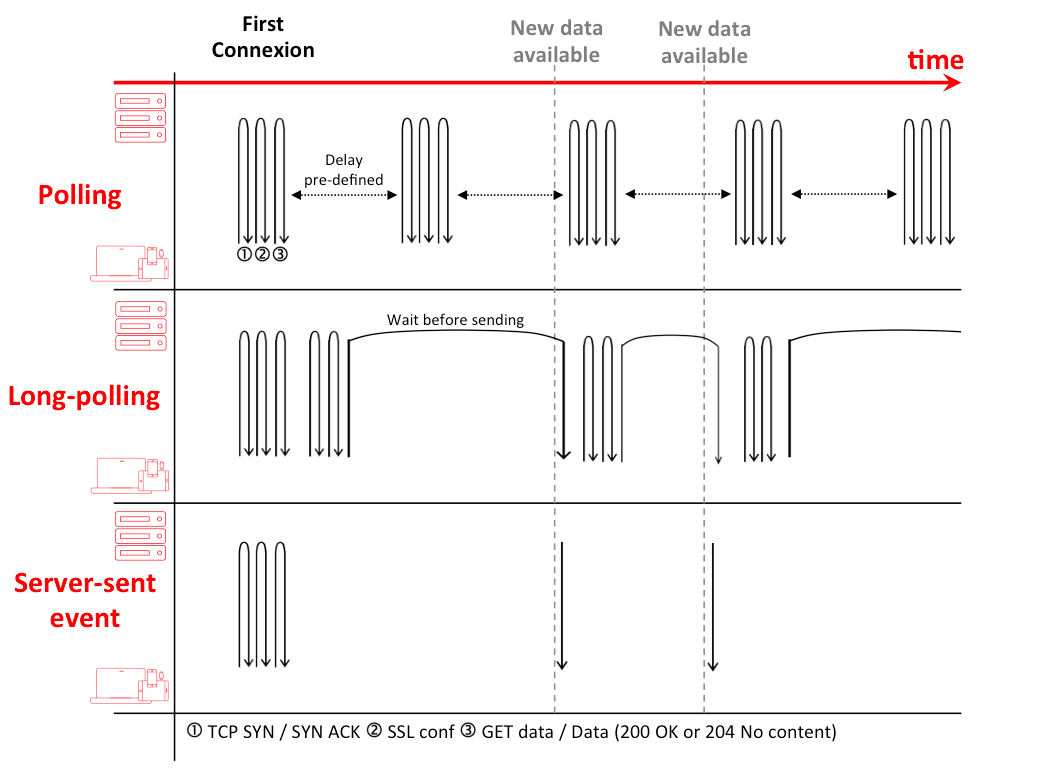
Communicatie Service in .NET Core

# SignalR

ASP.NET Core SignalR is een library die het toevoegen van real-time updates in applicaties heel erg vereenvoudigt. Het kan gebruik maken van polling, SSE of websockets en bepaald zelf per verbinding welke van deze opties het beste is om te gebruiken.

Er is een .NET Core server library beschikbaar en verschillende client libraries zoals Javascript, Typescript en C#.



Nadelen van de verschillende protocollen:

Short polling:

* Geen real time updates

Long polling:

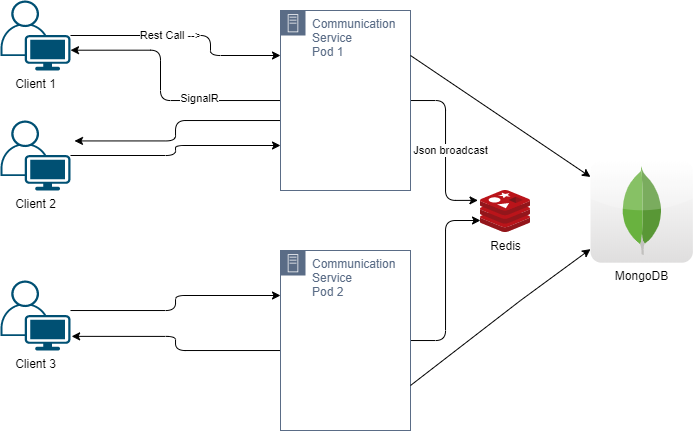
* Zwaar voor de server omdat per user er een request op de server nodig is

Websockets:

* Maakt geen gebruikt van HTTP wat problemen kan veroorzaken met Ingress of firewalls

## HA SignalR

Doordat een socket verbinding de backend applicatie stateful maakt kan dit problemen gaan veroorzaken wanneer de backend service opgeschaald wordt. Hieronder staat in een diagram weergegeven hoe de architectuur hiervan eruit ziet.



Wanneer client 1 een bericht wil sturen naar client 2 zal deze een REST verzoek doen met het bericht, deze zal worden opgeslagen in de MongoDB, SignalR zal een bericht naar client 2 sturen met de informatie, op dat moment zal de server ook aangeven dat dit bericht gelezen is.

Wanneer client 1 een bericht naar client 3 wil sturen kan de server deze informatie niet direct doorspelen aan client 3 omdat deze geen directe verbinding heeft met Pod 1. Hiervoor zal de server een bericht naar de Redis pub moeten sturen, deze zal dit bericht doorsturen naar alle verbonden pods en kan pod 2 deze aan client 3 sturen.

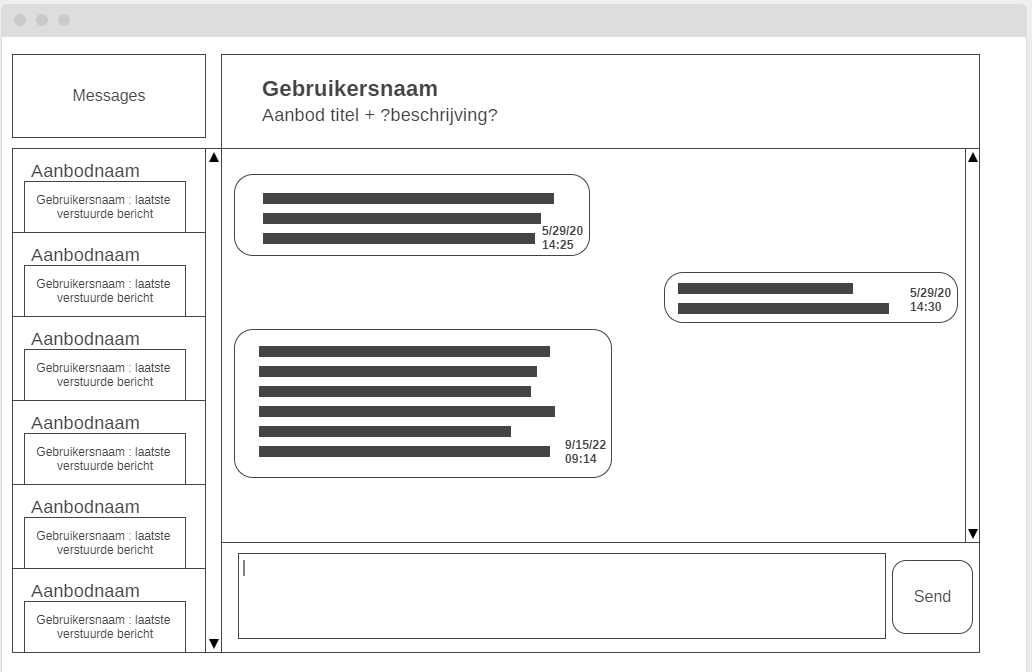
Wanneer client 1 een bericht wil sturen naar client 4 (die op dit moment niet verbonden is) zal geen elke pod dit request naar een client sturen en zal deze alleen in de MongoDB worden opgeslagen.

# 

# REST calls

|  |  |  |
| --- | --- | --- |
| Type | Name | Description |
| **GET** | **getChat/<chatId>** | Haalt de volledige chatgeschiedenis op van een specifieke chat. |
| **GET** | **getAllChatWithLastMessage/<userId>** | Haalt een overzicht op van alle chats van de gebruiker same met het laatste bericht dat verstuurd of ontvangen is in deze chat. |
| **POST** | **sendMessage** | Verstuurd een chatbericht. |
| **POST** | **confirmChatRead** | Een bevestiging die doorgestuurd wordt op het moment van het lezen van een ongelezen chat. |

# Wireframe



Hierboven is een schets te vinden waarin afgebeeld staat hoe wij de chat omgeving er uiteindelijk ongeveer uit willen laten zien.

Op deze pagina kunnen gebruikers met elkaar communiceren via chat berichten. In de lijst links van het venster wordt een overzicht gegeven van chats met verschillende gebruikers. Ook wordt hier het laatst ontvangen of verstuurde bericht in deze chat weergeven in het overzicht.

Als een chat geselecteerd wordt zal deze zichtbaar worden in het centrale venster. Hierin kunnen eerder gestuurde berichten worden ingezien. Door middel van het onderstaande tekst input veld kan een gebruiker een bericht typen om deze te versturen binnen de chat.