



---

# EMA

---

Matthew Mason: C6122243



## Table of Contents

PART 1 – CONTRASTING APPROACHES TO MOBILE DEVELOPMENT.....	2
PART 2 – MOBILE WEB APP .....	4
A.....	4
B.....	5
C.....	5
D.....	6
USER GUIDE:.....	6
BUILD GUIDE:.....	9
PART 3 - DEVELOPING THE APPS.....	11
Recommendation.....	11
Legal issues and Security risks.....	12
Storage, backup and maintaining accuracy and consistency.....	13
Scaling and Supporting application reliability.....	14
References.....	15

## PART 1 – CONTRASTING APPROACHES TO MOBILE DEVELOPMENT

	<i>Native</i>	<i>Web Based</i>	<i>Hybrid</i>
<i>Languages</i>	iOS: Swift/Objective-C Android: Java/Kotlin/C++ requires a wider specialised knowledge.	Web based technologies: HTML/CSS/JavaScript.	HTML/CSS/JavaScript: deployment of native code on multiple devices from a single code base.
<i>Hardware and data access</i>	Full access: can access all device hardware and storage, provides instant use of new features and technologies, and fastest performance.	Limited access: can use APIs to access hardware and web storage APIs to maintain session and persistent storage. Slowest of the three options.	Limited access: Uses JavaScript APIs core plugins to access hardware and storage and npm provides third-party plugins for other technologies such as VR, AR, AI. However, these can take time to develop and may not be available as soon as the technology is ready. Moderate response times.
<i>Development Support</i>	SDK and IDE tools for compiling native code, (Android studio/Xcode), testing, income generation, Android/Apple developer documentation.	A range of IDEs and text-based applications such as VS Code, Chrome Dev Tools, MDN Web docs, Git and GitHub, Openstack, Codecademy & Udemy courses, W3C standards, etc	Apache Cordova, Ionic, etc for compiling applications, Node.js, node package manager for plugins, Reactive Native, Flutter, JQuery UI, IDEs and text-based applications, Chrome Dev Tools plus all their related documentation.
<i>Publishing &amp; distribution</i>	Apple: Yearly fee (\$99), secured marketplace, strict guidelines taking 10-14 days before app/updates published. Android: one-off fee (\$25), published in a few hours, all apps require downloading & installation, apps only work on specified device.	Apps can't be downloaded so no need for an app store, however, a constant internet connection is required for access in most cases. Can use GitHub for simple testing and user projects. For enterprise web applications, a domain name is needed, a hosting provider or cloud hosting provider such as AWS. Can be updated without user having to download patches.	Very similar to Native experience except that strict guidelines must be followed for Apple and non-native apps can be denied as they cannot comply with guidelines for both operating systems. Launch faster than web apps and provide quality control through app store feedback.

<i>UI/UX</i>	Device specific interface controls, icons, animations, material design and standards providing familiarity, a professional feel and an improved, consistent learning experience.	Requires a responsive/progress design to work across a range of devices, provides consistent look and feel. Slower response times can lead to a more negative view of the user experience. The overhead of translating JavaScript into native instructions decreases performance efficiency.	A native look can be attempted by using libraries such as React Native, Onsen UI and JQuery UI but this adds complexity, and may add additional cost equal to developing separate native apps. Moderate performance provides good experience compared with web apps.
<i>Security &amp; Privacy</i>	Security advantages via strict app guidelines, restriction of permissions to those needed for the app to function.	Source code is available to view by anyone and securing communication channels between app and server complicates development, requiring CA certification and various encryption methods increasing cost.	The container that gives apps the same capabilities as Native apps can expose some features of the underlying platform. However, this does not make it less secure, it simply means more work must be taken to ensure the app provides necessary security features which can add to the cost. 3 <sup>rd</sup> party plugins may not be coded well and expose access to sensitive data.

Table 1: Contrasting Development Approaches

(Nunkesser, 2018), (Francese, et al., 2017), (Xanthopoulos & Xinogalos, 2013), (Charland & Leroux, 2011), (Lastovetska, 2021), (Buck, n.d.), (Google, 2023), (Apple, 2023), (Onegini, 2022), (Network, 2023)

## PART 2 – MOBILE WEB APP

A.

<b>FR</b>	<b>Name</b>	<b>Description</b>	<b>Obstacle</b>
<b>FR1</b>	Promotion	Salespersons will be able to apply a unique discount as a percent of whole total or specific amount.	The salesperson may be able to apply a discount that causes the sale to be made at a loss to the business by confusing the various types of applicable discounts.
<b>FR2</b>	New Client	Salespersons can add a new client to the database whilst in the field.	Salespersons may not have permission to adjust internal client databases from outside premises due to security risks.
<b>FR3</b>	Nearest Client	Salespersons can use geolocation services to identify the nearest client as the crow flies.	GPS signal may not be available if the user is in a remote location, inside a secure location that blocks signals, or the network is down.
<b>FR4</b>	Search	Salespersons can use search functionality to quickly identify a widget from the database as opposed to using forward and back buttons.	Repeated requests and traffic spikes from API calls may result in network congestion, slow response times and failure to provide appropriate responses.

*Table 2: Table of additional functional requirements for MegaMax's Sales app.*

B.



Figure 1: Mock-Up of MegaMax's Sales App

C.

see attached www folder.

## D

### USER GUIDE:

The screenshot shows the 'MegaMax Sale' app interface. At the top, there's a title 'MegaMax Sale'. Below it are three input fields: 'User ID' with the value 'i9a00252', 'Password' with a masked value '.....', and 'Client ID' with the value '1'. A 'Begin New Order' button is below these fields. The main content area features a large image of a brass self-tapping wood screw. To the left and right of the image are red circular buttons with white arrows pointing left and right. Below the image is a 'Widget Id' field with the value '1', followed by the text 'Brass, self tapping wood screw, 20mm'. Below this is an 'Agreed Price' field with '10' and a 'Quantity' field with '1'. An 'Add to Order' button is positioned below the price and quantity fields. At the bottom of the main content area is a 'Place Order' button. Below the main content area is a map with an 'Update Map' button.

Figure 2: Initial layout upon launching.

Figure 2 displays the initial layout of the app upon launching and loads the database of available widgets. To place an order, you must enter your user-ID, Password, and the client-ID. If no user-ID is supplied, an alert displays specifying these are required parameters and must be entered before you can continue (Figure 3 below).

Pressing 'Begin New Order' will open a new order for the specified client and display the time the order was created, the client's name, address, basket items and cost breakdowns in pounds (Figure 4).

You can navigate through the list of widgets using the arrow buttons either side of its image which will update its ID, price in pence and description. Once a price and quantity have been agreed upon, the 'Add to Order' button will add the item to the basket along with the agreed amounts and the subtotal, VAT and order total will be updated in pounds and pence (Figure 5 & 6).

The 'Place Order' button adds all the items to the current order, clears all data fields and re-initialise the layout to that of figure 2.

The map displays your current position, the 'Update Map' button centres the map on your current location, placing markers at sites where orders have been placed, centring on the most recent order.

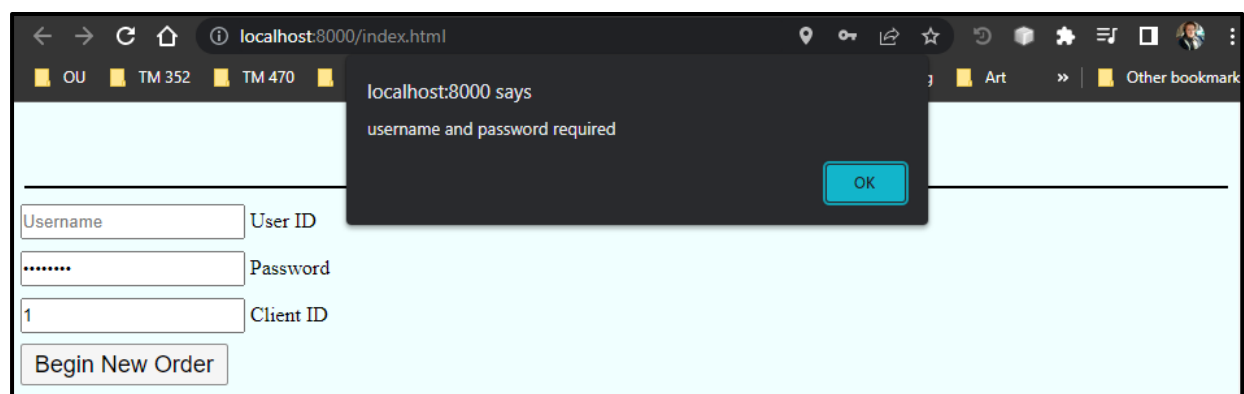


Figure 3: Alert for nonempty User ID field.

EMA

BSC (HONOURS) COMPUTING AND IT: TM-352

Matthew Mason: C6122243



Figure 4: A new order displaying the time the order was placed, client, address and items in basket.



Figure 5: Adding multiple items to the basket and updating the totals.





Figure 6: Adding multiple items to the basket and updating the totals.

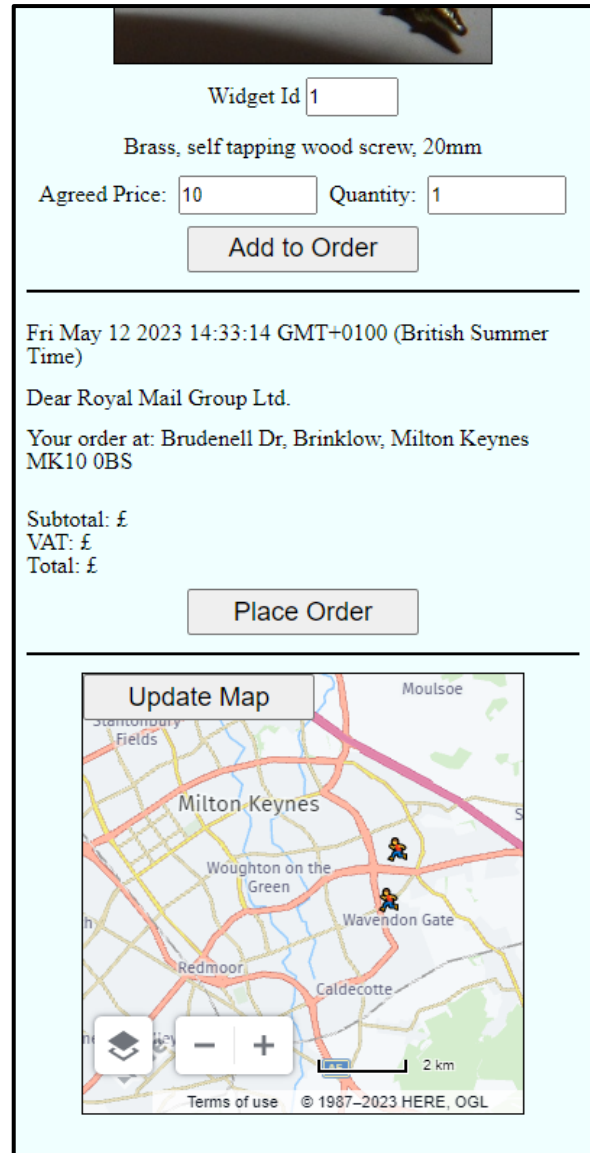


Figure 7: Map displaying order locations.

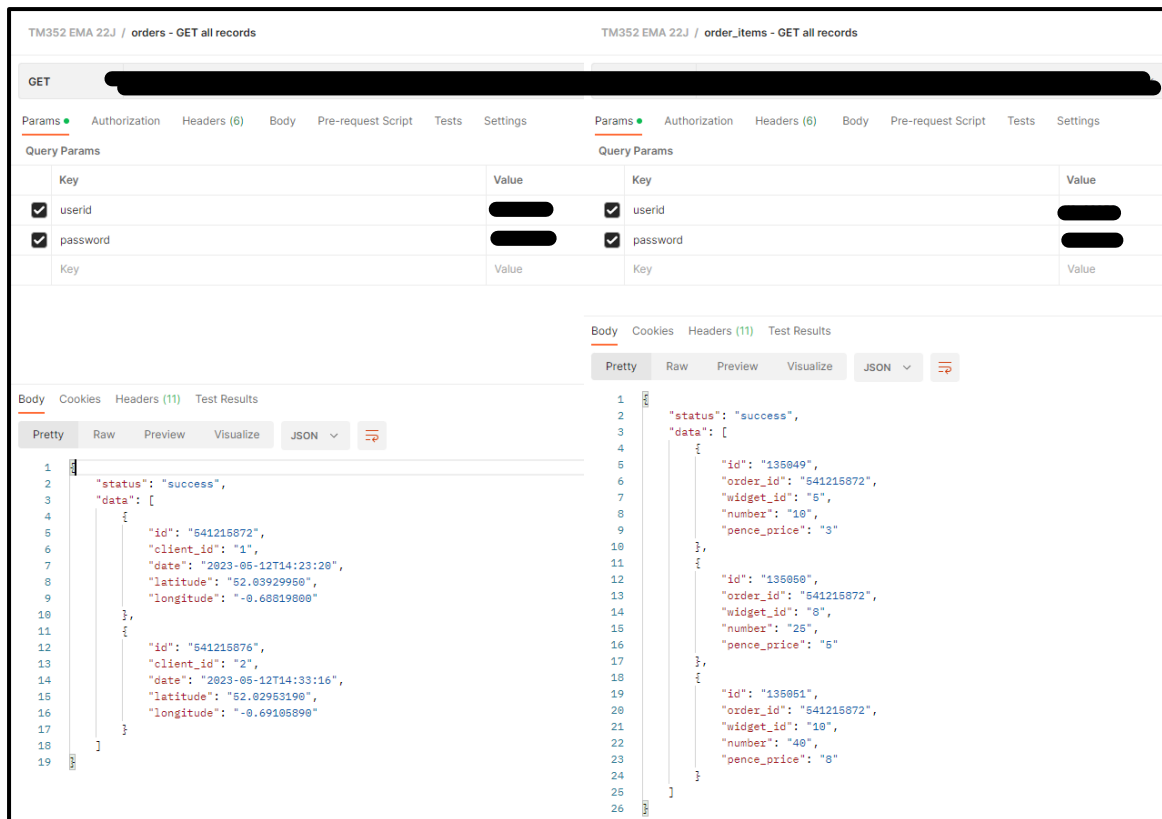


Figure 8: Using POSTMAN to check workings.

## BUILD GUIDE:

There are number of prerequisites before MegaMax’s sales app can be configured and deployed to staff and this guide will focus on android devices.

1. Download and install the appropriate LTS version of Node.js for your operating system. Add its location to your PATH variable. You should also add items to the path that will be required later (these can be installed in any location of your choice, we assume C drive for simplicity):  
C:\cmdline-tools\tools\bin  
C:\platform-tools  
C:\gradle\bin  
C:\emulator
2. Use npm (packaged with Node.js) via the cmd to install Cordova, version 10 globally: “npm -install -g cordova@10.0.0”.
3. create the sales app project directory, navigate to it within cmd and create a cordova project: “cordova create salesApp”.
4. From within your Cordova project, add any platforms you require: “cordova platform add android”.
5. Install Android SKD manager command line tools, extract the content and rename the files and location to match the path in part 1, then use cmd to install the following:  
“sdkmanager platform-tools”  
“sdkmanager platforms;android-29”  
“sdkmanager build-tools;29.0.3”

EMA

BSC (HONOURS) COMPUTING AND IT: TM-352

Matthew Mason: C6122243

6. Download and install gradle version 4.3.1, rename and move to the directory matching part 1.
7. Java version 1.8 is required, and other versions should be removed.

From here you can add additional platform for browsers or iOS and you will need to enable USB debugging via android developer options to deploy to an android device.

As the RESTful API uses http, the installation of a plugin to allow unencrypted services is required and installed via the command-line whilst in the cordova project directory: "cordova plugin add cordova-plugin-enable-plaintext-traffic".

The app uses HERE maps so an API key will need to be generated to make use of this feature otherwise the maps won't work.

## PART 3 - DEVELOPING THE APPS

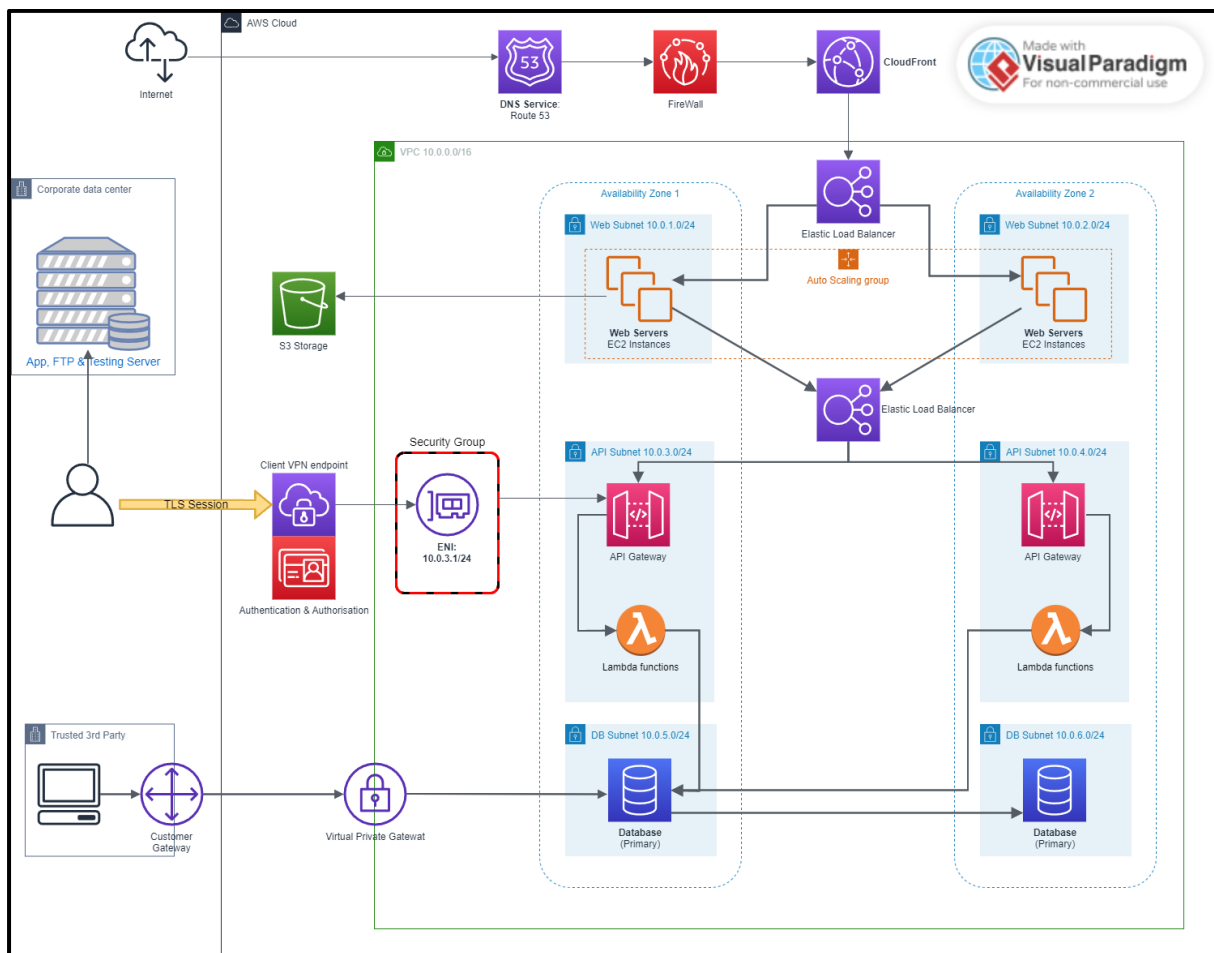


Figure 9: Cloud solution Architecture. (AWS, 2022), (Amazon, 2022), (Amazon, 2010), (Amazon, 2023),

### Recommendation

My recommendation for MegaMax is to utilise the AWS architecture in figure 9 to migrate the majority of their services to the cloud. This is very similar to the architecture in TMA02 with some small changes made such as the removal of the App servers and replacing them with an API gateway to manage endpoints and better utilise the cloud. The cloud provides multiple benefits; automatic horizontal scaling policies via elastic load balancer monitoring that adapts resource availability to traffic flow, reducing costs, capital expenditure gains for reinvestment in app and website development, equipment maintenance is provided for by the cloud provider along with N+1 fault tolerance redundancy across multiple availability zones and 99.99% SLAs to name a few. The API gateway configuration provides RESTful endpoints along with Lambda functions for a serverless environment, reducing costs further, callable to remote sales staff and from the website. Existing equipment can be repurposed to host the sales application, downloadable via FTP to staff, servers could provide development and testing environments before being pushed to the cloud and databases can be used for offsite backup if needed or another purpose. Alternative methods include a hybrid architecture using on premises equipment to configure OpenStack and use AWS for cloud scaling, however this would require significant overheads in training, configuration, maintenance, etc, onto two separate infrastructures, greatly increasing complexity and costs to invest in addition

EMA

BSC (HONOURS) COMPUTING AND IT: TM-352

Matthew Mason: C6122243

internet capacity, staff and equipment; similar problems would occur using a completely onsite OpenStack approach. While there are a number of risks and factors that affect cloud adoption such as security and multi-tenancy infrastructure, discussed below, the benefits outweigh the risks and an analysis using a taxonomy migration task framework can be utilised to build a comparison for estimating cost-benefits over traditional infrastructure expansion. As the complexity of the app and website is relatively low, and cloud services provided are very similar to the local servers and database, migration shouldn't require high overheads. (Tran, et al., 2011), (Gunka, et al., 2013), (AWS, 2022), (AWS, 2022).

I would also recommend the redesign of the website transitions from using JavaServer Faces to HTML, CSS and JavaScript plus other necessary frameworks and at the same time developing the sales app using a hybrid approach, the same languages and Cordova. This provides an easier maintenance environment as all sales apps uses a single code base, a similar look and feel across all devices with the ability to provide a native feel using select libraries and frameworks, however this does require more effort, time and resources to achieve, access to device hardware through plugins and deployment across a range of different devices. Developing native apps would require learning Swift or Objective-C for iOS, and essentially doubles maintenance and update/development efforts and alternatively, web applications can provide similar services but can be slow and require a constant internet connection to access and use, so I would not recommend either of these two approaches. The use of JavaScript can also be used for front-end interactivity and back-end request and response handling, such as using Node.js as the Lambda function language with API calls. Whilst the recommendations provide the best opportunity for MegaMax to grow their business in terms of availability, cost and sales, there are additional issues which need to be considered as part of the solution.

### Legal issues and Security risks

Measures taken to manage the security and legal risks of storing and transferring customer, client, and staff details as well as sensitive financial transactions and bank details can be handled in numerous ways such as encryption, creating VLANs to segment multitenancy infrastructure, and adhering to GDPR and DPA principals.

Financial transaction processing can be outsourced to Payment Card Industry (PCI), Data Security Standard (DSS) compliant 3<sup>rd</sup> parties to help mitigate risks. These companies help secure transaction data, managing the whole process from merchant to the bank through payment gateway technology. (PCI Security Standards Council, 2022), (emerchantpay, 2022). MegaMax would need to ensure their website is secure by using TLS/SSL certification meaning all transactions would be secure and encrypted. The certification is paid for by MegaMax and provided to them from a Certificate Authorities Organization that validates their public key, fully qualified domain name and company details. It needs to be installed on the server and is used to authenticate the organisation. To utilise encryption and secure data transmission, communicating parties such as clients' browsers would establish data exchange parameters with cloud servers via pre-installed crypto-graphic encryption algorithms, commonly, block ciphers such as Advanced Encryption Standards (AES) or Tripe Digital Encryption Standards (DES), then an encrypted symmetric session key can be used to provide a more efficient method of communication. User credentials can be configured on the server to provide client and staff authentication using encoded user-identifiers and passwords, cryptographically generated OAuth 2.0 bearer tokens with included timestamps or nonce values to check for freshness, and hash values to provide integrity and evidence no data alteration has occurred.

EMA

BSC (HONOURS) COMPUTING AND IT: TM-352

Matthew Mason: C6122243

The most cost effective and beneficial way to provide security in a multi-tenancy environment is to configure VLANs for each tenant. This uses a unique ethernet packet header tag to direct traffic to each instance providing isolation and segregation of data. Firewalls can then be used to screen any packets and assures VMs running on the same host can't gain privileged access to another.

As previously stated in TMA 02, "Cloud networks are accessible by anyone on the internet, therefore static data needs to be encrypted which can be done using symmetric keys. Using a single key for each tenant for decryption at scale is impractical so cloud providers offer a key management infrastructure that combines securely storing keys and authorising key usage. It means tenants don't have to write any additional code to manage these keys which requires considerable experience to ensure it would be fit for purpose. Penetration testing consultants can also be employed to test the security of infrastructures to eliminate vulnerabilities and ensure legal and privacy compliance regulations are being met. These practices should enable MegaMax to be confident that data is secure in transit and at rest whilst also adhering to General Data Protection Regulations (GDPR) (EU, 2018) and Data Protection Act (DPA) principals" (Parliament, 2018). There are also a range of security services that can assist with resources access at multiple levels such as AWS Identity and Access Management.

### Storage, backup and maintaining accuracy and consistency

There are multipurpose storage options available across cloud providers and to alleviate security concerns and assure confidentiality and unauthorised disclosure of data isn't possible, storage media has to be sanitized in accordance with the NIST publication 800-88 (NIST, 2014). The choice of storage also needs to be appropriate for MegaMax's intended use and as we can see in figure 9 the two main services are Amazon's Simple Storage Service and a MySQL database; the equivalent in a local infrastructure would simply be server hard drives and databases. There is also the use of the CloudFront service to cache frequently requested data resulting in quicker responses back to clients.

As the MegaMax sales application utilises a MySQL relational database, the equivalent cloud service needs to reflect this model to ensure migration complexity isn't increased. The Amazon relational database service provides the ability to configure a managed database service whilst providing a range of features such as scalability, automation of tasks, backups and recovery (Amazon, 2023). It also provides asynchronous replication abilities across multiple availability zones or regions for handling excess read requests, management and reporting tasks and disaster recovery by promotion of a replica to a standalone database instance (Amazon, 2023). These replica databases can also be exported externally from AWS to a local server for on premises backup. These features along with regular maintenance, optimisation and careful consideration of table and schema design, data validation and normalisation can help ensure data accuracy, integrity and consistency.

For medium term solutions, the RDS MySQL database provides all the necessary functionality and services to provide MegaMax with a reliable storage service and should they require an increased capacity, higher performance, additional services or decide on the need for a serverless environment, there is the options of using Amazon Aurora to handle these requirements.

The simple storage service (S3) provides storage for website resources in the form of HTML, CSS, JavaScript, images, order history, salespeople data, order forms, and any other resources necessary for the website and applications operation. It also provides scalability, security and 99.9999999999% durability. The S3 service can also help with MegaMax's backup concerns by utilising the AWS Data Sync service to automatically connect and transfer data via an on-premises

EMA

BSC (HONOURS) COMPUTING AND IT: TM-352

Matthew Mason: C6122243

software agent connected to the S3 service, to replicate data to local servers for backup. The CloudFront and S3 services can also work together to deliver content with low-latency and high transfer speeds, ideal for MegaMax's needs. Alternative storage options are available from providers, but these are aimed at specific use cases whereas the S3 service provides lowest cost requirements and a range of different storage classes depending on access frequency (Amazon, 2022).

### Scaling and Supporting application reliability

One of the clouds greatest advantages is its ability to respond to demand and this will have a major impact on MegaMax's website and application reliability. Rather than investing in traditional hardware, effectively overprovisioning and underutilising it, the cloud can respond dynamically to demand and supply capacity only when required. This means capital expenditure can be spent elsewhere and be used more effectively rather than having to be used on factors such as physical rack space, switches, internet access, power supply units, rent, employees, installation, redundancy, etc.

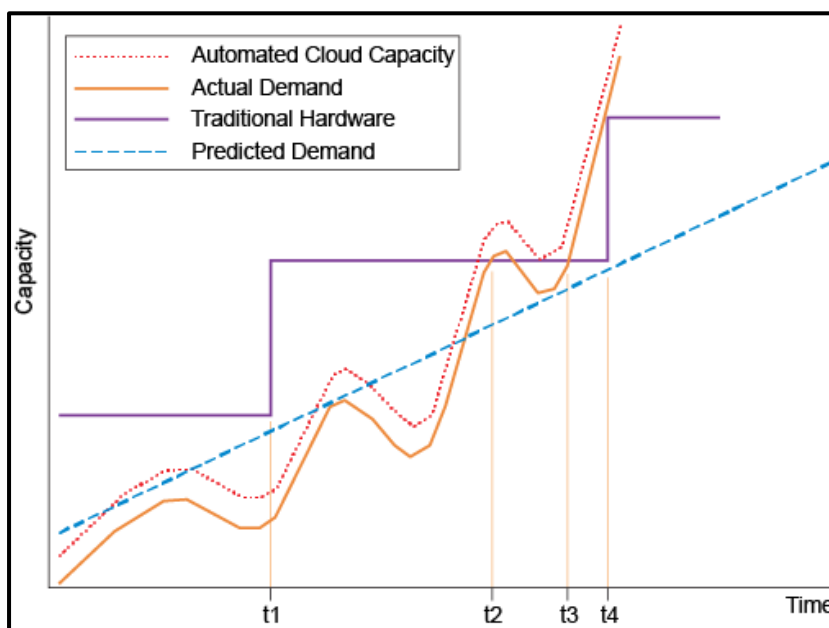


Figure 10: Comparison of demand and capacity for traditional and cloud deployments.

This response can be configured to provision resources automatically for optimal utilisation across multiple availability zones through horizontal scaling policies so when a threshold is reached, servers are commissioned and decommissioned dynamically based on traffic and other factors. This scalability also applies to API calls and Lambda functions concurrency which can be increased upon request (Amazon, 2023), (AWS, 2023).

This scalability will provide MegaMax with the assurance that no matter the amount of traffic, staff will have reliable access to the data they need to make sales, and clients and customer will be able to complete purchases without any hinderances. Cloud providers can also guarantee continuous operation through service level agreements to ensure services are running with a limited amount of downtime, redundancy options are available to help increases reliability and availability, meaning services are accessible at any time from any device for both MegaMax and their customers. Finally, it provides MegaMax with the opportunity to grow their business along with their staff, client and customer's needs.

## References

- Amazon, 2010. *Web Application Hosting in the AWS Cloud*. [Online]  
Available at: <https://d0.awsstatic.com/whitepapers/aws-web-hosting-best-practices.pdf>  
[Accessed 06 01 2023].
- Amazon, 2022. *AWS Client VPN | Remote access VPN | Amazon Web Service (AWS)*. [Online]  
Available at: <https://aws.amazon.com/vpn/client-vpn/>  
[Accessed 06 01 2023].
- Amazon, 2022. *Overview of Amazon Web Services - AWS Whitepaper*. [Online]  
Available at: <https://docs.aws.amazon.com/pdfs/whitepapers/latest/aws-overview/aws-overview.pdf#introduction>  
[Accessed 06 01 2023].
- Amazon, 2023. *Amazon API Gateway - Build and Run Scalable Application Backends*. [Online]  
Available at: <https://aws.amazon.com/blogs/aws/amazon-api-gateway-build-and-run-scalable-application-backends/>  
[Accessed 03 June 2023].
- Amazon, 2023. *Fully Managed Relational Database - Amazon RDS - Amazon Web Services*. [Online]  
Available at: <https://aws.amazon.com/rds/>  
[Accessed 03 June 2023].
- Amazon, 2023. *Multi-region API Gateway with CloudFront*. [Online]  
Available at: [https://docs.aws.amazon.com/architecture-diagrams/latest/multi-region-api-gateway-with-cloudfront/multi-region-api-gateway-with-cloudfront.html?did=wp\\_card&trk=wp\\_card](https://docs.aws.amazon.com/architecture-diagrams/latest/multi-region-api-gateway-with-cloudfront/multi-region-api-gateway-with-cloudfront.html?did=wp_card&trk=wp_card)  
[Accessed 31 May 2023].
- Amazon, 2023. *Working with DB instance read replicas*. [Online]  
Available at: [https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER\\_ReadRepl.html](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ReadRepl.html)  
[Accessed 03 June 2023].
- Apple, 2023. *Featured | Apple Developer Documentation*. [Online]  
Available at: <https://developer.apple.com/documentation>  
[Accessed 05 May 2023].
- AWS, 2022. *Amazon Compute Service Level Agreement*. [Online]  
Available at: [https://aws.amazon.com/compute/sla/?did=sla\\_card&trk=sla\\_card](https://aws.amazon.com/compute/sla/?did=sla_card&trk=sla_card)  
[Accessed 13 12 2022].
- AWS, 2022. *Data Centers - Our Controls*. [Online]  
Available at: <https://aws.amazon.com/compliance/data-center/controls/>  
[Accessed 13 12 2022].
- AWS, 2022. *Reference Architecture Examples and Best Practices*. [Online]  
Available at: [https://aws.amazon.com/architecture/?wasn\\_achp1&cards-all.sort-by=item.additionalFields.sortDate&cards-all.sort-order=desc&awsf.content-type=\\*all&awsf.methodology=\\*all&awsf.tech-category=\\*all&awsf.industries=\\*all](https://aws.amazon.com/architecture/?wasn_achp1&cards-all.sort-by=item.additionalFields.sortDate&cards-all.sort-order=desc&awsf.content-type=*all&awsf.methodology=*all&awsf.tech-category=*all&awsf.industries=*all)  
[Accessed 12 12 2022].

EMA

BSC (HONOURS) COMPUTING AND IT: TM-352

Matthew Mason: C6122243



- AWS, 2023. *Lambda function scaling - AWS Lambda*. [Online]  
Available at: <https://docs.aws.amazon.com/lambda/latest/dg/lambda-concurrency.html>  
[Accessed 03 June 2023].
- Buck, A., n.d. *Native Apps, Web Apps or Hybrid Apps? What's the Difference?*. [Online]  
Available at: <https://www.mobiloud.com/blog/native-web-or-hybrid-apps>  
[Accessed 05 May 2023].
- Charland, A. & Leroux, B., 2011. Mobile application development: Web vs. Native. *Communications of the ACM*, May, 54(5), pp. 49 - 53.
- emerchantpay, 2022. *What is a Payment Gateway and How Does it Work?*. [Online]  
Available at: <https://www.emerchantpay.com/insights/what-is-a-payment-gateway-and-how-does-it-work/>  
[Accessed 13 12 2022].
- EU, 2018. *General Data Protection Regulation (GDPR) - Official Legal Text*. [Online]  
Available at: <https://gdpr-info.eu/>  
[Accessed 13 2022 2022].
- Francese, R., Gravino, C., Risi, M. & Scanniello, G., 2017. *Mobile app development and management: results from a qualitative investigation*. Buenos Aires, Argentina, IEEE Press, pp. 133 - 143.
- Google, 2023. *Documentation | Android*. [Online]  
Available at: <https://developer.android.com/docs>  
[Accessed 05 May 2023].
- Gunka, A., Seycek, S. & Kuhn, H., 2013. *Moving an Application to the cloud - An Evolutionary Approach*. Prague, ACM, pp. 35 - 42.
- Lastovetska, A., 2021. *7 Pros of Native App Development vs. Hybrid & Web Apps*. [Online]  
Available at: <https://mlsdev.com/blog/native-app-development-vs-web-and-hybrid-app-development>  
[Accessed 05 May 2023].
- Network, M. D., 2023. *Web Storage API - Web APIs | MDN*. [Online]  
Available at: [https://developer.mozilla.org/en-US/docs/Web/API/Web\\_Storage\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API)  
[Accessed 05 May 2023].
- NIST, 2014. *Guidelines for Media Sanitization*. [Online]  
Available at: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-88r1.pdf>  
[Accessed 02 June 2023].
- Nunkesser, R., 2018. *Beyond web/native/hybrid: a new taxonomy for mobile app development*. Gothenburg, Sweden, ACM, pp. 214 - 218.
- Onegini, 2022. *Hybrid or Native: But what about security?*. [Online]  
Available at: <https://blog.onegini.com/native-vs-hybrid-apps>  
[Accessed 05 May 2023].

Parliament, U., 2018. *Data Protection Act 2018*. [Online]  
Available at: <https://www.legislation.gov.uk/ukpga/2018/12/part/4/chapter/2/crossheading/the-data-protection-principles/enacted>  
[Accessed 13 12 2022].

PCI Security Standards Council, 2022. *PCI\_DSS-QRG-v4\_0.pdf*. [Online]  
Available at: [https://docs-prv.pcisecuritystandards.org/PCI%20DSS/Supporting%20Document/PCI\\_DSS-QRG-v4\\_0.pdf](https://docs-prv.pcisecuritystandards.org/PCI%20DSS/Supporting%20Document/PCI_DSS-QRG-v4_0.pdf)  
[Accessed 13 12 2022].

Tran, V., Keung, J., Liu, A. & Fekete, A., 2011. *Application Migration to Cloud: A Taxonomy of Critical Factors*. Honolulu, ACM, pp. 22-28.

Xanthopoulos, S. & Xinogalos, S., 2013. *A comparative analysis of cross-platform development approaches for mobile applications*. Thessaloniki Greece, ACM, pp. 213 - 220.