Mobile Platforms and Application Development

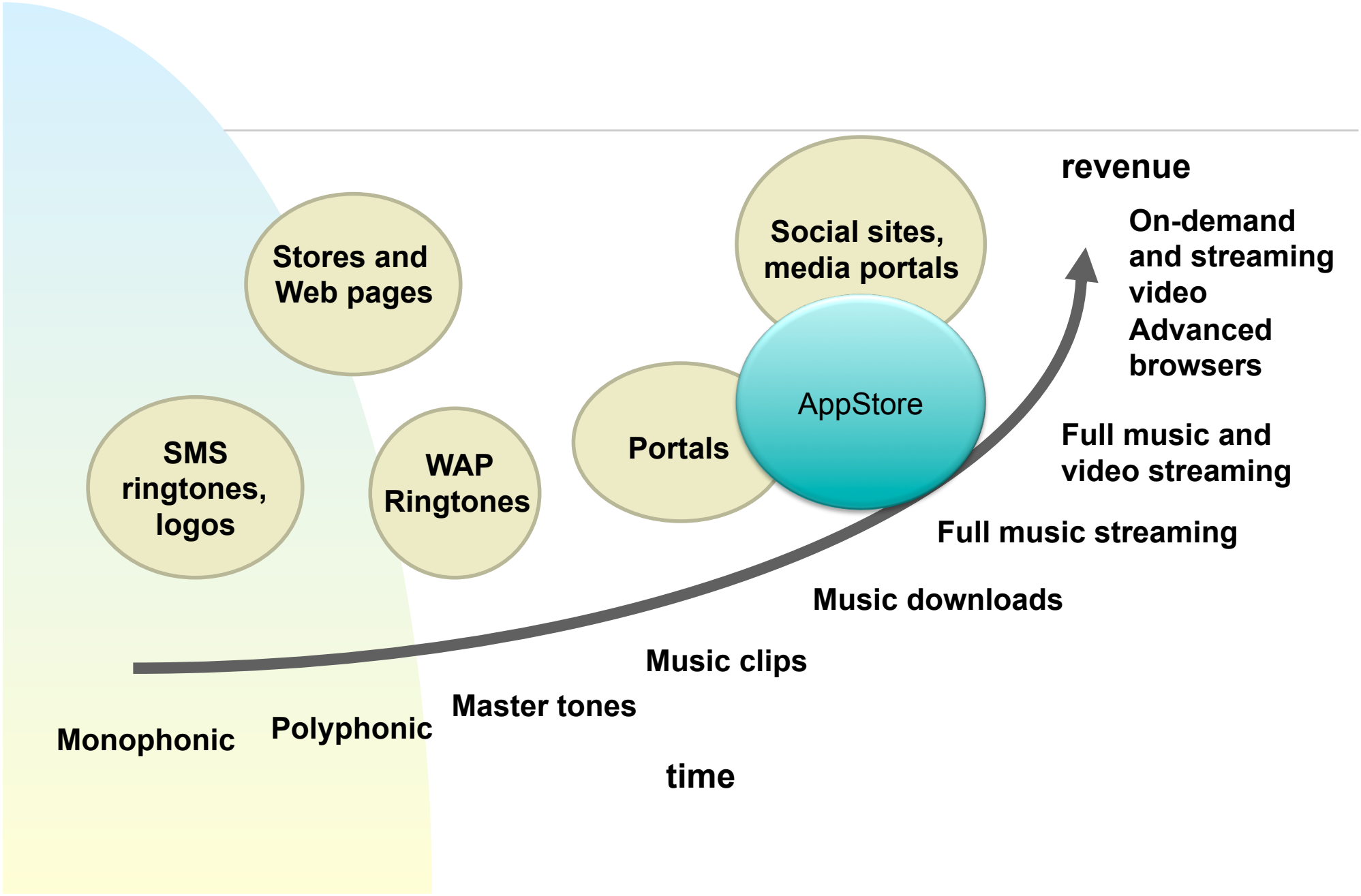
Prof. Sasu Tarkoma

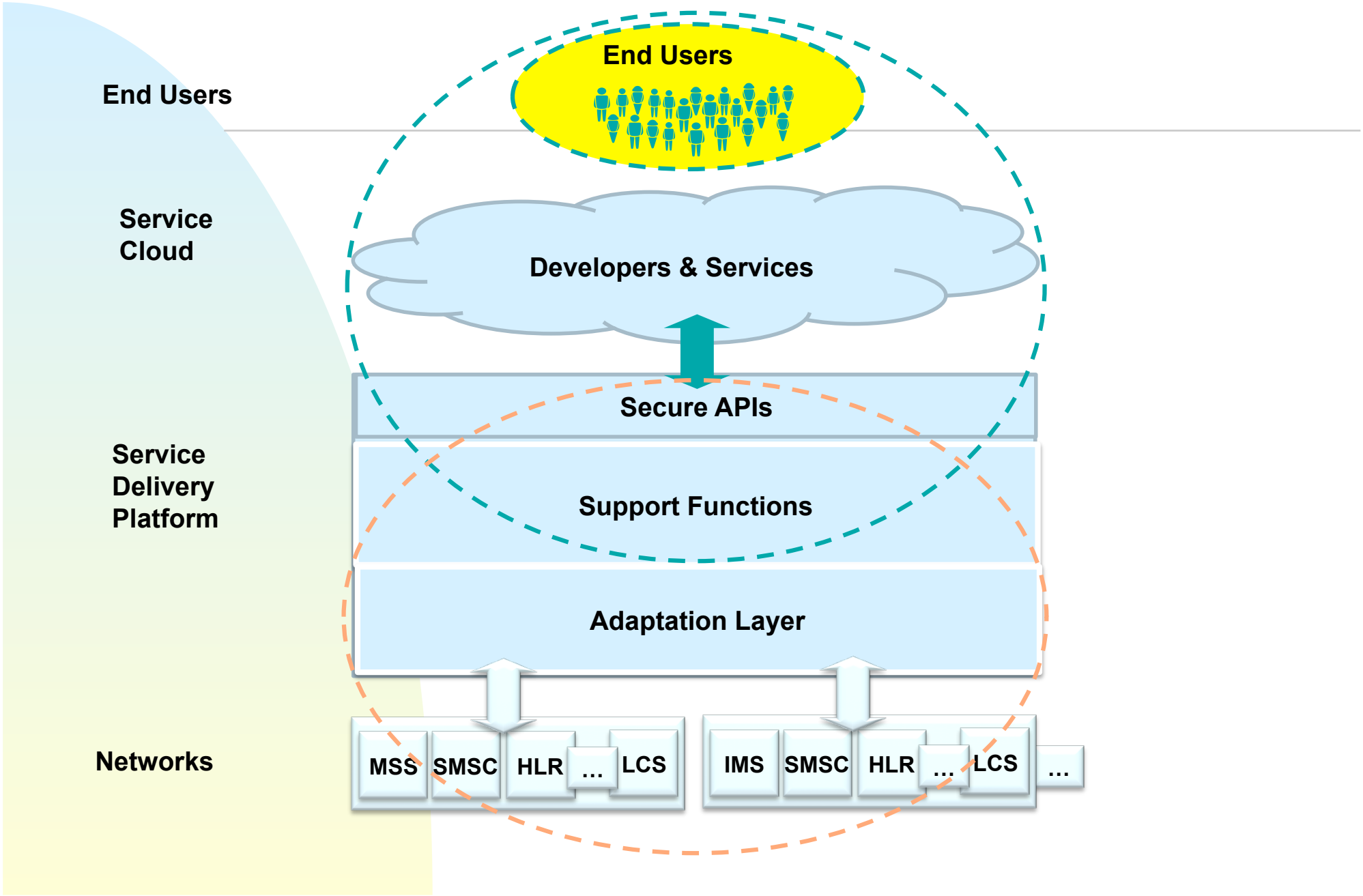
Introduction

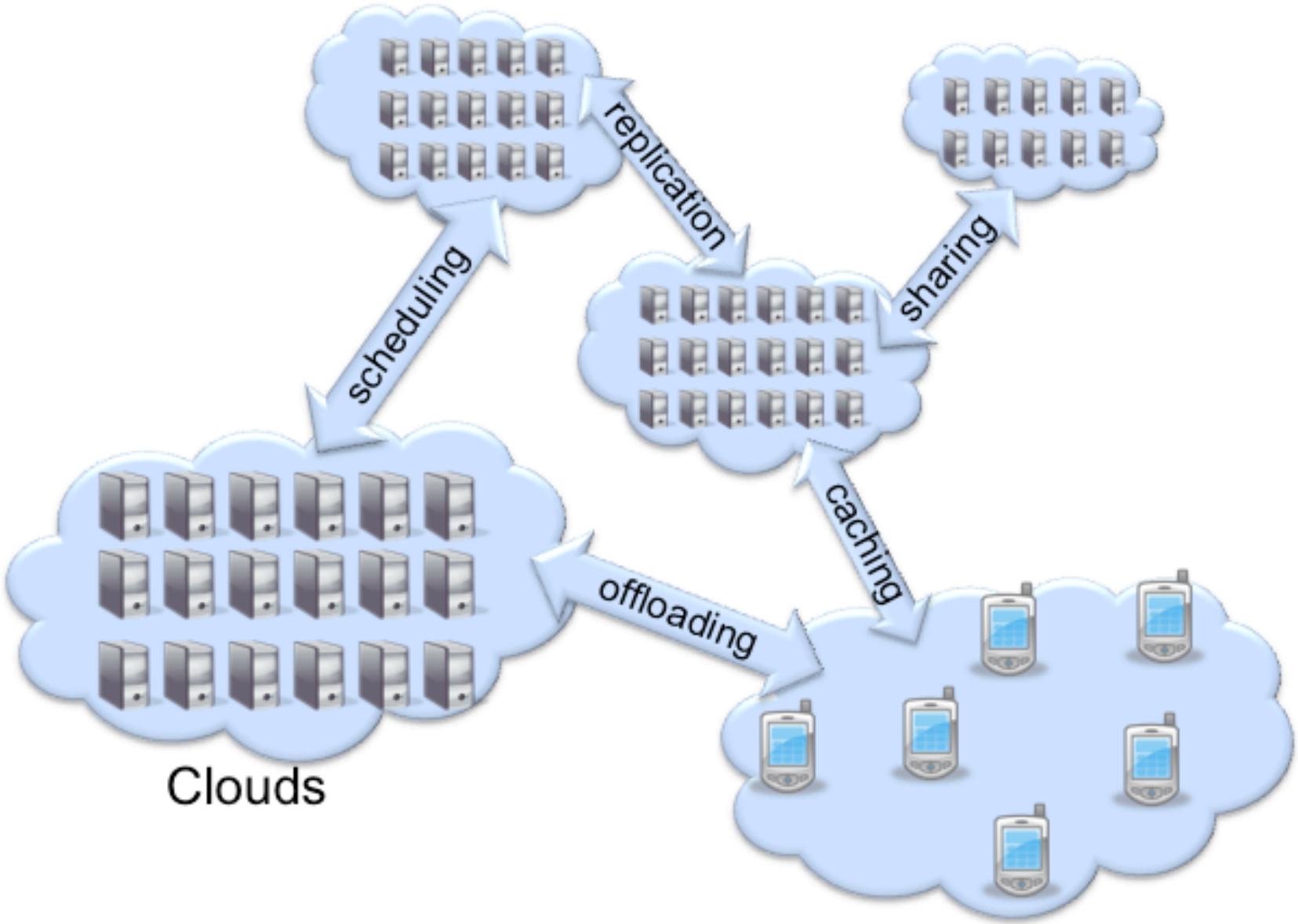
- Mobile software is a growing area
  - Ten billion downloads from Apple AppStore
  - Development processes, tools, APIs are crucial for the ecosystem
  - Integration with Web resources
- Key applications
  - Voice
  - Multimedia
  - Messaging
  - Web sites, mashups, services
  - Location-based services
- Forthcoming features
  - Context-awareness, adaptability, smart spaces

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.
  - Emergence of app stores.
  - Versatile devices: smartphones, pads.
  - Cloud-assisted applications with social networks.

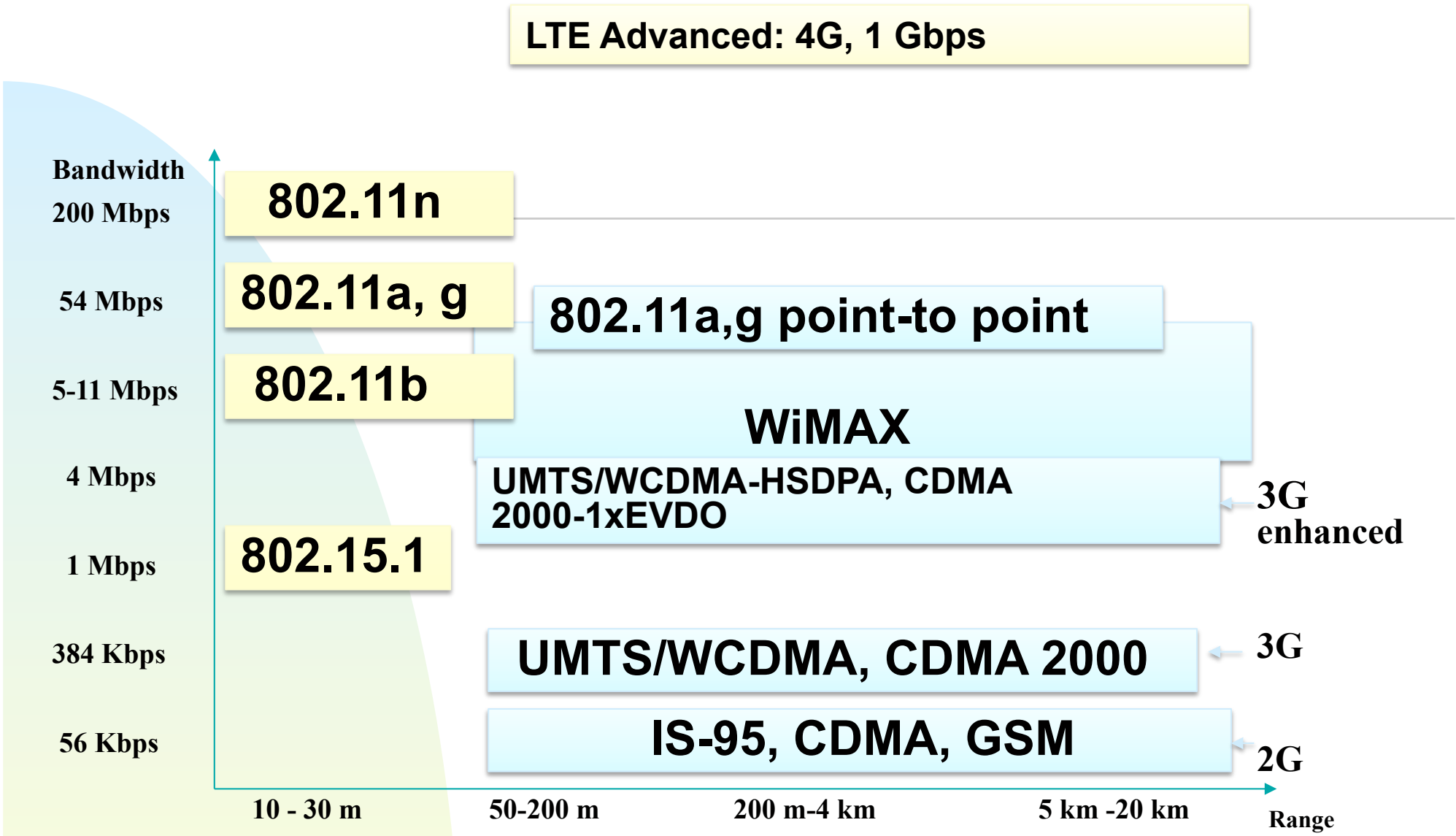

revenue
Social sites, media portals
On-demand and streaming video
Advanced browsers
Stores and Web pages
AppStore
Portals
Full music and video streaming
SMS ringtones, logos
WAP Ringtones
Full music streaming
Music downloads
Music clips
Master tones
Monophonic
Polyphonic
time


End Users
Service Cloud
Service Delivery Platform
Networks
End Users
Developers & Services
Secure APIs
Support Functions
Adaptation Layer
MSS
SMSC
HLR
...
LCS
IMS
SMSC
HLR
...
LCS
...


replication
scheduling
sharing
caching
offloading
Clouds

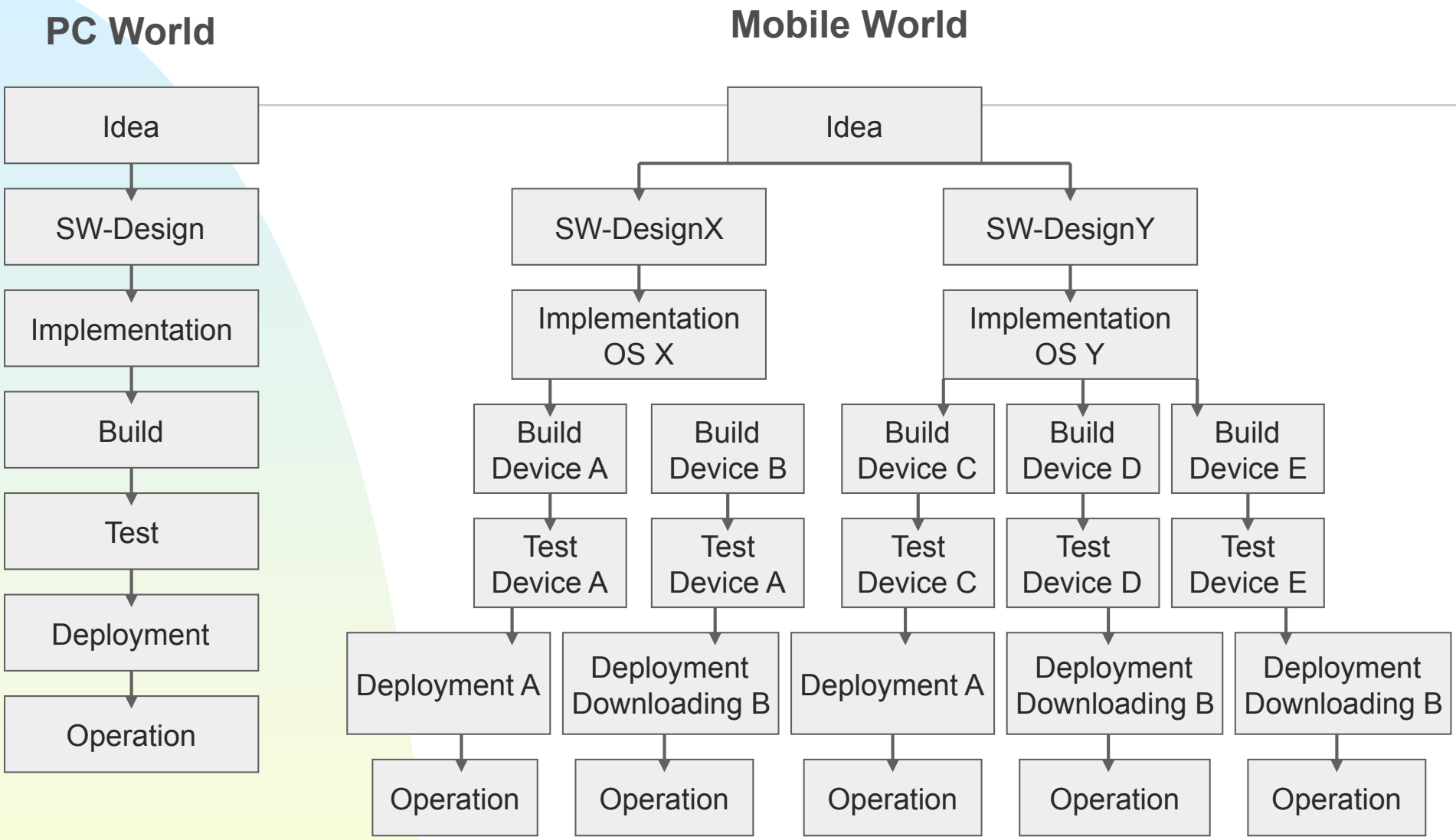
Wireless Technologies

- 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


LTE Advanced: 4G, 1 Gbps
Bandwidth
200 Mbps
802.11n
54 Mbps
802.11a, g
802.11a,g point-to point
5-11 Mbps
802.11b
WiMAX
4 Mbps
UMTS/WCDMA-HSDPA, CDMA 2000-1xEVDO
3G enhanced
1 Mbps
802.15.1
384 Kbps
UMTS/WCDMA, CDMA 2000
3G
56 Kbps
IS-95, CDMA, GSM
2G
10 - 30 m
50-200 m
200 m-4 km
5 km -20 km
Range

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

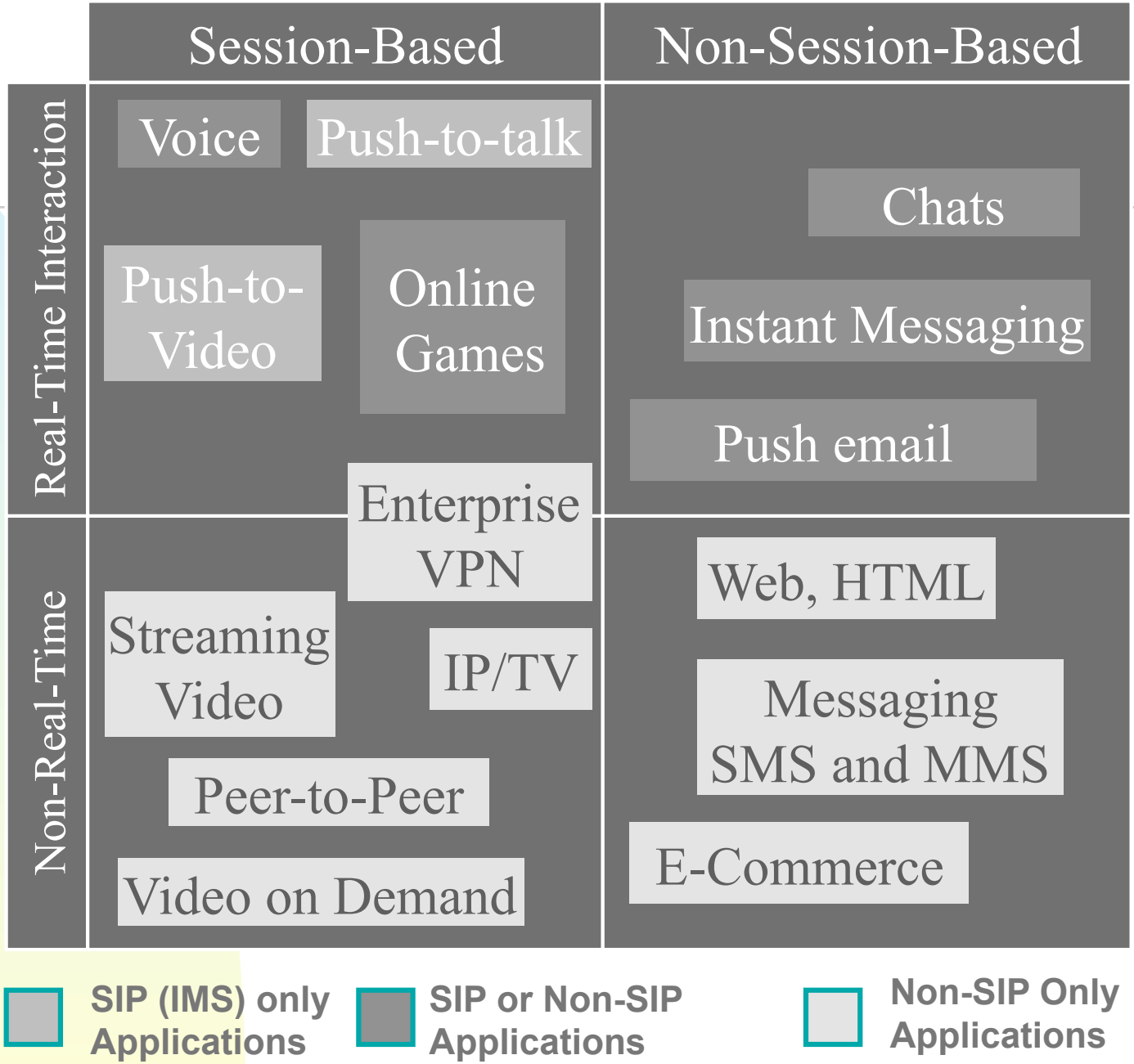

PC World
Idea
SW-Design
Implementation
Build
Test
Deployment
Operation
Mobile World
Idea
SW-DesignX
SW-DesignY
Implementation OS X
Implementation OS Y
Build Device A
Build Device B
Build Device C
Build Device D
Build Device E
Test Device A
Test Device A
Test Device C
Test Device D
Test Device E
Deployment A
Deployment Downloading B
Deployment A
Deployment Downloading B
Deployment Downloading B
Operation
Operation
Operation
Operation
Operation

IMS Service Approach

- Stovepipe and IMS service models
  - Stovepipe has separate service stacks for particular services
  - IMS service model has common layers
- The **IP Multimedia Subsystem** Provides Multimedia Services Across Networks (fixed & mobile), such as:
  - Instant Messaging, Video Sharing, Push-To-Talk, Gaming, Video Conferencing
- IMS Uses SIP protocol To Setup Multimedia Sessions Over IP Network
- SIP is a signalling protocol to:
  - Locate user given SIP Universal Resource Identifier (URL) (e.g., sip:jane@isp.com)
  - Set up session and negotiate its parameters

Network centric mobile application types

- Streaming Media
  - high jitter, low throughput
  - buffering, layered encoding
- Mobile Commerce
  - high latency, security
  - adaptive design, minimized comms.
- Pervasive Gaming
  - latency variations system
  - specific timeout values
- Web Browsing
  - low throughput, high load
  - phone caching,backoff algorithm


Session-Based
Non-Session-Based
Real-Time Interaction
Non-Real-Time
Voice
Push-to-talk
Chats
Push-to-Video
Online Games
Instant Messaging
Push email
Enterprise VPN
Web, HTML
Streaming Video
IP/TV
Messaging SMS and MMS
Peer-to-Peer
E-Commerce
Video on Demand
SIP (IMS) only Applications
SIP or Non-SIP Applications
Non-SIP Only Applications

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, 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

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

	Android Linux	Blackberry OS 5.0	iPhone OS	Java ME MIDP	Kindle SDK	MeeGo Linux	HP WebOS Linux	Symbian Series 60	Windows Mobile .NET and Windows Phone 7
Development	Java, native code with JNI and C/C++	Java MIDP, Blackberry APIs	Objective-C	Java ME	Java, Personal Basis Profile	C/C++, Qt APIs, various	Applications with Web tech. (HTML 5), C/C++	C++, Qt, Python, various	C# and .NET, various
Network scanning	Yes	Yes (hotspot API)	No	No	No	Yes	Limited (Web apps)	Limited	Yes
Network interface control	Limited	Limited (hotspot API)	No	No	No	Yes	Limited (Web apps)	Yes	Yes
Background processing	Yes (services)	Yes	No (Yes for 4.0)	Yes (multi-tasking support in MIDP 3.0)	No	Yes	Yes	Yes	Yes, not supported for third party applications in WP7
Energy and power monitoring and control	Yes	Limited (battery info)	Monitoring since 3.0	No	No	Yes	Yes (battery status, inform duration of activity)	Yes	Yes
Memory management	Yes	Yes (low-memory events)	Yes	Limited	Limited	Yes	Yes (no for Web apps)	Yes	Yes
Persistent storage	Yes	Yes	Yes	Limited, exension	Limited secure storage	Yes	Yes (HTML 5 storage)	Yes	Yes
Location information	Yes	Yes	Yes	Extension	No	Yes	Yes	Yes	Yes
HTML 5	Yes, support depends on version	Yes, support depends on version	Yes	N/A	N/A	Depends on WebKit version	Yes	No (Widgets and Javascript API)	No
SIP API support	Limited	No	Limited	Extension	No	Yes	No	Yes	No
Open Source	Yes	No	No	No	No	Yes	No (some parts are Open Source)	Yes	No
3rd party application installation	Certificate, Android store	Certificate	Certificate, Apple AppStore	Certificate	Kindle DRM	Certificate	Certificate	Certificate	Certificate, app store (WP7)
Level of fragmentation	Some fragmentation	Minor fragmentation	Minor fragmentation	Fragmented	Not fragmented	Not fragmented	Not fragmented	Some fragmentation	Some fragmentation

Application Trends

- WP7
  - Native apps, cloud integration
- iOS
  - Native apps
  - Potential for Web apps
- Android
  - Native apps
- WebOS
  - Web apps with HTML5
- Blackberry
  - Native and Web apps

HTML5

- HTML 5 is the next version of HTML
  - The first public working draft of the specification available in January 2008 and completion expected around 2012
- Improvements
  - Web Socket API, advanced forms, offline application API, and client-side persistent storage (key/value and SQL).
- HTML 5 support divides the platforms.
  - The iPhone platform has a very good support for HTML 5.

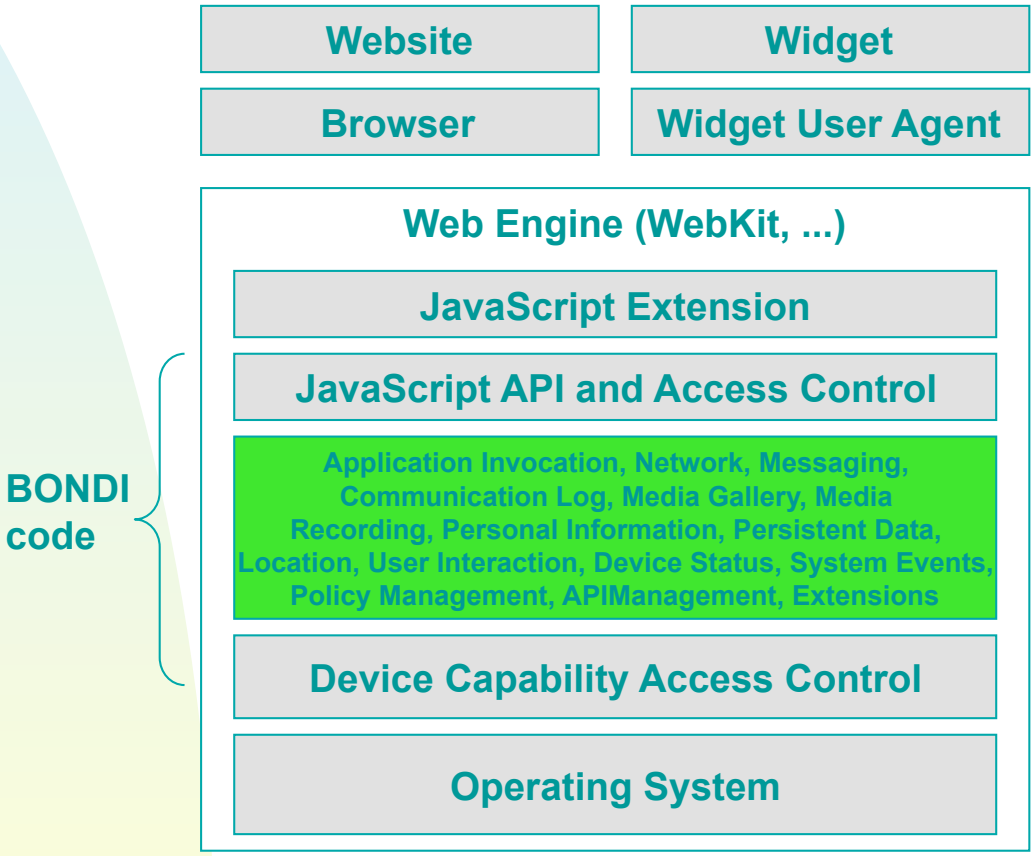
Browser support

Platform	Browser	Rendering engine	Flash	Widget engine	Comments
Android	Android	WebKit	Yes (for Android 2.1)	No	
Apple iPhone	Apple iPhone Safari	WebKit	No	Yes	
BlackBerry	BlackBerry	Mango	No	Yes	Proprietary push technology
Opera Mobile	Symbian S60, Windows Mobile	Presto	Yes	Yes	Opera Mini supports other browser platforms using a proprietary proxy
Symbian S60	Symbian S60	WebKit	Yes	Yes	
Windows Mobile	Internet Explorer	Trident/ MSHTML	Yes	No	Silverlight

JavaScript access

- The Open Mobile Terminal Platform (OMTP) group defines requirements and specifications that aim towards simpler and more interoperable mobile APIs
- BONDI defines requirements governing Device Capability access by JavaScript APIs to promote interoperability and security of implementations
- The recent 1.1 release of BONDI is compliant with the W3C Widgets: Packaging and Compliance specification

BONDI Architecture



System bus

- An asynchronous system-wide event bus is a basic solution for interconnecting various on-device components
- There is no single standard for this.
  - Android and Java ME use Java-specific events (Android Intent filtering)
  - MeeGo uses D-BUS
  - Palm's WebOS W3C Events
- One particular trend is to utilize URI-based conventions for naming system resources and services.
  - This is extensively used in the Nokia Platform Services, WebOS and the BONDI architecture.

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 (Java)
  - MeeGo, Qt, Symbian allow low-level access
  - SQLite, OpenGL ES, XML, Web services and REST
  - Browsers (HTML5) and Flash

Mobile Development

Starting points

- Interfaces to mobile devices, to mobile operators
- Simple design, rapid prototyping
- Interoperable formats: Web technology and XML
- Backend in the cloud
- Can leverage WebKit on most smartphones
  - Looks like an app but is a browser
- Use Web and HTTP to bridge different tech domains (Android, iPhone, Qt)

Cloud development

- Combine private and public clouds
- For example:
  - Local Eucalyptus cluster
  - Use Amazon as the public cloud
- Offloading

Telco viewpoint

- GSMA is standardizing the One API specifications
  - Initially Messaging and Location APIs
  - Charging, Data connection profile and User profile APIs will be next
  - Reference implementations available both in lightweight Web Services and with REST
- Open existing network infrastructure
- Broker useful for multi-operator deployment (bridging APIs across operators)

Recent trends

- Cloud and cloud-assisted services
- Push communications
- Stores and advertising
- Social networks
- Delay tolerant functions

Sw design for beyond Web apps

- Use MVC pattern to separate concerns
- Rely on brokers and proxies in the cloud
- Cloud to device messaging
- Event-based sw design is good for reactive apps
  - Context-awareness
  - UIs
- Fragmentation
  - C++, Java, Objective C
- Scripting for portability
  - Port the interpreter / VM
  - License issues

General sw issues

- Use native methods/functions
- Avoid unnecessary creation of objects
  - Pools
- Avoid floats / enums
- Avoid unnecessary updates (location, status, …)
- Avoid unnecessary WakeLocks (these keep the display on)
  - Use the minimum possible
- Keep sw responsive
  - Events, timers
- Engineering for the exceptions

User experience

- User experience
  - Keep it consistent, different modes
- Do not make assumptions about the screen size
  - Use vectors / high res graphics
- Use assertions for features
  - API availability
  - Check for GPS / compass / ..

Application stores

- App stores
  - Apple, Nokia, Android, WP7, …
  - In-app purchases
- Searching, purchasing, advertising, …
- How to do software updates
- How to support community buildup
- Push notifications
  - Dedicated push servers
  - Control plane
- Inter-app communication is still in early phases

Sensors

- The number of sensors will increase dramatically
- Innovative new applications
  - Pulse monitor, augmented reality, …
- Nokia N8 sensors:
  - Accelerometer Double Tap
    Accelerometer XYZ
    Ambient Light
    Magnetic North
    Magnetometer XYZ
    Orientation
    Proximity Monitor
    Rotation

Energy measurement

View	Unit and notes
Power	Watts, indicates backlight usage with a different background colour
Current	mA
Processor	Percentage
RAM Memory	MiB
Network Usage	kB/s, upload and download
WLAN	dBm
Signal Levels	dBm
Battery level	mAh. We have not seen values in this field.
Voltage	Volts

- Nokia Energy Profiler
- Other tools

- SW-based measurement is relatively easy, can be combined with hw-based measurements

Offloading

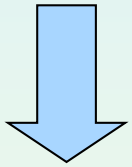
- Dessy: example of computational offloading
- IETF work on network controlled offloading (vertical handoffs)

Mobile Search: Dessy

- 2007: Search and Synchronization Prototype "Dessy" for Java Personal Profile (PDAs)
  - Finds local files by their content and metadata
  - Like *Google Desktop* for phones, but also syncs files between devices
- 2009: Dessy Ported to MIDP phones (Most Java-enabled phones)
- 2011: Dessy will allow offloading indexing to the cloud

Indexing: Desktop Search Basics

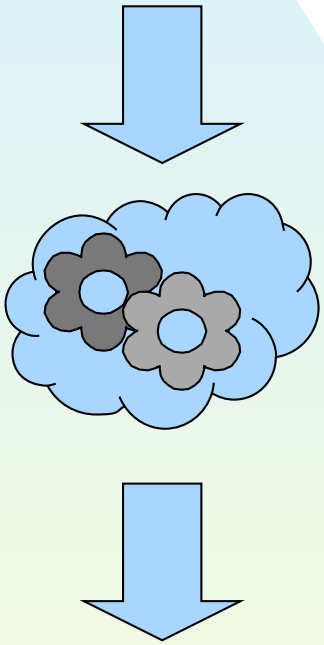
percom



P1.pdf, p65.doc, notes.txt

- Files are collections of words
- To find files by the words contained in them, we need to map words to collections of files that contain the word
  - This is called an *Inverted File, an Index*
- To create this index, we need to read through all the files, creating a mapping from all their words to the file identifiers
  - This is expensive CPU-wise

Offloading tasks to the Cloud



- Intuitively, CPU-intensive tasks can be offloaded
- Data transmission, waiting, and result reception energy should be smaller than execution energy on the phone
  - Then offloading is beneficial and we “win” battery life on the phone

Summary

- The offloading decision should depend on expected energy use
- High energy savings can be obtained when offloading CPU-intensive tasks
- Indexing can and should be offloaded, even on a slower network (Minimum efficiency for a 1 MB file with Dessy is 17 kB/J, offloading with 16 kB/J causes a 1.5 % energy loss)
- Future mobile platforms may use cloud services to enable fast and energy-efficient on-device search

IETF work on offloading

Conclusions

- Mobile development has changed
- Dominant platforms: Android, IOS, WP7
- Javascript/HTML5 as a promising alternative, with interfaces to device internals
- Cloud computing and mobile computing work together
- Applications partly in cloud, partly on the mobile
- Offloading as a service