Project 4
Inrichting Centrale Bank

Naam:	Wen Liu
Studentnummer:	0911282
School:	Hogeschool Rotterdam
Klas:	Ti 2A
Schooljaar:	2017-2018
Datum:	20 Juni 2018
Bytegroep:	11

Inhoudsopgave

1. Inleiding

In opdracht van mijn project docenten, moet ik een verslag schrijven voor mijn individueel deel van mijn project. Voor de individueel deel wordt er gekeken naar Beheren, Analyseren, Ontwerpen en Adviseren. In dit rapport komt dus 4 activiteiten te staan. Ook moet er 2 attributen per indicator uitgebreid uitgelegd. Deze zijn Security, Uitbreidbaarheid, Vertrouwen en tevredenheid. De documenten en code worden geplaatst op github, de link naar github wordt geplaatst onder beheren. Onderzoeksvraag luidt als volgt: *Hoe richt ik efficiënte manier de centrale bank in?*'

2. Beheren

De afgestudeerde is in staat om in een gegeven beroepssituatie het proces van ontwikkeling, ingebruikname en gebruik van ict-systemen beheersbaar te laten verlopen, rekening houdend met de context en relevante stakeholders. (Hogeschoolgids)

2.1 Github

Documentatie van elk student moet op Git staan (versiebeheer documenten + code).
Link naar mijn Github: https://github.com/HjimR/Project4Individueel

2.2 Issue Tracking

Groepsnaam: Saatbank

#	Datum	Issue	Verantwoordelijk	☺	Datum	Beschrijving
J1	**16-05-18**	**Nog niet helemaal een idee**	**Jimmy**	😐	**18-05-18**	**Idee binnen**
					19-05-18	**Bezig met ontwerp**

Tot nu toe niet veel issues (wat goed is)
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

Risico log

fx 16-05-2018

	A	B	C	D	E	F	G	H	I
1	#	Risico Beschrijving	Kans	Impact	Risico	Maatregel	Status	Status Omschrijving	Datum
2	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
3	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
4									
5									
6									
7									
8									
9									
10									

3. Analyseren

Huidige situatie:
De verbinding tussen de cliënt en een eigen server is momenteel direct met elkaar verbonden. Dat betekent natuurlijk dat er nog geen centrale server is. Waarom zou je een centrale server gebruiken?

- Het is efficiënter
- Makkelijker uit te breiden
- Klanten/gebruikers van andere banken kunnen bij elkaar pinnen

Voor de communicatie tussen de centrale servers met de cliënts kunnen we gebruik maken van de REST API[1][2][3][4]. REST API staat voor **RE**presentational **S**tate **T**ransfer **A**pplication **P**rogramming **I**nterface. Ook bekend als RESTful web servive. REST API is een applicatie programma interface dat HTTP requests verwerkt en een response terug geeft. Veel grote bedrijven zoals youtube, facebook en instagram maken gebruik van een API. Met de API kunnen we de banken laten communiceren via een centrale server. De API werkt met request and respond. Een API kan je vergelijken met een ober. Stel voor dat de keuken het systeem is en je de cliënt een besteller is. De cliënt besteld een gerecht en de ober geeft dat aan in de keuken, kort hierna brengt de ober de gekozen gerecht naar de gebruiker. De ober is dus niets anders dan een boodschapper (tussen persoon) die een request verstuurt naar de server en een antwoord terug geeft aan de cliënt.

3.1 Security[5]

Het is belangrijk om ervoor te zorgen dat de gegevens van klanten niet mag uitlekken. Dit voorkom je door de communicatie, cliënt en server te beveiligen. Niemand wilt hun gegevens te lezen zijn door andere mensen. Ook willen ze natuurlijk ook niet dat mensen geld stelen van hun bankrekening. Om de API en de communicatie te beveiligen kunnen we gebruik maken van **T**ransport **L**ayer **S**ecurity (TLS) protocol en encryptie/decryptie sleutels.

3.2 TLS[5]

TLS is een verbetering van Secure Sockets Layer protcol (SSL). TLS beveiligd data transfers tijdens het communiceren over computernetwerk. Hoe zorgt TLS voor beveiliging?
Bij het gebruik van TLS tussen de server en cliënt wordt de data geencrypt tijdens het versturen van de cliënt naar de server en omgekeerd. Data dat verstuurd wordt, wordt onleesbaar gemaakt voor derde partijen.TLS wordt voornamelijk toegepast in situaties waarin het nodig is om te verifiëren of men inderdaad met de juiste server is verbonden. Allereerst wordt de gevonden server geauthentiseerd middels een certificaat dat gebruik maakt van symmetrische cryptografie (public key). Hierdoor weten de gebruikers of het juiste server is.TLS wordt ook vaak toegepast in bancaire toepassingen (internetbankieren).
Als een gebruiker gebruik maakt van de pinautomaat, zal er vanuit de cliënt een JSON request opgestuurd naar de lokale server. Er wordt gevraagd naar de bankcode en saldo. Als er verbinding wordt gemaakt tussen de cliënt en server in TLS, zal de server een bericht terugsturen naar de cliënt middels een JSON response. De server stuurt nu met de JSON response de opgevraagde data terug naar de cliënt. Als de gebruiker ervoor kiest om geld op te nemen van zijn rekening wordt er weer een JSON request gestuurd naar de server van de bank. Geld wordt dan afgeschreven van de gebruikers rekening in de database. Kort hierna wordt er een response gestuurd naar de cliënt. Bij het communiceren wordt er gebruikt gemaakt van TLS. Dat betekent

dat de communicatie beveiligd is en dat de data dat verstuurd wordt onleesbaar wordt voor derden(boosdoeners/hackers). Voordat belangrijke data uitgewisseld wordt, wordt er eerst gekeken of de cliënt de juiste server heeft. Door eerst te kijken of er verbinding is en daarna elkaars certificaten te sturen en controleren. Als het de juiste cliënt en server is, kunnen ze eindelijk met elkaar belangrijke data uitwisselen. Als dit niet het geval is wordt de verbinding verbroken.

3.2 Uitbreidbaarheid

Door gebruik te maken van een centrale bank en API. Zorg je ervoor dat het makkelijk is om functies erbij te zetten. Kan makkelijk veel meer banken/cliënts aan sluiten. Men kan ook makkelijk banken/cliënts verwijderen. Kan verder uit bouwen zodat gebruikers online via desktop of een app online kunnen bankieren. Enz.

3.3 Vertrouwen

Vertrouwen is heel belangrijk voor gebruikers. Zij moeten ons kunnen vertrouwen dat niks fout gaat met hun gegevens en/of bankrekening. Niet dat hun rekening volgende dag ineens leeg staat. Er moet dus ervoor worden gezorgd dat de database beveiligd is en dat de communicatie beveiligd is. Wij moeten als bank de klant overtuigen dat we geen data van hun gaan lekken, daarom moet ons hele bank goed beveiligd ingericht worden. Ook is het belangrijk om geen enkel belangrijke data op de bon te laten printen, zoals de bankpas en/of wachtwoord.

3.4 Tevredenheid

Om ervoor te zorgen dat elke gebruiker tevreden is met onze product is het belangrijk dat onze geldautomaat het goed en snel doet. De transacties moeten snel en correct uitgevoerd worden. De klanten de keuze geven welk biljet ze willen en ook de keuze geven of ze een bon willen of niet. We willen niet dat de klant 1 minuut lang moet wachten voordat het naar het volgende scherm gaat. Of dat er verkeerde, te weinig of te veel biljetten worden gegeven aan de klanten. Ook is het belangrijk om de pinautomaat zo gebruikersvriendelijk te maken en niet al te complex.

3.5 Niet-functionele eisen

Eisen:

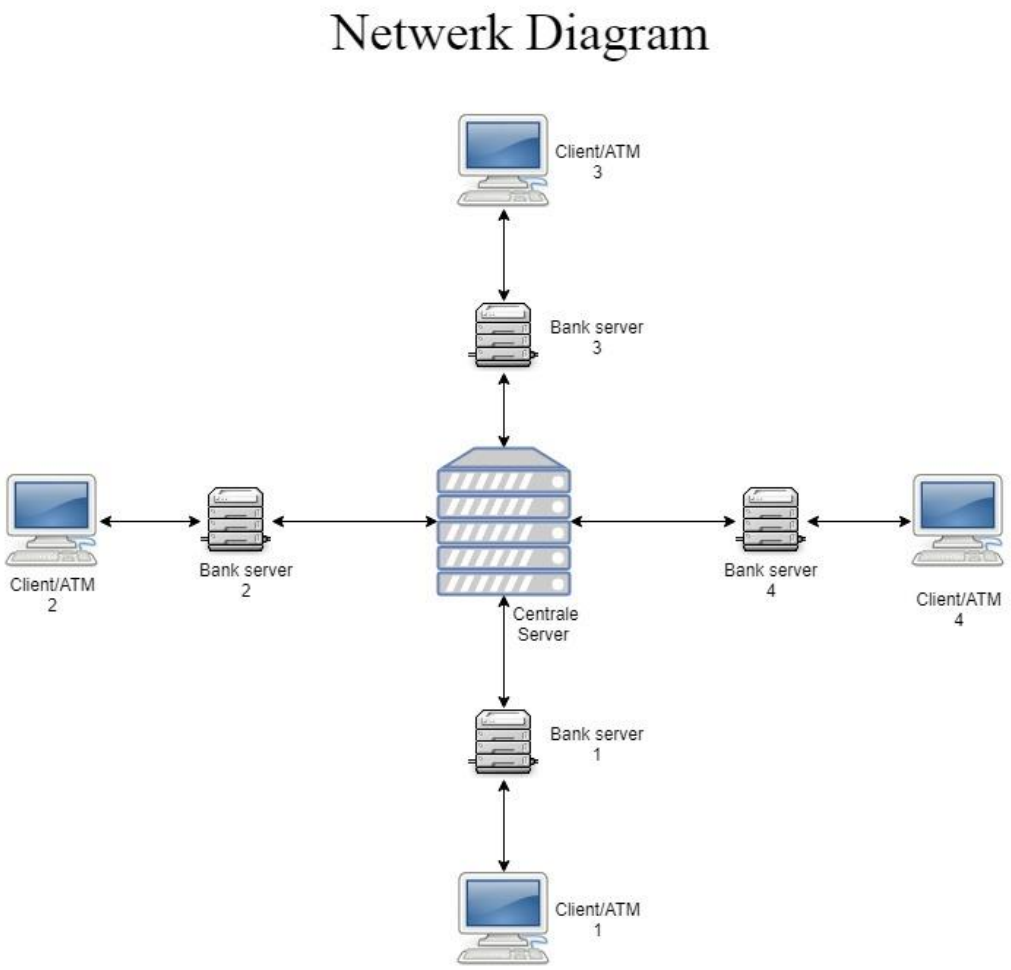
- Code moet leesbaar, commented en efficiënt zijn. (Misra-c guidelines volgen)[6]
- Onnodige dingen (code/poorten/wat je niet gebruikt) verwijderen
- Ervoor zorgen dat je TLS-protocol goed toepast

3.6 Alternatieven

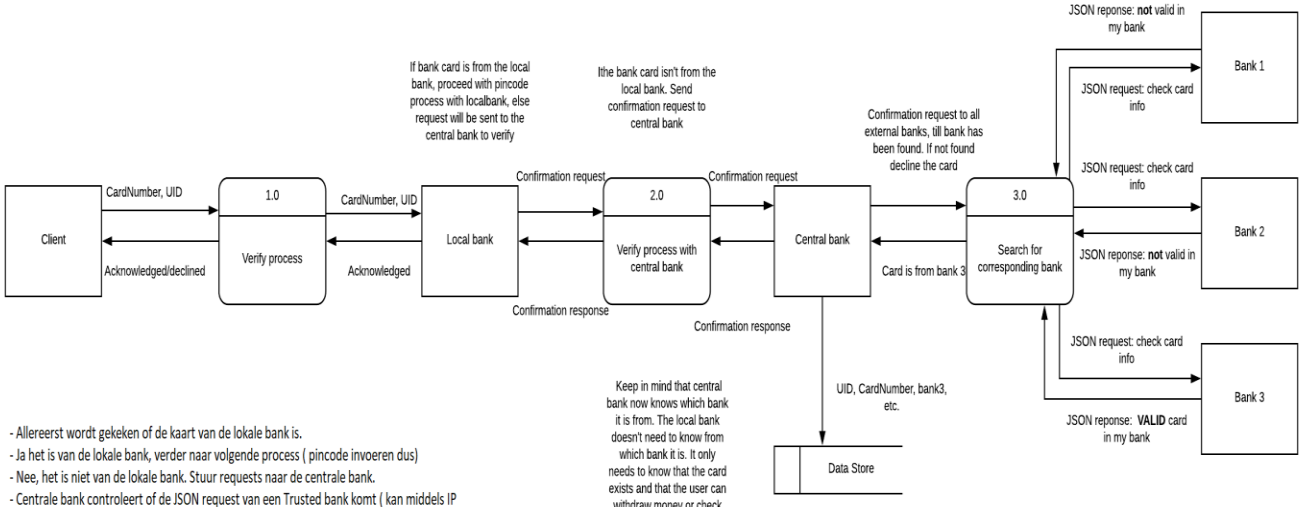
Voor de API kunnen we XML gebruiken in plaats van JSON[8]. Maar dat wordt sterk afgeraden, omdat JSON makkelijker, duidelijker, sneller en efficiënt vergeleken met XML. Voor TLS kunnen we SSL gebruiken, maar die is verouderd en dus ook niet aangeraden. We kunnen ook een directe verbinding maken met andere banken ipv met een centrale server. Maar dat is ook niet handig, niet alleen is dit niet efficiënt, maar ook onveilig. De Database is namelijk op de server geplaatst.

4. Ontwerpen

4.1 Netwerk Diagram

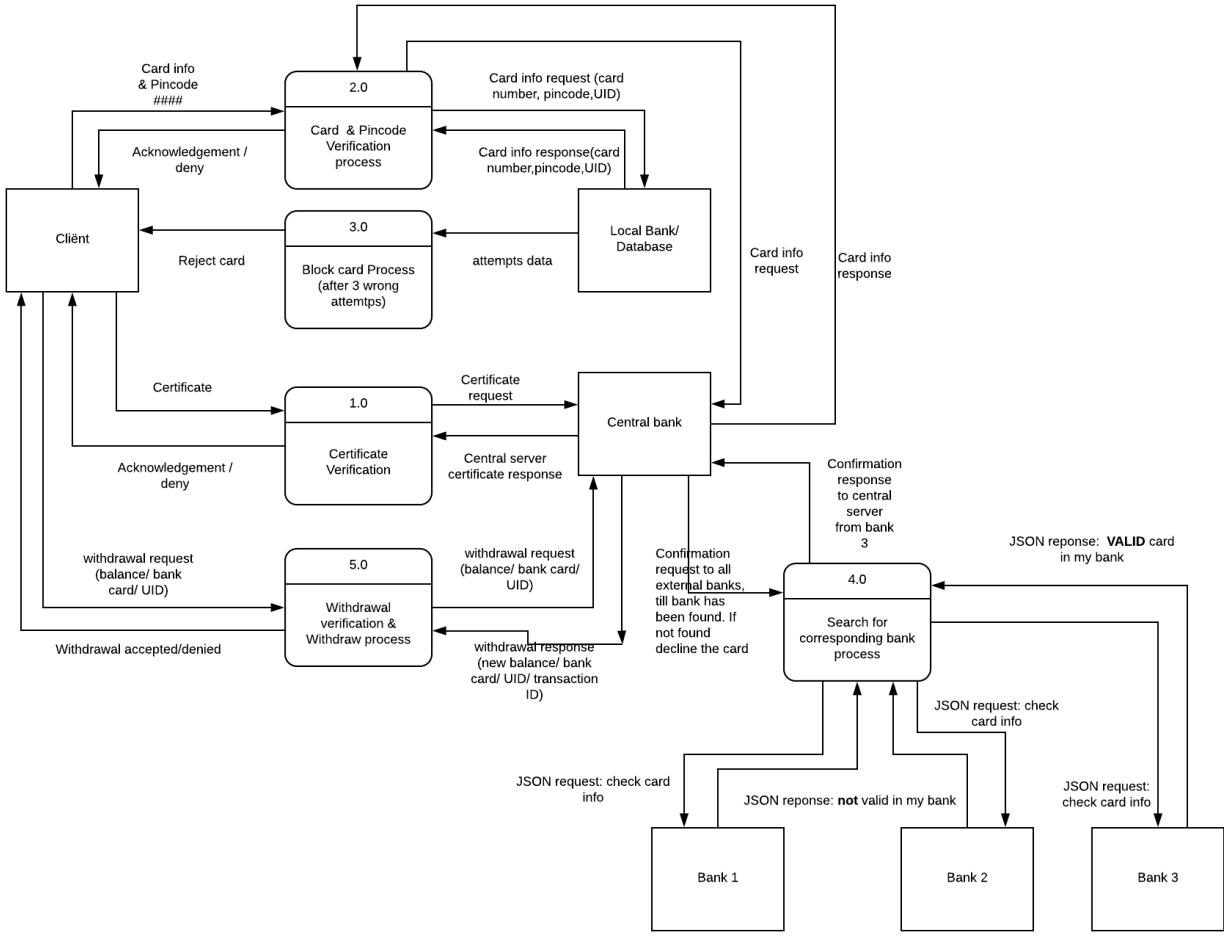


4.2 Dataflow Diagram



- Allereerst wordt gekeken of de kaart van de lokale bank is.
- Ja het is van de lokale bank, verder naar volgende process (pincode invoeren dus)
- Nee, het is niet van de lokale bank. Stuur requests naar de centrale bank.
- Centrale bank controleert of de JSON request van een Trusted bank komt (kan middels IP en/of ceritificaten controleren)
- Centrale bank vraagt de externe banken of de kaart van hun zijn
- Bank 3 reageert met een response dat het wel van hun is.
- Centrale bank stuurt de gegevens door naar lokale bank.
- Alle processen worden via de centrale bank gedaan.
- De lokale bank communiceert niet rechtstreeks met andere banken, dat wordt via de centrale bank gedaan. Door middel van JSON requests & responses.

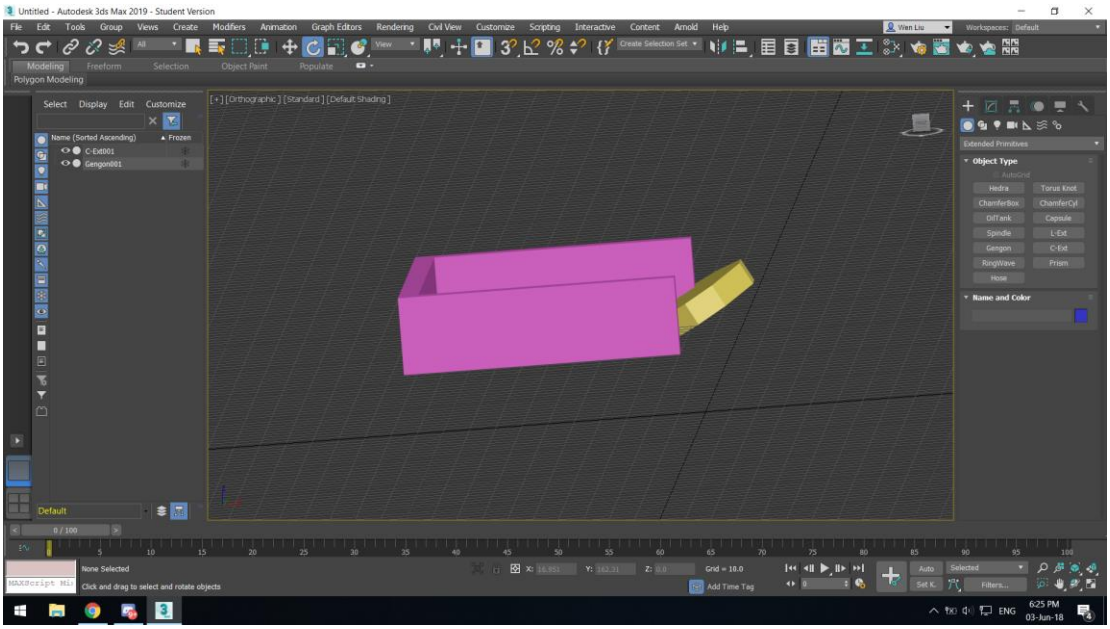
Efficiëntere manier: UID van kaarten opslaan bij de centrale bank zodat de centrale bank gelijk weet van welke bank het toehoort ipv steeds opvragen bij de externe banken. Bijv. UID van bank1 heeft bijv ABC in hun ID en bank 2 heeft DEF enz. Staat er ABC in de UID dan stuurt centrale bank gelijk een JSON request naar bank 1



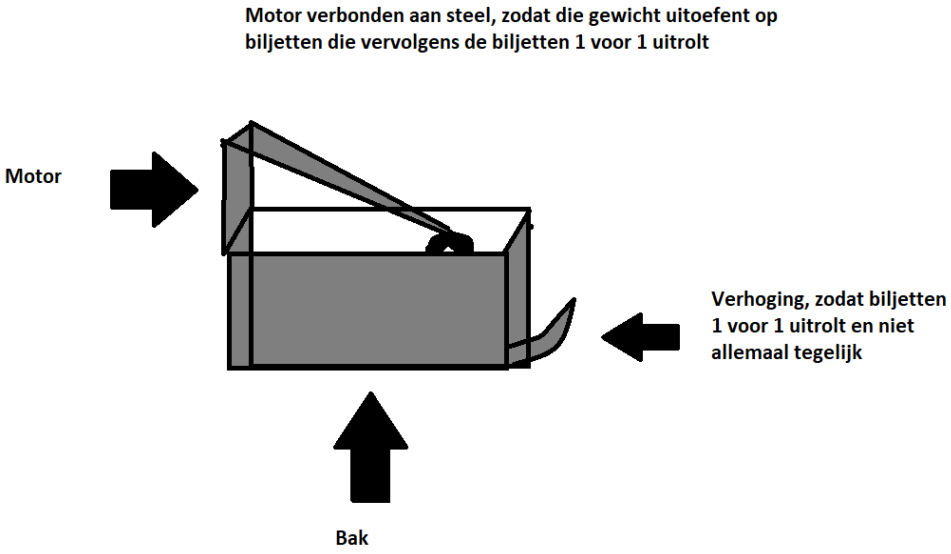
Hele proces van de banken DFD

JSON request/response example

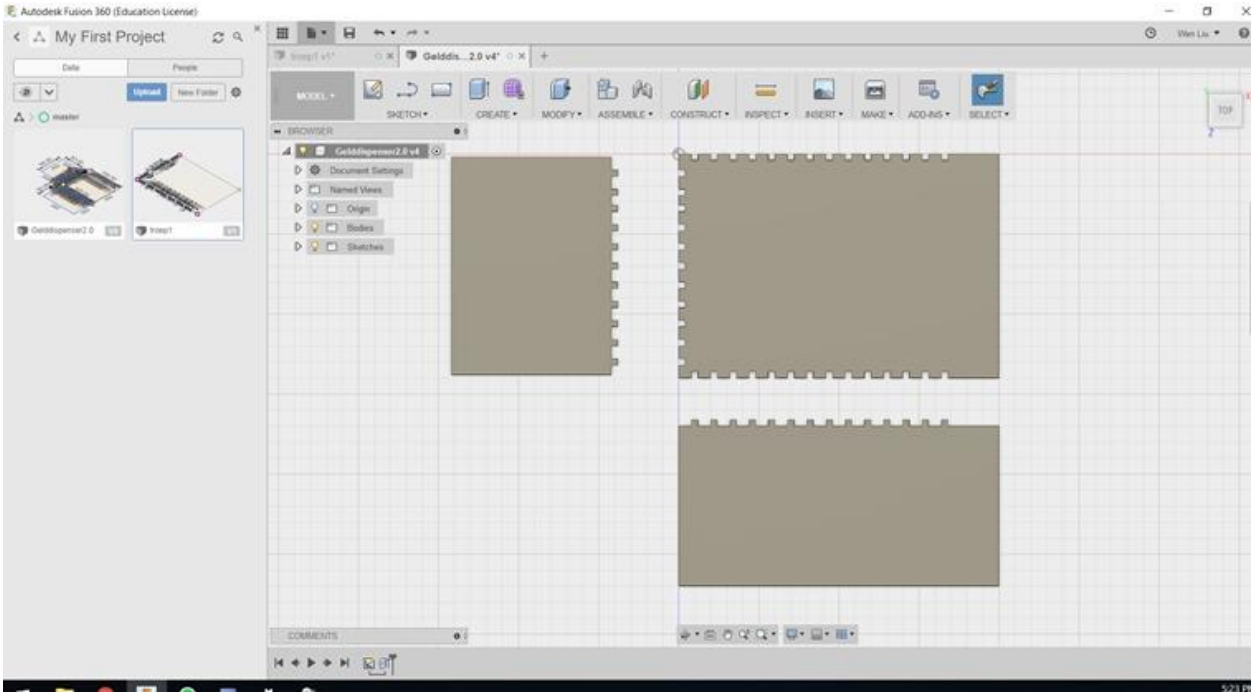
1.3 Ontwerp geld dispenser



Bak ontwerp in autodesk 3D MAX



Voorbeeld van geld dispenser



Bak ontwerp in autodesk fusion. Precies gemaakt zodat ze stevig in elkaar zitten wanneer je het in elkaar laat schuiven.

5. Advies

Liefst geen directe verbinding maken, omdat het niet veilig is en moeilijk uit te breiden. Voor uitbaarheid is het gebruikelijk om gebruik te maken van een API en centrale server. Je kan makkelijk banken toevoegen en verwijderen. Verder wordt er aangeraden om JSON te gebruiken en voor de beveiliging van de communicatie wordt TLS aangeraden. TLS is namelijk een verbeterde versie van SSL en JSON is sneller, efficiënter en ook makkelijk te implementeren. Het is ook slim om eerst bij de lokale server te controleren of een bankrekening van de lokale bank eerst is voordat je gelijk een request stuurt naar de centrale server. Hierdoor voorkom je onnodig zoekwerk voor de centrale server. Ook is het verstandig om de database van je bank te beveiligen.

6. Bronnenlijst

1. https://medium.freecodecamp.org/what-is-an-api-in-english-please-b880a3214a82
2. http://www.restapitutorial.com/lessons/whatisrest.html#
3. https://searchmicroservices.techtarget.com/definition/RESTful-API
4. http://jsonapi.org/
5. https://www.networkworld.com/article/2303073/lan-wan/lan-wan-what-is-transport-layer-security-protocol.html
6. https://www.misra.org.uk/LinkClick.aspx?fileticket=w_Syhpkf7xA%3D&tabid=57
7. https://www.w3schools.com/js/js_json_xml.asp
8. https://www.w3schools.com/xml/
9.