deti universidade de aveiro
departamento de eletrónica, telecomunicações e informática

CrowdWire

<!-- MASSIVE ONLINE MEETINGS -->

<>

OUTSIDE AT HOME

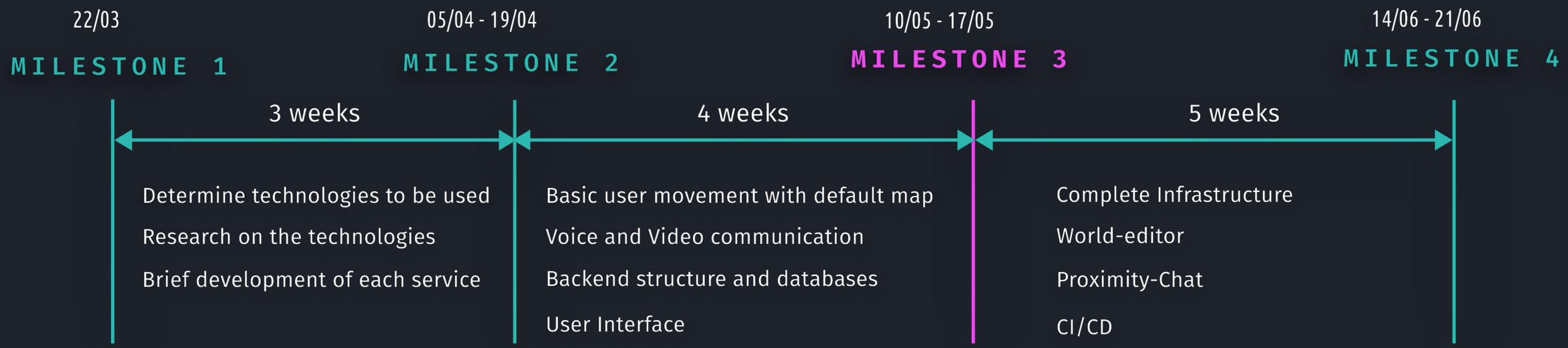
</>

LICENCIATURA EM ENGENHARIA INFORMÁTICA

PROJETO DE INFORMÁTICA

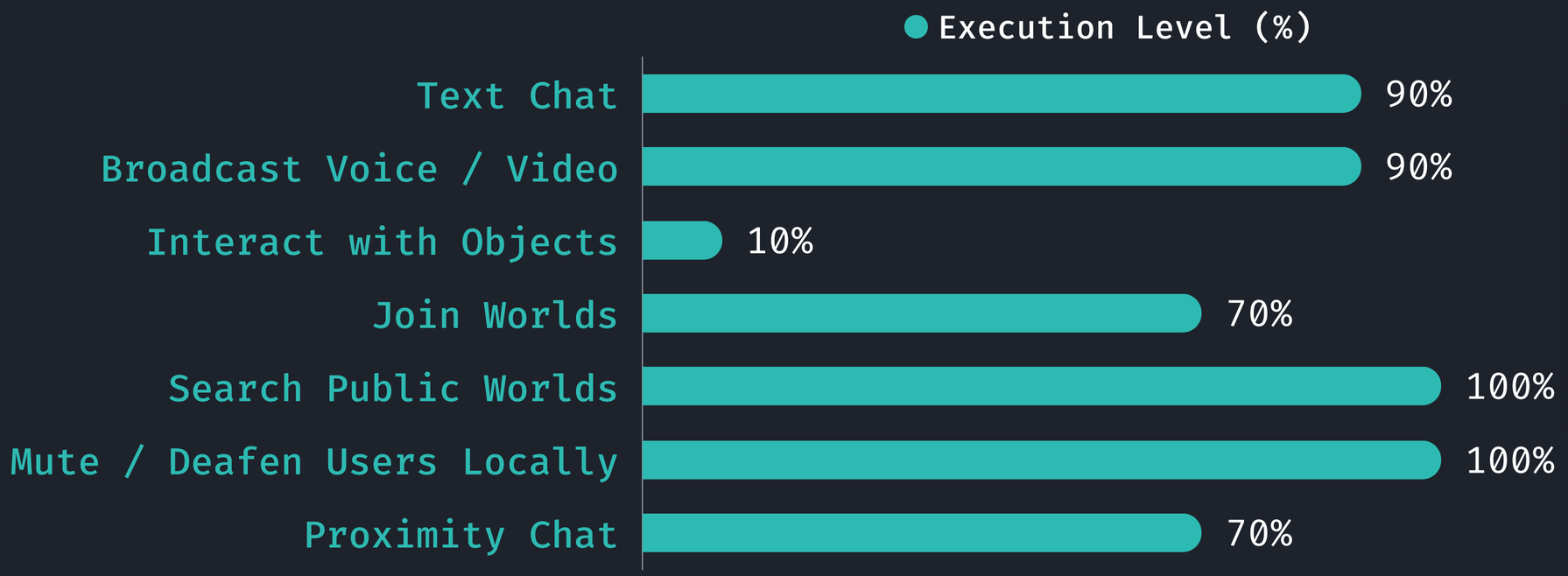
GRUPO 09

INITIAL PROJECT CALENDAR



- Milestone 1: Presentation of the lifecycle objectives and calendar for the Project
- Milestone 2: Presentation of the lifecycle Architecture; Validation of the Architecture
- Milestone 3: Prototype; Mid-Term Presentation with supervisors; Peer Evaluation.
- Milestone 4: Project Presentation; All Functionalities have been developed.

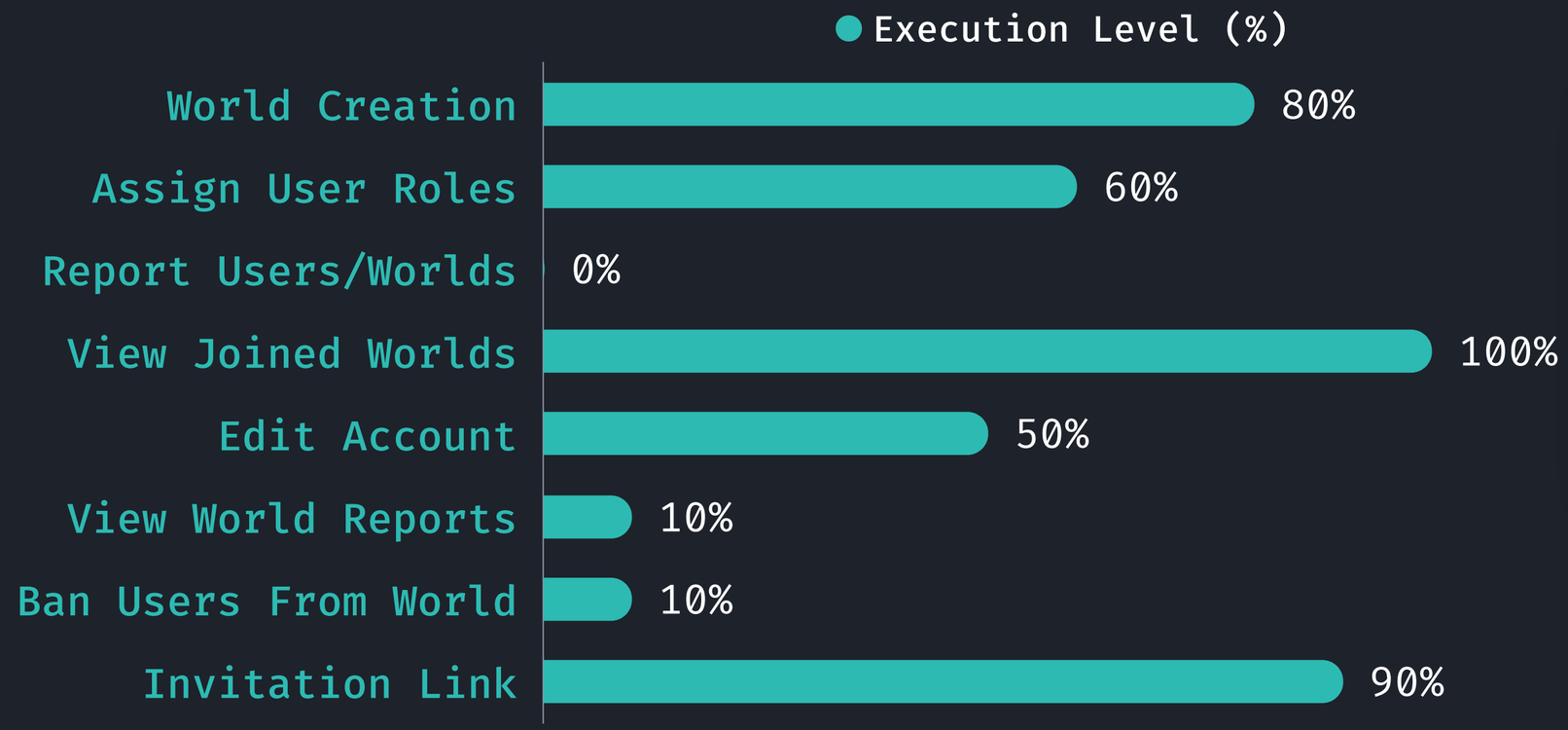
Requirements Execution Level（Guest)



Notes:

- The scale is from 0% to 100%.
- These values do not reflect the workload of the rearrangement to a replication infrastructure.

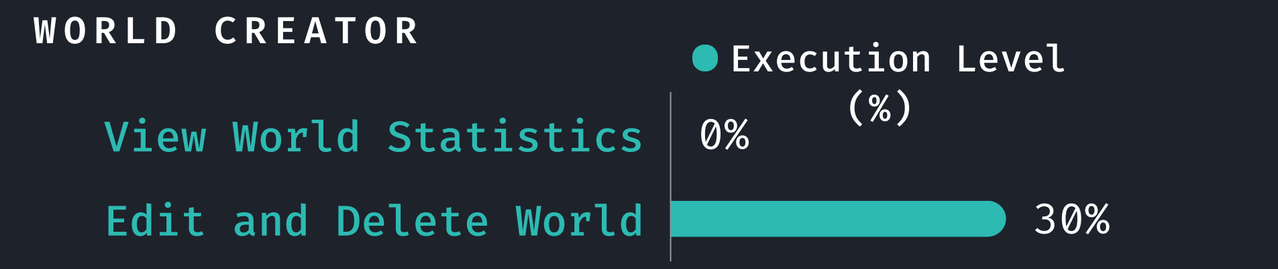
Requirements Execution Level (Registered User)

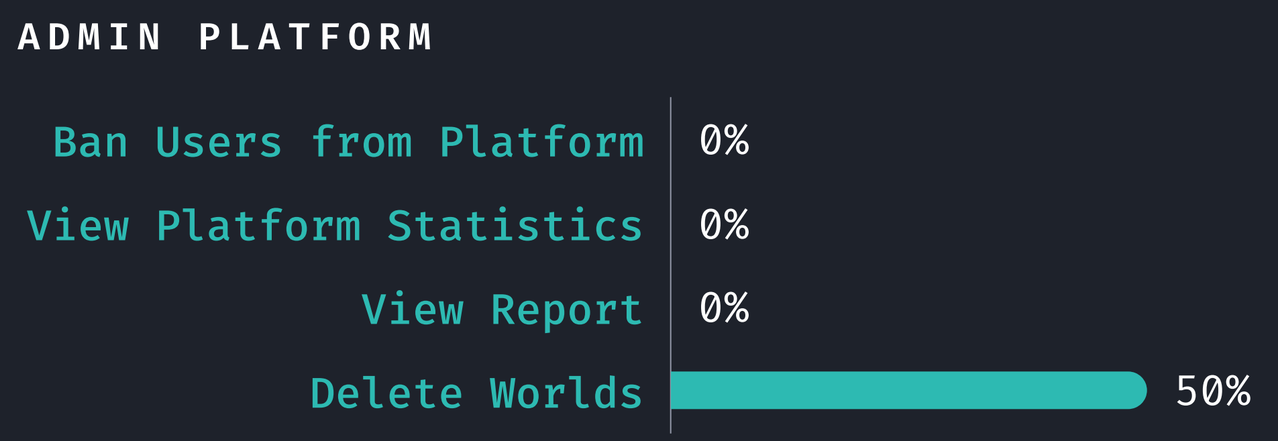


Notes:

- The scale is from 0% to 100%.
- These values do not reflect the workload of the rearrangement to a replication infrastructure.

Requirements Execution Level (Admin&World Creator)

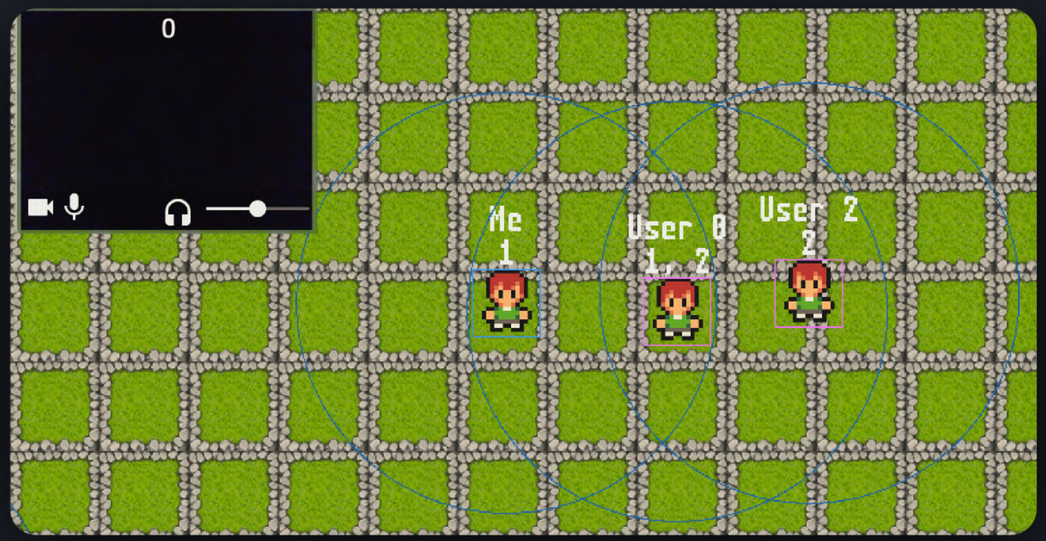


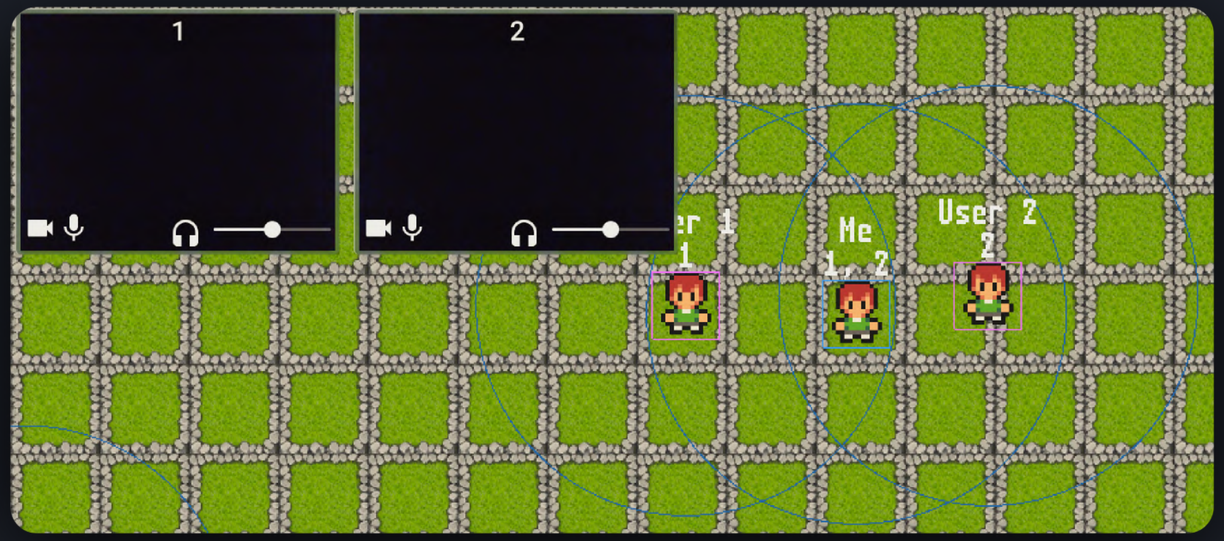


Notes:

- The scale is from 0% to 100%.
- These values do not reflect the workload of the rearrangement to a replication infrastructure.

Proximity chat example





Scalability Issues PROXIMITY CHAT

Client:

Video Upload:

Min: 0.25 MBps / Max: 0.375 MBps

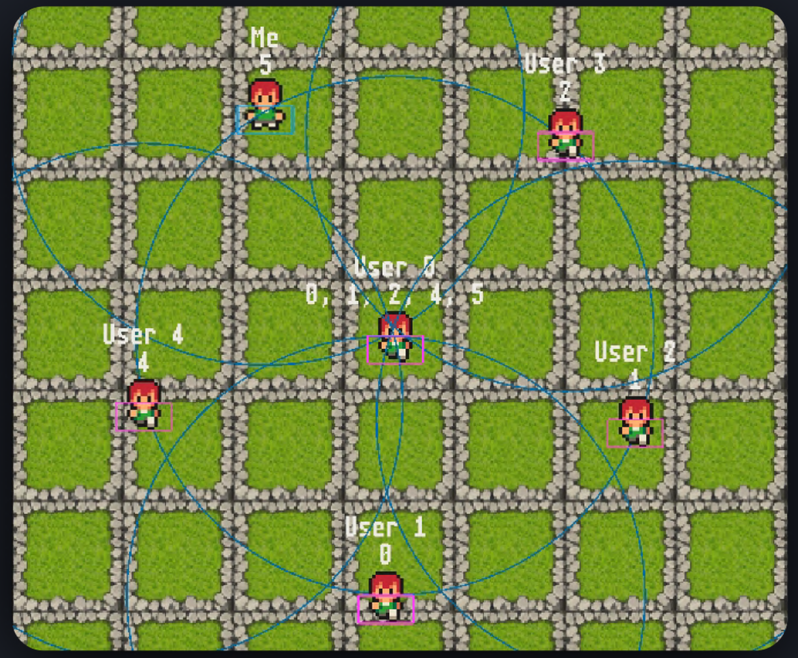
Audio Upload:

Min: 0.013 MBps / Max: 0.1 MBps

Worst Case Scenario

Client in 5 groups uploading audio and video with 10% offset:

(0.375 + 0.1) × 1.1 × 5 ≈ 2.61 MBps required for upload speed

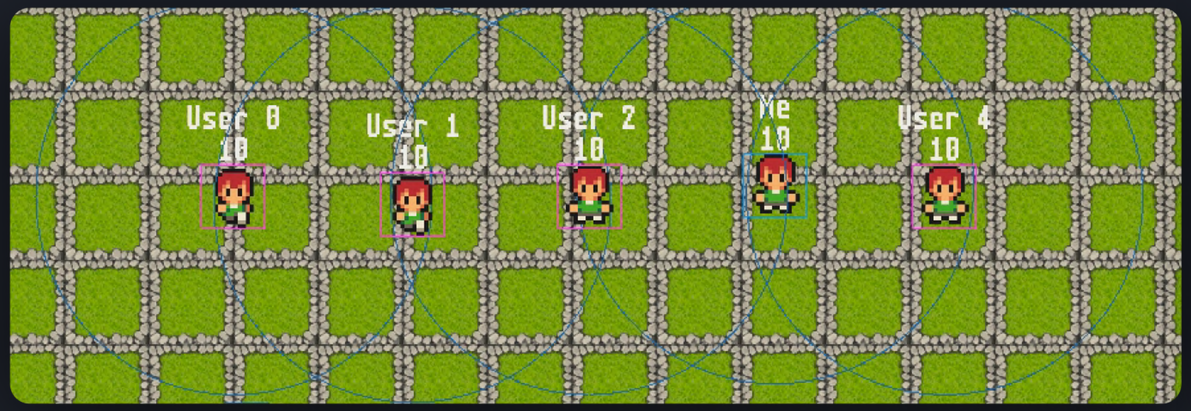


Scalability Issues PROXIMITY CHAT

Since bandwidth usage was not too high by being in more than one group, we decided to allow a user to be in multiple rooms at the same time.

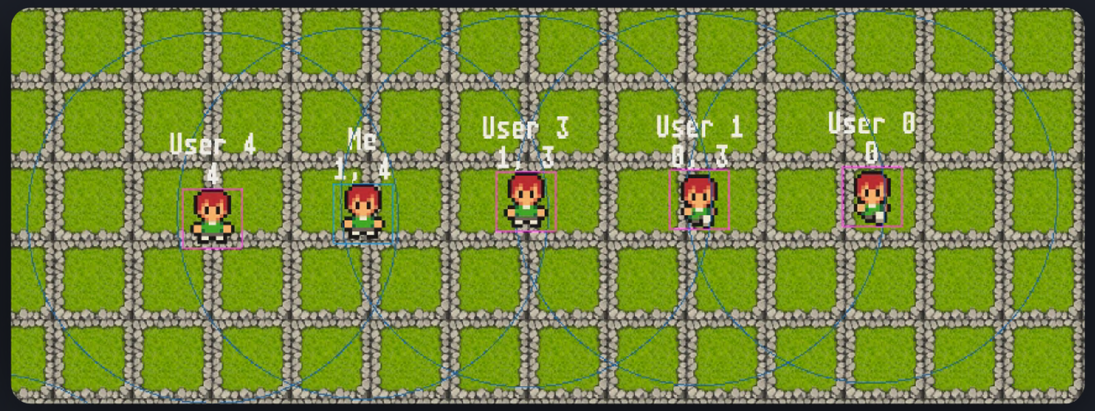
This allows the media server to scale easier because if all users were in a straight line, the user in one end would be in the same call as the user in the other end, possibly the whole world could be within a single call.

SCENARIO AVOIDED



Users restricted to one group

CURRENT SCENARIO



Users split through groups

Scalability Issues PROXIMITY CHAT

Backend

- Have a counter that when close to the limit of users per media server creates replicas.
- Following messages are sent to the new replica's queue.
- Queues are named after the replica.

VS

- Have pre-built replicas available through *Kubernetes.*
- Distribute requests throughout the available replicas according to the number of consumers.

Media Server

- Scale locally before creating another replica.
- Requires piping producers from one router to another.

Formula to calculate the number of Consumers:

$2x^2 - 2x$, x: Number of users producing both audio and video

Scalability Issues WEBSOCKETS

Websockets provide a stateful connection between the backend and the frontend. In order for the backend to communicate with every user in a given world it is required that the user has a way of sending messages to that websocket.

LOAD-BALANCING CONNECTIONS BASED ON WORLD

- Each replica handles every user of a set of worlds.
- Load-balancing is harder to manage.
- Connection distribution might be not uniform.
- Communication between users in the same world is easier.

VS

PUB/SUB COMMUNICATION BETWEEN REPLICAS

- Each replica handles a set of users from any world.
- Load-balancing does not need to be changed.
- Each replica handles a similar number of connections.
- Communications between users in the same world go through a pub/sub channel.

Infrastructure

1 - Kubernetes resources are assigned with a unique IP, that can be accessed through a DNS Name of a resource called Service.

2 - Each Kubernetes LoadBalancer can only redirect UDP or TCP traffic

Issues:

1 - Our microservices need to access each other in order to communicate, but how can we do it if IPs can always change?

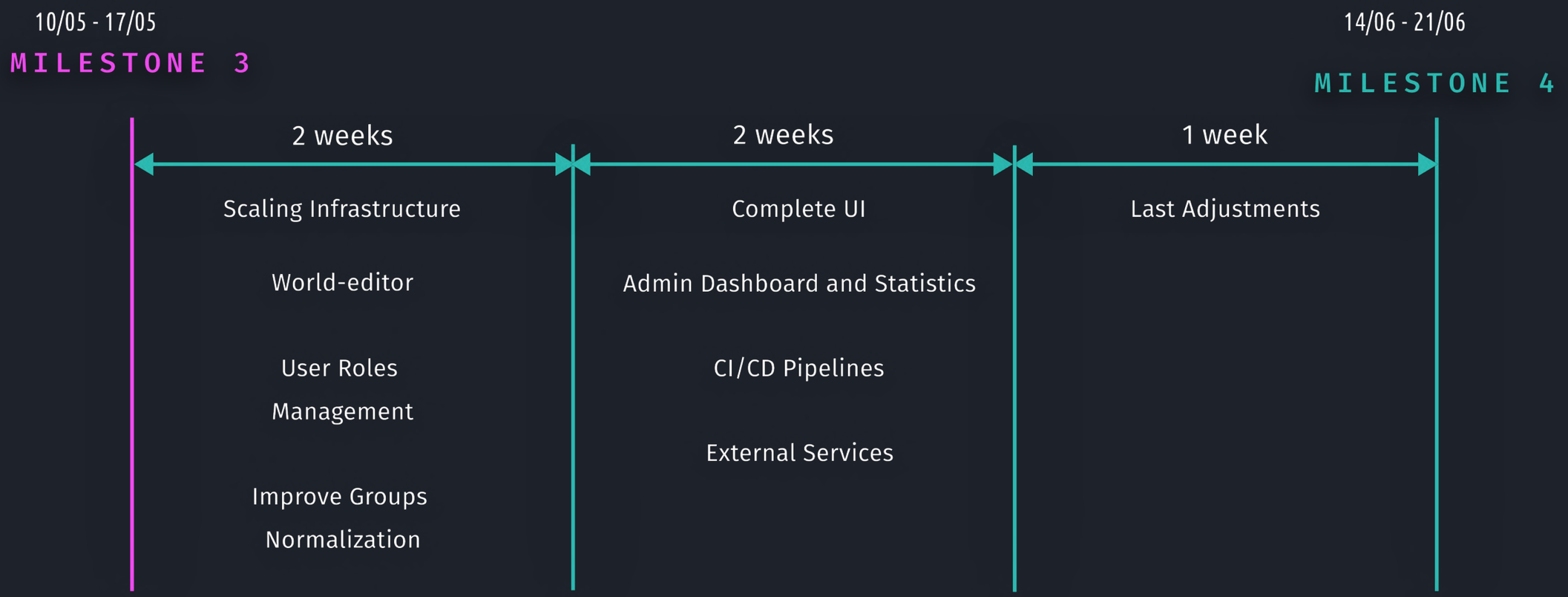
2 - MediaServer needs to send/receive both TCP/UDP Packets.

Solutions:

1 - Dynamic Env. Variables Stored inside Containers with the Services DNS Names.

2 - Creation of two LoadBalancers, one for TCP and the other for UDP.

CURRENT PROJECT CALENDAR



- Milestone 3: Prototype; Mid-Term Presentation with supervisors; Peer Evaluation.
- Milestone 4: Project Presentation; All Functionalities have been developed.

DEMO

https://crowdwire.duckdns.org