4+1 model

Inhoudsopgave

[Process view 3](#_Toc485633493)

[Logical diagram 3](#_Toc485633494)

[Deployment diagram 4](#_Toc485633495)

[Implementation view 4](#_Toc485633496)

[View 5](#_Toc485633497)

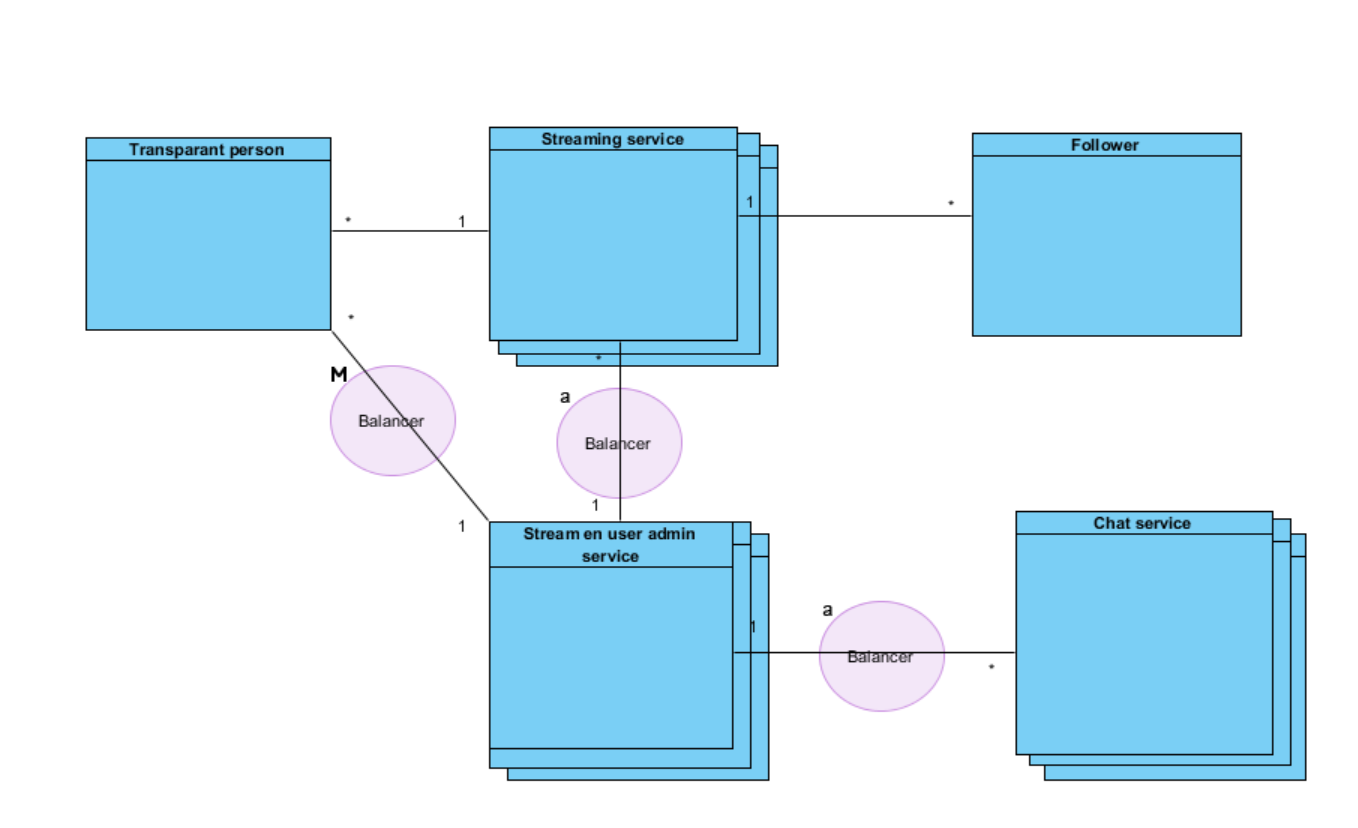
[Android stream flow 5](#_Toc485633498)

[Web App stream flow 5](#_Toc485633499)

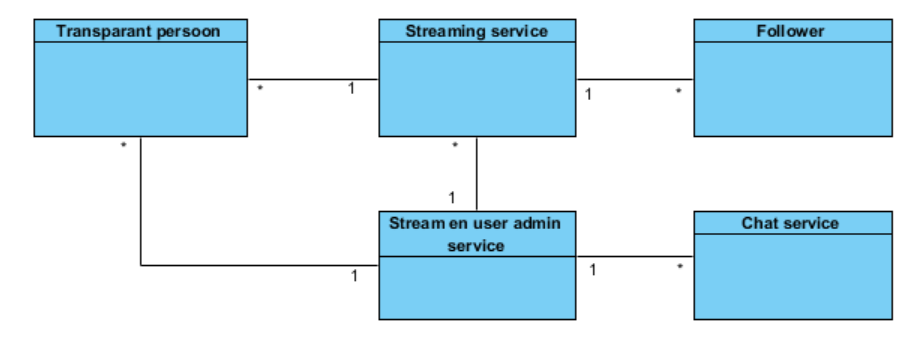
[Scalability 6](#_Toc485633500)

[FSD (fragmented signed data) protocol 6](#_Toc485633501)

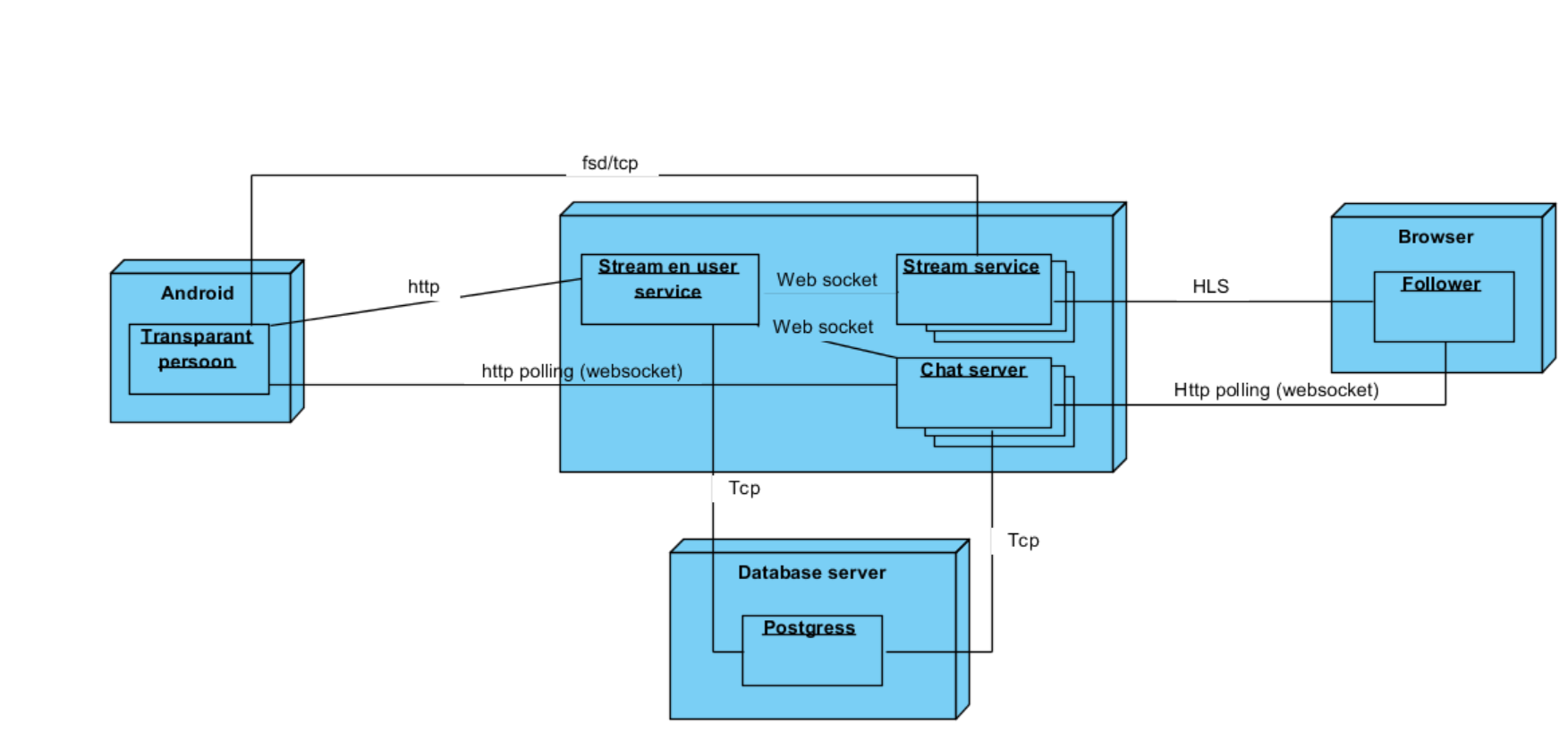
# Process view



# Logical diagram

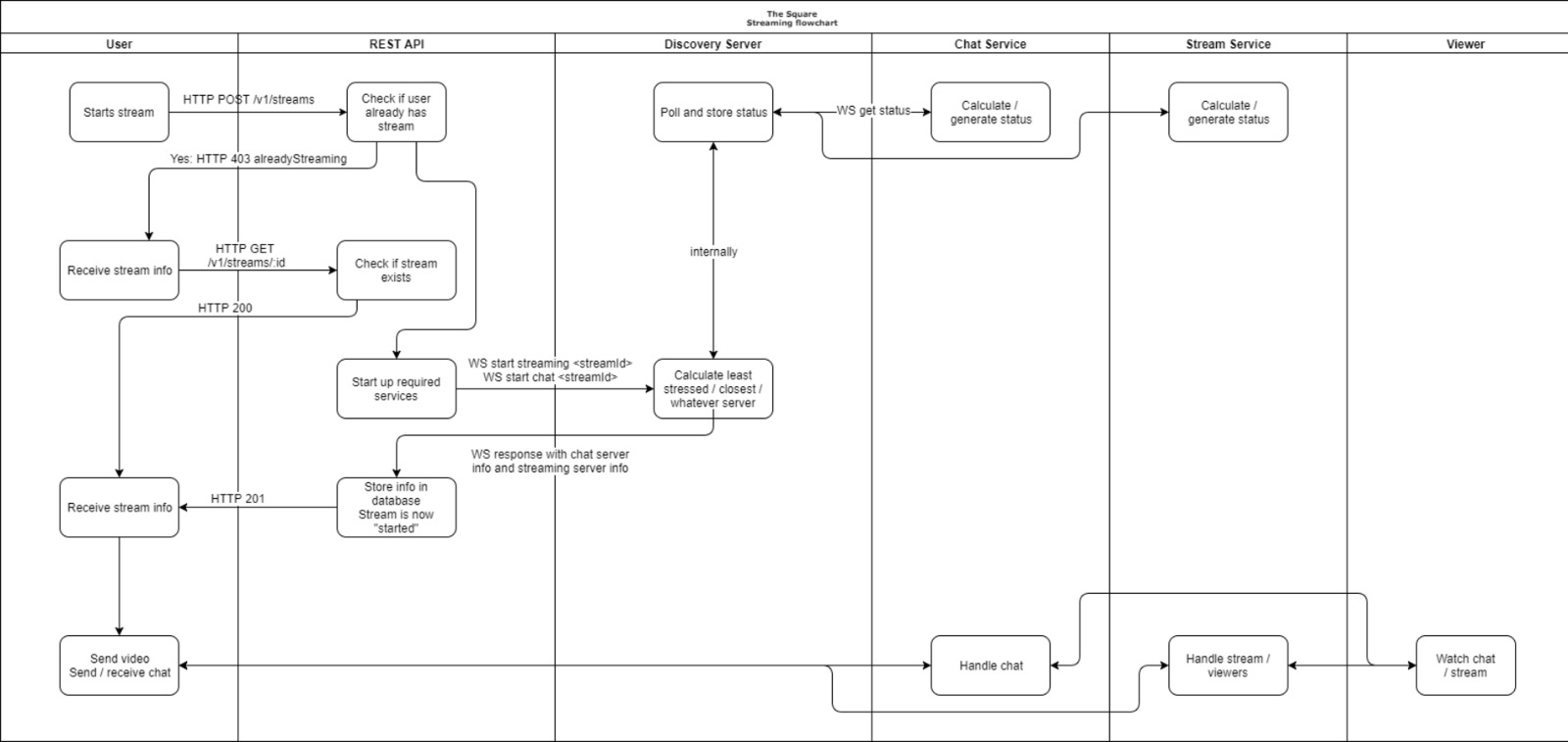


# Deployment diagram



# Implementation view

# View



# Android stream flow

Hieronder wordt de Android stream flow toegelicht:

1. Alice start de The Square applicatie op
2. Bij eerste gebruik moeten de juiste permissions gegeven worden
3. Vervolgens kan Alice een username ingeven en wordt er een private en public key gegenereerd
4. Alice kan de stream nu starten
5. Op het moment dat de stream gestart is, word er via de NGINX een post gedaan naar de rest api
6. De api vraagt aan de discovery service welke streaming en chat server gebruikt mag worden
7. Vervolgens geeft de discovery service dit door naar de android client
8. De android client staat nu in directe verbinding met de streaming en chat servers
9. Vervolgens gaat android een GET request doen naar de API voor de viewcount
10. Als Alice klaar is met de stream kan zij die afsluiten
11. De streaming server stuurt naar de API dat de stream klaar is
12. Hierna word de satoshi berekend via de streaming server aan de hand van de gestreamde tijd
13. De berekende satoshi word in de database opgeslagen
14. Alice kan in haar profiel kijken hoeveel satoshi ze heeft verdient

# Web App stream flow

Hieronder wordt de WebApp stream flow toegelicht:

1. Bob gaat naar thesquare.me en voert een username in
2. De eerste keer word een private en public key pair gemaakt
3. De gehashde public key word opgeslagen in de database.
4. Wanneer dit niet de eerste keer is checkt de API of de user valid is
5. Na verificatie kan Bob een stream kijken
6. Wanneer Bob een stream aan klikt word er naar de API een GET request gestuurd
7. De API stuurt dit door naar de discovery en deze bepaald op welke stream en chat server de content staat
8. De web app maakt nu rechtstreekse connectie met de stream en chat server en Bob kan nu de content bekijken en chatten

# Scalability

The Square maakt gebruik van microservices. Hierdoor zijn de API, Streaming en Chat servers makkelijk uitbreidbaar. Als er nood is voor een extra server door bijvoorbeeld overbelasting dan kan deze binnen vijf minuten een nieuwe server beschikbaar gesteld worden.

# FSD (fragmented signed data) protocol

Het FSD protocol kan gebruikt worden voor het overdragen van data tussen client en server. Het data fragment is altijd signed door gebruik te maken van een handtekening om data integriteit te waarborgen. Geen handdruk is bepaald, de client praat direct met de server. Als een handdruk is vereist, dan kan je een andere protocol gebruiken. 