HOGESCHOOL ROTTERDAM / CMI

**Project 4**

**Inrichting centrale bank**

**Periode 4**

Stan Boer

0931006

Vakcode: TIRPRJ014

Modulebeheerders:

P. Mazereeuw

R. van Doorn

G. Maas

Periode: 4

Inleverdatum: 30 mei 2018

Inlevermoment: Eerste gelegenheid

Inhoudsopgave

[**1. Inleiding**](#_baz5eq9xjoey) **2**

[**2. Beheer**](#_njuuv88mt8e8) **3**

[2.1 Git](#_bhmoywjtdn3x) 3

[2.2 Risicolog](#_g63ebfy94k6e) 3

[2.3 Issue tracking](#_t1nnugm3fwk) 3

[**3. Analyseren**](#_u37cimrzrt16) **4**

[3.1 Security](#_i3so81jeq70f) 4

[3.1.1 Directe verbinding](#_k5xqivxzk46q) 4

[3.1.2 TLS](#_gbbb4su4apg0) 4

[3.2 Overdraagbaarheid](#_ox3a7kfjysdw) 4

[3.3 Vertrouwen](#_cuprbz3l8vmq) 5

[3.4 Efficiency](#_7160wm4x8q5z) 5

[3.5 Functionele eisen](#_2gflip1fdnth) 5

[3.6 Niet-functionele eisen](#_woalo4p3svs6) 5

[**4. Ontwerpen**](#_s4mgt4fdls3) **6**

[4.1 Netwerk Diagram](#_lzs4w5jc2bv8) 6

[4.2 Dataflow Diagram](#_ws6uil3m8ndg) 6

[**5. Alternatieven**](#_vem4op5r8kvu) **7**

[**6. Advies**](#_svj93cwin0df) **7**

[**7. Bronnen**](#_gqt5l167g9b5) **8**

# 1. Inleiding

Om een goed beeld te krijgen van hoe de centrale bank zal moeten functioneren, wordt er voor dit onderdeel een adviesrapport beschreven. In dit adviesrapport zal een advies gegeven worden voor de vraag: ‘*Hoe richt ik op een efficiënte manier de centrale bank in?*’ In het rapport gaat het eerst over het beheer van het project, daarna worden de kwaliteitseisen van het eindproduct beschreven en hoe deze geïmplementeerd moeten worden. Er zal een kort gedeelte verteld worden over de eisen voor het gebruik van het product, waarbij tot slot een overweging gemaakt wordt tussen de alternatieven en er een conclusie getrokken wordt in de vorm van een advies.

# 2. Beheer

## 2.1 Git

De documenten en code met betrekking tot het project staan ten alle tijden op GitHub. Om naar de repository te gaan, klik hier: <https://github.com/Dragonstei/Project4Her>.

## 2.2 Risicolog

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Risico Beschrijving** | **Kans** | **Impact** | **Risico** | **Maatregel** | **😶** | **Status Omschrijving** | **Datum** |
| R1 | Stroeve communicatie begin project, laat bijeengekomen voor plan individuele bijdrage groepsdeel | 4 | 3 | 12 | Beter communiceren over bijeenkomsten plannen | :) | Probleem opgelost door betere afspraken te maken over communicatie | 09-05-2018 |
| R2 | Afwezigheid projectlid | 2 | 2 | 4 | Goede communicatie met betrekking tot afspraken en werk overnemen | :) | Geen probleem bij de groep, communicatie verliep goed | 16-05-2018 |

## 2.3 Issue tracking

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Datum In** | **Issue** | **Verantwoordelijk** | **😶** | **Status Log** | |
| **Datum** | **Beschrijving** |
| J1 | 02-05-2018 | Samenkomst projectgroep | Groepsleden met betrekking tot communicatie | :) | 02-05-2018 | Nog niet met projectgroep kunnen afspreken door andere vakken en herkansing project |
| 08-05-2018 | Na vakantie samengekomen om plan individuele bijdrage te maken |

😶 = Status: :) = opgelost, :| = bezig; :( = niet opgelost, N = nieuw

# 3. Analyseren

## 3.1 Security

Om ervoor te zorgen dat er een verbinding tot stand gebracht wordt tussen clients en servers, is het nodig om de verbinding te beveiligen. Dit om ervoor te zorgen dat de gegevens van gebruikers gewaarborgd blijven. Hieronder worden een paar mogelijkheden uitgelegd.

### 3.1.1 Directe verbinding

De client zal vanaf de client direct verbonden zijn met de servers. Hierdoor wordt er zonder enige versleuteling een verbinding gemaakt met andere banken. Alle data van de databases zal beschikbaar zijn en de gegevens om verbinding te maken staan in de code van de client geschreven.

### 3.1.2 TLS

Zodra er gepind wordt bij een pinautomaat, zal er een JSON request met de benodigde data (bankcode en saldo opzoeken) opgestuurd worden naar de lokale server. Als er een verbinding tot stand gebracht wordt over TLS, zal de server een JSON response terug sturen met de bankcode en met het saldo van de rekening. Hierna zal bij een transactie nogmaals een JSON gestuurd worden met het opgevraagde bedrag, wat hierna verwerkt zal worden in de database op de server van de bank en het bedrag terug sturen.

De JSON request wordt via TLS verzonden. De lokale server zal eerst proberen om verbinding te maken met de centrale server door zijn eigen certificaat en key te sturen. Hij zal vragen om de certificaat van de andere client. Als die certificaten niet gevonden, dan zal de verbinding verbroken worden. Als er gereageerd wordt en er een certificaat met key teruggestuurd wordt, dan zal en een verbinding tot stand worden gebracht. Om in een veilige connectie status te komen, veranderen zowel de client als server van cipher. Hierna zullen ze deze shared/session key gebruiken totdat de verbinding verbroken wordt.

De gebruiker zal ingelogd blijven totdat hij/zij klaar is met het gebruik van een pinautomaat, of zodra een timeout is bereikt.

## 3.2 Overdraagbaarheid

Om ervoor te zorgen dat het systeem niet beperkt blijft tot banken die vanaf het begin af aan zijn aangesloten op de centrale server, is het belangrijk om ervoor te zorgen dat het gebruikte systeem overdraagbaar is. Het moet dus zo zijn dat het systeem makkelijk te implementeren is bij andere banken, zodat ze zonder teveel moeite aangesloten kunnen worden op de centrale server.

Om gebruik te maken van TLS met JSON requests en responses, is het nodig benodigd om gebruik te maken van de REST API. Deze API zorgt voor de communicatie tussen de client en server. Hierdoor is het makkelijk om in een standaard format de requests te sturen, zodat alleen de benodigde opgevraagd hoeft te worden. Als nieuwe banken dezelfde structuur aanhouden, zal het moeiteloos toepasbaar zijn om hiervan gebruik te maken.

## 3.3 Vertrouwen

Om ervoor te zorgen dat klanten hun vertrouwen behouden in het gebruik van pinautomaten, is het belangrijk dat alle gegevens gewaarborgd blijven. Er moet dus voor gezorgd worden voor een zo’n veilig mogelijke verbinding, waardoor er geen ongewenste personen bij belangrijke gegevens kunnen komen.

Om geen belangrijke informatie door te sturen tijdens het pinnen, zal er bij de JSON requests alleen gevraagd worden naar het saldo van de gebruiker.

## 3.4 Efficiency

Voor een snelle transactie, is het nodig dat het gebruik van de pinautomaat soepel en snel verloopt. Er moet dus zo weinig mogelijke communicatie gebruikt worden om een transactie te voltooien. Daarom wordt er eerst gekeken of de gebruiker van de lokale bank is, zodat er niet onnodig een verbinding gemaakt hoeft te worden met de centrale server. Bij de verbinding met de centrale server wordt er eenmalig een verbinding uitgevoerd bij een transactie. Het is ook mogelijk om de verbinding eerder te verbreken indien nodig, zodat de verbinding niet onnodig lang tot stand blijft.

## 3.5 Functionele eisen

Om ervoor te zorgen dat het eindproduct functioneert zoals verwacht bij een pinautomaat, zijn er een aantal functionele eisen waaraan het eindproduct moet doen:

* De geldautomaat moet biljetten van tenminste vier verschillende waarden uit kunnen geven
* De gebruiker kan niet, zonder een pin-opdracht te geven, geld uit de automaat halen
* De geldautomaat geeft altijd het juiste bedrag
* De geldautomaat geeft alleen geld als het saldo toereikend is
* De gebruiker kan zelf selecteren welke biljetten hij/zij wil ontvangen
* De gebruiker kan geen biljetten kiezen die niet aanwezig zijn in de geldautomaat
* De geldautomaat is robuust (kan zelfstandig staan en valt niet om/uit elkaar tijdens gebruik)
* De biljetten in de geldautomaat mogen maximaal de dikte van een speelkaart hebben
* Na het pinnen wordt er een bon geprint met een bonnenprinter. Op deze bon staat in

ieder geval hoeveel geld er is opgenomen en bij welke (lokale of individuele) bank dit is gebeurd

## 3.6 Niet-functionele eisen

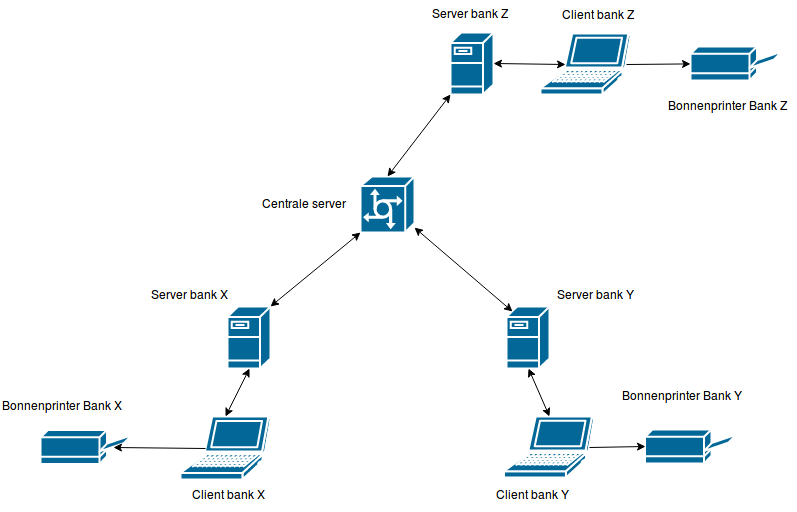
Voor de niet-functionele eisen geldt dat de communicatie tussen de clients en servers secure moeten zijn. Dit wordt gedaan door middel van het TLS protocol. De geschreven code moet ten minste voldoen aan de 3 volgende eisen, om ervoor te zorgen dat de code leesbaar en efficiënt is:

* DRY (Don’t Repeat Yourself)
* KISS (Keep It Simple, Stupid!)
* YAGNI (You Aren’t Gonna Need It)

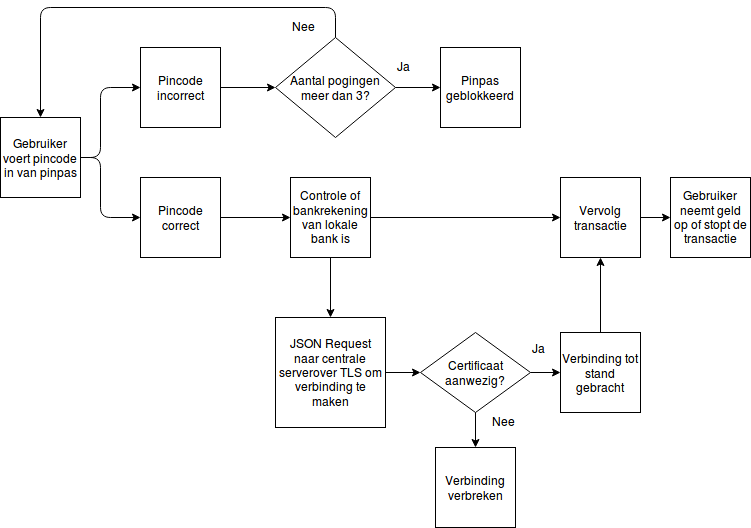
De JSON bestanden moeten gebruik maken van een bankcode, om de servers van andere banken te herkennen. Deze wordt meegegeven in de JSON requests.

# 4. Ontwerpen

## 4.1 Netwerk Diagram



## 4.2 Dataflow Diagram



# 5. Alternatieven

Om te verbinden met de server kan er gebruikt gemaakt worden van een directe verbinding. Dit houdt in dat alle data van de andere banken volledig beschikbaar zijn tot de client. Elk persoon zou zonder enige moeite kunnen verbinden met de andere banken. Dit is zeer onveilig, omdat belangrijke gegevens van klanten openbaar zullen zijn. Dit is een grote schending van privacy en zal ervoor zorgen dat er geen beveiliging aanwezig is bij de banken.

# 6. Advies

Er wordt aangeraden om gebruik maken van JSON requests in combinatie met TLS. Er wordt eerst gekeken of de pinpas van de lokale bank is. Dit wordt gedaan door te kijken of de bankcode overeenkomt met de bankcode van de lokale bank. Als dit niet het geval is, zal er een JSON request verstuurd worden naar de centrale server. De request komt binnen op de centrale server, waarna er nogmaals gekeken zal worden naar de bankcode. Als de bankcode overeenkomt met een van de servers van andere banken, zal de centrale server verbinding proberen te maken met die bank via TLS. Hierna zal er een verbinding tot stand komen, totdat deze verbroken wordt.

Omdat de REST API zeer makkelijk overdraagbaar is, is het aan te raden om hier gebruik van te maken. Het gebruiken van een vaste structuur van JSON bestanden en databases zorgt er voor dat het weinig moeite kost voor andere banken om zich aan te sluiten op de centrale server.

# 

# 

# 

# 

# 

# 

# 

# 7. Bronnen

Prodromou, A. TLS/SSL Explained – Establishing a TLS Connection, Part 5. Geraadpleegd van <https://www.acunetix.com/blog/articles/establishing-tls-ssl-connection-part-5/>

[TLS JSON Connection]. Geraadpleegd van <https://www.npmjs.com/package/tls-json>

REST API with JSON. Geraadpleegd van <https://technologyconversations.com/2014/08/12/rest-api-with-json/>

S Miguel, J. KISS, YAGNI & DRY, 3 Principles to Simplify Your Life as a Developer. Geraadpleegd van <https://itexico.com/blog/bid/99765/software-development-kiss-yagni-dry-3-principles-to-simplify-your-life>