Computação Distribuída

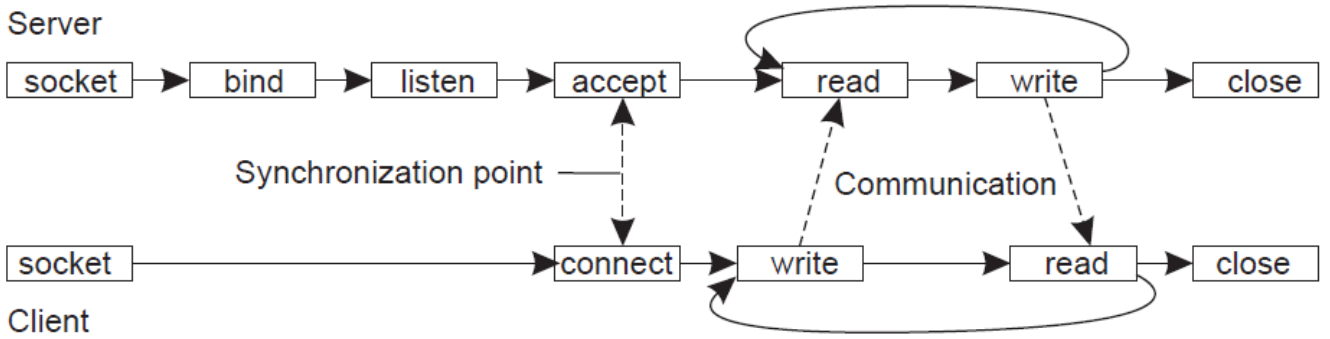
Odorico Machado Mendizabal

Universidade Federal de Santa Catarina – UFSC
Departamento de Informática e Estatística – INE

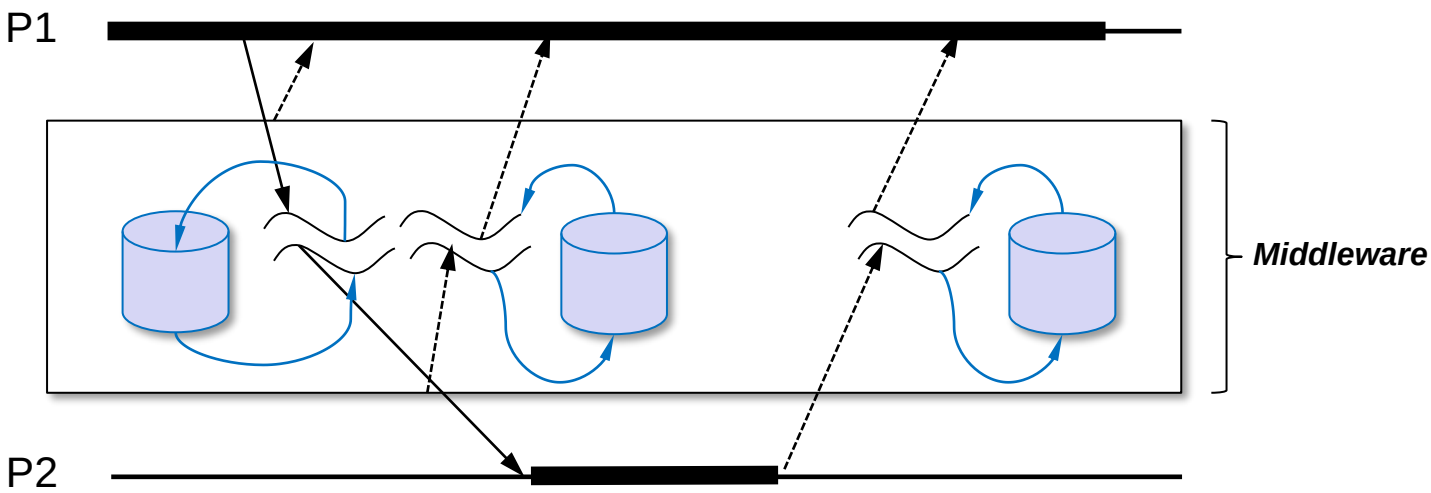
Middleware e serviços de mensagens

Comunicação Orientada a Mensagens

- **Mensagens transientes**
  - Ex. Sockets e canais de comunicação



- **Mensagens persistentes**
  - *Middleware* Orientado a Mensagens - MoM



Algumas Observações

Aplicações cliente/servidor normalmente baseiam-se em **comunicação transiente**:

- Cliente e servidor devem estar ativos durante comunicação
- Cliente envia mensagem e bloqueia até receber resposta (não pode executar outra atividade enquanto espera)
- Servidor é responsável por atender e responder requisições
  - *Comportamento reativo, espera pelo recebimento de mensagens e responde*

Modelo de **comunicação persistente**:

- Remetente não precisa esperar por respostas de requisições para prosseguir seu processamento
- Processos tornam-se mais autônomos (modelos diferentes do cliente/servidor)
- No modelo assíncrono com persistência, não há necessidade de ter os processos executando ao mesmo instante. Mensagens podem ser entregues posteriormente

Middleware Orientado a Mensagens - MoM

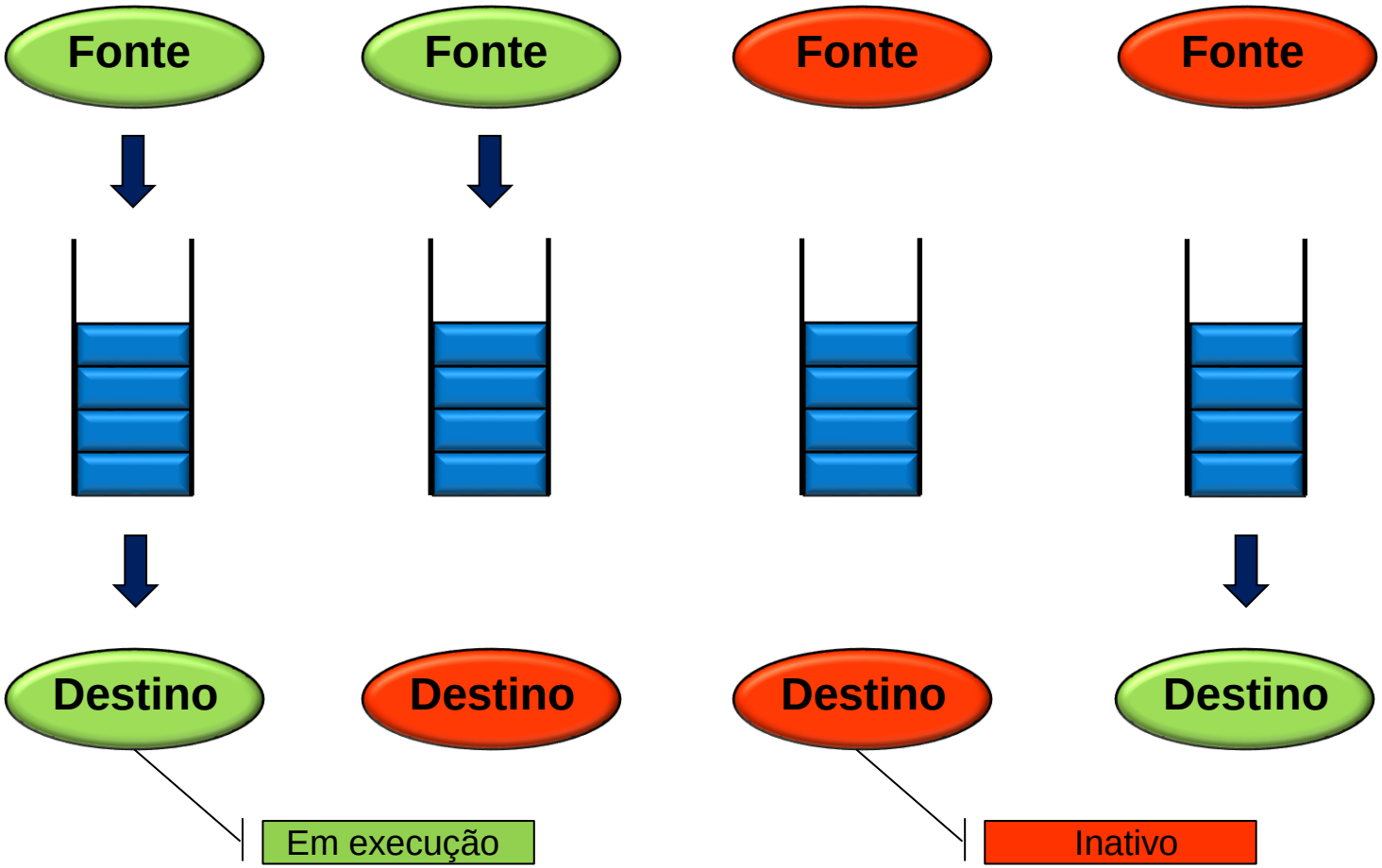
Middleware Orientado a Mensagens

Comunicação assíncrona persistente utilizando **Filas** implementadas pelo *middleware*

PUT	Adiciona uma mensagem a uma determinada fila
GET	Obtém uma mensagem de uma certa fila, bloqueando caso a mesma esteja vazia
POLL	Verifica a fila sem bloquear, obtendo uma mensagem caso a fila não esteja vazia
NOTIFY	Fornece um tratador (*handler*) para ser chamado quando uma mensagem for adicionada a fila

Filas de Mensagens

Estados possíveis durante a comunicação

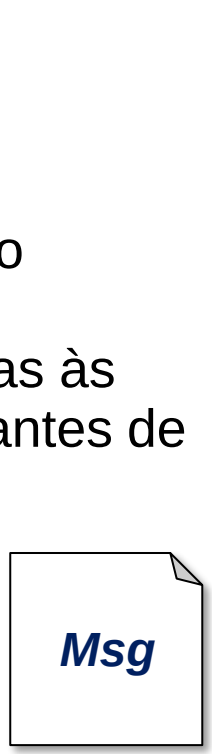


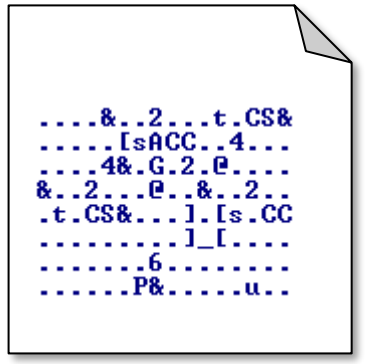
Formato das Mensagens

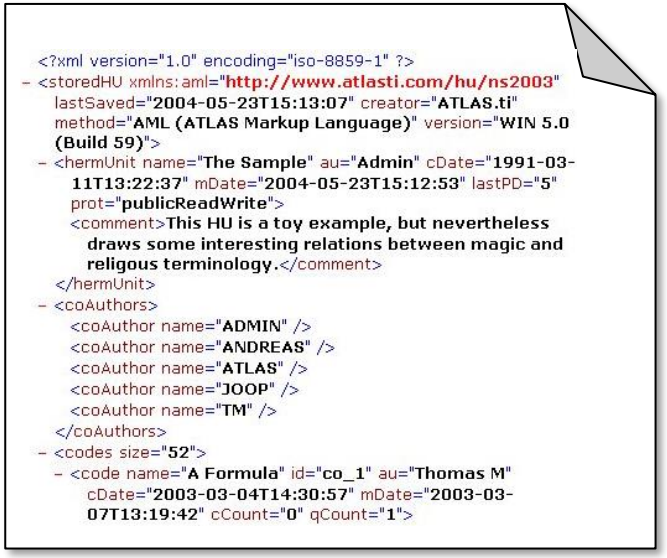
Mensagens podem ter os mais diversos formatos
- *String*
- XML
- Binário

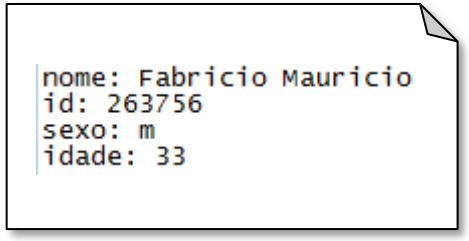
Cada Fila pode adotar um formato próprio

Regras de conversão podem ser aplicadas às mensagens por adaptadores (*adapters*) antes de serem colocadas na fila









Message Broker

Sistemas de Filas de Mensagens adotam um **protocolo de troca de mensagens comum**. Todas aplicações utilizam o mesmo formato de mensagens (estrutura e representação de dados)

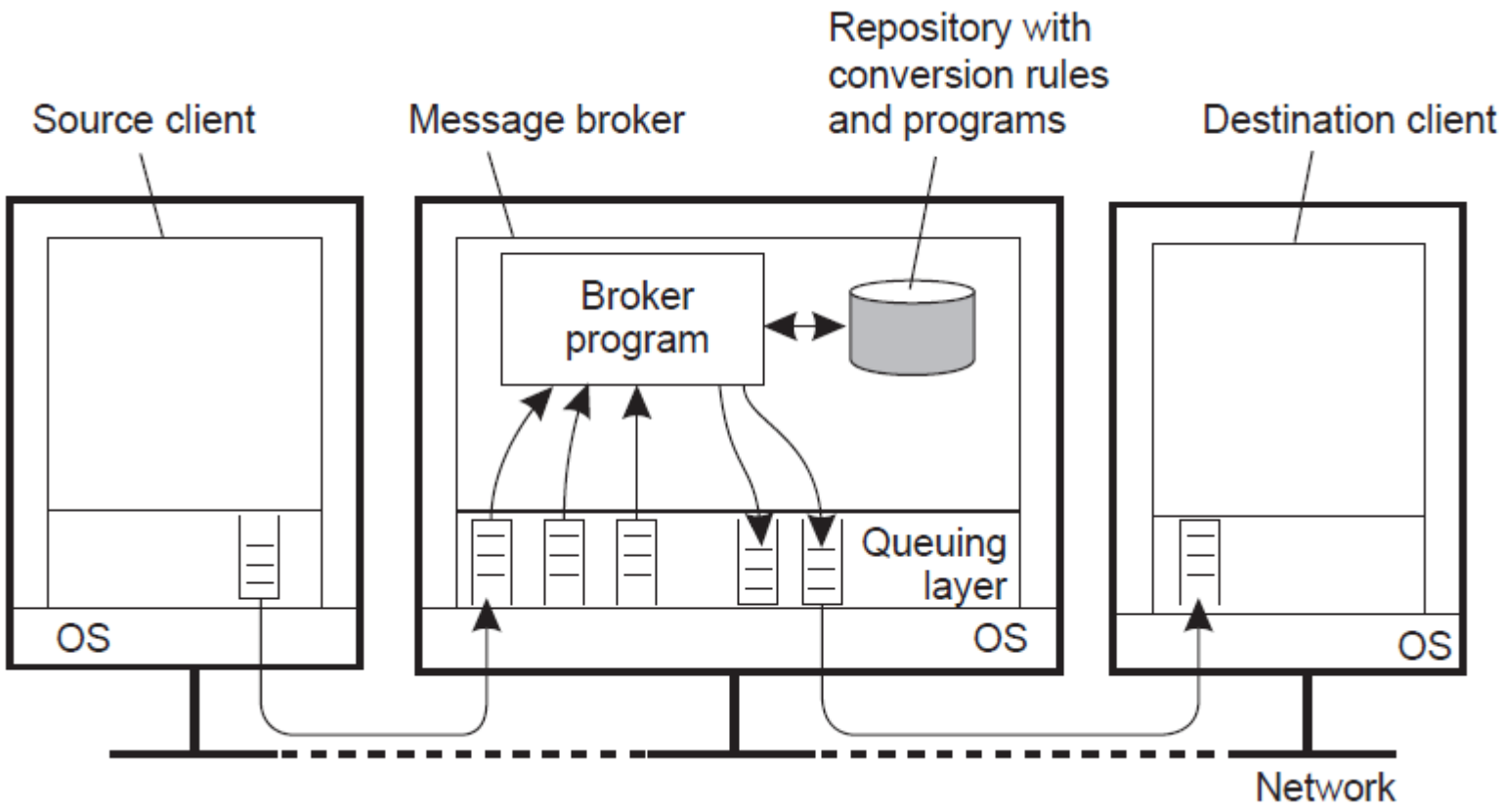
***Message Broker* (Agente)**

- Componente centralizado que permite a interoperabilidade em um sistema de Filas de Mensagem (MQ – *Message-Queuing*) - trata da heterogeneidade

- Transforma mensagens recebidas no formato alvo

- Pode atuar como *Gateway* de aplicação

Message Broker



Vantagens dos Sistemas MQ

- Heterogeneidade: Diferentes aplicações conseguem trocar mensagens entre si, independentemente da linguagem de programação, SO, HW

- Sistemas assíncronos, mensagens podem ser enviadas sem que cliente fique bloqueado esperando confirmação

- Persistência de mensagens: Mensagens enviadas são mantidas até que não sejam mais necessárias (replicação pode ser usada de modo a adicionar tolerância a falhas e aumentar disponibilidade das Filas)

- Múltiplos receptores podem consumir mensagens em uma fila sem que haja coordenação e sincronização entre eles (modelo *Publisher/Subscriber*)

Modelo *Publish/Subscribe*

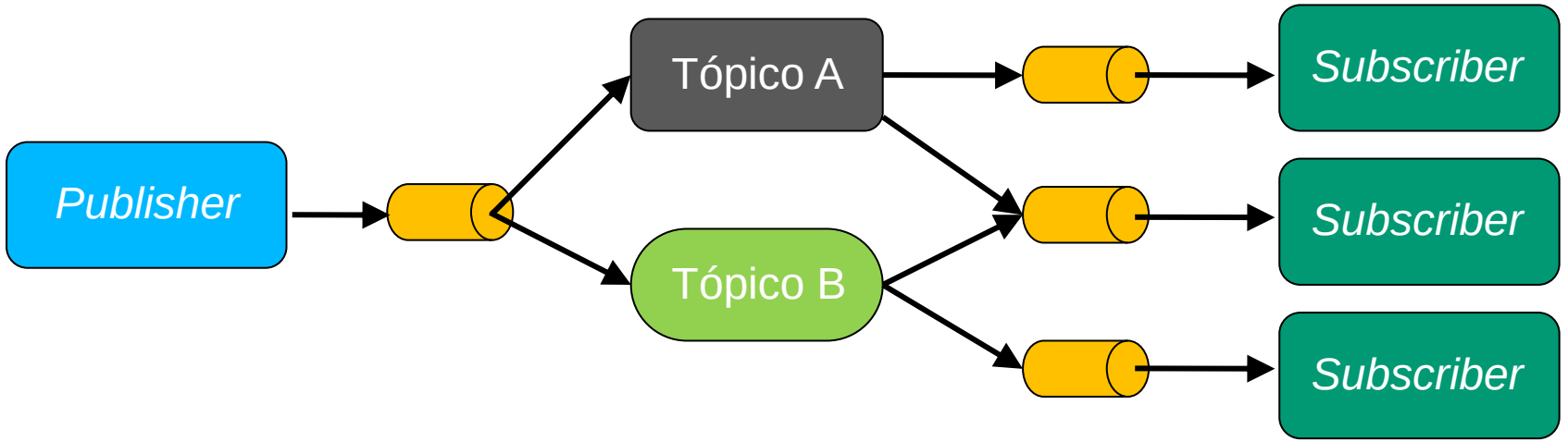
- Sistema onde processos registram interesse em receber mensagens (*inscrição em um tópico*) e também podem publicar mensagens (*publicação em um tópico*)
    - Filtragem de assuntos
- Diferentes níveis
    - Aplicação final: *newsgroups*
    - Primitivas de comunicação
- Modelos de comunicação
    - *push* e *pull*
    - Similar a sistemas baseados em eventos

Modelo *Publish/Subscribe*

Implementado na maioria dos *Message Brokers*

Funcionamento:

- *Publisher:* produtor de conteúdo a ser consumido
- *Subscriber:* "assina" um conjunto de conteúdos de seu interesse



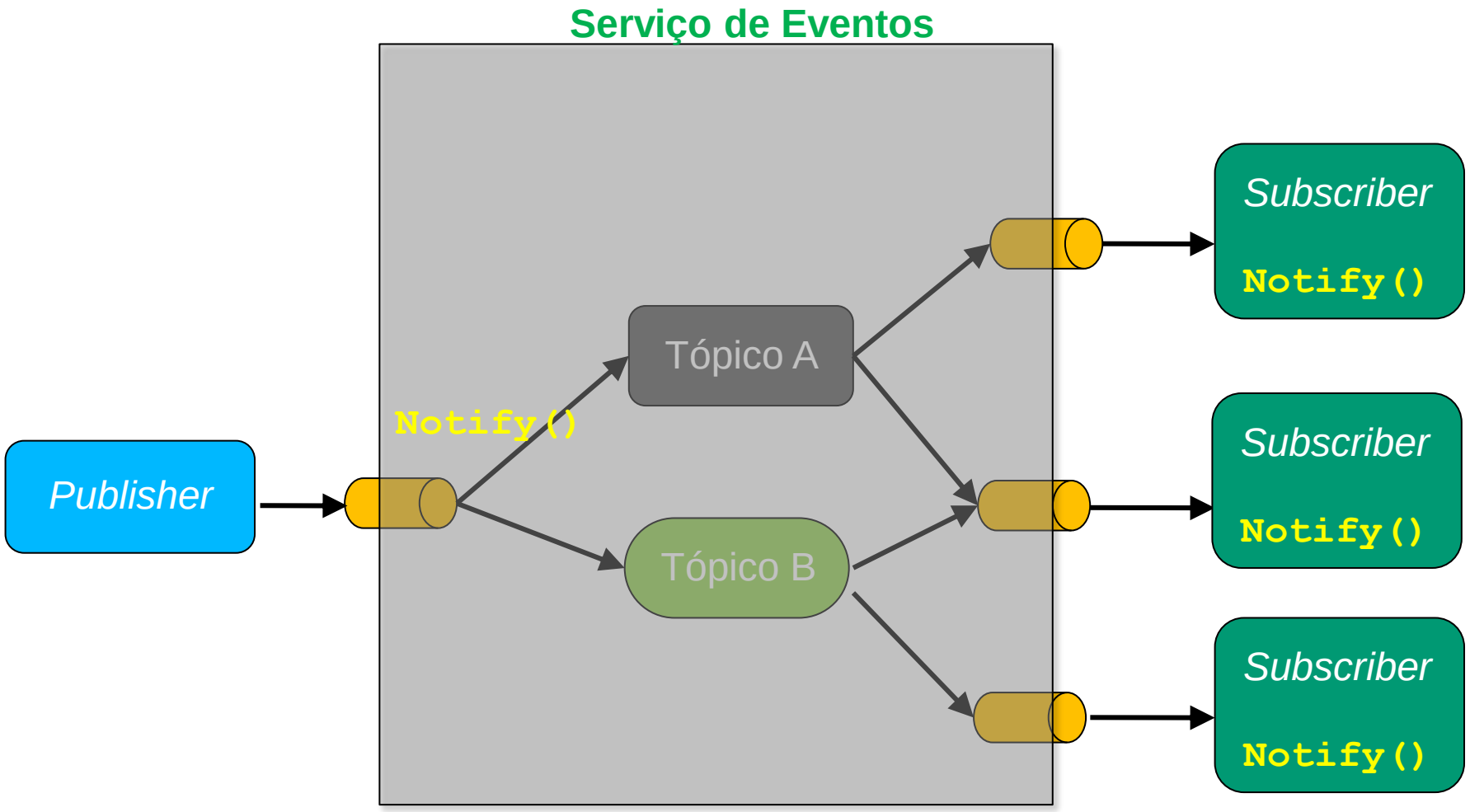
Exemplo: Dropbox

Modelo *Publish/Subscribe* – Padrão de Interação

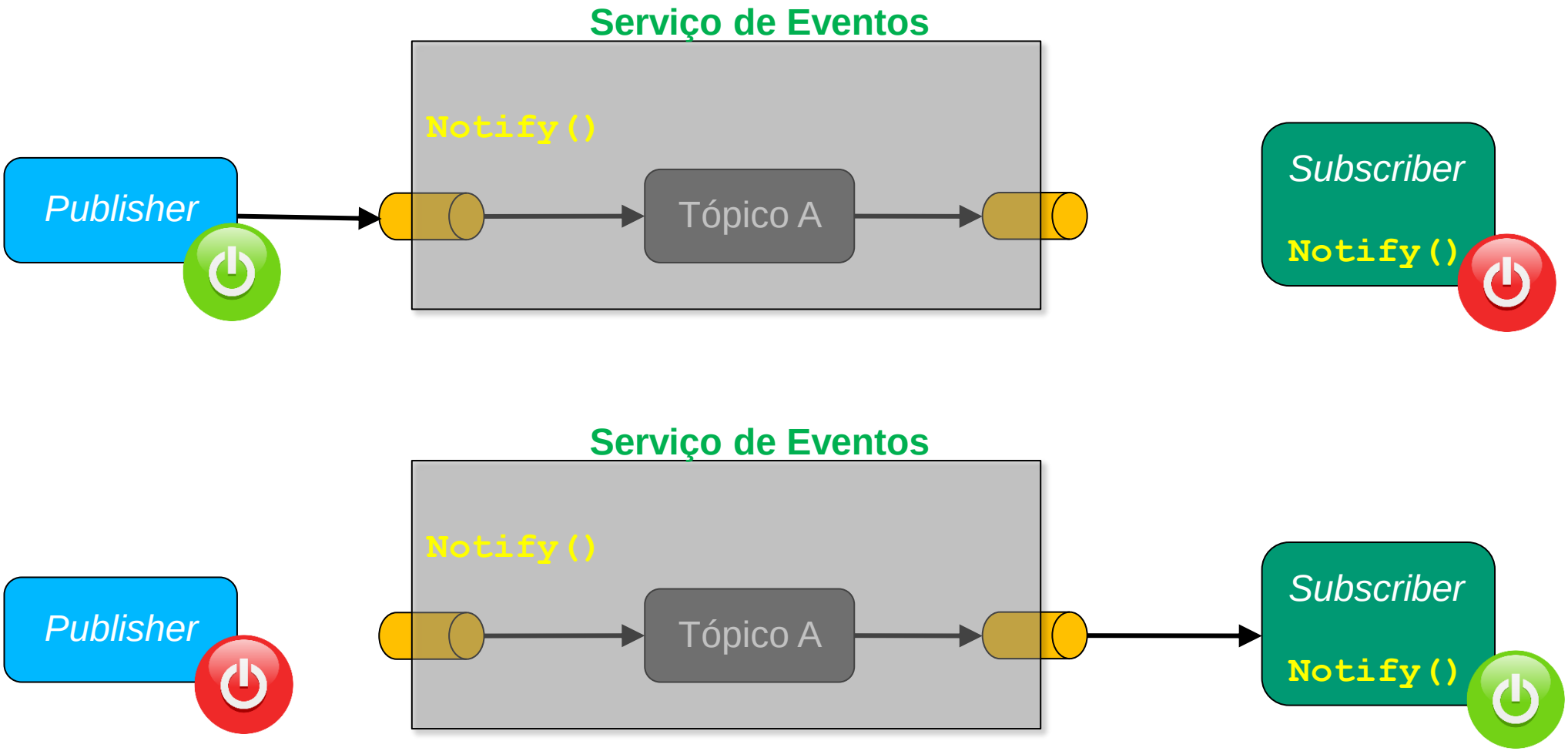
- Assinantes registram interesse em eventos ou padrões de eventos
- publicação de evento gera notificação assíncrona

- Desacoplamento
  - tempo
  - espaço
  - sincronização

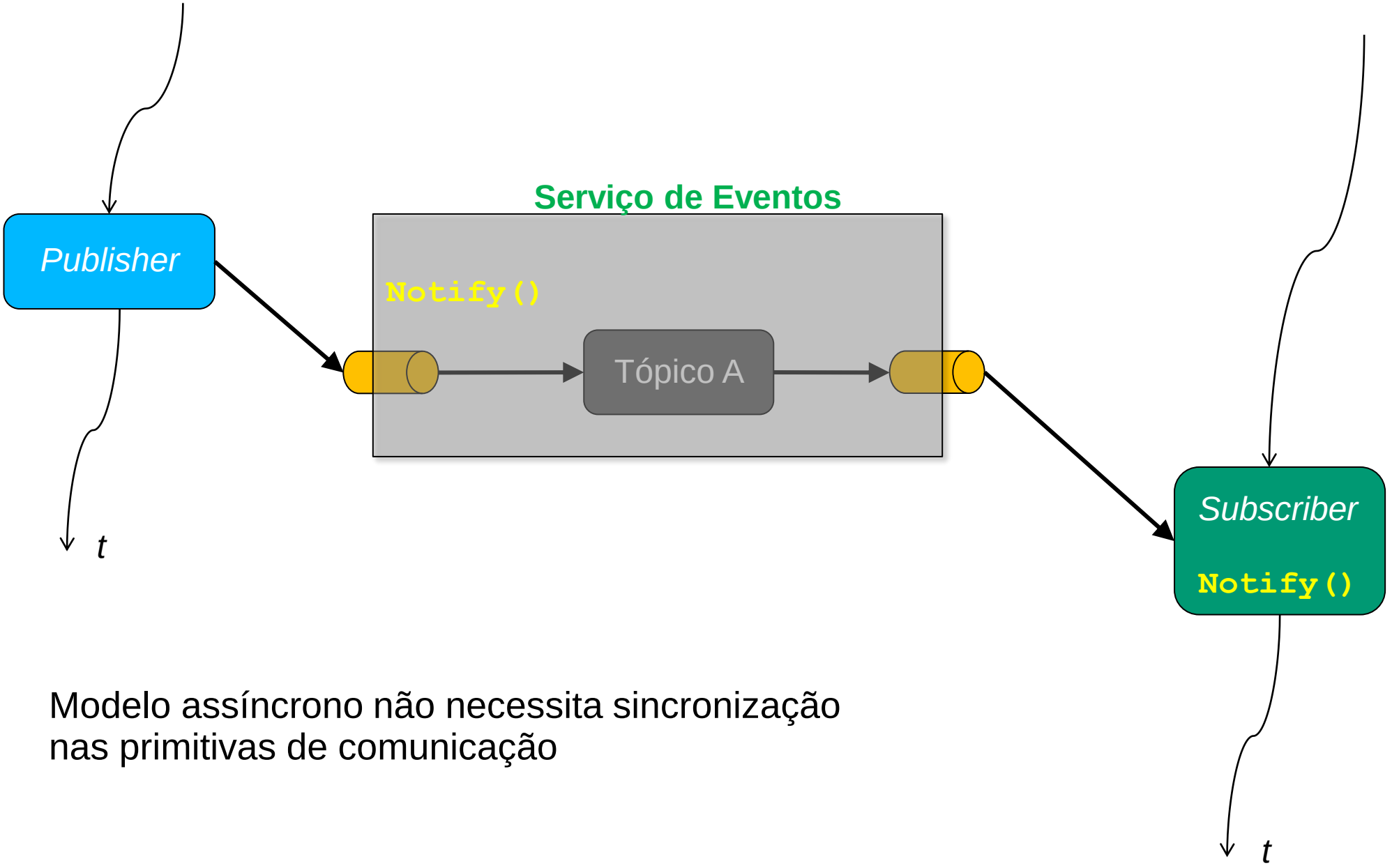
Desacoplamento Espacial



Desacoplamento Temporal



Desacoplamento de Sincronização



Modelo assíncrono não necessita sincronização nas primitivas de comunicação

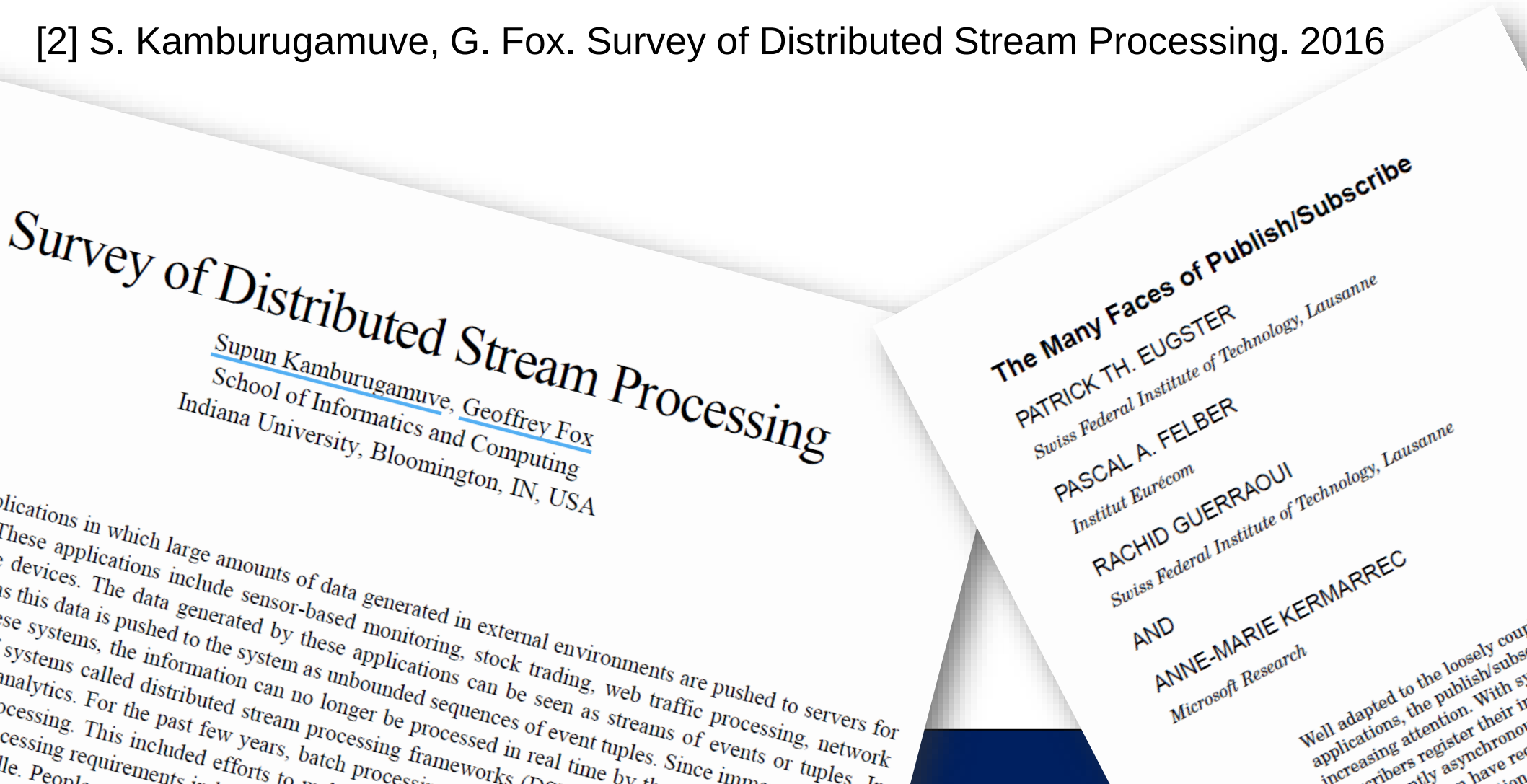
Padrões de Mercado

- JMS – Java Message Service
- IBM Websphere MQ
- Apache ActiveMQ
- BEA Weblogic JMS
- JBoss Messaging
- Microsoft Message Queue Server (MSMQ)
- RabbitMQ
- Zero MQ
- Nano MSG

Leituras Adicionais

[1] P. Eugster, P. Ferrer, R. Guerraoui e A. Kermarrec. The many faces of publish/subscribe. ACM Computing Surveys, 35(2), jun 2003.

[2] S. Kamburugamuve, G. Fox. Survey of Distributed Stream Processing. 2016



Survey of Distributed Stream Processing

Supun Kamburugamuve, Geoffrey Fox
School of Informatics and Computing
Indiana University, Bloomington, IN, USA

lications in which large amounts of data generated in external environments are pushed to servers for
These applications include sensor-based monitoring, stock trading, web traffic processing, network
devices. The data generated by these applications can be seen as streams of events or tuples. In
s this data is pushed to the system as unbounded sequences of event tuples. Since imm
ese systems, the information can no longer be processed in real time by th
systems called distributed stream processing frameworks (D
nalytics. For the past few years, batch processi
ocessing. This included efforts to
cessing requirements
le. People

The Many Faces of Publish/Subscribe

PATRICK TH. EUGSTER
Swiss Federal Institute of Technology, Lausanne

PASCAL A. FELBER
Institut Eurécom

RACHID GUERRAOUI
Swiss Federal Institute of Technology, Lausanne

AND

ANNE-MARIE KERMARREC
Microsoft Research

Well adapted to the loosely cou
applications, the publish/subs
increasing attention. With sy
scribers register their i
ntly asynchronou
have re

Referências

Parte destes slides são baseadas em material de aula dos livros:

- *Coulouris, George; Dollimore, Jean; Kindberg, Tim; Blair, Gordon. Sistemas Distribuídos: Conceitos e Projetos. Bookman; 5ª edição. 2013. ISBN: 8582600534*

- *Tanenbaum, Andrew S.; Van Steen, Maarten. Sistemas Distribuídos: Princípios e Paradigmas. 2007. Pearson Universidades; 2ª edição. ISBN: 8576051427*