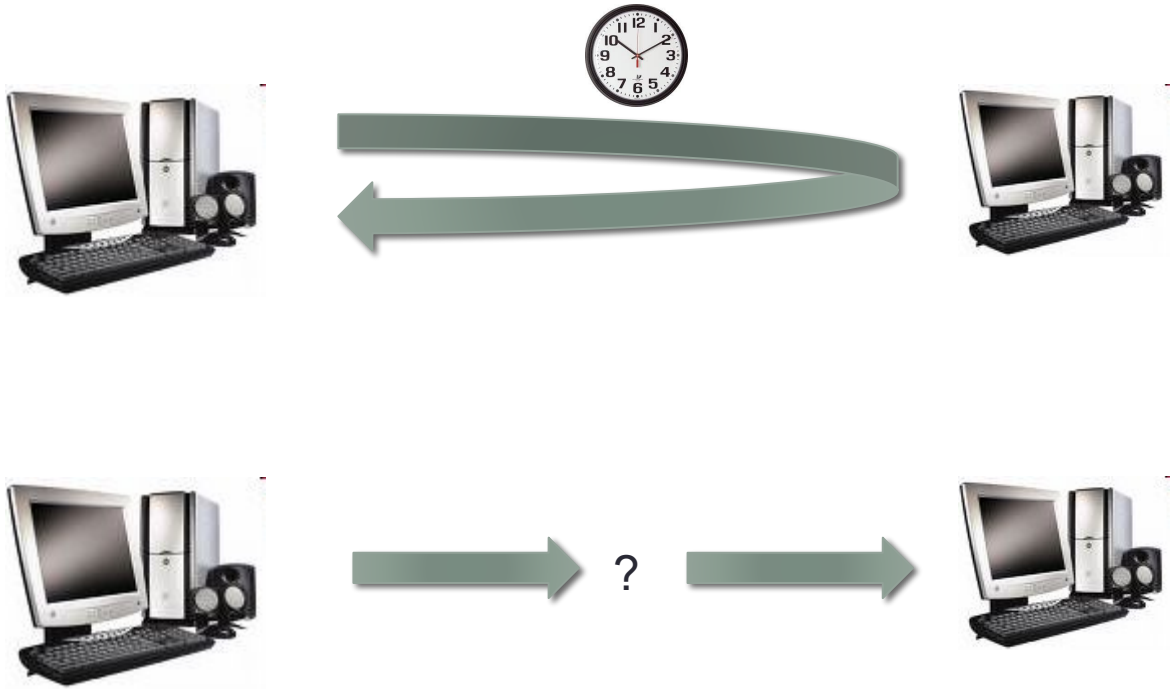
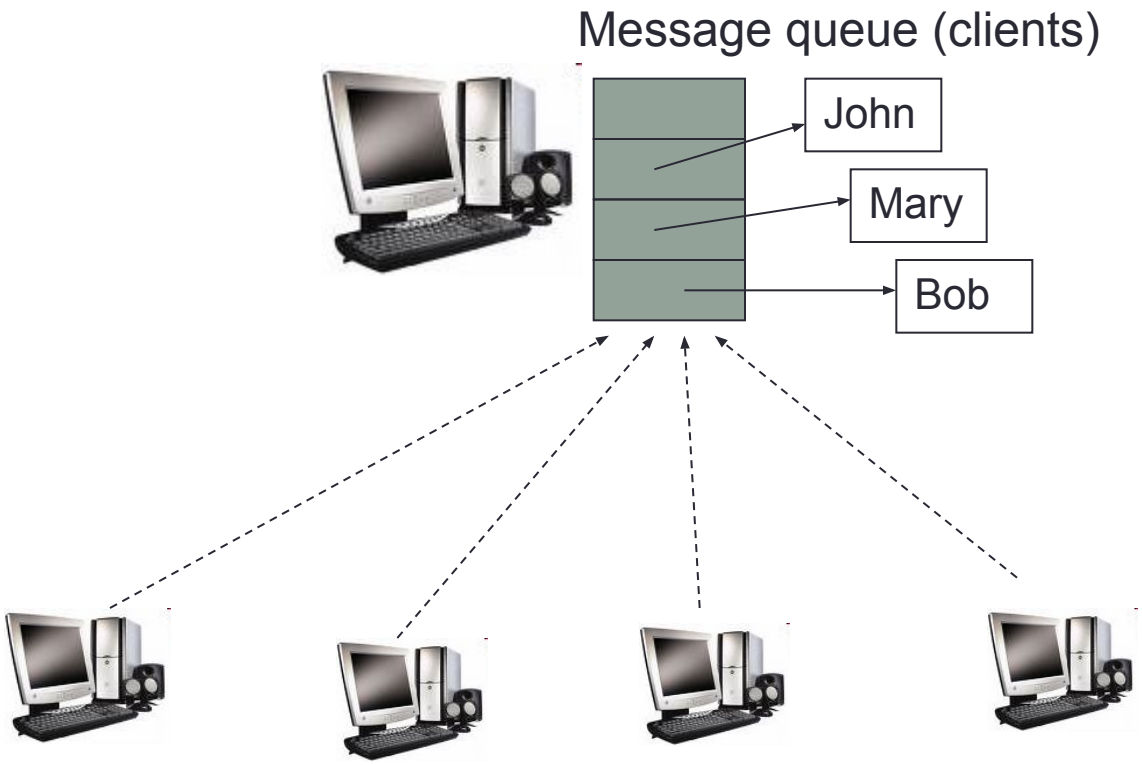
Estilos arquitectónicos basados en eventos

Synchronous P2P
vs Event-based P2P



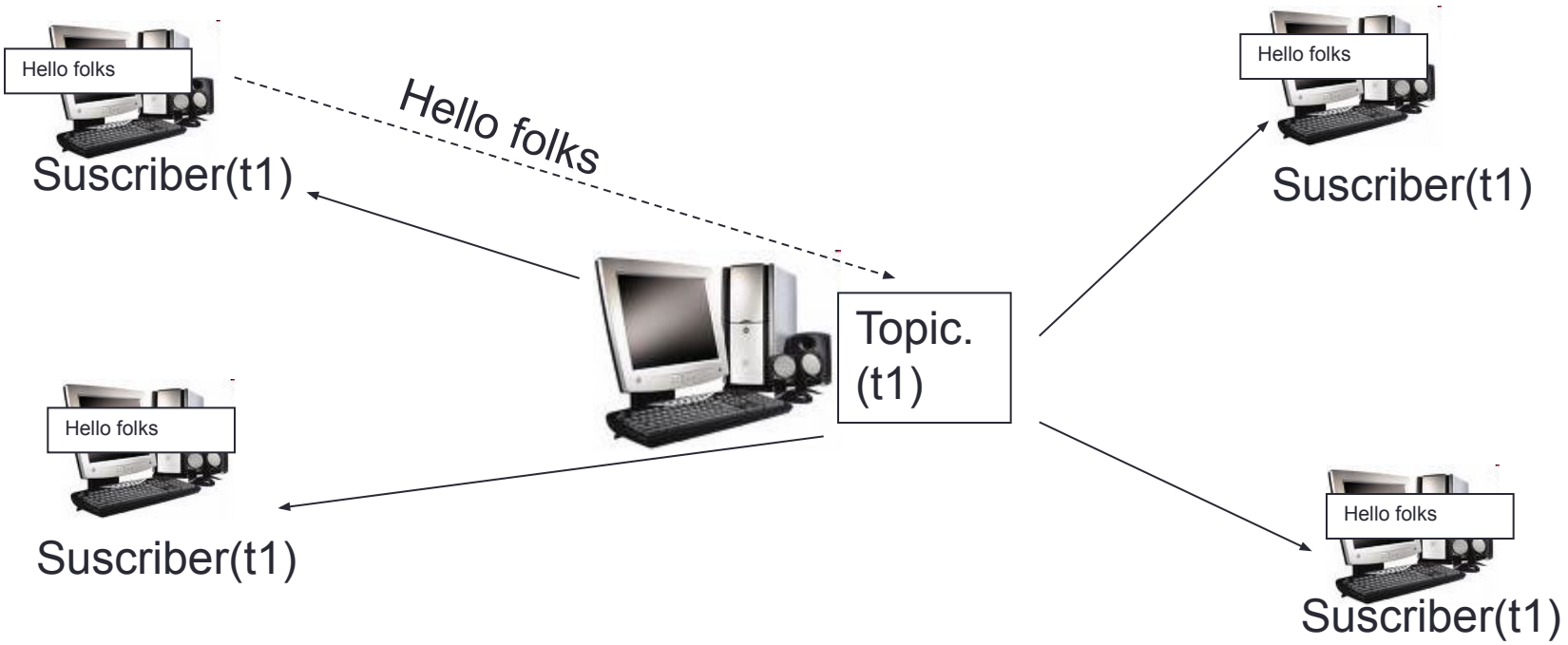
Point to point.

- Example: fighting for a task!

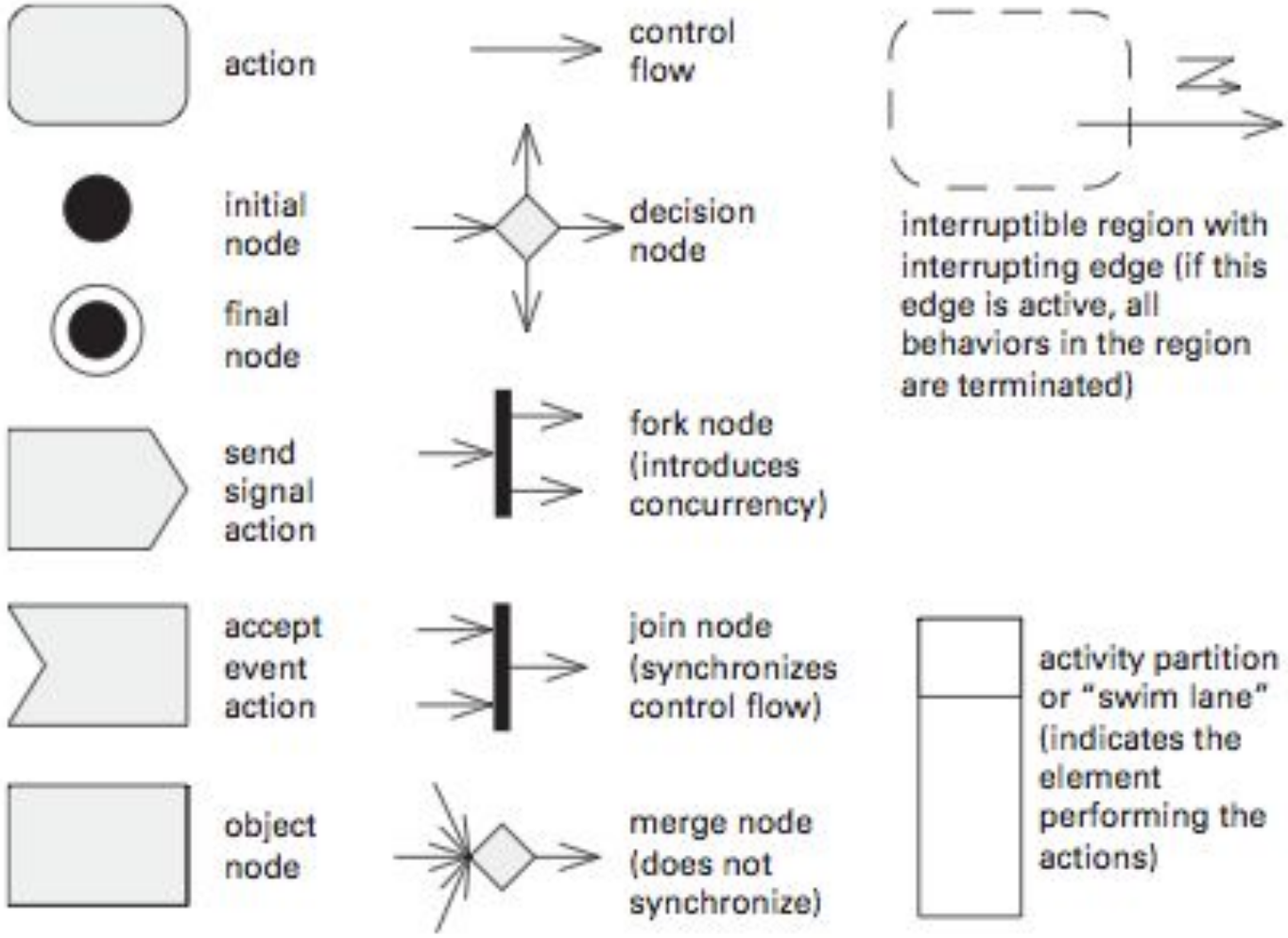


Publisher / Subscriber

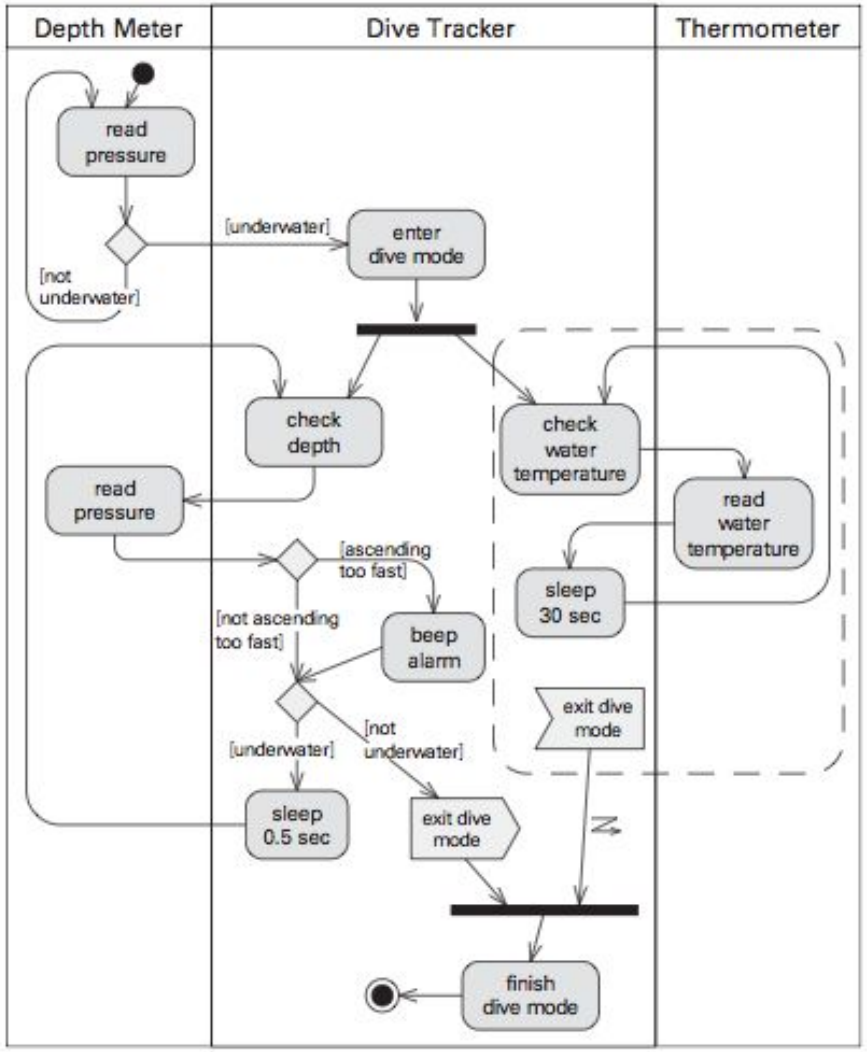
- Example: 2 LOC Simple broadcast chat.



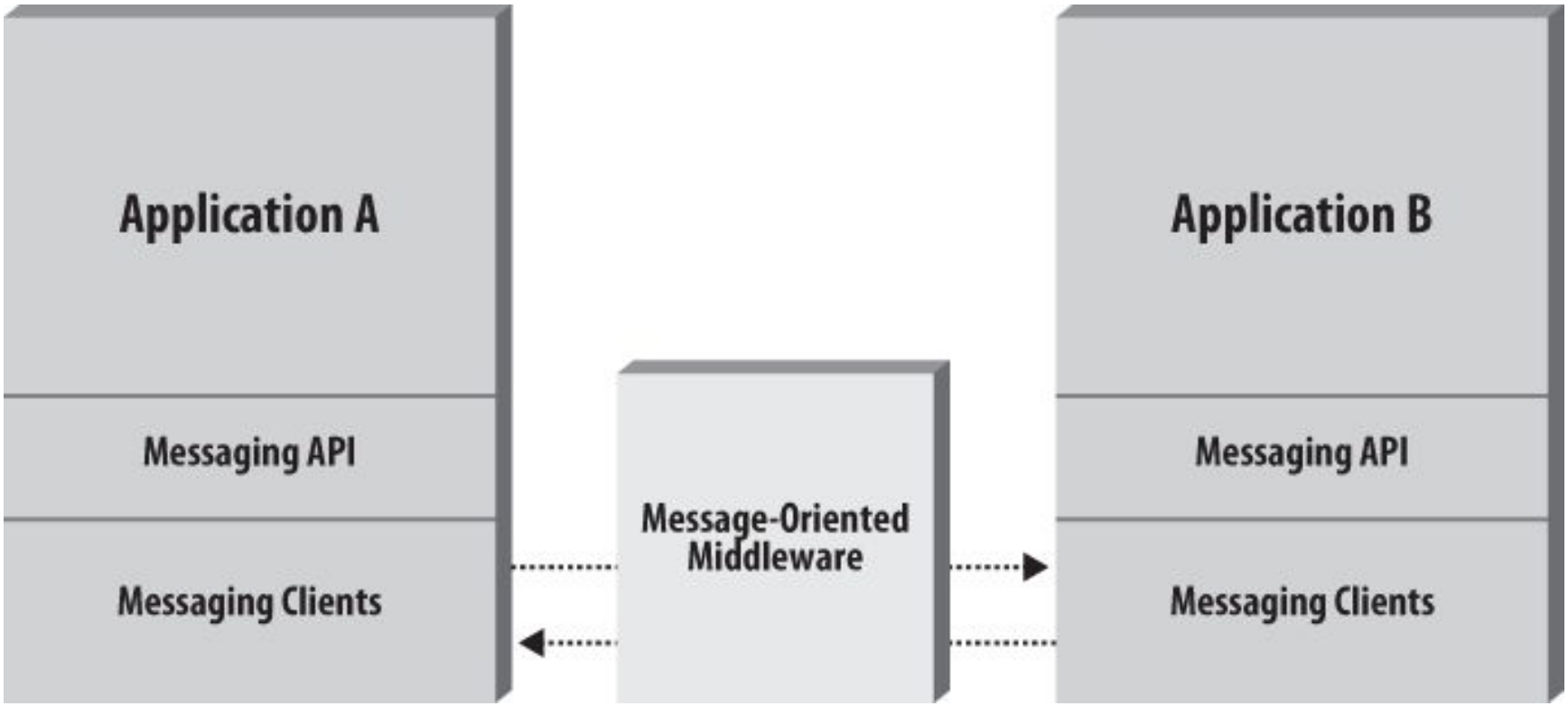
Documentación de arquitecturas: comportamiento dinámico.



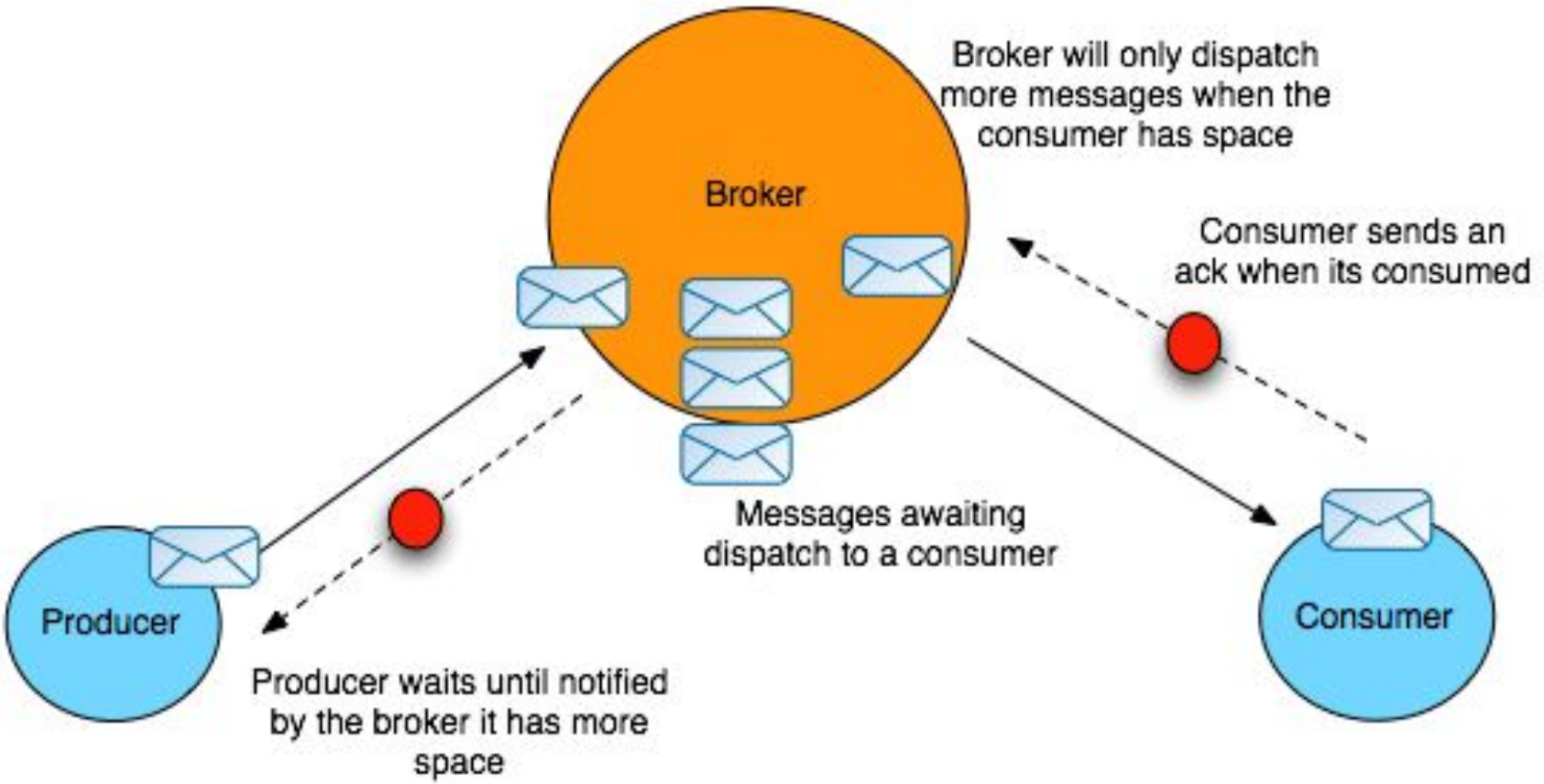
Diagramas de actividad



Message-oriented middleware

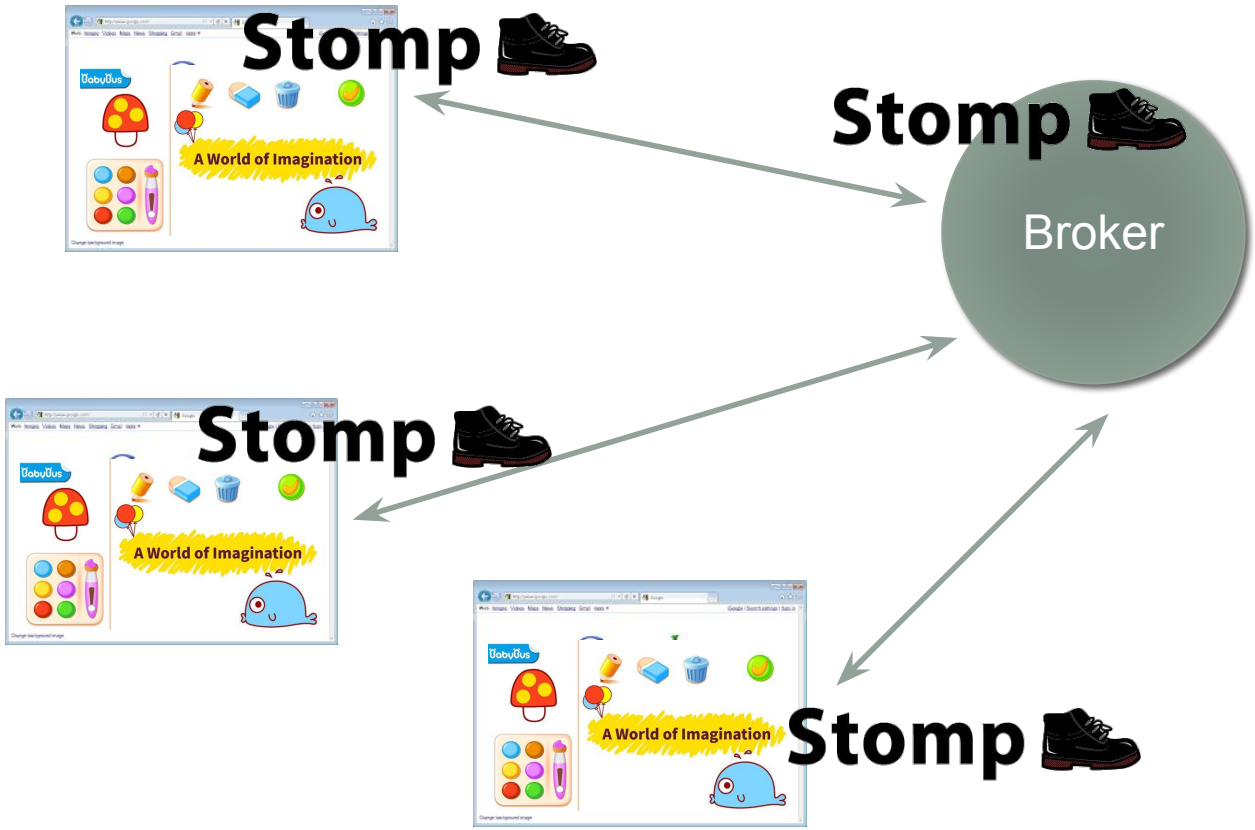


Message Brokers



Protocolos/plataformas de mensajería

Protocolos/plataformas de mensajería – Compatibilidad con clientes Web



Stomp

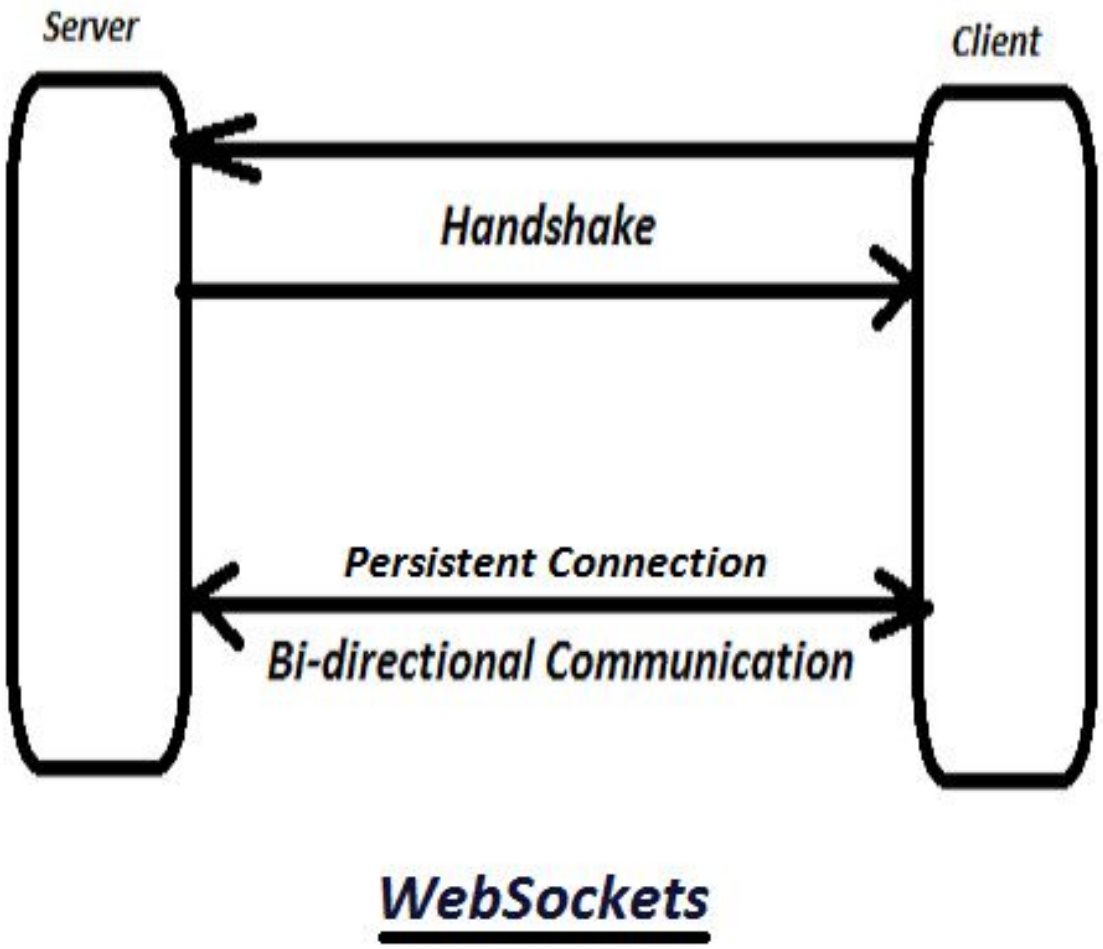
The Simple Text Oriented Messaging Protocol

STOMP Servers

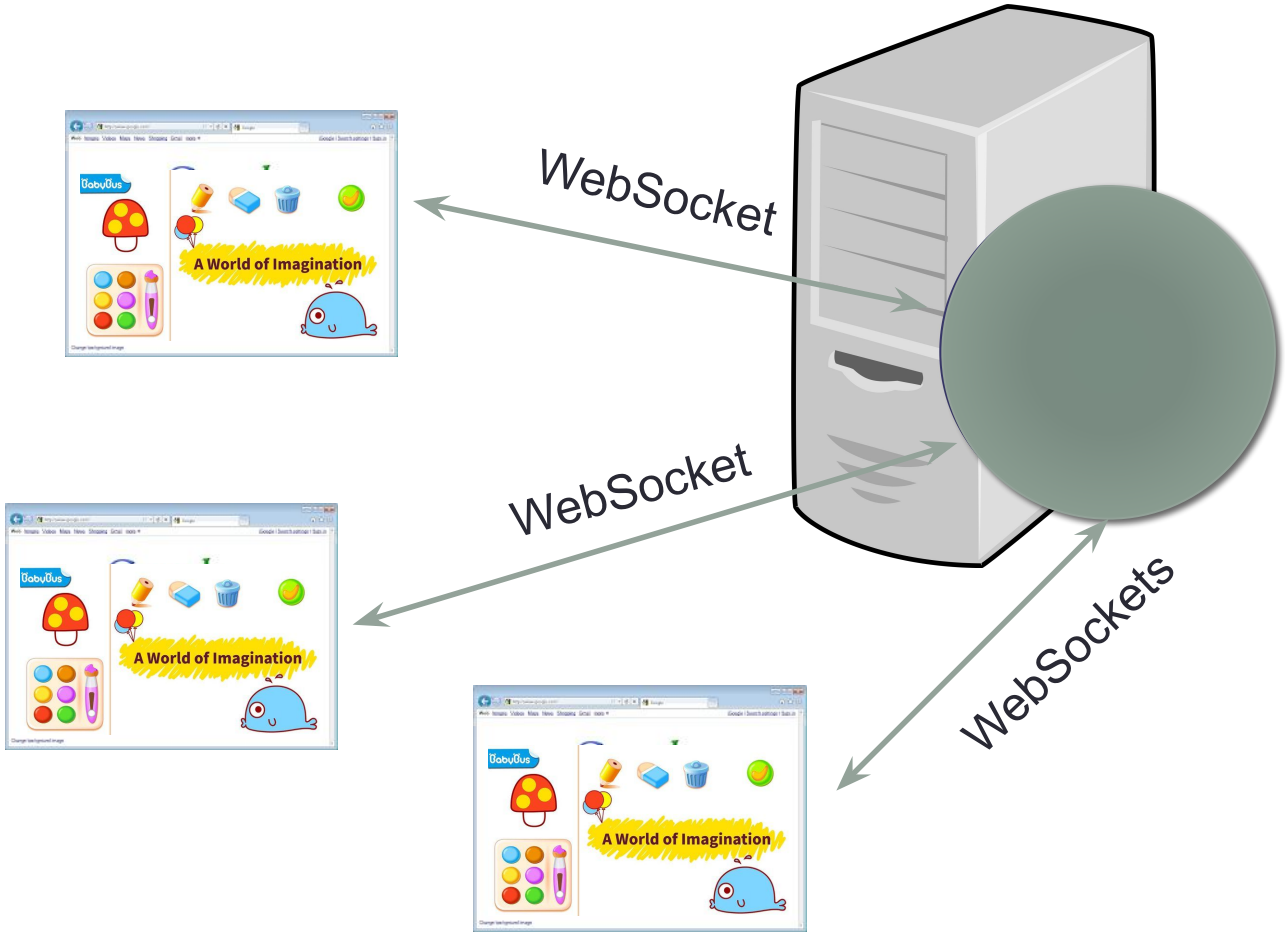
Here are the known STOMP compliant message servers:

Name	Description	Compliance
Apache ActiveMQ	the most popular and powerful open source messaging and Integration Patterns server	1.0 1.1
Apache ActiveMQ Artemis	Apache ActiveMQ Artemis has a proven non blocking architecture. It delivers outstanding performance.	1.0 1.1 1.2
Apache Apollo	a redesigned version of ActiveMQ	1.0 1.1 1.2
CoilMQ	a lightweight pure Python STOMP broker inspired by StompServer	1.0
Gozirra	a lightweight Java STOMP broker	1.0
HornetQ	puts the buzz in messaging	1.0
MorbidQ	a STOMP publish/subscribe server with absolutely no potential to cluster	1.0
RabbitMQ	an Erlang-based, multi-protocol broker with full support for STOMP via a plugin	1.0 1.1 1.2
Sprinkle	written in Python and runs on Unix type platforms	1.0
Stampy	a Java implementation of the STOMP 1.2 specification	1.2
StompConnect	provides a bridge to any other JMS provider	1.0
StompServer	a lightweight pure Ruby STOMP server	1.0

WebSockets


Server
Client
Handshake
Persistent Connection
Bi-directional Communication
WebSockets

Protocolos/plataformas de mensajería – Compatibilidad con clientes Web



STOMP/WS vs REST

StompWS

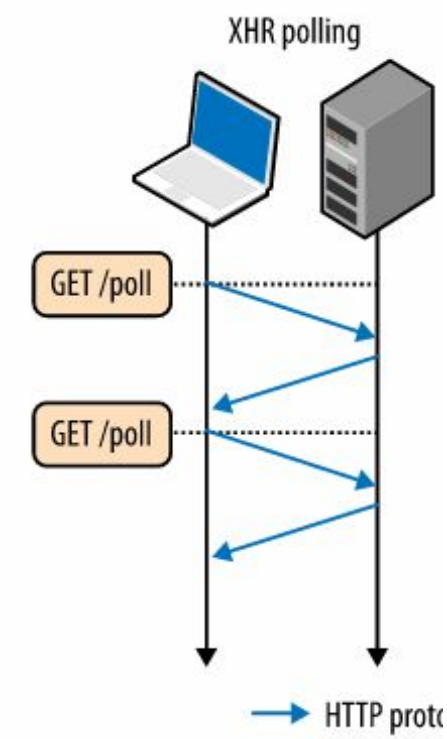
- Persistent Connection.
- Bi-directional communications.
- Not HTTP! (TCP channel)

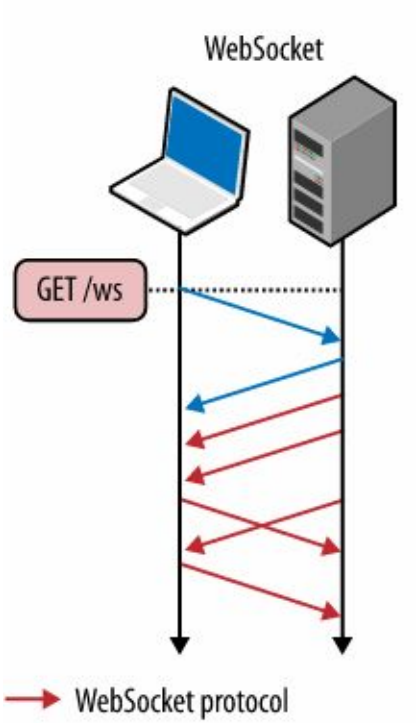
REST

- Stateless Uni-directional communications.
- HTTP Based protocol.

STOMP/WS vs REST

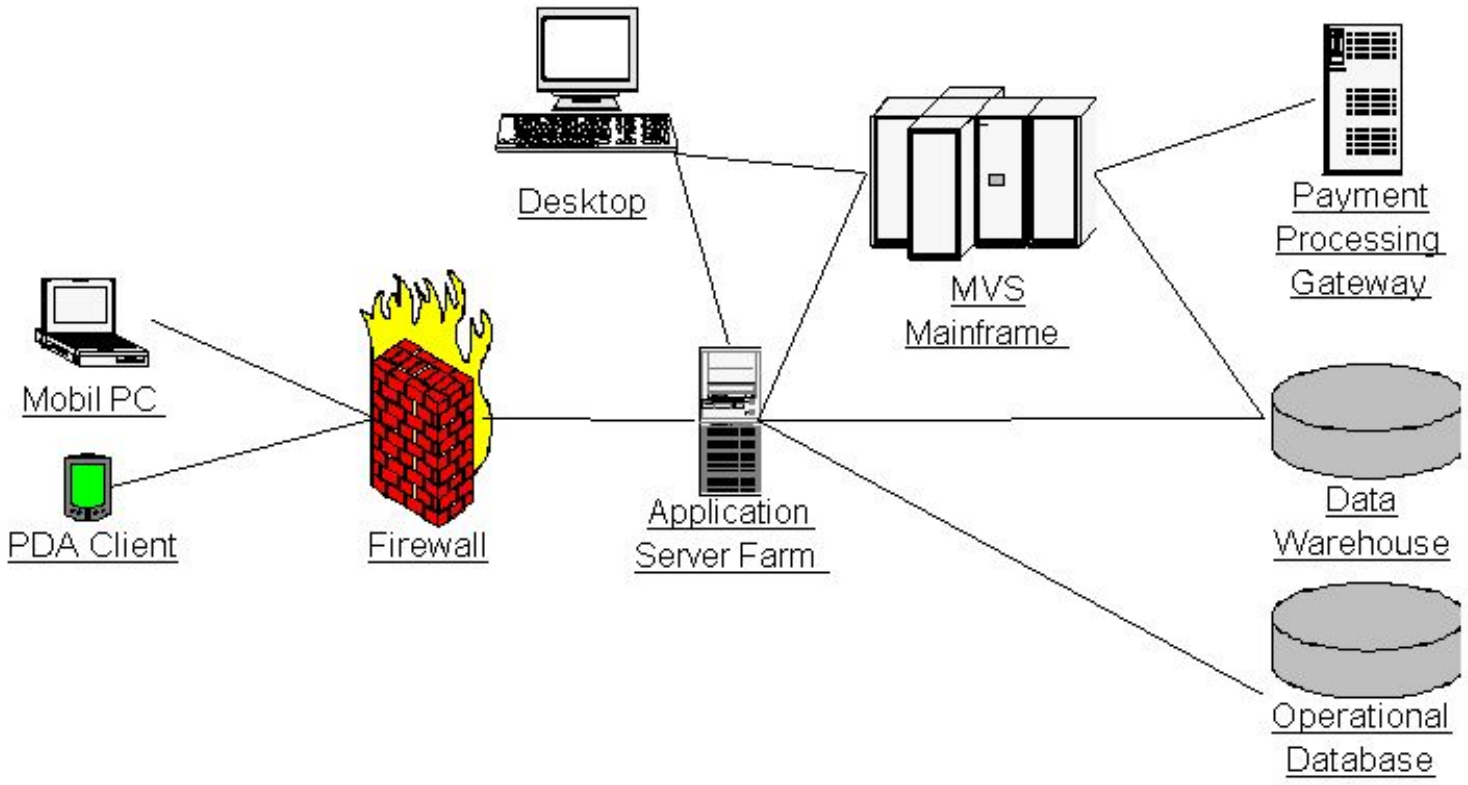
- STOMP/WS
  - High-frequency, Low latency communications.





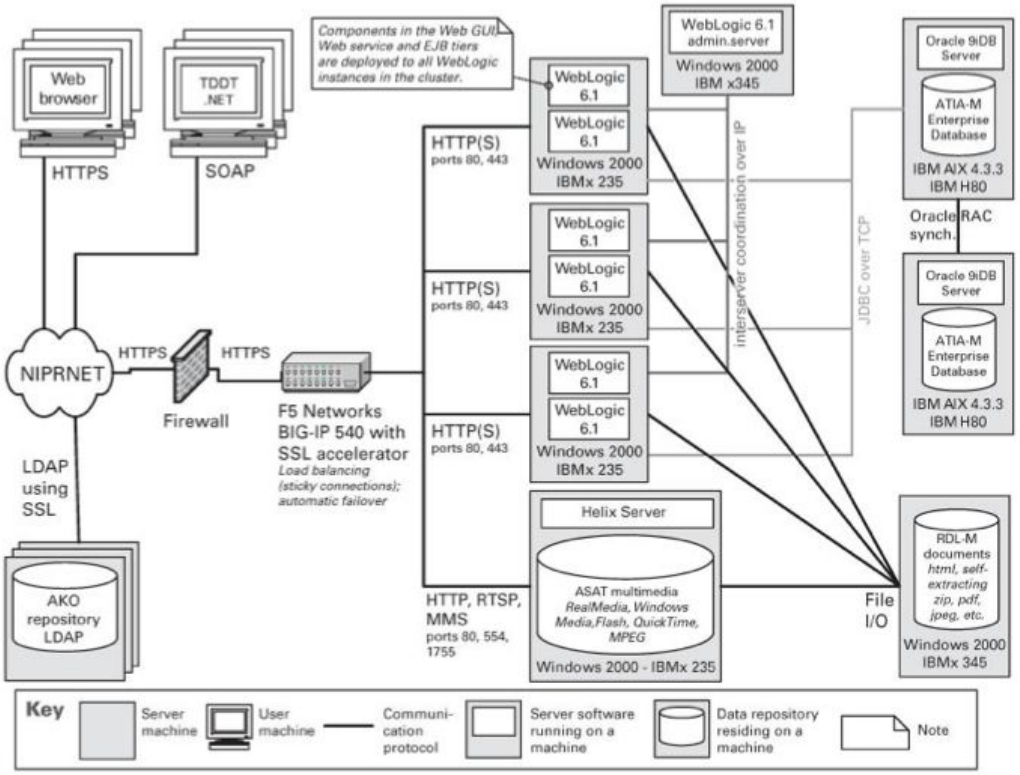
Documentación de arquitecturas: despliegue

- Vistas informales



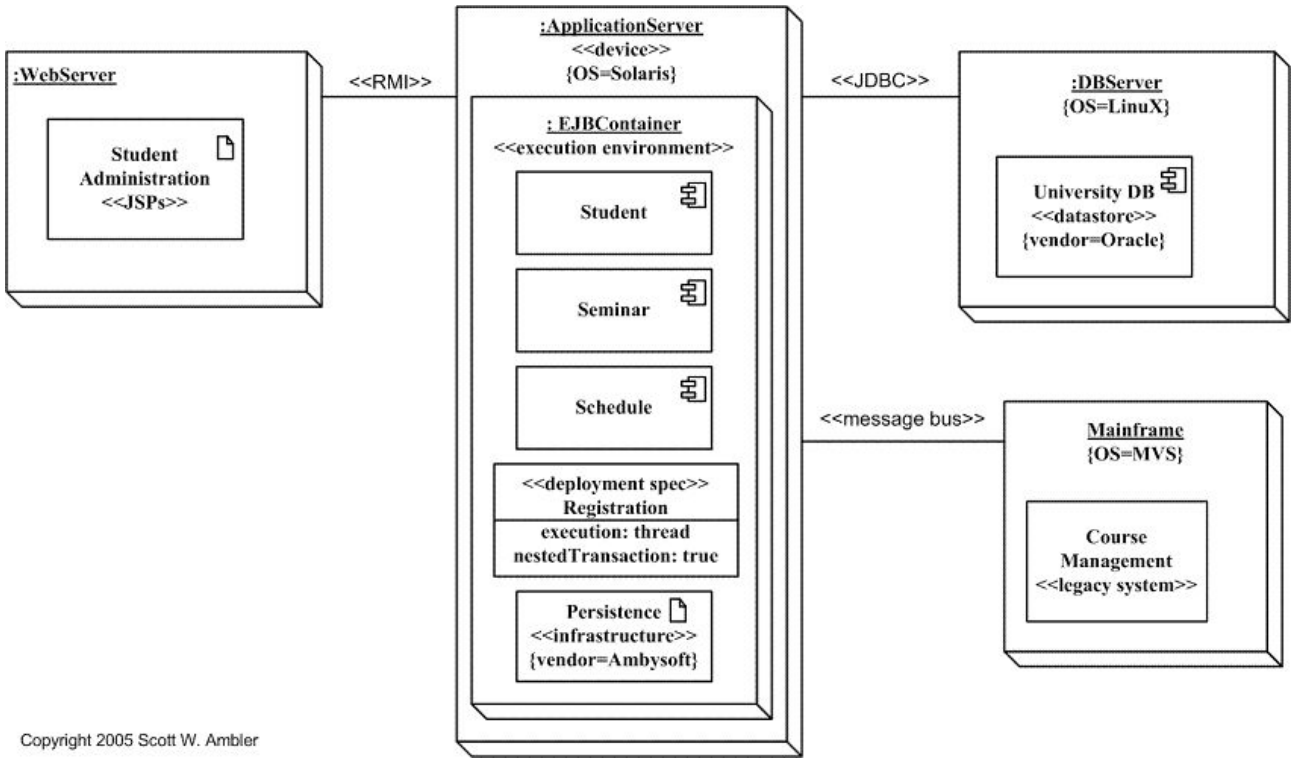
Documentación de arquitecturas: despliegue

- Vistas informales



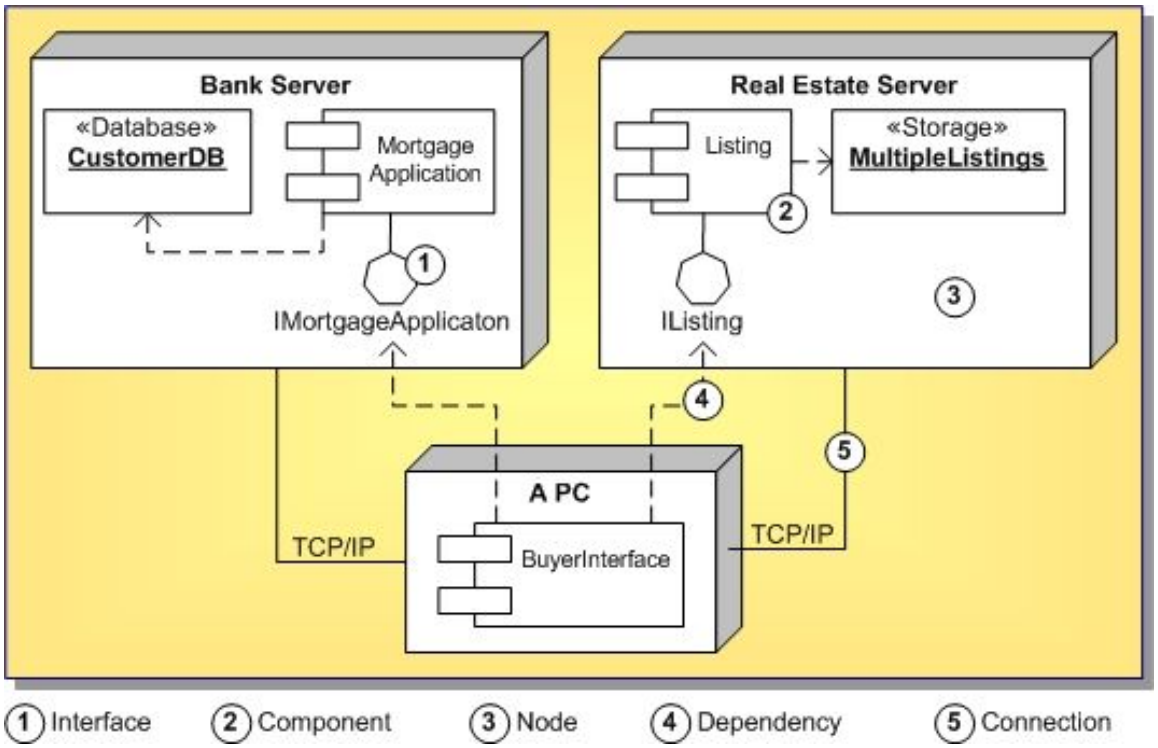
Documentación de arquitecturas: despliegue

- UML – Diagrama de despliegue (deployment)



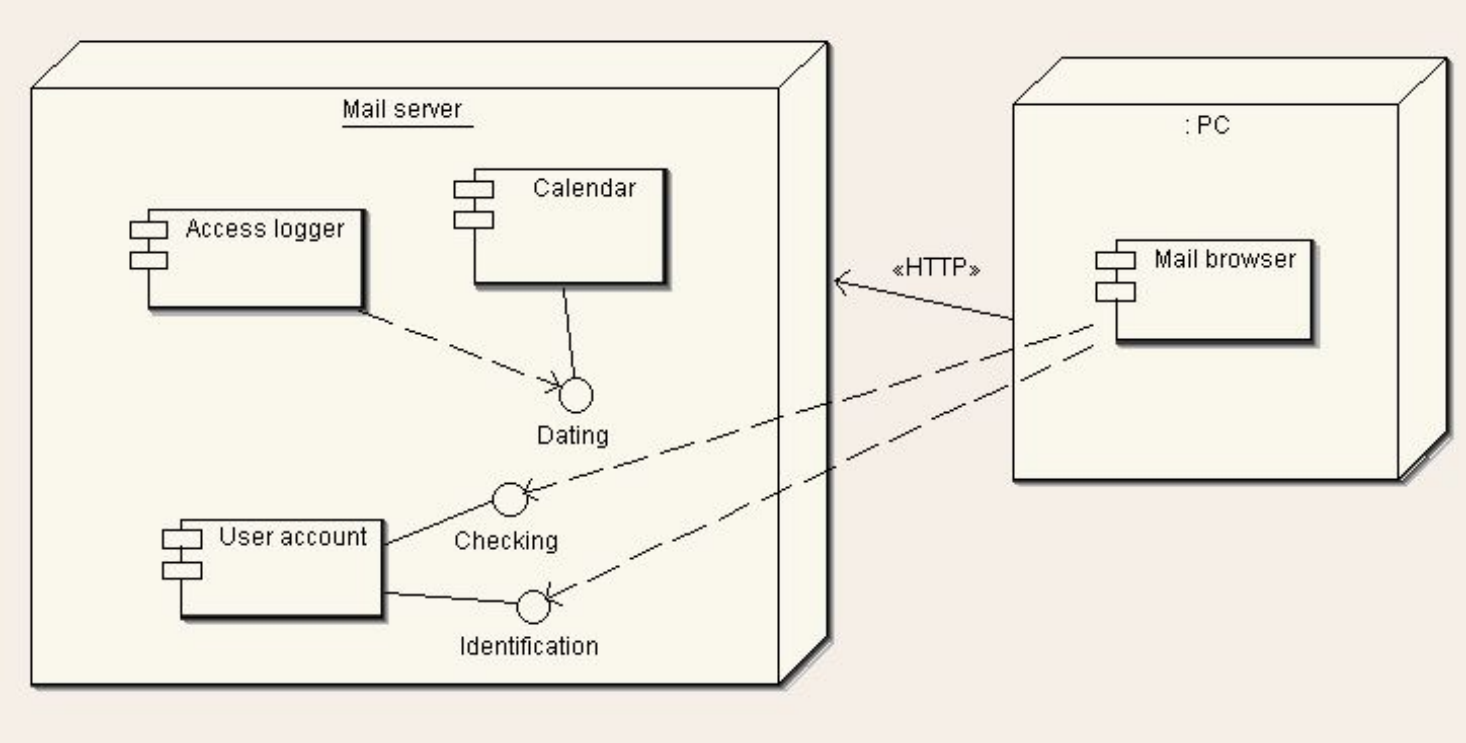
Documentación de arquitecturas: despliegue

- UML - Diagrama de despliegue (deployment)

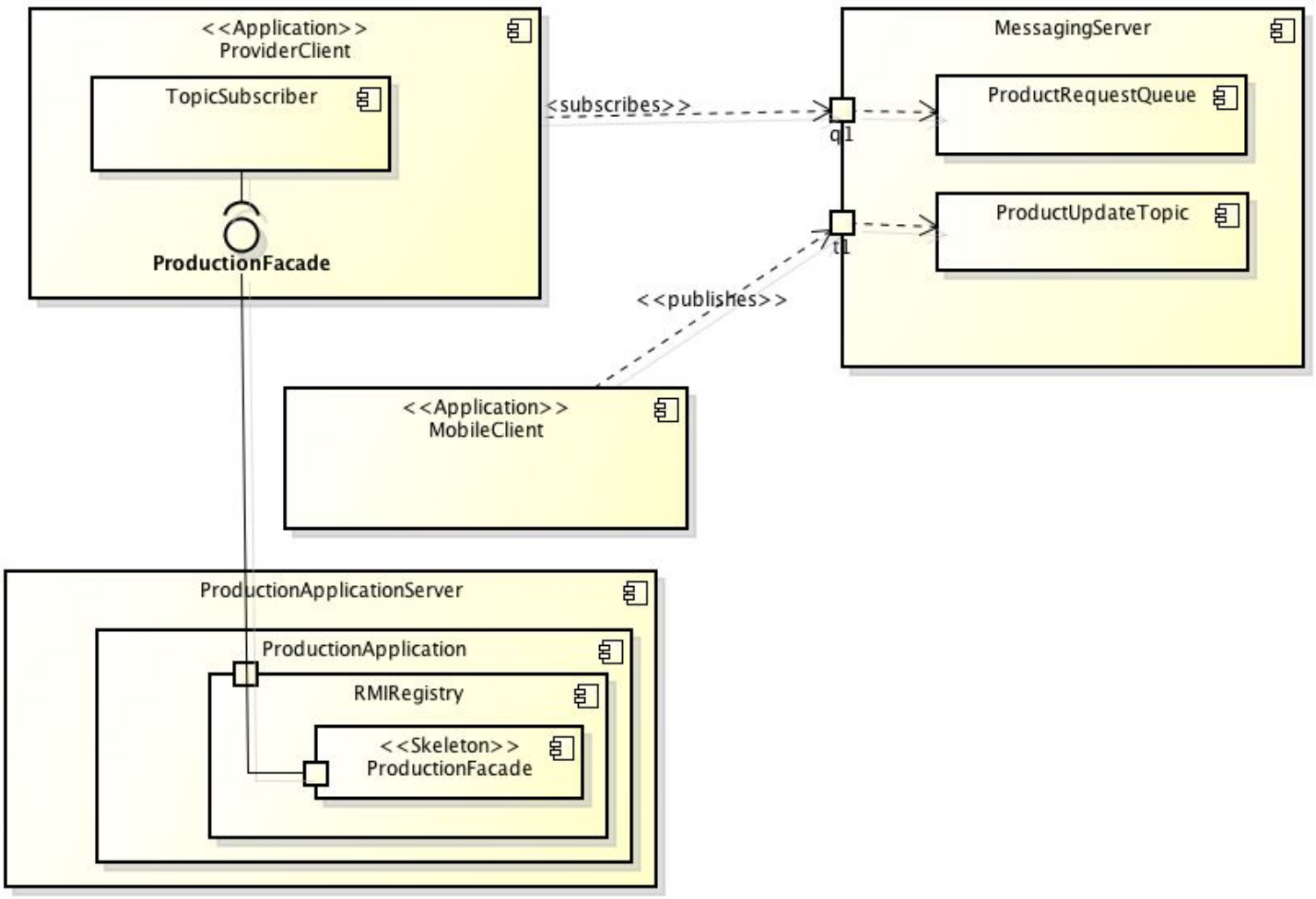


Documentación de arquitecturas: despliegue

- UML - Diagrama de despliegue (deployment)



Vistas estáticas (componentes)



Messaging Domains

- Point-to-Point (PTP)
  - built around the concept of message queues
  - each message has only one consumer
- Publish-Subscribe systems
  - uses a “topic” to send and receive messages
  - each message has multiple consumers

References

- *Some images and text extracted from www.rgoarchitects.com/Files/**ooprimer.ppt***

Books

- Fowler, Martin, Patterns of Enterprise Application Architecture, http://martinfowler.com/books.html#eaa
- Gamma, et al, Design Patterns, http://hillside.net/patterns/DPBook/DPBook.html

Blogs

- Miller, Alex, http://www.jroller.com/page/metalex
- Fowler, Martin, Inversion of Control Containers and the Dependency Injection pattern, http://www.martinfowler.com/articles/injection.html
- Oberg, Rickard, Dependency injection and open vs. closed designs, http://jroller.com/page/rickard/20040814#dependency_injection_and_open_vs
- Lee, Bob, Getter-Based Dependency Injection, http://weblogs.java.net/blog/crazybob/archive/2004/05/getterbased_dep.html
- Beust, Cedric, Getter-Based Injection, http://www.beust.com/weblog/archives/000134.html
- Denny, Mitch, IOC and .NET http://notgartner.com/posts/906.aspx
- Cazullino, Daniel, Lightweight Containers and Plugin Architectures: Dependency Injection and Dynamic Service Locators in .NET, http://weblogs.asp.net/cazzu/archive/2004/05/10/129140.aspx
- Weirich, Jim, Dependency Injection in Ruby, http://onestepback.org/index.cgi/Tech/Ruby/DependencyInjectionInRuby.rdoc
- Thomas, Dave, Transparent Inversion of Control, http://blogs.pragprog.com/cgi-bin/pragdave.cgi
- Mathew, Sony, Examining the Validity of Inversion of Control, http://stage.theserverside.com/articles/article.tss?l=IOCandEJB

Software Projects

- Spring, http://www.springframework.org/
- Pico, http://www.picocontainer.org/
- Hivemind, http://jakarta.apache.org/hivemind/
- Needle for Ruby, http://needle.rubyforge.org/