Server Architectures: Evolution of Software Technologies

January 2005

René J. Chevance

Foreword

- This presentation is an introduction to a set of presentations about server architectures.
They are based on the following book:

Serveurs Architectures: Multiprocessors, Clusters, Parallel Systems, Web Servers, Storage Solutions
René J. Chevance
Digital Press December 2004 ISBN 1-55558-333-4
http://books.elsevier.com/

This book has been derived from the following one:

Serveurs multiprocesseurs, clusters et architectures parallèles
René J. Chevance
Eyrolles Avril 2000 ISBN 2-212-09114-1
http://www.eyrolles.com/

The English version integrates a lot of updates as well as a new chapter on Storage Solutions.

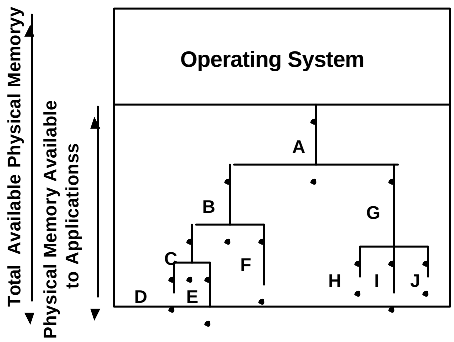
Contact: www.chevance.com rjc@chevance.com

Organization of the Presentations

- Introduction
- Processors and Memories
- Input/Output
- ➔ Evolution of Software Technologies (this presentation)
  - Virtual Memory
  - 64 bit Addressing
  - Operating Systems
  - Client/Server
  - Web Services
  - Transactional Monitors
  - RPC and MOMs
  - Distributed Object Model
  - Enterprise Java Beans
  - Web Servers
  - System Administration
  - Economic model
- Symmetric Multi-Processors
- Cluster and Massively Parallel Machines
- Data Storage
- System Performance and Estimation Techniques
- DBMS and Server Architectures
- High Availability Systems
- Selection Criteria and Total Cost of Possession
- Conclusion and Prospects

Virtual Memory

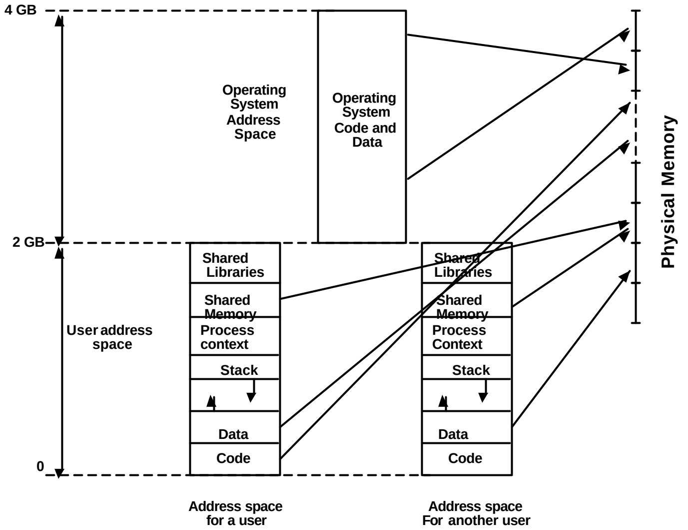
- Without the concept of virtual memory, the mechanism of overlays was used to support objects larger than the amount of physically available memory



- Virtual Memory
  - Virtual memory is (usually) realized onto physical memory on the basis of pages (e.g. 8 KB)
  - Whenever a virtual page has no realization in physical memory, page movement form disk (backing store) is automatically handled by the Operating System (page fault and demand paging mechanism)

Virtual Memory(2)

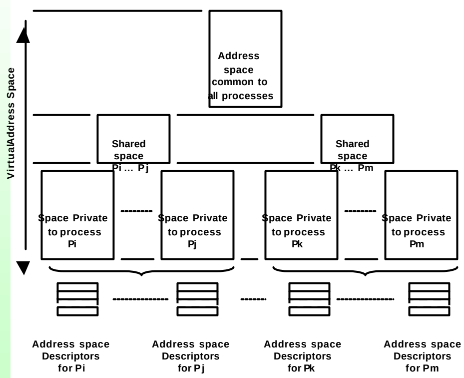
- Illustration of the concept of virtual memory (Unix, Windows)



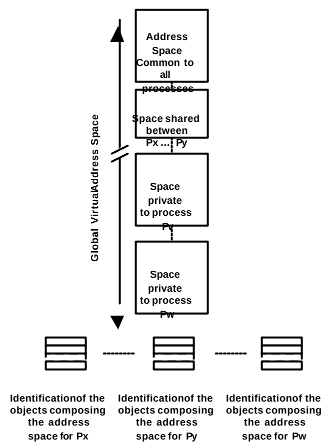
Page 5

Virtual Memory(3)

- Two models:
  - MAS Multiple Address Spaces (processes re-use the same address space)
  - SAS Single Address Space



a) – MAS Multiple Address Space

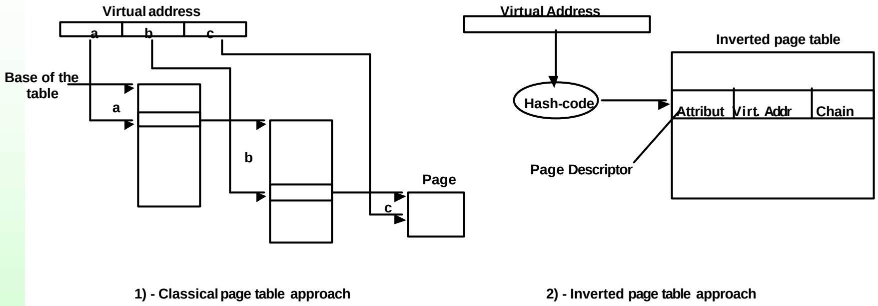


b) – SAS Single Address Space

Page 6

Virtual Memory(4)

- **Virtual Address Translation Mechanisms**



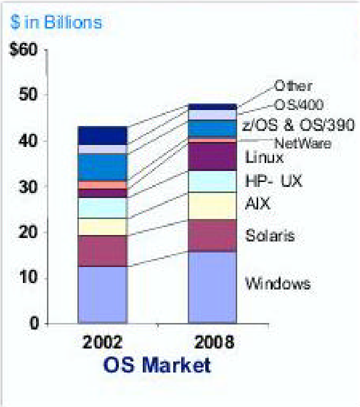
- **Address translation is performed by the processor using address translation caches (TLB – Translation Lookaside Buffers)**
- **Management of the TLB can be done in hardware (performance) or (more frequently now) by software (flexibility)**

64 bit Addressing

- **Supported by most of the processors (RISC, IA-64, AMD and Intel extensions to x86, IBM z Series)**
- **Supported by Operating Systems: Unix, Windows, z/OS**
- Advantages of 64 bit architecture
  - **Support of very large objects (files, databases) directly in virtual memory ➔ Performances**
    - **Address translation done by the processor**
    - **Data migration done by the Operating System (demand paging)**
    - **No need for the software to perform address multiplexing (like buffer management)**
  - **Support of very large file systems (>2 GB)**
  - **Management of very large physical memories**
  - **Becoming a typical requirement of DBMSes and CAD (Computer Aided Design)**

Operating Systems

- **Server sales forecast by operating system (Source Gartner 2003)**



Comments :

- Proprietary systems market share is decreasing sharply (with the exception of OS/400 and z/OS)
- Linux market share is expected to grow quickly
- Windows market share is expanding (specially for low and mid-range servers)
- The cost of developing and maintaining a proprietary version of Unix is approaching the cost of proprietary systems. This phenomenon is leading to a reduction in the number of Unix versions (to the benefit of Linux)

Page 9

Operating Systems(2)

- **Operating Systems Functionality**
  - **Scalability (ideally both dimensions)**
    - SMP
    - Clustering
  - **RAS: Reliability, Availability and Serviceability**
    - Masking hardware failures
    - Reconfiguration capability
    - On-line hardware and software updates
    - Checkpoint and restart capability
    - System partitioning and clustering
  - **File System**
    - Journaling File System
    - Logical Volume Management and support of very large files
    - Save and restore
  - **Distributed services and Internet support**
    - TCP/IP v6
    - Support of Internet tools: Browsers, Web Servers, e-Commerce,….
    - Directory Services
    - Security
    - Distributed File Services
    - Distributed Time Service
    - Inter-Program Communication: RPC (Remote Procedure Call), RMI (Remote Method invocation) or MOM (Message Oriented Middleware)

Page 10

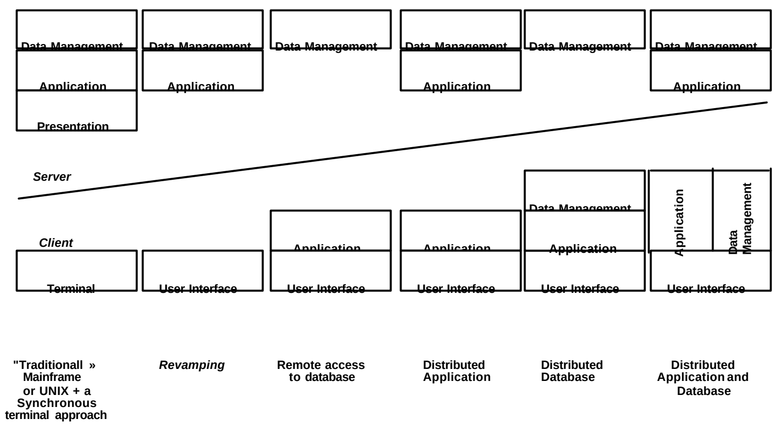
Operating Systems(3)

- **Operating Systems Functionality(2)**
  - **System Management**
    - **Hardware configuration management**
    - **Software configuration management**
    - **User management**
    - **Resource management**
    - **Remote management**
    - **Performance analysis**
    - **Batch processing optimizations**
  - **Capacity to simultaneously support various isolated workloads**
    - **Resource allocation must obey stated rules**
    - **Failure independence**
  - **PC Support**

Page 11

Client/Server

- **Client-Server Architecture options** (after Gartner)

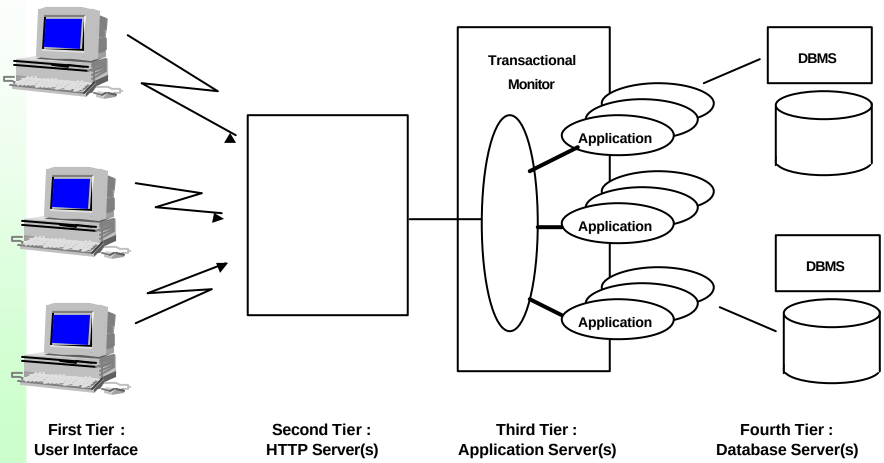


Note: In a Java environment, applications running on the clients are called applets, while those running on the server are known as servlets

Page 12

Client/Server(2)

Multi-Tier Client/Server



Page 13

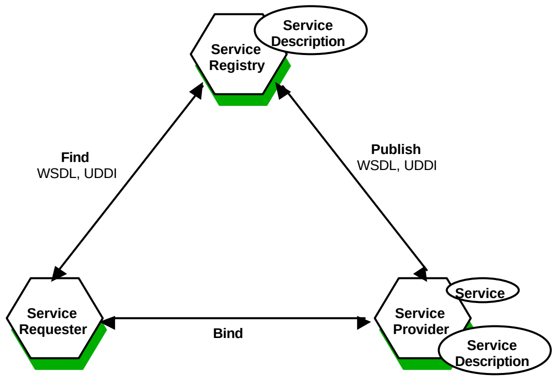
Components of client-server middleware

Middleware components [MEI99]

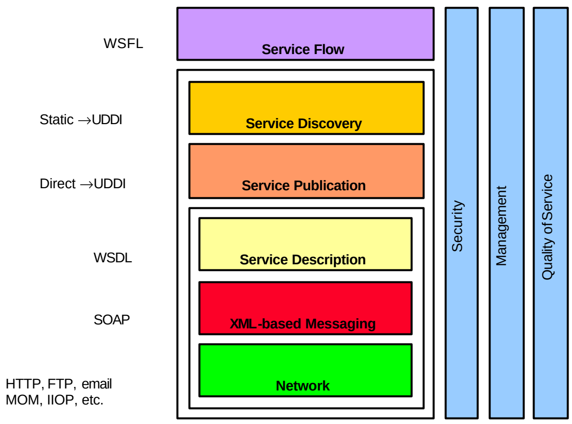
	Internet	Dialogue	Data Access	Transactions	Objects	
Application Services	HTTP S-HTTP SSL	HTML, Windows, Applets Java	SAG/CLI, RDA, DRDA, ODBC, JDBC	X/Open (Tuxedo, Encina, CICS 6000,...)	ORB (CORBA, COM+, ...)	•••
Distributed Environment Services	System Administration (SNMP, ..)	Directories (LDAP)	Security (Kerberos)	Distributed Time	PVM MPI	•••
Network OS Services	RPC	MOM Message Queues	IPC Remote Inter-Process Communication	Distributed File Systems (NFS, DFS)	•••	
Communication Services			TCP/IP			
Operating System Services	Processes And Threads	IPC Local Inter-Process Communication	Local File Systems	•••		

Page 14

Web Services Actors, Roles and Operations
(Source IBM)



Web Services Conceptual Stack (Source IBM)

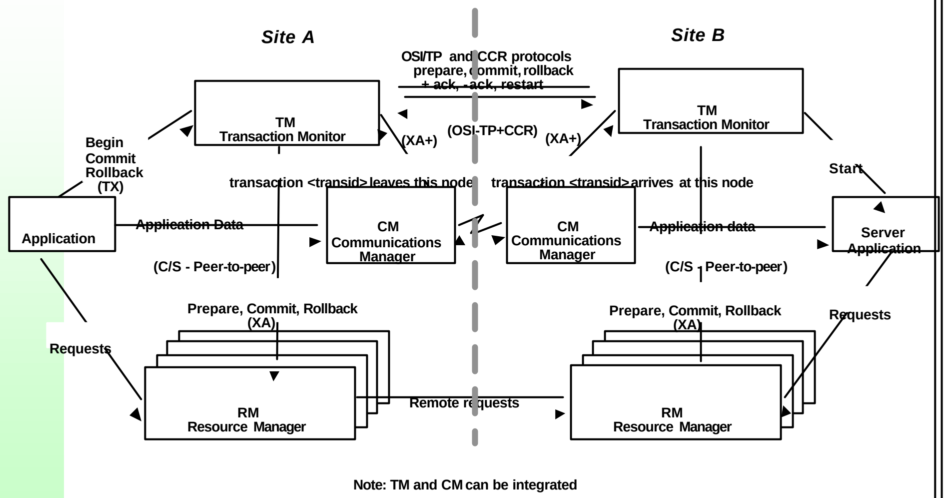


Transactional Monitors

- As very few Operating Systems integrate « native » transactional services, they resort to « transaction monitors » e.g. CICS, Tuxedo
- The support of very large numbers of simultaneous users causes performance problems. The concept of thread was created (in the 60's) to solve this issue (see SMP presentation)
- Functions of a transaction monitor:
  - Thread management, including launching the applications needed to handle user requests (whether users on workstations or requests coming from other systems), controlling their execution and doing load-balancing
  - Transaction management (ensuring that the ACID properties are respected) in a context which may be distributed and which can have several database managers involved in transaction execution.

Transactional Monitors(2)

- X/Open DTP Model [GRA93]



Transactional Monitors(3)

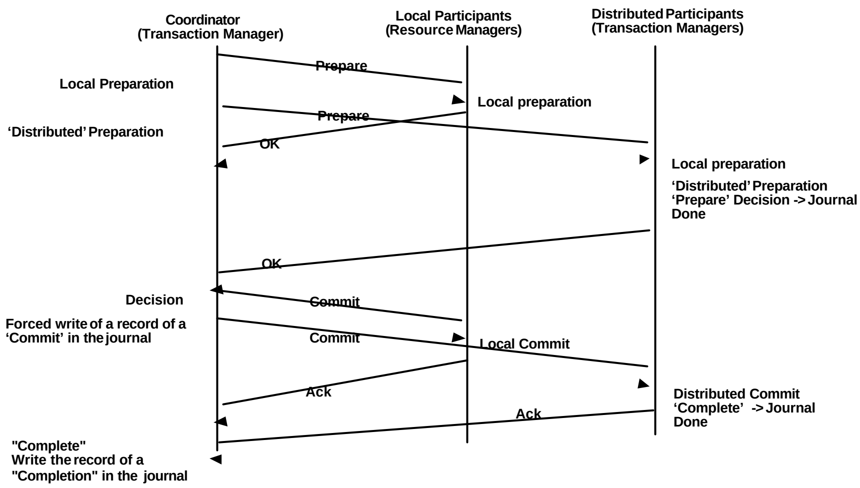
- **X/Open DTP Model(2)**

Participating elements	Protocol or interface (API)	Organization Involved
Application-TM	TX	X/Open DTP
Application-RM	Specific to RM	RM suppliers
Application- serveur	client-server type communications	OSI and application suppliers
TM-RM	XA	X/Open DTP
TM-CM	XA+	X/Open DTP
TM-TM	OSI-TP + CCR	OSI

- **Three possibilities for Client/Server communication:**
  - **A ‘transactional’ RPC (Remote Procedure Call) or RMI (Remote Method Invocation), so-called because the RPC or RMI mechanism (which here is just a CM) must use the XA+ interface to communicate with TM**
  - **A Peer to Peer dialog**
  - **A Message-Oriented Middleware, or MOM**

Transactional Monitors(4)

- **Principles of the Two-Phase Commit Protocol**



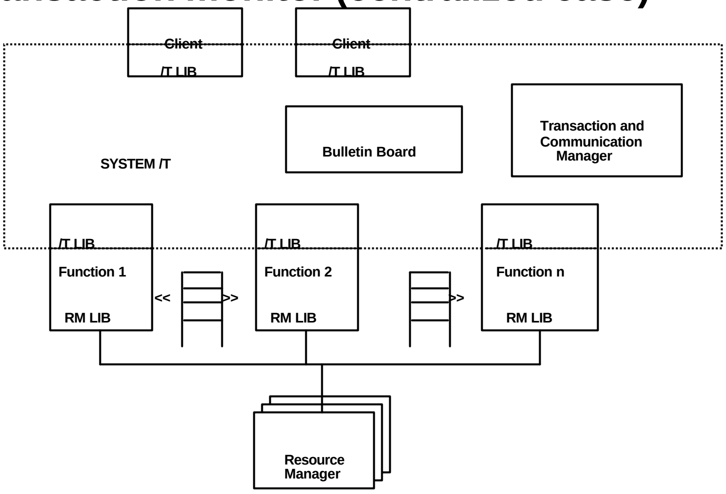
Transactional Monitors(5)

- **Example of a transactional monitor: Tuxedo**
  - Initially developed by AT&T for Unix to fufil its own, owned know by BEA (after USL - Unix System Laboratories and Novell)
  - Commercially available in 1989. Several thousands systems installed
  - Available on a wide variety of systems
  - Characteristics
    - Conform to the X/Open DTP Model (Distributed Transaction Processing)
    - Portability
    - High Level Language support (e.g. Visual Basic, Cobol)
    - Client/Serveur architecture
    - System management
    - Clients - Servers multiplexing (threading)
    - Queueing mechanism
    - Distributed transactions
    - Security

Page 21

Transactional Monitors(6)

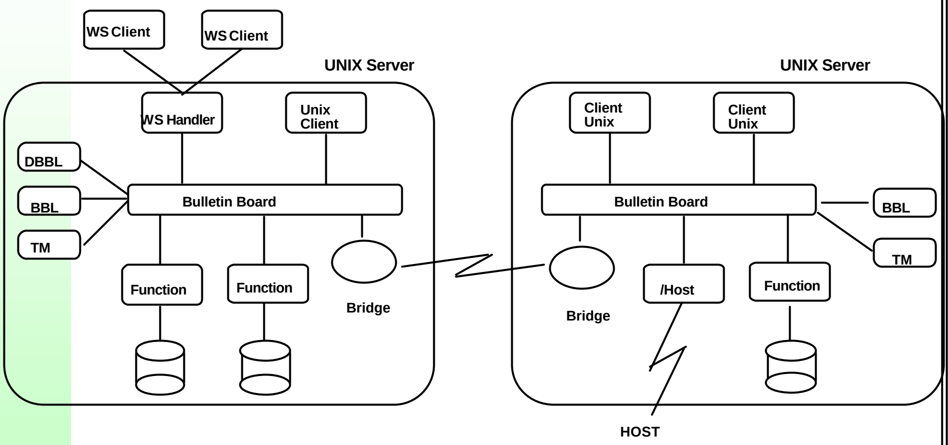
- **Tuxedo, an example of architecture of a transaction monitor (centralized case)**



Page 22

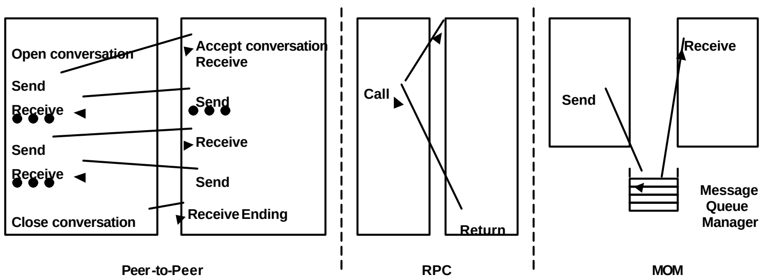
Transactional Monitors(7)

- **Tuxedo, an example of architecture of a transaction monitor (distributed case)**



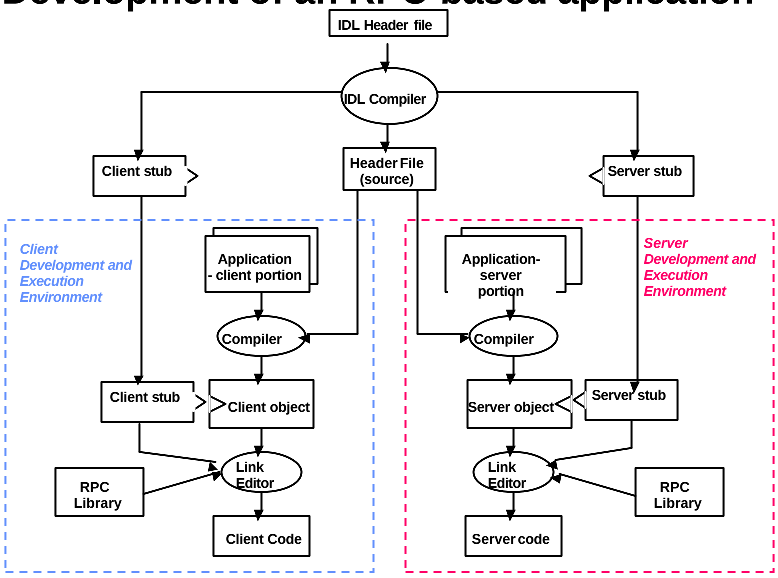
RPC and MOMs

- **Modes of communication between programs:**
  - **Peer-to-peer**
  - **RPC (Remote Procedure Call) or RMI (Remote Methode Invocation)**
  - **MOM (Message Oriented Middleware)**



RPC and MOMs(2)

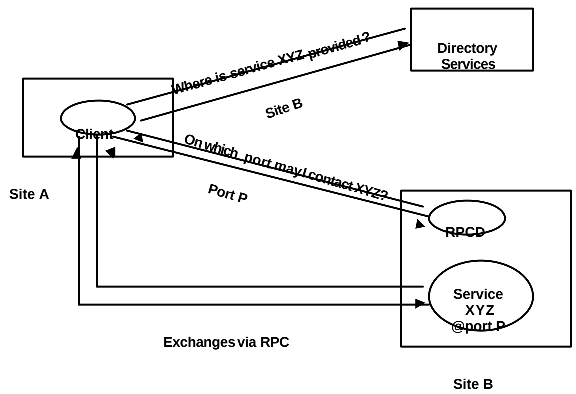
■ Development of an RPC-based application



Page 25

RPC and MOMs(3)

■ Operation of an RPC



Page 26

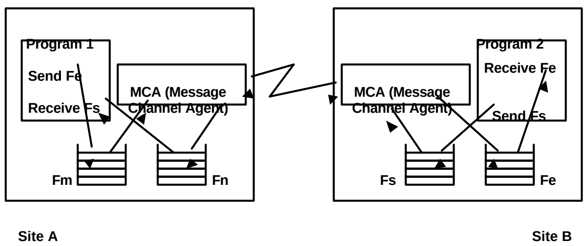
RPC and MOMs(4)

Comparing the Characteristics of RPC and MOM

Characteristic	MOM	RPC
Metaphor	Mail	Telephones (without answering machine)
Temporal relation between client and server	Asynchronous	Synchronous
Nature of the communications	Queue	Request-answer
Operational state of the server	Not necessary	Mandatory
Load Balancing	Policy of extraction of the messages (priority system)	By means of a transaction monitor
Transaction support	Depends on the product	Depends on the product (required of a transactional RPC)
Message filtering	Possible	No
Performance	Slow if messages are made secure by writing to disk	More effective than MOM since call parameters are not saved to disk

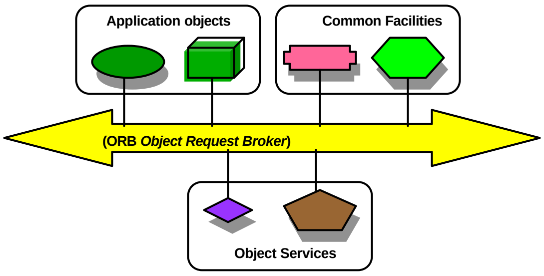
RPC and MOMs(5)

Communication by means of MQSeries

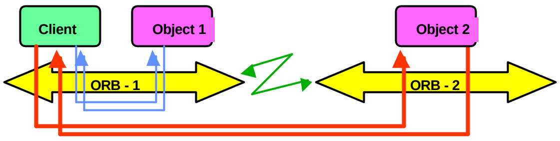


Distributed Object Model

■ **CORBA Reference Model**

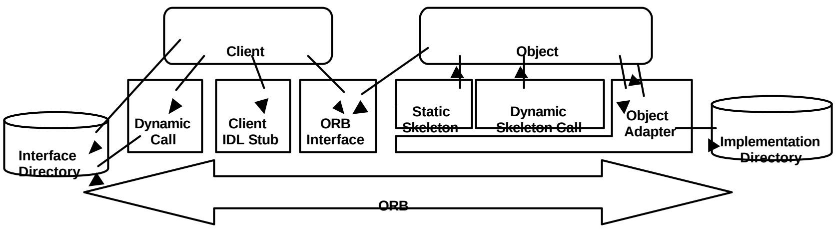


■ **Exchanges with the ORB**

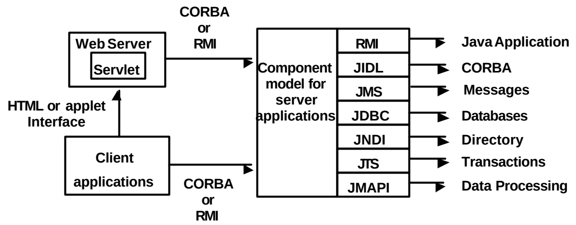


Distributed Object Model(2)

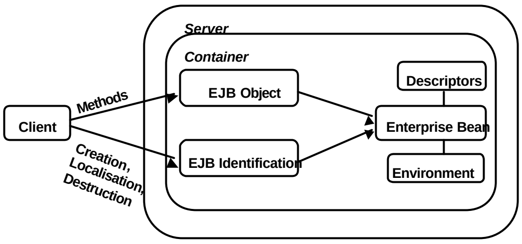
■ **Functional architecture of CORBA**



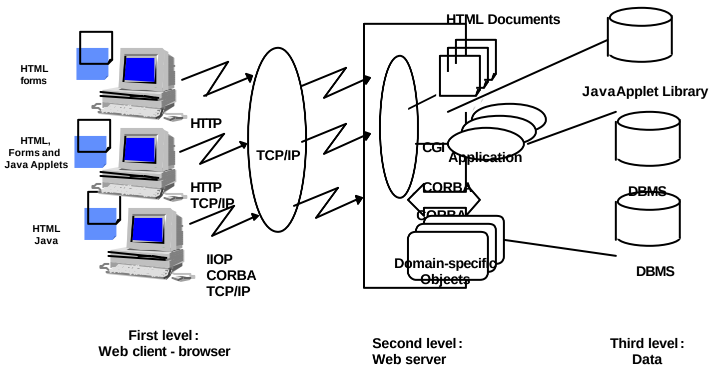
- **Enterprise Java Beans – A Component-Oriented Application Model**
  - **EJB Support Services**



  - **Structure of an EJB Container**

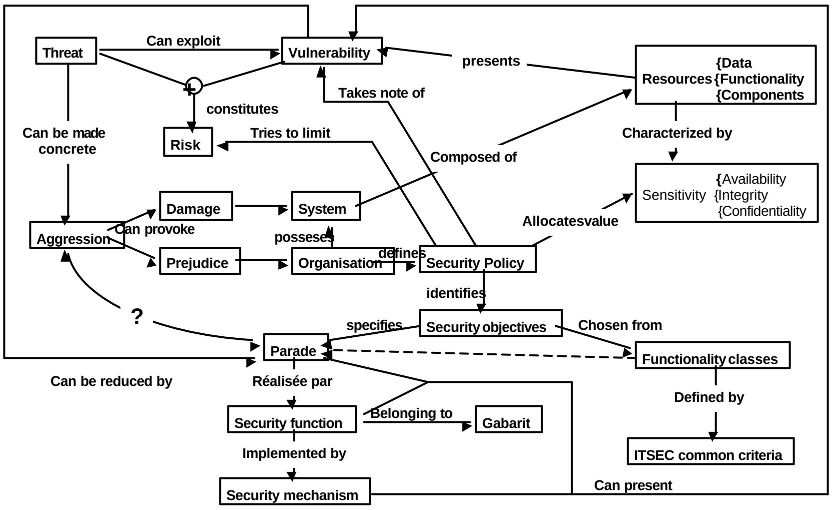


- **Examples of basic technologies used in Web Servers**



Also widely used : Script languages, Mobile code (Java), XML for data exchange, agents,....

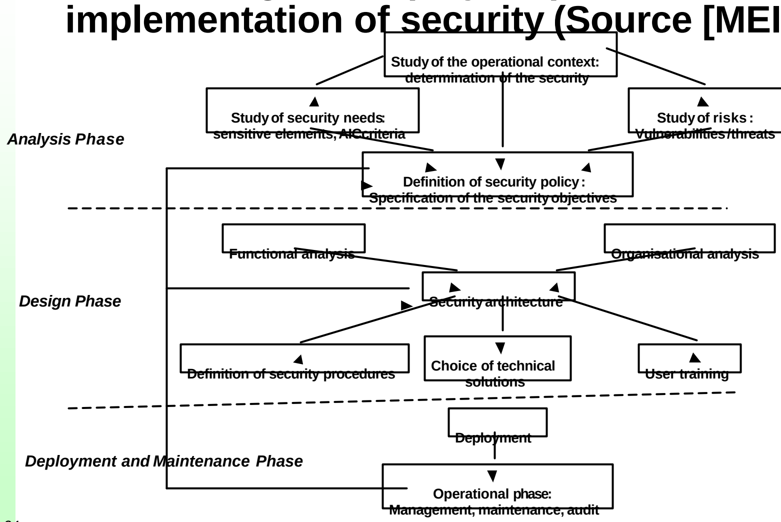
Security Concepts (Source [MEI98])



Page 33

© RJ Chevance

Methodological step by step implementation of security (Source [MEI98])

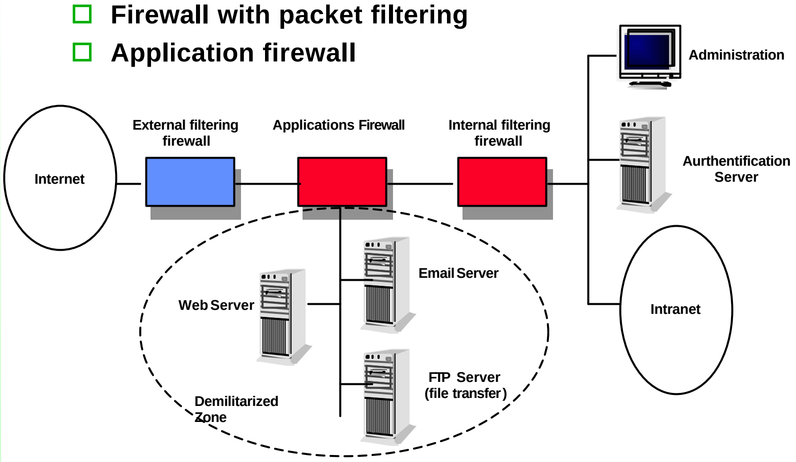


Page 34

© RJ Chevance

Security(3)

- Example of a system using the 2 firewalls techniques:
  - Firewall with packet filtering
  - Application firewall



Page 35

System Administration

- System administration covers:
  - Management of users and their rights with respect to resources and applications
  - Management of equipment, work groups and their rights
  - Administration of databases
  - Management of the equipment base and associated resources, software in particular, from the point of purchase to their destruction or sale
  - Incident management
  - Monitoring, optimization and automation of system usage, in particular for batch processing
  - Monitoring, optimization and automation of the use of networks
  - Management workstation (i.e, a single workstation to manage all systems, local and remote), definition of usage scenarios and reconfiguration (in general through the use of scripts)
  - Automatic management of backups and restores
  - Measurement of quality of service (QoS, publishing reports on QoS, improvement of QoS, etc.

Page 36

Economic Model

- **Scale Effects:**
  - Hardware: continuous price decrease due to to volume production (e.g. for microprocessors, below several millions of units, design cost redominates)
  - Software:
    - Manufacturing cost is almost « zero » (distribution via Internet, online documentation)
    - Design and development cost dominates:
      - e.g. for a $10M software component:
        - for a vendor of moderate volumes: never develop any software which is expected to sell less than 100,000 copies; given a price/development cost ratio of 10:1, this means the software must be sold for $1000.
        - for a high-volume vendor: never develop any software which is expected to sell less than 1,000,000 copies ; given a price/development cost ratio of 10:1, this means the software must be sold for $100

Economic Model(2)

- **Number of lines of code in various operating systems (Source: [MOO99])**

Operating system	Estimate of the number of lines of code (million of lines)
Windows 3.11	3
Windows 95	14
Windows 98	18
Windows NT 4.0	16.5
Windows 2000	35
OS/2	2
Netware 5.0	10
UNIX (average)	12
Linux	5 to 6 (still growing)
OS/400 (v4.r3)	40
MVS (OS 390 and extensions)	9-18

- **Comments:**
  - For some of these systems, the estimate integrates DBMS or user interface
  - Cost of man/year (US 2002): about $150K
  - Programmer's productivity: 2000 lines of code/year
  - So, $10M → 133,000 (new) lines of code

References

[GRA93] Jim Gray, Andreas Reuter « Transaction Processing: Concepts and, Techniques »
Morgan Kaufmann, San Mateo, 1993

[MEI98] Jean-Pierre Meinadier, Ingénierie et intégration des systèmes,
Hermès, 1998.

[MEI99] Jean-Pierre Meinadier, « Cours d'intégration des systèmes Client/-Serveur »,
CNAM, 1998-1999.

[MOO99] Fred Moore, *Storage Panorama 2000*,
StorageTech, *http://www.storagetech.com*.