Project no. IST-033576

XtreemOS

Integrated Project
BUILDING AND PROMOTING A LINUX-BASED OPERATING SYSTEM TO SUPPORT VIRTUAL ORGANIZATIONS FOR NEXT GENERATION GRIDS

Design of a Basic Linux Version for Mobile Devices
D2.3.3

Due date of deliverable: November 30th, 2007
Actual submission date: December 19th, 2007

Start date of project: June 1^{st} 2006

Type: Deliverable
WP number: WP2.3
Task number: T2.3.3

Responsible institution: Telefónica I+D
Editor & and editor's address: Luis Pablo Prieto
Telefónica I+D
Parque Tecnológico de Boecillo
47151 Boecillo (Valladolid)
SPAIN

Version 1.2 / Last edited by Luis Pablo Prieto / December 19th, 2007

Project co-funded by the European Commission within the Sixth Framework Programme		
Dissemination Level		
PU	Public	√
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Revision history:

Version	Date	Authors	Institution	Section affected, comments
0.1	2/8/07	Luis Pablo Prieto	TID	Initial template
0.2	17/8/07	TID team	TID	Added bibliography and section about distro election
0.3	8/10/07	TID team	TID	Added architecture chapter
0.4	16/10/07	TID team	TID	Added mobility enhancements chapter
0.5	17/10/07	Yvon Jégou	INRIA	Added VO support chapter
0.6	17/10/07	Luis Pablo Prieto	TID	Comments and modifications to the overall document
0.7	19/10/07	Yvon Jégou	INRIA	Updated VO support chapter
0.8	29/10/07	TID Team	TID	Added chapter about mobile Linux initiatives
0.81	30/10/07	Luis Pablo Prieto	TID	Reorganization of the content
0.9	31/10/07	Luis Pablo Prieto	TID	Introduction, future work and glossary, plus minor modifications
0.91	02/11/07	Luis Pablo Prieto	TID	Removed most comments and added figures. Ready for reviewing
1.0	30/11/07	TID team	TID	Added appendix, executive summary and several corrections
1.1	07/12/07	TID team	TID	Corrections from the first reviewers
1.2	19/12/07	TID team	TID	Corrections from the second reviewers

Reviewers:
Massimo Coppola (CNR), Jörg Domaschka (ULM)

Tasks related to this deliverable:

Task No.	Task description	Partners involved°
T2.3.3	Design of a basic Linux version for mobile devices (Linux-XOS for MD/PDA)	TID*, INRIA

°This task list may not be equivalent to the list of partners contributing as authors to the deliverable
*Task leader

Executive Summary

XtreemOS aims at embedding grid functionalities into the Linux operating system, in order to harness the heterogeneous computing resources available today, from clusters to mobile devices. XtreemOS-MD is the result of the efforts in adding these grid functionalities to mobile devices (first to PDAs, and later, also to mobile phones).

Previous documents have already pointed out the requirements that a *mobile* grid operating system must fulfill. But, most important of all, we first have to make Linux OS "aware" of *virtual organizations* (VOs) and their users. This document concentrates on defining which modifications must be made in the mobile operating system so that it adequately supports XtreemOS grid services.

After a thorough analysis of current mobile Linux distributions, we reached the conclusion that Ångström is the most suited distribution to use as a base for the PDA version of XtreemOS-MD, since it is an active community-based effort using state-of-the-art software packages and technologies. However, the selection of this distribution should not preclude XtreemOS-specific packages from being integrated into any other mobile Linux distribution.

In the process of designing this grid operating system for mobile devices, embedding *support for virtual organizations* into Linux was needed. In XtreemOS-MD, like in the rest of XtreemOS flavors, OS-level VO support to XtreemOS grid services is achieved by using standard Linux mechanisms like Pluggable Authentication Modules (PAM) and Name Service Switch (NSS). It was found that the basic VO support mechanisms devised for the XtreemOS standard flavor are perfectly applicable to the mobile device case, and that any differences in implementation are more likely to come from the user interface details of mobile devices, than from any internal workings of the system.

We have also found that other modifications to Linux are highly desirable, in order to unravel the full potential of mobile devices in the Grid (for example, having the ability to change access points while doing any grid operation). Thus, XtreemOS-MD will provide *terminal mobility* features, by including Mobile IPv6 (MIPv6) support in the distribution. More concretely, a version of the Mobile IPv6 for Linux (MIPL) implementation will be included in XtreemOS-MD.

Moreover, during this design process, we have become aware of the fact that the mobile world (and specially, the mobile Linux one) is not monolithic, and that a myriad of Linux distributions and initiatives compete and strive for attention. XtreemOS-MD, despite being a self-contained Linux OS, must keep portability in mind (more so than in any other flavor), as the ability to integrate with any Linux distribution and/or standardizing effort can become the main drive for the spread of our operating system in the future.

Contents

List of Figures

List of Tables

Glossary

API	Application Programming Interface
AEM	Application Execution Management
AMS	Account Mapping Service
ARM	Advanced RISC Machine
ATM	Asynchronous Transfer Mode
CDA	Certificate Distribution Authority
CELF	Consumer Electronics Linux Forum
CN	Correspondent Node
CoA	Care-of Address
DoS	Deny of Service
DWARF	Debug With Attributed Record Format
ELF	Executable and Linking Format
FLOSS	Free/Libre/Open-Source Software
FMIPL	Fast Handovers for Mobile IPv6 Implementation
FHMIPv6	Fast Handovers for MIPv6
FPL	(LiMo) Foundation Public License
FSG	Free Standards Group
FUSE	Filesystem in Userspace
GID	Group Identifier
GMAE	Gnome Mobile And Embedded
GNOME	GNU Network Object Model Environment

GNU	GNU's Not Unix
GPE	GPE Palmtop Environment
GPL	GNU Public License
GTK	GIMP Toolkit
GUI	Graphical User Interface
HA	Home Agent
HIP	Host Identity Protocol
HIPL	HIP for Linux
HMIPL	HMIPv6 for Linux
HMIPv6	Hierarchical Mobile Ipv6
HPC	High Performance Computing
ICMP	Internet Message Protocol
IM	Instant Messaging
IP	Internet Protocol
IPv4	Internet Protocol V4
IPv6	Internet Protocol v6
J2SE	Java Platform 2 Standard Edition
JVM	Java Virtual Machine
LGPL	Lesser GNU Public License
LiPS	Linux Phone Standards Forum
LSB	Linux Standard Base
LKKRS	Linux Key Retention Service
MANET	Mobile Ad-hoc NETwork
MAP	Mobile Anchor Point
MD	Mobile Device
MIPL	Mobile IPv6 for Linux
MIPv6	Mobile Internet Protocol v6

MLI	Mobile Linux Initiative
MMS	Multimedia Message Service
MN	Mobile Node
MPPWG	Mobile Phone Profile Workgroup
MTU	Maximum Transfer Unit
NAT	Network Address Translation
NGN	Next Generation Network
NSS	Name Service Switch
NUTSS	NAT, URIs, Tunnels, SIP and STUNT
OMA	Open Mobile Alliance
OPIE	Open Palmtop Integrated Environment
OSDL	Open Source Development Labs
P2P	Peer to Peer
PAM	Pluggable Authentication Modules
PDA	Personal Digital Assistant
PIM	Personal Information Management
POSIX	Portable Operating System Interface
RA	Router Advertising
RADVD	Router Advertisement Daemon
RCoA	Regional Care-of Address
RFC	Request For Comment
SA	Security Association
SAGA	Simple API for Grid Applications
SDK	Software Developer Kit
SIM	Subscriber Identity Module
SIP	Session Initiation Protocol
SMS	Short Message Service

SSH	Secure Shell
SSL	Secure Sockets Layer
STUN	Simple Traversal of UDP over NATs
STUNT	Simple Traversal of UDP over NATs and TCP too
TCP	Transport Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
UID	User Identifier
VO	Virtual Organization
VOM	Virtual Organization Manager
XtreemFS	XtreemOS filesystem
XtreemOS-MD	XtreemOS for Mobile Devices

Chapter 1

Introduction

When designing an operating system, there are myriads of factors that have to be taken into account, from memory and resource management to process scheduling or I/O issues. In the XtreemOS project, the election of GNU/Linux as the base for our grid operating system already narrowed down many of these decisions.

But still, in order to integrate grid functionalities into the operating system, several different paths are laid before the designer. The relative importance awarded to different design principles and goals like ease of use, scalability, reliability, etc, will shape the operating system and, in our case, the kind of grid that is available to users and administrators alike.

1.1 XtreemOS and XtreemOS-MD design principles

As already detailed in several of the project's documents [42, 45], the main design principles and goals of XtreemOS include the following:

Integration of grid and OS services As XtreemOS is a grid operating system, the grid services should be as integrated as possible into other standard OS services, and vice versa.

Unmodified Linux Kernel In order to favor its acceptance into the existing Linux community, XtreemOS will try to avoid unnecessary modifications in the kernel.

Transparency for the users One of the main gripes of current grid approaches is the amount of knowledge needed for doing even the simplest tasks in a grid. XtreemOS should minimize this required knowledge, being as transparent for the user as possible.

Transparency for the applications To minimize the amount of work required by developers of grid applications (or even grid-unaware developers wanting to exploit grid functionalities), XtreemOS must follow standard APIs (like POSIX or SAGA) as long as possible.

Ease of management Another common problem of grid systems is the amount of administrative and management work involved for the system administrators. Independence of users and resources and dynamic allocation of grid usage should allow administrators to be relieved of many of their current chores.

Standards compliance Another way to facilitate the acceptance and diffusion of XtreemOS is to follow industry-strength standards and specifications like POSIX, LSB or the recently initiated efforts into standardizing the mobile Linux market.

Coverage of application requirements Finally, XtreemOS must also cover the requirements provided by a wide range of grid applications, coming from both industry and academia, which have been selected to evaluate the operating system.

Apart from those design principles, there are a number of factors that concern only the mobile device flavor of the operating system:

Limited resources is one of the main features (or, more accurately, problems) of mobile devices. CPU, memory, storage or battery are the four main limitations of these small-format devices.

Unreliable connectivity is another, deriving from the wireless nature of the network connections that these devices use. This unreliability is seldom seen on more traditional Grid/HPC systems, and must be kept in mind at all times.

Terminal mobility is another aspect which should not go unregarded, as the ability to change from one access network to another is another trait that is not normally taken into account in traditional grids.

Together, these are the driving principles that will be taken into account in the rest of this document.

1.2 Document structure

This document describes the logical sequence that we have followed in the design of the Linux-XOS for mobile devices.

Chapter 2 describes the election of the base Linux distribution for mobile devices upon which the PDA version of XtreemOS-MD will be built (namely, the Ångström distribution).

In Chapter 3, we start delving into the general architecture of the XtreemOS-MD distribution, with a brief description of the different software components that will compose it, be them either standard Linux ones or modules specific to XtreemOS-MD.

Afterwards, Chapter 4 gives more details on the specific design of the Foundation layer of XtreemOS-MD, with the description of the VO support mechanisms that must be embedded into the operating system, in order to make it “aware” of virtual organizations. After analyzing the mechanisms proposed for the standard version, they were found to be perfectly applicable to mobile devices, apart from changes to the user interface.

Chapter 5 describes the other main modification to the operating system that will be done in XtreemOS-MD: the terminal mobility modules. After analyzing the different solutions for mobility available today, we conclude that the best option is to add Mobile IPv6 support to XtreemOS-MD, through the inclusion of the most widespread implementation of it (MIPL).

Then, Chapter 6 reviews the different mobile Linux initiatives and projects, their outputs and possible interactions with XtreemOS-MD, which could prove an important factor, not only in the design and implementation of the distribution, but also on its dissemination and acceptance.

Finally, the next steps and a number of research issues already identified are described in Chapter 7, which could be explored in the advanced version of XtreemOS-MD.

Moreover, this document includes an appendix describing a list of the already identified daemons, software packages, Linux modules and versions needed by the VO support and mobility features (appendix A). Also, a more extensive up-to-date analysis of the existing mobile Linux initiatives can be found in appendix B.

Chapter 2

First Step: Election of Base Distribution

Unlike what happens with other flavors of XtreemOS, the Linux distribution on which XtreemOS-MD will be based was not decided beforehand. One of the reasons behind this peculiarity is the fact that the distributions selected for other flavors (namely Mandriva, Red Flag and Debian) do not have a mobile counterpart[1]. Debian does have ports of their packages for mobile architectures like ARM [6], but the efforts to make a Linux distribution for mobile devices with those packages (e.g. the EmDebian distribution [7]) are either not very usable or no longer maintained at the moment.

With this situation in mind, it was decided at the start of the project to search for one of the specialized Linux distributions for mobile devices like PDAs and mobile phones (see [47] for a brief review of the main options), that suites the needs of the project.

In fact, very few Linux distributions can claim support for "mobile devices" in general, not even in a category of them (e.g. support for PDAs or for Mobile Internet Devices), due to the lack of standardization in the area, which makes hardware support a "one device at a time" process. The best a distribution can make is to support a long list of individual devices[2]. In fact, it cannot be assured that the distribution choice for XtreemOS in the PDA version will also be relevant for the mobile phone version. The discussion that follows applies only to the PDA or basic version of XtreemOS-MD.

The main requirements of XtreemOS, apart from those described in deliverable D2.3.2 [47], can be summarized as follows:

Open source This may seem a trivial requirement in Linux, but many Linux-based operating systems for mobile devices (specially commercial ones) do not

[1]Just as this document was being written, Red Flag announced a "mobile version" of the distribution, but aimed just at Intel's MID platform, thus not covering the wide range of devices that XtreemOS-MD required.

[2]By individual device, we mean manufacturer and model.

have their codebase publicly available (or easily accessible, at least).

Modern software As this is a research project, and the other flavors of XtreemOS are using fairly bleeding edge versions of libraries and tools, it would be pointless to build XtreemOS-MD over a distribution composed of outdated packages. It also would be more difficult to implement and would raise compatibility issues.

Community driven Products and projects that don't have a community behind, imply greater risks. Some companies change their policy and close the software. Small companies sometimes are bought by others with its own product or different business model. Sometimes companies or individuals quit a project. To have an active and large community is a good signal for the success of a product and a warranty of its continuity. Another reason in favor of communities is that the mobile version of XtreemOS is not so targeted to final users as it is targeted to developers and integrators (as most users of embedded software don't install the software themselves; they buy a product that has the software embedded and generally they don't know what operative system it is based on). It is easier to reach programmers contributing to a community based distribution than to another one that is a personal or closed project.

Wide hardware support As outlined above, hardware support is a very important (and difficult) issue in mobile devices. Thus, the choice of a base distribution should be heavily dependent on having good hardware support: not only a long list of devices, but also that these devices are varied enough (from different categories and device manufacturers), and also modern enough.

Availability for mobile phones This is a particularization of the previous one. Although it is quite difficult that a distribution complies with this requirement, it would be highly desirable that the same election would be valid for both PDAs and mobile phones.

With all these requirements in mind, our election is the **Ångström distribution** [2]. It is a community-driven effort, formed mainly by developers of the OpenEmbedded [31], OpenZaurus [34] and OpenSimpad [33] projects. The project also draws knowledge from the Familiar project [9] and the handhelds.org [16] community, which is a kind of "de facto" standard for Linux PDAs, with the widest hardware support, but quite outdated software versions. The scope of the project is quite broad, aiming "to unify their effort to make a stable and user-friendly distribution for embedded devices like handhelds, set top boxes and network-attached storage devices and more".

On the technical side, Ångström uses modern versions of much of its software and development tools (cross-compilers, for instance). It is also remarkable that the OpenEmbedded framework is used to build the distribution, which can be very valuable as this framework is becoming commonplace in many modern embedded

Linux projects. The graphic environment officially supported is the GPE Palmtop Environment [15], although OPIE [35] and even Enlightenment [8] environments are also possible. More details on the architecture of the distribution can be found on Chapter 3.

Ångström's official hardware support is growing as the community tests it in different devices. Although its wiki page states that 14 devices will be officially supported, as of this writing, 39 testing images for different devices can be found in their repositories [3], including a wide variety of devices like HP iPaq PDAs, Nokia internet tablets and even smartphones from Motorola or FIC (the mobile phone used by the OpenMoko project [32]).

Ångström is also very suited for the purposes of XtreemOS in that it is a quite streamlined distribution (it ships with less packages than the typical distribution), so that it can be later customized for various purposes (be it for PDA use, internet tablet use, set-top box or, in our case, to add grid functionality on top of it).

This distribution also fulfills some of the requirements detailed in deliverable D2.3.2, particularly those requisites not specifically related to grids. Table 2.1 shows a summary of the requirements supported. Those requirements not fulfilled by the distribution itself will have to be implemented by XtreemOS software and services, according to their importance.

D2.3.2 Requirement	Short description	Notes
R2.3.1	Hardware support	Built for ARM, many devices supported
R2.3.2	Battery shortage	GPE incorporates battery warnings
R2.3.3	Basic input methods	Touchscreen support
R2.3.4	Connectivity detection	GPE incorporates network monitor
R2.3.6	Offline use	Ångström is usable without grid users
R2.3.61	Java support	At least one JVM included (`jamvm`)

Table 2.1: Requirements fulfilled by Ångström

As far as we know, neither Ångström nor any other mobile Linux distribution have clear relationships with any of the standardizing efforts described in Chapter 6, mainly because of the novelty of these efforts, and their lack of clear specification outputs.

However, it is also important to note that, although the basic version of XtreemOS-MD will be integrated into the Ångström distribution, in XtreemOS we will not only provide a complete mobile Grid OS, but also a number of packages that can be integrated into any other mobile Linux distribution, either by end users themselves or (more likely) by device integrators.

Chapter 3

General Architecture

Before describing in detail the structure and interfaces of each of the modules composing the Foundation layer of XtreemOS-MD, it is necessary to describe the overall architecture of an XtreemOS-MD node, as it differs from an standard (i.e. PC) node in the XtreemOS Grid.

Figure 3.1 shows the architecture of an XtreemOS-MD node. As it can be seen, there will be three kinds of components in such a node:

Base Linux components are the components that come with the original Linux distribution for mobile devices (Ångström in this case), covering from a standard kernel, libraries, graphic environment and a Java VM to personal information management (PIM) applications.

XtreemOS-specific components are additional components that will be included in the XtreemOS-MD distribution. In the diagram, the modules covered by WP2.3 are labeled as "XtreemOS-F components", and provide the basic infrastructure for constructing mobility and grid services. The rest of the XtreemOS-MD components (covering the grid functionality itself) will be developed in WP3.6.

Applications This components will not be included in the XtreemOS-MD distribution, and manufactures, application developers or end users will have to provide them. This includes both grid-aware and traditional (grid-unaware) applications.

Base Linux components

Linux 2.6 Kernel + Kernel Mobility Modules XtreemOS-MD is based on Linux OS and will use Ångström distribution as its development base for the PDA version. This distribution uses a quite up-to-date 2.6 kernel, but in XtreemOS some changes will be done to this kernel (mostly in the compiling options, but also patches will be included) for adding support for terminal mobility.

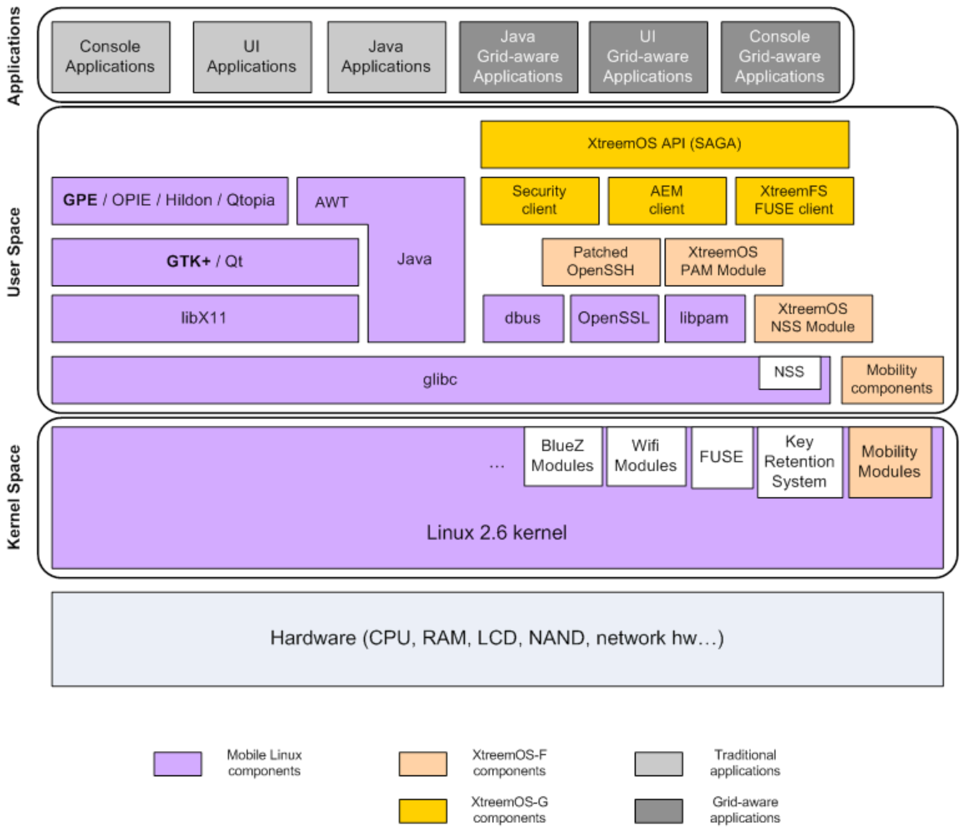

Figure 3.1: XtreemOS MD Software Stack

libC is the standard POSIX library for input/output, memory management, string manipulation, and the like. Most C programs in Linux are compiled and linked with them, being glibc the most widely used [14]. Ångström, as most PDA and MID distributions, includes glibc, because more complex programs are not portable to lighter alternatives (like ulibc or newlibc) without changes. No glibc alternative, for example, includes support for NSSwitch, a powerful extension mechanism used by XtreemOS and several Linux applications.

dbus D-Bus is a message bus system, used for applications to talk to one another. In addition to interprocess communication, D-Bus helps coordinate process life cycle, and easies the coding of "single instance" applications or daemons, and to launch applications and daemons on demand when their services are needed [11].

PAM (libpam) The Pluggable Authentication Modules (PAM) is a suite of shared libraries that enable the local system administrator to choose how applications authenticate users. It is used by XtreemOS to support virtual organizations natively in Linux.

OpenSSL is an open source implementation of the SSL and TLS protocols, implementing basic cryptographic and utility functions. OpenSSH is strongly dependent on this library, which includes all the cryptographic functions used by SSH. There are libraries lighter than OpenSSL, but only a few are API compatible, and those have more restrictive licenses.

Java (JVM) A Java interpreter, that converts the Java intermediate language (bytecode) into machine language and executes it [40]. Due to the lack of an official Sun J2SE Java Virtual Machine for the ARM architecture, probably one of the open source alternatives (as JamVM, which is included with Ångström), will be used in XtreemOS-MD.

Linux User Interface for GUI applications:

- **libX11** X11 libraries let the user build applications with a standard graphical interface, providing a basic framework, compatible with every other X11 system. Most Linux applications with a graphic user interface are X11 based, but normally using a toolkit as GTK+ or Qt. Some embedded single-application devices use directly the framebuffer instead X11.
- **GTK+** is a multi-platform toolkit for creating graphical user interfaces. It is used, among others, in the GPE graphical environment (see below).

GPE is an environment of components which makes it possible to use a GNU/Linux handheld for typical PDA tasks. It's based on GTK+ and X11 and its main modules are:

- **PIM** Personal Information Manager tool. Stores addresses, journal, reminders, emails, etc.
- **Multimedia** Software for viewing pictures and videos or listening to music.
- **Browser** Internet browsers adapted to small devices.

Although Ångström uses GPE for its graphical environment (and that is why it appears in the diagram), other alternatives are possible in other distributions and in other mobile devices. These include OPIE (Open Palmtop Integrated Environment) or Qtopia, based on the Qt toolkit, or the Hildon framework, used by Maemo (Nokia N800) and Intel, which is also based on GTK+. Although Ångström uses GPE, XtreemOS-MD components should be user interface platform neutral (i.e. should not have dependencies with GPE or GTK+), so as to increase portability.

XtreemOS-specific components

As with any other XtreemOS flavor, XtreemOS-MD grid-specific components can be grouped into two layers:

XtreemOS-F layer

The Foundation layer of XtreemOS comprises the components that modify the Linux OS itself, to make it aware of grid concepts like virtual organizations (VOs) and VO users, supporting the grid services of the XtreemOS-G layer.

Components for VO support in Linux OS

XtreemOS NSS Module This name service module adds virtual organizations support to basic Linux tools that show information about UIDs.

XtreemOS PAM Module This is a Pluggable Authentication Module that implements most of the low-level security and session aspects of Virtual Organizations.

Patched OpenSSH This is a modified version of OpenSSH. Patches allow to start a remote session using VO security and user mappings.

Components for enhanced mobility of the device

Kernel Mobility Modules Although XtreemOS is based on Linux OS and uses Ångström distribution as its development base, some special patches and modules are needed as a complement to the mobility enhancements (see below), to give terminal mobility at the system level.

XtreemOS-F Mobility Components This component includes the daemons and libraries necessary for providing Mobile IPv6 (MIPv6) support, thus giving terminal mobility to applications and services, in a transparent way.

XtreemOS-G layer

The Grid layer of XtreemOS is composed of the grid services themselves, that enable users to execute jobs in other machines, access remote data, etc.

XtreemOS-G components in a mobile device include:

- **Security Client** This component will include the communication tools necessary for the mobile device to talk to the VO Management and Security services.
- **AEM Client** This component will provide client access to XtreemOS Application Execution Management system.
- **XtreemOS FUSE client** This component will provide client access to the XtreemFS filesystem, using a FUSE module. FUSE is a Linux-specific technology that allows to implement file system modules in user space instead of in kernel space.

XtreemOS API This component will offer a subset of the XtreemOS Grid API (based on SAGA) to applications running on the mobile device.

Applications

JAVA Application/Services Any software created with a Java language compiler able to run in a JVM.

JAVA GRID Application/Services Any Java application/service that uses XtreemOS's Grid API.

Console Applications Any application or utility designed to run in text mode, without any GUI (Graphical User interface). It also applies to any service or daemon running in background.

Console GRID Applications Console, server and batch applications that use XtreemOS Grid services.

UI Applications GUI based applications using, for example, GPE framework.

UI GRID Applications This includes every application designed with a graphical user interface as GPE or OPIE, that makes use of XtreemOS Grid API.

Chapter 4

VO Support in Mobile Devices

In this chapter we will describe the mechanisms needed in mobile devices to adequately support virtual organizations and grid services, as XtreemOS understands them. We will begin by introducing the concept of VOs in XtreemOS and its general implementation principles (see [45] for more details). Afterwards, we analyze how this implementation is applicable in the case of mobile devices, and finally we will describe the interfaces offered and some concrete use cases in mobile device scenarios.

4.1 VO support in XtreemOS

4.1.1 Virtual organizations in XtreemOS

XtreemOS outsources user management to an external service, the Virtual Organization Manager (VOM). Users affiliated to a virtual organization can exploit grid resources (computation resources, data or knowledge bases, ...) although they are unknown on these resources. The VOM is in charge of managing its users, authenticating user requests, selecting adequate resources for handling user requests, monitoring, tracing and accounting resource usage.

Normally, one computer can act as a client of XtreemOS services (for instance, a user laptop from which a user can submit grid requests and monitor his applications), support basic XtreemOS services such as XtreemFS or provide computational resources for handling the execution of complex user-defined applications.

However, due to their limited capacity, mobile devices will only be used as clients or access points to more powerful grids: thus, a XtreemOS mobile device must support the XtreemOS client interface, apart from other XtreemOS-MD specific functionality like MIPv6 support.

4.1.2 Features and functionalities provided by generic XtreemOS VO support

XtreemOS standard flavor provides a number of VO-related features to grid users:

Credential management When a user process accesses some XtreemOS service, the credential management system can provide all necessary info (such as credentials) about the user, to the aforementioned service.

UID/GID dynamic management Inside a Linux operating system, access rights to files and processes are based on UID and GIDs associated to the process. In order to provide isolation and sharing, the generic XtreemOS VO support integrates a dynamic UID/GID allocation system (e.g. through the permissions granted to each of the mapped UID/GIDs).

Linux name service switch The Linux kernel uses numeric UID/GIDs internally. But, in order to provide a more pleasant user interface, the Linux operating system API uses text-based user and group IDs. In XtreemOS, global user and group identifiers are managed at VO level. The generic XtreemOS VO support provides the mapping between internal UID/GIDs and global identities.

Authentication XtreemOS provides a generic service for user and VO authentication. This generic service can be used by any operating system service.

Local policy enforcement – authorization The generic XtreemOS VO support provides a configurable common policy and authorization decision and enforcement point for local services.

Session management XtreemOS VO support provides a global support for session management: job state changes are notified to the job controller, accounting is automatically started at session initialization and reported to the job controller at session end, all processes and local files are cleaned after execution, and dynamically allocated UID and GIDs are freed.

4.1.3 Features and functionalities provided at higher level

The aforementioned VO functionalities are not useful per se, but support a number of functionalities that are provided by grid services at a higher level, such as:

XtreemOS overlay

Each XtreemOS VO provides an infrastructure that can support highly available and scalable grid services and applications, such that these kind of services can be developed independently from underlying instances of the XtreemOS operating system. An important part of this infrastructure is a P2P overlay, which represents the nodes of a certain VO. This overlay is used for node management, specially in the context of executing grid processes. In order to support XtreemOS global services associated to some VO, an XtreemOS resource must be connected to the corresponding overlay.

Application execution management

XtreemOS VO support also has implications in the execution of applications in the grid, through the AEM service. The AEM service provides functionalities for application submission, control, and monitoring, and for user management (through PAM modules). In order to be selected for job execution, a resource must be integrated in the VO overlay.

Data management – XtreemFS

XtreemOS VO support also has implications in the access to data services like the XtreemFS filesystem: access rights to the Grid filesystem are controlled using credentials managed by VOs and transmitted in the XOS-Cert. These credentials can be exploited directly (XtreemFS can retrieve the XOS-Cert from the process keyring) or indirectly through the internal UID/GIDs associated to the process from the XOS-Cert by the XOS PAM module.

4.2 Analysis of technological solutions

4.2.1 Evaluation criteria

When designing XtreemOS VO support, three criteria were mainly used for evaluation, and as guides for electing an adequate implementation:

Scalability One of the unfulfilled promises of current grid technologies is to provide resource usage across a virtually unlimited amount of resources. Thus, one of XtreemOS's main aims is to provide a grid operating system that is scalable enough to realize the vision that guided the concept of Grid from its very beginning.

Volatility Grid environments are volatile by nature, as the nodes are not guaranteed to be connected at all times. Moreover, the probability of any node disconnecting at any given time increases when the total number of nodes increase. A grid operating system like XtreemOS, so focused on scalability, must be resilient to changes in the membership of both users and resources.

Ease of use and management One of the main problems faced by current grid middlewares is the amount of administration chores, and the difficulties that users must endure to have a working system. This could be seen as another face of the scalability problem, as this management and usage issues increase with the size of the grid involved. Thus, XtreemOS should minimize the amount of management work necessary to run virtual organizations in the Grid.

Moreover, several new criteria arise from the fact that we are implementing this kind of support into a mobile device, which is quite limited, resource-wise:

Small footprint As mobile devices have a limited amount of resources (sp. memory and CPU), the way of supporting virtual organizations in XtreemOS (or at least in the mobile nodes participating) should be efficient and lightweight.

Connectivity Another characteristic essential to mobile devices is the unreliability of their network connections, compared to static nodes in the Grid. Thus, another important criteria in the implementation of VO support in XtreemOS-MD is that temporal network disconnections should not hamper it excessively.

Mobility As its name implies, mobile nodes in the grid can move, and change access points. Thus, the solution for VO support should also be resilient to these changes in node location. This aspect is mostly covered by other modules (see Chapter 5), but it should nevertheless be taken into account in the design of the VO support.

4.2.2 Selected solution

Using the basic node-level VO support proposed in deliverable D2.1.2 [45], a user requests an XOS-Cert from the VO CDA (Certificate Distribution Authority) when he logs into a VO. This login process is PAM-aware and forks a new process holding the XOS-Cert in its session keyring and possibly with a new local identity (UID/GID), depending on the XOS-Cert credentials. Note that the user can log in locally on his XtreemOS box before logging into the VO. Once the user is logged into some VO, he can access VO services directly (for instance, using AEM's job submission service) or indirectly through SAGA.

This approach clearly separates VO management from resource management. Resource running VO services need only be configured for accepting requests from the VOs. Resource re-configuration happens rarely. On the other hand, VO management (adding or removing users, managing user groups/roles, modifying global policies, ...) is completely independent from resource configuration. For instance, adding a new user does not necessitate any modification of resource configuration: a user account is created on some resource on the fly, only when this resource has been selected for running some application from the user. Moreover, applications from the same user can be run on different accounts depending on the group/role/capability requested by the user and allocated by the VO Manager.

When compared to the approaches taken by existing middleware, the XtreemOS solution is more scalable, can handle more volatile environments and allows more precise access control.

Scalability: In XtreemOS, user management is independent from resource management and configuration. The configuration of a resource is never modified when a user is added to or removed from a VO. The cost of user management does not depend on the number of resources. This solution is more scalable than solution based on gridmap-files (e.g. Globus/glite solutions) which must be updated (manually or periodically) in order to handle the mapping between global users and local accounts.

Volatility: In volatile environment (which is the case of grid environments in general and especially with mobile devices), all resources are not connected all the time. Unconnected devices cannot be updated (reconfigured) remotely. In XtreemOS, the VO configuration of resources is minimal and needs no (or very rare) reconfiguration: all configuration data (credentials, etc) are transmitted dynamically in XOS-Cert.

Access control: In XtreemOS, the credentials inserted by a VO Manager (or, more concretely, by the CDA) in the XOS-Cert can depend on the roles and capabilities requested by the user and allowed by the VO for a request. Dynamic management of user credentials allows a fine-grained management of access control on resources. Two different users can share files if their XOS-Certs contain common group credentials (it is even possible to map different users to the same account). On the other hand, it is also possible to completely isolate environments corresponding to the same user running requests with different roles.

Lightness: The OS-level approach to VO support taken in XtreemOS is particularly suited to mobile devices, as using technologies already present in the Linux kernel and standard libc libraries do not impose more burden in a mobile device's already taxed resources. It does use one additional daemon (the AMS service) and database (for storing the mappings), but their footprints are fairly reduced, and can be supported by current hardware.

Connectivity: Although a network connection is required for getting the XtreemOS certificate from the CDA, the VO support mechanism of XtreemOS does not require continuous network connectivity, and does not impose heavier network requirements than those needed by the grid services (e.g. AEM, XtreemFS) themselves.

Mobility: The transparent mobility features provided by the mobility modules described in Chapter 5 allow for other XtreemOS capabilities to be unharmed by changes in the terminal location, as with MIPv6 the apparent IP of the device remains constant when changing locations. In this way, terminal mobility should not pose a problem to the VO support capabilities of XtreemOS-MD.

After analyzing how XtreemOS implements virtual organization support, and the applicability of this implementation with regard to the aforementioned criteria, we can safely conclude that *the VO support solution in the standard flavor of XtreemOS is also applicable to mobile devices*, in a general sense. A number of changes, specially in the user interfaces of applications that use these VO support features (like, for example, login applications), are certainly foreseeable but, as they are more in the application layer, they will not be described in this document.

Furthermore, a more thorough performance analysis of this solution will be done in the implementation phase, to anticipate and eventually avoid any problems that could arise in the porting of the code to mobile devices, due to their limited resources.

4.3 Description of VO support modules in mobile devices

As it has been mentioned before, the design of the mobile device version of VO support modules largely coincides with the generic VO support conceived in XtreemOS standard flavor (please refer to [45] for more details).

In order to implement this generic VO support, the mobile device Linux flavor should support PAM modules, NSSwitch modules as well as kernel keyrings. Linux PAM modules are necessary to implement the dynamic management of user credentials, avoiding frequent reconfiguration, and avoiding kernel modifications as well as standard software modifications. Even if in the basic version of XtreemOS-MD the mobile device will not execute jobs from external users (it is a simple client), dynamic management of user credentials is still necessary to reflect the semantics of XtreemOS because the credentials of local users, once he logs into a VO, depend on the effective role/capabilities allocated by the VO. NSSwitch modules allow to reference global user and group names using the standard POSIX API. Note that the implementation of dynamic accounts also necessitates some kind of local database manager (db) for the Account Mapping Service (AMS).

Fortunately, the Ångström distribution includes `glibc` libraries (which in turn includes the NSS subsystem), as well as the PAM libraries needed by the aforementioned design. Moreover, most modern mobile Linux distributions include this kind of software, a fact that predicts the high grade of portability of this solution to other mobile Linux operating systems.

Figure 4.1 presents the three main components necessary for supporting virtual organizations in a Linux node. The main components are the PAM modules, the NSSwitch modules, and the Linux kernel key retention service. These main components also make use of some local services like the AMS (which in turn uses a database for storing the user mappings).

PAM modules Using Linux PAM, an application or service can delegate all steps necessary for user authentication, authorization and session management to a set of dynamically loadable and configurable modules managed by the PAM system.

The XtreemOS package provides two extra PAM dynamically loadable libraries: `pam_xos_acctmgt.so` for authentication, authorization and session management, and `pam_xos_plymgt.so` for policy management. These modules use the standard PAM API: these modules do not introduce any extra API. PAM-aware

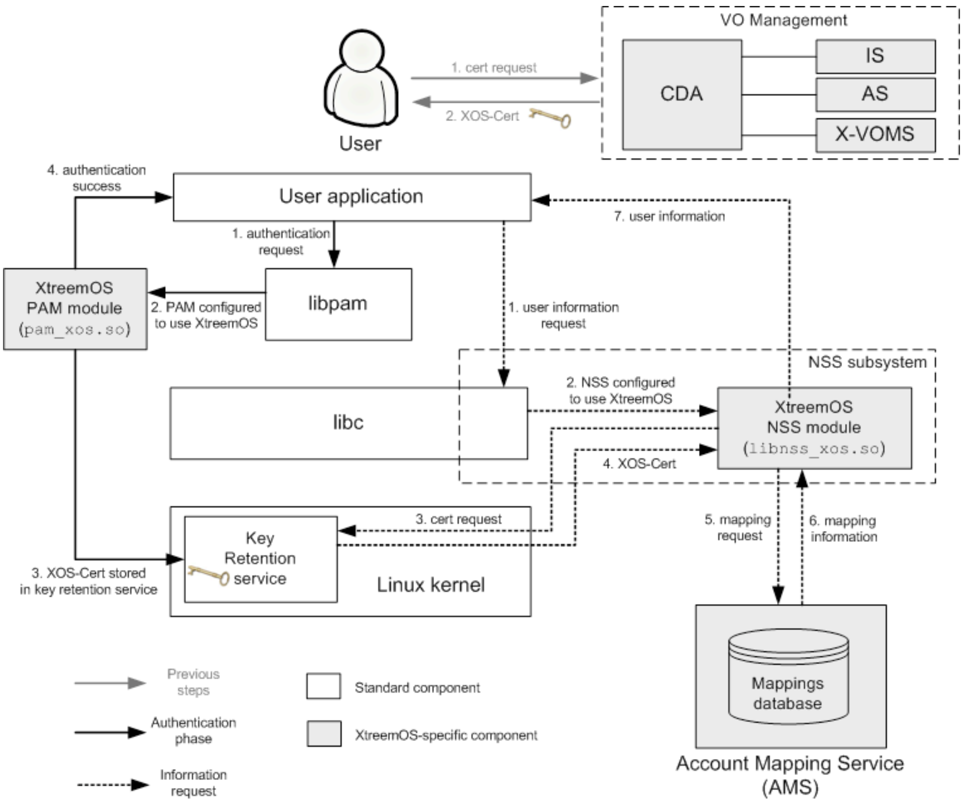

Figure 4.1: An architecture overview of Node-level VO support

applications wishing to exploit these new modules don't need to be modified. Only the PAM configuration file associated to the application file or service needs to be updated.

Name Service Switch The NSSwitch modules of Linux are in charge of translations between external names such as user names, group names, host names, service names, etc and their internal representation using the libc API. The `libnss_xos.so` module provided by XtreemOS is, for example, in charge of the translations between global user and group names provided by a VO and their kernel numerical representation.

The XtreemOS package provides one new NSS switch module which extends the standard `libc` name service to an extra database, the AMS database. No application or service needs to be modified in order to exploit the new module. It is automatically accessed through the standard `libc` library as soon as the module name has been added to the `nsswitch.conf` configuration file.

Linux Kernel Key Retention Service (LKKRS) The XtreemOS security certificates can contain VO-certified user credentials or attributes which can be used by XtreemOS services (internal or external) for various reasons such as policy decision, etc. In XtreemOS, the security certificate is securely attached to each user process through the kernel key retention service (also known as kernel keyrings).

Account Mapping Service (AMS) The Account Mapping Service is in charge of storing the mappings between local UID/GID and global user/group identities. Mappings are added/removed during session management in XOS PAM modules. Identity mappings can be requested explicitly through the AMS API or implicitly through `libc` using the NSS subsystem module.

The AMS service installed by the XtreemOS package is used internally by the PAM modules and the NSS switch module and can also be used internally by other internal services such as XtreemFS. Currently, it is not planned to provide AMS access to end users.

Higher level services Regarding the higher level services that make use of this VO support functionalities, as the mobile device flavor has the same range of functions as its standard counterpart, theoretically the whole range of services could be supported. But, at least in the basic version, not all of them are needed, as described in deliverable D3.6.1 [46]:

VO Overlay is not needed in mobile devices that only access XtreemOS services as clients.

AEM services are, for the most part, not needed. In mobile devices only the XATI interface to the AEM will be implemented.

XtreemFS Mobile users should be able to read and write files from the XtreemFS filesystem, and mount XtreemFS volumes.

4.4 Interfaces

Most of the APIs of XtreemOS-F VO support are internal. External APIs are native, well-known APIs of Linux or Unix systems, as POSIX, PAM or the Key Retention Service. For more information, please refer to deliverable D2.1.2 [45].

NSS API

The API of XtreemOS NSS module is transparent to applications. Glibc library is the only software that has direct access to NSS code. NSS module changes the behavior of the following glibc methods: `getpwnam`, `getpwuid`, `getpwent`, `getgrnam`, `getgrgid` and `getgrent` (actually the implemented methods are the reentrant variants, e.g.getpwnam_r, non-reentrant variants are coded as front-ends to reentrant ones).

There is more information about this methods in deliverable D2.1.2 [45], in Linux man pages and the texinfo libc pages. Lighter glibc alternatives don't include support for NSS: with these libraries the only solution is to overwrite this functions to invoke NSS module counterparts, thus discouraging their use in the implementation of XtreemOS-MD. Exported symbols in NSS module have _nss_xos_ prefix, e.g. _nss_xos_getpwnam_r.

Some applications that use functions affected by our NSS module include: ls, whoami, finger, login, sshd...

Some important PAM modules, all file managers and XtreemFS also make use the glibc NSS API. None of these applications or libraries need to be recompiled. glibc does not need recompilation either, because it loads NSS modules dynamically.

PAM API

As with NSS, the API of XtreemOS PAM modules is transparent to PAM-aware applications, and it's only visible to the PAM library. Applications that support PAM (e.g. login, su, sshd, Apache) only use the standard API of PAM, by using functions like pam_authenticate, pam_acct_mgmt, pam_setcred, pam_open_session and pam_close_session.

Internal symbols exported from PAM modules to the PAM library are: pam_sm_authenticate, pam_sm_acct_mgmt, pam_sm_setcred, pam_sm_open_session and pam_sm_close_session

Kernel Key Retention Service API

Most applications and libraries do not need access to XreemOS credentials, as the PAM module is the one who does the authentication. Other XtreemOS components and libraries like SAGA need access to the XtreemOS certificates (XOS-Cert) stored in this service. These components and libraries obtain the credentials using the Key Retention Service API that the Linux kernel provides. This API is available in two forms: direct calls to keyctl and req_key system API using syscall, or through the libkeyutils helper library (from a Red Hat package named keyutils). Also, libkeyutils provides request_key and keyctl functions. There is a full list of functions available in any Linux machine, in /usr/include/keyutils.h

AMS

The Account Mapping Service does not provide a client library to access AMS server, and its public API is the description of the network protocol used to communicate with it. Normally, user applications will not need direct access to AMS.

So far, only PAM and NSS modules need access to the AMS, which is a internal service of XtreemOS-F. In the future, the protocol can be extended to cover requirements of other components of XtreemOS such as XtreemFS.

XtreemOS API: SAGA

As mentioned in several XtreemOS documents [42], XtreemOS will provide a SAGA (Simple API for Grid Applications) API to any SAGA-aware grid application. Although the SAGA implementation in mobile devices is part of the work in WP3.6, it is worth noting that the VO support features like the usage of XOS-Certs, will be provided through SAGA's Contexts, the part of the SAGA "Look & Feel" that deals with security tokens.

4.5 Examples of use

Login on device startup

The login process can take place when the device is powered on: in this case, the user will see the usual login screen, but will be able to state a VO identity as a valid user for login.

Once the user is logged in, she can start a job management application (such as the JobMA application in WP4.2) for launching and monitoring her grid jobs from the PDA. Also, she can access files stored in her XtreemFS "home directory", which has been automatically mounted. No additional passphrases or security prompts are necessary when executing each of these applications.

Transparent grid usage

Alternatively, if we want to make grid functionalities completely transparent to the end user (e.g. in mass market applications), the system will directly login with the user's certificate, which must already be present at the terminal (most likely, put there by the integrator, or using operator-provided certificates stored in the SIM card).

After bootup, the user accesses a file sharing application, and searches, shares and accesses remote files (which are stored in XtreemFS, although she does not know it) in a secure and seamless way, and the file sharing service can track and bill the user according to her usage.

Afterwards, the user wants to write an e-mail, so she runs a mailing application, incorporating an advanced voice recognition system, which uses complex algorithms to translate natural language to written text. Without the user knowing it, the speech wave is submitted to one or more XtreemOS nodes, which run the algorithms (customizing them to our concrete user, who is identified by her certificate) and return the written text to the user. Of course, all this data transmission can be secured so that user confidentiality is always preserved.

Chapter 5

Mobility Enhancements

In this chapter we will describe a number of modifications to the operating system that will be applied in the mobile device flavor of XtreemOS. The goal of these modifications is to enhance the performance of XtreemOS-MD devices in mobile scenarios, by enabling them to change their location while performing grid operations, in a seamless and transparent way.

We will begin by reviewing what kind of mobility we aim to address with these modifications (namely, what is commonly known as terminal mobility), and which are the most common solutions for providing this kind of mobility. Afterward, we present the selected solution (Mobile IPv6), along with the details of the implementation in XtreemOS-MD, its interfaces and some concrete examples of use.

Please bear in mind that, although these modifications will only be included into the mobile device flavor of XtreemOS, the same principles could also be applied to standard XtreemOS nodes that are able to move (for example, laptops).

5.1 Mobility in XtreemOS

As follows from previous deliverables ([48, 47]), the concept of mobility in XtreemOS-MD can be seen from three different standpoints:

User mobility is achieved when the user can access the same (possibly personalized) services from any location and from different machines.

Session mobility allows users to transfer an ongoing session from one machine to another without interrupting it.

Terminal mobility is the ability to move the user's terminal from one access point to another (e.g. changing WiFi hotspots), without impacting in the operations that the user is performing.

User mobility is achieved in XtreemOS (not just in XtreemOS-MD) by the concept of global identity. By using his global identity, a XtreemOS user is able to log into XtreemOS grid from any machine in the grid (provided that the VO's policies allow for that kind of behavior). Once adequately identified, users can execute and manage jobs running in the grid, and access their files stored in the XtreemFS filesystem.

As for session mobility, and taking into consideration that session data (and the whole concept of "session") is almost completely application-dependent, it should be addressed at the application level.

Thus, the concept that we will be referring to as "mobility" from now on, will be the one of terminal mobility. To implement this kind of mobility, XtreemOS-MD will provide a *mobility enhancement* component. The selected approach to implement this component is the use of Mobile IPv6, with the reasons and design to this choice being explained in the following sections.

5.1.1 Features and Functionalities

This mobility enhancement module will provide, not only terminal mobility, but also will enable mobile devices to access XtreemOS distributed servers (as described in WP3.2 [43]), as MIPv6 support implies having IPv6 support:

Terminal mobility by the use of MIPv6, mobile nodes are able to change access points with minimum performance penalties and, most importantly, in a transparent fashion for applications and users.

Ad-hoc distributed servers are implemented as a collection of machines that are seen, from the client's point of view, as a single, fault-tolerant machine. This technology is also implemented using Mobile IPv6, and requires that any node that wants to interact with them supports the IPv6 protocol.

5.2 Analysis of technological solutions

In this section, current technological solutions to implement terminal mobility enhancements will be analyzed, selecting the most adequate for XtreemOS-MD. To obtain further information about this options, please refer to deliverables D2.3.1 [48] and D2.3.2 [47].

Mobile IPv4 is an extension to the IPv4 protocol that allows a mobile node to maintain its connections when moving from one access router to another (changing both the location and addresses). This process is called *handover*.

As mobility is handled at network level (IP), mobility is handled in a transparent way for user applications, which use it unknowingly.

- **PRO** There is an implementation of this for Linux OS.
- **CON** This protocol has the problem of triangular routing (see [48] and [47]), which imposes a performance penalty on roaming nodes.

- **CON** Additional network infrastructure (in the form of foreign agents) is needed.
- **CON** With each change of location, a mobile node has to send location updates to any node it corresponds with; moreover, updates have to be sent periodically. This involves a lot of signaling and processing, taking up considerable network and processing resources.
- **CON** It uses the same mechanisms for both local and global moves, which is an inefficient use of resources in the case of local mobility. Local mobility should be managed locally.
- **CON** There is a handover latency problem. In the time passed while handovers take place, communications stop; this becomes a problem when using real time applications that need a continuous flow of data.
- **CON** The protocol is not widely supported in current networks, and its progress seems to have been abandoned in favor of more modern alternatives like MIPv6.

Mobile IPv6 Similarly to MIPv4, it is an extension to the IPv6 protocol allowing for mobility between access routers (*handover*). The mobility enhancements it provides are achieved through the addition of header information, a new internet Control Message Protocol (ICMPv6), changes to router discovery messages and a set of mobility options included in mobility messages.

- **PRO** Route optimization feature is a fundamental part of the protocol (it was just optional in MIPv4), to avoid the problem of triangular routing.
- **PRO** There is an implementation for Linux, such as **MIPL** [26].
- **CON** As with the case of MIPv4, the signaling, both periodic and when handover occurs, can be rather costly.
- **CON** Updates occur for both local and global moves, as for MIPv4.
- **CON** During handover, there is a period in which the mobile node (MN) is unable to send or receive packets because of link switching delay and IP protocol operations. This phenomenon is known as *handover latency*.

To solve the problems of MIPv6, a number of extensions to the protocol have been proposed:

Hierarchical Mobile IPv6 is an extension to MIPv6. HMIPv6 manages two kinds of mobility: macro mobility (from one site to another) and micro mobility (within a site). It's designed to reduce the amount of signaling required and to improve handoff speed for mobile connections [39].

- **PRO** Compared to MIPv4, it adds a new node called Mobility Anchor Point (MAP) that serves as a local entity to aid in mobile handoffs, replacing MIPv4 foreign agent. It decreases handoff-related latency.
- **PRO** As already commented in previous paragraphs, compared to MIPv6 and MIPv4, it separates global and local mobility. The first one is managed by MIPv6 while the second one is managed locally.
- **PRO** There is an implementation of this protocol for Linux by Monash University [28], which enables full MN and MAP Basic Mode HMIPv6 functionalities.

Fast Handovers for Mobile IPv6 specifies a protocol to improve handover latency due to Mobile IPv6 procedures, but it does not address link switching latency.

- **PRO** Fixes Handover latency problem allowing the mobile device to send data packets as soon as a new IEEE 802.11 link is detected. Also, routers store packets sent to the device while it's disconnected; as soon as device presence is detected, stored packets will be sent.
- **PRO** An implementation for Linux exists called FMIPL (Fast Handovers for Mobile IPv6 Implementation) [10].
- **CON** The Linux implementation runs over Linux kernel version 2.4.26. Its portability to 2.6 kernels is unknown.
- **CON** There's still no public release for this implementation.
- **CON** Current implementation only supports one mobile node.
- **CON** Wireless devices (802.11 protocol) are not yet supported on the Linux implementation.

Host Identity Protocol IP addresses are used as network interface identities and as topological locators; this double role limits the Internet architecture flexibility; in particular, transport protocols are bound to IP addresses and disconnect as soon as address changes.

Host Identity namespace divides name and locator roles. Using HIP, transport layer works with Host Identities instead of IP addresses. At the same time, networking layer uses IPs as pure locators.

Communications between hosts are made in a safe way creating a pair of Security Associations (SA), one for each direction.

- **PRO** Provides a number of interesting features like: mobility, multihoming, dynamic IP renumbering, protocol translation/transition and defense against certain DoS attacks.
- **CON** It is highly experimental. As of now, the working group is producing RFCs on the "experimental" track. Although there is a Linux port for

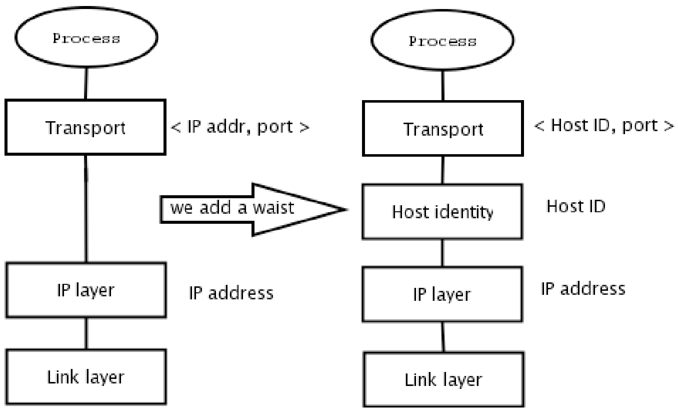

Figure 5.1: HIP Protocol

HIP (HIPL [18]), a light-weight HIP for small devices like the ones covered by XtreemOS-MD is still a work-in-progress.

Session Initiation Protocol is an application-layer signaling protocol for creating, modifying and terminating sessions with one or more participants [38]. It is text-based; this can be either a PRO (allowing for humans to read SIP messages and good for debugging) or a CON (wastes bandwidth).

PRO Being transport-layer-independent, SIP can be used over UDP, TCP, ATM and so on.

PRO It can be used on top of any regular IPv4 or IPv6 network without any special mobility support.

PRO Ångström distribution already ships with certain amount of SIP support.

CON Being an application-layer protocol, it requires applications to be modified in order to support mobility, a problem that the other lower-level solutions (like MIPv6) do not have.

NUTSS is a network architecture that uses signaling before establishing the data channel. While the data channel is a direct TCP/IP connection between two IP addresses and ports, the signaling is through SIP, which uses stable endpoint identifiers for routing [29].

CON Requires that all the routing/NAT boxes support this protocol, a fact that makes the deployment and adoption of this kind of solutions quite difficult.

5.2.1 Selected Solution

As it has already been pointed out, for the implementation of the mobility enhancements in XtreemOS-MD we have selected **MIPv6** technology. Several factors lead us to choose this solution over others:

- MIPv6 is based on IPv6, which is the most likely and future-proof network protocol as of this writing, even though it is still not deployed on a global scale (except in certain countries like China).
- Moreover, IPv6 is already a requirement for some of XtreemOS applications (see deliverable D4.2.3 [44]). Thus, it is only natural to choose an extension to this protocol for enhancing mobile devices' performance.
- MIPv6 provides better performance than other solutions for terminal mobility, with comparatively few handicaps.
- This solution can be combined with others (like FHMIPv6 or HMIPv6) to overcome some of these handicaps.
- And most important of all, these features are all provided in a transparent way, thus not requiring to change applications or (as it is our case) grid services, to obtain the benefits of its usage.

With regard to which implementation of MIPv6 for Linux to use, the best plan seems to be the adoption of **MIPL** [26], as it is the most actively developed and more thoroughly tested one, with an active community that continues to enhance it and strives for including it into the mainline kernel.

5.3 Description of mobility enhancements modules

5.3.1 MIPL

Mobile IPv6 extensions are still not integrated in the mainline kernel so, kernel must be patched to give it Mobile IPv6 support, as well as activating IPv6 support already present in the kernel.

The MIPL distribution is split in two packages necessary for MIPv6 support:

user space code User space code package consists of configuration files, initialization scripts, and specially the **mip6d daemon** to give IPV6 Mobility support.

kernel patch Since mobile nodes in MIPv6 require the presence of a special node in their home network, which takes care of the packets directed to them (i.e. the Home Agent), both the Mobile Node (MN) and Home Agent (HA) need their kernel patched, but in different ways. This kernel patch will be kernel version dependent.

5.3.2 Router Advertisement Daemon radvd

In MIPv6, mobile nodes arriving to a new network need to know who is the router of the new network, in order to get a new *care-of* address that belongs to this new network[1]. Router advertisement is the main mechanism used in this process.

IPv6 has support for autoconfiguration of the hosts of a network; in order to work, routers on the local network have to run a program which answers the autoconfiguration requests of the hosts. On Linux, this job is done by the **R**outer **Adv**ertisement **D**aemon, listening to router solicitations (RS) and answering with router advertisements (RA). Unsolicited RAs are also send from time to time.

Information inside RAs includes address prefixes, the MTU (Maximum Transfer Unit) of the link and information about default routers.

Since routers can't autoconfigure themselves, Radvd needs a radvd.conf file where values are set. The file contains one or more interface definitions of the form:

```
interface name {
        list of interface specific options
        list of prefix definitions
        list of route definitions
        list of RDNSS definitions
};
```

An example configuration file might look like this:

```
 interface eth0 {
        AdvSendAdvert on;
        prefix 5f15:9100:c2dd:1400:8000::0/80
        {
                AdvOnLink on;
                AdvAutonomous on;
        };
 };
```

For more information, please read man page of radvd.conf

5.4 Interfaces

5.4.1 MIPV6 related APIs

Understanding API as an entry point for application developers to access the mobility capacities of XtreemOS-MD (in this case, its MIPv6 capabilities), this module will not have any recognizable interface to any other XtreemOS component, as this

[1] Note that the any packets trying to reach the mobile node will be directed to the *home* address of the node, which does not change when it moves.

is precisely the goal of MIPv6, to provide terminal mobility to the higher layers (e.g. applications) *in a transparent manner*.

5.5 Examples of use

In order to fully understand the benefits of implementing these mobility enhancements in XtreemOS-MD, a number of scenarios that use this characteristics are presented here:

Nomadicity

A pharmaceutical company is set on several buildings across the city and, logically, each building has its own network. A user in the research building needs to talk and show some data heŠs working on to someone settled on the factory buildings. Thus, he takes his PDA with him and goes there. Then, in the factory building, he turns on his device and shows the results he had got. At the same time, the user's boss wants to access that very same data, which he knows is located in the user's PDA. The boss will have transparent access to the data, regardless of where the user is, as the requests are being redirected by the research building's Home Agent to the real location of the device. Note that network configuration and IP ranges may be different in both networks, but MIPv6 makes it transparent to the users.

Mobility

A user takes a taxi from his home to his office, where a meeting will take place in a couple of hours. Checking some files from his PDA he realizes that important data he needs has not been generated because a grid process was not launched. This user executes the job in the grid, using his PDA to keep an eye on the evolution of the job. This is done while moving, as his device has been switching networks to allow the session to be maintained seamlessly.

Chapter 6

Relationship with mobile Linux initiatives

During the last two or three years several consortia have been formed to identify and bridge gaps in the mobile Linux ecosystem, in order to promote its adoption in mobile devices. Much activity has been registered in this front during last year, including initiatives changing names and merging with each other - as it was the case of Open Source Development Labs (OSDL) and Free Standards Group (FSG) which merged into the Linux Foundation by the beginning of this year - as well as shocking announcements like Google's Android platform.

Furthermore, these organizations are aware of each other and seems to have the same intention of neither forking Linux nor fragmenting the industry. As a matter of fact, some of these initiatives have held several joint sessions and cross-organization presentations. Indeed, some companies are cross-involved in several organizations (Access, Trolltech, A la mobile, Motorola, Nokia, AMD, WindRiver, etc.).

Next, a brief explanation of the objectives of the most important Mobile Linux initiatives, and their potential relationship with XtreemOS-MD is discussed. A more detailed analysis of all of those initiatives can be found in appendix B.

6.1 Mobile Linux Initiative at the Linux Foundation (MLI)

The Mobile Linux Initiative (MLI) [36] aimed at assisting mobile industry in making Linux the best platform for mobile devices development. The MLI, founded by OSDL in 2005 and composed mainly by mobile hardware providers, became Mobile Linux Work Group [20] inside Linux Foundation [19] by the beginning of year 2007, when OSDL merged with FSG. The Mobile Linux Work Group inherited the mission of the former MLI, but interoperability got utmost importance among them. Therefore, providing developers with an interoperability standard, the corresponding mobile profile of the LSB, seems to be one of the most important tasks to be done by this Work Group.

XtreemOS-MD and MLI

The Mobile Linux Work Group is currently rechartering, focusing on three main issues: promotion of Linux in mobile space (by means of a Mobile Linux blog), providing legal support to device manufacturers and developers, and adaptation of the LSB to a mobile Linux development environment (although it does not seem that there is any direct contribution to LSB 3.2 specification to create such a mobile profile).

Since Mobile Linux Work Group belongs to the Linux Foundation and XtreemOS-MD must be consistent with overall XtreemOS policy, XtreemOS-MD should aim at being compliant with mobile profile of LSB, as long as XtreemOS is compliant with LSB. Nevertheless, a mobile LSB hasn't been published yet and it does not seem that any contribution to LSB 3.2 is being directly submitted by Mobile Linux Work Group.

On the other hand, a set of guideline specifications have been provided by MLI, regarding security, file system, developer tools and virtualization, and one more is foreseen addressing performance. As far as these guidelines could represent a preview of the mobile profile, XtreemOS-MD should follow them as far as possible depending on its Linux base distribution and LSB compliance of XtreemOS standard version.

6.2 Consumer Electronics Linux Forum (CELF)

The Consumer Electronics Linux Forum (CELF) [5] is a non-profit corporation formed in 2003 to promote the use of Linux in consumer electronic devices, including but not limited to mobile phones. The objective of CELF is to push projects developing Linux for such kind of devices by means of specifications for Linux enhancements, a code source tree and an Open Test Lab allowing for remote automated testing on different embedded boards. Several Work Groups have been established after a first period of identification of the areas of interests by the core members (consumer electronic devices manufacturers and their suppliers). These Work Groups aimed to create the specifications and review the implementations, while code development must be made by external companies, being or not CELF members.

XtreemOS-MD and CELF

The first CELF specification was released in June 2004, consisting of several parts: power management, system size, bootup time and security. One of the most active Work Groups, Mobile Phone Profile WG, has published for review a first release of the mobile phone functionality common API by January 2006, but CELF steering boards recommended dropping this activity due to low activity in the revision process. On the other hand, continuous CELF patches to Linux have been being gathered during last years.

CELinux and CELF specifications refer to an embedding operating system without taking into account the inclusion of mobile devices in a Grid. Anyway, it should be desirable that XtreemOS-MD as a mobile Linux operating system can integrate CELinux patches that enhance power management, system size or real-time behavior but, being these improvements out of the scope of XtreemOS-MD tasks, the CELinux integration must not be a need. On the other side, XtreemOS-MD as a Grid operating system could provide valuable feedback on decreasing power management or security issues (thanks to the security services defined in XtreemOS standard version).

Nevertheless, main external contributions to CELF are Linux modules and/or features developments fulfilling one or more specifications previously collected. Since it is out of the scope of XtreemOS to make developments of Linux kernel and modules other than those related to VO support and Grid services, there will not be many chances to submit official contributions under the current view.

6.3 GNOME Mobile and Embedded initiative (GMAE)

GNOME Mobile And Embedded (GMAE) initiative [13] was launched last April 2007 under the umbrella of GNOME project with the objective of building an open Linux platform based on GNOME components for mobile and embedded devices. Therefore, GMAE intends to ship a subset of the software components of GNOME desktop platform in mobile and embedded devices. GMAE initiative is supported by several open source community projects like OpenMoko or Maemo and other mobile Linux initiatives like CELF, Linux Foundation or LiPS.

XtreemOS-MD and GMAE

GMAE is a recent initiative but a first GNOME Mobile Platform has been already defined thanks to the GMAE members' long experience in GNOME components. Thus, a GNOME-like layout has been drawn including user interface, file accessing, service discovery, Bluetooth connectivity and other mobile functionalities components like Instant Messaging or Address Book. However, it doesn't seem to be much on going activity maybe because a clear roadmap is still missing.

Despite several common components can be found between GMAE layout and XtreemOS-MD architecture, it is more a consequence of the Linux base distribution selected in XtreemOS-MD than a strong decision on the design. Since XtreemOS-MD will try to provide Grid modules avoiding dependencies on the Linux base distribution as far as possible, and taking into account that the focus of GMAE initiative is far from XtreemOS-MD aims, no contribution is initially foreseen.

6.4 Linux Mobile (LiMo) Foundation

The Linux Mobile (LiMo) Foundation [21] aims at creating an open Linux-based software platform for mobile devices, and specifically mobile phones. This objective will be achieved not only by defining the reference architecture and the API specification of such an open mobile platform, but also by providing reference code modules and a test suite which will allow to demonstrate LiMo platform conformance. The LiMo Foundation was created by handset manufacturers and telecom operators in January 2007 and, even though other kinds of members have recently joined to this initiative (independent software vendors, software developers), the main target of the platform are mobile phones.

XtreemOS-MD and LiMo

The LiMo Foundation has defined an architecture reference including a kernel space (basically, the Linux kernel) and a user space consisting of: a middleware layer, including functionalities ranging from security framework to telephony capabilities, and the application layer. The first version of the platform is initially foreseen by the first quarter of 2008, with the advent of the first LiMo-compliant handsets, but only Foundation members are allowed to participate. Community participation is foreseen for the second and third releases (2ndquarter 2008 and 3rd quarter 2009, respectively).

There are some similarities in the scope of LiMo and XtreemOS architectures. For instance, XtreemOS-G components can be included at the same level as application engine layer and/or middleware layer, playing XtreemOS API the role of both application engine API and terminal service API. But besides the parallel structure of both architectures, LiMo platform is focusing on mobile phone features like multimedia and communication capabilities in a higher layer than XtreemOS-MD Grid services.

On the other hand, it is not clear what kind of collaboration is expected of non-members, and neither it has been specified how external contributions will be integrated. *Licensing scheme, that also includes proprietary developments, could be another barrier to integrate XtreemOS Grid modules in LiMo platform.*

6.5 Linux Phone Standards (LiPS) Forum

Linux Phone Standards (LiPS) Forum [23] is a telephony industry consortium founded in 2005 in order to define a standardized software platform which allows applications to be developed across any kind of Linux telephony terminal. Besides promotion of Linux adoption, this consortium pays special attention to interoperability issues by producing

- A standard API for Linux-based system services, to support user applications running on converging telephony devices

- Standard extensions to Software Development Kits, facilitating the development of those applications
- A testing methodology which allows to certify LiPS-compliant implementations

XtreemOS-MD and LiPS

As of this writing, LiPS has published a first draft of LiPS specifications (April 2007), which includes the description of the LiPS reference model and the specifications for the Address Book enabler, the Voice Call enabler and the User Interface services based on GTK+. These specifications will be part of the LiPS Enabler Release 1.0 to be delivered by the end of 2007, including the specification for other telephony capabilities. Next year, LiPS has planned to focus on the application framework, addressing lifecycle management, package management and inter-process communication.

Taking into account that LiPS specifications only deal with interfaces for telephony related services and behavioral requirements for those services, the overlap between LiPS reference architecture and XtreemOS-MD is almost non-existent. Maybe LiPS Operating Services, which provide a set of protocols and hardware independent interfaces as well as a subset of LSB and POSIX, are the intersection area with XtreemOS, but it is out of the main scope of LiPS.

To sum up, since XtreemOS-MD will not develop the telephony components of the LiPS reference architecture, the mutual feedback will be reduced to several isolated components (package management and security modules). Nevertheless, as Linux kernel and OS services will be defined by LiPS through the collaboration with other initiatives, other kind of collaboration could be envisioned in the long term (when addressing XtreemOS-MD advanced version for mobile phones).

6.6 Open Handset Alliance

The Open Handset Alliance initiative [30] has been announced at the beginning of November by Google, who leads the consortium of more than 30 companies from the mobile industry including mobile operators (Telefónica among them), semiconductor companies, handset manufacturers and software and commercialization companies. The intention of such a consortium is to foster the development of innovative mobile applications. And to that aim, Open Handset Alliance is developing a new open Linux-based platform called Android, which is much more than a theoretical framework. Since a first version of the Android SDK can be already downloaded, the target of the Open Handset Alliance is the deployment of handsets and services using Android platform by second half of 2008.

XtreemOS-MD and OHA

The launch of Android only one week after the announcement of the Open Handset Alliance is a symptom of the impact that this initiative could have in the mobile arena. Android is an open mobile platform including a Linux-based operating system, a middleware layer, an application framework, a SDK for developers including an Eclipse plugin, an Android emulator, and several application examples. Furthermore, a complete set of documentation has been provided as well as some explanatory videos published in the Android Youtube channel. Open Handset Alliance members are working on developing Android-based smartphones, while third party developers are invited to develop their own applications by means of the Android Developer Challenge (with an overall mount of $10 million for awards).

Despite Android could be considered to be a threat for XtreemOS-MD, as long as Android is built on an open Linux kernel, it should be considered a good opportunity, as one of the threats for XtreemOS-MD is that a Mobile Linux OS will not be that widespread and Telefónica is a founder member of the Open Handset Alliance. It must be kept in mind that the point of XtreemOS-MD is to provide modules for grid functionality that can be integrated into any mobile Linux.

In overall, Android architecture has a common layout with XtreemOS-MD, but there are many differences between them that should be thoroughly analyzed in order to determine their compatibility. However, no deep study can be done at this stage, since only the Android SDK is available.

6.7 Summary and conclusions

Table 6.1 summarizes the main results of our research into the mobile Linux initiatives with regard to XtreemOS-MD interaction:

We can conclude that, with the current state of affairs in the standardization of mobile Linux, few interactions with these organizations can be foreseen from XtreemOS, due to its status as a research project. The most likely ways of collaborating with these initiatives are:

- To make XtreemOS-MD comply with their specifications and guidelines, *once they are publicly available*. Taking into account that the scope of XtreemOS and most of these initiatives is quite different, the safest way of doing this is integrating XtreemOS-MD packages into an already-compliant mobile Linux distribution.
- Trying to include grid functionalities into the specifications and functionalities described by the initiatives. It is quite unlikely that XtreemOS alone can achieve this goal, unless the benefits of mobile access to grids are more clearly perceived within the mobile device industry.

Initiative	Probability of interaction	Notes
MLI	Medium	Compliance with mobile profile of LSB will be desirable if XtreemOS standard version fulfills LSB and the mobile profile version is available (not yet).
CELF	Medium	Since CELF specifications does not take into account Grid capabilities and CELF external contributions are focused on fulfilling the specifications, XtreemOS-MD contributions should demonstrate their usefulness to non-Grid users.
GMAE	Low	GMAE focus (software desktop platform) is far from XtreemOS-MD aims (supporting Grid capabilities and providing Grid services)
LiMo	Low	It is not clear what kind of collaboration can be achieved since it is not specified how external contributions will be evaluated and accepted (so far, only LiMo members are working on the first release of the platform.
LiPS	Low	Since XtreemOS-MD is not concerned about telephony components, the interaction is reduced to isolated components out of the focus of LiPS and in the long term (LiPS targets telephony devices).
OHA	To be determined	Although no thorough study is possible since only Android SDK is available, on the one hand Telefónica is member of this initiative but on the other, again the aim of OHA is quite different from XtreemOS-MD.

Table 6.1: Mobile Linux initiatives and their relationship with XtreemOS

Chapter 7

Future Work

7.1 Next Steps

The next phase toward the completion of the first basic version of XtreemOS-F for mobile devices is the implementation itself, which should be completed by April 2008 (M22), when the *first internal release* of the software is made. After a period of two months for testing and integration with the existing XtreemOS components, the *final release* of the basic version should be ready by June 2008 (M24). Afterwards, the software will be packaged by WP4.1, in order to have the *final packaged version* by August 2008 (M26).

In parallel with this packaging, the tasks related to the new advanced version of XtreemOS-MD targeted to mobile phones will begin. This will include taking a new decision on which is the best distribution to take as a base for these new target devices, and revising the requirements and specifications of Linux-XOS for mobile phones, including new research prospects as the ones detailed below.

7.2 Further Research Prospects

There are a number of issues that have been detected during the specification and design phases of this workpackage, and that could be worth analyzing in the advanced version of XtreemOS-MD. These issues include:

NAT, Teredo and Mobile IPv6

The issue of how to deal with Network Address Translation (NAT) and firewalls still remains unresolved on a project-wide scale. This and other workpackages will probably have to find a way to solve it if XtreemOS is to be widely used in practice. The usage of technologies like STUN or Teredo should be taken into account on a component-by-component basis, to detect potential problems in their usage.

It also remains to be investigated how these technologies can be integrated with modern protocols like Mobile IPv6, the one used in XtreemOS-MD to provide terminal mobility.

Integrating ad-hoc networks in the Grid

Ad-hoc networks, and more concretely mobile ad-hoc networks (MANETs), have already been spotted as a new technology that could be useful in certain scenarios (for example, crisis management situations), where grids also could provide advantages. As integration of mobile devices is the first step into extending the reach of grids nearer to the final user, integrating ad-hoc networking technology into XtreemOS could be a further step in that direction, bringing grid functionalities to users where virtually no infrastructures exist.

Integrating XtreemOS into telephony networks

It also remains a field of investigation how to properly integrate grid and next-generation networks (NGN), so that both technologies can exploit the benefits of each other, like using the telephony network's trust base (the operator) in a grid context.

Appendix A

Specification Lists

In this appendix, all the software dependencies are listed, including kernel options and modules, libraries, as well as the processes (e.g. daemons) which need to be running so that the software components described in the document can function properly.

A.1 Kernel options needed

The **VO support** features of XtreemOS-MD need the following kernel options in the *mobile devices*:

```
CONFIG_KEYS=y
CONFIG_KEYS_DEBUG_PROC_KEYS=y
CONFIG_SECURITY=y
CONFIG_SECURITY_NETWORK=y
CONFIG_SECURITY_CAPABILITIES=y
```

For activating the **terminal mobility** support, the following kernel options are needed in the *mobile nodes* (MDs) and *home agents*:

```
CONFIG_EXPERIMENTAL=y
CONFIG_SYSVIPC=y
CONFIG_PROC_FS=y
CONFIG_NET=y
CONFIG_INET=y
CONFIG_IPV6=y
CONFIG_IPV6_MIP6=y
CONFIG_XFRM=y
CONFIG_XFRM_USER=y
CONFIG_XFRM_ENHANCEMENT=y
CONFIG_IPV6_TUNNEL=y
CONFIG_IPV6_ADVANCED_ROUTER=y
```

```
CONFIG_IPV6_MULTIPLE_TABLES=y
```

Moreover, in the *mobile node* (i.e. the mobile device) the following options are also needed:

```
CONFIG_IPV6_SUBTREES=y
CONFIG_ARPD=y
```

And in the *home agent* node (i.e. a static node in the home network of the mobile devices), these options are also needed:

```
CONFIG_INET6_ESP=y
CONFIG_NET_KEY=y
CONFIG_NET_KEY_MIGRATE=y
```

A.2 Network parameters needed

In order for the **terminal mobility** support to be operational, the following network parameters must be tweaked in the *mobile node*:

```
# echo 0 > /proc/sys/net/ipv6/conf/eth0/forwarding
# echo 1 > /proc/sys/net/ipv6/conf/eth0/autoconf
# echo 1 > /proc/sys/net/ipv6/conf/eth0/accept_ra
# echo 1 > /proc/sys/net/ipv6/conf/eth0/accept_redirects
```

And also, the following parameters are needed in the *home agent* node:

```
# echo 1 > /proc/sys/net/ipv6/conf/eth0/forwarding
# echo 0 > /proc/sys/net/ipv6/conf/eth0/autoconf
# echo 0 > /proc/sys/net/ipv6/conf/eth0/accept_ra
# echo 0 > /proc/sys/net/ipv6/conf/eth0/accept_redirects
# ip route add (as needed)
```

A.3 Software packages needed

In order to get the **VO support** features in XtreemOS to run on a Linux mobile device, the following software dependencies must be fulfilled:

- Linux kernel 2.6
- glibc2 (or eglibc with NSS support)
- libpam
- openssl v0.98 or higher

- libz
- libdb4
- libkeyutils

A.4 Processes needed

VO support needs the following processes to be running on the *mobile devices*:

- AMS Daemon (see section 4.3)

Terminal mobility support also needs the following processes to be running in the *mobile nodes*:

- mip6d Daemon

In the *home agents*, the processes that need to be running are:

- mip6d Daemon

And finally, the *routers* need to run the following processes:

- radvd Daemon

Appendix B

Mobile Linux

B.1 Introduction: a mobile Linux world

Linux arrived late to the PDA operating system market, but the trend toward the mobile devices convergence (including telephony capabilities in computing devices like PDAs, as well as increasing computing power in mobile phones) and the advantages of Linux operating systems (reduced development and deployment costs, added value and open source features) are leveraging the adoption of Linux as mobile handsets operating system [12].

However, the use of Linux in mobile and embedded devices is relatively new and therefore there is not any widely adopted standard or convention for mobile Linux, or about integrating a Linux OS in mobile devices. There are currently several mobile industry and Linux community initiatives, Free/Libre/Open Source Software (FLOSS) projects and collaboration platforms seeking to foster (but also influence) the adoption of mobile Linux. Those initiatives are more focused on mobile phones (or at least mobile devices with telephony capabilities) than Personal Information Management (PIM) devices like PDAs.

In this section, the work of the most influential Linux initiatives will be explained from the point of view of XtreemOS-MD, that is, taking into account the potential collaboration and/or compliance with their specifications by XtreemOS-MD. Notice that we have considered GMAE a Linux community initiative and not only a FLOSS project because it fully belongs to GNOME initiative and has defined a layered platform (even if this platform represents Linux developments more than agnostic components).

It should be noticed that even though all of these initiatives and organizations can be focused on different areas of Mobile Linux environment, it is unavoidable to have several overlaps among them.

B.2 Mobile Linux initiatives

Following, the most important mobile Linux initiatives will be introduced from both technical and administrative point of view. For each initiative, objectives, members, work in progress and administrative issues will be addressed.

B.2.1 Mobile Linux Initiative at the Linux Foundation (MLI)

Objective

The Mobile Linux Initiative (MLI) [36] was created by the Open Source Development Labs (OSDL) on October 2005 with the aim of assisting mobile industry in making Linux the best platform for mobile devices development and deployment. By the beginning of this year, OSDL and Free Standards Group merged into the Linux Foundation [19], and MLI became Mobile Linux workgroup [20] inside this new foundation. The Mobile Linux workgroup, formed by major companies involved in the mobile industry, has as its mission to accelerate the adoption of Linux next-generation mobile handsets and other converged voice/data portable devices.

Taking into account that one of the main activities of the Linux Foundation is to deliver Linux Standard Base (LSB) to provide developers with an interoperability standard, the Mobile Linux workgroup will provide the corresponding mobile profile for the LSB.

Members

The founding members of the Linux Foundation are Fujitsu, Hitachi, HP, IBM, Intel, NEC, Novell and Oracle. Besides them, a number of companies are members of this foundation, from different technological areas like Linux industry (RedHat, RedFlag, Montavista, a la Mobile, Access, Trolltech), CPU and chipset manufacturers (AMD, Sun Microsystems), handset manufacturers (Nokia, Motorola, Siemens), Original Equipment Manufacturers (Alcatel-Lucent, CISCO), telecommunication operators (NTT, BT), other major companies (DELL, Google) and universities (Portland State University, Tokyo University of Technology). Despite not all of them contribute to the Mobile Linux workgroup, this workgroup comprises the whole spectrum of the mobile ecosystem.

Work in progress

The Mobile Linux workgroup is currently rechartering, focusing on three main issues:

- Promotion of Linux in mobile space: a Mobile Linux blog with a two-fold target, mobile and industry and Linux community, has been created; moreover, a “Mobile Linux Weather Forecast” is planned to be published; and collaboration with other organizations to promote mobile Linux activities by each organization is also planned.

- Provide legal support to device manufacturers and developers, for instance regarding licenses and the ability to change and update handsets.
- Adapt the LSB and its test suite to the mobile Linux development environment.

The work developed by the Mobile Linux workgroup so far consists of a series of guideline specifications, based on existing open source prototype implementations. There are four guidelines published (security, file system, developer tools, virtualization) and one more to be published (performance). Following, a brief analysis of these guidelines is provided:

Security The security guidelines intend to improve resilience to attacks on mobile Linux based systems and to ensure operational integrity of mobile devices. Specific guidelines are provided regarding confidentiality (robust set of cryptographic primitives), origin integrity (access and use of resources policy), and resistance to external threats (authentication and resource management).

Filesystem Operational and functional guidelines deal with the operation and management of mass storage used in mobile devices, so that Linux Virtual File System will be able to support multiple file system schemes on a single system. They seem to be very low level guidelines (boot-time speed, file system mount time, support for a Flash "configuration area", data compression or fast indexing) and not directly related to the requirements of a Grid file system.

Developer tools The guidelines to develop software for Linux mobile devices are divided in four categories, toolchain (cross compilation to create software), application binary interfaces (supporting of Executable and Linking Format – ELF), library (interoperability – glibc version 3.2.4 – and configurability), and debug features (architecture, format – DWARF – and user and kernel debugging modes).

Virtualization These guidelines try to merge the need for extensive legacy software reuse, the need to adopt additional new OS suites to execute new applications, and the need to deliver new features on a hardware constrained device. Virtualization is the selected choice, enabling the execution of multiple OS environments on a single processor, each environment being isolated within a dedicated partition.

Furthermore, the MLI workgroup tries to identify gaps in Linux kernel capabilities as well as in the software stacks above Linux, in order to instigate new projects which will tackle these gaps. On the other hand, the MLI workgroup is also in charge of the definition of the mobile profile for the Linux Standard Base (LSB). However, currently it does not seem that there is any direct contribution to LSB 3.2 specification in order to create that kind of LSB mobile profile.

Administrative standpoint

The Linux Foundation allows several kinds of members, platinum (founding) members, gold members, silver members which must correspondingly pay an annual fee of $500000 USD, $100000 USD and $20000 USD. There are also up to seven affiliate classes ranging from national government ($20000 USD fee) to individual affiliates ($25 USD fee), but without specific mention to collaboration of projects like XtreemOS.

Nevertheless, as the Linux Foundation is an open organization, there are other ways to be involved in its work, for instance joining and participating in workgroups mailing lists, or getting the LSB certificate. Specifically, the collaboration to the Mobile Linux workgroup can be carried out at first instance joining the MLI mailing list and eventually participating in the Mobile Linux Blog.

Apart from collaborating in Linux Foundation workgroups, there exist several collaboration fora (user and vendor advisory councils, technical advisory board, LSB steering committee) and foundation programs. The last ones provide funds for collaboration projects to accomplish key projects for Linux success.

B.2.2 Consumer Electronics Linux Forum (CELF)

Objective

The Consumer Electronics Linux Forum (CELF) [5] is a non-profit corporation formed in 2003 by Matsushita and Sony to promote the use of Linux in consumer electronic devices. The target of CELF is two-fold: (1) to publish specifications for enhancements that are desired in Linux, and (2) to make available a set of changes to Linux in the CELF source tree.

In summary, CELF will try to leverage the advancement and adoption of Linux as an open source platform for consumer electronic devices extending beyond just mobile handsets. Therefore, consumer electronic devices manufacturers and their suppliers are the core members of this forum.

Members

Matsushita, Sony, Hitachi, NEC, Philips, Samsung, Sharp and Toshiba can be considered the founder members of CELF. But there is a large list of members, including other CE manufacturers and suppliers (Fujitsu, HP, IBM, LG, Mitsubitshi, Motorola, Nokia, Panasonic, Sanyo, Sharp, Sun Microsystems, Thomson), CPU and chipset manufacturers (AMD, ARM, Intel, Texas Instruments), Linux related companies (Access, FreeScale, Montavista, Sun Wah Linux, Trolltech, WindRiver) and telecommunication operators (France Telecom). The impact of CELF is worldwide, as the members come from Asia, US, Europe and Australia, with a strong Asian core.

Work in progress

During a first stage, the Forum gathered requirements from its Member companies and several areas of interest were identified (real-time performance, memory usage, power management, bootup time, reliability) in order to establish different working groups. The working groups should create specifications and review implementations, but while both activities are considered part of the Forum work, the code is developed by external entities (either Members or non-Members) and submitted to the Forum for approval and inclusion in the CELF source tree.

Up to now, the more active working groups are the Real Time WG (devoted to improve the real-time capabilities of the Linux OS); the Audio, Video and Graphics WG (dealing with AVG requirements for CE devices); the Security WG (focused on establishing security requirements and evaluating candidate architectures and implementations) and the Mobile Phone Profile WG (which works on providing a Linux base for building mobile phones). This last one has been working on a common API for mobile phone functionality and a first release was published for review by January 2006. However, the Architecture Group and Board of Directors recommended dropping this activity due to the slow revision progress and the narrow focus.

Furthermore, CELF also provides a test lab, Open Test Lab, aiming at supporting remote automated testing on embedded target boards. Despite it is not a task for CELF, a large number of patches have been produced and included in the CELF source tree under the name of CELinux patches. However, CELF intention is that patches were incorporated into the mainline Linux kernel source tree, or in the corresponding SourceForge project for the technology the contributor is working on (if exists).

CELF specification v1.0 R2 was released in June 2004. It is divided in several parts:

- Bootup time specification, dealing with reducing the time required to boot a Linux kernel in a CE device
- Power management specification, introducing a power management architecture to allow applications for optimization of power use in battery-driven mobile devices
- Audio/Video/Graphics specification, addressing the different standards and specifications existing for audio, video and graphics processing
- Real-time specification, with the aim of improving Linux kernel and APIs to get a better real-time operating system

- System size specification, defining a method to compare Linux configurations size as well as a set of standard mechanisms for reducing system size
- Security specification (covering data security, privacy protection, system stability and data integrity

Besides these specifications that refer to an embedding operating system without taking into account the inclusion of the mobile device in a grid, it should be noticed two working groups whose work could be related to XtreemOS-MD: Security WG and Mobile Phone Profile WG. The first one is currently studying a set of Linux security frameworks (e.g. Linux Security Modules), components (e.g. SELinux) and features (e.g. NX patch). The MPP WG has been developing an API for handset functionality. Next, the reference architecture, which defines the different system layers and components, is introduced.

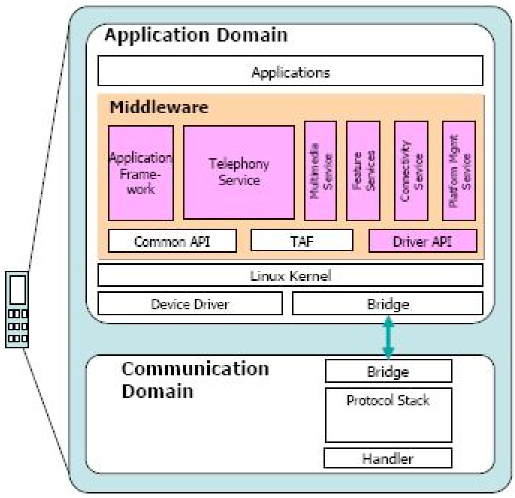

Figure B.1: CELF Mobile Phone Profile software architecture

The MPP WG architecture differentiates hardware (communication domain) and software (application domain) but it does not detail what kind of Linux modules are expected in the middleware layer. "Driver API" component provides upper layers with an abstract interface enabling development of portable middleware and applications programs amongst mobile devices. Despite this CELF Architecture Group recommended dropping the effort of developing this API, MPP WG will continue the work on reference architecture and mobile phones profile.

Administrative standpoint

There are several kind of members: statutory members (founding and appointed members) with the right to vote on Board meetings, and non-statutory members (associate members, supporting members, and special supporting members). All of them, except the special supporting members, have to pay an annual fee of $16000 USD, $12000 USD, $8000 USD and $4000 USD respectively. In spite of the fact that Working Group activities can be influenced by existing open source projects (suggestions can be sent to arch@celinuxforum.org), without being a member of the Forum, this influence will be determined by the overall impact of the project (e.g. XtreemOS) in the Linux community. It seems that CELF is not interested in adding Grid services by themselves, but only if they have impact on other features in mobile devices (as higher security or lower power consumption).

With regard to Intellectual Property Rights, the specifications produced by the forum will be released under a copyright license which allows free distribution and use. Source code will be made available under an Open Source license, usually the GNU General Public License (GPL) or GNU Lesser General Public License (LGPL).

B.2.3 GNOME Mobile and Embedded initiative (GMAE)

Objective

GNOME Mobile And Embedded (GMAE) initiative [13] focuses on building a mobile and/or embedded platform around the GNOME components already in wide use in other platforms. GMAE initiative was announced last April 19th 2007 during Embedded Linux Conference, but several meetings on this subject were held during the second half of year 2006 aiming at promoting the embedded usage of GNOME software in mobile devices. A list of GMAE contributors was identified as well as supporters from the open source community (projects like OpenMoko or Maemo) and industry (other Linux embedded initiatives like CELF, Linux Foundation or LiPS).

Therefore, GMAE intends to ship a subset of the software components of GNOME desktop platform in mobile and embedded devices.

Members

During Embedded Linux conference, several organizations announced their support to GMAE initiative. Amongst them, the GNOME Foundation supporters Access, Canonical, Debian, Igalia, Imendio, Intel, Nokia, OLPC, OpenedHand and RedHat. Moreover, potential GMAE contributors were identified, including CodeThink, Collabora, FIC, Fluendo, Kernel Concepts, Movial, Nomovok, Openismus, Vernier, Waugh Partners and Wolfson Microelectronics.

Work in progress

As part of GNOME platform, GMAE or GNOME mobile will focus on the mobile experience by integrating GNOME desktop and GNOME mobile technology. Even though GMAE is a quite recent initiative, a first GNOME Mobile Platform has been defined as a subset of GNOME platform.

But apart from this first GMAE components layout and the identification of several technologies candidates to be integrated in GMAE architecture, there has not been much discussions on GMAE mailing list (hardly two or three threads a month). Therefore, though the initiative has defined its mission (mainly building and promoting a GNOME mobile profile for mobile and embedded devices), specific actions and a clear roadmap are still to be determined.

Being GMAE the mobile initiative of GNOME, GMAE concentrates on providing an open source desktop environment for Linux users as well as a development framework for building applications integrated in such environment.

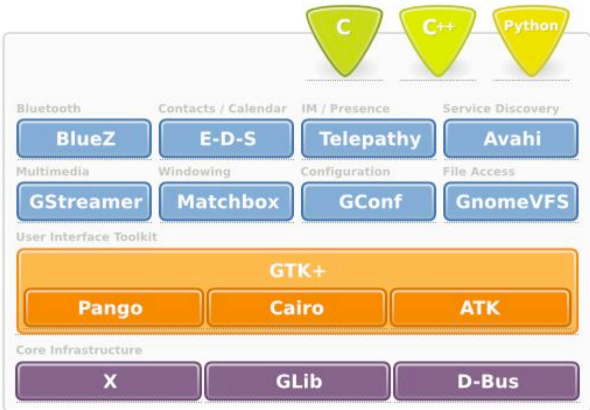

Figure B.2: GMAE software stack

GMAE software stack includes open source components like BlueZ (to integrate Bluetooth wireless standard), GTK+ and X libraries (to support the graphic environment), Glib (to provide the basics for building GNOME libraries and applications), and D-Bus (as a way of intercommunicating processes) among others. But all these core components of GMAE are outside the scope of XtreemOS-MD.

Administrative standpoint

Participating in GMAE implies participating in GNOME. Members and non-members can contribute in the same way, because the difference is that members get the right to vote for the Board of Directors (BoD), organize meetings and infrastructures, etc. But there is no need to be a voting member to contribute and influence the direction of GNOME. Moreover membership in the GNOME Foundation requires that the candidate has contributed to a non-trivial improvement in the GNOME Project.

GMAE platform is distributed under the terms of the LGPL following GNOME licensing strategy. This kind of license allows for its royalty-free use in proprietary software.

B.2.4 Linux Mobile (LiMo) Foundation

Objective

The LiMo (Linux Mobile) Foundation [21] aims at reducing the development costs and augmenting the flexibility in mobile industry, through the creation of an open Linux-based software platform for mobile devices. Despite the founding members come from mobile industry (mainly handset manufacturers and operators), the membership is open to other companies and organizations, like Independent Software Vendors and Software Developers.

Therefore, the LiMo Foundation intends to deliver a mobile software platform reference implementation comprising an API specification, several reference code modules and a test suite to demonstrate product conformance to the platform specification, which will allow products to display the LiMo Foundation logo.

Members

Motorola, NEC, NTT DoCoMo, Panasonic Mobile Communications, Samsung Electronics, and Vodafone are the founder members of the LiMo Foundation, covering several roles in the supply chain: from handset manufacturers to network operators. Other companies have recently joined to the Foundation either as Core members (Aplix, Celunite, LG Electronics, McAfee, and Wind River) or Associate members (ARM, Broadcom, Ericsson, Innopath, KTF, MontaVista Software and NXP B.V.).

Work in progress

The ongoing activity of the Foundation will be the continuous review of the architecture and development of the API and common code components. The current platform architecture proposed by the LiMo Foundation consists of a kernel space, basically the Linux Kernel, and a user space including the middleware layer and the application layer. Whereas the application layer is responsible for launching applications and defining the look and feel of the user interface, the middleware layer provides a set of daemons and services under a plug-in architecture. Functionalities like Registry (data storage) and Security (security policy enforcement) and frameworks like Telephony (network registration, call control, SIM / USIM handling, etc.) or Messaging (SMS, MMS, i-mode) are all part of middleware layer.

A first version of the platform is initially foreseen by the first quarter of 2008, and only Founder members will participate on it. The second version, in which all members will be allowed to contribute, is scheduled by the second quarter of 2008. Finally, a third version is already foreseen for the third quarter of 2009. Several Working Groups have been defined, one for each main component in R1 platform.

Despite the fact that this initiative is not limited to mobile phones, it seems to be quite oriented to this sort of devices, since there are several important components in LiMo R1 platform directly related to mobile phone features, like multimedia and communication (telephony, messaging, Digital Rights Management) capabilities.

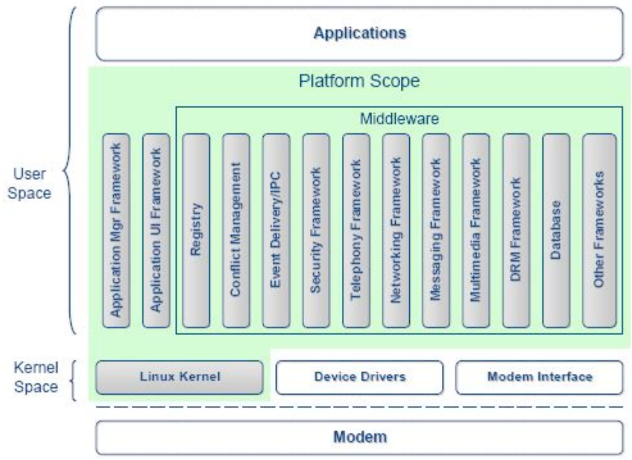

Figure B.3: LiMo Foundation software architecture

The architecture consists of two different spaces: kernel space and user space, being applications the higher layer of the user space. LiMo R1 platform has specified a minimum version for the kernel (2.6) and the libraries (2.4.x) and intends to avoid kernel changes as far as possible. Despite LiMo Foundation is developing its own components, it also foresees to point to other open (or proprietary) components where appropriate.

Administrative standpoint

The LiMo Foundation has a membership structure consisting of three types of member: founder members (who will become core members in 5 years time), core members and associate members. Despite the figure of non-member is recognized to develop applications, it seems quite difficult to influence on LiMo architecture without being a member. Nevertheless becoming a member of the LiMo Foundation requires the payment of an annual fee, \$800000 USD by founder members, \$400000 USD by core members and \$40000 USD by associate members. In addition to this, a membership agreement must be signed. In order not to prevent the participation in the Foundation, there are several license types from open source (GPL, LGPL, BSD, etc.) to proprietary (but terms must be offered to members on a non-discriminatory basis). One specific type of license, Foundation Public Licenses (FPL) is created to offer modules among the partners without copyright license fee.

B.2.5 Linux Phone Standards (LiPS) Forum

Objective

Linux Phone Standards (LiPS) Forum [23] is a telephony industry consortium aiming at producing standard APIs for Linux-based system services supporting user applications and services running on mobile and converging telephony devices, standard extensions to Software Development Kits facilitating the development of those applications and services, and a testing methodology which allows to certify LiPS-compliant implementations.

In summary, LiPS Forum was founded in 2005 in order to define a standardized software platform which allows applications and services to be developed and deployed across Linux telephony terminals, paying special attention to interoperability issues as well as promotion of Linux adoption.

Members

LiPS Forum members come from different sectors of the telecom industry and Linux community, including operators and service providers (France Telecom, Telecom Italia SpA, Bouygues Telecom, British Telecommunications), device manufacturers (Cellon, Freescale, Huaewi, Longcheer, NXP, Purple Labs, Spreadtrum communications, Texas Instruments, ZTE), and software vendors or Linux related companies(ACCESS, a la Mobile, ARM, Celunite, Esmertec, MIZI, Montavista, Movial, Open-Plug, Trolltech, VirtualLogix). There is an European core in LiPS, although members are distributed worldwide. Anyway, any related company or organization with an interest in the telephony sector and LiPS work is free to join the forum

Work in progress

The LiPS Forum has been working on creating an application environment for Linux mobile handsets, including usage profiles and functional requirements (jointly with OMTP and OMA), APIs defining Linux layers enabling applications in phones, SDKs and a testing and certification methodology. There are two working groups, the Requirement WG, whose mission is to define high-level requirements both functional (e.g. User Interface services) and non-functional (e.g. security constraints), and the Architecture WG devoted to obtain technical requirements from high-level requirements in order to define the LiPS Forum reference architecture.

As of this writing, LiPS has published a first draft of LiPS specifications (April 2007), which includes the description of the LiPS reference model and the specifications for the Address Book enabler, the Voice Call enabler and the User Interface services based on GTK. These specifications will form part of a wider release (LiPS Enabler Release 1.0) to be delivered by the end of year 2007 including the specification for other telephony capabilities (Telephony API, Messaging, IM and presence, etc.).

For year 2008 LiPS has planned to work on application framework (interprocess communication, package management, application lifecycle management), device management APIs (OMA DM), service APIs (IMS, local connectivity) and enabler APIs (multimedia).

LiPS reference model distinguishes five group of services sets:

- Application Management (AM) services, which implement application lifecycle including security models for accessing platform services
- User Interfaces (UI) services, only used by application services to configure the UI look and feel
- Enabler services, including application engines and servers that must not depend on AM and UI services
- Operating System services, providing the set of protocols and hardware independent interfaces to peripherals, as well as a subset of LSB and POSIX interfaces known as "Core Services"
- Platform Management services, aimed at the platform configuration, maintenance and software update

LiPS 1.0 specifications introduce this reference model and provide the specifications for several components of the UI services (text input methods, widgets set and key navigation), and two Enabler services (Addres Book and Voice Call). But LiPS is mainly dealing with mobile handsets which integrate telephony services, and even though LiPS reference model is referring to Linux kernel modules and OS services, these parts are out of the primary LiPS focus (UI services, telephony enablers, device management).

Administrative standpoint

The LiPS Forum allows three levels of membership: sponsor members, associate members and supporter members. Becoming a member seems to be a need to participate in the working groups and watch the work which is carrying out inside the Forum. Furthermore, in order to influence the forum direction and its outputs, it is necessary to be an sponsor member. It has a peculiar strategy for voting purposes, as the votes of each member are weighted according to the annual membership fee: the more a member pays the more number of votes it will have. The annual

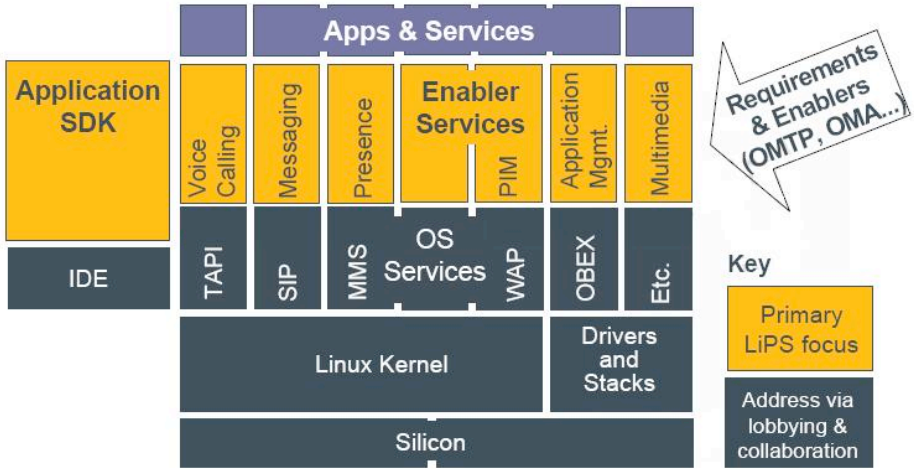

Figure B.4: LiPS software components

fee varies from 100000 € (sponsor members without any discount) to 5000 € (associate member). Since this Forum does not intend to produce source code, there is no specific policy regarding licensing of LiPS Forum outputs.

B.2.6 Open Handset Alliance

Objective

The Open Handset Alliance [30] is the most recent initiative to stimulate development of innovative mobile applications. To that aim this initiative is supporting a new open Linux-based platform called Android, developed by Google as a consequence of the acquisition of the same name company. Both the Open Handset Alliance and Android platform were announced last November 5th and an early version of the Android SDK was available since November 12th.

In order to make Android a commercial success, more than 30 technology companies from the mobile ecosystem were involved in the Open Handset Alliance by Google, who is leading Android. The aim of the initiative is to deploy handsets and services using Android platform for the second half of 2008. Despite it is not clear how an organization can become a member of this Alliance, companies contributing to openness in the mobile world are encouraged to join.

Members

The founder members of the Open Handset Alliance are 7 mobile operators from all over the world (including China Mobile, NTT Docomo, T-Mobile, Telecom Italia, Telefónica), 9 semiconductor companies (Broadcom, Intel, Marvell, NVidia, Texas

Instruments among them), 4 handset manufacturers (HTC, LG, Motorola and Samsung), 10 software companies (like eBay, Google or Nuance), and 4 commercialization companies (Aplix, Noser, TAT and WindRiver). Despite mobile industry is well represented by telecom operators and mobile equipment manufacturers, a noteworthy number of companies developing applications for mobile devices are also involved.

Work in progress

The first joint project of the Open Handset Alliance is Android [1], an open mobile platform including a Linux-based operating system, a middleware layer, an application framework, a SDK for developers and several application examples .

The launch of Android by Google at the same time than the previously unknown Open Handset Alliance has got a great impact on the mobile industry, because this announcement was the culmination of a long period with continuous rumors and a great expectation about the gPhone (a softphone which was supposed to be developed by Google). Google has far exceeded the expectations with not only a softphone but a whole Linux-based mobile platform which lets developers access and use the Android code as well as build their own applications over this platform.

Only in a week time Google has put Open Handset Alliance at the head of Mobile Linux Initiatives. Opposite to most of the other initiatives, the Open Handset Alliance did not stop at defining a theoretical framework; quite the contrary, Google has provided a detailed architecture design and a SDK - including an Android emulator and an Eclipse plugin - only a week after Android (and Open Handset Alliance) announcement. Moreover, a complete set of documentation has been released and some explanatory videos have been published in its own Youtube channel.

Members of the Open Handset Alliance are supposed to be working on developing handsets based on Android as well as services which can be killer applications to foster this platform. Despite the first services and handsets integrating Android are foreseen by the second half of year 2008, Google intends to get developers community involved from the very beginning, offering several awards in the so called Android Developer Challenge, consisting of a global amount of $10 million to be distributed to developers building the best applications over the platform SDK.

The Android architecture is composed by several layers, from the Linux Kernel layer to the application layer, where applications from providers and developers are supposed to run. It should be noticed that although it is not explicitly stated, Android is mainly targeting mobile phones and their evolution to smartphones.

Android platform includes a set of C/C++ core libraries (an implementation of libc tuned for embedded devices, a media library based on OpenCORE, a database engine, and several libraries dealing with 2D and 3D graphic interface among others), a Java runtime engine based on the Dalvik virtual machine that tries to re-

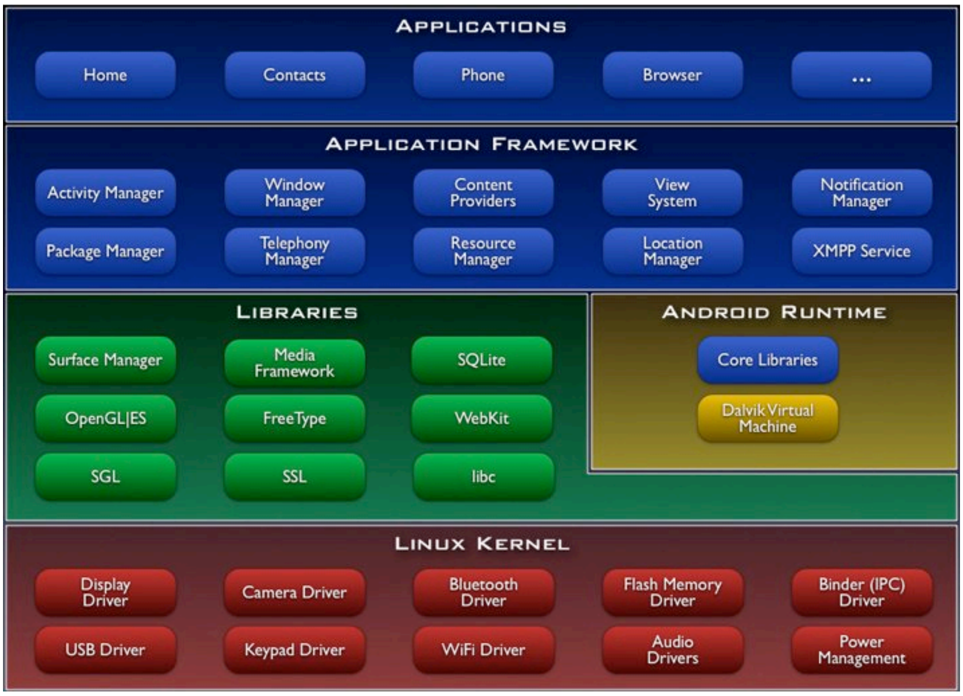

Figure B.5: Android platform architecture

duce the memory footprint, and an application framework including an Activity Manager (which manages the application lifecycle), a Notification Manager (enabling applications to display customized alerts), a Resource Manager (to provide resource access), Content Providers (to share data among applications) and a set of Views (e.g text boxes, lists, buttons, etc. used to build applications).

Administrative standpoint

It is not clear how organizations can contribute to the Open Handset Alliance in a way different to developing applications over Android SDK without being a member of the initiative. Fortunately, Telefónica is a founding member, so bi-directional feedback could be easily get.

Anyway, Android will be made available as open source via the Apache v2 license, which allows manufacturers and mobile operators to use the platform without the requirement to contribute those innovations back to the open source community.

B.3 Mobile Linux related projects

Besides Mobile Linux initiatives but quite close to the Mobile Linux movement, several open source projects and communities are facilitating the advent of Linux to mobile devices.

Linux-to-go [22] is a community hosting projects dealing with mobile and embedded Linux computing. To that aim it provides web hosting, file server, GForge project management and bug tracking. Among the projects supported by this community, the Ångström distribution and the OpenEmbedded build system [31] are the most interesting projects for XtreemOS-MD, since the first one will be the Linux base distribution for PDAs and the second one will be used by XtreemOS-MD developers. On the other hand, Linux-to-go is fully open to host new interesting projects, like XtreemOS-MD.

Handhelds.org [16] is also an open source software community devoted to create open software solutions on handheld computing platforms based on Unix-like systems (of course Linux, but not only). They provide information through web pages, communication and collaboration among users by means of mailing lists, and source code archiving and distribution via software repositories. Amongst the projects hosted by this community the most relevant for XtreemOS-MD are the Familiar distribution (Linux distribution for HP iPAQ PDAs over which Ångström was developed) and OpenMoko (a Linux distribution for cell phones - see next paragraph). However, none of them will be used by XtreemOS-MD.

OpenMoko [32] is an open source project devoted to create a free mobile phone operating system that shouldn't be tied to any particular phone (mobile phones supported so far are Neo1973, Treo650, Palm TX, and some models of Motorola and HTC). Up to date, OpenMoko is of alpha-quality and it is expected to reach some stability not before than the end of year 2007. OpenMoko GForge has some interesting projects (ad-hoc communication, BlueTooth remote, GPS location sharing), but out of scope of the current basic version of XtreemOS-MD for PDAs.

Hiker [17] is an open source Linux-based application framework aiming at smartphones and mobile devices. The Hiker application framework components are used by the proprietary Access Linux Platform (for which they were originally developed). This project mainly focuses on user space developing components like application server (to handle application lifecycle), exchange manager (to enable transparent access to data by the applications on the device), global settings (protected registry of system information) and other security related components (including the security module in the kernel space).

Despite the fact that XtreemOS-MD basic version aims at PDAs, the distinction between PIM devices, smartphones, ultra mobile PCs and mobile Internet devices is blurring, and therefore, initiatives targeting Internet tablets or Mobile Internet Devices (MID) should be watched:

- Mobile and Internet Linux (Moblin) project [27] focuses on the development of Linux for Intel-based devices. To that aim it will provide source code and binary images repositories. Several projects have been defined targeting different components of the software platform, from the kernel to the peripherals (like camera) and communications applications (chat built on Telepathy).
- Other community projects like Ubuntu MAE [41], RedFlag Midinux [25] or Pepper Linux [37] are close related to Moblin.org since they are also targeting Intel's MID.
- ARM Linux Mobile platform [4] aims at providing a better browsing experience on ARM devices by means of Mozilla Web Browser porting, open source modules supporting different languages and input methods, a special battery manager, wireless connectivity and using GTK+.
- Maemo [24] is the open source development project for Internet tablets driven by Nokia. It is a GPE-like project but also including a SDK to support creation and porting of new applications for Internet tablets.

References

[1] Android web site.
`http://code.google.com/android/`.

[2] The Angstrom Distribution.
`http://www.angstrom-distribution.org`.

[3] The Angstrom Distribution testing image repository.
`http://www.angstrom-distribution.org/unstable/images/`.

[4] ARM Linux Mobile Platform web site.
`http://www.arm.com/markets/cmc/linux.html`.

[5] Consumer Electronics Linux Forum (CELF).
`http://www.celinuxforum.org/`.

[6] Debian – ARM Port.
`http://www.debian.org/ports/arm/`.

[7] The Embedded Debian Project.
`http://www.emdebian.org`.

[8] Enlightenment web site.
`http://www.enlightenment.org`.

[9] The Familiar Project.
`http://familiar.handhelds.org`.

[10] Fast Handovers for Mobile IPv6 Implementation.
`http://sam.ccaba.upc.es/FIMPL.html`.

[11] Free Desktop.
`http://www.freedesktop.org/wiki/Software/dbus`.

[12] G. N. Brown. Linux - a platform for innovation in converged mobile handsets. *BT Technology Journal*, 25(2), April 2007.

[13] GNOME Mobile And Embedded initiative web site.
`http://www.gnome.org/mobile/`.

[14] GNU LibC web site.
http://www.gnu.org/software/libc/.

[15] The GPE Palmtop Environment.
http://gpe.linuxtogo.org/.

[16] Handhelds.org - Open Source Operating Systems for Handheld Devices.
http://handhelds.org.

[17] Hiker Project Portal Site.
http://www.hikerproject.org/.

[18] Host Identity Protocol for Linux.
http://infrahip.hiit.fi/index.php?index=hipl.

[19] Linux Foundation web site.
http://www.linux-foundation.org.

[20] Linux Foundation Mobile Linux Work Group web page.
http://www.linux-foundation.org/en/Mobile_Linux.

[21] LiMo Foundation web site.
http://www.limofoundation.org.

[22] Linux-to-go.org web site.
http://www.linuxtogo.org/.

[23] Linux Phone Standards Forum (LiPS) web site.
http://lipsforum.org.

[24] Official Maemo web site.
http://maemo.org/.

[25] Red Flag Midinux web site.
http://www.redflag-linux.com/chanpin/midinux/index.htm.

[26] Mobile IPv6 for Linux Homepage.
http://www.mipl.mediapoli.com/.

[27] Mobile Linux Internet Project web site.
http://www.moblin.org/.

[28] Monash Implementation for Hierarchical Mobile IPv6.
http://www.ctie.monash.edu.au/ipv6/hmipv6.htm.

[29] NUTSS Network Architecture.
http://nutss.gforge.cis.cornell.edu/.

[30] Open Handset Alliance web site.
http://www.openhandsetalliance.com.

[31] OpenEmbedded web site.
http://www.openembedded.org/.

[32] OpenMoko.org Portal Site.
http://www.openmoko.org/.

[33] OpenSIMpad web site.
http://opensimpad.org.

[34] OpenZaurus web site.
http://openzaurus.linuxtogo.org/wordpress/.

[35] Opie - Open Palmtop Integrated Environment.
http://opie.handhelds.org/.

[36] OSDL MLI web page (transitional, in the process of being transferred to the Linux Foundation website).
http://old.linux-foundation.org/lab_activities/mobile_linux.

[37] Pepper Linux distribution web page.
http://www.pepper.com/software/.

[38] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261 (Standards track), June 2002.

[39] H. Soliman, C. Castelluccia, K. El Malki, and L. Bellier. Hierarchical Mobile IPv6 Mobility Management (HMIPv6). RFC 4140 (Experimental), August 2005.

[40] TechWeb Java Virtual Machine definition.
http://www.techweb.com/encyclopedia/defineterm.jhtml?term=Javavirtualmachine.

[41] Ubuntu Mobile and Embedded Initiative web page.
https://wiki.ubuntu.com/MobileAndEmbedded.

[42] XtreemOS Consortium. Annex 1 – Description of Work. Integrated Project, April 2006.

[43] XtreemOS Consortium. Design of an Infrastructure for Highly Available and Scalable Grid Services D3.2.1. Integrated Project, December 2006.

[44] XtreemOS Consortium. Application References, Requirements, Use Cases and Experiments D4.2.3. Integrated Project, July 2007.

[45] XtreemOS Consortium. Design and Implementation of Node-level VO Support D2.1.2. Integrated Project, December 2007.

[46] XtreemOS Consortium. Requirements and Specification of Basic Services for Mobile Devices D3.6.1. Integrated Project, December 2007.

[47] XtreemOS Consortium. Requirements and Specifications of a Basic Linux Version for Mobile Devices D2.3.2. Integrated Project, June 2007.

[48] XtreemOS Consortium. Virtual Organization Basic Requirements and Specifications for Mobile Devices D2.3.1. Integrated Project, February 2007.