Trabalho Final
Yet Another Message Broker!

Aluno: Guilherme Avelino do Nascimento
Email: guiavenas@gmail.com
DRE: 117078497

Este documento contém o projeto da aplicação Yet Another Message Broker!, que é um sistema de publicação e subscrição de mensagens. O YAMB pode ser utilizado para integração entre sistemas e componentes de software, sem a necessidade de conhecimento mútuo entre esses componentes(desacoplamento). As seções a seguir descrevem os componentes principais e as formas de interação com a aplicação.

Seções:

1. Interface via linha de comando
2. Arquitetura de software
3. Arquitetura de sistema
4. Protocolo e formato do envelope das requisições

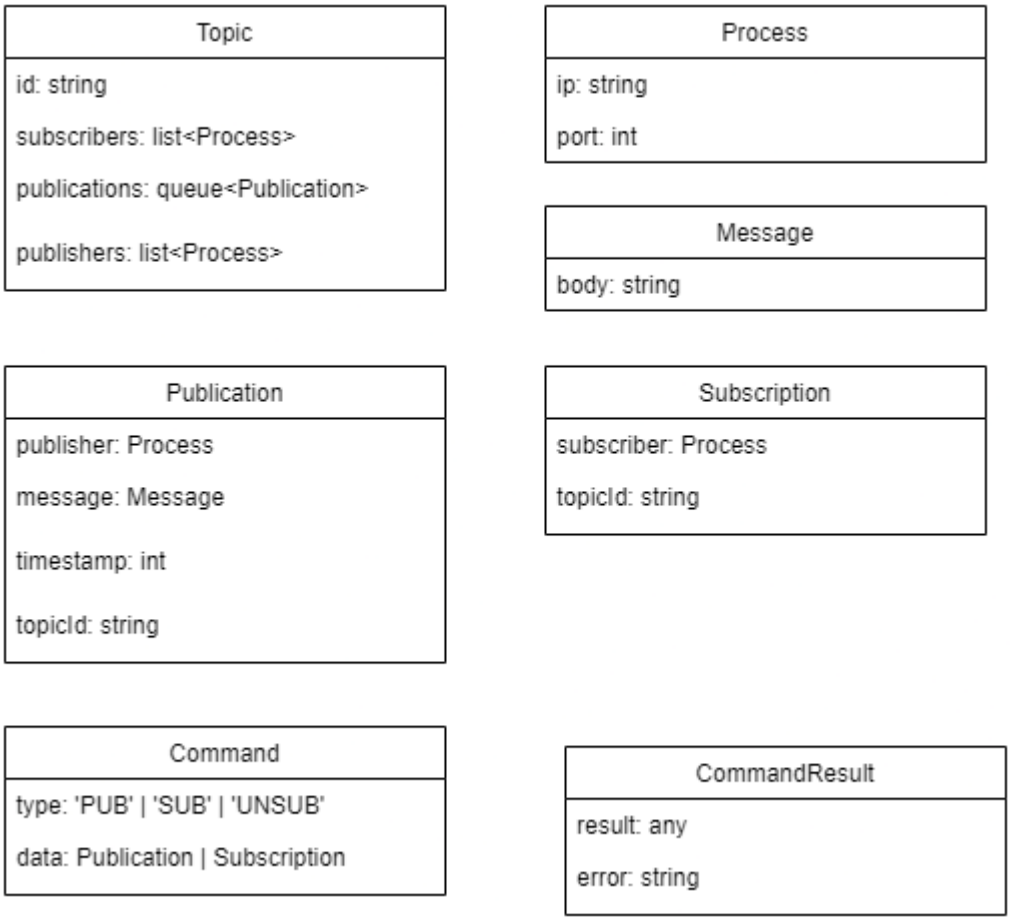
1 - Interface via linha de comando

Os seguintes comandos poderão ser enviados via linha de comando:

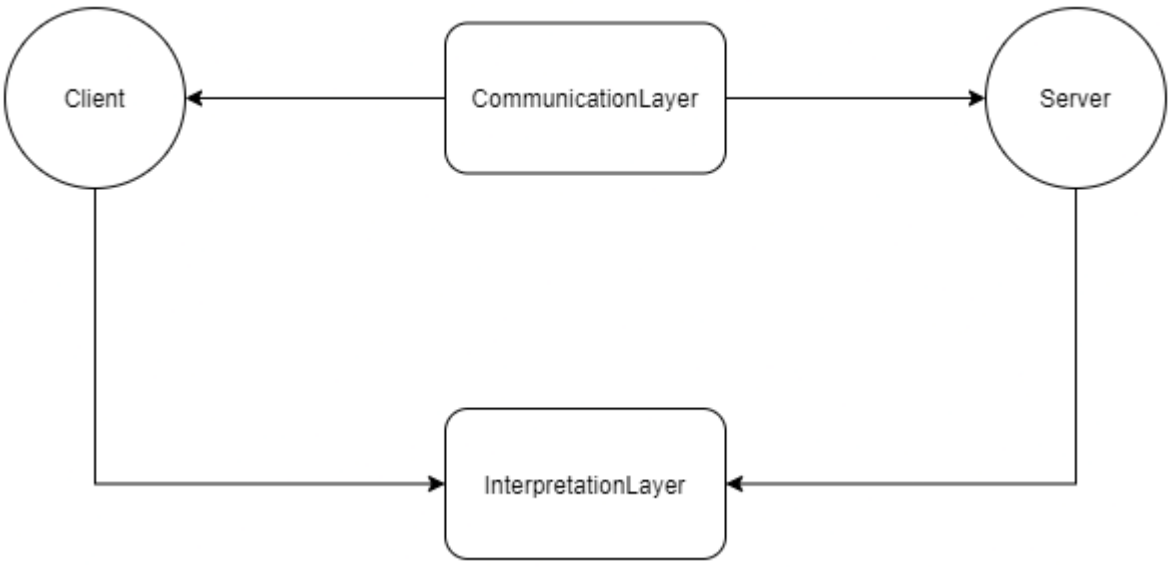
- **PUB {topicId} {message}**: Publica uma mensagem em um tópico
- **SUB {topicId}**: Cria a subscrição em um tópico
- **UNSUB {topicId}**: Remove a subscrição em um tópico
- **CREATE {topicId}**: Cria um tópico.
- **DELETE {topicId}**: Deleta um tópico.
- **LIST**: Exibe a relação dos tópicos criados.

2 - Arquitetura de software

As principais classes/ abstrações utilizadas serão as seguintes: Topic, Process(representando um produtor/consumidor), Publication(representando a publicação de uma mensagem em um tópico), Subscription(a subscrição de um consumidor à um tópico) e Message(os detalhes e o corpo da mensagem em si). Além disso, as requisições(PUB, SUB e UNSUB) serão enviadas para o servidor encapsuladas nas seguintes classes: Command(requisição que irá trafegar entre processos) e CommandResult(resultado das requisições). As propriedades de cada uma das classes estão descritas no seguinte diagrama:

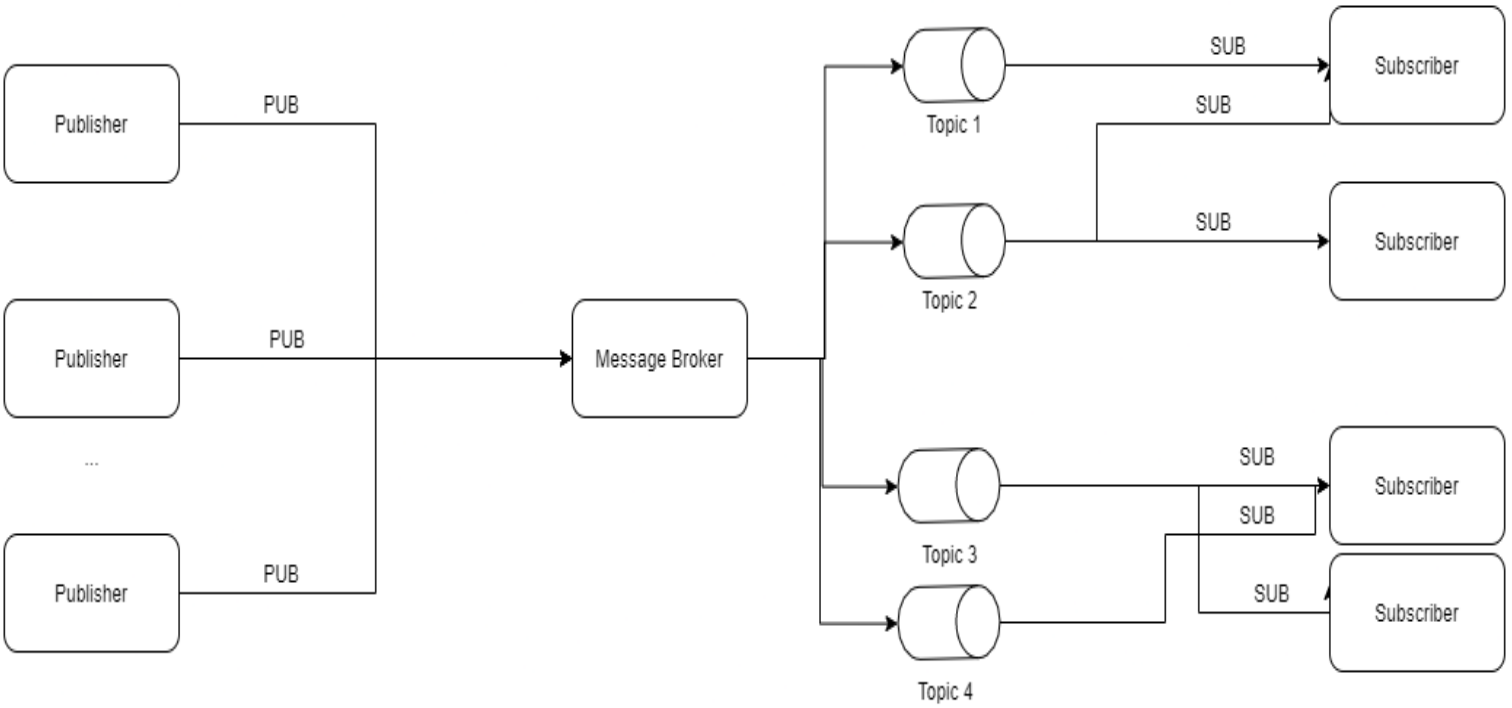


Seguindo o que foi utilizado no trabalho anterior(bate papo distribuído), os processos cliente e servidor utilizarão middlewares tanto para a interpretação dos comandos utilizados via linha de comando(InterpretationLayer), quanto para a comunicação inter-processos via rede(CommunicationLayer), como na figura a seguir:



3 - Arquitetura de sistema

Como padrão na maioria dos sistemas pub-sub, a arquitetura de sistema utilizada será a cliente-servidor. Os clientes podem enviar ao servidor as requisições de publicação e subscrição em tópicos, e o servidor ficará responsável por armazenar as subscrições e os tópicos e entregar as publicações corretamente a todos os processos interessados.
A figura a seguir representa de maneira geral a arquitetura do sistema:



A aplicação pode ser utilizada tanto localmente, iniciando os processos dos clientes e do message broker na máquina local, ou em uma plataforma de hosting remoto(como a AWS, por exemplo), desde que tenha um endereço IP acessível para a Internet.

4 - Protocolo e formato do envelope das requisições

A comunicação entre os processos será feita utilizando diretamente a API de sockets do sistema operacional, utilizando o protocolo TCP.

De maneira similar ao trabalho anterior, o envelope das mensagens enviadas segue a seguinte estrutura: os primeiros 64 bytes carregam o tamanho N da requisição, representado em base 10, e os próximos N bytes carregam o corpo da requisição, representado na estrutura JSON(com os mesmos campos das classes Command e CommandResult, indicados na seção 2).

Os comandos apresentados na seção 1 definem o protocolo da aplicação, e poderão ser dos seguintes tipos:

Executados pelas aplicações cliente:

- PUB
- SUB
- UNSUB

Executados pelas aplicações servidor:

- CREATE
- DELETE
- LIST