|  |
| --- |
|  |
| Rapport centrale bank |
| Project 4 |

|  |
| --- |
| Joey de Brabander (0955123)  18-5-2018 |

Inhoud

[Scope van het project 2](#_Toc515285734)

[Beheren 3](#_Toc515285735)

[Risicolog 3](#_Toc515285736)

[Issue tracking 3](#_Toc515285737)

[Analyseren 3](#_Toc515285738)

[Kwaliteitseisen 3](#_Toc515285739)

[Niet functionele eisen 4](#_Toc515285740)

[Oplossingen 5](#_Toc515285741)

[Advies 7](#_Toc515285742)

[Ontwerp 7](#_Toc515285743)

# Scope van het project

Project 4 is een vervolg op project 3. Waar je in project 3 een interface en de software van een compleet werkende pinautomaat hebt opgeleverd, gaan je in dit project aan de slag met communicatie tussen verschillende groepsbanken. Je werkt samen met de andere bytegroepen om tot een systeem te komen waarbij iedereen ook bij elke andere groep kan pinnen. Je moet daarbij rekening houden met de compatibiliteit van de eigen functionaliteiten zoals het invoeren van incorrecte pincodes, blokkades van passen, ophalen van saldo, etc. en die van anderen. Iedereen maakt hiervoor een advies wat we uiteindelijk ook gaan maken.

# Beheren

## Risicolog

Kans: schaal 1 (klein) t/m 5 (zeer groot)  
Impact: schaal 1 (zeer lage) t/m 5 (zeer hoge)  
Risico: = kans \* impact  
☺: [status] ☺ opgelost, 😐 bezig; ☹ niet opgelost; N nieuw

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | Risicobeschrijving | Kans | Impact | Risico\* | Maatregel | ☺ | Status Omschrijving | Datum |
| R1 | De school brandt af | 1 | 5 | 5 | Niet met vuur spelen | N | Net neergezet, nog geen maatregel voor genomen. Wel brandblussers gezien | 25-4-18 |
| R2 | Tijdnood | 3 | 3 | 9 | Planning maken en op tijd beginnen | ☺ | Ik heb een planning voor mezelf gemaakt voor de komende 2 weken. | 18-5-18 |
| R3 | 2e waternoodsramp | 1 | 5 | 5 | Backup maken | N | Net neergezet | 18-5-18 |

## Issue tracking

☺: [status] ☺ opgelost, 😐 bezig; ☹ niet opgelost; N nieuw

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | Datum In | Issue | Verantwoordelijk) | ☺ | Status Log | |
| Datum | Beschrijving |
| J1 | 16-05-2018 | Kiezen van manier van presentatie. | Joey de Brabander | ☺ | 16-05-2018 | Twijfelde eerst tussen een powerpoint of een rapport maar heb voor een rapport gekozen omdat je daar makkelijker meer informatie in kwijt kunt. |
| J2 | 16-05-2018 | Weet niet goed wat de opdracht precies inhoudt. | Joey de Brabander | ☺ | 16-05-2018 | Na alle bijlages lezen en een aantal dingen vragen aan andere studiegenoten is me duidelijk geworden wat de individuele opdracht precies is. |

# Analyseren

## Kwaliteitseisen

**Security**  
De beveiliging bij een centrale bank is uiteraard hartstikke belangrijk. Je wilt niet dat de gegevens van de klanten op ook maar 1 of andere manier te verkrijgen zijn door mensen die daar geen permissie toe hebben. Ook wil je niet dat transacties, inlogpogingen en andere vormen van aanvragen tussen de klant en de bank afgeluisterd kunnen worden. Om alle aanvragen en communicatie te beveiligen kan gebruik gemaakt worden van:

* VPN (Virtual Private Network): Met een VPN wordt al het verkeer door een tunnel verstuurt. Dus tijdens het versturen kan het niet afgeluisterd worden. Het probleem met een VPN is dat ergens die informatie de tunnel in gaat en weer uitkomt. Op die punten is de informatie niet beveiligd.
* SSL (Secure Sockets Layer): SSL is een protocol die de communicatie tussen computers/servers beveiligd door gebruik te maken van encryptie. Als informatie geëncrypt wordt, wordt de informatie omgezet naar onleesbare code. Alle servers bevatten een soort sleutel bestand. Dit sleutel bestand wordt gebruikt om de informatie te encrypten. Om de onleesbare code vervolgens weer leesbaar te maken en dus te decrypten kan de andere server ook gebruik maken van het sleutel bestand. Wanneer iemand dus de informatie onderschept maar het sleutelbestand niet heeft, dan is de informatie nutteloos omdat het onleesbaar is. Wanneer er dus een bytegroep bij komt hoeft de bytegroep alleen maar het sleutelbestand op de server te zetten zodat het kan communiceren met de centrale bank. **(Secure Sockets Layer (SSL))**
* IP Whitelist: Elke server heeft een unieke IP adres. De centrale bank kan dus een zogenoemde whitelist bij houden van elk IP adres van alle servers van de bytegroepen waarmee die mag communiceren. De servers van de bytegroepen kunnen ook een whitelist gebruiken met alleen het IP van de server van de centrale bank. Wanneer iemand anders vanaf een andere server of computer met de centrale bank of een van de servers van de bytegroepen probeert te communiceren dan wordt daar niks mee gedaan. Het nadeel van een IP Whitelist is dat IP’s kunnen veranderen en wanneer een IP veranderd van een server van een bytegroep, kan die niet meer communiceren met de centrale bank. Het voordeel is wel dat het makkelijk is op te zetten en dat het snel is. Je zou een IP whitelist kunnen combineren met SSL.

**Uitbreidbaarheid en overdraagbaarheid**  
Het systeem moet makkelijk uit te breiden zijn. Dit kan gedaan worden door de servers op een handige manier in te delen en het systeem zo te bouwen dat er makkelijk nieuwe functies geïmplementeerd kunnen worden in de toekomst. Wanneer het systeem op een handige en een niet te complexe manier is opgebouwd is die ook automatisch gelijk goed overdraagbaar.

**Vertrouwen**  
Het is belangrijk dat het systeem vertrouwd aanvoelt bij gebruik. De klant moet niet het gevoel krijgen dat die afgeluisterd kan worden. Het product moet ook betrouwbaar zijn in het gebruik. Er mogen absoluut geen fouten zitten in de programmeercode waar de klant last van heeft.

**Bruikbaarheid**  
Uiteraard moet de bruikbaarheid van het systeem goed zijn voor de klant. Wanneer een klant van een andere bank probeert te pinnen moet dat niet voor problemen zorgen. De klant moet zonder voorkennis makkelijk gebruik kunnen maken van de bank.

## Niet functionele eisen

|  |  |
| --- | --- |
| Eis beschrijving | Prioriteit |
| Alle communicatie tussen servers moet geëncrypt worden. | Must |
| De code moet op een nette manier geschreven worden zodat er makkelijk nieuwe functies geïmplementeerd kunnen worden en zodat de overdraagbaarheid goed is. | Must |
| Het systeem moet makkelijk uitbreidbaar zijn. Er moeten op een eenvoudige manier nieuwe bytegroepen aan toegevoegd kunnen worden. | Should |
| De code op de Arduino’s zijn met C++ geschreven. De GUI’s en de connectie naar de databases zijn gedaan door middel van Java. | Could |

## Oplossingen

Elke bytegroep heeft zijn eigen groeps server. Elke groeps server heeft zijn eigen groeps database. Wanneer een klant probeert in te scannen bij een pinautomaat wordt eerst gecontroleerd of de klant bestaat in de database van de groeps server van de aangesloten pinautomaat. Wanneer dat het geval is zal de klant inloggen en is er geen communicatie nodig met andere banken. Wanneer dat niet het geval is wordt het id van de pas van de klant en de naam van de groeps server naar de server van de centrale bank gestuurd. De server van de centrale bank stuurt vervolgens naar alle andere groeps servers de aanvraag of het id van het pasje bestaat in de database van de desbetreffende groeps server. Als het id van het pasje in een database voorkomt zal de groeps server van deze database dat terug sturen naar de server van de centrale bank die het vervolgens weer terugstuurt naar de oorspronkelijke groeps server.

De centrale bank bevat ook een database waarin wordt bijgehouden welk pasje op welke server database staat. Wanneer iemand dus ergens zijn pasje scant die niet voorkomt in de database van de groeps server van de aangesloten pinautomaat en dus een request moet doen aan de server van de centrale bank, dan wordt eerst gecontroleerd of er al een soortgelijke request is gedaan met hetzelfde pasje. Wanneer dat zo is weet de centrale bank naar welke server de request toe moet in plaats van dat de centrale bank opnieuw naar elke server een request moet doen. Dit scheelt behoorlijk in tijd.

Om de servers met de centrale bank te laten communiceren kan gebruik worden gemaakt van verschillende protocollen. Ik heb de verschillende protocollen met voor- en nadelen omschreven:

**MQTT (Message Queuing Telemetry Transport)**  
MQTT is een betrouwbaar en energiezuinig protocol dat een server in staat stelt te communiceren met andere servers. Het werkt met een publish/subscribe systeem. Clients kunnen verbinding maken met een zogenoemde broker. Een client kan zowel publisher of subscriber worden. Een publisher stuurt berichten naar een broker en alle subscribers krijgen dit bericht vervolgens binnen. Een mogelijkheid die MQTT heeft is Quality of Service (QoS). Dit geeft je de mogelijkheid om aan te geven of iets belangrijks is of niet:

* QoS 0:

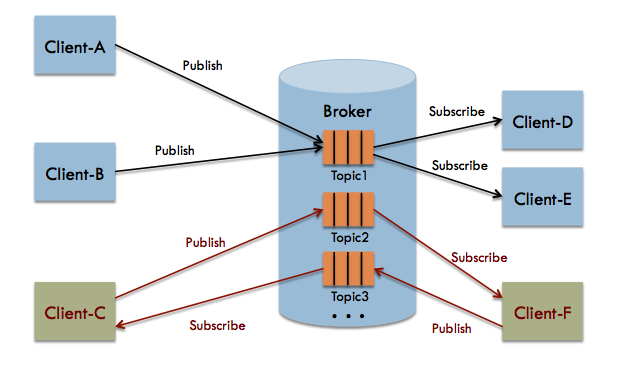
Quality of Service 0 hanteert ‘Fire & Forget’. Dat wil zeggen dat de publisher een bericht eenmalig verstuurt en er vervolgens niks meer mee doet.

* QoS 1:

Quality of Service 1 hanteert ‘At least once’. Dat wil zeggen dat de publisher een bericht blijft sturen tot de subscribers het ontvangen hebben. Het bericht kan meerdere malen gestuurt worden. Hierdoor wordt het bericht gegarandeerd minimaal 1x goed verstuurd.

* QoS 2:

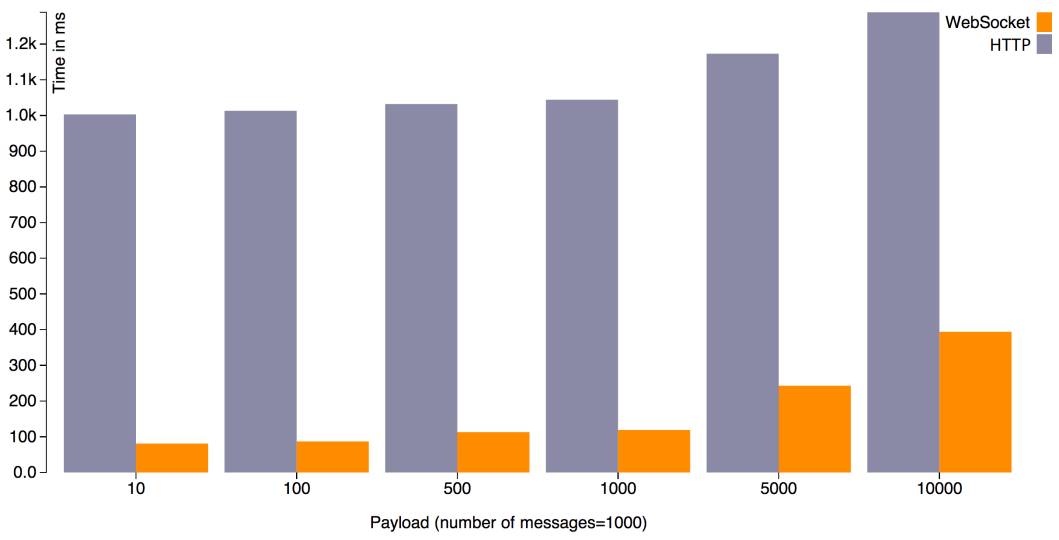
Quality of Service 2 hanteert ‘Exactly once’. Dat wil zeggen dat de publisher eerst kijkt of er een verbinding mogelijk is met de subscribers. Wanneer dat het geval is verstuurt die het bericht pas. Hierdoor wordt het bericht gegarandeerd gelezen en maar 1x verstuurd.

  
*Werking MQTT geïllustreerd.*

Het nadeel van MQTT is dat het niet veilig is. Je kan het veilig maken door de informatie d.m.v. SSL te encrypten. Het voordeel van dit protocol is dat het erg snel is en je ervan verzekerd kan zijn dat een bericht gegarandeerd 1x aankomt bij gebruik van QoS 2. Bij een transactie is dit bijvoorbeeld belangrijk. Als bank moet je erop kunnen vertrouwen dat een transactie eenmalig afgehandeld wordt. Nog een voordeel van MQTT is dat het erg redundant is. Zodra een subscriber of dus pinautomaat crasht, blijft de verbinding in stand. **(MQTT)**

**HTTP**  
Door gebruik te maken van het protocol genaamd HTTP kan ook gecommuniceerd worden tussen servers. Het is eenvoudig op te zetten. Het nadeel van dit protocol is dat iedereen mee kan luisteren en dat het heel erg langzaam is. Het meeluisteren kan echter verholpen worden door het bericht te encrypten zodat iedereen die meeluistert, er niks aan heeft. De server die een HTTP request uitvoerd, krijgt ook geen zekerheid dat de request goed is aangekomen. HTTP is ontworpen voor het wereldwijde web en is dus gelimiteerd. Nog een nadeel van HTTP is dat wanneer er wordt gecommuniceerd tussen de servers door middel van HTTP, er telkens een nieuwe verbinding wordt aangemaakt, wat dus een tijdrovend proces is. **(Communication Networks/HTTP Protocol)**

**WebSockets**  
WebSockets vertoont enige gelijkenis met HTTP. Servercommunicatie is echter een stuk sneller dan met HTTP. Wanneer servers met elkaar proberen te communiceren door middel van websockets, wordt er een vervinding aangemaakt. Deze verbinding blijft bestaan waardoor in het vervolg de servers makkelijk met elkaar kunnen communiceren zonder steeds opnieuw een nieuwe verbinding hoeven aan te maken in tegenstelling tot HTTP. Nog een voordeel van websockets tegenover het gebruik van HTTP is dat de berichten een stuk smaller zijn. Bij HTTP kunnen de berichten honderden bytes bevatten, bij websockets zijn dit er maar 2 tot 6. Het nadeel van websockets is wel dat het tijd kost om te configureren op elke server. Hieronder vind je een diagram die de snelheid van HTTP en websockets naast mekaar legt. **(WebSockets)**

  
*Snelheid WebSockets vs HTTP.*

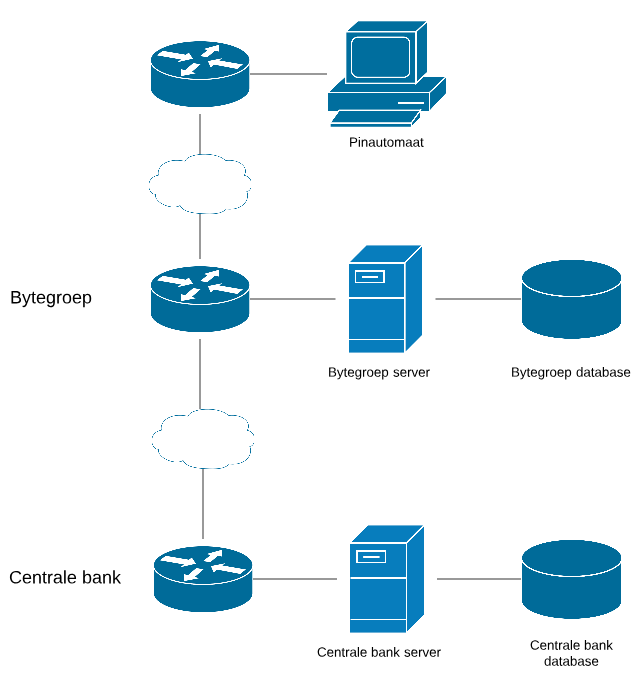
# Advies

Mijn advies is om als beveiliging SSL te gebruiken. Je zou er ook een IP whitelist bij kunnen gebruiken maar doordat we nog maar 3 weken hebben is dat niet aan te raden wegens tijdsgebrek. SSL op zichzelf is genoeg om er volledig zeker van te zijn dat er geen gegevens onderschept kunnen worden. Om de servers met elkaar te laten communiceren zou ik gebruik maken van MQTT. Niet alleen de mogelijkheid van Quality of Service is bij een banksysteem erg handig, ook is redundatie noodzakelijk. Je wilt altijd met zekerheid kunnen zeggen dat bijvoorbeeld een transactie altijd correct wordt uitgevoerd.

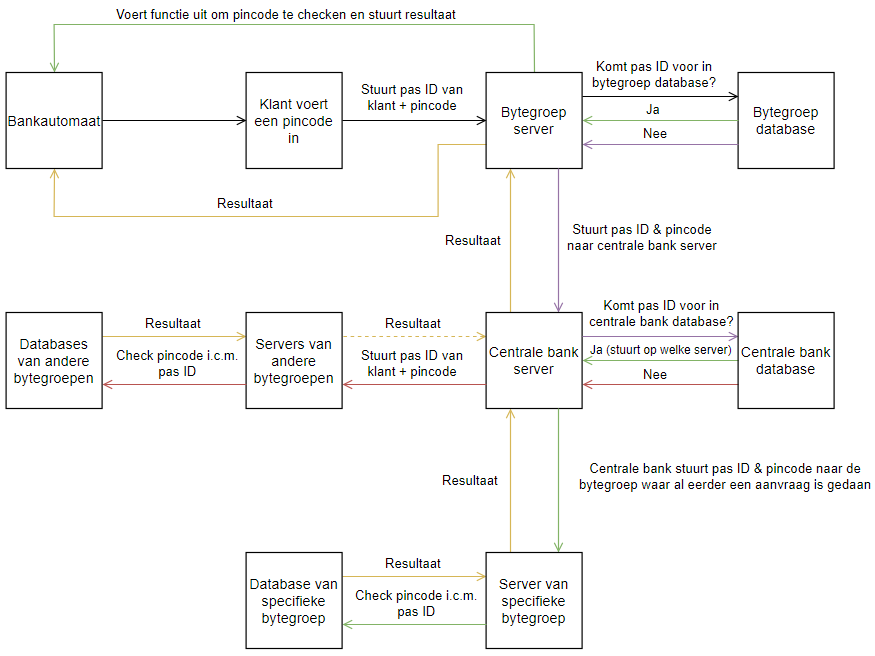
Ik heb niet gekozen voor HTTP omdat MQTT verreweg het snelst is. Het gebruik van websockets duurt te lang om te configureren met de gegeven tijd. Daarbij heb je bij HTTP of Websockets geen Quality of Service systeem, wat je bij MQTT wel hebt.

# Ontwerp

## Netwerk diagram



## Dataflow diagram



In het dataflow diagram hierboven wordt behandeld hoe een bytegroep server met een actie omgaat. Als voorbeeld heb ik de controle van een pincode laten zien. Door middel van gekleurde lijntjes kan je precies de dataflow volgen.

Wanneer het resultaat bij de gestipelde gele pijl aangeeft dat de pas ID bestaat in een server dan zet de centrale bank dit pas ID + de servernaam in zijn eigen database. Wanneer er in de toekomst nog een actie gebeurt met hetzelfde pas ID dan weet de centrale bank direct naar welke server die de aanvraag moet versturen in plaats van dat die de aanvraag naar elke server moet versturen.

# Bronnen

*Communication Networks/HTTP Protocol.* (n.d.). Retrieved mei 27, 2018, from Wikibooks: https://en.wikibooks.org/wiki/Communication\_Networks/HTTP\_Protocol

*MQTT.* (n.d.). Retrieved mei 27, 2018, from Infvo: https://infvo.nl/lab/index.php/Internet\_of\_Things/Protocollen/MQTT

*Secure Sockets Layer (SSL).* (n.d.). Retrieved mei 27, 2018, from Wikipedia: https://nl.wikipedia.org/wiki/Transport\_Layer\_Security

*WebSockets.* (n.d.). Retrieved mei 27, 2018, from   
Wikipedia: https://nl.wikipedia.org/wiki/WebSocket