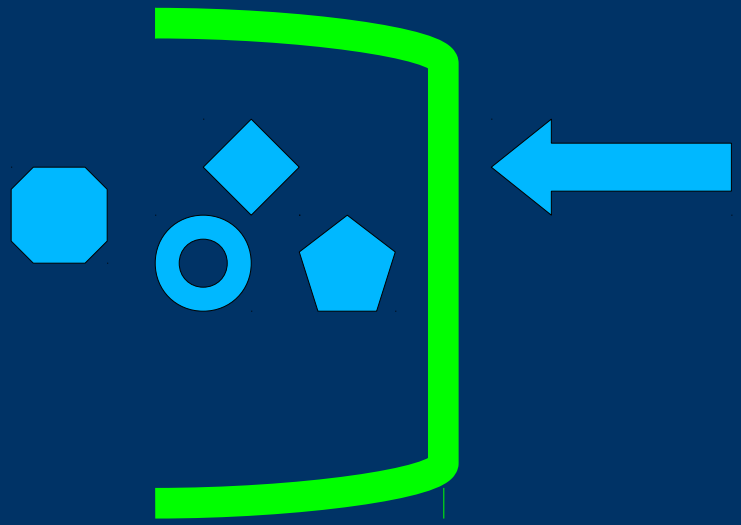
Architectural Patterns

CS 718 lecture series

Prof. Rushikesh Joshi
IIT Bombay

wrapper



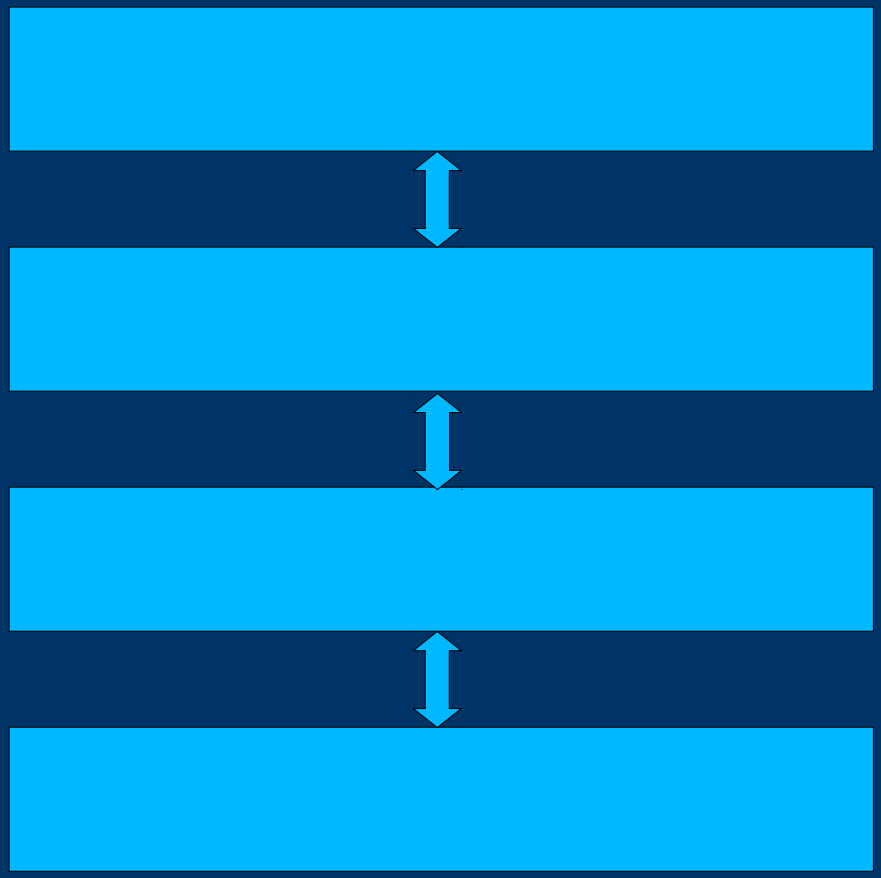
provide a single layer of abstraction on top of many related functions

wrapper – an example

- the semop system call in unix-- it wraps around many functions related to semaphores
  –

```
int semop(int semid, struct sembuf *sops, unsigned nsops);
```

Layering

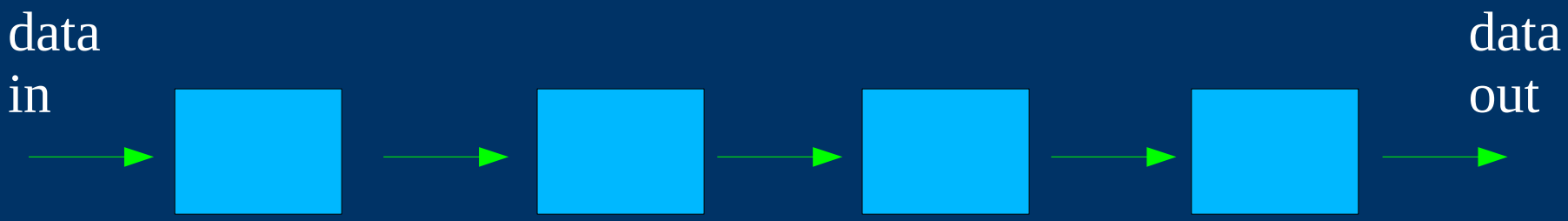


A large system is decomposed into layers of abstractions

Layers: examples

- operating systems – system calls, inner kernel, hardware abstractions, hardware
- networking layers
- APIs-platform independent Implementation-Platform dependent implementation
- 3-tiered architecture – UI, BL, Data

Pipelines



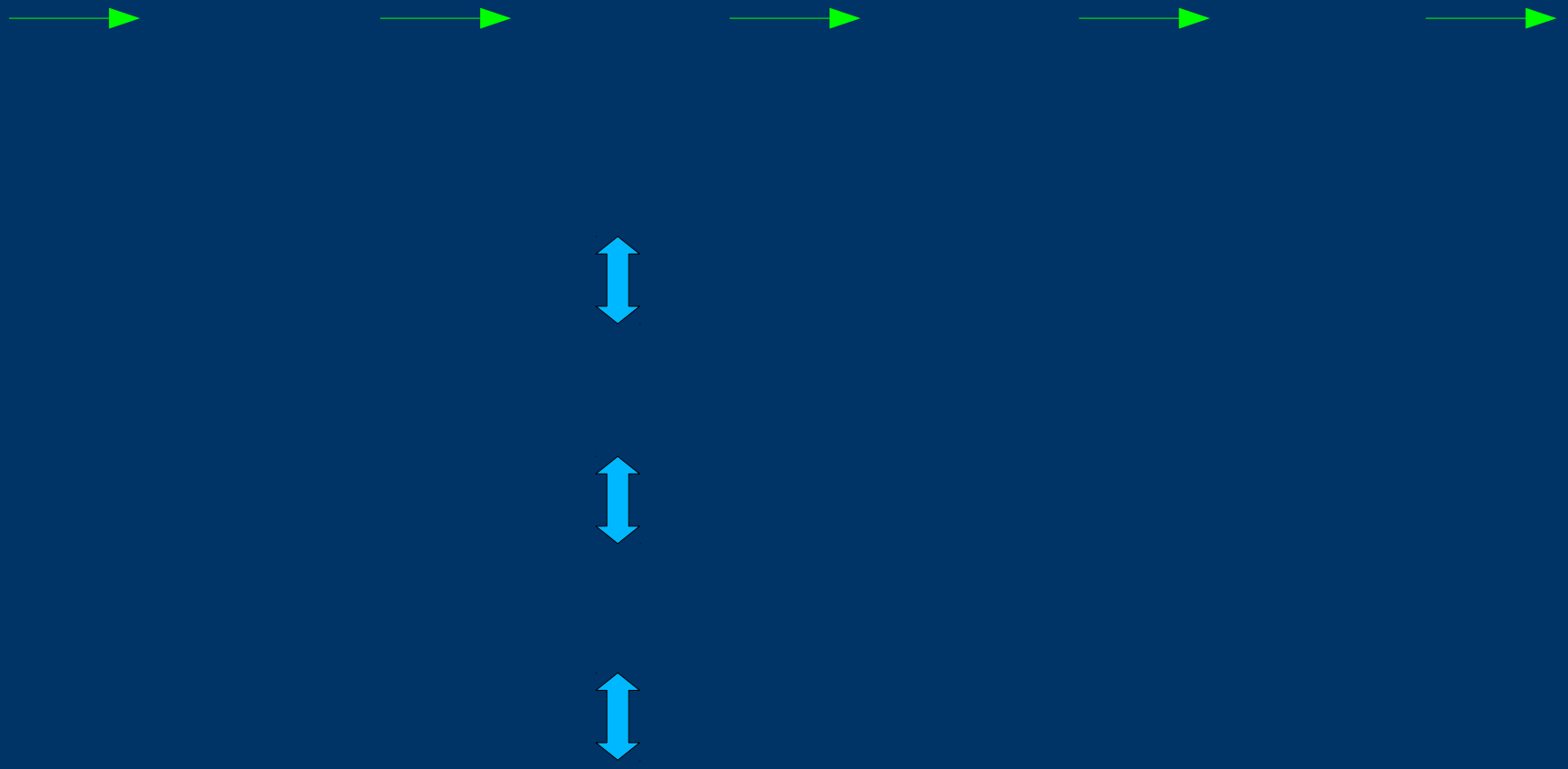
processing steps in a pipeline are sometimes referred to as filters, and the connectors as pipes

stream of data passes through pipes and filters

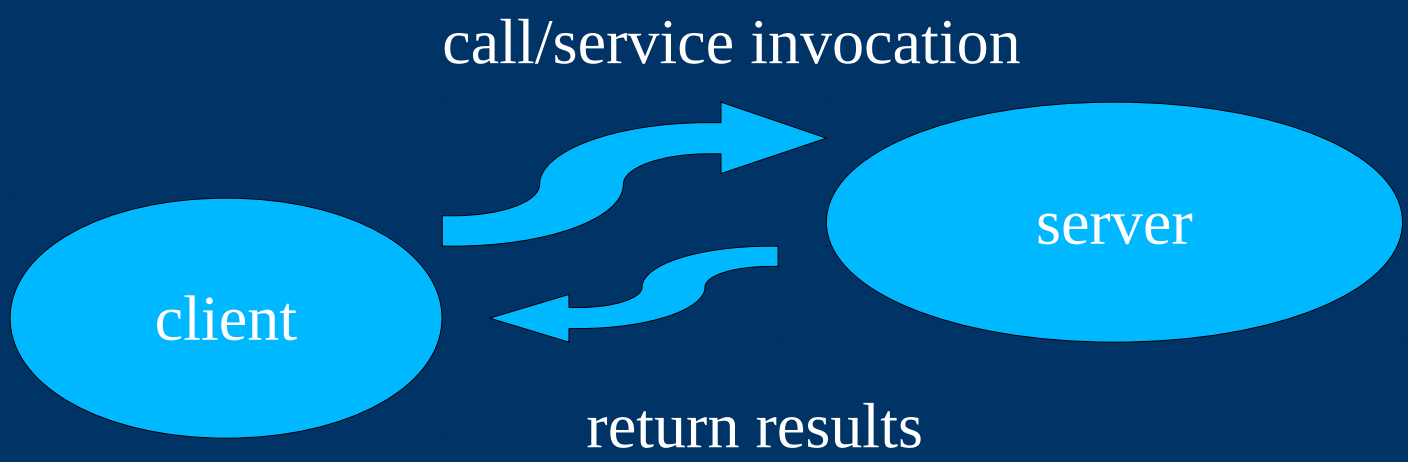
pipelines-- examples

- language processing – lexical analysis, syntax analysis, semantic analysis, code generation, optimization
- unix pipes and filters
- instruction pipelines

The difference between layers and pipes



client-server

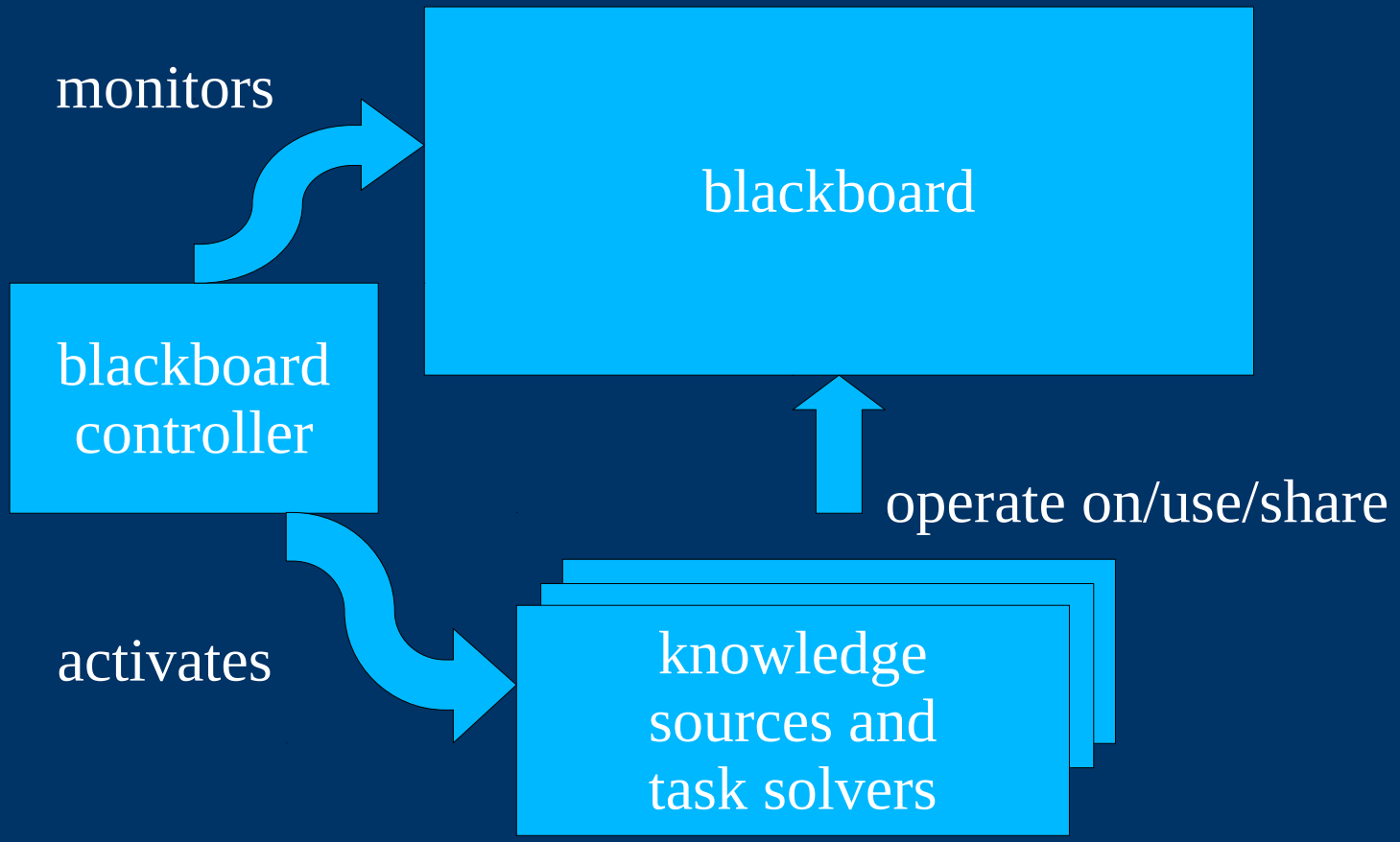


client does not provide an interface, the only communication from the server to client is through return results or exception/error values

client-server examples

- Lan services on a unix machine/windows machine
  - ldap, portmapper, nslookup, directories
- RMI, RPC, Web-servers and web clients

Blackboard

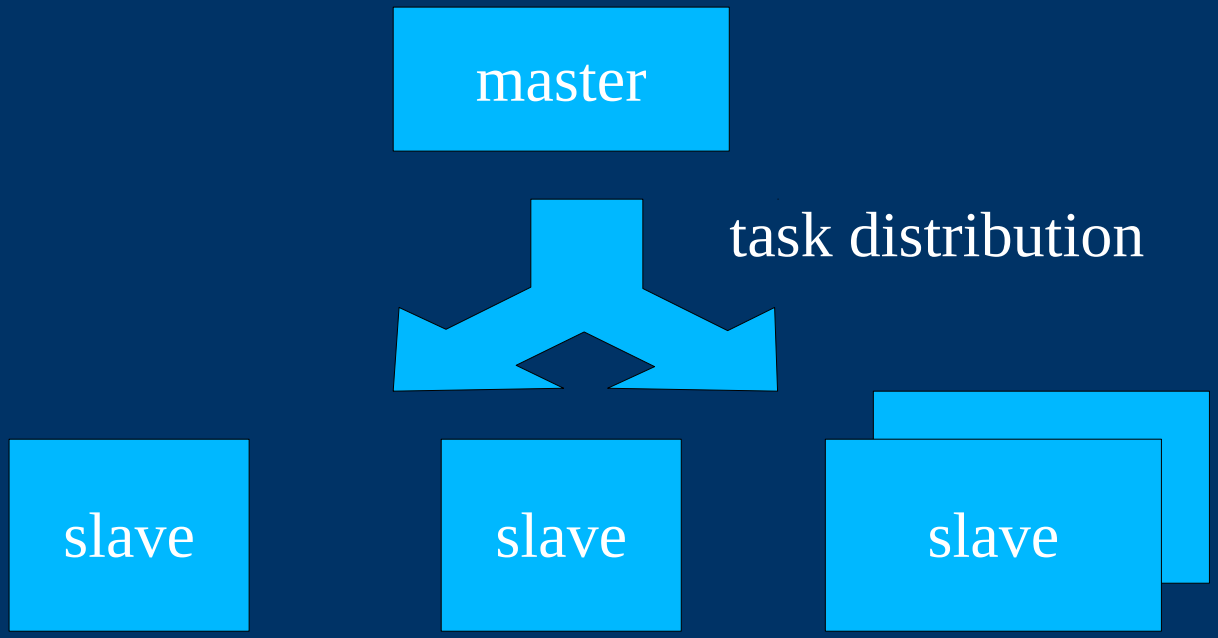


collaborative problem solving through knowledge sharing

blackboard-examples

- Linda tuple space
- shared memory based parallel/distributed problem solvers

Master-Slave



master-slave example

- used in parallel computing on clusters
  - master process splits a big task into subtasks, distributes and coordinates slave processes, collects and collates results

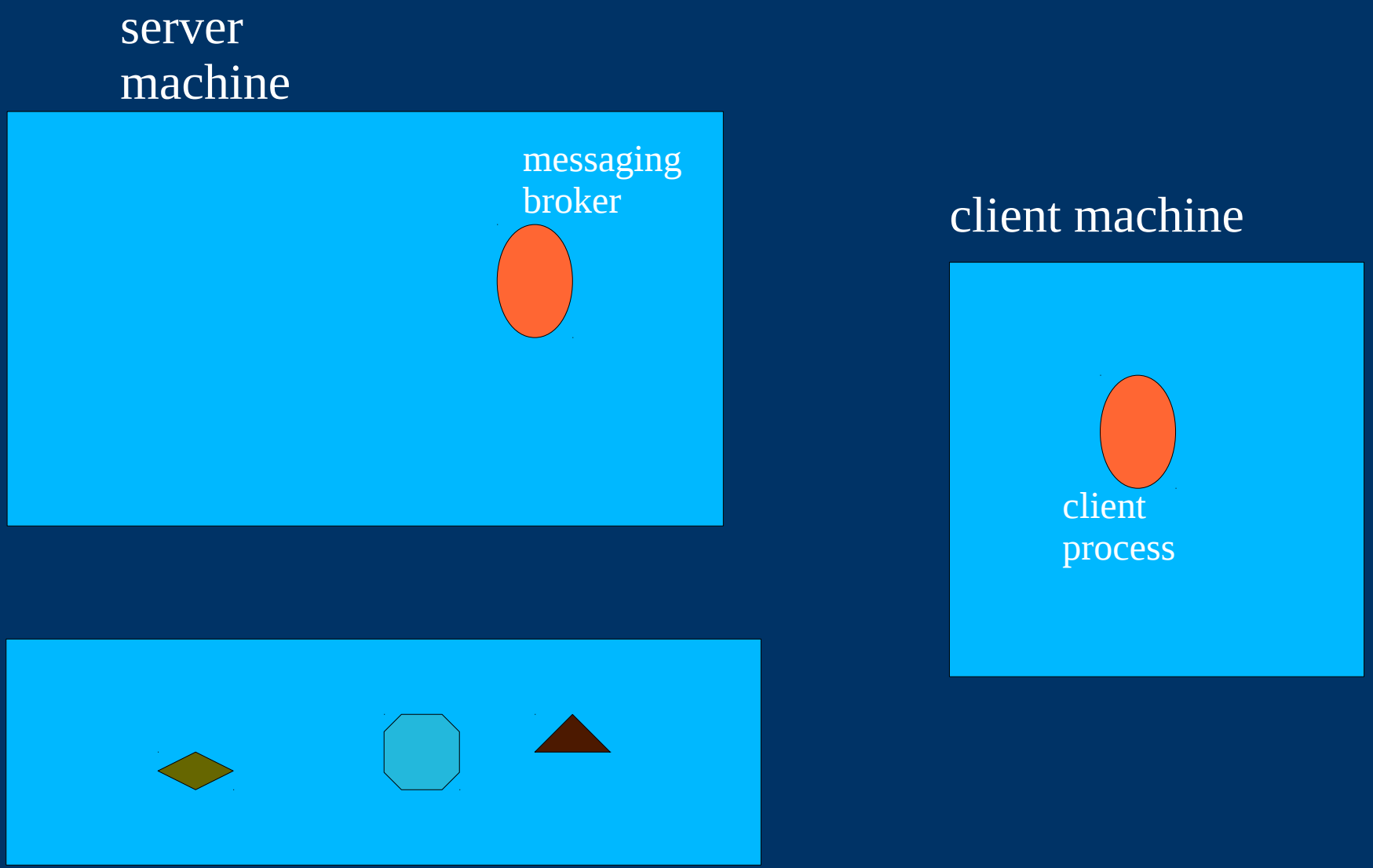
Broker

coordinating communication (requests, replies, exceptions)
in a distributed remote service invocation scenario

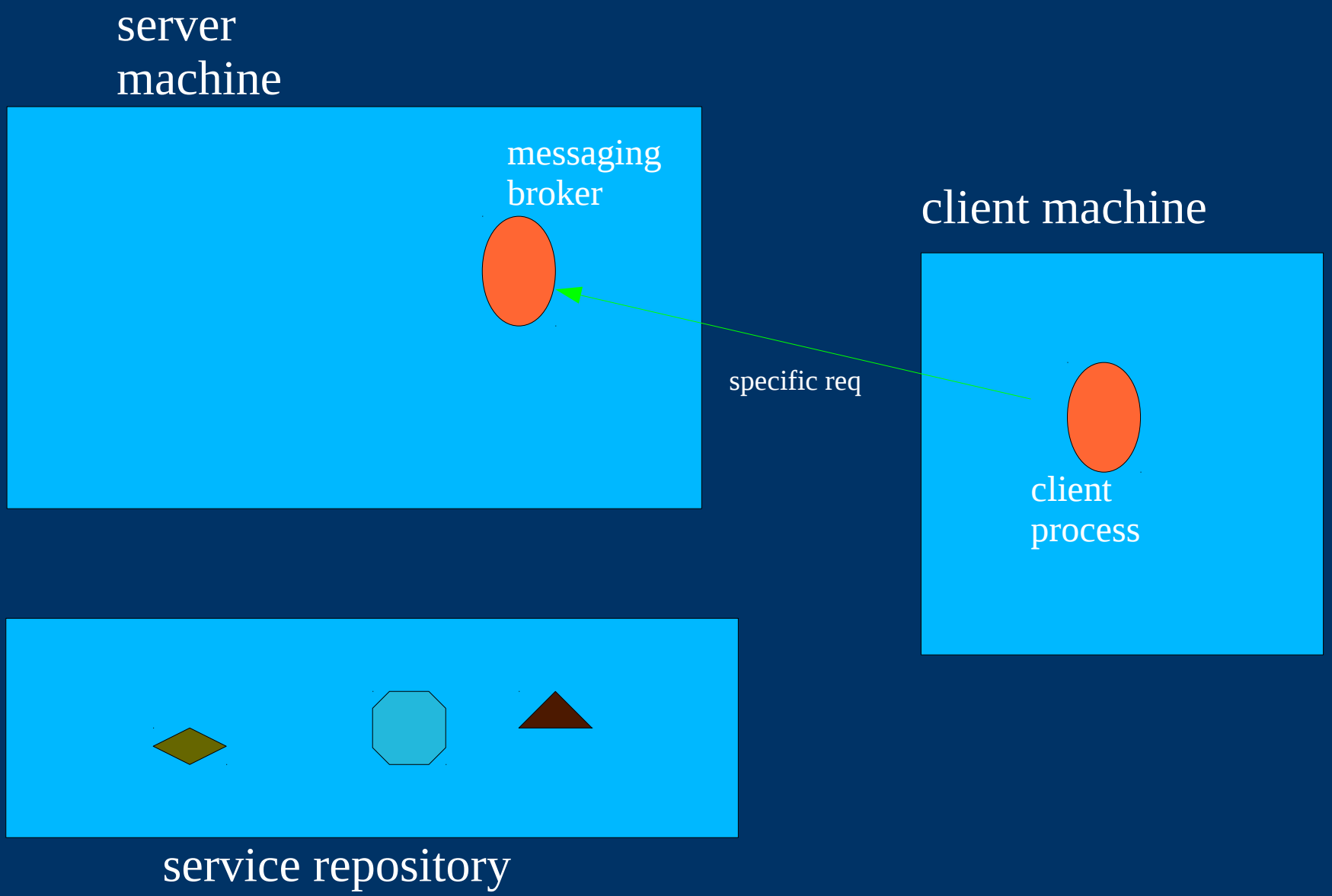
broker--examples

- direct communication broker- connect and then let client communicate directly
- trader broker (select one of many servers)
- middleware brokers (e.g orb in corba)
  - locating servers, supporting interoperability

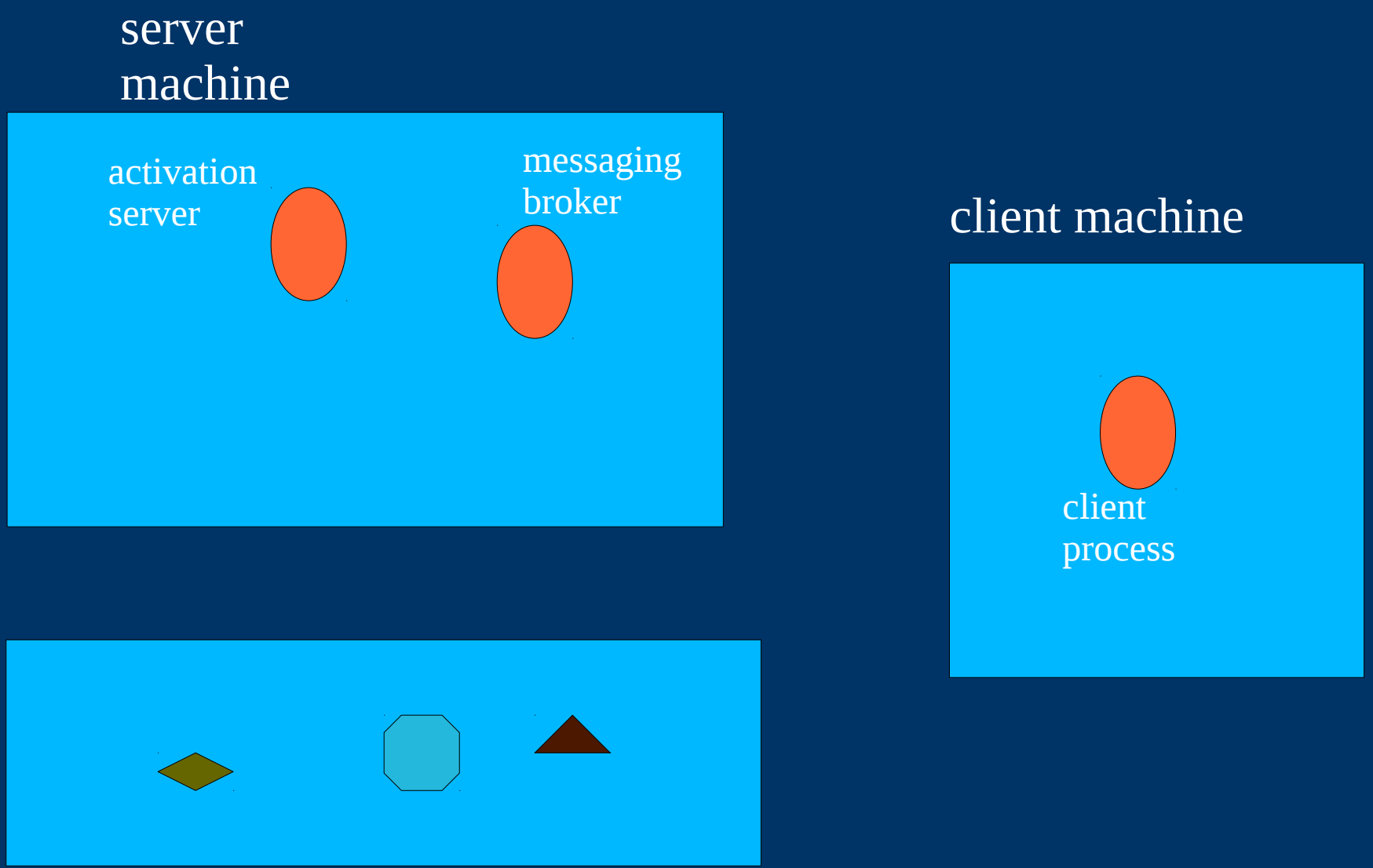
Example: Activation Broker -1



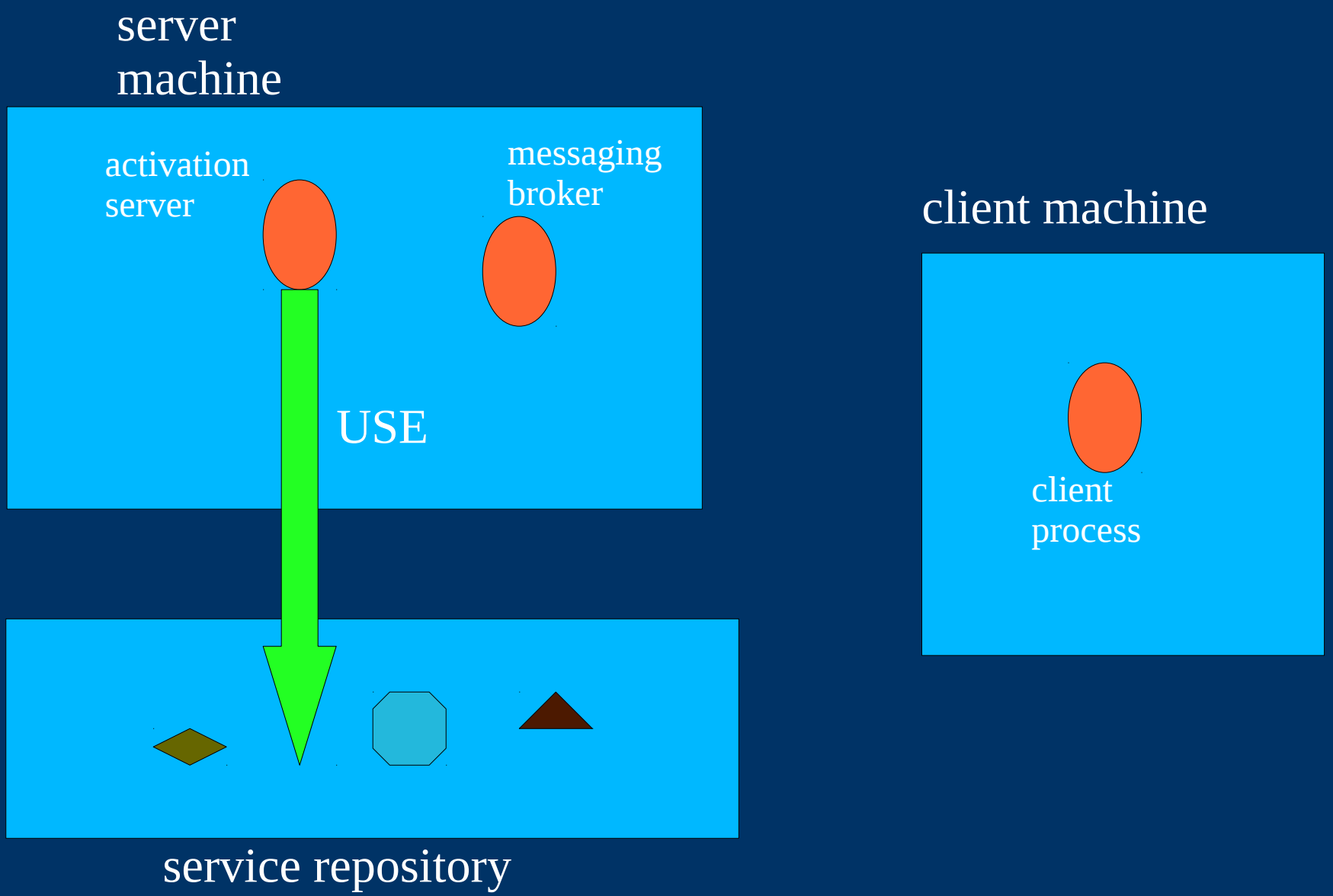
Example: Activation Broker -2



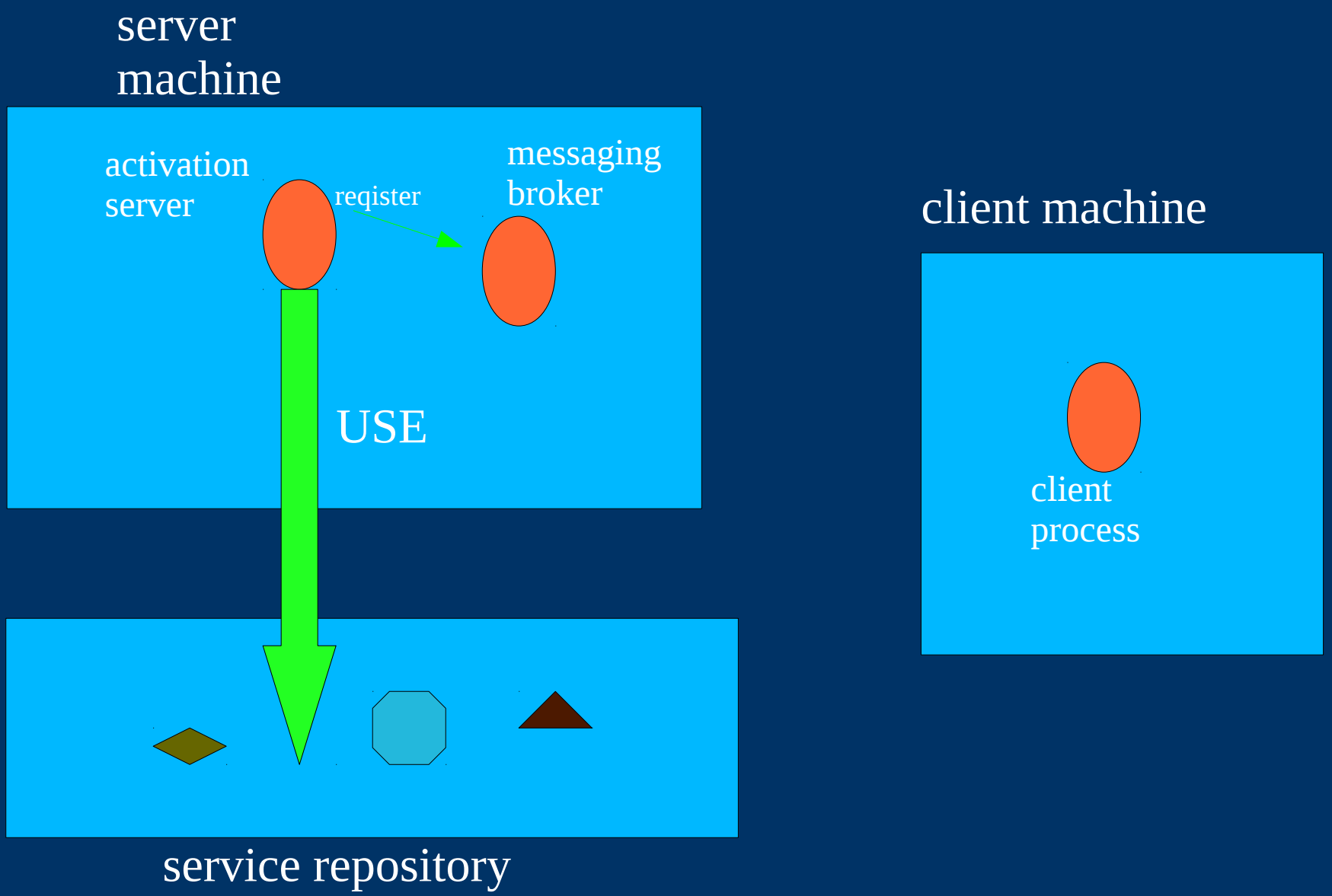
Example: Activation Broker -3



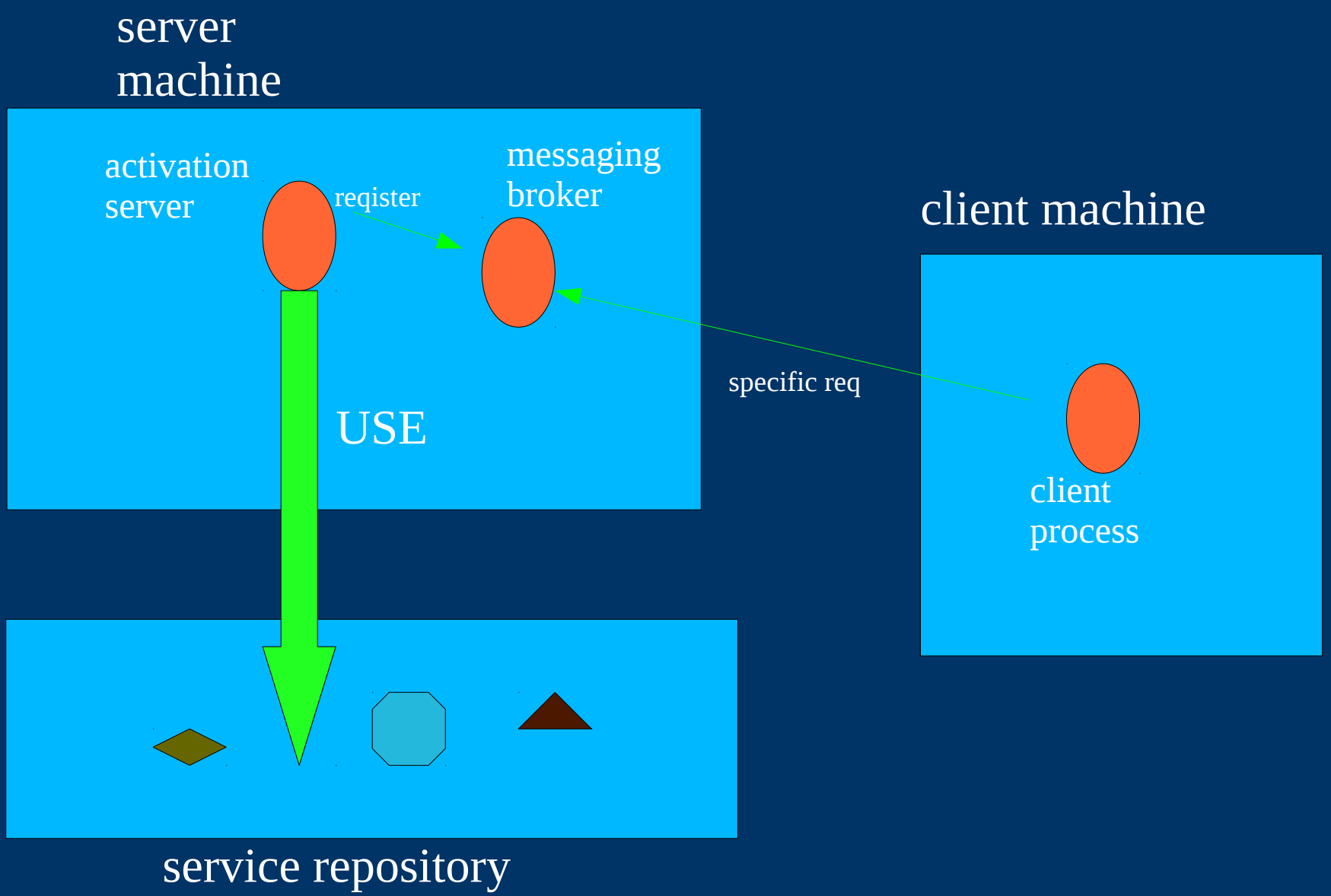
Example: Activation Broker -4



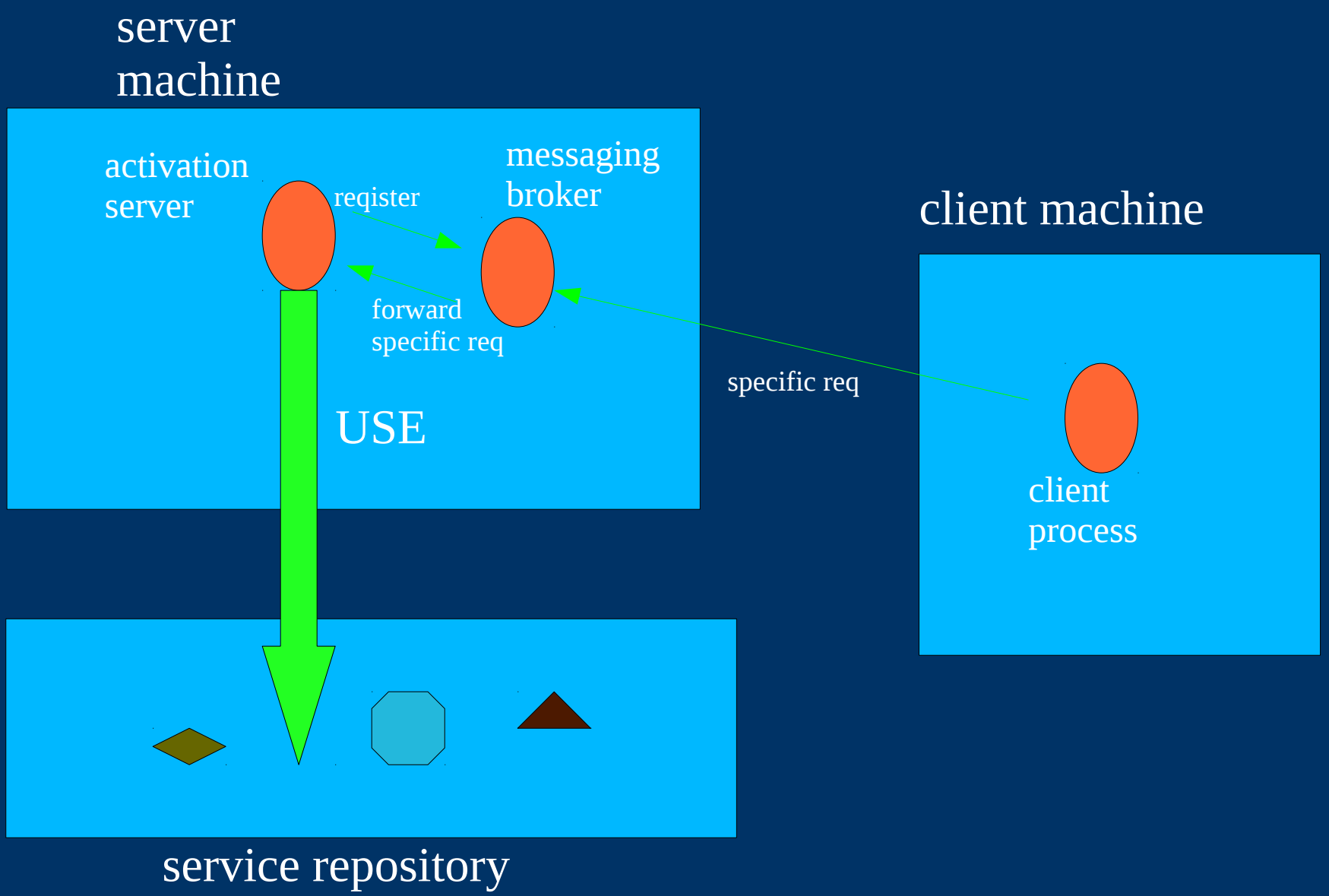
Example: Activation Broker -5



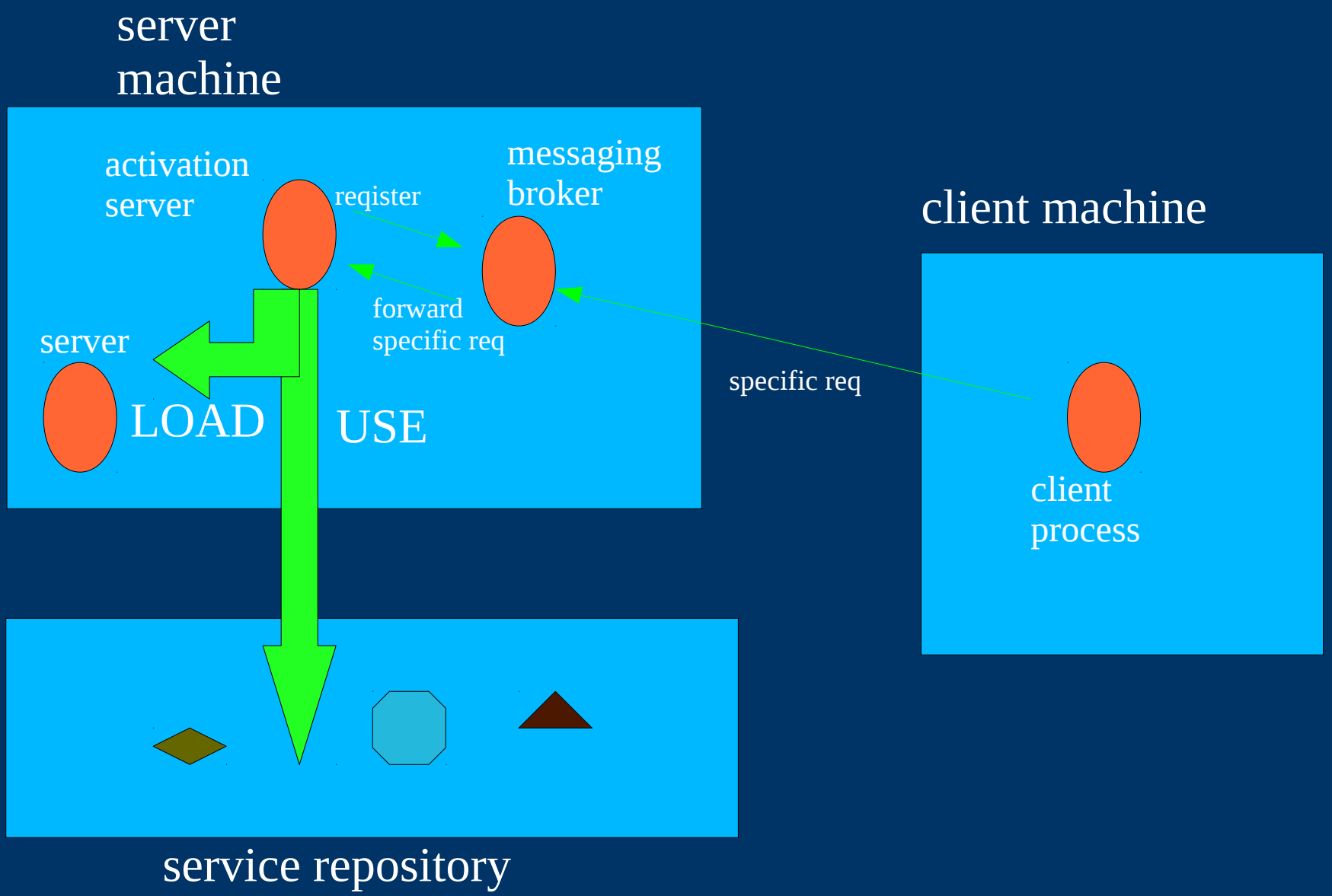
Example: Activation Broker -6



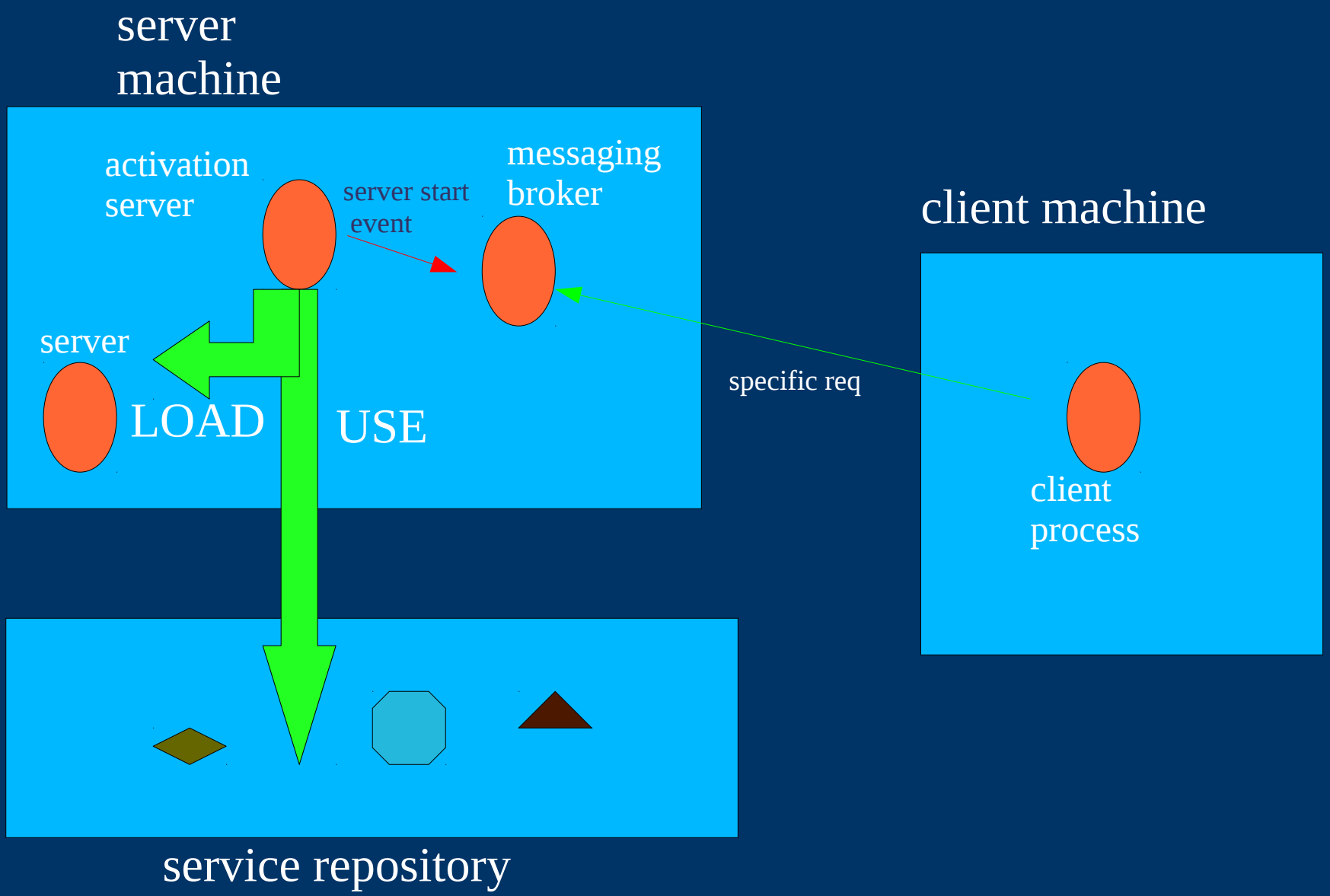
Example: Activation Broker -7



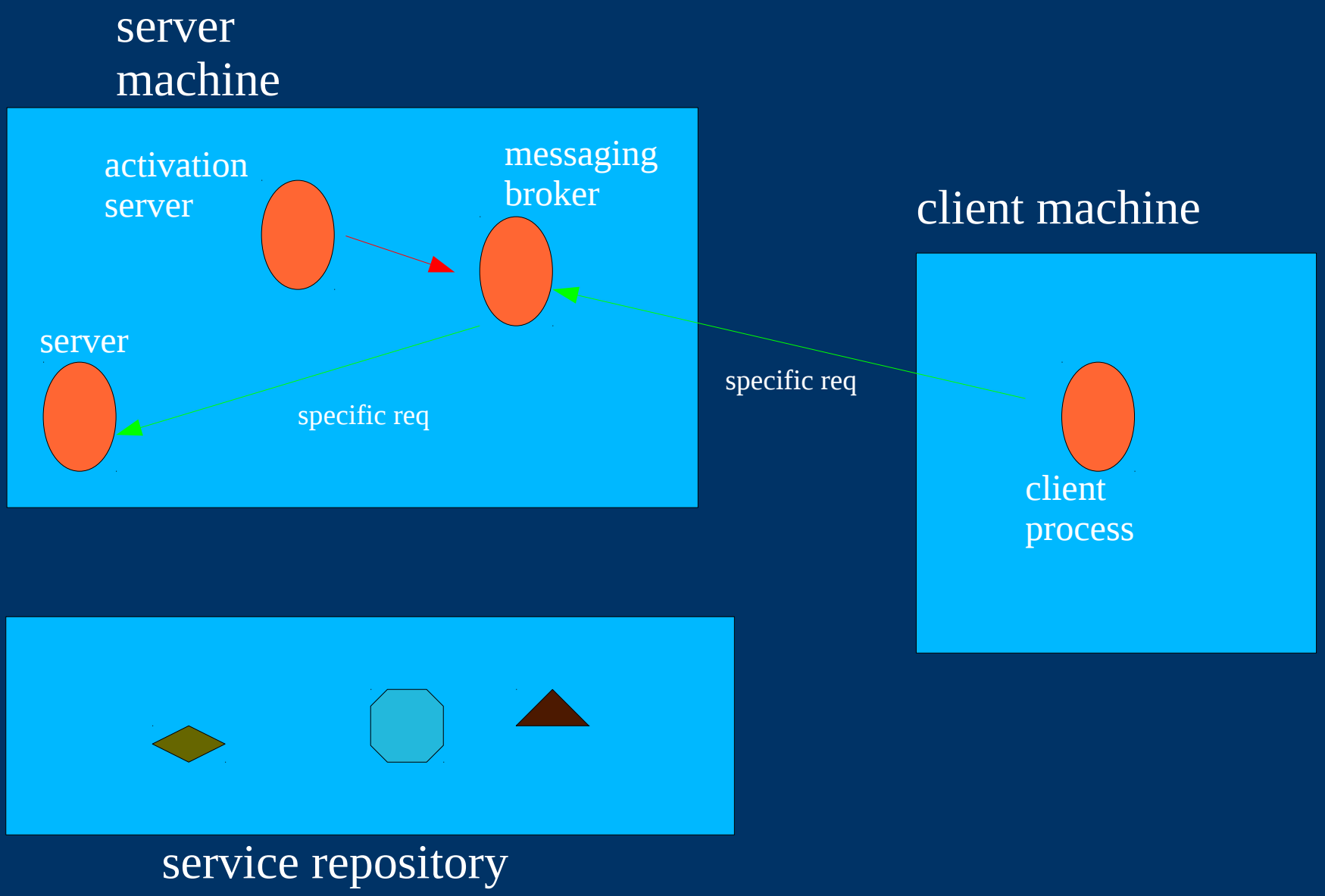
Example: Activation Broker -8



Example: Activation Broker -9



Example: Activation Broker -10



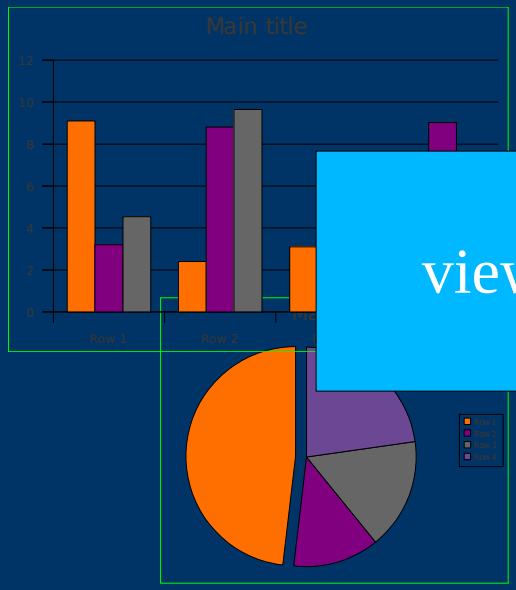
Activation Policies

- Per client's request
- Per client
- Per service request
- Per server

Who does the registration into repository

- A separate application that creates server implementations
- Server implementations are registered in implementation repositories
- Server interfaces can be registered with interface repositories

Model - View – Controller -1



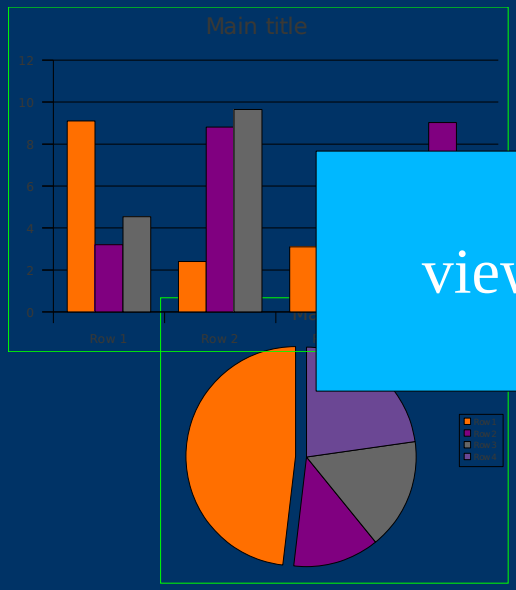
view

controller

Model

pradeep		20	1
vinay	20	10	2
namita	30	15	2
yogita	30	12	1

Model - View – Controller -2



view

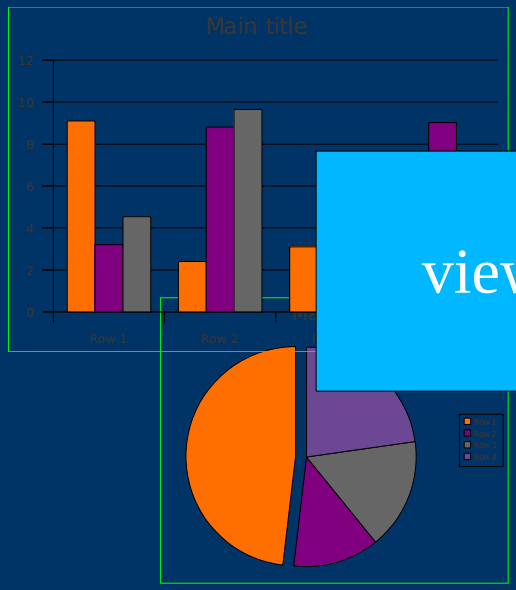
user/control inputs

controller

Model

pradeep		20	1
vinay	20	10	2
namita	30	15	2
yogita	30	12	1

Model - View – Controller -3



view

user/control inputs

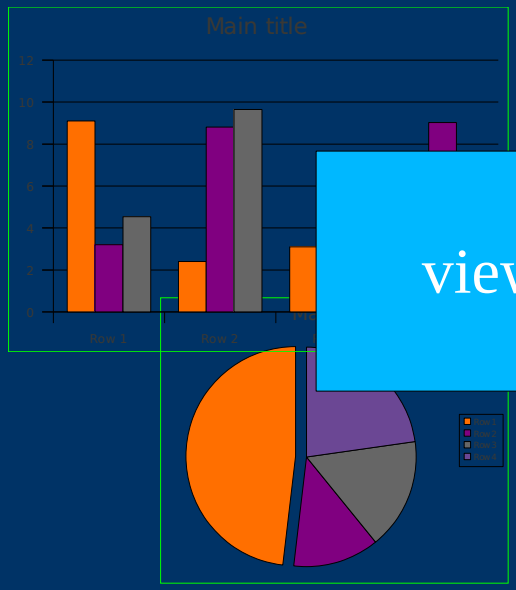
controller

update model

Model

pradeep		20	1
vinay	20	10	2
namita	30	15	2
yogita	30	12	1

Model - View – Controller -4



view

user/control
inputs

controller

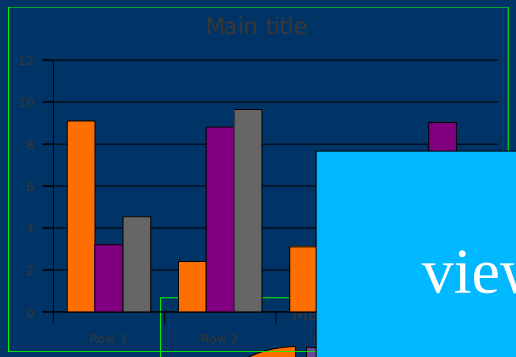
change
intimation

update
model

Model

pradeep		20	1
vinay	20	10	2
namita	30	15	2
yogita	30	12	1

Model - View – Controller -5



view

controller

user/control
inputs

change
intimation

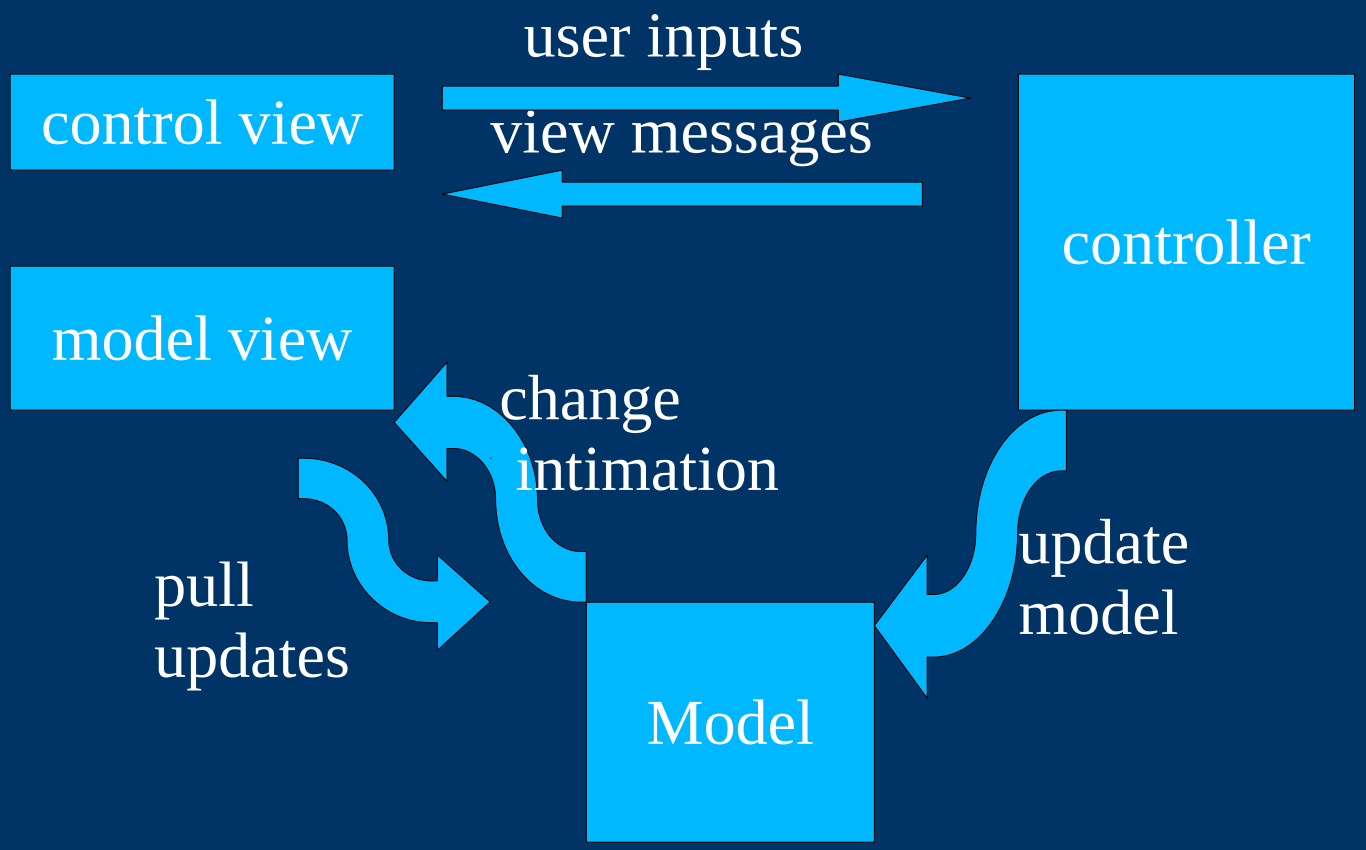
pull
updates

update
model

Model

pradeep		20	1
vinay	20	10	2
namita	30	15	2
yogita	30	12	1

Model - View – Controller -6



input events sent to controller
model contains the state information
model is displayed in view

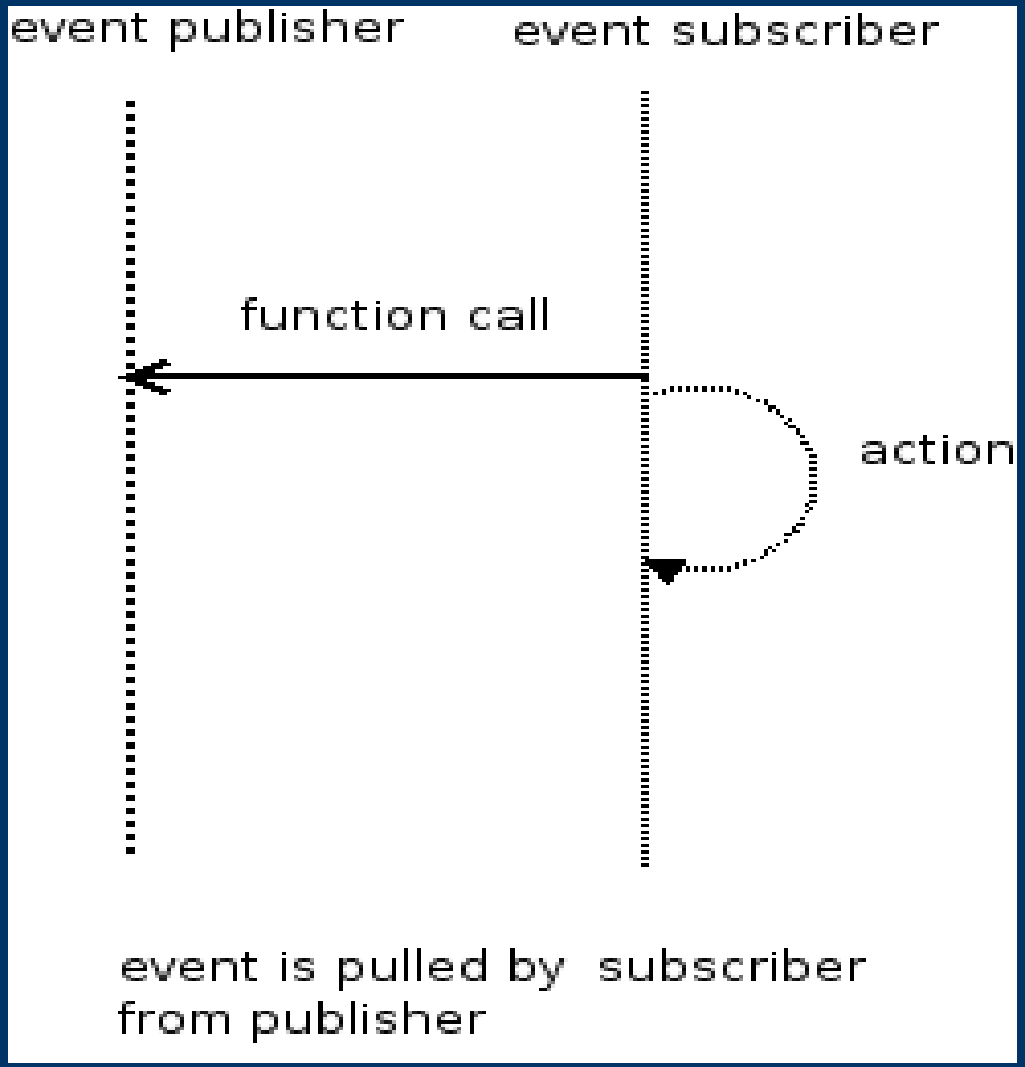
Properties of MVC

- controller not aware how the state is displayed
- model not aware of how the state is displayed
- controller does not directly inform the view about updates
- model sends update intimations to view
- the view can pull in the updates

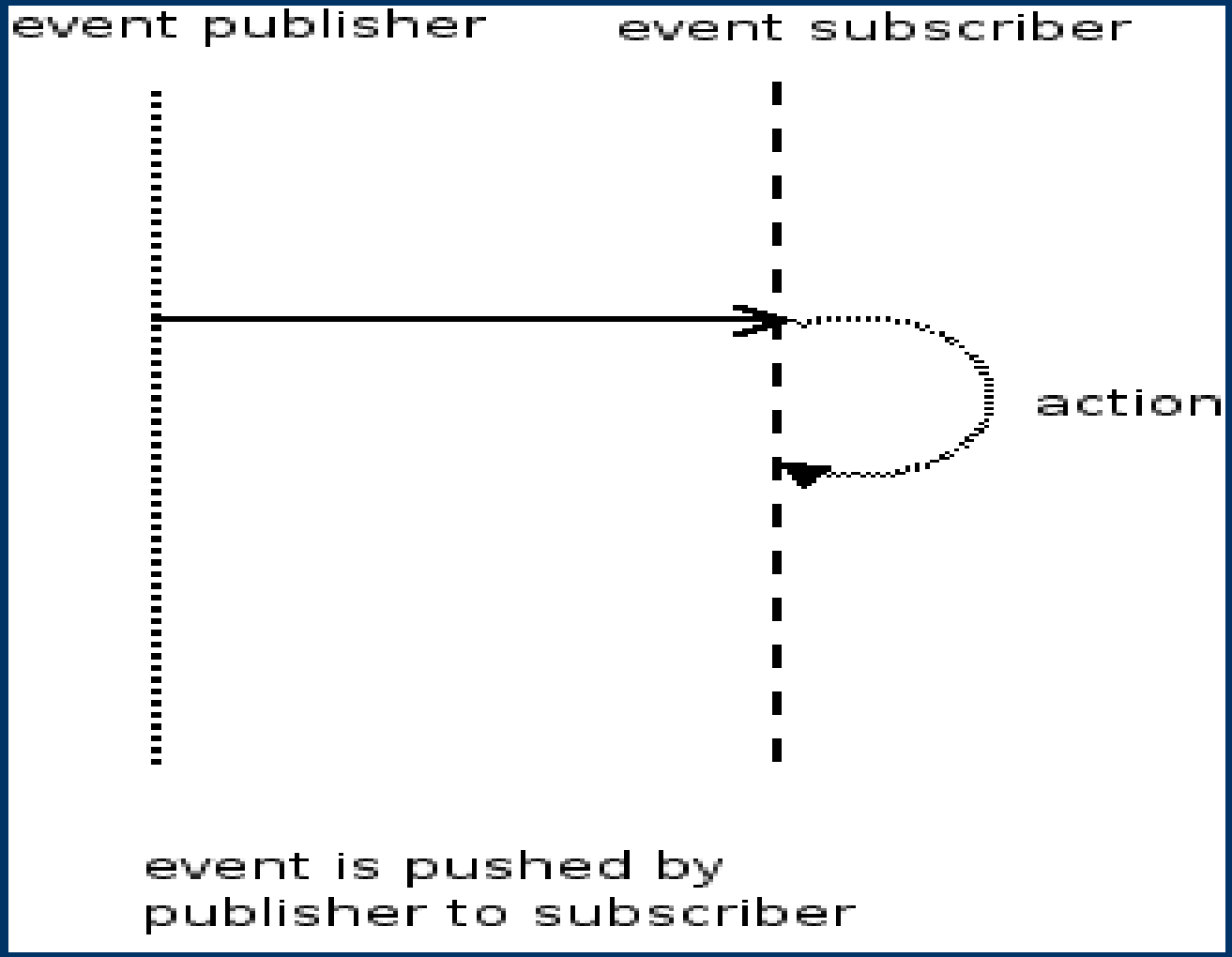
mvc -- examples

- smalltalk MVC – different look and feel standards
- Java pet-store example
- observer based systems

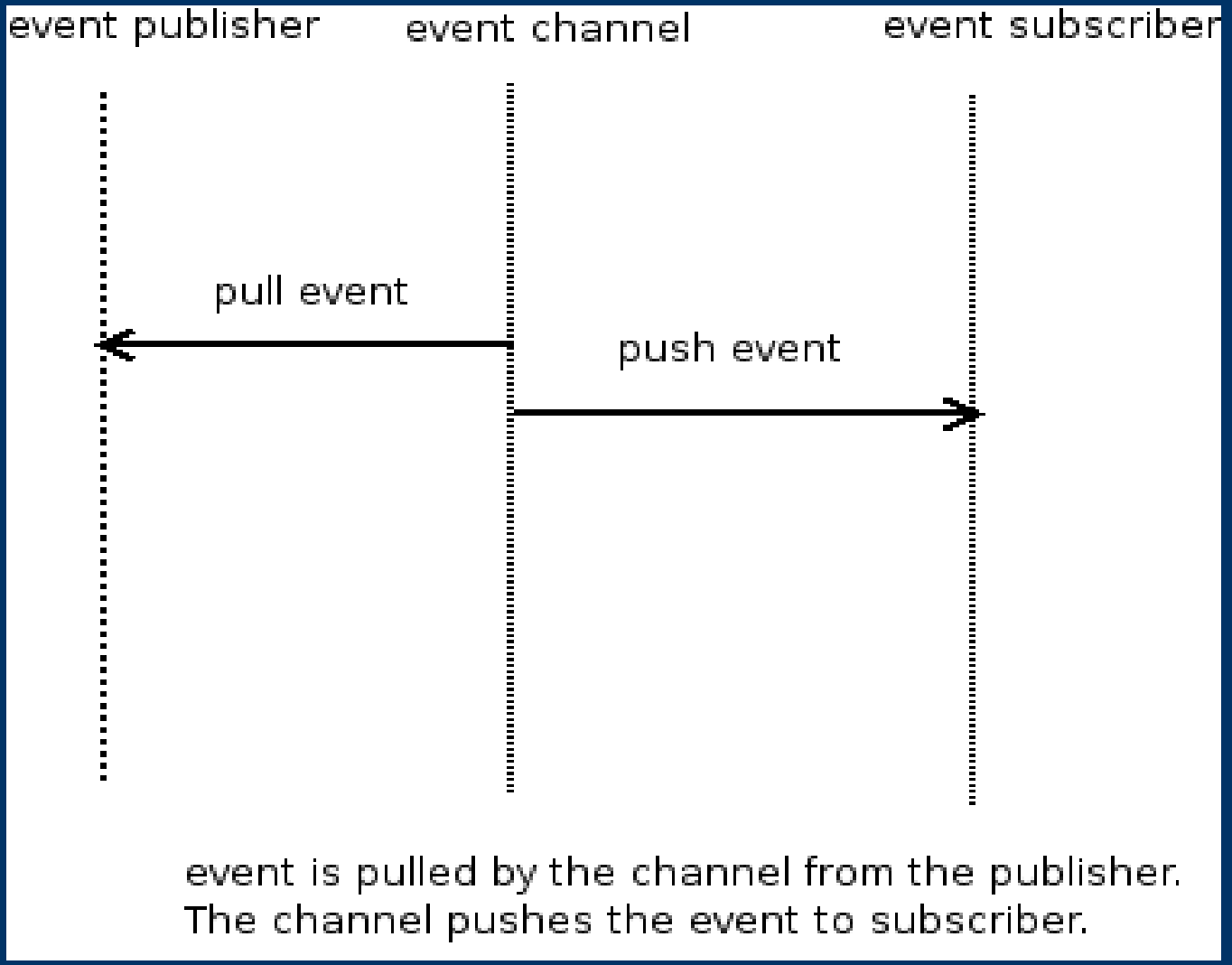
Pull Type Architecture



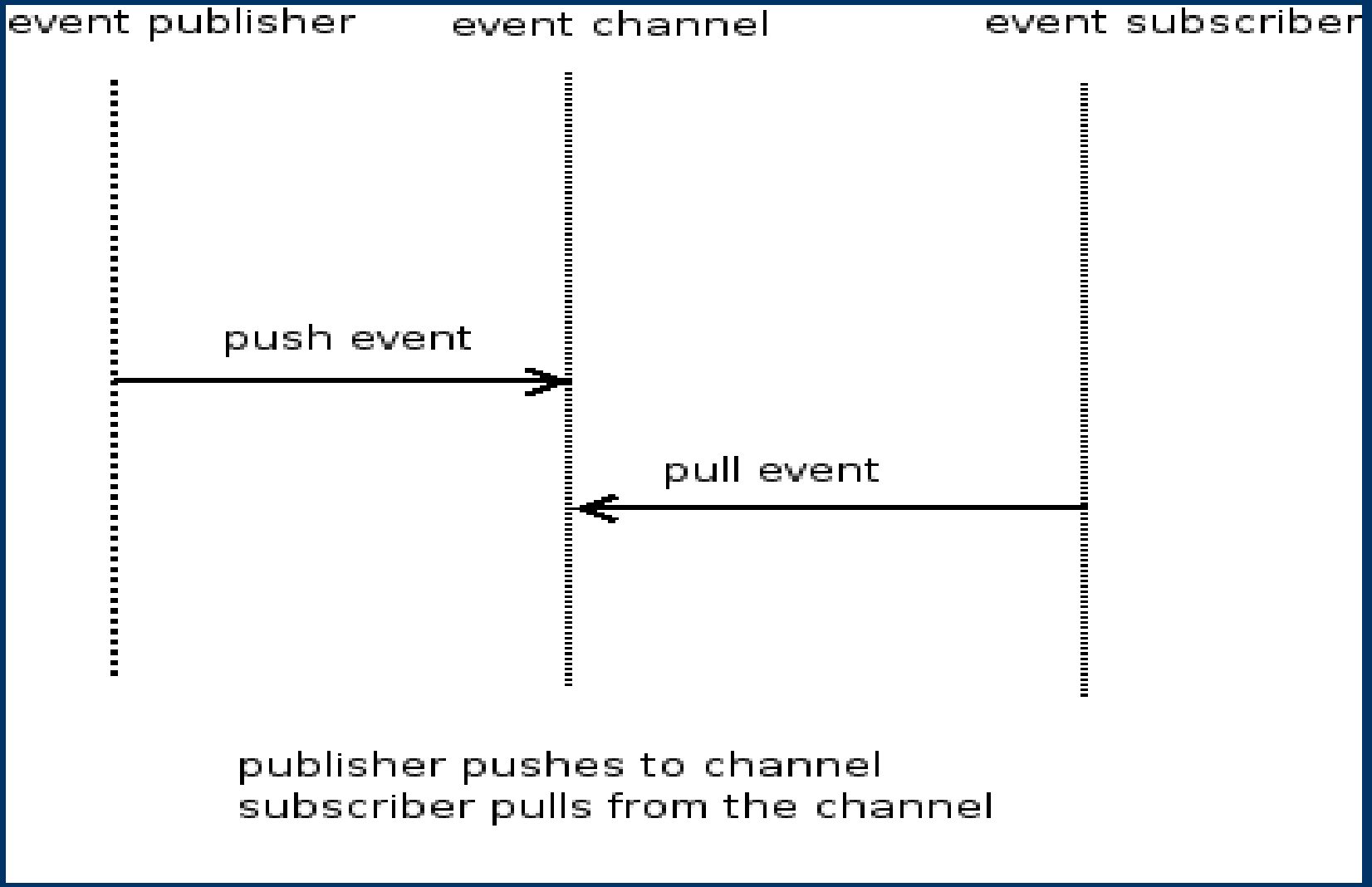
Push type Architecture



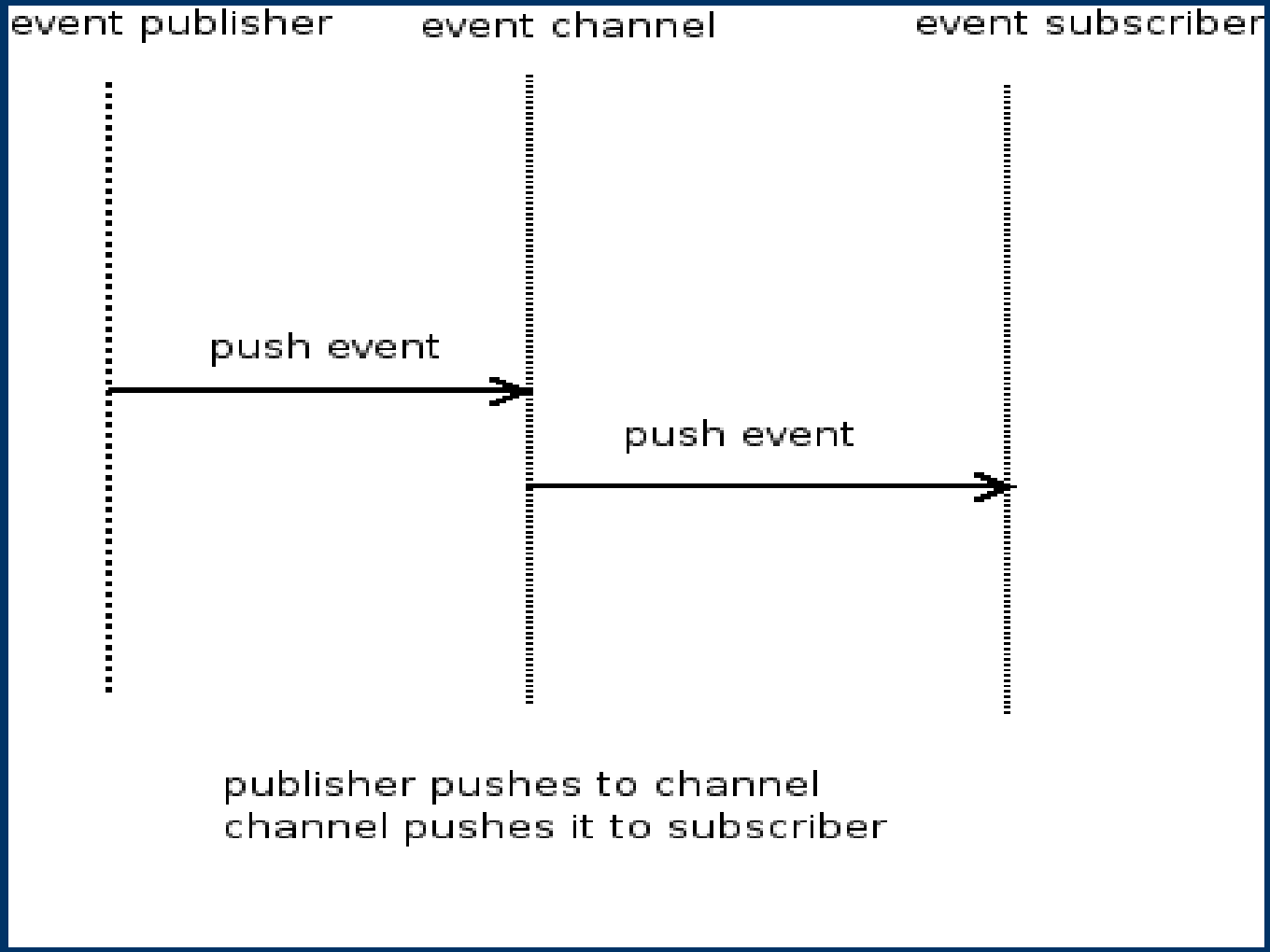
Combining them.. Pull Push Architecture



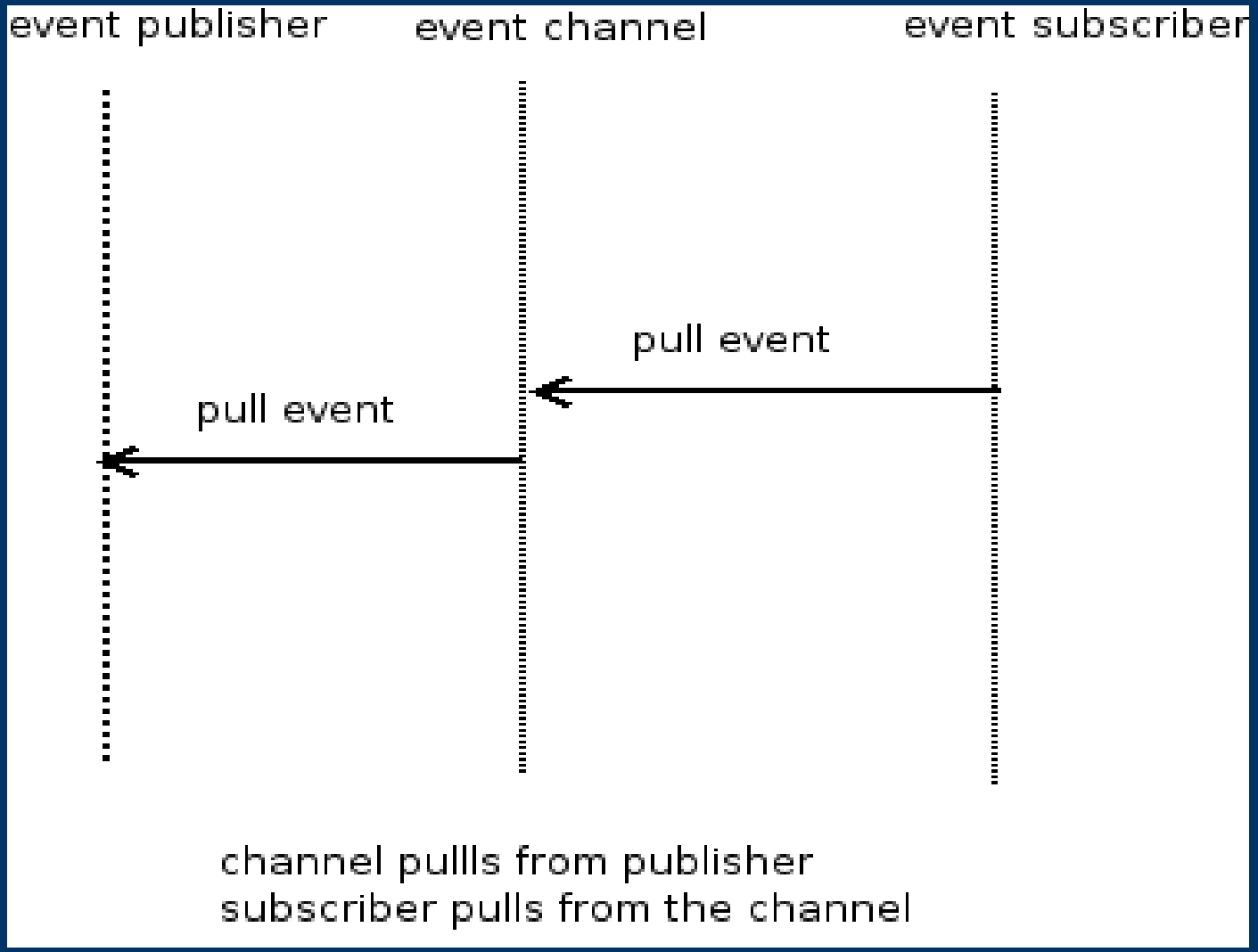
Combining them.. Push Pull Architecture



Combining them.. Push Push Architecture



Combining them.. Pull Pull Architecture



Design issues

- Buffer spaces
  - at source
  - at intermediate channel
- Connectivity
  - dynamic
  - disconnections
- Service orientation, payment model
- Failure Handling
- Performance

Design of interfaces

```
interface pushConsumer {

      notifyEvent(Event e);
      disconnectingPushPublisher();

}
```

```
interface pushPublisher {

      subscribe (pushConsumer   c)
      unsubscribe (pushConsumer  c)

}
```

```
interface pullPublisher {

     boolean subscribe (pullConsumer c);
     unsubscribe (pullConsumer c);
     Event pullevent ();
}
```

```
interface pullConsumer {

      disconnectingPullPublisher();
}
```