

Antti Repo

**UTILIZING BLOCKCHAIN TECHNOLOGY IN A  
ROAD TOLL ARCHITECTURE**



UNIVERSITY OF JYVÄSKYLÄ  
FACULTY OF INFORMATION TECHNOLOGY  
2019

## ABSTRACT

Repo, Antti

Utilizing blockchain technology in a road toll architecture

Jyväskylä: University of Jyväskylä, 2019, 68 pp.

Information systems science, master's thesis

Supervisor: Veijalainen, Jari

Blockchains are a fairly new technology with few real-world applications beyond Bitcoin, the first mainstream cryptocurrency. Road tolling is a rather widely used method of monetizing transportation infrastructure projects by collecting payments from the infrastructure users. In Norway, there is a road tolling system called AutoPASS, of which multiple companies are a part of, working together and some providing essentially the same service. This can cause data redundancy and creates a need to transfer data between companies efficiently. This raised the question whether this could be done in a decentralized manner using blockchains. In this thesis the feasibility of blockchain technology in a road toll architecture was evaluated on a high level. Specifically, the Norwegian road tolling system AutoPASS was investigated on a business and technological level and using a blockchain feasibility model it was evaluated whether a blockchain could be utilized in said setting. The research was done as a literature review combined with constructive research utilizing a design science methodology. The result of the research was that it can be justified and feasible to utilize a public permissioned blockchain in a road toll system such as AutoPASS.

Keywords: databases, peer-to-peer networks, blockchain, road tolls, AutoPASS

# TIIVISTELMÄ

Repo, Antti

Lohkoketjuteknologian hyödyntäminen tietulliarkkitehtuurissa

Jyväskylä: Jyväskylän yliopisto, 2019, 68 s.

Tietojärjestelmätiede, pro gradu -tutkielma

Ohjaaja: Veijalainen, Jari

Lohkoketjut ovat kohtalaisen uusi teknologia, joilla ei vielä ole juurikaan todellisia käytännön käyttökohteita lukuun ottamatta Bitcoinia, ensimmäistä valtavirran kryptovaluuttaa. Tietullit taas ovat melko yleinen tapa rahoittaa liikenneinfrastruktuuriprojekteja keräämällä käyttömaksuja infrastruktuurin käyttäjiltä. Norjassa on käytössä tietullausjärjestelmä nimeltään AutoPASS, johon kuuluu useita yhteistyössä olevia yrityksiä, jotka tarjoavat periaatteessa samaa palvelua. Tämä voi aiheuttaa datan päällekkäisyyttä ja luo tarpeen siirtää dataa yritysten välillä tehokkaasti. Tästä nousi kysymys voisiko tämän toteuttaa hajautetusti lohkoketjujen avulla. Tässä tutkielmassa lohkoketjujen soveltuvuutta tietulliarkkitehtuuriin arvioitiin korkealla tasolla. Erityisesti norjalaista AutoPASS-tietullijärjestelmää tutkittiin liiketoiminta- ja teknologianäkökulmista ja käyttäen lohkoketjujen soveltuvuusmallia arvioitiin, voisiko lohkoketjua käyttää kyseisessä ympäristössä. Tutkimus tehtiin kirjallisuuskatsauksena sekä hyödyntäen suunnittelutieteellistä metodologiaa. Tutkimuksen tuloksena voitiin todeta, että voi olla perusteltua ja soveltuvaa käyttää julkista luvanvaraista lohkoketjua Autopassin kaltaisessa tietullijärjestelmässä.

Asiasanat: tietokannat, vertaisverkot, lohkoketju, tietullit, AutoPASS

## FIGURES

FIGURE 1 Joining the Department and Employee tables in a relational DB allows accessing data in both tables simultaneously .....	13
FIGURE 2 Simplified network structure based on the P2P model in which resources are shared by interconnected nodes (“peers” or “servents”) without a central entity such as a server. The computer icons depict network nodes, and the black lines depict connections between nodes. ....	22
FIGURE 3 Network structure based on the client-server model, in which individual clients request resources and services from a central administrative system such as a server. Computer icons depict network nodes, server icon depicts a central administrative system, and black lines depict connections between nodes and central system. ....	22
FIGURE 4 An overlay network (Dunaytsev et al., 2012; Eberspächer et al., 2004) .....	23
FIGURE 5 P2P Organizational borders .....	27
FIGURE 6 Blockchain block structure (a) and smart contract structure (b) (Hong, Wang, Cai & Leung, 2017).....	33
FIGURE 7 DSRM sequence (Peffer et al., 2006) .....	46
FIGURE 8 A simplified model of the parties in the AutoPASS Samvirke network and their legal and contractual relations. (AutoPASS, 2019) .....	50
FIGURE 9 AutoPASS charging point (Wærsted, 2005).....	52
FIGURE 10 Operation principles among participants and the blockchain in the Unified Tolling Network (Milligan partners, 2019).....	57
FIGURE 11 Flow chart for determining whether blockchain is appropriate for solving a problem (Wüst & Gervais, 2017) .....	59

## TABLES

TABLE 1 Summary of P2P and client-server network types (Eberspächer et al, 2004). In the diagrams the computer icon depicts a network node, the server icon depicts a central administrative system, and black lines depict connections between nodes and the central system. Dashed lines in the centralized P2P diagram depict search queries to a centralized system.....	26
TABLE 2 Data change governance (Lewis, 2017).....	36
TABLE 3 CPE-CS payment data files (Pedersli, 2012).....	52
TABLE 4 AutoPASS stakeholders and duties .....	54

# TABLE OF CONTENTS

ABSTRACT.....	2
TIIVISTELMÄ .....	3
FIGURES.....	4
TABLES.....	4
TABLE OF CONTENTS .....	5
1 INTRODUCTION .....	7
1.1 Motivation.....	8
1.2 Research questions .....	9
1.3 Research methods.....	9
2 DATABASES.....	11
2.1 Data concepts .....	11
2.2 Traditional database solutions.....	12
2.3 ACID properties.....	16
2.4 Autonomy in databases .....	16
2.5 Cloud services .....	17
3 PEER-TO-PEER NETWORKS .....	20
3.1 Technology .....	21
3.2 Autonomy in P2P networks .....	27
4 BLOCKCHAIN .....	31
4.1 Overview.....	31
4.2 Blockchain technology .....	32
4.3 Key features .....	36
4.4 Four key concepts .....	40
4.5 Autonomy in blockchain systems .....	43
5 DESIGN SCIENCE RESEARCH PROCESS .....	45
6 ROAD TOLL SYSTEMS.....	48
6.1 Road toll systems in Norway.....	49
7 ROAD TOLL SYSTEM ARCHITECTURE DESIGN WITH BLOCKCHAIN .....	54
7.1 Problem identification and motivation.....	54

7.2	Objectives for a solution .....	55
7.3	Design and development.....	57
7.4	Evaluation.....	59
8	DISCUSSION .....	61
	REFERENCES .....	63

# 1 INTRODUCTION

In the last decade or so blockchain technology has paved its way into people's awareness, but there are still few widely adopted applications based on blockchains, excluding for example the cryptocurrency Bitcoin, which, while very well known, has yet to become the widely used means of trade it was hoped to be. Banks, health institutions, and governments have investigated the possibilities provided by blockchain technology, which tells us there is real interest in the technology. For example, in Finland it has been researched whether blockchains could be utilized in the planned social and health service renewal (Salonen, Halunen, Korhonen, Lähteenmäki, Pussinen, Vallivaara, Väisänen & Ylén, 2018), and in the United States blockchain technology has been piloted in voting (Palermo, 2018).

There are many potential use cases for blockchains – in the end, it is basically a way to store data in a decentralized manner – and it sure has its benefits to offer, we just need to identify and research the correct ways and places to apply it. The aim of this study is to investigate whether blockchain technology could be utilized in a road tolling architecture and evaluate its suitability.

The thesis begins with a look into databases and peer-to-peer networks in chapters 2 and 3 to provide a basic understanding on current technologies, building a base for the introduction of blockchain technology in chapter 4. The blockchain chapter covers the main concepts of blockchains, providing insight into the technology's capabilities. Chapter 5 presents the research methodology DSRP, the design science research process, and in chapter 6 road toll systems are examined, with the main focus on the Norwegian AutoPASS road tolling system. In chapter 7 these topics are brought together in the form of evaluating whether and how blockchain technology could be utilized in a road toll setting.

## 1.1 Motivation

In this study the combination of road toll systems and blockchain technology is examined, with the aim of seeing how well these two concepts could fit together. The subject domain was chosen due to two reasons: first, the potential of blockchains in real-world use cases is still not fully clear since widely adopted real-world applications are still quite rare. Also, as a technology with potential in privacy and verifiability aspects, blockchains should be researched further. Second, despite – yet also because – of blockchains having been the subject of considerable hype in the media and road tolls being a frequent topic of public discussion in Finland (Sito Oy, 2016; Helpinen, 2016; Lempinen, 2017), I consider the subject to be quite current and above all interesting.

In many countries the costs of maintaining infrastructure (including – but not limited to – for example highways, bridges, and tunnels) are covered by collecting payments from their users, i.e. drivers or owners of vehicles. There have been plans to introduce road toll collection on some roads in Finland as well, but actual timetables for the project(s) are unclear. The surveillance and collection of road toll payments would likely be executed with an automated system. Typically, the gathering of payments is done in two ways: the first way is by utilizing physical toll booths, where the vehicle must stop in front of a gate and the driver pays the toll officer a fee, deposits money in a machine, or provides evidence of earlier payment to be able to continue the journey. The second method does not require stopping the vehicle. Instead, an automated system either communicates with a transponder device in the passing vehicle to verify or initiate payment transaction, or it uses image recognition to identify the vehicle's number plates and charges the vehicle's owner if no transponder system is installed. Electronic systems like license plate recognition can of course be utilized in physical toll booths as well to enable automatic gate operation.

Cryptocurrencies, the most famous one being Bitcoin, have made blockchain technology known in the last few years. Blockchains are a fairly new method to save data in a decentralized manner enabling direct transactions between users without dependence on centralized actors, all the while providing transparency, security, and data privacy aspects. Blockchains also allow for so called "smart contracts", which allow the performance of credible transactions without third parties. Smart contracts were first introduced by computer scientist Nick Szabo in 1994. Blockchains can be categorized into three generations based on their properties (Casino, Dasaklis, & Patsakis, 2019). The first generation includes applications that enabled digital cryptocurrency transactions; the second generation introduced smart contracts and a selection of applications that extend beyond cryptocurrency transactions; and the third generation blockchain which holds applications in areas that are beyond the previous two generations, such as health, science, government, and Internet of Things. The introduction of blockchains has been described to be as revolutionary as the invention of Internet itself



was, or that blockchain will do to transactions what the Internet did for information (Gupta, 2017), but it seems it will still take some time for that.

Another topic discussed in this thesis is the concept of autonomy in the contexts of databases, p2p networks and blockchains. Who has the control over the data in an organization, and what kind of control is it? In a centralized system setting defining these answers might not be the most difficult thing to do, but in a decentralized system it might prove challenging. The four types of autonomies – organizational, design, execution, and communication autonomy– are discussed in the context of databases, peer-to-peer networks, and blockchains in their respective chapters.

## 1.2 Research questions

The hypothesis is that blockchain technology can propose features for data security, openness, and transparency not presented by commonly used database structures today. Therefore, the research questions are as follow:

- How could blockchain technology be utilized in a road toll system?
  - o Is decentralized technology suitable for a road toll architecture?
  - o What could the architecture of a road toll system utilizing blockchain technology be like?
  - o Is using blockchain technology in a road toll system justified?

To provide answers to these questions, the main issues to research are the ecosystem of a road tolling system: what actors participate in such an ecosystem, what kinds of equipment is utilized, and what kind of data is generated and transferred in the system? The next step would be to examine what blockchain technologies would offer in terms of these requirements, and what properties are required from the blockchain technology.

## 1.3 Research methods

The research was done as a literature review combined with constructive research: articles and research papers about databases, peer-to-peer networks, and blockchains were gathered to provide understanding of the technologies. Then the existing road tolling system were investigated, and the system being used in Norway was selected mainly because documentation of the system was quite comprehensively available. The gathered information was then combined using the design science research process, DSRP, a research model by Peffers, Tuunanen, Gengler, Rossi, Hui, Virtanen and Bragge (2006). The six steps of the DSRP model are

1. problem identification and motivation
2. objectives of a solution
3. design and development

4. demonstration
5. evaluation and
6. communication.

The model is described in more detail in chapter 5. With the help of the DSRP model and its six core steps, a possible blockchain-road toll implementation to the Norwegian road tolling system is considered on a moderately high abstraction level. The developed artifact is a recommendation or a guideline on what kind of blockchain could be feasible to be used in an ecosystem such as the AutoPASS road tolling system.

## 2 Databases

In this chapter traditional database models and architectures are presented and their properties are discussed. The goal is to provide an understanding on the underlying technology virtually all web services and systems utilize in one way or another. Also, in this chapter the concept of autonomy in the context of databases is explored.

All information systems utilize data in some form, and these data resources must be organized and structured in a logical way so that accessing them is easy, their processing is efficient, they can be retrieved quickly, and they can be managed effectively. To efficiently organize and access the data stored by information systems, many kinds of simple and complex data structures and access methods have been devised. (O'Brien & Marakas, 2010.)

Nowadays practically all data you will ever access is stored and organized in some form of database, and O'Brien and Marakas (2010) suggest that if you find yourself asking "Should I use a database?" the question should instead be "What *kind* of database should I use?"

A big quantity of data that is stored in a computer can be called a database, and the basic hardware and software that is designed for managing this data can sometimes be called a database management system ('DBMS'). Not all software used for managing data are DBMSs, though. A DBMS provides the commands to manipulate the database, i.e., the database operations. It is very typical that the DBMS has alongside it large and ever-growing software that can be used to access and modify the stored information. (Abiteboul, Hull, & Vianu, 1995.)

### 2.1 Data concepts

To differentiate between different groupings of data, a conceptual framework of multiple data levels has been developed with which data can be logically sorted into characters, fields, records, files, and databases. (O'Brien & Marakas, 2010.)

A *character* can be a single alphabetic, numeric, or other symbol. It is the most fundamental data element. This is the logical view as opposed to the physical or hardware view of data, according to which the bit or byte is the most basic element. So, from the user's viewpoint, a character is the most basic data element to be manipulated and observed. (O'Brien & Marakas, 2010.)

A *field*, or data item, is the next higher level of data, and it is a collection of related characters. As an example, the characters in a person's name can constitute a name field, and the grouping of numbers in a person's salary amount forms a salary field. A data field usually represents an attribute (a characteristic) of an entity (a person, an object, a place, or an event). (O'Brien & Marakas, 2010.)

A *record* is formed when all the fields that are used for describing an entity's attributes are grouped together. Therefore, a record represents a number of attributes which describe a single entity instance. For example, an employee's payroll data which contains data fields for such attributes as name, social security number, and amount of salary, is a record. Normally in a database, the record's first field is utilized for storing a unique identifier of a chosen type; this is called the *primary key*. This key will be used to identify a unique entity instance and distinguish it from other instances. The value of the key can be anything that suits this purpose. For example, in a student record the student ID number can be applied as the primary key for identifying individual students from other students in the same category. If there is no explicit data to be used as the primary key, the designer of the database may assign the records an extra field containing a sequential number to be used as the key. This way all records will always have a unique primary key. (O'Brien & Marakas, 2010.)

A *data file* in DB context is a set of records that are in relation to each other. A file may sometimes be called a *table* or a *flat file*. A single table can be called a *flat file* when there are no other files related to it. According to O'Brien and Marakas (2010) a database of flat files should not contain anything but data and characters separating the data – the delimiters. In a broader sense, the term “flat file” can refer to a database existing in an individual file containing rows and columns, without links or relationships among records and fields apart from the table structure. But, despite the name, records that are related to each other and are grouped in any way in tabular form (rows and columns) can be referred to as a *file*. For example, the records of a company's employees would often be stored in an employee file. (O'Brien & Marakas, 2010.)

A *database* is a consolidated group of data elements that are related to each other logically. A database integrates records that were stored previously in independent files into a data pool of elements that several applications can utilize. The stored data in a database are usually not locked into any specific application program that utilizes the data or hardware that the data are located on. In short, a database contains data elements that define entities and their relations among one another. All in all, databases can be quite simple: they just are supposed to make the data organized and accessible. (O'Brien & Marakas, 2010.)

## 2.2 Traditional database solutions

O'Brien & Marakas (2010) identify five database structure types: *relational*, *hierarchical*, *network*, *object-oriented* and *multidimensional* models. Out of these five structures the relational model is the most commonly used; the others are not often found in modern organizations.

In early DBMS packages using the hierarchical structure was common. In it the record relationships constitute a tree structure, or a hierarchy: in the traditional hierarchical model, there is a single *root* record and a number of lower level records. This means all the record relationships are *one-to-many*, because all single

elements are related to just one element above it in the tree structure. The root element is the record on the hierarchy's top level, and it is possible to reach any data element in the database by progressing down from the root element and through the "tree branches" until the desired data record is found. (O'Brien & Marakas, 2010.)

The network structure can express more complicated logical relationships than the hierarchical structure. It permits *many-to-many* relationships amongst records, which means that in the network model it is possible to access a data element by following one of many paths of relations. This is because a record or a data element can be in relation to a practically unlimited amount of more data elements. (O'Brien & Marakas, 2010.)

The relational database model is the most commonly used one. In the relational model all data are regarded as being stored as rather uncomplicated tables. These tables may then implement the concept of relations: columns in tables can contain data types, and columns can be related to each other, within tables and across other tables. A table can have multiple copies of the same row, whereas a relation is a set that only contains unique entities.

Figure 1 demonstrates the relational database model by showing how the relationship between the departmental and employee record is established. In the relational model, it is possible to "connect" data in a table with data in some other table in another file, provided that both files have the key attribute, i.e. a common data field or element. This attribute is called the "foreign key". This way a manager, for example, can fetch the name and salary of an employee from the employee table, and the department of the employee from the department table with just one query. This way, by retrieving data from numerous tables new information can be created, even if the tables are physically stored in different locations. (O'Brien & Marakas, 2010.)

Department table			
Deptno	Dname	Dlocation	Dmanager
Dept A			
Dept B			
Dept C			

Employee table				
Empno	Ename	Etitle	Esalary	Deptno
Emp 1				Dept A
Emp 2				Dept A
Emp 3				Dept B
Emp 4				Dept C
Emp 5				Dept B

FIGURE 1 Joining the Department and Employee tables in a relational DB allows accessing data in both tables simultaneously

According to O'Brien & Marakas (2010), the multidimensional model is an alteration of the relational model. It utilizes multidimensional structures for data organizing and expressing data relationships. Multidimensional structures can be

visualized as “cubes of data and cubes within cubes of data”, with each cube side considered as a dimension of said data. (O’Brien & Marakas, 2010.)

O’Brien & Marakas (2010) consider the object-oriented model as one of the essential technologies of new multimedia applications based on the Web. In the object-oriented model we can distinguish between two levels: the schema level and the instance level. The object-oriented schema contains all the object types the database will contain; these types contain the object type name and signature. The instance level consists of operation interface specifications. Object instances consist of the interface operation implementations and the actual data values. The code for operations is not usually replicated to every object instance but is stored into the code repository part of the object-oriented database. This is called *encapsulation*, and it allows the handling of complex data types such as graphics, pictures, audio, and text more effortlessly over other types of database structures. Additionally, *inheritance* is supported by the object-oriented model, which means it is possible to automatically create new objects by replicating characteristics of a *parent* object and adding characteristics of a *child* object. (O’Brien & Marakas, 2010.)

Next, we will take a look at common types of databases, which, according to O’Brien & Marakas (2010), are the operational, distributed, external, and hypermedia databases.

Operational databases are used to store elaborate data that is needed for supporting a company’s business processes as well as operations. These databases can also be called subject area databases (‘SADB’), transaction databases or production databases. Examples of such databases are the customer database, inventory database, human resource database and other databases that contain data generated by business operations. (O’Brien & Marakas, 2010.)

A distributed database is a DB that has its entirety or parts of it replicated or partitioned to network servers at different physical sites. These distributed databases may be situated on servers on the Internet, on corporate extranets or intranets, or on other organizational networks. According to O’Brien and Marakas (2010), making sure the data in the distributed databases of an organization are concurrently as well as consistently kept up to date is a significant challenge of managing distributed databases.

There are advantages and disadvantages to having distributed databases. One key advantage is in data protection: in the case all the data of an organization is stored in only one physical location, an event such as a fire or other damage to the storage devices containing the data could result in devastating data loss. By distributing the databases in several physical locations, the unwanted consequences can be minimized. (O’Brien & Marakas, 2010.)

Another advantage of having a distributed database can be recognized in the requirements for storage. By maintaining the logical relationship that the stored data has with the storing location, it is possible to distribute a massive DB system into smaller size databases. A company that operates across multiple branches can have its data distributed based on the branches, so for example a company’s branch office in Helsinki holds only the data relevant to that location.

Since in distributed systems it is possible to join databases together, all locations can have control over their local data, all the while also allowing for other branch locations to access the company's other databases if necessary. (O'Brien & Marakas, 2010.)

Alongside the advantages in distributed databases, however, there often are disadvantages; the main one being the challenge of data accuracy maintenance. Distributing its database to multiple locations, a company must then manage updating the data in all locations when a change in the data occurs in one location. O'Brien and Marakas (2010) identify two ways of updating data: replication and duplication. (O'Brien & Marakas, 2010.)

According to O'Brien and Marakas (2010) replication means utilizing special software that searches for changes in the distributive database. (The usual meaning for replication, though, is that the same data is stored onto several sites. What O'Brien and Marakas (2010) describe is in fact a special way of keeping replicated data consistent.) When the system has detected the changes, the replication process modifies all the DB's to be identical. Because of the complexity of the process it may take significant amounts of time and computing power, depending on the number of databases to be modified as well as their size. (O'Brien & Marakas, 2010.)

Duplication, in contrast to replication, is less complex. In the process one database is identified as the master and that database is duplicated at another site. Usually the duplication is carried out at a pre-defined time, for example, at night or when usage is at its lowest. The purpose of this is to make sure all distributed locations do not have differences in their data. In the process users are permitted to make changes only to the master database, so that data stored locally does not get overwritten. (O'Brien & Marakas, 2010.)

Duplication, however, can be considered a special case of replication. In distributed databases data can be replicated to multiple sites, but also partitioned to multiple sites. In replication the same rows or portions of rows are stored onto many computers, whereas in partitioning a table can be partitioned and the parts saved on different locations, or different tables can be partitioned and saved in different locations. These can also be combined.

*External databases* are what O'Brien and Marakas (2010) call databases that have content available online, be it with or without charge, that can be accessed via the World Wide Web. When using a search engine such as Google or Yahoo, you are using a large external database.

The fast increase in Internet websites and corporate intranets as well as extranets has greatly added to the use of hypertext and hypermedia document databases. Websites can provide a broad assortment of hyperlinked websites of multimedia content stored in these hypermedia databases, consisting of hyperlinked multimedia pages (text, images, video clips, audio, etc.). (O'Brien and Marakas, 2010.)

## 2.3 ACID properties

In computer science and databases, there is a set of properties called ACID properties. ACID properties are an important set of principles that should be considered in the design of a database management system. ACID stands for Atomicity, Consistency, Isolation, and Durability, and they are a set of properties of database transactions for guaranteeing data validity in case errors, electric power failures, etc. occur. ACID properties are especially important in distributed databases, because when executing a transaction to multiple locations simultaneously, the integrity of the transaction must be maintained, and successful processing must be ensured.

*Atomicity* means that a transaction must be executed entirely or not at all. A database system used in a bank cannot, in a currency transaction situation, take currency from the bank account of the sender A and then not place the correct amount to the account of the recipient B. Therefore, a transaction may consist of multiple parts, but it is still regarded as a single transaction.

An example on atomicity in a banking scenario using the SWIFT system: a transaction is for example the moving of currency from account A in bank X to account B in bank Y. This is regarded as only one transaction, even though it may consist of several components: the owner of account A orders his bank X to pay a certain amount of selected currency into the recipient's account B with his bank Y, and the reimbursement of this transfer might be through a correspondent bank Z. Therefore, the sending bank A notifies the recipient bank B of the funds transfer and sends the cover by bank transfer to the correspondent bank Z. Upon receipt, the corresponding bank lets the recipient bank know about the receipt by confirming the credit. (Veijalainen, Eliassen & Holtkamp, 1992.)

*Consistency* means that a successful transaction moves the consistent state of the database to another consistent state. The new state of the database after a successful transaction must not be a faulty one.

*Isolation* means that a transaction acts as if it was the only transaction in the system, and the database operations are executed one at a time as if in a series - not simultaneously.

*Durability* means that after the results of a transaction that was committed remain in the database until another committed transaction changes them. In the case of various database failures, for example, a power failure or a disk crash, the system cannot erase the committed data.

## 2.4 Autonomy in databases

Organizations create, store and utilize vast amounts of data in their databases. The utilizers can be internal or external: internally, they can be, for example, the employees of the organization, and externally they can be the organization's customers, affiliate organizations, or even competitors (e.g. banks). However, not



every party has the same access to the data, and especially not the same control over it. Therefore, it is important to distinguish the different roles in the access and control of the organizations data. This leads us to the concept of *autonomies*. Merriam-Webster's online thesaurus defines autonomy as "the act or power of making one's own choices or decisions" (Merriam-Webster, 2019).

According to Veijalainen, Eliassen and Holtkamp (1992) databases can be observed to hold properties of autonomy. There are four types of autonomy: O-, D-, C-, and E-autonomy.

*Organizational autonomy* (O-autonomy) means that distinct organizations are not in control of each other despite being in contact with each other in business related matters or otherwise. The organizations have the volition to act on their own. Banks, for example, can simultaneously cooperate and compete. It is often the case that O-autonomous banks also want to remain D-autonomous. (Veijalainen et al., 1992.)

*Design autonomy* (D-autonomy) means that organizations can make their own decisions regarding the systems they decide to utilize. This means that the organizational environment can be heterogenous, which in turn means the possibility of using multiple types of hardware and software solutions such as servers and database management systems. Banks, for example, usually want to decide for themselves what data processing systems they use and who can access them. (Veijalainen et al., 1992.)

*Communication autonomy* (C-autonomy) means that organizations have the autonomy to choose which other organizations they communicate with and when. Two banks in two different time zones might not be able to communicate during office hours, unless the system used by the organization is able to save messages and send them when the recipient's system is ready to receive them. The messages in question can be transaction requests, for example. (Veijalainen et al., 1992.)

*Execution autonomy* (E-autonomy) is one of the consequences of Organizational autonomy. This means that an E-autonomous organization does not necessarily need to process all the messages it receives. Therefore, a bank can refuse giving service for several reasons: lack of trust, erroneous messages, authorization failures, or the message simply does not require taking action. (Veijalainen, 1992.)

## 2.5 Cloud services

Organizations often select the database systems they use based on the system's suitability and price to best cater to their needs. The databases can be maintained either by the organization themselves or by a third party, such as a cloud service provider. In the case that the organization manages both the hardware and software of their own databases, questions regarding data ownership and data access rights may not be too difficult to answer. If, however, an organization utilizes the data services of a third party, for example, an outside cloud service provider for

outsourcing database infrastructure and software, such questions become more important.

Cloud services can be roughly divided into three distinctive categories: infrastructure as a service ('IaaS'), software as a service ('SaaS'), and platform as a service ('PaaS'). In short, an IaaS provider provides scalable virtual machines or storage on demand, SaaS means software hosting in the cloud so the software does not use an organization's local resources, and PaaS is a category of cloud computing services that offer an environment that allows customers to develop, run, and manage their applications without the need to manage the complexities of building and maintaining the infrastructure that may often come when you develop and launch an application. (Butler, 2013.)

Data access and ownership are important questions in using PaaS environments. For example, in a PaaS category of service, questions such as "who actually owns the data?", "who can access the data?", and "who can utilize the data?" are examples of questions that need to be expressed and agreed upon in contracts and service agreements between organizations and cloud database providers.

Amazon.com Inc. (later "Amazon") is an American company offering e-commerce and cloud computing services, which has a subsidiary called Amazon Web Services ('AWS'). According to their website AWS provides on-demand cloud computing platforms to companies, governments, as well as individuals (Amazon Web Services, Inc, 2018). Data privacy aspects concerning the data ownership and customer content control are specified in their terms of service. They specify five aspects of data privacy: access, storage, security, disclosure of customer content, and security assistance.

Access is defined as customers managing access to their content and user access to AWS services and resources, and to help with this, AWS provides a set of access, encryption and logging features. (Amazon Web Services, Inc, 2018.)

By storage AWS means the possibility to choose in which geographical region the customer content is stored. AWS promises not to move or copy customer content outside the selected geographical region without consent from the customer. (Amazon Web Services, Inc, 2018.)

The security aspect means the customer chooses how their content is secured. AWS states they provide encryption for customer content both in transit (when sending the data over a connection) and at rest (while the data is simply located on disk and not being operated upon), and there is an option to manage your own encryption keys. (Amazon Web Services, Inc, 2018.)

Concerning disclosure of customer content AWS states that they do not disclose customer data unless required to do so by law or to comply with a valid and obligatory order of a governmental or regulatory entity. (Amazon Web Services, Inc, 2018.)

As a security assurance AWS states – quite vaguely – that they "have developed a security assurance program that uses best practices for global privacy and data protection to help you operate securely within AWS, and to make the best use of our security control environment. These security protections and control

processes are independently validated by multiple third-party independent assessments.” (Amazon Web Services, Inc, 2018.) More specific details of these security measures are perhaps available for AWS customers.

Amazon assures in its terms of service that they do not access the data storages of their customers without permission, and that they do not use customer data, nor do they infer information from it for marketing or advertising purposes (Amazon Web Services, Inc, 2018). Amazon also states that they offer customers the possibility to choose the data storage’s geographical location, and access rights to their data based on location.

Questions regarding data control and ownership can become exceedingly complex and pronounced as organizations’ data storage solutions become more dispersed. Utilizing the services of a multitude of IaaS, SaaS, or PaaS providers, or distributed databases in different geographical locations, can make the service contracts between cloud service providers and their customers complex because data ownership, use, and control rules must comply with both organizations’ business requirements and the legislations of possibly multiple countries. Within organizations, the autonomies discussed before can be said to manifest themselves through technical choices and contractual agreements.

Many of the traditional database structures and technologies presented above are well established and widely in use. They are not without problems, however. Storing the data of an organization in a single centralized database location may place the organization in a vulnerable situation. An attacker may gain access to the database by various malicious means and obtain or otherwise manipulate the organization’s possibly sensitive data. This threat can be reduced by utilizing distributed databases so that not all data is stored in a single location, and of course by utilizing the latest database security measures. On the other hand, distributing the data on multiple sites provides potential attackers with more attack surface, so this can be a tricky issue.

### 3 Peer-to-peer networks

*Peer-to-peer networking* (“P2P networking”) is an essential set of technologies that need to be addressed when moving towards discussing blockchain technologies, which have a lot in common with P2P.

The term P2P is almost synonymous to illegal downloading due to it being the technology behind various file sharing services such as BitTorrent, Gnutella, DC++, EMule, Kazaa, and Napster. Regardless, P2P networking is not only for file sharing: other well-known services use the technology as well. For example, the voice over IP (‘VoIP’) internet telephony service Skype utilized P2P technology for years until it ran into performance issues and switched to cloud infrastructure in 2017 (Unuth, 2018). Also, the email transfer protocol SMTP uses server-to-server type of P2P networking. P2P techniques are also popular in cloud computing. In short, P2P can be used for file sharing, communication, and distributed computation, for example.

While the traditional client-server network architecture often involves a computer as a data receiver and a server as a data provider, most P2P networking solutions get rid of the designated server. Depending on the used protocol, P2P networks designate participating computers as both the client and the server, so to speak. The main idea behind P2P is to be able to establish connections and transfer files directly between computers without the need for a central entity to manage the connections. Besides file transfers, P2P can also be utilized for transferring other resources, such as control and computational power. As with any technology, there are both advantages and disadvantages to using P2P technologies compared to client-server technologies.

The main benefit of a P2P network structure is that such networks are easy to set up and maintain because each participating computer manages itself. Another benefit is that a P2P protocol network does not usually require setting up a separate, always-online server; although, in the case of for example a centralized P2P network a central register is required. To generalize, for the end user creating a P2P network only costs the price of the node computers, but of course other costs must be taken into consideration as well, such as network and electricity infrastructure, etc. Because of this lack of a central server data is stored on the participating computers. This allows for the possibility of high availability of content in the network, of course depending on multiple nodes possessing the same data and that the communication infrastructure is reliable. A P2P network can also provide good load distribution under high demand, high availability, and offers good scalability as well as good fault tolerance. (Lissounov, 2016.)

Depending on the used network type, one disadvantage to using a P2P network can be, for example, that there might be no central data storage, but instead the data are located on multiple independent nodes and therefore it might be difficult to create backups. In P2P networks security must be applied to each node separately, which might leave some parts of the network vulnerable to threats such as trojans or viruses.

There has been relatively little quantitative measurement on what percentage of IP traffic consists of P2P, but the rough number is surprisingly high. A now well over a decade old research by Azzouna and Guillemin (2004) claimed that their simple observation of a particularly loaded link of a France Telecom IP network showed that approximately 50 percent of global traffic was caused by P2P protocols. A more careful examination of IP packets that took the application level into consideration revealed the share of P2P traffic to be closer to 80 percent (Azzouna & Guillemin, 2004). Other studies regarding the proportion of P2P traffic versus other non-P2P traffic have been conducted on different levels, but the results vary noticeably (Bartlett, Heidemann, Papadopoulos & Pepin, 2007; Madhukar & Williamson, 2006). A study by Schulze and Mochalski (2009) working for the company Ipoque describes a quite vast selection of measurements that were conducted in eight geographic regions between 2008 and 2009. This study showed that P2P networks were responsible for generating the majority of internet traffic in all monitored regions, ranging from 43% in Northern Africa to 70% in Eastern Europe. However, these data are over 10 years old, and since then for example video-on-demand services such as YouTube and Netflix have become big actors on the internet, creating massive amounts of non-P2P traffic. Also, other big players such as Facebook and Google have expanded their share of total internet traffic. 73% of internet consumer traffic was video traffic in 2016 (Cisco, 2017), and in 2017 15% of all downstream traffic worldwide was created by Netflix (Cullen, 2018).

It used to be that P2P networking generated most of internet traffic in most parts of the world, but it seems that services such as video-on-demand have made video pirating a less desirable option for consumers, and therefore the portion of P2P traffic in total internet traffic has reduced.

### 3.1 Technology

A peer-to-peer network is a network of interconnected nodes (i.e. independent computers, clients) that share data between one another with no need for a centralized administrative system such as a central server (figure 2). This differs significantly from the client-server network model (figure 3), in which individual clients (independent computers) connect to centralized servers.

Schollmeier (2002) defines a client-server network as a distributed network consisting of a system of higher performance called the server, and often multiple lower performance systems called the clients. The server acts as both a central registering entity and the provider of services and content. Basically, the only task a client does is requesting content or the executing services. It does not share its own resources with others. (Schollmeier, 2002.)

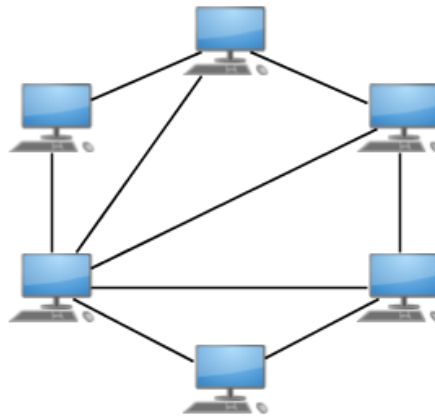


FIGURE 2 Simplified network structure based on the P2P model in which resources are shared by interconnected nodes (“peers” or “servents”) without a central entity such as a server. The computer icons depict network nodes, and the black lines depict connections between nodes.

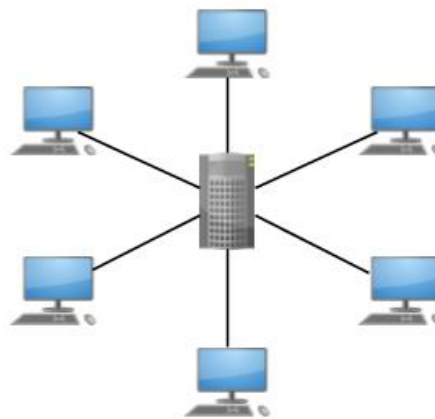


FIGURE 3 Network structure based on the client-server model, in which individual clients request resources and services from a central administrative system such as a server. Computer icons depict network nodes, server icon depicts a central administrative system, and black lines depict connections between nodes and central system.

Schollmeier (2002) suggests the term *servent* (a contrived word derived from the words *server* and *client*) for describing the capability of the nodes of a peer-to-peer network to act both as a server and a client. This is different from client-server networks, because in those networks the participating nodes can be either a server or a client. They cannot have both capabilities. (Schollmeier, 2002.) According to Schollmeier’s (2002) definition of peer-to-peer networks a distributed network architecture can be called a P2P network in the case the participating nodes share their hardware resources among other participants. These resources may be computing power, storage space, network capacity etc. Furthermore, the resources shared by the participants are fundamental in providing the service as well as the content the network offers. Examples of these services are e.g. shared

collaboration workspaces or file sharing. Other users, peers, can access the resources directly with no need to go through any intermediary entities. Therefore, the participants of a P2P network are both the resource providers and the resource requestors. (Schollmeier, 2002.)

As mentioned previously, for the end user setting up a P2P network typically only requires a computer and appropriate software. This is a key feature of P2P networks: in P2P computing, according to Kitembe and Jeberson (2017), nodes organize themselves as an overlay network, in which transmission of packets on each of the overlay links uses standard Internet protocols, which are the user datagram protocol (UDP) and transmission control protocol (TCP).

An overlay network is a network in which links between peers are based on logical relationships in a virtual network built on top of physical communication infrastructure (figure 4). The overlay is a logical depiction that does not necessarily follow the actual physical network topology. (Dunaytsev, Moltchanov, Koucheryavy, Strandberg & Flinck, 2012; Eberspächer, Schollmeier, Zöls & Kunzmann, 2004.)

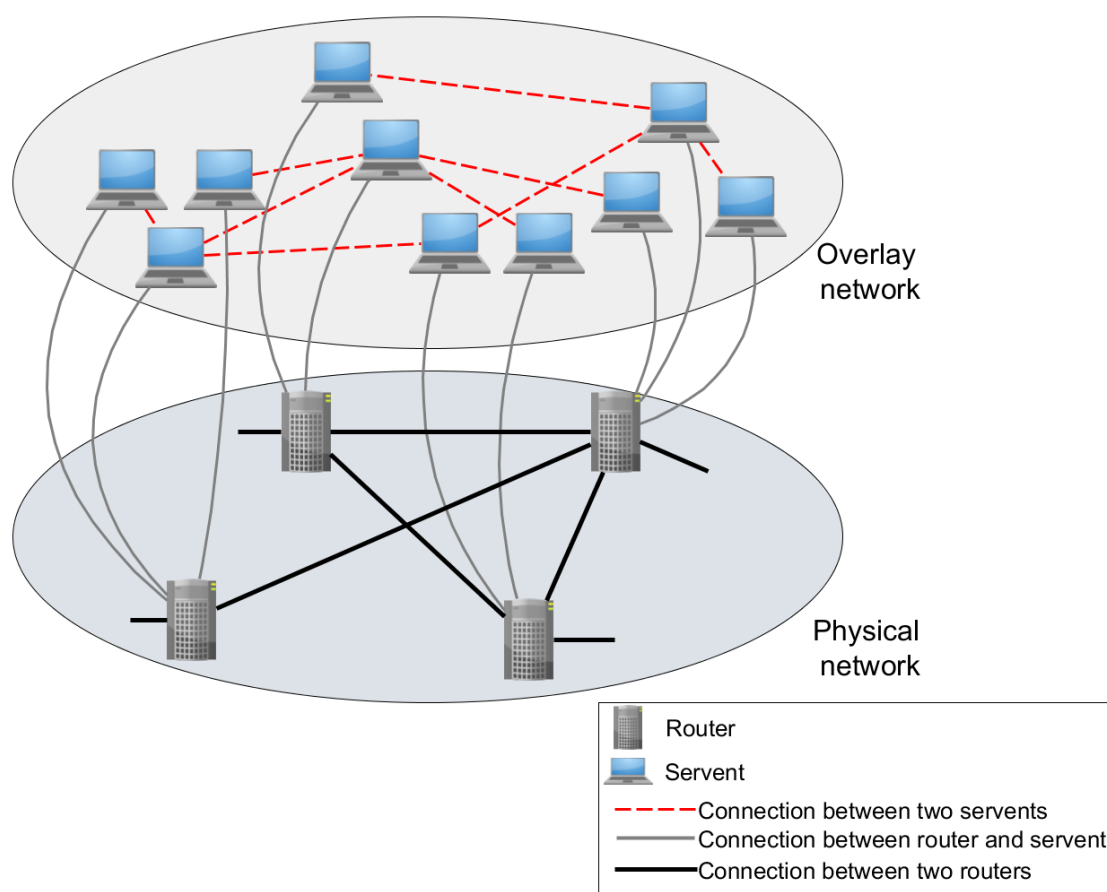


FIGURE 4 An overlay network (Dunaytsev et al., 2012; Eberspächer et al., 2004)

Zhu (2010) categorizes P2P systems into two groups: structured P2P systems and unstructured P2P systems (table 1). In structured P2P systems the connections between the network's peers are fixed, and these peers hold the information

about the content their neighbor peers possess. This way data queries can be channeled to the neighboring peers who have the desired data, even when the data is very rare in the network. To enable effective data discovery, structured P2P systems prescribe constraints on node graph (the topology of the overlay network) and data placement. The Distributed Hash Table (“DHT”) indexing is the most common means of indexing used for structured P2P systems. The DHT is based on a *key* and *value* pairing system, by which any participating peer is able to retrieve the value that is associated with a certain unique key. (Zhu, 2010.)

In unstructured P2P systems the connections between a network’s peers are formed arbitrarily in hierarchical or flat manners. In order to find as many peers with wanted content as possible the peers query data based on multiple techniques such as flooding, random walking, and expanding ring. (Zhu, 2010.)

According to Eberspächer, Schollmeier, Zöls, and Kunzmann (2004) and Zhu (2010), unstructured P2P systems can be further categorized into centralized P2P systems, hybrid unstructured P2P systems, and decentralized (or pure) unstructured P2P systems (table 1).

In centralized P2P systems, a central entity such as a server is used for indexing the entire system, which means keeping record of file locations, but not the files themselves. For example, in the music sharing service Napster the peers announced their IP address and filenames of their shared files to the indexing server, which then created a dynamic and centralized database that mapped content names into a list of IP addresses. Peers could then search and download content from each other utilizing this server-maintained list. Napster and the file sharing service BitTorrent are both examples of an unstructured centralized P2P network. The downside of this structure can be that the server is effectively a single point of failure: in a situation where the central index server crashes or is otherwise taken off network, the entire network will also collapse. (Zhu, 2010.)

A hybrid unstructured P2P network enables for the existence of so-called infrastructure nodes that can be referred to as “super-nodes” or “super-peers”. The hybrid model is unstructured, except that it divides peers into two logical layers: super-peers and ordinary peers. The super-peer concept was coined after it was realized not all peers have the same capabilities (bandwidth, processing power, disk space, etc.), and that the peers with lower capabilities could cause bottlenecks in a network’s performance (Min, Holliday & Cho, 2006).

A hybrid network is a hierarchical overlay network, addressing problems with scaling present in pure unstructured P2P networks, an example of which is the file sharing service Gnutella. Over time in this kind of a network a peer can typically change roles and, for example, become a super-peer that participates in the coordination of the P2P network structure. The super-peers are designated users (network participants) who preferably have high processing power and disk space, as well as bandwidth. When a peer enters a network, it is assigned to a super peer, to which the peer announces its shared content. (Zhu, 2010.) While a super-peer is connected to a set of ordinary peers, an ordinary peer can only be connected to one super-peer. In these hybrid P2P systems an ordinary peer is



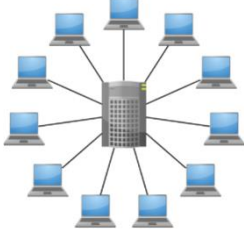
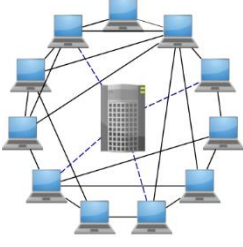
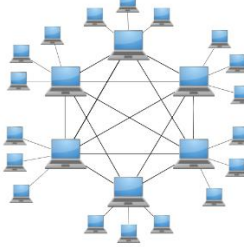
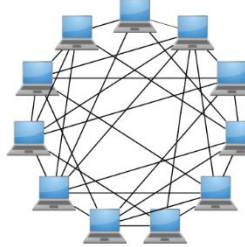
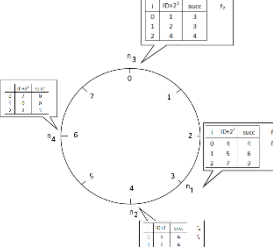
often assigned to a super-peer through random selection, which is a simple technique, but does not deal well with the participating peers' heterogeneity considering both content similarity and the peers' dynamic capabilities. In case no super-peers are online in the network at a given time, the system appoints an ordinary peer with suitable properties as a super-peer. (Min, Holliday & Cho, 2006.)

The super-peer manages search functions by maintaining a database mapping content to peers. The role of the super-peer is not unlike in the centralized design as the super-peer acts as a directory server, though the role is assigned to peers. Together these super-peers form a structured overlay network of super-peers, which makes content search efficient. (Zhu, 2010.)

A pure, decentralized unstructured P2P network is an overlay network, which is a logical network. In a pure P2P network, there is no central server managing the network, or super-peers. An example of a pure P2P network application is Gnutella version 0.4. In Gnutella 0.4, peers do not hold information about the content other peers are sharing, they are only aware of the location of their neighbor peers (IP address and port). As a result of this, search queries are conducted by a "flooding" mechanism: a peer interested in certain content broadcasts a query to its neighbors, who then forward the query to their neighbors. This continues until a holder of the desired content receives the query, who then sends a "query hit response" back to the peer who started the query, indicating that the peer has the content. Of course, the original sender might receive query hit responses from multiple peers who have the desired content, which leaves choosing the download location to him. This flooding mechanism has been criticized for its non-scalability, due to its tendency to enable linear query traffic growth along with the total query number, which grows as the system grows. Also, because there is a query time out or a depth of search limit mechanism in the Gnutella protocol, users might not find what they are looking for, especially if the desired content is rare. (Zhu, 2010.)

An earlier study by Schollmeier (2002) suggests a simpler division of P2P networks than that of Zhu's (2010): according to his paper, P2P networks can be simply divided into two sub-definitions - the hybrid and the pure P2P network structures - without first categorizing them into structured or unstructured types. In Schollmeier's (2002) division, the centralized P2P and hybrid P2P structures are essentially the same, and the concept of the super-peer was introduced by Zhu in 2010.

TABLE 1 Summary of P2P and client-server network types (Eberspächer et al, 2004). In the diagrams the computer icon depicts a network node, the server icon depicts a central administrative system, and black lines depict connections between nodes and the central system. Dashed lines in the centralized P2P diagram depict search queries to a centralized system.

Client-Server	Peer to Peer			
<p>1. Server is the central entity and sole provider of content and service -&gt; Network is managed by the server</p> <p>2. Server is the network's higher performance system</p> <p>3. Clients are the lower performance system.</p> <p>Example: The World Wide Web</p>	<p>1. Resources are shared between peers</p> <p>2. Resources can be accessed directly from other peers</p> <p>3. Peer is both the provider and requestor (servent)</p>			<p><b>Structured P2P</b></p>
	<p><b>Unstructured P2P</b></p>			<p><b>DHT based</b></p>
	<p><b>Centralized P2P</b></p>	<p><b>Hybrid P2P</b></p>	<p><b>Pure P2P</b></p>	<p><b>DHT based</b></p>
	<p>1. Includes all P2P features</p> <p>2. Requires central entity to provide service</p> <p>3. Central entity is a form of index/group database</p> <p>Example: Napster</p> 	<p>1. Includes all P2P features</p> <p>2. Possible to remove any terminal entity without losing functionality</p> <p>3. → Dynamic central entities</p> <p>Examples: Gnutella 0.6, JXTA</p> 	<p>1. Includes all P2P features</p> <p>2. Possible to remove any terminal entity without losing functionality</p> <p>3. → No central entities</p> <p>Examples: Gnutella 0.4, Freenet</p> 	<p>1. Includes all P2P features</p> <p>2. Possible to remove any terminal entity without losing functionality</p> <p>3. → No central entities</p> <p>4. Connections in the overlay are "fixed"</p> <p>Examples: Chord, CAN</p> 

### 3.2 Autonomy in P2P networks

As previously with databases, we should consider the concept of autonomies also in P2P networks by investigating how the different autonomy types (organizational, design, communication, and execution autonomy) manifest themselves in P2P networks.

When considering autonomies, the concept of organization should be defined. Therefore, we should begin by defining what constitutes an organization in a P2P network, i.e., when the P2P network can be considered to exist. In P2P organizations the organization can be considered to exist at  $1 - 1+n$  users + P2P software (one to one plus  $n$  users plus P2P software). Therefore, one way to look at the definition of a P2P organization is that the formed P2P network – the group of connected nodes – is the organization. Also, since the organization forms itself, the structure thereof is not robust or constant: depending on the network structure type users may join or leave the network (the organization) as they please.

Another interpretation of the  $1 - 1+n$  users + P2P software formula is that each user plus software is one organization, and these organizations are interconnected in the P2P network, creating an organization of organizations, so to speak. By this interpretation, there is an organizational border between nodes as well as between interconnected sets of nodes (figure 4).

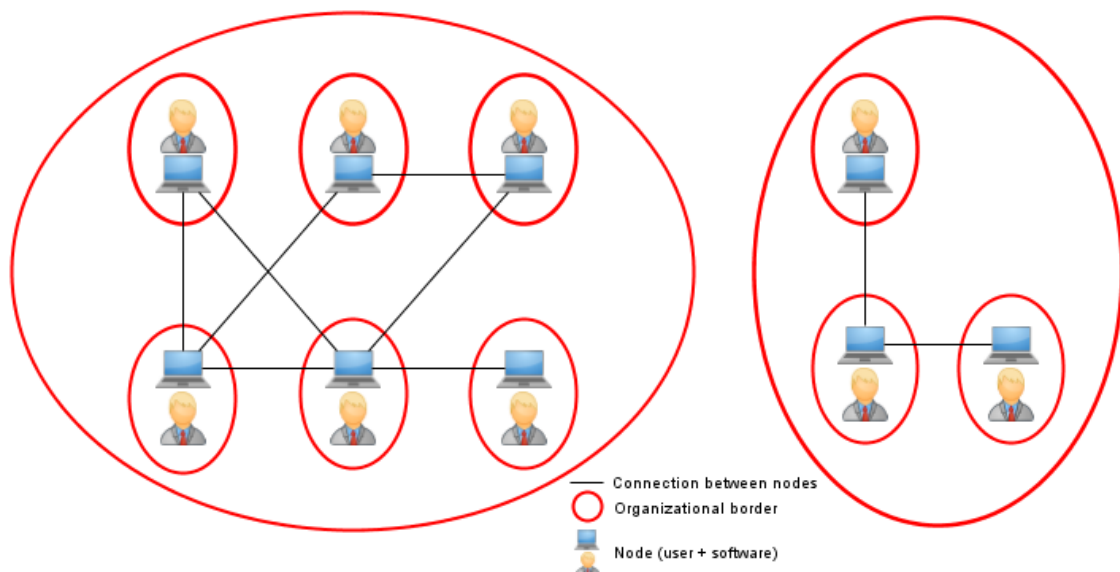


FIGURE 5 P2P Organizational borders

In the case of an unstructured centralized P2P network, the organization can be considered to exist at  $1 - 1+n$  users + P2P software + central entity, because without a central entity such as a server the centralized P2P network cannot function by definition. For example, in the case of BitTorrent, a central “tracker” server was needed for the service to function before it moved to utilize DHT for “trackerless” torrents. The tracker servers assist in the communication between peers

to find suitable peers to download desired files from. The servers were usually operated by private individuals, such as volunteers.

As stated earlier, organizational autonomy means organizations are not in control of each other and can act on their own (Veijalainen et al., 1992). How this relates to P2P applications is that while BitTorrent and Kazaa, for example, are both P2P applications, they do not affect each other's operation since they operate in their own separate networks. Unlike traditional organizations such as banks or companies, they are not in contact with each other. Users might prefer one P2P application over another, which can mean more users for that application and fewer for the other, but the organizations, as defined here, do not have direct control over each other; the networks retain autonomy. Therefore, the different P2P networks that do not communicate with each other can be said to be organizationally autonomous. On the other hand, when considering individual nodes (participating in a P2P network) as organizations, they are not entirely autonomous. Some nodes may have power over other nodes: administrators of a Direct Connect hub can prevent users from accessing the hub, and users on a Kazaa P2P network may refuse other users from downloading files from them.

Design autonomy means organizations can make their own decisions regarding the systems they intend to use (Veijalainen et al., 1992). In the case of a P2P organization, to some degree, this does apply, and is also rather restrictive. As the organization is formed, it is already locked into the technology choice of the network's founder, as the organization was previously defined to exist at 1 + 1 + n users + software, with said software being the P2P application of choice. Users wishing to join the P2P network have to use the software used by the founder: it is not possible to join a BitTorrent P2P network with a Kazaa client, for example. However, different versions of the same P2P network application client may work with each other, and there usually are no hardware restrictions imposed on the system composition of the P2P network's peers. In this sense, P2P networks can be said to possess design autonomy. Virtually anyone can join a P2P network, guaranteed that it is public, and that the node obeys the protocol defined for the P2P system in question. From a node point of view design autonomy can be summarized as the node's freedom to select from different P2P software or program the software himself.

Communication autonomy refers to the fact that organizations have the autonomy of choosing which other organizations they communicate with and when (Veijalainen et al., 1992). Let's first consider C-autonomy through the relationship between two P2P organizations utilizing distinct P2P protocols. Since most P2P applications require the user to use a specific application to access the desired P2P network because of technical decisions and protocols, the P2P applications do not communicate with each other (e.g. BitTorrent and Kazaa). The users may communicate with each other, if permitted by the properties of the application, by sending written messages, or by requesting, sending and downloading files, for example, but this has little to do with the definition of communication autonomy. The participants of a chosen P2P network can often communicate with each other inside the P2P organization, but not with users outside the organization.

Hence, it could be argued that P2P organizations utilizing different P2P protocols possess communication autonomy, the reason being there often is no choice.

We should also consider communication autonomy from the perspective of the nodes within a P2P organization and the relationships between them. Within a P2P organization the nodes can be said to possess communication autonomy, since they can decide for themselves what other nodes they communicate with and when. This applies also in the case of a centralized P2P network where there is a central entity: the nodes have autonomy over whether or not they communicate with the central entity. The central entity, however, does not have autonomy towards the nodes it is connected to: for a centralized P2P network to operate, the central entity must be online and communicate with the nodes around the clock.

Execution autonomy means an organization does not necessarily need to process all messages it receives (Veijalainen et al., 1992). P2P organizations are inherently composed of their users and distinct software (and sometimes a central entity), so depending on the protocol being utilized they may not communicate with other P2P organizations as they are defined above, so P2P organizations utilizing protocols that do not communicate with other protocols can be argued to uphold execution autonomy; they have protocol level independency. But, within organizations (e.g. node-to-node or central entity-to-node communication) execution autonomy manifests itself as how nodes or central entities react to messages sent over organizational boundaries. Can the organization ignore the incoming protocol messages and not react to them, refuse to retrieve or return requested data? If the receiving organization has autonomy over these sorts of decisions, it can be seen to possess execution autonomy. When examining a single node as an organization, execution autonomy can be defined separately for each pair of nodes within a P2P network utilizing a certain protocol.

Nevertheless, some P2P organizations, such as organizations created with applications using the Direct Connect ("DC") protocol, could be argued to consist of "sub-networks" or "sub-organizations" called *hubs*. This form of a P2P network is essentially a centralized P2P network, because the hubs do not host data; they act as indexing servers only. This "sub-organization" argument is based on the principles how Direct Connect clients work: in a DC client such as DC++ users can create hubs - chatroom-like environments - where users can see lists of other users and the content they have shared for others to download. These hubs often have their own rules for connecting to the hub, sharing content, and downloading content, etc. If a user does not abide by these rules, they can be removed from the hub and thus cannot access the content shared by other users on the hub anymore. In this sense the hubs possess organizational autonomy (hubs do not control other hubs) and design autonomy (hub administrators can choose which DC software and which version to use). In accordance with the definition of communication autonomy the hubs can choose who to communicate with, but the communication is restricted to hub-to-user interactions, since hubs do not in effect communicate with each other, except in the sense that a hub can remove a user by redirecting them to another hub without restrictions. Whether or not the

hub the user is redirected to accepts the connection of the redirected user is up to the hub's rules.

Concerning execution autonomy: the hubs do not need to process the messages (text messages to other users, search requests, etc.) the users send on the hub if the messages are against the hub's rules or not accepted by the utilized protocol, and since a user plus P2P software constitutes a P2P organization, it can be argued that E-autonomy is defined between hubs and individual user nodes, and both possess execution autonomy in this case, since individual nodes also have the choice over what to do with messages received from the hub.

On one hand P2P organizations can be said to not be autonomous at all, because such organizations have no choice to be open to other P2P organizations utilizing different P2P protocols. They are locked into their technology from beginning, from the creation of the organization. For example, a P2P organization using a DC protocol network cannot communicate with a P2P organization using the BitTorrent protocol. A P2P organization could, of course, choose to run several P2P applications (i.e. users can run several applications on their computer simultaneously), but applications based on different protocols cannot directly manipulate each other.

It should be noted that due to the inherently closed nature of P2P networks we should probably consider the meaningfulness of discussing the autonomies of P2P networks. It seems a P2P organization is inherently autonomous, because it cannot be directly influenced or communicated with by outside P2P organizations of other network types or P2P organizations utilizing different P2P applications. In different P2P network types (in their applications, in effect) the autonomies manifest themselves in different ways, depending on the definition of an organization.

## 4 Blockchain

*“It’s not shared data, it’s shared control” (Lewis, 2017.)*

### 4.1 Overview

Blockchain is still a fairly new technology for storing information in a distributed network and making secure and anonymous (or pseudonymous, to be precise) transactions without a central operator such as a bank. Blockchain technology became known in 2009 along with the rise of Bitcoin, the oldest and currently most valuable cryptocurrency created by the pseudonym Satoshi Nakamoto, a person or a group of people who worked on cryptography. The real identity of Nakamoto is still unclear. Blockchain is the technology behind today’s most known cryptocurrencies. A cryptocurrency, such as Bitcoin, is a decentralized online currency.

In this context “decentralized” means that the information stored in the blockchain is not stored (or better – replicated) only in one location, but on the devices of everyone participating in the blockchain. Blockchains use cryptography to solve trust related problems that previously have been assigned to a third party like a bank. In a banking scenario where sender A transfers currency to the bank account of recipient B inside the same bank, the bank is responsible for removing the currency from A’s account and placing it onto B’s while respecting the ACID properties as well as consistency constraints (sender and recipient bank accounts’ balance must remain constant) of the transaction. Curiously, currently in the Bitcoin network coins can be sent to “nowhere”; if a bitcoin is sent to a nonexistent address, it is simply “destroyed” and becomes permanently unavailable to anyone to use.

In blockchains cryptography is used to ensure the legitimacy of the transactions within the chain. Bitcoin and other blockchain technologies have allowed mutually mistrusting parties to make financial transactions with no reliance on a central trusted third party while also allowing for a transparent data storage (Nakamoto, 2008). Because of the aforementioned properties, blockchain technology has gained attention not only in the financial sector, but in other sectors as well: companies and organizations working in smart property, Internet of Things (IOT), cloud storage, supply chain management, ownership and royalty distribution (such as in the music industry), decentralized autonomous organizations, and healthcare have taken interest in the possibilities offered by blockchain technology (Wüst & Gervais, 2017).

According to one description blockchain can be defined as “a database that is shared among its users and allows them to transact valuable assets in a public and pseudonymous setup without the reliance on an intermediary or central authority (Glaser, 2017; Risius & Spohrer, 2017). Another description by Honkanen

(2017) does not define blockchains as databases but rather describes blockchain technology as an open, distributed, cryptographically linked ledger stored in a network of multiple devices, and the contents of the ledger can be verified by multiple users simultaneously from multiple locations. While traditional database systems may be centralized or distributed, the key difference is that blockchains do not have a centralized control mechanism; instead they work entirely in a distributed P2P network formed usually by thousands of computers called “nodes”. The data stored in a blockchain is therefore stored on every node participating in the network: while a bank may hold a ledger of all accounts and transactions in one or more locations, in blockchains the ledger is replicated on all participants’ devices, which through a consensus mechanism ensures the consistency of the ledger on all nodes. Operating in a distributed manner without a central operator (excluding the blockchain developers) is a key property that differentiates blockchain technology from traditional centralized database structures. In the latter the starting point is in controlling and storing data via a central server and the hierarchy thereof (Honkanen, 2017), which does not apply in the case of a distributed database, though, and especially in the case of so-called federated databases. They do not require a central node, and any node can start a distributed transaction. In distributed transaction processing there exists no central control, and the nodes must obey a two-phase commit protocol, which is a type of atomic commitment protocol.

Lewis (2017) introduces an interesting way to look at blockchains – or distributed ledgers – is that it is not about sharing data, but rather sharing the control of data. Before distributed ledgers, data was (and still often is) controlled by single entities. For example, you can upload a photo to Facebook and configure who can see it there, but ultimately it is Facebook who controls the data. This can be called control by power; Facebook has the power over its data and its users. With distributed ledgers, there can be pre-agreed technical rules regarding the handling of data, and participants of the distributed ledger are subject to these rules upon creating or joining a distributed ledger network. These rules are not enforced by any single entity and therefore participants could ignore them and create or upload invalid data, but there can be approval mechanisms in the network so that other participants can validate the data before it is added to the ledger. (Lewis, 2017.)

## 4.2 Blockchain technology

The term “blockchain” describes the way information is stored: a blockchain is a chain of blocks – data structures made up of digital information – that are linked to the previous block with a cryptographic *hash*; hence the term “chain”. The approved changes in the database are saved to the database collectively at predefined intervals in blocks (Mattila & Seppälä, 2015). Each of these blocks is essentially a data structure that can store a list of transactions. These transactions are created by the users of the blockchain network (Wüst & Gervais, 2017). As the



number of transactions increases, so does the length of the blockchain (Gupta, 2017). The state of the blockchain is modified by the transactions, which can be used to exchange monetary amounts, but they are not restricted to financial transactions. (Wüst & Gervais, 2017.)

Each block added to the chain has a hash, which is a unique identifier, like a digital fingerprint. The hash is basically a sequence of letters and numbers. The blocks also contain timestamped groups of recent valid transactions, and the previous block's hash. (Gupta, 2017.) The blocks are linked to the previous one by referencing previous block's hash, thus forming a chain of blocks (Hong, Wang, Cai & Leung, 2017). This also prevents any kind of tampering of block content. This way each consecutive block strengthens the previous block's verification and therefore the whole blockchain. This hashing process is what makes blockchains immutable. (Gupta, 2017.)

In addition to the hash there can be three other features saved in the block: a *nonce*, a 4-byte field, which is used to prevent malicious nodes from flooding the network; a list of transactions; and a timestamp. For storage saving purposes, the transactions are often saved in a Merkle root format in each block (Hong et al., 2017). The timestamp marks the time for each transaction and is used to show that the blocks are in a chronological order. It proves what has happened on the blockchain and when; it acts as a sort of a notary. These features are depicted in figure 6a.

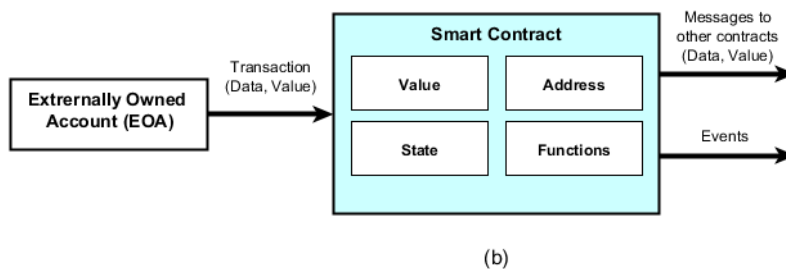
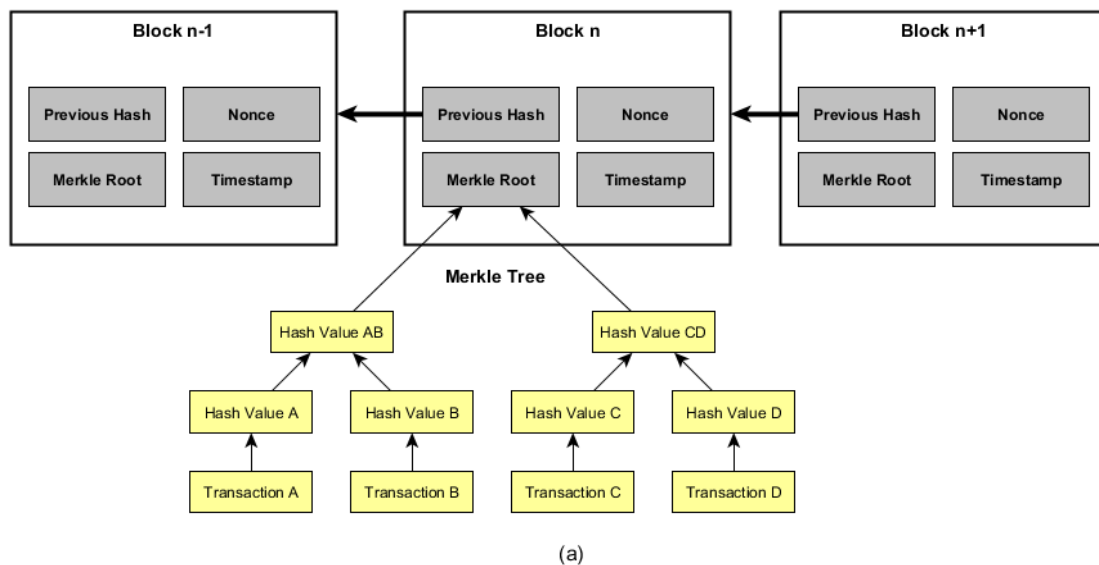


FIGURE 6 Blockchain block structure (a) and smart contract structure (b) (Hong, Wang, Cai & Leung, 2017)

Concerning publicity and transparency, not all blockchain technologies are the same. Blockchains are often divided into three categories: *public* or *permissionless*, *permissioned*, and *private [permissioned]* blockchains. On a public blockchain all transactions can be verified in the public log, but the contents of these transactions are not inherently public. Instead, the publicity of the information content of the transaction is controlled by the holder of the private key. The private key is generated through a complicated algorithm by the web wallet upon creating an account on a web wallet service such as Coinbase. Keeping this private key secure is essential for protecting your account: some users keep the private key written on paper or in physical cryptocurrency wallet devices. Another key, the public key, is in fact a representation of the address of the cryptocurrency wallet. The keys are not stored on cryptocurrency networks themselves, but are instead created by and stored in cryptocurrency wallets, which exist independently of the network.

In public blockchains it can be verified that something has happened regarding an address and possibly with what address, but not what the transaction withholds. (Honkanen, 2017.) This doesn't necessarily mean that a transparent ledger is completely insecure: a distributed ledger can still offer public verifiability of its state without giving away information about the state of individual participants (Wüst & Gervais, 2017): even if the transactions are public, the contents of the transactions may not be. In public blockchains it is therefore possible to know that something has happened regarding an address and possibly with what other public address, but not what the transaction contains. (Honkanen, 2017.)

In a permissioned blockchain the user needs a permission that allows participating in consensus creation and/or creating smart contracts inside the blockchain (Honkanen, 2017). In a permissioned blockchain all participants have unique identities, with which they can place constraints on network participation, and accessing rights to transaction details. Network participation constraining could be useful to organizations, because they could comply with required data protection regulations with less effort. The consistency of the data getting appended to the blockchain can also be controlled more efficiently by utilizing a permissioned blockchain. (Gupta, 2017.)

Private blockchains, on the other hand, are closed blockchains designed for example for the internal use of companies, where verification and consensus mechanisms typical of public blockchains are not needed because trust is an inherent component in this type of situation. Depending on context, it might prove problematic who actually decides the rules of the blockchain, and who can be allowed to participate in it, and who can be removed from the chain. According to some views it is questionable whether private blockchains can offer anything new in contrast to traditional data structures (Honkanen, 2017). Table 2 describes how and by which participants data is governed in three types of data storage solutions (Lewis, 2017).

As all technologies, blockchains have also evolved over time: there are different generation blockchain types, and the third one is currently the newest. The

first generation includes the first applications of the blockchain technology, such as the cryptocurrency Bitcoin. These cryptocurrencies are “simple”, as they can be described as simple digital currency. The second generation of blockchain applications includes projects such as Ethereum, a blockchain-based programming platform for creating distributed applications such as smart contracts, and other cryptocurrencies operating within the Ethereum platform. First and second-generation cryptocurrencies most often use the Proof of Work consensus algorithm for creating new blocks, which is the major difference between them and third generation cryptocurrencies, which aim towards the use of Proof of Stake algorithms. The use of Proof of Stake can provide electricity savings, since individual nodes do not need to compete with computing power. In addition to using less power, newer generation cryptocurrencies can provide better scalability and higher network capacity over the older generation cryptocurrencies.

TABLE 2 Data change governance (Lewis, 2017)

How are changes to data governed?			
	Traditional single-entity databases	Public blockchains	Private distributed ledgers
Who enacts changes to data?	Single entity enacts changes	Individual data-adders ('miners') propose new changes ('blocks') and submits to entire networks	Individual participants suggest changes and submit to relevant subset of network.
What constraints changes?	Changes constrained by user agreements and terms of service. No technical constraints.	Changes constrained by technical rules (e.g. maximum block size).	Changes constrained by technical rules (e.g. producing relevant digital signatures). Legal agreements between participants may also play a part, e.g. mutual terms of service.
Who needs to technically agree changes to data?	No technical policing.	Network participants police by accepting or rejecting data	Configurable: data changes can be policed network-wide, or by relevant participants only.
Dispute mechanism	Disputes can go to legal system	No built-in legal dispute framework, by design	Disputes can go to legal system.

### 4.3 Key features

One of the major concepts in the digital world which blockchain technology has great promise to substantially impact is the concept of trust. In a centralized system trust is traditionally managed by the provider or administrator of the system. Since there are no centralized operators or "middlemen" in blockchain systems, functionalities associated with trust that are not found in traditional centralized databases are required (Honkanen, 2017). In blockchain systems functionalities like this are cryptography and the consensus mechanisms which determine which changes will be applied in the chain. The consensus mechanism could be described as a council formed from the participating nodes, which decides which blocks will be added to the blockchain.

Gupta (2017) argues that there could be benefits for businesses from utilizing blockchain technology. The benefits could be savings in time and money as well as promises of tighter security. Because transactions via blockchains do not require verification from central authorities, the transaction settlement times can be fast. This is of course dependent on the utilized blockchain technology, of which there are several. Other beneficial aspects are that utilizing blockchains removes single points of failure in systems, prevents censorship, and ensures transparency and trust between all parties involved in an interaction.

In 2016 Visa claimed to process 150 million transactions per day (Vermeulen, 2016) which amounts to roughly 1700 transactions per second, while Bitcoin has

been estimated to be able to handle between 3 and 7 transactions per second. In 2013 Visa claimed they could process more than 24,000 transactions per second (White, 2013) and in 2016 Visa estimated that VisaNet could be capable of processing over 56,000 transactions per second (Vermeulen, 2016), but these claims have been questioned (Sedgwick, 2018). Mastercard claims their network can handle roughly 44,000 transactions per second (Mastercard, 2012). Regardless of the exact number of transactions per second, the traditional card payment networks are currently the fastest and most widely used payment options. Some blockchain companies claim that their technologies can handle tens or hundreds of thousands of transactions per second: for example Metahash claims their network can handle 50,000 to millions of transactions per second (Metahash, 2019), and researchers at an Australian university running a project called Red Belly Blockchain (RBBC) claim to have reached a speed of more than 100,000 transactions per second (Red Belly Blockchain, 2018). Considering that these transaction speeds have been reached in test networks much smaller than the globally encompassing networks of the large payment providers (the RBBC test network consisted of 100 computers), the transaction speeds would most likely be significantly lower in large-scale real-world use. So far one of the biggest challenges for blockchains has been maintaining high transaction speeds as the network size grows.

In suitable contexts, cost savings from utilizing blockchains are possible by needing less oversight because the network is self-policing by the network's participants. Also, the number of intermediaries is decreased as participants can trade items of value directly between each other, and duplication of effort is removed as all participants can access the shared ledger. (Gupta, 2017.) The main cost, though, is keeping the ever-growing ledger copies consistent. Underwood (2016) claims blockchain technology has the properties to revolutionize industry and commerce and drive global economic change. These expectations have risen because a blockchain can be immutable, transparent, trust-redefining, while enabling fast, secure, transparent and trustworthy transactions that can be either private or public. Underwood (2016) claims blockchain could even provide solutions to the challenges of people in developing countries, such as asset ownership, financial inclusion, and identity recognition. Also, according to Underwood (2016), blockchains could help avert a repetition of the 2008 financial crisis, improve supply chains, support efficient healthcare programs, and maybe even reduce corruption and other unethical behavior in high-value markets like diamond trading. (Underwood, 2016). While Underwood's (2016) article proposes a plethora of theoretical benefits of using blockchain, it should be taken with a grain of salt; blockchain can be a useful technology in the right setting, but it is no silver bullet to all the world's problems, especially since the technology is still new and there are several unsolved issues with it. Nevertheless, an argument can be made that utilizing blockchain technology can potentially offer benefits ranging from technical benefits to economic and strategic.

As mentioned above, a central concept to blockchains is that of trust. Blockchains have been said to remove the need for trust between parties by replacing

it with cryptography (Beck, Czepluch, Lollike & Malone, 2016). The parties in this definition are the end users of the blockchain network, excluding the developers. There still exists a strong trust relationship between the parties using the blockchain system and the developer community: the developers are supposed to design a system that meets the requirements of non-trusting parties interacting. Additionally, the parties using the system must trust that the system is implemented correctly. These two aspects can often be assumed to hold trivially, but in reality, this is not clear: there have been cases where cryptocurrency wallet implementations have allowed simple keys to be used and money has been stolen from such wallets.

There are P2P marketplaces where trust between parties is required. For example, the apartment renting service Airbnb involves three parties in its transactions: the providers of the apartments, the consumers who wish to rent an apartment, and the online platform acting as a two-sided marketplace that matches supply (apartments) with demand (renters). A traditional currency transaction process requires trust from all parties involved. People have to trust each other, they have to trust banks, and banks have to trust people and other banks. For example, when making a monetary transaction using traditional banking services, person A has to trust bank C to take the exact defined amount of currency from A's account and add that amount to the account of person B. Person B also has to trust bank C to do this correctly. In addition to trusting the third party, the bank, both parties A and B have to trust each other: if for example B sells a product to A online, A must trust that B will deliver the bought product, and B must trust A to carry out the monetary transfer. This is one of the things many blockchain and smart contract projects aim to simplify; to replace the need of trust among parties through algorithms and cryptography. Nevertheless, despite minimizing trust requirements between parties, the parties must have trust towards the algorithms, selected cryptography and the blockchain developers. It could be said that as long as there are human parties involved in any way, requirements for trust cannot be eliminated completely.

Blockchains are inherently decentralized, but according to Vitalik Buterin (2017), the developer of Ethereum - a decentralized platform that runs smart contracts - the definition of decentralization has remained unclarified for very long. Buterin (2017) therefore gives his definition for three types of decentralization:

- *Architectural (de)centralization*: how many physical computers a system consists of, and how many of those computers it can endure failing at any point in time? (Buterin, 2017.)
- *Political (de)centralization*: how many organizations or individuals are ultimately in control of the computers that the system consists of? (Buterin, 2017.)
- *Logical (de)centralization*: if the system is divided in half, including both users and providers, will the halves continue to operate fully as independent units? (Buterin, 2017.) This definition reaches also into the realm of autonomies, more specifically organizational autonomy, and the definition thereof in terms of blockchains.

Blockchains can be examined through the aforementioned type definitions. According to Buterin (2017), blockchains can be said to be politically decentralized because no one ultimately controls them, and architecturally decentralized because there is no infrastructural central point of failure. But, blockchains are logically centralized because of the fact that there is one commonly agreed state in the chain at a time (due to the operation of the consensus mechanism) and that the system behaves like a single computer. (Buterin, 2017.) The agreed upon state of the chain is not a given, though: it is a result of a race of nodes trying to solve a mathematical puzzle as fast as possible to decide who can add the next block containing a set of transactions to the chain. So a blockchain ledger can have diverse attempts to continue it.

With a blockchain, parties unknown to each other can collectively produce and maintain many kinds of databases in a decentralized fashion. The database or a part of it is copied to everyone participating in the network. All participants can make changes in the database, which will be processed and approved based on predetermined rules through a consensus mechanism described earlier. (Mattila & Seppälä, 2015.) While Mattila and Seppälä (2015) use the term “database” here and while a blockchain can contain transaction data, it is important to clarify that it is not a replacement for databases, transaction processing, messaging technology, or business processes. Blockchains can hold verified proof of completed transactions. Gupta (2017) argues that a blockchain basically serves as a database for transaction recording, but the term “ledger” would probably be more suitable. He also claims that the benefits of blockchains extend beyond those of traditional databases, which is likely to be true in many cases, though as Wüst & Gervais (2017) argue, there is a time and a place for the utilization of blockchain technology, as we will discuss later.

There are attributes often linked to blockchain technology that make it stand out from other database solutions. Blockchains have been described as

- tamper proof: the history of records cannot be modified afterwards
- decentralized: data is located in multiple locations and is backed up automatically
- encouraging collaboration: trust between participants is programmed
- disintermediated: enables confidential operations between participants without a third party. (Kinnunen, Leviäkangas, Kostiainen, Nykänen, Rouhiainen & Finlow-Bates, 2017)

Honkanen (2017) argues that the cryptographic protection of a blockchain cannot practically be broken with current technology, and that an essential property of blockchains is also that after reaching consensus and encryption the information stored in a blockchain cannot be modified. Therefore, it is possible to store information in a blockchain in a permanent manner so that the information cannot be accessed by anyone except the user, or a person or party authorized by the user. Even the person (the user) who stored the information cannot remove or modify the information once it has been stored; this way blockchains are immune to forgery. (Honkanen, 2017.) Although, since every network participant has a copy of

the ledger stored on their computer, they can try to modify the ledger as they wish and attempt to pass that along as the new ledger. Of course, this would require multiple nodes doing the same thing with the same ledger in order for it to become the chain accepted by the majority. This is called a “51% attack” in which a group of miners (in the case of Bitcoin, for example) controlling more than 50 percent of the network’s computing power can prevent other miners from creating the blocks and thereby monopolize the mining of new blocks and earn all the rewards of mining, or possibly add erroneous data into the blocks. The reward for creating new blocks is often a certain amount of the cryptocurrency in question.

#### 4.4 Four key concepts

There are various listings of the key concepts of blockchains, each containing more or less the same concepts. The four blockchain concepts presented here are *shared ledger*, *permissions*, *consensus*, and *smart contracts*, and they are based on the book by Gupta (2017).

A blockchain is basically a shared ledger for keeping records. What’s different to a traditional bookkeeping ledger is that a blockchain is an immutable record of every transaction that occurred in the blockchain network, and that it can be accessed by all network participants. With a shared ledger like this, transactions are only recorded once, which eliminates the trouble with duplication which is typical to traditional business networks. The ledger can have the following characteristics (Gupta, 2017):

- The ledger records all transactions in the network: the ledger is the system of record, and therefore holds the single truth of transactions. (Gupta, 2017.)
- The ledger is shared among all network participants, so that each participant has an exact duplicate copy of the ledger through replication. (Gupta, 2017.)
- The ledger can be permissioned, which means network participants may only view transactions they are authorized to. The participants have identities linking them to the transactions (e.g. Bitcoin wallet address, in the case of Bitcoin), but they might be able to choose what information other participants can see. (Gupta, 2017.)

The transactions saved into the blocks are saved in the main ledger called the log. The log is shared by all participants of the chain (they all possess a copy of it on their device), which means the new added data is replicated on all participating nodes when they are online and connected to the chain. The transactions are timestamped, which can be used to identify the point in time when for example the ownership of a product has been transferred to another party. With the timestamp it is afterwards possible to verify whether and when the transaction has actually happened. Since nodes’ system time can vary between each other,



an independent timestamping solution has been implemented. In the case of Bitcoin, for example, the timestamp in each block is a Unix time timestamp (Bitcoin wiki, 2019). Unix time is a system for describing a point in time, and it is the number of seconds that have passed since 00:00:00 Thursday, 1<sup>st</sup> of January 1970 UTC minus leap seconds. According to the Bitcoin wiki (2019), in Bitcoin, a timestamp is accepted as valid if the time value is greater than the median timestamp value of the previous 11 blocks in the Bitcoin blockchain and less than the network-adjusted time plus two hours. Network-adjusted time is the median of the timestamps returned by all the nodes connected to the user. This results in block time stamps not being entirely accurate, but they do not need to be; accuracy within 1-2 hours is seen as enough. (Bitcoin wiki, 2019.)

An example of transaction anonymity and confidentiality would be that when person A transfers assets via blockchain to person B, both A and B can view the transaction details. Person C is able to see that A and B have performed a transaction but cannot see the details of the transaction. (Gupta, 2017.)

The time and sequence of the transactions are recorded into the blocks, which can be added to the blockchain after being verified through consensus (i.e. agreement), which can be reached through, for example:

- *Proof of work*: The network protocol makes every node storing a copy of the ledger solve a complex puzzle (this is called “mining”), and the node that solves this puzzle gets to add their block to the chain, if the majority of other nodes agree that the block is valid. This is the current method of adding blocks to the Bitcoin ledger, but it has been criticized for consuming significant amounts of computing power and therefore electricity. (Gupta, 2017.) The electricity consumption of the Bitcoin network is roughly equivalent to that of Ireland and is estimated to grow (de Vries, 2018).
- *Proof of stake*: In order to validate transactions, the validators have to possess a certain share of the network’s total value. This could offer higher protection against malicious attacks on the network by making attacks less desirable and very expensive. (Gupta, 2017.)
- *Multi-signature*: The majority of validators (e.g. two out of three) have to agree that a transaction is in fact valid. (Gupta, 2017.)
- *Practical Byzantine Fault Tolerance (PBFT)*: This is an algorithm for settling disputes between computing nodes in a situation where one node in a set of nodes generates discrepant output from other nodes in the set. (Gupta, 2017.)

The term “smart contract” was coined by Nick Szabo, a cryptographer, in 1994. A smart contract can be an agreement or a set of rules that define how a business transaction is executed. It is basically code that executes automatically as part of a transaction when certain terms are met. (Gupta, 2017.) The smart contract can be stored either on the blockchain or outside of it. Because of the property of immutability, storing program code on the blockchain can cause difficulties, if the code is to be modified. This is especially true if the code is not correctly programmed when it is saved into the chain, so in a sense, in order to establish a

trust-free system, there needs to be trust towards the programmer and ultimately the code. There is also the chance that the code will become obsolete in some time frame and that it will not work as intended anymore.

According to Gupta (2017) smart contracts can have many contractual clauses that can be made to be fully or just partially self-executing or self-enforcing, or both. The aim of the clauses is to provide security that is superior to traditional contract law, while decreasing delays and costs that traditional contracts may entail. For example, a smart contract could hold the terms and conditions of a person's travel insurance, which would execute automatically in the case the person's flight is delayed by more than five hours. (Gupta, 2017.) Another example of using smart contracts is that when the funding goal of a Kickstarter crowd-funded project is met, the pledgers' money would be transferred to the Kickstarter project's account, or a bank loan is granted when certain loan conditions are met. Along with other information on the blockchain, also smart contracts are immutable and distributed, which means they can't be changed after creation, and that the output of the contract is validated by everyone on the blockchain network.

A smart contract designed for online payments and product mailing could work in the following fashion: person B (product seller) creates a smart contract with payment details (e.g. price) and sends a link to it to person A (product buyer). Person A transfers agreed sum to the smart contract. Neither A or B can now access the money; it is locked in the contract until the conditions are fulfilled. B sends the product to person A via mail and registers the product's tracking code to the smart contract as proof that the package has been sent. Person A receives the tracking code, and the smart contract releases the money to person B's account. The details (the money transfers and tracking code exchange) have been recorded in the blockchain and are indisputable. Blockchains have been said to have the potential to take us to a "cryptographically secured trust-free transactions economy" (Beck et al., 2016).

An interesting application for blockchains worth mentioning is the decentralized autonomous organization (DAO). A DAO is basically a business or an organization in which decisions are made automatically based on programmed code or based on the members' votes. It can operate like a company, but it has no centralized management or hierarchy. It is a system of hard coded rules - programs, smart contracts, and so on - that define which actions an organization will take, often based on majority vote in the blockchain network.

Theoretically, a DAO could exist and run autonomously on the Internet if the platform provides adequate rules and flexibility. However, the system would rely heavily on hiring individuals to perform the actions the system cannot do itself. Also, a blockchain network could contain multiple DAOs, that could communicate with each other. The DAO has been predicted to be one of the business models of the future.

## 4.5 Autonomy in blockchain systems

Since blockchains networks are essentially types of P2P networks, they share many commonalities regarding autonomies. We can, for example, identify the different organizational entities in a blockchain network: the users (nodes), and the blockchain developers, who could be a company, a group of volunteers, or any other interconnected group with shared goals. Bitcoin, for example, is being developed and maintained by volunteers, while some other cryptocurrencies and blockchains designed for other purposes besides monetary transactions may be developed and maintained by for-profit companies, as an example.

In accordance to definitions previously presented in chapter three, there is an organizational border between nodes in a blockchain network; in other words, a node participating in a blockchain network is an organization in itself. Furthermore, nodes “work for the common good”: they participate in creating and securing the blockchain – and thereby form an organization as well, the blockchain, so in this sense a blockchain network is a loose organization of organizations.

Blockchain networks can be considered as organizationally autonomous in the sense that for example the Bitcoin network does not communicate with the Ethereum network, so on this organizational level they are autonomous. Also, nodes may choose for themselves which blockchain network they wish to participate in, giving them autonomy in the choice of used network.

Regarding design autonomy, nodes may choose for themselves which blockchain network they wish to participate in, and on which hardware, as long as the hardware supports the software required to participate in the blockchain network. As with other distributed P2P systems, the most important requirement is that the node obeys the protocol defined for the blockchain network in question. The companies or communities who develop the blockchains also retain autonomy in the choice of technology: they may choose what technology they use and how regardless of other companies’ or communities’ choices.

Having communication autonomy means organizations decide for themselves how, when, and who they communicate with. In a blockchain network, the communication can be for example two nodes making transactions (which it mainly is, e.g. in the case of Bitcoin). Anyone on the blockchain network can make transactions with anyone, and the transactions are public in the sense that the contents (e.g. amount of currency transferred) of the transaction can be verified by anyone on the network, but the sender and recipient remain relatively anonymous; in the case of Bitcoin, they are only identified by their Bitcoin wallet address. Due to the decentralized nature and lack of a central entity, blockchain network nodes can decide for themselves who they make transactions with. But that is where the autonomy can be considered to end; the nodes are usually limited to communication among the users within the selected blockchain protocol. Also, if receiving transactions is considered as communication, then nodes do not have autonomy in deciding whether or not they wish to receive, for example,

cryptocurrency through a network if someone was to transfer them some. Furthermore, if while a node is offline for a long enough period, the longest, most widely adopted blockchain changes (new blocks are added), when online again, the node is required to download the most recent version of the ledger if he wishes to participate in the most current chain. Otherwise the node can start creating its own side chain (a “fork” of the most widely adopted chain) or join another chain that has the same ledger content as his. Considering blockchain networks as organizations, they possess communication autonomy in the sense that they are not connected to each other, such as Bitcoin and Ethereum networks.

Execution autonomy is defined as the organization’s freedom to react to messages it receives the way it wants. For a blockchain network’s nodes this does apply, because they may have the choice to accept or reject the blocks to be added to the chain. Whatever the majority of nodes decide as valid transactions will be committed to the longest valid chain. If a node decides not to accept a new block, this can create a new fork of the most widely accepted chain. Considering blockchain networks as organizations, the situation is similar to that of communication autonomy: the blockchain networks do not communicate with each other, therefore there are no messages to be executed.

## 5 Design Science Research Process

The research part of this thesis will be conducted using the design science research process (DSRP), which is a process model for conducting design science research. It also provides a mental model for presenting DS research in information systems. (Peffer et al., 2006.) The DSRP model consists of six steps, which are (in nominal sequence) problem identification and motivation, defining the objectives for a solution, design and development, demonstration, evaluation, and communication (Peffer et al., 2006). The model is presented graphically in figure 7.

DSRP provides a model for identifying a problem and designing a solution for the problem, as well as testing and evaluating the solution. The DSRP activities in detail are as follows:

1. *Problem identification and motivation.* In the first step the research problem is defined, and the solution's value should be justified. Because the artifactual solution is developed based on the problem definition, breaking the problem into smaller pieces conceptually may be useful so that the solution matches the complexity of the problem. It is also useful to provide justification for the value of the solution, as it manages to do two things: it can motivate the conductor of the research as well as the audience of the research into pursuing the problem's solution and accepting the results, and it also helps shed light onto the researcher's reasoning in understanding the problem. The resources that the completion of this first stage are needed include insight regarding the problem's state and the relevance of the problem's solution. (Peffer et al., 2006.)
2. *Objectives of a solution.* The objectives of a possible solution are inferred from the definition of the problem. The objectives may be qualitative (how a new artifact could support solutions to problems that have not been addressed so far) or quantitative (how a new solution would be better than the current solutions). These objectives should be deduced from the specification of the problem. The resources this stage requires include insight regarding the state of the problems and the current solutions and their potential effectiveness. (Peffer et al., 2006.)
3. *Design and development.* Creating the actual artifactual solution. The artifacts may be, for example, constructs, instantiations, models or methods (Hevner et al., 2004). In this step the goal is to determine the desired functionality and architecture of the artifact, as well as actually create it. To advance from the objectives stage to design and development stage, the resources required include insight into theory that can effectively be utilized as a solution. (Peffer et al., 2006.)
4. *Demonstration.* Demonstrating the artifact's efficiency to solve the problem. This step may include the use of the artifact in a case study, experimentation, proof, simulation, or other applicable activity. The resources this

- stage needs include efficient insight on how to utilize the created artifact in answering to the problem. (Peffer et al., 2006.)
5. *Evaluation.* Measuring and observing how suitable the artifact is in providing an answer to the problem. In this stage the solution's objectives should be compared with confirmed observed results that have come up in the demonstration step from the use of the artifact. This calls for insight on suitable analysis techniques and metrics. Depending on the problem venue's nature and the artifact, in the evaluation step it can be justified to compare the artifact's functionality with the solution objectives from stage 2, perform objective quantitative measures (e.g. budgets or produced items), do satisfaction surveys, collect client feedback, or do simulations. After this step iterating back to stage three is possible in an effort to improve the artifact's effectiveness or to continue to the next step and leave further improvements to following projects. (Peffer et al., 2006.)
  6. *Communication.* The concluding stage involves communicating the essence of the problem as well as its importance, the artifact itself, the artifact's innovativeness and utility, its design, and its effectivity to researchers and other applicable audiences. In research papers of scholarly nature researchers can utilize the process' structure to structure their paper in the same manner as a they would use the structure of an empirical researching practice (defining the problem, conducting literature review, developing hypothesis, collecting data, conducting analysis, describing results, discussion, and presenting a conclusion). Resource required by this communication phase is insight into the disciplinary culture. (Peffer et al., 2006.)

Nominal process sequence

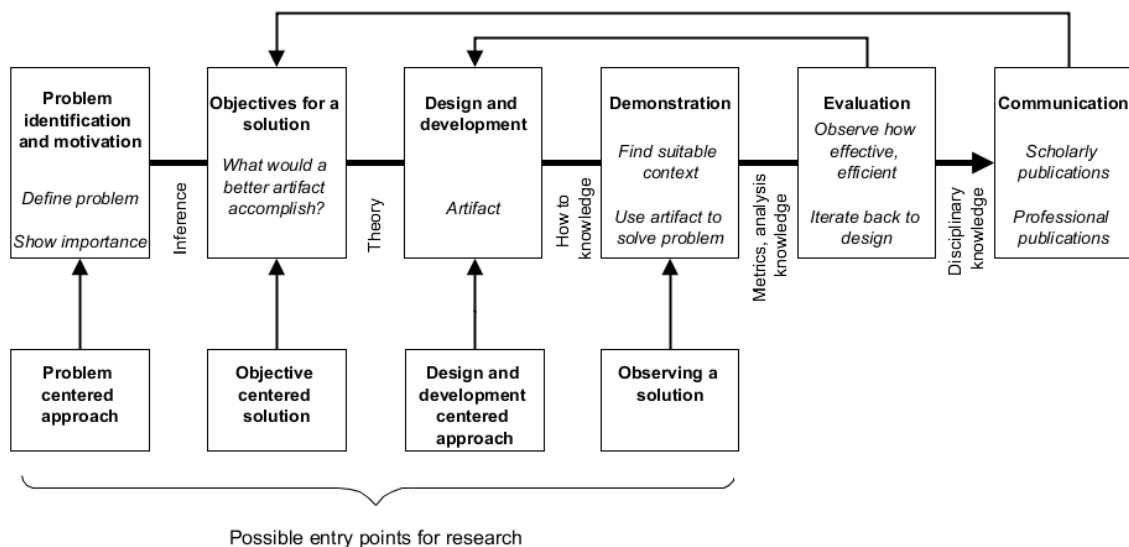


FIGURE 7 DSRM sequence (Peffer et al., 2006)

As mentioned, the steps are in nominal order, so it is not mandatory to proceed subsequently from step one through step six. It is, in fact, possible to start at almost any step and continue outwards. In effect, there are four possible entry

points for researchers in the design science research process (Peppers et al., 2006). The entry points are *problem centered approach*, *objective centered solution*, *design and development centered approach*, and *observing a solution*. Researchers may start with the problem centered approach and proceed forwards from there if the idea for their research results either from observing the problem or from proposed future research described in a prior project paper. This approach would start from the first activity, problem identification and motivation. An objective centered solution could stem from, for example, consulting experiences where the results of a system development project were not as hoped, and clients wish some aspect of the project could have been done differently. This solution approach would start with activity 2. A design and development centered approach, starting with activity 3, would be the result of having an artifact that exists already but so far has not been explicitly considered as a solution to the problem scope in question. An artifact like this could emerge from a different research domain, it is possible it has been utilized in solving some other problem in the past, or it could have surfaced as an analogical idea. Finally, starting with activity 4, observing a practical, working solution, could result in a design science solution in the case that researchers work backwards to retroactively apply accuracy or precision to the process. (Peppers et al., 2006.)

Adapting the DSRP model description of Peppers et al. (2006), an appropriate entry point for this research could be the third step of the DSRP model, design and development. This is because we already have an existing artifact, the blockchain, which we will be considering as a solution to the problem at hand. The fourth step, demonstration, will be skipped due to the fact that actually creating a road tolling system utilizing blockchains is beyond the scope of this work. The aim is to define a rough outline of a road toll architecture that utilizes blockchain technology. At the base of the research is a description of a road tolling system architecture used in Norway called AutoPASS, and the goal is to look at ways how blockchain technology could be implemented in such an architecture, or rather, what aspects of the architecture should be reconsidered from a more business oriented viewpoint for blockchain to be implemented. The implementation is done on a conceptual level, so no actual road toll system utilizing blockchains is going to be constructed. The artifact produced is a guideline or a recommendation on the type of blockchain to consider for use in a road tolling system such as AutoPASS.

## 6 Road toll systems

In this chapter a road toll system is examined from both the business logic point of view and technical point of view without going extensively into technical details, to the extent that information is available.

A road toll is a fee collected for using a public or a private road. It is essentially a type of road pricing, and the collected money is typically allocated into constructing and maintaining roads. The amount collected often varies by vehicle type, weight, or number of axles, with freight trucks commonly charged more than other types of vehicles. Road tolls are often collected on highways, or on certain bridges or tunnels or for the access to a city etc. A road toll system is a system designed for this tolling purpose.

The charging methods for road tolls can vary, but common ones are *time based charges and access fees*, *motorway and other infrastructure tolls*, and *distance or area charging*. The first method means the infrastructure user must pay a price that grants him the right to utilize the infrastructure for a given period of time. The access fee is essentially the same; in it the user pays for accessing a restricted area for a period of time. The second method is often utilized in collecting payments for usage of well-defined special and often expensive infrastructure, such as tunnels, bridges, parts of a motorway or some other infrastructure. The charge collection is often handled with tolling stations which may be open or closed, with some variations.

There are different types of toll collecting methods. For example, the road toll systems in the USA can be generally categorized into three types: open, closed and open road systems. Open toll systems have tolling locations along the highway, and all vehicles must stop at these locations to pay the toll. This can cause congestion on the highways, plus it can be expensive to build several tolling locations along a long highway. The difference between an open and a closed system is that there are no toll collection booths on the entrances or exits to the highway, only “mainline” toll booths. Since there is no toll collection on the entrances or exits, it can be possible to avoid paying the toll altogether, if no tolling station exists on the used section of the highway.

Another toll type is the closed road system. This system consists of toll booths at entrances and exits of the highway. In this type of system, the driver may collect a ticket upon entering the highway and pays the toll on exit. The toll is calculated based on the distance between the entry and exit tolls. In many European countries such as France, Italy, and Spain, almost all road toll systems are closed systems (Waersted, 2005).

The third type, open road system, does not utilize toll booths, but instead uses automated toll collecting. This way vehicles do not need to stop at the highway, reducing congestion. These open road systems often utilize electronic toll collection based on license plate recognition or toll-to-car transponder communication. License plate recognition utilizes cameras and image recognition software to capture license plate information from vehicles. Transponders often have an



ID number or some other measure of recognition that is used to verify the validity of a toll use subscription.

In this study we will be focusing on the automated electronic tolling systems that do not require drivers to stop at any tolling stations. In this chapter we look at a road tolling system utilized in Norway based on available literature. I managed to find quite extensive public specification documents regarding the Norwegian AutoPASS tolling system, which is part of the reason why I chose to focus on the Norwegian system.

## 6.1 Road toll systems in Norway

According to Wærsted (2005) all road toll systems in Norway are open road toll systems where drivers pay a fixed fee when passing a toll station, regardless of how long a distance they have travelled on the toll road. There are no start or end toll stations on a given stretch of road. The tolls are electronic and automatic, so when a vehicle passes a tolling station (which often resemble traffic light fixtures) the vehicle driver's or the owner's account is billed a certain amount.

According to the AutoPASS domain statement documentation (AutoPASS, 2019a) the organizational structure of the toll roads in Norway is such that there is a network called AutoPASS Samvirke, which is for electronic toll payments on public roads as well as public ferry services. This network is to ensure interoperability between parties in the AutoPASS Samvirke. The Norwegian Public Roads Administration (NPRA) is responsible for directing and managing the network. The AutoPASS Samvirke network is modeled in figure 8. It is composed of the NPRA managing the network, the AutoPASS service providers, the toll chargers, and the users. The toll road companies are required to participate in the AutoPASS Samvirke as toll chargers. In this context, a road toll company is a company that has made an agreement with the Norwegian Ministry of Transport and Communication or the NPRA to have the right to collect tolls on public roads. (AutoPASS, 2019a.) An AutoPASS service provider is a company or some other legal entity that concludes agreements with toll chargers as well as users once it has been approved by the NPRA. (AutoPASS, 2018.)

As mentioned earlier, all the Norwegian toll roads that are in the AutoPASS Samvirke are open road systems, or so-called free flow systems, using DSRC charging points that do not require vehicles to stop for toll charging. According to AutoPASS (2019a) there are two main methods for road toll collection:

- A user signs an agreement with an AutoPASS service provider. The user and the actual owner of the vehicle do not have to be the same person. The AutoPASS service provider then bills the user on behalf of the toll charger company. The user's payment to his AutoPASS service provider fulfills the user's payment obligations towards the toll charger company in question. (AutoPASS, 2019a.)

- If there is no user agreement registered on the vehicle, the road toll company collects the road tolls directly from the vehicle's owner. (AutoPASS, 2019a.)

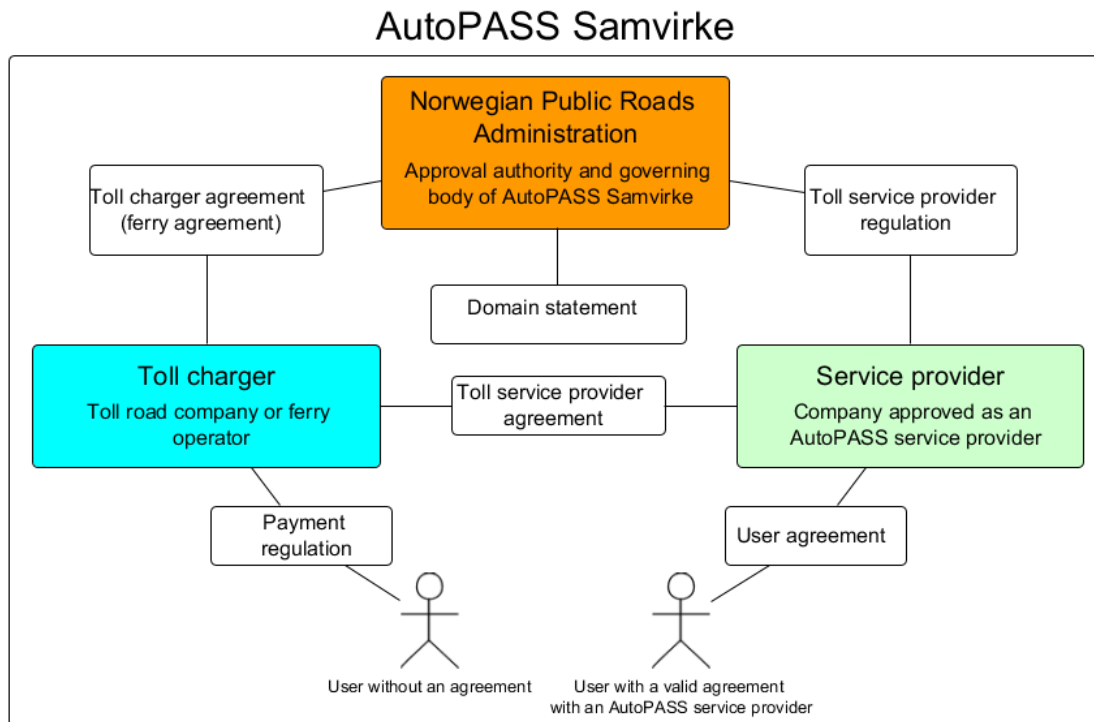


FIGURE 8 A simplified model of the parties in the AutoPASS Samvirke network and their legal and contractual relations. (AutoPASS, 2019)

For each toll road project there is a dedicated toll company. This company has limited responsibility, and its duties are to operate the toll system and to handle the collected money. Regulations for operating come from the Ministry of Transport and Communications. (Wærsted, 2005.) Currently there are five regional toll companies that handle the day-to-day operation of the tolling systems, and they are regulated by the NPRA, which is the executive authority on behalf of the government. According to AutoPASS (2019b), in 2017 there were 63 road toll projects, most of which are now part of the five regional toll road companies, so apparently some form of company reorganization has been carried out.

A paper by Wærsted (2005) describes how AutoPASS uses Electronic Fee Collection (EFC) to enable “fully automatic free flow toll stations”. Most toll collection systems in Norway support the usage of an On Board Unit (OBU) where the user is assigned to a central account. The road toll charging systems in Norway that belong to the AutoPASS system support the use of an AutoPASS OBU. This AutoPASS OBU is a Dedicated Short-Range Communication (DSRC) unit that communicates wirelessly with tolling stations in Norway in congruence to the AutoPASS radio link specifications. (Lysfjord, 2013.) The device itself is roughly the size of a small cellphone.

All tolling companies using EFC are part of the AutoPASS project and interoperability between companies is defined contractually, which means convenience for consumers. Drivers who have the OBU installed do not have to sign a subscription contract with every toll company to be allowed to legally drive in the no stop lanes around Norway. Upon signing the contract to become a subscriber with their “home” toll company, they can also subscribe to the “national payment system” which gives them legal access to all AutoPASS lanes in Norway. (Wærsted, 2005.)

Since the NPRA has the power to harmonize Norwegian road toll collection systems, its role is vital in setting up codes of road tolling conduct, building toll stations, and implementing toll collection equipment. NPRA owns the AutoPASS system, i.e. the technical specification, the tags, and all the toll road equipment. Because of NPRA’s major role in this, they have also been taking part in a project on road tolling operations together with Finland, Denmark, and Sweden. (Wærsted, 2005.) Currently there is a partnership called EasyGo that enables the use of a single electronic toll tag on toll roads, ferries, and bridges in all EasyGo member countries Norway, Sweden, Denmark, and Austria.

In the AutoPASS system automated toll stations (Charging Points, “CP”) are equipped with video cameras that record the vehicle’s license plate and a device that reads the AutoPASS OBU installed on the windscreen of the car. Every time a vehicle passes an AutoPASS road toll station, a set of information is saved locally onto the toll station system, including OBU information, transaction data, and license plate images. The images are stored locally only temporarily. Transaction data, the picture file, and exception messages are uploaded to the central system based on a schedule or upon request by the central system (Pedersli, 2012). Whether the connections between the Charging Points and the central system are wireless or fixed fiber was not clearly defined in the AutoPASS documentation.

When a vehicle approaches the tag reader -equipped gantry, the antenna on the gantry reads the tag’s unique ID number from the OBU. On the gantry there is also detector that registers the vehicle and sends a quick message to the video camera to start taking several pictures of the vehicle’s front license plate (figure 9). To account for night conditions there is a light source halfway between the vehicle detector and the camera which illuminates the license plate. On the highways’ tolling stations there are tolls which require vehicles to stop as well as AutoPASS no-stop lanes. Passing through the AutoPASS payment lane the station computer checks if the vehicle’s tag number correlates to a valid subscription. The state of the subscription is communicated to the driver immediately at the end of the payment line as a visual signal on a traffic light-like display. (Wærsted, 2005.) In case a vehicle from a foreign country is not equipped with an OBU, an invoice is sent by the NPRA to the vehicle’s owner based on information obtained from various vehicle licensing authorities in the vehicle owner’s country. (EPCplc, 2019).

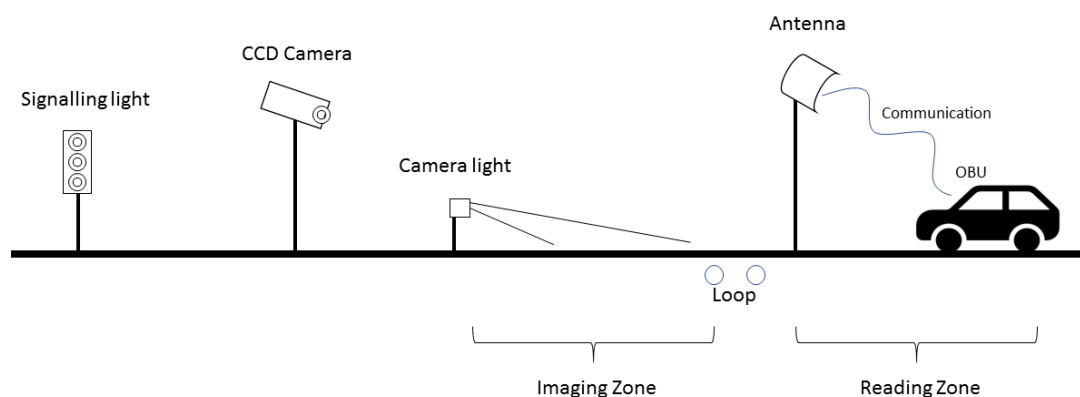


FIGURE 9 AutoPASS charging point (Wærsted, 2005)

According to Wærsted (2005) the EFC lanes have a capacity of handling 1600 vehicles per hour. Wærsted (2005) claims the EFC system is capable of registering tag numbers and photographing license plates even when vehicles are passing the tolling station at high speeds; for example, the speed of 150km/h was mentioned to cause no problems. However, for safety reasons the maximum speed limit through the tolling stations has been set at 60km/h when not in highway environments. The weakest point of the system was noted to be the success rate of the video system due to the possibility of dirt on vehicles' license plates in poor weather conditions. In contrast to AutoPASS lanes, tolling stations with personnel and stations with coin or other payment method machines possibly have a capacity ranging between 200 and 400 vehicles per hour depending on the type machines, drivers, and fees.

The vehicle OBUs communicate with the tolling station system called the Charging Point Equipment (CPE) and the EFCs communicate with the Central System (CS). There is also a Monitoring and Control System that is used for automatic monitoring, controlling, and for maintaining the tolling station equipment. (Pedersli, 2013.)

The tolling stations exchange data with the central system. This data is mainly payment related data, and it is organized in several different files. The files are described in table 3.

TABLE 3 CPE-CS payment data files (Pedersli, 2012)

File type	Description
Transaction File	A transaction list generated at one or more Charging Point
OBU Status File	A 'grey' list that contains the statuses of the operational OBUs
Price File	A list of prices for use at the Charging Point(s)
Picture File	A list of pictures and data included in the picture file that is used for exception handling
Picture Text File	Format of a picture text file

Exception Messages	A list of exception messages
Currency File	A list of various currencies for use at a Charging Point
Operator File	A list of users who have access to the system
Blacklist Credit cards	A list of invalid credit cards
Blacklist	A list containing the foreign (EasyGo) OBUs in operation that are to be rejected
Issuer list	A list that contains all valid Contract Providers

In the AutoPASS system the file traffic goes as follows: the files that are generated by the Charging Point and sent to the Central System are Transactional file, Picture file and Exception file. The other files described in table 3 are generated by the Central system and are sent to the Charging Points to keep their data up to date. (Pedersli, 2012.)

According to Pedersli (2012) in the AutoPASS specification the transaction file generated at the Charging Point is transferred to the Central System periodically and suggests an interval of each 24 hours, but no actual concrete period is defined. This could be problematic if the interval is too long: if something was to happen to the Charging Point, the transaction data could be lost.

Each time a vehicle passes a CP a record is created in the transaction file of whatever payment means the driver of the vehicle used. The fields included in the transaction file may contain, for example, vehicle specific information such as vehicle measurements and properties, direction of passage, lane used, license plates, country; payment information such as price paid, type of payment, credit card information; and OBU information such as OBU ID, OBU status, and OBU passage counter.

To use the AutoPASS system, a user (an individual) has to register an account with any of the toll service providers. From the system's perspective, it doesn't matter which provider the user registers with; the OBUs are interoperable between all providers' systems. A user can drive on the toll roads without an account and an OBU, but he has to pay the bill afterwards and is not eligible to any possible discounts the tolling service provider would have offered to registered customers.

## 7 Road toll system architecture design with blockchain

In this chapter we will take a look at how blockchain technology could be utilized in the road toll architecture of the Norwegian AutoPASS system on a conceptual level. Above all, the feasibility of a blockchain solution is evaluated based on an article by Wüst and Gervais (2017). The chapter's structure will follow the six-stepped DSRP model by Peffers et al. (2006) also presented in chapter 5 and figure 7. The final step, Communication, will not be its own chapter, since the step is inherently included through the creation of this thesis.

### 7.1 Problem identification and motivation

In its current state, the Norwegian AutoPASS tolling system requires users to register an account with a tolling service provider company to be able to utilize the automated electronic road toll payment system. The service provider bills the user on behalf of the toll charger company. Also, since users can choose for themselves which provider they wish to register with, the companies need to exchange user information between each other to maintain toll charging interoperability. In other words, tolling companies, service providers, and the NPRA need to have their own systems for maintaining customer data, and these systems need to facilitate data exchange with each other. These various stakeholders are described in table 4.

TABLE 4 AutoPASS stakeholders and duties

Stakeholder	Duties
NPRA	Directing and managing the Samvirke network interoperability.
AutoPASS Service Provider	User contract management, providing technical solution for vehicle detection (electronic tag), payment collection, customer service for users, pass payments to Toll Companies
AutoPASS Toll Company	Collect payment in case no user agreement is made, finance roads
OBU supplier	Supply users with NPRA approved OBUs
Charging Point Equipment supplier	Supply Tolling Companies with NPRA approved CPE
User	Pay road tolls to Service provider or Toll Company

A registered customer can view certain tolling information on their personal AutoPASS website, including information on when your registered vehicle has passed a toll station. According to the AutoPASS website (AutoPASS, 2019c), it

can take from a few hours to a few days for the toll passages to be visible on the AutoPASS website. This is presumably due to the varying data upload intervals of the Charging Points.

Therefore, the problems identified with the current AutoPASS system are the following:

- There are multiple organizations providing the same service while dependent on each other for user data
- A central entity is required for managing the data between toll companies
- Security and data durability risks: Charging Points may lose transaction data before uploading it to Central System.

Blockchain is a fairly new technology with quite few widely adopted real world applications so far. The purpose of this study is to investigate whether blockchains could provide solutions to the aforementioned problems.

## 7.2 Objectives for a solution

A feasible solution to the problems addressed in the previous chapter would possibly be to utilize blockchain technology. The potential benefits of utilizing blockchains include high data availability as well as always having the latest data versions due to the decentralized nature of blockchains, plus high data security provided by the immutability of blockchains. Blockchains do not necessarily require a central entity, which could be a good argument for cost savings.

One challenge for organizations presented by blockchains today is standardization: the data added to the chain obviously needs to be of certain format in order for mutually independent parties to be able to start utilizing the shared blockchain. In the case of AutoPASS, the blockchain standardization could be provided by the government or its mandated agent, and the road toll companies could agree to these standards and modify their interfaces to be compatible with them.

Another thing to consider is the type of the blockchain; what would be the most suitable in the case of AutoPASS? A permissionless blockchain, a public permissioned, or a private permissioned blockchain? This is dependent on the high-level organizational structure of the road tolling ecosystem, including government level entities, road tolling companies, end customers (vehicle owners and drivers), and also providers of the equipment (such as the OBU devices). The choice of possibly suitable blockchain for the AutoPASS system is determined in the next chapter.

In related work, in 2019 Milligan partners launched a proof of concept called Tolling.Network, which is a distributed ledger solution aimed at realizing “open source blockchain solutions in tolling, transit, parking, and other transportation services” (Milligan partners, 2019). It aims to provide an open source blockchain system to enable for more streamlined road toll interoperability capabilities. It utilizes the open source blockchain Hyperledger.

In the Milligan partners solution, there is no central entity; instead the participants in the tolling system are the customer, the toll agencies, and the asset suppliers. The customer is the end user of the service – the vehicle owner or driver. Toll agencies are the organizations that provide the road tolling capabilities. Asset suppliers are the companies that provide for example the communication devices, the OBUs, for the customers. (Milligan partners, 2019.)

The system's operation consists of a few basic ideas: user certification, an electronic wallet, asset registration, and a transaction ledger. The main concept therefore consists of three topics (Milligan partners, 2019):

- A transparent, shared, and replicated ledger for all tolling transactions
- A secure and unified customer, license plate, and transponder register
- A method to allow any customer to transact directly with any agency

The blockchain utilized in this case is a permissioned blockchain. The blockchain holds information regarding the transactions, the assets, and certifications. The high-level operation is pictured in figure 11. The figure consists of two flows: the registration part and the transaction part.

In the registration part of Milligan partners' model (figure 11), the road toll companies and equipment suppliers register an account on to the blockchain, and the asset suppliers also register their assets (e.g. OBUs) into the blockchain. Since Hyperledger does not rely on cryptocurrency, toll payments have to be processed some other way and therefore Tolling.Network supports electronic wallets such as Apple Pay or Google Wallet. Customers then are required to register an electronic wallet with the service, and information regarding this is stored into the blockchain. (Milligan partners, 2019.)

In the transactions part of the Milligan partners' (2019) model (figure 11) is pictured how the electronic toll payments are collected in the Tolling.Network proof of concept. As a vehicle passes a tolling station, its OBU is read and its status is verified, and the connected account information is also verified on the blockchain to identify the customer. If the transaction is approved, the transaction is stored into the blockchain, and the payment is sent to the tolling agency. (Milligan partners, 2019.)



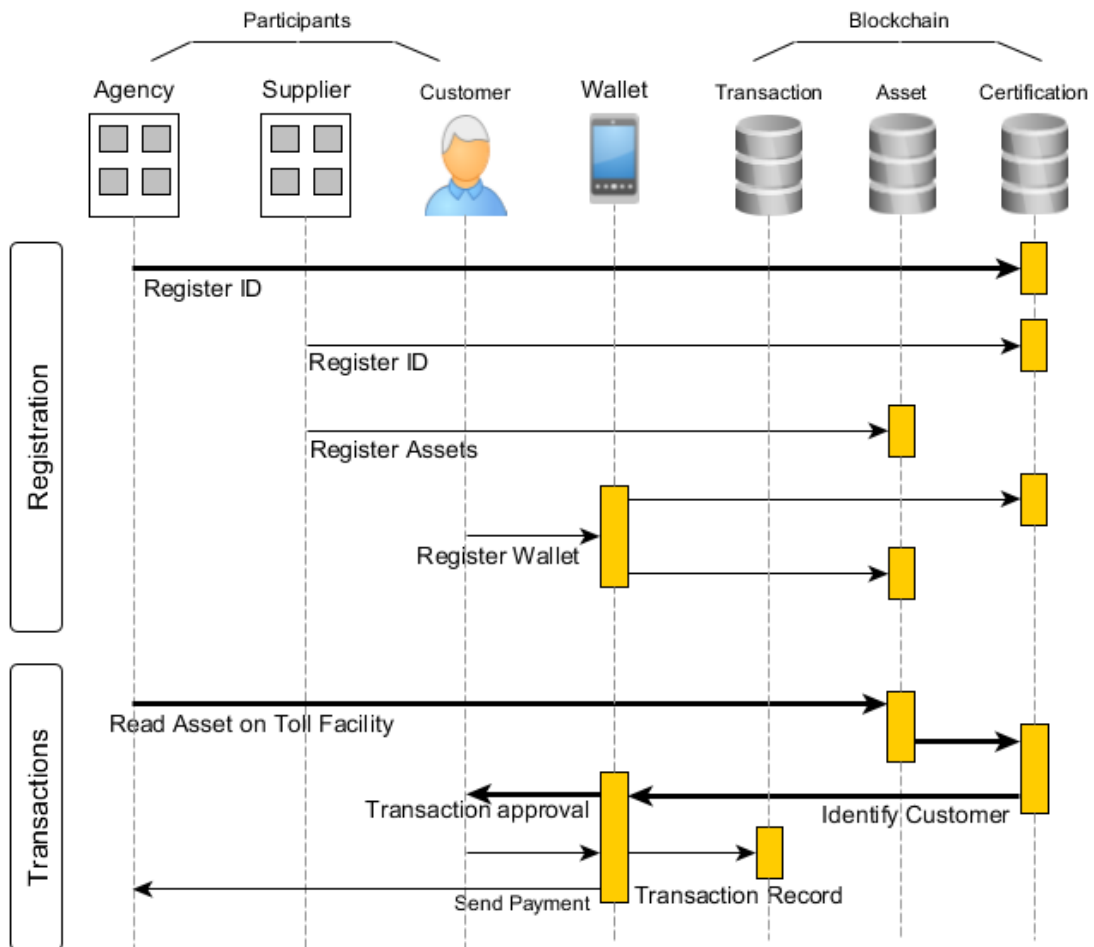


FIGURE 10 Operation principles among participants and the blockchain in the Unified Tolling Network (Milligan partners, 2019)

### 7.3 Design and development

To evaluate the feasibility of blockchains in the AutoPASS system, we will be utilizing a flow chart designed by Wüst and Gervais (2017) and presented in figure 9. With this, we can get a sense of what type of blockchain could be suitable in this situation. The chart has 4-6 questions with yes/no transitions for evaluating the applicability of blockchains in a given situation, finishing in a recommendation of either a permissionless blockchain, a public permissioned blockchain, a private permissioned blockchain, or to not use a blockchain to begin with.

The flow starts with the prompt “Do you need to store state?”. Storing state basically means storing data. If there is no need to store data, using any form of database is of no use (Wüst & Gervais, 2017). In the current AutoPASS system architecture there is a need to store, for example, customer information such as subscription status, and transaction data, so the flow continues to “Are there mul-

multiple writers?”. Considering the multitude of AutoPASS Charging Points as writers pushing data into the blockchain, the answer is yes. Therefore, the flow continues to “Can you use an always online TTP?”. TTP stands for Trusted Third Party. In the case of AutoPASS, the TTP could be the NPRA since it is the administrative entity responsible for the road tolling capabilities in Norway. It seems though that the NPRA does not handle the subscription or tolling transaction data; it is the duty of the Service Providers or the Tolling Companies, which handle the day-to-day road tolling operations. There is a multitude of Service Provider companies in Norway and, though governed by common rules set by the NPRA, they are individual companies with their individual systems which need to communicate with each other. Therefore, it can be said that there is no common always online TTP in the AutoPASS system. Also, considering the fact that the AutoPASS service can also be used with the same OBU in Sweden and Denmark through their EasyGo partnership, blockchain could be a suitable solution. In a case where there are several countries with their individual road tolling administrations, service providers, and central systems, it could be reasonable to utilize blockchains. A blockchain based system would allow for cutting out the central systems and letting the Charging Points upload tolling data directly to the blockchain and check valid user and OBU subscriptions, upheld by the tolling companies. This would remove the need for independent companies to each store their own data and transfer it among one another, since all participants would hold a copy of the same data in the blockchain ledger.

Continuing with the scenario where there is not an always online TTP, the next step in the flow by Wüst and Gervais (2017) is “Are all writers known?”. The writers in the case of AutoPASS would be the Charging Points: they are the parties appending e.g. vehicle passage and transaction data into the blockchain. Also, the Service Providers can be considered writers since they are the party handling user subscriptions. The writers are, therefore, known, and this rules out the feasibility of utilizing a Permissionless blockchain.

The next question is about trust towards the writers, which had earlier been defined as the Charging Points. Charging Points are managed by their respective companies which, though governed by the NPRA, don't necessarily uphold trust towards one another in the sense that trust is defined in the context of blockchains. This leads us to the choice between a Public permissioned blockchain and a Private permissioned blockchain, the choice of which is determined by whether public verifiability is required. The main difference between the two is that in a public blockchain anyone can read the contents of the chain and thereby verify the data validity, while in a private blockchain only a limited number of participants is allowed to read the chain contents (Wüst & Gervais, 2017). A private permissioned blockchain does not provide significant benefits over a traditional database solution, so if we wish to promote openness and gain the benefits of utilizing blockchains, the reasonable choice is Public permissioned blockchain. This way the blockchain participants can have access to the chain contents, and the tolling data can be utilized by other parties outside the tolling companies as well.

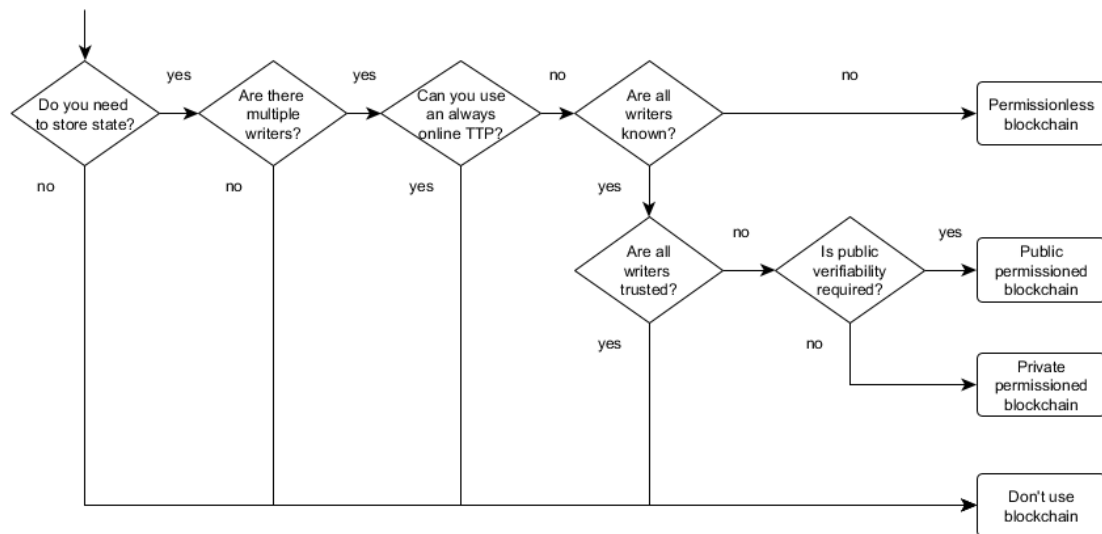


FIGURE 11 Flow chart for determining whether blockchain is appropriate for solving a problem (Wüst & Gervais, 2017)

Strictly following Wüst and Gervais' (2017) blockchain usage feasibility flow, the conclusion would be that utilizing a Public Permissioned blockchain could be justified in the case of AutoPASS. The specific choice of which blockchain to use – for example Ethereum for its smart contract capabilities or Hyperledger for its lack of cryptocurrency qualities – should still be carefully considered.

Also, with 5G communication on the way, it should be reconsidered whether the Charging Points need to be physical structures along the roads to begin with. Their functionalities along with road toll payment capabilities could be integrated within the vehicle's systems, and therefore the Charging Points could be virtual. They could simply be a set of GPS coordinates that the infrastructure in the vehicle monitors and launches a payment transaction when a specific coordinate has been reached on the road. Virtual road toll locations could be managed by, for example, the regional service providers, and they could be dynamic: toll locations could be changed depending on congestion and to channel traffic.

## 7.4 Evaluation

Based on the results gathered from looking at AutoPASS through the flow chart by Wüst and Gervais (2017), it can be said that utilizing a blockchain in a road toll system such as AutoPASS might be a feasible solution over traditional, centralized database solutions (table 1). A public permissioned blockchain could possibly be the best blockchain based solution.

Wüst and Gervais shape their opinion on the use of blockchains as follows: "In general, using an open or permissioned Blockchain only makes sense when

multiple mutually mistrusting entities want to interact and change the state of a system, and are not willing to agree on an online trusted third party.” (Wüst & Gervais, 2017, p. 2). In the process flow examined in chapter 7.3 the conclusion was that even though in AutoPASS there is an agreed upon and trusted central entity, the NPRA, it is not the party that is responsible for data handling; it is primarily a governing entity for the entire tolling ecosystem. Presumably, the tolling companies and the service providers are ultimately the parties facilitating and executing the day-to-day data handling and operations. Therefore, having a multitude of companies doing essentially the same thing, it would be justified to consider utilizing a public permissioned blockchain solution to reduce redundancy in, for example, transaction and user data. This leads us again to the concept of autonomies: a shared blockchain solution would still require standardization either from a higher authority such as the NPRA, or through an agreement between the individual companies. This is because otherwise the companies could decide for themselves what technology solutions they wish to utilize (organizational-, and design autonomy). In the case of AutoPASS the standardization is defined by the NPRA, but this is an especially notable issue when considering multinational road tolling systems: different countries have different legislation and therefore different requirements for the system (e.g. considering data security and privacy). The standardization for such a system can be difficult.

Since Milligan partners used the Hyperledger blockchain protocol in their road tolling concept, there is no cryptocurrency involved. Instead, they implemented a wallet payment system, in which, for example, Apple Pay can be utilized to pay the tolls. Therefore, no data regarded as currency is stored in the blockchain; only the transaction, asset, and certification data are stored in the chain. The payments go directly from the customer’s wallet to the tolling agency’s account. If a different blockchain was utilized – Ethereum for example – the cryptocurrency capabilities could be implemented as well. This would therefore require the creation or selection of a suitable cryptocurrency, and a downside of this would be that the cryptocurrency would be locked to be used only in this environment, whereas Apple Pay and such can be more versatile in their implementation and be utilized across a variety of services. This would present a barrier – albeit a relatively small one – to the adoption process of the system for end users. In order to start using the system, users would first need to convert traditional currency to cryptocurrency and place it into their toll subscription account or wallet. Also, the road tolling companies would have to agree on a mutual cryptocurrency – or standardize and create their own. For example, the value of Bitcoin fluctuates excessively to be utilized in a system such as this. The cryptocurrency for road toll payments – regardless whether the system is national or multinational – needs to be tied to an existing FIAT currency for its value to be predictable and equal for all users.

## 8 Discussion

The aim of this thesis was to investigate whether blockchain, the decentralized transaction ledger technology, is suitable for use in a road tolling architecture. The research questions were as follow:

- How could blockchain technology be utilized in a road toll system?
  - o Is decentralized technology suitable for a road toll architecture?
  - o What could the architecture of a road toll system utilizing blockchain technology be like?
  - o Is using blockchain technology in a road toll system justified?

The research was done as a literature review combined with a design science method, the Design Science Research Process. To conduct the literature review, information of electronic road tolling systems was needed. I managed to find quite detailed information on AutoPASS, a Norwegian road tolling system, and used this as the main literature material. The suitability of blockchain technology for AutoPASS was then evaluated using DSRP as the research model and a flow chart by Wüst and Gervais (2017) as a step-by-step reference guide. Based on the research, the conclusion was that it can be reasonable or suitable to utilize blockchain technology. Therefore, the answer to the first research question “Is decentralized technology suitable for a road toll architecture?” is yes, because decentralization could remove the need for a centralized system and streamline operations among a multitude of companies’ systems, and also remove the need for an always online third party, and therefore lower overall costs of operation, for example. Utilizing blockchains would inevitably require creating standards for the used blockchain system, in order to be able to expand the tolling system to neighboring countries with less effort, for example.

The second research question was “What could the architecture of a road toll system utilizing blockchain technology be like?”. The system could be one which does not utilize cryptocurrency; instead the road toll payments could be paid via an electronic wallet such as Apple Pay or Google Wallet. This wallet feature would be implemented as an essential part of the tolling system. No data considered as currency would be stored in the blockchain; only the data regarding transactions, assets (e.g. communication devices such as OBU), and certifications is stored in the chain.

The third research question was “Is using blockchain technology in a road toll system justified?”. The answer is yes, since blockchain protocols are usually open source software and therefore available for all to use, which means utilizing them can create cost savings in the case there are several actors with similar needs that could be met with a decentralized solution. Also, blockchains can create trust in an untrusting environment by offering equal transparency for participants.

A further study into this topic could delve deeper into the specific properties of suitable blockchain protocols. For example, Ethereum, Cardano, or Hyperledger could be investigated, compared and it could be researched what particular properties they would offer in the design of a road tolling architecture.

Also, road tolling systems from other countries in addition to Norway could be investigated.

## REFERENCES

- Abiteboul, S., Hull, R., Vianu, V. (1995). *Foundations of Databases*. Addison-Wesley Publishing Company, Inc.
- Amazon Web Services. (2018). Data privacy. Retrieved 3.9.2018 from <https://aws.amazon.com/compliance/data-privacy-faq/>
- AutoPASS. (2018). Regulation on toll service provision for tolls and ferry tickets (the Toll service provider Regulation). Retrieved 29.5.2019 from <https://www.autopass.no/en/about-autopass/toll-service-provision>
- AutoPASS. (2019a). Domain statement for AutoPASS Samvirke Revision 1.1. Retrieved 29.5.2019 from <https://www.autopass.no/en/about-autopass/toll-service-provision/attachment/2502667?ts=16a69016038&download=true>
- AutoPASS. (2019b). About toll service provision. Retrieved 29.5.2019 from <https://www.autopass.no/en/about-autopass/toll-service-provision>
- AutoPASS. (2019c). AutoPASS Customer service. Retrieved 28.5.2019 from <https://minside.autopass.no/kundeservice>
- Azzouna, N., B. & Guillemin, F. (2004). Impact of peer-to-peer applications on wide area network traffic: an experimental approach. In *IEEE Global Telecommunications Conference (GLOBECOM '04.)*, vol. 3 (1544-1548). Dallas, Texas.
- Bartlett, G., Heidemann, J., Papadopoulos, C., Pepin, J. (2007). *Estimating P2P Traffic Volume at USC*. Technical Report ISI-TR-2007-645. USC/Information Sciences Institute.  
[https://www.researchgate.net/publication/268427175\\_Estimating\\_P2P\\_traffic\\_volume\\_at\\_USC](https://www.researchgate.net/publication/268427175_Estimating_P2P_traffic_volume_at_USC)
- Beck, R., Czepluch, J. S., Lollike, N. & Malone, S. (2016). Blockchain – the gateway to trustfree cryptographic transactions. *Association for Information Systems Research Papers*, 153. [https://aisel.aisnet.org/ecis2016\\_rp/153](https://aisel.aisnet.org/ecis2016_rp/153)
- Bitcoin wiki. (2019). Block timestamp. Retrieved 30.5.2019 from [https://en.bitcoin.it/wiki/Block\\_timestamp](https://en.bitcoin.it/wiki/Block_timestamp)
- Buterin, V. (2017). The Meaning of Decentralization. Retrieved 17.5.2019 from <https://medium.com/@VitalikButerin/the-meaning-of-decentralization-a0c92b76a274>

- Butler, B. (2013). PaaS Primer: What is platform as a service and why does it matter? Retrieved 3.9.2018 from <https://www.networkworld.com/article/2163430/cloud-computing/paas-primer--what-is-platform-as-a-service-and-why-does-it-matter-.html>
- Casino, F., Dasaklis, T., K., Patsakis, C. (2019). A systematic literature review of blockchain-based applications: Current status, classification and open issues. *Telematics and Informatics*, 36, 55-81.
- Cisco. (2017). Cisco Visual Networking Index: Forecast and Methodology, 2016-2021. Retrieved 22.11.2018 from <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf>
- Cullen, C. (2018). Global Internet Phenomena Report: Netflix is approximately 15 per cent of worldwide downstream traffic. Retrieved 22.11.2018 from <https://www.sandvine.com/blog/global-internet-phenomena-report-netflix-is-15-of-worldwide-downstream-traffic>
- Dunaytsev, R., Moltchanov, D., Koucheryavy, Y., Strandberg, O., Flinck, H. (2012). A Survey of P2P Traffic Management Approaches: Best Practices and Future Directions. *Journal Of Internet Engineering*, 5(1), 318-330.
- Eberspächer, J., Schollmeier, R., Zöls, S., & Kunzmann, G. (2004). Structured P2P networks in mobile and fixed environments. In *Proceedings of 2nd International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks (HET-NETs)*, T4, (1-25), Ilkley, UK.
- EPCplc. (2019). Norwegian Road Tolls. Retrieved 28.5.2019 from [https://www.epcplc.com/norwegian\\_road\\_tolls](https://www.epcplc.com/norwegian_road_tolls)
- Glaser, F. (2017). Pervasive Decentralisation of Digital Infrastructures: A Framework for Blockchain enabled System and Use Case Analysis. In *Proceedings of the 50th Hawaii International Conference on System Sciences (HICSS 50)*, (1543-1552). Waikoloa Village, Hawaii.
- Gupta, M. (2017). *Blockchain for Dummies, IBM Limited Edition*. John Wiley & Sons, Inc.
- Helpinen, V. (2016, 11 February). Tietulleja Helsingin ympärille puuhataan taas - 340 euron lisälasku autoilijalle. Yle.fi. Retrieved 31.5.2019 from <https://yle.fi/uutiset/3-8662824>
- Hevner, A.R., March, S.T., & Park, J. (2004). Design Research in Information Systems Research. *Mis Quarterly*, 28(1), 75-105.



- Hong, Z., Wang, W., Cai, W. & Leung, V. C. M. (2017). Connectivity-Aware Task Outsourcing and Scheduling in D2D Networks. In *26th International Conference on Computer Communication and Networks (ICCCN)*, (1-9). Vancouver, BC, Canada.
- Honkanen, P. (2017). Lohkoketjuteknologian lupaus. Arcada Working Papers 1/2017.
- Kinnunen, T., K., Leviäkangas, P., Kostainen, J., Nykänen, L., Rouhiainen, K., Finlow-Bates, K. (2017). *Lohkoketjuteknologian soveltaminen ja vaikutukset liikenteessä ja viestinnässä*. Liikenne- ja viestintäministeriön julkaisuja 12/2017.
- Kisembe, P. & Jeberson, W. (2017). Future Of Peer-To-Peer Technology With The Rise Of Cloud Computing. *International Journal of Peer to Peer Networks*, 8(2), 45-54.
- Lempinen, T. (2017, 31 May). Trafi varautunut jo tietullien käyttöönottoon Suomessa – tämä sanavalinta kertoo paljon. Retrieved 31.5.2019 from <https://www.is.fi/autot/art-2000005233714.html>
- Lewis, A. (2017). Distributed Ledgers: Shared control, not shared data. Retrieved 23.5.2019 from <https://bitsonblocks.net/2017/01/09/distributed-ledgers-shared-control-not-shared-data/>
- Lissounov, K. (2016). What's the difference between peer to peer and client server? Retrieved 30.10.2018 from <https://www.resilio.com/blog/whats-the-difference-between-peer-to-peer-and-client-server>
- Lysfjord, N. (2013). AutoPASS Requirement Specification 1.3 – On Board Units for Norwegian AutoPASS – Project description and Scope of Work. Retrieved 28.5.2019 from <https://kgv.doffin.no/ctm/Supplier/Documents/Folder/74570>
- Madhukar, A. & Williamson, C. (2006). A Longitudinal Study of P2P Traffic Classification. In *14th IEEE International Symposium on Modeling, Analysis, and Simulation*, (179-188). Monterey, CA, USA.
- Mattila, J., Seppälä, T. (2015). Laitteet pilveen – vai pilvi laitteisiin? Keskustelunavauksia teollisuuden ja yhteiskunnan digialustojen uusista kehitystrendeistä. ETLA Raportit No 44. Retrieved 24.5.2019 from <https://www.etla.fi/wp-content/uploads/ETLA-Raportit-Reports-44.pdf>
- Mastercard Incorporated. (2012). *Fiscal Year 2012 Form 10-K Annual Report*. Retrieved 23.5.2019 from

[https://s2.q4cdn.com/242125233/files/doc\\_financials/annual/MA-2012-Annual-Report.PDF](https://s2.q4cdn.com/242125233/files/doc_financials/annual/MA-2012-Annual-Report.PDF)

- Merriam-Webster. (2019). Synonyms and Antonyms of autonomy. Retrieved 23.5.2019 from <https://www.merriam-webster.com/thesaurus/autonomy>
- MetaHash AG. (2019). *MetaHash whitepaper*. Retrieved 23.5.2019 from [https://static.metahash.org/docs/MetaHash\\_WhitePaper\\_EN.pdf?v=5](https://static.metahash.org/docs/MetaHash_WhitePaper_EN.pdf?v=5)
- Min, S., Holliday, J., Cho D. (2006). Optimal Super-peer Selection for Large-scale P2P System. In *2006 International Conference on Hybrid Information Technology*, (588-593). Cheju Island, South Korea.
- Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved 28.2.2018 from <https://bitcoin.org/bitcoin.pdf>
- O'Brien, J. & Marakas, G.M. (2010). *Management Information Systems*. New York, NY: McGraw-Hill Irwin.
- Palermo, F. (2018, 2 November). Is Blockchain The Answer To Election Tampering? Retrieved 31.5.2019 from <https://www.forbes.com/sites/forbestechcouncil/2018/11/02/is-blockchain-the-answer-to-election-tampering/#53b8515a19c4>
- Pedersli, P., E. (2012). AutoPASS – Requirement specification 4.3 – Charging point equipment – Central system interface. Retrieved 28.5.2019 from <https://kgv.doffin.no/ctm/Supplier/Documents/Folder/74570>
- Pedersli, P., E. (2013). AutoPASS – Requirement specification Doc 4.1 – Charging point equipment – Requirements for automatic charging points. Retrieved 28.5.2019 from <https://kgv.doffin.no/ctm/Supplier/Documents/Folder/74570>
- Peffer, K., Tuunanen, T., Gengler, C., E., Rossi, M., Hui, W., Virtanen, V. & Bragge, J. (2006). The Design Science Research Process: A Model For Producing And Presenting Information Systems Research. In *Proceedings of First International Conference on Design Science Research in Information Systems and Technology (DESRIST'06)*, (83-106). Claremont, CA.
- Red Belly Blockchain. (2018). The Red Belly Blockchain Experiments. Retrieved 23.5.2019 from <https://redbellyblockchain.io/papers/redbellyblockchain-experiments.pdf>
- Risius, M. & Spohrer, K. (2017). A Blockchain Research Framework. *Business & Information Systems Engineering*, 59(6), 385–409.
- Salonen, S., Halunen, K., Korhonen, H., Lähteenmäki, J., Pussinen, P., Vallivaara, V., Väisänen, T., Ylén P. (2017). Lohkoketjuteknologian

mahdollisuudet ja hyödyt sosiaali- ja terveydenhuollossa. Valtioneuvoston selvitys- ja tutkimustoiminnan julkaisusarja 80/2017.

<http://urn.fi/URN:ISBN:978-952-287-490-0>

- Schollmeier, R. (2002). A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications. In *Proceedings of the First International Conference on Peer-to-Peer Computing*, (101-102). IEEE, Linköping, Sweden.
- Schulze, H., Mochalski, K. (2009). Ipoque Internet Study 2008/2009.
- Sedgwick, K. (2018). No, Visa Doesn't Handle 24,000 TPS and Neither Does Your Pet Blockchain. Retrieved 12.12.2018 from <https://news.bitcoin.com/no-visa-doesnt-handle-24000-tps-and-neither-does-your-pet-blockchain/>
- Sito Oy. (2016). *Ajoneuvoliikenteen hinnoittelun hallinnollis-lainsäädännöllinen selvitys*. HSL:n julkaisuja 3/2016. Retrieved 31.5.2019 from [https://www.hsl.fi/sites/default/files/uploads/hsl\\_julkaisu\\_3\\_2016\\_ajoneuvoliikenteen\\_hinnoitteluselvitys\\_hallinnollinen.pdf](https://www.hsl.fi/sites/default/files/uploads/hsl_julkaisu_3_2016_ajoneuvoliikenteen_hinnoitteluselvitys_hallinnollinen.pdf)
- Underwood, S. (2016). Blockchain Beyond Bitcoin. *Communications of the ACM*, 59(11), 15-17.
- Unuth, N. (2018). Skype Changes From P2P to Client-Server Model. Retrieved 1.10.2018 from <https://www.lifewire.com/skype-changes-from-p2p-3426522>
- Veijalainen, J., Eliassen, F. & Holtkamp, B. (1992). The S-transaction Model. In book A. K. Elmagarmid, Database Transaction Models For Advanced Applications.
- Vermeulen, J. (2016). VisaNet – handling 100,000 transactions per minute. Retrieved 11.12.2018 from <https://mybroadband.co.za/news/security/190348-visanet-handling-100000-transactions-per-minute.html>
- de Vries, A. (2018). Bitcoin's Growing Energy Problem. *Joule*, 2(5), 801–809.
- Wærsted, K. (2005). Urban Tolling in Norway – Practical Experiences, Social and Environmental Impacts and Plans for Future Systems. *PIARC Seminar on Road Pricing with emphasis on Financing, Regulation and Equity*. Cancun, Mexico, 2005, April 11-13. Retrieved 10.7.2019 from <https://www.piarc.org/ressources/documents/281,2.1-Waersted-0405C11.pdf>
- White, E. (2013). A Look Inside the Visa Network Center Powering the Global Economy. Retrieved 23.5.2019 from

<https://www.visa.com/blogarchives/us/2013/03/05/a-look-inside-the-visa-network-center-powering-the-global-economy/index.html>

Wüst, K., Gervais, A. (2017). Do you need a Blockchain? IACR Cryptology ePrint Archive 2017, 375.

Zhu, C. (2010). *Streaming Media Architectures, Techniques, and Applications: Recent Advances*. IGI Global; 1st edition.