UNIFESSPA
UNIVERSIDADE FEDERAL DO SUL E SUDESTE DO PARÁ

Universidade Federal do Sul e Sudeste do Pará

Sistemas Distribuídos

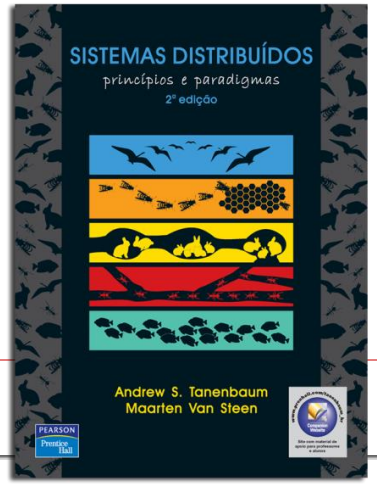
Prof.: Warley Junior
wmvj@unifesspa.edu.br

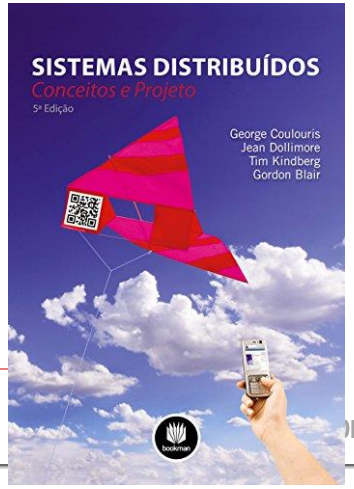
Agenda

- AULA 9:
- Middlewares
  - MOM (Middleware Orientado a Mensagem)

Leitura Prévia

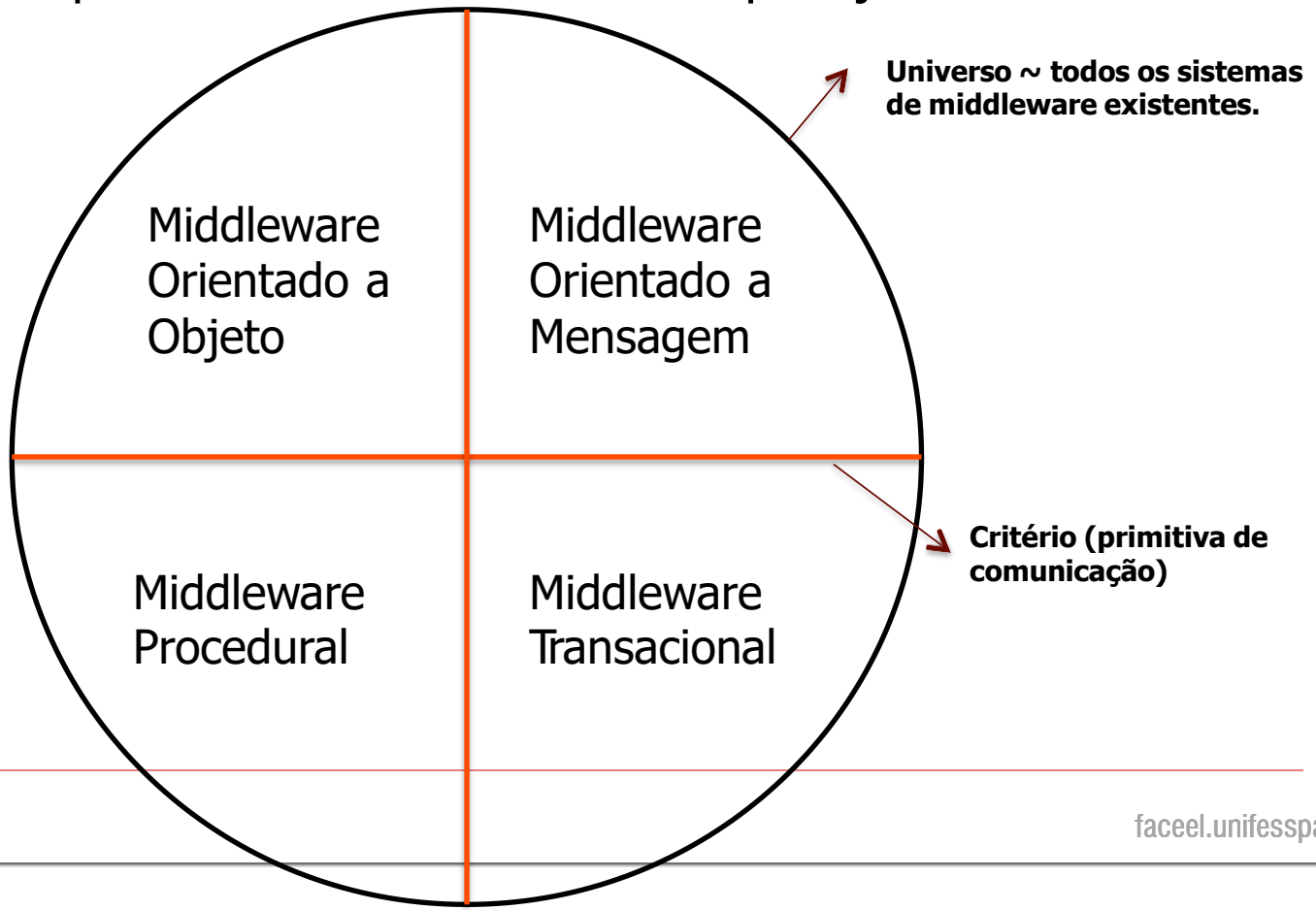
- COULOURIS, George. Sistemas distribuídos: conceitos e projetos. 5ª ed. Porto Alegre: Bookman, 2013.
  - Capítulo 6.
- TANENBAUM, Andrew S. Sistemas distribuídos: princípios e paradigmas. 2ª ed. São Paulo: Pearson Prentice Hall, 2007.
  - Capítulo 4.





Middlewares - Classificação

- Critério: tipo de primitiva de comunicação fornecida pelo middleware para o desenvolvimento de aplicações distribuídas.



Middleware Orientado a Mensagem (MOM)

Comunicação orientada a mensagem

- Middlewares RPC e RMI podem ser inadequados.
- Receptor sempre acordado?
- O comportamento de sincronismo e bloqueio pode ser inadequado em ambientes com dispositivos voláteis
  - RSSF
  - IoT
  - Computação Ubíqua e Pervasiva

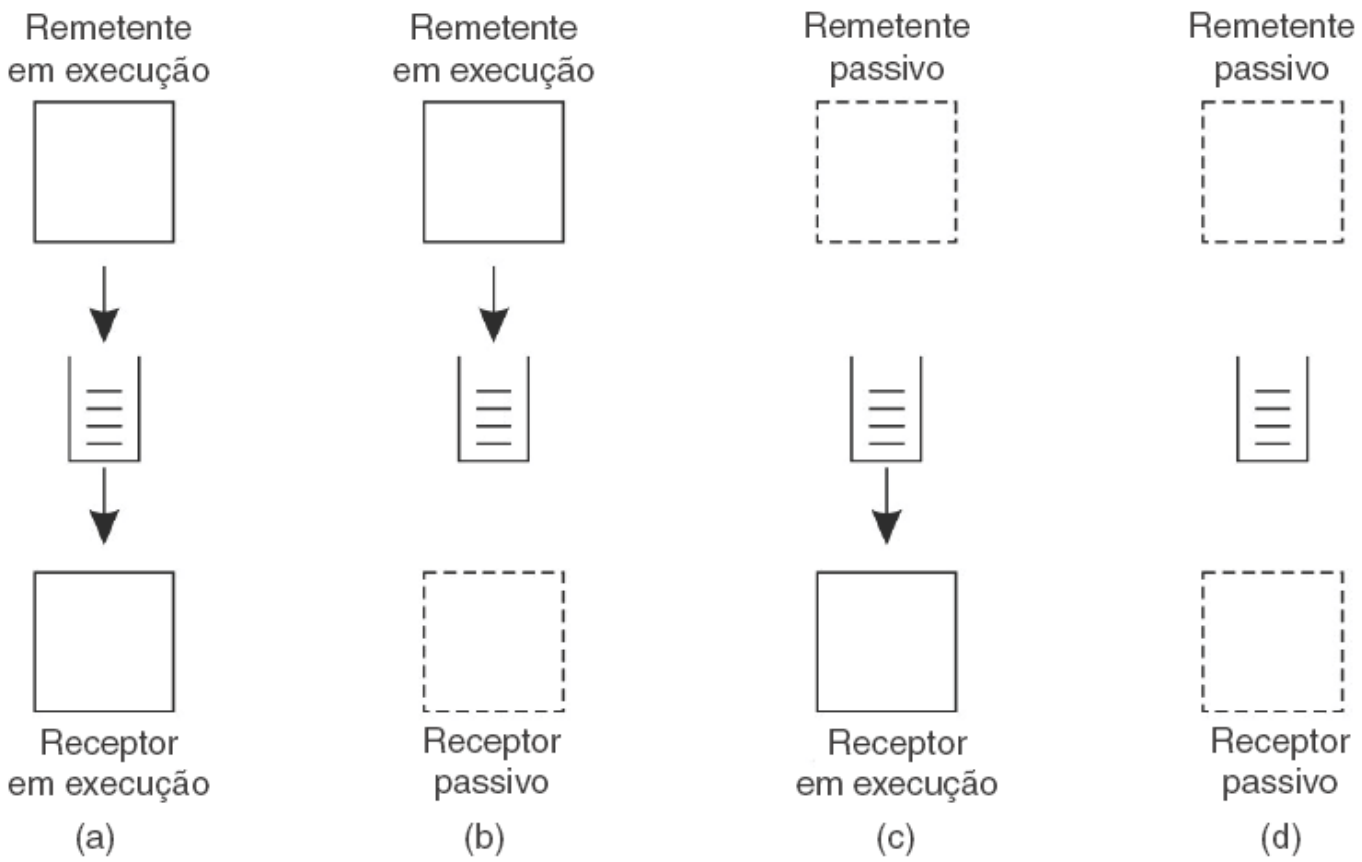
Middleware Orientado a Mensagem (MOM)

- Suporte a comunicação assíncrona e persistente
- Capacidade de armazenamento de médio prazo para mensagens trocadas
- Ideia básica: Aplicações se comunicam retirando e inserindo mensagens em filas específicas
- Mensagem será eventualmente entregue ao receptor
- Comunicação fracamente acoplada

MOM: Ideia básica

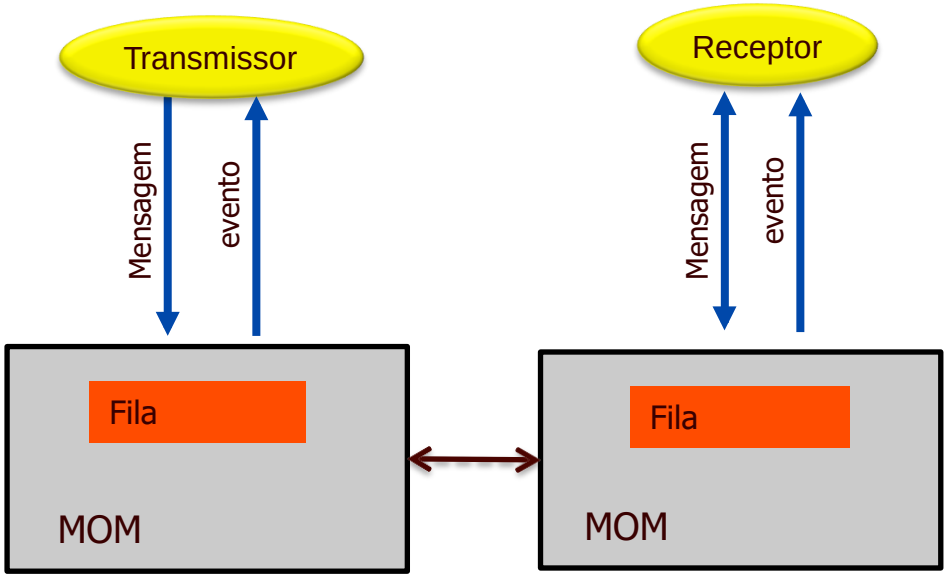
- Aplicações se comunicam inserindo mensagens em filas específicas
- As mensagens são repassadas por uma série de servidores de comunicação
- Essas são entregues ao destinatário mesmo que ele não esteja em funcionamento
- Exemplo: e-mail

Quatro combinações no modo de execução

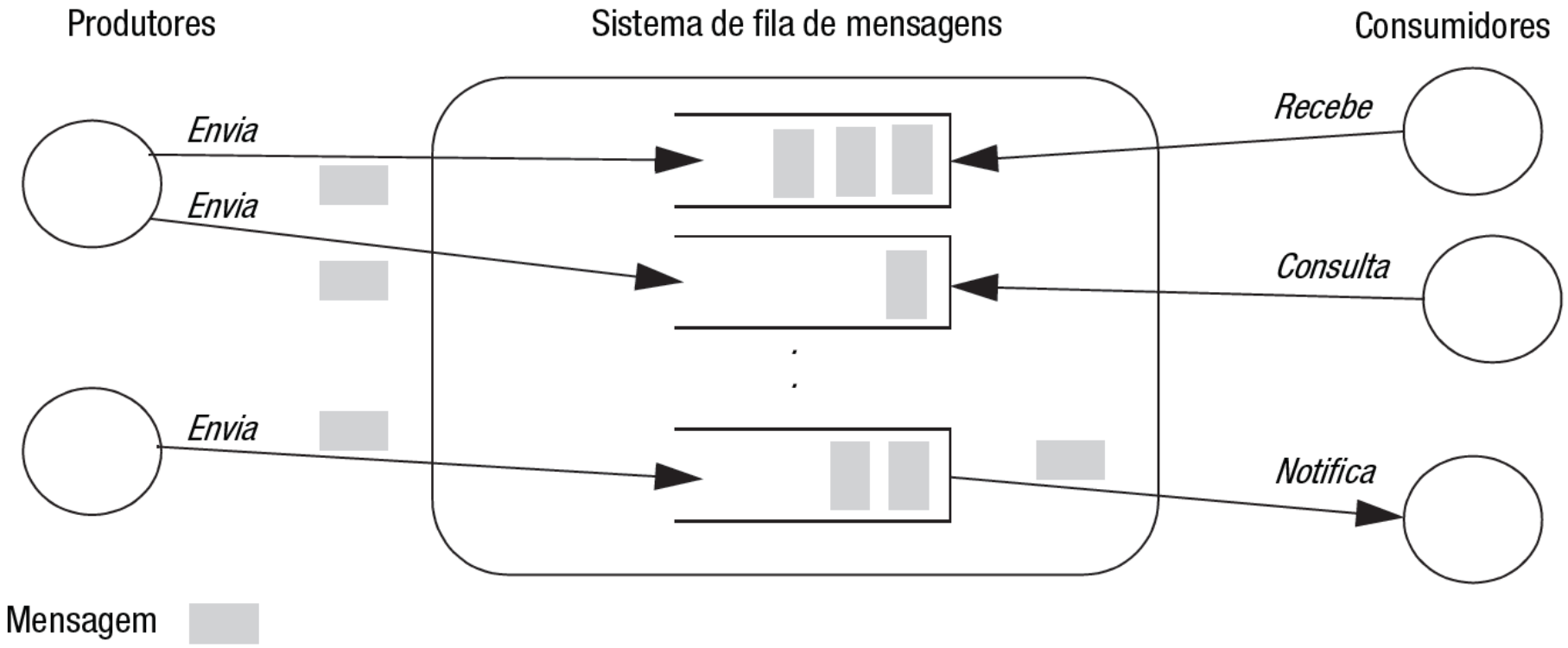


Middleware Orientado a Mensagem (MOM)

- Primitiva de interação: passagem de mensagem
- Comunicação assíncrona/em grupo naturalmente implementadas
- Uso de fila (temporárias, persistentes) de mensagens
- Sub-tipos:
  - fila de mensagem
  - *publish/subscribe*
- Comunicação: 1-1, 1-N

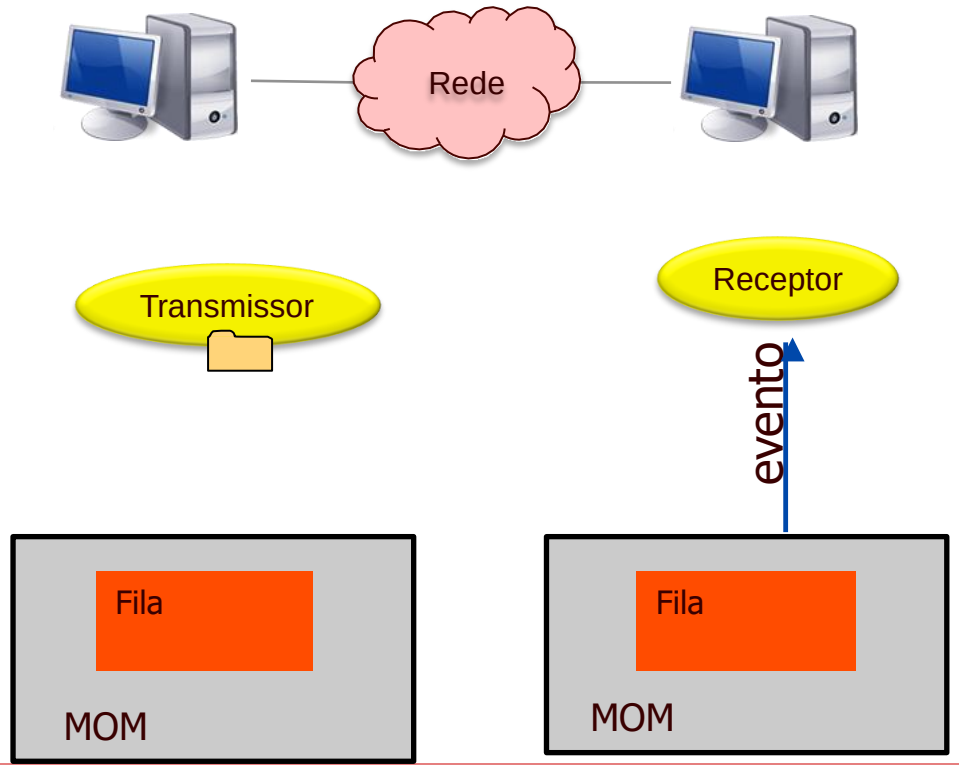


O paradigma da fila de mensagens.



Middleware Orientado a Mensagem (MOM)

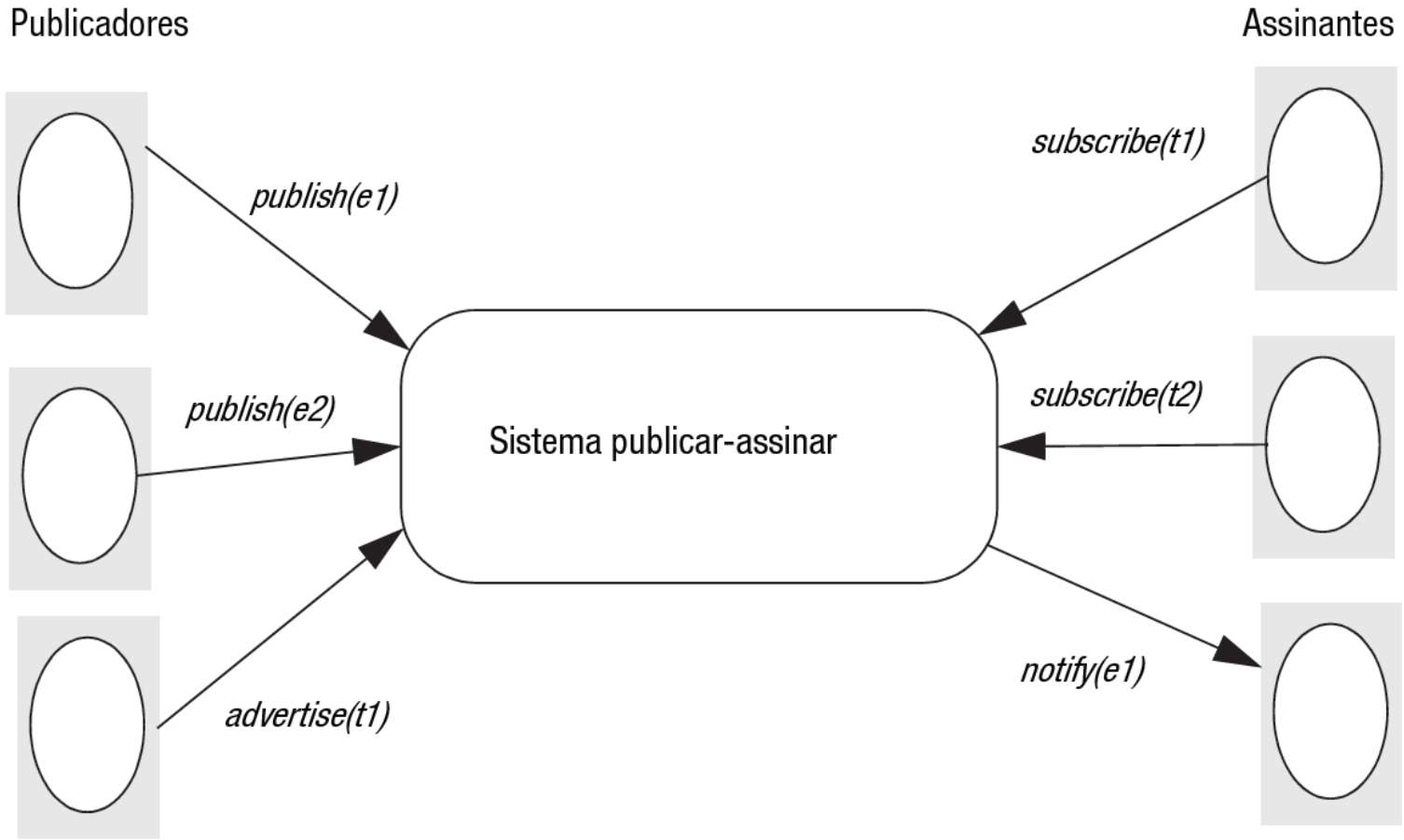
- Fila de Mensagem



Interface básica para uma fila

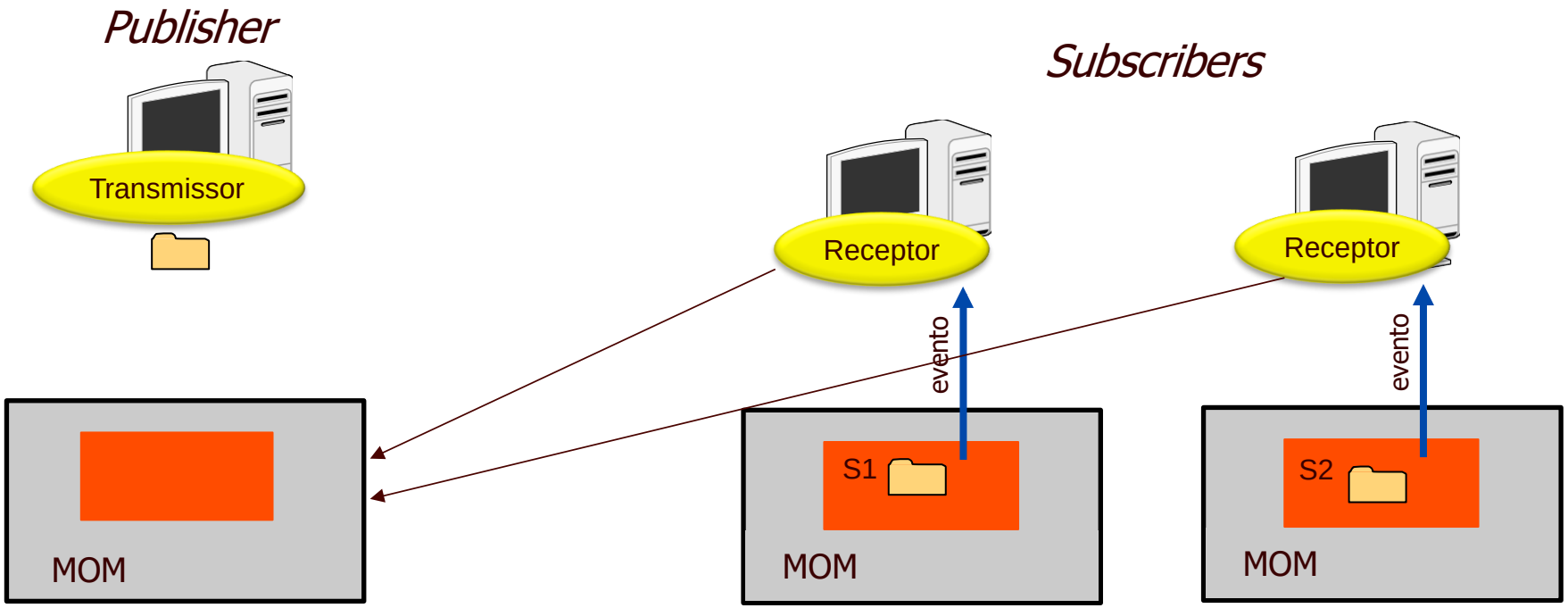
Primitiva	Significado
Put	Anexe uma mensagem a uma fila especificada
Get	Bloqueie até que a fila especificada esteja não vazia e retire a primeira mensagem
Poll	Verifique uma fila especificada em busca de mensagens e retire a primeira. Nunca bloqueie
Notify	Instale um manipulador a ser chamado quando uma mensagem for colocada em uma fila específica

O paradigma publicar-assinar.



Middleware Orientado a Mensagem (MOM)

- *publish/subscribe*

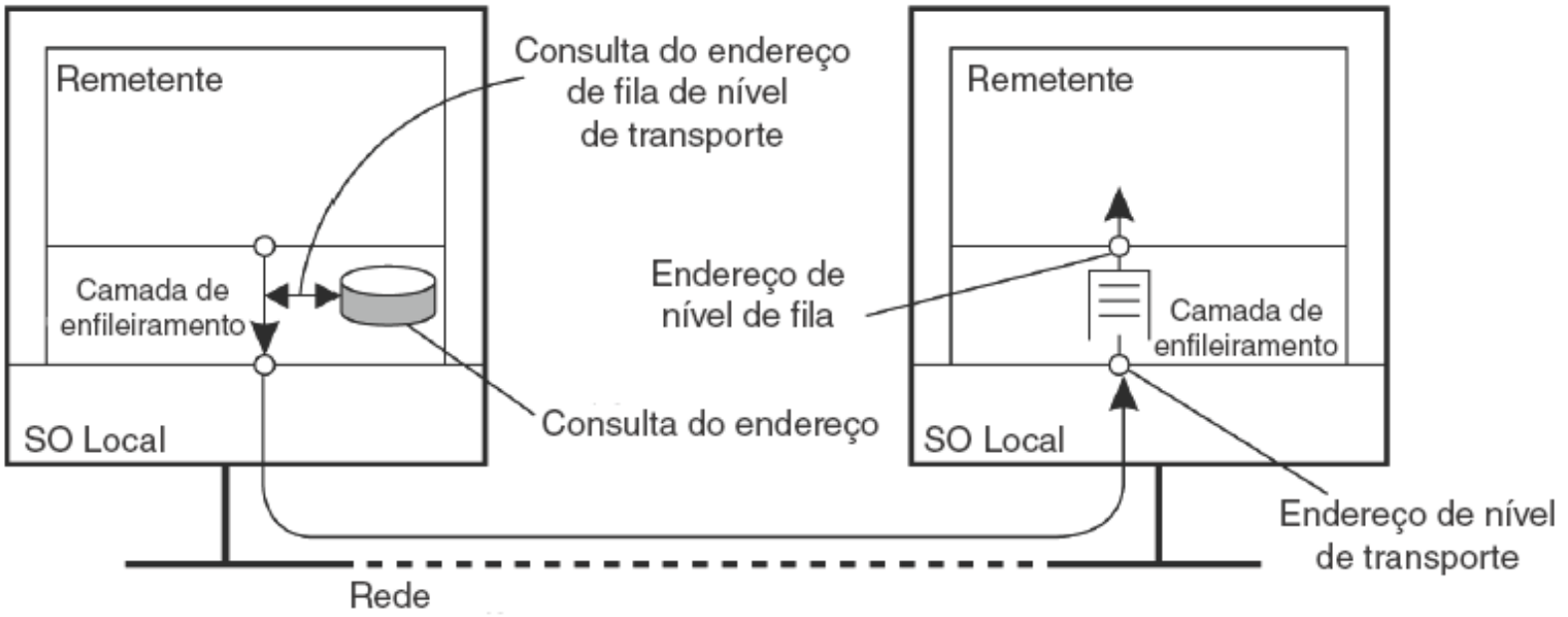


MOM: Componentes da Arquitetura

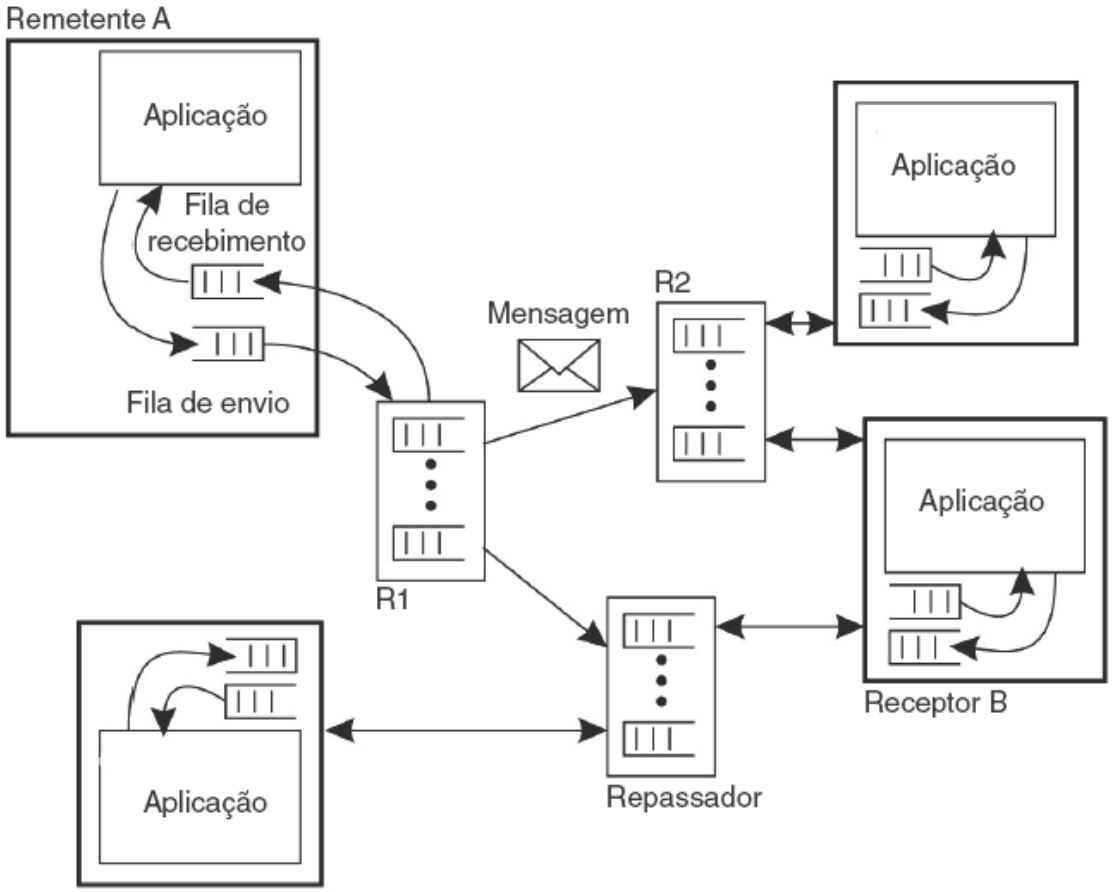
- Fila de Fonte
- Fila de Destino
- Gerenciadores de Fila
- Repassadores

MOM: Componentes da Arquitetura

- A organização da fila é feita pelo gerenciador da fila (FIFO? LIFO?)
- Repassadores também são chamados de roteadores



MOM: Componentes da Arquitetura

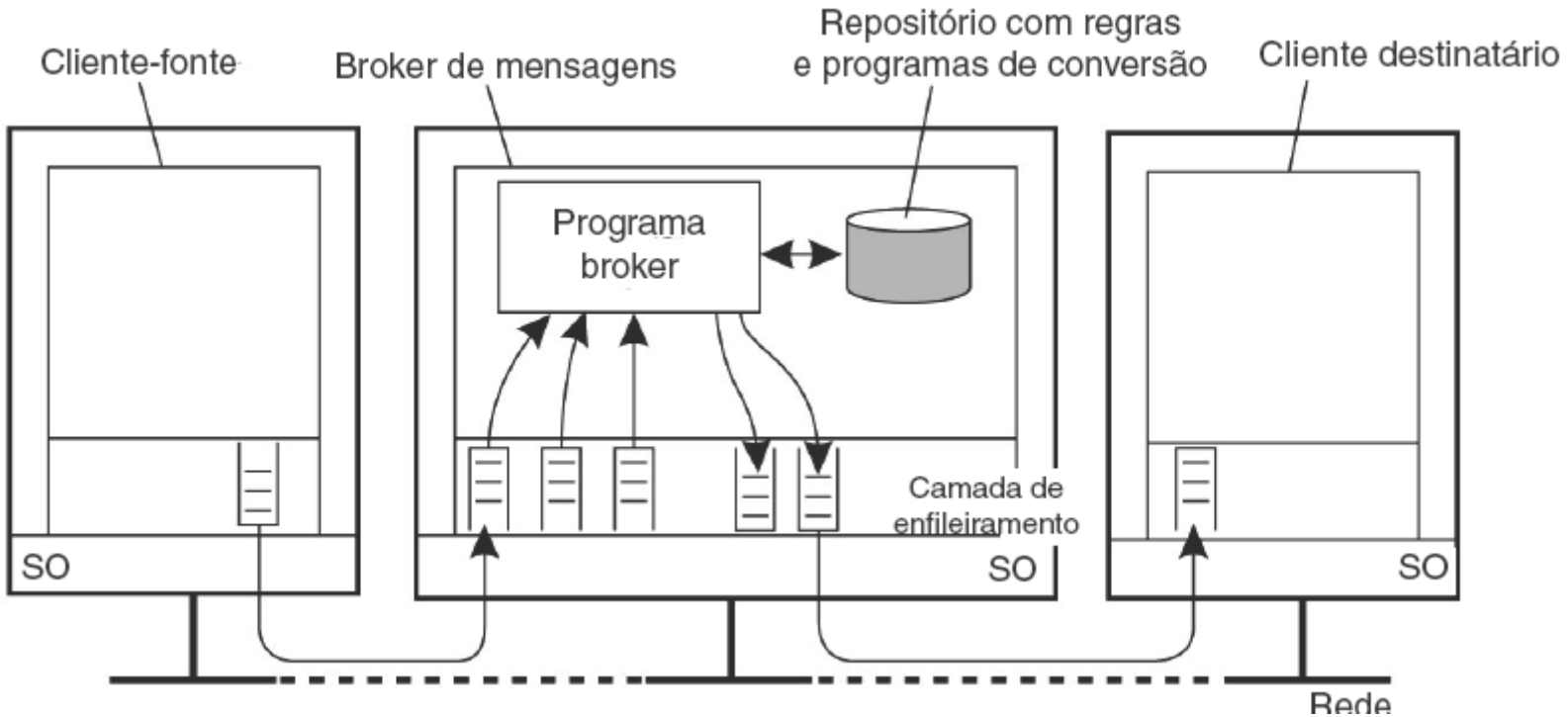


MOM: Mais informações sobre os repassadores

- Repassadores ajudam a construir sistemas escaláveis de gerenciamento de fila.
- Atualizações de remoção e adição de filas devem ser informadas aos repassadores.
- Gerenciadores de fila devem saber onde está o repassador mais próximo.

Brokers de mensagens

- Aquitetura de um *Message Broker/Provider*

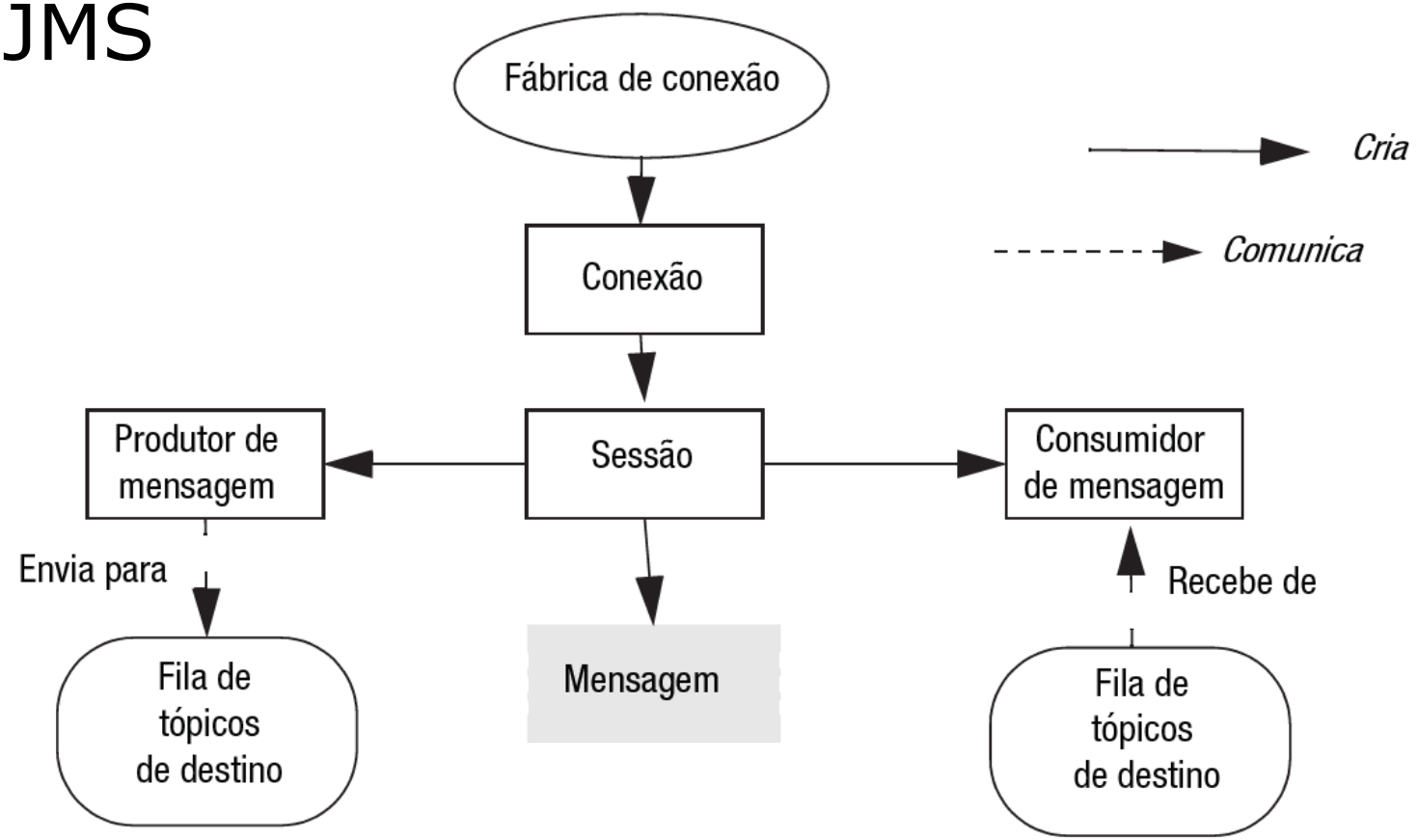


JMS (*Java Messaging Service*)

- Padrão de interface para acesso a MOMs
- Independente de fornecedor, mas não de linguagem
- Suportado por diversos MOMs e por grande parte dos servidores de aplicação
- Elementos
  - Provedor JMS
  - Clientes JMS
    - Produtores
    - Consumidores

JMS (*Java Messaging Service*)

- Modelo de programação oferecido pelo JMS



JMS (*Java Messaging Service*)

- Fábrica de conexão: responsável por estabelecer conexões com as propriedades exigidas.
- Sessões: série de operações envolvendo criação, produção e consumo de mensagens.
- Mensagem: um *cabeçalho*, um conjunto de *propriedades* e o *corpo* da mensagem.

JMS (*Java Messaging Service*)

- Produtor de mensagem: objeto usado para publicar mensagens sob um tópico específico ou para enviar mensagens para uma fila.

- Consumidor de mensagem: objeto usado para assinar mensagens relativas a determinado tópico ou para receber mensagens de uma fila.

JMS (*Java Messaging Service*)

Classe Java *FireAlarmJMS*

```java
import javax.jms.*;
import javax.naming.*;

public class FireAlarmJMS {

    public void raise() {
        try {
            Context ctx = new InitialContext();
            TopicConnectionFactory topicFactory =
                    (TopicConnectionFactory)ctx.loo kup("TopicConnectionFactory");
            Topic topic = (Topic) ctx.lookup("Alarms");
            TopicConnection topicConn =
            topicFactory.createTopicConnection();
            TopicSession topicSess = topicConn.createTopicSession(false,
                    Session.AUTO_ACKNOWLEDGE);
            TopicPublisher topicPub = topicSess.createPublisher(topic);
            TextMessage msg = topicSess.createTextMessage();
            msg.setText("Fire!");
            topicPub.publish(msg);
        } catch (Exception e) {
        }
}
```

JMS (*Java Messaging Service*)

Classe Java *FireAlarmConsumerJMS*

```
import javax.jms.*;
import javax.naming.*;

public class FireAlarmConsumerJMS {

    public String await() {
        try {
            Context ctx = new InitialContext();
            TopicConnectionFactory topicFactory =
                (TopicConnectionFactory)ctx.lookup("TopicConnectionFactory");
            Topic topic = (Topic) ctx.lookup("Alarms");
            TopicConnection topicConn =
            topicFactory.createTopicConnection();
            TopicSession topicSess = topicConn.createTopicSession(false,
                    Session.AUTO_ACKNOWLEDGE);
            TopicSubscriber topicSub = topicSess.createSubscriber(topic);
            topicSub.start();
            TextMessage msg = (TextMessage) topicSub.receive();
            return msg.getText();
        } catch (Exception e) {
            return null;
        }
}
```