Mobile Middleware Course

Introduction and Overview

Sasu Tarkoma

Contents

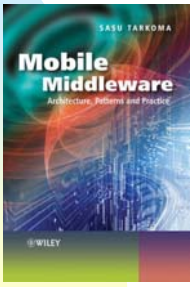
- Lecture outline
- Motivation
- Mobile middleware overview
- Examples
- Summary

Lecture Outline

- 13.3. 16-18 Introduction and overview.
- 17.3. Mobile platforms.
- 24.3. Principles and patterns. Messaging and pub/sub.
- 31.3. Principles and patterns continued.
- 7.4. Interoperability and standards.
- 14.4. (Easter holiday)
- 21.4. Applications and service case studies.

- Exam: 16-19 CK112

Course Book



- Mobile Middleware – Architecture, Patterns, and Practice published by Wiley
  - Publication date 27.3.2009

Motivation

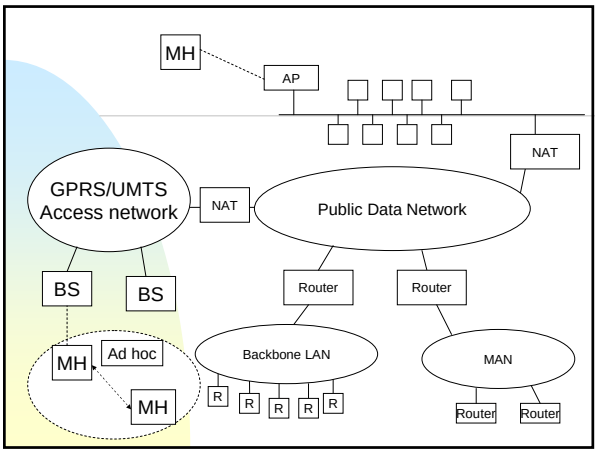
- Mobile computing has become one of the breakthrough technologies of today
  - Over 3.3 billion mobile phones in use, 4 billion SIM cards
- Current trend is converged communications
  - Web resources integrate seamlessly with mobile systems
  - Mobile systems are increasingly dependent on software
- The course provides a comprehensive overview of mobile middleware technology

Mobile Evolution

- 1st generation (1990-1999)
  - Text messages (SMS) and mobile data. Speeds up to tens of Kbps.
- 2nd generation (1999-2003)
  - Limited browsers, WAP, iMode, and MMS. Speeds up to 144Kbps.
- 3rd generation (2003-2008)
  - Mobile platforms, middleware services. Series 60, J2ME, Android, iPhone. Speeds up to several Mbps.
- 4th generation (2008-)
  - Adaptive services, user interfaces, and protocols. Context-awareness, always-on connectivity. Speeds up to hundreds of Mbps.

Wireless Technologies

- Global System for Mobile (GSM),
- General Packet Radio Service (GPRS)
- Universal Mobile Telecommunications System (UMTS)
- Long Term Evolution (LTE)
- Wireless LAN (WLAN)
- Worldwide Interoperability for Microwave Access (WiMax)
- Ultra-wideband (UWB)
- Wireless Personal Area Network (WPAN)
- Bluetooth, Wibree
- RFID



Mobility in the Internet

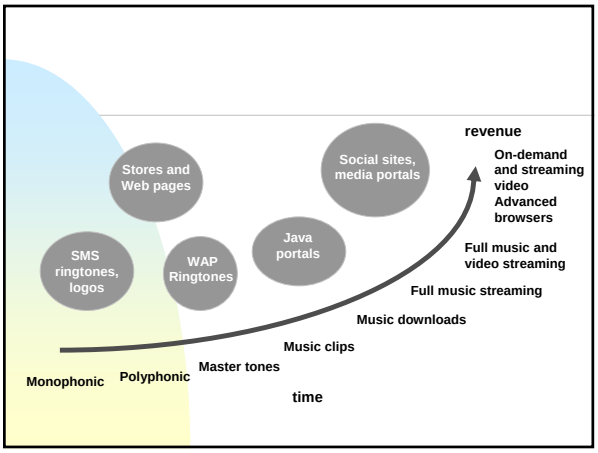
- This topic pertains to mobility of
  - Networks
  - Hosts
  - Transport connections
  - Sessions
  - Objects (passive, active)
  - Services
  - Users
- Many solutions are needed on multiple layers
  - Link layer, network, transport, application

Users of Middleware

- **End user.** The goal of middleware is not to directly interact with the end users, but rather support the applications and services that are visible to the users. This means that middleware should provide sufficient APIs and mechanisms to cope with different kinds of failures and faults, and in general support enhanced usage experience.
- **Device Manufacturers.** Device manufacturers use middleware in order to provide extended features that interface with device drivers.
- **Internet Service Providers.** Internet service providers utilize middleware to monitor and administer the network.
- **Platform Providers.** Platform providers develop middleware platforms that integrate with different operating systems.
- **Application Service Providers.** Application service providers utilize middleware in order to facilitate application development and deployment in a scalable and secure manner.

Key Elements

- Accessibility
  - Resources are available and accessible for end users irrespective of the current location or where resources are located
- Reachability
  - Resources should be available in any location. Reachability cannot be taken for granted in today's dynamic environment
- Adaptability
  - The environment is subject to changes. Mobile service usage needs to adapt to the operating environment.
- Trustworthiness
  - The various entities in the environment need to have certain level of trust that operations are carried out according to expectations (and contracts).
- Universality
  - Universal data access is one of the key reasons for the success of the Internet. This is a key element for the success of the mobile service ecosystem



Middleware

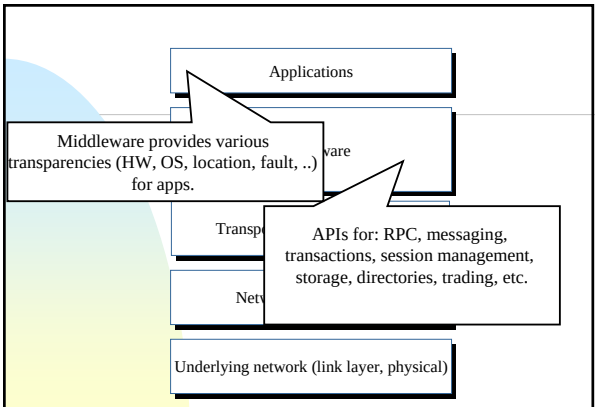
- Widely used and popular term
- Fuzzy term
- One definition
  - "A set of service elements above the operating system and the communications stack"
- Second definition
  - "Software that provides a programming model above the basic building blocks of processes and message passing" (Colouris, Dollimore, Kindberg, 2001)

Why Middleware?

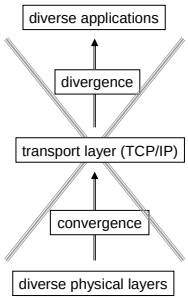
- Application development is complex and time-consuming
  - Should every developer code their own protocols for directories, transactions, ..?
  - How to cope with heterogeneous environments?
    - Networks, operating systems, hardware, programming languages
- Middleware is needed
  - To cut down development time
    - Rapid application development
  - Simplify the development of applications
  - Support heterogeneous environments and mask differences in OS/languages/hardware

Middleware cont.

- Middleware services include
  - directory, trading, brokering
  - remote invocation (RPC) facilities
  - transactions
  - persistent repositories
  - location and failure transparency
  - messaging
  - Security
- Network stack (transport and below) is not part of middleware



The Hourglass

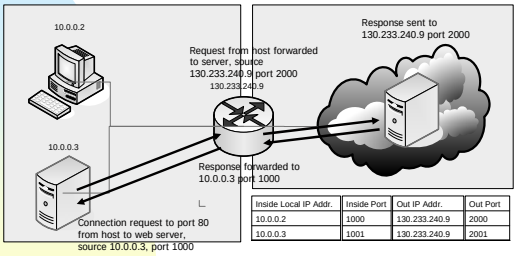


Examples

- Remote Procedure Call (RPC)
  - call of a remote procedure as it were local
  - marshalling / unmarshalling
- Remote Method Invocation (RMI)
  - call of a remote method as it were local
  - marshalling / unmarshalling
- Event-based computing
  - entities receive asynchronous notifications
  - a notification causes a state change
- Overlays and P2P content delivery

Transparencies

- Location transparency
  - RPC and RMI used without knowledge of the location of the invoked procedure / object
- Transport protocol transparency
  - RPC may be implemented using any transport protocol
- Transparency of OS and hardware
  - RPC/RMI uses external data representation
  - Presentation is important
  - XML is becoming increasingly important
- Transparency of programming languages
  - language independent definition of procedures: CORBA IDL, WSDL



Inside Local IP Addr.	Inside Port	Out IP Addr.	Out Port
10.0.0.2	1000	130.233.240.9	2000
10.0.0.3	1001	130.233.240.9	2001

Mobile Middleware I

- Middleware is typically designed and implemented for fixed-network hosts
  - High bandwidth, low latency, reliable communication
  - Persistent storage, sufficient computing power, power supply
  - No mobility
- Mobile environment requires new solutions
  - Existing middleware services do not scale
  - Small devices and embedded systems pose different challenges

Mobile Middleware II

- Goals for middleware:
  - fault-tolerance, adaptability, heterogeneity,scalability, resource sharing
- Mobile middleware
  - dynamically changing context
  - Decoupled in space and time
    - Asynchronous events, tuple spaces
  - Basic solution for wireless
    - Use a proxy

Reflection

- In general, middleware provides transparency to the underlying conditions
  - Location transparency in RPC/RMI
- In mobile environments this is not always desired
  - Sometimes we need to know about location changes, QoS changes, etc.
- Requirement for reflective middleware
  - Should expose APIs for querying / changing important parameters
  - Under development

Introduction to Platforms

- Mobile middleware aims to support the development, deployment, and execution of distributed applications in the heterogeneous and dynamic mobile environment.
- The goals for mobile middleware include adaptability support, fault-tolerance, heterogeneity, scalability, and context-awareness.
- The industry solution to these challenges has been to create middleware *platforms*.
- A platform collects frequently used services and APIs under a coherent unified framework.

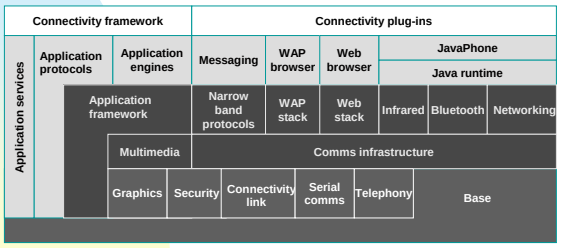
Platforms continued

- A service platform is the realization of a service architecture following its principles and patterns.
- A Service Delivery Platform (SDP) is a set of components that are used to implement a service delivery architecture that includes service creation, life-cycle, session control, and security support.
- A service platform revolves around three main actors: service providers, service requesters and a service registry.
- Service providers publish service descriptions, and service requesters discover services and bind to the service providers.
- Publication and discovery are based on service descriptions.
- Current telecom SDP's utilize SIP, IMS, Web, and IPTV technologies in delivering services to mobile users.
- The Service Creation Environment (SCE) is responsible for supporting end user or developer driven software development. The main motivation for a SCE is to support easier and more flexible service creation and deployment.
- Typically, the SCE is used by a developer experienced with the supported scripting language and other tools; however, it is also envisaged that end users use SCEs to compose services.
- For end users, scripting and programming are expected to be replaced with adapting pre-generated service composition templates for current needs and requirements.

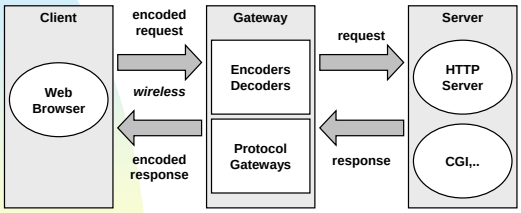
Platforms

- Java Micro Edition (Java ME)
- iPhone
- Symbian and Series 60
- BREW
- WAP
- Windows Mobile and .NET Compact Framework
- NoTA
- Linux Maemo
- Android
- OSGi
- Python
- Flash Lite
- Opera Mini

Example: Symbian



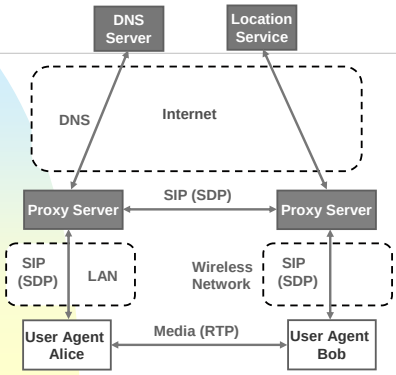
Example: WAP Programming model



Support Technologies

- Session Initiation Protocol (SIP)
- IP Multimedia Subsystem (IMS)
- Web Services
- IP Television (IPTV)
- SQLite
- OpenGL ES
- PAMP

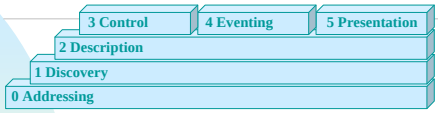
Example: SIP



Service Discovery

- UPnP
- Jini
- Service Location Protocol (SLP)
- ZeroConf

Example: UPnP



0 Control point and device get addresses

1 Control point finds interesting device

2 Control point learns about device capabilities

3 Control point invokes actions on device

4 Control point listens to state changes of device

5 Control point controls device and/or views device status using Web browser

Mobility Solutions

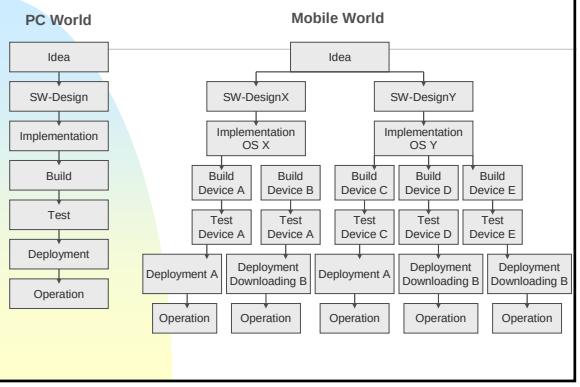
- Mobility and multi-homing can be realized on different layers
  - Network
    - Mobile IP, HMIP, NEMO
  - Between network and transport
    - Host Identity Protocol (HIP)
  - Transport (SCTP)
    - TCP extensions, SCTP (TrASH)
  - Application
    - SIP, Wireless CORBA, overlays
    - Re-establish TCP-sessions after movement

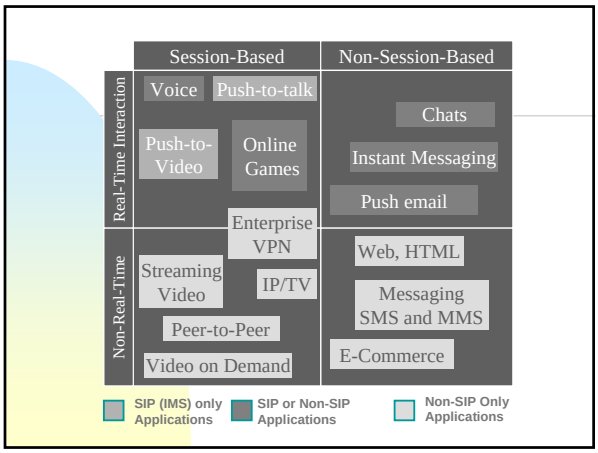
Advanced Topics

- Overlay Networks
- Context-awareness
- Service composition
- Security and Trust
- Charging and Billing

Mobile Service Development

- The mobile landscape is fragmented
  - Heterogeneous device base
  - Many different wireless technologies
- The situation is challenging for the developer
  - Many APIs
  - Many middleware platforms
  - APIs evolve over time
- Current challenge of the industry pertains to improving the development processes





Summary

- Mobile middleware
  - Desktop middleware not usable on small, mobile devices
  - The mobile and wireless environment is different from the traditional fixed-network computing environment
  - Special solutions are needed
    - Decoupling, context-awareness, adaptation, ..
  - J2ME, WAE, Wireless CORBA, ..
- Current trends
  - Flexibility, decoupled nature
  - Convergence / unification