



smarts 

Mobile based self-service
platform powered by crypto

Yellowpaper

Smarts OÜ
Tartu Maantee 53, 10115 Tallinn, Estonia
info@smarts.store

Table of Contents

Table of Contents	1
1. Smarts Summary	2
1.1 Overview of self-service strategies.....	2
1.2 The future of cryptocurrencies.....	2
1.3 What is Smarts.....	3
2. Technical overview	3
2.1 Smarts Core.....	4
2.1.1 Architecture.....	4
2.1.2 Payments and bills.....	6
2.1.3 Scanning.....	7
2.2 Security.....	7
2.2.1 Security of the system.....	8
2.2.2 Security of the purchase.....	8
2.2.3 Security of the payment.....	8
2.3 Speed.....	9
2.4 Scalability.....	10
3. Managing, monitoring and developing the system	10
4. Testing the system	11
Economic view	11
Smarts OÜ share and token offering S/TO	12-14
The future of Smarts	14
Bibliography	15-16

1.Smarts Summary

1.1 Overview of self-service strategies

The aim of self-service is to reduce the amount of human resources needed for providing the service and put a partial responsibility on the customer. Another main purpose for implementing self-service is cost savings. In addition, providing self-service opportunities is innovative and attractive for the customer.

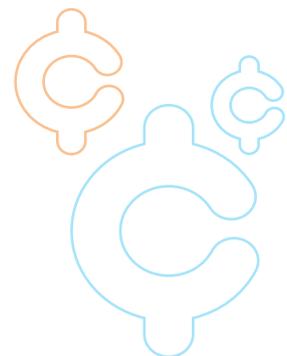
The most common self-service methods are terminal based self-service, web-based self-service and mobile-based self-service. In 2005 a study with 654 participants was done to evaluate those three self-service strategies. Among other things the participants were asked about ease of use, usefulness, the need for communication and risk. The results of the study provided a fundamental difference between the three self-services. The most useful self-services were terminal and mobile based. Terminal-based self-service was the easiest to use, which was the main point of the study. The riskiest was internet banking, but considering that in 2005 web-based self-service was still a really new thing, the participants of the study did not yet have a firm opinion of it.

Smarts tries to develop it's system on three main features, which are ease of use, availability and profitability.

1.2 The future of Cryptocurrencies

The past decade has shown a huge increase of demand in the world of cashless and digital payments. From 2014 to 2015 the global growth volumes grew over 11%, reaching 433 billion transactions. Emerging Asia being the biggest driving force, with a growth of 43.4%.

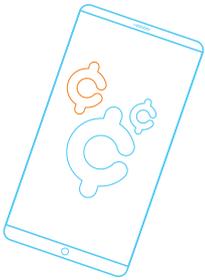
The volume of mobile contactless payment is projected to grow by a compound annual growth rate (CAGR) of 80% between 2015 and 2020 to 4 Trillion USD. It mainly comes thanks to different mobile payment solutions such as Google pay, Alipay, Apple pay and many others.



The crypto-economy and cryptocurrencies are also at a remarkable growth. Estimations show, that by 2020 the crypto-economy will pass three trillion dollars and by 2022 equal 10% of world GDP.

But how do you connect all this into the everyday world and the lives of everyday people? This is where Smarts comes in.

1.3 What is Smarts



Bringing cryptocurrencies into the lives of everyday people does not happen until there is a simple and quick way to use cryptocurrencies the same way that we use traditional currencies. Smarts is here to provide a solution for that - a mobile self-service platform.

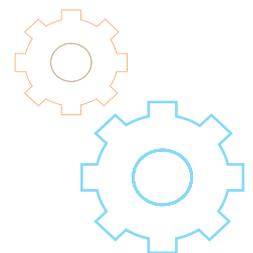
A system that allows for a customer to walk into a store, grab the product they wish to purchase, scan the product's barcode they want to buy and make the payment by using their smartphone. And for payment methods you can use either traditional or cryptocurrencies.

What that means is that we can bring the benefits of cryptocurrencies - speed of payment; transaction costs and removal of national borders into the hands of everyone who has a smartphone with them, whilst also removing one of the biggest hassles of everyday shopping - checkouts.

2. Technical overview

This paragraph gives a technical overview of Smarts. The technical part is based on two previously mentioned master's thesis works, one of which was written by the author of this article.

The development and research of this system is based on the methodology of Design Science Research. The systems are developed on the principles of agile software development methods.



2.1 Smarts Core

2.1.1 Architecture

For the creation of this platform it was necessary to find the best solution with which to deal with great volumes of data, was fast enough and easily scalable, for when data volumes get bigger. Then we studied service-based architectures such as microservices and SOA/ESB (Service Oriented Architecture/Enterprise Service Bus). What we discovered was that the architecture of microservices is more easily scalable (especially horizontally) than SOA architecture. Also, SOA is mainly used on enterprise level solutions, when microservices are rather used for project-based approaches. SOA problems were also that the integration component might get too complicated and it's architecture has a risk of turning monolithic.

Based on those facts, Smarts system was built on the microservice principal. Smarts core was first divided into 10 different microservices. With every development there will be additional microservices. The table below consists of the first microservices.

Smarts Gateway service

Gateway through which clients communicate with the Smarts system.

All the services are in the intranet and unavailable directly from the external network.

Smarts Service Registry Service

Service registry for retaining the service locations and work as a load balancer.

Smarts Auth Service

Authorization based on OAuth2.0 protocol. As the name states, this service is responsible for authorization.

Smarts Subscription Service

Through the subscription service the stores can connect themselves into the system.

Smarts Communication Service

The purpose of the communication service is to communicate with external systems and to locate product data.

Smarts Payment-and-Bills Service

The service of payment and bills authorizes and supervises payments.

Smarts Configuration Service

Service that administers all of the configuration.

Smarts QA Service

Quality assurance tests the quality of connected systems during the connection and later on with periodic testing.

Smarts Monitor Service

Smarts monitoring service monitors the the running of all services. When an error occurs, this service makes sure that the error gets announced and corrected as fast as possible.

Smarts Deploy Service

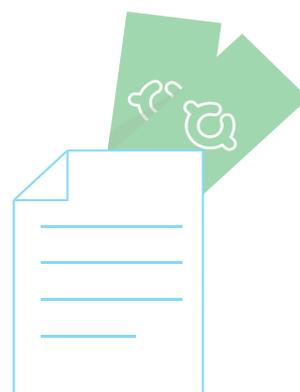
Continuous integration service with which we can deploy a new release with zero downtime.

2.1.2 Payments and bills

Smarts treats all bills on the e-invoice standard, which is very minimally customized to allow the introduction of cryptocurrencies as a method of payment. Smarts issues bills to the stores on the e-invoice standard.

Smarts will support cryptocurrencies as well as traditional internet-based card-payment methods. The first supported cryptocurrencies will be Ethereum, EOS and Bitcoin for which will be created smart contracts, if possible, and the transactions will take place through the smart contracts.

The service provider will get two options on how they wish to receive their payment. If the provider supports crypto payments, then Smarts will transfer the fee in cryptocurrencies. But if they are unable to accept crypto payments, Smarts will use an external exchange to convert the cryptocurrency into traditional currency.



2.1.3 Scanning

The main activity of Smarts' users is scanning codes. Smarts uses scanning to enter to the service provider (QR Code), to purchase (one- and two-dimensional barcodes) and to exit from the service provider and pay for the purchases. On the user of the self-service platform (UOP) end, scanning will be used by the purchase inspector.



We compared and tested the efficiency of different mobile-based one- and two-dimensional scanning algorithms. The best results were provided by a third-party, Scandit's, algorithms, which were taken into use.

2.2 Security

Smarts divides security into three groups:

1. Security of the system
2. Security of the purchase
3. Security of the payment.



2.2.1 Security of the system

The system uses OAuth2.0 protocol for authentication. For bigger purchases, customers must go through the KYC process. Customers' sensitive info will be stored in secure databases with limited third-party and external access. Only the ip-addresses of the Operation team members will be whitelisted. Smarts doesn't have permission to share the private info of it's customers.

Smarts will create a secure zone for the intranet services, that doesn't have access from the external network or by unauthorized users. We are using a gateway solution to provide allow communication between external and internal services.

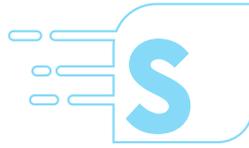
2.2.2 Security of the purchase

UOP has many options in the admin panel, to set up different methods of purchase control. Smarts offers a variety of methods that are configurable daily or hourly. For example you can check every customer, every X-th customer, or by random. Along with the product information, the UOP can send along additional safety strategies for example, making sure a person buying alcohol is of legal age.

2.2.3 Security of the payment

For the secure management of crypto payments, Smarts uses smart contracts. End users' crypto wallets will be generated inside their smartphone but Smarts won't have access to them. The transaction will be initiated by the user and smarts will mediate the transaction into the blockchain. The signing of the transaction will be done inside the user's phone.





2.3 Speed

Smarts has to be as fast as possible to provide the best shopping experience for the end-user. To make that happen, instead of using relational databases Smarts uses NoSQL database engines. We compared potential NoSQL database read and write speeds and found that right now the most optimal for this system is to use Couchbase NoSQL database engine.

Smarts caches product information, which does not contain any client-based discounts (from memberships etc). The stores can manage it's products and also update, empty and add new ones to the cache through the SDK (Software Development Kit). If Smarts finds that a scanned product is not in the in the cache and no discounts or markups are applied then Smarts Core will cache the product automatically. With that, Smarts reduces the amount of queries needed to be made at the UOP node server. To get the products from the cache, Smarts uses algorithms and methods optimized for search.



Active bills are held in the cache. Smarts uses Redis cache. If the bill is paid or has expired, Smarts will automatically delete it from the cache.

We compared different one- and two-dimensional image-based scanning algorithms and found, that not only by fault tolerance and accuracy but also by the scanning speed the Scandit algorithms were the best for scanning the product's barcode.

Communications between different services are done by message brokers. From our tests and comparisons we found out, that the fastest message broker for our system was Kafka, but we chose to use RabbitMQ. Even though if it is a little slower than Kafka, it allows for a better managing of routing logics.

2.4 Scalability

Smarts Core uses a microservice architecture which allows for a better horizontal scanning of the system. Smarts places multiple instances across regions, so the queries from different countries are received at the instance closest to them. Database capabilities are made better by applying master-slave pattern and using a cluster of replicated databases of which some are for writing processes and the others are for reading processes.



3. Managing, monitoring and developing the system

To manage and monitor the system, smarts creates different microservices. Service registry monitors the status and conditions of the service machines. Monitoring the traffic is done in the gateways. To manage logs we will use elastic search with the Kibana interface.

Systems microservices are hosted and managed with the help of Docker. In addition, we will also use Kubernetes.

To deploy new releases we are using means of continuous integration. For project management, documentation and storing the code we use Jira, Bitbucket and Confluence.

4. Testing the system

Before publishing a new release, Smarts system is tested thoroughly which includes automatic testing, stress testing, security testing, regression testing, testing the black box; white box and smoke testing. In the development of business-critical components we use TDD (test drive development) developing method. If an error gets through the testing phase, it will be top priority and gets fixed as soon as possible.



Economic view

Service fees are collected on every purchase which will be added to the bill of the end-user. If the service user is not prepared to accept crypto currency, Smarts converts it into traditional currency, brokering fee will be added. With the brokering fee Smarts is able to cover the risks of crypto currency fluctuation.

5. Smarts OÜ Share and token offering (S/TO)

Smarts is on the opinion that those who provide us with their help and trust should also see a deserved return on it. That is why we conduct our Utility Token sale in a way that gives our investors the opportunity of acquiring our company shares in complete agreement with the law.

5.1 Overview of the sale

Smarts OÜ, the company behind Smarts Self-Service Platform will be split into class A and class B shares - 75% and 25% respectively. During our S/TO the 25% of Smarts OÜ will be made available to our customers during the sale.

The shares will be sold to our customers who have purchased our Smarts utility token (SMT) as an advance payment for future services. As our customers, the SMT holders have the option to get a percentage of Smarts if they so wish. The B-shares on offer carry no obligations but they do have prior rights to dividends (profits will first be shared with B shareholders and only after then with the A shareholders) with Smarts obligated by law to pay the dividends to the holders of the B-shares when the company becomes profitable. All investors must go through a Know Your Customer (KYC) process before they receive their SMT and B-Shares.

The tokens on sale are ERC-20 Utility tokens, with the total amount 100 million SMT being divided between the two sales Tiers 25% and 50%. The Remaining 25% will be reserved for marketing, advisors, team and company reserve.

5.2 S/TO - Share and Token Offering

Smarts S/TO will take place in two Tiers. With Tier 1 being the presale and Tier 2 the second round of the crowdsale. During the presale, a total amount of 25 million SMT will be made available for purchase as well as 10% of Smarts OÜ in the shape of class B-shares. As per Estonian law only 800 investors can become the owners of the 10% during our presale, the minimum investment for a percentage of Smarts OÜ will be around 3000 (the final price will be in ETH and set at the launch date).

The Tier 2 sale will consist of 50 million SMT (plus the unsold tokens from the presale) and 15% of Smarts OÜ in the shape of class B shares. The details for the Tier 2, including the dates and pricing will be made available after the presale.

5.3 Presale walkthrough

The presale will commence no later than September 1st 2018 and close on the 1st of November or when 2.5 million EUR worth of SMT are sold. During the sale, Investors will have the opportunity to enter our Token Sale environment and register their account. The next step would be to follow the instructions in the Token Sale interface and purchase their Smarts Tokens. When the purchase is validated, the option to go through the KYC process is enabled. If the account holder has successfully passed the KYC they need to wait until the end of the presale before they receive their SMT and class B shares of Smarts OÜ.

If they fail to commence the KYC process their money will be considered as a donation. If they do not pass the KYC, their money will be returned.

When the presale ends, all investors who are eligible for a class B share of Smarts OÜ will be recorded as the shareholders of Smarts OÜ in the Estonian Business Registry by first and last names.

All of the Smarts OÜ class B shareholders will receive the dividends from any profits that Smarts OÜ will make and they have the prior rights to the dividends ahead of all the class A shareholders.

5.4 Know Your Customer (KYC)

Before receiving any SMT or class B shares of Smarts OÜ, the investor must go through a KYC process. The KYC will be provided to us by KYC Center OÜ (<https://kyc-center.com>). All the private information that the account holder provides to the KYC Center will be confidential, securely encrypted and Smarts OÜ will have no access to it. Smarts OÜ will only receive the info if the KYC of a certain investor has been successful or not. This process is needed to make the S/TO in compliance with Estonian law and to protect all the investors and their investments.



5.5 Conclusions

Smarts will be leading the way in offering a crypto-investment opportunity which is legal, safe and with great odds for ROI. The team of Smarts OÜ believes in what they are creating - a world-changing enterprise profitable to everyone involved which can only be achieved with full transparency and compliance with laws so to first and foremost protect the investors and to repay the faith they put into the Smarts Self-Service Platform.

The future of Smarts

Following the completion of the Smarts Core system, Smarts will start building a global network of self-service systems. We will try to involve as many interested services, stores and institutions as possible. In parallel to building that network, Smarts will start building an AI team, whose goal is to develop a hands-free shopping system. They will mainly concentrate on perfecting computer vision, which would elevate the confidence of computer-based decision making.



References

Curran, J. M., & Meuter, M. L. (2005). Self-service technology adoption: comparing three technologies. *Journal of Services Marketing*, 19 (2).

Cap Gemini World Payments Report 2017

IDC Financial Insights' 2014 Consumer Payments Survey



Deloitte estimations

Namiot, D., & Sneps-Sneppe, M. (2014). On Micro-services Architecture. *International Journal of Open Information Technology*, 2(9).

Villamizar, M., Garcés, O., Castro, H., Verano, M., Salamanca, L., Casallas, R., & Gil, S. (2015). Evaluating the Monolithic and the Microservice Architecture Pattern to Deploy Web Applications in the Cloud. Colombia: Systems and Computing Engineering Department.

Kristo Truu. Mobile based self-service platform 2018 (Tallinn University Of Technology, Informatics, Msc.)

Estonian Banking Association "Guidelines for Description of Estonian Electronic Invoice. Sending e-invoices to the bank and presentment of e-invoices at the bank".

Cerny, T., Donahoo, M. J., & Trnka, D. (2017). Contextual Understanding of Microservice Architecture: Current and Future Directions. *APPLIED COMPUTING REVIEW*, 17 (4).

Curran, J. M., & Meuter, M. L. (2005). Self-service technology adoption: comparing three technologies. *Journal of Services Marketing*, 19 (2).

Namiot, D., & Sneps-Sneppe, M. (2014). On Micro-services Architecture. *International Journal of Open Information Technology*, 2(9).

Villamizar, M., Garcés, O., Castro, H., Verano, M., Salamanca, L., Casallas, R., & Gil, S. (2015). *Evaluating the Monolithic and the Microservice Architecture Pattern to Deploy Web Applications in the Cloud*. Colombia: Systems and Computing Engineering Department.

Kristo Truu. *Mobile based self-service platform 2018* (Tallinn University Of Technology, Informatics, Msc.)

Hans-Toomas Saarest *Design, analyze and development of mobile base self-service integration components 2018* (Tallinn University Of Technology, Informatics, Msc.)