User access control on the blockchain for supply chain visibility

Julian Flapper
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
j.h.j.flapper@student.utwente.nl

ABSTRACT

Manufactured products flow through complex supply chains involving many actors. Visibility within the supply chain is a key business challenge with the potential to improve business performance through higher efficiency in processes as a result of this increased visibility. This visibility is difficult to achieve as it requires sharing sensitive data, requiring a high level of access control, trust and security. Blockchain technology may provide the trust and security aspects, but some challenges are present. These include user access control to manage who can perform certain actions and access data. In this research, the necessary components for a blockchain based system for supply chain visibility are identified, a prototype is developed, leading to a system architecture and knowledge of further implications that need to be overcome.

Keywords

supply chain visibility, blockchain, user access control, XACML, BigchainDB

1. INTRODUCTION

Everyday, billions of products are being manufactured across the globe through complex supply chains. Very little is known of how, when and where these products were originated, manufactured and used through their life cycle - although the rise of the Internet of Things is likely to change that in the future, for example with RFID tags on products that automatically update the location or other information[13][12]. These goods travel through an often vast network of retailers, distributors, transporters, storage facilities and suppliers that participate in design, production delivery and sales [1]. For example, Maersk (the world's largest carrier, responsible for over 21% of the world's shipping volume), found in 2014 that just a simple shipment of refrigerated goods from East Africa to Europe can go through nearly 30 people and organisations, including more than 200 different interactions and communications among them [8].

Visibility within the supply chain is a key business challenge, because end to end supply chain transparency and visibility can help model the flow of products from raw materials to manufacturing, testing, and finished goods, enabling new kinds of analytics for operations, risk and sustainability [1]. This leads to more informed decision making and potentially improved performance [11][20]. For example, currently, manual interaction is required between companies if a container suddenly has to be assigned to a different carrier, resulting in business process inefficiencies that can be solved using supply chain visibility. Another example is the optimisation of planning processes by predicting logistics processes as a result of increased visibility in the supply chain [9].

Achieving this visibility is extremely difficult as it requires sharing data between many companies. These companies are heterogeneous, meaning they operate differently and have different IT systems with different data models[24][20]. Furthermore, every time they want to share data, they need to agree on how to give their machines and users access to this data and how to trust each other with this data [9]. Other initiatives have failed as they can not achieve this trust, security, access control and a common interface as data model. Moreover, companies benefit from information asymmetry as they can extract value from this difference in information. While supply chain visibility may improve business performance, companies need to be convinced to give up the benefits of information asymmetry.

A substantial amount of research has been done regarding supply chain visibility, but it proves to be a challenge to achieve as there is still no final solution. Technology pushes such as blockchain technology open up new possibilities in achieving important aspects of supply chain visibility such as trust and security. In a blockchain, trust is gained through decentralisation where no single party has all the power, as each participant holds the data and consensus has to be achieved by the rest of the network in order to reach consensus. Additionally, security is gained through cryptography and this consensus mechanism. Thus, a blockchain implementation could provide a solution for supply chain visibility [25][1][14]. In addition to improving supply chain visibility, blockchains could also allow for operational improvements (for example, less IT staff required to maintain the system). Challenges are data ownership and intellectual property being difficult to define, and protecting commercially sensitive information and privacy [25]. This access control is an important aspect that should be considered. Furthermore, it is difficult to share information in global supply chains due to many different code schemes [14].

2. RESEARCH QUESTIONS

In order to benefit from supply chain visibility and other blockchain features, these challenges need to be overcome. This paper will explore the possibilities of overcoming the

31^{th} *Twente Student Conference on IT* Jul. 5^{nd}, 2019, Enschede, The Netherlands.

challenging aspect of user access control in a blockchain-based supply chain visibility solution.

We will try to resolve the following research question:

RQ How can user access control be implemented in a blockchain for supply chain visibility?

And in order to do so, we will answer the following sub-questions:

RQ1 What are the requirements for user access control for supply chain visibility?

RQ2 Which existing user access control method could be used for such an implementation?

RQ3 Which existing blockchain could be used for such an implementation?

RQ4 Which components are required to create a prototype of such an implementation?

2.1 Methodology

In order to conduct this research we will first solve the sub-questions (**RQ1, RQ2, RQ3** and **RQ4**) by means of exploring existing literature and solutions. The literature will be searched using keywords such as *"supply chain visibility"*, *"blockchain"* and *"user access control"*, and combinations of those keywords using *AND* modifiers. Literature will be chosen on aspects such as relevance, citations and date published. Newer articles are preferred in order to capture the state of the art.

With this knowledge, a prototype implementation will be made that provides user access control and data storage in a decentralised manner. The development will mainly consist of experimentation and attempting to connect the various components. With the knowledge gained from the development of this prototype, a system architecture will be made showing these components and their connections. Validation will be discussed using internal and external validity [17].

This would demonstrate the possibilities of such an implementation and research whether the challenges regarding user access control in this context can be overcome. The resulting system architecture will answer the main research question **RQ**.

3. BACKGROUND

3.1 Supply chains

3.1.1 General

Carter et al (2015) describe supply chains as follows: A supply chain can be seen as a network of nodes and links. Each node is an agent with the ability to make decisions and aims to maximise its own gain within its parameters. A supply chain is relative to a particular product and agent. In this context a product is either an input or an output of the agent, which physically moves in or out of the node. Links are connections between the nodes, such as transportation. Each agent tries to focus on centrally controlling their operations in order to increase performance for their own benefit. However, each node is bounded by its visible nodes. A node is visible to another node if the latter has knowledge of existence, location and activities, of the first node. The supply chain often continues beyond this visible horizon and there are additional nodes and links the node is unaware of. As a result, the agent (node) has no choice but to accept what happens beyond the visible range.

Furthermore, support supply chains exist, consisting of supporting nodes such as financial institutions, brokers and transportation [3].

3.1.2 Supply chain visibility

This visibility in the supply chain is described by Baratt (2007) [11] as *"the extent to which actors (agents) within a supply chain have access to or share information which they consider as key or useful to their operations and which they consider will be of mutual benefit"*. It is a key business challenge which enables new kinds of analytics for operations, risk and sustainability [1], potentially leading to improved performance due to more informed decision making [11].

3.2 Blockchain

3.2.1 General

Blockchain is a distributed ledger technology aiming to achieve three goals: anonymity, an unchangeable record and independence of any central or trusted authority. These goals are achieved by means of three main components. The first being the ledger, which is a series of blocks that form the public record of transactions and the order of these transactions. The second component is the consensus protocol, which allows members of the community to agree on the values stored in this ledger. This is often a Proof of Work mechanism, where so called miners have to perform calculations (work) in order to reach consensus. And finally, a digital currency forms the third component and provides the reward for those willing to do work of advancing the ledger [10].

3.2.2 Smart contracts

Blockchain functionality may extend beyond mere transactions by means of smart contracts and decentralised applications. The Ethereum blockchain, for example, is a blockchain with a built-in Turing-complete programming language, allowing anyone to write smart contracts and decentralised applications where they can create their own arbitrary rules for ownership, transaction formats and state transition functions [21].

3.3 Access control

Access control concerns determining the allowed activities of users (actors) in a system, where every attempt by a user to access a resource (object) in the system is mediated. Access control systems can be implemented in an information technology infrastructure in many places and different levels. Three main types of access control are defined. Discretionary Access Control (DAC) determines access by the owner of an object or other authorised actors. Mandatory Access Control (MAC) relies on a central authority instead of the object owner to determine access to an object. Furthermore, the owner can not change access rights. Role-Based Access Control (RBAC)[5] determines access based on the role of an actor in an organisation [22].

4. RELATED WORK

4.1 Practitioner relevance

A number of companies are currently working on blockchain implementations for logistics. IBM and Koopman have conducted a case study [7] on using blockchain to gain real-time visibility, reduce fraud, eliminate paperwork, accelerate deliveries and cut supply chain costs. Their results were a potential 775 million euros in savings in the

EU, potential real-time decision making due to increased transparency and reduced commercial friction and fraud as a result of increased trust.

IBM and Maersk [8] have also collaborated on a blockchain solution with similar goals of increasing transparency and providing secure information sharing, with the potential to save the industry billions of dollars.

Provenance's white paper [16] describes a prototype that, using blockchain, provides secure traceability of certifications and other information of physical products in supply chains. This provides proof of the authenticity and origin of a product.

A survey from Gartner [4] states 90% of blockchain-based supply chain projects are failing due to lack of important use cases for the technology. Only 19% of the respondents believed blockchain to be an important technology for their business and only 9% have invested in it. A combination of technology immaturity, lack of standards, overly ambitious scope and a misunderstanding of the impact of blockchain on the supply chain causes most of these projects to remain pilot projects.

iShare [9], is a current initiative with similar goals as this research, attempting to provide supply chain visibility. However, they do not use blockchain technology. Instead, trust is gained from being part of the iShare network, requiring an agreement with the iShare foundation. This agreement includes committing to rules on, for example, technical requirements and how to deal with data confidentiality. They use OpenTripModel as data model.

4.2 Literature

Various researchers have looked at the possibilities blockchain technology can offer. Ter Stege (2018) [19] concludes blockchains have a potential to disturb logistics on the long term and to be applied on the short term. He concludes especially track & trace can benefit from a blockchain implementation and that higher efficiency can be achieved if processes are redesigned and automated for optimal improvements.

Abeyratne and Monfared (2016) [1] propose a system for a blockchain based supply chain management system and identify various actors and processes. This concept is visualised in Figure 1. Their concept system includes a high-level overview of authentication, validation and storage. The system shows a blockchain component with which various stakeholders interact, guarded by an authentication component. The blockchain, containing the relevant data, is accessed through this authentication component by means of a client or directly by a product. This allows for users to interact and influence the data through a client, and for a product to influence the data by itself using IoT applications such as RFID tags. Furthermore, they conclude there are cultural and technical challenges to overcome, but the benefits and impact on the environment are sufficient motivation to progress.

Nakasumi (2017) [14] states information in supply chains is one of the most valuable resources for manufacturers in order to build competitive supply chains. Information asymmetry between actors in the supply chain increases the severity of capacity risk, resulting in lower efficiency. He identifies the following benefits of information sharing in the supply chain: clarification, automatic supplying, auto-selecting of distributor, capacity optimisation, optimisation in transportation, reduction of sale opportunity loss and on time collection and delivery. Furthermore, he concludes it is difficult to share information in a global supply chain due to many different code schemes.

5. RESEARCH

5.1 Requirements

In order to create a blockchain implementation for supply chain visibility with user access control, it is important to know the requirements for such as system. Trust, security and decentralisation have already been named as important aspects of such an implementation, other important aspects exist. Abeyratne & Monfared (2016) [1] propose a blockchain based system for supply chain visibility and identify various actors and types of data to be shared amongst these actors. Identified actors include registrars, certifiers & standards organisations, producers & manufacturers, retailers, distributors, consumers, and waste management. Furthermore, this proposed system includes important aspects such as authentication, validation and storage. An important and useful aspect of their proposed system is how data entries are divided into sub-entries, where access can be given or denied to an actor for each sub-entry. This allows for sharing only a portion of the data with an actor and thus for very precise control of data access. Authentication is performed by means of public and private keys, a widely used and secure method of authentication in information technology. A high-level overview of the proposed system can be seen in Figure 1.

5.2 Access control

Controlling who has access to which data is an important aspect of this system as it involves potentially sensitive data and many parties and stakeholders. Not everyone who may see the data should be able to edit it, and thus clear policies have to be defined. Therefore, **RQ2** aims to find out which access control method should be used for this research. Various access control methods exist, as discussed in Chapter 3.3, however there is one method which is capable of implementing all of them and more: Attribute-Based Access Control (ABAC) [26][23]. A widely used implementation of ABAC is XACML [15], an XML-based standard defining various components to manage and perform attribute-based access control. Maesa et al. (2019) [6] have implemented and tested XACML on the Ethereum blockchain using smart contracts, thus resulting in a fully decentralised access control system with the flexibility of ABAC.

The fact that this ABAC implementation using XACML is flexible, already widely used and can be decentralised using smart contracts make it a perfect choice for this research. This technology decision answers **RQ2**.

5.3 Blockchain implementation

As this research aims to decentralise supply chain visibility using blockchain technology, an important research subquestion is **RQ3**, which aims to find out which existing blockchain implementation may be of use to this research.

As mentioned in the previous chapter (*Chapter 5.2*), Maesa et al. (2019) [6] present an implementation of XACML attribute-based access control on the Ethereum blockchain [21]. The smart contracts of this blockchain allow for the decentralised execution of code, thus opening up the possibility to decentralise the access control system as demonstrated in their implementation. Thus, the Ethereum blockchain is a perfect choice for this research. However, while the Ethereum blockchain provides the smart contracts feature for the access control aspect, it does not include a proper method of storing data.

As supply chain visibility focuses on sharing data between various stakeholders, this data needs to be stored somewhere. An important requirement is that this data is also decentralised in order to make sure it is immutable, secure and can be trusted. Conventionally, data is stored in a database which can be queried. Immutable distributed databases exist which combine the properties of a blockchain, such as immutability, decentralisation and data ownership, with the properties of a database, such as indexing & querying of structured data, high transaction rate and low latency. An existing implementation for this is BigchainDB [2]. BigchainDB includes a method of storing *assets* with changeable *metadata* and data ownership by means of private keys. Other noteworthy features[1] include MongoDB queries to search all content, low latency, customisability, Byzantine Fault Tolerance (up to one third of the notes can be experiencing faults and the rest of the network will still function) and an open source code.

These features make the Ethereum and BigchainDB blockchains very good solutions for this research. Furthermore, it is out of the scope of this research to look at and compare all existing blockchain implementations. This technology decision answers **RQ3**.

5.4 Data model

In order to allow communication between companies, these companies need to agree on how to store and request data. This requires a data model providing rules on how to format this data and specifying an API on how to interact with the data. In the logistics sector, OpenTripModel[2] (OTM) is an open standard which is free, lightweight and easy to use. OTM is already used by 3 large-scale shippers, 20 transport carriers within the Netherlands, the Dutch postal service provider and local road authority administrations.

This simplicity, openness and usage by existing companies makes it a solid choice for this research. Furthermore, like with the blockchain implementation, it is out of the scope of this research to look at and compare all existing data models.

This is one of the requirements for supply chain visibility, as without it, companies can not communicate with each other and thus no visibility can be achieved. Thus, this adds to sub-question **RQ1**.

6. PROTOTYPE

In order to gain more insight into the possibilities of user access control on the blockchain for supply chain visibility, a prototype will be developed. This chapter will discuss the identification of components and the development of the prototype.

6.1 Components

As a result of the research sub-questions, answered in Chapter 5, a number of components can be identified which form the system for user access control on the blockchain for supply chain visibility. With that knowledge, this chapter will answer **RQ4**. Users interact with a *server* through a *client*. This client may be a website or a smartphone application and does not contain any logic other than sending, showing and receiving data, and attaching a cryptographic key to the sent data to provide the identity of the user for authentication. Processing of the data, such as authentication, access control, validation and storage, happens on the server side.

[1]https://www.bigchaindb.com/features/
[2]https://www.opentripmodel.org/

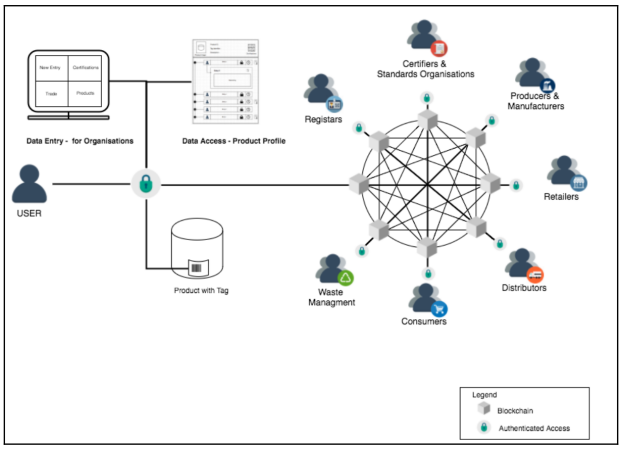

Figure 1. Proposed concept by Abeyratne and Monfared (2018) [1]

The server consists of multiple sub components that provide these functionalities by communicating with each other.

6.1.1 Communication

First, data needs to be sent and received. A widely used, secure (with HTTPS) and flexible method of achieving this is through a HTTP REST API. OpenTripModel, the chosen data model for this research, is defined using the OpenAPI[3] specification, which can be used to generate a client and REST server in a variety of programming languages. These support sending and receiving data formatted in either XML or JSON data formats. The decision was made to opt for JSON as it is believed to provide a more human-readable and thus user friendly syntax and is often used in web applications.

6.1.2 Client

The client will be generated in JavaScript, as JavaScript can run natively in every browser (both desktop and smartphone). Combined with HTML and CSS, a web-based user interface can easily be created. Using a web development framework such as Bootstrap[4], this interface can be made responsive to fit all device sizes and thus allow for increased accessibility.

6.1.3 Server

As the server contains a variety of different components that need to interact with each other, it is important to generate it in a programming language that is supported by all or most of these components. Therefore, the decision was made to choose Python, which is easy to use and deploy as it does not require a compiler or an IDE (Integrated Development Environment). Furthermore, the HTTP REST server, BigchainDB driver and Ethereum blockchain interface are all available in Python, making this a solid choice. Many packages are available to aid in development, providing functions for, for example, cryptographic operations.

6.1.4 Authentication & user access control

In order to provide proof of identity, a user needs to be able to authenticate themselves. This is commonly done using a username and password, but this is inefficient as the password will have to be sent with every request, and unsafe as a password is relatively susceptible to hacking.

[3]https://www.openapis.org/
[4]https://getbootstrap.com/

The system proposed by Abeyratne & Monfared (2016)[1] uses public-key cryptography for authentication. Upon user registration, a public and private key are generated for the user. A user is identified using their public key, and may interact with the network using their private key - together providing secure authentication.

Besides identifying a user, the system has to decide if a user has rights to view or edit data in the system. This is where the implementation of Maesa et al. (2016) [6] can be implemented to provide ABAC with XACML in a decentralised manner. In their implementation the Ethereum blockchain is used and the access control logic is implemented using smart contracts.

Thus, this component exists on the Ethereum blockchain in smart contracts, which use the Solidity programming language, and receives a public or private key for authentication and an identifier for a data object and the desired action to this data object (view or edit). The smart contracts compare the user and data identities, together with the desired action, against the policy of that data object. These policies may be stored on the BigchainDB blockchain, as this blockchain is specialised for the purpose of database-like storage. This will be further discussed in Chapter 6.1.6.

The Ethereum blockchain can be interfaced with using Python as to provide integration possibilities with the rest of the system.

6.1.5 Validation

The data (be it received or to be sent), needs to be validated in order to make sure correct data is entered into the fields. A field that may only contain a string should not contain an integer, and requests that violate this requirement need to be refused. While the OpenAPI specification of the OpenTripModel data model does include a description of fields and their data types, it is unknown if the generated server actually performs validation as a lot of the logic happens behind the scenes in external libraries and the documentation does not seem to be clear about this either. The importance of this component is recognised, however, and plays an important part in ensuring integrity of the data.

6.1.6 Storage

The storage of logistics data and access control policies takes place on the BigchainDB blockchain, which combines the characteristics of a database with the blockchain. Data is stored using transactions, which contains elements such as owners, an asset, and metadata. BigchainDB assets can be 'linked' by referencing the ID of another asset. With this, it is possible to store data and reference an XACML access control policy, which the authentication & access control component evaluates in order to decide on the access decision.

Data stored on a BigchainDB blockchain is just plain data and readable by everyone who has access to the blockchain. Thus, it is important to encrypt this data before it is stored and decrypt it when requested, with cryptographic keys of the owners. This encryption and decryption may happen in the storage component, but could also take place in the authentication component or REST API component.

Furthermore, while the system aims to improve transparency, transaction data (who submits, requests, edits or transfers data) is sensitive data that should not be known to everyone as it could indicate the activity of a company or partnerships between companies. Whether to share this information should be up to the company to decide. Transactions contain a public key to identify parties that interacted with an asset, and these public keys (identities) are not hidden by default. A system may be implemented to provide this, but research is still being performed on how to achieve this for BigchainDB.

A Python driver is available for BigchainDB, allowing for integration with the rest of the system.

6.2 Development

Development was started by evaluating each component separately and becoming familiar with the mechanics using the documentation and experimentation.

Many obstacles presented themselves during the development of the prototype. First, the BigchainDB Python driver did not work on Windows, requiring wrapping it in a Docker container as to place it in a virtual Linux environment.

Second, the BigchainDB testnet became unavailable, requiring the self-hosting of a node. This did not work on the Windows machine either and a Linux server had to be set up.

Third, part of the reason why OpenTripModel was chosen as a data model is because it is specified using the OpenAPI specification. OpenAPI (formerly known as Swagger) is an open specification supported by large IT companies such as Google, Microsoft and IBM[5]. With OpenAPI, an API can be specified with elements such as endpoints, request types, data types and examples. This specification can be used to generate documentation, a client and a server, allowing for supposedly easy development. However, the predicted ease of use turned out to be misplaced, as generating the Python server and JavaScript client code using the OpenAPI specification for OpenTripModel did not work as the file contained errors. These errors did not show up in all OpenAPI validators and the one where it did show up did not indicate the line number of the error. Eventually the errors were solved using a validator from IBM[6] which provided more detailed feedback. Furthermore, the JavaScript client was difficult to use as it required Browserify to build the file on each change in the code, and automating this using GulpJS did not seem to work.

Fourth, the OpenAPI server did not receive JSON objects on a PUT request properly, indicating that the object was missing while the HTTP request body clearly did contain it. This was solved by removing this object requirement, but this is a bit of a dirty fix.

Once these obstacles were overcome, some of the components could be connected. The client and server can communicate, and within the server a GET request to an endpoint can successfully query for the appropriate data on the BigchainDB blockchain and return this to the client. However, it lacks validation and also the user access control component could not be implemented due to lack of time, developers and knowledge of the Ethereum blockchain and its Solidity programming language. With more time and resources, the prototype could have been extended by requiring identification of the user before processing the request, and checking the identity and request against an XACML-based ABAC system on the Ethereum blockchain.

However, the development and its problems lead to experience which contributed to the research as the specific

[5]https://www.openapis.org/membership/members
[6]https://github.com/IBM/openapi-validator

workings and characteristics of each component were discovered. Insights were gained in how to connect the components and where challenges still lie.

7. RESULTS

While the developed prototype is not a fully functioning system, but rather a collection of components, the knowledge of identifying and working with these components did result in insights of how such a system should be built up and what connections are to be made. The attempt to develop a prototype resulted in a model for the architecture of the system. This model can be found in Figure 2. It builds upon the model of Abeyratne & Monfared (2016) [1] and provides a more detailed overview of the internal components and technologies that may be used, such as XACML for user access control and BigchainDB for storage.

As can be seen in Figure 2, the resulting model identifies two blockchains on which processing and storage take place, and a REST API as intermediate interface between these blockchains and a client. This decentralises both the user access control and storage processes. In the model, the REST API is depicted as a separate component due to it not being a blockchain implementation that can run on, for example, the Ethereum blockchain. It could be possible to adapt the source code of BigchainDB nodes to each include the required REST API component, which would also decentralise the REST API. However, this would result in a custom version of BigchainDB, resulting in higher development costs to keep up with features, security patches and bug fixes.

The system works as follows. Through a client, a user identifies themselves with a cryptographic key and performs a request to an endpoint of the API, following the specification of OpenTripModel. An example would be sending a GET request to the endpoint */locations/*{*uuid*} to request information about a location with a certain *uuid*. This information is sent to the XACML access control system on the Ethereum blockchain and evaluated by its sub components. The Policy Decision Point (PDP) searches for a policy on BigchainDB that matches the requested asset identifier and evaluates the policy. The result is returned through the Policy Enforcement Point (PEP) and depending on this outcome the asset may be queried from the BigchainDB blockchain and decrypted so that the user may view its contents.

Similarly, data owners can manage policies through the same client but instead of the PEP, the Policy Administration Point (PAP) is addressed and appropriate actions are taken.

7.1 Validation

The design can be validated on internal validity, trade-offs and external validity [17]. The internal validity questions if the design, implemented the problem context, would satisfy the identified criteria. The identified criteria are user access control, trust, security and a common interface for data. These are achieved in the system by each of the components. User access control through XACML, trust and security through the blockchains and a common data interface through OpenTripModel. Thus, in the domain of supply chain visibility, the proposed solution should have the desired effects and these effects should satisfy the stakeholder criteria.

Regarding trade-offs, components in the designed system may be switched for alternatives that achieve the same functionality or provide greater functionality and the system should still satisfy the criteria.

External validity should also be applicable. The system is built to provide access control with a high level of trust and security. The system may thus also be applied in other domains where these requirements exist, such as the medical sector. Only the data model would have to be changed.

7.2 Discussion

During the research and development of the prototype, it became clear why there is no existing implementation for supply chain visibility that uses blockchain technology combined with user access control. Aside from the obstacles that presented themselves during development, other obstacles remain that still stand in the way of supply chain visibility.

Regardless of the technology push of blockchain technology that has emerged in the past couple of years, and the potential it may have to solve supply chain visibility[25][1][14], the current (used) implementations still lack important features. For example, in order to provide supply chain visibility, it is important that interactions to the blockchain are anonymous as to protect matters such as business connections or an indications of activity within the company.

Another important obstacle is the fact that companies are heterogeneous and thus each operate differently[24]. This means different processes, data formats, IT systems and IT budgets. A company will have to adapt to a common standard in order to use the system, and this may be difficult as a result of these differences.

Controlling data access and ownership is an issue that spreads across more than just the domain of supply chain visibility and may also be applied to a domain such as healthcare. Sensitive information is at stake and thus there has to be a guarantee that this will not fall into the right hands. Again, the blockchain may offer a solution here but only with appropriate access control and encryption methods.

However, companies do want to collaborate and share data in order to improve their business processes and achieve higher efficiency in their processes [18].

This willingness is mentioned in the paper of iShare [9], which was briefly mentioned in Chapter 4.1 as an initiative with similar goals regarding supply chain visibility. While they also use OpenTripModel, they do not use blockchain technology. Instead, their trust and security are based on agreements. While this may be a solution, there is still reliance on trusting a central authority: the iShare foundation. And who is to say these companies keep to the rules? Perhaps the concepts of iShare and this research can be combined to achieve a working system for supply chain visibility.

8. CONCLUSION

The logistics sector could benefit from supply chain visibility and the proposed system in this paper could be a possible solution, attempting to overcome some of the challenges that are currently present, such as trust, security and user access control. As a result, business processes may become more efficient and less resources are wasted by means of increasing visibility, allowing for more timely reactions to a change in situations.

The proposed system includes OpenTripModel as a data

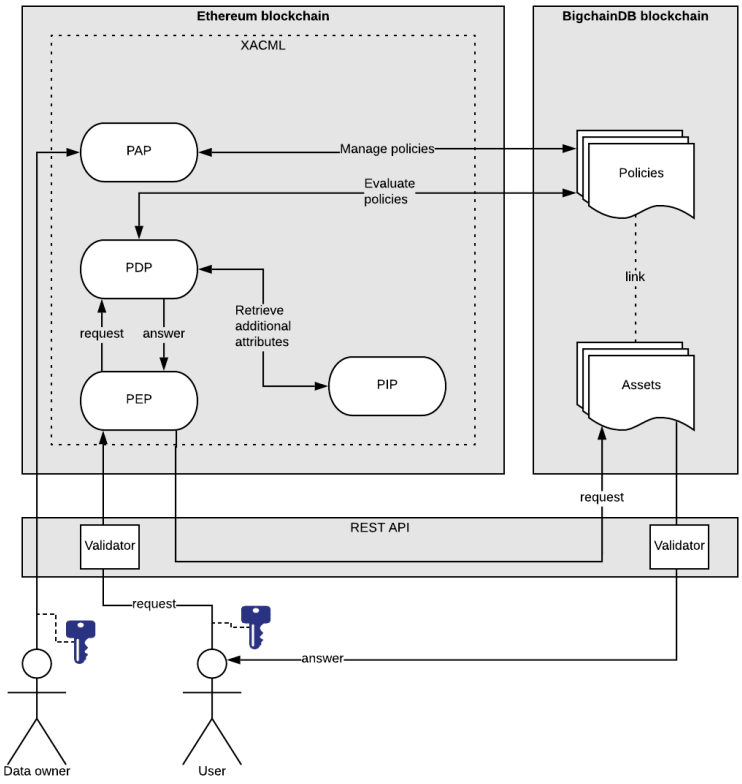

Figure 2. Resulting model of components forming the system

model, because it is an open standard, already implemented by a number of logistics companies and is built with the OpenAPI specification, allowing for easy server and client code generation. In order to achieve trust and security, two blockchains are included, Ethereum for providing decentralised execution of code using smart contracts, and BigchainDB for providing database characteristics such as querying of structured data, low latency and high transaction rate. Access control is achieved by means of XACML, which can be decentralised on the Ethereum blockchain using smart contracts. XACML provides ABAC (attribute-based access control), a powerful access control mechanism capable of implementing and supplementing existing access control mechanisms such as DAC, MAC and RBAC.

From this, various insights were gained and obstacles were identified. Each of the individual components turned out to have its own obstacles in getting them to function as desired. Moreover, connecting the components was difficult and due to lack of time and developers. Blockchain technology still has its limitations, such as lacking anonymity in transactions and encryption of the data (aside from known limitations such as scaleability, resource efficiency and transaction speed).

In order for these challenges to be overcome, a company or collection of companies has to dedicate resources, such as time and manpower, for the development of such a system. And even then, the previously mentioned limitations of blockchain technology have to be overcome. And once a fully functioning system has been realised, fulfilling all requirements, it will still be a challenge to motivate companies to migrate their current IT infrastructure to the new system.

8.1 Future work

From the limitations and challenges that have come forth from this research, several topics for additional research came to mind.

Blockchain technology still has its limitations in terms of efficiency with computing resources and scaleability in terms of transaction throughput. While these are known to be actively researched right now, it could be interesting to research how (if at all) the system as proposed in this paper would scale to the size of something as large as the global logistics sector.

A similar issue is the difference in data models. It may be interesting to research how companies can be encouraged to change their data model to a common model such as OpenTripModel. It would be interesting to know how compatible OTM currently is with a variety of large companies, and whether it has to evolve in order to become more compatible, making a transition easier for companies.

It may also be interesting to know the financial impact on implementing a system as proposed in this paper, taking into account the impact of achieving supply chain visibility, and possible savings in the IT department due to not having to maintain a proprietary system. Performing transactions on the blockchain is not free Additionally, it may be interesting to get an estimate of the development, deployment and maintenance costs of the system.

9. ACKNOWLEDGEMENTS

I would like to thank my supervisor, Simon Dalmolen, for his advice, support and feedback. Furthermore, I would like to thank Hans Moonen for his feedback, Jacco Spek for sharing his knowledge of BigchainDB and Wout Hofman

for sharing his knowledge of current initiatives regarding blockchains in supply chains.

10. REFERENCES

[1] S. A. Abeyratne and Monfared". Blockchain ready manufacturing supply chain using distributed ledger. *International Journal of Research in Engineering and Technology*, 05(09):1–10, 2016.

[2] BigChainDB GmbH. Bigchaindb 2.0 the blockchain database. 2018.

[3] Carter, Craig R. and Rogers, Dale S. and Choi, Thomas Y. Toward the theory of the supply chain. *Journal of Supply Chain Management*, 51(2):89–97, 2015.

[4] D. C. Klappich, et al. Gartner predicts 2019: The future of supply chain operations. 2018.

[5] D. Ferraiolo, R. Kuhn, R. Chandramouli. Role-based access control. 2013.

[6] D. Maesa, P. Mori and L. Ricci. A blockchain based approach for the definition of auditable access control systems. *Computers Security*, 84:93–119, 2019.

[7] IBM. Koopman case study. `https://www.ibm.com/case-studies/koopman-blockchain-logistics`, accessed 13 june 2019.

[8] IBM. Maersk and IBM unveil first industry-wide cross-border supply chain solution on blockchain. 2017.

[9] iSHARE. ishare en opentripmodel: noodzakelijke bouwblokken voor delen van data. 2019.

[10] J. J. J Waldo. A hitchhiker's guide to the blockchain universe. *Communications of the ACM*, 62(3):38–42, 2019.

[11] M. Barratt, A. Oke. Antecedents of supply chain visibility in retail supply chains: A resource-based theory perspective. *Journal of Operations Management*, 25(06):1217–1233, 2007.

[12] M. E. Porter, J. E. Heppelman. How smart, connected products are transforming competition. *Harvard Business Review*, 92(11):64–88, 2014.

[13] M. E. Porter, J. E. Heppelman. How smart, connected products are transforming companies. *Harvard Business Review*, 92(10):97–114, 2015.

[14] M. Nakasumi. Information sharing for supply chain management based on block chain technology. *2017 IEEE 19th Conference on Business Informatics (CBI)*, 01:140–149, 2017.

[15] OASIS. extensible access control markup language (xacml) version 3.0. 2013. `http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html`, accessed june 20th 2019.

[16] Project Provenance Ltd. Provenance white paper. 2015.

[17] R. Wieringa. Design science as nested problem solving. 2009.

[18] T. C. Du, V. S. Lai, W. Cheung, X. Cui. Willingness to share information in a supply chain: A partnership-data-process perspective. *Information Management*, 49:89–98, 2012.

[19] S. ter Stege. Blockchain in logistics : Is blockchain in logistics hyped, or has it true potential to be a game changer? June 2018.

[20] The Digital Transport and Logistics Forum. Data sharing in supply and logistics as commodity. 2018. `http://ec.europa.eu/transparency/regexpert/index.cfm?do=groupDetail.groupMeetingDoc&docid=14275`, accessed june 24th 2019.

[21] V. Buterin. A next generation smart contract decentralized application platform. 2018.

[22] V. C. Hu, D. Ferraiolo and D.R.Kuhn. Assessment of access control systems. *US Department of Commerce, National Institute of Standards and Technology*, 7316, 2006.

[23] V. C. Hu, D. Ferraiolo, R. Kuhn. A unified attribute-based access control model covering dac, mac and rbac. *Special Publication (NIST SP)*, 800-162, 2014.

[24] W. Hofman, J. Spek, S. Dalmolen. Supply chain visibility ledger. *6th International Physical Internet Conference*, 2019.

[25] Wang,Y. and Singgih,M. and Wang,J. and Rit,M. Making sense of blockchain technology: How will it transform supply chains? *International Journal of Production Economics*, 211:221–236, 2019.

[26] X. Jin, R. Krishnan and R. Sandhu. A unified attribute-based access control model covering dac, mac and rbac. *Lecture Notes in Computer Science*, 7371:41–55, 2012.