CPE 545

Course - Introduction

What is Communications Software?

- Software that facilitates communication between remote systems to exchange information in various formats: text, audio, video.

- What is embedded communications software?
  - When the remote systems are embedded devices like cell phones, watches, routers, switches, MP3 players…

- Communication systems consist of devices that have a wide range in complexity
  - Software design is determined by the function each kind of device performs in the network

Communication Software Strategy

- Issues common to communication equipment
  - Protocols and standards
    - International Organization for Standards (ISO), Internet Engineering Task Force (IETF), 3rd Generation Partnership Project (3GPP)
  - Code size
  - Real-time processing
- Issues due to increasingly more complex next generation smart devices
  - Running tasks on multiple processors
  - Inter Process Communication (IPC)
  - Reducing development cycle
  - Micro-kernels
  - Middleware distributed software (RPC, MOM)

OSI Layers

- Key Observations
  - 7 layers
  - Each of them can be implemented in hardware or software
  - Lower layers are mostly implemented in hardware while the higher layers are mostly in software
  - All 7 are not required to be implemented to execute a system

Physical Layer

- Provides hardware means to transmit/receive data
  - Electrical specifications
  - Functional and procedural specifications
  - A/D conversion
  - Modulations/Demodulations
  - Handles the physically adjacent nodes (point-to-point)
  - All in hardware. Examples:
    - RS232, RS422
    - RF Modules and Modems in wireless
    - PHY Ethernet Controllers

Data Link Layer

- Provides for more reliable transmit/receive of data between peers
  - Data Integrity through error detection and correction
    - (Cyclic Redundancy Check) CRC
    - Collision Detection and Avoidance
    - Viterbi or Turbo decoding
  - Both Hardware and Software
  - MAC (Media Access Layer) layer
    - Provides addressing and channel access control
  - LLC (Logical Link Control) layer
    - Provides Multiplexing, flow and error control
  - Examples:
    - ISDN, ATM, Ethernet MAC

Network Layer

- Responsible for delivery of packets from source to destination
  - Allows for independent link layer end points
  - Provides network addressing to route the data packets
  - Routers are used to forward packets from node to node
  - Routing protocols are usually implemented in software

Transport Layer

- Provides network independent end-to-end integrity between the end points
- Transmission Control Protocol (TCP) - connection oriented
  - Connection established prior to data transmission
  - Ack sent upon receipt of data providing reliability, ordering and data integrity
- User Datagram Protocol (UDP) - connectionless
  - Assumes that error checking and correction is either not necessary or performed by higher layers
- Usually implemented in software except when to support large number of connections.
- TCP Offload Engine
  - As name suggests offloads TCP/IP stack processing to a dedicated hardware unit, Network Interface Cards
  - Hard to integrate into computing systems, requiring extensive changes in the operating system

Communication Devices

- **Repeaters to regenerate signal (Physical Layer)**
- **Layer 2 switches (Data Link Layer)**
  - Switches MAC frames between LAN segments on the same subnet based on their dest address
  - Monitors traffic and learns about the MAC address of the nodes on its ports
- **Layer 3 switches or Routers (Network Layer)**
  - Forwards IP frames across LAN segments and WANs based on their dest address
  - Hosts are not on the same sub-net
  - If dest host is not directly connected, it forwards the packet to another router (routing tables)
  - Routing tables gets updated by neighboring routers according to the routing protocols
    - Routing Information Protocol (RIP)
    - Open Shortest Path First (OSPF)
    - Border Gateway Protocol (BGP)

Communication Eco-system

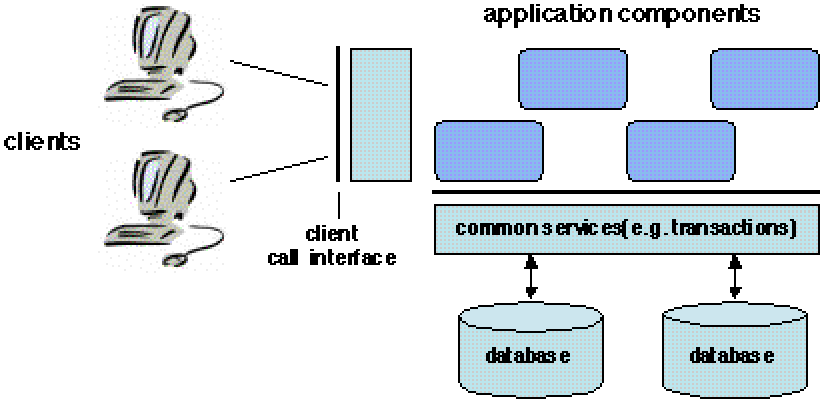
- **Communication devices can refer to any kind of device that facilitates communication**
  - These devices transmit data between different computers or computing devices
  - Each device plays a specific role and implements specific communication functions
- **Protocol and Systems Software**
  - Architected for performance and memory efficiency
- **Communication Companies**
  - Electronic Design Automation (EDA) tool vendors (Altera, Xilinx)
  - Semiconductor component vendors (ASIC design - fabless)
  - RTOS, tools and software vendors
  - Contract Manufacturers (CMs)
  - Equipment Manufacturers (EMs)
  - Home, Enterprise and service-provider users

What is Middleware?

- Middleware’s multiple definitions.
  - Software that mediates between an application program and a network.
  - Software that contains a set of services to allow multiple processes running on one or more computers to interact across a network
  - General term for any programming that serves to glue together separate, already existing applications

Motivation for Middleware

- Component Based Software
  - Applications composed of software components
    - Client user interface (presentation layer)
    - Information management (Database)
    - Application specific functionality (business logic)



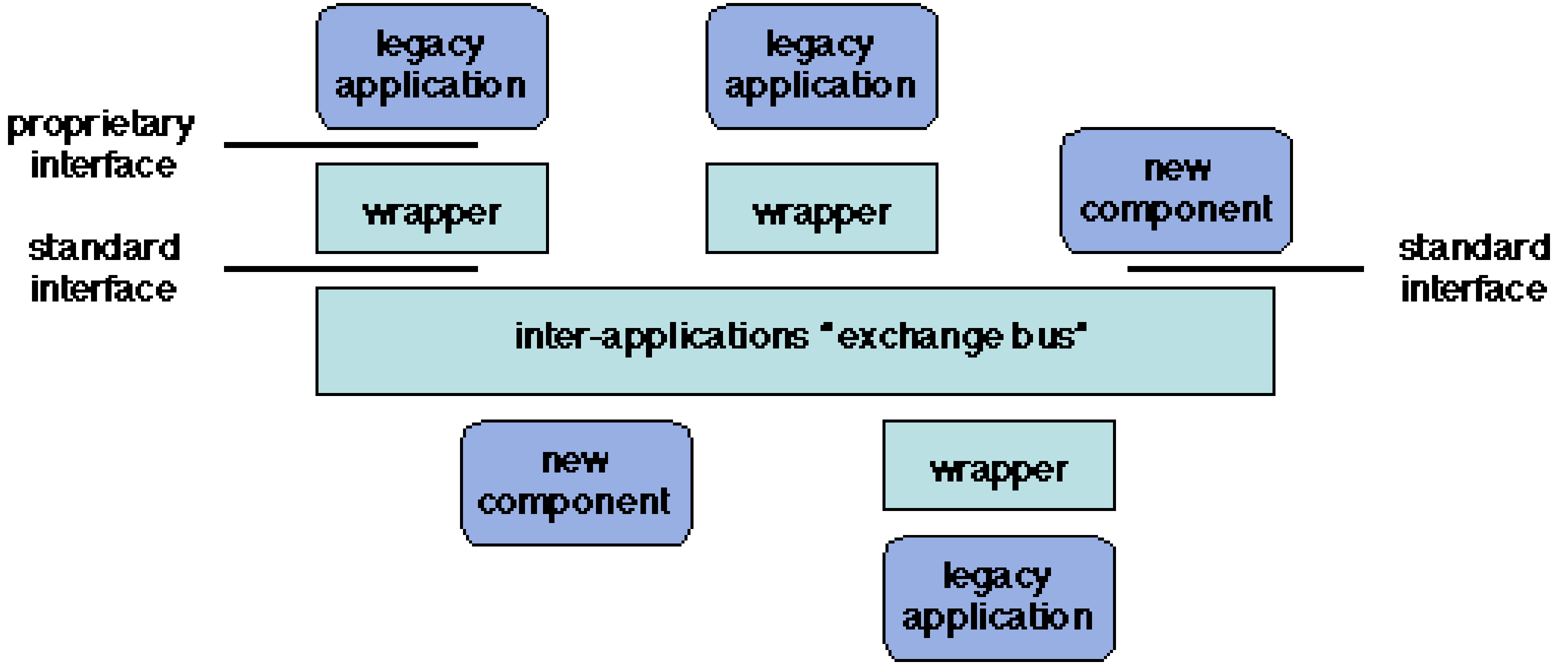
Motivation for Middleware

- Applications use an intermediate software (middleware) that resides on top of the operating systems and communication protocols to perform the following functions:
  - Hiding *distribution*.
  - Hiding the *heterogeneity* of the various hardware components, operating systems and communication protocols that are used by the different parts of an application.
  - Providing uniform, standard, high-level *interfaces* to the application developers and integrators, so that applications can easily interoperate and be reused, ported, and composed.
  - Supplying a set of common *services* to perform various general purpose functions, in order to avoid duplicating efforts and to facilitate collaboration between applications.

Motivation for Middleware

- Reusing Legacy Software
  - Adopt a common language -- independent standard to interconnect with other applications
  - Define the interface and interchange protocols implemented in a software acting as exchange-bus or broker
  - User wrapper to bridge the legacy interface with the new interface

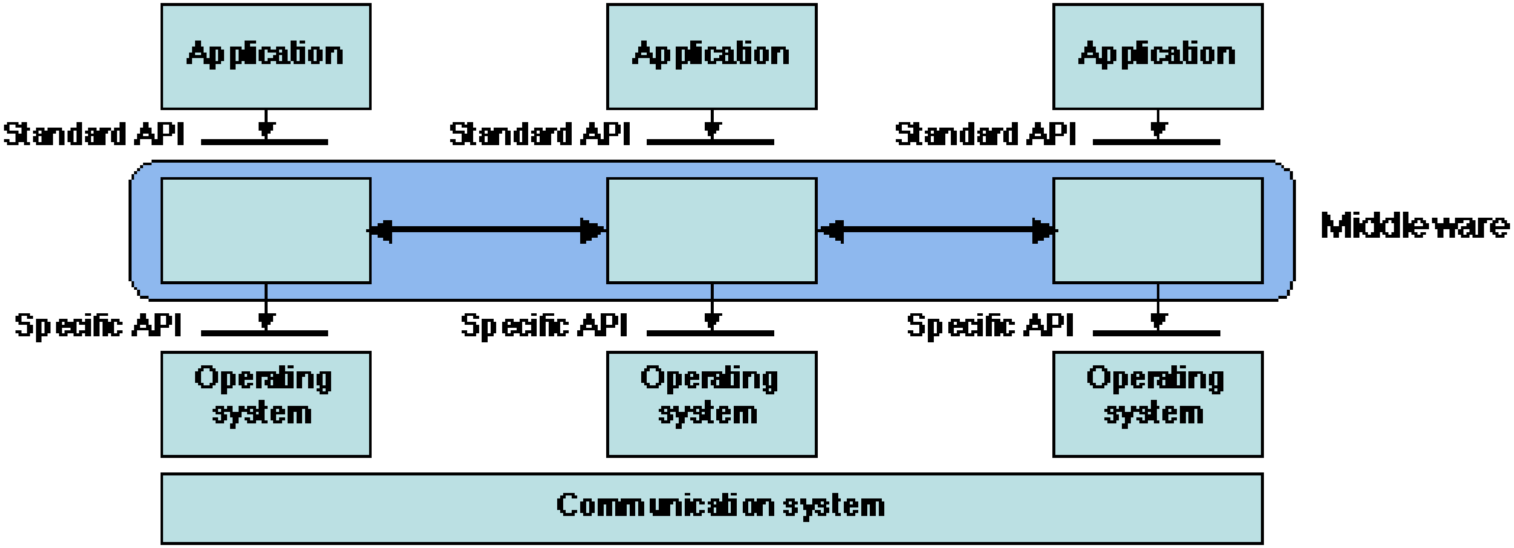
Motivation for Middleware



Motivation for Middleware

- The middleware architecture of a system can be designed according to the following requirements and motivations:
  - *Managed entities*. Middleware systems manage different kinds of entities which differ by their definition, properties, and modes of communication (e.g. objects, agents, and components).
  - *Service provision structure*. The entities managed by a middleware system may have predefined roles such as:
    - *client* (service requester) and *server* (service provider),
    - *publisher* (information supplier) and *subscriber* (information receiver).
    - *peer to peer* - all entities are at the same level and a given entity may indifferently assume different roles
  - *Service provision interfaces*. Communication primitives provided by a middleware system may follow the synchronous (blocking) or asynchronous (non-blocking) paradigm.

Motivation for Middleware


Application
Application
Application
Standard API
Standard API
Standard API
Middleware
Specific API
Specific API
Specific API
Operating system
Operating system
Operating system
Communication system