Project no. IST-033576

XtreemOS

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

Requirements and Specifications of a Basic Linux Version for Mobile Devices
D2.3.2

Due date of deliverable: May 31st, 2007
Actual submission date: May 30th, 2007

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

Type: Deliverable
WP number: WP2.3
Task number: T2.3.2

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 1.0 / Last edited by Luis Pablo Prieto / May 30th, 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	5/12/06	Daniel Galindo	TID	Initial template
0.2	16/03/07	TID Team	TID	First draft of state of the art section
0.3	26/03/07	TID Team	TID	First draft of requirements section
0.4	10/04/07	Yvon Jégou	INRIA	Integrated VO support requirements and specifications
0.5	16/04/07	TID Team	TID	Integrated the rest of the specifications. General corrections
0.6	20/04/07	TID Team	TID	Executive Summary, Introduction, Conclusions, Future Work sections included
0.7	26/04/07	Luis Pablo Prieto	TID	Minor corrections and general proofreading
0.8	22/05/07	Yvon Jégou	INRIA	Updated VO support requirements (removed non MD-specific requirements)
0.9	24/05/07	TID Team	TID	Corrections derived from internal reviewers
0.91	28/05/07	Oscar David Sánchez	INRIA	Updated federation management requirements
0.92	29/05/07	Luis Pablo Prieto	TID	Merge of all changes
1.0	30/05/07	Luis Pablo Prieto	TID	Finishing touches

Executive Summary

In this document, the requirements and specifications of Linux-XOS for MDs (the Foundation layer of XtreemOS-MD) are proposed. Linux-XOS for MDs will be a personalised version of Linux operating system providing the key features to natively support virtual organisations and specially designed for mobile devices.

The state of the art regarding operating systems for mobile devices shows a wide variety of choices. The open-source alternative, represented by Linux-based operating systems, is rapidly becoming more and more popular. For PDAs, there are several Linux distributions available, of which Familiar and Ångström seem to be the most suitable to be used as the base distribution for Linux-XOS for MDs. XtreemOS enhancements that will be implemented on top of this base distribution include, among others, native support for virtual organisations and Mobile IPv6.

Taking input from the state of the art, as well as from the different workpackages and deliverables produced so far, a series of requirements to be met by the basic version of Linux-XOS for MDs are presented. These requirements are classified into general, VO support, checkpointing, interfaces, services, application and data management, security and support for mobile devices. Moreover, requirements are classified according to different importance levels: Basic requirements are those that will be implemented for the basic version of XtreemOS-MD. Advanced requirements will be met by the advanced version of XtreemOS-MD. Optional requirements are of less importance and will be offered as optional features.

Requirements that do not explicitly affect mobile devices will not be included in this deliverable, as they will be handled by the respective workpackages. This document focuses on requirements concerning mobile devices, including (but not limited to): it is necessary that XtreemOS-MD is able to run on the most common hardware available on the market; it should be possible to monitor applications running on other XtreemOS nodes from MDs; there is also a need for supporting terminal and user mobility.

The final specification of Linux-XOS for MDs is based on the previously extracted set of requirements together with the results from the state of the art and other deliverables. This specification determines the different components that are required in order to have an XtreemOS-compliant mobile device in terms of its operating system. Just like requirements, specifications are also classified into basic, advanced and optional. According to the specifications, XtreemOS-MD will be available for the ARM architecture and will provide kernel support for the Mobile IPv6 protocol. Specifications also guarantee that other Grid nodes will be aware of the mobile nature of mobile devices nodes. Other important specifications ensure support for the execution of Java applications, Linux Pluggable Authentication Modules, or the use of lighter security methods.

Future steps to be taken include the election of a Linux distribution which will serve as the basis for XtreemOS-MD. The selection of the different components

that will ground the Foundation layer of XtreemOS-MD will also be considered. Finally, a selection of packages and kernel modules will be made for XtreemOS-MD. The specifications obtained in this deliverable will be a key factor in this process, as they will ensure that the decisions taken are not in conflict with XtreemOS-MD expected behaviour.

Contents

Glossary

ARM	Advanced RISC Machine
ALP	Access Linux Platform
AODV	Ad-hoc On-demand Distance Vector protocol
ARM	Advanced RISC Machine
BGP	Border Gateway Protocol
C#	C Sharp
CELF	Consumer Electronics Linux Forum
CN	Correspondent Node
DSR	Dynamic Source Routing
ELC	Embedded Linux Consortium
F-layer	Foundation layer
FMIPv6	Fast Handovers for Mobile IPv6
FOSS	Free and Open Source Software
GMAE	GNOME Mobile And Embedded initiative
GNU	GNU is Not Unix
GPE	GPE Palmtop Environment
GPL	General Public License
HA	Home Agent
HMIPv6	Hierarchical Mobile IPv6
IETF	Internet Engineering Task Force
IDE	Integrated Development Environment

ISV	Independent Software Vendor
LiPS	Linux Phone Standards Forum
IP	Internet Protocol
MAC	Media Access Control
MANET	Mobile Ad-hoc Network
MAP	Mobility Anchor Point
MIPL	Mobile IP for Linux project
MIPv4	Mobile Internet Protocol version 4
MIPv6	Mobile Internet Protocol version 6
MLI	Mobile Linux Initiative
MD	Mobile Device
MN	Mobile Node
MR	Mobile Router
MRC	Metadata and Replica Catalogue
NEMO	Network Mobility
NEPL	NEMO Platform for Linux
NFS	Network File System
NIS	Network Information Service
OLSR	Optimised Link State Routing protocol
OMTP	Open Mobile Terminal Platform
OPI	Open Platform Initiative
OPIE	Open Palmtop Integrated Environment
OS	Operating System
OSD	Object Storage Device
OSDL	Open Source Development Labs
OSPF	Open Shortest Path First
OSS	Open Source Software, also Object Sharing Service

OSX	Operating System Ten
PAN	Personal Area Network
PDA	Personal Digital Assistant
PACE	Palm Application Compatibility Environment
PIM	Personal Information Management
PNO	PACE Native Objects. Pieces of ARM code performing common tasks
RA	Router Advertisement message
RFC	Request For Comments
RIP	Routing Information Protocol
RMS	Replica Management System
RS	Router Solicitation message
SDK	Software Development Kit
STCP	Stream Control Transmission Protocol
TCO	Total Cost of Ownership
TCP	Transport Control Protocol
UDP	User Datagram Protocol
UMTS	Universal Mobile Telephone System
USAGI	Universal Playground for IPv6
WIDE	Widely Integrated Distributed Environment
WLAN	Wireless Local Area Network
WM	Windows Mobile
XtreemOS-MD	XtreemOS for Mobile Devices

Chapter 1

Introduction

The main objective of this document is to extract a series of specifications to be met by the Foundation layer of XtreemOS flavour for mobile devices (XtreemOS-MD).[1] XtreemOS-F can be considered as the modified Linux operating system embedding VO support mechanisms and providing an appropriate interface to implement XtreemOS-G services [52], as depicted in Figure 1.1.

Once extracted the hardware and network specifications of mobile devices to run XtreemOS-MD within D2.3.1 [60], software and operating system specifications are now addressed to complete the set of specifications that will be necessary to tackle the development activities.

Because of the reasons mentioned in the introductory chapter of D2.3.1 [60], we have mainly focused on PDAs (mobile phones will be considered on the advanced version of XtreemOS-MD). Laptops have not even been considered because they are basically identical to standard PCs and thus, they will run the standard flavour of XtreemOS.

1.1 Methodology

Our approach has been to first collect a set of requirements that XtreemOS-MD should meet. This set of requiremens has been extracted from applications (D4.2.1 [57]), hardware and network requirements (D2.3.1 [60]) and the state of the art in operating systems for mobile devices, Linux for mobile devices and mobility in Linux. These requirements have been the actual source to make the current list of specifications for XtreemOS-MD. So to speak, requirements describe **what** must be done and specifications describe **how** it must be done, i.e. how requirements will be met.

[1] XtreemOS Foundation layer (XtreemOS-F) and Linux XOS both refer to the same software layer in XtreemOS, and are used interchangeably throughout this document.

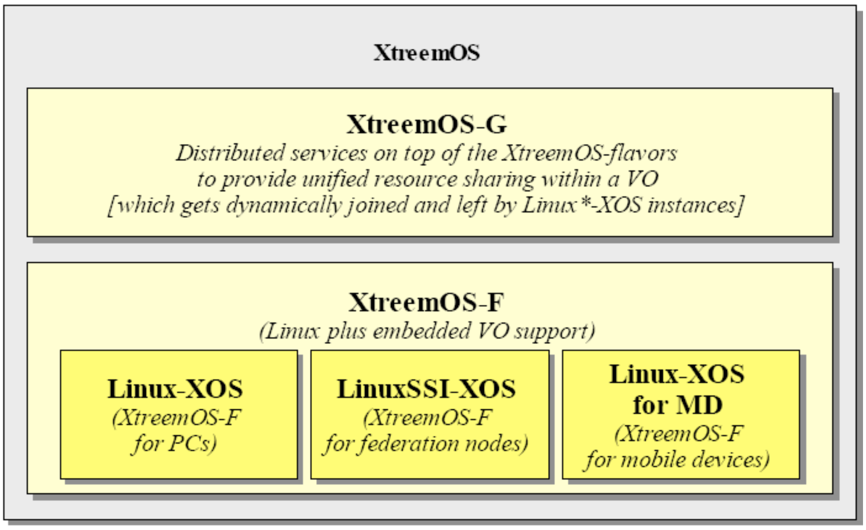

Figure 1.1: Summary of XtreemOS flavours

1.2 Document structure

This document is structured as follows:

In chapter 2 we describe the state of the art in the relevant topics for the development of XtreemOS-MD, such as the current operating systems for mobile devices, Linux and mobile devices and the different ways mobility can be handled within a Linux operating system.

The list of software requirements to be met by XtreemOS-MD is detailed in chapter 3.

Chapter 4 contains the list of specifications of XtreemOS-MD derived from all the aforementioned requirements.

In chapter 5 we present some open issues to be specified in further stages of the project, and chapter 6 gathers the main conclusions derived from this document.

Finally, in chapter 7 we outline the next steps to be followed after this document in order to design the XtreemOS-MD distribution itself. These steps will be reflected in D2.3.3.

Chapter 2

State of the Art

One of the most rapidly evolving areas in software development and IT in general, is that of mobile devices. Every week, new devices, software manufacturers and initiatives flood the market with their output, in search of the next “killer device” or “killer application”. In this context, having up-to-date knowledge of the mobile arena is of utmost importance when undertaking a project aimed at mobile devices.

In this section, three areas of this mobile market are surveyed, not only to provide an adequate context for making informed design decisions for XtreemOS-MD, but also to extract requirements to make XtreemOS-MD a usable and competitive operating system:

- First, an analysis of current **mobile operating systems** is carried out, not in order to ascertain which operating system will be used in XtreemOS-MD, but to see what the competition is offering, and later extract requirements that make XtreemOS-MD a competitive operating system for PDAs.
- Then, the focus will move to **Linux in mobile devices**, with a survey of the most important Linux distributions for PDAs (which will be used to choose the most adequate distribution for the needs of XtreemOS). Other important aspects to the development of a mobile operating system (namely, development tools, communities and standardisation initiatives) will also be reviewed, as they can also impact the design, implementation and exploitation of XtreemOS-MD.
- Finally, the status of **mobility enhancements in Linux** is examined, to establish the starting point for further enhancements and to draw requirements regarding mobility, in order to make XtreemOS-MD a truly mobile and ubiquitous operating system.

2.1 Operating Systems for Mobile Devices

2.1.1 Overview

As previously shown in deliverable D2.3.1 [60], in regards to operating systems in mobile devices, a wide variation across studies can be noticed, depending on which devices are counted as smartphones or as PDAs. All in all, major options seem to be [22, 15, 24]:

- **RIM** for the Blackberries: owned by Research In Motion.
- **Palm OS**: owned by PalmSource.
- **Windows Mobile** (Pocket PC): based on Windows CE kernel, and owned by Microsoft.
- **Symbian OS** (specially in smartphones): owned by Ericsson, Motorola, Panasonic, Nokia, Samsung, Siemens and Sony Ericsson.
- Various operating systems based on the **Linux** kernel. These include private companies (MontaVista, Trolltech) as well as communities (Familiar, GPE, OPIE).

In the following sections, we are going to review the key features of these operating systems from the XtreemOS point of view.

2.1.2 RIM

RIM (Research In Motion) is a Canadian company experienced on wireless solutions for the mobile communications market. RIM provides integrated hardware, software and services in order to access different types of information, most of them time-sensitive: email, phone, SMS, MMS, and other web based information and services.

Emphasis must be laid on the fact that, initially, Blackberry was a whole solution involving proprietary software for specific hardware. This is no longer true. For example, Blackberry licences a BlackBerry e-mail client which connects BlackBerry servers to other mobile phone manufactures like Nokia, Motorola and HTC.

Focusing on software running on MDs, RIM provides a proprietary operating system (OS) installed on BlackBerry, with the following characteristics:

- It is a multi-tasking OS
- It makes heavy use of the device's specialised input devices, (QWERTY keyboard, thumbwheel or scroll wheel, etc).
- Concerning Java Runtime Environment (JRE), it provides support for MIDP 1.0 and a subset of MIDP 2.0 in its last version.

Third-party developers can write software using MIDP (Java) and proprietary APIs (C++). Nevertheless these applications must be digitally signed and associated to a developer account, which actually means that there is little place for third party modifications or free software in general on the RIM platform.

2.1.3 Palm OS

Palm OS is an operating system licenced by PalmSource, Inc., the Palm spin-off devoted to OS development (while Palm, Inc concentrates on hardware issues). Palm OS runs on different manufacturer's devices, including Palm Inc., Samsung, Lenovo and Sony.

Palm OS was originally released in 1996 and since version 5.0, Palm OS supports ARM architecture.

Last stable version of Palm OS is named Palm OS Garnet (corresponds to 5.4.9). There also exists a 6.0 version named Cobalt but no device has been released with this version so far.

Apart from built-in applications (like Calendar, Address Book, and other PIM applications), there are more than 20,000 third-party applications available for the Palm OS platform, which have various licensing types, including open-source, freeware, shareware, and traditional commercial applications. PalmSource also develops several programs for the Palm OS.

Applications are primarily coded in C/C++. Two compilers exist: a commercial one (CodeWarrior Development Studio for Palm OS), and an open source tool based on gcc. A version of the latter is included in a free Palm OS Developer Suite. Software developers can take advantage of the ARM processors with PNO (PACE Native Objects), small units of ARM code, also sometimes referred to as 'ARMlets'.

There are also higher level development tools available for Palm such as CASL, AppForge Crossfire (which uses Visual Basic, Visual Basic.NET, or C#) and Handheld Basic or HB++ (which uses Visual Basic).

In September 2005, PalmSource was acquired by ACCESS and in February 2006, PalmSource announced ACCESS Linux Platform, "the latest evolution of Palm OS for Linux". It is planned to be released on Q2 2007. At that time, Palm also announced a new device running Windows Mobile OS, signalling Palm's diversification into two OSes for their handhelds. These two measures can probably mean the fade out of Palm OS as a separate operating system.

2.1.4 Windows Mobile

Windows Mobile (WM) is an operating system for mobile devices from Microsoft Corporation, based on the Microsoft Win32 API. It is designed to run on PDAs, smartphones and other portable devices like portable media centers.

The latest version of Windows Mobile has been released on February 12, 2007 (see [42]). Its core (kernel, mainly) is based on the foundations of Windows CE like all previous releases. Windows CE is multithreading and multitasking OS.It is also a modular operating system and it is possible to purchase the Platform Builder Kit, which contains all those components and tools to develop a custom platform, which means that WM can be adapted to different devices by developers.

Three different versions focus on different device categories: Windows Mobile 6 Standard for smartphones (usability without touchscreens), Windows Mobile 6 Professional for PDAs with phone functionality (Pocket PC Phone Edition), and Windows Mobile 6 Classic for plain PDAs.

WM includes a set of basic applications like mobile versions of Microsoft Office applications and Activesync (a tool for synchronising the data in the mobile device with a desktop PC).

The development of applications for WM 6 is based on Windows Mobile 6 Software Development Kit (SDK). This SDK provides the tools and libraries needed to start developing Windows Mobile 6 applications using Microsoft Visual Studio 2005. Applications can make use of .NET Compact Framework 2.0 and SQL Server 2005 Compact Edition, as these two APIs must be present in the ROM of all WM 6 devices.

2.1.5 Symbian OS

Symbian OS is an operating system licenced by Symbian Ltd. It is exclusively designed for mobile devices (advanced 2.5G and 3G mobile phones) with ARM architecture.

Symbian is owned by several mobile phone manufacturers with the following shares: Ericsson (15.6%), Nokia (47.9%), Panasonic (10.5%), Samsung (4.5%), Siemens AG (8.4%), and Sony Ericsson (13.1%).

Symbian OS supports a wide range of devices from these providers, with several user interfaces, developed on common Symbian OS APIs. Symbian OS main design characteristics are:

- Pre-emptive multitasking, multithreading, memory protection
- Multi-language development support: C++, J2ME MIDP 2.0
- Reference telephony abstraction layer for 2G, 2.5G and 3G provided
- Flexible device creation. Easy creation of virtual devices for testing purposes. There exist many SDKs for developing software for Symbian OS, from different companies, like Nokia and SonyEricsson. Software licences of available software range from freeware to commercial, and there are amounts of applications from communities and software companies.

2.1.6 MD Operating Systems at a Glance

We summarise most important characteristics of the analysed OSes in the following tables.

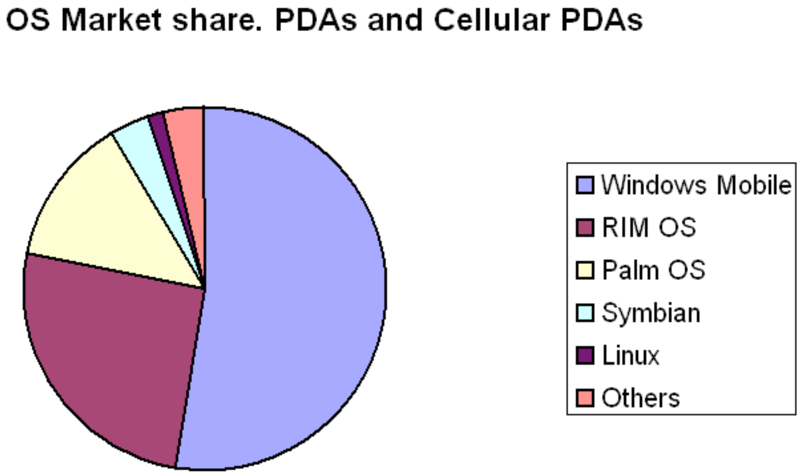

Figure 2.1: PDAs Market Share

	RIM	**PalmOS**
Multithread/Multitask	Yes/Yes	Yes/Yes
Devices supported	RIM	Many (Palm, Samsung, Sony...)
Developement tools	Java MIDP, C/C++	C/C++ VB, VB .NET, C#, HB++

	Windows Mobile	**Symbian OS**
Multithread/Multitask	Yes/Yes	Yes/Yes
Devices supported	PDAs, smartphones and other portable devices from many manufacturers	Many manufacturers, mainly Symbian shareholders
Developement tools	Visual C++, C#, VB .NET	C++, J2ME MIDP 2.0

2.1.7 Market Trends and Shares

According to [23], where evolution of PDA market in 2006 is analysed, Microsoft's Windows Mobile surpassed the 50 percent threshold for the first time in first quarter of 2006, with a share of 52.6 percent. RIM OS was second at 25.5 percent market share, followed by Palm OS at 13.4 percent, Symbian at 3.6 percent and Linux at 1.2 percent. This market research does not include smartphones, but include cellular PDAs.

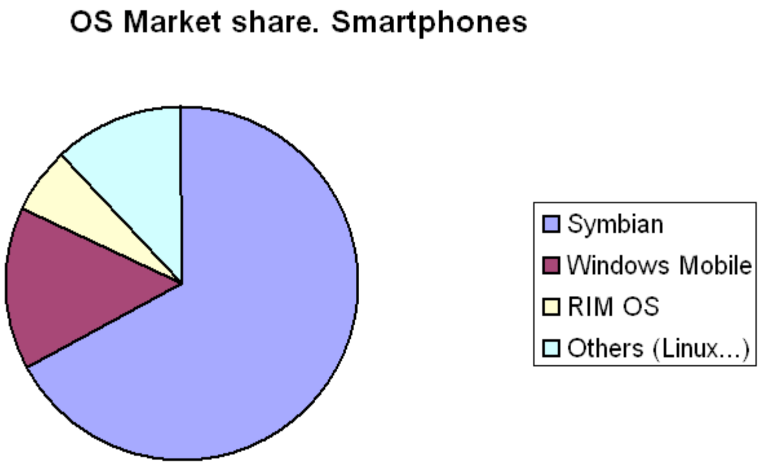

Figure 2.2: Smatphones Market Share

In the case of smartphones, which is considered an independent market, and therefore studied separately, the leading OS is Symbian with 67 percent, followed by Windows Mobile with 15 percent and 6 percent for RIM OS (figures from [5]).

Research firm Gartner reports that worldwide mobile phone sales grew more than 20 percent in 2006, with almost one billion phones sold that year [29]. Other studies [24, 50] predict a steady growth in Linux market share for high end mobile phones, which can only be shadowed by Microsoft's Windows Mobile. Thus, the future for Linux looks quite rosy in the mobile phones area, not only in smartphones, but also for low-end phones: as they become packed with more and more features, Linux will become a sensible option for them.

2.1.8 Conclusions

The market of OSes for PDAs and smartphones is boiling because of its inmature nature. The whole market is growing every year, in customers, in services and technical solutions offered to them. The blurred division between connected PDAs and smartphones has also increased this instability, with all the OSes trying to cope with the whole range of devices. This is the case of Windows Mobile and Symbian, for example. This fact is increasing competence because two different markets (smartphones and PDAs) are becoming one.

Two of the most valued requirements of MD OSes are customisability and total cost of ownership (TCO), and OS vendors will have to offer the most balanced solution to the market.

Phone vendors and network operators like the customisability of Symbian OS relative to Windows. This customisability, though, makes integrating a Symbian OS phone more difficult. That is the main drawback of Linux distributions for

MDs; it is too hard to make a phone from Linux at the moment.

The ability to balance TCO and customisability will probably become a key issue for OS developers' success.

If customisability is a strength of Linux distributions, TCO can be a weakness. That's why in the case of Linux distributions, special attention must be paid to TCO, term that involves not only the cost of purchase, but also installation costs, costs of training (ease of use) and maintenance costs, among others.

During design of XtreemOS-MD, those characteristics of mobile OSes must be taken into account, TCO and customisability, because they will be the key issues in its subsequent expansion and success.

2.2 Linux and Mobile Devices

In this section, the state of the art in mobile Linux is reviewed, including the reasons for adopting Linux as an operating system for mobile devices, the most important mobile Linux distributions, as well as the most important initiatives related to mobile Linux adoption and standardisation.

2.2.1 Rationale

Linux has become a serious alternative to proprietary operating systems in the mobile world. Linux on MDs has experienced a considerable growth over the past two years, as can be deduced from the following statistics [4]:

- Close to a quarter of all smartphones sold during last year were running Linux.
- Over the year 2006 the percentage of wireless developers targeting the Linux platform with their applications has risen from 24% to 37%, i.e. from 2Q2005 to 2Q2006 there has been a 54% increase in Linux development in the mobile space.

This growth of Linux on MDs is expected to continue, driven by a number of factors [4, 7, 35]:

- **Stability**: Being open source, bugs can easily be found and corrected. Moreover, Linux (specially the kernel) has been tested by thousands of users and developers all over the world for many years.
- **Security**: Linux security flaws can be easily detected in its source code, and it's code has been reviewed and enhanced by many security experts.
- **Flexibility**: Having the code available allows developers to customise it to the specific needs of each project. In fact, the Linux kernel has already been ported to more than 15 different **hardware architectures**, both in 32 and 64bit variants.

- Stronger **networking** support than many other operating systems, something that is becoming more and more important, now that almost every piece of electronic hardware can be connected to a network.
- **Interoperability**/Compliance to standards: the Linux community is very standard-conscious, and whenever a standard arises in any IT area, there is a high probability that some kind of Linux implementation will appear in a short time.
- Good **performance** for both application layer and system boot up.
- **Independence** and open structure, a very valuable feature for hardware and software companies that want to avoid being tied to certain potential competitor companies (as it happens in other OSes, like Windows Mobile and Microsoft or Symbian and Nokia).
- Community **support**: Linux communities offer free support for the software they develop, either through mailing lists, fora, chats, etc. But this community support can also be seen as a disadvantage, as will be seen later. In any case, there are also software companies that offer more conventional support for Linux software.

However, Linux is not a perfect platform, and also has its own **weaknesses**. Some technical drawbacks or inhibitors that can be pointed out are [4, 7, 35]:

- Linux often lags behind other popular OSes when it comes to **hardware support**, because of the wealth of different devices and architectures that are appearing in the market, and also due to the reluctance of many hardware manufacturers to develop Linux drivers, or even to disclose precise specifications for their devices.
- **Fragmentation** is another of the common criticisms that Linux faces, as the Linux community is composed of many smaller communities, initiatives and companies, which overlap (and sometimes diverge) in their goals and objectives. This often translates into:
  - Lack of widely-used standards inside Linux itself
  - Interoperability problems above the kernel, between different distributions, graphic toolkits, and middlewares. Fortunately, this problem has been acknowledged, and several initiatives are trying to bridge the gaps in this area (see section 2.2.2).
- Community **support**, although cheap, has no contractual obligations. Thus, its effectiveness depends greatly on the strength of the community itself, and open source projects dying from loss of interest by the community are not uncommon.

Apart from the aforementioned technical issues concerning Linux adoption on MDs, some **economic** factors are also relevant, and must be taken into account. The lack of license fees is the main motivation for both device manufacturers and software developers to adopt Linux. This can lead to a lower total cost of ownership and therefore an increase of profit margin, which is very attractive for hesitant companies. However, the risk of discovering something that has been already patented is implicit in OSS development; this would mean an amount of resources unfruitfully spent. To diminish the effect of this, some organisations like the Open Invention Network (OIN), formed by IBM, Novell and others, offer royalty-free patents, and Nokia has announced that they will not pursue their mobile Linux patents [4].

2.2.2 Current State of Mobile Linux

In this section we will describe the general outlook of Linux-powered mobile devices, focusing in the area of PDAs, which are the target for the basic version of XtreemOS-MD. As it was already pointed in deliverable D2.3.1 [60], some mentioning of mobile phones will be unavoidable, as these two kinds of devices are increasingly taking over each other's functionalities.

We will also review the main initiatives that foster Linux adoption in mobile devices, as well as the companies that have already adopted it or those who are working on development tools for it.

2.2.2.1 Mobile Linux Arena

Linux arrived quite late to the PDA operating system arena, thus missing that market's biggest expansion. Linux has remained a minor player ever since, overshadowed by the likes of Palm and, more recently, Microsoft. In fact, Linux PDAs are very hard to come by in Europe and America, and only Asian markets have seen a considerable variety of Linux-powered PDAs.

This outlook is changing in the recent years, with the advent of "connected PDAs". The need for more advanced features, together with the mobile phone cross-breeding, is forcing some manufacturers to reconsider their operating system decisions. A good example of this is Nokia which, even being the main Symbian OS promoter, has released two models of PDA-like devices (the 770 and 800 Internet Tablets) featuring a Linux operating system.

As the mobile market grows, handset manufacturers are turning to Linux in order to provide more capabilities, flexibility, reduce time-to-market and, therefore, reduce costs. Several groups and initiatives have raised during the last few years in order to foster the adoption of Linux on MDs. This adoption involves companies from all levels of the value chain: device manufacturers, kernel developers and application developers, as well as those software companies providing the so necessary development tools for Linux on MDs.

2.2.2.2 Mobile Linux Initiatives

As it has been already mentioned, several consortia have been formed to identify and bridge gaps in the Linux and FOSS ecosystem, in order to accelerate its adoption on MDs. It should be noted that there has been much activity in this front in the last months, with initiatives changing names and merging with others:

OSDL MLI (Mobile Linux Initiative) Its mission is to accelerate adoption of Linux on next-generation mobile handsets and other converged voice/data portable devices, trying to identify and fill gaps in the Linux platform and in the ecosystem above and around it [38]. To do so, MLI will perform the following actions:

- Identify and address technical and nontechnical industry requirements.
- Create and foster implementations in open source.
- Advocate/explain industry needs to the kernel/open source community.
- Promote mobile Linux (including education of Carriers about benefits of open source).
- Clarify legal and regulatory issues surrounding mobile phones as they relate to Linux and open source.
- Enable and foster preplatform developer ecosystem.

Among others, this initiative absorbed the Embedded Linux Consortium (ELC) in 2005, and was in turn incorporated recently to the Linux Foundation (see below).

Linux Foundation The Linux Foundation is a nonprofit consortium dedicated to foster the growth of Linux [25]. The Linux Foundation promotes, protects and standardises Linux by providing unified resources and services needed for open source to successfully compete with closed platforms. It was born with the fusion of the Open Source Development Labs and the Free Standards Group, and its scope is not limited to mobile adoption (by the inclusion of OSDL's MLI), but also including legal protection, promotion, collaboration and standardisation (they are behind the Linux Standards Base and Carrier Grade Linux, for example).

This consortium comprises the whole spectrum of the mobile ecosystem, including CPU and chipset manufacturers (Intel, AMD...), Linux distribution and platform suppliers (a la Mobile, Access, MontaVista...), ISVs for middleware and applications (Trolltech), handset manufacturers (Fujitsu, NEC, Nokia...), integrators and mobile and wireless carriers and operators (NTT Corporation).

LiPS (Linux Phone Standards Forum) This consortium (created in 2007) is founded by a group of telephony operators, device manufacturers, silicon and software vendors who have a strategic focus on Linux telephony [28]. It was created to define a standardised software platform which allows applications and services to be deployed across all types of Linux telephony terminals, fostering interoperability and promoting adoption of Linux.

Current LiPS members include operator and services providers (France Telecom, Telecom Italia), equipment and chip manufacturers (Freescale Semiconductors, Texas Instruments...), software vendors (a la Mobile, ARM Limited, McAfee Mobile Solutions, Palmsource, Montavista, Movial Oy, Trolltech...) and independent supporter members.

LiMo Foundation (Linux Mobile Foundation) This initiative is led by four large mobile phone vendors, together with two major wireless operators: Motorola, NEC, NTT DoCoMo, Panasonic, Samsung and Vodafone. These companies aim at creating an independent foundation to develop and market an open Linux implementation, including an API specification, architecture, and source code-based reference implementations of components and tools [26]. The additional goals of the foundation try to mitigate some of the drawbacks pointed out in section 2.2.1:

- Implement a fair, balanced, transparent contribution and participation process for current and future members.
- Establish safeguards to minimise fragmentation.
- Collaborate on a mobile Linux developer ecosystem.
- Coordinate with existing industry organisations.
- Seek participation from all interested companies across the value chain, including device manufacturers, operators, chipset manufacturers, independent software vendors, integrators and third-party developers.

OMTP (Open Mobile Terminal Platform) OMTP is an operator-sponsored forum that serves the needs of each and every link in the mobile phone value chain by gathering and driving mobile terminal requirements. It is technology neutral, with its recommendations intended for deployment across the range of technology platforms, operating systems (OS) and middleware layers. Carriers, content providers, middleware vendors, handset manufacturers and users all stand to gain from the forum's recommendations [34]. It is currently formed by 9 operators, including Cingular, Vodafone and Telefonica. It is OS-agnostic, but it is worth including here as a force driving the mobile device characteristics.

Handhelds.org is a community, not an initiative with industrial backing. Their main goal is to encourage and facilitate the creation of open source software for use on handheld and wearable computers [14]. Although originally supported by Compaq, it is now an independent community dedicated to porting Linux to a wide variety of devices, not limited to HP/Compaq PDAs. It is one of the most active communities, which has spawned several tools and distributions for PDAs, including Familiar and Intimate (see below).

Linux-to-go.org is another community, an "effort to create a general platform to work collaboratively on common interests", "mostly concerned about mobile Linux applications and systems, i.e. everything in the area of mobile and embedded Linux computing." [27]. This community hosts several mobile Linux projects, most notably the Ångström and OpenZaurus distributions, the OpenEmbedded framework and the GPE and G(PE)^2 graphic environments (see below).

2.2.2.3 Current Linux Distributions for PDAs

Currently, there exist a number of Linux distributions aimed at PDAs. As it often happens in open source projects, some of these are more active than others, but in any case, the main players in the Linux PDA area are:

Familiar is one of the distributions developed by the Handhelds.org community. The Familiar Project is composed by a group of developers all contributing to creating the next generation of PDA OS [11]. Although a great number of handheld devices are supported, their efforts are currently devoted to produce a stable and full featured Linux distribution for the HP iPAQ series and other PDAs, as well as applications to run on top of the distribution.

Currently Familiar's Linux distribution supports some of the following key features:

- Choice of user environments, both with full PIM suite and other applications: GPE and Opie.
- Full package management based on ipkg.
- Many system programs are implemented using busybox, to save space.
- Dropbear SSH server included by default.
- Built entirely using the OpenEmbedded build system.

One of the best features of Familiar is that it has a very active community, which is constantly adding support for new devices and features.

Intimate is also part of Handhelds.org. The Intimate Project is a fully blown Debian based Linux distribution for the Compaq iPAQ, combining the work being done by the Familiar Project and the Debian package management [17].

There is an important drawback concerning the Intimate distribution: the minimum requirements are currently around 140MB of storage for the base image, which probably won't fit a PDA without external storage (i.e. Flash cards). Besides, the developments on this distribution seem to have stopped in 2005, and are based on a quite outdated version of Familiar.

OpenZaurus This project was created as an alternative image for the Sharp Zaurus personal mobile tool, with the purpose of creating an image (kernel + root filesystem) which was a bit closer to what the developer community specifically desired [37]. But shortly after this, it was revamped completely, becoming a Debian-based (although it uses the ipkg format) embedded Linux distribution built from source, from the ground up. The efforts of OpenZaurus and Familiar along with other embedded Linux projects were integrated into the OpenEmbedded Project, which now provides the common framework for these projects [36].

Ångström is another distribution for embedded devices, which was started by people from the OpenEmbedded, OpenZaurus and OpenSimpad projects 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 [3]. It started in 2006, and it seems that many OpenZaurus efforts are being redirected to this distribution.

EmDebian (Embedded Debian) This project intends to make Debian GNU/Linux the mainstream choice for embedded projects, trying to strip Debian down to be a much smaller system whilst keeping all its advantages [8]. This is different from Debian ARM, which is the project that only tries to port Debian GNU/Linux to ARM architecture.

ALP (Access Linux Platform) is the next version of Palm OS developed by Access Co. This Linux platform, based on the Linux kernel 2.6.14, provides a commercial OS with application compatibility layers for traditional Palm OS, Java, GTK+ and Linux shell applications [2]. This OS was first announced in 2006 and, although it already counts with a development kit, it is not yet supported by Palm Inc.'s terminals.

2.2.2.4 Development Tools

Another important point to have in mind when developing software for mobile devices, is that there are some additional issues, which come from the fact that these devices use different hardware architectures than a PC (for example, ARM). These can include:

Cross-compilation occurs when a compiler running on one system produces executables for another system – this is an important concept when the target system doesn't have a native set of compilation tools, or when the host system is faster or has greater resources. This is accomplished using a cross-compiler toolchain and cross-compiled libraries.

Automated build tools Open Source Software (OSS) projects widely employ automated build tools to ease the development and testing processes. GNU

Autoconf and Automake are two very common examples. They may automatically detect and recognise certain features of the build environment, and this would not be desirable when the build environment (host platform) is different to the target platform.

Non-standard input/output Mobile devices have much smaller screens than PCs, and also have very different input devices, like stylus, touchscreen, etc. This poses a problem when designing GUIs for these devices, because standard graphical toolkits widely used in Linux cannot be used in the same way.

There are a number of tools and frameworks that try to solve those problems, just to name a few:

- **Scratchbox** is a cross-compilation toolkit designed to make embedded Linux application development easier [46], by sandboxing the compilation, so that compilers and build tools "see" the target machine instead of the host machine. It also provides a set of tools to integrate and cross-compile an entire Linux distribution, and allows testing the compiled software either in a real target machine or in a virtual one using the QEMU emulator.
- **OpenEmbedded** is a development tool for building entire distributions from source, specially embedded distributions [36]. It supports several machine/distribution configurations, and is specially useful as a integration and release tool, since it is able to generate flashable images to be installed directly in the target device.
- Regarding **GUI toolkits**, the main players in mobile Linux are:
  - Qtopia (developed by Trolltech, the creators of the Qt graphical libraries) is one of the most used commercial[1] software stacks for mobile devices, with several versions, either for PDAs and mobile phones.
  - The GPE Palmtop Environment (GPE) is a collection of integrated software components optimised for (but not limited to) handheld and other input constrained and resource limited devices. It also has recently launched a sibling project, GPE Phone Edition or G(PE)^2, targeted to mobile phones.
  - In April 2007 it was announced the GNOME Mobile And Embedded initiative (GMAE), a proposal by the GNOME Foundation to "advance the use, development and commercialization of GNOME components as a mobile and embedded user experience platform". It seems quite similar to GPE, and its evolution should be followed closely.

[1] Although there is an Open Source version, for OSS projects.

2.2.3 Conclusions

The study of all these distributions, initiatives and tools will be very useful when choosing a base distribution for XtreemOS-MD, and also for monitoring the most relevant mobile Linux initiatives and defining an adequate development environment.

In fact, we can already point out the most probable choices in many of these areas, although a more thorough analysis and explanation will be given in future reports (D2.3.3):

Base distribution As of this writing, the most promising options for a base distribution for PDAs are Familiar and Ångström. **Familiar** has the advantage of having a very active community behind, and also its support for a wide range of PDAs. Unfortunately, it has the problem that most of its packages are quite outdated, and this can be a problem since most of the work done in other areas of XtreemOS seems to be using very up-to-date versions of their required libraries. **Ångström**, on the other hand, is a new distribution with modern versions of most of its packages, but it doesn't support as many devices as Familiar. In the end, probably the best option would be to use Ångström, but reuse as much of the work and knowledge from Familiar as it is possible.

Development tools From the study, several tools arise as useful additions to an eventual development environment for XtreemOS-MD. As a bare minimum, every developer should have a **cross-compilation toolchain** (as compiling natively in the device would be too slow). **Scratchbox** is another tool that can greatly ease the development of software for mobile devices. **OpenEmbedded** can also ve very valuable, specially in the integration and packaging stages of the project, to generate the binary images of the software. In the case of developers not having a physical mobile device at their disposal, **emulation** software such as QEMU or SkyEye should be added to this developer toolkit.

Linux initiatives This is probably the most unclear area, as most of the initiatives are brand new, and many of them were created during the last year. The best strategy here seems to wait and see how they evolve, and see which of them are left behind and which are the real drivers of the mobile Linux evolution.

2.3 Linux and Mobility

In this section, current state of the art in mobility in Linux systems will be analysed, with special emphasis in Linux support for the Mobile IP protocol and its different optimisations.

In deliverable D2.3.1 [60] we already defined the different kinds of mobility that can be found on a system:

Terminal Mobility The user can change his location (i.e. access point), be it either in offline mode (nomadicity) or online mode (mobility). In this last case, the session should be maintained.

Linux terminal mobility support is currently based on the Mobile IP protocol for both IPv4 [39] and IPv6 [19]. Terminal mobility requires support from the kernel and from several user-space daemons. In the following pages, Linux support for these two protocols will be thoroughly discussed.

User Mobility The user can access Grid services (maybe personalised to fit his needs) from any location and device.

User mobility in Linux is implemented by its login mechanisms. User mobility is normally limited to an administrative domain and requires the configuration of the Network Information Service (NIS) server and Network File System (NFS) server. A Linux user is able to log into his account and access his files from any computer in a given domain.

Session Mobility The user can transfer an ongoing session from one device to another, maintaining the session.

Linux currently offers no support for session mobility and it has to be implemented at the application level.

In the XtreemOS Project, user mobility will be handled by WP2.1, whereas session mobility will be mostly handled by the different applications and by the different XtreemOS services for mobility (WP3.6). Given that WP2.3, together with WP3.6 will be in charge of terminal mobility, in this section special focus will be placed upon Linux support for terminal mobility.

2.3.1 Linux and Mobile IPv4

Mobile IPv4 specified the first set of enhancements that allowed the transparent routing of IP datagrams to and from mobile nodes, incorporating mobility capabilities in Linux systems.

There exist several implementations of Mobile IPv4 for Linux. Unfortunately, most of the non-commercial implementations are outdated and no longer maintained. The main reason for this is that Mobile IPv4 had several limitations (such as the problem of triangular routing or the need for a foreign agent element) that kept it from being as widely deployed as it was expected to be.

Dynamics Mobile IP This implementation of Mobile IPv4 is available for kernel versions 2.2.x and 2.4.x under the GNU General Public License (GPL). It does not require any kernel patching, although some kernel options are required (such as `CONFIG_NET_IPIP` or `CONFIG_IP_ADVANCED_ROUTER`) and hence the kernel may need to be recompiled in order to support them. Most likely, this code can be adapted to run on 2.6.x kernels, although this hasn't been tested yet.

MosquitoNet Linux Mobile IP This implementation was performed in 1996 for kernel version 1.2.13 and currently supports up to kernel version 2.2.16. This implementation tries to avoid the necessity of a foreign agent by incorporating a foreign agent in the mobile node. In this way, when the mobile node enters a network without foreign agent, the mobile node itself decapsulates the packets from the home agent.

MosquitoNet's main website [31] is down at the time of this writing and it does not seem to be maintained.

2.3.2 Linux and Mobile IPv6

Mobility support went through major improvements in IPv6. Mobile IPv6 solves the problem of triangular routing (packets do not need to go through the home agent in order to reach the mobile node) by introducing an extensible packet header that includes both the home and care-of addresses. Moreover, Mobile IPv6 also eliminates the need to have foreign agents in visited networks. Instead, the mobile device can generate its own IP address by combining the prefix of the visited network on which it is located with a unique device identifier, such as its Media Access Control (MAC) address.

The most popular implementation of Mobile IPv6 for Linux is represented by the Mobile IPv6 for Linux (MIPL) [30] project. This project started up in 1999 as a student project at the Helsinki University of Technology. The initial implementation was further developed in order to match new drafts of the protocol as they came along. In late 2003, MIPL was completely rewritten and started being referred to as MIPL2. This complete rewrite is composed of a user-space daemon which handles MIPv6 signalling, plus a thin in-kernel layer represented by a patch for kernel version 2.6.16.

MIPL2 was taken up by the Universal Playground for IPv6 (USAGI) [49] project, which has as its main goal the improvement of the IPv6 environment in Linux. USAGI is now in the process of merging MIPL2 patches into the mainline Linux kernel. This merge is planned to be completed soon and once completed it will be possible to install Mobile IPv6 support without any kernel patching.

MIPL2 includes support for nodes acting as mobile node (MN), correspondent node (CN) or home agent (HA).[2] In the latter case of a node acting as an HA, the Linux IPv6 Router Advertisement Daemon (`radvd`) [44] is also needed in order to send the Router Advertisement (RA) messages as specified by [32]. These messages are sent periodically and whenever requested by a node sending a router solicitation (RS) message. RA and RS messages are required for IPv6 stateless autoconfiguration. The `radvd` daemon runs in user-space and does not require any changes to the kernel.

[2] See deliverable D2.3.1 [60], page 17.

2.3.2.1 Multiple Care-Of Address Registration

Using this approach, a mobile node would be assigned multiple care-of addresses which would be bound to a single home address. In this way, a mobile node would be able to take advantage of all of its network interfaces in a seamless way, because correspondent nodes would only know about the mobile node's unique home address.

The current Mobile IPv6 specification lacks this capability, although work is being done on this topic [9, 10] and an IETF RFC (currently an Internet-Draft) [51] is being prepared for Mobile IPv6 to support multiple care-of address registration.

2.3.2.2 Bootstrapping MIPv6 Handovers

Handovers in Mobile IP have the drawback of being rather slow (two seconds approximately) for critical applications such as audio or video broadcasting. In order to mitigate this effect, several improvements have been presented for Mobile IPv6 and are now standardised by the Internet Engineering Task Force (IETF):

Hierarchical Mobile IPv6 (HMIPv6) [47] This protocol introduces a hierarchical structure of mobile agents in order to reduce the registration latency and the possibility of an outdated CoA.

The most current implementation for Linux of HMIPv6 dates back to year 2003 and is based on a draft version of the standard, which was released in 2005. This implementation consists of a series of patches for MIPL version 0.9.4, and the Router Advertisement Daemon (`radvd`) version 0.7.2. The fact that this implementation is rather outdated implies that this protocol has not reached a great popularity. This may be due to the fact that HMIPv6 requires network support with the addition of new network elements like the Mobility Anchor Point (MAP), which is used by MNs as a local home agent.

Fast Handovers for Mobile IPv6 (FMIPv6) [21] This protocol tries to reduce the handover latency by configuring a new IP address before the MN enters the new subnet. FMIPv6 requires link-layer information about radio signal strength. This is a functionality that most wireless cards provide.

Given that FMIPv6 does not require any changes to the network infrastructure, it has been more widely deployed than HMIPv6. There is fairly recent implementation of FMIPv6 for Linux [12] which is based on MIPL2 patches for kernel 2.6.16.

Furthermore, FMIPv6 and HMIPv6 can be used together to further improve handover performance. This combination has been reported to give the best results in terms of handover delay and packet loss [13].

2.3.3 Other Mobility Optimisations

In this section, other mobility optimisations that would be important to have on a mobile device and their Linux support will by analysed.

2.3.3.1 Multihoming Support

Multihoming is a technique that increases network reliability and is often used on nodes with several network interfaces (i.e. a WiFi interface and a UMTS interface). In order to obtain a durable and wide area network connectivity it is important for mobile devices to use various types of network interfaces.

There are basically two ways to support several network interfaces on a mobile device:

Assigning a Different IP Address to Each Network Interface With this approach, each interface is given a different IP address. This is the most widely used approach in current Linux systems. However, this approach has the disadvantage that if one of the interfaces loses its connectivity, existing connections cannot be transferred to the other interface because neither the TCP nor the UDP protocols support this feature. A new transport protocol, the Session Control Transmission Protocol (SCTP) [48] does support this situation and is supported by the Linux kernel (still in experimental status) since version 2.6.12. Unfortunately, this protocol is not as widespread in current applications as TCP or UDP are.

Assigning a Single IP Address to Several Interfaces This approach is what is usually meant by multihoming. In this case, several interfaces on a node share the same IP address. If one of the interfaces fails, packets will continue to reach the node through the other interface. This may be very useful for a mobile device which may, for example, have network access via two wireless access points.

Enabling this kind of multihoming is not an easy endeavour, and support from a routing protocol such as the Border Gateway Protocol (BGP) [45] is needed. With the help of this protocol, packets coming from remote nodes will arrive through any of the MD's interfaces.

An implementation of the BGP protocol for Linux platforms is provided by the GPL licensed software Quagga [43]. Quagga is a routing software suite, which was born as a fork of the GNU Zebra [61] project and implements other routing protocols besides BGP such as the Open Shortest Path First (OSPF) (versions 2 and 3) or the Routing Information Protocol (RIP) (versions 1, 2 and 3).

Nevertheless, even though theoretically it is possible to use the BGP protocol to enable this kind of multihoming, this feature is not yet standardised for IPv6 and it is still unclear how asymmetries on the interfaces will exactly be handled (i.e. different bandwidth, error rate, etc.).

2.3.3.2 Mobile Ad-hoc Networks (MANET)

A mobile ad-hoc network (MANET) is a self-configuring network of mobile routers and associated hosts connected via wireless links. In a MANET, network topology may change rapidly and unpredictably as a consequence of the movements of the nodes. The great advantage of MANETs is that they can bring network connectivity to places that would otherwise be unreachable. For XtreemOS-MD, supporting MANET routing protocols can improve MDs connectivity and service pervasiveness even in the absence of network access points.

There are several routing protocols that enable MANETs [1]. These protocols can be divided into three different groups:

Proactive (global) protocols In proactive protocols, the routes to all the destinations are determined at startup and maintained by periodic updates.

Reactive (on-demand) protocols In this case, routes are determined only when they are required via a route discovery process.

Hybrid protocols This is a combination of both proactive and reactive routing.

In the following paragraphs, three of the most used routing protocols for MANETs are described:

Dynamic Source Routing (DSR) [18] The DSR protocol requires every packet to carry every hop in the route in order to arrive to its destination node. Hence this protocol will not be very efficient in large networks, as the amount of overhead will increase with network size.

The only implementation for Linux of the DSR protocol [41] dates back to 2001 and consists of a kernel module for kernel 2.4.3 and a user-space routing daemon. This implementation was tested on an iPAQ PDA running a Familiar Linux 0.4 distribution.

Ad-hoc On-demand Distance Vector (AODV) [40] This protocol is based on DSR but in AODV each packet carries only the destination address rather than the whole route.

Kernel AODV [20] is an implementation for Linux consisting on a kernel module for the Linux kernel 2.4.x series. This module implements AODV routing between computers equipped with WLAN devices. Kernel AODV also requires netfilters support in the system, for which rebuilding the kernel may be needed.

Optimised Link State Routing (OLSR) [6] As opposed to DSR and AODV, which are both reactive protocols, OLSR is a proactive protocol, which means that the routes to all the nodes in the MANET need to be maintained all the time.

OLSR is perhaps the routing protocol for MANETs which is best supported in Linux, as it is also the most recently developed. Some OLSR implementations are even IPv6-enabled. The most widely used implementation of the OLSR protocol is OLSRd [33]. This implementation is available for Linux, Windows, and OSX systems. OLSRd is implemented as a user-space daemon and does not require any patching to the Linux kernel.

2.3.4 Conclusions

Mobile IP specifies a set of enhancements to allow the transparent routing of IP packets between mobile nodes. Therefore, Mobile IP and its different optimisations are a basic requirement for supporting terminal mobility in XtreemOS-MD.

To improve handover times in Mobile IP, optimisations such as HMIPv6 or FMIPv6 have been proposed. There exist implementations of these optimisations available for Linux and XtreemOS-MD can take advantage of them in order to improve handover times.

XtreemOS-MD can also implement multihoming techniques so that mobile devices can utilise several network interfaces simultaneously.

Finally, MANETs is another concept that XtreemOS-MD can take advantage of. By allowing mobile terminals to act as routers forwarding packets to other mobile terminals, the network can be extended to places otherwise unreachable. Virtual organisations can provide MANETs with the security that they need in order to safeguard the privacy of the forwarded packets.

Chapter 3

Requirements for XtreemOS-MD F-layer

In this section, we will list and explain the main requirements for XtreemOS-F in its flavour for mobile devices. These requirements have been gathered from several sources, including:

- Research on state of the art in Linux for mobile devices and other mobility issues in Linux (summed up in chapter 2)
- Requirements derived from the hardware/network study of mobile devices done in deliverable D2.3.1 [60].
- Requirements from XtreemOS Grid applications, extracted in WP4.2 (mainly deliverable D4.2.1 [57]) and further refined here.
- Security requirements from D3.5.2 [58].
- Requirements derived from the first specifications of other XtreemOS components (XtreemOS filesystem, Application Execution Management, etc.), in deliverables D2.1.1 [55], D3.1.1 [54], D3.2.1 [53], D3.3.1 [56], and D3.4.1 [59].

These requirements have been classified into three categories, according to their importance and degree of complexity. These categories are:

Basic These are the first and most important requirements for XtreemOS-MD, and without them, fulfilling the others would not be feasible. Thus, these requirements should be implemented first.

Advanced These are requirements that, due to their increased complexity or lesser importance, should be implemented in the advanced version of XtreemOS-MD. Nevertheless, if time allows, some of them can be included in the release of the basic version.

Optional These are less important requirements, although they are desirable in a mobile and ubiquitous system.

Therefore, the necessary information to define a requirement is:

- Requirement identifier (R2.3.X).
- Short name.
- Brief description.
- Detailed description.
- Source of the requirement (dependency):
  - State of the Art (chapter 2), indicated as "State of the Art"
  - Deliverable D2.3.1, indicated as RMD2.3.X
  - Deliverable D4.2.1, indicated as RXX
  - Deliverable D3.5.2, indicated as GSRXX
  - Other XtreemOS deliverables, denoted with the deliverable code (DX.Y.Z) and the requirement specific code, if available.
- Level of importance: Basic|Advanced|Optional.

3.1 General Requirements

R2.3.1: Compatibility with Common Hardware

XtreemOS-MD must run on the most common hardware.

The XtreemOS-MD flavour must be able to run on the most common hardware for mobile devices (PDAs and mobile phones). As of today, this means being able to run on ARM architecture devices.

Depends on: RMD2.3.1, RMD2.3.17
Importance: Basic

R2.3.2: Battery Shortage

XtreemOS-MD must notice users and services in MDs of low battery levels.

XtreemOS-MD must warn the users and any grid services that require it, of low battery levels, so that they can take actions accordingly (i.e. logout from the grid, close the applications, mark the node as "about to leave", etc.).

Depends on: RMD2.3.2
Importance: Basic

R2.3.3: Input Methods

> *XtreemOS-MD must allow users to easily interact with the operating system and applications.*

XtreemOS-MD must have one or more means for the user to interact with it. This means being able to input text, select menus and options, etc. using the available MD hardware (stylus, touchscreen, keyboard, special keys...).

Depends on: RMD2.3.3, RMD2.3.4
Importance: Basic

R2.3.4: Connectivity Detection

> *The user must be able to know the network connectivity status (online/offline) and the quality of it.*

XtreemOS-MD must warn the users and any grid services that require it, of low connectivity levels, so that they can take actions accordingly (i.e. logout from the grid, close the applications, mark the node as "about to leave", etc.).

Additionally, the user should also be notified when the connection is completely cut off.

Depends on: RMD2.3.7
Importance: Basic

R2.3.5: IPv6 Support

> *XtreemOS-MD must support IPv6.*

The XtreemOS-MD kernel and components must support the IPv6 network protocol.

Depends on: R11, State of the Art
Importance: Basic

R2.3.6: Offline Operation

> *Users of XtreemOS-MD should be able to access MDs even if it has no access to the Grid.*

XtreemOS-MD should be usable even if it has no network connectivity or if the local user has not been identified as a valid grid user.

Depends on: RMD2.3.9
Importance: Basic

R2.3.7: Software Licensing Mechanisms

> *XtreemOS must support software licensing mechanisms.*

XtreemOS-MD must support the same software licensing mechanisms as any other XtreemOS node (i.e. for the distribution of license files, or connecting to license servers). This requirement would only make sense if grid applications could be run on MDs, and thus it is labelled optional.

Depends on: R9
Importance: Optional

R2.3.8: Context Support

> *XtreemOS-MD must provide the means of extracting available information about the devices hardware and connectivity.*

XtreemOS-MD must be able to extract information about the hardware and connectivity in the device (free memory, processor type, available storage, connectivity, etc.), in order to be used by context-aware applications.

Depends on: RMD2.3.5
Importance: Optional

R2.3.9: Grid Caching Mechanisms

> *Users of XtreemOS-MD should be able to perform certain grid operations even when offline.*

XtreemOS-MD users should be able to perform certain grid operations even if temporarily disconnected. Those operations would be stored locally and performed when connectivity (and proper authentication) resumes.

Users should be made aware that these operations are done locally (and tentatively, as there is the possibility of them not being possible once the user is reconnected).

Depends on: RMD2.3.9, RMD2.3.10
Importance: Optional

R2.3.10: Easy Installation and Customisation of Mobile Grid Services

> *XtreemOS-MD must provide the users with some means for easy installation and customisation of XtreemOS grid services.*

Due to the limited resources of MDs, it would be desirable to offer the users a way of installing only the grid services that they expect to use in the future. This can be implemented, for example, through packages, with a "core" package with the basic support for VO operation, and other packages with additional functionalities (i.e. XtreemFS access, AEM access, etc).

Depends on: State of the Art
Importance: Optional

3.2 Virtual Organisation Support in XtreemOS

R2.3.11: A VO management interface has to be provided

> *The management of VOs (creation, deletion, modification, etc.) must be possible from MDs via an API and a GUI. It must also be possible to monitor VO information.*

VO management is a critical aspect of XtreemOS. Hence, XtreemOS-MD needs to provide administrators with all the possible means to manage VOs even when they are out of the office. An API will be designed so that MDs' applications can take advantage of all of the functionality of XtreemOS VO management. The GUI will provide an easy-to-use graphic interface that will allow system administrators to add users or resources, change permissions, roles and/or policies from MDs. The specification of the GUI will consider the reduced graphical interface capabilities of mobile devices.

Depends on: R19, R25, R79, GSR4, GSR5, GSR10-GSR15
Importance: Basic

R2.3.12: Integrity in VO management operations

> *VO management operations performed from MDs will not be compromised by MD's churn.*

VO management operations are critical, and when performed from mobile devices, their unreliability (i.e. sudden loss of connectivity) could lead to inconsistencies on VO-related stored metadata, compromising system integrity. For this reason, the necessary mechanisms will be implemented on MDs so as to ensure the atomicity of VO management operations.

Depends on: R20, R27, R99
Importance: Advanced

3.3 Checkpointing and Restart

R2.3.13: Checkpointing and restart from MDs

> *XtreemOS-MD must support checkpointing and restart management from MDs.*

It must be possible to manage checkpointing and restart of applications running on other nodes from MDs.

Depends on: R28–R35
Importance: Basic

R2.3.14: Checkpointing and restart will not be compromised by MD churn

Unreliability of MDs will not affect checkpointing and restart of Grid applications.

It is necessary that checkpoints and restarts performed from MDs are completed successfully, even if the MD loses its connectivity while the checkpoint is being taken.

Depends on: R28–R35
Importance: Advanced

3.4 Federation Management

R2.3.15: Access to federations

It must be possible to access federations from MDs.

Clusters running XtreemOS federation flavour will appear to be single Grid nodes. It must be possible to access them from MDs.

Depends on: R36–R42
Importance: Basic

3.5 XtreemOS Interfaces

R2.3.16: Interoperability with XtreemOS Standard Flavour

XtreemOS-MD should be compatible with XtreemOS Standard Flavour.

XtreemOS-MD must be able to interoperate with the standard version of XtreemOS to the greatest possible extent.

Depends on: D2.1.1
Importance: Basic

R2.3.17: Demand for POSIX-like Extension

Xtreemos-MD should support POSIX extensions that can be useful to implement mandatory Access Control Lists to files.

Access to files in the XtreemFS filesystem [59] will be controlled by POSIX Access Controls Lists (ACLs). XtreemOS-MD must fully support the management of these control lists via the standard POSIX interface.

Depends on: R45
Importance: Basic

R2.3.18: XtreemOS API Language Support

> *XtreemOS-MD must support the most widely used programming languages in MDs.*

Emphasis must be laid on the fact that widespread languages in PC platforms (such as the latest releases of the Java API) may not be available for ARM architectures. That's why the translation of this requirement to XtreemOS-MD might not be a trivial task and will be monitored.

Depends on: R46
Importance: Basic

R2.3.19: Interoperability with other Linux-based Operating Systems

> *XtreemOS-MD must be compatible with other popular Linux standards for mobile devices.*

In order to promote the adoption of XtreemOS-MD, it is important that the operating system can easily interoperate with other Linux-based operating systems for mobile devices. The reutilisation or easy adaptation of software from other Linux devices are an important factor for XtreemOS-MD's adoption.

Depends on: State of the Art
Importance: Optional

R2.3.20: API Standards as Basis for XtreemOS API

> *XtreemOS-MD must offer support for the most popular Grid APIs, which could be adapted in order to fit MD's particularities.*

XtreemOS-MD must offer support for the most popular Grid APIs (i.e. SAGA). These APIs could be adapted in order to fit MD's particular features.

Although the implementation of this requirement will be done in WP3.6, any implication for XtreemOS-F must be taken into consideration in this WP, for providing adequate support at the OS level.

Depends on: R44
Importance: Optional

R2.3.21: Interoperability with Middlewares

XtreemOS-MD should be compatible with other Grid middlewares.

A main issue in XtreemOS fast adoption might be its interoperability with preexistent Grid middlewares (i.e. Globus Toolkit). That interoperability is seen as an optional requirement as it is not a key feature for XtreemOS but a way to leverage XtreemOS spread.

Depends on: State of the Art, R47, D3.3.1
Importance: Optional

R2.3.22: Session Mobility Interfaces

XtreemOS-MD will support applications with session mobility.

Most of the session mobility support will be developed in WP3.6. Nevertheless, any implication for XtreemOS-F will be taken into consideration in this WP, once technical details are specified in the corresponding task.

Depends on: State of the Art
Importance: Optional

3.6 Highly Available and Scalable Grid Services

R2.3.23: Publish/Subscribe Client

XtreemOS-MD must provide access to publish/subscribe service.

XtreemOS-MD will not act as a publish/subscribe server but as a client (publisher and subscriber). This requirement is not going to be developed in WP2.3 but in WP3.6, but any implications for XtreemOS-F will be taken into consideration in this WP, once this feature is specified in the corresponding task.

Depends on: D3.2.1
Importance: Basic

R2.3.24: Directory Service for Node Monitoring. Subscription

XtreemOS-MD must be client of directory service for other nodes' monitoring and failure detection.

XtreemOS-MD will be able to connect directory service to query and monitor Grid nodes. It is a basic feature for XtreemOS-MD to be able to be up to date of other nodes' status as those nodes can be in charge of execution of jobs launched from MDs.

Depends on: D3.2.1
Importance: Basic

R2.3.25: Directory Service for Node Monitoring. Publishing

> *XtreemOS-MD must be client of directory service to publish node information and node failures.*

XtreemOS-MD should be able to connect directory service to publish node information. As it is not compulsory for XtreemOS-MD to execute Grid applications, this requirement is considered as optional.

Depends on: D3.2.1
Importance: Optional

R2.3.26: XtreemOS-MD Support of Collections

> *XtreemOS-MD must support WP3.2 collections as explained in D3.2.1 ([53])*

XtreemOS-MD must support collections of nodes (grid-wide, application-wide unstructured, application-wide structured collections). The special characteristics of MD nodes must be taken into account, being important to analyse these characteristics and determine which collections can be joined with full functionality and which cannot. It doesn't make sense that MD nodes belong to application execution collection if they are finally not sharing their execution capabilities.

Depends on: D3.2.1
Importance: Basic

3.7 Application Execution Management

R2.3.27: Job Management

> *It must be possible to manage jobs from MDs.*

It must be possible to control the execution of jobs (stop, resume, etc.) from MDs, to change users' privileges, and to apply the new privileges while the user's jobs are running. This requirement could have certain implications for XtreemOS F-layer.

Depends on: R62–R64
Importance: Basic

R2.3.28: Job Monitoring System

> *It must be possible to monitor running applications from mobile devices.*

Monitored information includes (but is not limited to) status, resource consumption and notifications. In any case, this requirement will not be developed in WP2.3 but in WP3.6. Nevertheless, any implication for XtreemOS-F will be taken into consideration in this WP, once technical details are specified in the corresponding task.

Depends on: R52
Importance: Basic

R2.3.29: Job Monitoring Data

> *The monitoring system must provide MD's with sufficient monitoring data.*

This requirement will not be developed in WP2.3 but in WP3.6. Nevertheless, any implication for XtreemOS-F will be taken into consideration in this WP, once technical details are specified in the corresponding task.

Depends on: R54, R55
Importance: Basic

R2.3.30: Tracing Information

> *It must be possible to obtain tracing information about application execution and resources being used from MDs.*

It should be possible to determine which information is considered sufficient. It is important to stress the fact that this requirement will not be developed in WP2.3 but in WP3.6. Nevertheless, any implication for XtreemOS-F will be taken into consideration in this WP, once technical details are specified in the corresponding task.

Depends on: R56–R59
Importance: Basic

R2.3.31: Resource Accounting

> *It must be possible to record mobile users' resource usage.*

Given mobile device's churn rate, resource accounting for jobs launched from mobile devices can be a challenging task (i.e. if accounting is not implemented carefully, a mobile device could lose its connectivity after submitting a job, but before being charged for it). For this reason, mechanisms may be needed so as to ensure that mobile users are properly accounted for their use of resources.

Depends on: R53
Importance: Basic

R2.3.32: Scheduling

> *XtreemOS-MD must be able to co-allocate an application on different resources of several different sites.*

This requirement assumes the co-allocation is performed by the user and not by the system. XtreemOS-F may not be influenced by this requirement but possible implications will be evaluated when the feature is designed in the corresponding WPs.

Depends on: R60
Importance: Basic

R2.3.33: Resource Planning from MDs

> *XtreemOS-MD will permit MD users to perform resource reservations on other Grid nodes.*

It must be possible to perform the reservation of resources for specific intervals, as well as defining specific characteristics of the required resources from mobile devices.

Depends on: R61
Importance: Basic

R2.3.34: Resource Planning on MDs

> *XtreemOS-MD should permit users to perform resource reservations on MD Grid nodes.*

It should be possible to perform the reservation of resources (such as cameras, sensors,...) for specific intervals, as well as specifying specific characteristics of the required resources on mobile devices.

Depends on: R61
Importance: Optional

R2.3.35: Distribution

> *It must be possible to classify MDs by location.*

That means that when launching jobs from MDs, it must be possible to limit their geographical distribution (for example, to nodes geographically near the MD).

Depends on: R67
Importance: Advanced

3.8 Data Management

R2.3.36: Integrity in Filesystem Operations

> *XtreemOS-MD should ensure that filesystem operations done from MDs do not compromise the filesystem integrity.*

Due to unreliability in the MD's network connections, the necessary mechanisms must be put in place to ensure that sudden loss of connectivity or battery does not affect the grid filesystem integrity.

These mechanisms can include using the transactional features of the filesystem, or other custom methods.

Depends on: State of the Art, R81, R82
Importance: Basic

R2.3.37: POSIX-like Client Access to XtreemFS

> *XtreemOS-MD must provide client access to the XtreemOS grid filesystem using a POSIX-like interface.*

XtreemOS-MD must allow users to access XtreemFS grid filesystem using a POSIX-like interface, so that XtreemFS-unaware applications and users can access the data in the grid.

This requirement is included because, although the implementation of this access is a task for WP3.6, this can have some implications for the F-layer (namely, support for json, neon and openssl libraries).

Depends on: R68–R76, D3.4.1
Importance: Advanced

R2.3.38: XtreemOS Custom Access to XtreemFS

> *XtreemOS-MD must provide native client access to the XtreemOS grid filesystem, using its custom interface.*

XtreemOS-MD must allow users to access XtreemFS grid filesystem using the XtreemFS native interface, so that XtreemFS-aware applications and users can access the data in the grid with the maximum control and efficiency.

Again, although the implementation of this access is a task for WP3.6, this can have some implications for the F-layer (support for Python and certain other dependencies).

Depends on: R68–R76, D3.4.1
Importance: Advanced

R2.3.39: Meaningful Filesystem Information

Users of XtreemOS-MD should be able to see meaningful information when accessing files in XtreemFS.

Users and user applications should be able to see XtreemFS as any other filesystem. Thus, meaningful information about grid users and VOs should be given to them when necessary (i.e. fstat displaying VO IDs and global UIDs, instead of local ones).

Depends on: R68–R76, D3.4.1
Importance: Optional

R2.3.40: OSS Implementation in MDs

XtreemOS must implement Object Sharing Service (OSS) mechanisms.

XtreemOS should offer OSS mechanisms for memory sharing between grid applications, as any other XtreemOS node.

This kind of implementation would only be useful in the case that grid applications can be executed in MDs, and thus is labelled optional. Again, the implementation would be done in WP3.6, but can have implications in lower layers.

Depends on: R68–R76, D3.4.1
Importance: Optional

R2.3.41: Filesystem Servers in MDs

XtreemOS-MD must include the necessary software so that a mobile node can act as a XtreemFS server.

XtreemOS-MD must implement one or more of the XtreemFS entities (OSD, MRC, RMS) in order to act as a filesystem server.

The implementation of XtreemFS entities in MDs (OSD, MRC, RMS) would be carried out in WP3.6. But some implications in the F-layer are foreseen in that case.

Depends on: R68–R76, D3.4.1
Importance: Optional

3.9 Security in Virtual Organisations

R2.3.42: User Validation

Users' access policies must be checked before executing any operation on a MD.

Apart from user identification, XtreemOS-MD must also implement user validation mechanisms to check if the user is allowed to perform certain operations according to the security parameters and rules negotiated to become a member of that VO.

Depends on: R85
Importance: Basic

R2.3.43: Access to User's Credentials

> *Applications run by global user's should carry the global user's credentials*

This requirement is necessary in order to implement credential delegation. An application running under a certain global user's credentials must be able to launch another application (possibly on a different node) under the same user's credentials.

Depends on: R84, R85, GSR39
Importance: Basic

R2.3.44: Single Sign-on, Authorisation and PKI

> *Single sign-on mechanisms must be implemented on XtreemOS-MD to avoid the user authenticating each time he accesses a resource.*

XtreemOS-MD must implement single sign-on mechanisms to gain authorised access to resources in a VO. To this aim, PKI mechanisms as well as authorisation mechanisms are necessary.

Depends on: R83, R84, GSR6
Importance: Basic

R2.3.45: VO Access from MDs

> *Users must be able to access VOs from MDs.*

Users connected to the Grid from MDs can become members of a VO.

Depends on: R79, GSR4, GSR5, GSR10-GSR15
Importance: Basic

R2.3.46: Confidentiality

> *Information exchanged within a VO must be transported via secure channels.*

All data sent and received by an MD as a VO member must be confidential, i.e. transported via secure channels to be protected from invalid observers. Encryption methods are a must to keep confidentiality. These confidentiality mechanisms should be applied to all data in which a user of a VO was not able to select his own security preferences (i.e. default option).

Depends on: R80
Importance: Basic

R2.3.47: Network Traffic Validation

> *XtreemOS-MD must implement a monitor to control both incoming and outgoing messages and redirect them to the corresponding processing mechanism.*

There is the need for an OS reference monitor mechanism to capture and validate all incoming and outgoing network traffic to ensure that data integrity and confidentiality mechanisms (e.g. encryption and digital signature) are applied. This monitor could be a kind of agent or OS daemon.

Depends on: R81, R82
Importance: Basic

R2.3.48: Integrity

> *XtreemOS-MD must keep data integrity with digital signature mechanisms.*

XtreemOS-MD must implement digital signature mechanisms to prevent and detect the loss of integrity of stored data (i.e. illegal changes to data). These illegal changes mean possible changes to data performed by other VO members who are not the owners of the data.

Depends on: R81, R82
Importance: Basic

R2.3.49: Lightweight Security Standards

> *Possibility to implement lightweight security standards on XtreemOS-MD.*

If possible, light security standards (e.g. for PKI, PAM and SSH) will be implemented on MDs, due to their computation, storage and battery limitations.

Depends on: R90
Importance: Basic

R2.3.50: Security Parameters Negotiation

MDs must negotiate security parameters before becoming a member of a VO.

MDs will be able to negotiate their own security parameters before becoming members of a VO. In fact, each VO can try to achieve a different set of security objectives and information flow policies. Thus, XtreemOS-MD must implement these negotiation mechanisms.

Depends on: R80, R89, GSR3
Importance: Basic

R2.3.51: Security Assessment

It must be possible to evaluate how secure an MD is.

XtreemOS-MD must implement a security assessment mechanism to evaluate their actual security conditions and therefore be aware of how secure they are. Hence, the VO administrator could be able to classify nodes according to service and trustworthiness level.

Depends on: R95, GSR9
Importance: Basic

R2.3.52: WS Support

XtreemOS-MD must support WS.

XtreemOS-MD must support WS to be compliant with WS security standards (e.g. Globus GSI).

Depends on: GSR1
Importance: Basic

R2.3.53: Logging

A configurable logging service must be available on XtreemOS-MD.

XtreemOS-MDs must implement a logging service to keep track of the critical operations performed on it. Due to the little storage capacity of MDs, it should be possible to store log files in other nodes. It will also be possible to specify different logging levels to control the amount of stored information.

Depends on: R86
Importance: Basic

R2.3.54: Authentication of Software or Components

An authentication for software components must be provided.

As well as for users, XtreemOS-MD must implement an authentication mechanism for software or components in order to ensure the identity and reputation of the software provider. This requirement only makes sense in the case of grid applications being executed on MDs.

Depends on: R91
Importance: Optional

R2.3.55: Support for Crypto Accelerators

It will be studied the possibility to implement crypto accelerators on XtreemOS-MD.

XtreemOS-MD will support crypto accelerators to perform encryption tasks more rapidly, due to their limited computation capacity.

Depends on: R94
Importance: Optional

R2.3.56: Reputation

XtreemOS-MD must implement a reputation system.

XtreemOS-MD must implement a reputation system to control reputation of users, resources, software, etc. to be taken into account in access control decisions. This means e.g. a user that has previously performed (or tried to perform) some illegal operations within other VOs (and therefore has a bad reputation) will not be allowed to access critical data in our VO.

Depends on: R91
Importance: Optional

3.10 Support for Mobile Devices

R2.3.57: Network connectivity

XtreemOS-MD must provide the necessary software to handle a mobile network connection.

Network connectivity is essential for Grid operation and communication among nodes must be ensured by XtreemOS. XtreemOS-MD must be able to handle network connections through a wireless interface. In any case, the special particularities of MDs and the particular characteristics of wireless connections commonly used by these nodes must be taken into account.

Depends on: R10
Importance: Basic

R2.3.58: MDs Should Be Considered as Special Nodes.

> *Low capacity together with the fact that MDs are not permanently connected, makes it convenient to mark them as special nodes.*

This requirement will prevent other XtreemOS nodes from scheduling jobs on MDs as they might not be reliable enough to carry out some tasks

Depends on: R100
Importance: Basic

R2.3.59: Terminal Mobility

> *XtreemOS-MD must provide terminal mobility.*

XtreemOS-MD must provide terminal mobility, either in offline mode (nomadicity) or online mode (mobility).

Depends on: State of the Art
Importance: Basic

R2.3.60: User Mobility

> *XtreemOS-MD must provide user mobility.*

XtreemOS-MD will let users access the same Grid services from any location and device.

Depends on: State of the Art
Importance: Basic

R2.3.61: Java Support

> *XtreemOS for MD must support Java*

As pointed out in D4.2.1, the applications to be run over XtreemOS-MD require Java support. This means that support for a certain version of Java over ARM architecture will be needed, from the ones available at the moment. It should be noted that developing Java Virtual Machines is *not* among the project's goals.

Depends on: R96
Importance: Basic

R2.3.62: Session Mobility

> *XtreemOS-MD must provide session mobility.*

When a user changes his/her location, it should be able to maintain his/her session. This issue is solved at the application layer, so this requirement will monitor if any special functionality is needed to be implemented at XtreemOS-F level.

Depends on: State of the Art
Importance: Optional

R2.3.63: Modes of Operation

> *XtreemOS-MD must support different modes of operation, depending on the power source and network connectivity.*

Mobile devices' conditions can vary over time. Depending on these conditions, it can be necessary to temporarily disable some services (i.e. to save battery live).

Depends on: State of the Art
Importance: Optional

Chapter 4

XtreemOS-MD F-layer Specification

This section gathers all the requirements in Chapter 3 and refines them into more concrete specifications, thus transforming *what* has to be fulfilled by XtreemOS-MD (in its Foundation layer) into *how* it will be done. This specifications will determine which software will have to be included in the basic version of this VO-enabled Linux OS.

4.1 Basic Specifications

S2.3.1: ARM Architecture

> *XtreemOS-MD must be available for the ARM architecture.*

Because it is the most widespread architecture in the market of embedded devices, XtreemOS-MD must offer support of the ARM architecture.

Depends on: State of the Art, R2.3.16
Importance: Basic

S2.3.2: MD Nature Publishing

> *XtreemOS-MD must publish the mobile nature of the node.*

When an MD node joins the Grid, XtreemOS-MD must communicate its mobile nature to the node monitoring system and any other discovering service to let the rest of the nodes know its particular characteristics as a mobile node.

Depends on: R2.3.58
Importance: Basic

S2.3.3: Basic Input – Hardware

> *XtreemOS-MD must support the basic input interfaces that the PDA provides.*

Available input interfaces will depend on the particular device. However, basic input interfaces (i.e. a keypad) must be supported by XtreemOS-MD in order to be a usable distribution.

Depends on: R2.3.3
Importance: Basic

S2.3.4: Basic Input – Virtual Keyboard

> *XtreemOS-MD must include a virtual keyboard program in order to make up for the absence of a physical one.*

This virtual keyboard program must allow the user to input characters as if he had a physical keyboard. In the case of mobile devices provided with touchpad screens, the user must be able to enter characters by pushing on the keys displayed on the MD's touchpad screen.

Depends on: R2.3.3
Importance: Basic

S2.3.5: Normal Offline Operation

> *XtreemOS-MD must be a usable OS in the case of not being connected to the network.*

Local users must be able to log into the mobile device even in the absence of network connectivity.

Depends on: R2.3.6
Importance: Basic

S2.3.6: IPv6 Kernel Support

> *XtreemOS-MD must provide kernel support for IPv6.*

Kernel support for the new version of the IP protocol is required, not only for MDs, but also for any XtreemOS-compliant device.

Depends on: R2.3.5
Importance: Basic

S2.3.7: Support for MIPv6

XtreemOS-MD must support Mobile IPv6.

In mobile devices, the mobile version of IPv6 (MIPv6) must also be supported in order to handle terminal mobility issues. By using Mobile IPv6, mobile devices will be able to seamlessly roam from network to network, maintaining a single IP address regardless their physical location and maintaining ongoing connections alive. Xtreemos-MD implementation of MIPv6 will be very probably based on MIPL2 as it has been recently merged to become part of the Linux kernel mainline.

Depends on: R2.3.57
Importance: Basic

S2.3.8: Java Support

XtreemOS-MD must support the execution of Java applications.

XtreemOS-MD must provide some kind of Java Runtime Environment (JRE), to support the execution of several XtreemOS components and reference applications.

This specification not only asks for the inclusion of a Java Virtual Machine, but also a set of standard class libraries (or API) consistent with it. The election of a concrete JVM and API should take into account, among other things, their license policies and the fact that they implement a modern API (at least J2SE 1.4 API).

Depends on: R2.3.40, R2.3.41
Importance: Basic

S2.3.9: PAM Support

XtreemOS-MD must support the Linux pluggable authentication modules (PAM).

In order to support various VO models and various security models without the need to recompile all secutity-dependent applications each time a new model is introduced, XtreemOS-MD applications and services will use PAM modules for authentication, authorisation and session management.

Depends on: R2.3.39
Importance: Basic

S2.3.10: Linux System Databases and Name Service Switch Support

XtreemOS-MD must support Linux System Databases and Name Service Switch (NSSWITCH).

In order to support dynamic configuration of user accounts and the on-the-fly translation between local user IDs and global user IDs, XtreemOS-MD will use the NSSWITCH name service management support of Linux.

Depends on: R2.3.17, R2.3.39
Importance: Basic

S2.3.11: Local Mapper

> *XtreemOS-MD must offer a service for the mapping of global to local identities and vice-versa.*

This service will allow global users from mobile devices to obtain information about global identities of both users (UIDs) and virtual organisations (GIDs).

Depends on: R2.3.39
Importance: Basic

S2.3.12: POSIX ACLs Support

> *XtreemOS-MD must support POSIX ACLs as defined in IEEE 1003.1e/2c [16].*

ACLs support has been imposed by applications and will also be supported in mobile devices. Besides, these ACLs must be extended to the semantics of global users and virtual organisations.

Depends on: R2.3.17
Importance: Basic

S2.3.13: Publish/Subscribe Client Implementation Support

> *XtreemOS-MD must implement the client of the publish and subscribe service*

That means to provide the needed libraries (like a TCP stack to enable connection to the service) to allow Publish/Subscribe client work once ported to the basic MD hardware architecture, ARM. The porting of the client itself is not a task scheduled in this WP but in WP3.6. In this WP we must provide appropriate foundations for the client execution.

Depends on: R2.3.23
Importance: Basic

S2.3.14: FUSE Support

XtreemOS-MD kernel must support the FUSE module.

Support for the FUSE (Filesystem in Userspace) kernel module is required in order to access the XtreemFS. FUSE is merged into the mainline Linux kernel since version 2.6.14.

Depends on: R2.3.37
Importance: Basic

S2.3.15: Miscellaneous Libraries for Supporting VO Management

XtreemOS-MD must include the necessary libraries for supporting VO management.

The following libraries are needed in order to support the work developed in WP2.1:

- `libpam0`
- `libc6`
- `libdb` (version 4.3 or higher)

Depends on: R2.3.11
Importance: Basic

S2.3.16: Miscellaneous Libraries for Supporting XtreemOS API

XtreemOS-MD must include the necessary libraries for supporting XtreemOS API.

The following libraries are needed in order to support the work developed in WP3.1:

- `libboost` (version 1.33 or higher)
- `xmlrpc++` (version 0.7 or higher)
- One of:
  - `sqlite3` (version 3.0 or higher)
  - `postgresql` (version 8.0 or higher)

Depends on: R2.3.16, R2.3.18, R2.3.20
Importance: Basic

S2.3.17: Miscellaneous Libraries for Supporting XtreemOS AEM

> *XtreemOS-MD must include the necessary libraries for supporting XtreemOS AEM.*

The following libraries are needed in order to support the work developed in WP3.3:

- `log4j` (version 1.2 or higher)
- `dom4j`

Depends on: R2.3.27, R2.3.31, R2.3.33, R2.3.34
Importance: Basic

S2.3.18: Miscellaneous Libraries for Supporting the XtreemFS

> *XtreemOS-MD must include the necessary libraries for supporting the XtreemFS.*

The following libraries are needed in order to support the work developed in WP3.4:

- `libssl` (version 1.2 or higher)
- `libc6`

Depends on: R2.3.37, R2.3.38
Importance: Basic

S2.3.19: Implementation of Security Mechanisms

> *XtreemOS-MD F-layer will support the concrete security mechanisms specified for XtreemOS standard version.*

XtreemOS-MD will implement the precise client-side security mechanisms specified for XtreemOS standard version. The Foundation layer of XtreemOS-MD will support this functionality at the operating system level.

Depends on: R2.3.46, R2.3.50, R2.3.36, R2.3.48, R2.3.47, R2.3.44, R2.3.42, R2.3.53, R2.3.54, R2.3.56, R2.3.51, R2.3.52
Importance: Basic

4.2 Optional Specifications

S2.3.20: Location Information

> *XtreemOS-F must be able to handle positioning information concerning the MD, provided that the MD is able to provide this information.*

This information may be given in terms of geographical coordinates obtained from a GPS interface, GSM cell id, user metadata or any other appropriate means. Location information will be offered to higher level services.

Depends on: State of the Art, R2.3.35
Importance: Optional

S2.3.21: Battery Warnings

> *XtreemOS-MD must support the issue of notifications when certain battery level is reached.*

As this feature may require kernel support for the specific mobile device it is considered as optional

Depends on: R2.3.2
Importance: Optional

S2.3.22: Network Warnings

> *XtreemOS-MD must support the issue of notifications when certain network level is reached.*

A system service must continuously monitor network signal strength level on MDs. This level can then be queried by applications or by other XtreemOS nodes. It may depend on specific device driver and is considered optional.

Depends on: R2.3.4
Importance: Optional

S2.3.23: Advanced Input – Handwriting recognition

> *XtreemOS-MD must include handwriting recognition applications.*

Advanced input methods are necessary so that users feel comfortable using the mobile devices. Users must be able to enter text by handwriting the characters on the MD's touchpad screen.

Depends on: R2.3.3
Importance: Optional

S2.3.24: MIPv6 Enhancements

XtreemOS-MD might implement MIPv6 enhancements.

Enhancements like FMIPv6 or HMIPv6 to improve MIPv6 operation, reduce handover time and might be implemented in XtreemOS-MD. XtreemOS-MD should, in that case, integrate or adapt correspondent implementations.

Depends on: R2.3.59
Importance: Optional

S2.3.25: Isolation

Isolation in XtreemOS-MD.

XtreemOS-MD must provide the same level of isolation as XtreemOS standard. Due to the limited capabilities of mobile devices, isolation can be implemented by other means than virtualisation, LSM or SElinux as it has been proposed in XtreemOS-F. Isolation can, for instance, be provided through policy rules which avoid simultaneous execution of some applications on the same mobile device.

Importance: Optional

S2.3.26: MLI Support

XtreemOS-MD must follow the guidelines provided by the Mobile Linux Initiative (MLI).

These guidelines must be met in order to be interoperable with other existing operating systems for mobile devices.

Depends on: R2.3.19
Importance: Optional

S2.3.27: SAGA Support

XtreemOS-MD must offer support for applications using the SAGA API.

The SAGA API will be officially supported by XtreemOS. Moreover, mobility extensions may be added to this API in order to fit MDs particularities.

Depends on: R2.3.20
Importance: Optional

S2.3.28: API Languages Support

> *XtreemOS-MD must offer a development environment so that applications can be created for this flavour of XtreemOS. This development environment will be available at least, for the C, C++, and Java programming languages.*

It is important to offer support for these programming languages, in order to ease the development of applications for XtreemOS-MD. Due to the limitations of mobile devices, the development environment should also be available for cross-compilation.

Depends on: R2.3.18
Importance: Optional

S2.3.29: Resource Control

> *Mobile devices must support a service in charge of keeping track of available resources (cameras, sensors, etc.), as well as their usage.*

In order to implement resource planning, applications must be able to reserve MD's resources for a determined amount of time. MDs must expose the resources that they have so that applications can use them.

Depends on: R2.3.33, R2.3.34
Importance: Optional

S2.3.30: Java XMLRPC support

> *XtreemOS-MD must support XMLRPC for Java applications.*

XtreemOS-MD must include the necessary libraries to support XMLRPC protocol in Java applications, as several XtreemOS components require it.

The election of a concrete implementation of Java-XMLRPC should take into account, among other things, the license, in order to be distributed with XtreemOS.

Depends on: R2.3.41
Importance: Optional

S2.3.31: Lightweight Security Methods

> *XtreemOS-MD might implement lightweight security methods.*

If XtreemOS security methods are found to be heavy for MD's processing capacity, lighter security methods could be integrated in XtreemOS MD to increase performance.

Depends on: R2.3.49
Importance: Optional

S2.3.32: Implementation of Security Mechanisms for Execution in MDs

> *XtreemOS-MD F-layer will support the concrete security mechanisms specified for the execution of jobs in XtreemOS standard version.*

XtreemOS-MD will implement the precise security mechanisms specified for the execution of jobs in XtreemOS standard version. The Foundation layer of XtreemOS-MD will support this functionality at the operating system level.

Depends on: R2.3.46, R2.3.50, R2.3.36, R2.3.48, R2.3.47, R2.3.44, R2.3.42, R2.3.53, R2.3.54, R2.3.56, R2.3.51, R2.3.52
Importance: Optional

Chapter 5

Open Issues

Although this document contains a fairly detailed description of the features that XtreemOS-F for mobile devices should fulfill, there are still a number of issues which remain unspecified, or about which a final decision has not been taken. In this section we will take a look at all of them, establishing the nature of this "open questions", their importance and why they remain open at the time of this writing.

5.1 Access Networks, Firewalls and NAT

One issue that has not been specified in XtreemOS so far (although it has been already pointed out in the latest XtreemOS meetings) is the influence of the access networks in the services and functionalities that XtreemOS will provide.

In practice, most private users and many corporate and academic users will access the Grid from behind some kind of firewall or Network Address Translation (NAT) mechanism. This means that, if proper measures are not taken to avoid it, the communications to and from XtreemOS services can be disrupted by these mechanisms. The problem is even more urgent when talking about mobile devices, as practically 100% of them are in the private network of some operator, and thus they will be accessing the Internet (and the Grid) through a NAT device.

A study about the most widespread kinds of NAT and the main NAT traversal techniques should be carried out. This, coupled with the detailed characteristics of the services XtreemOS intends to implement, should give us the best solucion to be applied on the whole of the XtreemOS project. Possible solutions may include one or more of the following:

- Adoption of one or more NAT traversal techniques across the XtreemOS grid (i.e. STUN, Teredo tunnelling...), maybe even using different techniques for different grid services.
- Addition of new services and grid nodes specifically devoted to firewall and NAT traversal operations (i.e. STUN servers).

- Specification of network hardware models that support the selected technique(s).

Moreover, the proposed solution(s) should be evaluated in the concrete environment of wireless and cellular operators (for PDAs and mobile phones, respectively), to see if they are adequate for the special case of mobile devices.

5.2 Security Mechanisms

As the reader may have noticed, although the security requirements are quite defined, both in this document and in WP 3.5 documentation (deliverable D3.5.2, "Security Requirements for a Grid-based OS", mainly), there's still a lack of concrete specifications for the security mechanisms in XtreemOS.

Our work here will be specially concerned with the low-level security protocols and mechanisms such as communications encryption (i.e. IPsec), the nature of the certificates (i.e. X.509 or otherwise), etc.

As workpackage 3.5 is releasing a specification document in parallel with this one, a more detailed specification of the kernel options, modules and software packages needed in XtreemOS-MD will be provided in the near future, either as an update of the present document, or as part of the design document for the Foundation layer of basic XtreemOS-MD.

5.3 Licensing

Although the way XtreemOS components should be distributed and licensed is being studied by the Intellectual Property Use and Dissemination Committee (IPUDC), mobile devices deserve a small comment in this regard.

Mobile devices are very different from ordinary PCs in that there is a lack of open source drivers for much of their hardware, as standards are much less advanced in this regard compared to desktop PCs. Thus, we must be very careful when packaging and distributing XtreemOS-MD, in order not to violate the driver's licenses. Either we must distribute XtreemOS-MD without any proprietary drivers (making the distribution less usable by end-users), or distribute them only in binary form (if licenses allow for this).

5.4 MIPv4 Support

Another issue that has not been thoroughly covered in this report is the matter of Mobile IPv4 support. XtreemOS needs to support IPv6 not just because it will surely be the Internet Protocol in the future, but also because some of the applications require it. Mobile IPv6 also seems a good candidate for providing enhanced performance in mobility situations, and will make XtreemOS-MD more competitive in this increasingly mobile world. But, what about its predecessor, MIPv4? should we support it to provide backwards compatibility to existing installations?

Currently, we tend to think that MIPv4 support is not all that desirable, since right now it is not very widely used at all, and will be soon made obsolete by the new and improved MIPv6 protocol and its extensions.

Chapter 6

Conclusions

The final result from this document is the list of specifications that the basic version of Linux-XOS for MDs must meet. In order to obtain this list of specifications, a methodology has been followed in which the current state of the art and requirements lead to the final list of specifications.

A complete state of the art has been outlined with information about the MDs' operating systems and the addressing of mobility issues concerning MDs, introducing insights into the whole problem of OSes for MDs. Linux is increasing its market share in the arena of operating systems for mobile devices, specially in Asia. The most important mobile Linux distributions have been described, emphasising the open-source distributions Familiar and Ångström which are seen as two of the most appropriate candidates to become the foundation of XtreemOS MD flavour.

Afterwareds, the requirements to describe XtreemOS-MD F-layer features and behaviour from the user's point of view have been gathered. Requirements were taken from different sources, specially from the aforementioned state of the art and from work by other workpackages (always looking at the issues from a mobile device point of view), to cover all facets of XtreemOS-MD foundation layer. Those facets include security, VO support, data management, interfaces, etc. Requirements show the need for supporting the most common hardware available on the market, the need for node-level VO support, as well as the need for supporting, at the OS level, the access to XtreemOS Grid services from MDs. Finally, requirements also demand terminal and user mobility.

Once a vision of XtreemOS-MD F-layer has been depicted, our work has focused on translating the requirements into precise specifications for the F-layer of XtreemOS running on MDs. These specifications guarantee the availability of XtreemOS for the ARM architecture. Also, specifications ensure that other Grid nodes will be aware of the mobile nature of MDs. Other specifications provide support for the execution of Java applications, Linux Pluggable Authentication Modules, or the use of lighter security methods.

Chapter 7

Future Work

Now that we have laid out the concrete requirements and specifications for the mobile device version of XtreemOS (or, more concretely, of its Foundation layer), the next step will be the design of XtreemOS-MD. In this endeavour, we foresee several steps:

- First of all, we must decide which Linux distribution is best suited to serve as the starting point for the basic XtreemOS-MD. The main distributions for PDAs will be taken into account, and choice will be done regarding the wideness of its adoption, the strength of the community behind them, the licensing policies and, of course, their initial compliance with the requisites and specifications detailed in this document.

- Afterwards, the main components and building blocks for the Foundation layer of XtreemOS-MD will be detailed, including an extensive list of features and functionalities, the components that implement them and the interfaces between them and with other XtreemOS components, both to support Virtual Organisations in mobile devices and to enhance the mobility of the devices themselves.

- Also, a detailed list of software packages, kernel modules and kernel configurations will be elaborated. This software will serve as a checklist to guarantee that XtreemOS-MD complies with the requirements in this document.

All these steps will be reflected in a report, D2.3.3, entitled “Design of a basic Linux version for mobile devices”.

References

[1] M. Abolhasan, T. A. Wysocki, and E. Dutkiewicz. A review of routing protocols for mobile ad hoc networks. *Ad Hoc Networks*, 2(1):1–22, 2004.

[2] ACCESS Linux Platform (ALP) web site.
`http://www.access-company.com/products/linux/alp.html`.

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

[4] Albion M. Butters. Linux in the Mobile Space: Today and Tomorrow.
`http://www.access-company.com/PDF/Linux_Mobile_Space.pdf`.

[5] Canalys report: Smart mobile device market growth remains steady at 55
`http://www.canalys.com/pr/2006/r2006071.htm`.

[6] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). RFC 3626 (Experimental), October 2003.

[7] Embedded Linux Survey.
`http://www.bluemug.com/research/els/els.shtml`.

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

[9] T. Ernst, N. Montavont, R. Wakikawa, C. Ng, and K. Kuladinithi. Motivations and scenarios for using multiple interfaces and global addresses. Internet-Draft: draft-ietf-monami6-multihoming-motivation-scenario-01 (work in progress), October 2006.

[10] T. Ernst, N. Montavont, R. Wakikawa, C. Ng, and K. Kuladinithi. Analysis of multihoming in mobile ipv6. Internet-Draft: draft-ietf-monami6-mipv6-analysis-02 (work in progress), February 2007.

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

[12] fmipv6.org project.
`http://www.fmipv6.org/`.

[13] Shaojian Fu and M. Atiquzzaman. Handover latency comparison of SIGMA, FMIPv6, HMIPv6, FHMIPv6. In *Procedings of IEEE Global Telecommunications Conference, 2005 (GLOBECOM '05)*, volume 6, pages 3809–3813, November 2005.

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

[15] IDC. Worldwide Handheld QView, April 2006.

[16] Ieee 1003.1e and 1003.2c: Draft standard for information technology–portable operating system interface (posix)–part 1: System application program interface (api) and part 2: Shell and utilities, draft 17 (withdrawn), October 1997. (withdrawn).

[17] The Intimate Project.
`http://intimate.handhelds.org`.

[18] D. Johnson, Y. Hu, and D. Maltz. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. RFC 4728 (Experimental), February 2007.

[19] D. Johnson, C. Perkins, and J. Arkko. Mobility Support in IPv6. RFC 3775 (Proposed Standard), June 2004.

[20] Kernel AODV.
`http://w3.antd.nist.gov/wctg/aodv_kernel/`.

[21] R. Koodli. Fast Handovers for Mobile IPv6. RFC 4068 (Experimental), July 2005.

[22] Todd Kort, Roberta Cozza, Kanae Maita, and Lillian Tay. Dataquest Alert: 3Q06 PDA Shipments of 4.5 Million Are All-Time High, November 2006.

[23] Todd Kort, Roberta Cozza, Kanae Maita, and Lillian Tay. Dataquest alert: Research In Motion lifts PDA market to 6.6 percent growth in 1Q06, May 2006.

[24] Todd Kort, Roberta Cozza, Kanae Maita, and Lillian Tay. Forecast: Mobile Terminals, Worldwide, 2000-2009 (4Q05 Update), January 2006.

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

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

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

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

[29] Carolina Milanesi, Ann Liang, Hugues J. De La Vergne, Nahoko Mitsuyama, and Tuong Huy Nguyen. Market Share: Mobile Devices, Worldwide, 4Q06 and 2006, March 2007.

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

[31] MosquitoNet Project.
http://mosquitonet.stanford.edu/.

[32] T. Narten, E. Nordmark, and W. Simpson. Neighbor Discovery for IP Version 6 (IPv6). RFC 2461 (Draft Standard), December 1998. Updated by RFC 4311.

[33] olsr.org.
http://www.olsr.org/.

[34] Open Mobile Terminal Platform web site.
http://www.omtp.org.

[35] Michael Opdenaker. Embedded Linux Kernel and Driver Development.
http://free-electrons.com/training/drivers.

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

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

[38] 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.

[39] C. Perkins. IP Mobility Support for IPv4. RFC 3344 (Proposed Standard), August 2002.

[40] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (Experimental), July 2003.

[41] picoNet — A wireless ad hoc network for mobile devices.
http://piconet.sourceforge.net/.

[42] Microsoft reveals new Windows Mobile 6.
http://www.microsoft.com/presspass/press/2007/feb07/02-11WM6SoftwareP%R.mspx.

[43] Quagga software routing suite.
http://www.quagga.net/.

[44] Linux IPv6 Router Advertisement Daemon (radvd) Homepage.
http://www.litech.org/radvd/.

[45] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). RFC 1771 (Draft Standard), March 1995. Obsoleted by RFC 4271.

[46] Scratchbox web site.
http://scratchbox.org/.

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

[48] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson. Stream Control Transmission Protocol. RFC 2960 (Proposed Standard), October 2000. Updated by RFC 3309.

[49] USAGI Project Homepage.
http://www.linux-ipv6.org/.

[50] Hugues J. De La Vergne and Roberta Cozza. Forecast: Smartphones, Worldwide, 2004-2009, March 2006.

[51] R. Wakikawa, T. Ernst, K. Nagami, and V. Devarapalli. Analysis of multihoming in mobile ipv6. Internet-Draft: draft-ietf-monami6-multiplecoa-02 (work in progress), March 2007.

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

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

[54] XtreemOS Consortium. First Draft Specification of Programming Interfaces. Integrated Project, December 2006.

[55] XtreemOS Consortium. Linux XOS Specification. Integrated Project, December 2006.

[56] XtreemOS Consortium. Requirements and specification of XtreemOS services for job execution management. Integrated Project, December 2006.

[57] XtreemOS Consortium. Requirements Capture and Use Case Scenarios D4.2.1. Integrated Project, December 2006.

[58] XtreemOS Consortium. Security Requirements for a Grid-based OS. Integrated Project, December 2006.

[59] XtreemOS Consortium. The XtreemOS File System - Requirements and Reference Architecture. Integrated Project, December 2006.

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

[61] Gnu Zebra.
`http://www.zebra.org/`.