|  |
| --- |
|  |
| Advies over de communicatie tussen de groeps-servers |
| De centrale bank |

|  |
| --- |
| Paul de Hek  0941736  Ti1C |

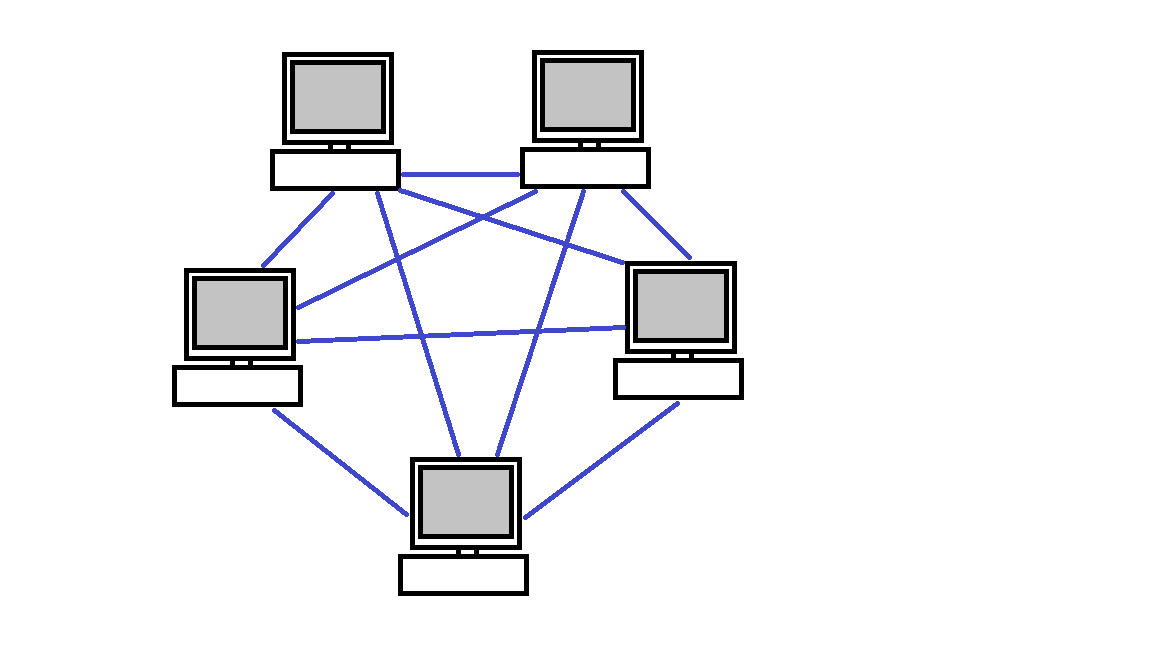
Opties voor communicatie

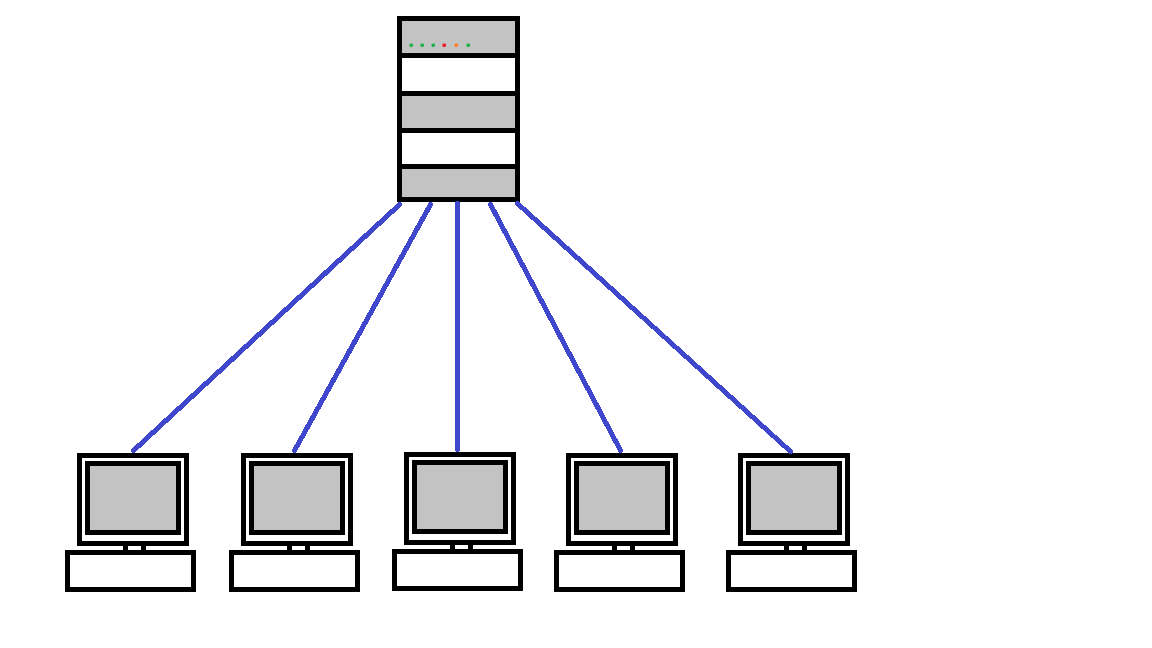
* Peer-to-peer netwerk
* Centrale server

**ANALYSE**

**communicatie**

**Peer-to-peer netwerk:**  
In deze vorm van communicatie zijn de servers van elke bank verbonden met alle andere servers van de andere banken. Hieronder is dit schematisch weergegeven.

In deze oplossing zijn alle groepsservers (hierna servers) direct verbonden. Dit moet uiteraard veilig gebeuren en efficiënt gebeuren. de grootste uitdaging zal zijn verschillende groepen verschillende programmeertalen hebben gebruikt. Er zijn twee manieren waarop dit opgelost kan worden. De eerste is dat er één programmeertaal word gekozen waarin de servers met elkaar communiceren. Dit is waarschijnlijk de taal waarin de meeste servers geschreven zijn (in dit project zal dit Java zijn), hierdoor zal een deel van de servers niks hoeven te vertalen maar een server die in een andere taal is geschreven, zoals C, zal zowel bij het verzenden als het ontvangen de data moeten omzetten in zijn taal waardoor deze bank langzamer zal werken. De tweede oplossing is dat elke server in meerdere talen kan data kan ontvangen en dit zelf omzet naar de taal die binnen de server word gebruikt. Er zal een afweging gemaakt moeten worden aangezien bij de eerste oplossing er alleen werk zal moeten worden verricht als de server niet is geschreven in de afgesproken communicatie taal en is er voor een deel van de servers geen extra vertaal werk nodig. Bij de tweede oplossing is voor iedereen verzenden even snel omdat iedereen data verzend in zijn eigen taal maar het ontvangen zal langer duren. Ook is dit voor iedereen meer werk omdat elke server elke programmeertaal moet kunnen ontvangen die in het systeem word gebruikt.  
Security is bij een peer-to-peer netwerk lastig omdat er zoveel verbindingen zijn die allemaal beveiligd moeten worden. Als er TLS gebruikt word zal dit over elke connectie gebruikt moeten worden. Dit is mogelijk maar het is niet heel erg praktisch om te doen.

**Centrale server:**  
Bij deze vorm van communicatie zijn alle servers verbonden met één centrale server die de data tussen de servers verstuurd. Dit is hiernaast weer schematisch weergegeven.

Bij dit model zullen de servers verbinding maken met een centrale server die de data heen en weer stuurt tussen de goede servers. Een scenario kan als volgt gaan:

Een server leest een bankpas die hij niet in zijn database heeft staan, hij zal dan naar de centrale server een bericht verzenden om te kijken of de bankpas in een database van een andere server staat. De centrale server zal dan een bericht sturen naar alle andere servers. Als een server terug stuurt dat hij de pas herkent zal de centrale server dit terug sturen naar de originele server. Ook zal de centrale server doorsturen bij welke bank de pas hoort.

Het scenario wat ik hierboven heb beschreven is de eerste interactie tussen een bepaalde bankpas en de server.  
Bij deze vorm van communicatie is hetzelfde probleem aanwezig als bij het peer-to-peer netwerk. De servers zijn in verschillende talen geschreven. Het is nu alleen makkelijker om alle servers in hun eigen taal de data te laten sturen aangezien er maar een ontvanger is, de centrale server. Deze word waarschijnlijk in Java geschreven en er zullen klasses zijn om de data uit verschillende talen om te zetten naar iets wat de centrale server kan lezen. Ook zullen deze klasses de data weer terug omzetten zodat de server de data in zijn eigen taal ontvangt. Deze methode is ook via TLS te beveiligen

**Vergelijken**

Er zijn uiteraard verschillen tussen deze twee manieren van verbinden ik zal hier de meest belangrijke verschillen behandelen en de voor- en nadelen ten opzicht van elkaar

Een peer-to-peer netwerk heeft als voordeel dat de communicatie in theorie iets sneller is omdat de data direct van de ene server naar de andere server gaat. Bij een centrale server gaat dit nog door een extra server die dan nog een keer extra de data moet verwerken waardoor het langzamer is.  
Dit heeft wel een keerzijde waardoor een peer-to-peer netwerk in praktijk zelfs langzamer kan zijn dan een centrale server gebruiken. Dit komt omdat bij een peer-to-peer netwerk de server die de request doet zelf alle informatie moet verwerken om te kijken met welke server hij verder moet communiceren. Het is makkelijker om dit op een centrale server te doen die alleen maar de data moet doorgeven van de ene server naar de andere

Bij een peer-to-peer netwerk heb je ook meer controle over met wie je welke data deelt dan met een centrale server. Bij een centrale server krijg je namelijk alleen een aanvraag om bepaalde data te verzenden of te verifiëren, dat stuur je terug naar de centrale server en die bepaalt waar de data naar toe gaat. Bij een centrale server heb je dus geen controle over hoe en naar wie jouw data verstuurd wordt. Bij een peer-to-peer netwerk kan je zelf bepalen naar wie en welke dat je verstuurd wat je dus meer controle.