30-4-2018

De centrale bank

Project 4

Naam: Mohamed Abdi
Studentnummer: 0916844

Voorwoord

Hierbij is een adviesrapport over het inrichten van de centrale bank in een efficiënte manier. Het adviesrapport is bedoeld voor alle eerstejaars studenten.

Dit onderzoeksrapport betreft de communicatie tussen de servers. Er wordt in dit rapport over de communicatie onderzocht, beschreven en aanbevolen.

Inhoudsopgave

Inleiding

Deze adviesrapport is bedoeld voor de projectgroepen die in het eerste jaar zitten. Het probleem van de communicatie heeft geleid tot dit adviesrapport. In dit project moet de eerstejaars studenten hun servers met elkaar communiceren zodat de ene groepslid bij de andere groepsautomaten kan inloggen zonder problemen. De vraag is hoe kunnen de groepen de centrale bank in een efficiënte manier inrichten?

In dit adviesrapport zal duidelijk worden gemaakt hoe de servers met elkaar moeten communiceren. Om een advies te kunnen geven, moet eerst gekeken worden naar de opties die er zijn om de communicatie tussen de banken te kunnen realiseren. Dat betekent dat ik deze opties moet bestuderen. Wanneer ik duidelijk beeld heb van deze opties, kan ik er nauwkeurige advies geven aan de hand van mijn bevindingen.

In deze adviesrapport wordt alleen naar twee opties gekeken en wordt de voordelen en nadelen van deze twee mogelijkheden benoemd. Uiteindelijk wordt alleen een van de twee opties gekozen en wordt de gekozen mogelijkheid verdedigd met argumenten.

Beheren

- Alle documenten zijn terug te vinden op git.
- De issue tracking is ook terug te vinden op git.
- De link: https://github.com/Gewad/Project4Bankalicious/issues
- Hieronder ziet u de Risicolog.

Risicolog:

Risico Beschrijving	Kans	Impact	Risico*	Maatregel	☺	Status Omschrijving	Datum
De school brandt af	2	5	10	De school verlaten	😐	Net neergezet, nog geen maatregel voor genomen. Wel brandblussers gezien	25-4-2018
Een van mijn groepsleden verlaat de groep	3	3	9	Zijn taken verdelen over de groep	☺	Er staan afspraken hierover in het samenwerkingscontract	9-5-2018
Geen concentratie	2	3	6	Energie verdelen over de dag	😐	Ik probeer na elke 45 minuten pauze van 20 minuten te nemen	9-5-2018
Te weinig informatie op internet	3	4	12	De vragen die je hebt stellen aan de docent	☺	Vragen zijn beantwoord door docenten	9-5-2018
Git verknoeien	2	2	4	Git leren	😐	Ik probeer git te begrijpen	16-5-2018
Groepslid komt meerdere keren niet.	2	3	6	Taart meenemen voor elke keer dat je afwezig bent	N	Nieuw, de bestaande maatregel is niet heel sterk	16-5-2018
Mijn computer crasht	1	4	4	Git gebruiken of Dropbox	N	Mijn bestanden op Git bewaren	23-5-2018
Ziek zijn	2	5	10	Medicijnen innemen	☺	Ik ben beetje beter geworden	25-5-2018
Geen motivatie door het warme weer	1	2	2	Mezelf dwingen om te werken	N	Beetje gewerkt, maar niet klaar met individuele deel	28-5-2018

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

De communicatie opties:

Optie 1: De Centrale Server

De eerste optie is dat er een centrale server komt tussen de groepsserver. De centrale server zorgt voor de communicatie tussen de groepsserver. De centrale server werkt als volgt: Wanneer iemand van groep A met zijn RFID probeert in te loggen in groep B, de server van groep B bekijkt eerst in zijn eigen database. Als de server de gegevens van die persoon niet vindt, maakt die vervolgens verbinding met de centrale server. Op dat moment maakt de centrale server gelijk verbinding met andere servers. Wanneer de centrale server response krijgt van de server van groep A met de gegevens van die persoon, stuurt de centrale server de gegevens verder aan de gevraagde server namelijk aan groepsserver van groep A.

Voordelen

- Minder verbindingen: er is maar één verbinding tussen de centrale server en de groepsservers.
- Groepsservers weten niks van elkaar.

Nadelen

- Zonder de centrale server er is geen verbindingen tussen de groepsservers.

Optie 2: Direct verbindingen tussen de groepsservers.

De bedoeling van deze oplossing is dat elke groepsserver een cliënt wordt bij de andere groepsservers. Wanneer er een groepslid pint bij een andere groepsbank met zijn RFID-Card. De server ontvangt de gegevens van de card, kijkt als eerste of de gegevens van deze persoon in de database is, als dat niet het geval is, verstuurt de server de data naar de andere groepsservers en wacht op een respons. Op dat moment wordt de server een cliënt voor andere groepsservers. Dit betekent dat alle groepsservers rechtstreeks met elkaar communiceren.

Voordelen

- Als een van de groepsservers uitvalt, blijven de overige servers functioneren.

Nadelen

- De servers moeten meer verbindingen maken om de data te verkrijgen.
- Door de verbindingen met meerdere servers worden de servers belast en er ontstaat de risico dat een server uitvalt en niet meer functioneert.
- Te veel dataverkeer.

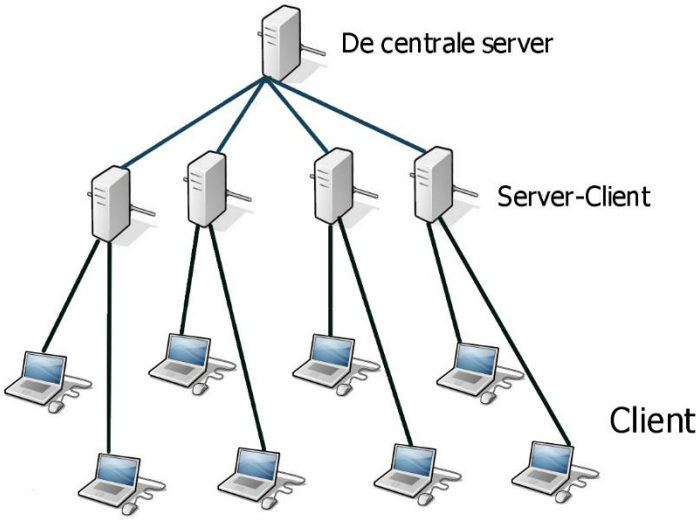
Analyse, Advies en ontwerp

Naar mijn mening de groepen kunnen het beste gebruik maken van de eerste optie namelijk een server tussen alle groepsservers, zogenaamd centrale server. Ik raad de Java sockets aan voor het inrichten van de centrale bank.

Het idee van een server tussen de groepsservers is beter dan het direct verbinden van de groepsservers met elkaar. De reden daarvoor is dat de tussen server minder nadelen heeft dan wanneer je de groepsservers met elkaar direct verbindt. De taak van het zoeken naar data ligt nu bij de centrale server en daardoor wordt de risico dat de groepsservers uitvallen of niet functioneren helemaal vermeden. Een groepsserver kan uitvallen op het moment dat er te verbindingen wordt gemaakt met andere groepsservers.

<u>Maakbaarheid:</u>

Met Java sockets worden de groepsservers als cliënten gezien tegenover de centrale servers.
Java biedt klassen aan voor het maken van sockets om de communicatie over het internet mogelijk te maken. Sockets zijn de eindpunten van logische communicatie tussen cliënten en servers en het wordt gebruikt om data te versturen en te ontvangen. Java sockets lijkt op I/O operations, dat wil zeggen deze programma's lezen en schrijven van en naar de sockets.



Java sockets werkt als volgt: De cliënt verstuurt requests naar de server. Vervolgens de server reageert op de cliënt. De cliënt moet in het begin een connectie maken met de server. De server kan op dat moment accepteren of weigeren. Op het moment dat er een connectie is tussen de server en de cliënt, de cliënt en de server kunnen communiceren via de sockets. Een voorwaarde: wanneer de cliënt een connectie wil maken met de server, de server moet aan het runnen zijn. De server wacht op een request van de cliënt.

Java sockets maken gebruik van IP (Internet Protocol) adressen en porten. Elke specifieke server kan een aantal verschillende functies uitvoeren. Daarom moet de server onderscheid kunnen maken tussen de verschillende verzoeken die hij krijgt van een cliënt. Bijvoorbeeld hij moet onderscheid kunnen maken tussen email request en request voor webpagina's. Door deze porten nummers kunnen de cliënten bepaalde data vragen.

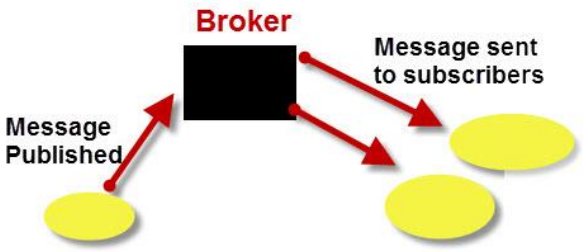
Aangezien dat alle groepssleden dit jaar Java uitgebreid hebben behandeld, is het handig om Java socket te gebruiken. Sommige groepen hebben Java al in project 3 geïmplementeerd. Bovendien er zijn veel informaties over Java sockets te vinden op internet.

Protocollen:

Er zijn verschillende manieren om de data over het netwerk te kunnen versturen. Deze manieren moeten volgens bepaalde protocollen. Deze protocollen moeten opgevolgd worden om data in goed orde af te leveren. In adviesrapport licht ik MQ en MQTT-protocollen uit. Uiteindelijk wordt een van deze twee protocollen gekozen.

MQTT

MQTT oftewel Message Queue Telementry is ontwikkeld in 1999 door IBM. MQTT wordt gebruikt voor telemetrie en het is geschikt voor het transporteren van berichten die kleine hoeveelheden data bevatten. Dit protocol is een open source protocol. MQTT werkt met brokers. Als een broker een bericht krijgt van bijvoorbeeld een cliënt, kan die aan de hand van de Quality of Service die hij heeft, het bericht door sturen aan de subscribers.



MQTT- Publish Subscribe Model

MQ

MQ oftewel Message Queuing lijkt op MQTT, maar dat klopt niet. Bij MQ wordt het bericht in plaats van naar alle subscribers naar één consumer gestuurd. MQ is een op een communicatie. Bij MQ moeten de kanalen aangemaakt worden terwijl bij MQTT de kanalen automatisch worden aangemaakt.

Ik raad MQTT aan, omdat het meer efficiënter is dan MQ. MQTT heeft hoge networklatency en verbruikt laag stroom. Berichten bereiken snel hun bestemmingen. Bovendien MQTT is schaalbaar. Dat wil zeggen dat MQTT aanpasbaar is in capaciteit. We kunnen meer subscribers toevoegen. Bij MQTT de server raakt ook niet overbelast, omdat de data die verstuurd wordt lichter is dan bij MQ.

Security (beveiliging):

Data die tussen de Centrale server, groepsservers en cliënten gaat kan makkelijke toegankelijk zijn bij derden. Vooral wanneer de data uit persoonlijke gegevens bestaat, bijvoorbeeld pincode en creditcardnummers, moeten er maatregelen komen om die data te kunnen beschermen van afluisteringen. Het is ook belangrijk dat de data beveiligd wordt van veranderingen tijdens het versturen van die data. The Secure Sockets Layer (SSL) en

Transport Layer Security (TLS) waren ontworpen voor de beveiliging en de volledigheid van data die over het internet gaat. TLS is de huidige uitvoering van SSL.

Java Secure Socket Extension (JSSE) maakt beveiligde internetcommunicatie mogelijk. Java biedt een implementatie voor SSL en TLS-protocollen. Met deze protocollen kan de data versleuteld worden tijdens het transport tussen de centrale server, groepsservers en de cliënten. Daardoor blijft de data beveiligd en onveranderd.

Wanneer iemand wil inloggen in zijn bank gaat de cliënt op dat moment een verzoek sturen naar de groepsserver. De groepsserver stuurt een sleutel terug naar de cliënt. Met deze sleutel kan de cliënt de informatie van die persoon inpakken. De bedoeling van de sleutel is dat de cliënt de informatie met dezelfde sleutel terugstuurt. Wanneer de cliënt de informatie codeert, moet hij ook een willekeurig getal verzinnen zodat de groepsserver en de cliënt privégesprek kunnen voeren. Bij het uitpakken van de gecodeerd informatie heb je een andere sleutel voor nodig en die heeft alleen de groepsserver.

Vertrouwen:

Naast het beveiligen van de data die tussen de server en de cliënt gaat, maakt SSL-certificaat de data tijdens het transport vertrouwelijk en onleesbaar. De bedoeling van deze certificaat is het registreren van de sleutels. Wanneer een cliënt een publieke sleutel krijgt, kan hij navragen of dit inderdaad een geregistreerde sleutel is die door de groepsserver gebruikt wordt.

Ik raad de projectgroepen TLS te gebruiken, omdat die een verbeterde en veiligere versie is van SSL. SSL-protocollen zijn verboden te gebruiken voor bedrijven. In TLS is een nieuwe beveiligingsmechanismen toegevoegd en daardoor is het meer betrouwbaar dan SSL.

TLS-certificaat werkt als volgt:

1. De cliënt maakt connectie met de groepsserver.
2. Daarna vraagt de cliënt de identificatie van de groepsserver aan.
3. De groepsserver moet vervolgens een kopie van de certificatie sturen.
4. Nadien bepaalt De cliënt of het certificaat te vertrouwen is. Als dat het geval is, krijgt de groepsserver een bericht van de cliënt.
5. De versleutelde verbinding wordt opgezet.

Er is een kans dat de centrale server uitvalt en er geen communicatie meer is tussen de groepsservers. Om de communicatie van de groepsservers betrouwbaar te maken, moet er een oplossing komen voor dit probleem. De oplossing voor dit probleem is de redundantie server. De redundantie server werkt als een back-up voor de centrale server. Dat wil zeggen dat de redundantie server de taak van de centrale server overneemt wanneer de centrale

server niet meer functioneert. De redundantie server heeft de functionaliteit van de centrale bank.

Efficiency:

Een server inrichten tussen de groepsservers is efficiënter dan wanneer je de groepsservers met elkaar direct verbindt. De groepsservers direct verbinden kunnen voor zorgen dat die vaak uitvallen en niet werken en daardoor zal meer tijd kosten om die te repareren.

Dataflow diagram: