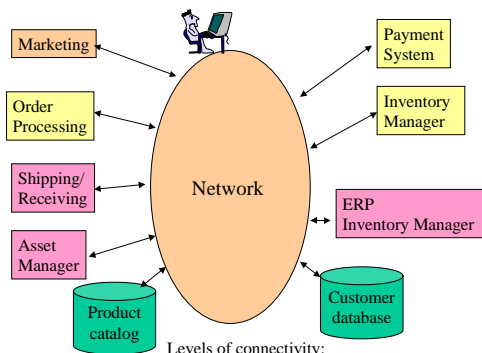


Application Interconnectivity and Middleware

- Application Interconnectivity
- Middleware characteristics
- Connecting remote applications - RPC
- Remote database access
- Message oriented
- Publish-subscribe

Amjad Umar

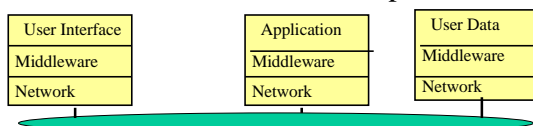
Application and Database Connectivity



Levels of connectivity:
•Network to transport information
•Middleware to connect above network

Copyright (A. Umar) 2002

Middleware Concepts



Definition: MIDDLEWARE is a set of common business-unaware services enabling applications and end users to interact with each other across a network.

It resides above the network and below the business-aware software.

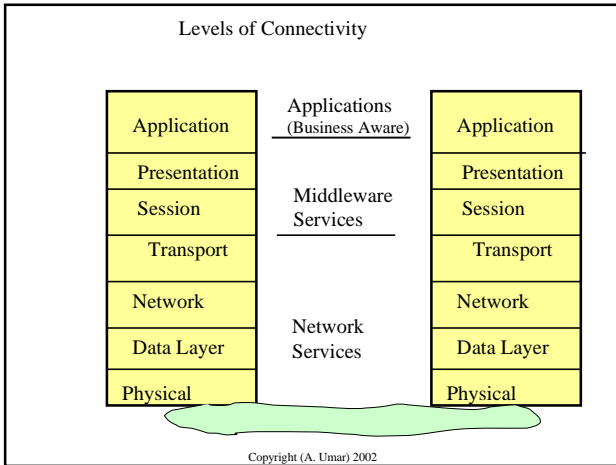
Examples: Email, Web, ODBC, EDI, distributed transaction processors, CORBA, DCOM etc

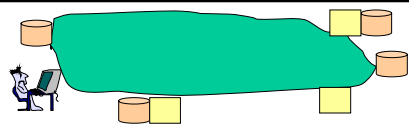
Important: Middleware typically supports client/server interactions across machines.

Example: a client uses middleware to access a machine for a catalog. However, some information is in master catalog.

Thus the middle machine may house a client that accesses a third machine

Copyright (A. Umar) 2002

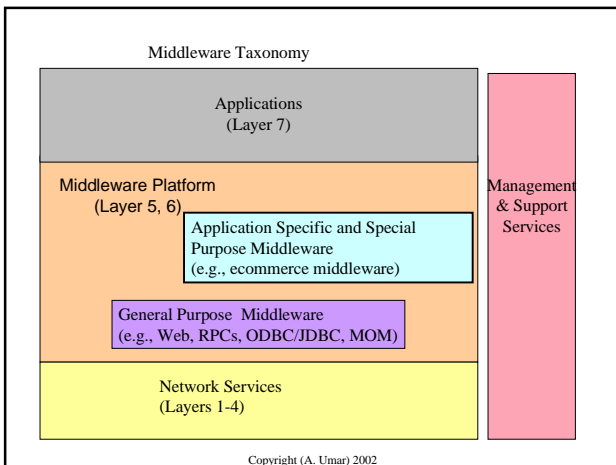


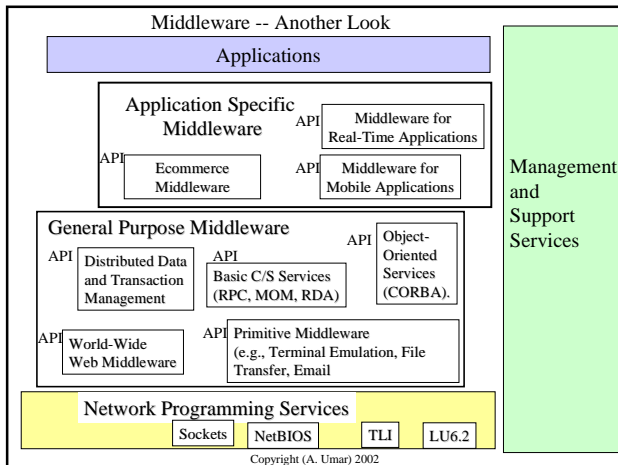


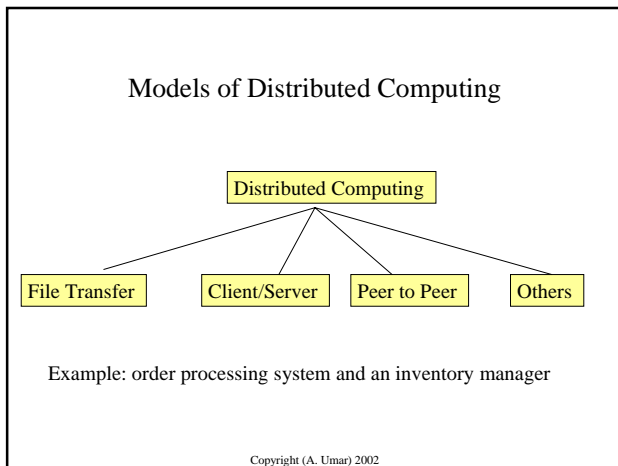
Middleware Services Overview

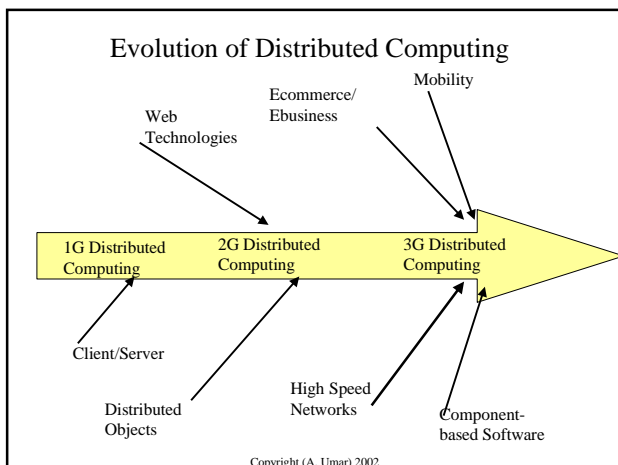
- Remote logon; extend local logon
- Remote file transfer; extend local file transfer
- Client/Server Services; extend local program and data access
 - Remote procedure call (RPC)
 - Message oriented middleware (MOM)
 - Remote database access (RDA)
 - Remote Presentation access (RPA)
- Web Middleware (Web Browsers, Web Servers, Web Gateways)
- Distributed Object Technologies (CORBA, DCOM)
- Distributed Transaction Processors (TP-Lite, TP-Heavy)

Copyright (A. Umar) 2002

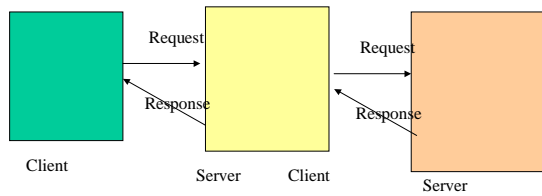








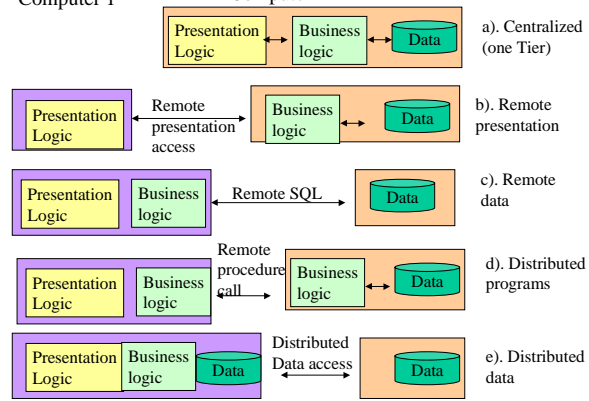
Conceptual Model of Client/Server



Copyright (A. Umar) 2002

Computer 1

Computer 2



Thin Client Versus Fat Client

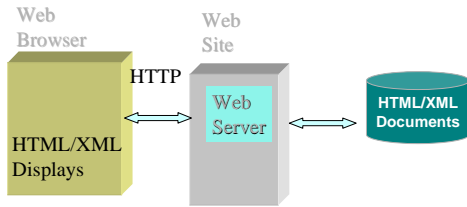
Copyright (A. Umar) 2002

Evaluating Architectures

	- Presentation on Server - Business logic on server - Data on Server	- Presentation on Client - Business logic on server - Data on Server	- Presentation on Client - Business logic on Client - Data on Server	- Presentation on Client - Business logic on Client - Data on Client
Performance				
Security				

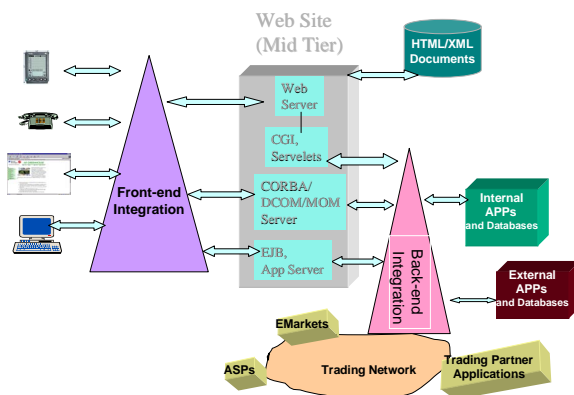
Copyright (A. Umar) 2002

Simple Application Architectures



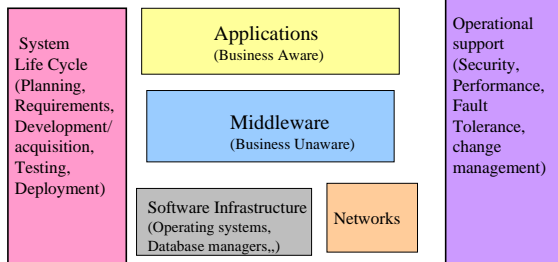
Copyright (A. Umar) 2002

Distributed Application Architectures

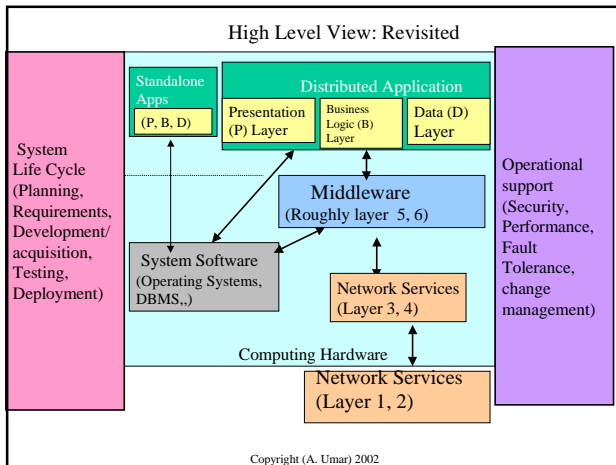


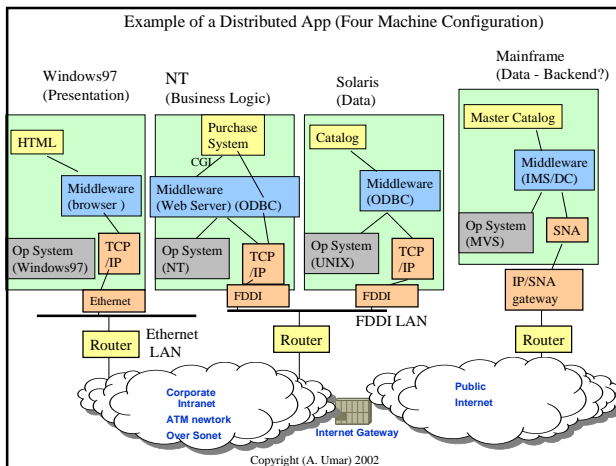
Copyright (A. Umar) 2002

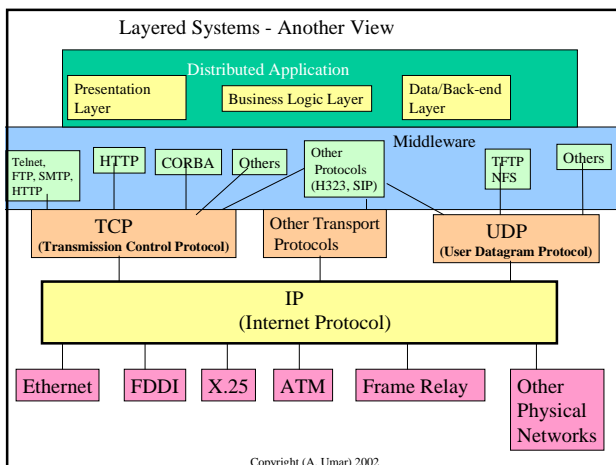
Key Building Blocks



Copyright (A. Umar) 2002

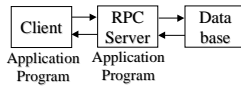




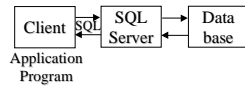


Basic Remote Interactions Services

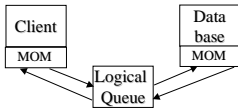
RPC (eg: HTTP, CORBA)



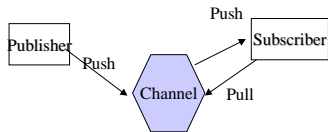
RDA (ODBC)



MOM (e.g., IBM's MQ)

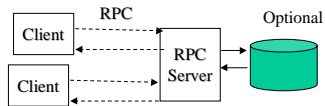


Publish/Subscribe (e.g., JMS)



Copyright (A. Umar) 2002

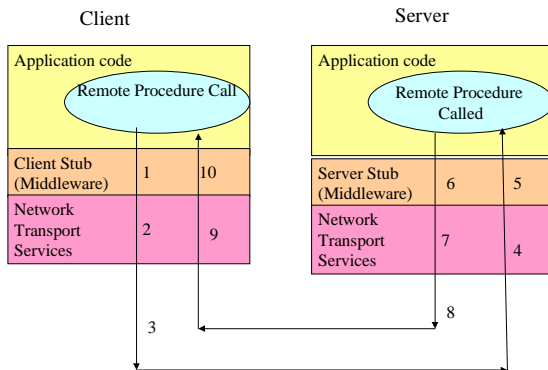
Remote Procedure Call (RPC)



- Supports apps distributed across multiple machines.
- Uses IDL to specify API
- Does not support improvised queries.
- Two types:
 - Asynchronous RPC
 - Highly scalable.
 - Synchronous RPC
 - Best recoverability of all C/S services.

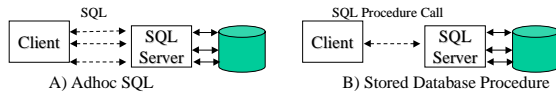
Copyright (A. Umar) 2002

RPC Flow



Copyright (A. Umar) 2002

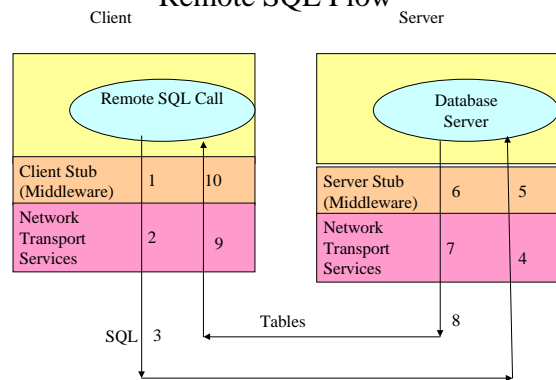
Remote Data Access (RDA)



- Client programs and/or end-user tools issue improvised queries, usually in SQL and without the need for prestructuring of remote databases.
- Transactions are usually synchronous.
- RDAs can support virtual databases through a distributed-query processor, allowing joins among remotely located tables.
- RDAs have wide support by DMBS vendors.

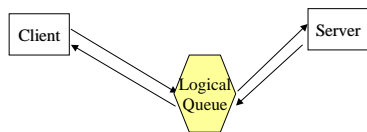
Copyright (A. Umar) 2002

Remote SQL Flow



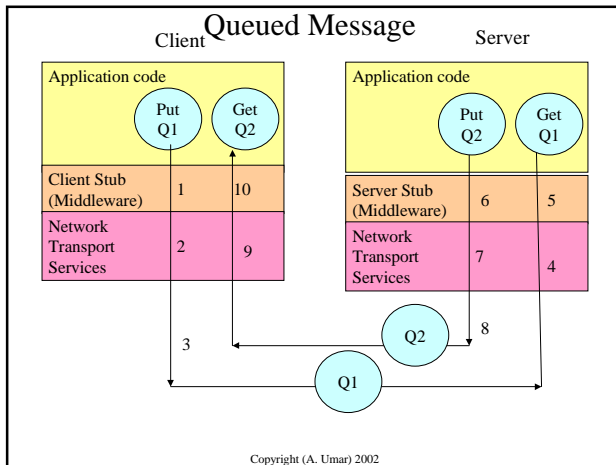
Copyright (A. Umar) 2002

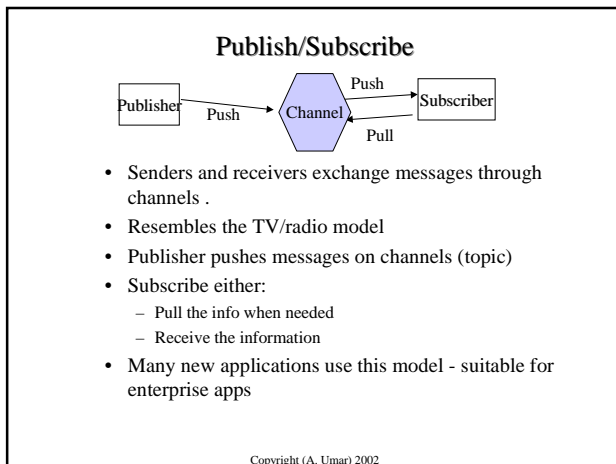
Message-Oriented Middleware (MOM)

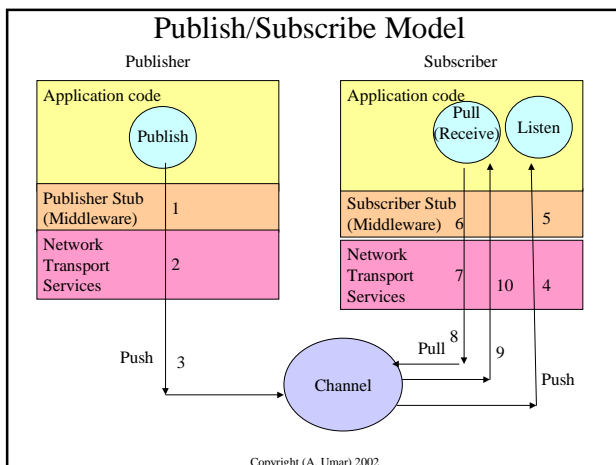


- Senders and receivers exchange messages through queues.
- Example: Queued Message Processing (QMP).
- Supports guaranteed message delivery between client and database.
- Asynchronous.
- Some recovery through queue if system fails.

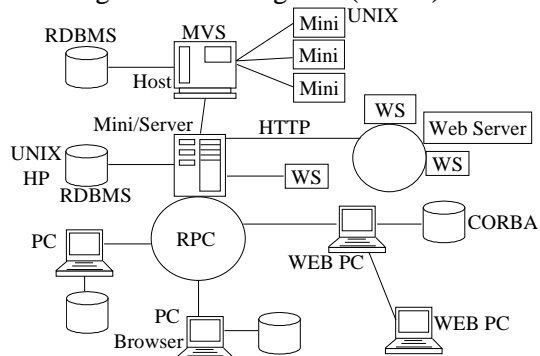
Copyright (A. Umar) 2002





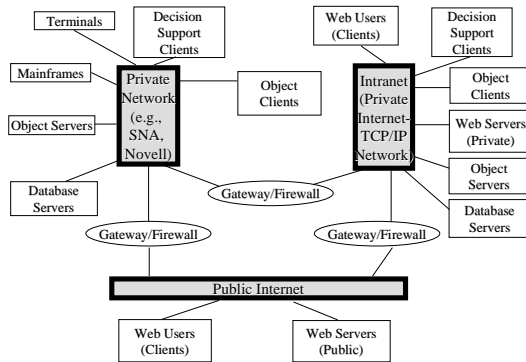


Putting The Pieces Together (View1)



Copyright (A. Umar) 2002

Putting The Pieces Together (View2)



Copyright (A. Umar) 2002

Evaluating Middleware Services

- Synchronous or Asynchronous?
- Loose versus tight coupling
- Scalability
- Recoverability
- Cost

Copyright (A. Umar) 2002
