Middleware

2009년 1학기

yiyoon@sm.ac.kr

011-451-0388

숙명여자대학교 멀티미디어과학과

Table of Contents

- **Introduction**

- **DB Connectivity Products**
- **Remote Procedure Calls**
- **Message-Oriented Middleware**
- **Object Request Brokers**
- **DCE middleware**
- **Distributed Transaction-Processing Monitors**

Background (1)

- The early 1980s
  - Emphasize parallel and interconnection architecture
- The late 1980s
  - Connecting heterogeneous platforms had become a priority business
    - based on workstation and mainframe
- The 1990s
  - Increasing Desktop PC
    - based on PC

Background (2)

- Cooperate (Collaborative) Systems
  - no longer operate on one computer
  - no longer support one network protocol
  - no longer access one data type

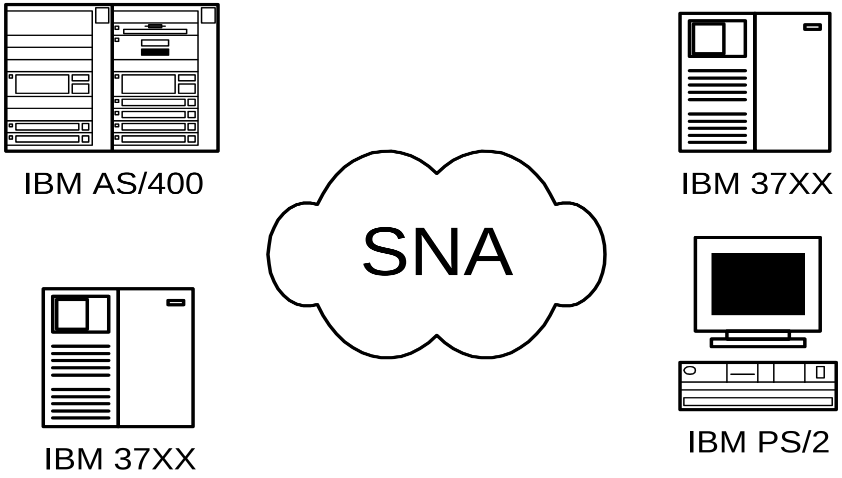
  ⇒ having functions on a variety of computers
  connecting via different protocols
  storing information in heterogeneous

- Vendors
  - designed for its HW, SW, Network Equipment

  ⇒ cannot support heterogeneous environment

Background (3)

- Example
  - IBM Mainframe Application
    - connected via IBM’s System Network Architecture protocol
    - basically developed by using OS/2 and DB/2



Background (4)

- Programmers
  - tailor applications to each OS, DB, and Network Protocol
  - depend on the variety of options required

  ⇒ Inefficient in Timing and Cost

- SW vendors
  - develops **<u>Middleware</u>**
    - allows Application Programming Environments for the differences between computer components

  → Considers Domain, Service, Contents

Background (4)

- Middleware
  - enables companies to build modular applications

- Programmers
  - can typically write the application logic separately from the middleware layer by using middleware APIs

⇒ significantly simplifying application design

- no longer directly access of OS services, network protocols, DBMSs
- works with one tools that provides links to different vendors’ products

Background (5)

- Middleware
  - the key to Distributed Computing
    - role it performs in developing distributed software
      - a large information pipe

- Six Types of Middleware
  - DB Connectivity Products
  - Remote Procedure Calls
  - Message-Oriented Middleware
  - Object Request Brokers
  - DCE middleware
  - Distributed Transaction-Processing Monitors

Key Questions (Important Subjects)

- What is middleware?
- What benefits does it offer?
- What are the main types of product available?
- How do they work?
- What are their advantages and disadvantages?
- What are the trends in product development?
- What is the size of the middleware market?

What is middleware (1)

- Middleware
  - one of the most confusing and complex subjects in IT
    - vendors offering a variety of products
      - » sometimes complementary, sometimes competing
  - need to sweep away this confusion,
  - provide a clear understanding of different types of middleware
  - indicating when it is appropriate to use them

- **Obectives:**
  - performs a distinct and specific role
  - has well defined functionality
  - is a strategically important product to practically every IT department

What is middleware (2)

- **Definition**
  - off-the-shelf connectivity software
    - supports **distributed processing** at run-time and
    - is used by developers **to build distributed software**
  - the specific purpose of the software
    - must be to enable distributed computing by providing **connection services**
      - » GUI builders are nor classified as middleware, because their prime purpose is to enable a developer to build applications

⇒ prime purpose: connectivity

What is middleware (3)

- Example
  - – the ‘/’ in the middle of client/server
    - confirms the position and significance of middleware
      - » the necessary link between client and server
    - indicates why so many underestimate its importance
  - – Applications in C/S
    - separately developed on client, on server, on different platforms

    ⇒ **middleware is ‘bit in the middle’**

    ⇒ **‘/’ is silent**

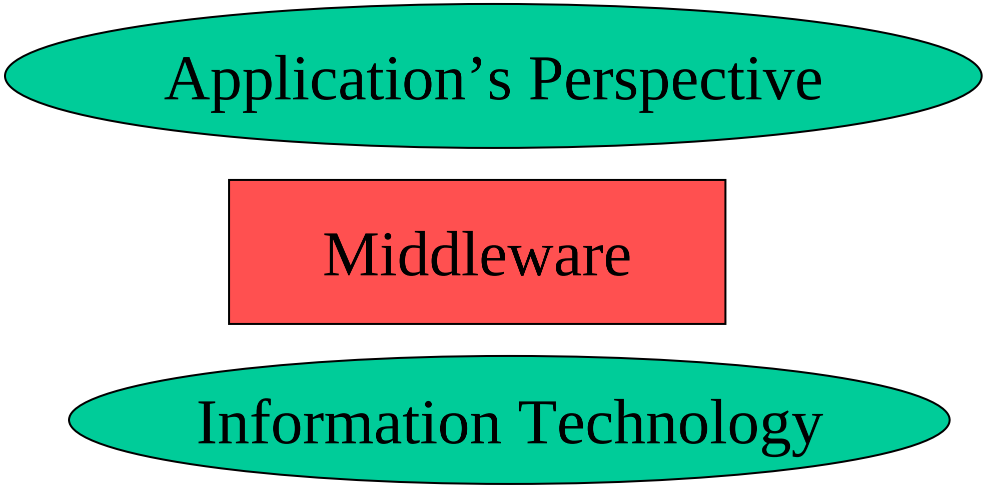
What is middleware (4)

- The key to distributed computing
  - Middleware will become a key software technology
  - Its impact will be similar to the introduction of DBMSs
    - » strategic and standard
  - It will glue together heterogeneous architectures and deliver considerable benefits

- Model of middleware
  - to achieve the above problems
  - enables to distinguish between generic middleware services and the specific functionality of particular products
    - » compare and contrast the six different types

Status of Middleware (1)

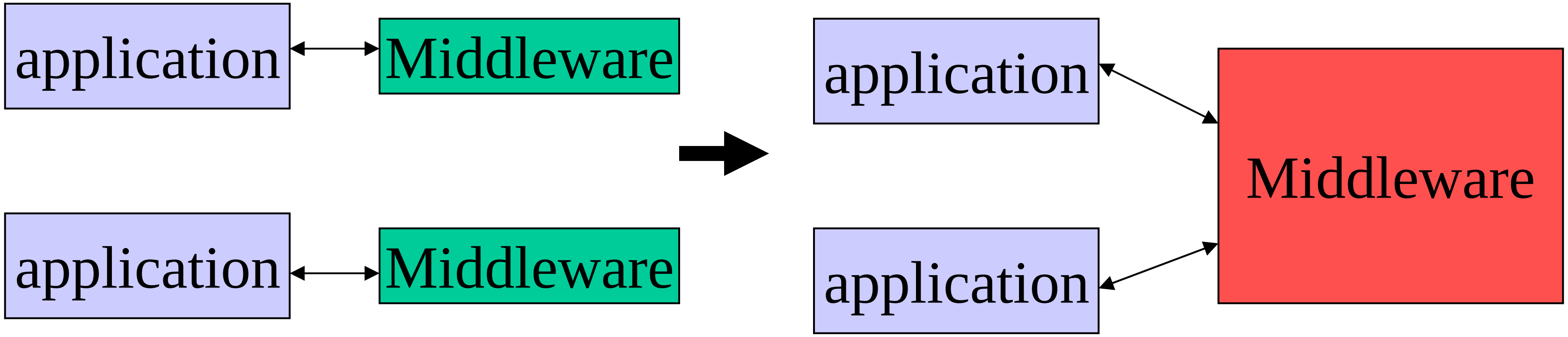
- Middleware
  - the bridge between the underlying technology and the application



- Value of Middleware
  - Closely to the availability of applications
- Successful middleware products
  - Industry standard interfaces
    - » ODBC, OLE, API, etc

Status of Middleware (2)

- Direction of Middleware
  - application-centric
  - a quality application-oriented product
- Function of Middleware
  - a single shared service
    - » enables large numbers of application



Driving Trends of Middleware

- Driving Trends
  - two of the key trends for distributed computing
    - implementing reliable systems
    - providing users with direct access to information without the need for programmer or IT intervention
  - no a single technology, product or solution
    - a class of software and solution
- Different types of middleware
  - one based on messaging activity
  - the other based on database activity, especially for DB access

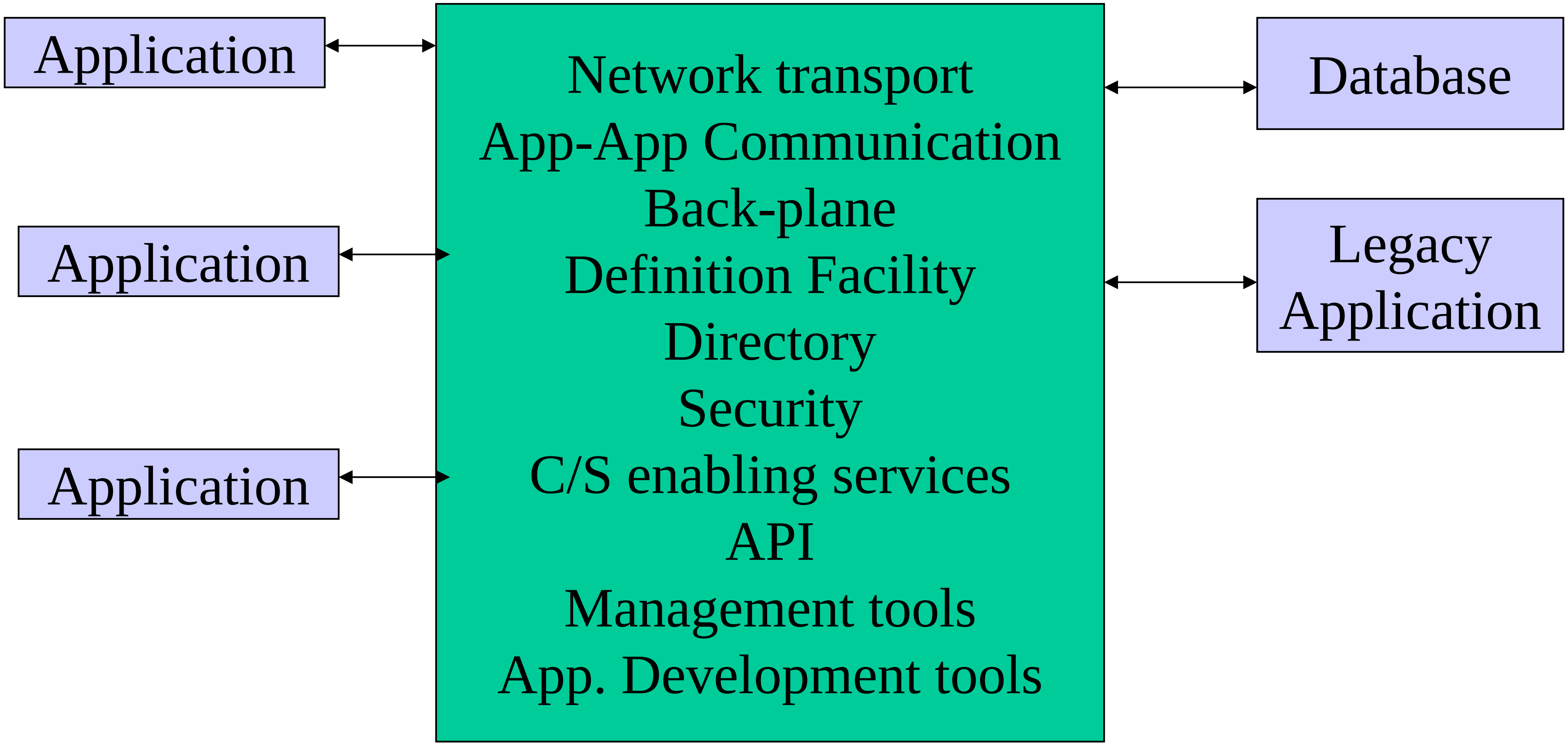
What middlware needs to offer

- Standard
  - apply technology and take into industry standards
  - OLE, ORBs, etc
- Simple
  - abstract the complexity of an underlying service technology
- Complete
  - provide that functionality which is relevant to current requirements
- Efficient
  - highly specialized tool sets to maximize application performance
- Middleware provides technology to application development by way of *standard*, *simple*, *complete*, *and efficient* software interface.

Basic Requirements for Middleware

- Common application interface
- Scalability and Flexibility
  - must be protected from interface function changes which reflect evolution of the technology
- Encapsulation of services
  - must abstract a complex service for the requesting application
- Individual Exception Processing
- Reporting and Diagnostic Capabilities

Typical services provided by Middleware



Application

Application

Application

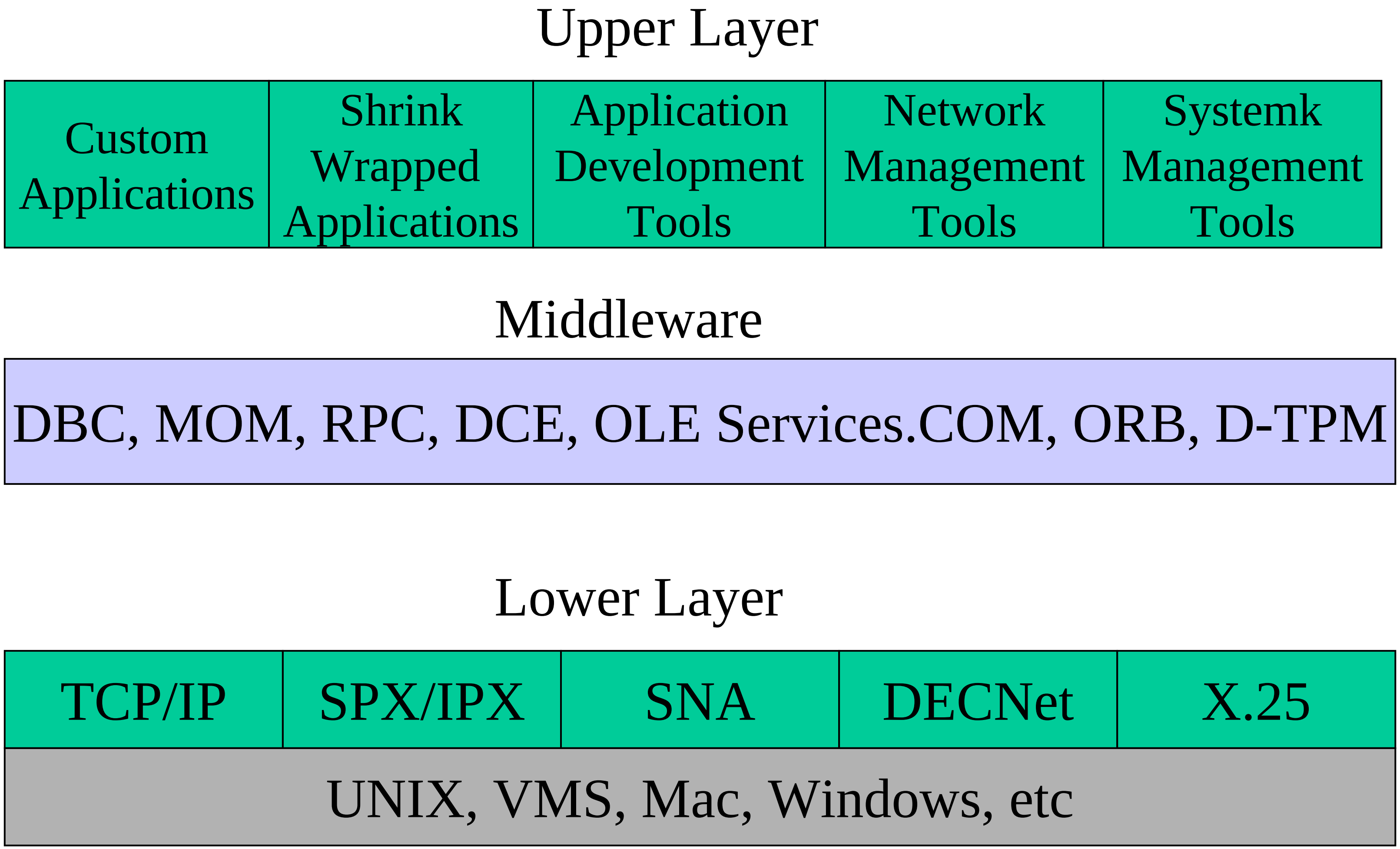
Network transport
App-App Communication
Back-plane
Definition Facility
Directory
Security
C/S enabling services
API
Management tools
App. Development tools

Database

Legacy Application

Middleware protects applications from Service Provider changes

Where Middleware Fits



Upper Layer

Custom Applications

Shrink Wrapped Applications

Application Development Tools

Network Management Tools

Systemk Management Tools

Middleware

DBC, MOM, RPC, DCE, OLE Services.COM, ORB, D-TPM

Lower Layer

TCP/IP

SPX/IPX

SNA

DECNet

X.25

UNIX, VMS, Mac, Windows, etc

Distributed Middleware Melting Pot

- Distributed Transaction Processing
  - » TPM
- Multi-platform workflow
  - » DCE
- Heterogeneous database access
  - » ODBC
- Asynchronous application execution
  - » Message Queue
- Multifaceted object request brokers
  - » ORB
- Multiple parallel communication modes
  - » RPC
- Mobility

- Combinations of two or more of the above

Why do you need middleware?

Why do you need middleware? (1)

- Coping with heterogeneity
  - need a legacy of systems built on different HW and SW
  - will continue to add new machines and new SW
    - purchasing power is distributed
    - reduced costs and improved functionality are sought
    - winners and losers can change
  - must find better ways dealing with heterogeneity

  ⇒ **Middleware** helps scope with heterogeneity by:
  - insulating against complexity
  - insulating against change
  - providing control over the environment

- Complexity
  - The number of APIs used to access SW is already large and is growing
    - A standard restricts the ability to innovate and beat the competition.

  ⇒ **Middleware** hides complexity

Why do you need middleware? (2)

- Change
  - a company may be changing the residency of its data, SW, HW, Network
    - a large amount of time wasted
    - the potential for making mistakes

  ⇒ **Middleware** is specifically designed to make the system more adaptable to change

- Control
  - in heterogenous environment,
    - the potential for message corruption, loss and error increases

  ⇒Middleware provides control over the environment
    - checks are made that messages are not lost
    - if there are errors they are trapped and handled correctly
    - the security of message is ensured

  ⇒ **Middleware** makes sure that inter-software communication is handled correctly, securely, efficiently and effectively⇒

Key messages to users

- Prepare the ground
  - Middleware cannot and should not be used to solve the problems of poorly structured data or outdated applications
  - Middleware cannot turn a poor system into a better one

  ⇒You have to to that yourself : by a strategy study for your system/envri.

- Appoint a technical architect
- Develop a technical architecture
- Implement and test the architecture you have designed
- Evaluate the options
- Development tools
- Manage the risks

Key messages to suppliers (1)

- Make the message clearer
- Eradicate the services overlap
- add more network transparency
- Provide architecture-visualization tools
- Add ease-of-use features
- Manage expectations
- Test, test and test again
- Increase your customer-support team

Middleware(1)

- Category of Middleware
  - Transport Stacks Middleware
    - TCP/IP, NetBIOS, IPX/SPX, DECnet, OSI, SNA/APPN, …
  - Network Operating System Middleware
    - NetWare, OSF/DCE, …
  - Server Specific Middleware
    - Database Middleware: ODBC, DRDA, Oracle Glue, CLI, ...
    - Transactional RPC Middleware: Tuxedo ATMI, X/Open TxRPC, IBM CICS, ...
    - Groupware Middleware: MAPI, VIM, Lotus Notes API, ...
    - Object Middleware: OMG CORBA, ODMG-93 API, ...
    - Distributed System Management Middleware: SNMP, CMIP

Middleware(2)

- Client/Server를 위한 통신 Middleware
  - Computer Communication Protocol Overview
  - OSI 7 Layer Reference Model
  - Transport Middleware Stack
    - INTERNET
    - SNA
    - IPX/SPX
    - XNS
  - RPC(Remote Procedure Call)/MOM(Message-oriented-middleware)
    - RPC : high level and synchronous on TCP/IP only
    - MOM : low level and asynchronous on TCP/IP, DECnet, SNA, and IPX