

Mobile Middleware Course

Mobile Platforms

Sasu Tarkoma

## Contents

- Overview
- Platforms
- Research Systems
- Summary

## 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
- Server side scalability can be achieved by using traditional solutions:
  - ◆ clusters, caching, geographical distribution, load balancing, data centers

## Challenges

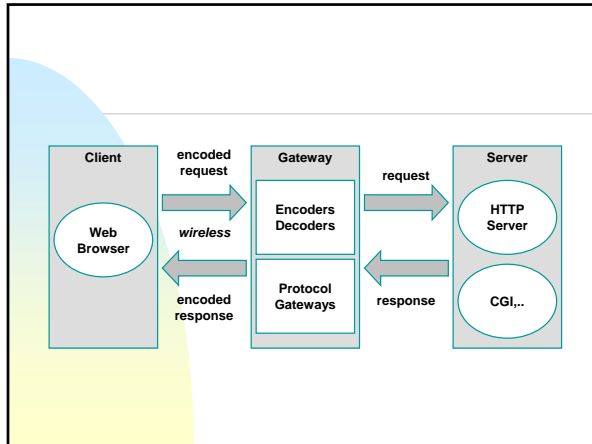
- Fragmentation is a major problem
- Different types of fragmentation
  - ◆ device-level fragmentation
  - ◆ standard fragmentation
  - ◆ implementation fragmentation
- Security is also a problem
  - ◆ Sandboxed environments and privileged operations require certification
  - ◆ Certification is difficult for developers
  - ◆ Current trend is towards application stores and more lightweight certification processes

## Examples

- WAE
- Java ME
- Symbian
- MAEMO
- Android
- Pervasive Computing Middleware
- Spice
- Fuego

## WAE

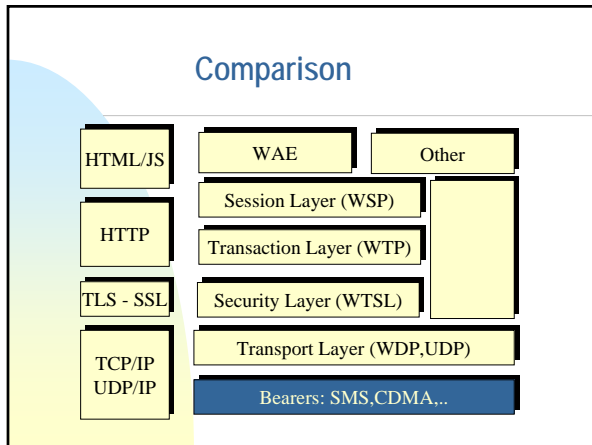
- Wireless Application Environment (WAE)
  - ◆ A suite of protocols and specifications for optimizing data transfer for wireless communication
- WAP stack
  - ◆ Focus on binary transmission
  - ◆ Protocols
    - WSP (Wireless Session Protocol)
      - HTTP replacement, "compressed"
    - WTP (Wireless 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

- **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

## Comparison



## WAE: current status

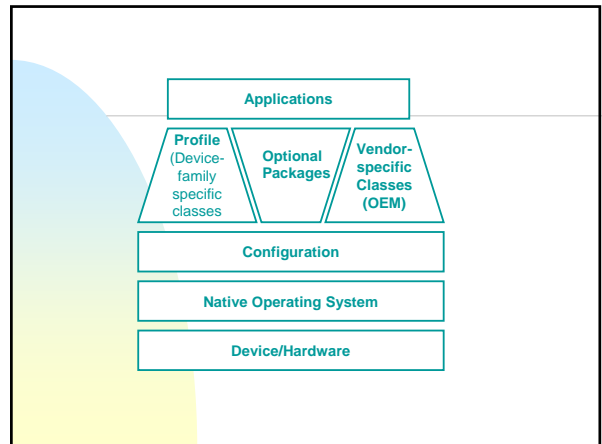
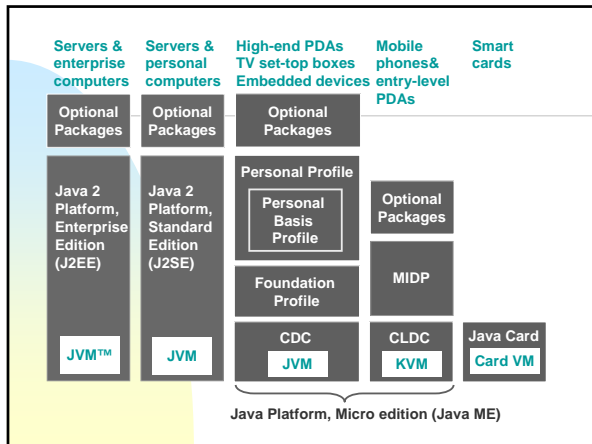
- WAP Forum now in OMA (Open Mobile Alliance)
- WAP 2.0, is a re-engineering of WAP using a cut-down version of XHTML with end-to-end HTTP
- Gateway and custom protocol suite is optional.
- WAP used by many handsets
  - ◆ 1.2 version introduced WAP Push (typically using an SMS message)
- Typically versatile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

## Java Micro Edition (Java ME)

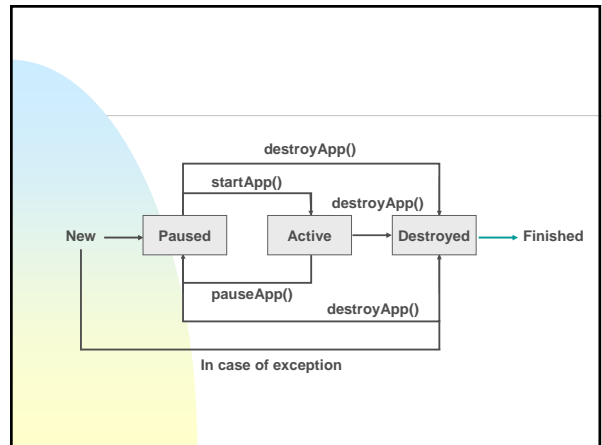
- Java for consumer electronics and embedded devices
- A virtual machine and a set of APIs
- Configurations and profiles
  - ◆ **Configurations**
    - two-low level APIs and optimized VMs
      - CDC, CLDC
  - ◆ **Profiles**
    - API specification on top of a configuration for complete runtime
   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

## Java Editions

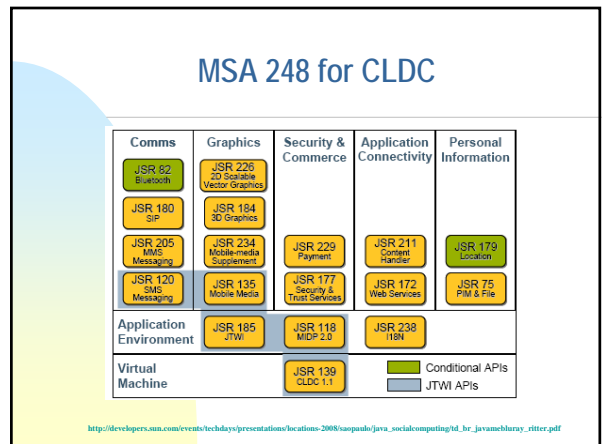
- Java Card, which allows small Java-based applications to be executed on smart cards
- Java ME (Micro Edition, formerly J2ME), which specifies several different profiles, collections of libraries
  - ◆ for devices that are sufficiently limited that it is not feasible to support the full Java platform 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.



- ### MIDlet lifecycle
- MIDlets follow a specific lifecycle
    - Paused
    - Active
    - Destroyed.
  - While creating the MIDlet, the runtime system calls the constructor of the MIDlet class
  - MIDlet's constructor cannot access the Display. When the application receives focus startApp() is executed.
  - Application then should set the desired display by calling Display.getDisplay(this).setCurrent() method.
  - When the application loses the focus, the method pauseApp() is called.
  - When paused, the display is not active and it is not shown to the user. While paused, all timers and threads are still running.
  - The application can call setCurrent() to indicate the screen that should be presented when returning from the Paused state.
  - The method destroyApp() is called when the application is being destroyed.



- ### 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**
  - 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 will be backward compatibility with MIDP2 content

## 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)

## Symbian

- OS for handheld devices with limited resources
- User interface framework
- APIs (C++)
- Tools
- Operating System
  - ◆ Pre-emptive, multitasking, multithreading, memory protection
  - ◆ Event-based, active objects
  - ◆ Memory conservation, reliability, CPU optimizations

## Software Components

- Kernel
  - ◆ Manages and controls access to hw
  - ◆ Hw-supported privileges, kernel mode
- Application
  - ◆ Program with a user interface
  - ◆ Runs in user mode in its own process
- Server
  - ◆ Program without a user interface
  - ◆ Manages resources, provides interface to clients
  - ◆ File server, window server, comms, ..
- Engine
  - ◆ Application part that manipulates data, typically separate DLL

## Key layers

- The Symbian OS System Model contains the following layers:
  - ◆ UI Framework Layer.
  - ◆ Application Services Layer.
  - ◆ Java ME.
  - ◆ OS Services Layer: generic OS services, communications services, multimedia and graphics services, connectivity services.
  - ◆ Base Services Layer.
  - ◆ Kernel Services and Hardware Interface Layer.

## Symbian OS 9.x

- Recently, in Symbian OS 9.x, most applications must be signed using a centralized process provided by Symbian Ltd. in order to be installable and executable on a mobile phone.
- An unsigned applications have very limited features and at least in theory cannot perform harmful actions.
- One motivation for application signature process, called Symbian Signed, is to improve mobile phone security by preventing the installation and execution of unknown and possibly hazardous programs.
- Several viruses and trojan horse programs have been developed for the OS, for example Cabir, which have caused some concerns for the trustworthiness of mobile software and prompted a number of anti-virus products for mobile phones.

## WebKit for S60

- The Nokia Web Browser is based on the S60WebKit
- WebKit contains the WebCore and JavaScriptCore components that Apple uses in its Safari browser.
- Based on KHTML and KJS from KDE's Konqueror open source project, this software offers improvements in Web site usability on smartphones through the re-use of an existing desktop rendering engine that has been developed and optimized by a large open source community over many years.
- The Nokia Web Browser supports Dynamic HTML, AJAX applications, and W3C's XHTML 1.0, DOM, CSS and SVG-Tiny.
- Other supported Web standards include SSL and ECMAScript; and Netscape style plug-ins such as Flash Lite and audio.

## Windows Mobile

- Windows Mobile 6 was released by Microsoft at the 3GSM World Congress 2007 and it comes in three flavours
  - ◆ standard version for smartphones
  - ◆ a version for PDAs with phone functionality
  - ◆ a classic version for PDAs without phone features.
- Based on the Windows CE 5.0 operating system and has been designed to integrate with Windows Live and Exchange products.
- Software development for the platform is typically done using Visual C++ or .NET Compact Framework.
- When native client-side functionality is not needed, server-side code can be used that is deployed on a mobile browser, such as the Internet Explorer Mobile bundled with Windows Mobile.

## Compact Framework

- The .NET Compact Framework is a subset of the .NET Framework and shares many components with the desktop software development environment.
- The framework includes
  - ◆ optimized *Common Language Runtime (CLR)*
  - ◆ a subset of the .NET Framework class library, which supports features such as *Windows Communication Foundation (WCF)* and *Windows Forms*.
  - ◆ It also contains classes that are designed exclusively for the .NET Compact Framework.
- A platform adaptation layer exists between Windows CE and the common language runtime
  - ◆ mapping the services and device interfaces required by the CLR and Framework onto Windows CE services and interfaces.
- The expectation is that managed components developed using .NET languages, such as C#, are used to create the applications.
- It is also possible to use the Win32 API with .NET CF, which allows directly to access OS features.

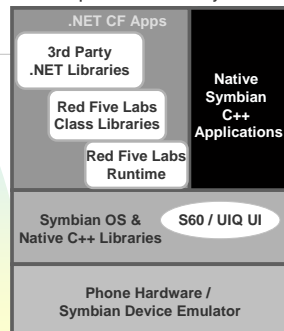
## Runtime

- The .NET Compact Framework CLR is made up of the following three component:
  - ◆ class libraries
  - ◆ execution engine
  - ◆ platform adaptation layer
- The purpose of the class libraries is to provide a basic set of classes, interfaces, and value types
  - ◆ the foundation for developing applications in .NET.
- The execution engine is the core component of the CLR. It provides the fundamental services needed for executing managed code.
  - ◆ The execution engine includes components such as a JIT compiler, a class and module loader, and a garbage collector.
- The PAL layer maps calls from the execution engine to the functions of the underlying operating system.

## Symbian port of .NET CF

- The main differences observed in the porting process were the following:
  - ◆ A C++ dialect that redefines basic language structures.
  - ◆ No writable global and writable static variables allowed in DLLs.
  - ◆ Extensively used client/server model that, for example, implies constraints for accessing file and networking functions.
  - ◆ Event-driven programming model with a focus on non-preemptive multitasking.
  - ◆ Symbians error handling and cleanup model.
  - ◆ Concepts from the Unix/Windows world such as environment variables as well as several file and networking functions are missing.

### .NET Compact Framework on Symbian OS

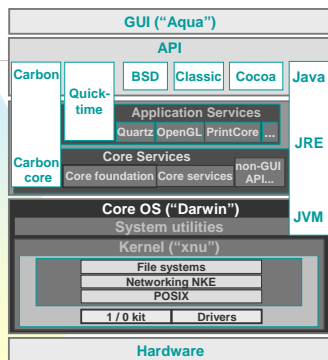


## iPhone

- The iPhone OS is a mobile operating system developed by Apple Inc. for their iPhone and iPod touch products.
- The OS is derived from Max OS X and uses the Darwin foundation.
- Darwin is built around XNU, a hybrid kernel that combines the Mach 3 microkernel, various elements of Berkeley Software Distribution (BSD) Unix, and an object-oriented device driver API (I/O Kit).
- The iPhone OS is based on four abstraction layers, namely the *Core OS layer*, the *Core Services layer*, the *Media layer*, and the *Cocoa Touch layer*.

## iPhone OS

- The iPhone OS's user interface is based on *multi-touch gestures*. Interface control elements consist of sliders, switches, and buttons.
- Interaction with the OS includes gestures such as swiping, tapping, pinching, and reverse pinching.
- Additionally, using internal accelerometers, rotating the device on its y-axis alters the screen orientation in some applications.

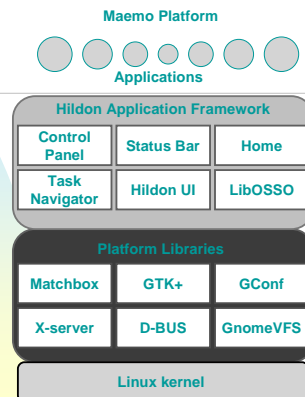


## MAEMO I

- Open Source development platform for Nokia Internet Tablets and other Linux-based devices
- Maemo SDK
- [www.maemo.org](http://www.maemo.org)

## MAEMO II

- The Maemo platform from Nokia includes the Internet Tablet OS, which is based on Debian GNU/Linux and draws much of its GUI, frameworks, and libraries from the GNOME project.
- Maemo is based on the Linux operating system kernel, which is a monolithic kernel that supports multiple hardware platforms.
- It uses the Matchbox window manager, and like Ubuntu Mobile, it uses the GTK-based Hildon as its GUI and application framework. The Maemo platform is intended for Internet tablets, which are smaller than laptops, but larger and more versatile than PDAs.
- A tablet may have a small keyboard, and central characteristics include a stylus and a touch-sensitive screen.
- Graphical interfaces must be designed with the touch screen in mind.



## Android I

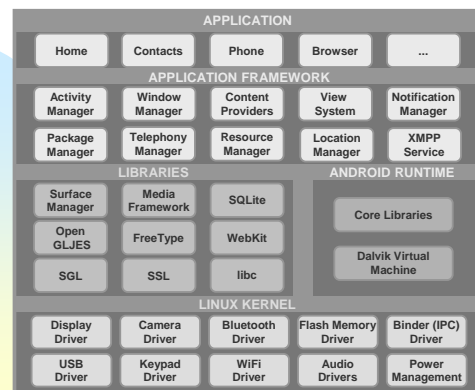
- Mobile OS and application platform from Google
- Open Handset Alliance
- Linux kernel
- Open Source
- Uses Java to build applications (Java SE class library parts from Apache Harmony project)
- Optimized virtual machine called "Dalvik"
  - Runs .dex files (derived from .class or .jar)
  - Relies on underlying system for process isolation, memory mng, and threading
- Independent of Sun and JCP
- Java APIs for basic comms, location, SQLite, OpenGL, SyncML

## Android II

- Android includes a set of C/C++ libraries used by various components of the Android system. The capabilities of these libraries are exposed to developers through the Android application framework APIs.
- The core libraries include:
  - System C library, a BSD-derived implementation of the standard C system library (libc), adapted for embedded Linux-based devices.
  - Media Libraries based on PacketVideo's OpenCORE.
  - Surface Manager that manages access to the display subsystem and seamlessly renders 2D and 3D graphic layers from multiple applications.
  - LibWebCore, a web browser engine which powers both the Android browser and an embeddable web view.
  - SGL, the underlying 2D graphics engine.
  - 3D libraries, an implementation based on OpenGL ES 1.0 APIs.
  - FreeType, bitmap and vector font rendering.
  - SQLite, a lightweight relational database engine available to all applications through the framework API.

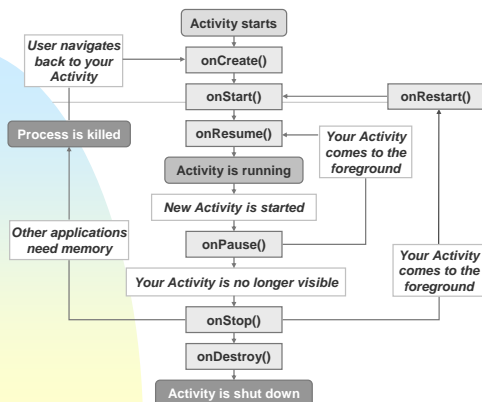
## Android: Key Components

- AndroidManifest.xml. This XML document contains the configuration that tells the system how the top-level components will be processed.
- Activities. An activity is an object that has a life cycle and performs some work. An activity can involve user interaction. Typically one of the activities associated with an application is the entry point for that application.
- Views. A view is an object that knows how to render itself to the screen.
- Intents. An intent is a message object that represents an intention to perform some action.
  - In Android terminology, an application has an intent to view a Web page, and generates an Intent instance in order to implement the intent. In this case, a browser would be used to load and display the Web page.
- Services. A service is code that runs in the background. The service exposes methods for to components. Other components bind to a service and then invoke methods provided by using remote procedure calls.
- Notifications. A notification is a small icon that is visible in the status bar. Users can interact with this icon to receive information.
- ContentProviders. A ContentProvider provides access to data on the device.



## Activity states

- An activity has four main states:
  - Active. An activity is active when it is in the foreground of the screen and at the top of the activity stack.
  - Paused. An activity is paused when it has lost focus, but is still visible. A paused activity is alive, but can be destroyed by the system if memory needs to be freed.
  - Stopped. An activity is stopped when it is obscured by another activity. The stopped activity retains its state, but it is no longer visible and can be destroyed by the system when memory is needed.
  - Destroyed/Inactive.
- If an activity is paused or stopped, the system can remove the activity from memory. This can happen in two ways, the system can ask the application to finish or simply destroy the process.



### Pervasive computing middleware

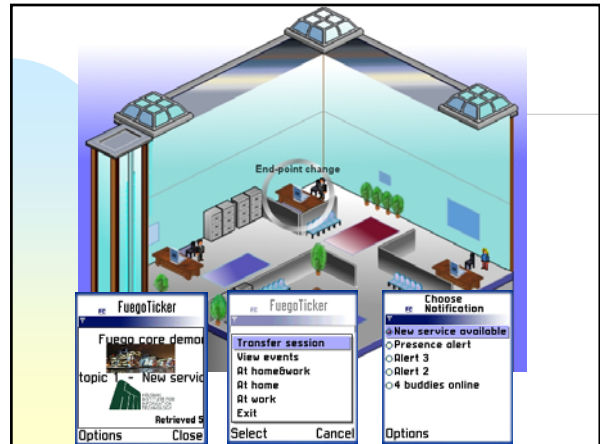
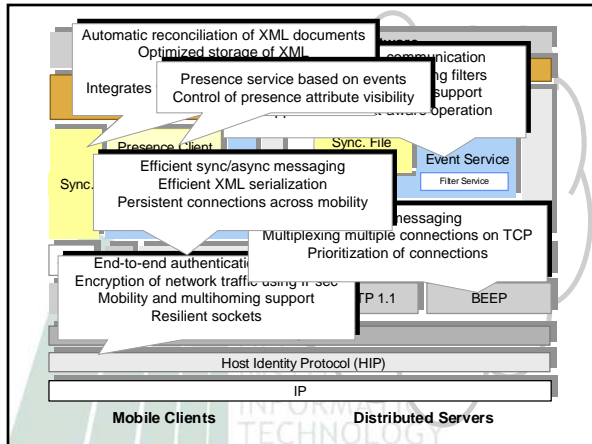
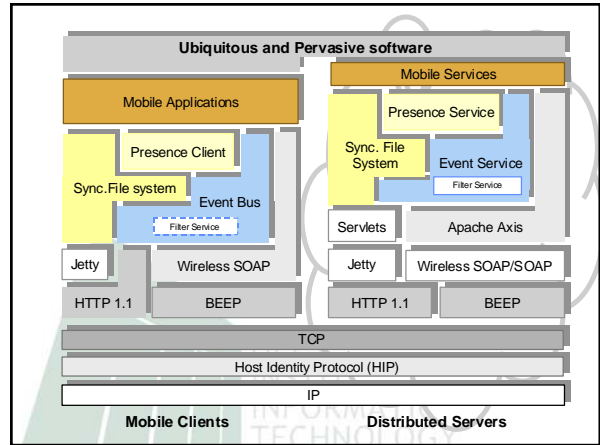
Projects	Key Issues
UIC	<b>Heterogeneity of devices and networks:</b> It helps users to specialize to the particular properties of different devices and network environments
X-Middle	<b>Disconnected operations in mobile applications:</b> It allows mobile users to share data when they are connected, or replicate the data and perform operations on them off-line when they are disconnected; data reconciliation takes place when user gets reconnected
Gaia	<b>Dynamic adaptation to the context of mobile applications:</b> It supports the development and execution of portable applications in active spaces
Lime	<b>Programming constructs which are sensitive to the mobility constraints:</b> It explores the idea by providing programmers with a global virtual data structure and a tuple space (Tspace), whose content is determined by the connectivity among mobile hosts
Tspaces	<b>Asynchronous messaging-based communication facilities without any explicit support for context-awareness:</b> It explores the idea of combination of tuple space (Tspace) and a database that is implemented in Java. Tspace targets nomadic environment where server contains tuple databases, reachable by mobile devices roaming around
L2imbo	<b>QoS monitoring and control by adapting applications in mobile computing environment:</b> It provides the facilities of multiple spaces, tuple hierarchy, and QoS attributes
Aura	<b>Distraction-free pervasive computing:</b> It develops the system architecture, algorithms, interfaces and evaluation techniques to meet the goal of pervasive computing

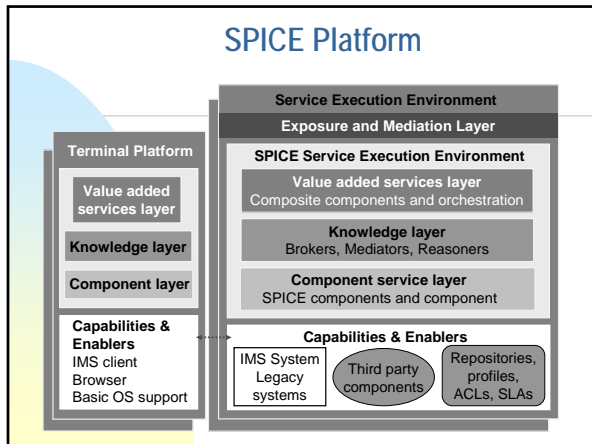
### Fuego Core (HIIT)

- Mobile and wireless environments have different requirements than desktop systems
  - User mobility, terminal mobility, connectivity, device characteristics, dynamic environments
- Vision: A service application is distributed among various application servers, network elements and terminals
- Three year Tekes project (2002-2004)
  - Industrial partners: Nokia, TeliaSonera, Elisa, Ericsson, Movial
  - Open source software
- Demonstration at WMCSA 2004

### Fuego Architecture

- The Fuego middleware service set for mobile computing
  - Data communication
    - Efficient wireless SOAP
    - Efficient content-based routing (asynchronous events)
  - Data synchronization
    - 3-way XML document merging
    - XML-aware distributed file system
  - Applications
    - Presence service, mobile ticker, image-album





- ### Summary
- Fragmentation is a current problem
    - ◆ Device, standard, implementation
    - ◆ Standards efforts are addressing this (JSR 248, ..)
  - APIs seem to be converging
    - ◆ Java is one of the key languages
      - Java ME poses significant problems regarding local storage and access to device resources
      - Android allows better access
    - ◆ SQLite, OpenGL ES, XML, Web services and REST
    - ◆ Browsers and Flash
  - Active research areas
    - ◆ Mobile virtualization as a solution to fragmentation?
    - ◆ Energy-awareness
    - ◆ Distribution of tasks
    - ◆ ..