Distributed Applications

Software Engineering 2017

Alessio Gambi - Saarland University

Based on the work of Cesare Pautasso, Christoph Dorn, and other sources

ReCap

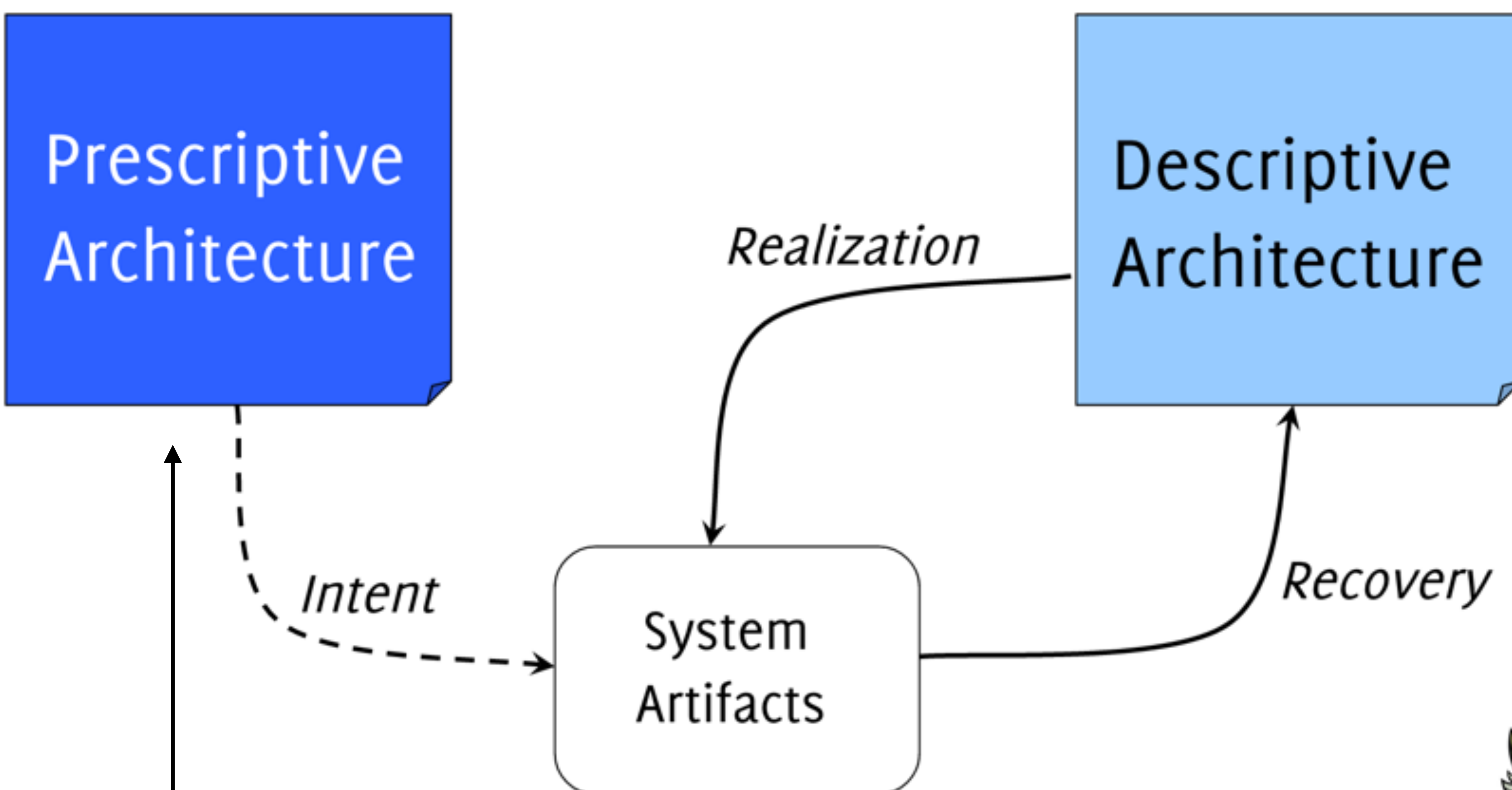
Software Architecture

A software system's architecture is the set of principal design decisions made about the system.

N. Taylor et al.

Abstraction	Communication
Visualization and Representation	Quality Attributes

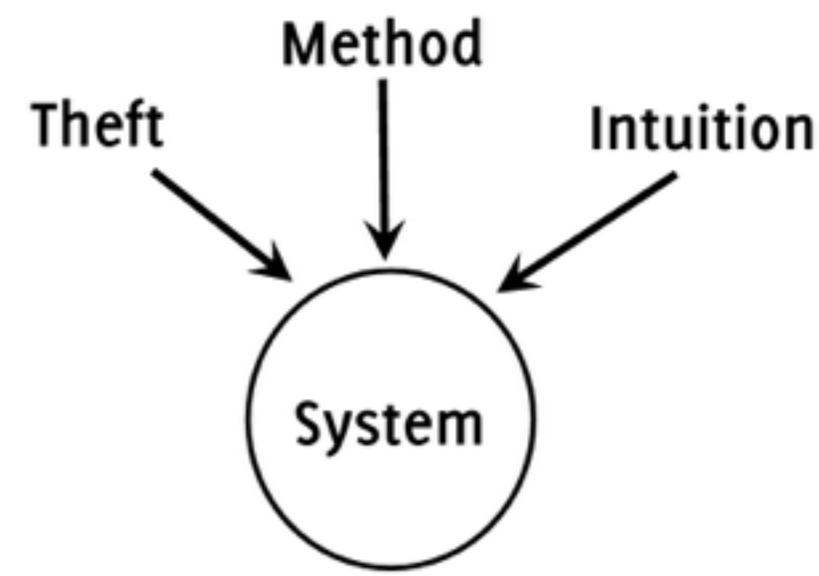
Every system a software architecture has



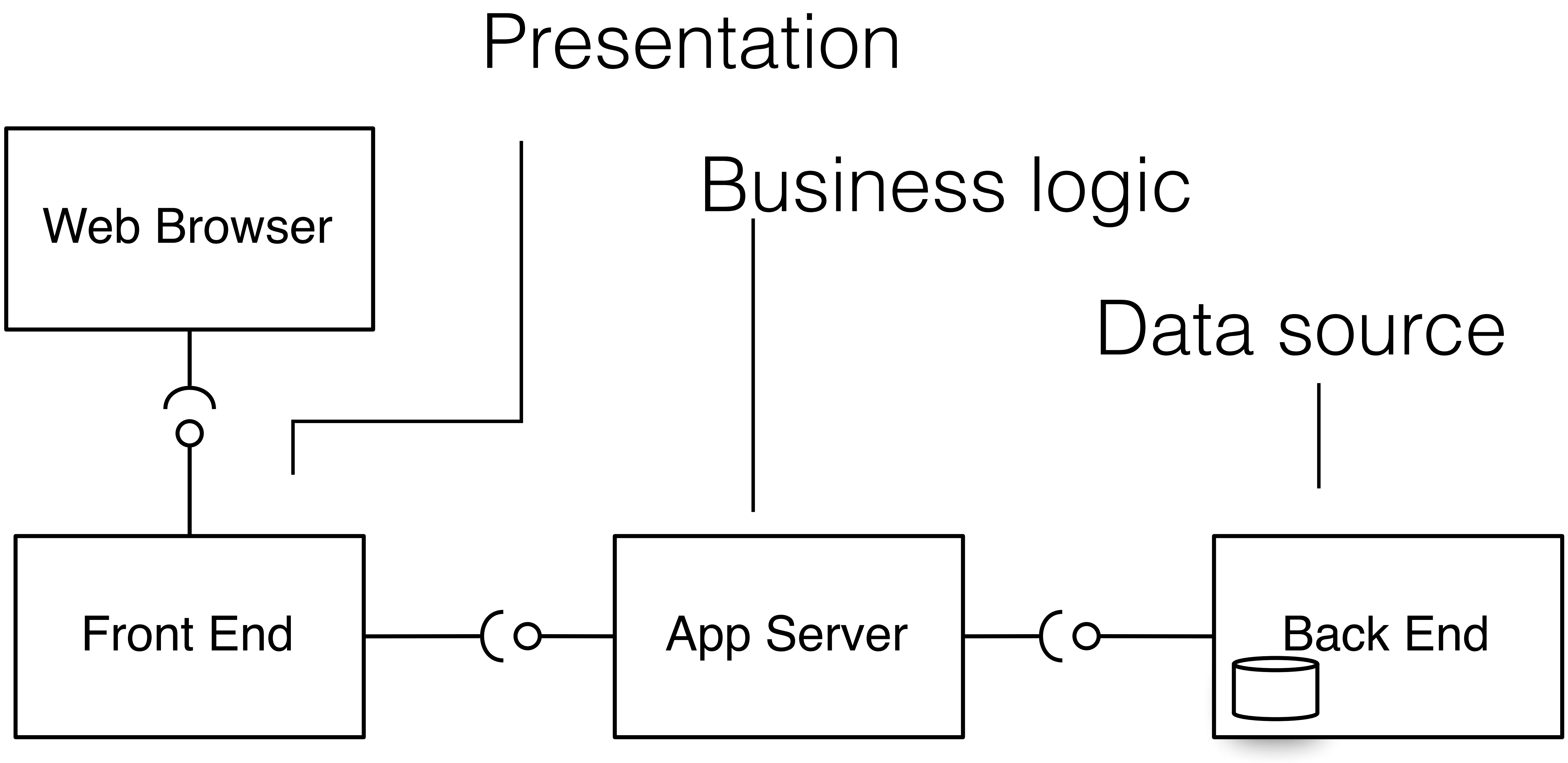
What designers want

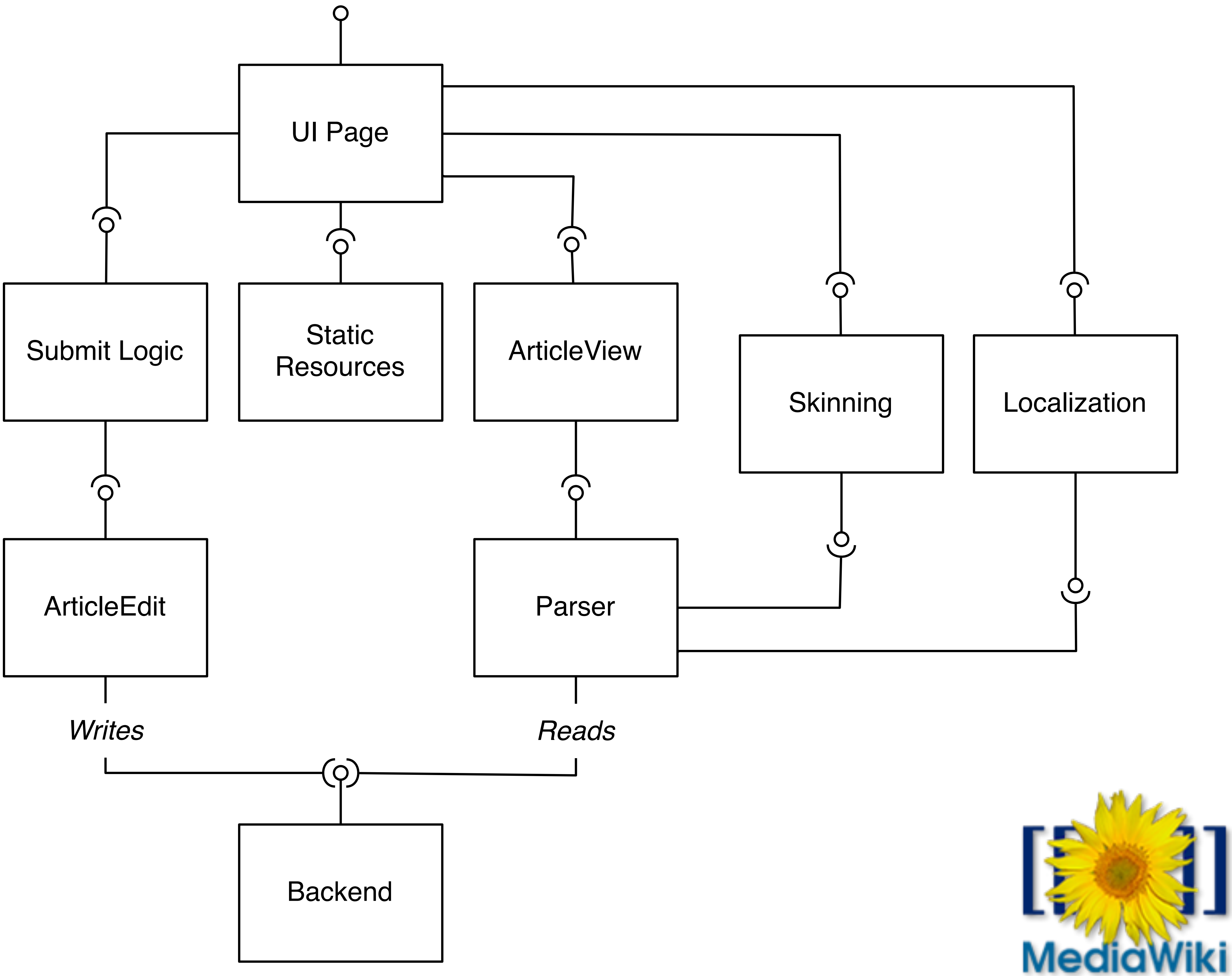
Design

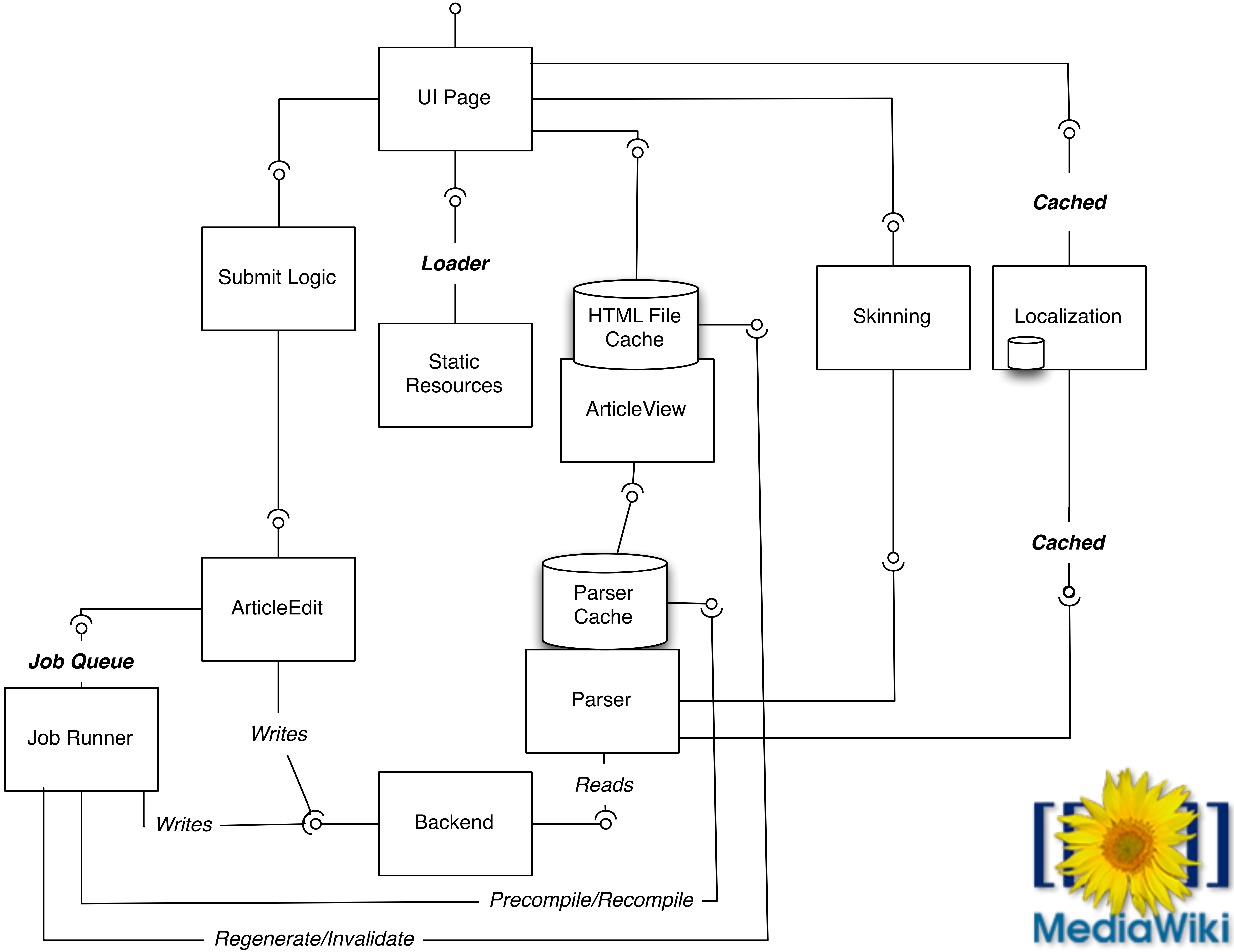
- Architectural Styles
- Architectural Patterns
- Building Blocks
  - *Software Components*
  - *Component API/Interfaces*
  - *Software Connectors*



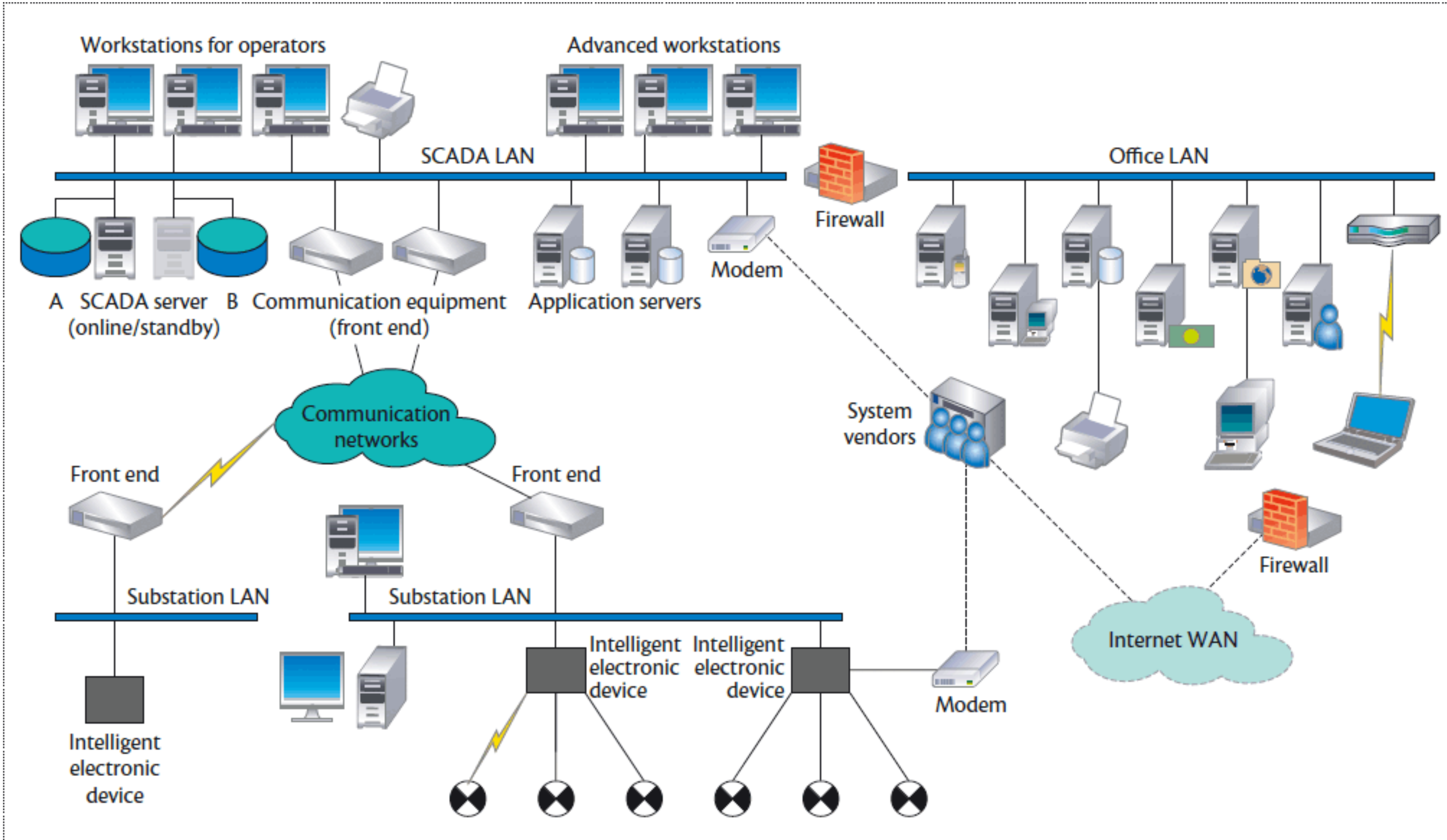
3-Tier Architecture




UI Page
Submit Logic
Static Resources
ArticleView
Skinning
Localization
ArticleEdit
Parser
Writes
Reads
Backend
MediaWiki

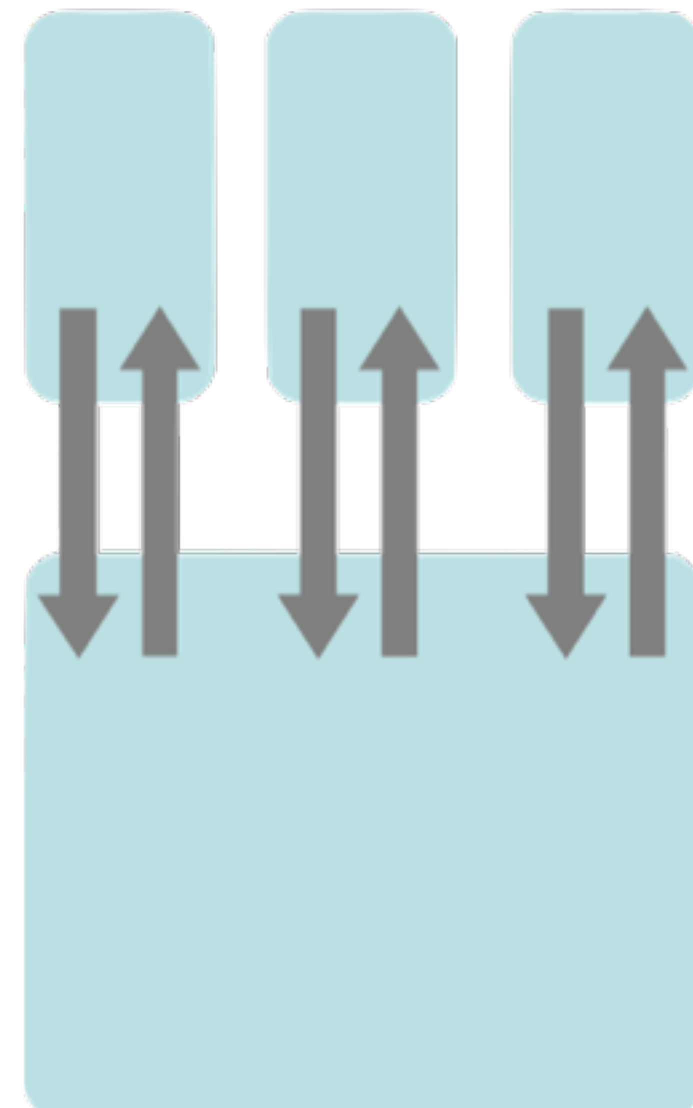

UI Page
Submit Logic
Loader
Static Resources
HTML File Cache
ArticleView
Skinning
Cached
Localization
Cached
ArticleEdit
Parser Cache
Parser
Job Queue
Job Runner
Writes
Reads
Writes
Backend
Precompile/Recompile
Regenerate/Invalidate
MediaWiki

Distributed Applications



Client/Server

- Many clients, active, close to users
- One server, passive, close to data
- Single point of failure, scalability
- Security, scalability



Distribution and Lifecycle

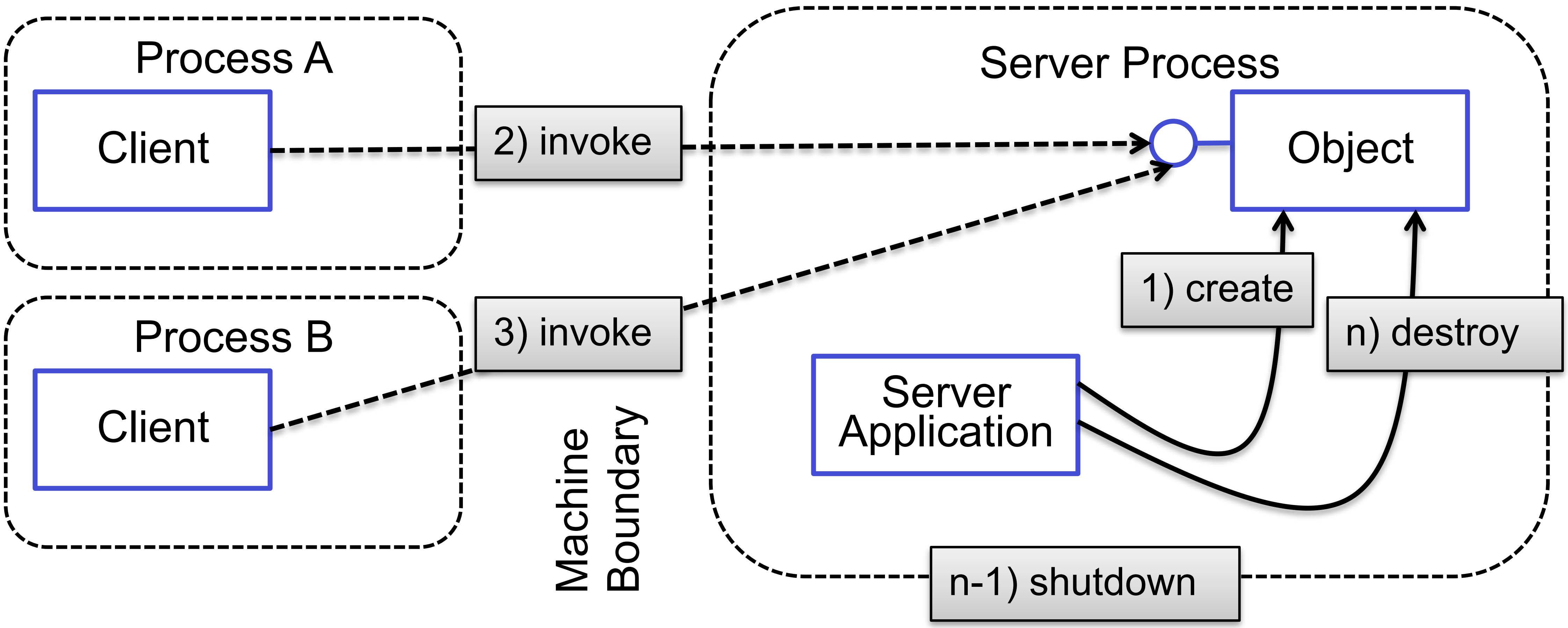
In distributed applications the lifecycle of remote objects is disjoint from the local ones.
We must explicitly design the lifecycle of those remote entities

Distribution and Lifecycle

In distributed applications the lifecycle of remote objects is disjoint from the local ones.
We must explicitly design the lifecycle of those remote entities

Static and Lazy instances	Leasing
Per-request instances	Pooling
Client-dependent instances	Passivation

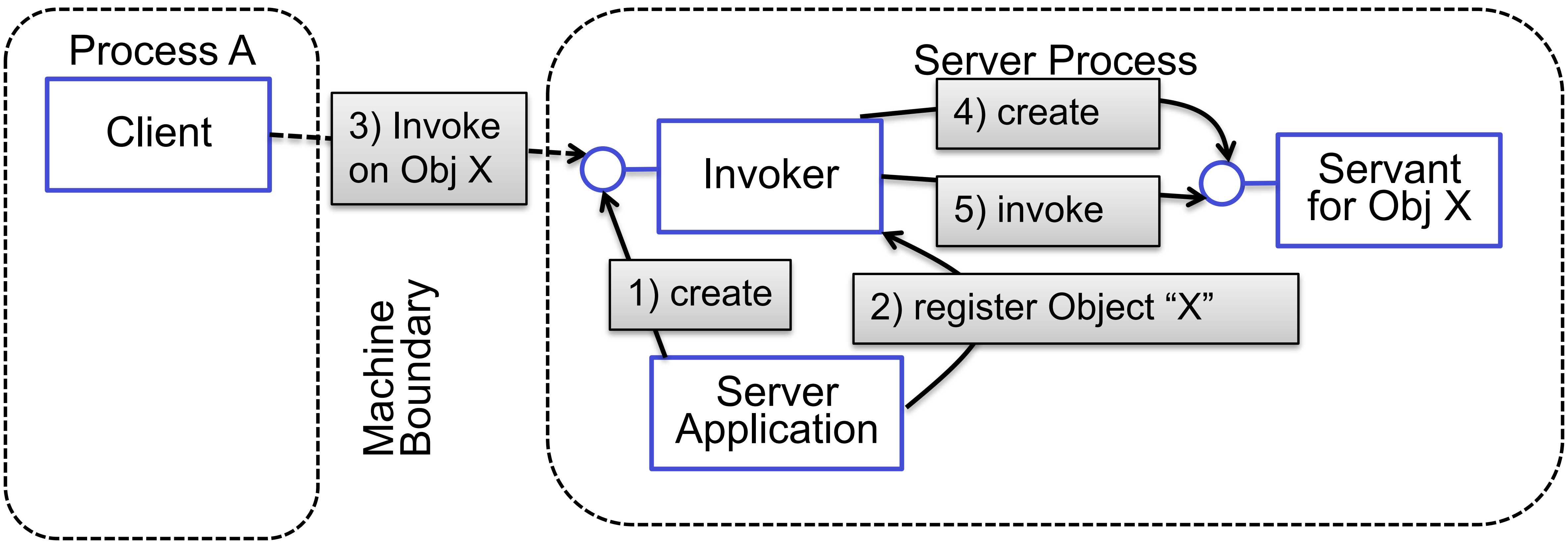
Static Instances



Remote object instances exist independently of any clients

They last as long as their container (server)

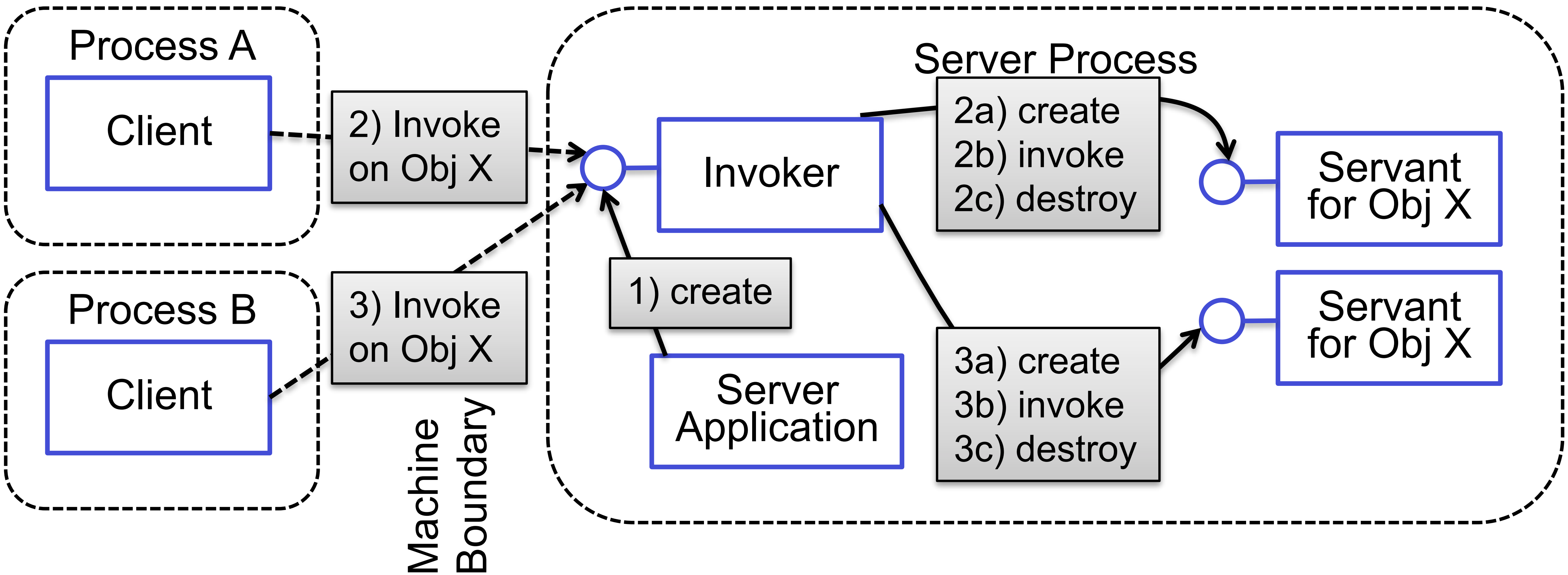
Lazy Instances



Instantiate object upon first request

Save computational resources

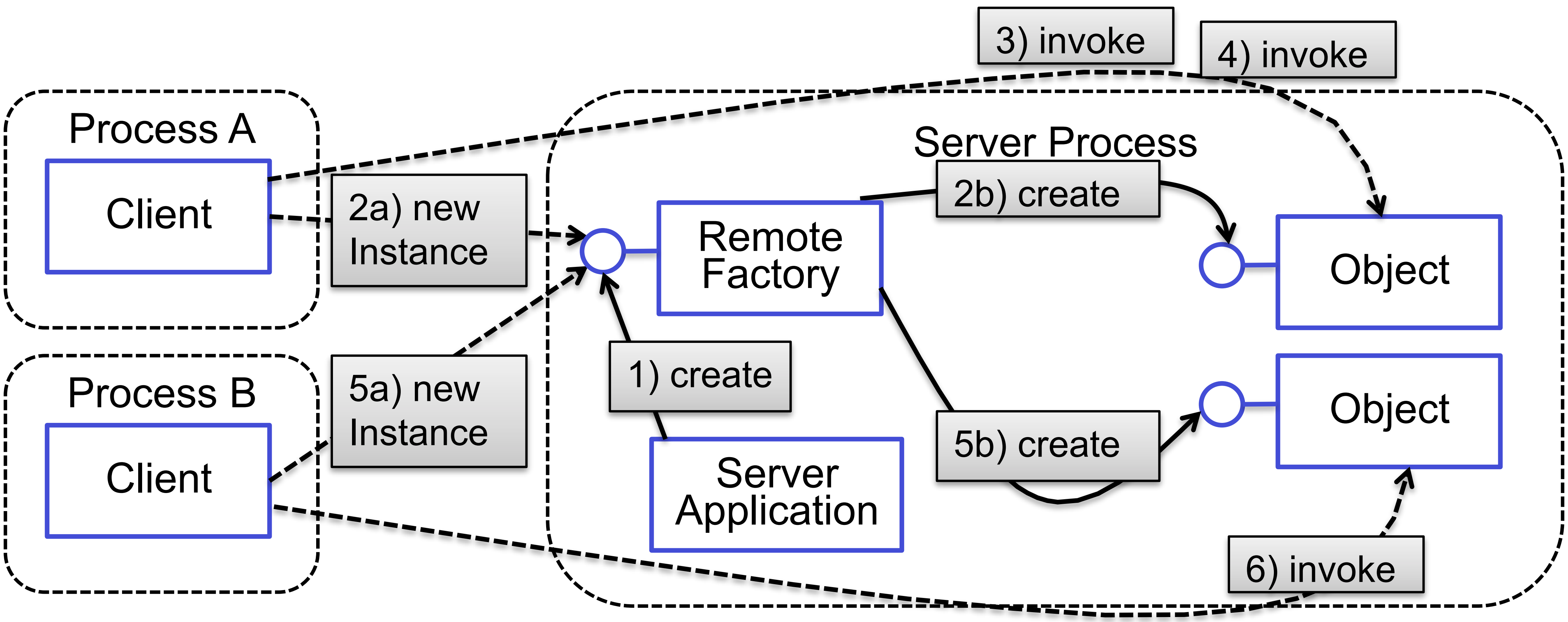
Per-request Instances



Each request processed by a fresh instance

Provide max logical isolation (but high cost)

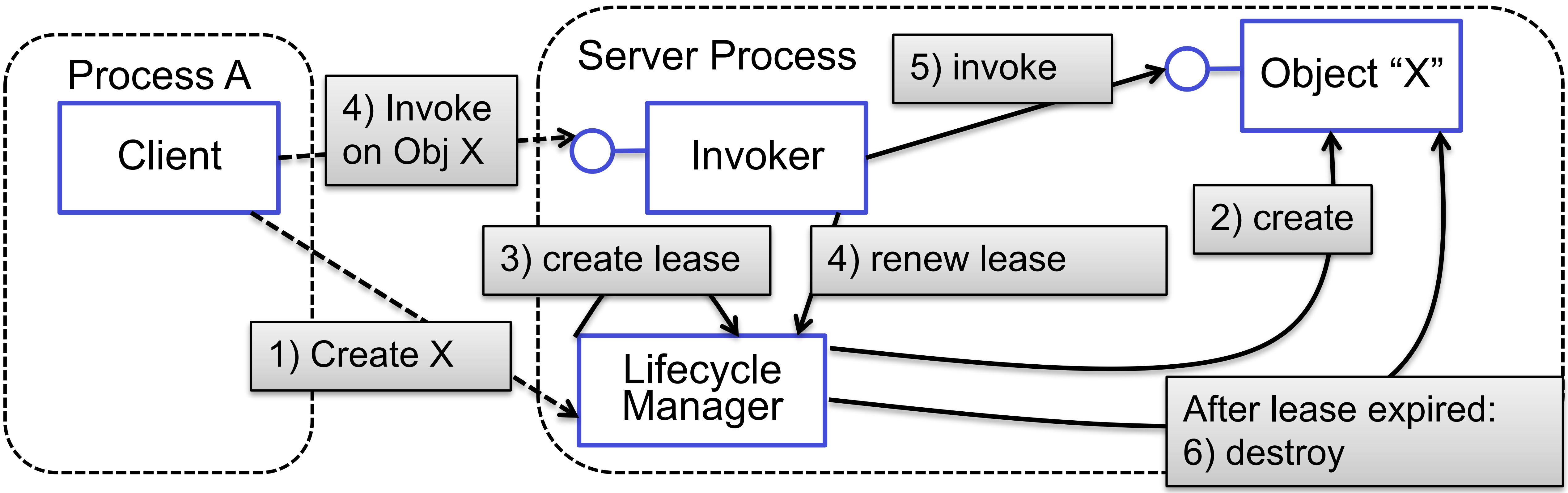
Client-dependent Instances



Requests from the same client processed by the same instance (but there might be a one-to-many mapping)

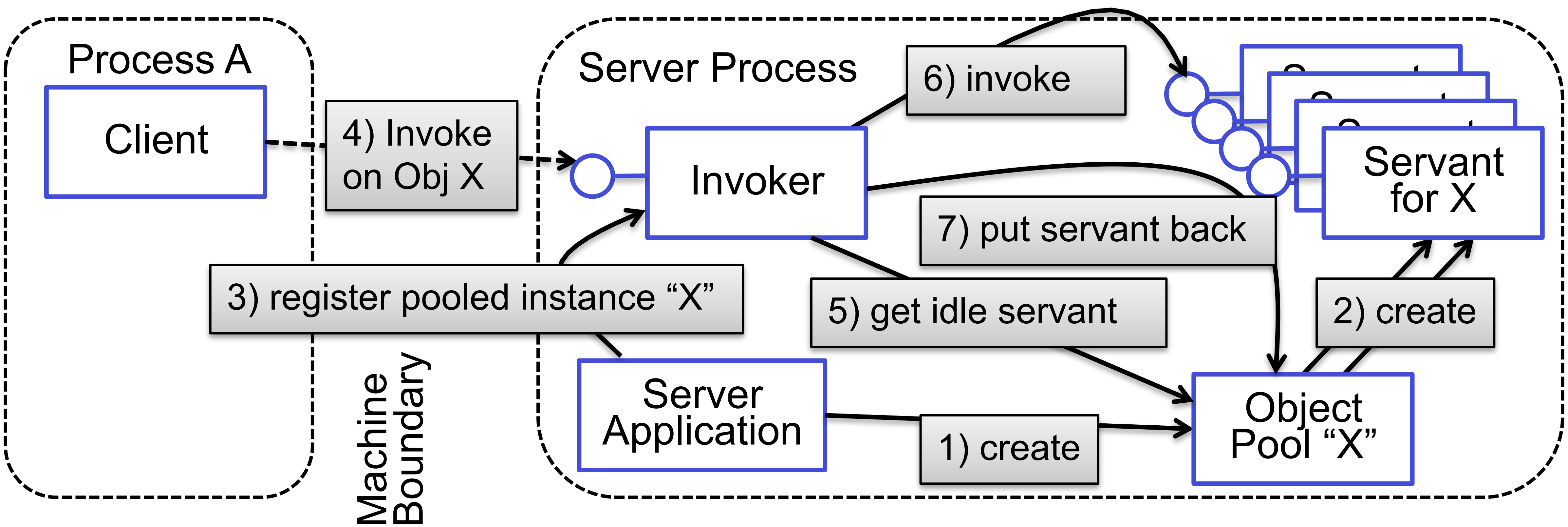
Remote objects extend client logic and share its state

Leasing



Avoid removal of per-client objects when not used by periodically renew the lease

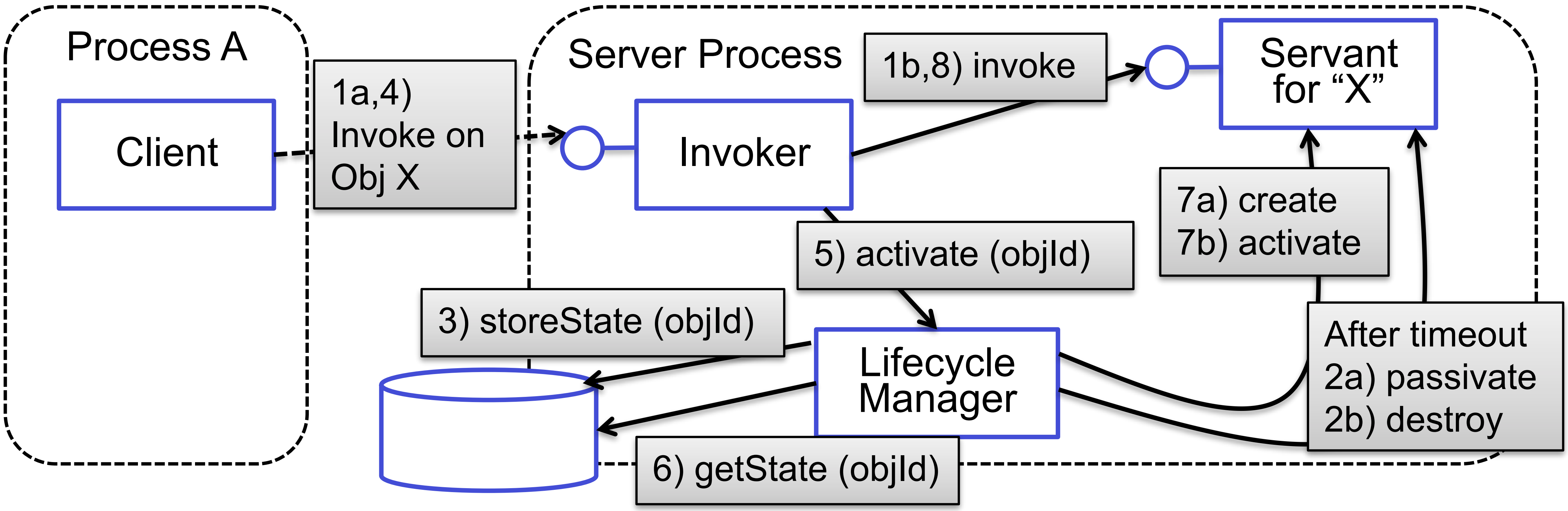
Pooling



Maintain a (possibly dynamic) set of generic objects to serve clients requests

Clean up state before returning to the pool

Passivation



Save resources by freezing “per-client” objects

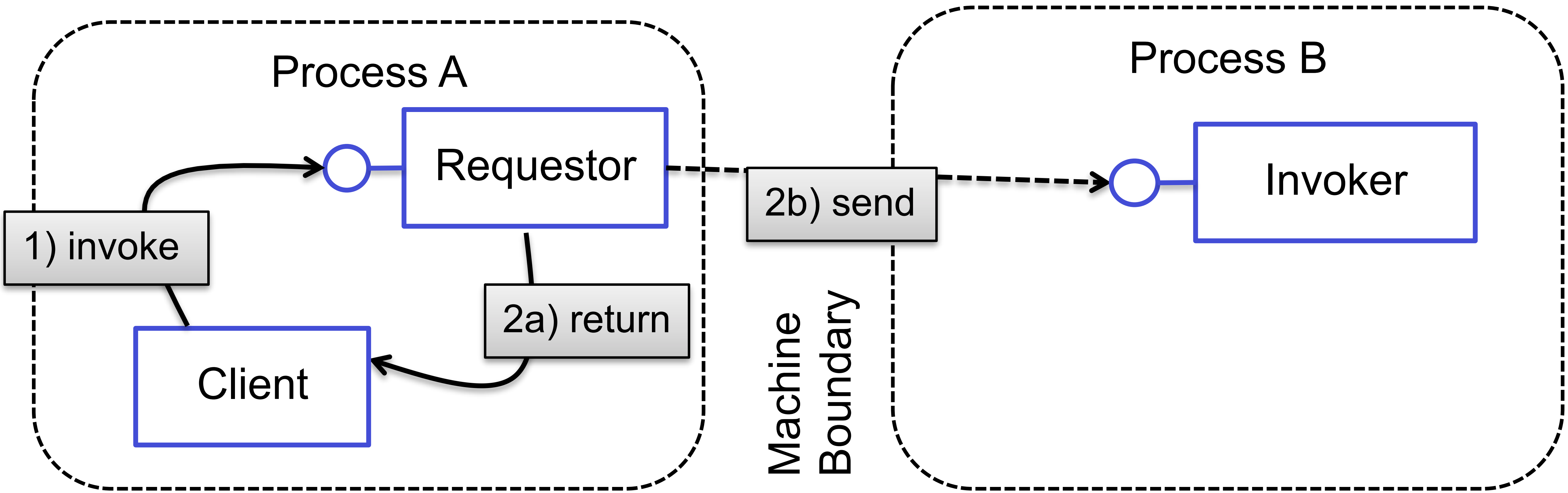
Objects are reactivated upon first request

(A)Synchronization

Remote invocations can be either synchronous or asynchronous. For asynchronous invocations we must handle the evolution of the distributed state across the nodes.

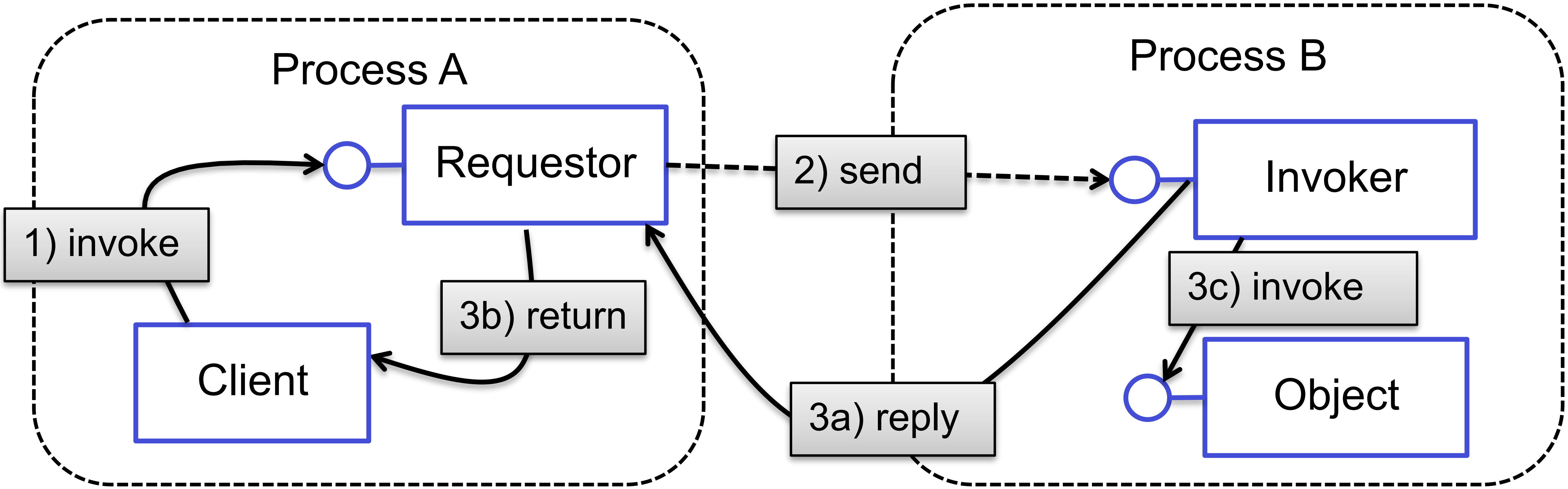
One-way Patterns	**Two-way Patterns**
Fire and Forget	Poll Object
Sync with Server	Callback

Fire and Forget



Best effort (or nobody cares) semantics

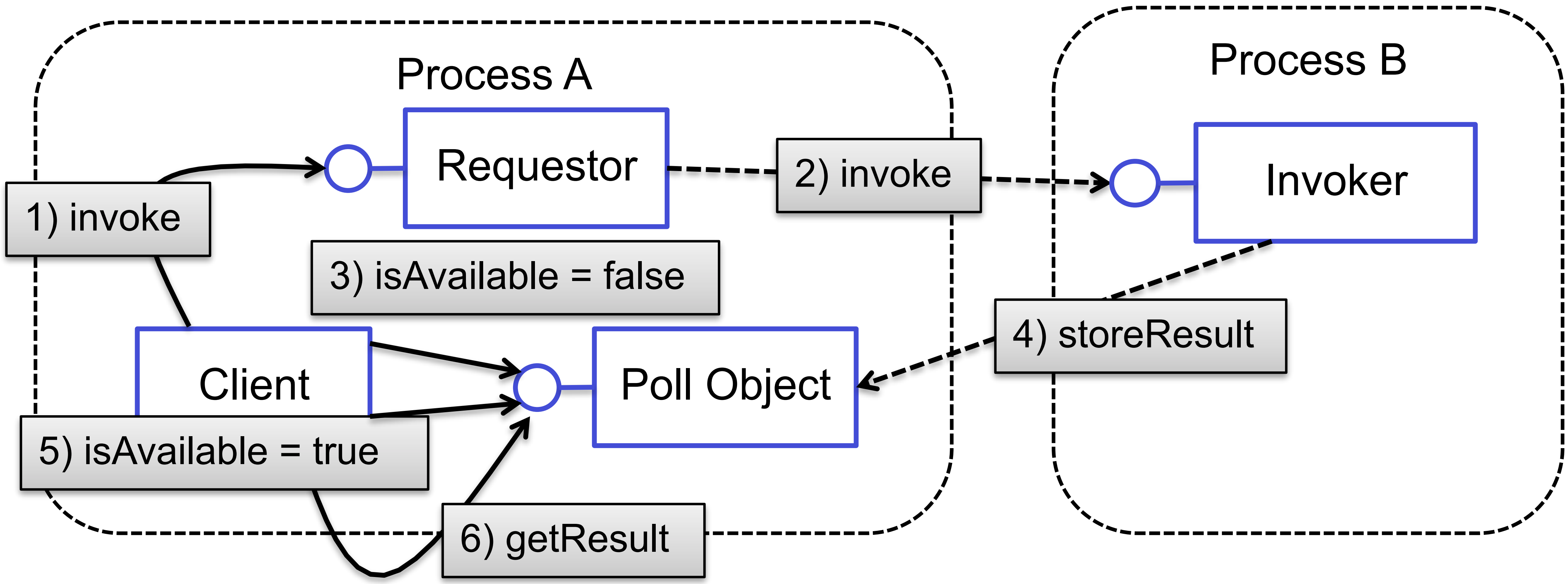
Sync with Server



Requestor ensures that the request correctly arrived to server (but not processed)

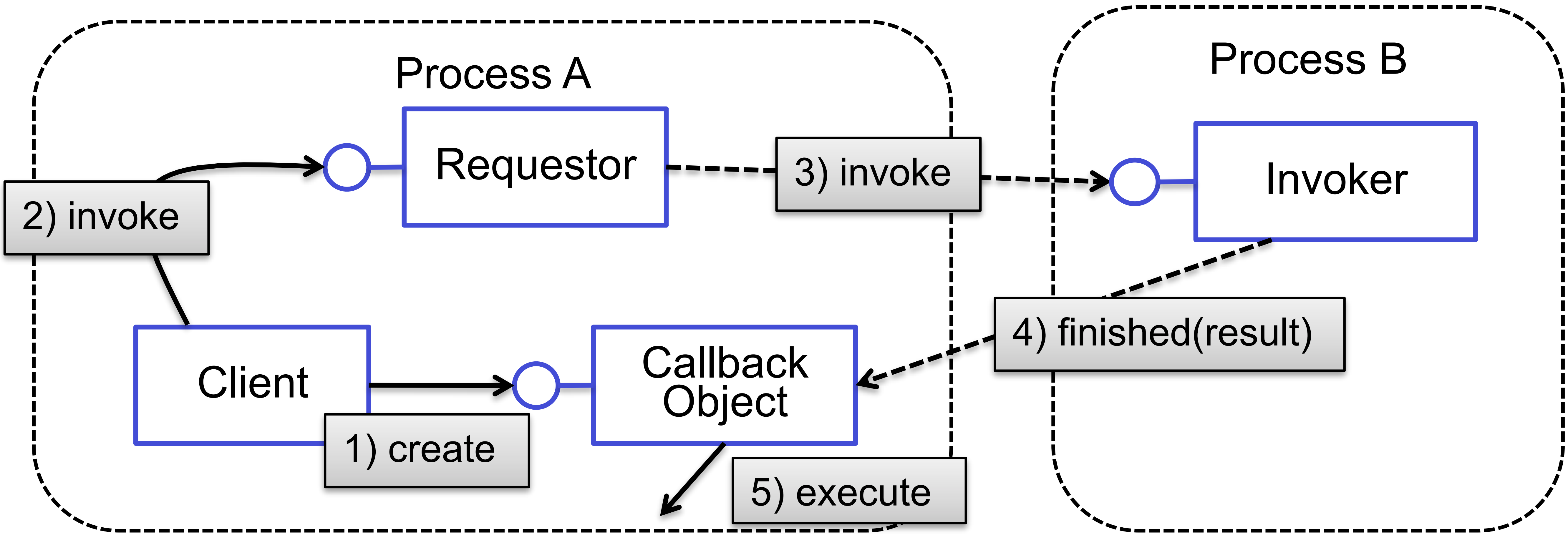
Delivery confirmation semantics

Poll Object (or Future)



Local stub on client's machine checks if results are ready

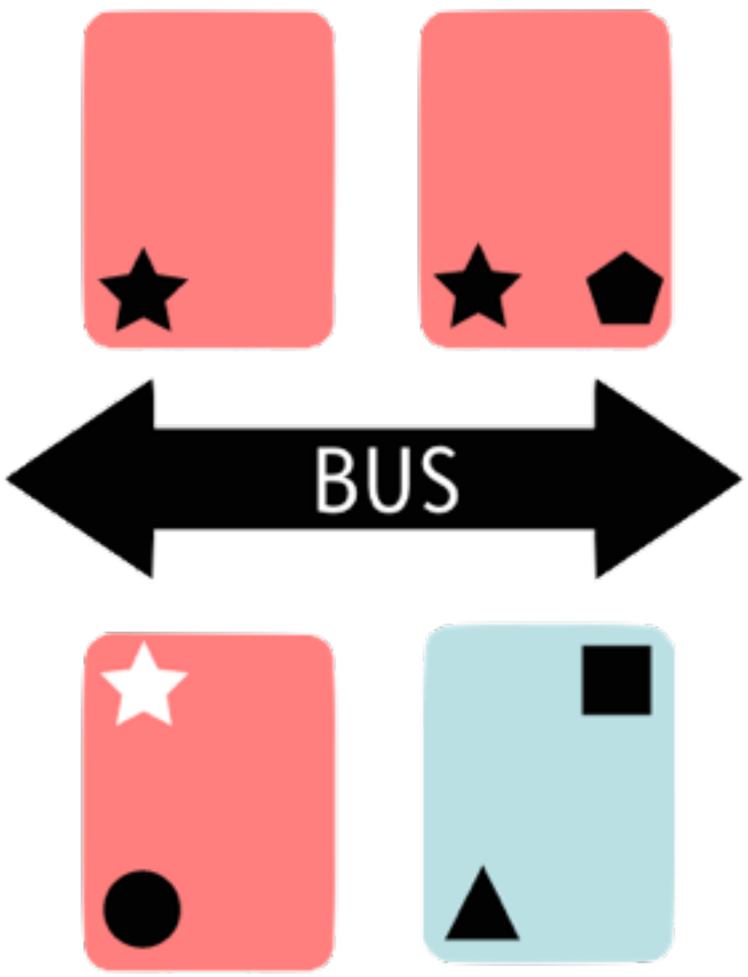
Callback



Execute code whenever the remote request returns

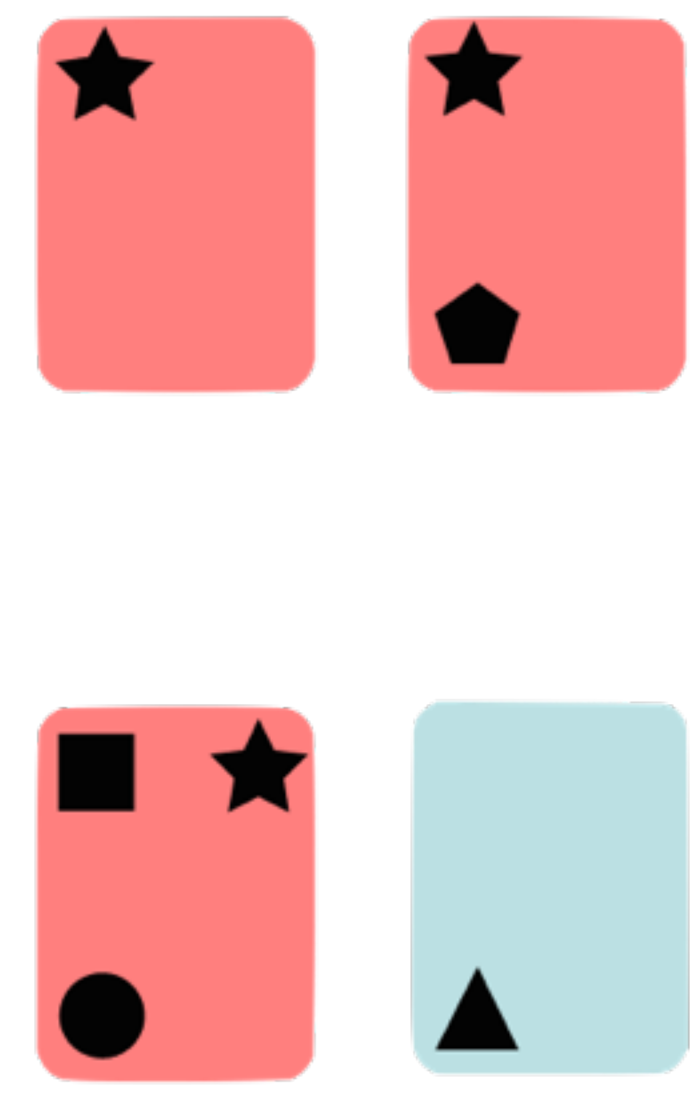
Publish/Subscribe

- Subscription to queues or topics
- Loose coupling



Pub/Sub vs Event-Driven

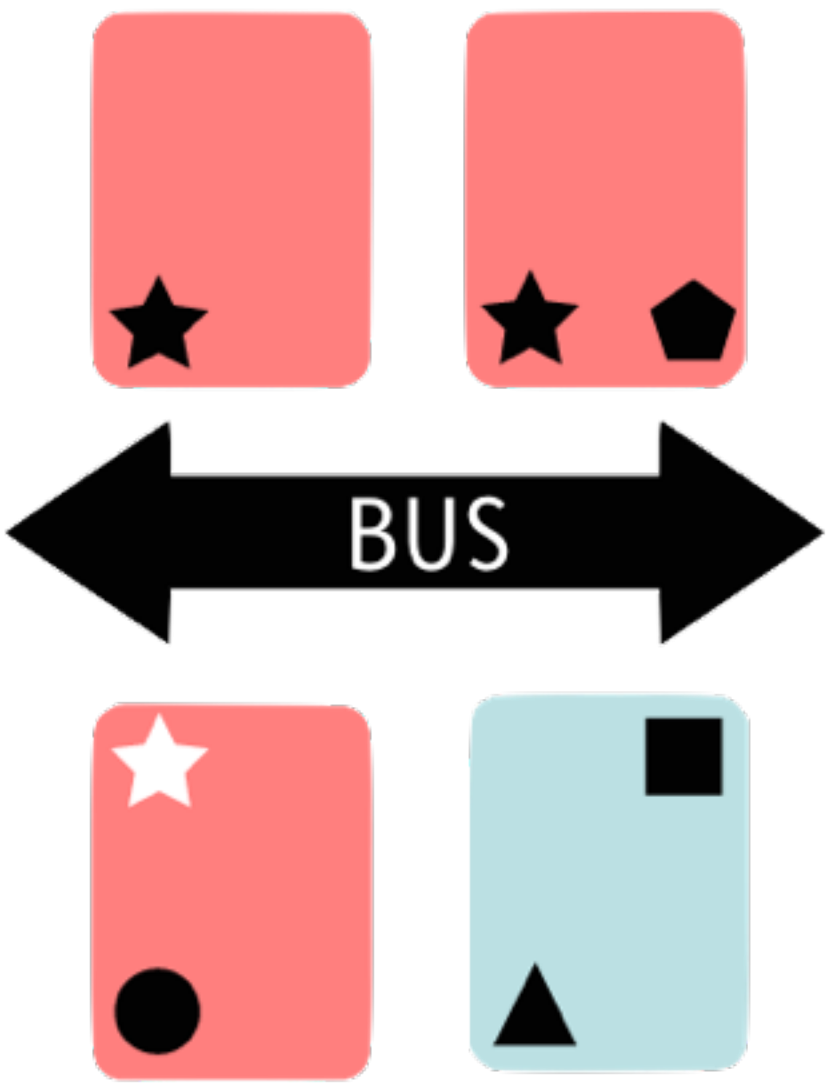
Pub/Sub vs Event-Driven



no specific roles

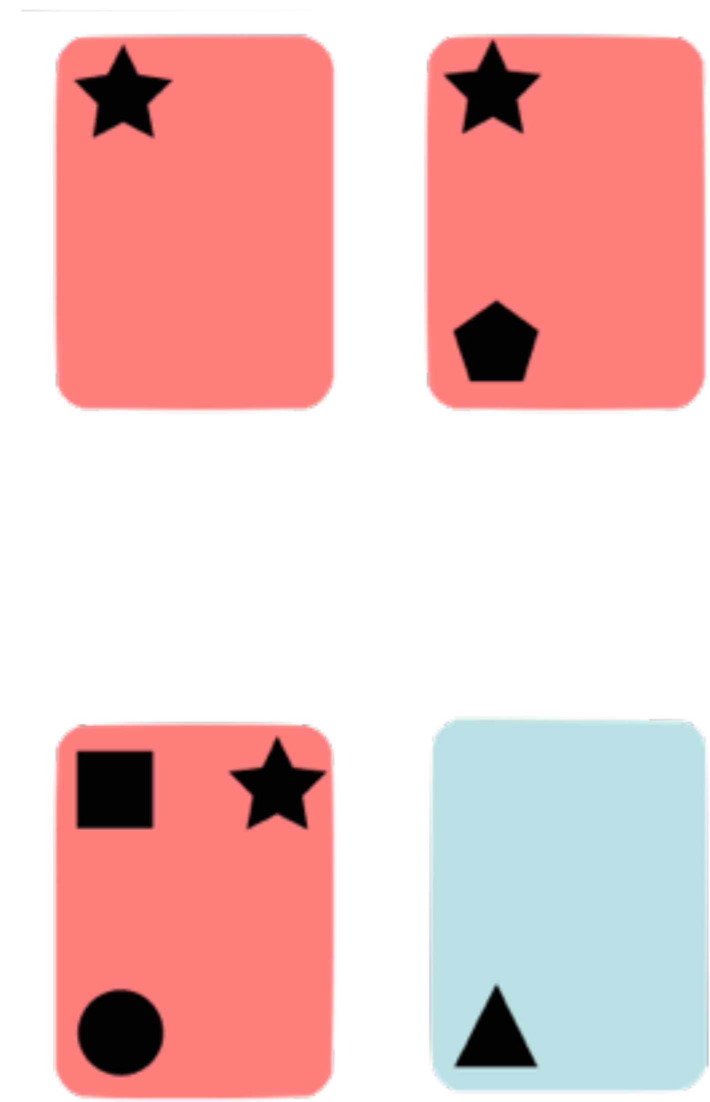
local/distributed

Pub/Sub vs Event-Driven



opposite roles

mostly distributed

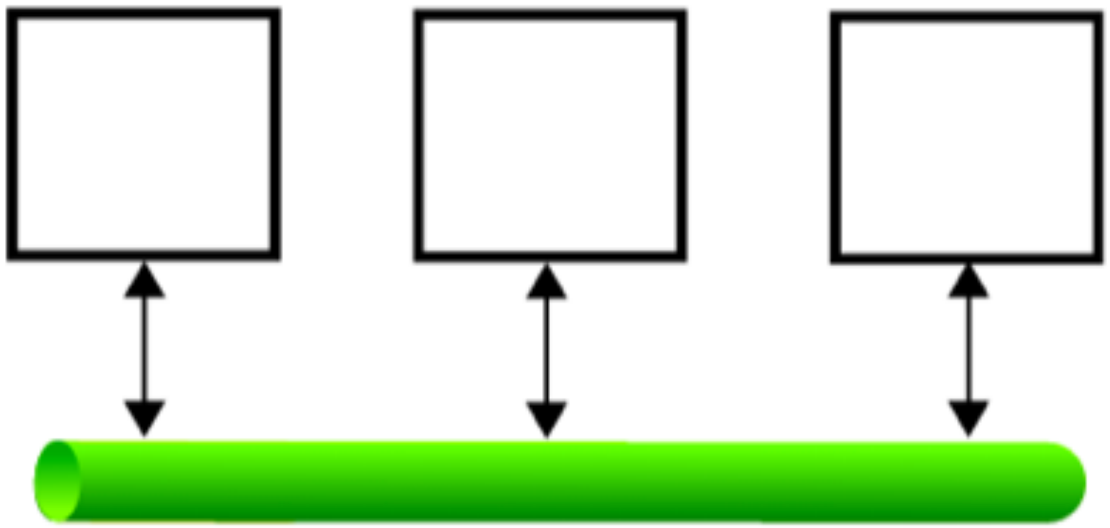


no specific roles

local/distributed

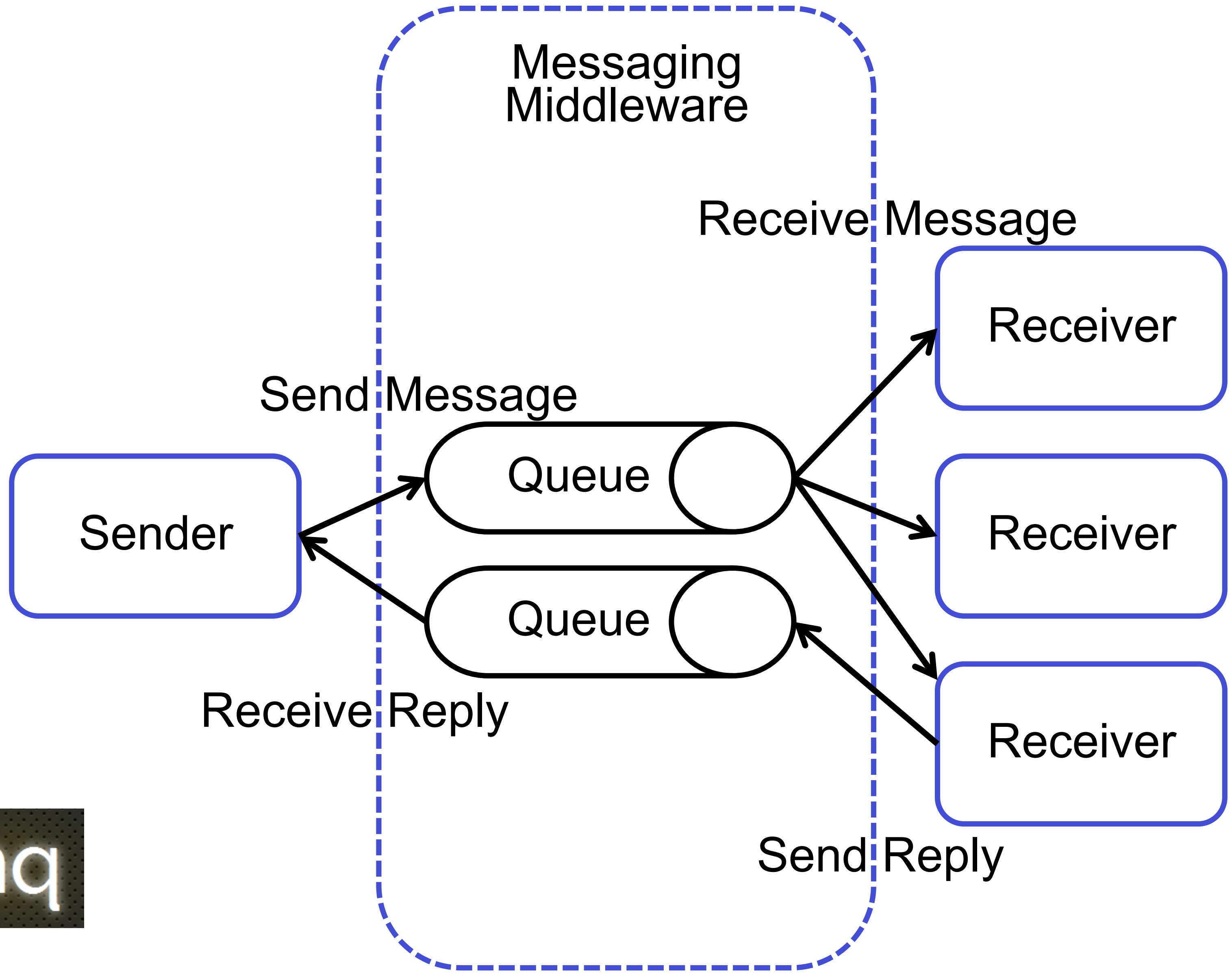
Message Bus

- Publish
- Subscribe
- Notify



MOM

Message-Oriented Middleware



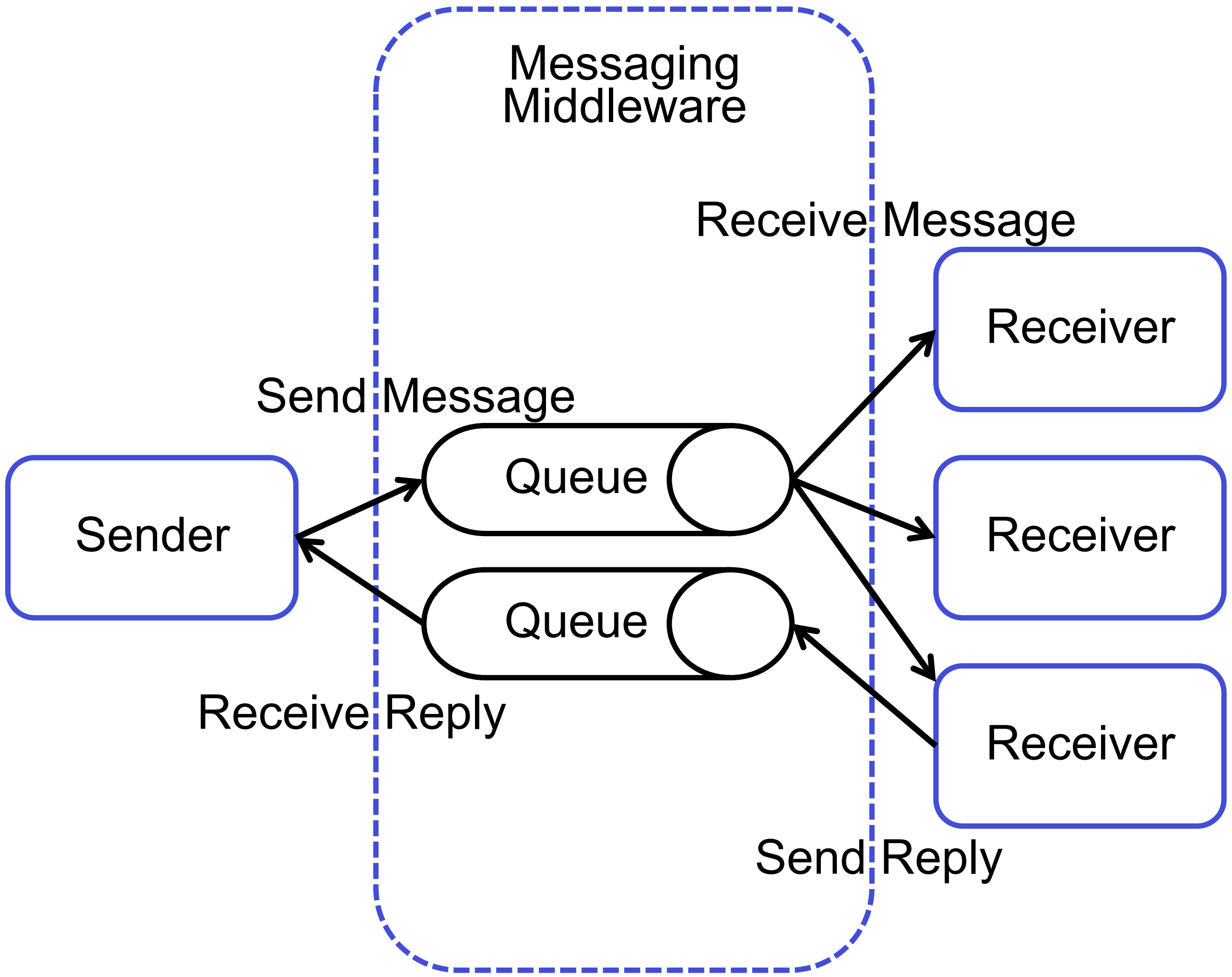
JMS

stormmq

RabbitMQ
Messaging that just works

MOM

Message-Oriented Middleware



- Processing always on consumer
- Queues provide persistence and decoupling (async)

Reply or don’t reply?

MOMs enable both request-only and request-reply interactions despite sender/receiver do not know each other addresses

Reply or don’t reply?

MOMs enable both request-only and request-reply interactions despite sender/receiver do not know each other addresses

Uniquely identify a request message (ID)

+

MessageType=REQUEST|REPLY & MessageID = ID

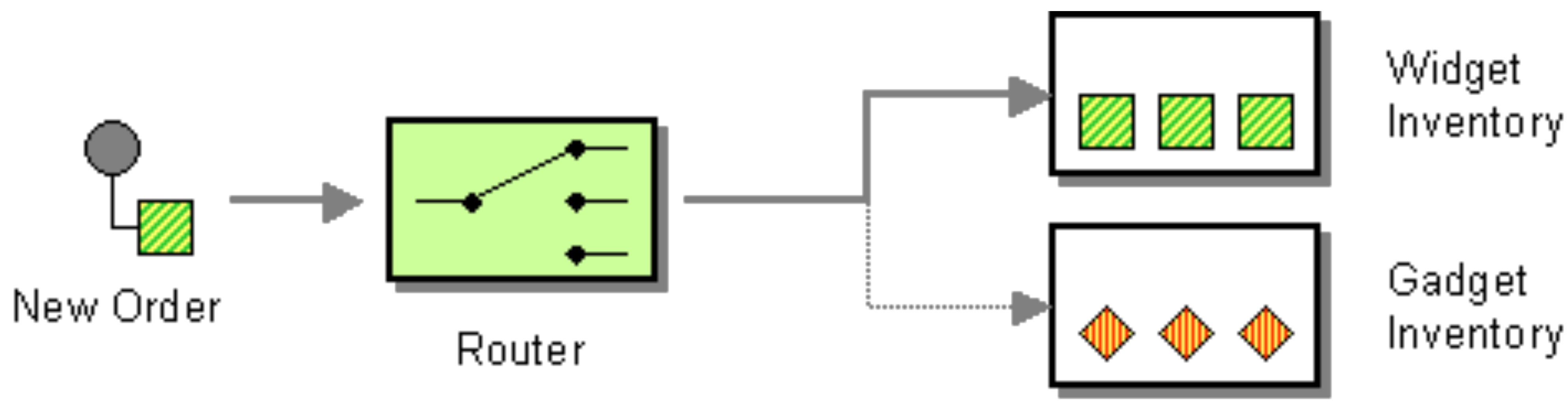
=

Correlation between the requests and replies

Handling Messages

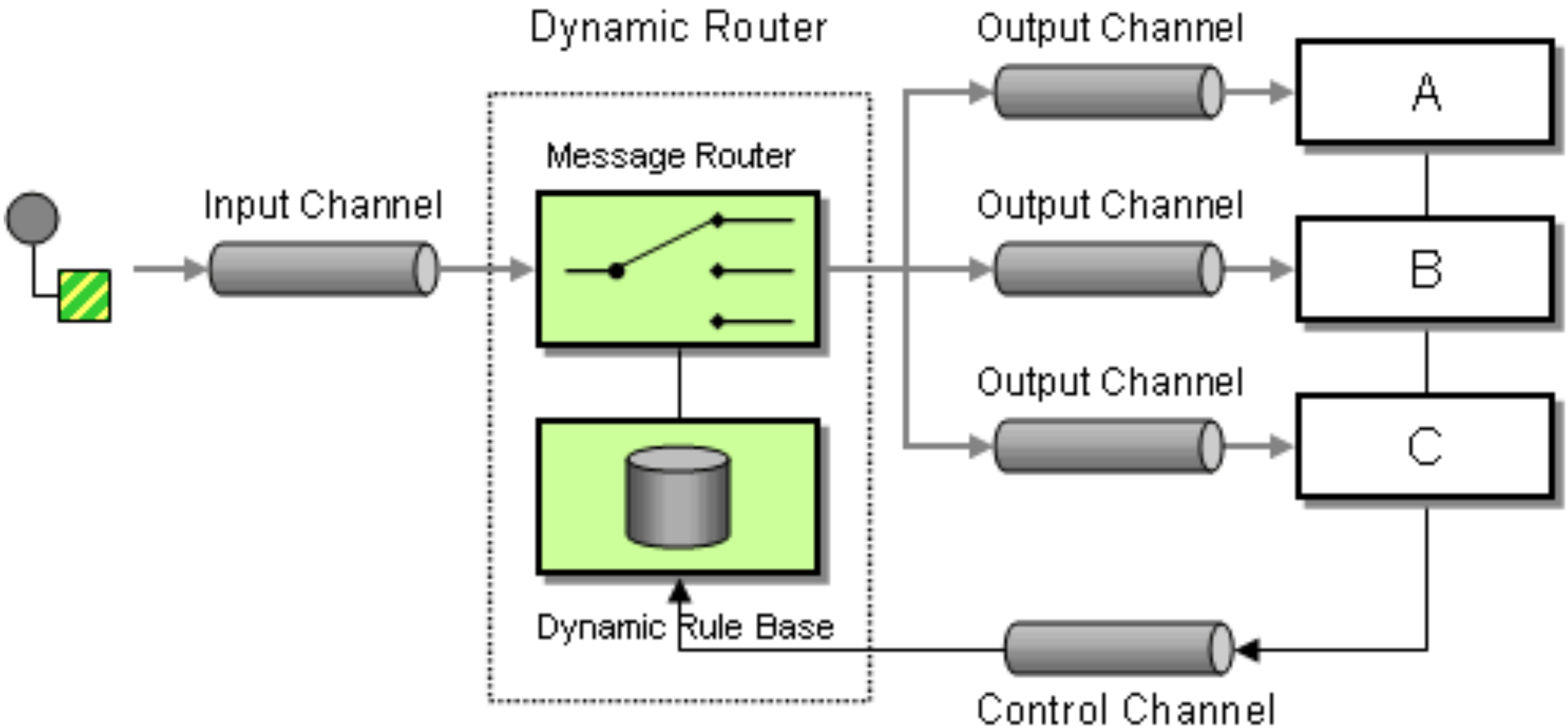
- Routing
  Content-based, Dynamic
- Filtering
  Message filter
- Transforming messages
  Splitter, Aggregator
- Transforming messages content
  Normalizer, Content Enricher, Content Filter
- Transforming message envelope
  Envelope wrapper

Content-based Routing



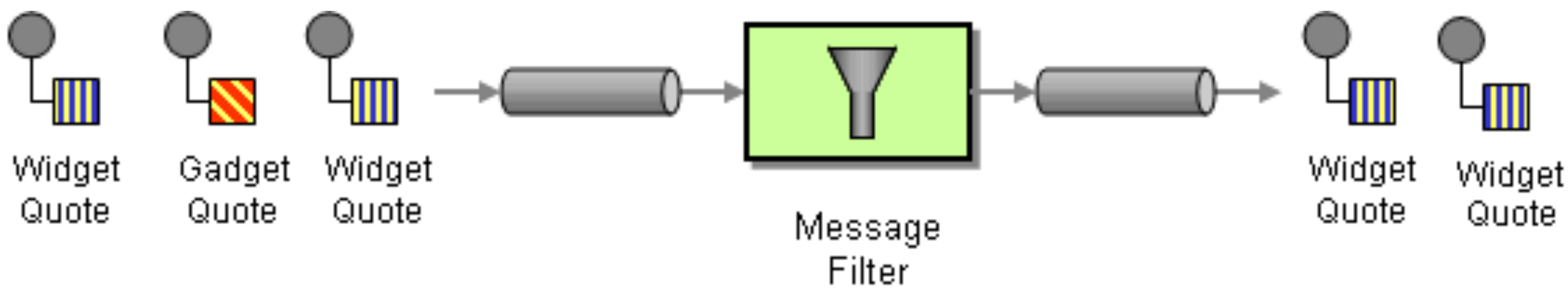
Destination decided using the payload

Dynamic Routing



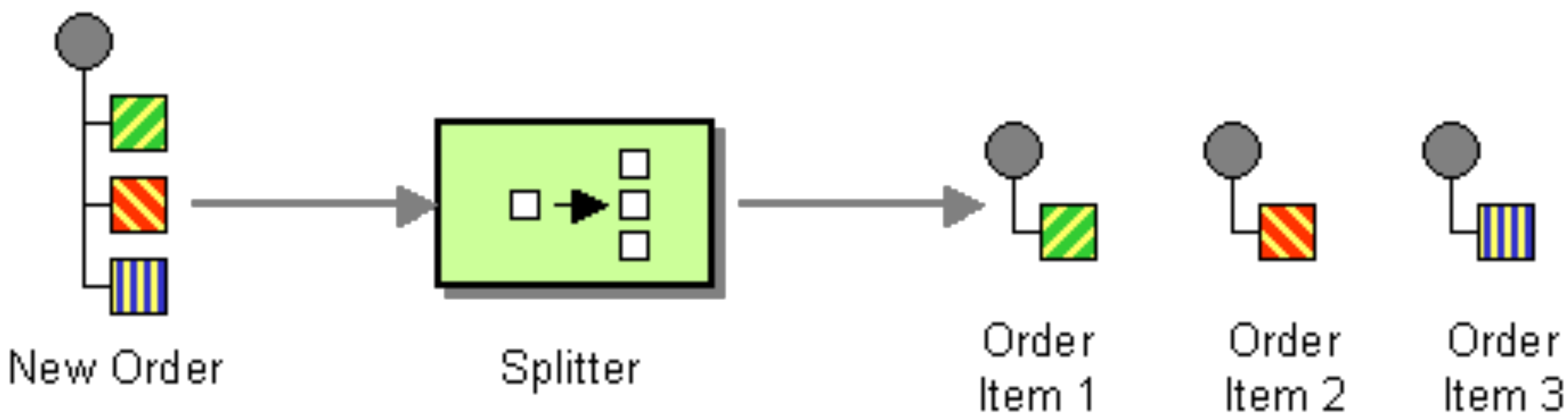
Destination not fixed but chosen using rules

Message Filter



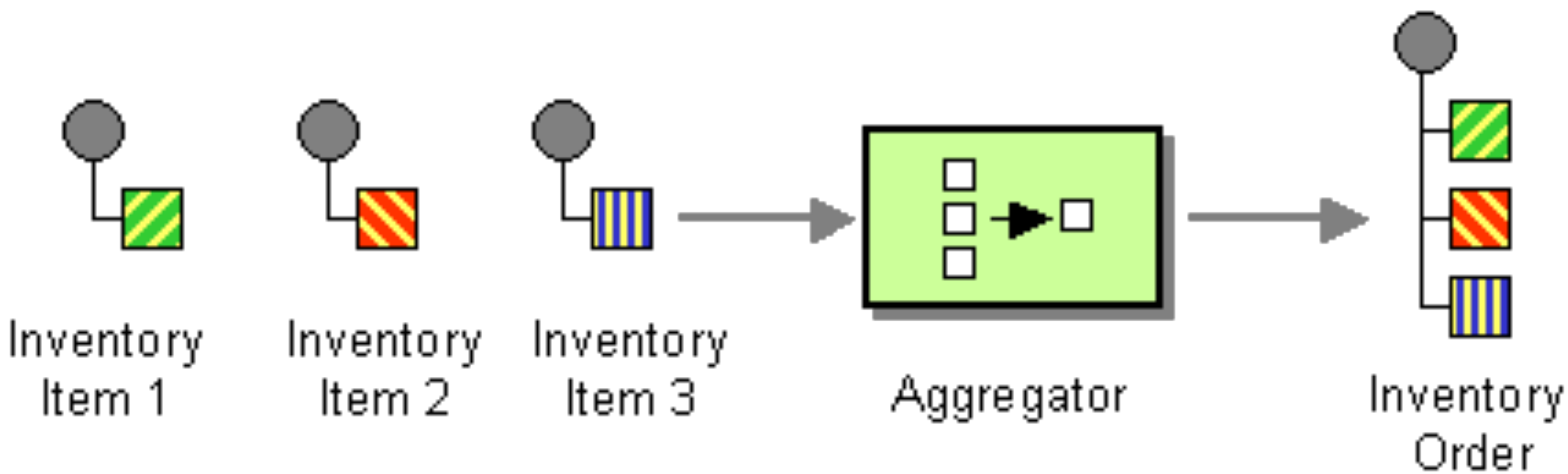
Remove un-needed messages

Splitter



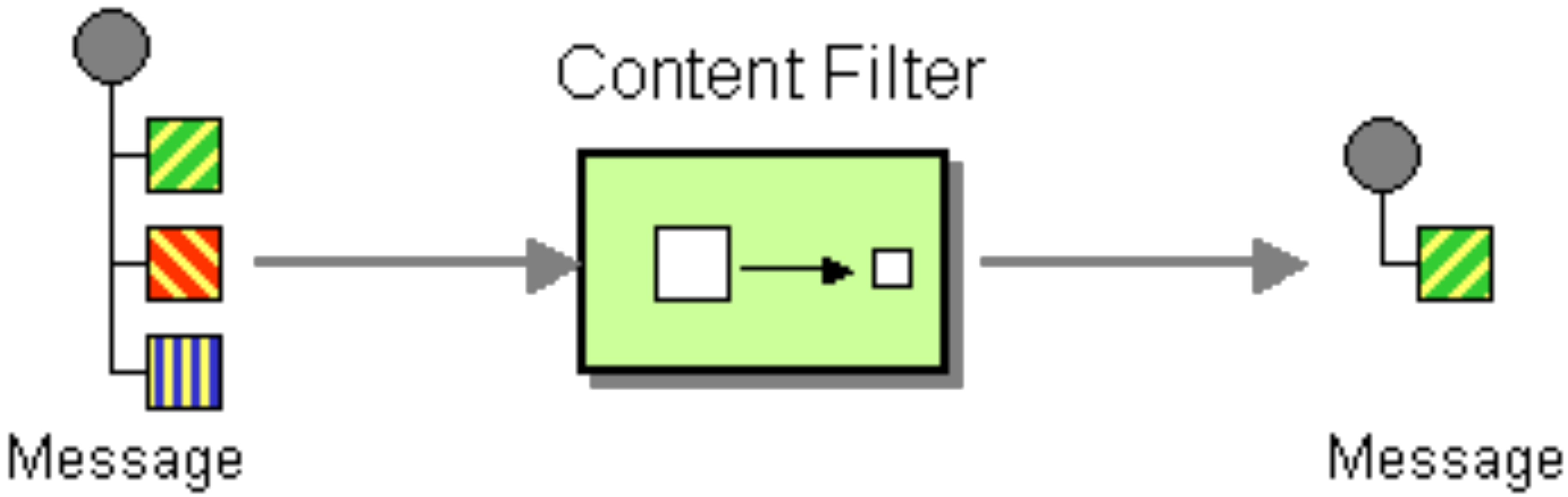
Decompose a composite message in parts

Aggregator



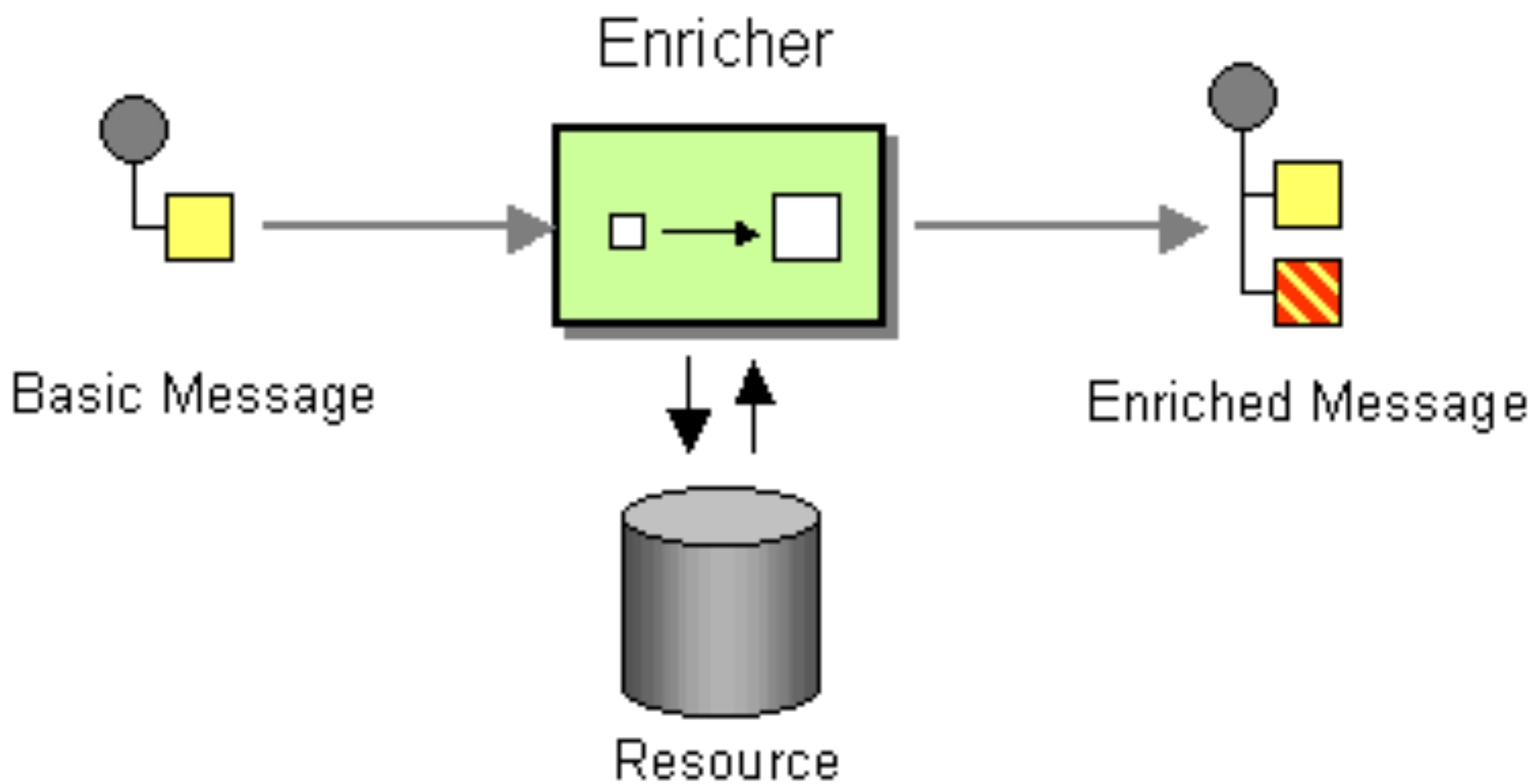
Use the parts to create a composite message

Content Filter



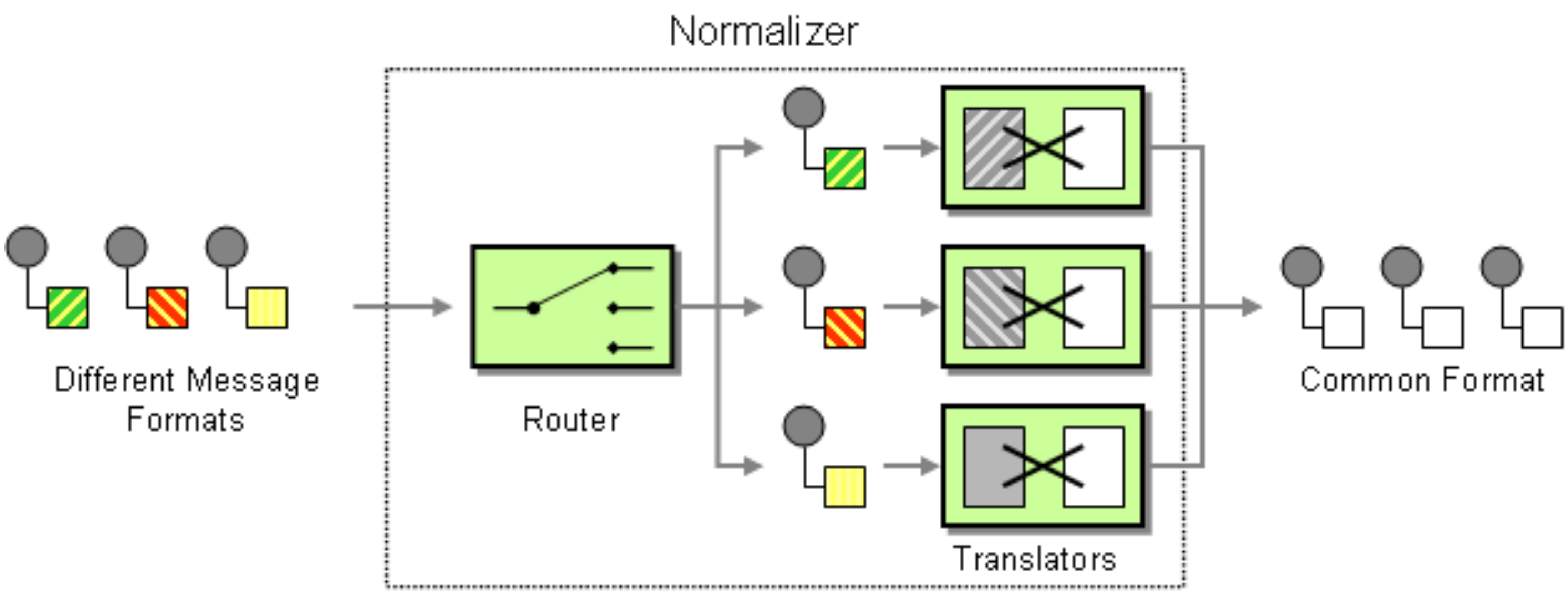
Filter from a composite message unneeded payload

Content Enricher



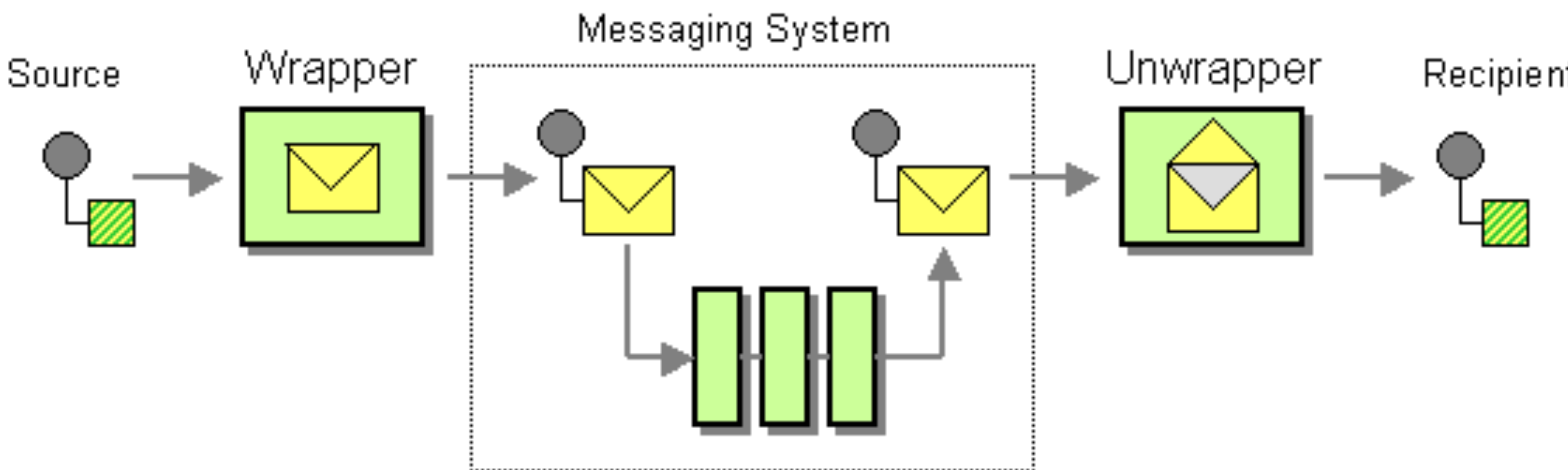
Use additional data to augment messages

Normalizer



Route messages to translators which transform the to a common format

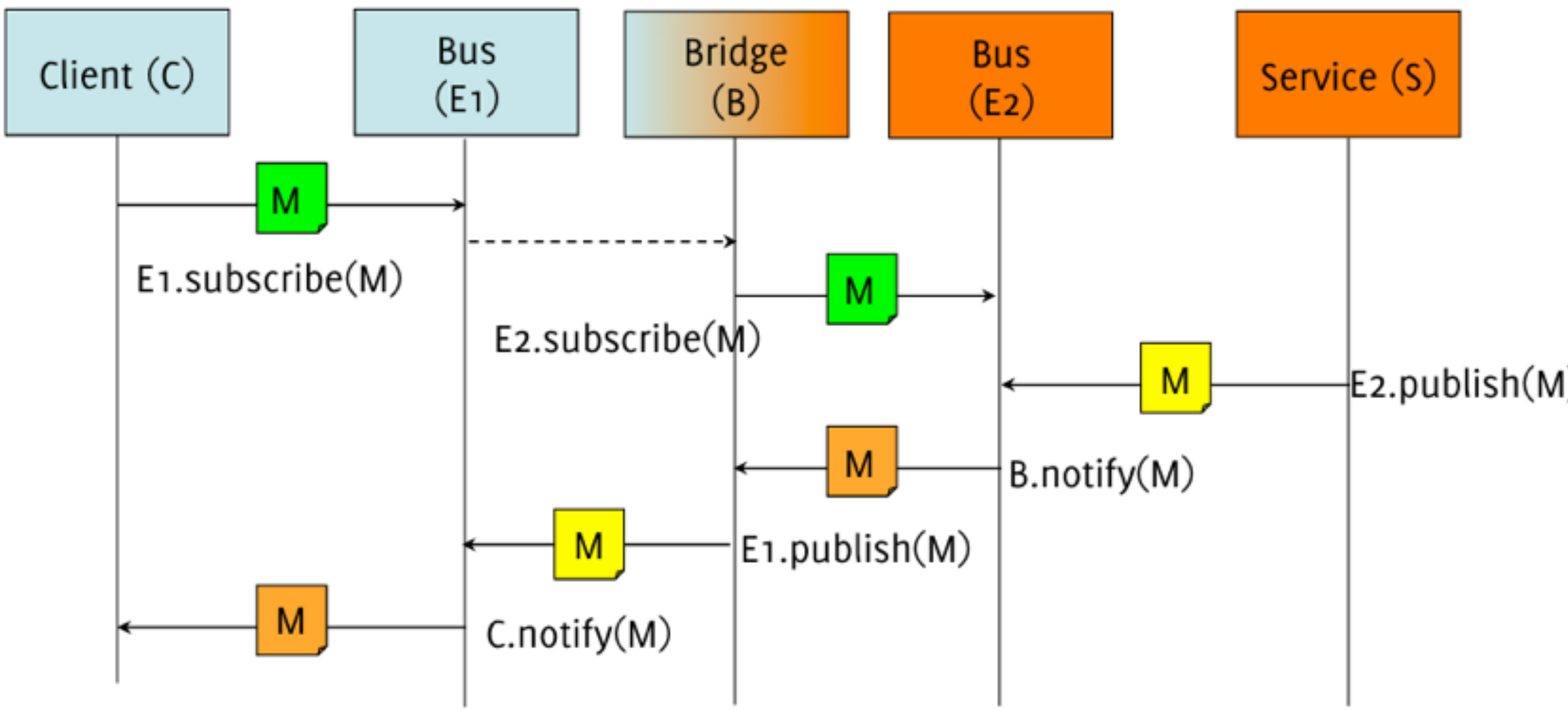
Enveloper Wrapper



Bridged delivery via wrapping messages into other messages

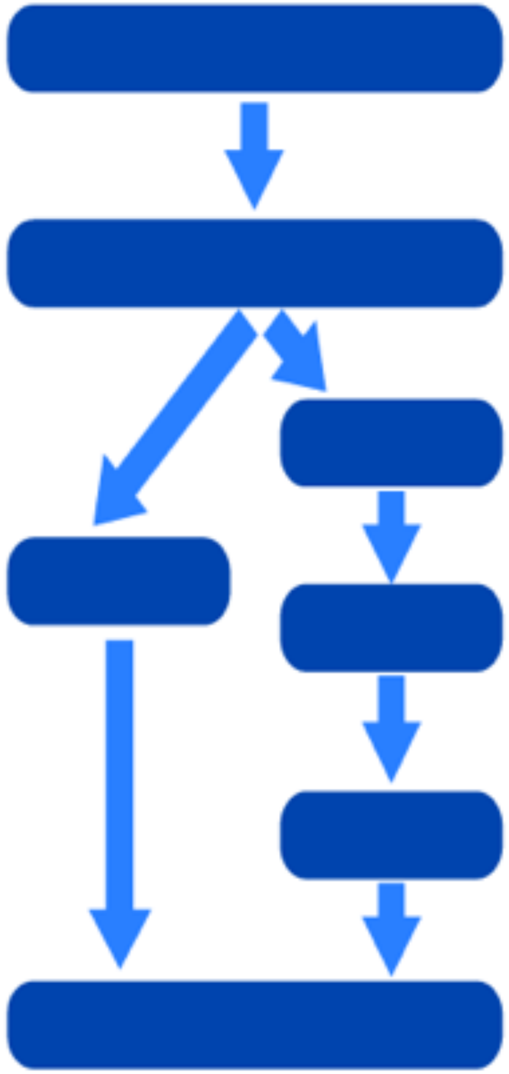
Messaging Bridge

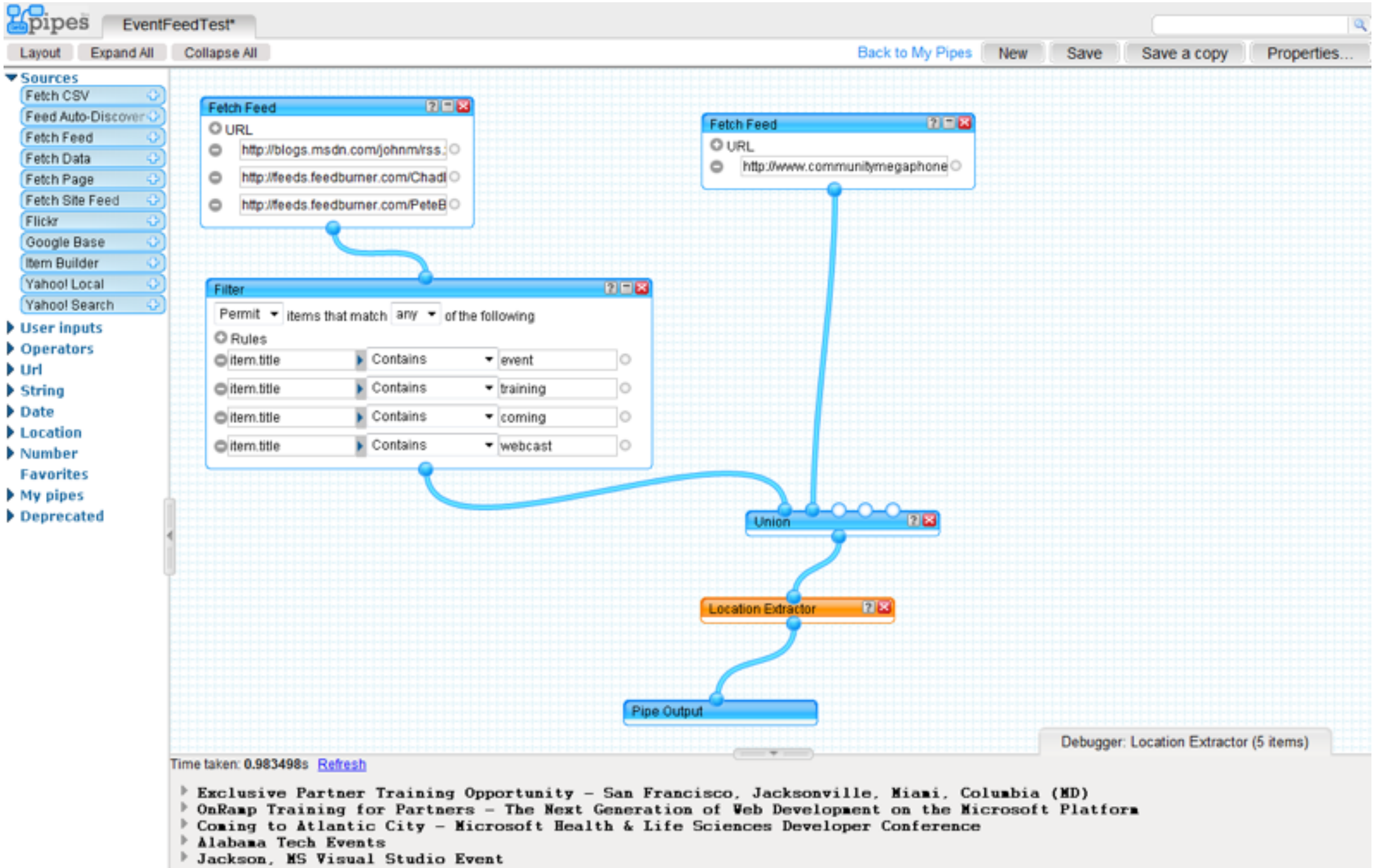
link multiple messaging systems to make messages exchanged on one also available on the others



Pipe & Filter

- Clean separation: filter process, pipe transport
- Heterogeneity and distribution
- Only batch processing, serializable data
- Composability, Reuse




pipes
EventFeedTest*
Layout
Expand All
Collapse All
Back to My Pipes
New
Save
Save a copy
Properties...
Sources
Fetch CSV
Feed Auto-Discover
Fetch Feed
Fetch Data
Fetch Page
Fetch Site Feed
Flickr
Google Base
Item Builder
Yahoo! Local
Yahoo! Search
User inputs
Operators
Url
String
Date
Location
Number
Favorites
My pipes
Deprecated
Fetch Feed
URL
http://blogs.msdn.com/johnm/rss.
http://feeds.feedburner.com/Chadl
http://feeds.feedburner.com/PeteB
Fetch Feed
URL
http://www.communitymegaphone
Filter
Permit items that match any of the following
Rules
item.title Contains event
item.title Contains training
item.title Contains coming
item.title Contains webcast
Union
Location Extractor
Pipe Output
Debugger: Location Extractor (5 items)
Time taken: 0.983498s Refresh
Exclusive Partner Training Opportunity - San Francisco, Jacksonville, Miami, Columbia (MD)
OnRamp Training for Partners - The Next Generation of Web Development on the Microsoft Platform
Coming to Atlantic City - Microsoft Health & Life Sciences Developer Conference
Alabama Tech Events
Jackson, MS Visual Studio Event

Stream

- Send
- Receive

Streaming

- Infinite sequence of messages

  simple/primitive, complex

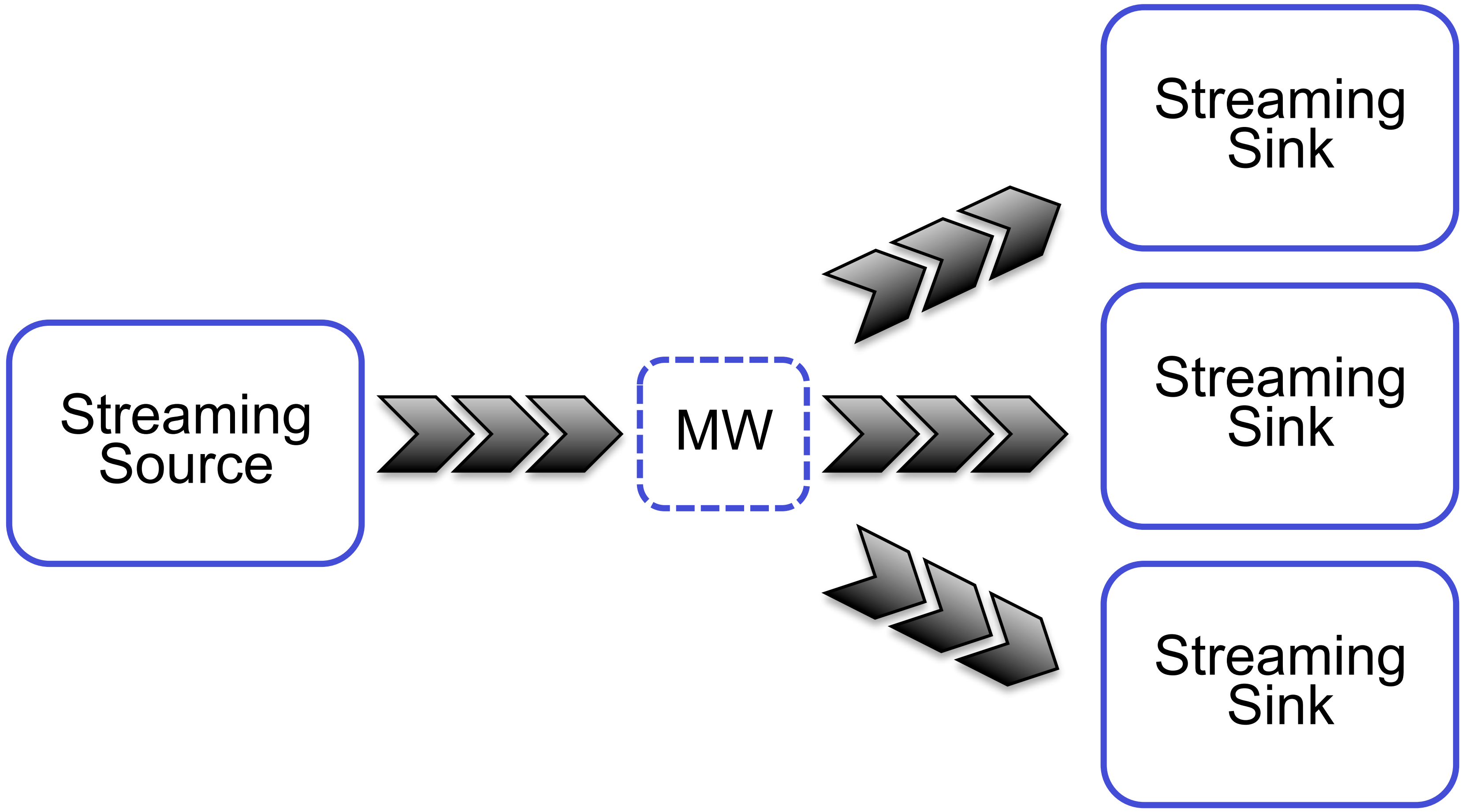
- Discrete - Messages

  stock markets, twitter

- Continuous - Data

  video, audio

Streaming and Data Analytics



Unicast or multicast communication channels

No discrete unit of interaction (request/response)

Low overhead, but mostly transport/communication

Sync/Async Streams

- Synchronous
  Time matters (e.g., minimum transfer rate)

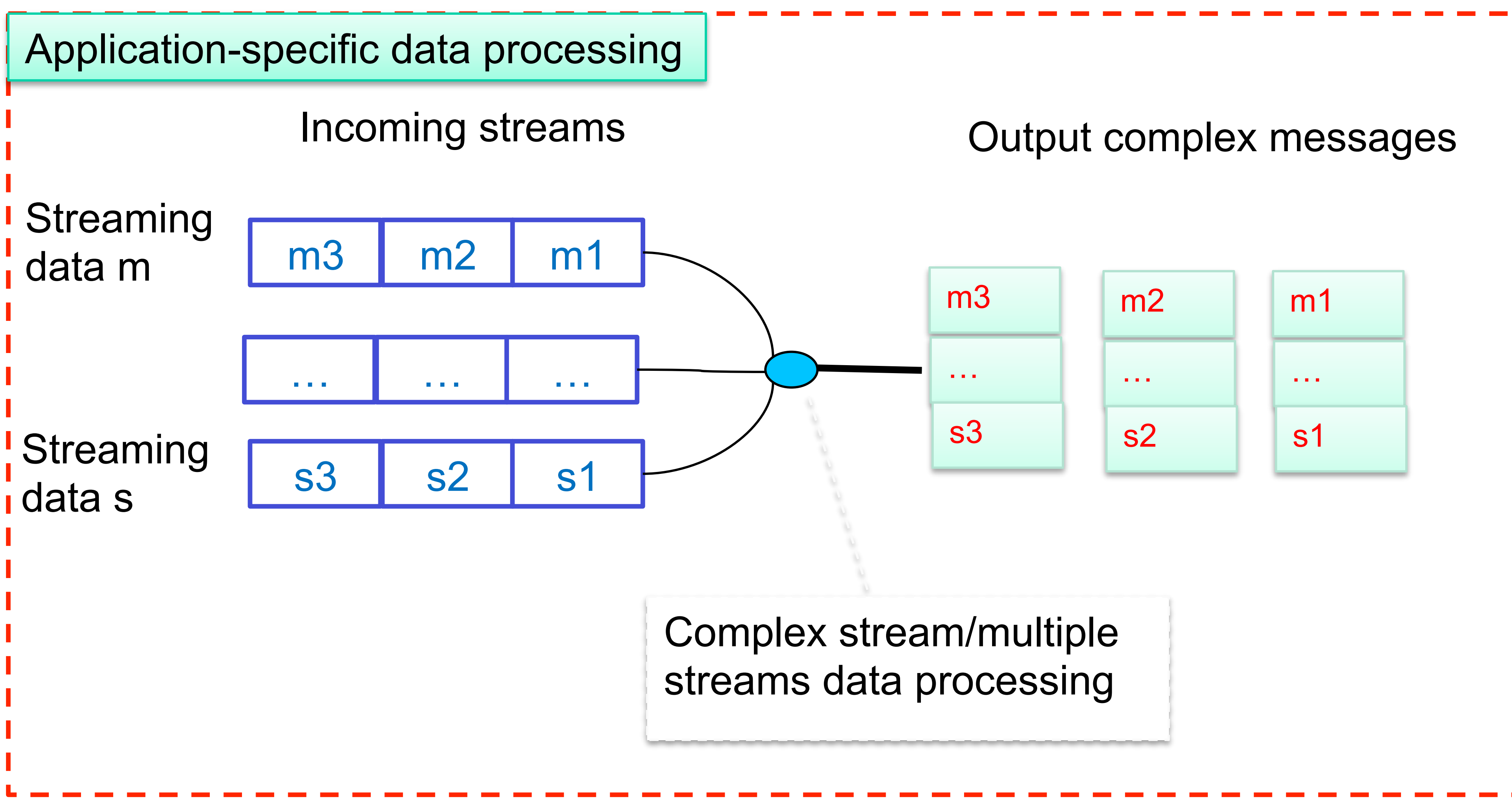
- Asynchronous
  Sequence matters (e.g., no specific transfer rate)

Sync/Async Streams

- Synchronous
  Time matters (e.g., minimum transfer rate)
- Asynchronous
  Sequence matters (e.g., no specific transfer rate)
- Isochronous
  Time is essence (e.g., both min and max transfer rate)

Processing Model

Complex Event Processor



EsperTech

S4 distributed stream computing platform

Processing Model

Complex Event Processor

- Event representation
  POJO, Maps, Object-Arrays, XML, etc..
- Continuous processing
  events processes as they arrive and sent to output
- Listeners and notifications
  both incoming and outgoing events
- Domain specific languages (DSL)
  describe conditions, transformations, etc.

EPL

Event Processing Language

Specify interests on certain types of events

event-patterns, correlations of events, and more

High-level language SQL-like

standard and new clauses

Streams replace tables; events replace rows

it’s just an analogy

Statements target single and multiple data streams

https://docs.oracle.com/cd/E13157_01/wlevs/docs30/epl_guide/overview.html

EPL

Event Processing Language

- Standard clauses

  SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY

- Re-casted clauses

  INSERT INTO

- New clauses

  RETAIN, MATCHING, OUTPUT

https://docs.oracle.com/cd/E13157_01/wlevs/docs30/epl_guide/overview.html

EPL

Event Processing Language

- Retain
  virtual window (constraint amount of data)

- Matching
  sequence of events (logical and temporal operators)

- Output
  control/stabilize the output rate

Event Processing

Incoming events processed through sliding windows

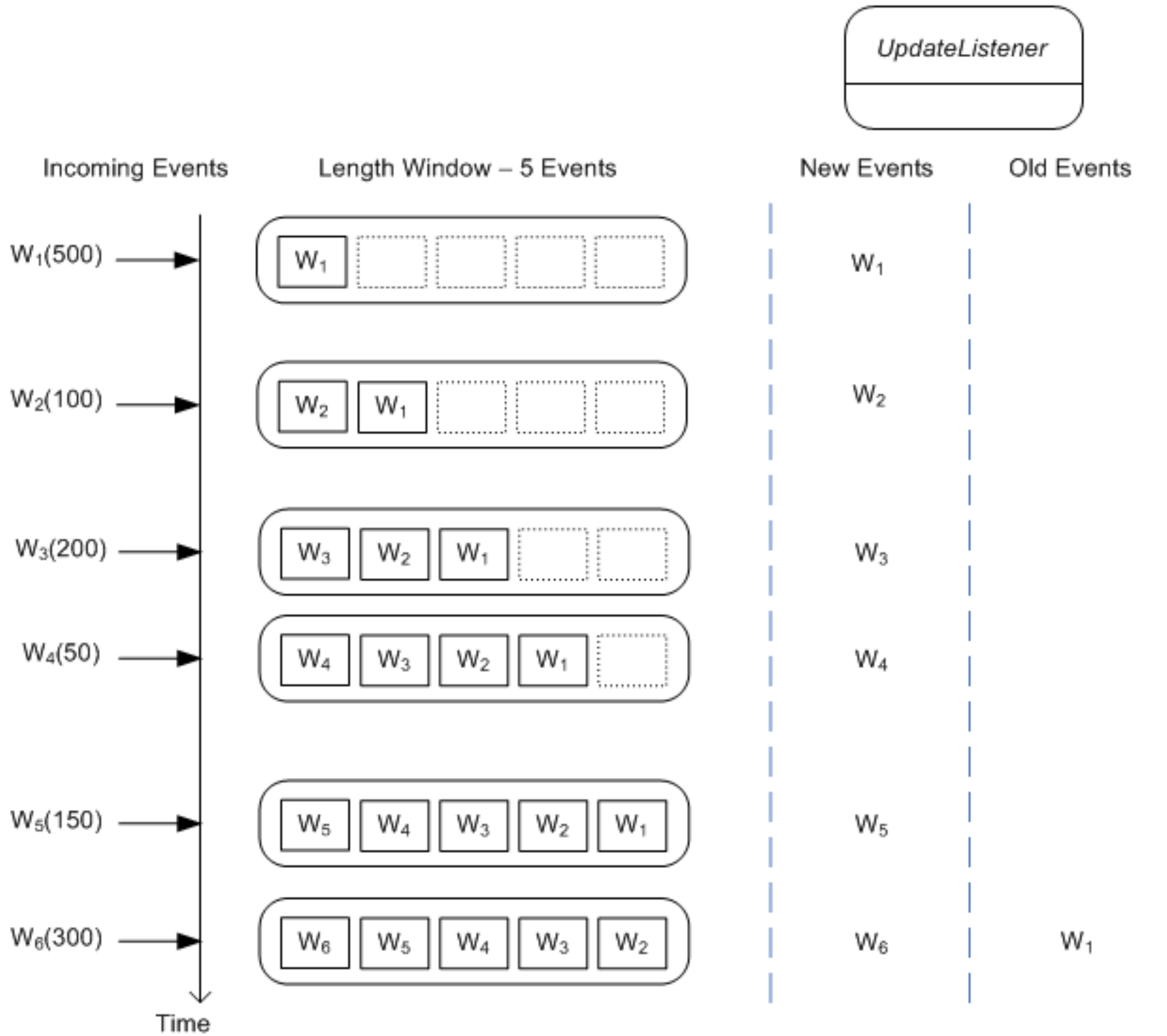
- *incremental, one event*
- *batched, chunks of events*

Size of window limits the maximum number of events or the maximum amount of time to keep them

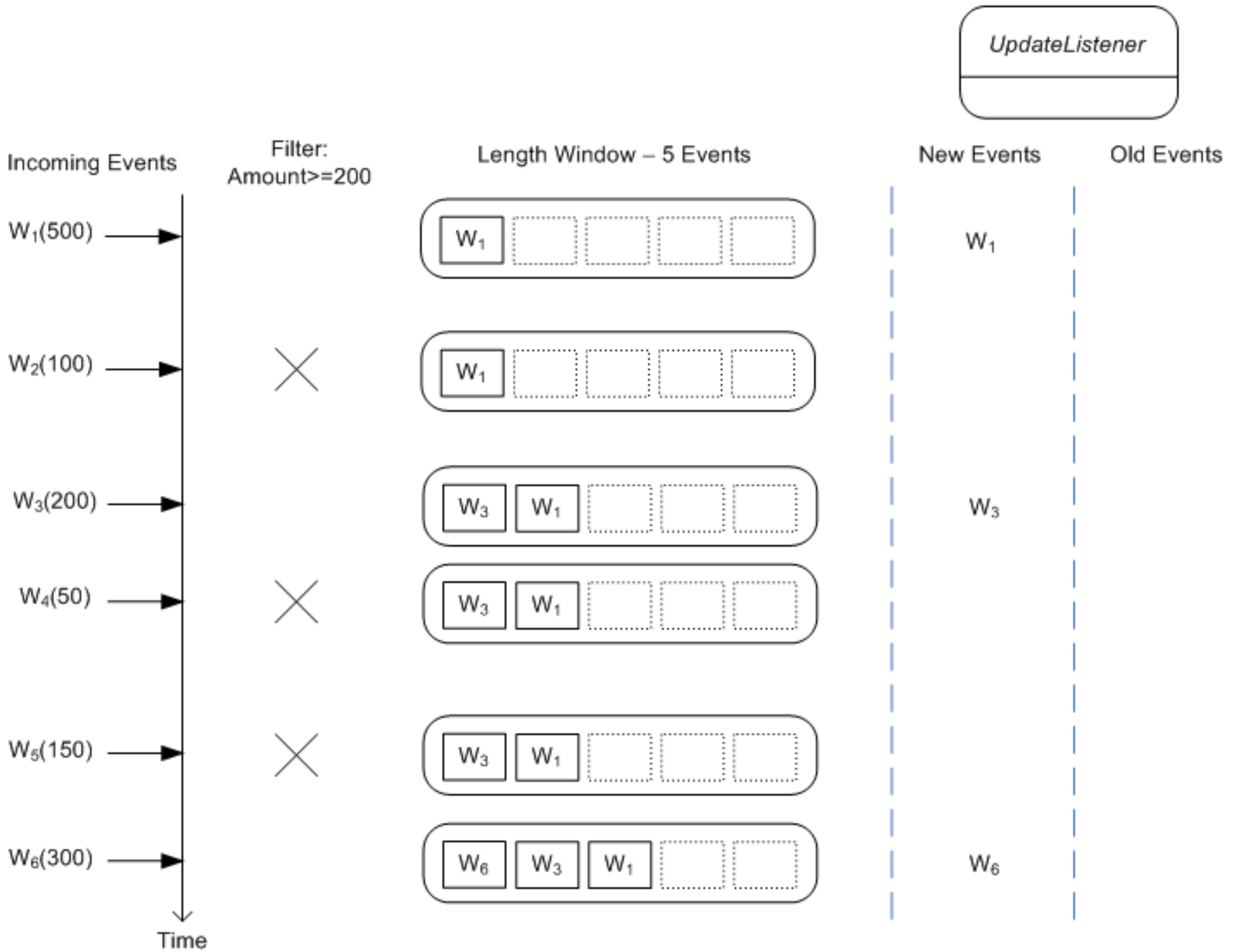
- *time*
- *length*

Conditions expressed on the window and events

Sliding Window with Length

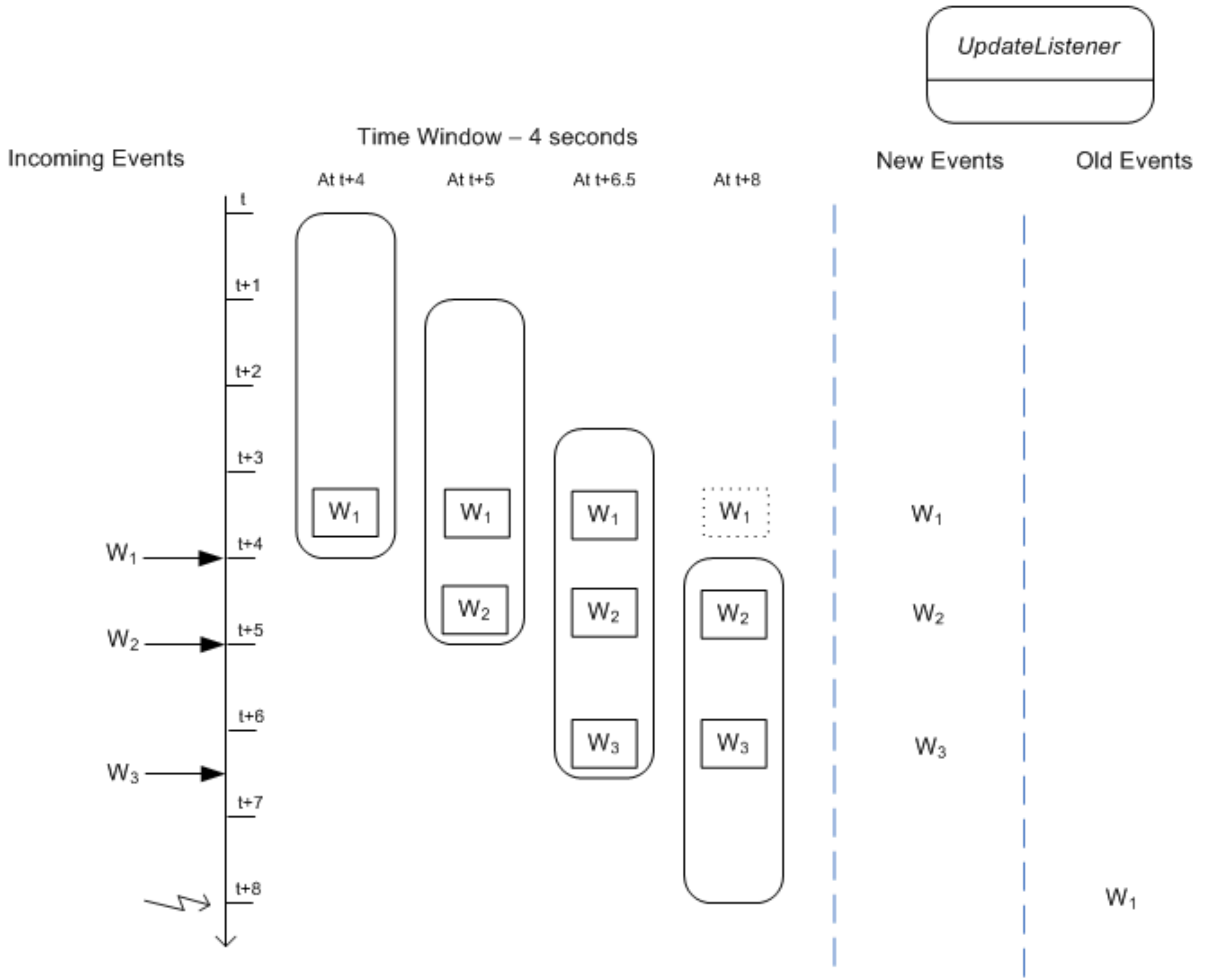


Filter & Slide

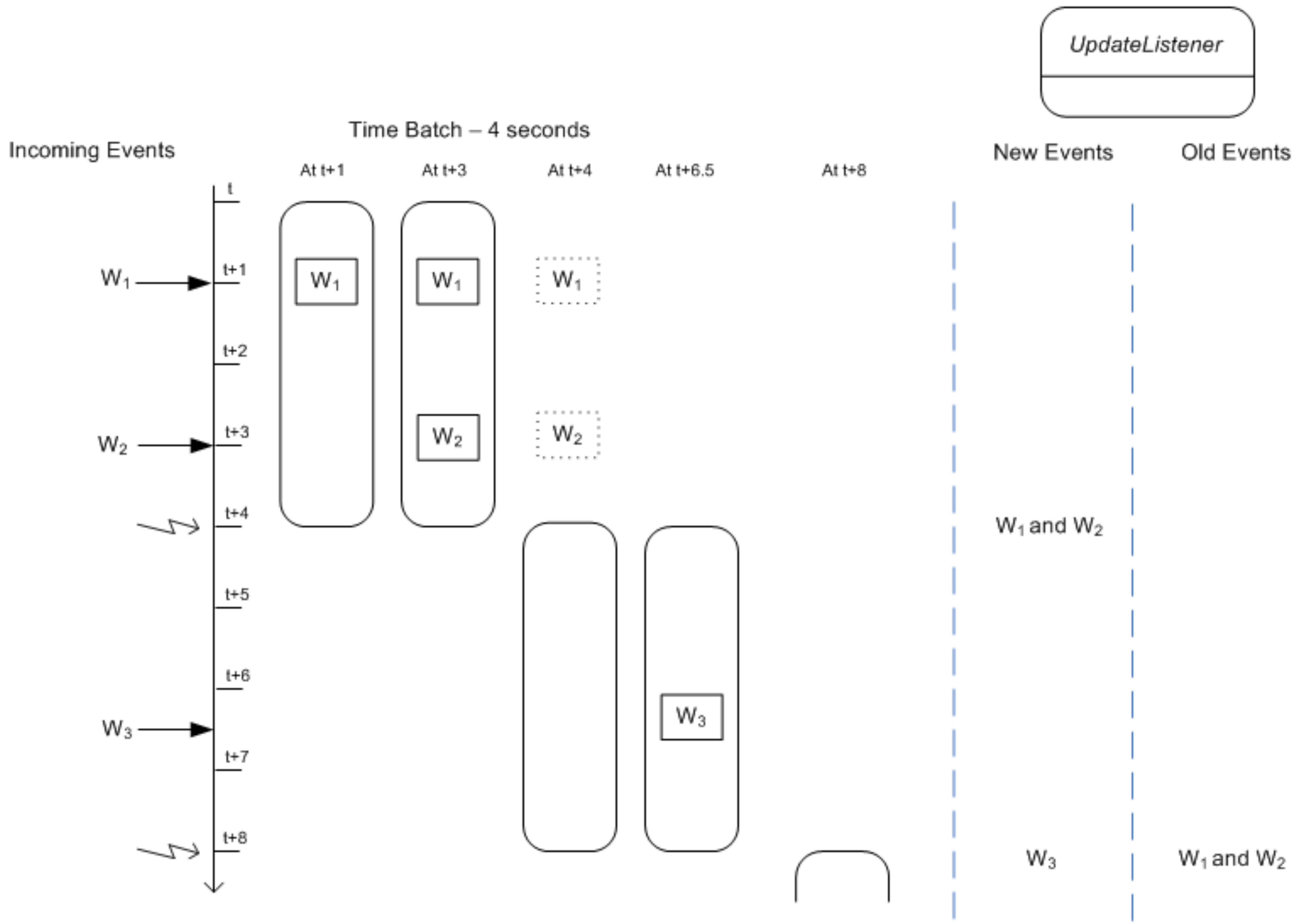


Slide & Filter

Sliding Window with Time

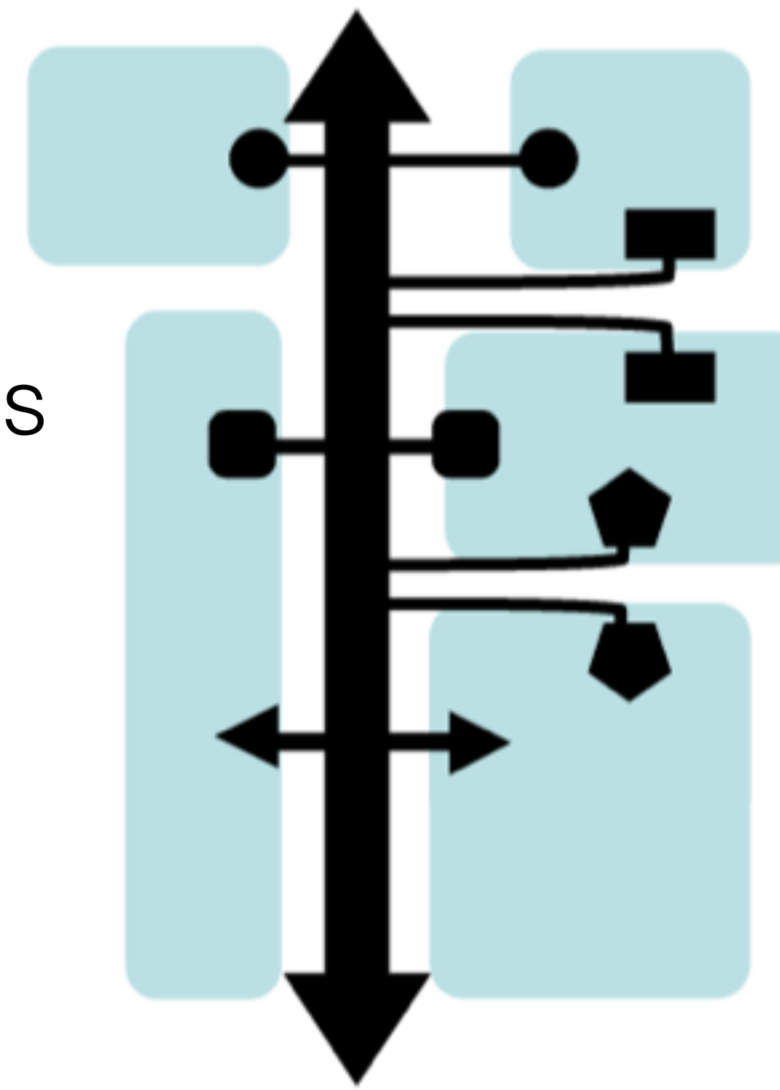


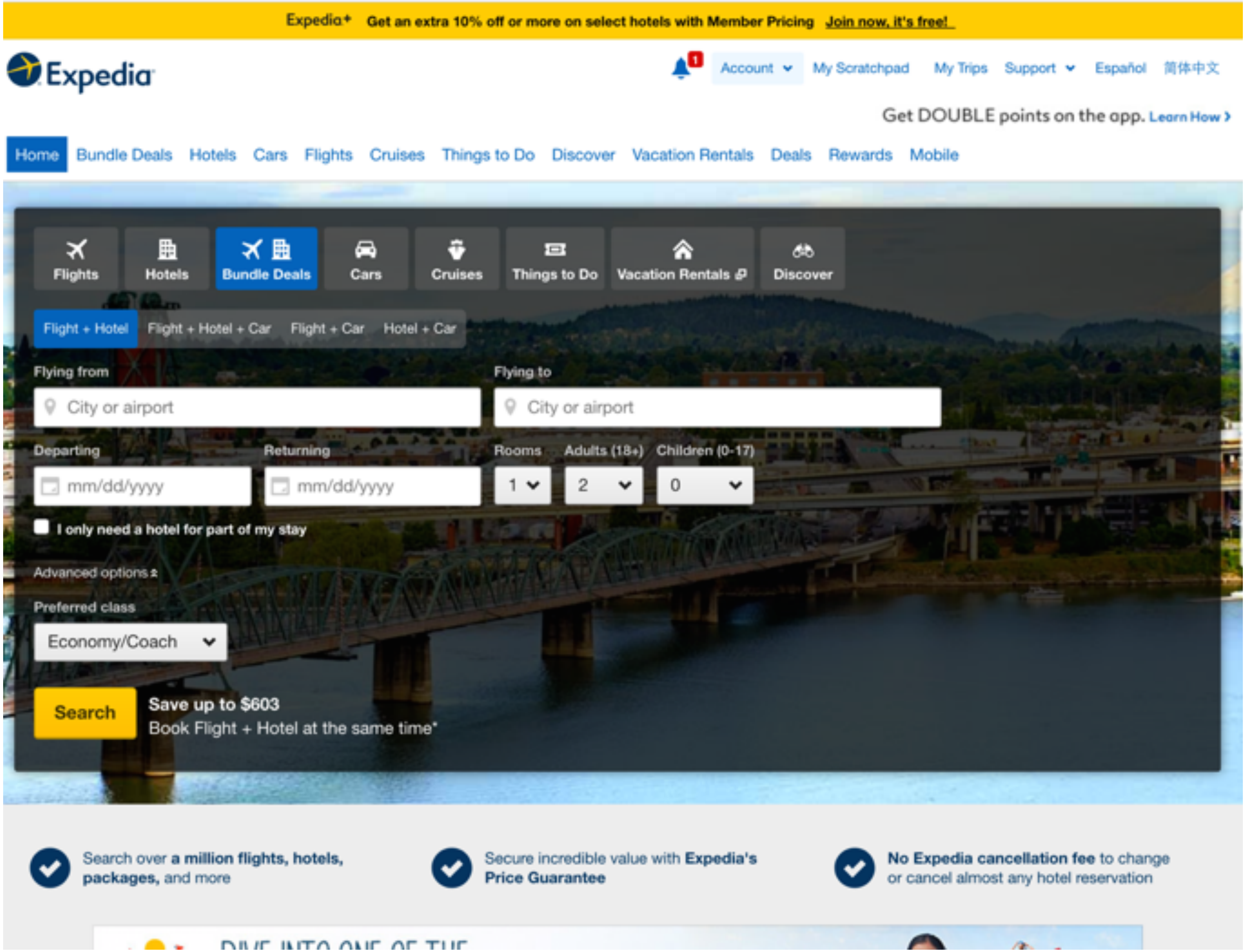
Batched Window with Time

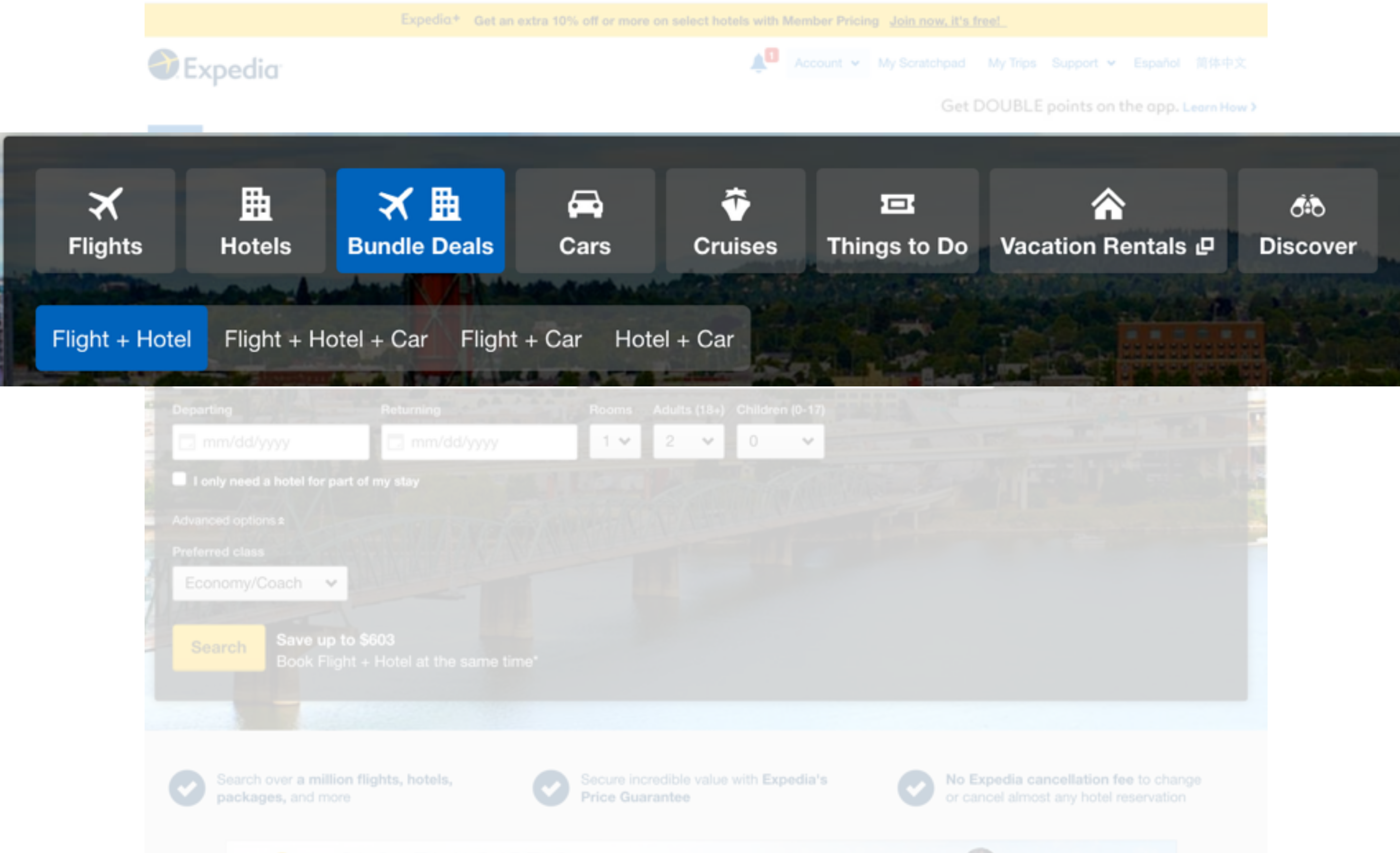


Service Oriented

- Components outside control
- Standard connectors, precise interfaces
- Interface compatibility problem
- Loose coupling, reuse



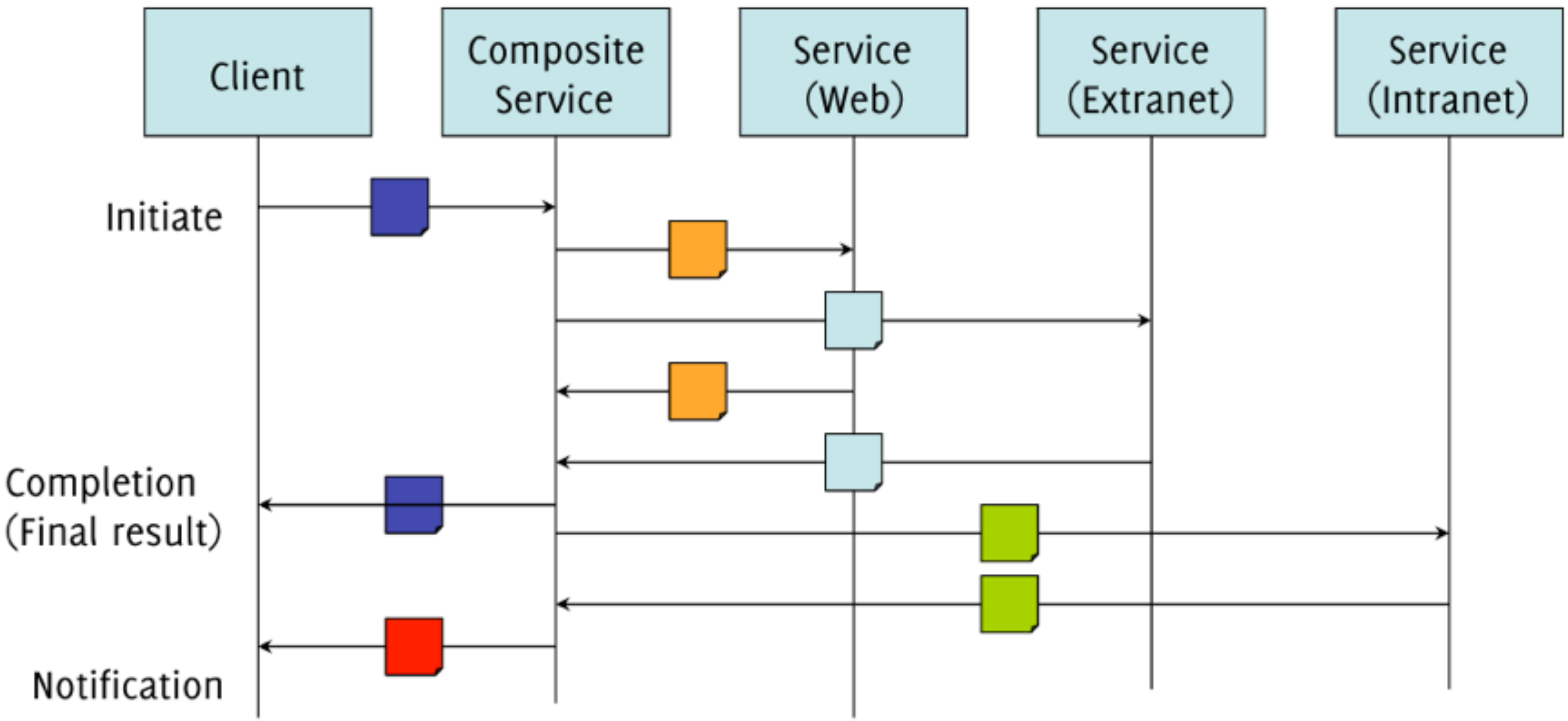

Expedia+ Get an extra 10% off or more on select hotels with Member Pricing Join now, it's free!
Expedia
Account
My Scratchpad
My Trips
Support
Español
简体中文
Get DOUBLE points on the app. Learn How
Home
Bundle Deals
Hotels
Cars
Flights
Cruises
Things to Do
Discover
Vacation Rentals
Deals
Rewards
Mobile
Flights
Hotels
Bundle Deals
Cars
Cruises
Things to Do
Vacation Rentals
Discover
Flight + Hotel
Flight + Hotel + Car
Flight + Car
Hotel + Car
Flying from
City or airport
Flying to
City or airport
Departing
mm/dd/yyyy
Returning
mm/dd/yyyy
Rooms
1
Adults (18+)
2
Children (0-17)
0
I only need a hotel for part of my stay
Advanced options
Preferred class
Economy/Coach
Search
Save up to $603
Book Flight + Hotel at the same time*
Search over a million flights, hotels, packages, and more
Secure incredible value with Expedia's Price Guarantee
No Expedia cancellation fee to change or cancel almost any hotel reservation


Expedia+ Get an extra 10% off or more on select hotels with Member Pricing Join now, it's free!
Expedia
Account
My Scratchpad
My Trips
Support
Español
简体中文
Get DOUBLE points on the app. Learn How
Flights
Hotels
Bundle Deals
Cars
Cruises
Things to Do
Vacation Rentals
Discover
Flight + Hotel
Flight + Hotel + Car
Flight + Car
Hotel + Car
Departing
Returning
Rooms
Adults (18+)
Children (0-17)
mm/dd/yyyy
mm/dd/yyyy
I only need a hotel for part of my stay
Advanced options
Preferred class
Economy/Coach
Search
Save up to $603
Book Flight + Hotel at the same time*
Search over a million flights, hotels, packages, and more
Secure incredible value with Expedia's Price Guarantee
No Expedia cancellation fee to change or cancel almost any hotel reservation

Components	Services
Tight coupling	Loose coupling
Client requires library	Message exchanges
Client / Server	Peer-to-peer
Extendable	Composable
Fast	Some overhead
Small/Medium	Medium/Large
Buy and install	Pay-per-use
Local	Remote

Composition/Orchestration

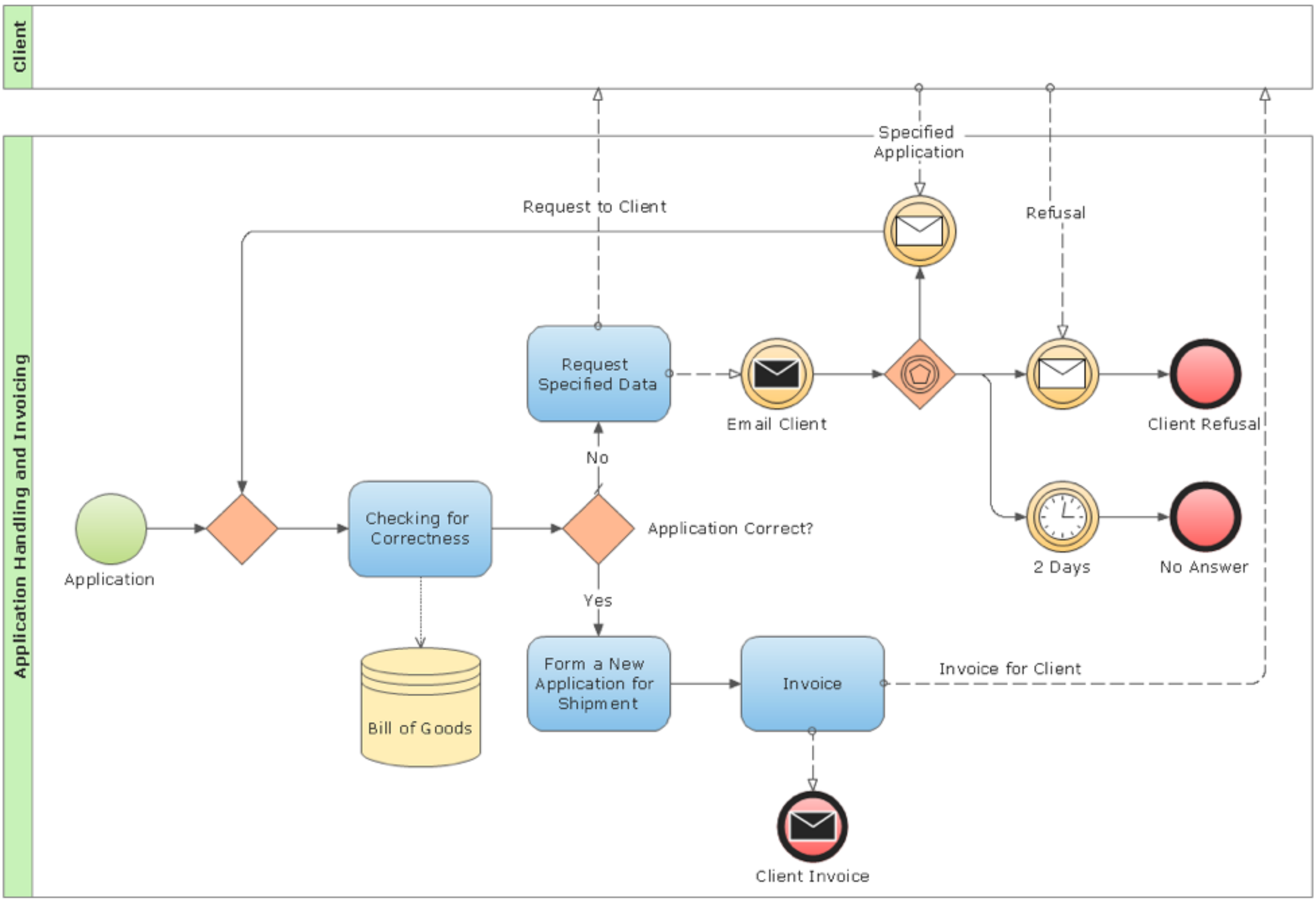
build systems out of the composition of existing ones



Business Processes

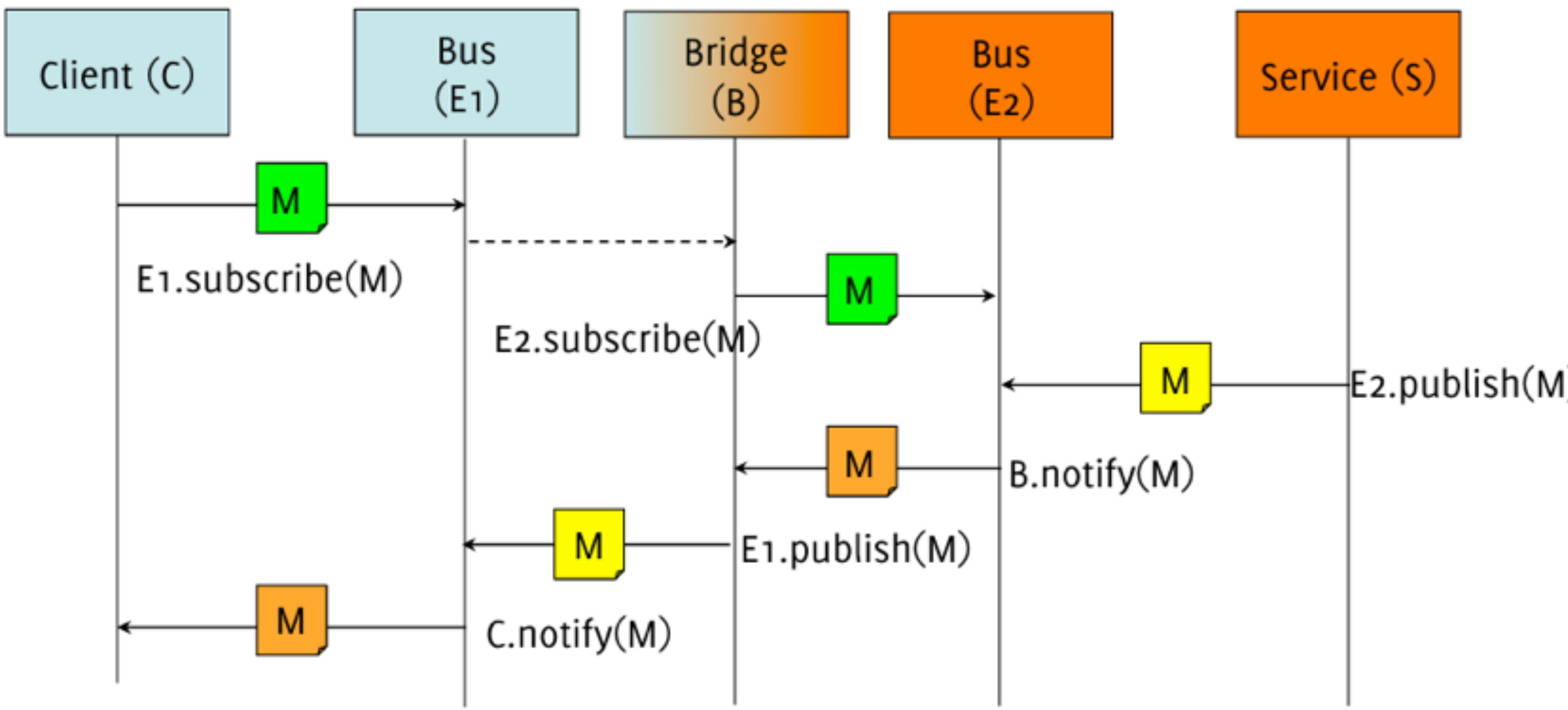
- Many alternative notations and languages
  WSCI, BPML, BPEL4WS, BPSS, XPDL
- Standard protocols and technologies
  WSDL, XML, HTTP, JSON, SMTP, FTP, …
- Two “BIG” players
  SOAP + WS-*, HTTP+RESTFul

Application Handling and Invoicing Process

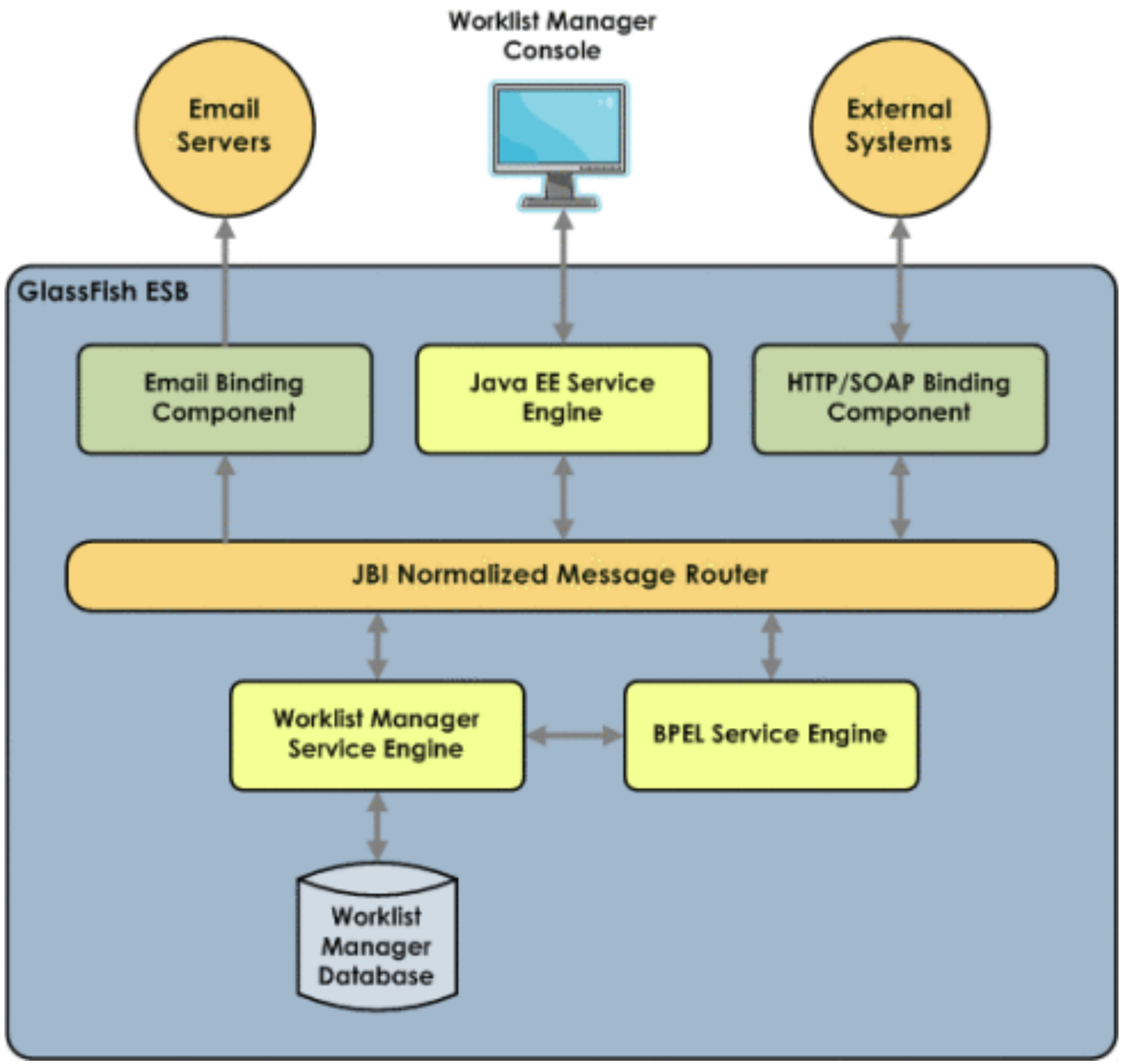


Messaging Bridge

link multiple messaging systems to make messages exchanged on one also available on the others



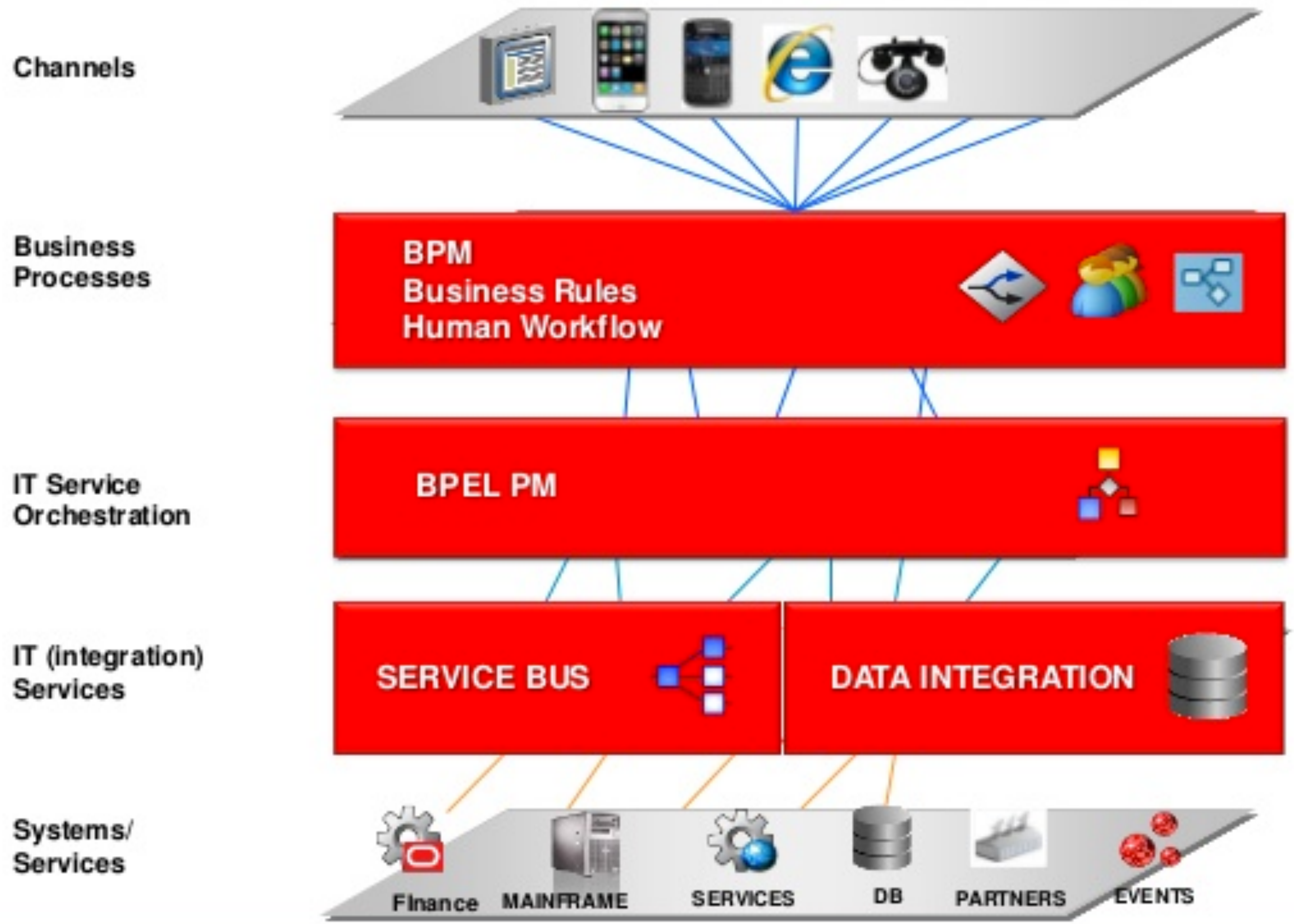
Software Architecture



- Work Flow Engine
- Enterprise BUS

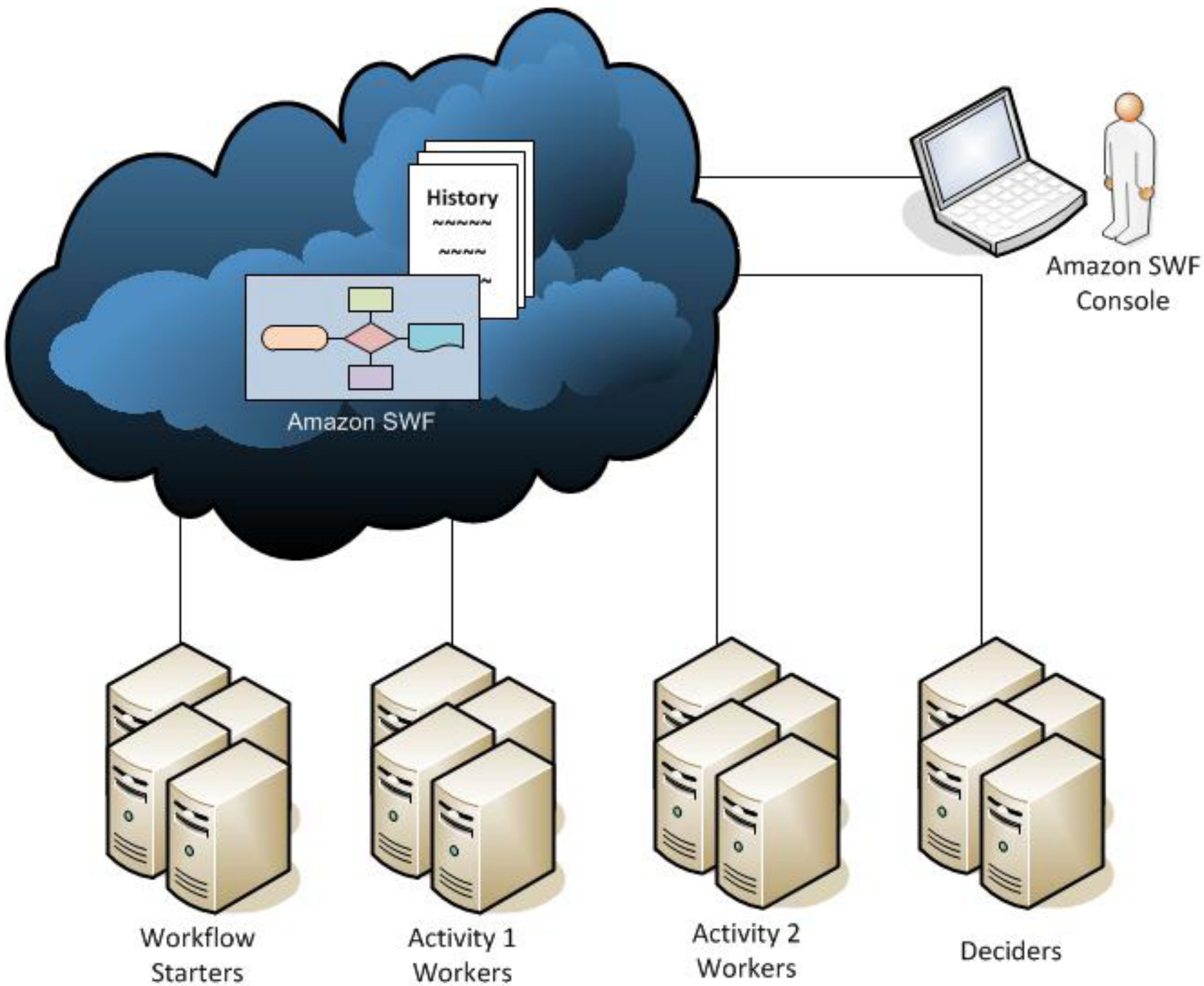
Layered Target Architecture

SOA / BPM architecture



https://www.slideshare.net/kumargaurav66/oracle-soaand-bpm

Say “what what” ? In the cloud!

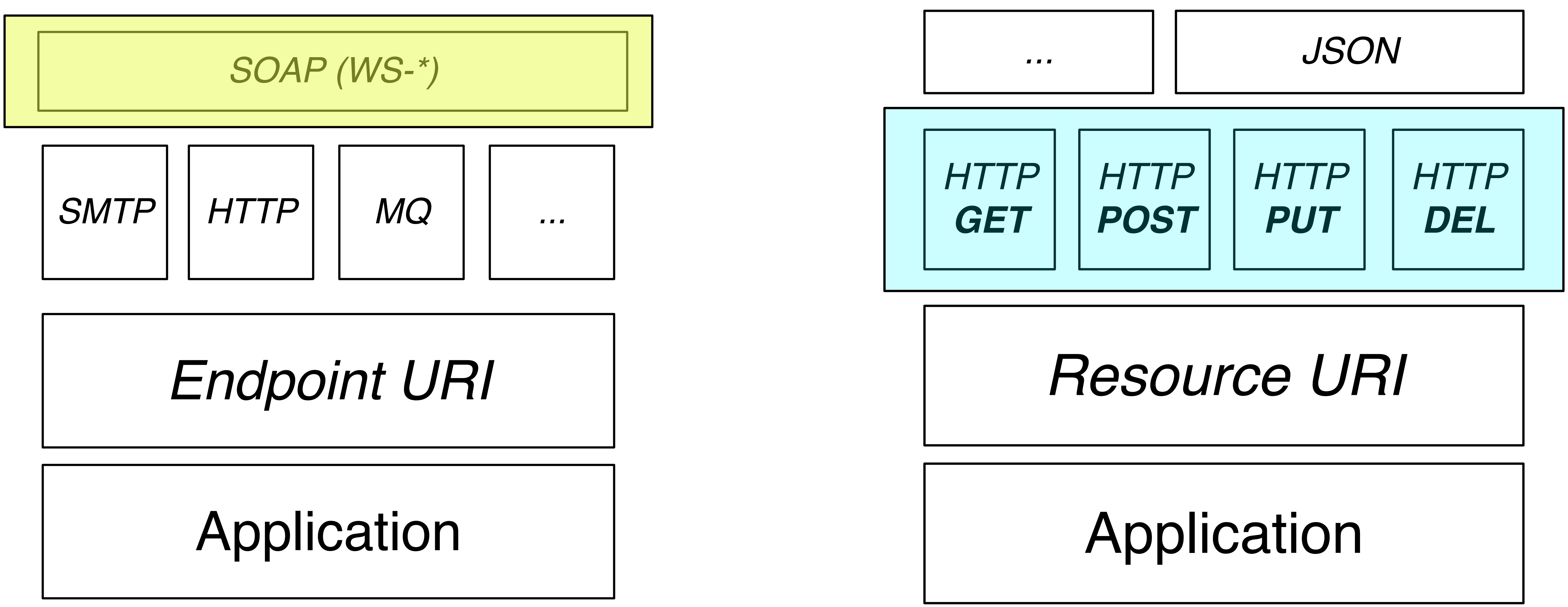


Heavy vs Light

Old vs New

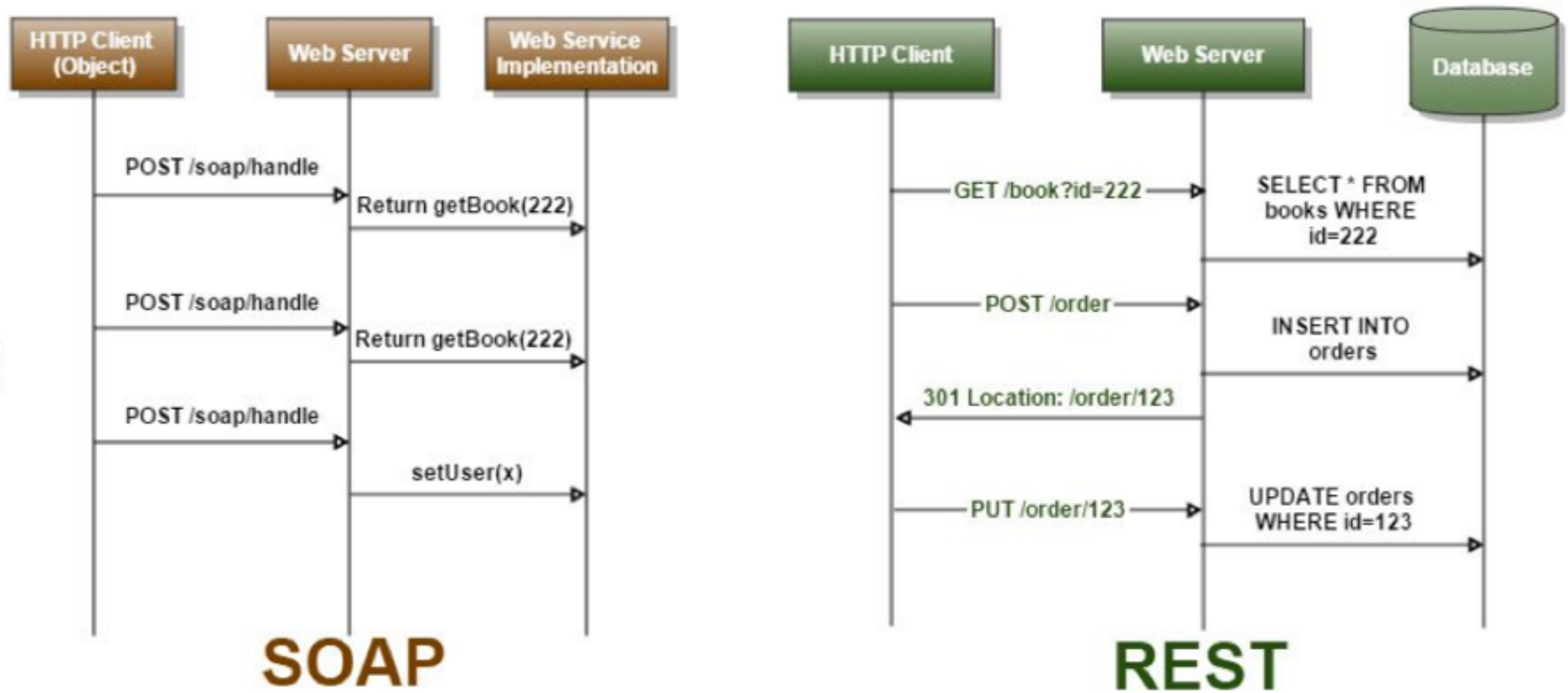
#	SOAP	REST
1	A XML-based message protocol	An architectural style protocol
2	Uses WSDL for communication between consumer and provider	Uses XML or JSON to send and receive data
3	Invokes services by calling RPC method	Simply calls services via URL path
4	Does not return human readable result	Result is readable which is just plain XML or JSON
5	Transfer is over HTTP. Also uses other protocols such as SMTP, FTP, etc.	Transfer is over HTTP only
6	JavaScript can call SOAP, but it is difficult to implement	Easy to call from JavaScript
7	Performance is not great compared to REST	Performance is much better compared to SOAP - less CPU intensive, leaner code etc.

Layered View



https://i.stack.imgur.com/WQsEJ.jpg

Process View



http://webtechsharing.com/soap-vs-rest/

Summary

- Understand the size and complexity of your system and distribute functions and data (lifecycle)
- Embrace diversity: not only RPC, not only sync
- Aim at satisfy your the requirements with the right method (different patterns/styles for different parts)

Additional Readings

- Sam Newman, Building Microservices, 2015. http://de.slideshare.net/spnewman/principles-of-microservices-ndc-2014
- Markus Völter et al.: Remoting Patterns – Foundation of Enterprise, Internet and Realtime Distributed Object Middleware, Wiley Series in Software Design Patterns, 2004
- Thomas Erl: Service-Oriented Architecture – Concepts, Technology and Design, Prentice Hall, 2005
- Roy Fielding's PhD thesis on REST: http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm
- Martin Fowler's blog: https://martinfowler.com/
- Fay Chang et al. Bigtable: a distributed storage system for structured data. (OSDI '06)
- Eric Redmond and Jim R. Wilson: Seven Databases in Seven Weeks – A Guide to Modern Databases and the NoSQL Movement
- CAP: http://www.julianbrowne.com/article/viewer/brewers-cap-theorem
- Eventual consistency: http://queue.acm.org/detail.cfm?id=1466448
- Java Message Service: http://www.oracle.com/technetwork/java/index-jsp-142945.html
- Integration patterns: https://camel.apache.org/enterprise-integration-patterns.html, http://www.espertech.com/esper/release-5.0.0/esper-reference/html_single/index.html