

Väliohjelmistojen tyypit

Tyypit

- Proseduuripohjainen
- Viestiperustainen
- Tapahtumaorientoitu
- Objekti- ja komponenttipohjainen

Teemat

- Arkkitehtuurimallit
- Palvelut
- Sisärakenteet
- Laajennustrendit
- Realisaatioita

Proseduraaliset väliohjelmistot

Teemat

- Järjestelmämallin tarpeet
- Palvelut ja rajapinnat
- Sisärakenteet
 - networking and interoperability
 - server control
 - system administration infrastructure
- Sovellustilanteen arkkitehtuurimallikkeita
- Realisaatioita
- Laajennustrendit

RPC-malli

- tavoitteena saantituntumaton proseduurikutsu
- muttei voi täysin täyttää tätä tavoitetta
 - Nimentä, vikaantumiset, suorituskyky paljastavat
 - Globaalit muuttujat, kontekstiriippuvat mjat, osoittimet
 - Call-by-reference vs. call-by-value
- Kutsusemantiikat
 - Maybe, at-least-once, at-most-once
 - Exception delivery
 - Asynchronous RPC
 - Multicast, broadcast
- Muut kutsujen ominaisuudet
 - Location transparency, migration transparency, ...
 - Concurrent processing

Request-Reply protocol

- Issues in marshalling of parameters and results
 - input, output
 - data representation
 - passing pointers
- Distributed object references

Handling failures

- Types of failure
 - client unable to locate server
 - request message lost
 - reply message lost
 - server crashes after receiving a request
 - client crashes after sending a request
- Reporting failures breaks transparency

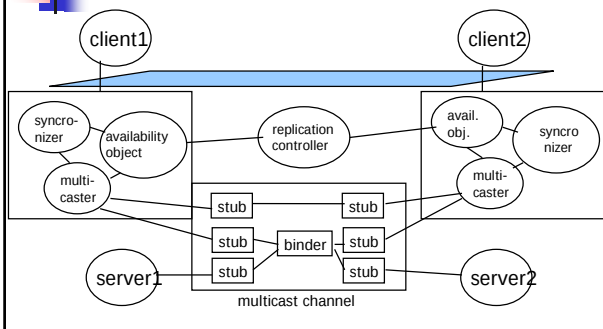
Handling failures

- Lost request message
 - retransmit a fixed number of times before throwing an exception
- Lost reply message
 - client resubmits request
 - server chooses
 - re-execute procedure: service should be idempotent
 - filter duplicates: server should hold on to results until acknowledged
- Client crashes
 - orphan detection: reincarnations, expirations

Fault tolerance measures

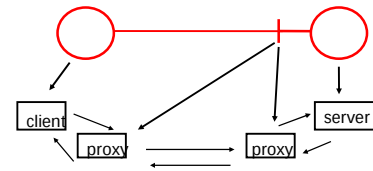
Retransmit request	Duplicate filtering	Re-execute/retransmit	invocation semantics
no	not appl	not appl	maybe
yes	no	re-execute	at-least-once
yes	yes	retransmit reply	at-most-once

"Replication transparent RCP"

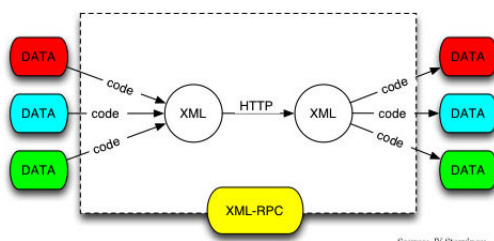


RPC middleware - model

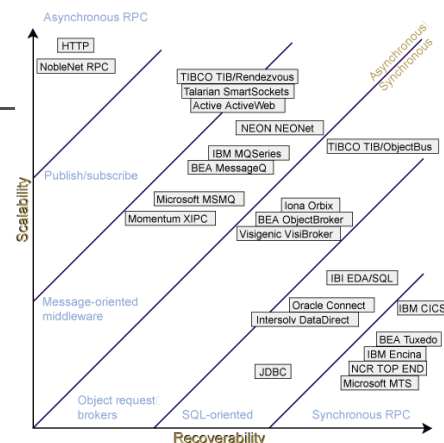
- First class - integrated with language, e.g. Java RMI
- Second class - language-independent IDL, e.g. CORBA



XML-RPC



Source: JV Strevinsou



RPC:n ongelmia

- Synkroninen vuorovaikutus
 - tiukka sidonta asiakkaan ja palvelijan välillä
 - voi olla odotustilassa pitkään
 - johtaa monisäikeiseen ohjelmointiin
- Hajautustuntumattomuus
 - ei voida peittää kaikkia ongelmia
- Palvelun käsite puuttuu
 - ohjelmoija ei ole kiinnostunut palvelijasta/palvelimesta
- Objektittomuus
 - kutsut funktio/proseduurikutsuja eivätkä metodi/operaatiokutsuja

Viestipohjaiset väliohjelmistot

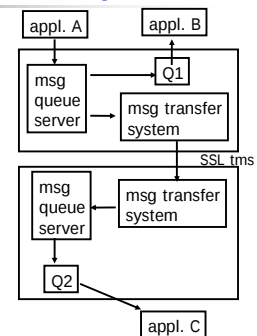
Teemat

- Järjestelmämallin tarpeet
- Palvelut ja rajapinnat
- Sisärakenteet
 - networking and interoperability
 - server control
 - system administration infrastructure
- Sovellustilanteen arkkitehtuurimallikkeita
- Realisaatioita
- Laajennustrendit

Viestipohjaiset väliohjelmistot

Arkkitehtuurimalli

- asynchronous messages
 - reliable, fault-tolerant
 - no loss, duplication, permutation, cluttering
- persistent subscriptions
- models supported
 - message queue
 - request-response
 - multicast
 - publish-subscribe



Viestipohjaiset väliohjelmistot

- MOM = message oriented middleware
- asynchronous messaging natural, synchronous communication cumbersome
- message queues support reliability of message transport
- violates access transparency, no support for data heterogeneity unless in programming language mapping, no support for transactions
- suitable for event notifications, publish/subscribe-based architectures
- persistent message queues support fault tolerance

Viestipohjaiset väliohjelmistot

- Perustila: client-server-malli, putki
 - vrt pistoketason ohjelmointi
- Topics for variation and development
 - persistent/transient msgs
 - FIFO/priority queues
 - translations of msgs
 - abstractions on msg ordering
 - multithreading, automatic load balancing
 - msg routing (source, cost, changes in topology etc)
 - secure transfer of msgs (at least between msg servers)



Viestipohjaiset väliohjelmistot

- Services (MQSeries)
 - 4 types of messages
 - Datagram: one way message, no reply
 - RequestMessage: query initiating message
 - ReplyMessage: response to RequestMessage
 - ReportMessage – "exception report"
 - API: 11 calls
 - Connect/disconnect queue manager, open/close queue, inquire/set (queue) attributes, get/put message from/to queue, put1 (open, put, close), commit/rollback



Rajaavampi määritelmä

- The term "messaging" most often refers to electronic mail systems. MOM message movers, however, differ radically from mail systems, because they are high-speed, generally connectionless and are usually deployed for concurrently executing applications with a non-blocking sender. Message-queuing middleware (MQM) is a type of MOM that combines a high-speed message mover and a queuing (message storage) service.

Korhosen seminaarityö