messaging works on both slow and fast internet connections.

Advantages and disadvantages

Instant messaging

It is instant. Assuming the intended target is online, the message will be received as soon as you send it. Instant messaging tends to be confined to a single thought or sentence rather than a lengthy essay. It is conversational. Instant messaging has an immediate back and forth exchange that lets you collaborate in real-time. Some services like WhatsApp and other does not obtain users details, so you don't worry about your identity being unmasked. But it is intrusive. Instant messages pop up on the screen when you are in the middle of doing something, but you can disable them from the application, but some people wouldn't like to miss their incoming messages. It is proprietary. It requires the both parties to have the same app, both must be connected to the internet and both must be login in. You lose the history. Some application only allows you to see the backdated history up to few months and some of them don't.

The last disadvantage of using instant message is you cannot send a text to a group of people unless you are participating in a group chat.

Email

Short for electronic mail, email (or e-mail) is defined as the transmission of messages over communications networks. Typically, the messages are notes entered from the keyboard or electronic files stored on disk. Example Gmail, Hotmail etc. it is universal. Almost everyone has from any email provider to any email provider. The email can be sent to an individual person or group of people, rather than sending out separate messages and you can also communicate with an entire team or list of the individual at once. But it is slow. "Slow" is a relative term, since messages can be delivered to the other side of the world in a minute or a second but when a message is actually retrieved is a function of the mail server and email client software, and you don't know when it is actually received. The worst thing about emails, they are filled with spam emails. And companies like Microsoft are trying to take down botnets and to reduce but still many users' get spams. And according to a recent Symantec Intelligence Report suggests spam still accounts for nearly three-fourths of all email. It is a primary means of spreading malware.

3. Existing Applications

This section of the document will focus on the different type of existing application that uses hosted and self-hosted technologies and how are they related to the application going to be developed.

3.1 Matthermost

Matthermost is an open source, self-hosted Slack-alternative. Slack is a cloud-based instant messaging and collaboration system that allows teams to communicate with each other. As an alternative to proprietary SaaS (Software as a Service) messaging, Matthermost brings all your team communication into one place, making it searchable and accessible anywhere. The strength of the application is it's a self-hosted, which means you don't have to rely on someone else's servers, you can host matthermost on your own server. But not all operating systems are supported by matthermost. It has available on Android, IOS, but there are no official apps for any of the OS [2].

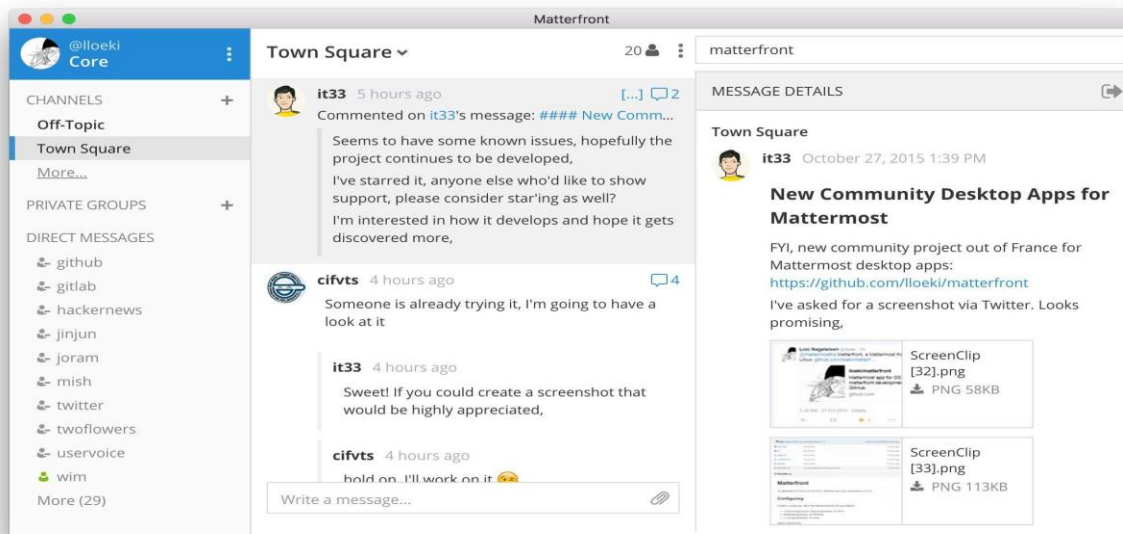


Figure: 2 Mattermost UI
<https://about.mattermost.com/>

3.2 WhatsApp

WhatsApp [3] is a texting service between mobile phones as a replacement for the regular SMS text messages. Over 900 million users are active worldwide using the WhatsApp service. WhatsApp uses an internet connection between phones. The application available on cross-platforms devices.

You can easily send free messages to any part of the world without any charges and you that app will allow you to import the contacts from your phone and tells you that how many of your friends are using the app. Photos, locations, status can be shared with friends. Send 100 messages a month to any of your friends not using WhatsApp for free. However, everything having a plus point also has some negative points too. So here are some disadvantages of this brilliant App. You can chat to friends that only have smartphones supporting this application and to friends having their account on WhatsApp. You must have access to the internet to send and receive messages for free and the messages are also not sent to the phone inbox.

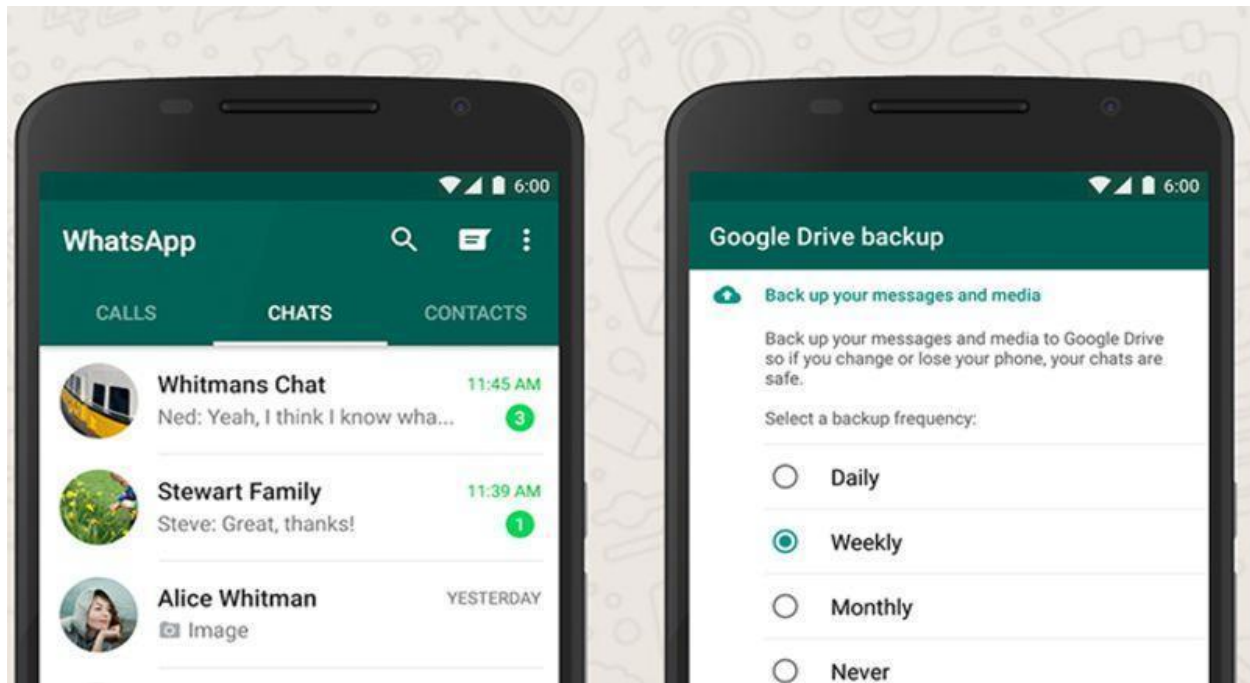


Figure: 3 WhatsApp UI
<https://www.whatsapp.com>

How are they different from this project's app?

From the end-user, the application is going to be like these existing applications mentioned above. However, the major difference is in where the platform is running and who owns the data of the platform. The backend development will be different from the existing applications mentioned as these application uses monolithic architecture. Monolithic architecture is where the application functionality is distinguishable. For example data input and output, data processing, error handling, and the user interface are all interwoven, rather than containing architecturally separate components. However, the application will be built using the micro-services architecture which means it's a suite of independently deployable, small, modular services in which each service runs a unique process and communicates through a well-defined, lightweight mechanism. The application is going to be self-hosted which means it going to run on your server unlike web-hosted they rely on someone else's server. That will give complete freedom on the application and the data of the application.

4. Hosting solutions

4.1 Self-hosted vs Web-Hosted

Do you prefer using a car? Or do you prefer taking the bus?. Cars allow you to go anywhere anytime you want. You can customise it. You can choose between a car with good mileage and a car with more storage space. You can choose how fast you get to your destination and you can choose what radio station to listen to etc. But of course, there are some downsides of owning a car. You have to pay for the car itself, parking, filling it up with gas, maintenance, and repair etc. If you like the freedom of owning a car you'll probably like a self-hosted solution. Or you could, of course, take public transportation. It goes to where most people want to go, it's a maintenance free (for you). The downside of using buses is there are many busses but you can't leave exactly when you want you might have to wait a bit to get on the bus. There are usually reasonable monthly fees instead of a big upfront cost when purchasing a car. You can't customise anything on the bus. You have to follow their rules etc. If you like taking public transportation you'll probably like a hosted solution.

Technical Explanation

Hosted

A hosted solution is a piece of software that runs on someone else's server. In most cases, you don't have access to the code that runs your site or the app. You use their website to make changes to your site. You don't manage it & you don't run it. You create an account and then edit your profile via their site. A good example is Facebook, WhatsApp, Viber etc. you use the app but you don't worry about how it runs and how it's maintained.

Advantages and disadvantages of a hosted solution.

Ease of Use, by far one of the best reason to use a hosted solution is they're easy to use. It's about as complex as signing up for a Facebook account. With a hosted solution you never have to worry about updates are made to everyone immediately. The downside is that it's like using a bus, you have to pay for a ticket. These fees can be monthly fees or annual fees.

Self-Hosted

Self-hosted alternatively is a piece of software that you run on your own machine (or a machine you rent from someone). You usually see the code and are told where to upload it. A good example would be a WordPress. You can control exactly when you want to update it, but you are ultimately the one responsible for pressing the update button. If you have technical problems you have to figure them out or pay for someone else to fix it for you.

Advantages and disadvantages of a self-hosted solution.

Customisation, on the flip side of easy to use, is how much you can customise something. When you have the code you can do anything or you can pay a developer to customise it for you. Do you hate when Facebook changes their policies and all of a sudden your private images, messages and other information are available to the world at large? That can happen with hosted solutions. They can change the rules whenever they want. Your hosted solution could close overnight and you'll be stuck. You don't own the code, you don't have access to your content or any of your information. It's going to be a pretty miserable time to get back up and running. But hosting your own software it's a freedom and if another platform becomes more popular you can stay on your existing platform for as long as you want. And that can also happen with an application like WhatsApp or Viber etc. Down of a self-hosted is that it runs on your server so you will end up maintaining it and fixing any bugs.

4.2 Architecture

Micro-services vs Monolithic architecture

Micro-services splits the application into sub-applications which means the application is going to be divided into processes each process is separate. This will result in the codebase will be easier to manage (each application is independent of the other applications to stay up and running). Therefore, upon doing this successfully you will have the ability to add new features to the application. In a monolithic architecture, the application is a complete unit, it will be hard to add new features especially if you are to develop a large application. Also, deploying the application is easier since you are building the independent micro-services separately and deploying them on separate servers.

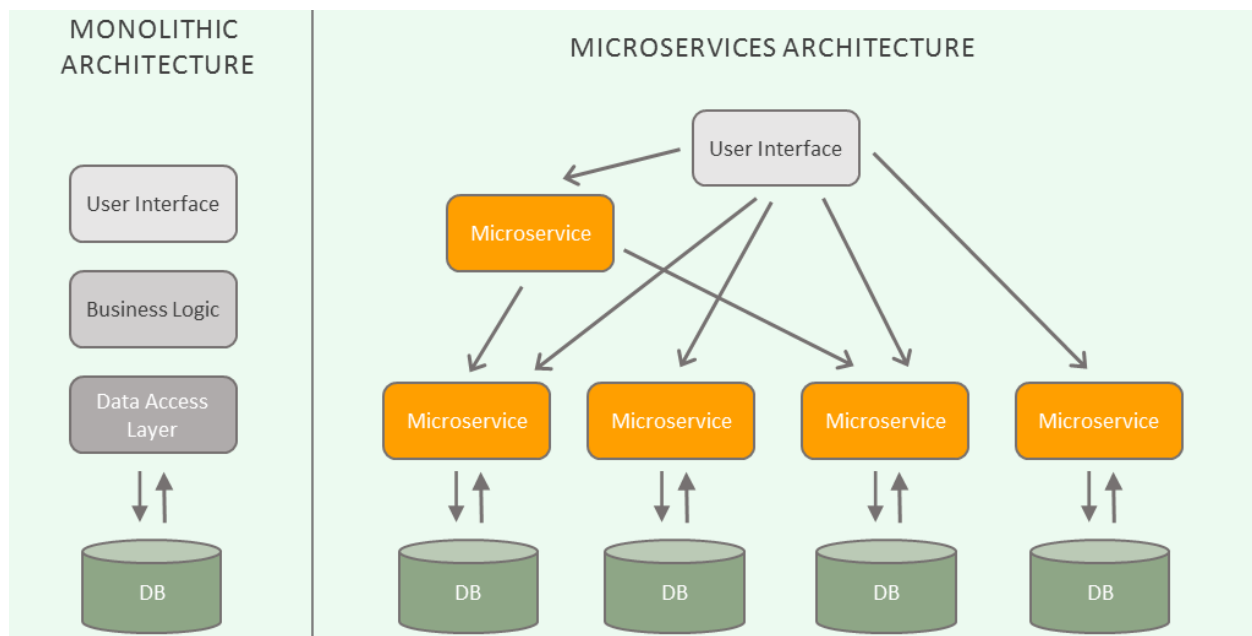


Figure: 4 Microservices vs Monolithic architecture

<http://3gamma.com/insights/architecting-speed-agile-innovators-accelerate-growth-microservices/>

5. Development Tools

5.1 Eclipse



Eclipse [4] is a popular IDE (Integrated Development Environment) commonly used among Java developers. It is a free open source platform tool but it can also be used to develop in many other languages. There are plugins available for the Eclipse IDE that can provide additional functionalities and customisability to enhance the development of applications. Cordova plugin can be added into eclipse to start creating a mobile application.

5.2 Android Studio



Android Studio [5] is an integrated development environment developed by Google like Eclipse in it also open source. It's now the officially integrated development environment (IDE) for android platform development. On May 16, 2013, at Google, I/O conference was announced and it was available under the Apache Licence 2.0. It is developed with android in mind that has all the required packages and tools, unlike eclipse which requires the ADT plugin to develop for android.

Conclusion

Due to the fact that Android Studio is now the officially integrated development environment (IDE) for the Android platform and more features are including such design feature that is not included. So I will be using Android studio to the app.

6. Platforms

This section will list and compare the platform suited for running the application.

6.1 Android

Android [6] is an open source operating system developed for touchscreen devices like smartphones and tablets. It is one of the most popular mobile platforms in use today alongside Apple iOS. The platform is constantly evolving, offering more features and a nice, fluid interface. Updates are made every six months to nine months with the latest being Android 7.0 "Nougat". At, released on August 22, 2016, with Nexus devices being the first to receive the update. Android is used by many companies such Samsung, HTC LG etc. as their main operating system for their newly released devices. With a large community of developer android apps written in java and one of the largest and most well-renowned communities for android development is the XDA Developers forum, which has over 5 million members. According to data from the International Data Corporation (IDC) Worldwide Quarterly Mobile Phone Tracker, over 344.7 million shipments and nearly 87.6% of the market share in 2016 [7].

Earlier versions are becoming outdated as people continue to upgrade their devices. Most devices are using version 4.0 onward so these remain the biggest target audience in the Google Play Store.

| Version | Codename | API | Distribution |
|------------------|-----------------------|-----|--------------|
| 2.2 | Froyo | 8 | 0.1% |
| 2.3.3 - 2.3.7 | Gingerbread | 10 | 1.5% |
| 4.0.3 - 4.0.4 | Ice Cream Sandwich | 15 | 1.4% |
| 4.1.x | Jelly Bean | 16 | 5.6% |
| 4.2.x | | 17 | 7.7% |
| 4.3 | | 18 | 2.3% |
| 4.4 | KitKat | 19 | 27.7% |
| 5.0 | Lollipop | 21 | 13.1% |
| 5.1 | | 22 | 21.9% |
| 6.0 | Marshmallow | 23 | 18.7% |

Figure 5: Android versions distribution

<https://developer.android.com/about/dashboards/index.html>

6.2 iOS

Developed by Apple. The iOS was originally developed only for the iPhone but since more devices have been created by Apple, now runs on the iPad and Apple-TV. IOS was first released in 2007 and today the current version is iOS 9.1. [8]. It requires Apple Macintosh or a laptop with the iOS SDK and Xcode IDE installed. Xcode is the official Apple IDE for developing an application for Apple platforms, much like Eclipse is the official IDE for developing Android apps. Developers cannot test their app on physical devices or publish them to the app store until they pay a fee of \$99.00. OS apps are written in Objective-C, and it's a superset of the C programming language and inherits the syntax, primitive types and flow control statements of C.

6.3 Cordova

Apache Cordova [9] (formerly PhoneGap) is a popular mobile application development framework originally created by Nitobi. Adobe Systems purchased Nitobi in 2011, rebranded it as PhoneGap, and later released an open source version of the software called Apache Cordova. The framework is used to develop mobile applications. Developers use knowledge of HTML, CSS, JavaScript and other web programming languages to build a hybrid application (mix of native app and web app) that can run on IOS or android. However, android uses Java as their main language for developing android app while IOS uses Objective C or Swift. Much like web apps, the entire layout is done using web views instead of the native UI (user interface) but they can still be considered native platforms application and can deploy to the Play Store or the Apple Store. It works just like the native app, allowing device APIs access such interact with the GPS, camera etc. In early days before Apple allows developers create native apps for their device, web app where common and although their popularity has declined in recent years, every so often you will discover an app developed using this framework.

Conclusion

There are many ways of developing an app these days and looking at the statistics from the International Data Corporation (IDC) which shows that Android has a market share of over 80% so it seems the best platform to develop for. 4. X is the most popular, having a share of 78% on all android devices so developing my app to support those versions lower than 4.x will be not viable. I have studied android development in the past and it was little and I would like to improve my android development skills. Also, I think a native app will work a lot better in terms of performance compared to Cordova which is mainly HTMLan HTML. I would like to build a multi-platform or a cross-platform app using Cordova but I don't think Cordova is the right technology for this project. But at a later stage, I might decide to change this technology. Regarding iOS, I think it's a strong option and I would like to learn a bit more IOS development.

7. Technologies

This section will discuss the frontend and the backend of the application. Where the frontend application will focus on user side or the client app and the backend technologies for this project must provide a means of external applications, processing their output and communicating with the database and user interface. Also, will discuss the suitable database system for the application and since the architecture of our system is going to be micro-services architecture, the below technologies will be used to develop the server side.

7.1 Backend technologies

This sub-section is going to discuss the languages, databases and the architecture of the system.

7.1.1 Node.js



Node.js is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine). Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Asynchronous event driven IO helps concurrent request handling.

npm, the Node packaged modules has already become huge and still growing [10].

7.2 Database Systems

The client application will require some data storage mechanism, and the most two common use data storage mechanisms are, object-oriented and relational databases but are database types for each we can look at the most commonly used databases MySQL and Mongo DB and other alternative options that we can look at.

7.2.1 Mongo



Document oriented database also is known as NoSQL databases different in terms of how the data is structured and stored. NoSQL allows much more flexibility at the cost of later search and retrieval performance. These Systems become more popular as in many cases they provide faster write operations, and a better fit for an object oriented programming, especially when used in conjunction with JavaScript Object Notation (JSON), Representational state Transfer (REST) APIs, and JavaScript based applications.

The analysis performed by DB-engine portal and presented in the figure (number) clearly shows that Mongo DB is the most popular document-oriented database and it's also well used with Node.js. In addition, Mongo DB is an open source with a history of stable releases and an established community as well as it supports flexibility, unlike relational databases. Its elastic scalability, which means its scales out on low cost, commodity hardware, and allowing for almost unlimited growth. Another Document-oriented database that I can use is Redis which is an open source (BSD licensed), in-memory data structure store, used as database, cache and message broker. It supports data structures such as strings, hashes, lists, and sets, sorted sets with range queries, bitmaps, hyperlog logs and geospatial indexes with radius queries. Redis comes in third place in figure 5 below and it shows it's widely used.

7.2.2 MySQL

In 1970 E. F. Codd Proposed a relational model that stores the data in tables, which are a natural and easy to understand the way of structuring repetitive data. With the use of primary and foreign keys the data can be stored with a minimal duplication and therefore much higher search efficiency [12]. According to statistics created by db-engines.com presented in figure 5, it shows that six out of the top 10 databases are relational. In addition, there is a significant gap between the top three, which consists of the relational databases only, and the rest of the systems. MySQL and SQLite are the most commonly used relational databases.

MySQL is a popular open-source relational database management system (RDBMS) developed, distributed and supported by Oracle Corporation. Like other relational systems, MySQL stores data in tables and uses structured query language (SQL) for database access. While SQLite is an in-process library that implements a self-contained, server-less, zero-configuration, transactional SQL database engine.

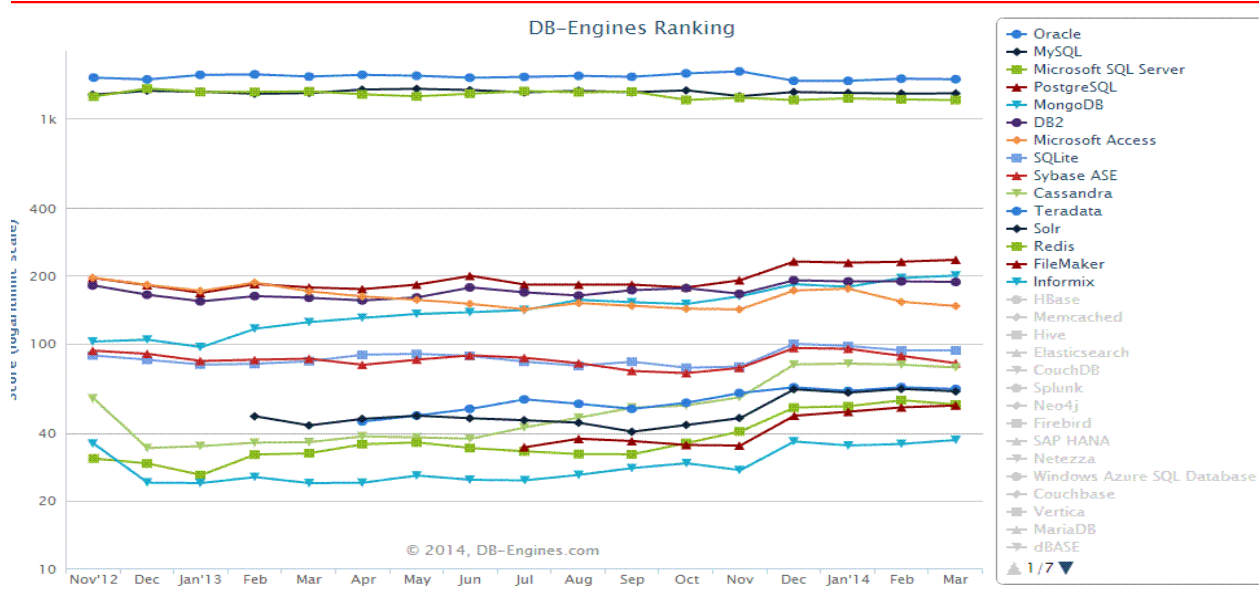


Figure 6 : Top database engines

<http://db-engines.com/en/>

You like without having to rebuild the rest of your application. Since the different services are small and deployed separately, it's obvious easier to scale them, with the advantage that you can scale specific services of your application (with a monolithic you scale the complete "thing", even if it's just a specific part of the application that is getting an excessive load). However, for small applications, you can use monolithic and if you intended to increase the size of the application it's you use monolithic architecture.

Conclusion:

I will be using Mongo DB as the main storage for my app because I have never worked with Mongo DB before and I would like try this technology.

8. Other technologies

The frontend is associated with how the user uses the application or known as UI (user interface) will be created using the technologies (mentioned below).

8.1 Java



<https://java.com/>

Java is a widely used object-oriented programming language and the main programming language used on the Android mobile operating system. Java development for mobile is often done via the Android Studio environment and there are extensive online tutorials detailing how to get started along this development route. Java was not chosen as it would effectively limit application deployment solely to Android devices [13].

9. Messaging protocols

This section will describe the different type of messaging protocols that can be used in this project. Also, will discuss how these protocols work and what are the advantages and disadvantages of using these protocols.

9.1 Websockets

WebSocket is a computer communications protocol, providing full-duplex communication channels over a single TCP connection. WebSocket is designed to be implemented in web browsers and web servers, but it can be used by any client or server application. The WebSocket Protocol is an independent TCP-based protocol [14]. In this project I will be a concern will Socket.io. Socket.io is a JavaScript Library that enables communications between and clients and servers. Socket.io is represented by the figure (7). The difference between WebSocket and HTTP is WebSocket is a protocol providing full-duplex communication channels over a single TCP connection, where HTTP provides half-duplex communication.

Information exchange mode of WebSocket is bidirectional. Means, the server can push information to the client (which does not allow direct HTTP).

Architecture

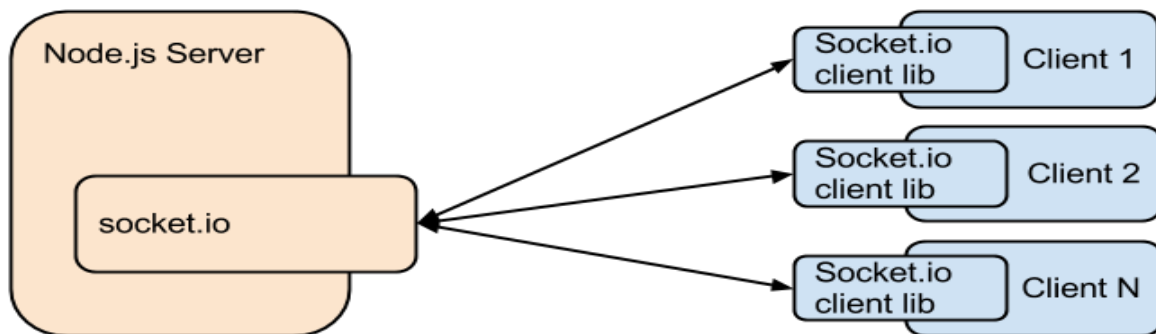


Figure: 7

<http://blog.lightstreamer.com/>

9.2 XMPP



Extensible Messaging and Presence Protocol (XMPP) is a communications protocol for message-oriented middleware based on XML (Extensible Markup Language) [15].

This type of messaging protocol is used by the famous messaging app (WhatsApp).

Conclusion:

Bases on the above the research that regards which messaging protocol to use I found WebSocket as a good starting point. Due to its simplicity and numbers of tutorials that are available, there I found WebSocket (socket.io) is the number one choice.

Although XMPP remains a great choice due to the fact it's a powerful and as we know that an app like “WhatsApp” uses that technology but I thought using WebSocket, for now, is a good choice because the project type is very light. Any changes to this choice I will update later.

10. Conclusion

1. This research manual focuses on technologies and knowledge required to develop a self-hosted application. The platform chosen for the development of this application is an android application, which will develop using Java.
2. To accomplish this, the application will be developed using full software production pipeline techniques.
3. The technologies to be used will be used is Java, chosen as a native development language for developing Android applications.
4. Node.js chooses a server-side language for its rising popularity and large community.
5. The database chosen for this development of this application is Mongo database because it works well node.js. MySQL has a lot more rigidness and pre-defined relationships up front. So you have NoSQL (Mongo database) gives you much more flexibility initially and you can fire off data to store (and retrieve) without needing to worry about that side.
6. Websocket protocol (socket.io) will be used as an instant message protocol for communication between client and the server.
7. The user interface will be developed using XML, which easy to parse and manipulate programmatically, it's basically a tree structure and most UI creation tools already use it. It has nothing to do with decoupling business logic because you can define in the Java code, also the UI will be delivered to reach the target audience for better user experience. Other necessary technologies and frameworks will be discovered later.

10. References

- [1] computer.howstuffworks. 2016. History of Instant Messaging. [ONLINE] Available at: <http://computer.howstuffworks.com/e-mail-messaging/instant-messaging1.htm>. [Accessed 24 October 2016].
- [2] mattermost. 2016. mattermost. [ONLINE] Available at: <https://www.mattermost.org>. [Accessed 26 October 2016].
- [3] howdoesappingwork. 2016. How does Whatsapp Work?. [ONLINE] Available at: <https://www.howdoesappingwork.com/whatsapp/>. [Accessed 26 October 2016].
- [4] Notes, Eclipse ADT. 'ADT Plugin Release Notes | Android Developers'. Developer.android.com. N.p., 2014. Web. 28 Oct. 2016.
- [5] Wikipedia,. 'Android Studio'. N.p., 2014. Web. 6 Nov. 2016
- [6] Wikipedia,. 'Android (Operating System)'. N.p., 2016. Web. 3 Nov. 2016.
- [7] idc. 2016. Smartphone OS Market Share, 2016 Q2. [ONLINE] Available at: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>. [Accessed 24 October 2016].
- [8] wikipedia. 2016. iOS. [ONLINE] Available at: <https://en.wikipedia.org/wiki/iOS> . [Accessed 1 November 2016].
- [9] Wikipedia. 2016. Apache Cordova. [ONLINE] Available at: https://en.wikipedia.org/wiki/Apache_Cordova. [Accessed 1 November 2016].
- [10] nodejs. 2016. node.js. [ONLINE] Available at: <https://nodejs.org/en/>. [Accessed 1 November 2016].
- [11] firebase.2017. [ONLINE] Available at : <https://firebase.google.com/>. [Accessed 1 February 2017].
- [12] Wikipedia. 2016. Relational database. [ONLINE] Available at: https://en.wikipedia.org/wiki/Relational_database. [Accessed 1 November 2016].
- [13] wikipedia. 2016. HTML. [ONLINE] Available at: [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language)) . [Accessed 1 November 2016].
- [14] Websocket. 2013. [ONLINE] Available at: <https://en.wikipedia.org/wiki/WebSocket>. [Accessed 4 November 2016]
- [15] Wikipedia. 2013. XMPP (instant messaging protocol). [ONLINE] Available at: <https://en.wikipedia.org/wiki/XMPP> . [Accessed 4 November 2016].