

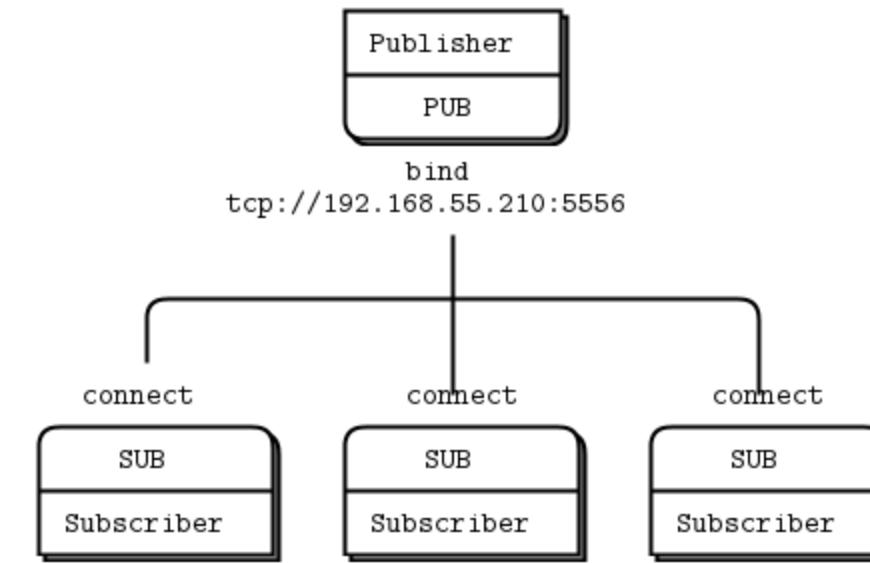
# Publish-subscribe

# Exemplos de PubSub

- Robot Operating System (ROS)
- Padrão de filas no RabbitMQ
- Apache Kafka pode ser usado dessa forma
- GCP Pub/Sub

# Publish-subscribe vs Request-reply

- **Request-reply:** cliente envia mensagem (request) e servidor responde (reply)
- **Publish-subscribe:** publisher publica uma mensagem e subscriber recebe a mensagem



# **Pub-sub**

- A mensagem é enviada de um (publisher) para todos (subscribers)
- Voltado para escalabilidade:
  - Aplicado para grande volume de dados que pode ser enviado para várias aplicações
  - Publisher envia mensagem, subscriber lê mensagem, nenhuma outra troca de mensagens acontece (back-chatter)
- Para ficar mais fácil de escalar, a mensagem do publisher é enviada para grupo de multicast e subscribers se conectam a este grupo

# Problemas do formato mais simples de pub-sub

- Publishers não sabem se subscribers estão conectados, se reconectaram ou pararam de funcionar
- Subscribers não conseguem controlar a velocidade que recebem as mensagens

Pub-sub is like a radio broadcast; you miss everything before you join, and then how much information you get depends on the quality of your reception.

<https://zguide.zeromq.org/docs/chapter5/>

# **Problemas para o Pub-sub ser confiável**

- Conexão tardia do subscriber, perdendo mensagem
- Demora para leitura de mensagens pelo subscriber, perdendo mensagens
- Subscriber perde conexão, perdendo mensagens
- Subscribers pode travar, perdendo mensagens e os dados que possuía
- Rede pode ficar sobrecarregada e perder dados
- Rede pode ficar devagar, causando overflow na fila do publisher e possível travamento

# Para testar

- Verificar os arquivos na pasta `src/pubsub`
- O `Dockerfile` é o mesmo da aula anterior
- Execução: `docker compose up`

# Para testar com proxy

- Verificar os arquivos na pasta `src/proxy`
- O `Dockerfile` é o mesmo dos outros exemplos
- Execução: `docker compose up`

# Testes

O que podemos testar:

1. Subscriber demora para conectar
2. Publisher não espera para publicar
3. 1 publisher e N subscribers
4. N publishers e 1 subscriber
5. N publishers e N subscribers
6. Uma mensagem pode ser publicada em um tópico. Como isso é feito no ØMQ?