Mobile Technologies

Department of Computer Science
VŠB-Technical University of Ostrava

by Roman Szturc
2006

It Started with the Telegraph

"We call the electric telegraph the most perfectinvention of modern times … as anything more perfect than this is scarcely conceivable, and we really begin to wonder what will be left for the next generation, upon which to expend the restless energies of the human mind."

— an Australian newspaper
1853

The Vision

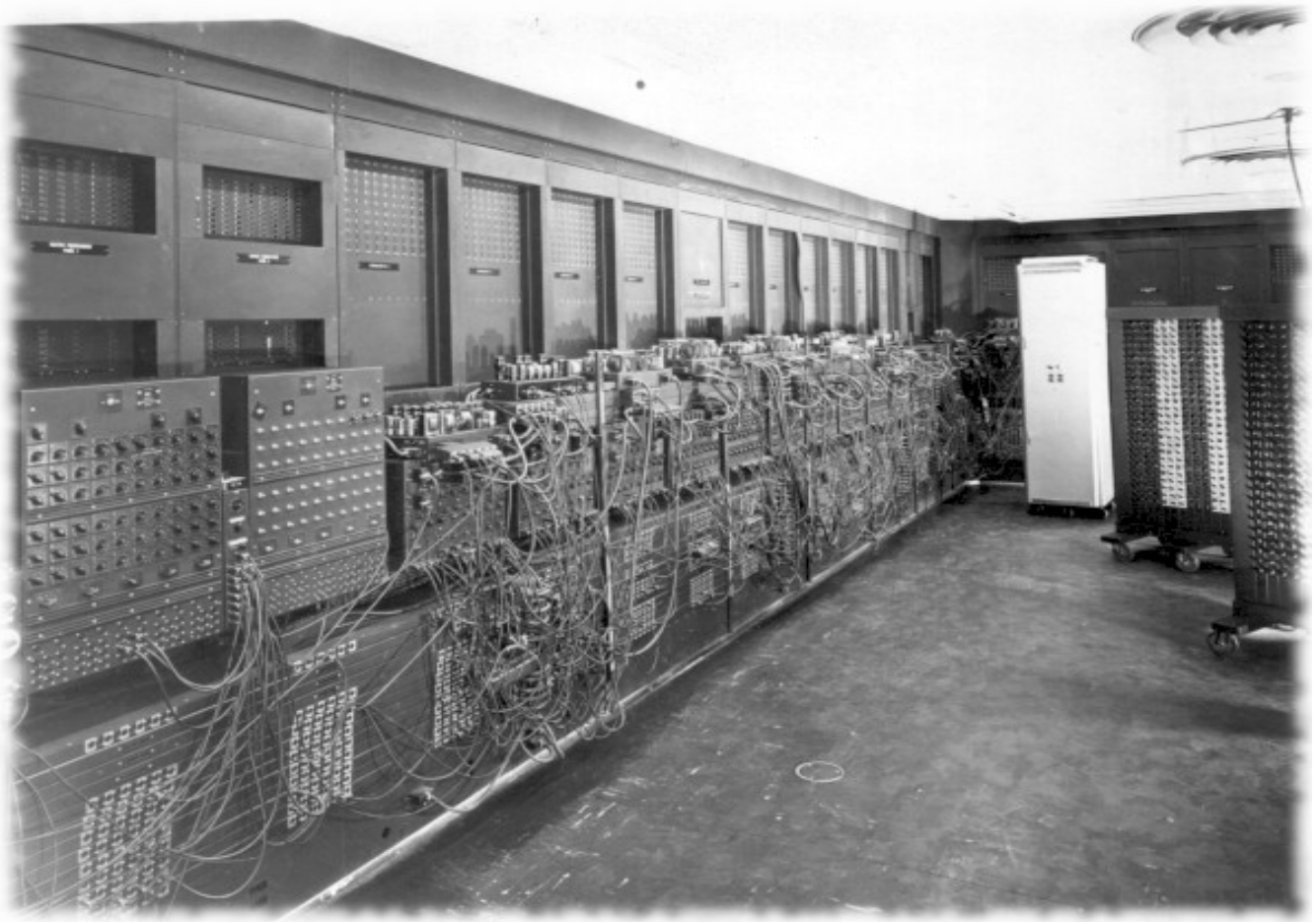
"People and their machine should be able to access information and communicate with each other easily and securely, in any medium or combination of media—voice, data, image, video, or multimedia—any time, anywhere, in a timely, cost-effective way."

— George H. Heilmeier
1992

History

History

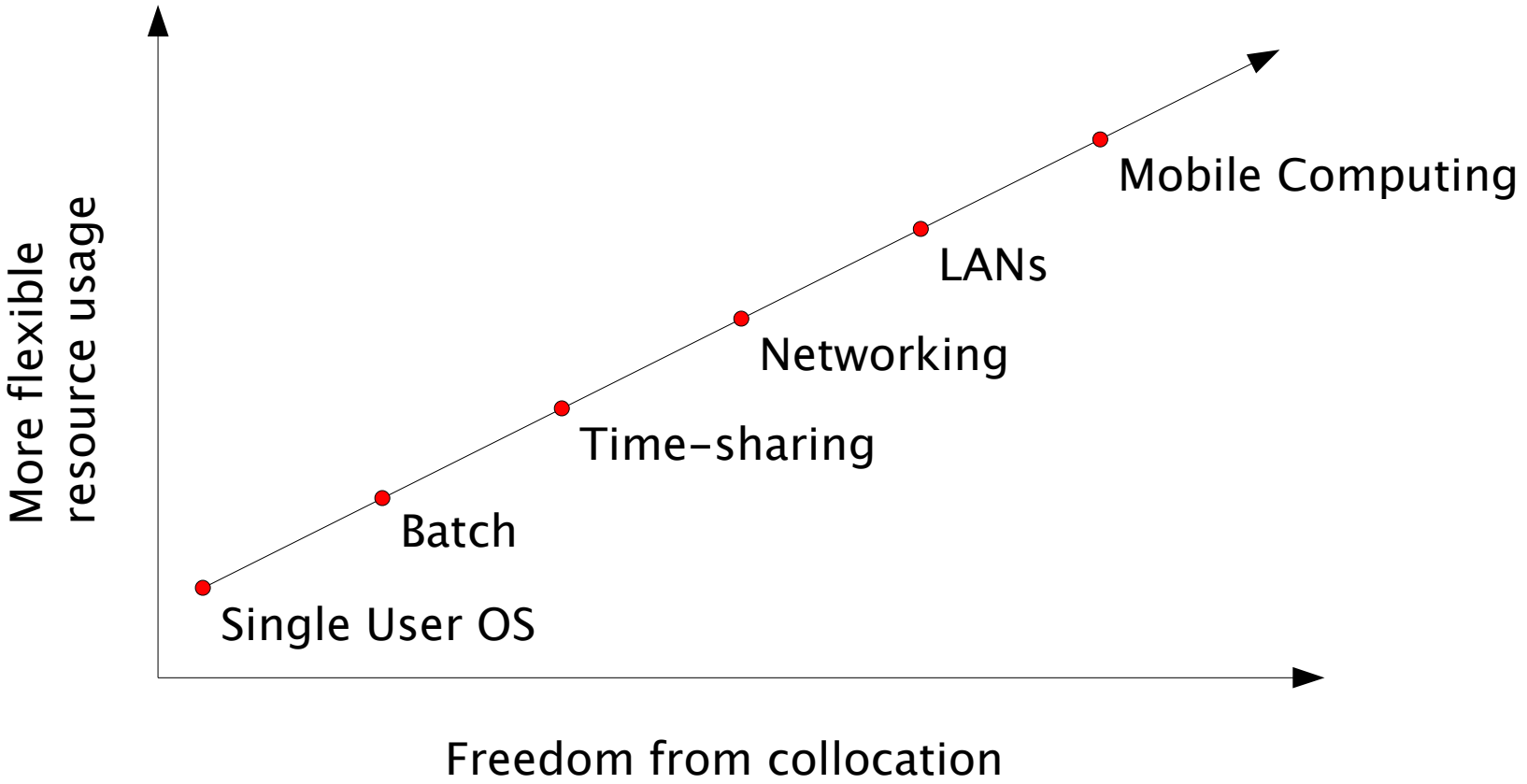
ENIAC was a monster. It weighed 27 t, was roughly 2.5 m by 1 m by 30 m and consumed 150 kW of power.



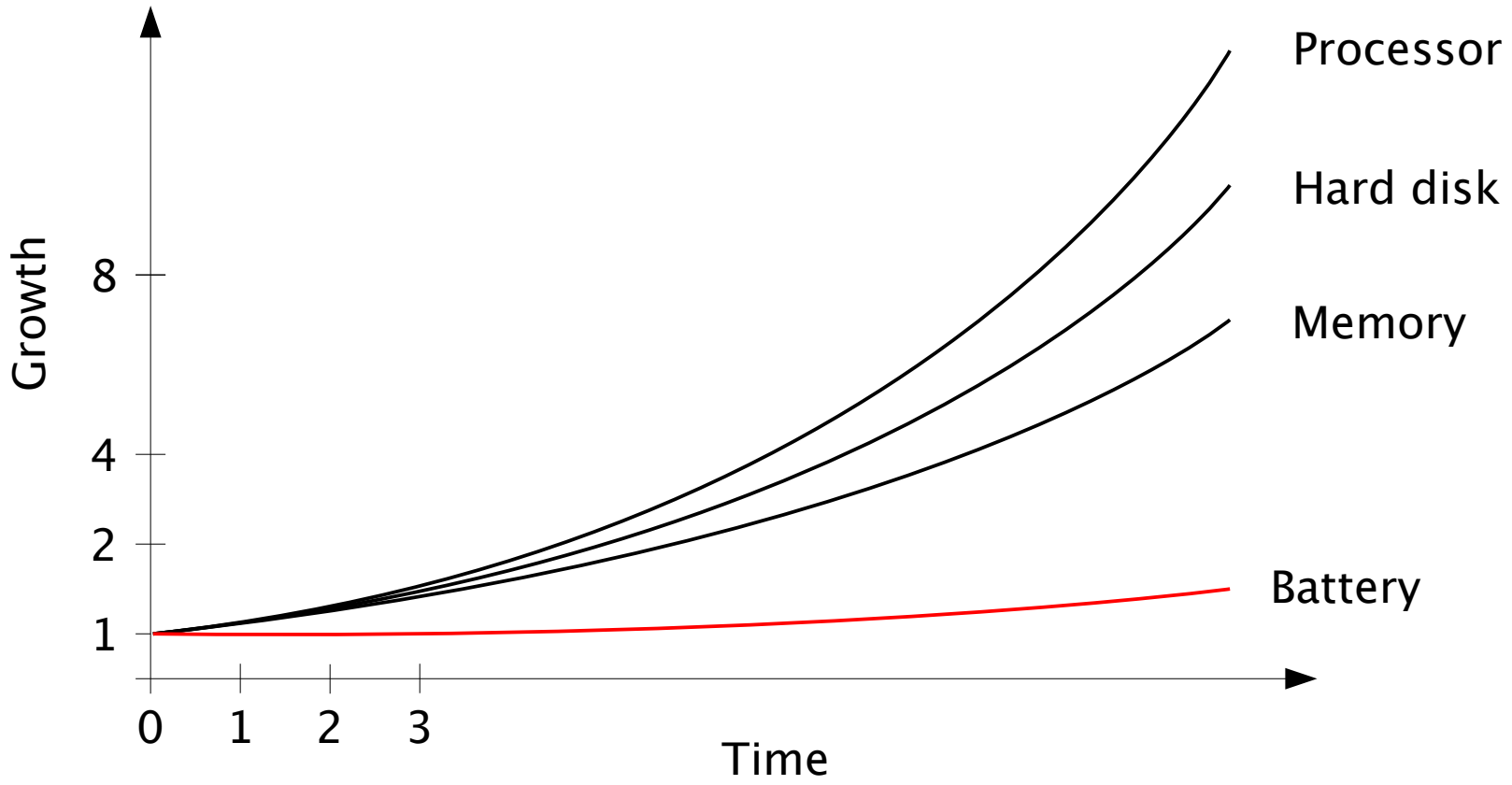
History

The first smartphone was called Simon designed by IBM in 1992. Besides a mobile phone, it also contained a calendar, address book, world clock, calculator, note pad, e-mail, and games. Customers could also use a stylus to write directly on its screen to create facsimiles and memos.

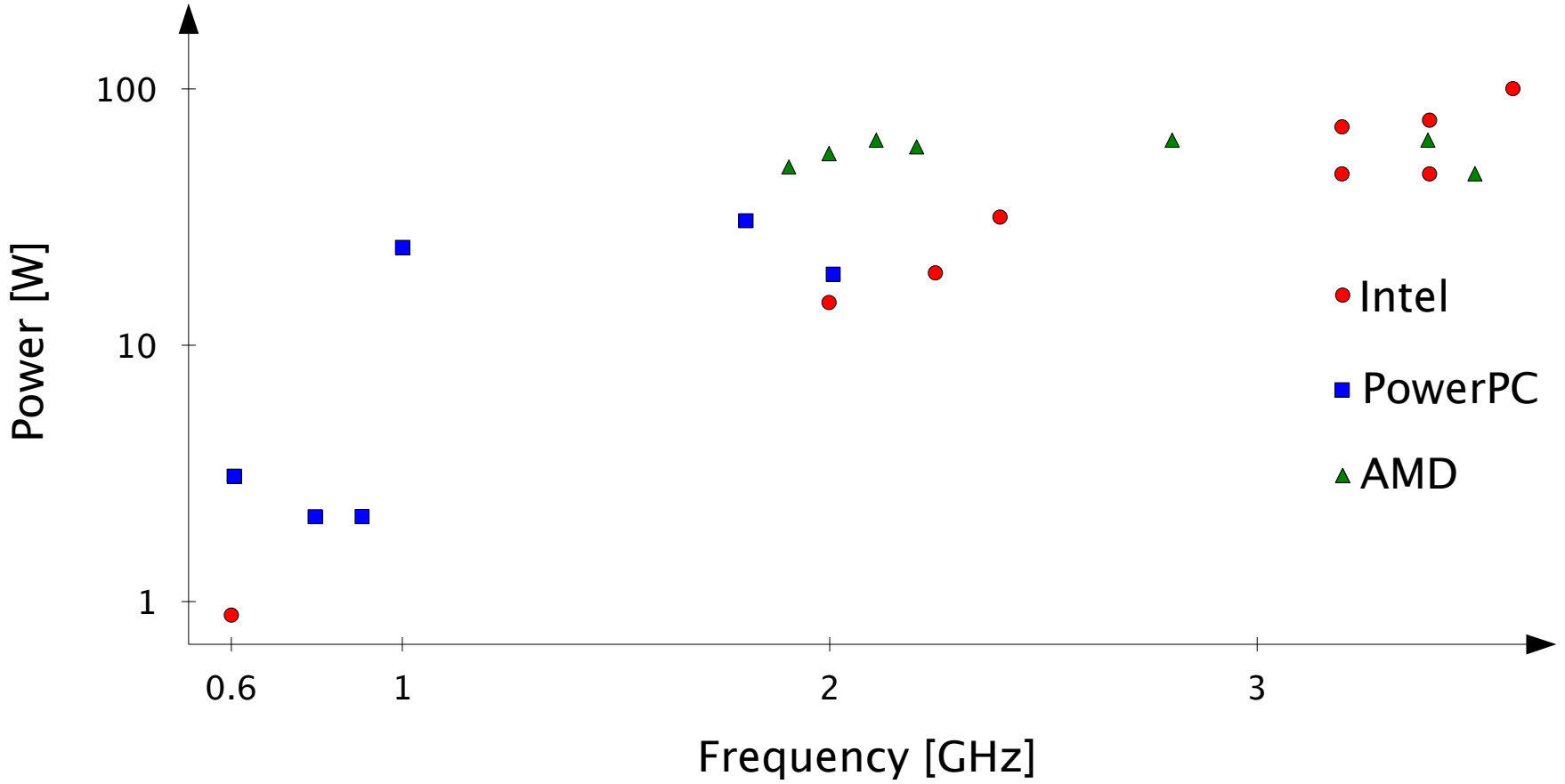
Natural Evolution of Computing



Technologies Improvement



CPU Power Dissipation



Note: The figure does not include computational performance.

Operating Systems

There are significant differences between general purpose OS and embedded ones. An embeded OS not only provides the interface between application and hardware, but often also enables the handling of tasks with real-time requirements. The main properties that make an embedded system different include:

- hardware constraints in terms of working memory, nonvolatile storage and power consumption;
- user interfaces with significantly limited capabilities;
- often application specific software and hardware is used rather than the traditional general purpose hardware and software;
- streaming media applications which requires specific real-time requirements are often the dominant type of application.

Real-Time Operating Systems

Although computational performances of general-purpose processors highly outperform the performances of DSP processors, the latter have I/O characteristics that are much better suited for media applications and therefore often imply better overall system performance. Modern mobile processors most often target a soft real-time operating system (RTOS) rather than a hard RTOS.

- For hard RTOS, the most important requirement is that throughput and computational requirements latencies are deterministic and that all deadlines are always met.
- For soft RTOS, the main issue is tailoring the operating system to the strict hardware and power requirements, while the deadlines for real-time tasks occasionally may not be met.

Compilers

In addition to the OS, utility programs play an important role in the acceptance and use of mobile processors. Utility tools are usually organized in integrated development environments, which include tools such as project builders, source level debuggers, event analyzers, performance profilers, run-time error checking tools, graphical code browsers, specialized text editors, and version control systems. The most important utility tools for mobile processors are compilers, linkers and loaders. Development of power and memory utilization sensitive compilers is still mainly in the research phase.

Operating Systems

"People and their machine should be able to access information and communicate with each other easily and securely, in any medium or combination of media—voice, data, image, video, or multimedia—any time, anywhere, in a timely, cost-effective way"

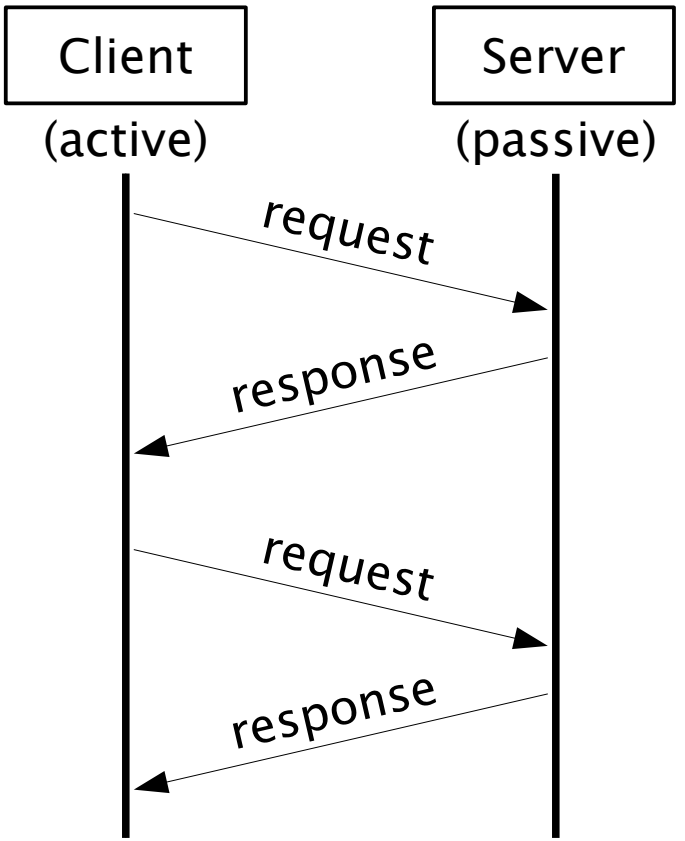
- **Symbian OS**
  http://jcp.org/en/home/index/
- **Palm OS**
  http://java.sun.com/j2me/
- **Windows CE / Windows Mobile**
  http://www.microsoft.com/windowsmobile/
- **Linux**

OS Market Shares

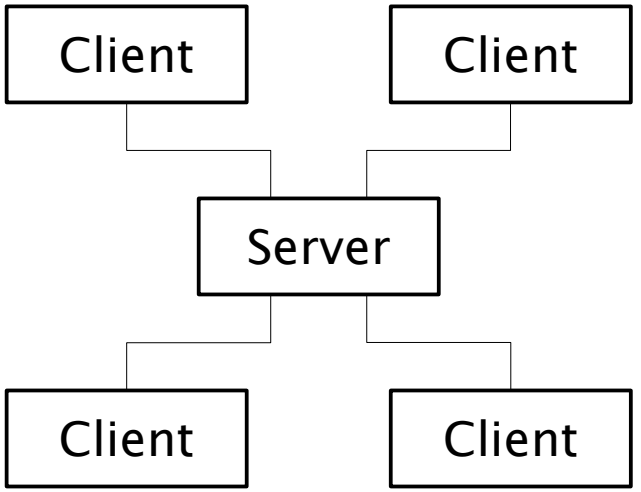
OS Vendor	Q2 2005 % share	Q2 2004 % share
Symbian	63	41
Microsoft	16	23
Palm Source	9	22
Others	12	14

Client-Server

Client-Server (C-S) is the reference model for distributed applications. The model is based on a rigid distinction of roles between the client nodes and the server nodes. The server nodes provide the services, but they are not capable of taking any initiative as they are fully reactive and they can just wait for being invocated by the client nodes. Client nodes, as opposite, concentrate all the initiative of the system: they access and use the services, but they do never provide any capability.

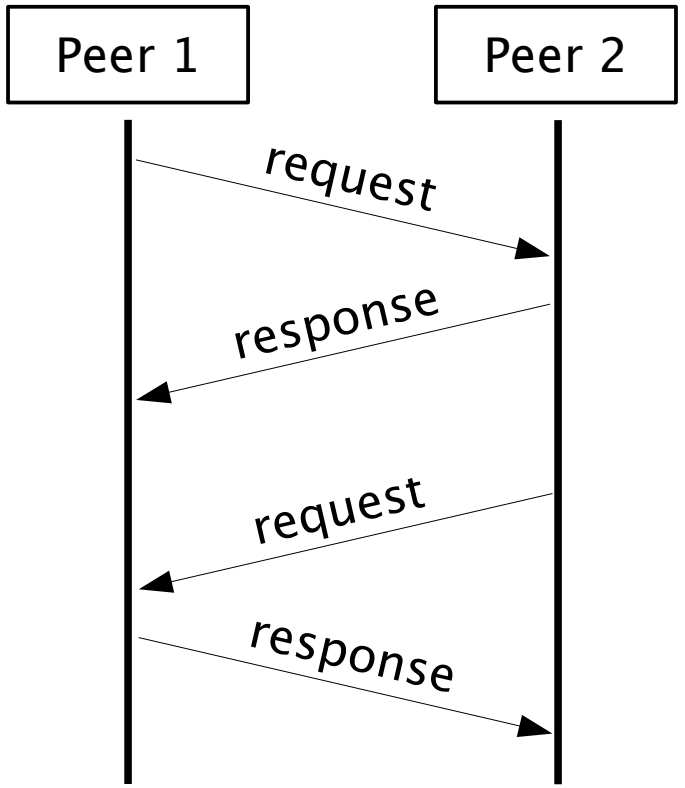


Client-Server Architecture



Clients can appear and disappear at any time; generally, they have dynamic addresses, while servers must typically provide some guarantees of stability and generally listen to a well-known and static address. Clients communicate with the servers, but they cannot communicate with other clients. On the other hand, server cannot communicate with their clients until the clients have taken the initiative and decided to activate a communication session with the server.

Peer-to-Peer

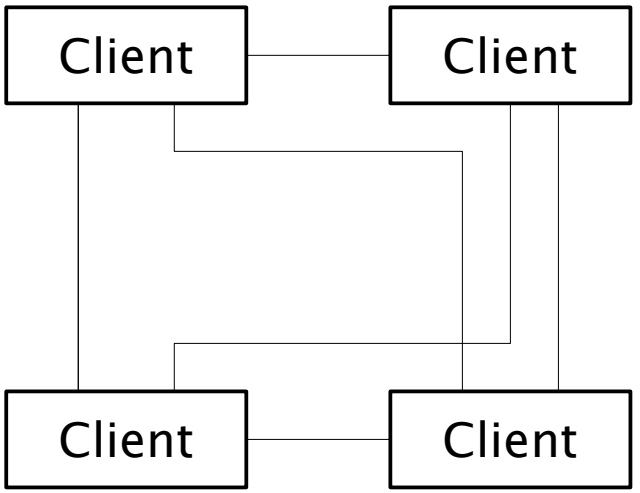


In the peer-to-peer (P2P) model there is no distinction of roles and each peer is ca-pable is a mix of initiative and capabili-ty:

- each node can initiate the communication, be subject or object of a request;
- the application logics is no more concentrated on the server but distributed between all the peers;
- each node is capable of discover each other, it can enter, join or leave the network anywhere anytime.

Pure Peer-to-Peer Architecture

A pure P2P network is fully decentralized and the peers are fully autonomous. The absence of any reference node makes more difficult to maintain the coherence of the network and the discovery of the peers, with a complexity and bandwidth that tends to grow exponentially with the number of nodes. Also security is quite demanding as each node is entitled to join the network without any control mechanism.



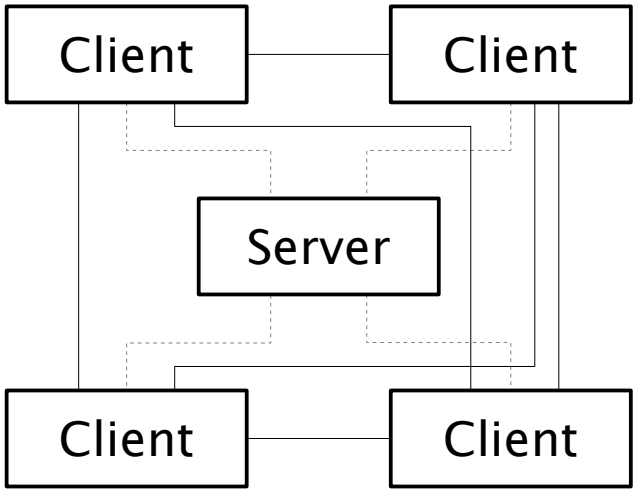
Node Discovery

In the C-S systems, clients must know their servers but they do not need to know other clients. In P2P systems, who-knows-whom is fully arbitrary and the system must provide proper services that allow peers to enter, join, or leave the network at any time as well as to search and discover other peers. These services are usually the white and yellow page mechanisms that allow publishing and discovering the features and the services offered by a peer.

- **White pages**
  A section of a directory that alphabetically lists the names of people. For example Knowbot, Netfind, whois, X.500 and finger.
- **Yellow pages**
  A section of a directory that lists businesses, services, or products alphabetically according to field. For example NIS, UDDI, ebXML.

Hybrid Peer-to-Peer Architecture

A hybrid architectures, instead, are based on a special node that provides a service that simplifies the look-up and discovery of the active peers, their list of capabilities, and their list of provided services. These types of networks, usually, generate less traffic and are more secure as they tend to require also the registration and authentication of the peers. On the other hand, their functioning depends on the availability of the index nodes that might become a central point of failure and attack.



Mobility

Mobility in distributed systems offers some benefits.

- **Benefits**
  Mobility in distributed systems offers some benefits, such as enabling movement toward desired resources, the use of resources while moving and improved flexibility.
- **Deployment Challenges**
  The major challenges in deploying mobility are the lack of applications, widespread infrastructure and global scalability.
- **Technical Issues**
  Finally, there are technical issues that need to be resolved while supporting mobility: naming, locating, security, reliability and disconnected behavior.

Moving Towards a Desired Resource

By local accessing data or other resources, performance can be dramatically improved. Sometimes, only local access is possible because of the semantics or security requirements.

- **Process Migration**
  A process might move toward an underloaded computer, a specific database, or some rare hardware device.
- **Mobile Agents**
  Agents migrate toward source of information, or a computer that they manage.

Using Computer Resources while Moving

It allows users to take a computer away from its usual workplace and still be productive. Process of migration and mobile agents provide additional support, by enabling movement of the programming environment and applications along with the mobile computer.

Flexibility

Flexibility can mean easier reconfiguration or improved reliability.

- **Easier Configuration**
  If a wireless phone cannot connect from a specific area, moving to a new area may overcome some natural obstacles that prevented original connection.
- **Improved Reliability**
  If a computer has a partial failure, or if it is about to shut down, a process can migrate to another computer and continue to execute there.

The Lack of Applications and Infrastructure

The lack of applications is the biggest challenge for deploying any form of mobility.

- **Remote execution**
  This mechanism offers less performance, functionality and flexibility.
- **Reconnection of mobile devices**
  Absence of support for connecting computers while or after they migrate, or for programming infrastructure that will allow migrated processes or agents to visit remote computers.

Security

Security problems that need to be addressed in mobile environments are similar to those addressed by classic security: authentication, authorization, integrity, privacy, prevention or the denial of service and non-repudiation.

- **Unknown environment**
  It is relatively easy to access data sent from a mobile computer or to interpose between a remote agent and the source of its incoming messages.
- **Non-repudation**
  It is harder to support non-repudation for mobile entities, especially those that move frequently and are short lived, than it is for static, long-lived objects.

Reliability and Disconnection

Caching and hoarding
Mobile computers typically cache files. Mobile processes cache code and data from their address space and open data files.

Naming and Locating

Without the support for locating a mobile object, other functionality, such as communications or control, is not achievable. When a mobile entity is moved from its original location and then reconnected at some new location, communication channels must be reestablished with other entities.

Process Migration

In the late 1970s, people commonly moved data from one computer to another by copying it onto magnetic tapes. Since the early 1980s, problem of process migration is researched.

The basic idea is to move an executing process from one node to another. Migration provides several benefits:

- **Load distribution**
  This allows a heavily loaded node to migrate away some subset of its active processes to a less loaded node.
- **Fault resilience**
  Migration improves the ability of a process to overcome localized errors or faults.
- **Data access locality**
  Locality of reference can provide significant performance gains.

Process Migration

Despite the promises of migration, most successful implementations face several chanalges that work against their widespread acceptance and popularity:

- **Social issues**
  Given an ability to move a process from one node to another requires a social contract between owners of nodes.
- **Code complexity**
  Migration mechanisms tend to be complex and difficult to maintain.
- **Naming migrated processes**
  Process names have to maintain existing single node semantics and yet uniquely identify each migrated process.
- **Locating migrated processes**
  Process migration has to track and maintain up-to-date information about the location of migrated processes.

Process Migration

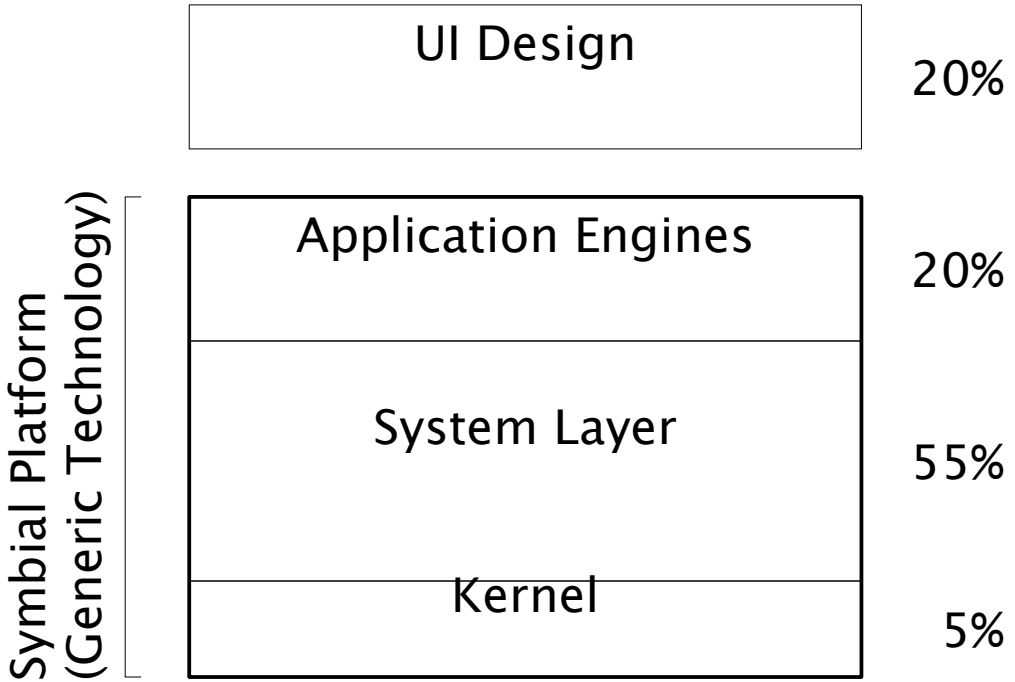
- **Controlling migrated processes**
  A proxy mechanism is used to controll migrated process. All control requests are forwarded to the process's current node by the proxy.
- **Exporting and transferring process state**
  A process has several types of data, such as user data, stack, memory-mapped files and so on. Migrating a process requires complex support for encapsulating this state.
- **File systems**
  Process migration is usually supported by distributed file systems.
- **Security**
  Single node security systems have to be extended to support distributed applications execution.

Using Computer Resources while Moving

It allows users to take a computer away from its usual workplace and still be productive. Process of migration and mobile agents provide additional support, by enabling movement of the programming environment and applications along with the mobile computer.

Main OS Layers

Themajority of the features of the OS are screen and user input independent. All these features amount to around 80% of the OS and are known as Symbian Platform or Generic Technology (GT). Symbian is wholly responsible for these parts of the system.



GUI Designs

Series 60

- 176 x 208 pixel GUI,
- one handed operation,
- e.g. Nokia 6630.

UIQ

- 208 x 320 pixel GUI,
- pen based + keyboard,
- e.g. Sony Ericsson P910.

Series 80

- 640 x 320 pixel GUI,
- qwerty keyboard,
- e.g. Nokia 9300.

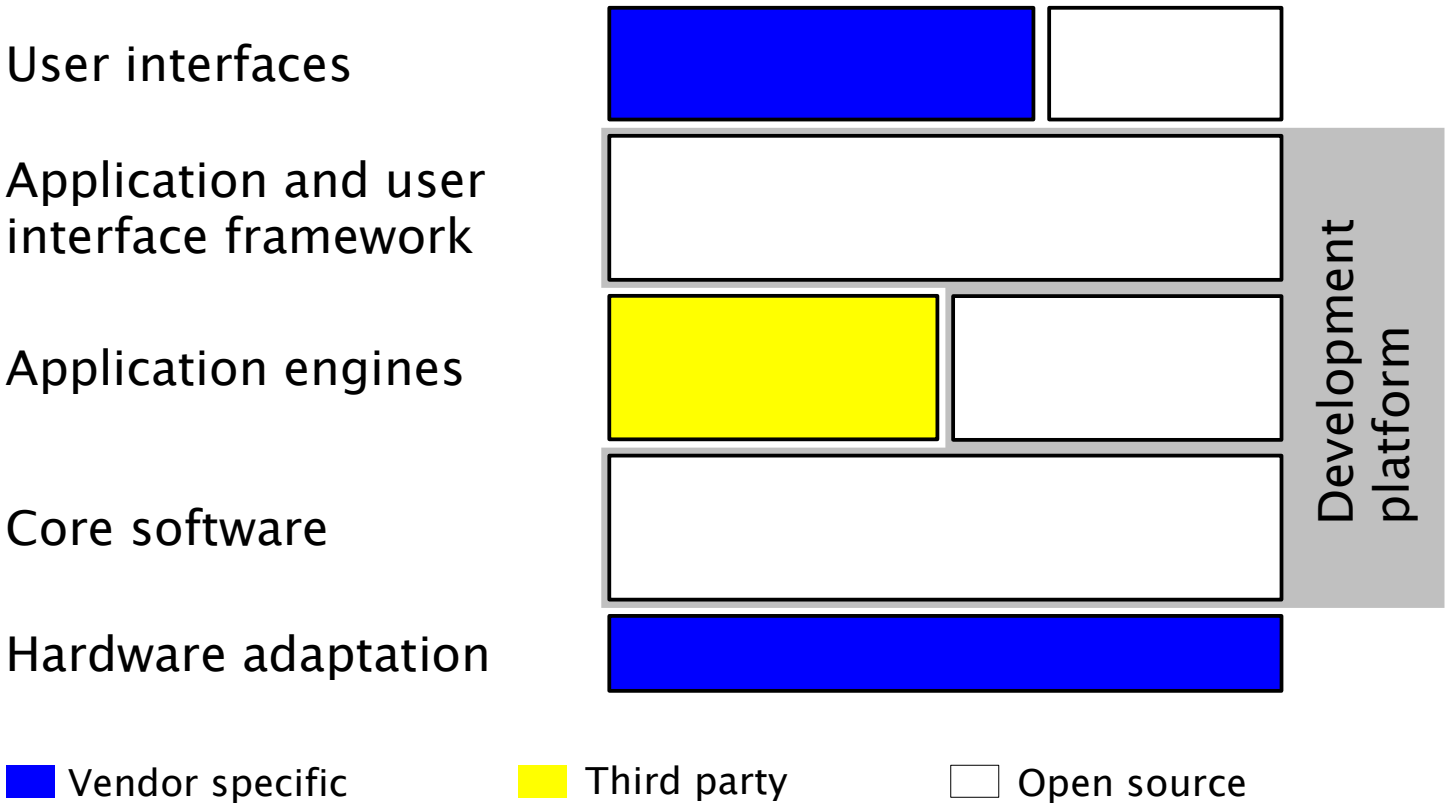
Others

Symbian Platform Evolution

	Series 60	Series 80	QUI	Others
v6.0		Nokia 9210		
v6.1	Nokia 7650			
v7.0			Sony Ericsson P910	Nokia 7710
v7.0s	Nokia 6600	Nokia 9300		
v8.0	Nokia 6630			
v9.1	Nokia E60			

Linux-Based OS

xxx



The Vision

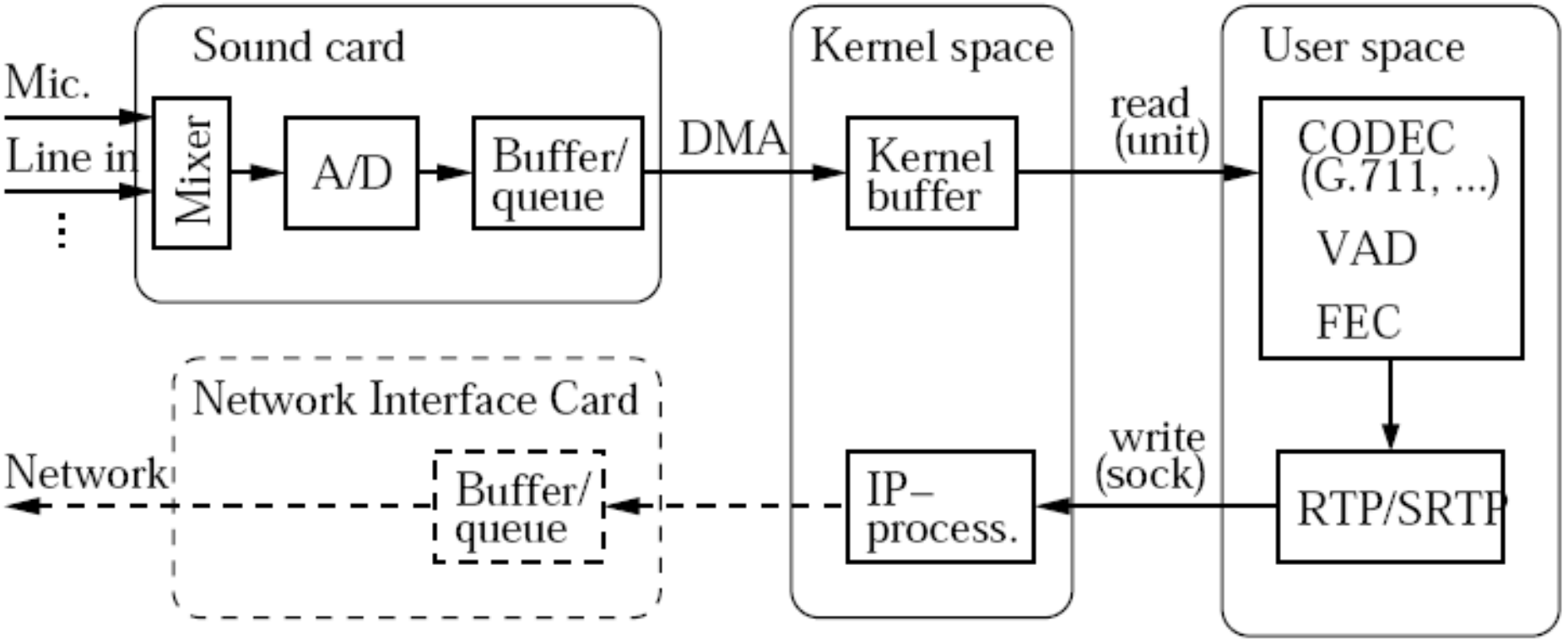


Figure 2.1: Generating and transmitting audio data at the sender.