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: 14th October 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 2.1/ Last edited by Telefónica I+D/ Date 12-10-2009

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
1.0	17-June-09	Telefónica I+D	Telefónica I+D	Included comments and modifications from Alvaro's review
1..1	10-Sep-09	Telefónica I+D	Telefónica I+D	First version for internal review after feedback from EC.
2.0	1-Oct-09	Telefónica I+D	Telefónica I+D	Internal revised version.
2.1	12-Oct-09	Telefónica I+D	Telefónica I+D	Version ready for submission

Reviewers
Alvaro Arenas (STFC), Toni Cortés (BSC), Yvon Jégou (INRIA)

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

XtreemOS aims at natively embedding grid computing functionalities into the Linux operating system, in order to harness the heterogeneous computing resources available today, from clusters to mobile devices. XtreemOS-MD is the result of the efforts in adding these native grid functionalities to mobile devices.

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

In the present document, we first analyze in detail the Linux-based mobile phone market, in terms of evaluation of relevant and suitable platforms for XtreemOS-MD. This evaluation is carried out by first presenting general evaluation guidelines and considerations in order to establish our rationale appropriately and always by following XtreemOS main principles and objectives. Then, a set of evaluation criteria, involving technical and non-technical aspects, is presented. After that, the selected suitable platforms are evaluated accordingly, finishing with a set of final conclusions that will drive the design and implementation process.

After the state of the art evaluation, the document focuses on the XtreemOS-MD advanced version architecture and design for the fulfillment of the requirements and specifications identified in D2.3.5, including the following group of features: VO management and security, resource sharing, context-awareness and installation, configuration and additional enhancements.

For each group of features, the document includes a specific chapter providing the details about the new features to be included in the advanced version, several possible scenarios and the design procedures that will profile the XtreemOS-MD-F layer implementation process. Specific interest has been put on the possible resource sharing features in the mobile device, in the context of native support in Linux-based mobile devices. Several options for resource sharing in mobile device are proposed, including data sharing, input/output device sharing and network access sharing, and the required advanced features and modifications highlighted. Moreover, additional security considerations, specific for resource sharing in the mobile device, are discussed.

In addition, context-awareness features and its applicability to mobile device flavor are covered. Those context-awareness APIs will be used by upper G-layer services and applications. Finally, it is explored several important installation and configuration features, mainly focused on the necessity of providing a successful user experience for mobile device native Grid services.

Table of contents

List of figures

Glossary

AEM	Application Execution Manager
ARM	Advanced RISC Machine
API	Application Programming Interface
IMA	Instant Messaging Application
IPv6	Internet Protocol v6
JobMA	Job Management application
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
SDK	Software Development Kit
SSH	Secure Shell
SSL	Secure Socket Layer
UID	User Identifier
VO	Virtual Organization
WP	Work Package
XtreemFS	XtreemOS File System
XtreemOS-MD	XtreemOS for Mobile Devices

1 Introduction

Previous documents in this work package have defined (D2.3.2, see [1]), designed (D2.3.3, see [2]) and implemented (D2.3.4, see [3]) the XtreemOS-MD basic version and also have analyzed the requirements and specifications for an advanced version of XtreemOS-MD (D2.3.5, see [4]), including new functionalities and also extending hardware support to mobile phones. In this document we are designing the advanced version of XtreemOS-MD to fulfill the requirements previously defined in D2.3.5 [4].

1.1 Advanced XtreemOS-MD for mobile phones

One of the most important features of the advanced version of XtreemOS-MD is the support for mobile phones, while the basic XtreemOS-MD version was designed to be executed on more powerful platforms, like PDA ones. Nevertheless, the mobile device market is changing and evolving very quickly, and there is a clear tendency from mobile operators to offer more advanced mobile terminals with the main goal of capturing new customers, usually bundling the mobile device with some data or voice subscription. In fact, the traditional hardware and software main differences between feature phones, smartphones and PDAs are progressively disappearing, and the concept of smartphones is becoming the natural evolution of PDAs when adding common mobile phone features. Not surprisingly, consumer and media oriented smartphones are taking the center stage in carrier portfolio selection and subsidy allocation.

However, the mobile phone market for Linux is quite fragmented and the different mobile Linux platforms are not always compatible between them. Hence, we have carefully analyzed the available platforms and terminals, in order to prioritize the platforms in which XtreemOS-MD advanced version should be implemented. Of course, we target our analysis in Linux-based terminals and we exclude other mobile phone OSs, such as Windows Mobile or other proprietary OSs, following XtreemOS goal of "Build and Promote a **Linux-based** Operating System to Support Virtual Organizations for Next Generation Grids".

The platform and terminal candidates we have analyzed in this deliverable are terminals based on OpenMoko platform, Nokia Maemo 5, Android and LiMo platforms, the ones that we initially consider the most appropriate potential target for XtreemOS-MD advanced version.

1.2 Document structure

The document structure is as follows:

- Chapter 2 performs a state of the art analysis of Linux-based mobile platforms. This analysis is performed to obtain useful conclusions and considerations that should guide the rest of the design process.
- Chapter 3 covers the architecture of XtreemOS-MD advanced version, including specific design for some of the analyzed platforms.
- Chapter 4 is focused on the new functionalities related to VO support and security, those that are precisely some of the key core functionalities of XtreemOS-MD F-layer.
- Chapter 5 goes into the design of resource sharing for mobile device, taking into account the special characteristics of MDs in comparison with other flavors.

- Chapter 6 is dedicated to context-awareness APIs that will be offered to other layers and applications to expose the context-aware information of the mobile device.
- Chapter 7 gives the design for new installation, configuration and additional enhancements that were previously identified in the requirements for the advanced version.
- Finally, Chapter 8 is our future work chapter that exposes the next steps to implement the XtreemOS-MD-F layer advanced version within the context of the project.

2 Linux-based mobile phone's state of the art analysis

2.1 Analysis guidelines and considerations

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

Our interest on mobile phones is not only related to the mobile device voice and messaging capabilities or to the mobile access to data networks (2G/3G and beyond), but also to the dissemination possibilities of XtreemOS as a whole: XtreemOS-MD software may be provided to an increasing number of Linux-based mobile devices that are usually distributed through mobile operators' sale channels or sometimes directly to the customer and free of any operator contract agreements. In fact, all the major mobile operators typically offer subsidized mobile device terminals, which benefits final customers as they may purchase high quality devices (feature phones and smartphones) and also the operators because they have mechanisms to retain their customers or to capture new ones. XtreemOS-MD is also benefited from that model because more users will be able to obtain access to Linux-based mobile terminals with the minimum hardware and software characteristics required to fully exploit XtreemOS capabilities as a whole.

Once focused on Linux-based mobile terminals, there are several considerations that should be taken into account:

Platform fragmentation

Linux-based mobile phone market is very fragmented in platforms that are incompatible among them. For instance, in some Linux-based platforms like Google Android and Palm Pre the applications that users are allowed to install are not Linux native applications, and developers have to program their applications using HTML/Javascript/Java languages and platform dependent SDKs.

Native applications

In the Linux arena there are thousands of open source C/C++ native applications for PC that may be easily ported to Linux-based mobile devices without having to rewrite all the code. Many of these native applications may require access to mobile device's input/output peripherals, like GPS receivers, accelerometers, camera, voice, modem, etc. In a native code application, accessing these peripherals is done by using standard input/output libraries and APIs of the operating system itself, which allows for better control of the device, code optimization, application response time, and finally better user experience, a key factor to succeed on the very dynamic mobile device marketplace.

Application model

A Web applications model is not always the best solution, in particular for Linux-based devices, because porting every single well-known native application to a Web application will require a significant effort. Moreover, in many cases, as accessing local peripherals is very limited in the Web apps model, not to say accessing to basic operating system services, the adaptation of the native application will require modification of system stacks, which may be out of the public specification, and consequently could be different for each mobile device or version.

Not surprisingly, major players who follow the Wep app model approach, sometimes called "WebOS", know quite well this type of issues and provide mechanisms to access some of the user device's peripherals from within the Web app. For instance, Google Gears software is a plug-in component for Web Browsers. It includes a geo-location API that provides an estimated user's position using either onboard GPS or other means (network location providers). Google Gears is included in Google Chrome browser (for PC) and also in Google Android OS for Android mobiles. Yet another contender we should mention is Google Chrome OS, that was announced July 09 as an operating system **for netbooks** (Intel x86 an ARM architectures) but **not for mobile phones**. According to Google, this new OS for netbooks is expected to be available for consumers in the second half of 2010, and it's promising "the web as a developer platform". Apart from that news, we know nothing more about the technical details of this new OS, as they are not published at the moment of writing this deliverable.

However, not all mobile OS vendors are following the *WebOS* approach. A clear example is Apple, that provides a developer platform SDK in native code (*Objective-C*), for his famous iPhone/iPod proprietary OS. According to Apple, "*the iPhone SDK provides the tools and resources needed to create native iPhone applications that appear as icons on the user's Home screen. Unlike a web application, which runs in Safari, a native application runs directly as a standalone executable on an iPhone OS–based device. Native applications have access to all the features that make the iPhone and iPod touch interesting, such as the accelerometers, location service, and Multi-Touch interface. They can also save data to the local file system and even communicate with other installed applications through custom URL schemes.*" This native application model for the Apple's iPhone and also the success of the application distribution model based on an online service ("AppStore") from which users may download and pay for the applications, **demonstrate that there is a large developer community that still prefers developing their own apps directly over the device native OS**. However, iPhone OS is closed and proprietary and it still has important limitations, in particular user applications cannot be run in the background. In fact, iPhone OS does not allow the user to run or install daemon services. In the last published released of iPhone OS (iPhone 3.0), the manufacturer has provided "push notification" service [5] as a sort of mechanism to alleviate this issue. This local service listens to push messages originated from a remote a web server. Push messages contain a data payload and destination application. Upon receiving the push notification the service starts user's application in charge of managing the data. User's application needs to be already installed in the device and push notification providers must route their data through Apple's gateways. Hence, the issue of running applications in the background is still not clearly solved and thus it limits the scope of applications and services for the iPhone smartphone.

Native support

We should also consider that the primary objective of XtreemOS as a whole is "*To design, implement and evaluate a Grid operating system in open source with **native support** for Virtual Organizations (VO)*", and this is also an inherited objective for the Mobile Device flavor. XtreemOS and XtreemOS-MD foundation layer's main components are Linux native, because that provides a seamless integration with Linux OS.

Hence, **a major consideration is that XtreemOS-MD advanced version still requires that native support**. Moreover, XtreemOS-MD reference applications (IMA, JobMA, etc) and many other open source already available Linux applications will take benefit of XtreemOS services (VO, XtreemFS, AEM, security) by following the XtreemOS approach. One of the most important benefits is that developers won't need to rewrite their applications but use a set of simple APIs provided by XtreemOS, and sometimes, without the need of any application modification at all.

This direct approach not only impacts on creating a developer community around XtreemOS, but also in improving the time to market for those mobile device vendors or operators willing to offer to their customers a real access to Grid services.

Security model

Another topic we shall mention is the applications' **security model**. Many Linux-based mobile platforms have an applications' security model that is not compatible with XtreemOS security model. In particular, there are cases where the applications are completely isolated, not having access to the operating system itself 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, partially removing the advantages that XtreemOS integration would offer.

Software installation

In addition, some mobile device Linux-based distributions prevent users from installing software components when the installation process requires gaining root access to the OS. Installing XtreemOS-MD modules requires root privileges, thus in those platforms it is needed that mobile device manufacturer, OS manufacturer or mobile operator authorizes its inclusion in their base distribution. From a technical point of view, that is not a blocking issue, because XtreemOS-MD is designed as a set of standard Linux-based components, providing modularity and easy integration with Linux base devices. Of course, **in a production scenario, the mobile device vendor is in charge of the inclusion and adequate integration of XtreemOS-MD components**.

Mobile Internet

Another consideration has to do with Mobile Internet market, which is currently under rapid expansion. Mobile operators are highly interested in selling feature phones and smartphones, because these terminals give access not only to their voice services but also to their data services. Although some operators include PDA-only devices in their Mobile Internet devices' portfolio, it is much more common to find feature phones and smartphones as the preferred devices for voice and data access. However, many operators are offering 3G-modems pluggable to netbooks or laptops. In fact, some new netbooks are already equipped with an onboard SIM card reader plus 3.5G hardware. After sales of 13 million netbooks in 2008, this portable mini-PC is emerging as the mobile Internet device of choice for mass-market customers. Some market researchers forecast that in 2014 more than half of all netbooks will be sold by operators bundled with a mobile internet service. Although laptop market is almost monopolized by Microsoft Windows OS, netbooks market is a good candidate for Linux. It is quite common that netbooks are sold with a preinstalled Linux, usually a customized version of a standard Linux distribution. As a consequence, we believe that Linux OS is well positioned for the emerging smartphone and netbook market, and that XtreemOS-MD advanced version design should be done considering this fact. Thus, **another consideration is that the easy porting of XtreemOS-MD from mobile devices to netbooks is a strongly desirable feature**.

Dissemination

An additional consideration is related to the dissemination of XtreemOS-MD. In order to promote XtreemOS-MD, the existence of a developers' community around XtreemOS-MD is very important. This way, XtreemOS will be better known by interested Linux developers and thus this community would push the adoption of XtreemOS-MD into the Linux-based mobile device market. The existence of this developer community should be a good reason why mobile device manufactures and operators would adopt XtreemOS-MD as an available platform preinstalled on their devices. **On top of XtreemOS-MD ready mobile devices, developers would be**

encouraged to create new applications with embedded Grid support, and this model should be useful to create an XtreemOS "market place".

Market trends

It only remains to add some final considerations about the trends of the mobile device market and their impact in XtreemOS-MD.

In the past, the smartphones were defined as mobile devices that provided the standard features of mobile phones (voice communications and messaging), as well as personal information management (PIM) applications and wireless communications capabilities (WiFi, Bluetooth). On the other hand, PDAs provided more hardware resources (CPU and memory) but a minor set of hardware peripherals (camera, GPS, 3G modem, accelerometers).

Nowadays the previous differences between PDAs and smartphones are smoothly disappearing, and basically following a convergence path. Some good examples of this convergence are the new generation of smartphones. For instance, the new Palm Pre hardware is more advanced than Nokia N8x0 series and also Nokia has recently announced the new Nokia N900 smartphone [6], equipped with a more powerful hardware platform and of course including a complete set of hardware peripherals (camera, GPS, 3G).

Moreover, attending to sales reports [7] that show the smartphones evolution (±30% market share in 2009) and the experts' opinion about future trends in the mobile device world[1], we may deduce that the smartphones, more than any other devices, will be able to take advantage of these trends and they will become the natural evolution of mobile phones.

In conclusion, the mobile phone market is quickly converging to an all-in-one mobile device; hence **XtreemOS-MD should be focused on a unique and common version**. Of course, this version may be modified or adapted to support specific cases like several ranges of smartphones and netbooks.

2.2 Analysis criteria

As commented in the introduction section, the Linux-based mobile device market is very fragmented in platforms that are sometimes incompatible among them. In order to cope with this fragmentation issue, our strategy is to **analyze suitable mobile device platforms, based on a set of evaluation criteria**. This process of analysis will guide our design in such a way that it should be feasible or even immediate to port between those platforms that reasonably follow the selected criteria. **Thus, by performing this analysis we are ensuring that XtreemOS-MD advanced version will be available for dissemination to the developers' community and thus we are leveraging the adoption of XtreemOS-MD to more OS and device manufacturers, which are the key players for a success story of XtreemOS as a whole.**

In order to proceed with this analysis, we have defined some criteria that will allow us to compare the different possibilities, quantifying in some manner the strengths and weakness of each one. We also distinguish between technical and non-technical criteria.

[1] As said by Juergen Stark, corporate VP and general manager of Motorola's Mobile Devices, "*There are three key trends that will cause a fundamental change in the role and nature of smartphones over the next few years: broadband everywhere, digitization of all content, and pocket computing power"*

2.2.1 Technical criteria

2.2.1.1 Terminal availability

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

1. Availability of developer terminals: Even if final users (i.e. non-developers) may be not allowed to install XtreemOS-MD in their devices, it's highly recommended that at least any developer can buy an unlocked phone (i.e. developer terminal) at a reasonable price, install XtreemOS-MD and develop their own Grid-enabled applications. The success of our dissemination heavily depends on gaining a developers community and this is one of the main advantages of public projects with open source licenses.
2. Continuity and availability of commercial terminals: 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 to the end users 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.

2.2.1.2 Native applications support

Some smartphone platforms are based on Linux, but Linux is not "visible" to the applications. Indeed these platforms could migrate in the 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 actually intended to support existing Linux applications, but to develop new ones based on vendor specific APIs. If the platform does not allow running native applications, our reference applications will not be able run in the smartphone. This issue is especially important for our instant messaging reference application (IMA), which serves as good example of how a traditional Linux application can be easily extended with XtreemOS-MD.

Moreover, thousands of open source applications already available in the Linux world will not be able to be executed or even ported to the XtreemOS-MD smartphone. As we have already stated before, XtreemOS-MD requires native support and also native support for applications is needed if we really want to disseminate XtreemOS-MD. Having native applications support gives developers the opportunity to easy porting existing native applications or developing new ones, and directly use the grid services provided by XtreemOS.

2.2.1.3 Security model compatibility with XtreemOS

Some Linux-based mobile phone 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 is 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 jobs, because each job would run with a different identity.

Avoiding sandboxing for XtreemOS-MD applications, one possible solution would consist in running signed XtreemOS aware applications with a different security model than the generic applications' security model. Of course, this type of modification directly impacts on the vendor's security model, and probably on their marketing model, so it should be allowed and developed by the OS vendor.

2.2.1.4 Integration risk

Mobile device's platforms are changing quickly and there is not a favorite platform. Probably, most of the platforms will evolve quickly and suffer significant changes to become more robust, provide more features and perhaps ensure compatible between them.

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 there are integration risks because could base systems are evolving quickly and manufacturers even could replace its own base system without modifying their public applications' API. In order to successfully deploy XtreemOS in a selected platform, it is important to keep a low integration risk.

2.2.1.5 Users may install XtreemOS

To achieve maximum dissemination, it is necessary that final users are allowed to install XtreemOS MD modules as any other application or software, at least in the short-term. This is the case with the basic version for PDAs and MIDs, but unfortunately, this is not always possible in many mobile phone platforms, which are locked by default even when the platform is announced as "open" by their vendor.

2.2.1.6 Development tools availability

One of the most important criteria to decide the platform - following the premise "to achieve a community around the project and maximum dissemination for XtreemOS"- is the set of tools offered by each platform to facilitate the development work.

The platforms offer tools for developers such as:

- Emulator: It allows developers to test and debug quickly the applications in different version of platforms, applications, modules, etc., without a real installation in a terminal.
- Plugins: Specific plugins for IDE's (integrated development environment) that ease the applications development or modifications, automating common tasks (compilations, tests, debugging, etc.)
- Documentation: A platform should offer a lot of detailed information about its architecture, technical characteristics, how it works and several guidelines for developers.
- Developers community: A platform should have a large and active developers community powered by technical webs, IRC channels, wikis, etc., where the technical information are present as well as doubts and errors resolution and new ideas and implementations for the platform.

In short, it is necessary to study the availability of development tools of each platform, taking into account that these tools allow us to facilitate the integration and minimize the learning curve.

2.2.2 Non-technical criteria

2.2.2.1 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.

2.2.2.2 Expected future of the platform

This criterion is about the probability that mobile operators will sell Linux-based mobile phones based on 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 Detailed analysis

2.3.1 Platforms summary

This section presents a summary of platforms that will be analyzed. The platforms analyzed in detail are OpenMoko, Nokia Maemo 5, Android and LiMo.

2.3.1.1 Openmoko

OpenMoko [7] is a project dedicated to deliver mobile phones with an open source software stack. 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 announced in 2006 by *First International Computer* (FIC), their founders. There are two main phones, the first one was the Neo 1973 (GTA01) [9], followed by the current model, Neo FreeRunner (GTA02) [10], that is available for any user or developer.

2.3.1.2 Nokia Maemo 5

Maemo [11] is the Linux platform launched by Nokia for Mobile Internet Devices (MID). Nokia MID hardware is very similar to most modern smartphones. The next generation of Nokia devices for Maemo will include support for 3G data and voice, thus bringing to the market the first Nokia smartphone based on Maemo. This new smartphone will be governed by a new software release of Maemo, Maemo 5 (codename Fremantle). This software will be different from Maemo 4, the software already supported by XtreemOS-MD basic version. Indeed, Maemo 5 won't be available for Nokia N8x0 terminals, just for the new Maemo smartphone.

2.3.1.3 Android

Android is 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. Every application is 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**.**

2.3.1.4 LiMo

The LiMo (Linux Mobile) Foundation is an industry consortium aiming [12] "to create an open, Linux-based software platform for use by the whole global industry to produce mobile devices through a balanced and transparent contribution process enabling a rich ecosystem of differentiated products, applications, and services from device manufacturers, operators, ISVs and integrators" (deliverable D2.3.3 includes an in-depth description of this organization). LiMo has completed two release specifications, the last one (R2) was announced on June 09. According to LiMo Foundation, LiMo R2 compliant commercial handsets will be available starting Q4 2009.

2.3.2 Platforms detailed analysis

2.3.2.1 Openmoko evaluation

OpenMoko		
Criterion	**Discussion**	**Result**
Terminal availability	FreeRunner, the current OpenMoko terminal, is available and it is fully unlocked. There is no distinction in OpenMoko between "smartphone for users" and "smartphone for developers". The only caveat is that FreeRunner terminal may become a bit obsolete in several	Fair

	months compared with more modern devices, and the future plans to launch a new terminal have been suspended.	
Native applications	OpenMoko applications are standard Linux applications. Applications available in Angstrom distribution can also be compiled to run in OpenMoko.	Yes
Security model compatiblity	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.	Yes
Integration risk	The main risk for successfully deployment XtreemOS-MD in OpenMoko platform comes from the quick and deep software changes between different versions of the platform (e.g. from OM2007 to OM2008 platform moves from Gnome Mobile to Qt). Last version (2009) is a beta version under development.	Medium
Users may install XOS	OpenMoko was born to allow users to have full control of the terminal, and so users are allowed to install any software.	Yes
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.	Yes
Expected future of the platform	It's not probable that the full OpenMoko platform will be 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. OpenMoko 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.	Reference implementation platform
Development tools	Openmoko offers a multilingual wiki [13] where it is described how to start an Openmoko development plan and development environment options: Toolchain [14] to develop a single application and OpenEmbedded [15] which builds its own cross compiler for System Integration and customizing a distribution. There is not a practical emulator for Openmoko terminals. Even if it is possible to run Openmoko as a QEMU image, the result is not satisfactory. Most Linux libraries are ported to Openmoko platform and they are available in the repositories. In addition, this platform allows building easily other native code with a cross compiler.	Medium

2.3.2.2 Android evaluation

Android		
Criterion	**Discussion**	**Result**
Terminal availability	There are several Android-based commercial smarphones already in the market. Commercial Android terminals, i.e.	Fair

	those distributed by mobile device vendors are operators, are locked, and users are only allowed to install applications but not to modify the 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]). However, ADP1 distribution is limited, it can only be purchased by previously paying a fee not related to the terminal itself, and it is not always immediate to obtain the terminal. That issue limits the terminal availability for developers.	
Native applications support	Android SDK does not support native applications. Android third party applications developed with Android SDK are signed packages, compiled with bytecode to run in Dalvik VM. After releasing the Android SDK, Google released the Android NDK (Native DeveloperS Kit) in June 2009. According to Google: "The NDK allows developers to implement parts of their applications using native-code languages such as C and C++. This can provide benefits to certain classes of applications, in the form of reuse of existing code and in some cases increased speed." Although this is good news for the Linux community it is still too early to know whether Android NDK will provide a path to natively support XtreemOS applications or not. At least, converting a native application to Android using the NDK will require breaking the application in three elements: a library that provides the main application functionality, a JNI interface that gives access to the library methods and a Dalvik program that acts as a launcher of the application. Given that Android does not support X-Windows server, in many cases the Dalvik program should be also in charge of the application GUI, thus adding extra effort in porting the application. We should be here conservative and we evaluate native application support in Android as partial.	Partial
Security model compatibility	Android's third party installable applications based on Android SDK run each one with a different UID. Only applications signed with the same certificate can run with the same UID as referred in [18].	No
Integration risk	XtreemOS integration on Android requires important modifications of the base system. These modifications have to be designed in such a way that the applications' API is not broken or modified. Taking into account that Android OS implements less features than ordinary Linux systems (for example, *libc* library may cause compatibility problems), we may conclude that the integration risk is high.	High
Users may install XOS	Regular users are not allowed to install software 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 [18], but XtreemOS-MD installation requires root permissions (e.g. to	No

	install FUSE module).	
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.	Yes
Expected future of the platform	Android platform has been adopted very quickly by the community and many mobile operators (T-Mobile, Telefónica, Vodafone, etc.) are offering Android terminals. Even it's also possible to install Android, with some limitations, on FreeRunner terminal [19], on Nokia Maemo devices [20] and also on BeagleBoard [21].	Good
Development tools	Android SDK provides a variety of tools to build applications in this platform. Android development tools (ADT) [22], is an Eclipse plugin which allows a quick creation and debugging of applications for Android. Also Android offers a based QEMU emulator in order to help developers to test and debug applications. The Android toolset [23] is completed with other tools for debugging, packaging, and installing. i.e.: Hierarchy Viewer (debug and optimize the user interface), sqlite3 (access to SQLite data), Traceview (traces and logs analysis), UI/Application Exerciser Monkey (stress-test applications), etc. On the other hand, Android offers a large developers community and a extend websites with guidelines, tutorials, source code, samples code, best practices guides, etc. An additional consideration, to take into account for the applications final tests, is that Android supports different phones. Finally it is necessary to remark that if it is necessary a concrete Linux library for an application, it should be built using a cross compiler, what may be a non trivial procedure in Android.	Excellent

2.3.2.3 Nokia Maemo 5

Nokia Maemo 5		
Criterion	**Discussion**	**Score**
Terminal availability	Current Maemo terminals support Maemo4; Maemo 5 terminal with 3G is expected to be available before the end of this year, although there is not official commitment from Nokia at this moment. In the meanwhile, a Maemo5 SDK is already available to test the platform (also it is possible to use as hardware the Beagle Board [24]). Previous Nokia Maemo terminals are unlocked and available in online shops like Amazon.	Fair
Native applications	One of the Maemo strengths is the possibility of running Linux desktop native applications, optionally adapted to the mobile device.	Yes
Security model compability	Maemo platform does not implement sandboxing and it allows the installation and execution of applications under the same	Yes

	conditions than a classic PDA or a Linux desktop version.	
Integration risk	Nokia implementations of Maemo are stable and very well documented. In fact, Maemo 5 API is already published, even before the availability of Maemo5 compatible terminals.	Low
Users may install XOS	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 it is not necessary.	Yes
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.	Partial
Expected future of the platform	The new Nokia device based on Maemo5 is the Nokia N900 smart phone which, according to Nokia, will be commercially available October 2009 in selected markets.	Good
Development tools	Maemo Community [25] offers a development's mailing list, an IRC channel, forums and code samples. A developer guide is available in its web as well as other reference to complete the information for Maemo platform. Also the Maemo training courses [26] provide an overview of the tools for developing applications and platform services. There is a specific IDE for Maemo: VistaMax IDE [27] and an Eclipse plugin (Laika) [28] for scratchbox. The list of tools for Maemo developers[29] is extensive, including tools for Code Analysis, Debugging, Networking, Performance, Resource Usage, Test Automation, Wireless Tools based on C programming in Linux-environment, Debian development tools and Debian utilities used in software packaging. Maemo provides tools in order to debug applications or services using the real device not an emulation device. On the other hand, the developments will be tested easily in a unique terminal (Nokia N900). The most of Linux libraries are ported to Maemo platform and they are available in the official repositories. In addition, the platform allows us to build easily other native code with a cross compiler	Excellent

2.3.2.4 LiMo evaluation

Other important platform that it is claimed to be supported by several mobile manufactures is LiMo. It's an interesting platform, because it allows Linux native applications and there are several terminals available in the market [30]. However, LiMo is not an open platform. Source code is under FPL (*Foundation Public License*) that is not an Open Source License. Only LiMo core members have the rights to commercially distribute under this license and non-members have no access to the source code [31][32][33][34]. Therefore, in this Work Package only Telefónica could access the source code, but the modified code may not be either published or shared with other XtreemOS project members or with XtreemOS-MD developers' community. This restriction does not necessarily imply that XtreemOS-MD advanced version will be useless for LiMo terminals. On the contrary, as LiMo is a

native Linux platform, potentially, the XtreemOS-MD software may be adapted and ported to LiMo based devices by LiMo device manufactures.

2.3.3 Conclusion

Once we have provided detailed analysis of several outstanding platforms, and in order to maximize the success rate of XtreemOS-MD advanced version, we conclude:

- From a technical point of view the platform best situated is OpenMoko, Maemo5 followed by LiMo and Android. Maemo5 main risk comes from the fact that terminals are not available yet, although the Nokia N900 terminal is announced by Nokia to be commercially available this year.
- From a non-technical point of view, it is extremely difficult to anticipate the future of each platform and how they will evolve in the near future, especially within the timeframe of XtreemOS project. Hence, from a non-technical point of view we cannot actually provide a clear picture.

Thus, as a summary, our best conclusion is: **in order XtreemOS-MD to be compatible with as many platforms as possible, the design of XtreemOS-MD advanced version components will be standard and as general as possible, always keeping in mind that native support is a must in XtreemOS and that the porting effort of the software to more platforms should be minimized**.

3 General Architecture

The architecture for the advanced version of XtreemOS-MD Foundation Layer is based on the one designed for the basic version, described in D2.3.3 [2], and it adds the necessary elements and modifications to support the advanced features covered in this deliverable.

The full XtreemOS-MD architecture includes the following set of components (see Figure 3-1. XtreemOS-MD architecture for OpenMoko and for the integration of these components in smartphone platforms):

- **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 diagrams, the modules covered by WP2.3 are labeled as "XtreemOS-F components", and provide the basic infrastructure for constructing mobility, resource sharing 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 general, in the XtreemOS-MD distribution; manufactures, application developers or end users will have to provide them. That includes both grid-aware applications (represented as dark grey colour in the figures) and traditional or grid-unaware applications (light grey). Nevertheless, as a demonstrator, XtreemOS-MD will include in the final package:
    - An application for Instant Messaging in mobile devices (a traditional grid-unaware application modified to benefit from Grid features like access to XtreemFS).
    - An specific application for job management that will offer a graphical interface designed for mobile devices.

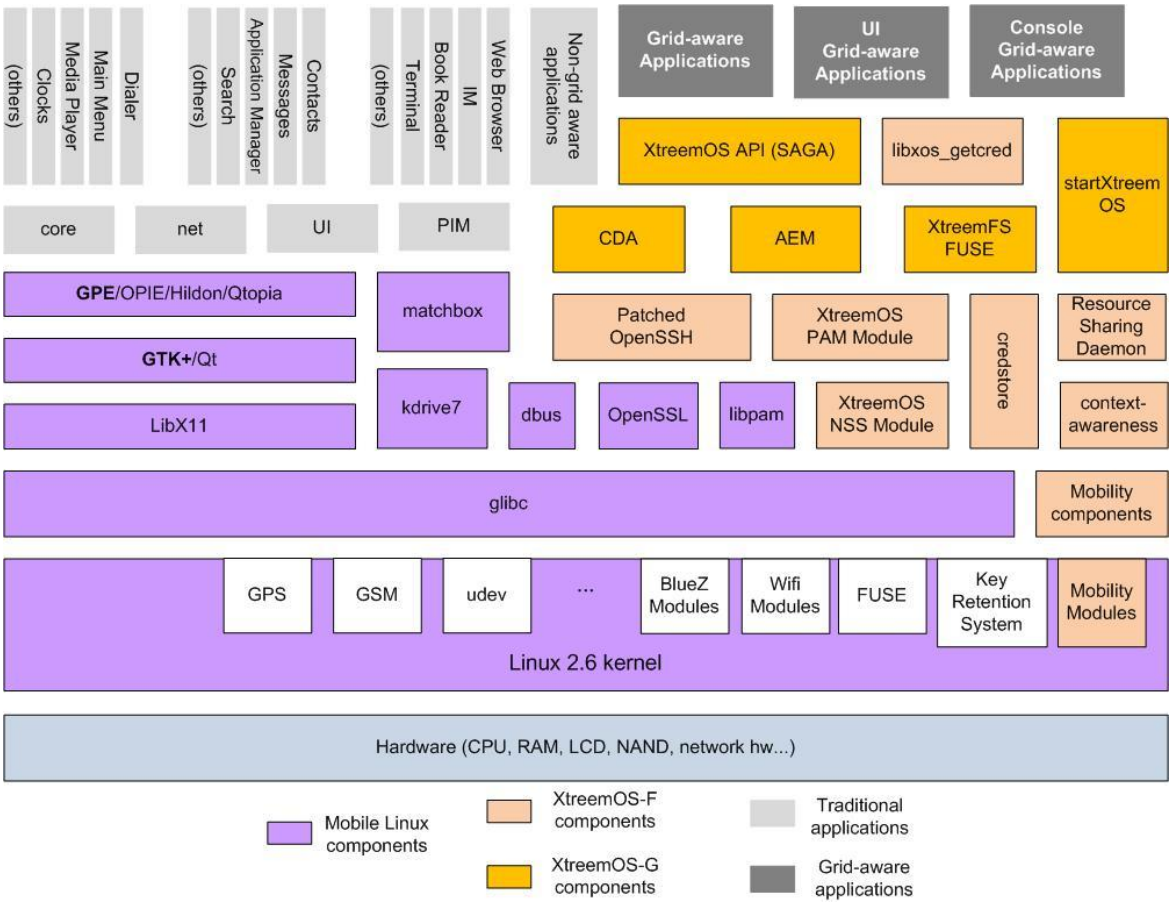

Figure 3-1. XtreemOS-MD architecture for OpenMoko

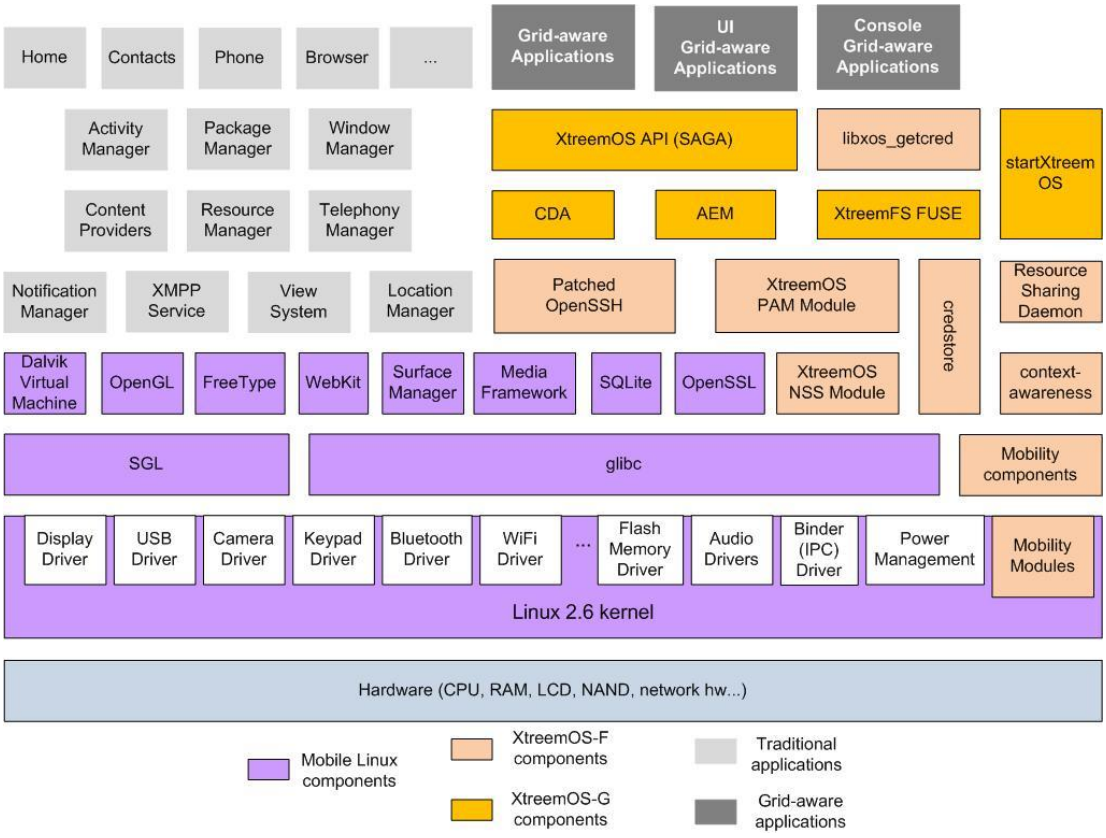

Figure 3-2. XtreemOS-MD architecture for Android

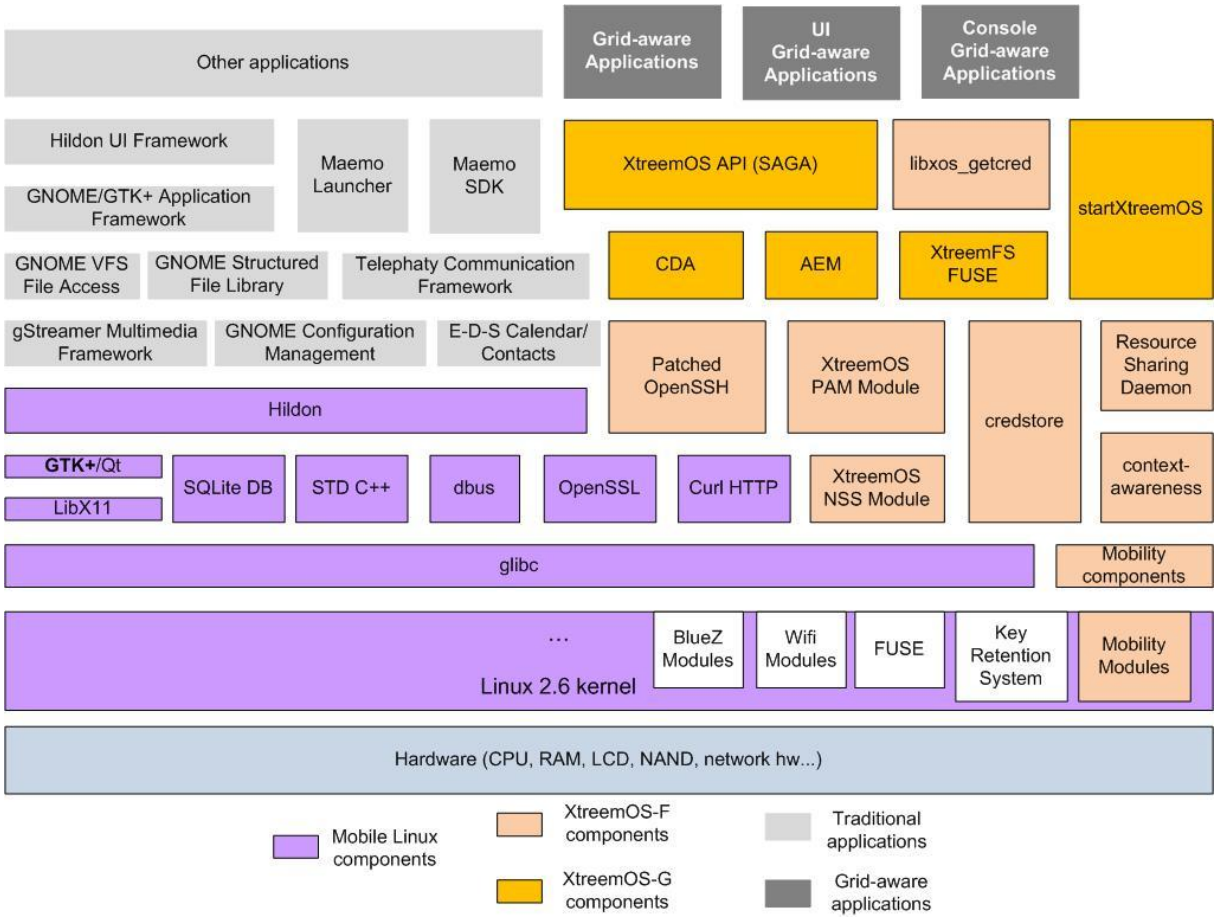

Figure 3-3. XtreemOS-MD architecture for Maemo

3.1 Base Linux components

- **Linux 2.6 Kernel + Kernel Mobility Modules**: XtreemOS-MD advanced version will support different base distributions being the common base of all them a Linux kernel 2.6 but with different minor version number, each one including a different set of modules. Mobility modules at kernel level are also included in order to support mobile IPv6 at least at kernel level.
- **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 the launching of 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's used by XtreemOS to support virtual organizations natively in Linux.

- **OpenSSL** is an open source implementation of the SSL and TLS protocols, implementing basic cryptographic and utility functions. OpenSSH is strongly dependent on this library, which includes all the cryptographic functions used by SSH.
- **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.

3.2.2 Components for VO support in Linux OS

The main components are:

- **XtreemOS NSS Module**: This name service module adds virtual organization support to basic Linux tools that show information about UIDs.
- **XtreemOS PAM Module**: This is a Pluggable Authentication Module that implements most of the low-level security and session aspects of Virtual Organizations.
- **Patched OpenSSH**: This is a modified version of OpenSSH. Patches allow 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.
- **`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.

3.2.3 Components for enhanced mobility of the device

The main components are:

- **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 providing terminal mobility to IPv6-aware applications and services.

3.2.4 Components for resource sharing:

The main component is:

- **Resource sharing daemon**: This daemon will be responsible for providing the resource sharing features in mobile devices. Resource sharing will be implemented using modules to be loaded by the resource sharing daemon. This mechanism allows a modular resource sharing implementation. Resource sharing modules may be installed as an application and vendors will be able to allow or prevent installation of specific modules or require them to be signed in the order to apply the vendor's policy. Users will be able to enable or disable the service and configure each resource sharing module.

It is expected that a high percentage of the code needed for resource sharing implementation will be completely new, due to the following main reasons:

1. New resource types: several resources types are not available yet in XtreemOS PC flavor (e.g. sharing the video camera, microphone, GPS) so those resource types will have to be added to the general architecture of XtreemOS.
2. XtreemOS PC flavor is coded in Java (1.5 and 1.6) which imposes a huge dependency regarding memory and local data storage for mobile device (including smart phones). Currently, there are not any Linux smartphone supporting an official JRE (Android VM is based in a Java dialect, but doesn't support standard Java API).

3.2.5 XtreemOS-G layer

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

- **Security component**: This component includes the communication tools necessary for the mobile device to talk to the VO Management and Security services.
- **AEM component**: This component provides access to XtreemOS Application Execution Management system.
- **XtreemOS FUSE**: This component provides access to the XtreemFS file system, 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 [35] for further details**).**

3.2.6 Applications

We are not focusing directly applications on this level, as this document is mainly focused on F-layer design. Anyway, D2.3.3 [2] can be used as reference, as there were already some explanations in the architecture of XtreemOS-MD basic version that will be applicable to the advanced version as well.

4 VO management and security

XtreemOS-MD F-layer basic version provides a rich security infrastructure to support a secure access to the Grid. XtreemOS-MD implements a modular, pluggable SSO solution, which allows users forgetting about credential administration. Security in mobile devices is very important, as there are personal data involved, and it's also a challenge: many telephone users have not technical skills and don't understand security concepts like certificates, etc. 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 mobile phones.

4.1 VO Management from MDs

XtreemOS-MD basic version allows VO Management through *VOLife* web interface. The advanced version will support the new, redesigned, *VOLife* web interface (split 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. User logs in with his new user account, selects an existing VO and applies for inclusion into 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 their mobile device a notification for each request to join a VO, allowing him to approve or deny it immediately.

It could 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 may 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*.

4.2 Security enhancements

4.2.1 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 pluggable, modular SSO implementation and the new XtreemOS SSO design, implemented using a modified *OpenSSH*, which will be ported to XtreemOS-MD.

4.2.2 User interface enhancements for security

Mobile applications must be adapted to support a limited interface. For instance, many terminals don't include a fully functional or friendly keyboard and sometimes typing text into the mobile is a boring and difficult task. Specifically, an enhancement that XtreemOS-MD advanced version may provide 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. In order to simplify this process and improve user experience we propose the following new features:

- XtreemOS-MD may provide alternative mechanisms to authenticate against the CDA-proxy without typing a password, but using Bluetooth-pairing technology. For instance, a VO user may run a CDA-proxy module into his PC. The PC may be paired with a mobile device with XtreemOS-MD. The mobile device may access the Grid transparently because there is already a trust relationship between the PC and the mobile device.
- XtreemOS-MD may 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.

4.2.3 Credential modules and cryptography

4.2.3.1 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.

4.2.3.2 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 [36]).

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.

4.2.3.3 `libcredstore` access control by group

This feature provides Grid access to a mobile device application only if it belongs to an authorized group. Currently, when a user starts a Grid session, all his applications have access to the *credstore*. A useful 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 Resource sharing

As it was stated in previous deliverables, XtreemOS-MD basic version (i.e. XtreemOS-MD 1.0) does not provide sharing of mobile device resources. In deliverable D2.3.5, we obtained the first requirements and specifications regarding resource sharing in mobile devices. But we should always keep in mind that, even if the basic idea of creating a node in the Grid is to add some shared resources (like storage or computing capacity), when applying that concept to mobile devices, this is not so evident because of the intrinsic mobile phone limitations: very little storage capacity (in general), reduced computing capacity, and battery life and consumption dependency. In this chapter we will present different resource sharing options specifically adapted to the mobile device flavor in XtreemOS. We will consider three different options for resource sharing: data sharing, input/output device sharing and network access sharing. Each type of resource sharing will be first explained and then we will present some useful technical design considerations that we have to follow in order to provide appropriate resource sharing for mobile devices.

5.1 Resource sharing types

5.1.1 Data sharing

Data and file sharing is a well-known feature in peer-to-peer Grid computing and a common service in Internet social networks. Mobile phones usually provide very low storage capacity, especially when comparing with PCs or netbooks capacity. However, the mobile phones market is changing this tendency and it is now common to find some phones with 8 to 16GBytes of memory for file storage. In this type of terminals it could be useful to share files between users.

In the mobile device flavor we should take into account that users are not expecting to consume their battery very quickly only because they want to share their files. In order to allow file sharing in mobile devices we have to give control to the user and consider especial circumstances such as battery consumption, network stability and network interface to be used for sharing (3G, WiFi, etc). Hence, for the data sharing scenario we will consider two different advanced features that will be combined to efficiently allow a transparent file sharing in the mobile device:

5.1.1.1 On demand file sharing

XtreemOS-MD advanced version may provide a mechanism to upload file uploading from the MD only when other nodes request the file contents. This is a specific feature just for MDs, so it will be developed for XtreemOS-MD. In other words, when a MD user wants to share a file with other users, he will mark the file to be shared but it will not be actually shared until some conditions are met. The first condition is related to network access: users might want their files to be shared only when WiFi access is available and not when 3G access is available. The second condition is related to the moment when other users request the shared file. At that moment, the first time that the file is requested by another user, the file would be replicated to XtreemFS; later on, other users will get the file from the XtreemFS replica. Whenever the original file is changed by the owner it may be updated in XtreemFS.

5.1.1.2 Offline mode (AS2.3.4)

The importance of providing an offline-mode is quite clear, especially in the mobile device world, as mobility may imply continuous changes in network connectivity. A temporary network access loose should not result on a user loosing data, and moreover, users will be interested in being able to work when they are offline and once the connection is recovered, automatically and transparently replicate

their local data changes. Hence, offline mode is necessary for a full support of file sharing in mobile devices.

Offline-mode support is one of the new features that may be implemented in XtreemOS-MD advanced version. Even if this feature will be implemented mainly in G-layer, the F-layer will provide automatic information about network availability and battery level to upper layers, in order to automatically change to offline/online mode when required. In addition, the F-layer will provide a user interface and API to manually switching to offline mode.

5.1.2 Sharing of Input/Output devices

The possibility of sharing input/output devices or equipment has been considered in other grid related projects [43]. However, when thinking on mobile devices and mobile users we anticipate that sharing I/O devices possess some peculiarities. For instance, users might be reluctant to share their cameras with other users, because they could feel that there is a potential security risk of sharing their camera with a malicious user who could invade their privacy. Hence, within XtreemOS-MD whenever an application is trying to gain access to a MD I/O device, XtreemOS-MD will request to the user the permission for this access.

The I/O devices that may be shared by the mobile device in a VO are: GPS receiver, microphone, loudspeaker, screen and camera. Sharing those I/O devices may be useful for many different applications that could be envisioned and developed by the XtreemOS developers' community. Just as an example, sharing GPS position may be used in social network applications or statistic applications, among others.

5.1.3 Network access sharing

Nowadays, many devices are equipped with several network interfaces using different technologies like WiFi, Ethernet, Bluetooth, 3G, WiMAX, etc. In a VO it should be possible to share those interfaces as yet another type of resource. Basically, a device will offer its network access as a resource in the grid. Just as an example, a mobile device might be connected to an internal network (ad-hoc, private at home, etc) and could provide to the VO access to internal resources by means of sharing the access to that network. Not only a mobile device but any networked element in the VO could share the network access. Moreover, if the sharing node is a PC or netbook it could also share file space or the possibility to run processes, thus creating a full VO with full grid capabilities.

5.2 Design considerations about resource sharing

5.2.1 Security and privacy

XtreemOS for mobile device is not only different because it is targeted to mobile devices, but also because the potential users are not necessarily the same as in PC or cluster flavor. We may expect that XtreemOS-PC users will be relatively skilled, mainly in IT technology. At least most of them will have to install a new full Linux into their PCs and they will require a basic to medium knowledge of Linux applications and sometimes command-line interfaces. Some of the PC users will install XtreemOS-PC nodes on their network and probably many of them will install the native client version of XtreemOS-PC to gain native access to the Grid.

However, the situation in the mobile device world is expected to be somehow different. First of all, users are not usually going to open command-line windows into their mobile device and run complicated scripts to gain access to the Grid. Rather than that, probably they will be expecting to

have the same kind of easy, friendly and native user interfaces and experiences as when working with their preferred mobile applications (photo, agenda, calendar, music, video, office apps, games, navigation, social networking, etc.). Very likely, they will prefer to carry only one mobile device to keep connected to their families, friends and coworkers, and use it to run any kind of application, either for their private or non-private life.

In addition, mobile terminals contain sensitive user information (contacts, messages, music, personal videos, private files, call history, paired devices, credentials, etc.). Final users really need to be sure that sharing their devices with others will not expose under any circumstance their private data. It may be considered the possibility of using virtualization techniques in order to separate the shared resources from the private ones. Some authors propose to use virtualization in grid nodes to run un-trusted code [37], but this technology was designed originally to server consolidation, not as a security mechanism. Other authors alert about security concerns of virtualization [38]. Moreover, virtualization always leads to a performance overhead and requires extra memory for each virtual machine, a resource heavily limited in mobile phones. Perhaps, the way to keep separated private info from non-private info in a mobile device would require embedding two different machines (CPU, memory, storage, etc) inside the same phone, but the current trend in software and hardware for mobile phones is not that one.

Besides, most of the MD users will lack of technical skills. Unfortunately, many MD users may be unconscious of the importance of security requirements, or sometimes mobile devices may be lost or stolen. Thus, certificate managing, and especially machine certificates management (like RCA certificates) has to be considered carefully in the mobile device case. Besides, considering the UI limitations of MDs, it could be somehow inconvenient for a MD user to manage a second certificate apart from his personal certificate. As a possible work-around, it might be reasonable to consider that vendors may issue and pre-install those machine certificates into their mobile devices and give clear and comprehensive information to the user about those certificates, their usage and maintenance and what to do in case of losing the terminal.

Finally, we should also consider the real application of resource sharing in the mobile device market. In this market, it is a common practice that many vendors mandate over which software should be accepted to be executed into their device. For instance, applications developers have to sign their applications by following a vendor-based signing procedure, and the application features allowed will depend on vendor specific policies. Moreover, if there is a risk that, by any chance, the application may break or put into danger user's privacy or security, it will be usually rejected. In the better case it may be accepted by the vendor with some restrictions but it will always be forced to alert the users that they are going to install or run an application not authorized by the vendor, so the users will run it under their own risk. The success of sharing resources in the mobile device market (like file sharing which is very common into the fixed market, although there are many legal concerns about content rights, etc.) not only depends on the usage trend and the willingness of the users, but also on the vendors decision on incorporating that application to their portfolio and perhaps on some other legal issues that we may expect to arise. So, from a technical and sustainability point of view, we are obliged to consider these issues and to provide good enough security and privacy protection rules into our XtreemOS advanced version design.

5.2.2 Terminal availability

XtreemOS-MD design is necessarily going to be driven by one specific characteristic of mobile devices: the terminal availability. Terminal availability depends on several key factors that are common to mobile devices, but do not affect in general to PC or cluster flavors:

- **Connectivity**: A good performance when sharing resources depends on good device connectivity. However, mobile devices may be connected intermittently: the terminal could be moving, or the network connectivity may be sometimes limited only to voice or SMS depending on mobile coverage. Network performance (jitter, latency) is usually worse than in fixed networks. XtreemOS-MD services may be affected by network performance, so connectivity conditions will have to be carefully addressed.
- **Bandwidth**: Resource sharing in the mobile device network is constrained by the bandwidth. A pretty-good resource sharing experience may require high mobile bandwidth, upstream (HSDPA) and upstream (HSUPA). Other options are WiFi whenever users have access to a WiFi hotspot and their devices have both Wifi and 3.5G network interfaces.
- **Battery life**: MDs depend on batteries and current battery technology does not allow for long term intensive utilization of mobile device hardware. For instance, a good value for battery life during continuous talk is 8 hours. Thus, any application that provides continuous resource sharing should limit as much as possible the battery consumption. XtreemOS-MD software will take into account battery information, network access and general terminal availability limitations so as to provide a minimum set of conditions to run services on the mobile devices.

5.2.3 Mobile device's OS supported software

XtreemOS node server for PC and cluster flavor is currently written in Java and requires JRE 1.6. However, there are no implementations of JRE 1.6 available for ARM devices[2]. None of the analyzed mobile platforms support Java 1.6 or Java 1.5. In particular, Maemo and Openmoko platforms do not support Java 1.6 JVMs, and even if Android uses a Java-like language for application development, JRE1.6 applications are not supported in Android.

Indeed, the need of full Java support is a quite heavy requirement for a mobile device platform as it was already stated in previous deliverables. However, XOSd module (one of the main services of XtreemOS) is not designed to cope with the restrictions of embedded devices. XOSd daemons currently communicate each other by exchanging Java objects. That may be convenient when the resource node is developed in Java for PCs or clusters, and we must admit that there is a very good Java developers community that will probably continue this line of work. Unfortunately, this is not the case for mobile devices, and it actually seems it won't be the case within XtreemOS project timeframe. The exchanging of Java objects outside Java programs implies a very tricky tough work on serialization of Java objects, and it will lead to a buggy and unreliable software. Hence, resource sharing software for mobile device will need to be developed almost from scratch, and it will need to be optimized for MDs. Moreover, some modifications of other XtreemOS flavors are expected to arise, especially when considering inter-process communication between mobile devices and other non-mobile nodes in XtreemOS.

5.2.4 People as resources

In the context of MD, in previous deliverables (see for instance D2.3.5) it was presented the concept of people as resources, as a mechanism that "allows a user order to participate in the application by performing a certain task in the way that is needed". In XtreemOS as a whole it is already identified the feature of interactive jobs that provide the same type of opportunity for users participating in the execution of a job. Thus, from now on, in order to avoid unnecessary overlaps, people as resources will be part of the more general concept of job interactivity.

[2] *OpenJDK* has not been officially ported to ARM and GNU releases only implements **partially** the Java 1.6 API.

6 Context-awareness

One of the advanced features considered in 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:

- Geolocation API: based on GPS coordinates, CellID information or other means.
- 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 Geolocation API

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.

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 switch to offline mode and/or shutdown grid services.

6.3 Active network interface

The API will provide information about the network(s) interfaces(s) that are active. The information will include the type of active network interface (2G/GPRS/3G/3.5G/Wifi/Bluetooth, etc) as long as the terminal provides that data. This is useful when the user only wants to switch to online mode when WiFi is available but not when the GPRS is available. The API will also provide the IP details of the connection (IP address, etc), which could be useful to detect a possible change of IP address that might be problematic when dealing with opened connections (XtreemFS service, etc).

6.4 Additional information

There are other types of context-aware information handled by the terminal that may be useful for the user or the applications. For instance, terminal's operation mode (normal, silence, meeting, etc) may be provided by the context-awareness interface and, in the future, other context information could be provided by the API if necessary.

7 Installation, configuration and additional enhancements

One of the main objectives of XtreemOS is making the grid easy to use. Moreover, applications for smartphones 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.

7.1 Installation enhancements

XtreemOS-MD basic version is very easy to install and almost plug-and-play (e.g. users just need to provide their VO username during installation). The advanced version will provide several enhancements, with the main objetive of reaching a full unattended installation whenever possible:

- XtreemOS-MD advanced version 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.
- XtreemOS-MD may be installed as a dependency of an application, for example the IMA application, etc
- XtreemOS-MD advanced installer may search a user identity in the mobile device, for instance, a user's account (IM, email) belonging to a specified domain or user's phone number. This method may be useful with enterprises that implement SSO with the same username in every internal server and also in a CDA server/proxy. This method may be also useful for mobile operators.

7.2 Configuration enhancements (AS2.3.7)

One of the main principles of XtreemOS-MD basic version was the full automatic configuration, so that the users don't need to configure anything: administrators or service providers 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 users gain root privileges and use 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 modify it, 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 a malicious application could alter the configuration.
- It could be useful for implementing parental control.
- For those service providers wishing to keep full control of the configuration.

In addition, XtreemOS-MD advanced version will support multiple configurations, allowing users to choose which one is the default setting. Each configuration file includes data like CDA IP addresses, XtreemFS server IP, AEM configuration parameters, etc., as shown below:

```
[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
```

On the other hand, regarding credential acquisition process, one of the advantages of XtreemOS-MD is that users don't need to get the credentials manually from CDA. The software does all the work automatically. In this context, there are two points that advanced version will enhance:

- 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.
- 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.

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

7.3 On-demand starting enhancements

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

- Limited resources of MDs.
- User preferences.
- 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 tries to read a file available in an un-mounted XtreemFS (auto-mounting software that will be implemented by layer G, will invoke *libxos_getcred*).

7.4 Service resuming (RMD2.3.10)

One common issue in mobile networks, which impacts in many mobile device services, is related to network instability, which could require the restarting of some XtreemOS daemons (like for instance the SSO daemon) that require a permanent 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 or other means), the wrapper will use this information to wait until network connectivity is restored. Sometimes, services may terminate abruptly because of memory problems: the wrapper will also test this condition before restarting the services.

7.5 Transparent access to grid resources

XtreemOS provides transparent access to file 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.

7.6 XtreemOS-MD for netbooks (AS2.3.6 and RMD2.3.24)

In D2.3.5 [4], 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 believe that Ubuntu could be a good choice as the GNU/Linux distribution to port XtreemOS-MD. These are the main advantages of Ubuntu:

1. Ubuntu distribution is available for x68 architectures, 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 [39][40][41].
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 more convenient 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 [42]). 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.

8 Future work

The next phase towards the completion of the advanced version of XtreemOS-F for mobile devices is the implementation itself, which is expected to be completed by November 2009 (M42) as first internal release. After a period of two months for testing and integration with the existing XtreemOS components (core node, resource node, etc), the final release of the XtreemOS-F advanced version code should be ready by Feb 2010. In parallel, the tasks related to the design and implementation of advanced version of XtreemOS-G layer will carry on (WP3.6). Afterwards, all the software (F layer and G layer) will be packaged at WP4.1, for both smartphones and PDAs, in order to have the final packaged version of XtreemOS-MD by the end of the project (M48).

A more detailed future work is as follows:

1. The XtreemOS-MD 1.0 version, available for PDAs, shall be ported to mobile phones first. This is necessary mainly because many of the software modules already developed shall work accordingly into the more restricted mobile phones platforms. We will strictly follow the considerations and guidelines provided in this deliverable, in order to ease the availability and porting of XtreemOS-F advanced version to Linux-based OSs for mobile devices.

2. XtreemOS 2.0 (core node, resource node, PC and cluster versions) is currently on its way to a stable release. Several important changes have been done from XtreemOS 1.0 to XtreemOS 2.0, including protocols and APIs in XtreemFS, AEM, etc. Those changes directly impact into XtreemOS-MD advanced version, not only for smartphones but also for the already available PDA version. Thus, the second clear step towards and advanced version of XtreemOS-MD is to modify the current MD software so as to guarantee compatibility with XtreemOS 2.0 as a whole. Of course, PC and cluster version shall be back-compatible with MD flavor, thus some request for changes are expected to arise from the MD side that will promote the associated changes in the PC or cluster version, in order to guarantee full compatibility and interoperability.

3. After those two main previous steps, the work will continue by adding advanced features to XtreemOS-MD, with a certain priority scheme, as we have identified an important number of advanced features in this deliverable:
   - First, we will work on VO management and security enhancements (see section 4).
   - Right after that, we will work on resource sharing (see section 5), first focusing on data sharing and after that on input/output device sharing and network access sharing.
   - Then, the work will continue developing the context-awareness APIs (section 6) plus the installation and configuration enhancements (section 7).

All these developments will be part of the last deliverable of this work package: D2.3.7 “Linux-XOS for MD/MP”.

The very last step will be the porting of XtremOS-MD to netbooks (section 7.6), once all the software pieces including G-layer are stable enough. This porting is expected to be available by the end of the project.

Bibliography

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

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

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

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

[5] Apple Push Notification Service Programming Guide. http://developer.apple.com/iphone/library/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Introduction/Introduction.html

[6] Nokia Press release, The new Nokia N900, http://www.nokia.com/press/press-releases/showpressrelease?newsid=1337594 (last visited 31/08/09)

[7] Gartner Group, Gartner Q2 world smartphone sales figures, http://www.gartner.com/it/page.jsp?id=1126812, (last visited 31/08/09)

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

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

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

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

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

[13] OpenMoko wiki. http://wiki.openmoko.org/wiki/Main_Page (last visited 01/09/09)

[14] Toolchain description. http://wiki.openmoko.org/wiki/Toolchain (last visited 01/09/09)

[15] OpenEmbedded description. http://wiki.openmoko.org/wiki/OpenEmbedded (last visited 01/09/09)

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

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

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

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

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

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

[22] Android Development Tools. http://developer.android.com/guide/developing/tools/adt.html (last visited 01/09/09)

[23] Tools for Andriod. http://developer.android.com/guide/developing/tools/index.html (last visited 01/09/09)

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

[25] Maemo development. http://maemo.org/development/ (last visited 01/09/09)

[26] Maemo training. http://maemo.org/development/training/ (last visited 01/09/09)

[27] VistaMax IDE. http://www.wirelexsoft.com/ (last visited 01/09/09)

[28] Laika - IDE for maemo development. http://www.cs.tut.fi/~laika/ (last visited 01/09/09)

[29] Maemo tools. http://maemo.org/development/tools/ (last visited 01/09/09)

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

[31] LIMO foundation web site. http://www.limofoundation.org/images/stories/pdf/090223%20benefits%20table%20v4.pdf

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

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

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

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

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

[37] Sriya Santhanam, Pradheep Elango, Andrea Arpaci-Dusseau, Miron Livny. Deploying Virtual Machines as Sandboxes for the Grid. http://www.usenix.org/event/worlds05/tech/full_papers/santhanam/santhanam_html/

[38] Steven J. Vaughan-Nichols, Virtualization Sparks Security Concerns, Computer, vol. 41, no. 8, pp. 13-15, Aug. 2008, doi:10.1109/MC.2008.312 http://www2.computer.org/portal/web/csdl/doi/10.1109/MC.2008.312

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

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

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

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

[43] A.S. Grimshaw and A. Natrajan. Legion: Lessons learned building a grid operating system. *Proceedings of the IEEE*, Vol. 93, No. 3, pp. 589-603, March, 2005.