UNIVERSITY OF TARTU

Institute of Computer Science

Software Engineering Curriculum

Sven Mitt

Blockchain application - case study on Hyperledger Fabric

Master’s Thesis (30 ECTS)

Supervisor(s): Luciano García-Bañuelos  
Fredrik Milani

Tartu 2017

Blockchain application - case study on Hyperledger Fabric

Abstract:

Blockchain and smart contract based systems are slowly gaining more foothold. There is research how it impacts the software architecture and what can be done with smart contracts. In this paper we study how implementation of blockchain based on system Hyperledger project compares to currently used systems. We implement a parking spot application for multisided market using Hyperledger Fabric and compare it to iterative language contract. In the end we have a working prototype covering predetermined use cases.

Keywords:

Distributed Ledger Technology, Smart contract, Blockchain Technology, Hyperledger Fabric

**CERCS:**

Pealkiri eesti keeles

Lühikokkuvõte:

Blockchain tüüpi rakendused on vaikselt hakanud maad võtma ning …

Võtmesõnad:

Distributed Ledger Technology, Smart contract, Blockchain Technology, Hyperledger Fabric

**CERCS:**

Table of Contents

[1 Introduction 4](#_Toc499584003)

[1.1 Scope 4](#_Toc499584004)

[1.2 Motivation 4](#_Toc499584005)

[1.3 Research problem 4](#_Toc499584006)

[1.4 Summary of contribution 4](#_Toc499584007)

[2 Related Work 5](#_Toc499584008)

[2.1 Search strategy 5](#_Toc499584009)

[2.2 Technology for two- vs multi-sided markets 5](#_Toc499584010)

[2.3 Blockchain 5](#_Toc499584011)

[2.4 Distributed Ledger Technology 6](#_Toc499584012)

[2.5 Smart contract 6](#_Toc499584013)

[3 Background 7](#_Toc499584014)

[3.1 Current business models in use 7](#_Toc499584015)

[3.2 Chain core 7](#_Toc499584016)

[4 Evaluation 9](#_Toc499584017)

[4.1 Business model for parking spot application 9](#_Toc499584018)

[4.1.1 Adding and removing parking spot from listing 9](#_Toc499584019)

[4.1.2 Using parking spot: paying in advance 10](#_Toc499584020)

[4.1.3 Using parking spot: paying after parking ended 10](#_Toc499584021)

[4.1.4 Reserving a parking spot 11](#_Toc499584022)

[4.1.5 … 11](#_Toc499584023)

[4.1.6 … 11](#_Toc499584024)

[4.2 Software architecture 11](#_Toc499584025)

[4.3 Implementation 11](#_Toc499584026)

[4.4 Smart contract for parking spot 12](#_Toc499584027)

[5 Conclusions 13](#_Toc499584028)

[6 References 14](#_Toc499584029)

[Appendix 15](#_Toc499584030)

[I. Glossary 15](#_Toc499584031)

[II. License 16](#_Toc499584032)

# Introduction

## Scope

Scope of this thesis is to research blockchain suitability for parking spot application and implement using predetermined use cases. Platform for blockchain is Hyperledger Fabric using smart contract language chaincode.

## Motivation

Current parking applications are divided into two categories.

In first one you show up, find a parking spot and buy a ticket for predetermined time. This is done using separate machine, which accepts cash or card and prints out the parking ticket, or app on a smartphone. The cost of using parking app in Estonia is 32 euro cents. That is mediator fee collected by the parking app platform owner.

In the second type, you get a ticket before you enter through a barrier gate. Paying for parking time is done before leaving the parking lot. The fee for platform owner is included in the parking price. The standard fee is between 10-20%.

Both of those technical solutions is driven by a third party that has built a platform for managing the app and hardware and takes a hefty cut when you use their service.

We are interested in searching a cheaper solution. One that has smaller mediator fee and does not include vendor lock-in. That means the platform is shared between the companies that are interested providing parking spots.

What if you are in a hurry and need a parking spot and you do not have time to lose. There is no reservation possibility in current business models nor assigning specific parking spot per vehicle.

## Research problem

We define research questions accordingly:

1. Is blockchain viable for implementing parking solution?
2. Is Hyperledger Fabric viable for implementing parking solution?
3. Is Hyperledger Fabric smart contract language chaincode powerful enough to implement parking solution on its own?

To answer these questions, we research whitepapers that deal with differences of software systems used today and compare them to blockchain based solutions. Before implementing such solution, we need to know if blockchain is at all appropriate for these type of systems.

## Summary of contribution

In the end of this theses we have implemented selected use cases for parking application and shown that it is possible to provide a viable system using Hyperledger Fabric as a blockchain platform.

This theses is structured as follows.

* Section 2 presents related work and state of art.
* Section 3 describes the technical background and section 4 implementation of the software.
* Section 5 presents the evaluation of results
* Section 6 summarizes the paper and outlines future work to be done

# Related Work

## Search strategy

To find state of art we executed a search using keywords:

* Blockchain
* Smart contract
* Distributed Ledger Technology
* Product-centric blockchain
* Blockchain software connector
* Two sided and multisided markets
* Hyperledger Fabric

Search was conducted in digital libraries to find any whitepaper related to search string. Databases used:.

1. Scopus
2. Web of Science
3. IEEE Xplore
4. ACM Digital Library
5. SpringerLink
6. ScienceDirect
7. Google Scholar

## Technology for two- vs multi-sided markets

To understand how blockchain enables to provide better business value, we have to understand how it is produced currently [1].

Most business models today depend on two-sided platforms, where a trusted third party is involved for mediating the interaction between end-users. End users do not trust each other, so they need a third party that can be trusted by all. Transactions and resolving data left on the shoulders of the mediator and its platform. The platform must keep records and use architectural patterns to resolve transaction problems. All that involves cost since the third party must develop and maintain a platform for mediating and will charge end users for using it.

Multisided platforms enable end users trust each other since data is decentralized and all parties have a full copy of database. The problem is how to resolve simultaneous changes and how to synchronized data to all parties.

That is the reason why blockchain is multisided platform enabler [2]. It allows to create specific datastore that contains immutable blocks and gives order to transactions. Mediator in this case will only monitor the trading and revenue is split according to contract and business rules.

## Blockchain

Blockchain data structure is list of blocks that are timestamped, immutable and in strict order. All transactions are immutable and that allows involved parties to be certain that no data, that already has been submitted to blockchain, is not tampered with. Blockchain enables this behavior by requiring consensus from interested parties and only committing the transaction after reaching consensus.

If there are concurrent changes, the consensus algorithm will guarantee that only one version of database will be in use. That provides companies the insurance that they can see the entire database in and they will left out in vote for consensus.

Criteria’s for platform to be blockchain enabled [3]:

1. Must be durable or capital good – parking spot is an example
2. Fully shared database, cannot be fragmented
3. Multiple concurrent writers
4. Need for one absolute version (not like git)
5. Must depend on previous modifications
6. Need for trust for untrusted parties and permissions on items
7. Need to remove intermediation (someone who maintains joint database and enables trust and multiversion concurrency)
8. Blockchain mining mechanism increases latency – since parking spot app is not real time system, then it is not an issue
9. Data stored in block chain is limited and should be minimal

There can be other technologies for implementing the solution for parking platform, but this theses concentrates on enabling platform on blockchain and specifically Hyperledger Fabric infrastructure and using chaincode

## Distributed Ledger Technology

Distributed ledger technology uses blockchain to verify and store transactions in distributed manner.

## Smart contract

Smart contract [4] adds functionality to blockchain as it is a computer program that can access blockchain blocks and the history. Functionality was added to second generation of blockchain.

Smart contract is a computer program that is stored in distributed database. They can act like database triggers, conditionals and add business logic to transactions. Quite powerful tool if smart contract implementation is Turing complete – this is the reason Hyperledger Fabric was chosen as backend for this theses.

Since smart contracts move digital values that also have real monetized value in the real world, a bug in a contract will be costly. Writing even a simple smart contract need economic thinking and can lead to unexpected results [3].

# Background

## Current business models in use

We provide a description of business model of parking model using Business Process Model and Notation (BPMN).

In conventional application, the platform owner takes the responsibility to mediate the transaction between the parking spot owner and renter.

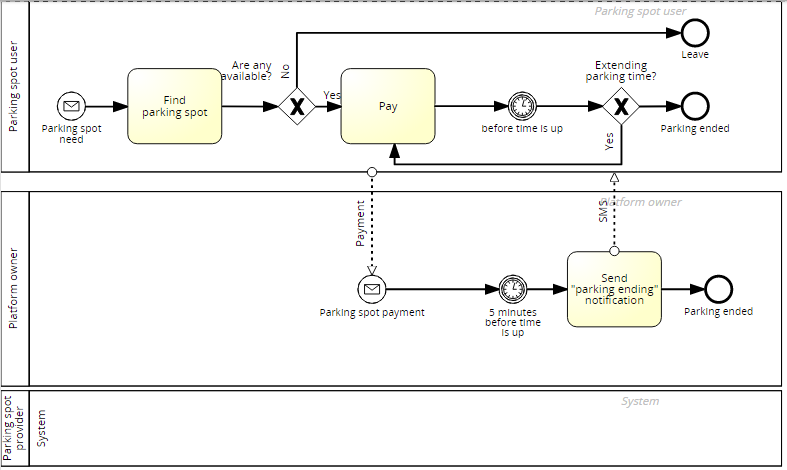


Figure 1: BPMN of conventional parking business model

## Hyperledger fabric

Founded by Linux Foundation, Hyperledger project builds upon collaborative approach to develop blockchain technologies. Hyperledger Fabric is one of the projects.

Hyperledger Fabric is blockchain based enterprise software for issuing and transferring financial assets [5]. It also includes role based permission handling, a must have capability for private and enterprise usage. It is open source software licensed under Apache License, Version 2.0 [6].

To implement permissioned ledgers, it offers *channels* – allowing a transaction participants to keep the details of transaction only visible to themselves.

Ledger consists of two components:

* world state – it contains the status of data of any given point in time
* transaction log – contains the changes that are applied to data

Smart contract language is called chaincode – it has support for Java, Node.js and Go-lang.

Data model:

* **Asset** – any type of capital or durable goods. Issuing and retiring assets is supported by the platform.
* **Transaction** – move value from inputs to output. Input can be previous transaction output or in case of assignment of new Asset, a new issuance of unit
* **Blocks** –multiple transactions are batched into blocks and those make up the blockchain. New block is written when a consensus program is executed successfully.

# Evaluation

In this section, we describe the implementation of application and give an answer to research questions:

1. Is blockchain viable for implementing parking solution?
2. Is Hyperledger Fabric viable for implementing parking solution?
3. Is Hyperledger Fabric smart contract language chaincode powerful enough to implement parking solution on its own?

The system must be implemented for multisided market, application that brings together parking spot owners and parties, who wish to rent parking spots.

## Business model for parking spot application

We provide a description of business model to be implemented using Business Process Model and Notation (BPMN).

We have three actors to business transactions: parking spot provider, parking spot renter (person who wishes to reserve a parking spot) and the system (implemented in thesis) that makes this possible.

## Adding and removing parking spot from listing

Parking spot provider can be any person or business who has business interest of creating revenue by renting out one or more parking spots. Owner must be able to add the parking spot to list of available parking spots and remove later.

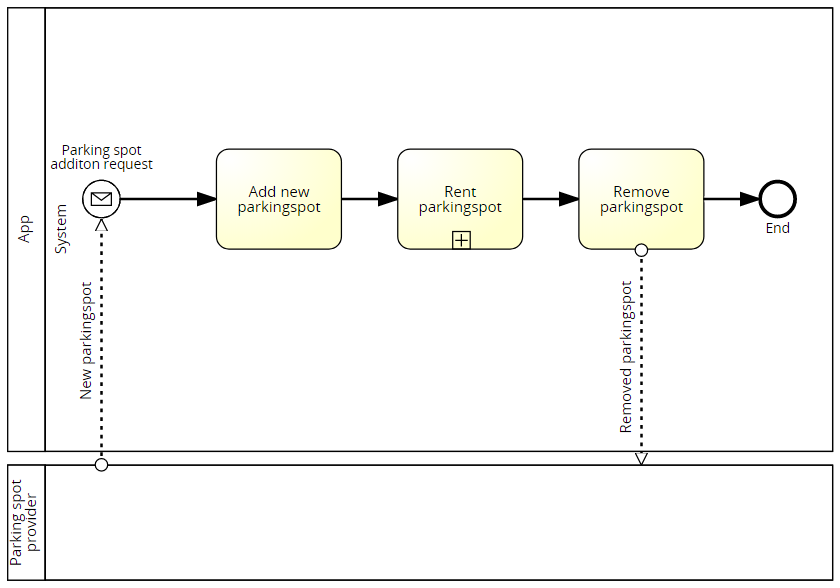


Figure 2: BPMN of adding and removing parking from listing

## Using parking spot: paying in advance

In this scenario, person looking for a parking spot pays the full amount beforehand. The renter will specify the timeslot for using the parking spot. When the time is ending, the renter will receive a message from the system to create a new timeslot (and keep the same parking spot). If no notification is received from the renter, the parking spot will be available for usage after time slice has ended.

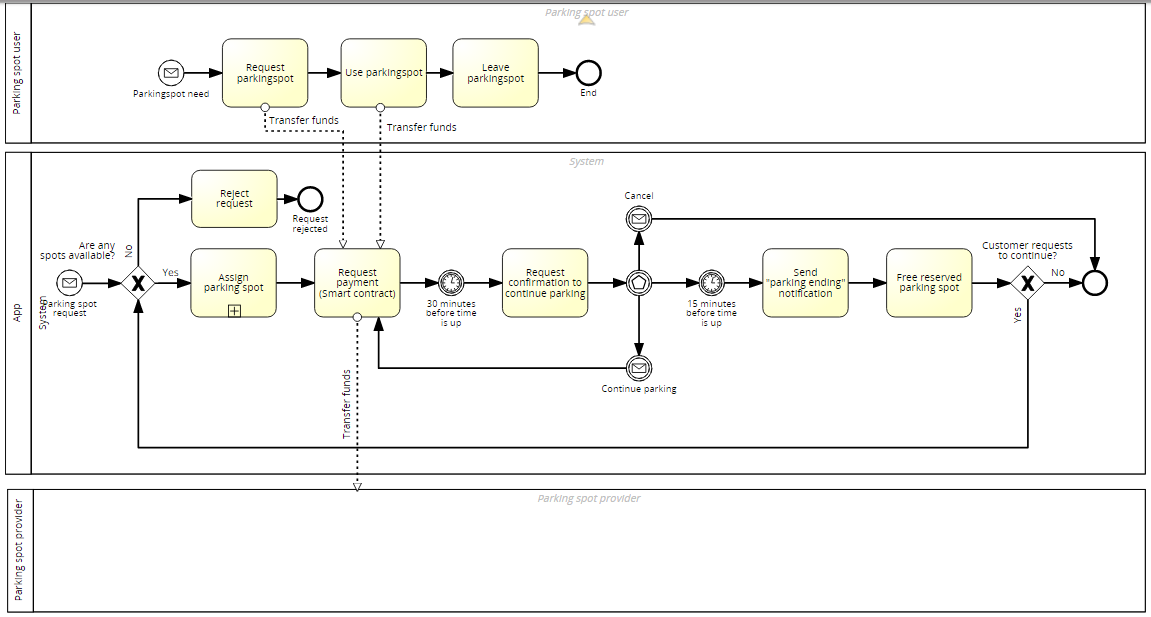


Figure 3: BPMN of paying for parking spot in advance

## Using parking spot: paying after parking ended

Not always renter knows how long he will be needing a parking spot. In that case it is more client friendly to pay after user has ended the contract and calculate the cost. This business model needs a predetermined maximum time limit, in case the customer forgets and the fee can expensive.

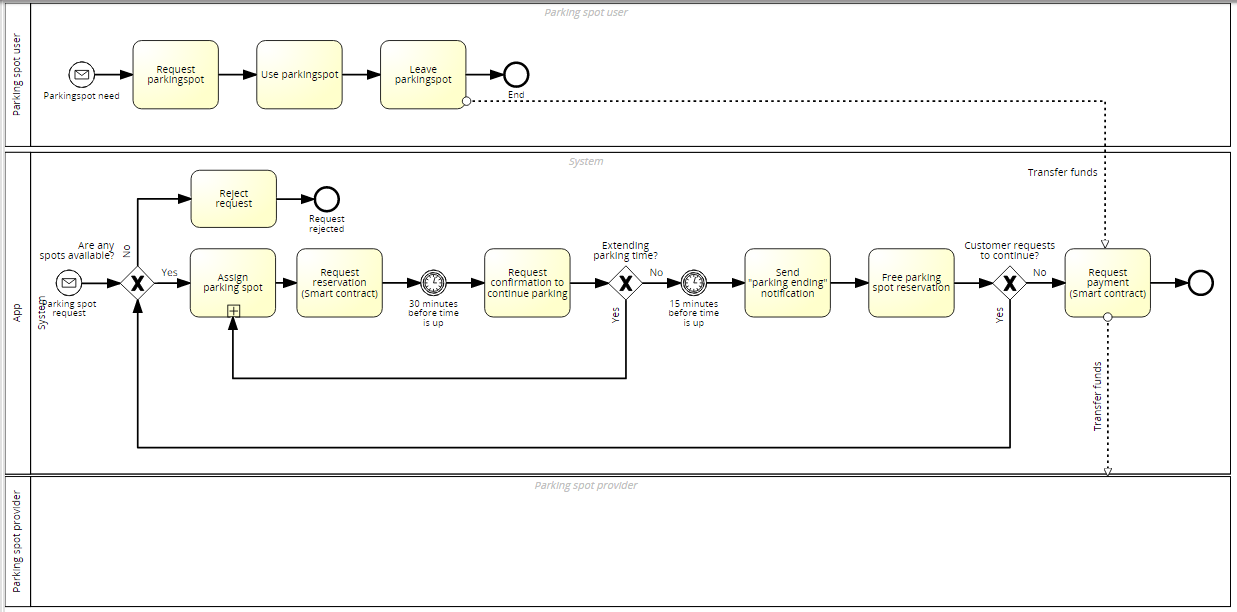


Figure 4: BPMN of paying for parking spot after using

## Reserving a parking spot

This is a use case that is not available in conventional parking spot business models. This gives the client the ability to reserve parking spot, let’s say for period of meeting, so he can be sure that upon arrival, the spot is free.

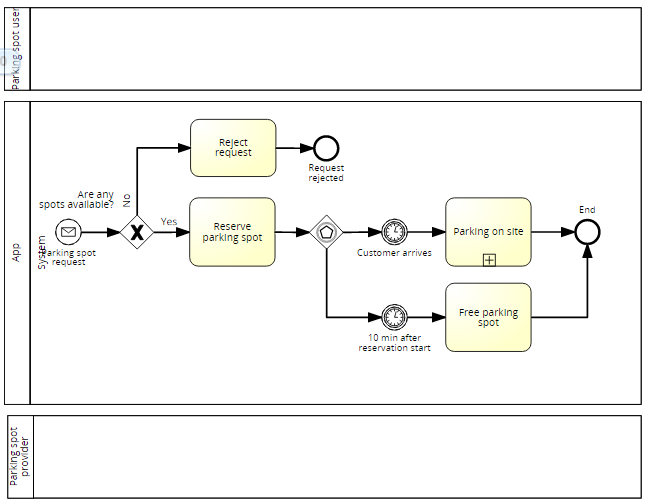


Figure 5: BPMN of parking spot reservation

## Software architecture

The central implementation will be a smart contract framework Hyperledger Fabric. It will guarantee that transactions are accepted by all parties, are immutable after transaction has ended and allows access for interested parties, so the business transaction is transparent.

The parking spot owner will use a web app to manage parking spot and cash flow.

The parking spot renter will use a smartphone app to find and rent parking spots.

## Implementation

Implementation will start by defining user stories and use cases. Use cases will be transformed into functional tests and implementation code. In the end, this project will make code and underlying system available, using Docker platform.

*This is TODO, since the business mode and the architecture is not fully set.*

## Smart contract for parking spot

Smart contract is piece of software that runs the transactions, makes sure that the transactions is valid and calculates end results. This will make it the most valuable piece of application and bugs may have expensive outcomes.

Smart contract must allow:

* Parking spot owner, to define
  1. a fee per hour, for the parking spot
  2. time for parking spot availability
  3. a fee for canceling the reservation
* Parking spot renter
  1. to search a parking spot in given location and time
  2. to make sure that parking spot is not double used (somebody has already using it)
  3. ability to cancel reservation and make sure only cancelation fee is taken

Simple smart contract for renting out a parking spot with predetermined time/monetary value and cancelation fee:

*//buy for N minutes***contract ParkingContract(**

**assetRequested: Asset, *//EUR* amountRequested: Amount, *//predetermined per period (let’s say 450 cents for 45 min)* amountCollateral: Amount, *//Colleteral in case of cancel* parkingspotOwner: Program, *//Selver* cancelKey: PublicKey *//Buyer public key for canceling* ) locks parkingSpot** {  
  
 **clause payForParking()**

**requires payment: amountRequested of assetRequested** {  
 lock payment with parkingspotOwner *//pay seller* lock parkingSpot with parkingspotOwner*//give back to seller* }  
 **clause cancel(buyerSig: Signature)**

**requires payment: amountCollateral of assetRequested** {  
 *//verify amountCollateral > amountRequested* verify checkTxSig(cancelKey, buyerSig) *//check it is the buyer (owner of cancelKey)* lock payment with parkingspotOwner  
 lock parkingSpot with parkingspotOwner*//give back to seller* }  
}

# Conclusions

# References

|  |  |
| --- | --- |
| [1] | T. S. J. H. Juri Mattila, "Product-centric Information Management: A Case Study of a Shared Platform with Blockchain Technology," UC Berkeley, Minneapolis, 2016. |
| [2] | C. P. L. Z. V. G. A. P. Xiwei Xu, "The Blockchain as a Software Connector," 13th Working IEEE/IFIP Conference on Software Architecture, 2016 . |
| [3] | M. A. A. K. A. M. a. E. S. Kevin Delmolino1(B), "Step by Step Towards Creating a Safe Smart Contract: Lessons and Insights from a Cryptocurrency Lab," 2016. |
| [4] | R. M. G. M. B. Kewell1, "Distributed ledger technology: Applications and implications," John Wiley & Sons, Ltd, 2017. |
| [5] | "Hyperledger Fabric documentation," 2017. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release/blockchain.html. |
| [6] | "Hyperledger Fabric licence," 2017. [Online]. Available: https://github.com/hyperledger/fabric. |
| [7] | K. C. L. W. N. K. Daniele Magazzeni and Peter McBurney, "Validation and Verification of Smart Contracts: A Research Agenda," *COMPUTER ,* no. september, pp. 50-57, 2017. |

Appendix

1. Glossary

|  |  |
| --- | --- |
|  |  |
|  |  |

1. License

**Non-exclusive licence to reproduce thesis and make thesis public**

I, **Sven Mitt**,

(*author’s name*)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

of my thesis

**Blockchain application - case study on Hyperledger Fabric**,

*(title of thesis)*

supervised by      ,

*(supervisor’s name)*

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, **27.11.2017**