DSU Blockchain

Student Name: King Him Cheung

Supervisor Name: Dr. Ivrissimtzis

Submitted as part of the degree of MEng Computer Science to the

Board of Examiners in the Department of Computer Sciences, Durham University

Abstract —

Context/Background

Blockchain technology in recent years has been trending in popularity to the extent that many organisations have dedicated resources to its development. The Blockchain technology itself promises the decentralisation of resources. Companies have proposed their own schemes to further enhance this technology and are currently still providing updates and further improvements in the hopes of delivering a practical and impactful product

Aims

The aim of this project is to research a modern blockchain technology and utilise it to produce an application for the student body and student groups to enable them to perform their weekly activities without the need of a centralised body. Furthermore, we will apply benchmarking tests to measure performance, in combination with user surveys to gather data for the consideration the suitability of blockchain.

Method

We will implement a blockchain network using Hyperledger Fabric and develop a Hyperledger Composer business network application layer in which a REST API can be formed from. This REST API is used to produce our Angular application which will act as the user interface for student and student groups. Benchmark tests will be undergone via Hyperledger Caliper and a survey will be created for gathering user feedback.

Results

From our results we have demonstrated Hyperledger Fabric's capabilities in enabling a satisfactory software application and thus giving us a generalisation of modern private blockchains. We see there is flexibility in network performance relying largely on hardware resources. Nevertheless showing capabilities of better performance than public blockchain technologies.

Conclusion

Hyperledger provides suitable software development tools to produce viable blockchain applications, where even performance can be managed with additional resources. However, justification is required to give a convincing argument on the need to use blockchain instead of another more established technology. Nevertheless, as demonstrated from user feedback the underlying blockchain technology can be made viable from some scenarios.

Keywords — Blockchain, Hyperledger Fabric, Hyperledger Composer, Hyperledger Caliber, Angular, REST API, Business Network

I INTRODUCTION

This project considers the possible practical uses of blockchain technology and in particular focus on it's potential to either enhance the current Durham Student Union (DSU) technologies or at least provide an alternative option with considerations of the benefits and downfalls of this blockchain variant. Overall, we will produce a user interface for both Students and Societies. From this, we will gather information on how current blockchain technologies may impact future trends in the incorporations of such technologies in standard software infrastructures.

A Project Domain

The blockchain can be defined as a chain of data blocks of which contains details about transactions. These blocks have identifiers in the form of a hash. Moreover, this identifier and it's block-data is used to create the next identifier for the next block through hashing and hence forming the central concept of the blockchain whereby linking all data. This creates an immutable ledger where previous transactions can no longer be altered without causing major disruptions to the overall chain of data. This chain of immutable records are maintained by a group of peers whereby the whole system is not owned by any individual but instead be owned and managed by the cluster of individuals.

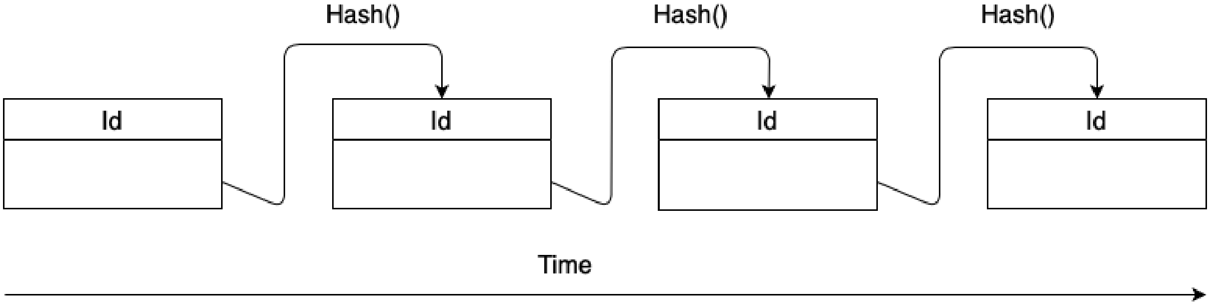


Figure 1. Concept of Blockchain

Bitcoin's consensus algorithm is based on proof of work. Individual's in the cluster of peers all work in ”mining” a block that could potentially be the next block added to the blockchain. The difficulty in mining the block is where the name ”proof of work” is derived. Moreover, this difficulty is managed by the network of peers and allows for the control in the speed of blocks being added to the chain all the while preventing peers forking the blockchain and hence preventing double spending.

There are other consensus algorithms not based on proof of work such as proof of stake employed in blockchains such as for the cryptocurrencies EOS and NXT. These mechanisms aims to reduce the waste in energy during resource utilisation since the other proof of work mechanisms contributes to a high proportion of energy consumption (Li et al. 2018) . By removing the middle man of the mining process this allows for higher stake individuals to probabilistically have higher likelihood of discovering the next block without doing the work.

Second generation cryptocurrencies employ blockchains also but further adds Turing Complete smart contract capabilities along with its ledger capabilities.

On the other hand, in more recent approaches, companies have considered a more enterprise friendly blockchain called private blockchains that is dependent on a consortium model in which all peers are identifiable. This is more approachable in applications whereby data transparency is fundamental such as for usage in the financial market. An enterprise friendly blockchain also aims to deliver high transaction throughput, permissioned networks, privacy and confidentiality of transactions.

Hence, the consensus algorithms utilised in these private blockchains can be more traditional approaches of crash fault tolerant (CFT) consensus protocol and byzantine fault tolerant (BFT) consensus protocols. This allows for the faster consensus times and hence higher transaction throughput.

It is these private blockchains that I will consider in my application due to the above mentioned benefits. The current DSU system is fully centralised by the DSU, however we could argue that the Student Society Groups may benefit becoming decentralised due to their heavy reliance on the DSU's ability to provide quick and impactful responses. It is here that the problem occurs, due to the heavy workload required of the DSU, society groups may suffer in the form of delayed responses or human errors.

From the current system, we see that many operations between student groups rely on the DSU to act as an intermediary. This process is similar to real life bank transactions which also occurs a cost for the two parties in terms of a transaction fee. Hence, we consider the possibility of removing this middle man and thus removing costs much like the proposed solution given in Bitcoin. We propose a private blockchain in the form of Hyperledger Fabric an open source blockchain implementation developed by the Linux Foundation.

Hyperledger Fabric is a consortium based network that relies on a trusted membership provider, an ordering service, certificate authorities and peers. Unlike a completely decentralised network we have a semi-decentralised approach in which the network is decentralised to the existing administrators but only chosen individuals can join and act as an administrator. Administrators can then define a consortium between organisations and thus these organisations can then form a channel to communicate between in our network.

B Project Overview

With the consideration of time frames and computing resource, we have found that the Hyperledger systems provides powerful frameworks and toolsets that allow us to develop a system in which meets our criteria's as a proof of concept application for student groups. Our developed main system will provide basic capabilities needed for a functioning body of student groups. This includes a web based front end to interact with the blockchain.

In addition to Hyperledger Fabric, we will also be utilising Hyperledger Composer a framework to develop business networks on top of the Fabric blockchain network, enabling us to develop smart contract code using higher level languages such as Javascript and their provided modelling language. This enables a much needed reduction in our time to market. We can view this composer framework as our smart code development environment, enabling us to provide definitions to assets, participants and transactions.
The business network developed through Hyperledger Composer act as our back end to the system as it provides a REST api and moreover to develop a user application we will need a front end framework and thus we will be using Angular in our development of a Web based application.

Moving on, we will create a survey to gather user experience data and gather insight on the effectiveness of our blockchain implemented application. Finally, we will also test the efficiency of the backend system in comparison to more conventional public blockchain networks using

Hyperledger Caliper and from this gain insight on the capabilities of this technology.

C Project Deliverables

The overall aim of this project is to implement a blockchain based application for the interactions between Student Groups and Student body. To do this we have utilised a variety of open source frameworks provided by the Linux Foundation under the Hyperledger namespace. From this we will make conclusions on the blockchains effectiveness from the infrastructure metrics, user experience and software development approach.

We describe the tasks completed to achieve the aims of this project below:

Basic deliverables:

- Implement a single peer fabric network
- Implement basic smart contract code for the interaction between student groups and student body

Intermediate Deliverables:

- Implement a multi-peer fabric network
- Design and implement a user application through Angular that provides a front end GUI for both students and student groups
- Implement advanced smart contract code for the interaction between student groups and student body

Advanced Deliverables

- Survey users on their experience with the application and gain insight on their perspective on the blockchain backend
- Implement blockchain performance benchmarking for our system using Hyperledger Caliper

II RELATED WORK

In our solution we will examine the infrastructure implemented during this project and the various technologies involved. From sources including cryptocurrency whitepaper's (Nakamoto 2008) we have examined a variety of possible solutions we could have taken in our implementation. Ultimately we have chosen Hyperledger's frameworks though there are other possible technologies such as other blockchain frameworks and also Non-blockchain technologies. We will now discuss these related technologies.

A Ethereum

Ethereum differs from Bitcoin in a variety of ways (Buterin et al. 2013). Ethereum is account based and so behaves like states allowing transactions to act as state changes. Ethereum relies on two concepts ether and gas, where gas is used to pay for the smart contract transactions during implementation and thus prevents over usage of resources. Thus, gas acts as a way a unit of measure of work and is defined in smart contracts and transaction calls to ensure both parties are satisfied in the resource usage. Furthermore, Ethereum defines a variety of different memory types that are allocated to smart contracts and accounts of which again costs gas to use. Hence, with the combination of memory, state changes and smart contracts we find that this mimics a Turing machine allowing this decentralised network to act as a Turing complete system.

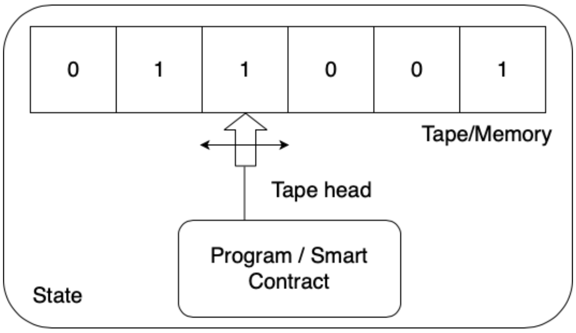


Figure 2. Turing Machine parallels with Ethereum

This is a powerful statement, allowing for the development of many applications (Hu et al. 2018). One such example is Cryptokitties, an asset generation and trading application (Axiom-Zen 2017). We consider this application due to its surge in popularity and thus resulting affect on the Ethereum network. At it's peak cryptokitties contributed over 10% of the ethereum transactions, hence causing ethereum transactions to become more costly. Cryptokitties are a smart contract application that consists of non fungible tokens displayed as cryptokitties, a transaction that allows for breeding and a descending clock auction that allows for users to bid tokens. Therefore, such a complex application demonstrates Ethereum's ability to produce other such applications. Nevertheless, such a system is not suitable for use cases such as our own where user anonymity is not needed. Moreover, transaction information is open to the entire network an attribute again not suitable for our problem.

B Blockchain in applications

From a variety of papers, there are research in the prospect of blockchain technologies for application development in commercial fields such as in finance and education. From the research we see a general positive outlook on blockchain though a general consensus that current capabilities of the most popular blockchain implementations may not be the most suitable. Hence, we see the major concerns highlighted in these papers being: Privacy, security and scalability.(Bonneau et al. 2015)

Privacy becomes an issue in the financial industry where regulations are of such high importance. Current challenges to financial blockchain applications involve the required privacy of transaction details (Guo & Liang 2016). Furthermore interoperability with legacy systems and scalability also provide issues to the incorporation of blockchain technology. As mentioned above, there is a positive outlook on the potential of blockchain in the financial sector where banks have already began investigating the technology and testing its viability to be integrated with their current systems. JPMorgan have developed their own blockchain named Quorum (Baliga et al. 2018), an Ethereum based enterprise blockchain. UBS have formed their own

Blockchain research team investigating the potential of smart contracts. Over 40 banks have gathered together to form a consortium to investigate private blockchains.

Moving on, blockchain technologies have also influenced the education sector (Sun et al. 2018). One difficulty with online education is the lack of recognised certification throughout. Blockchain technology provides a potential solution, an immutable ledger is highly desirable for this purpose. Certification that is awarded to individuals are kept on this ledger providing a suitable data storage that is ultimately set and unchanging. This provides reassurance to learners that their accomplishments can always be recognised.

In this paper (Grüner & Alexander Mühle 2018) we see a focus on the identification of the need for blockchain solution which focuses on two factors: the remediation of central governance and the management of digital objects. The first concerns the need for a central body governing the system. If such a body is not desired then this suggests a higher tendency towards a blockchain approach. The second concerns digital assets that needs tracking. Physical objects are highly likely to be loosely coupled to their digital identification hence meaning it is unsuitable to use blockchain ledger. For example, looking at an identity management scenario. It is clear that identity creation, modification and deletion should not be controlled by any entity apart from the individual. Moreover, digital identities act as a digital asset useful for authorisation and authentication for services. We therefore see a suitable use case for a potential blockchain implementation.

C Non-Blockchain systems

A vital question we must ask is whether or not we should be using a blockchain implementation. In our research we have also considered whether a non blockchain alternative is more suitable. We have looked at the current DSU system to make comparisons, the current system is based on client and server. The system enables student groups to perform a variety of functions including linking to a webpage, setting up membership and events, providing a view of members and their details, and finally setting up subscriptions. However, this system provides no interaction between other student groups of which may occur during the organisation of events. Student groups will communicate and at times interact by exchanging room bookings or assets. Thus, with the current system any records that needs to be kept must go through the Student Union. A step that is not needed. Focusing back to the previous factors that suggest a blockchain system may be suitable, we see that a governing body is not needed here as it only occurs overhead in the response times. Furthermore, room bookings fit the criteria of digital objects in which management of their identification, ownership and details may be benefited from an immutable decentralised ledger.

D Hyperledger Fabric Blockchain

We now discuss the Fabric Blockchain that we will implement later in our solution (Androulaki et al. 2018), this blockchain implementation varies to public blockchains (Guegan. 2017), removing the anonymity of peers and differing in consensus algorithms. Fabric's consensus approach is very different to public blockchains. In Bitcoin, consensus is achieved via a single algorithm that achieves an agreed upon order of a batch of transaction. Instead consensus is achieved via validation checks throughout the lifecycle of a transaction from proposal, to endorsement, to ordering, validation and commitment. Checks are required in every step of the transaction cycle such as endorsement policies that indicate which members endorse specific transaction types. System chaincode ensure that these policies are met. Hence, prior to the commitment of transaction, peers run the system chaincode to ensure that the policies are met and that enough endorsements have been achieved. Versioning checks are also employed, ensuring all ledgers are up to date prior to new commitments. Thus, removing the opportunity for double spending.

The Fabric blockchain are made up of many components including: orderer, certificate authorities, organisations, membership service providers and channels. Below is a general diagram of a fabric network:

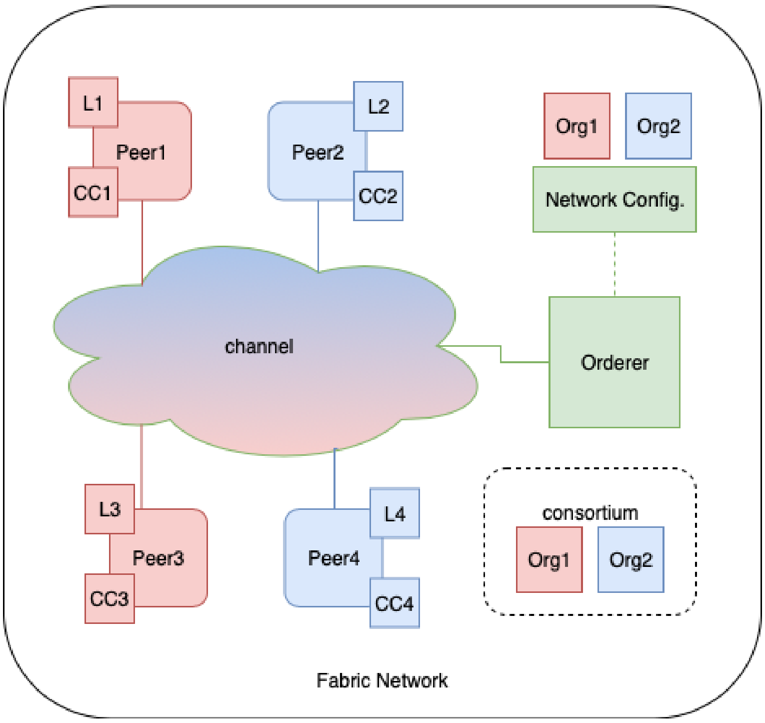


Figure 3. Fabric Blockchain network

The above network demonstrates the relationships between actors. The orderer is the first to be created, closely followed by a network configuration. The owner of the network configuration dictate the administrator of the network. However, to be a decentralised network multiple actors must therefore have access to this network configuration. Hence, multiple organisations do share access to this configuration. Organisations that have network access can form consortiums which allows for their communication. Hence in the example above, Org1 and Org2 are joined together in one consortium. Once in a consortium a channel is formed on the network where all communication between members of the consortium occur. All the while the certificate authorities dispenses certificates to components to identify them belonging to an organisation. This process is achieved via a structure called a membership service provider. Finally, once a channel is formed peers from the organisations that are members can join and acquire the ledger and are able to promote their chaincode onto the channel.

III SOLUTION

Before we started the implementation of our application, we first created a requirements specification. The benefits from doing so is that we now have more insight to how the development process should proceed. We now have functional requirements that should be implemented as well as of having a plan of achieving non-functional requirements. A website template have also been created to match up these requirements and again help in the development process, this gives us a thorough concept of what the end product should look like and reduce experimentation during the development. Without experimentation, we can focus on achieving our requirements.

Firstly, we will look at the design choices made for the implementation of our blockchain backed user interface. The implementation details regarding the blockchain network via Hyperledger Fabric have been briefly touched upon above, the diagram in figure 3. is the actual network architecture of our implementation, we have three external hosts of: an orderer and four organisations' peers. The peers host the ledger and smart contract code where as the ledger enables the consensus of transactions made within the network. This fabric network consists of one channel that provides communication between the two organisations which we now relate back to our problem scenario as being student groups or being students. With the channel definition it is possible for further organisations to join the network as either students or student groups. Thus, this single channel enable user applications to be developed upon. Due to the channel configuration, students and student groups have differing permissions. This is not generally possible with public blockchains. For simplicity we have developed an application on a single host, providing the host with cryptographic identification to act as both students and student groups for access of the channel due to ease of producing a single application. In actuality, we would split the applications into two separate entities hosted on two different virtual machines.

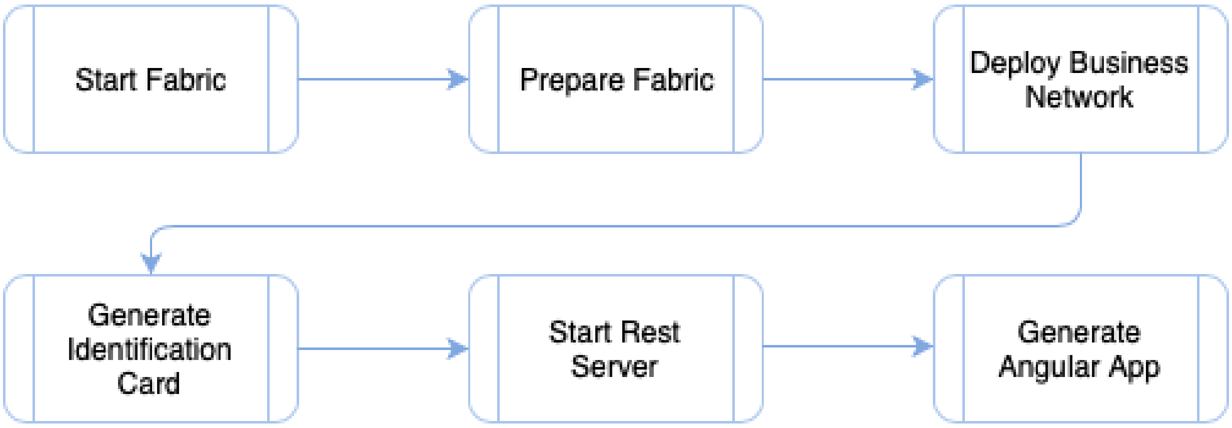


Figure 4. Flow diagram for building our angular app

The above is a generalised flow for the development of our application. As we move through the solutions section of this paper, we will discuss the different points in our flow diagram and explain the implementation process involved at each stage.

First we implemented three virtual machines using Google cloud and Amazon web services for both flexibility of resources in case one provider suffers downtime but also provides us with a good demonstration of the decentralisation of our network. With these virtual machines we have used Ubuntu LTS 16.04 with 50gb of storage and vCPU 3.75GB. We have implemented docker to run multiple images required for the complete picture of our fabric network. Hence, a host that is not the orderer will house the certificate authority, and two peers. These docker images communicate externally via docker swarm and by forming a subnetwork in which all peers including those from other hosts can communicate.

Moving on, once we have instantiated our blockchain network, we then produced our business network that sit on top and which utilises our blockchain, we used Blockchain Composer that defines the business via four parts: the model, script, access control and queries. The model gives us the definitions for assets, participants and transactions.

In our scenario the assets involved are: Room Bookings, Room Booking Tokens, Room Booking Votes and Student Wallets.

The participants are: Student, Student Groups and the DSU (the latter is not needed but is here for completeness).

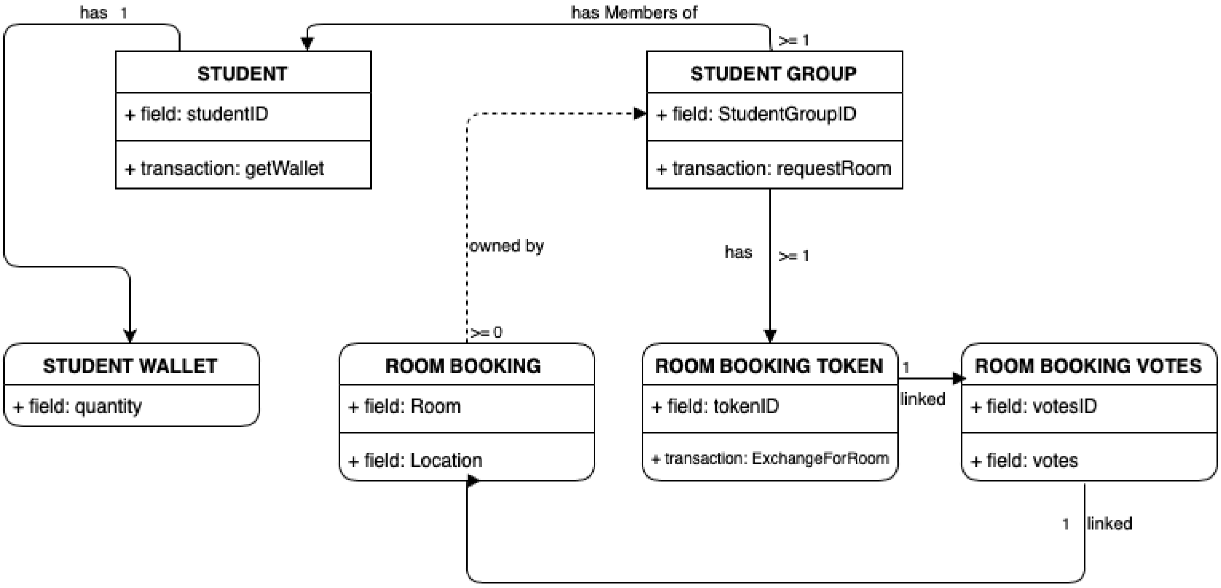


Figure 5. Class diagrams for our Business Network

The script provides us the smart contracts of our transactions. There are numerous transactions involved in this application, so I will give a brief description on some of the capabilities of this network: Students may purchase memberships of a Student Group, Student Groups may exchange room bookings and Student Groups vote for Room Bookings they deem suitable. A more complete description may be found in our requirements specification.

Access controls enable us to manage business network level permissions which is much needed for our application. This involves read permissions such as Students being only capably of reading their own details. Students may only join Student Groups if they have enough money in their wallet to pay for membership. Student Groups can only book a room if they own the corresponding 'Room Booking Votes' asset and of which have the sufficient number of votes needed.

After completing the business network implementation, we ran tests using Composer's playground environment allowing us to experiment with the business network layer. By creating test participants within this environment we are able to test transactions and access controls. Moreover, we are given two versions of Composer's playground: Web based and Fabric based. With the first, we are able to export our business network onto a web backend and not on any fabric network. This enables us to quickly test results and interactions, every transaction and asset has been tested via this method to ensure that the generated REST API will perform as it should, moreover deploying a business network onto this web based version is considerably more efficient and hence helping reducing the time needed for development. On the other hand, the second Fabric based version allows us to test our business layer on our blockchain network giving us run time estimations.

Once testing has been completed we move onto generating the REST api's required for the communication between web application and blockchain network. These REST api's are generated using the Loopback framework. Moreover, we enable OAuth authentication via Github to allow us easy verification of user's and for enabling user accounts without setting up our own authentication layer. An application level architecture is given below:

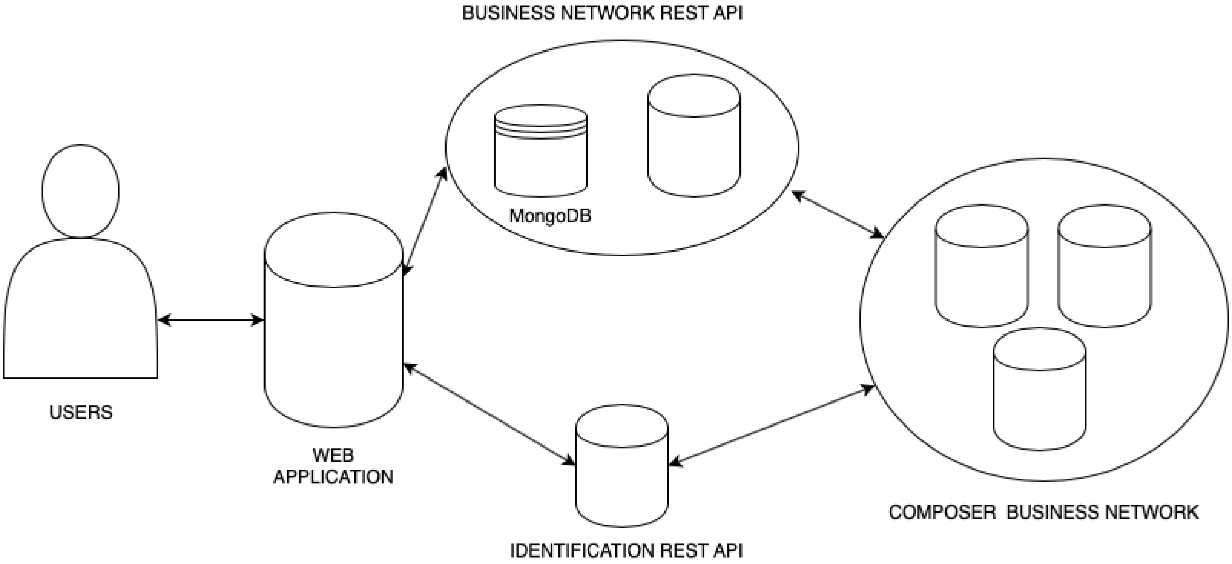


Figure 6. Application software architecture

The above diagram details the software infrastructure deployed for our application. One control feature of our REST api's is "Wallet" definitions, these wallets contain business network cards that contains connection profiles and certificates to enable identification of business network users. One design choice made was to allow users to generate multiple business network connection profiles in the use case where a user will require both Student profile as well as a Student Group profile connection.

Hence, this "Wallet" definition enables us to generate a REST API allowing identification that depends on provided business network cards. This first REST API prevents users accessing other connection profile permissions. However, we now need a separate REST API that allows for the generation of these business network cards and hence connection profiles. We have defined a separate "profile provider" participant with access controls to only be able to generate these business network cards. Moreover, only the network admin have access controls to generate this participant so we are assured that only one of these participant exists and have limited control. Finally, this second REST API is generated with a set connection profile of this profile provider. Therefore, this second REST API has the sole purpose of generating connection profiles for users of which can be uploaded to the first REST API's wallet to grant user's permission for access to their assets and transactions.

These two REST api's enable us to communicate with the fabric blockchain network and thus we are now able to produce our front end application. Our application relies on the Angular framework and hence the superset language of Javascript: Typescript.

The benefit of Typescript is its statically typed nature enabling for type inference. During development type inference allows source code editors to infer method calls and parameter checks. Typescript is also a compiled language and hence detects type errors and syntax errors before runtime hence quickly detecting coding errors, this is very useful in large applications development (Fischer & Hanenberg 2015).

By using Angular we are able to bootstrap our application. Moreover, the modular capabilities of Angular allow us to streamline our code base. The file structure of our application is split into four main folders: app, services, environment and assets. The app folder contains our modules and components. Services contain data services and in our case, methods for our REST API calls. These calls will implement promises hence allowing for asynchronous calls and hence not halting the website when gathering data. Environment contains environment definitions such as for production or testing. Finally the assets folder contains all image assets and CSS files that may be required throughout the application.

We now look at the main application code in the app folder, each component have their own folder containing html, css, script and declaration of services. Hence each component act independently and are exported to be used elsewhere in our application. It is also possible for components to use other components as subcomponents. This allows for a lot of flexibility in our design choices.

A major design choice made during development is restricting the services required for each component. By doing so, we reduce repeatability of REST API calls which both improves efficiency and improves modularity. The modularity of code is highly important for code maintainability and readability. Hence, services in components are restricted to call REST API's for one asset, transaction or query, it is now easy to see what data each component is capable of accessing and hence see what they should be capable of doing.

We have implemented a couple of subcomponents, this includes a calendar component and a fading component. These are designated for our room booking page and logging in page respectively.

The calendar component outputs the selected date to the parent component that is our request room booking page, this date is used during our REST API to generate a corresponding Room Booking Token and Room Booking vote asset, these are then utilised during the voting of requested rooms.The fading component is an animated component that fades in and out by providing a ”isVisible” boolean parameter to the parent component that sets the visibility of the component.

Angular components are also used throughout as routes in our web application, each component can act as a section of a page of which can be split into pages for Students and pages for Student Groups. The main encompassing body in our application is our app module that contains all routes to other pages. initially this app module is set up as an authentication and authorisation page. Rerouting back onto this module once authentication is complete allows access to components depending on the identity of the user. Our authentication method is through Github and provides an access token for our first REST API.

The Student's pages allow routing to view their memberships, events of societies that they are members of, exploration page for all student groups and finally a page for personal information.The Student Group's pages allow for routing to view members, room bookings and a request room page. Below is a diagram that details the page routing of our web application.

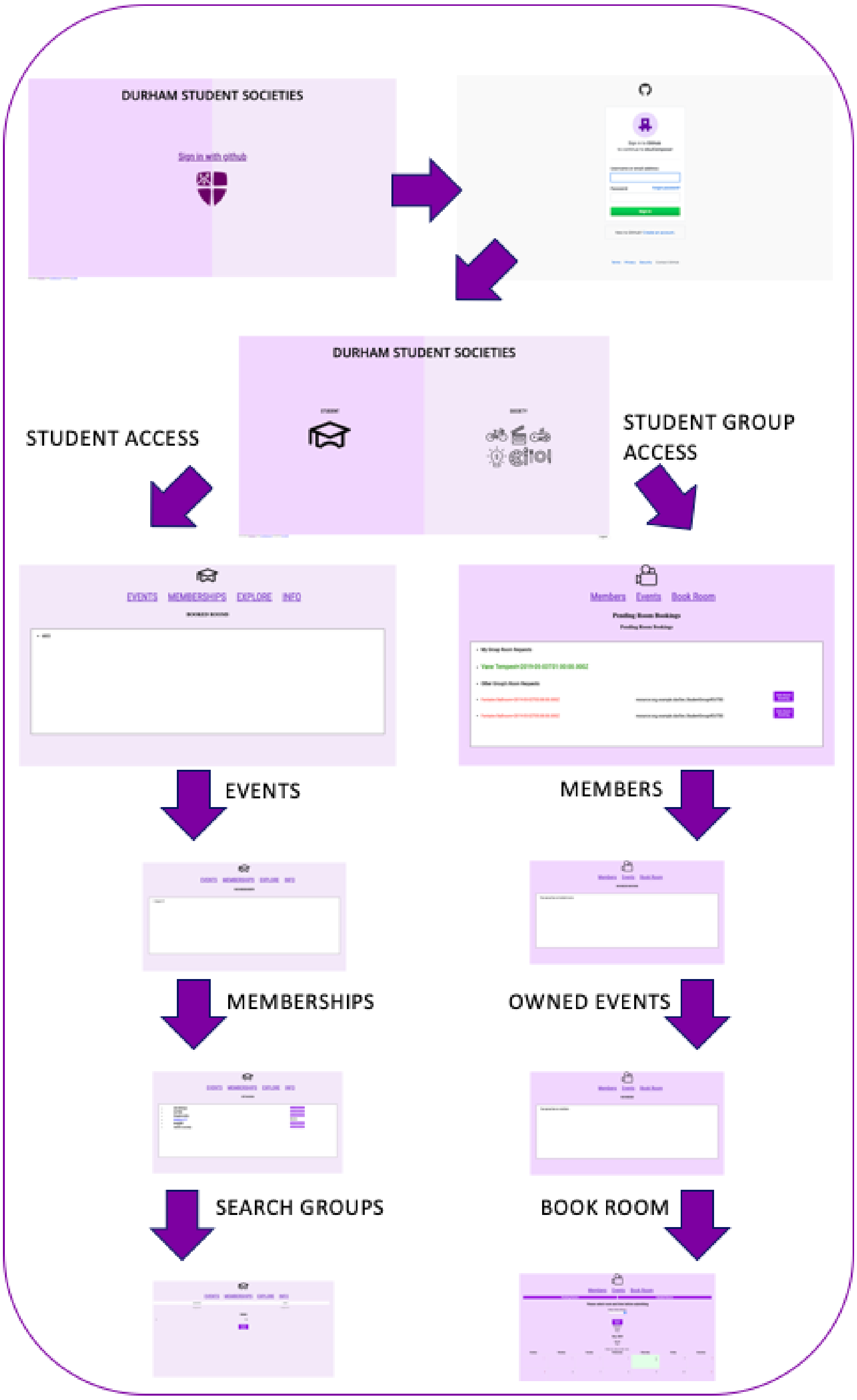


Figure 7. Web application flow diagram

To gain results for this project we decided with two approaches. Technical results are gathered through blockchain benchmarking using Hyperledger Caliper. A survey was created to gather user feedback.

For our technical results we implemented benchmarking of our fabric blockchain by defining tests that will generate rounds of transactions, these transactions are ran and performance metrics are gathered on their minimum latency, maximum latency, average latency and throughput. Alongside these performance metrics are also resource consumption measurements of which contains details for each docker container that is required for our network.

We have implemented three transactions to test: details update, asset generation and finally asset generation with an update to their details. Overall, we will run three tests with the corresponding transactions, each of these transactions take as input the number of individual transactions to run hence we have decided to run 25 or 50. In summary, we will generate three overall tests each of which contain 10 rounds of 50 transactions. At each round we measure the average throughput of that rounds transactions.

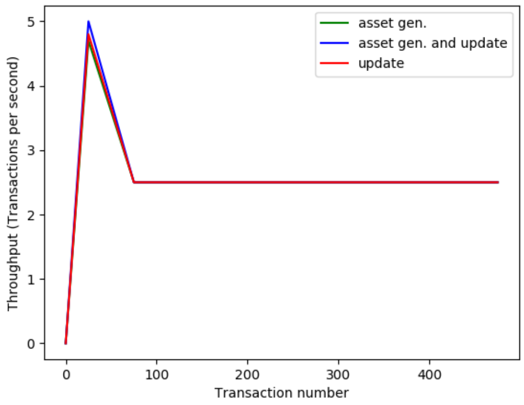
For the first transaction that is the update, we initialise the fabric blockchain with the necessary Student asset, during the main testing stage these Student assets are updated. In the second transaction a very similar approach is taken except that the asset generation is moved to the main testing stage and we remove all asset updates. Finally, for the last transaction we have asset generation and updates in the main testing stage.

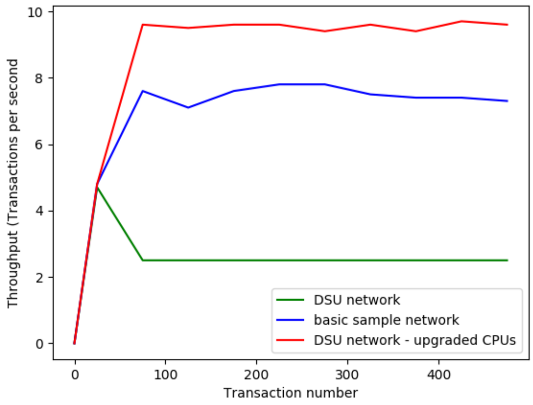
For comparison, we will also run a smaller fabric network Caliper test, this will be a sample network given by Hyperledger Fabric. This basic network contains only a simple asset and so we will run a much similar asset generation test. The metrics of this smaller network will give us some insight on the scalability of Hyperledger Fabric as we will see how performance differs depending on network complexity.

Finally, for our user feedback results we have generated a survey using free online software, we have used 8 questions with opened comments as the last. This survey is designed to gain an understanding on the ease of user-friendliness of the application. Hence, these results lends itself more towards the software engineering approach of this project and will give us conclusions on how to best improve the application if it were to be an iterative development cycle. These conclusions with the addition of a reflection of the development process are indicative of how well suited the blockchain technology towards this problem. From these 8 questions we will compile results and explore the trends that have emerged. These survey questions have varying answer types depending on their suitability, questions may have ratings from 1 to 10 or ratings from very satisfied to very not satisfied.

IV RESULTS

For our results we will discuss the experimental data gathered from the benchmarking done via Hyperledger Caliper. We will first compare the performance outcome of our three tests, looking at the throughput and latency.





Left: Figure 8. Throughput findings of our application network benchmarks, Right: Figure 9. Throughput findings from varying network setups benchmarks tests

From figure 8, we see that the throughput of our network caps at the beginning, the transaction therefore become backlogged and as we see in our graph stabilises at only 2.5 transactions per second. This occurs with every one of our transaction types.

All three transaction types display similar results and demonstrate a very common problem with blockchain technologies in its scalability. This is further demonstrated in figure 9.

Figure 9. compares our network with a very simple sample network that has very limited capabilities. This shows that an increased complexity of network have major consequences on the performance of our network. In the basic network the bottlenecking doesn't occur and so we have a stable throughput rate throughout, moreover the throughput values are significantly higher than in our network, demonstrating at times 2× better performance. Looking now at the latency of our networks, our comparisons show how differing complexities of the transaction themselves have an affect on performance. From figure 10. we see an increase in latency when combining both asset generation and its update. This increase though obvious is not multiplicatively large thus it may be possible to increase transaction complexity further without hitting large performance deficits.

Moving on, now looking at the resource utilisation of our network, we give a snapshot of the utilisation during a round of transactions, more accurately for the ”asset generation and update” transaction. From figure 11. we see the memory usage by all docker containers found in our network, we see that our peers require the bulk of memory. The dev peers house the smart contracts and our mongodb databases house the ledger hence both these containers also use significant amounts of memory. The actual memory usage change from one round to another is significantly small hence demonstrating that memory is not the limiting factor to the scalability of our network.

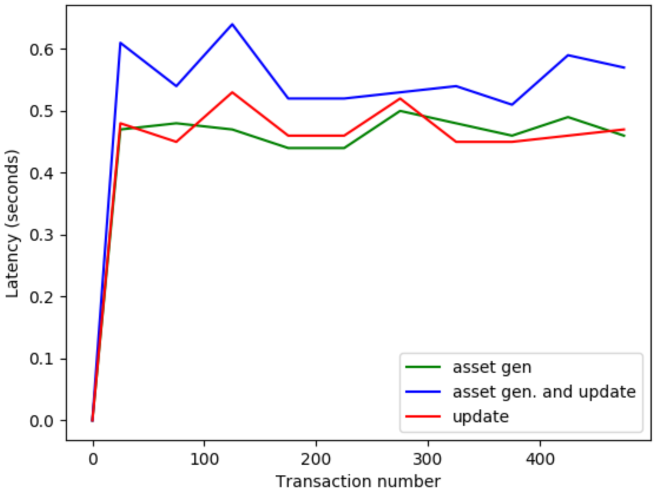
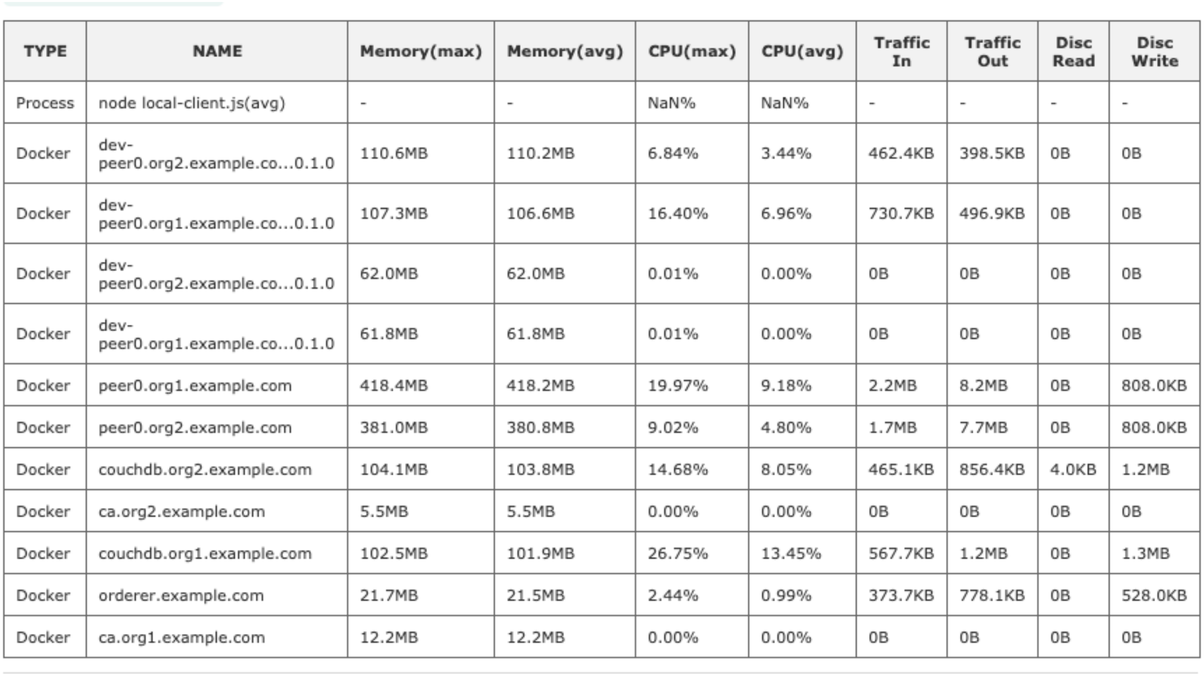


Figure 10. Latency findings of our application network benchmarks



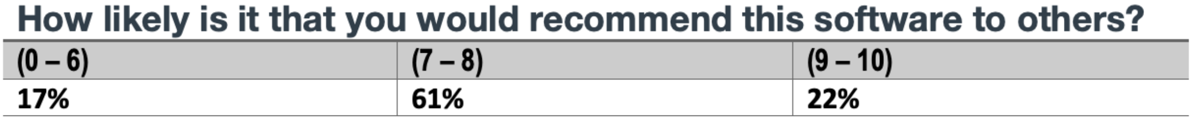
TYPE	NAME	Memory(max)	Memory(avg)	CPU(max)	CPU(avg)	Traffic In	Traffic Out	Disc Read	Disc Write
Process	node local-client.js(avg)	-	-	NaN%	NaN%	-	-	-	-
Docker	dev-peer0.org2.example.co...0.1.0	110.6MB	110.2MB	6.84%	3.44%	462.4KB	398.5KB	0B	0B
Docker	dev-peer0.org1.example.co...0.1.0	107.3MB	106.6MB	16.40%	6.96%	730.7KB	496.9KB	0B	0B
Docker	dev-peer0.org2.example.co...0.1.0	62.0MB	62.0MB	0.01%	0.00%	0B	0B	0B	0B
Docker	dev-peer0.org1.example.co...0.1.0	61.8MB	61.8MB	0.01%	0.00%	0B	0B	0B	0B
Docker	peer0.org1.example.com	418.4MB	418.2MB	19.97%	9.18%	2.2MB	8.2MB	0B	808.0KB
Docker	peer0.org2.example.com	381.0MB	380.8MB	9.02%	4.80%	1.7MB	7.7MB	0B	808.0KB
Docker	couchdb.org2.example.com	104.1MB	103.8MB	14.68%	8.05%	465.1KB	856.4KB	4.0KB	1.2MB
Docker	ca.org2.example.com	5.5MB	5.5MB	0.00%	0.00%	0B	0B	0B	0B
Docker	couchdb.org1.example.com	102.5MB	101.9MB	26.75%	13.45%	567.7KB	1.2MB	0B	1.3MB
Docker	orderer.example.com	21.7MB	21.5MB	2.44%	0.99%	373.7KB	778.1KB	0B	528.0KB
Docker	ca.org1.example.com	12.2MB	12.2MB	0.00%	0.00%	0B	0B	0B	0B

Figure 11. Resource Utilisation findings of our application network benchmarks

Now looking at the CPU usage we see that the couchdb databases are the heavier users due to amount of disk reads and writes needed for the update of our ledgers at every asset addition and update. Moreover, it is here where we see the limiting factor for our performance. The summation of the CPU (avg) column gives approximately 40%, a significantly large percentage of all CPU utilisation, at peak usage this will be quite a proportion higher. We can now give one advantage of this form of blockchain network, the chosen virtual machines are scalable hence performance boosts are possible, this is unlike any public blockchain that depends on the resources of the entire network and thus is dependent on all other anonymous peers.

Finally, we now look at the user experience survey results. Our survey took results from 23 users, of which answered 8 questions in total, the first of which requires a selection from 1 to 10 on how likely they are to recommend this application another person.

For the first question, we see a overall positive response on the website and a significant portion of the users show general openness of recommending this application to other users, a positive outlook for blockchain technologies as a whole since there is not complete dissatisfaction with the system. Given that the survey was preceded with information that this system is blockchain based, with a quick background of the technology, suggests that encounters with blockchain is now more common.



How likely is it that you would recommend this software to others?

(0 – 6)	(7 – 8)	(9 – 10)
17%	61%	22%

Figure 12. Question 1 results from our user survey

The next 6 questions deal with the general adequacies of software, including reliability, security, ease of use, look and feel, and experiences with its functionality. The below graphs demonstrate a general positive view on the application bar two key factors, the responsiveness and the capabilities of interaction with other users.

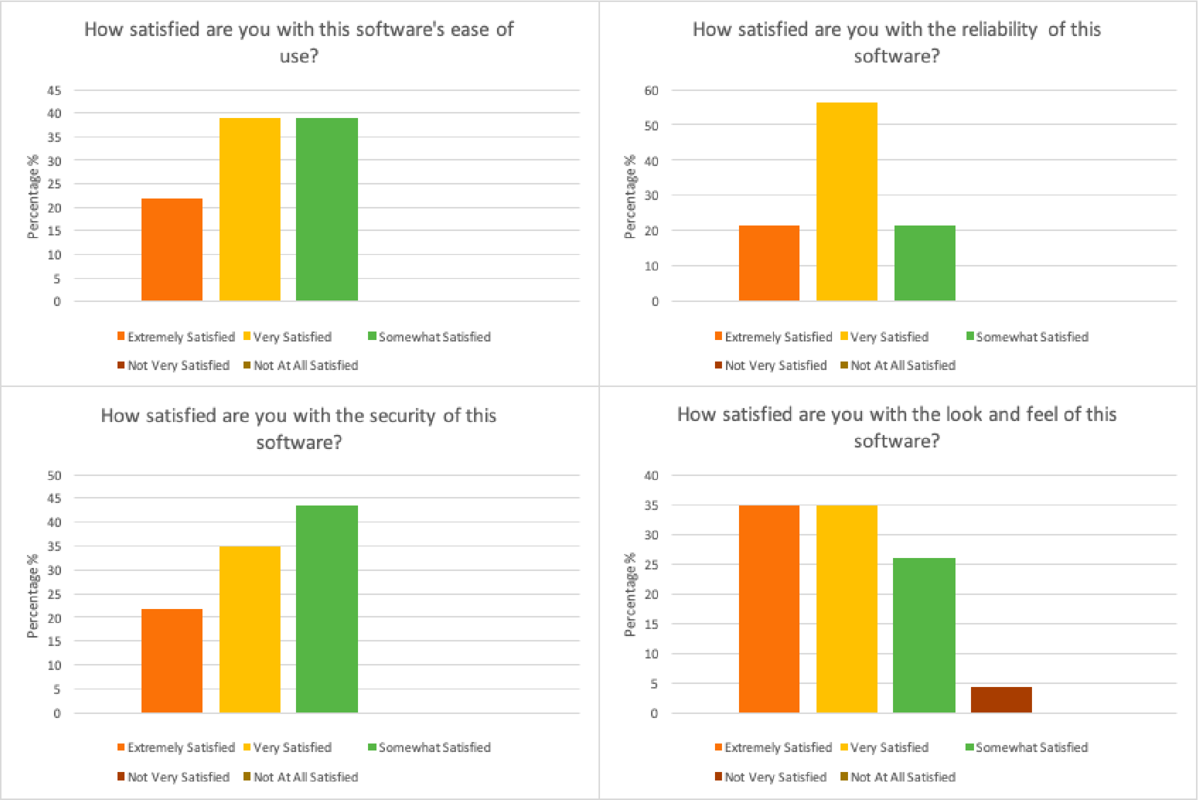


Figure 13. Question 2,3,4 and 5 results from our user survey

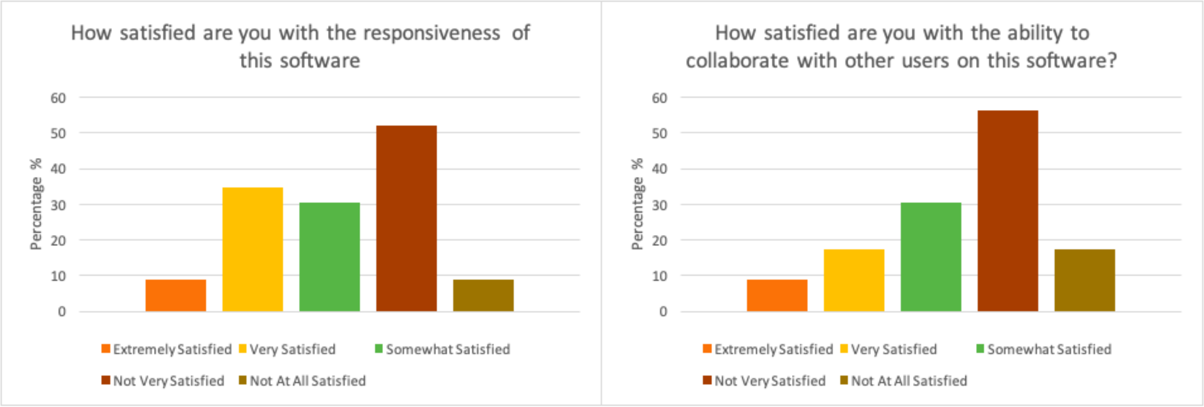


Figure 14. Question 6 and 7 results from our user survey

We shall now delve some of the reasons why these two negative responses have occurred.

The first is responsiveness, as mentioned previously during testing we saw our throughput quickly backlogged, however in this review scenario not many users would have been live so encountering throughput issues would be uncommon. Nevertheless, looking instead at the latency we see that our more complex network increased latency and it is here that would have caused slower responsiveness.

The second negative response is the lacking some of the capabilities of interacting with other users, when considering that this application is targeted at a community of closely tied users. Due to going to the same University of at least sharing the same facilities. Hence, the ability to communicate with others is highly valuable in this scenario. This is a minor issue in that it does not negatively impact our arguments for developing with blockchain. The blockchain is capable of providing such functionalities and it was down to the developers in their design choices that the some functionalities were not present.

On a more positive note, looking back on the other 4 questions dealing with the applications competency, we again see a positive response, the look and feel of the application gathered good results, so did the ease of use, reliability and security. We will touch on reliability and security again as this is a major talking point in regards to blockchain technologies, though these users would have believed that the blockchain technology had little influence due to it being out of sight. In fact, the reliability of the system when excluding throughput is largely reliant on Fabric's consensus algorithm. Moreover, data immutability ensures that there is the added security of having a log of all transactions and events, and as a result malicious users can be isolated.

The look and feel of the application, setup experience and ease of use gained positive responses due to the preplanning done prior to the start of development, this provided us a clear approach of how the interactions should act.

Finally, we look at the last question of the survey of which is an open question on how to improve the software. As expected the general response is that the responsiveness of the software needs improving and as previously mentioned this is can be achieved via improving the CPU capabilities as is possible with cloud infrastructures. We can demonstrate this via upgrading our CPU resource allocation for Caliper benchmarking purposes. Instead of the 1vCPU at 3.75GB we have upgraded to 4vCPUs at 15GB and thus we gained a vast improvement.

Figure 9. also compares our original network benchmarks with the upgraded CPU virtual machines, the improvements results in the complete removal of backlogging of transactions, moreover with these updated resources both latency and throughput improve drastically. This demonstrates that unlike public blockchains, we have flexibility on the performance of our networks, even showing a vast improvement on some public blockchain transaction throughput. Bitcoin's current throughput is approximately 3.5 transactions per second.

V EVALUATION

In this section of the paper we will focus back on the original aim of this project in: to develop an blockchain application suitable for the Student body and Student Groups, moreover, to examine the potential of blockchain in software development. From the process that we have undergone to pursue these aims, we can highlight some of the strengths and weaknesses of our approach.

A Strengths

From developing our own blockchain application, we have gained hand's on experience with some of the modern blockchain technologies, hence the strength of our solution spans from having a physical modern application to test and gain insights from. Moreover, by selecting a most appropriate approach we have achieved a wide reach of blockchain understanding.

B Weaknesses

Moving on the weaknesses, we see a direct counterpart to our strengths, having produced a modern application under the limitation of time meant that we could not have produced multiple implementations from a variety of blockchain technologies, hence in our final results we do not have conclusive results on the strength of every blockchain approach but instead have a general sense of how modern blockchain's have developed and improved as seen from our individual current blockchain's results.

C Limitations

The main limitations found during this process is time restraints and CPU capabilities. Firstly, having to produce an entire web application requires a large quantity of time hence shortcuts in implementation has been made. Including in our situation, the combination of Student and Student Group applications into a single entity. However, the network configuration separates the access controls between students and student groups meaning it is a bad design decision to combine the access capabilities of both groups into a single application access point. The other limitation of CPU restraints span from the virtual machines dedicated to establish the network, due to cost restraints we only have access to the lower level CPUs in both Google cloud and Amazon Web Services. As mentioned in our results, this heavily impacts the throughput of our network and therefore our application. This has negatively impacted our survey results. Nevertheless, this limitation has been taken into consideration and as mentioned demonstrate one advantage of our chosen blockchain technology.

D Project Organisation

The overall organisation of the project has its strengths however the main focal point here is that it is tending towards a software development project rather than a research based project. Therefore, in our results we not only focus on user feedback but also on performance and resource utilisation metrics, this allows us to gather more quantitive data and present a more scientific approach to the process. Given our stance that this project is to produce an application, the learning process throughout has been advantageous in developing new technical skills of which deepens our abilities in software engineering.

VI CONCLUSION

In this project we aimed to develop a blockchain based application that can be used by the Student Groups and Students in replacement of the current systems provided by the Durham Student Union. In our investigation we develop a private blockchain application as was the most suitable option for our scenario. However, another goal was to investigate whether or not current blockchain technologies are suitable in developing software applications such as this, hence metrics were gathered and a survey was used to accomplish this task.

A Findings from benchmark tests

We gathered metrics using Hyperledger's Caliper tool by running benchmarking tests on our network as well as a comparison network smaller in size to contrast with our networks results. These metrics gave us insight on the performance of our network as well as resource utilisation. The comparison made between networks allowed us to gain an understanding of the scalability of the networks, of which shows that more complex network do in fact take a significantly longer time to transact with reduction in throughput by almost a half in our case. Nonetheless, resource utilisation data gave us insights on the possible improvements that can be made by upgrading resources, we found that the limiting factor in our scenario was CPU performance and so further tests showed that these upgrades performed significantly better. This demonstrates that there is flexibility in private blockchain implementations and their performance a factor that can be further considered in further applications.

B Findings from surveys

Finally, our survey's showed that user response on the application was tending towards more positive, this suggests that current blockchain technology could be a viable option in application development given the right problem cases as mentioned in our related works. Survey results showed that the main issue with the application was the responsiveness and as mentioned before our tests with more capable resources demonstrated that this can be resolved. Nevertheless, further research is required to examine whether or not the costs of upgrading to hardware capable of running a blockchain network is suitable. Instead, it may be most appropriate to change technologies all together and so reduce hardware costs.

C Possible Future Works

Possible extensions to this work include further research in resource utilisation and a more in depth look at the scalability of Fabric networks with regards to hardware resources. We may extend this research into other blockchain technologies and examine application development with public blockchains and the problems that may arise there, much like in the case of Cryptokitties (Axiom-Zen 2017). Further expansions on the Angular application can be made to further demonstrate the capabilities of the blockchain network. Finally, it would beneficial to explore further the scope of real world problems that can be tackled by blockchain technologies. Further improvements of this technology is currently still being made. Hence, by exploring new problems we discover more and more the capabilities that blockchain technologies may provide.

References

Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., Caro, A. D., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., Muralidharan, S., Murthy, C., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulou, C., Vukolic, M., Cocco, S. W. & Yellick, J. (2018), 'Hyperledger fabric: A distributed operating system for permissioned blockchains', *arXiv: 1801.10228v2* .

Axiom-Zen (2017), 'Cryptokitties: Collectible and breedable cats empowered by blockchain technology'.

Baliga, A., I, S., Kamat, P. & Chatterjee, S. (2018), 'Performance evaluation of the quorum blockchain platform', *arXiv: 1809.03421v1* .

Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J. A. & Felten, E. W. (2015), 'Sok: Research perspectives and challenges for bitcoin and cryptocurrencies', *IEEE Symposium on Security and Privacy* .

Buterin, V., Wood, G. & Lubin, J. (2013), 'A next-generation smart contract and decentralized application platform'.

Fischer, L. & Hanenberg, S. (2015), 'An empirical investigation of the effects of type systems and code completion on api usability using typescript and javascript in ms visual studio', *Proceedings of the 11th Symposium on Dynamic Languages* .

Grüner, A. & Alexander Mühle, C. M. (2018), 'On the relevance of blockchain in identity management', *arXiv: 1807.08136v1* .

Guegan., D. (2017), 'Public blockchain versus private blockchain', *halshs-01524440* .

Guo, Y. & Liang, C. (2016), 'Blockchain application and outlook in the banking industry', *Financial Innovation, Springer Open* .

Hu, Y., Liyanage, M., Manzoor, A., Kanchana, Thilankarathna, Jourjon, G., Seneviratne, A. & Ylianttila, M. (2018), 'The use of smart contracts', *arXiv:1810.04699v1* .

Li, J., Li, N., Peng, J., Cui, H. & Wu, Z. (2018), 'Energy consumption of cryptocurrency mining: A study of electricity consumption in mining cryptocurrencies', *Elsevier Energy 168* .

Nakamoto, S. (2008), 'Bitcoin: A peer-to-peer electronic cash system'.

Sun, H., Wang, X. & Wang, X. (2018), 'Application of blockchain technology in online education', *iJET - vol. 13, No.10* .