Mobile Middleware Course

Mobile Platforms

Sasu Tarkoma

Contents

- Middleware
- Mobile middleware
- Challenges
- Platforms
- Comparison
- Outlook

Middleware

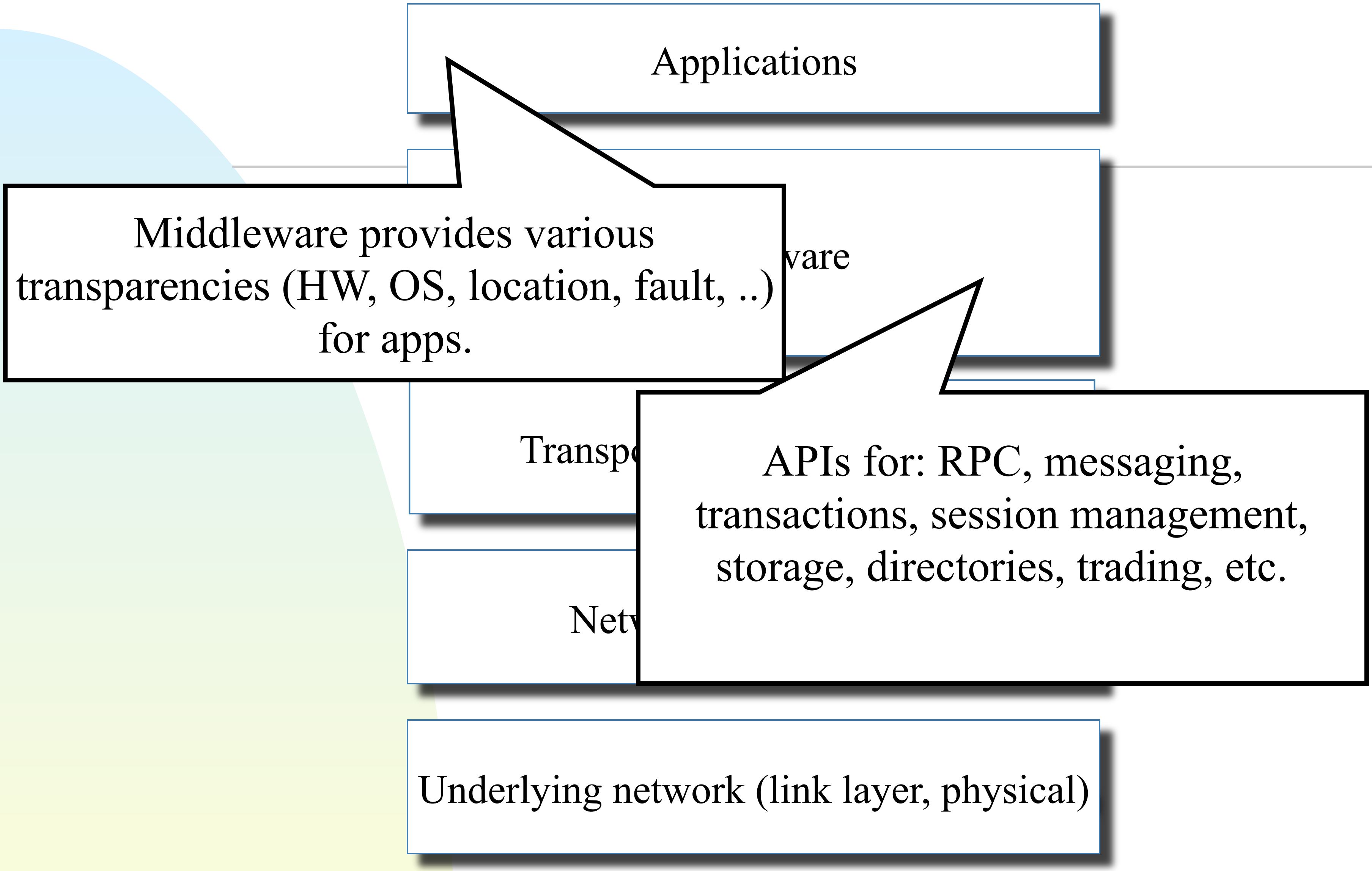
- Widely used and popular term
- Fuzzy term
- One definition
  - “A set of service elements above the operating system and the communications stack”
- Second definition
  - “Software that provides a programming model above the basic building blocks of processes and message passing” (Colouris, Dollimore, 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 and events
  - Security
  - synchronization
- Network stack (transport and below) is not part of middleware



Applications

ware

Transp

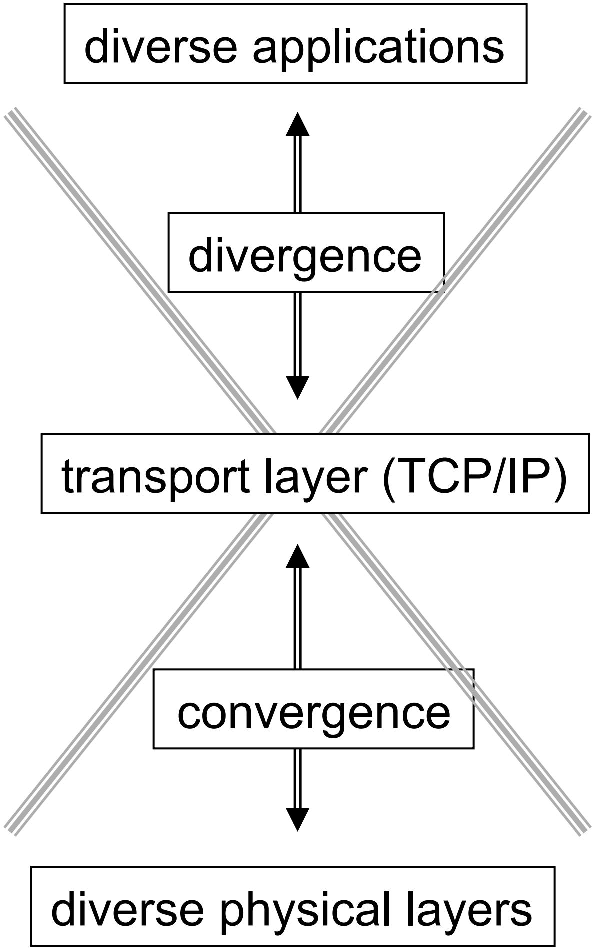
Netw

Underlying network (link layer, physical)

Middleware provides various transparencies (HW, OS, location, fault, ..) for apps.

APIs for: RPC, messaging, transactions, session management, storage, directories, trading, etc.

The Hourglass



diverse applications

divergence

transport layer (TCP/IP)

convergence

diverse physical layers

Mobile Platforms

- Collections of central services and libraries with both reactive and proactive functions

- APIs typically logically centralized

- Distributed between elements of the environment
  - Multi-tier client-server
  - Peer-to-peer
  - Hybrids

- The platform running on the mobile terminal and the characteristics of the device determine how service is rendered for the end user

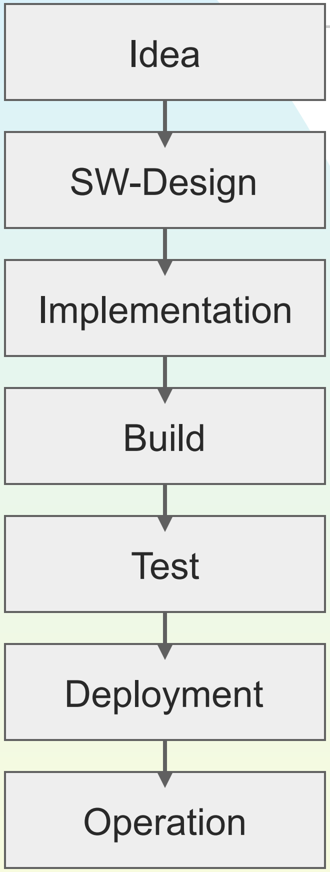
Wireless and Cloud

- Wireless hop is the limiting factor
  - Bandwidth, connectivity, reachability, tail energy, costs
- Server side scalability can be achieved by using traditional solutions:
  - clusters, caching, geographical distribution, load balancing, data centers
- Cloud computing
  - Integration, offloading
  - Web apps vs. native apps

Mobile Service Development

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

PC World



Idea

SW-Design

Implementation

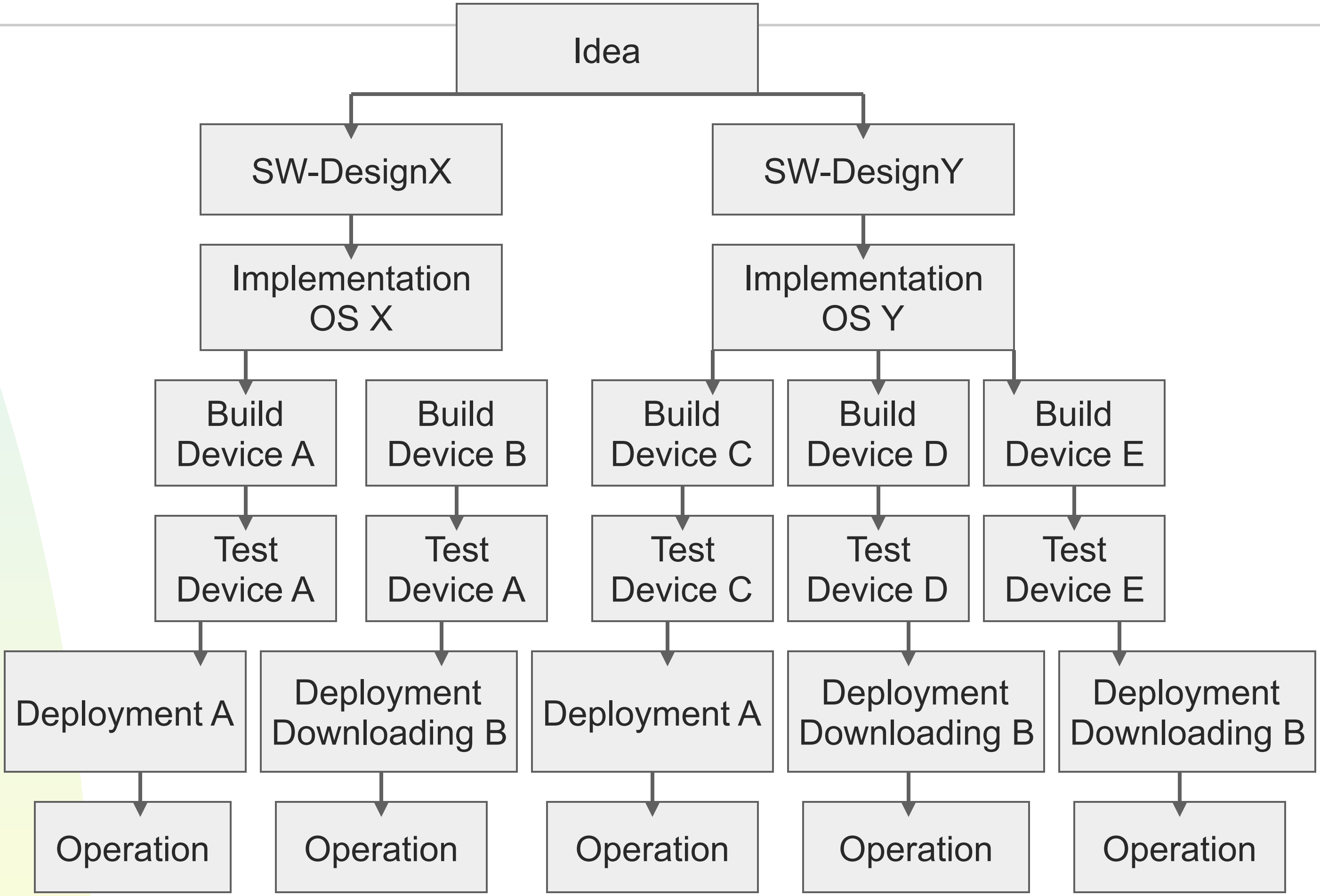
Build

Test

Deployment

Operation

Mobile World



Idea

SW-DesignX

SW-DesignY

Implementation OS X

Implementation OS Y

Build Device A

Build Device B

Build Device C

Build Device D

Build Device E

Test Device A

Test Device A

Test Device C

Test Device D

Test Device E

Deployment A

Deployment Downloading B

Deployment A

Deployment Downloading B

Deployment Downloading B

Operation

Operation

Operation

Operation

Operation

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

- 2009
  - Java Micro Edition (Java ME)
  - iOS
  - Symbian and Series 60
  - Windows Mobile
  - Linux Maemo (MeeGo)
  - Android
  - BREW
  - WAP
- 2012
  - iOS
  - Android
  - Windows Phone 7
  - HTML5 web apps

Application Trends

- WP7
  - Native apps, cloud integration
- iOS
  - Native apps, cloud integration
  - Potential for Web apps
- Android
  - Native apps, cloud integration
- WebOS
  - Web apps with HTML5
  - Obsolete (open source)
- Blackberry
  - Native and Web apps

Challenges

- Fragmentation is a major problem
  - device-level fragmentation
  - standard fragmentation
  - implementation fragmentation
- Energy consumption
  - Modelling: where is the energy going
  - Optimization: how to improve things
- Security is also a problem
  - Sandboxed environments and privileged operations require certification
  - Certification is difficult for developers
  - Current trend is towards application stores and more lightweight certification processes
  - No malware for iOS, plenty for Android

Update problems

- WP7 and Android updates not available for all devices
- Microsoft update development process
  - OEMs work with Microsoft to customize the update with their handset firmwares
  - The carriers (who also have code on the OEM phones) check these updates
  - Same as with Android, Google tightening control.
- Recent problems: phones bricked (out of battery when updating)
- Older problems: excessive background data transfer, update problems

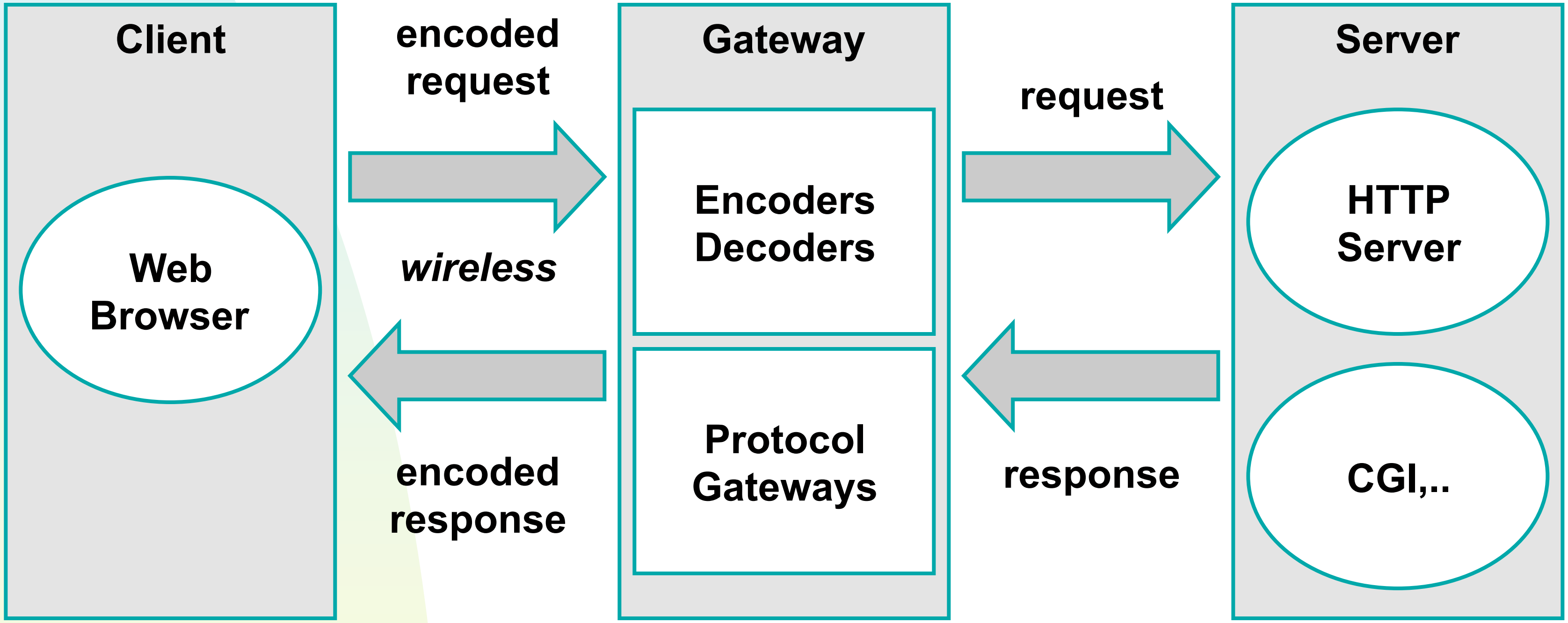
Examples

- Classical examples
  - WAP
  - Java ME
  - Symbian
  - MAEMO / MeeGo
- Current Platforms
  - Windows Phone 7
  - iPhone
  - Android
  - Web apps

WAE

- Wireless Application Environment (WAE)
  - A suite of protocols and specifications for optimizing data transfer for wireless communication
- WAP stack
  - Focus on binary transmission
  - WSP (Wireless Session Protocol)
    - HTTP replacement, “compressed”
  - WTP (Wireless Transaction Protocol)
    - Request/response, more efficient than TCP
  - WTLS (Wireless Transport Layer Security)
    - Based on TLS, may not be end-to-end with a gateway
  - WDP (Wireless Datagram Protocol)
    - UDP replacement

Web Access with Gateway



Web Access

- Data transformation
  - WAP gateway performs data transformation between WML (or XHTML) and HTML
- Data compression
  - Technique are used for dealing with images and other graphics
- Adaptability
  - User profile and device characteristics are stored in the WAP gateway
- Security
  - Secure Enterprise Proxy (SEP) using 128-bit encryption in WAP 1.2
- Service discovery and mobility support
  - WAP’s “walled garden” – WAP gateways are provided by ISP such as AOL

WAE: current status

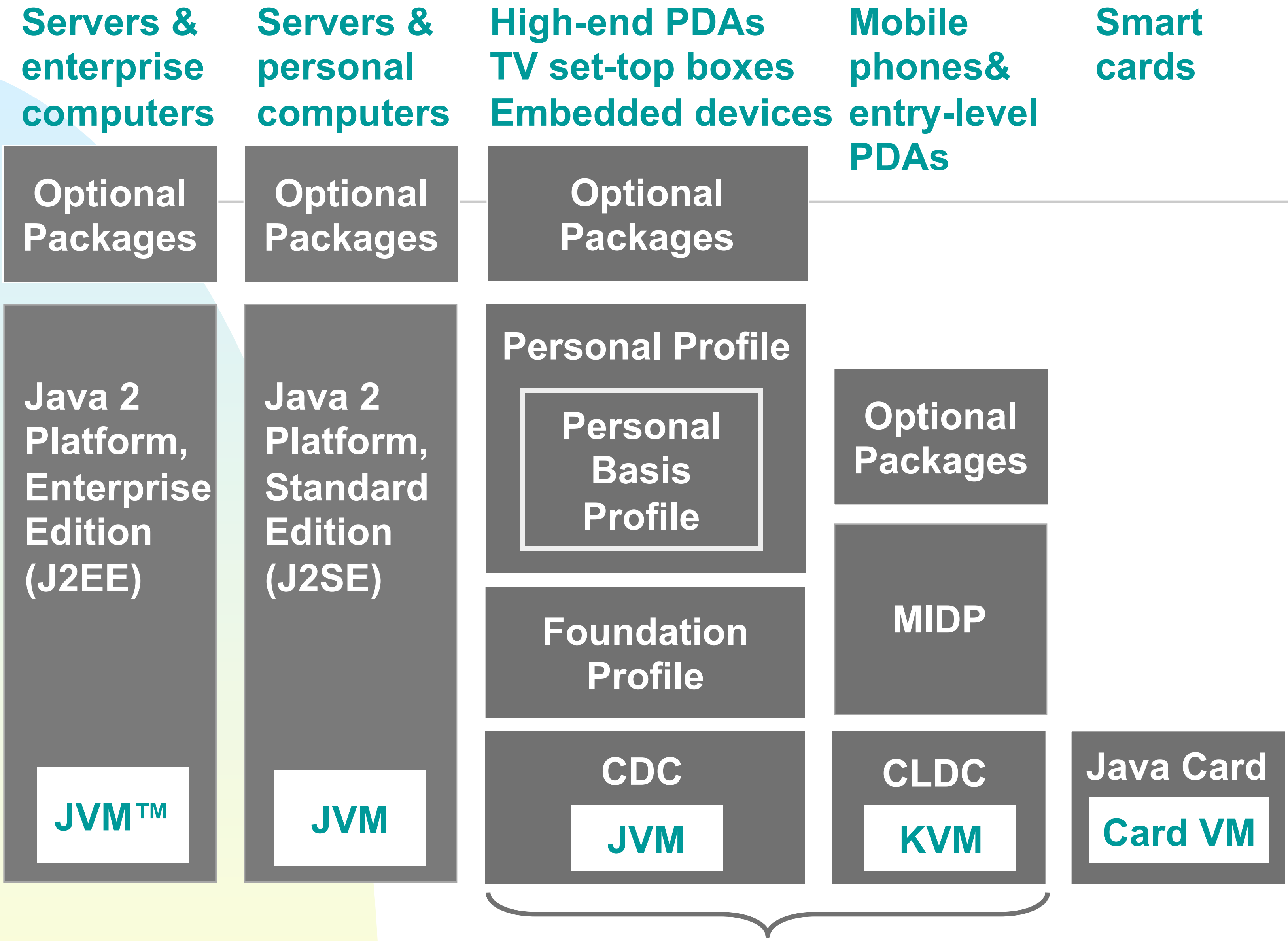
- WAP Forum now in OMA (Open Mobile Alliance)
- WAP 2.0, is a re-engineering of WAP using a cut-down version of XHTML with end-to-end HTTP
- Gateway and custom protocol suite is optional.
- WAP used by many handsets
  - 1.2 version introduced WAP Push (typically using an SMS message)
- Typically versatile networking stacks with also IPv6 support

Java Micro Edition (Java ME)

- Java for consumer electronics and embedded devices
- A virtual machine and a set of APIs
- Configurations and profiles
  - Configurations
    - two-low level APIs and optimized VMs
      - CDC, CLDC
  - Profiles
    - API specification on top of a configuration for complete runtime
    - CLDC: MIDP
    - CDC: Foundation, Personal Basis, Personal
    - Profiles defined using Java Community Process (JCP)

Java Editions

- Java Card, which allows small Java-based applications to be executed on smart cards
- Java ME (Micro Edition, formerly J2ME), which specifies several different profiles,collections of libraries
  - for devices that are sufficiently limited that it is not feasible to support the full Java platform on them.
- Java SE (Standard Edition), which is the platform for general purpose desktop PCs.
- Java EE (Enterprise Edition), which includes the Java SE and a number of additional
  - APIs for multi-tier client-server enterprise applications.



Servers & enterprise computers

Optional Packages

Java 2 Platform, Enterprise Edition (J2EE)

JVM™

Servers & personal computers

Optional Packages

Java 2 Platform, Standard Edition (J2SE)

JVM

High-end PDAs
TV set-top boxes
Embedded devices

Optional Packages

Personal Profile

Personal Basis Profile

Foundation Profile

CDC

JVM

Mobile phones& entry-level PDAs

Optional Packages

MIDP

CLDC

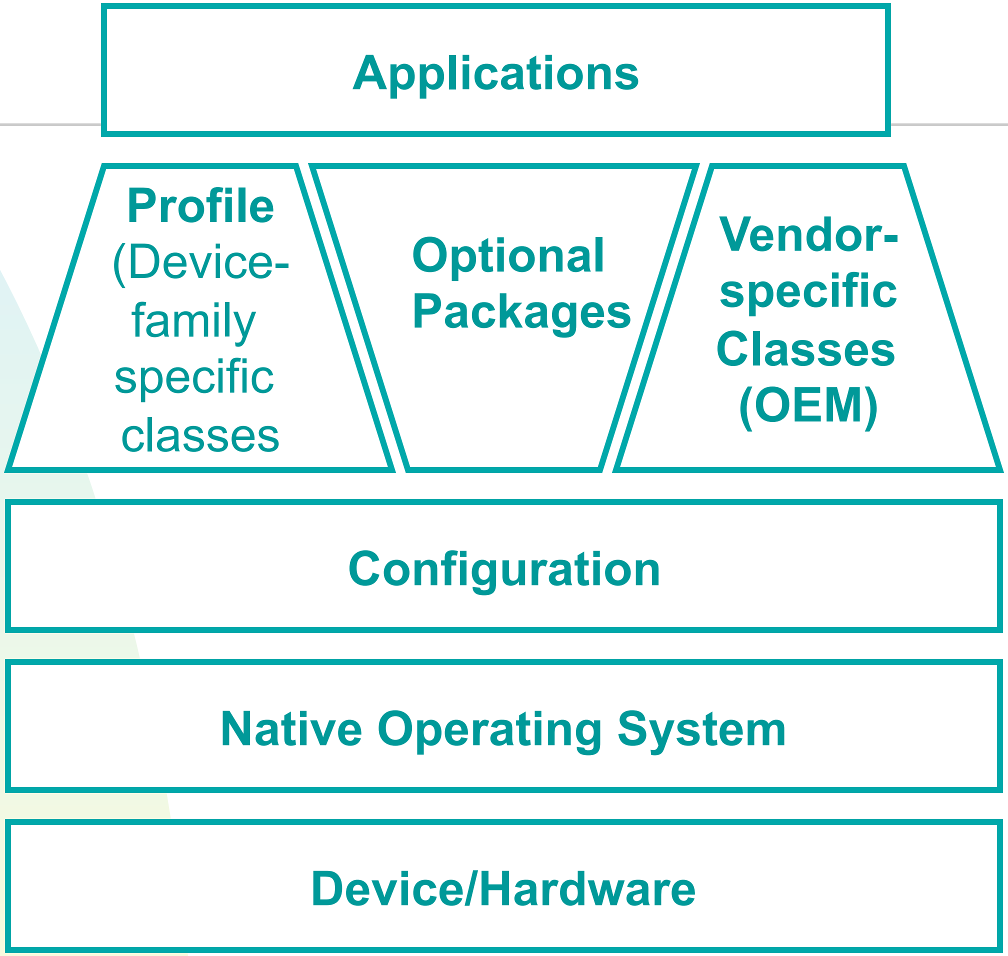
KVM

Smart cards

Java Card

Card VM

Java Platform, Micro edition (Java ME)



Applications

Profile (Device-family specific classes)

Optional Packages

Vendor-specific Classes (OEM)

Configuration

Native Operating System

Device/Hardware

Important JSRs

- 75 File Connection and PIM
- 82 Bluetooth
- 120 Wireless Messaging API (WMA)
- 135 Mobile Media API (MMAPI) Audio, video, multimedia
- 172 Web Services
- 177 Security and Trust Services
- 179 Location API
- 180 SIP API
- 184 Mobile 3D Graphics
- 185 Java Technology for the Wireless Industry (JTWI) General
- 205 Wireless Messaging 2.0 (WMA)
- 211 Content Handler API
- 226 SVG 1.0
- 229 Payment API
- 234 Advanced Multimedia Supplements (AMMS) MMAPI extensions
- 238 Mobile Internationalization API
- 239 Java Bindings for the OpenGL ES API
- **248 Mobile Service Architecture General**
  - **Collects useful specifications**
- 256 Mobile Sensor API
- 287 SVG 2.0

MIDP 3.0

- MIDP 3 specified in JSR 271 will specify the 3rd generation mobile APIs.
  - AMS (Application Management System)
  - Multitasking
  - Provisioning and OTA
  - Shared libraries
  - Security and access control
  - Service framework
  - Inter-MIDlet communication
  - User Interface improvements
- A key design goal of MIDP3 is backward compatibility with MIDP2 content
- Approved in Dec, 2009. Not supported by current phones.

CDC Technology

- Connected Device Configuration (CDC); JSR 36, JSR 218
- Foundation Profile, (FP); JSR 46, JSR 218
- Personal Basis Profile, (PBP); JSR 129, JSR 217
- Personal Profile, (PP); JSR 62, JSR 216
- J2ME RMI Optional Package, (RMI OP); JSR 66
- JDBC Optional Package for CDC/Foundation Profile API; JSR 169
- Java TV API; JSR 927
- Java Secure Socket Extension for Connected Device Configuration (JSSE)

APIs

- Mobile Sensor API, Contactless Communication API, and Location API
- The Mobile Broadcast Service API supports the delivery of streaming multimedia to mobile phones
- Converged communications support is provided by the XML API and *IP Multimedia Subsystem (IMS) Services API*
- *The Mobile User Interface Customization API and Scalable 2D Vector Graphics API*

Summary

	Java ME MIDP
Development	Java ME
Network scanning	No
Network interface control	No
Background processing	Yes (multi-tasking support in MIDP 3.0)
Energy and power monitoring and control	No
Memory management	Limited
Persistent storage	Limited, exension
Location information	Extension
HTML 5	N/A
SIP API support	Extension
Open Source	No
3rd party application installation	Certificate
Level of fragmentation	Fragmented