

Abstract geometric lines in the top-left corner of the slide, consisting of several thin black lines forming various polygons and intersecting each other.

KARAOKE

COMPUTAÇÃO DISTRIBUÍDA

Tomás Matos 108624

Rodrigo Graça 107634

ARQUITETURA

Fui usado uma Server-Cliente, onde os diversos servidores, Workers, que estiverem disponíveis recebem as mensagens do Cliente, api, e processam a música

WORKERS

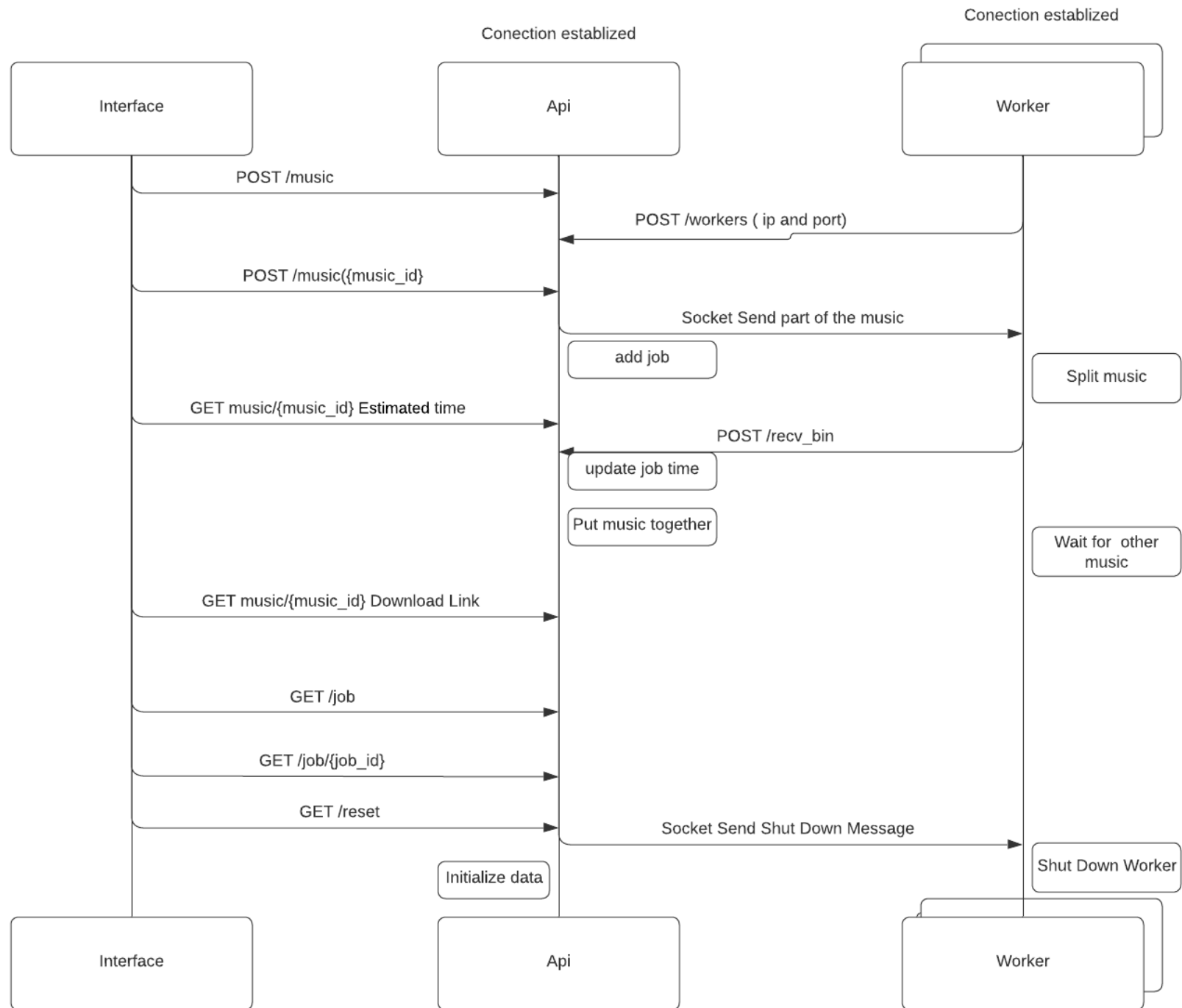
Pensamos em usar os Workers como servidores por causa da tolerância a falhas. Se um Worker falhar, outros podem assumir o seu trabalho e o sistema não falha.

WORKER->API

Os Workers após serem inicializados postam através de HTTP o seu ip e a sua porta, de modo a poderem ser contactados pelo cliente. O processo Worker-API acontece através de HTTP

API->WORKER

No envio da música da Api (Cliente) para o Worker são utilizadas Sockets



RESULTADOS OBTIDOS

- As músicas estão a ser divididas por diversos Workers e a ser concatenada no final, podendo cada Worker estar num computador diferente.
- Tentamos cumprir as mensagens indicadas no Swagger.
- Foi implementada uma Queue, que armazena as músicas enquanto outra está a ser processada.
- Quando um Worker falha o processamento, este volta a tentar processar, se falhar 5 vezes (número escolhido aleatoriamente) a música é mandada novamente da Api para o determinado Worker voltar a tentar.
- Reset durante o split em instrumentos de uma música, foram usadas Threads para fazer este fim.



CONCLUSÃO

Pensamos que os requisitos foram minimamente cumpridos, apesar da tolerância a falhas não estar 100% completa