Project no. IST-033576

XtreemOS

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

D2.3.6
Design of an advanced Linux version for mobile devices

Due date of deliverable: 31st May 2009
Actual submission date: 17th June 2009

Start date of project: June 1st 2006

Type: Deliverable
WP number: WP2.3
Task number: T2.3.6

Responsible institution: Telefónica I+D
Editor & editor's address: Santiago Prieto
Telefónica I+D, Parque Tecnológico de Boecillo
Boecillo (Valladolid)
SPAIN

Version 1.0/ Last edited by Telefónica I+D/ Date 17-06-2009

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)	

Keyword List: design, linux, mobile device, smartphone

Revision history:

Version	Date	Authors	Institution	Sections Affected / Comments
0.1	12-Jan-09	Telefónica I+D	Telefónica I+D	First draft
0.2	30-Apr-09	Telefónica I+D	Telefónica I+D	Included smartphone comparison and initial contributions to the rest of chapters
0.3	20-May-09	Telefónica I+D	Telefónica I+D	Completed resource sharing and context awareness chapters. Included executive summary. Advances in the rest of the sections
0.4	26-May-09	Telefónica I+D	Telefónica I+D	Draft version for review
0.5	05-June-09	Telefónica I+D	Telefónica I+D	Included comments and modifications from Toni's review
0.6	17-June-09	Telefónica I+D	Telefónica I+D	Included comments and modifications from Alvaro's review

Reviewers
Alvaro Arenas (STFC), Toni Cortés (BSC)

Tasks related to this deliverable

Task No.	Task description	Partners involved °
T2.3.6	Design of an advanced Linux version for mobile devices (Linux-XOS for MD/MP)	TID*, INRIA

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

* Task leader

Executive Summary

Grid computing and other sibling concepts like metacomputing, virtual organizations (VOs) or the more recent “cloud computing”, have been around for more than two decades. However, they have failed to reach wide acceptance, except among certain scientific communities and, even there, complaints are often heard about the difficulties of using and administrating it, especially in large scale projects. The XtreemOS project aims to change this situation by delivering a modified Linux operating system with grid capabilities embedded in it, so that operating inside VOs is transparent for users, easy to manage and scalable to thousands of nodes.

Moreover, it has also been an old aspiration of grid computing to extend its reach, not only to PC computers, but also to other devices, like in this case Mobile Devices that are the focus of this deliverable and also of the whole work done by WP2.3. As commented in previous deliverables, the mobile devices (MDs) considered are PDAs and mobile phones, but the concept could be extended to other kinds of embedded devices such as set-top-boxes or satellites. This could have big impact thinking on the dissemination of XtreemOS, taking into account that there are hundred of millions of mobile devices around the world.

The previous document (D2.3.5, see [1]) has defined the vision for the advanced version of XtreemOS-MD, extending hardware support to more limited devices (such as smartphones), and including also advanced optional functionalities not present in the basic version.

In the present document, we are first analyzing carefully the smartphone market, comparing the current platforms following different criteria, like platform openness, availability and future of the platform used, compatibility between the security model and the XtreemOS philosophy, etc. The main platforms tested have been OpenMoko, Nokia Maemo and Android (LIMO has also been considered, but early discarded as it’s not an open platform). We have decided to prioritize the selection of the platform (not directly discarding the rest), and after the evaluation OpenMoko has been assigned with the highest priority, followed by Nokia Maemo and finally Android. But, depending on the availability of terminals or in possible future considerations of the criteria selected, the priorities could be changed.

After the smartphone evaluation, the document focus on the XtreemOS-MD advanced version architecture and design for the fulfillment of the requirements and specifications identified in D2.3.5, including:

- enhancements on installation, configuration and usability
- VO management enhancements, ad-hoc VO support, security

- design of an API to expose context (location, remaining power…) to other layers or applications

A special interest has been put on the possibility of sharing the resources available on the MD. In previous documents it was already commented that the CPU and storage capacities of the MDs are really limited and not interesting to be offered as Grid capacities so. But after analyzing the possibility of sharing the camera, GPS modules or network access, and taking into account the privacy issues associated, the security problems introduced and the software porting needed (and consequently the resource expenditure needed), we have decided to limit the scope of mobile devices as Grid clients, not just as special nodes. Anyway, interesting features like file sharing will be implemented, as event not being the MD a node there are ways to achieve them.

Finally a section dedicated to software porting to PC platforms has also been included, as it could be interesting to be able to run XtreemOS-MD on a PC (especially thinking on Netbooks).

Table of contents

List of figures

Glossary

AMS	Account Mapping Service
ARM	Advanced RISC Machine
API	Application Programming Interface
IPv6	Internet Protocol v6
KRS	Key Retention Service
MD	Mobile Device
MID	Mobile Internet Devices
MIPv6	Mobile Internet Protocol v6
NSS	Name Service Switch
OHA	Open Handset Alliance
PAM	Pluggable Authentication Modules
PDA	Personal Digital Assistants
RIA	Rich Internet Applications
SAGA	Simple API for Grid Applications
SSH	Secure Shell
SSL	Secure Socket Layer
UID	User Identifier
VO	Virtual Organization
WP	Work Package
XtreemOS-MD	XtreemOS for Mobile Devices

1 Introduction

Previous documents in this work package have defined (D2.3.2, see [2]), designed (D2.3.3, see [3]) and implemented (D2.3.4, see [4]) XtreemOS-MD basic version and also have analyzed the requirements and specifications for an advanced version of XtreemOS-MD (D2.3.5, see [1]), including new functionalities and also extending hardware support to more limited devices (such as smartphones).

In this document we are designing the advanced version of XtreemOS-MD to fulfill the requirements previously defined in D2.3.5 [1].

1.1 Advanced XtreemOS-MD for smartphones

One of the most important differences of the advanced version of XtreemOS-MD is the support for smartphones, while the basic version was just designed to be executed on more powerful platforms, like PDA ones. Nevertheless, the differences between mobile phones and PDAs have been progressively disappearing, and the concept of smartphones is becoming the evolution of both PDAs and mobile phones as will be commented in chapter 2.

As the smartphone market is quite fragmented and the different platforms are not compatible between them, we have carefully analyzed the available platforms and terminals, in order to prioritize the XtreemOS-MD implementation for them. The candidates analyzed are OpenMoko, Nokia Maemo, Android and LIMO, the ones that we initially considered more appropriated as target of XtreemOS-MD (specially for the Linux-based OS running in all of them). The evaluation has been based on criteria like platform openness, availability and future of the platform used, compatibility between the security model and the XtreemOS philosophy, etc.

1.2 Document structure

- First in chapter 2, the mentioned evaluation of smartphones platforms is compiled, including a “selected solutions” section where the final order of priority is assigned.
- Then, in chapter 3 we are covering the architecture of this XtreemOS-MD advanced version, including figures for the higher priority platforms
- Chapter 4 is dedicated to the enhancements of installation, configuration and usability, one of the modifications defined in the requirements for the advanced version
- Chapter 5 is focused on the new functionalities related to VO support, that is precisely one of the core functionalities of XtreemOS-MD F-layer
- In chapter 6 we have described the API that will be offered to other layers and applications to expose the context of the user
- Chapter 7 includes an analysis of the possibilities related to resource sharing, taking into account how different is a MD respect to a PC, that would run the XtreemOS “normal” flavour software
- Chapter 8 is dedicated to XtreemOS-MD porting for PCs, specially useful for the “Netbooks” that recently arrived to the market

- Finally a future work chapter has been included, to identify next steps to finalize the XtreemOS-MD advanced version

2 First Step: Election of Base Distribution for Smartphones

2.1 Introduction

For XtreemOS-MD basic version, the selected distributions were Angstrom and Maemo, really suitable for PDA and MID. Nevertheless, XtreemOS-MD advanced version must be available also for mobile phones.

Our interest on mobile phones is not related to the voice capabilities, but to the dissemination possibilities, as XtreemOS-MD could be provided to the millions of devices sold and subsidized by mobile operators.

Mobile operators don't sell PDAs, but rather devices as smartphones, which use their communications infrastructure for voice or data communication. In fact, sometimes even laptops or netbooks with a data plan are also distributed. Nevertheless, laptop market is almost monopolized by Microsoft Windows OS (excluding ARM netbooks, for which "full" Windows is not available), while emerging smartphone market offers opportunities to other operative systems as Linux or MacOS (iPhone).

In general, phone users are not worried about the OS used by their device and that gives some advantage, to Linux, which allows a more easy porting of desktop applications than Windows (full Window applications are not available today to mobile users).

Unfortunately, Linux smartphone market is very fragmented in platforms that are incompatible among them. Even in some Linux-based platforms, the applications that users can install are not Linux native applications, but developed in Java/Javascript, etc. The integration of native applications (like XtreemOS-MD reference applications for instance) in these platforms is very difficult: in some cases it's needed the adaptation of the stacks, which are out of the public specification, and consequently could be different for each device or version.

Another problem is that some distributions prevent users from installing XtreemOS-MD modules; this issue implies that XtreemOS may be available just if the mobile manufacturer or the mobile operator authorizes its inclusion.

The security model behind many of the Linux-based mobile platforms it's also incoherent with XtreemOS philosophy: there are cases where the applications are completely isolated, not having access to the operating system or even to the file system. In those cases some kind of "gateway" module would be needed to access the file system or to call the native libraries, removing the advantages that XtreemOS integration would offer.

On the other hand, even if the selection of the platform could affect the dissemination, we should take into account that the smartphone users would not be in general the ones installing XtreemOS-MD, but directly the mobile operators and manufacturers, so in this case our first target are them and not directly the end users. In order to have acceptance from the manufacturers and operators, it's important the existence of a developers' community behind the product: that's one of the reasons of the Linux success in embedded devices against other free and good solutions from the technical point of view (like eCOS [5], for instance) but not so "known", lacking a big developers' community.

2.2 Differences between Smartphones and PDAs

As commented in the introduction, the interest on smartphones is not directly related to the intrinsic voice capabilities offered by the phones, but to the bigger market penetration of smartphones against PDAs. The need of a specific version was based on the differences between PDAs/MIDs and smartphones. However, those differences are currently under discussion, as modern smartphones (like for example iPhone, PalmPre, etc.) are breaking the frontiers with PDAs.

Initially, smartphones were known like a type of cell phones which facilitated data access and process with a notable computing power.

As occurred with cell phones, smartphones could provide voice communications and messaging functionality, but also personal information management (PIM), some applications and wireless communications capabilities

The traditional differences considered where:

1. **Hardware**: Traditionally, PDAs provide more hardware resources (CPU, memory) than telephones. This asseveration is today under discussion with the launching of very advanced smartphones: for example, Palm Pre hardware is more advanced than Nokia N8x0 one. On the contrary, hardware peripherals (camera, GPS, 3G modem, accelerometers) were more common in smartphones than in PDAs or MIDs, but many PDAs include currently those elements (or at least they might be added through Bluetooth) and there are some smartphones not including some of them (like OpenMoko reference terminal FreeRunner for instance, that do not provide camera).
2. **Telephone capabilities**: Even if this is the most evident difference, many devices support VoIP using Wifi. On the other hand, voice capabilities are isolated in most smartphone models. If we consider just GPRS or 3G modem capabilities, we should notice that some devices not categorized as smartphones (as the future successor of Nokia N8x0) will include 3G support, and any model with Bluetooth support (including Nokia N800, the PDA selected for the basic version) may use an external 3G modem.
3. **User experience**: Even if we can consider than the evolutions of cell phones and PDAs are converging to a single device, the smartphone, there are some additional factors traditionally considered related to the user experience:
    - Weaknesses of smarphones:
        1. PDAs are often cheaper than a smartphone over the life of the device. Although the initial purchase price of some smartphones is less than the cost of a PDA, users have to pay an extra monthly fee adds due to wireless carrier subsidies, making smartphones more expensive in the long run.
        2. Smartphones are often tied to a wireless carrier's network. If users wish to switch from a wireless carrier to another one, the smartphone they used with the first wireless carrier is unlikely to work on the second one. This means users will have to buy a new smartphone. With a PDA, changing wireless providers is not an issue.
        3. If the smartphone breaks or become lost or stolen, all of the information users have stored on it is also gone. If users have a PDA and a cell phone, on the other hand, they could still use their PDA to look up a phone number even if their cell phone becomes inoperable.
    - Strengths of smartphones:
        1. Smartphones connect to a cellular network, just like a cell phone. With a wireless data plan, smartphones can surf the Internet from anywhere a cellular

signal is available (though speeds vary). PDAs do not connect to cellular networks and are thus unable to provide the same range of connectivity to the Internet.

2. All users' data in one portable device. If users didn't have the phone number they wanted in the phone, they had to put down the phone, find it in the PDA, look up the number, put down the PDA, pick up the phone, and dial the number in this cell phone. However, users solve this problem thanks to smartphones, because they have both functionalities on them.
3. A smartphone can reduce what you carry to one compact package. As a phone, it can handle voice and data communications, including e-mail and text messaging. As a PDA, it boasts the processor and memory to run pocket versions of favorite applications and the ability to browse the Web.

Despite the fact that PDAs and smartphones have different options and applications, both often use the same, or very similar, operating systems. As a result, both types of devices can support third-party software programs that will increase the functionalities of these devices. Finally, we can conclude saying that a smartphone is a converged device that combines the functionalities of a PDA and a cell phone.

2.3 Evaluation criteria

As commented in the introduction section, smartphone market is very fragmented in platforms that are incompatible among them. In order to prioritize the development for a concrete platform, we have defined some evaluation criteria that will allow us to compare the different possibilities, "quantifying" in some manner the strengths and weakness of each one (we will assign 3 points if the evaluation is "very good", 2 points means "good", 1 point is "not too bad" and 0 points is "bad").

2.3.1 Users may install XtreemOS

To achieve maximum dissemination, it would be interesting that final users could install XtreemOS MD modules as any other application, at least in the short-term. This is the case with the basic version for PDAs and MIDs, but unfortunately, this is not possible in most telephones, which are locked by the mobile operator, even if the platform is open. This is the reason to consider mobile operators, manufactures and developers as our "direct target" (even if the indirect final target remain the end users), but that could delay a bit the dissemination, as in general there won't be any device with XtreemOS-MD embedded until some months after the software is released (except if the firmware is updatable).

2.3.2 Terminal availability

It's important that XtreemOS-MD may be tested in real devices. Mobile phone availability implies two conditions:

1. Continuity of a terminal model at software release time: Terminal models' lifetime is in general very short and it could be the case that a popular model during the design phase is finally discontinued and not available at the end of the project (at software release time). That shows clearly the importance of not developing code much dependant of a specific model, to allow its reutilization.
2. If final users cannot install XtreemOS-MD, it's interesting that at least any developer can buy an unlocked phone at a reasonable price. Otherwise XtreemOS-MD code only will be only available to operators and mobile manufactures. We think that it's positive that XtreemOS-MD gains a developers community, this is one of the main advantages of public projects with open source licenses.

This criterion is important mainly when considering developers community as the target. It's always possible to use an old telephone as a proof of concept for a mobile operator or manufacturer, but to gain a developers community that's not enough.

2.3.3 Platform openness

This is a general requirement of XtreemOS project and it remarks the importance of achieving a community around the project. This requirement does not imply that XtreemOS-MD will not be available in closed platforms, as XtreemOS-MD open source license allows it. Owners of closed platforms may use the reference implementation of XtreemOS-MD and port it to their platforms.

2.3.4 Allowance of native applications

Some smartphone platforms are based on Linux, but Linux is not visible to applications. Indeed these platforms may migrate in a future to other operative system without breaking its public API. These platforms (e.g. Android, WebOS) support applications written in languages as subsets of Java or Javascript, and are more intended to run RIA (*Rich Internet Applications*). That is, these platforms are not for running existing Linux applications, but to allow more advanced web applications with better integration.

If the platform does not allow running native applications, our reference applications could not be able to run in the smartphone (the needed work for porting the code will not be worthy). This problem is especially important for our instant messaging application, which serves as an example of how a traditional Linux application can be extended through the XtreemOS integration.

XtreemOS-MD offers integration of grid service at operative system level. If applications has no access to operative system and need an intermediate layer, it's unclear what advantages offer XtreemOS over other solutions based on middleware.

An interesting option could be the possibility of modifying smarphone OS to allow the parallel execution of native applications and smartphone applications. But this task is on one hand very complex (possible competition to gain access to the screen and input devices if the terminals are not using X-Window) and on the other hand it's also risky: the system base to modify is not part of the public specification, and therefore it could be terminal-specific and it could be changed in any moment as well.

2.3.5 Security model compatibility with XtreemOS philosophy

Some smartphone platforms allow native applications, but their security model prevents applications to access disk, or run each application with a different UID ("sandbox model"). This sandbox model it's not optimal for XtreemOS VO support, as if a user invokes two XtreemOS applications using the same credentials, these applications must share the access to XtreemFS, which should be mounted several times, once for each UID. This is significantly more problematic thinking on mobile devices because of their limitations.

Sandboxing model also conflicts with SSO philosophy of XtreemOS-MD. The use of a different certificate with each application is not user-friendly and it's not aligned with the SSO philosophy of XtreemOS-MD. It also introduces further problems: e.g. smartphone users cannot use JobMA application to monitor processes, because each one runs with a different identity. It would be possible to avoid these restrictions using a middleware, but in this case XtreemOS is not offering any advantage against other middleware solutions

A different solution will consist in the modification of the smartphone software to avoid sandboxing. But this is not a good idea, because it breaks the smartphone's security model, that applications have been designed to use.

A possible solution will consist in running signed XtreemOS aware applications with a different security model than generic applications, to avoid sandboxing for them.

2.3.6 Expected future of the platform

This criterion is about the probability that mobile operators will sell smartphones based in the evaluated platform. It's a strategic, long term criterion for project dissemination, while "Users may install XtreemOS" is a short term criterion.

The implementation of XtreemOS-MD over a distribution already adopted by mobile operators or manufactures is useful to achieve early XtreemOS-MD integration but it could also be a weakness for mobile operators or manufacturers that have adopted a different platform.

Some distributions are currently oriented only to developers and enthusiastic users. Of course these platforms are worse evaluated under this criterion that platforms that has gained the interest of mobile operators and manufacturers. But thinking in the "neutrality" of those base distribution, not directly adopted by most manufacturers or mobile operators but with a code base that may be ported and reused easily to any other platform, it could even become an advantage (as an example, some software is published for Debian instead of Red Hat or Novell for this reason).

2.3.7 Risk evaluation

Linux smartphone world is changing and there is not a favorite platform. Probably, many platforms will evolve quickly and suffer big internal changes to become more compatible among them, as no one has the needed size to compete with big smartphone actors.

XtreemOS-MD is implemented at base operative system, not using a high-level API. This is good to be adopted for other platforms thinking in its neutrality, but could be a problem when base systems are evolving quickly and manufacturers can change it without modifying the public API offered.

The risk associated to fast evolution of mature platforms and possible obsolescence of them exists, but the risks associated to a platform still under design and without compliant terminals available are still more important.

2.4 OpenMoko stack evaluation

OpenMoko is a project dedicated to deliver mobile phones with an open source software stack [6]. Openmoko uses a Linux core, with a graphical user environment built with the server X.Org, GTK+/Qt toolkits and windows management. It is based in the *OpenEmbedded* framework and the *opgk* package system.

OpenMoko was published in 2006, thanks to its founders First International Computer (FIC). There are two main phones, the first one was the Neo 1973 (GTA01) [7], followed by the current model, Neo FreeRunner (GTA02) [8], that is available for any user or developer.

2.4.1 Users may install XtreemOS

OpenMoko was born to allow users to have full control of the terminal, and so users are allowed to install any software.

Therefore OpenMoko fulfills this criterion as “very good”.

2.4.2 Terminal availability

FreeRunner, the current OpenMoko terminal, is available and it is fully unlocked: there is not distinction in OpenMoko between “smartphone for users” and “smartphone for developers”. The only caveat is that FreeRunner may become a bit obsolete in several months compared with more modern devices, and the future plans to launch a new terminal have been suspended. Other FreeRunner limitations: neither 3G support (although it provides wifi and Bluetooth) nor webcam.

OpenMoko fulfills this criterion as “good”.

2.4.3 Platform openness

OpenMoko is an open source project. Any developer can contribute code to project and any manufacturer or operator is authorized to use the software freely.

Therefore, OpenMoko fulfills this criterion as “very good”.

2.4.4 Allowance of native applications

Open Moko applications are Linux applications. Applications available in Angstrom distribution can also be compiled to run in OpenMoko.

Therefore, OpenMoko fulfills this criterion as “very good”.

2.4.5 Security model compatibility with XtreemOS philosophy

OpenMoko platform does not implement sandboxing and allows installation and running of applications under the same conditions than a classic PDA or a desktop Linux.

Therefore, OpenMoko fulfills this criterion as “very good”.

2.4.6 Expected future of the platform

It's not probable that OpenMoko is adopted directly by any mobile operator, but it's more probable that the most successful code showed by OpenMoko will be merged in other mobile platforms.

Open Moko is interesting as a neutral, reference implementation, with code that may be ported easily to any device because is based in *OpenEmbedded*. *OpenEmbedded* is a project with a developers’ community, quite similar to Debian in Linux world. OpenMoko it is also interesting as a project to experiment new ideas and as a proof of concept.

Therefore, OpenMoko fulfill this criterion as “bad”.

2.4.7 Risk evaluation

The main risk of OpenMoko platform is the uncertainty about its future, after canceling the design of the new smarpthone model (GT03), even if the rumors about a possible closing of the project have been recently denied [9].

Another risk is the existence of several OpenMoko distributions, including drastic changes from one to (e.g. from OM2007 to OM2008 platform moves from Gnome Mobile to Qt).

Risk evaluation conclusion about OpenMoko is “not too bad”.

2.5 Nokia Maemo 5 (Fremantle) evaluation

Maemo [10] is the Linux platform launched for Nokia for Mobile Internet Devices (MID). Nokia MID hardware is very similar to most modern smartphones. Despite to be Nokia devices, MIDs are not telephones, although they support VoIP. Next generation of Maemo hardware will include a 3G modem: current generation uses a Wifi connection or a Bluetooth connection with an external telephone to access Internet.

We will consider next generation of Nokia device as a smartphone, independently of it is really a phone that also use the modem to voice communication or it is a PDA that only use modem to data communications. As commented previously for us the main difference between PDAs and telephones are not the voice capabilities but the possibility of a subsidized commercialization of those devices

Next generation Nokia MIDs will be governed by a new software release, Maemo 5 (codename Fremantle). This software will be different from Maemo 4, the software already supported by XtreemOS-MD basic version. Indeed, Maemo 5 wont be available for Nokia N8x0 terminals.

2.5.1 Users may install XtreemOS

Maemo devices allow users to install any program with a single click. Applications installers run with root privileges. Users also may gain root account if they want, although is not necessary.

Therefore Maemo fulfills this criterion as “very good”.

2.5.2 Terminal availability

Current Maemo terminals support Maemo4; Maemo 5 terminal with 3G modem should be available before the end of the year, and in the meanwhile, an SDK is available to test the platform (also it is possible to use as hardware the Beagle Board [11]).

Nokia Maemo terminals are unlocked and available in online shops like Amazon.

Maemo fulfills this criterion as “not too bad”, as even if it’s not available yet, at expected release date of XtreemOS-MD advanced versions it will probably do.

2.5.3 Platform openness

Strictly speaking, Maemo is a Nokia platform, not an open platform. But Nokia has contributed under open source licenses most parts of Nokia platform (for example, Hildon is part of Intel Moblin). In practice, XtreemOS-MD may be fully implemented just using open source parts of Maemo platform.

Therefore, Maemo gains “not too bad”.

2.5.4 Allowance of native applications

One of the Maemo strengths is the possibility of running Linux desktop native applications, optionally adapted to device.

Therefore Maemo punctuation is “very good”.

2.5.5 Security model compatibility with XtreemOS philosophy

Maemo platform does not implement sandboxing and it allows the installation and execution of applications under the same conditions than a classic PDA or a Linux desktop version.

Therefore, Maemo fulfills this criterion as “very good”.

2.5.6 Expected future of the platform

New Maemo devices are not strictly phones [12], even if they will include a 3G modem: the modem is intended to data communication, not to voice functions. Although Apple iPhone is redesigning the concept of phones, Maemo MIDs are a little bigger than smartphones and most current Maemo users prefers to have a Maemo device and a slim telephone with Bluetooth support.

But probably, with the inclusion of 3G modem, mobile operators will sell these Nokia devices (subsidized with a data plan) and some users will use it as smartphone.

It is also possible that Nokia will launch a true smartphone based on Maemo instead of Symbian. Nokia and Intel has announced *Ofono* project [13] an open source telephony solution but it’s not yet decided if *Ofono* will be included in Fremantle [14].

Therefore, Maemo fulfills this criterion as “not too bad”.

2.5.7 Risk evaluation

Nokia implementations of Maemo are stable and very well documented. But it is a real risk the lack of a release date for Fremantle terminal.

Therefore, Fremantle scoring in this criterion is “not too bad”.

2.6 Android evaluation

Android is a software platform and an operating system for mobile devices based on the Linux kernel. It was initially developed by Google and later adopted by the *Open Handset Alliance* (OHA), a consortium formed by 48 hardware, software and telecommunications companies, to promote open standards for mobile devices.

Android allows developers to write code thanks to *Dalvik* (a Java-like programming language) and control devices through libraries. Dalvik virtual machine is optimized for mobile devices. Android includes an integrated browser, based on WebKit, the open source engine. It also provides a touch screen, optimized graphics with a 2D graphics library, 3D graphics based on OpenGL ES 1.0 specification and SQLite for structured data storage. Android supports various media formats for audio, video and image, such as MPEG4, H.264, MP3, JPG, PNG, etc., and it provides some hardware-dependent features, such as GSM Telephony, Bluetooth, EDGE, 3G, WiFi, Camera, GPS, compass, and accelerometer.

Android's architecture is formed by:

- **Applications**: including an email client, calendar, maps, contacts, etc. and every application developed under Dalvik programming language.
- **Application frameworks**: developers have full access to the same framework APIs used by applications base. The architecture is designed to simplify the reuse of components and any

application can publish its capabilities and any other application can then use these capabilities.

- **Libraries**: a set of libraries C/C++ used by several components of the Android system. These capabilities are exposed to developers through the Android application frameworks.
- **Android runtime**: every Android application runs its own process, with its own instance of the Dalvik virtual machine, which has been written to run multiple virtual machines in an efficient manner. Virtual machine is based on registers, and runs classes compiled by the Java compiler that have been transformed to .dex format. Those .dex files are executed by Dalvik and are optimized for minimum memory.
- **Linux kernel:** acts as an abstraction layer between the hardware and the rest of the software stack. Android relies on a Linux version 2.6 for the basic system services such as security, memory management, process management, network stack and driver's model**.**

2.6.1 Users may install XtreemOS

Users are not allowed to install applications that requires root privileges to run (Google considers that users must not gain root privileges, but the first update to G1 phones was to remove a method that allowed to open a root shell [15], but XtreemOS-MD installation requires root permissions (e.g. to install FUSE module).

Therefore, Android fulfills this criterion as "bad".

2.6.2 Terminal availability

There are several Android smarphones already in the market or at least planned to be released in next months. Unfortunately, terminals sold by mobile operators to end-users are locked: *bootloader* only allows starting signed firmware. There is a special Android smartphone for developers available from Google, the ADP1 (according to Google, "*Android Dev Phone 1 devices are not intended for non-developer end-users*" [16]). This is a weakness comparing to unlocked consumer-smartphones, as these last ones allows enthusiast end-users to "become" developers.

It's also possible to install Android, with some limitations, on FreeRunner terminal [17], on Nokia Maemo devices [18] and also on BeagleBoard [19].

Android matches "good" in this criterion.

2.6.3 Platform openness

Android platform is open source, under a non-restrictive license. Software is under the umbrella of Open Handset Alliance but everyone has access to the source code and may modify it. It's not required to be member of OHA to contribute.

Therefore, Android gains "very good" in this criterion.

2.6.4 Allowance of native applications

Android does not support the execution of native applications. Android third party applications are signed packages, compiled with bytecode to run in Dalvik VM, although application may include also a native library to be invoked using JNI. Usually, it is not a valid option to run a native application to convert it to native library and use a Dalvik program as launcher: it's not possible to install share libraries, system C library is limited and there is not X-Window server.

SAGA API is not available for Dalvik applications. C++ version is not available and Java version is not compatible with Dalvik.

Therefore, Android is considered “bad” under this criterion.

2.6.5 Security model compatibility with XtreemOS philosophy

Android’s third party installable applications run each one with a different UID. Only applications signed with the same certificate can run with the same UID as referred in [20].

This is a big limitation and then, Android’s scoring in this criterion is “bad”.

2.6.6 Expected future of the platform

Android is already adopted by mobile operators like Telefónica, T Mobile and Vodafone.

Therefore its punctuation is “very good”. Anyway it’s a big challenge to find a killer feature that XtreemOS-MD may contribute to Android and it's not available with middleware Grid implementations XtreemOS integration with the operative system is a big strength, but unfortunately it is not useful in platforms as Android where applications are fully OS independent.

2.6.7 Risk evaluation

XtreemOS integration requires the modification of the base system, but Android API is at higher level. As commented previously, this may imply a change in Android’s base system without affecting the public API, loosing the compatibility with XtreemOS-MD, which porting effort is on the other hand difficult to estimate, as Android implements less features than ordinary Linux systems (for example, *libc* library may cause compatibility problems).

Therefore, Android score in this criterion is “not too bad”.

2.7 Other options

Other important platform that it is supported by several manufactures and mobile operators is LIMO. It's an interesting platform, because it allows Linux native applications and there are several terminals available in the market [21].

The big problem with LIMO is that it is not an open platform. Source code is under FPL (*Foundation Public License*) that is not an Open Source License: just the LIMO core members have the rights to commercially distribute under this license and non-members have no access to the source code [22], [23], [24], [25]. Therefore, in this Work Package only Telefónica I+D may access the source code, but the modified code cannot be released to public nor shared with other XtreemOS members. Of course, LIMO members may port XtreemOS-MD implementation to LIMO platform under the auspicious of LIMO foundation, because XtreemOS open source licenses allow it.

Almost all new Linux smartphones use Android or LIMO. Main exception is Palm Pre, which has raised great expectation as a possible “IPhone killer”. Palm Pre operative system is WebOS [26]. But WebOS is not an open platform (Palm is the owner).

Note that the availability of SDKs for smartphone developers it's not enough to port XtreemOS. These SDKs are for developing *applications*, but access to the source code and the possibility to update the smartphone firmware is also required for us.

2.8 Selected Solutions

We have decided not to select only one platform and exclude the others, but prioritize the development. We only guarantee the full development of one platform, but some of the others may be supported at “best effort”.

We have assigned different scorings to each platform for each criterion. Every platform receives 3 points if the evaluation is “very good”, 2 points means “good”, 1 point is “not too bad” and 0 points is “bad”. We think that “openness” is a very important criterion, and therefore their points are doubled. Risk evaluation is also important, but the three platforms have the same poor score in this criterion.

	users may install XtreemOS	***terminal availability***	***openness***	***native applications***	***security model is compatible***	***future***	***risk evaluation***	***TOTAL***
OpenMoko	3	2	6	3	3	0	1	**18**
Fremantle (Nokia)	3	1	2	3	3	2	1	**15**
Android	0	2	6	0	1	3	1	**13**

OpenMoko is the platform with the highest priority. Fremantle has priority over Android, but this could change if Fremantle terminals are not available in next months. Also, if Fremantle terminals are finally available soon, they could be prioritized against OpenMoko.

Even if Android would finally be supported, it would be just a partial: e.g. SAGA won't be available, reference applications cannot run in Android because this platform does not run native code

3 General Architecture

The architecture for the advanced version of XtreemOS-MD Foundation Layer remains basically the same already designed for the basic version, described in D2.3.3 [3], where the differences with the XtreemOS standard node (PC) where remarked. In this deliverable, as shown in Figure 1, Figure 2 and Figure 3, we are just showing the integration of the different smartphone platforms and the three kinds of components of XtreemOS-MD:

- **Base Linux components** are the components that come with the original Linux distribution for mobile devices, covering from a standard kernel, libraries, graphic environment to personal information management (PIM) applications (shown in purple in the figures)
- **XtreemOS-specific components** are additional components that will be included in the XtreemOS-MD distribution. In the diagram, the modules covered by WP2.3 are labeled as "XtreemOS-F components", and provide the basic infrastructure for constructing mobility and grid services. The rest of the XtreemOS-MD components (covering the grid functionality itself) are being developed in WP3.6 ("XtreemOS-F components" in light orange and "XtreemOS-G components" in dark orange in the figures).
- **Applications.** These components will not be included in the XtreemOS-MD distribution, and manufactures, application developers or end users will have to provide them. This includes both grid-aware and traditional (grid-unaware) applications (light grey for "Traditional applications" and dark grey for "Grid-aware applications" in the figures).

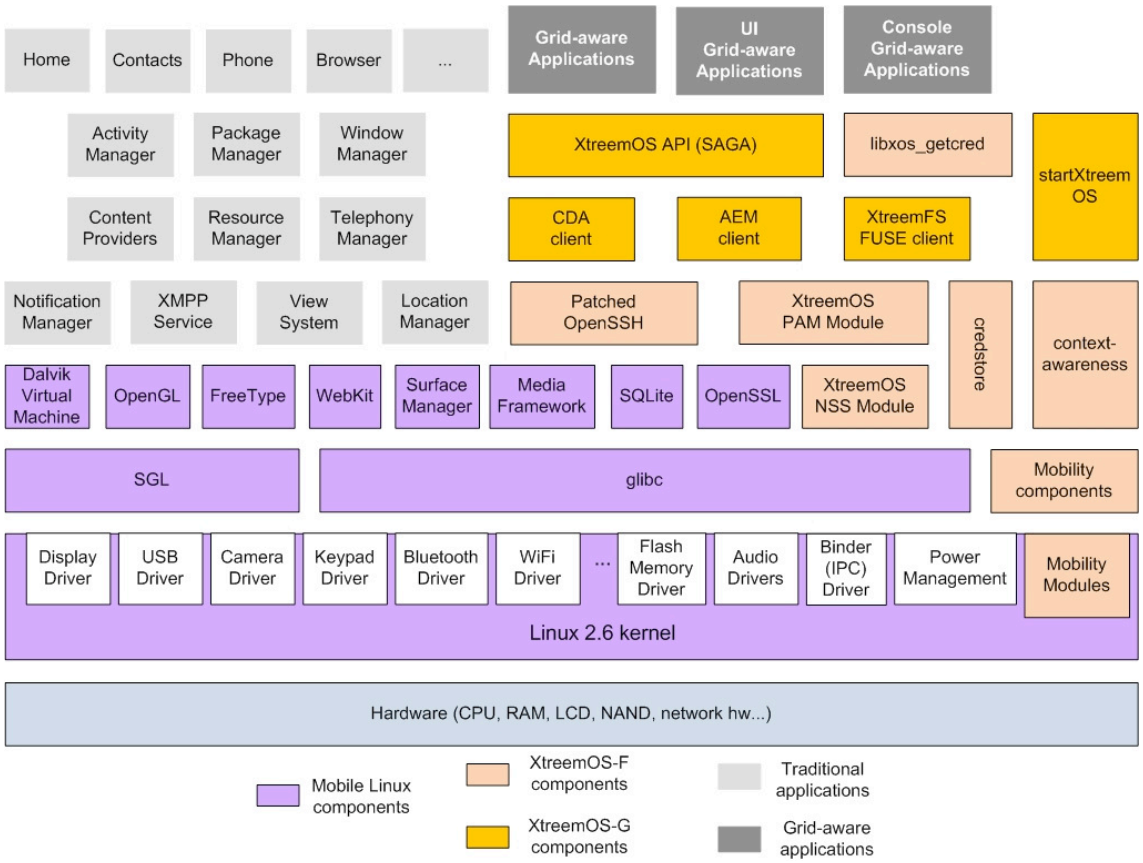

Figure 1. Android architecture

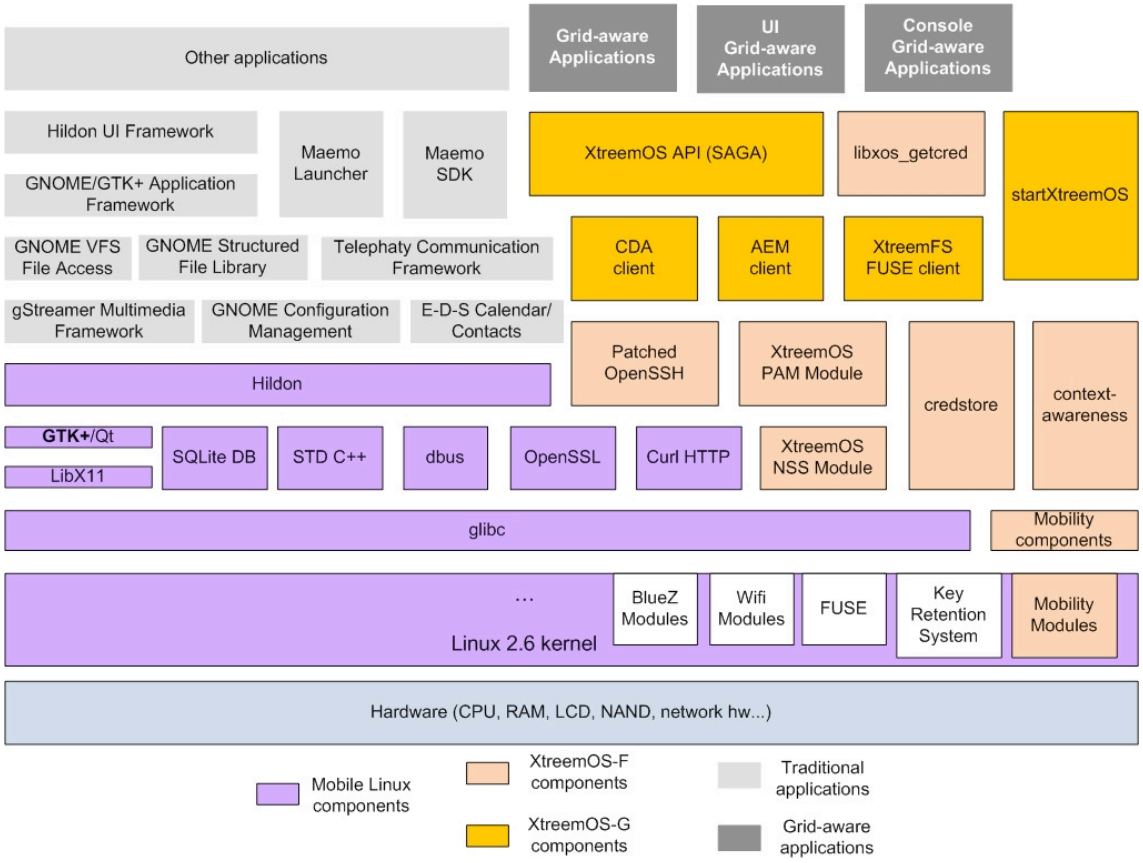

Figure 2. Maemo architecture

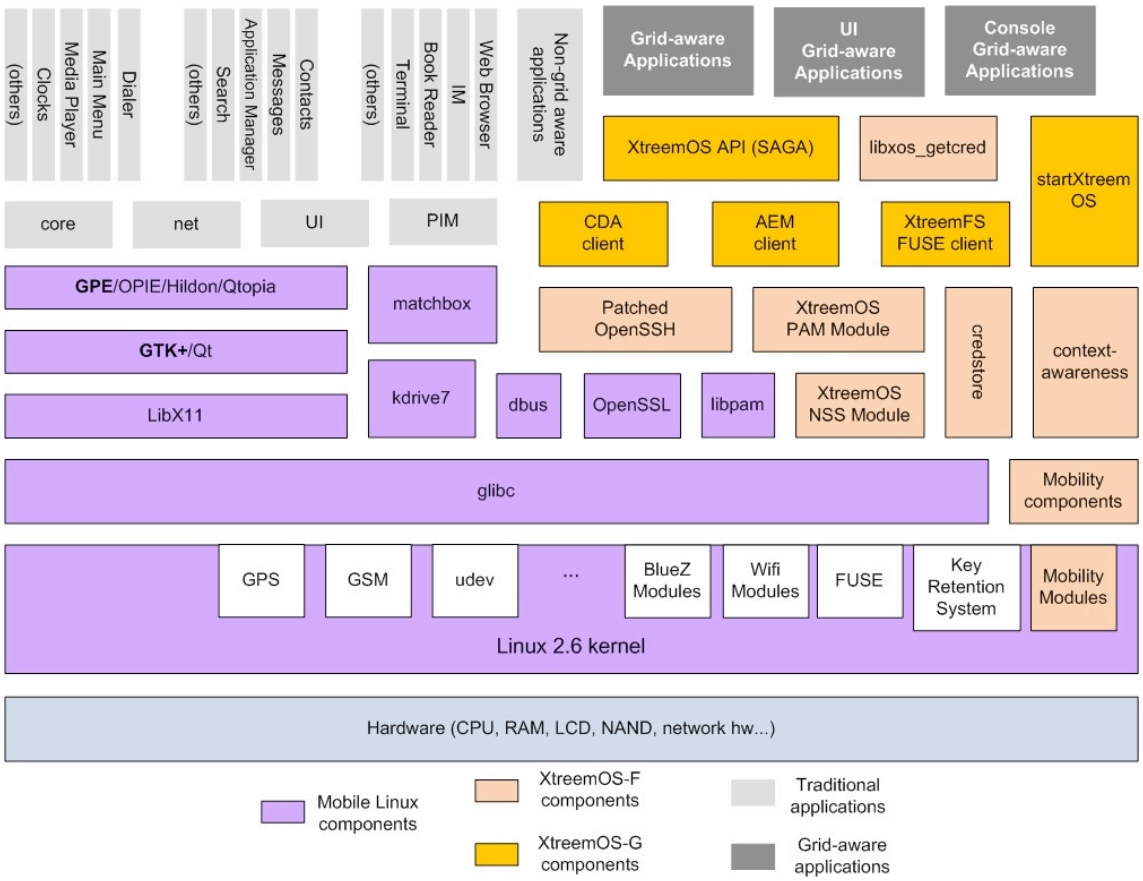

Figure 3. Open Moko architecture

3.1 Base Linux components

- **Linux 2.6 Kernel + Kernel Mobility Modules**: XtreemOS-MD supports different base distributions; common base of all them is Linux kernel 2.6 but with different minor version number, each one including a different set of modules. XtreemOS-MD requires mobility IPv6 compatibility, so we have recompiled the distributions not including the correspondent modules by default, in order to support it.
- **libC** is the standard POSIX library for input/output, memory management, string manipulation, and the like. Most C programs in Linux are compiled and linked with them, being *glibc* the most widely used. The only analyzed platform that does not include *glibc* is Android, that it is based in a lighter, featureless, implementation of *libc*.
- **D-Bus:** D-Bus is a message bus system, used for applications to talk to one another. In addition to inter-process communication, D-Bus helps coordinating process lifecycle, and easies the coding of "single instance" applications or daemons, and to launch applications and daemons on demand when their services are needed. Linux platform also uses D-Bus to notify events to applications about hardware, network availability, etc. In XtreemOS-MD advanced versions of D-Bus will be used, when available, to get context information useful and also to offer *libxos_getcred* API and to allow the implementation of *creduiagent* modules.
- **PAM (libpam)**: The Pluggable Authentication Modules (PAM) is a suite of shared libraries that enable the local system administrator to choose how applications authenticate users. It is used by XtreemOS to support virtual organizations natively in Linux. Anyway, as PAM modules are just required for Grid nodes, they are not mandatory for XtreemOS-MD, as we can consider that it will operate just as client-only mode, as will be explained later in the document.
- **OpenSSL** is an open source implementation of the SSL and TLS protocols, implementing basic cryptographic and utility functions. OpenSSH is strongly dependent on this library, which includes all the cryptographic functions used by SSH. There are libraries lighter than OpenSSL, but only a few are API compatible, and those have more restrictive licenses.
- **Linux User Interface** for GUI applications:
    - **libX11:** X11 libraries let the user build applications with a standard graphical interface, providing a basic framework, compatible with every other X11 system. Most Linux applications with a graphic user interface are X11 based, but normally using a toolkit as GTK+ or Qt. Some embedded single-application devices use directly the *framebuffer* instead X11.
- **GTK+** is a multi-platform toolkit for creating graphical user interfaces. Even if XtreemOS-MD reference applications are written using GTK+, XtreemOS-MD core is independent of any GUI library. In fact, the concept of *creduiagent* modules was created to make *credagent* modules GUI-agnostic.

3.2 XtreemOS-specific components

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

3.2.1 XtreemOS-F layer

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

Components for VO support in Linux OS

- **XtreemOS NSS Module**: This name service module adds virtual organizations support to basic Linux tools that show information about UIDs.
- **XtreemOS PAM Module**: This is a Pluggable Authentication Module that implements most of the low-level security and session aspects of Virtual Organizations. As commented in previous section, this module is not required in a client-only device
- **Patched OpenSSH**: This is a modified version of OpenSSH. Patches allow starting a remote session using VO security and user mappings and are also used in the new SSO implementation.
- **Startxtreemos**: utility to start a XtreemOS-MD client session, manually or indirectly when an application that uses *libxos-getcred* is started. *Startxtreemos* gets the credential from CDA, stores it in *credstore* using *libcredstore*, mounts XtreemFS and configures AEM client.
- **libxos_getcred**: library that can be used by the applications to read the credential, available through *credstore*. If the credential is not available in *credstore*, *startxtreemos* is invoked automatically.
- **libxos_credagent**: library used by *startxtreemos* to get the credential. The functionality is implemented using *credagent* and *creduiagent* modules.
- **Context-awareness**: library that provides useful context information (like location, remaining power, etc.) to other layers and applications.

Components for enhanced mobility of the device

- **Kernel Mobility Modules**: XtreemOS needs some special kernel modules as a complement to the mobility enhancements (see below), to give terminal mobility at the system level. Although this code is part of the official kernel, many distributions don't compile it by default.
- **XtreemOS-F Mobility Component**: includes the daemons and libraries necessaries for providing Mobile IPv6 (MIPv6) support, thus giving terminal mobility to applications and services, in a transparent way.

3.2.2 XtreemOS-G layer

The Grid layer of XtreemOS is composed of the grid services themselves, which enable users to execute jobs in other machines, access remote data, etc. XtreemOS-MD advanced version layer G is being handled by WP3.6 and the components included are:

- **Security Client**: This component includes the communication tools necessary for the mobile device to talk to the VO Management and Security services.
- **AEM Client**: This component provides client access to XtreemOS Application Execution Management system.
- **XtreemOS FUSE client**: This component provides client access to the XtreemFS filesystem, using a FUSE module. FUSE is a Linux specific technology that allows implementing file system modules in user space instead of in kernel space.
- **XtreemOS API**: This component will offer a subset of the XtreemOS Grid API (based on SAGA) to applications running on the mobile device (see [27], figure 1 and section 2.2, for further details)

3.2.3 Applications

We are not focusing on this level, as this document is mainly focused on layer F. Anyway, D2.3.3 [3] can be used as reference, as there where already some explanations in the architecture of XtreemOS-MD basic version.

4 Installation, configuration and usability enhancements

One of the main objectives of XtreemOS is making the grid easy to use. Moreover, applications for telephones are in general easy to install and configure as phone users are not necessary technically skilled, so for XtreemOS-MD is even more important this idea of use simplicity.

4.1 Installation enhancements

XtreemOS-MD basic version is very easy to install: users just need to fill the username field. Even, there is a possible enhancement: avoid requesting the username if it's possible to guess it by other means: that is, a full unattended installation;

- XtreemOS-MD advanced installer will search in the mobile device if there is an account (IM, email) belonging to a specified domain: if it is present, the installer will extract the username from that account. This method is useful with enterprises that implement SSO with the same username in every internal server and also in a CDA server/proxy. This is also useful with mobile operators that provide a user account.
- XtreemOS-MD may be installed as a dependency of an application, for example XOS-Pidgin (the IMA application). It's possible that this application knows the username to use and pass it to the installer.

XtreemOS-MD advanced server will allow better granularity in installation: it would be possible to install it without AEM or without XtreemFS if only one of these two components is needed.

4.2 Configuration enhancements (AS2.3.7)

One of the main philosophy principles of XtreemOS-MD basic version was the full automatic configuration, so that the users didn’t need to configure anything: administrators or mobile operators would set the needed configuration in a single file that would be included in the installation. This configuration file is currently not editable by the users, unless gaining root privileges and using a text editor to modify it.

The XtreemOS-MD advanced version will allow advanced users to change the configuration by using a program with an intuitive GTK+ interface. Anyway, the configuration file will optionally limit what users are allowed to change, as there are some cases where the old behavior (preserved if root is selected as the only authorized user to edit the configuration file) may be interesting:

- the need of root privileges to edit configuration files may be considered a security feature as it avoids that malicious applications could alter the configuration
- It could be useful for implementing parental controls
- for companies which don't want a modification of the configuration done by the users

XtreemOS-MD advanced version will support multiple configurations, allowing users to choose which is the default one. Each configuration includes data like CDA IP adrresses, XtreemFS server IP, Xatica configuration etc., as shown below in a real example of one of this configurations

```
[general]
credagent=xos_credagent_cdaclient
creduiagent=xos_creduiagent_gtk
```

```
[credagent]
cda_host=192.168.1.141
cda_port=6730
cda_vo=TID
cda_user=testinguser
ask_confirm=true
use_ssl_proxy=false
use_proxy=false
cache_file=/etc/xos/creds/cached-credential.pem
fsuidisuid=true

[creduiagent]
autokill_after=50
grabserver=true

[xtreemfs]
xtreemfs_server=192.168.1.141
xtreemfs_mountpoint=~/xtreemfs

[xatica]
xatica_remoteserver=192.168.1.141
xatica_remoteport=55000
xatica_bindaddress=0.0.0.0
xatica_localport=10000
xatica_showmyipserver=192.168.1.141
xatica_showmyipport=8081
```

One advantage of XtreemOS-MD is that users don't need to get the credentials manually from CDA; the software does all the work automatically. But there are two points that advanced version will enhance:

1. Automatic renewal of expired credentials: *startxtreemos* will set a timeout in *credstore* to disable expired credentials. This implies auto–renewal of the credential, because *startxtreemos* will be invoked if *credstore* is empty and a new credential will be obtained from the CDA server.
2. A new function *getnewcred* in *libxos_getcred* library. This call will force removal of cached credentials and will request a new one. This is useful when applications detect that the current available credential is not valid anymore for some reason (e.g. credential was revoked). This implies also a new function in *libxos_credagent* library and in *credagent* modules.

XtreemOS-MD advanced version will support also the configuration of multiple volumes to be mounted when starting the connection to the Grid (it was just one volume in the basic version).

4.3 User interface enhancements

Mobile applications must be adapted to support a limited interface: most terminals don't include a functional keyboard.

One enhancement of XtreemOS-MD advanced is related to password typing:

- Mobile telephones usually request short numeric PINs instead of long alphanumeric passwords. The security of PINs is based on setting a limited number of typing errors. For example, after three failed tries, the PIN is blocked and the user needs to introduce the PUK (longer and then more secure). Again, in case of failing many times when typing the PUK, the terminal will be locked.
- XtreemOS-MD will provide a PIN mechanism, but instead of a numeric PUK, the PUK will be the full pass phrase, with no limit in the number of tries. The authentication process is unmodified, except that the user will have a configurable number of opportunities to introduce

a short PIN instead of the full pass phrase used normally. *CDAProxy* and *libcredagent* will be the responsible for this mechanism implementation.

- XtreemOS-MD will provide also a new mechanism to authenticate against the CDAProxy without typing a password, but using Bluetooth-pairing technology.

4.4 Offline mode (AS2.3.4)

XtreemOS-MD will support to work in offline-mode. This mode, just available for filesystem operations, will be implemented mainly in layer G, as part of the XtreemFS enhancements. F-layer will just provide a user interface for switching to offline mode and the following API:

int set_offline_mode(char mode);
Mode parameter is 1 to set "offline mode", 0 to return to "online" mode. The function returns 0 if success, -1 otherwise.

int get_offline_mode();
Function returns 1 if in online mode; otherwise it returns 0.

On the other hand, layer F will also provide information about battery and network availability, useful to automatically change to offline mode; as part of the "context awareness API".

4.5 Service resuming (RMD2.3.10)

One problem with mobile devices is related to network instability, which could require the restarting of some XtreemOS daemons (like for instance the SSO daemon) which needs a network connection. XtreemOS-MD advanced version will provide a launcher wrapper for checking unexpected program deaths, automatically restarting them. This wrapper will use timeouts and counters to avoid launching continuously a program with permanent errors. If the underlying platform provides asynchronous information about network availability (via D-BUS API), the wrapper will use this information to wait until network connectivity is restored.

Sometimes, services may die abruptly because of memory problems: the wrapper will also test this condition before restarting the services.

4.6 On-demand starting enhancements

There are some reasons that recommend not starting XtreemOS-MD until users really need it, like:

- limited resources of MDs compared to PCs
- MD users are more sensible to the starting time
- security reasons

Users may start the software manually running *startxtreemos*. *Startxtreemos* makes available the credentials to applications through *credstore*, getting it from the CDA server if needed, mounting XtreemFS and autoconfiguring AEM. *Startxtreemos* is a console application, but the advanced version will provide a GTK+ version that will offer to the user a list of the available configurations.

One of the strengths of XtreemOS-MD is that it provides auto-starting through *libxos_getcred* library: *startxtreemos* is launched automatically when an application linked with *libxos_getcred* is launched. *Startxtreemos* is only launched once per session and the credential is stored in a *credstore*.

Applications may use transparently *libxos_getcred* without accessing the source code, just re-linking the application with *libxos_wrapopen* (that overwrites some *glibc* functions).

It's also possible to use *libxos_wrapopen* without re-linking through LD_PRELOAD mechanism, useful when only final executable is available or software is not C/C++ code (e.g. Java or Python code), but this method has some limitations:

1. It's unavailable to SUID and SGID applications for security reasons.
2. It's necessary to define the variable before starting desktop session (applications are launched using and environment that cannot be modified after starting session).
3. Parallel effects: setting LD_PRELOAD implies that open system calls are overwritten for **every** application, including applications not related to XtreemOS.

XtreemOS-MD advanced version will provide an alternative to overwriting open system calls, through a FUSE filesystem. A pseudo filesystem using *libxos_getcred* will provide four virtual files to read the credential, certificate, key and configuration name..

XtreemOS-MD advanced version will also provide a wrapper to *libxos_getcred* through a D-BUS service, so that *libxos_getcred* will also be available for programs written in non-C languages as Python or Java. An advantage over FUSE filesystem, is that a running daemon is not required: D-BUS services are auto-launched when a ".service" file is installed.

One application that uses *libxos_getcred* is *xsub*, the utility to launch applications in the Grid through AEM: this implies auto-starting when users demand the execution of applications in the Grid. A similar interesting feature that advanced version will provide is that grid software will be started automatically when an application try to read a file available in an unmounted XtreemFS (auto-mounting software, that will be implemented by layer G, will invoke *libxos_getcred*).

4.7 Transparent access to grid resources

XtreemOS provides transparent access to files resources through XtreemFS FUSE module. However, the execution of processes in the Grid is not transparent: a launcher as *xsub* or the JobMA application is needed at this moment, and even if it's proposed to include a new shell to have the possibility of launching jobs directly, just typing the name of the processes, it's not really an "OS feature" and it's not directly applicable in the MD world, where opening a shell is not the usual "way of working".

The Linux kernel provides a mechanism (*binfmt_misc*) that allows the execution of code that needs a special interpreter or launcher as native binaries (e.g. this mechanism allows running directly Java .class files). This kernel module allows two new features about launching processes in the Grid:

1. Run JSDL files
2. Run i386 executables in XtreemFS volumes in the Grid (local processor only runs ARM binaries). When launching */usr/bin/progname*, the wrapper will also read, if available, the file */usr/bin/progname.jsdl*, otherwise it will generate one automatically.

Unfortunately, *binfmt_misc* kernel module is not compiled by default and many mobile devices don't include it, but there are some alternatives:

- Associate a launcher to JDSL MIME type. Even if this is not a universal solution, it's available with web browsers and with some file managers for example.
- Use a *#!/bin/launcher* header in the jsdl file. But of course, this requires the modification of utilities like *xsub* or JobMA to ignore this header, not valid in a XML document.

5 VO support advanced features

XtreemOS-MD F-layer basic version provides a rich security infrastructure to support access to grid securely. XtreemOS-MD implements a modular, pluggable SSO solution, which allows users forgetting about credential administration.

Security in mobile devices like smartphones is very important, because there are personal data involved. Security in mobile devices is a challenge, because many telephone users have not technical skills and don't understand security concepts like certificates and so on. KISS security principle (*Keep it Short and Simple*) is enforced when possible: unfortunately, long passwords and keys are securer than shorter ones, but they are more difficult to use in embedded devices like smartphones.

5.1 VO Management from MDs

XtreemOS-MD basic version allows VO Management through VOLife web interface. Advanced version will support the new, redesigned, VOLife web interface (splitted into 2 different interfaces, VOFrontEnd and RCAFrontEnd) and will provide as well an alternative interface for joining a VO, optimized for the MD interface. Joining a VO involves the following steps:

1. Users create a new account in VOLife
2. They log in using the new account, select an existing VO and apply for inclusion in that VO.
3. The administrator must login periodically in VOLife, check if there are new applications and approve or deny each one.

The periodical check in VOLife done by the administrator is unproductive and introduces a delay in approval. A better option for the administrator would be receiving in the mobile device a notification for each request to join a VO, allowing him to approve or deny it immediately.

It would also be convenient for users to apply for VO joining through an instant messaging application, instead of needing less comfortable procedures involving a web interface and additional steps like creating an account. XtreemOS-MD solution for this will be implemented using a Jabber application. Let's see an example of the process:

1. John will request inclusion in the buddy list of a virtual user *vo_example*, representing the VO that John wants to join to.
2. *vo_example* administrator will receive the notification in his IM program; if he accepts John in the buddy list, John will be included in the VO.

Some organizations may prefer an automatic approval of joining request of their authenticated users, instead of a direct default inclusion of every user in a VO, as some users will probably not make use of the Grid and the membership could even have an associated cost. This operation will be implemented by the *CDAProxy*.

5.2 Security enhancements

5.2.1 New credagent and creduiagent modules

Basic version of XtreemOS provides a modular architecture to get the credentials, using two types of modules: *credagent* modules, to obtain the credential, and *creduiagent* modules, that allows *credagent*

to interact with the user. XtreemOS-MD basic version provides a *credagent* module that uses *libcdaclient* and a *creduiagent* module that uses GTK+ API to interact with local user.

Advanced version of XtreemOS-MD will include new *credagent* and *creduiagent* modules. These new modules are intended to use external remote methods, as for example, to request the authorization using an instant messaging application. This provides several advantages:

1. Better integration with legacy SSO solutions. For example, a company may use a web based SSO or support authentication methods not available in the mobile device (e.g. a smartcard).
2. Security: as when a window dialog requests a password, the user is not really sure if the request comes from the *creduiagent* module or from some "malware".
3. Flexibility: user's PC could be used for obtaining the credential when starting the session, etc.
4. Delegation and hierarchical access control: it's possible to delegate the authorization in other user (e.g. a security administrator) or machine. This also allows parental control (e.g. before a child get access to a VO, his parents must approve it).
5. Support to new devices: although XtreemOS-MD is intended mainly to personal mobile devices as PDAs, MIDs or smarpthones, there are other embedded devices running Linux as NAS devices or set-top boxes that have very limited user-interface capabilities. For example, many multimedia devices using SIGMA chipset don't allow ordinary applications screen access.
6. Enhanced user experience. Some users will prefer to receive notifications by Instant Messaging than by other kind of alerts. For example, user may have configured his messenger application to play a sound when a new message arrives and the PDA is inside his pocket.

This solution will be implemented by modules using *libcurl* for web interaction, and *libloudmouth* for XMPP interaction.

A new *creduiagent* using D-BUS will be provided as well. This module will allow the registry of a generic application to interact with the user.

5.2.2 libcredstore access control by group

Currently, when a user starts a Grid session, all his applications have access to the *credstore*. A interesting feature to enhance isolation is the implementation of a *credstore*'s access restriction to applications running as members of a special group. Only the applications with SGID bit active and with that special group as "group owner" will have access to *credstore*. Another option is to use a launcher with SGID bit active or SUID root.

This solution is only available if using *uskeystore* module, not using the KRS module, as any application could bypass the mechanisms implemented in KRS invoking directly the API offered by the kernel.

5.2.3 SSO integration

XtreemOS-MD basic version already provides a pluggable and modular Single Sign On (SSO) solution. This solution doesn't cover the authentication phase; it's SSO in the KRS sense: a key is available to every application, not being necessary additional interaction with the user nor the CDA server, but applications must implement authentication using the key themselves. WP3.5 is defining at the time of writing this document a SSO implementation that covers the authentication phase and implements other features as delegation. Both implementations are complementary, because each one of them covers different phases.

XtreemOS-MD advanced version will provide integration between XtreemOS-MD plug-able, modular SSO implementation and the new XtreemOS SSO design, implemented using a modified *OpenSSH*, which will be ported to XtreemOS-MD.

5.2.4 Support for Crypto Accelerators (R2.3.25)

Some embedded devices provide crypto accelerators by hardware, to surpass limitations of its low-power CPU. Sometimes this hardware support is used transparently by *OpenSSL*, but in others a special module known as "engine" must be loaded. Engines modules are automatically loaded if added to */lib/engines* but a more flexible solution is based on the modification of the configuration file */etc/ssl/openssl.cnf* . (see [28]).

Using a crypto accelerator is transparent for applications if using openSSL 0.9.8. All XtreemOS-MD applications that require cryptographic features use openSSL 0.9.8.

Engines accelerate only a subset of cryptographic algorithms: so, it's convenient that applications use these algorithms with higher priority than algorithms without hardware acceleration support. The *credagent* module implemented will read a configuration file with the preferred list of algorithm to be used for SSL communication. This feature is also useful to select between low or high security ciphers in case of hardware acceleration not available (e.g. *openssl ciphers LOW: MEDIUM: TLSv1*). See requirement RMD2.3.18.

5.3 Ad-hoc VO support

One interesting feature for mobile devices would be the possibility of creating ad-hoc VOs. A possible scenario could be the creation of a ad-hoc VO for the people attending a meeting, in order to use the PDAs to access to the Internet through a PC connected with ADSL. Each mobile device owner will join the VO and the PC will share resources like the network connection, file space and the possibility to run processes.

This topic is currently under analysis and evaluation in order to estimate the resources needed for implementing it.

6 Context-awareness

One of the advanced features considered for XtreemOS-MD is the inclusion of context information. In mobile devices there are several considerations when using grid services, as differently from other Grid nodes or clients, the mobile devices could change their location, could run out of battery or even could alternate between different network accesses (like 3G, Wifi, etc.).

The context information is useful at higher levels (G-layer services and specially applications), but at F-layer we are going to provide an API to allow them retrieving this context. The main information provided by this API could be separated in:

- Geographical location: based on GPS coordinates or CellID information
- Remaining power: level of battery
- Network access used: to identify if the mobile device is connected using 3G, Wifi, GPRS, etc.
- Additional information: like operation mode (silence, meeting, etc.), etc.

6.1 Geographical location

The API will provide the coordinates relative to the mobile device location, depending on the kind of location information offered, such as GPS coordinates or CellID information:.

1. If location is based on GPS, it can be expressed as UTM coordinates or as geographical coordinates:

```
getLocationUTM(){
        double x;
        double y;
        int spindle;
        string hemisphere;
}

getLocationGeographical(){
        int grad_lat;
        int min_lat;
        float sec_lat;
        int grad_long;
        int min_long;
        float sec_long;
        char cardinal;
        float lat;
        float long;
}
```

2. Geographical location can also be based on CellID information

```
getLocationCellID() {
        int CellID;
        int ECellID;
        int hour_EOTD;
```

```
        int min_EOTD;
        int sec_EOTD;
        int hour_time;
        int min_time;
        int sec_time;
        int grad_angle;
        int min_angle;
        float sec_angle;
    }
```

6.2 Remaining power

The level of battery, the remaining power, is a very important "context information" related to the mobile device, taking into account the big limitations of the current batteries and the problems associated (not just for XtreemOS) to running out of battery. The information about remaining power could be used for example in applications that could decide a possible change to a disconnected status when running out of battery, in order not to waste if completely. For example, if the percentage of the battery is under 10%, the mobile device could automatically disconnect from the Grid.

```
getRemainingPower() {
    float remainingPercentage;
}
```

6.3 Network access used

The API will provide the type of access used, such as 3G, Wifi or GPRS, and also the current IP of the terminal. This information could be useful for example to detect a possible change of IP address, what could be problematic with the open connections (for example, for XtreemFS service)

```
getNetworkAccess() {
    string accessType;
    string IP;
}
```

6.4 Additional information

There are other context information handled by the terminal, but not all would be useful for XtreemOS. At this moment, we are just envisioning to offer the current terminal's operations mode (silence, normal, meeting…), but in the future, other context information could be provided by the API if necessary.

```
getOperationMode() {
    string mode;
}
```

7 Resource sharing

During the phase of identification of requirements and specifications for the XtreemOS-MD advanced version, it was commented several times that the basic version did not consider the possibility of sharing resources, and that would be one of the optional enhancements associated to the advanced version. This chapter is focused on this resource sharing, but we should first take into account that even if the basic idea when creating a node in the Grid is to add some shared resources, like storage or computing capacity, when thinking on mobile devices, this is not so evident because of their limitations: very little storage capacity (in general) and reduced computing capacity because of the use of not powerful processors. Then, other resources should be considered, like special hardware (cameras, GPS, etc.) and also we will analyze the possibility of considering people as resources, one of the requirements commented in D2.3.5 [1].

7.1 Special hardware

Taking into account the very limited storage and computing capacity of the mobile devices considered, the interest could be focused on concrete hardware available in those terminals, like the cameras, the GPS modules, etc.

Camera

In some cases it could be interesting to be able to access the camera of a device, for example a satellite camera, but thinking on the MDs we are tackling in XtreemOS (smartphones and PDAs), it's quite difficult to imagine a suitable scenario to take advantage of sharing the camera. Moreover, mobile device owners would be really reluctant to share their cameras, what for sure would be considered a privacy "violation".

GPS modules

As not every mobile device provides a GPS module, this could be one of the more interesting resources to share. The problem is that there are just 2 meaningful scenarios to take advantage of it, and both have several limitations or problems associated, apart from the inherent invasion of privacy:

- Use the shared GPS module of a user in the surrounding area to obtain the approximate own location. But this would be useful just if we can assure that the GPS module used is really close to us, for example if we are travelling together
- Locate the MD that is sharing the GPS: even if authorized (to avoid privacy issues), there are simpler ways to implement it, like sending the GPS coordinates in a message, by GPRS, etc.

Network access

In order to share the 3G access to the Internet for example. This case links directly with the scenario considered in the section related to ad-hoc VO support (section 5.3). A mobile device could share the 3G access with other terminals registered in the same VO. This could be useful if for example the other terminals are not provided with 3G access or don't have a flat rate, etc.

7.2 People as resources

One of the ideas related to resource sharing that was commented in the past, and that has been analyzed now, is the possibility of considering the mobile device users as "resources". Anyway, there are simpler ways, not based on XtreemOS, to implement this kind of resource sharing: for example,

with a phone call to request "resources" or even using the publish/subscribe mechanisms offered by XMPP protocol), so we have decided to abandon this idea and consider it out of the scope of XtreemOS project

7.3 Problems

7.3.1 Java software

XtreemOS node server is written in Java and requires JRE 1.6 or higher versions, but there are not open implementations of JRE 1.6 available for ARM devices (officially, *OpenJDK* has not been ported to ARM and GNU releases only implements partially the 1.6 API).

Indeed, the need of Java is a quite heavy requirement for a mobile device if it is not used also by other local applications. None of the analysed platforms uses Java, and even if Android uses a Java-like language, JRE1.6 applications are not supported)

Apart from the language issue, the software for nodes is designed to run over PCs, it's not optimized for embedded devices, and the needed porting would be a high resource consuming task, apart from the difficult version synchronization if the software keeps evolving during the porting,

7.3.2 RCA certificates

There are two problems related with the certificates, one from the "Grid perspective" and the other related to usability:

Firstly there is a big security risk if the Grid gives a machine certificate to a mobile device, as differently from other Grid nodes, mobile devices are managed by end-users, which implies several considerations:

- MD users are not necessarily familiarized with computers
- MD users are in general unconscious of the importance of security requirements
- Mobile devices are easily stolen or they can simply be manipulated by third users

Them, taking into account the interface limitations of the MDs, it would be very inconvenient for a MD user to mange a second certificate apart from the personal certificate. Even managing just one certificate is not too comfortable, so we are analyzing and implementing different solutions to facilitate the task, like the use of a PIN for authentication, etc.

7.3.3 Security concerns

The use of a mobile device as a Grid node has several security problems implicitly associated with the normal use of those terminals.

From the point of view of the MD owner, there are security problems (especially related with privacy) like:

- lack of privacy if sharing the camera
- possible access to personal information like contacts, agenda, personal messages, etc: even if considering the possibility of using virtualization in order to separate the resources shared and the private ones, there are some voices advising that virtualization doesn't grant the possibility of accessing to other data physically stored in the terminal
- monetary, privacy and even legal risks if a remote user "jailbreak" the smartphone and takes control of it

- MD users' lack of technical skills (as commented previously): this is not just a problem for the machine certificate issue, but also they could not be aware of what "resource sharing" is and its implications.

From the point of view of the Grid user, there are also some security concerns, especially related in this case with the lack of confidence on the MD owner (already commented in section 7.3.2):

- the MD owner (or a third MD user) could hack the terminal to give wrong results when executing applications
- the MD owner could physically remove data stored in the terminal
- lack of privacy as the MD owner could easily access to the data stored or even directly monitor or sniff every packet arriving to the terminal

7.3.4 Other concerns

Apart from the mentioned security and privacy issues, there are other concerns related to the mobile devices like:

- low battery life, making the MD owners reluctant to share the computing capacity of the terminal, that could quickly reduce the remaining power
- lack of availability of the terminal, as the owner can switch off the terminal whenever he wants to, even if there are processes running or pending operations
- frequent connectivity problems, specially when the MD owner is moving

7.4 Considerations

Taking into account the lack of a "killer use case" where the mobile device act as a Grid node[1], the security concerns exposed and the low value added by the possible hardware shared, we have decided to consider the mobile device just as a Grid client, and never a Grid node.
Also, for taking this decision, it has been considered the low battery life of the current mobile devices and the high effort needed to implement and port the software to use XtreemOS-MD in a node (see 7.3.1)

Probably, the most interesting feature not implemented if we limit to the MD-as-client scenario is the file sharing (R2.3.41), and specifically the on-demand file sharing (sharing local files not uploading them until another user requests them). But this feature should not be really considered as part of "resource sharing", as XtreemFS nodes are really sharing their disk space, not the local files. Therefore, XtreemOS-MD could provide a method to implement on-demand file sharing even if MD will be just a Grid client. Other problems related to the lack of a persistent connection between XtreemFS clients and servers would appear, but this issue will be deeper analyzed by WP3.6, in charge of the G-layer, where XtreemFS is one of the services offered.

[1] While mobile operators may be receptive to integrate grid client software to gain new features without compromising the security, they are very reluctant to introduce modifications that could imply some security risks if those additional features don't clearly improve the user experience

8 PC porting of the software (AS2.3.6 and RMD2.3.24)

In D2.3.5 [1], one of the specifications defined was related to making XtreemOS-MD available for x86 architectures (AS2.3.6). The main idea behind this specification was the possibility of using XtreemOS-MD in other kinds of mobile devices, like MIDs, UltraMobilePCs (commonly known as "netbooks"), etc., revealing the need of porting the software to PC (x86) architecture. Even if it will always be possible to use the XtreemOS PC flavour, the "user experience" is different. On the other hand, it could also be useful for testing purposes, in order to test XtreemOS-MD on "PC" devices.

We have analyzed the possibilities and we have decided to select Ubuntu as the PC GNU/Linux distribution to port XtreemOS-MD. These are the main advantages of Ubuntu:

1. Ubuntu distribution is available for PCs, but there is also a specific version for Moblin (Intel Mobile Internet Devices with PC compatible processor), Atom Netbooks (Ubuntu Netbook Remix) and ARM Netbooks [29], [30], [31].
2. Ubuntu uses DEB packages: Nokia Maemo, one of the supported platforms by XtreemOS-MD, also uses DEB packages. Both Maemo and Ubuntu are distributions based in Debian. This implies than porting to Ubuntu or Debian is easier than porting to other distributions (e.g. RPM based distributions).
3. Ubuntu is a very popular distribution; according to *distrowatch* statistics, Ubuntu is the first in the "2008 Page hit ranking" (see [32]). But Ubuntu is not yet supported by XtreemOS, so porting XtreemOS-MD to Ubuntu will allow new users to discover XtreemOS.
4. Ubuntu has a predictable release plan and a longer time support than other free-available GNU/Linux distributions.

9 Future work

9.1 Next steps

Next step is developing the solutions described in this deliverable. This document is the last one about design of XtreemOS-MD layer F, but new features may be explored before the end of the project if applications require it.

Bibliography

[1] XtreemOS Consortium. *Requirements and Specifications for Advanced VO Support in Mobile Devices* D2.3.5. Integrated Project, November 2008

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

[3] XtreemOS Consortium. *Design of a Basic Linux Version for Mobile Devices* D2.3.3. Integrated Project, December 2007.

[4] XtreemOS Consortium. *Linux-XOS for MDs/PDAs* D2.3.4. Integrated Project, May 2008

[5] eCOS web site.
http://ecos.sourceware.org

[6] OpenMoko source code.
http://wiki.openmoko.org/wiki/Source_Code

[7] OpenMoko GTA01 model.
http://wiki.openmoko.org/wiki/Neo_1973

[8] OpenMoko Neo_FreeRunner model.
http://wiki.openmoko.org/wiki/Neo_FreeRunner

[9] Article in OpenMoko web site.
http://lists.openmoko.org/pipermail/community/2009-April/044915.html

[10] Nokia Maemo web site.
http://www.maemo.org/

[11] Beagle Board web site.
http://beagleboard.org

[12] Maemo wiki
http://wiki.maemo.org/Tablet_death

[13] Ofono web site;
http://ecos.sourceware.org/

[14] Ofono news in linuxdevices.com.
http://www.linuxdevices.com/news/NS6762676329.html

[15] Android new in slasphone.com.
http://www.slashphone.com/google-push-out-new-android-update-to-fix-root-access-loophole-083163

[16] About Android Dev Phone.
http://developer.android.com/guide/developing/device.html

[17] OpenMoko wiki about Android.
http://wiki.openmoko.org/wiki/Android

[18] About Android in Maemo devices.
http://guug.org/nit/nitdroid

[19] Porting Android to Beagle Board.
http://labs.embinux.org/index.php/Android_Porting_Guide_to_Beagle_Board

[20] About Android's security.
http://developer.android.com/guide/topics/security/security.html

[21] LIMO foundation web site.
http://www.limofoundation.org/solutions/index.php

[22] LIMO foundation web site.

http://www.limofoundation.org/images/stories/pdf/090223%20benefits%20table%20v4.pdf

[23] LIMO foundation web site.
http://www.limofoundation.org/images/stories/pdf/090310_limo_ipr_policy_final%20%282%29.pdf

[24] LIMO foundation web site.
http://www.limofoundation.org/images/stories/pdf/limo_foundation_ipr_guide.pdf

[25] LIMO foundation web site.
http://www.limofoundation.org/images/stories/pdf/LiMo_Foundation_Bylaws_v2.pdf

[26] Palm developer web site.
http://developer.palm.com/

[27] XtreemOS Consortium. *Design of Basic Services for Mobile Devices* D3.6.2. Integrated Project, May 2008.

[28] About OpenSSL configuration.
http://www.daemon-systems.org/openssl.cnf.5.html

[29] Ubuntu web site.
http://www.ubuntu.com/products/mobile

[30] Ubuntu web site.
http://www.ubuntu.com/news/ubuntu-9.04-unr

[31] Ubuntu web site.
http://www.ubuntu.com/news/arm-linux

[32] Ranking of Ubuntu in distrowatch.com.
http://distrowatch.com/index.php?dataspan=2008