

Individueel deel: Inrichting centrale bank

Boraida Al-Assbahi (0947569)

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	Tijdsnood	3	3	9	Planning maken en op tijd beginnen	😊	Uitzoeken wat er allemaal nog moet gebeuren.	23-5-18
R3	Gebrek aan communicatie	2	3	6	Houd de communicatie binnen de groep actief	N	Net neergezet, nog geen maatregel voor genomen.	28-5-18

Issue tracking

- De Issue Tracking wordt bijgehouden tijdens het project.
- Template;

○

#	Datum In	Issue	Verantwoordelijk		Status Log	
					Datum	Beschrijving

- □: [status] :) opgelost, :| bezig; :(niet opgelost; N nieuw

Analyseren

Kwaliteit: Security

- We werken met gevoelige data en geld, daarom is er gekozen om op 'security' te concentreren.
- De API waar de geldautomaat mee communiceert moet gebruik maken van SSL.
- JSON Web Token (JWT) moet worden gebruikt om de ingelogde gebruiker op de geldautomaat te identificeren.
- Elke request naar de API moet de JWT meegestuurd worden.
- De Message Queue Broker wordt alleen gemachtigd via de API.
- De centrale server moet verifiëren dat de geldautomaat wel bestaat.

Kwaliteit: Privacy

- Bij het inloggen op de geldautomaat krijgt de klant alleen zijn saldo te zien, zodat zijn identiteit geheim blijft.
- Alleen de administrator heeft de mogelijkheid om de gegevens van de klant op te vragen.

Kwaliteit: Tevredenheid

- We moeten de klanten tevreden houden.
- Wanneer de klant bijvoorbeeld geld wilt opnemen moet het niet lang duren voordat het gebeurt.
- Moet feilloos werken zonder dat het vastloopt.

Kwaliteit: Vertrouwen

- Security en privacy is 'key' in dit project.
- De centrale server moet ten alle tijden vertrouwd aanvoelen tijdens het gebruiken.
- Daarom wordt er alleen het saldo meegegeven na het inloggen.

Eisen: Functioneel

- 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 bonprinter. Op deze bon staat in ieder geval hoeveel geld er is opgenomen en bij welke (lokale of individuele) bank dit is gebeurd

Eisen: Niet-functioneel

- Communicatie tussen servers moet geëncrypt worden.
- De code moet op een nette manier geschreven worden zodat er gemakkelijk nieuwe functies geïmplementeerd kunnen worden.
- De code voor de Arduino wordt geschreven in C++.
- Klassen mogen niet meer dan 400 regels code hebben.

Alternatieve oplossingen

- WebSockets [<https://www.maxcdn.com/one/visual-glossary/websocket/>]
 - Lijkt op HTTP, maar is het veel sneller. Het is ook een real-time verbinding die je vaststelt met de server. Dit kan heel handig zijn, maar dan moet er continu verbinding zijn.
- Hypertext Transfer Protocol (HTTP) [<https://searchwindevelopment.techtarget.com/definition/HTTP>]
 - HTTP is een onveilige protocol. Je kunt het gemakkelijk gebruiken voor websites, maar wanneer het over privacy en/of security gaat is het beter dat we HTTPS gebruiken. HTTP is heel makkelijk af te luisteren.
- Directe database verbinding
 - Dit is heel erg onveilig, want dat betekent dat de client (geldautomaat) toegang heeft tot alle informatie in de database. Dat is een schending van privacy en security.

Adviseren

Client API (HTTPS)

- De geldautomaat maakt gebruik van de API van project 3.
 - Het dient als middleware tussen de geldautomaat en de Message Broker.
 - Het is beveiligd en encrypted met SSL.
 - Het is onleesbaar wanneer het onderschept wordt.
-
- Bron: <https://www.instantssl.com/ssl-certificate-products/https.html>

JSON Web Token (JWT)

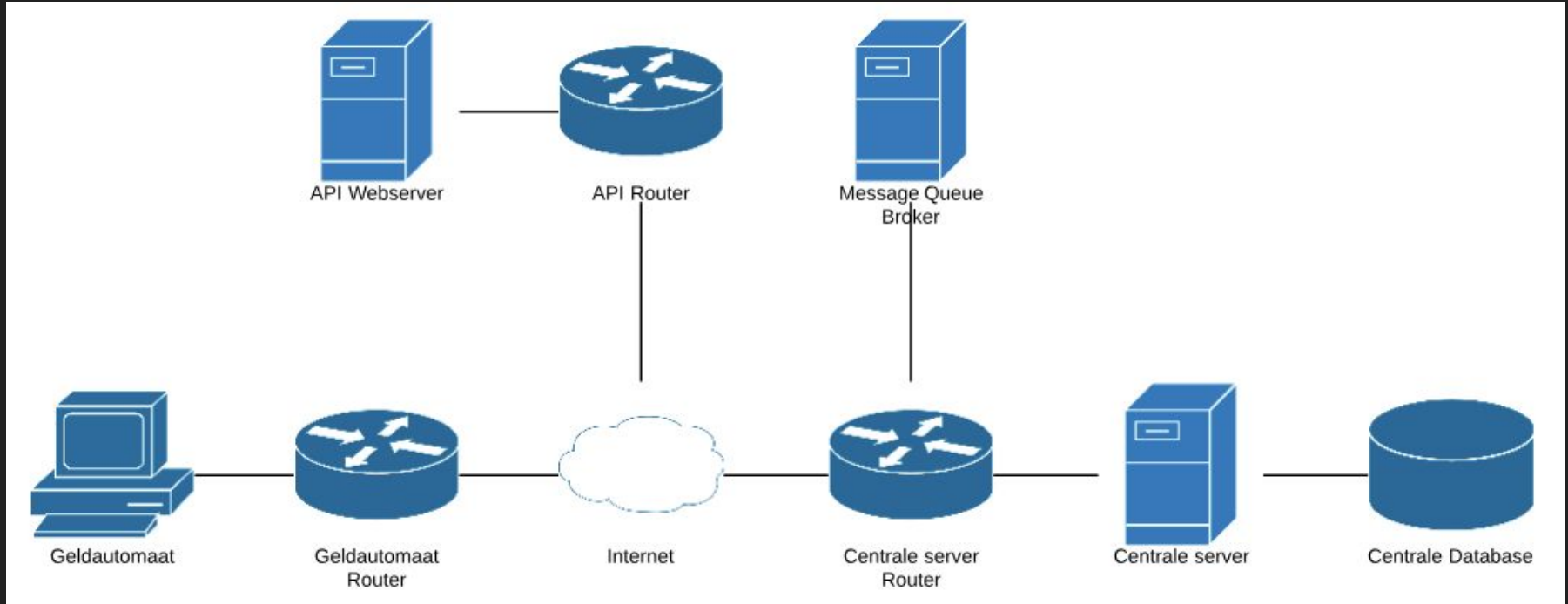
- JWT wordt gebruikt door bijvoorbeeld de ID van de gebruiker te encrypten.
 - Gestuurd vanaf de API naar de geldautomaat .
 - Geldautomaat moet altijd de JWT meesturen voor elke request naar de API.
 - Elke JWT zal een vervaltijd hebben (15 minuten), maar deze wordt weer vernieuwd wanneer de gebruiker actief is.
-
- Bronnen:
 - <https://jwt.io/introduction/>
 - <https://openid.net/specs/draft-jones-json-web-token-07.html>

AMQP & Message Queue Broker

- Deze wordt gemachtigd via de API waar de geldautomaat mee communiceert.
 - Dit wordt gebruikt als een centrale punt van informatie tussen de API en centrale server.
 - Het slaat de informatie op, dus ook al is de API of centrale server offline wanneer ze weer online zijn pakken ze hun werk weer op.
-
- Bronnen:
 - https://en.wikipedia.org/wiki/Message_broker
 - <https://whatis.techtarget.com/definition/Advanced-Message-Queuing-Protocol-AMQP>

Ontwerpen

Netwerkdigram



Data flow diagram

