

# Technical design

---

## Inleiding:

Dit is het technical design van Team 3 over project F.Y.S. Hierin zal o.a. het volgende staan:

- Wat een technical design is.
- De benodigdheden om een captive portal op te zetten.
- Hoe zal het eindproduct er uitzien?
- Een bijbehorende flowchart bij de captive portal.

## Wat is een technical design?

Een technical design is de verzamelnaam voor de documentatie met betrekking tot een product. De term verwijst naar verschillende documenten met product gerelateerde gegevens en informatie die gebruikt worden en opgeslagen voor verschillende doeleinden. Verschillende doeleinden verstaan: Productdefinitie en specificatie, ontwerp, productie, productaansprakelijkheid, productpresentatie; beschrijving van de eigenschappen, functies en interfaces.

## De benodigde software:

Het systeem moet om goed te functioneren de volgende 'core' functies hebben.

=> Captive portal

=> Database

=> Access Point

=> Firewall

=> Webserver (voor captive portal)

=> Infrastructuur

## Captive portal

De captive portal moet aan verschillende eisen voldoen. Zo moet deze een inlogmogelijkheid hebben en moeten de gebruikers voorwaarden accepteren. De captive portal is de website die de gebruikers met de server laat communiceren. Er wordt aan de gebruiker gevraagd om ticket nummer en achternaam, daarna of de gebruiker het eens is met de voorwaarden. Wanneer de gebruiker alles correct heeft ingevuld, gaat de captive portal communiceren met de database en de firewall. Er zal dan een programma worden uitgevoerd om de firewall in te stellen.

De regels voor de firewall worden in het script `"/Portal/rules.sh initialize"` in geladen<sup>1</sup>.

Wanneer de gebruiker succesvol heeft ingelogd wordt er vanuit de applet een command uitgevoerd. `"/Portal/rules.sh grand <IP gebruiker>"` is de command die een IP toegang geeft tot het internet<sup>2</sup>. De gebruiker kan daarna gebruik maken van het internet.

## Database

Via de database kan de captive portal aan gegevens komen. Het is daarom ook belangrijk dat alle data van de eventuele gebruikers daarin staan en dat erin kan worden gezet dat een gebruiker van het internet gebruik heeft gemaakt. De database wordt opgehaald door de CenD api. Hier later meer over omdat deze functie nog druk in ontwikkeling is.

### **Acces point**

In een vliegtuig gaan maximaal 189 mensen. Deze mensen mogen maar met 1 apparaat tegelijk verbonden zijn. Er kunnen dus maximaal 189 apparaten verbinden. Er zouden per acces point 30 mensen kunnen verbinden. Het projectteam raad 7 acces points aan dus kunnen er in theorie maximaal 210 apparaten verbinden. Hierdoor worden de acces points niet maximaal belast en kan iedereen van een stabiel internet gebruik maken.

De IP-range van de access points ligt tussen de 10.111.11.10 en 10.111.11.200. Deze range is nodig omdat de 10.XXX.XXX.XXX range staat voor interne publieke netwerken. Net als 192.168.XXX.XXX. Er is voor 10.XXX.XXX.XXX gekozen omdat deze range professioneler is dan een IP adres dat men thuis ook heeft. De TTL van een IP DHCP uitgave staat op forever. Dit komt doordat men per vlucht de gehele tijd van internet gebruik mag maken. Wanneer men door een andere tijdzone vliegt, kunnen de apparaten van de gebruiker hun tijd automatisch aanpassen. Wanneer dan een IP uitgave is verlopen, moet de gebruiker opnieuw inloggen en dat kan irritaties veroorzaken.

### **Firewall**

Voor de beveiliging van gebruikers moeten bepaalde inkomende en uitgaande connecties worden toegestaan en/of worden geblokkeerd. Zo moeten poorten 80/tcp en 443/tcp voor HTTP en HTTPS uitgaand open worden gezet. En moeten 25/tcp, 110/tcp, 993/tcp voor e-mail uitgaand open worden gezet. Poort 80/tcp moet inkomend open worden gezet zodat de captive portal bereikt kan worden. Er moet wel een aantal inkomende poorten worden opengezet voor de eth0 interface op de 'Master access-point' zodat de 'slaves' IP adressen voor hun cliënten kunnen aanvragen en andere verificatie kunnen doen. Hier later meer over omdat deze functie nog in ontwikkeling is.

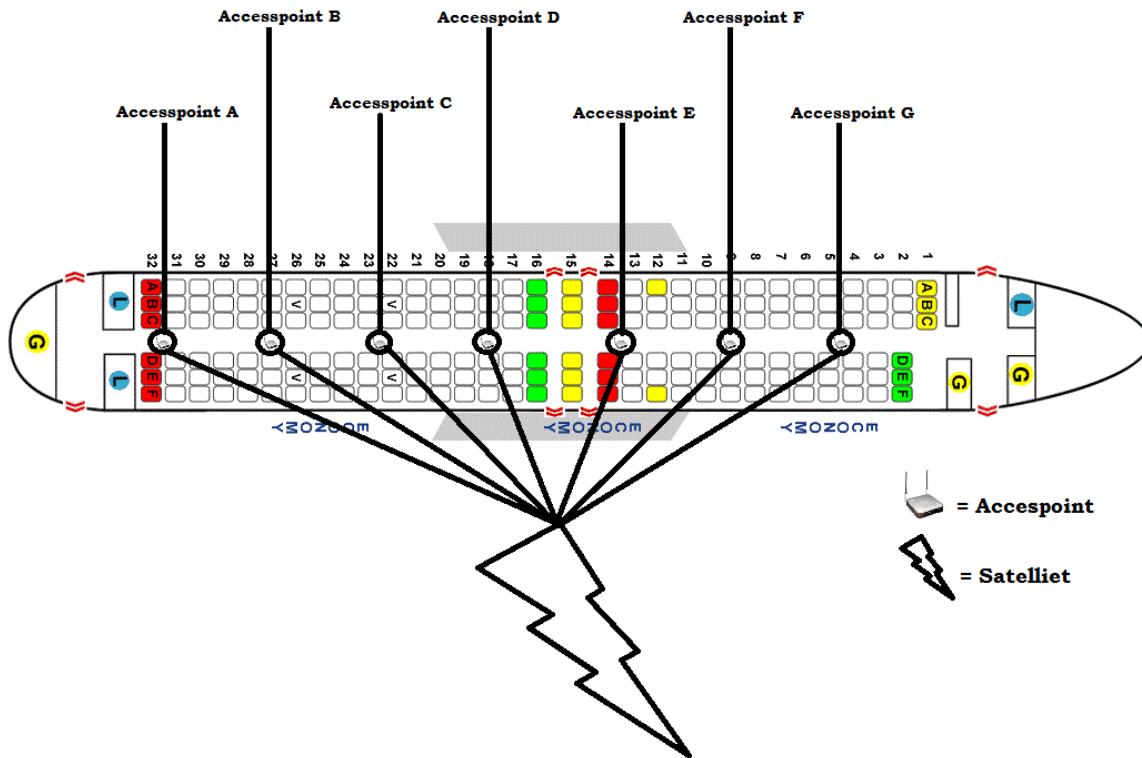
### **Webserver (voor captive portal)**

De webserver moet uit apache en tomcat bestaan. Er wordt apache gebruikt omdat dit de meest gebruikte en stabiele webserver op unix systemen is. Tomcat wordt gebruikt omdat tomcat hosten van java webpages mogelijk maakt. Op de webserver staat een JAVA servlet. Deze servlet maakt het mogelijk voor de gebruiker om in te loggen en om de gebruiker toegang tot het internet te geven<sup>3</sup>.

### **Infrastructuur**

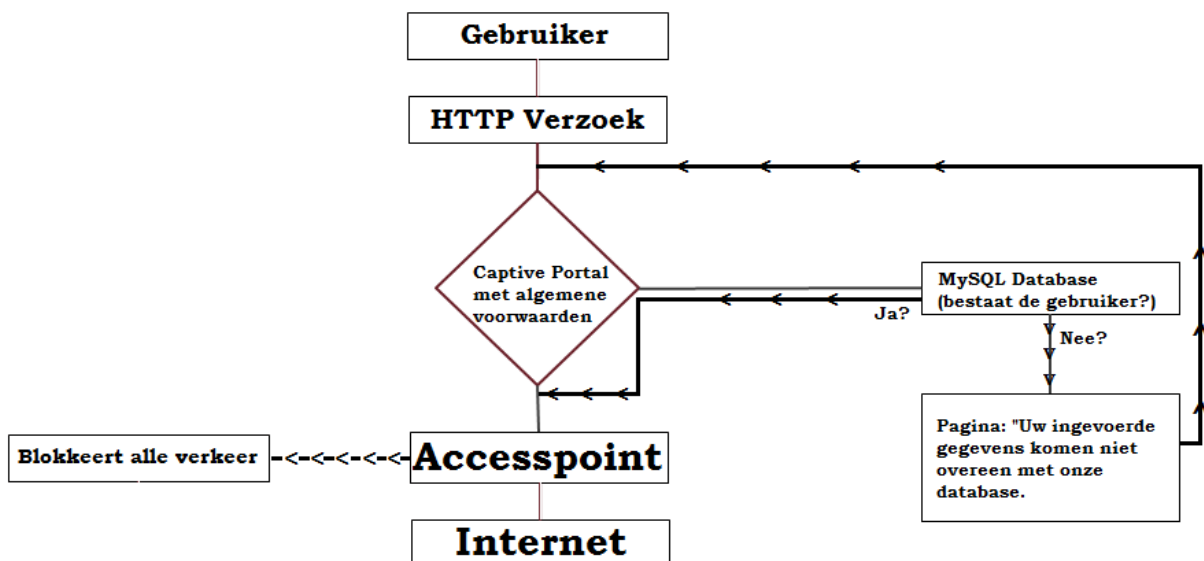
Alle 7 raspberry's gaan naar een 8-poorts switch. Aan de overgebleven poort word dan de satelliet modem aangesloten. Alle apparaten staan dan met elkaar in verbinding en iedereen kan met optimale snelheid van het internet gebruik maken. De apparaten staan met elkaar in verbinding over het zelfde subnet als dat van de WiFi gebruikers. De raspberry's hebben een IP range tussen de 10.111.11.1 en de 10.111.11.9. Zo is er plek voor 7 raspberry's. Het satelliet modem krijgt een IP adres van 10.111.11.254. Hierdoor blijft het netwerk keurig van elkaar gescheiden en kan iedereen van veilig en snel internet gebruik maken.

## Hoe zal het eindproduct er uitzien?



Het eindproduct zal er zoals hierboven uitzien. In de boeing 737-800 passen 189 mensen. Elke passagier mag maar maximaal 1 apparaat verbinden met een access point. Er zijn dus in totaal 7 access points nodig met in totaal 210 verbindingen. Deze access points houden zo genoeg ruimte over om nog eventueel 21 extra apparaten te verbinden met de accesspoints. Zoals hierboven al is verteld zal het maximale aantal apparaten dat een gebruiker mag verbinden met een accesspoint één zijn. De 7 access points zijn verbonden met een satelliet modem.

## Flowchart:



De gebruiker opent zijn internet en dient een HTTP(S) verzoek in. Vervolgens komt de gebruiker bij de Captive portal met de algemene voorwaarden. Hier moet de gebruiker zijn/haar ticketnummer en achternaam invoeren. Als alles correct is, dan kan de gebruiker gebruik maken van het internet nadat de verbinding met de access point (die alle verkeer blokkeert) tot stand is gekomen<sup>2</sup>. Als het ticketnummer en de achternaam fout zijn, dan krijgt de klant een pagina waarin hij/zij ziet dat de gegevens onjuist zijn. Als dit het geval is dan komt de klant weer bij de captive portal terecht waar hij/zij weer zijn/haar ticketnummer en achternaam moet invoeren.

## Appendix

1. Dit onderdeel is te zien met uitleg op [github](#). Gaat in vanaf regel 13 tot 86.
2. Dit onderdeel is te zien met uitleg op [github](#). Gaat in vanaf regel 88 tot 101.
3. Dit onderdeel is te zien op [github](#). Gehele bestand. Dit bestand is onderdeel van de portal en werkt dus alleen in combinatie met de andere bestanden(index.java, index.html, process.java, loading.html, command.java en database.java)
4. How the process works (also on [github](#))  
HTTP request  
||>>By just typing in any website  
User redirected to the captive portal  
||>>By javascript  
User fills in ticket number and lastname  
||>>By a HTML form  
Forms goes to proccess.java  
\_\_\_\_||  
\_\_\_\_||>>lastname and ticketnumber return 1 row? (lastname and ticketnumber combination is unique)  
\_\_\_\_|| Then continue  
\_\_\_\_|| else Tell the user the lastname and ticketnumber combination was not correct  
\_\_\_\_||  
\_\_\_\_||  
\_\_\_\_||>> Update Database (tell the database the user is using internet)  
\_\_\_\_|| Succes? return continue  
\_\_\_\_|| Failure? Tell the user someting went wrong  
\_\_\_\_||  
\_\_\_\_||>>Grand IP adress  
\_\_\_\_|| Succes? continue  
\_\_\_\_|| else update Database (tell the database the user is not using internet)  
\_\_\_\_|| Tell the user someting went wrong  
\_\_\_\_||  
\_\_\_\_||>>Everything OK?  
\_\_\_\_|| Then show the user the loading.html page and The user is free to use the internet  
\_\_\_\_|| else Tell the user someting went wrong