Chat-backend

Inhoud

[Stack & technologie 3](#_Toc106193260)

[Events 3](#_Toc106193261)

[Risico analyse 3](#_Toc106193262)

## Stack & technologie

We realiseren de chat functionaliteit doormiddel van een express JS applicatie. Met behulp van de socket.io library die gebruik maakt van de websocket protocol kunnen we deze datastream realiseren en kunnen andere applicaties de socket.io-client library gebruiken om te communiceren met de chat server. Dit leek ons de meest efficiënte optie om deze functionliteit te realiseren.

## Events

Luisterd naar:

* joinStream: message -> gebruiker joint een stream
* chatMessage: message -> gebruiker verstuurt een chat bericht in de gejoinde stream
* disconnect: x -> gebruiker verlaat socket
* disconnectUserFromStream: message -> gebruiker verlaat een stream

Stuurt:

* message
* streamUsers

## Risico analyse

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risico | Kans | Impact | Kans \* impact | Maatregel |
| Stack veranderd | 3 | 2 | 6 | Plan uitdenken en basis opbouwen om te weten of het mogelijk is om aan alle security goals te voldoen en aan de functionaliteitseisen. |
| Stack van andere applicaties veranderd | 3 | 0.5 | 1.5 | Door het single responsibility principe toe te passen voorkomen we dat onze applicatie aangepast moet worden. Autonomie is ook erg belangrijk hierbij. |
| Lastige componenten die veel tijd kunnen kosten e.g. signature | 3 | 1 | 3 | Eerst zelf uitzoeken maar als het niet lukt, technisch advies vragen aan anderen. |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |