

Relatório Projeto SO

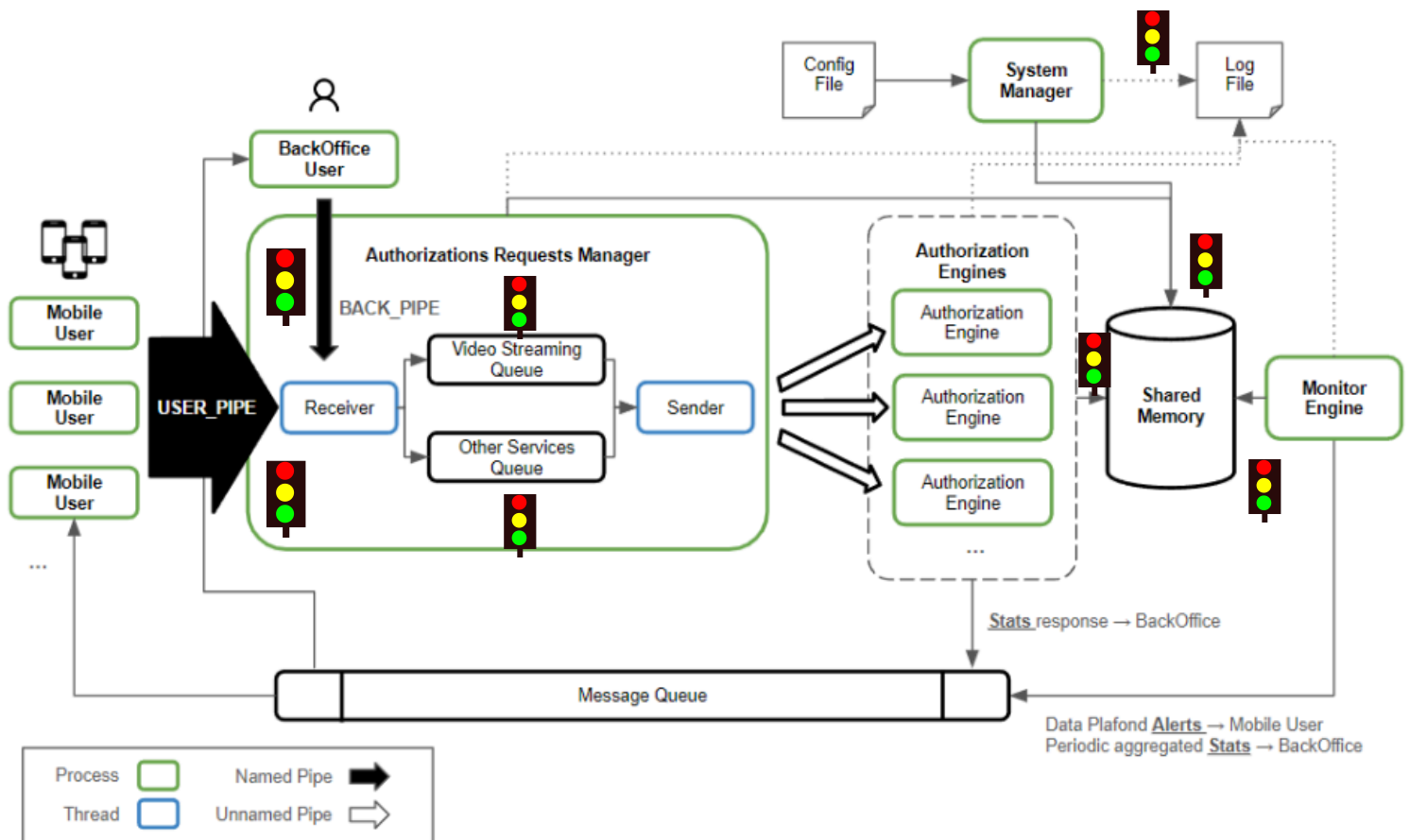


Figura 2: Arquitetura técnica do simulador

Mecanismos de Sincronização

Mobile User & Backoffice User:

Utilizamos um semáforo para controlar o envio de mensagens para a thread receiver que depois as recebe e as coloca na fila correspondente (video queue ou other queue), ou seja, o semáforo é bloqueado, escreve de seguida a mensagem no USER_PIPE ou BACK_PIPE e quando esta é enviada liberamos o semáforo sem_post.

Authorizations Request Manager:

Para controlar o acesso das threads receiver e sender às filas, utilizamos um mutex que sempre que cada uma destas threads tentar acessar uma fila, o mutex é trancado, após acessarem as filas, o mutex é destrancado. E porque é que usamos isto? Ao usarmos isto conseguimos garantir que apenas uma thread, de cada vez, possa acessar as filas, prevenindo assim corrupção e possíveis erros de dados.

System manager para o ficheiro logs:

Usamos um mutex dentro da função do `sent_to_logs()` para garantir a sua proteção, isto é, garantimos que apenas uma thread ou processo de cada vez escreva no ficheiro log os dados que precisa. De modo a prevenir corrupções de dados.

Acesso à shared memory:

Usamos um semáforo sempre que acessamos a shared memory de modo a não causar conflitos entre os dados, apenas podendo uma thread ou processo aceder à memória de cada vez.

André Albuquerque uc2022231505

Rodrigo Borges uc2022244993