UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA EN INFORMÁTICA

PROYECTO FIN DE CARRERA

DEVELOPING A HEAVY CLIENT-SIDE WEB APPLICATION: SCALENET

Author: **VÍCTOR PIMENTEL RODRÍGUEZ**
Tutor: **DR. JOSÉ IGNACIO MORENO NOVELLA**

MARCH 2011

A mis padres y hermana: Inma, Julián y Sandra.

(no me echéis de casa que ya me voy yo)

Resumen

Internet ha causado un tremendo impacto en muchos aspectos de nuestra vida cotidiana. A medida que la sociedad se va acostumbrando a las facilidades de trabajar en línea, los hábitos cambian de manera acorde. Aplicaciones que tradicionalmente se ejecutaban de manera nativa en la máquina del usuario se están, gradualmente, convirtiéndose en aplicaciones web ejecutadas remotamente.

Al mismo tiempo los navegadores han ido mejorando progresivamente hasta convertirse en potentes plataformas de desarrollo. Esta mejora ha dado lugar a la aparición de aplicaciones web de una gran complejidad basadas en HTML, CSS y JavaScript, distribuyendo una carga de procesamiento importante al cliente. A la vez, se obtienen interfaces flexibles capaces de adaptarse a dispositivos muy dispares.

En este proyecto se documenta el desarrollo de una aplicación web avanzada cuyo propósito es controlar la reproducción de contenidos multimedia en varios dispositivos. Esta aplicación se ha realizado en colaboración con *Deutsche Telekom AG*, durante un estancia de seis meses en Berlín como parte del programa *Erasmus Placement* en 2010.

Dicha aplicación se enmarca dentro del proyecto ScaleNet (2005-2009), una Red de Siguiente Generación (NGN) cuyo fin es un sistema que permita una integración escalable, rentable y eficiente de las diferentes tecnologías de acceso inalámbrico y por cable. El componente desarrollado, la *Interfaz de Administración de la Red Personal* (PNAI), es solo una pequeña parte de ScaleNet que sirve como ejemplo de aplicación sobre esta red.

Aunque la interfaz para estas operaciones ya existía, se solicitó un rediseño completo que integrara mayor funcionalidad y que ofreciera una experiencia de usuario más agradable. Además de la interfaz principal para ordenadores de escritorio, también se explica el desarrollo de una interfaz web para dispositivos táctiles modernos.

Abstract

Many aspects of our everyday life have been drastically affected by the Internet. As society becomes accustomed to the possibilities of working online, habits change accordingly. Traditional applications that are executed natively on the user's machine are gradually becoming web applications running remotely.

Meanwhile on the client, browsers are steadily improving to become powerful development platforms. This improvement has led to the emergence of highly complex web applications based on HTML, CSS and JavaScript, distributing significant processing loads to the client. At the same time, you get flexible interfaces able to adapt to very different devices.

This thesis documents the development of an advanced web application whose purpose is to control the playback of multimedia content across multiple devices. This application was completed in collaboration with *Deutsche Telekom AG*, during a six-month stay in Berlin as part of the *Erasmus Placement* in 2010.

This application is part of the ScaleNet project (2005-2009), a Next Generation Network (NGN) aimed at a system that provides a scalable, cost effective and efficient integration of different wireless and wireline access technologies. The developed component, the *Personal Network Administration Interface* (PNAI), is only a small part of ScaleNet that serves as an example application on this network.

Although the interface for these operations already exist, a complete redesign was requested to integrate more functionality and to provide a more pleasant user experience. In addition to the primary interface for desktop computers, this document also covers the development of a mobile web interface for modern touch devices.

Acknowledgements

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Contents

List of Figures

List of Tables

List of Listings

Chapter 1

Introduction and Objectives

MICHAEL SCOTT : I enjoy having breakfast in bed. I like waking up to the smell of bacon —sue me— and since I don't have a butler, I have to do it myself.
So most nights before I go to bed I will lay six strips of bacon out on my *George Foreman* Grill. Then I go to sleep.
When I wake up, I plug in the grill. I go back to sleep again.
Then I wake up to the smell of crackling bacon. It is delicious. It's good for me. It's the perfect way to start the day.
Today I got up, I stepped onto the grill and it clamped down on my foot. That's it. I don't see what's so hard to believe about that.

The Injury
THE OFFICE

1.1   Motivation

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

1.2   Objectives

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

1.3   Project Phases

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

1.4 Document Structure

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Chapter 2

State of the Art

Don Draper : Well, technology is a glittering lure.
But there is a rare occasion when the public can be
engaged on a level beyond flash —if they have a
sentimental bond with the product.
My first job I was in house at a fur company, with
this old pro of a copywriter, a Greek, named Teddy.
Teddy told me the most important idea in advertis-
ing is *new*. It creates an itch. You simply put your
product in there as a kind of calamine lotion.
He also talked about a deeper bond with a product:
nostalgia. It's delicate, but potent. Sweetheart.
[starts slide show featuring photos of Draper's family.]
Teddy told me that in Greek, nostalgia literally
means the pain from an old wound. It's a twinge
in your heart, far more powerful than memory
alone.
This device isn't a space ship, it's a time machine.
It goes backwards, forwards. It takes us to a place
where we ache to go again. It's not called a wheel,
it's called a carousel.
It lets us travel the way a child travels. Round and
a round, and back home again.
To a place where we know we are loved.

The Wheel
Mad Men

2.1   Introduction

As the Internet evolves, Web development tools mature and multiply at an incredibly fast pace. In a few months the best choices becomes superseded by better and new tools. If we are dealing with a rewrite that is something to take into account.

This chapter describes the technologies used for this project. Since we are working on a existing system, most of them cannot be changed and are system constraints. And since ScaleNet includes so many different modules written in different languages and tools, it is wise to avoid adding even more layers of complexity.

A special case was the existing Java applet. Because of new requirements, an alternative had to be considered. Eventually it was replaced by a new module called the Ajax Push Engine (APE) server. The other major addition to the system was the JavaScript Framework called MooTools. Both decisions are explained and justified in §2.7 and §2.8.

2.2   Existing System: ScaleNet

ScaleNet [1] is a research project developed between 2005 and 2009. Partly sponsored by the *German Ministry of Education*, several major corporations participated, including *Deutsche Telekom AG*, *Alcatel SEL AG*, *Eriksson GmbH*, *Lucent Technologies* and *Siemens AG*. Deutsche Telekom Laboratories (T-Labs) was specifically one of the departments more closely involved.

The aim of ScaleNet is to provide a Next Generation Network (NGN) that integrates different wireless and wireline access technologies. It is advertised as a scalable, cost effective and efficient Fixed and Mobile Convergence (FMC) solution.

Figure 2.1: ScaleNet logo

2.2.1 System Overview

ScaleNet addresses both service and network convergence. At the lower level, the system supports a multitude of heterogeneous physical and logical network elements of fixed and mobile networks into one single all-IP infrastructure. Figure 2.2 lists some of the protocols that could be used [2].
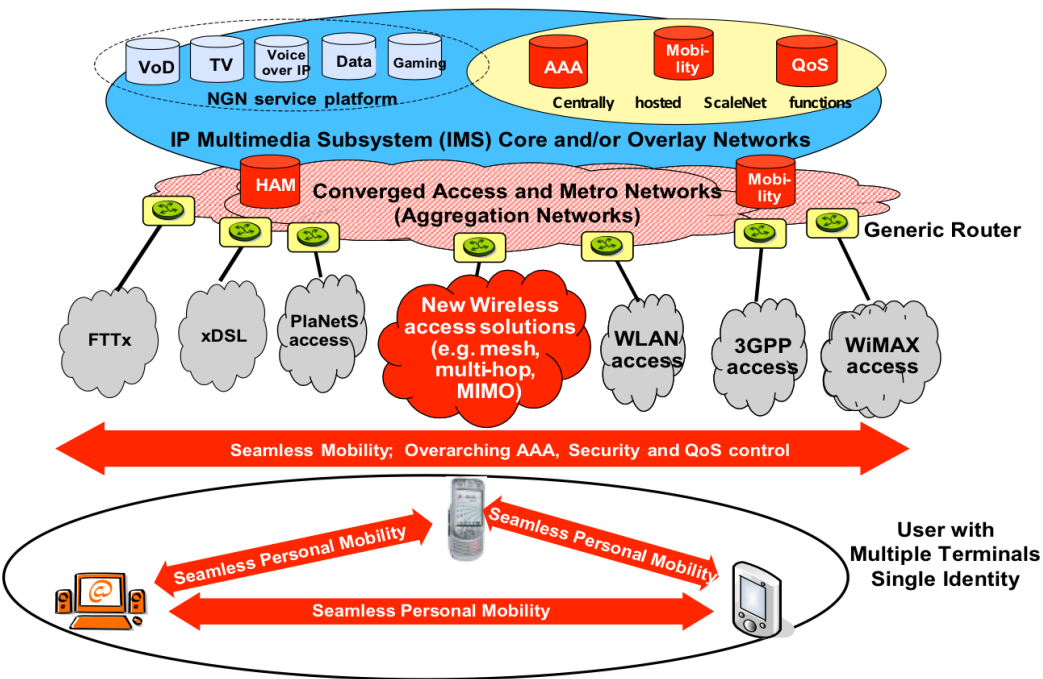
Figure 2.2: Structure of the system

At a upper level, multimedia services relay on the IP Multimedia Subsystem (IMS) framework for the delivery. Theoretically ScaleNet could support other protocols like Overlay Networks or Peer-To-Peer (P2P), but IMS is the one used by the current implementation.

It is important to notice that the own network is user-centric, and transparently handles identities by using Session Initiation Protocol (SIP). This eases handling users with multiple devices; therefore applications do not have to worry about that part.

It is also important to define what a session means in this system. A session refers to the current use of a service, so for every service that the user is enjoying a session is created. For example, if it is viewing a movie but also talking on the Internet Protocol (IP) phone, there are two sessions at the same time.

The creation of a session implies that a new service is created, but it goes the other way around too. If a session is deleted, that service must stop. If the user ends the service, the session must be deleted. That means sessions have to be synchronized with the actual services.

A session is also linked to the device that the user is using. The system allows the copy and transfer of sessions to other devices that he owns, wherever it makes sense. Since the current implementation has also basic social capabilities, that session can also be transferred or copied to a user's contact. In the context of this application a user's contact is called "buddy". Figure 2.2 lists some of the services that can be offered:

- Voice & Video Calls

- Mobile TV & Video on Demand (VOD)

- Massively Multiplayer Online Games (MMOGs)

- Internet Access

The work described in this document is primarily focused on the second application, i.e., video streaming. The idea is that the user can buy a video and play it anywhere using any supported device.

2.2.2 IMS Demonstrator

A logical view of the system is depicted in Figure 2.3, explaining the important nodes based on the capabilities needed. The information relevant to this project is contained in the upper right corner of the figure, the nodes behind the control layer.

In the offices of T-Labs in Berlin and Darmstadt there is a demonstrator with a working implementation of ScaleNet. That demonstrator is composed by several servers and a network infrastructure that enables access to the system using different network protocols and devices. In Figure 2.4 the actual network and hardware are exposed, replacing the same space as in the logical view (Figure 2.3).

Figure 2.5 describes the setup in a better way and highlights the three different planes of the demonstrator. The developed web application is executed from the Web Server and the Application Server, since it belongs
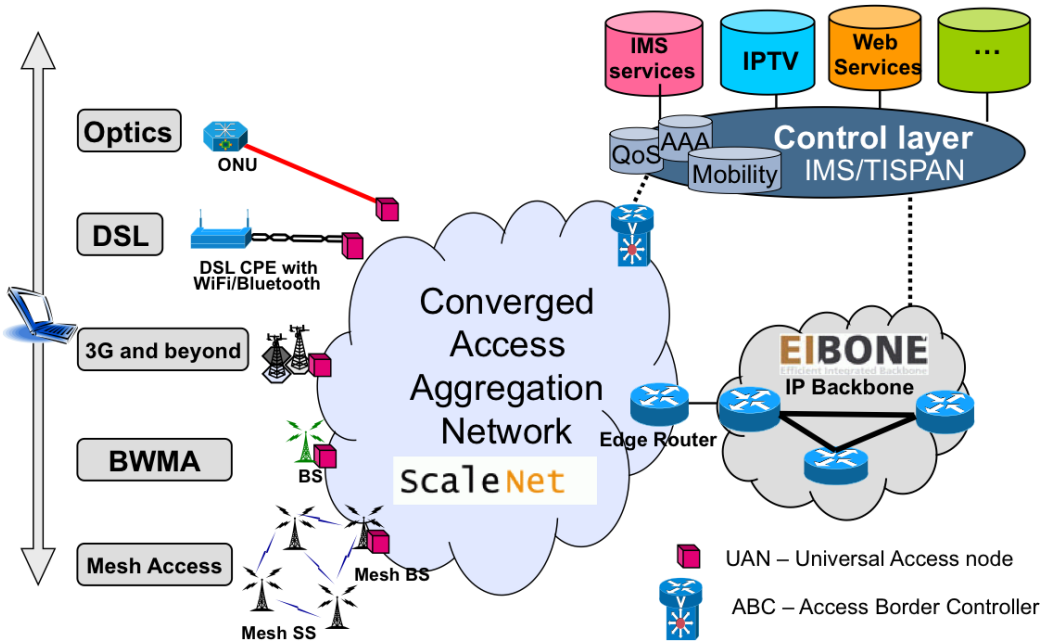
Figure 2.3: System architecture
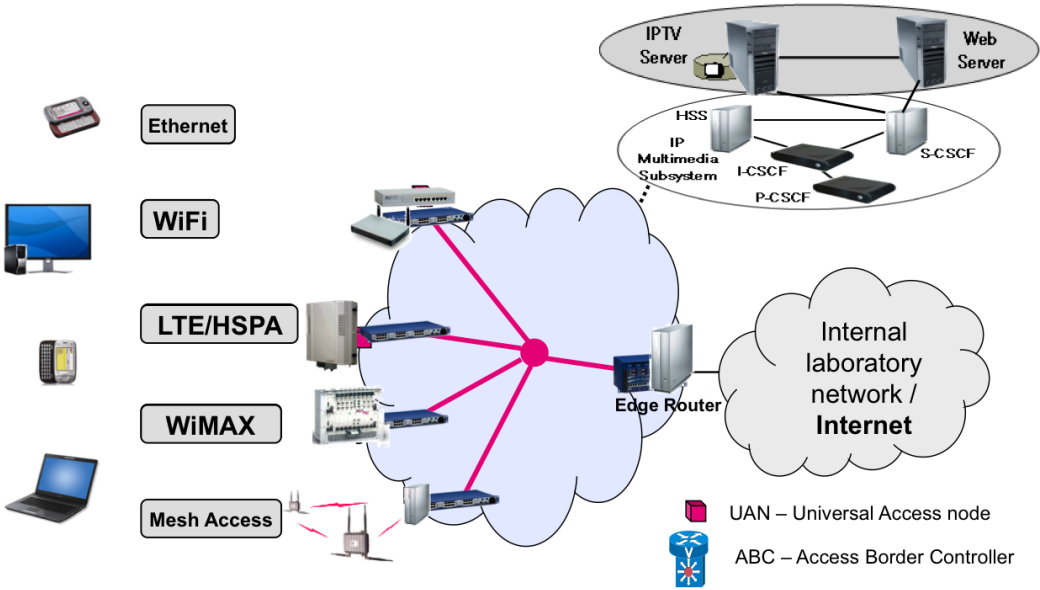
Figure 2.4: Architecture of the demonstrator
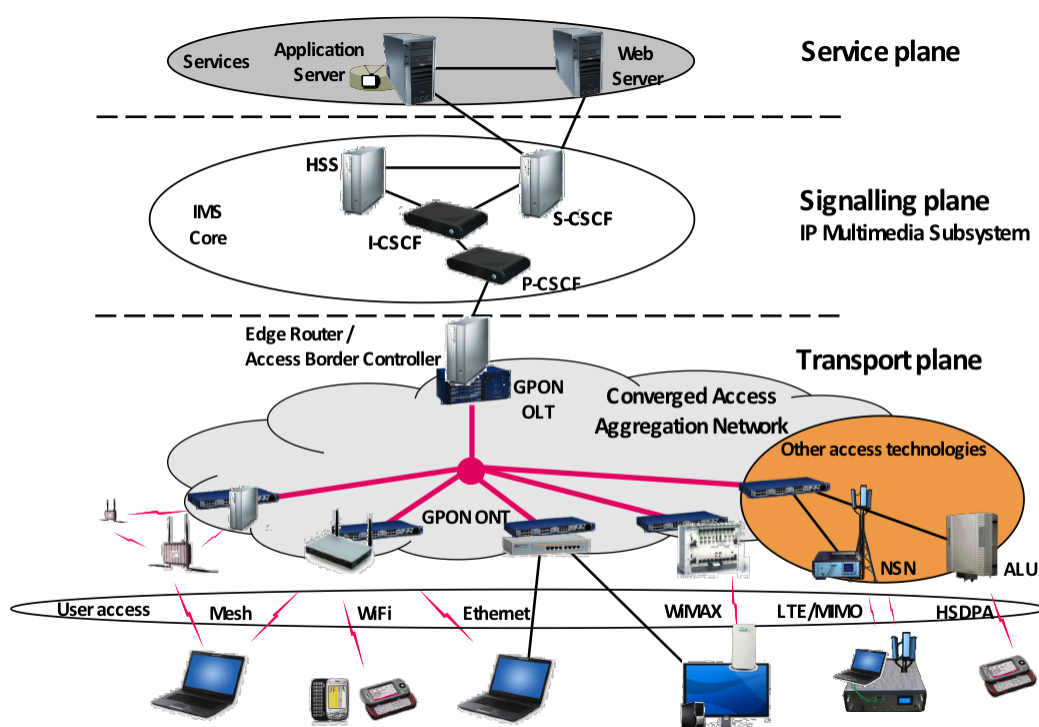
Figure 2.5: Setup of the demonstrator

to the service plane. The signaling plane has also to be taken into account, because it communicates directly with the servers.

However, that is not the real deployment of the hardware used. Whether for convenience or efficiency, tasks are distributed between two main servers. This does not affect the logic of the system, since those tasks could be easily decoupled in an alternate deployment with more servers. Anyway, the interesting pieces of hardware for this project are:

IMS core This machine contains the IMS server[1], but since the IMS load is not very high, it is responsible for other things. It acts as a Web Server (using Apache Web Server[2]) serving PHP applications. It is also the internal DNS server.

Application Server This is the IPTV server, where the video content is streamed. It is also a Web Server, but it serves Java applications based on the OSGi framework[3].

---

[1]The IMS core is open source software from Fraunhofer FOKUS and it can be freely downloaded from: http://www.openimscore.org/

[2]http://httpd.apache.org/

[3]http://www.osgi.org/

User Devices Devices intended for the user to access the services. There is a TV, a laptop and several phones. All of them run a custom IMS client that holds a connection to the servers, allowing the identification and adding IPTV and VoIP capabilities to those devices. In the last phase of the development, an iPhone was added for testing purposes.

This demonstrator contains several demo applications running. The interesting one for this project is the application that handles IPTV streaming.

2.2.3   Personal Network Administration Interface (PNAI)

The Web interface used for the management of sessions is called Personal Network Administration Interface (PNAI). From this interface the user can obtain this information:

- All devices and registered in the system for that user and their online status.

- All buddies for that user and their online status.

- All multimedia sessions related to the user. This includes:

   – The sessions running on his devices, no matter who paid for that content.

   – The sessions running on devices from his buddies and started/-paid by that user.

Those are passive actions, but from that same view the user can initiate some operations to control the system. In Figure 2.6 all the available operations relating sessions are listed following a use case diagram.

In that diagram colors are used to differentiate the different kind of use cases covered. Also two visual marks (* and **) are added in case this is a copy in black a white. The meaning of the colors are explained according to this legend:

Green Available already in the main PNAI page.

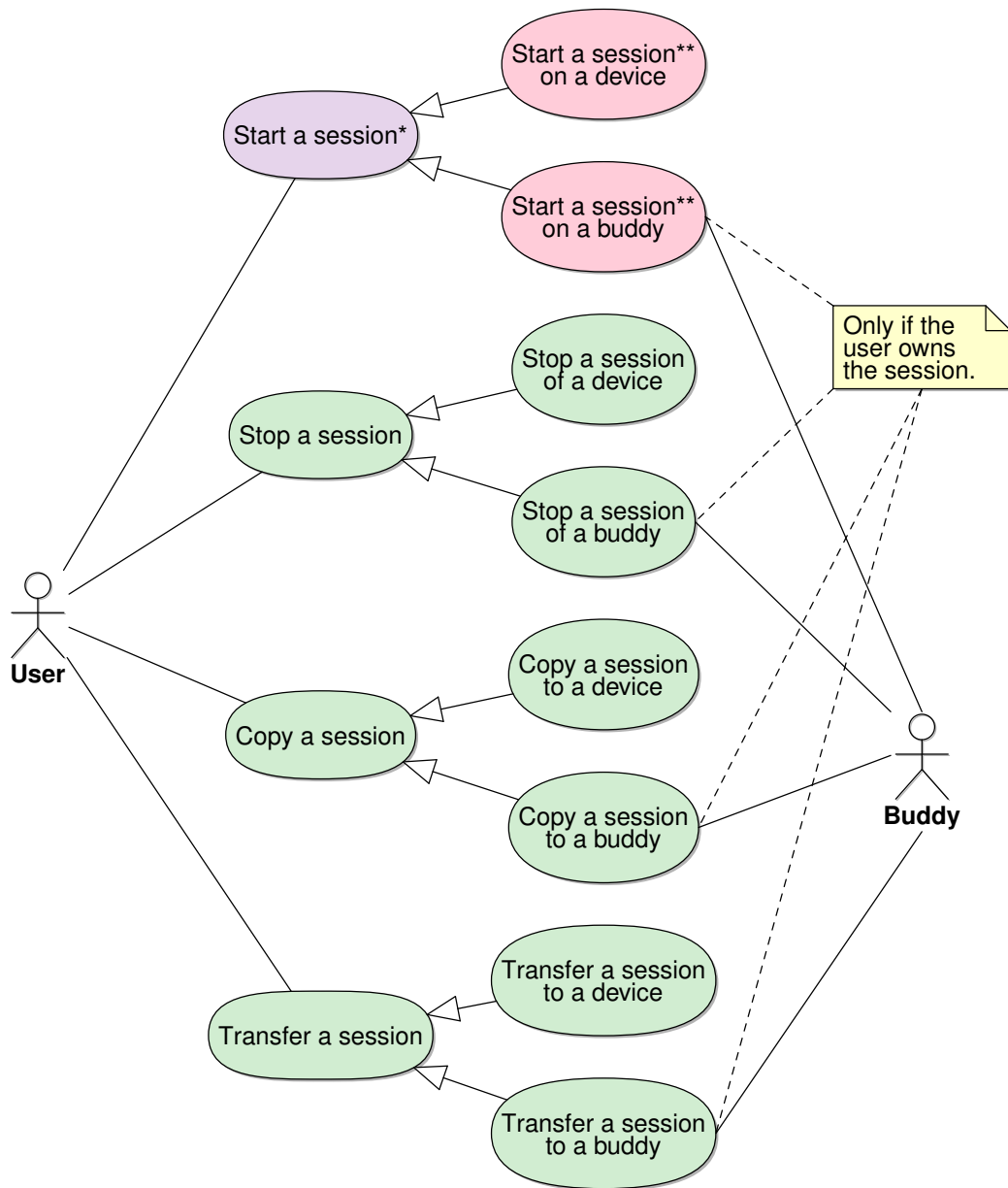**Purple (*marked with* *)** Available in an individual page outside of the main PNAI page.

Figure 2.6: Use cases for the IPTV application

Red (*marked with ***) Not implemented.

As we can see, the main PNAI page has already a lot of functionality, but it can contain even more. Basically the actions available to that user in that page are:

- Terminate a session of a user device or of a buddy if the session is owned by that user.

- Transfer (handover) or copy (duplication) a existing session to a user device or to a buddy if the session is owned by that user. That is, if one buddy bought the content for us, we cannot transfer again that content to another buddy.

Beside of these session related operations, there are other management operations. For example, selecting which device is the default, adding/removing devices or adding/removing buddies. For this document they are not relevant since they remained untouched.

Figure 2.7 shows the old appearance of the main page for a logged user, before any work began. On the left side of the page there is the Device List, where the devices owned by that user are drawn. On the right side there is the Buddy List, where the user's buddies are listed. Finally, the trash bin is in the lower right corner of the Device List.

The devices that are offline are disabled and are drawn with a dimmed appearance. The buddies that are online are preceded by a green icon, while the ones that are offline are preceded by a red icon.

Devices or buddies that are online act as session containers. The reason for the devices to be so big is because inside them the current sessions are drawn. Besides the name of the content playing, session have an icon that changes depending on the type of session (video, audio, call, etc).

The user can interact with the sessions through the mouse using drag&drop. For example, the user can *grab* the icon he wants and drop it in another container to copy or transfer that session. That is a very visual and fast way to manage sessions.

When a user drops the session in another container, a menu will appear to ask the user if he wants to copy or transfer that session. The trash bin acts also as a container, but in a special way: when a session is dropped in the trash bin, that session is automatically deleted, with no menu involved.
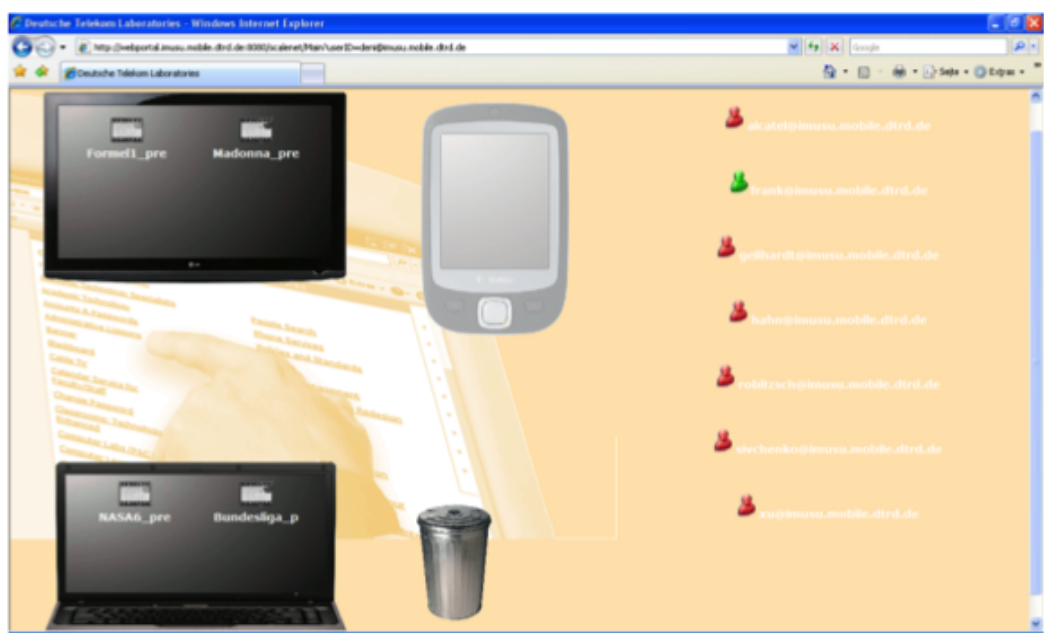
Figure 2.7: Old main PNAI page

Use Cases

In the following tables the current supported use cases are explained step by step. The first use case is detailed in Table 2.1, explaining the situation where the user wants to stop/delete the session in a device.

Table 2.1: Use case 1 – Stop a session of a device

Use Case 1	Stop a session of a device
Actor	System user
Precondition	A session is already running on a device, and it is showing in the PNAI interface inside of that device.
Postcondition	Session must terminate, i.e., the content must stop playing. The user must be notified with a popup and the session icon must be deleted from the PNAI interface.
	continued on next page

Table 2.1: Use case 1 – Stop a session of a device (continued)

USE CASE 1	**Stop a session of a device**
Main Path (M)	1. User starts dragging the session icon. 2. A copy of the session icon appears under the user's cursor, and follows the cursor until the user drops it. 3. User drops the cloned session icon into the trash. 4. A popup appears to notify the user that the action is in progress and the cloned session icon is deleted from the view. 5. The content stops playing. 6. The popup disappears and the original session icon is deleted from the view.
Alternate Path (A1)	3b. User drops the session into a blank space. 4b. Action is cancelled.
Alternate Path (A2)	5c. There is an error with the server and the content keeps playing. 6c. The content of the popup changes to notify the user that there was an error with the server and the action could not be completed. After 5 seconds it disappears. 7c. Action is cancelled.

Figure 2.8 shows how the page looks when it is waiting for a response to the server for the previous use case. Since the user interface does not block in the process, the communication between the front end and the back end must be asynchronous. The use case for terminating a session that a buddy is playing and that we own is very similar, as Table 2.2 exposes.

Figure 2.8: Old main PNAI page: Deleting a session

Table 2.2: Use case 2 – Stop a session of a buddy

Use Case 2	Stop a session of a buddy
Actor	System user
Precondition	A session owned by the user is running on a device, and it is showing in the PNAI interface near that buddy's name.
Postcondition	Session must terminate, i.e., the content must stop playing. The user must be notified with a popup and the session icon must be deleted from the PNAI interface. The buddy is *not* notified, the content stops without warning.
	continued on next page

Table 2.2: Use case 2 – Stop a session of a buddy (continued)

USE CASE 2	Stop a session of a buddy
Main Path (M)	1. User starts dragging the session icon. 2. A copy of the session icon appears under the user's cursor, and follows the cursor until the user drops it. 3. User drops the cloned session icon into the trash. 4. A popup appears to notify the user that the action is in progress and the cloned session icon is deleted from the view. 5. The content stops playing. 6. The popup disappears and the original session icon is deleted from the view.
Alternate Path (A1)	3b. User drops the session into a blank space. 4b. Action is cancelled.
Alternate Path (A2)	5c. There is an error with the server and the content keeps playing. 6c. The content of the popup changes to notify the user that there was an error with the server and the action could not be completed. After 5 seconds it disappears. 7c. Action is cancelled.

Tables 2.3, 2.4, 2.5 and 2.6 show how the user could copy or transfer a session to another device or buddy.

Table 2.3: Use case 3 – Copy a session to a device

Use Case 3	Copy a session to a device
Actor	System user
Precondition	A session is already running on a device/buddy, and it is showing in the PNAI interface inside of that device/buddy. Also, there is another device online.
Postcondition	Session must be copied to that device, i.e., the content must be duplicated and played on that device. The user must be notified with a popup and the session icon must appear in the PNAI interface for the second device.
Main Path (M)	1. User starts dragging the session icon. 2. A copy of the session icon appears under the user's cursor, and follows the cursor until the user drops it. 3. User drops the cloned session icon into another device that is online. 4. A popup menu appears where the user dropped the session, giving options to copy/duplicate the session, transfer the session or cancel the action. 5. The user clicks on the copy/duplicate option. 6. The popup menu disappears. 7. A popup appears to notify the user that the action is in progress and the cloned session icon is deleted from the view. 8. The content starts playing on the destination device. 9. The popup disappears and the same session icon appears inside of the destination device.

Table 2.3: Use case 3 – Copy a session to a device (continued)

Use Case 3	Copy a session to a device
Alternate Path (A1)	**3b.** User drops the session into a blank space. **4b.** Action is cancelled.
Alternate Path (A2)	**5c.** The user clicks on the cancel option. **6c.** Popup menu disappears and action is cancelled.
Alternate Path (A3)	**8d.** There is an error with the server and the content is not duplicated. **9d.** The content of the popup changes to notify the user that there was an error with the server and the action could not be completed. After 5 seconds it disappears. **10d.** Action is cancelled.

Table 2.4: Use case 4 – Copy a session to a buddy

Use Case 4	Copy a session to a buddy
Actor	System user
Precondition	A session is already running on a device/buddy, and it is showing in the PNAI interface inside of that device/buddy. Also, there is another buddy online.
	continued on next page

Table 2.4: Use case 4 – Copy a session to a buddy (continued)

USE CASE 4	Copy a session to a buddy
Postcondition	Session must be copied to that buddy, i.e., the content must be duplicated and played on the buddy's default device. The user must be notified with a popup and the session icon must appear in the PNAI interface near the name of that buddy. The buddy is *not* notified, the content plays without warning.
Main Path (M)	1. User starts dragging the session icon. 2. A copy of the session icon appears under the user's cursor, and follows the cursor until the user drops it. 3. User drops the cloned session icon into another buddy that is online. 4. A popup menu appears where the user dropped the session, giving options to copy/duplicate the session, transfer the session or cancel the action. 5. The user clicks on the copy/duplicate option. 6. The popup menu disappears. 7. A popup appears to notify the user that the action is in progress and the cloned session icon is deleted from the view. 8. The content starts playing on the buddy's default device. 9. The popup disappears and the same session icon appears inside of the destination buddy.
Alternate Path (A1)	3b. User drops the session into a blank space. 4b. Action is cancelled.

Table 2.4: Use case 4 – Copy a session to a buddy (continued)

Use Case 4	**Copy a session to a buddy**
Alternate Path (A2)	**5c.** The user clicks on the cancel option. **6c.** Popup menu disappears and action is cancelled.
Alternate Path (A3)	**8d.** There is an error with the server and the content is not duplicated. **9d.** The content of the popup changes to notify the user that there was an error with the server and the action could not be completed. After 5 seconds it disappears. **10d.** Action is cancelled.

Table 2.5: Use case 5 – Transfer a session to a device

Use Case 5	**Transfer a session to a device**
Actor	System user
Precondition	A session is already running on a device/buddy, and it is showing in the PNAI interface inside of that device/buddy. Also, there is another device online.
Postcondition	Session must be transferred to that device, i.e., playback must be stopped at the source and started at the destination device. The user must be notified with a popup and the session icon must appear in the PNAI interface for the second device.
	continued on next page

Table 2.5: Use case 5 – Transfer a session to a device (continued)

USE CASE 5	Transfer a session to a device
Main Path (M)	1. User starts dragging the session icon. 2. A copy of the session icon appears under the user's cursor, and follows the cursor until the user drops it. 3. User drops the cloned session icon into another device that is online. 4. A popup menu appears where the user dropped the session, giving options to copy the session, transfer/hand over the session or cancel the action. 5. The user clicks on the transfer/hand over option. 6. The popup menu disappears. 7. A popup appears to notify the user that the action is in progress and the cloned session icon is deleted from the view. 8. The content stops playing on the source device. 9. The content starts playing on the destination device. 10. The popup disappears, the session icon is deleted from the view and created again inside of the destination device.
Alternate Path (A1)	3b. User drops the session into a blank space. 4b. Action is cancelled.
Alternate Path (A2)	5c. The user clicks on the cancel option. 6c. Popup menu disappears and action is cancelled.
	continued on next page

Table 2.5: Use case 5 – Transfer a session to a device (continued)

Use Case 5	Transfer a session to a device
Alternate Path (A3)	**8d.** There is an error with the server and the content is not transferred. **9d.** The content of the popup changes to notify the user that there was an error with the server and the action could not be completed. After 5 seconds it disappears. **10d.** Action is cancelled.

Table 2.6: Use case 6 – Transfer a session to a buddy

Use Case 6	Transfer a session to a buddy
Actor	System user
Precondition	A session is already running on a device/buddy, and it is showing in the PNAI interface inside of that device/buddy. Also, there is another buddy online.
Postcondition	Session must be transferred to that buddy, i.e., playback must be stopped at the source and started at the buddy's default device. The user must be notified with a popup and the session icon must appear in the PNAI interface near the name of that buddy. The buddy is *not* notified, the content plays without warning.
	continued on next page

Table 2.6: Use case 6 – Transfer a session to a buddy (continued)

USE CASE 6	Transfer a session to a buddy
Main Path (M)	1. User starts dragging the session icon. 2. A copy of the session icon appears under the user's cursor, and follows the cursor until the user drops it. 3. User drops the cloned session icon into another buddy that is online. 4. A popup menu appears where the user dropped the session, giving options to copy the session, transfer/hand over the session or cancel the action. 5. The user clicks on the transfer/hand over option. 6. The popup menu disappears. 7. A popup appears to notify the user that the action is in progress and the cloned session icon is deleted from the view. 8. The content stops playing on the source device. 9. The content starts playing on the buddy's default device. 10. The popup disappears, the session icon is deleted from the view and created again inside of the destination buddy.
Alternate Path (A1)	3b. User drops the session into a blank space. 4b. Action is cancelled.
Alternate Path (A2)	5c. The user clicks on the cancel option. 6c. Popup menu disappears and action is cancelled.
	continued on next page

Table 2.6: Use case 6 – Transfer a session to a buddy (continued)

USE CASE 6	Transfer a session to a buddy
Alternate Path (A3)	**8d.** There is an error with the server and the content is not transferred. **9d.** The content of the popup changes to notify the user that there was an error with the server and the action could not be completed. After 5 seconds it disappears. **10d.** Action is cancelled.

Figure 2.9: Old main PNAI page: Duplicating a session

Figure 2.9 shows how the popup menu is displayed to the user. It is a very simple menu with only three links, each of which correspond to an action.
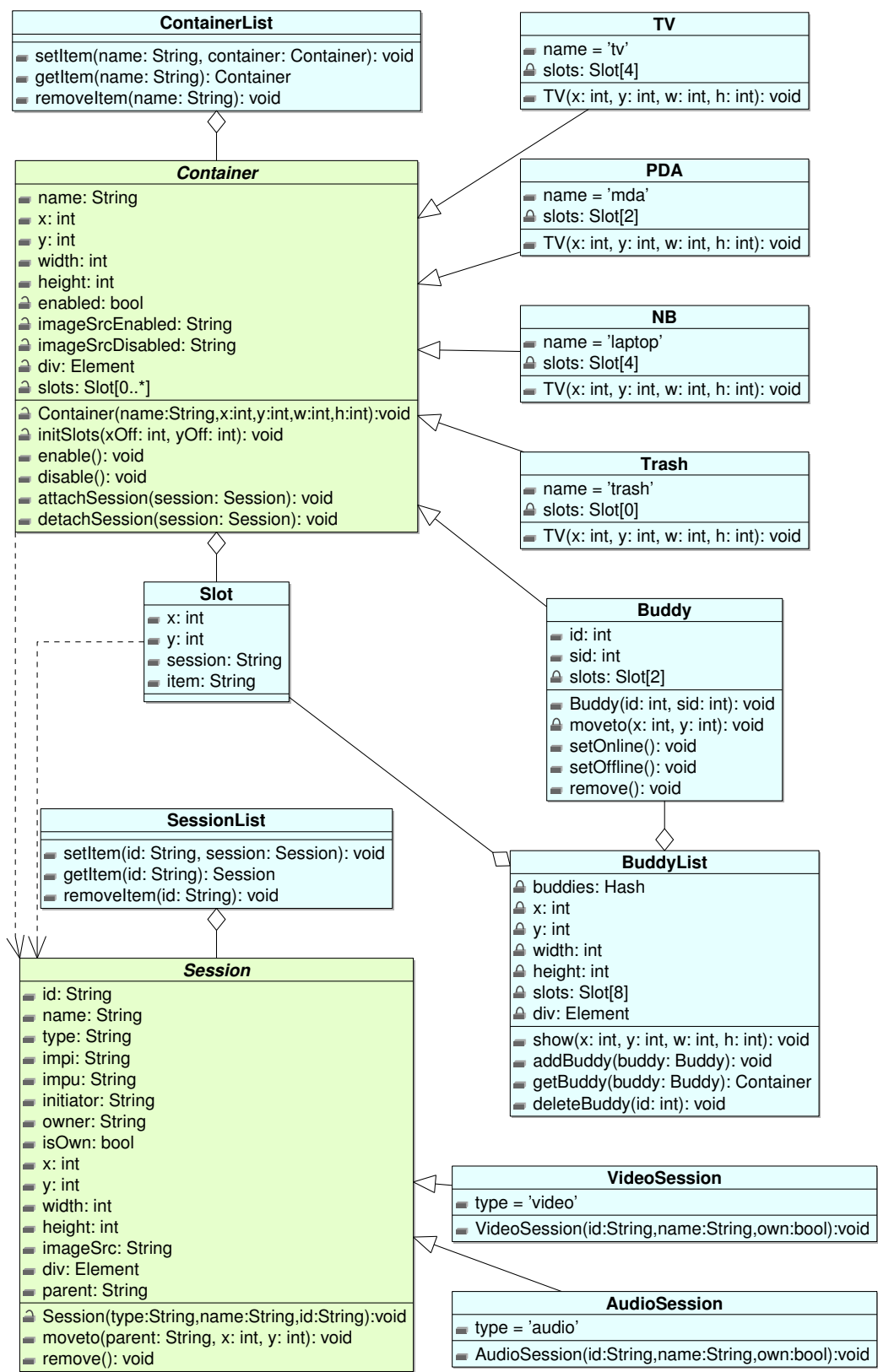
Class Diagrams

ContainerList

- setItem(name: String, container: Container): void
- getItem(name: String): Container
- removeItem(name: String): void

Container

- name: String
- x: int
- y: int
- width: int
- height: int
- enabled: bool
- imageSrcEnabled: String
- imageSrcDisabled: String
- div: Element
- slots: Slot[0..*]

- Container(name:String,x:int,y:int,w:int,h:int):void
- initSlots(xOff: int, yOff: int): void
- enable(): void
- disable(): void
- attachSession(session: Session): void
- detachSession(session: Session): void

Slot

- x: int
- y: int
- session: String
- item: String

SessionList

- setItem(id: String, session: Session): void
- getItem(id: String): Session
- removeItem(id: String): void

Session

- id: String
- name: String
- type: String
- impi: String
- impu: String
- initiator: String
- owner: String
- isOwn: bool
- x: int
- y: int
- width: int
- height: int
- imageSrc: String
- div: Element
- parent: String

- Session(type:String,name:String,id:String):void
- moveto(parent: String, x: int, y: int): void
- remove(): void

TV

- name = 'tv'
- slots: Slot[4]

- TV(x: int, y: int, w: int, h: int): void

PDA

- name = 'mda'
- slots: Slot[2]

- TV(x: int, y: int, w: int, h: int): void

NB

- name = 'laptop'
- slots: Slot[4]

- TV(x: int, y: int, w: int, h: int): void

Trash

- name = 'trash'
- slots: Slot[0]

- TV(x: int, y: int, w: int, h: int): void

Buddy

- id: int
- sid: int
- slots: Slot[2]

- Buddy(id: int, sid: int): void
- moveto(x: int, y: int): void
- setOnline(): void
- setOffline(): void
- remove(): void

BuddyList

- buddies: Hash
- x: int
- y: int
- width: int
- height: int
- slots: Slot[8]
- div: Element

- show(x: int, y: int, w: int, h: int): void
- addBuddy(buddy: Buddy): void
- getBuddy(buddy: Buddy): Container
- deleteBuddy(id: int): void

VideoSession

- type = 'video'

- VideoSession(id:String,name:String,own:bool):void

AudioSession

- type = 'audio'

- AudioSession(id:String,name:String,own:bool):void

Figure 2.10: Class diagram for the old PNAI

2.3 Server Programming Language: PHP

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

2.3.1 Quick Overview of the Language

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

2.3.2 Where it is Used?

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

2.4 Server Programming Language: Java

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim

veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

2.4.1   Quick Overview of the Language

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

2.4.2   OSGI

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

2.4.3   Java Applets

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

2.5 Interface: HTML and CSS

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

2.5.1 HTML

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

2.5.2 CSS

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

2.5.3 HTML5 and CSS3

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim

veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

2.6   Client Programming Language: JavaScript

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

2.6.1   Quick Overview of the Language

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

2.6.2   Compatibility

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat

cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

❧

2.7 JavaScript Framework: MooTools

Before explaining why MooTools has been chosen for this application, another important question needs to be resolved.

2.7.1 Why Use a JavaScript Framework?

There are several reasons that lead to this conclusion, the most important are:

- Because we want to support different browsers. If we do not use a framework a lot of time would be spent debugging the huge differences between Internet Explorer and the rest of the browsers.

- Because we want to facilitate the development, since usually these frameworks cover several holes in the JavaScript specification that allows us fixing common issues with less code.

- Because we want the interface to have advanced effects. We could just search for several scripts that makes one individual effect, but that will result in redundancies, differences in quality code and waste time in searching.

2.7.2 Making the Decision

By the previous standards, we have plenty of options to choose from: jQuery[4], Prototype[5], Dojo[6], YUI[7], GWT[8], Ext JS[9], etc. Overall, these are very

---

[4]http://jquery.com/
[5]http://www.prototypejs.org/
[6]http://www.dojotoolkit.org/
[7]http://developer.yahoo.com/yui/
[8]http://code.google.com/webtoolkit/
[9]http://www.extjs.com/

popular and they offer high quality and plenty of functionality. However, for this particular project, and after some consideration, MooTools[10] was considered the best option. The reasons for this decision are:

Compact It has a low footprint on the site load because it is reasonably lightweight for the functionality it offers. Particularly, it is more optimized in this aspect than Prototype, YUI or Dojo, but it is also slightly more compact than jQuery.

Modular-Based Because of that, the installation can be customized to get only the modules we need, and the creation of our own extensions is easier.

Compatible It has been tested with most browsers: Internet Explorer 6+, Firefox 2+, Opera 9+, Safari 2+ (and other Webkit-based browsers, like Chrome).

Functional It offers all the functionality required for the first phase of the project: drag&drop, resize, animations, etc.

It also offers other functionality like AJAX support, `Hash` handling or `Cookie` handling, that ease the development in different browsers.

Object-Oriented By adding *Classes* to JavaScript, an abstraction that it is perfect for this application, since the server code is written is Java.

This way, we can use similar concepts both in the server and in the client. Moreover, the inherited code for ScaleNet already used JavaScript objects.

Extensive It also has a repository for official plugins called MooTools More (with similar code quality and documentation to the MooTools Core) and other third-party plugins can be found in the web.

Well-documented It has extensive documentation for every class of the framework.

Well-structured Its structure is perfect for a professional web application. Frameworks like jQuery are more focused in reducing the lines of

---

[10]http://mootools.net

code that in encouraging robust coding. MooTools also helps reducing the lines of code, but it has more tools for writing code in a very modular, reusable and robust way, for example by using classes and other abstractions.

It also improves the readability of the code, something hard to do in JavaScript. Another important point of this framework is that it is based on prototype extensions (mainly `DOM` extensions), so the syntax is very Object-Oriented and the code seems very clean.

Used by the APE server So if we use that component, it will be very straightforward to write extensions in JavaScript also in the server. This will mean that we could use the same coding style and the same tools in the server as in the client.

2.8   Push Server: the APE Server

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

2.9   Mobile Web Development

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate

velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

2.9.1 Touchscreens

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

2.9.2 Webkit

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Chapter 3

Development

WALTER :	Did you learn nothing from my chemistry class?
JESSE :	No. You flunked me, remember? You prick! Now let me tell you something else. This ain't chemistry —this is art. Cooking is art. And the shit I cook is the bomb, so don't be telling me.
WALTER :	The shit you cook is shit. I saw your set-up. Ridiculous. You and I will not make garbage. We will produce a chemically pure and stable product that performs as advertised. No adulterants. No baby formula. No chili powder.
JESSE :	No, no, chili P is my signature!
WALTER :	Not anymore.

Pilot

BREAKING BAD

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

3.1 How the Devices Are Placed

The first time that the user visits the page in a new browser, the system has to place the devices in the screen. Since the number and type of devices are different for each user, an algorithm must be used to placed the devices in the available space.

3.1.1 Simplified Algorithm

The restrictions that we have to comply are:

- We have an undetermined number of elements to place.

- For the sake of simplicity, the elements and the canvas are rectangular.

- Every element, including the canvas, has a different size.

- We have to place them in the most comfortable way possible, ideally using all the space we have.

The solution to the above problem is known as a variant of 2-D rectangle packing and it is, regrettably, NP-hard. At this point, it is clear that we need a simplification. Furthermore, the original problem also presents a big issue, as it does not address an important constraint: the possibility of resizing the elements to fit the canvas.

A simplified algorithm is proposed and implemented, relaxing some terms while obtaining acceptable results. The important concepts are:

- The main goal is to draw a virtual grid of M×N cells (like a table). Every cell can be a position for a device.

- Indeed, we are going to calculate the smallest grid of M×N cells in which the devices can be placed.

- Finally, we are going to resize the elements to fit within that cell, giving that every cell has the same size.

To obtain the squared grid for N elements, we simply have to calculate the ceiling of the squared root of N as seen in eq. (3.1).

$$Grid_x = \lceil \sqrt{N} \rceil \tag{3.1}$$

This formula synthesizes the idea that, given a certain number of elements N:

- If N is a square number (that is, it exists an integer x that fulfills $x^2 = N$), then you could fit a whole grid of $x \times x$ with N elements.

- Otherwise, this integer N must be between the square of two consecutive integers, that we are going to call x and $x + 1$. That is, $x^2 < N$ and $(x + 1)^2 > N$. In visual words, that means that a grid of size x cannot hold that number of elements, and a grid of size $x + 1$ can hold that number of elements but there will be empty *cells*.

- In that case, we choose the grid of size $x + 1$, this way we want to apply the ceiling function to the squared root to obtain the next following integer.

In the following figures we can appreciate what all of this means with real examples, using different number of elements.

(insert images with the disposition of different number of elements in the grid)

If we look at the examples below, we can guess an improvement without making the calculation severely complicate. We can see that, at certain points, a whole row of the grid is completely empty, so we are wasting vertical space. In theory, we could detect when this happens and try to reduce the vertical height of the grid by one, effectively converting this squared grid into a rectangle grid with different number of columns and rows.

Parting from an example: if we have $N = 19$, then we obtain $x = 5$ by applying the previous formula. This will lead us to a 5x5 grid, but the last row will be completely empty. The question that we have to ask ourselves is: how big has to be the rectangle grid in order to be able to place this number of elements? Briefly, the answer is $x \cdot (x - 1)$. That is a mathematical way of describing that we are decreasing the number of rows by one. In this case, we need a grid of 5×4 elements: that will hold up to 20 elements.

To discover whether we have to decrease the number of columns or not, we must compare that number $(x \cdot x - 1)$ with the actual count of elements. If we know that this number is bigger or equals to the number of elements, then we know that a grid of $x \cdot (x - 1)$ elements can hold those elements. On the contrary, if we know that this number is strictly lower than the number of elements, then we know that a grid of $x \cdot x$ is unavoidable. This can be formulated as in eq. (3.2).

$$Grid_y = \begin{cases} Grid_x - 1 & \text{if } N \leq Grid_x \cdot (Grid_x - 1), \\ Grid_x & \text{if } N > Grid_x \cdot (Grid_x - 1). \end{cases} \tag{3.2}$$

Another question appears: Do we need to shrink the grid only by one? Is there any case in which we have to shrink the grid by two or more?

The answer is no.

We can prove why not by calculating if a grid of $(x - 1) \cdot (x - 1)$ elements can hold more elements that the grid of $x \cdot (x - 2)$. If that is the case, then we would not need to shrink the grid in any case by two because the grid will be already horizontally shorter. It is quick to prove this is true following the steps explained from eq. (3.3) to eq. (3.5).

$$(x - 1) \cdot (x - 1) \; < \; x \cdot (x - 1) \tag{3.3}$$
$$x^2 - 2x + 1 \; < \; x^2 - 2x \tag{3.4}$$
$$1 \; < \; 0 \tag{3.5}$$

Then, using this final algorithm, the previous examples will change:

(insert images with the disposition of different number of elements in the grid using the final algorithm)

Using this algorithm we can calculate the height, width, vertical and horizontal offset for every element. If we want to work with percentages, we only have to divide the size of the container between the number of columns and rows. For example, if we want to fill the container at 100%, then every element will be of size $100/x \times 100/y$. The actual values for the offset needed for every particular element is then easy to calculate if we fill the canvas one by one.

3.1.2   Storage Positioning

Next time an user visits this page, the system will remember the last position of those elements instead of calculating the grid again. Because the user can change the size of the window at any time, we cannot relay on fixed positioning with pixels, because our canvas could be bigger or, worse, smaller than the one we have calculated. Besides being not very elegant, we can find several situations where the page is unusable.

The best way to avoid all that trouble is treating every position or size in terms of percentages. This is how is done in the code, and it allows the user to resize the window at any time: the devices will be resized dinamically according to that window size.

To store and retrieve painlessly these values, we are going to take advantage of an useful MooTools class: `Hash.Cookie` [3]. With this utility, we only have to specify the name for the `Cookie` and we can store a `Hash` into a `Cookie` without worrying about the `Cookie` itself. Besides loading the data of the `Cookie` directly on the `Hash` at its creation, if we change a value of the `Hash` it will be automatically updated in the `Cookie`.

The reason for using a `Cookie` is mostly because it reduces complexity on the server, since it does not have to store the position of every device. Other good reason is that it is the most simple way of allowing different arrangements in different places; for example the user may want to arrange radically different its devices in a big screen like in a TV or on a smaller screen like in a netbook. Finally, it is universal as it is supported by almost every browser.

The final decision is to have one `Cookie` for each device. This is very straightforward for the implementation, since a `Cookie` can have the name

of the container. Each `Hash` that is stored in every `Cookie` is composed by the four values needed for positioning the element: `offsetX`, `offsetY`, `width` and `height`. These values are percentages respect the container (the devices list) and an `Hash` example is presented in Listing 3.1.

Listing 3.1: Cookie Hash example

```
{
  offsetX: 15,
  offsetY: 50,
  height: 10,
  width: 20
}
```

Then, each time the object size or dimension changes, the `Hash` (and therefore the `Cookie`) is updated. These changes happen mostly in two situations: when we resize a device (changing its size but no its position) or when we move around a device (changing its position but not its size).

❧

3.2 APE Server Installation and Configuration

In this section the installation and configuration of the APE server are defined step by step.

3.2.1 Install the Server

The APE download page [4] contains packages for different operating systems and architectures. In this case, since the system is Debian-based we should use the DEB package. Once the correct package is downloaded, it can be installed on the Application Server by typing Listing 3.2 from the same directory as the package is stored.

Listing 3.2: APE installation command

```
sudo dpkg -i ape-1.0.i386.deb
```

After that, the APE server daemon (`aped`) is automatically started with the default configuration [5]. It can be checked by visiting the url `webportal.imusu.mobile.dtrd.de:6969`.

3.2.2 Configure BIND

The IMS core is the machine that provides the DNS service through BIND, and that service needs to be configured to allow the APE server to use a lot of different dynamic subdomains like `1.ape.webportal`, `2.ape.webportal`, `567.ape.webportal`, etc.

This is how the APE server works by default, and it appears that there is no way to configure the APE server for using only one domain [6].

So, in the file `/etc/bind/imusu.dnszone` located in the IMS core we have to look for the *webportal* entry and change that section to look like Listing 3.3.

Listing 3.3: BIND configuration

```
webportal          1D IN A       192.168.5.234
ape.webportal      1D IN A       192.168.5.234
*.ape.webportal    1D IN CNAME   ape.webportal
```

To apply the changes, we have to restart BIND using the command in Listing 3.4.

Listing 3.4: BIND restart command

```
sudo /etc/init.d/bind restart
```

Chapter 4

Discussion and Outlook

> **Fry** : But I know you in the future.
> I cleaned your poop.
> **Nibbler** : Quite possible. We live long and are
> celebrated poopers.
>
> *The Why of Fry*
> **Futurama**

4.1   Discussion

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

4.2   Outlook

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Appendix A

Budget

<div style="text-align: right">

DEXTER : It seems ironic that I, an expert on human
dismemberment, have to pay 800 dollars
to have myself virtually dissected.

The Lion Sleeps Tonight
DEXTER

</div>

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Appendix **B**

One More Thing

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea

commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Bibliography

[1] Matthias Siebert et al. *ScaleNet – Converged Networks of the Future*, April 2006. it - Information Technology, ISSN: 1611-2776, volume 48, pages 253–263. Also avaiblable online from:
http://telematics.tm.kit.edu/publications/Files/204/
it0605_253a.pdf. 2.2

[2] Dmitry Sivchenko et al. *ScaleNet – IMS Demonstrator*. Deutsche Telekom Laboratories, August 2008. Internal Document. 2.2.1

[3] MooTools. *MooTools Docs: Class Hash.Cookie*, Version 1.2.5. Last checked on February 2011.
http://mootools.net/docs/more/Utilities/Hash.Cookie. 3.1.2

[4] Weelya. *APE Server Download*, Version 1.0. Last checked on February 2011.
http://www.ape-project.org/download/APE_Server.html. 3.2.1

[5] Weelya. *APE Setup Manual*, Community Wiki. Last checked on February 2011.
http://www.ape-project.org/wiki/index.php/Setup. 3.2.1

[6] Weelya. *Advanced APE Configuration*, Community Wiki. Last checked on February 2011.
http://www.ape-project.org/wiki/index.php/Advanced_APE_
configuration. 3.2.2

Acronyms

AJAX   Asynchronous JavaScript and XML

APE   Ajax Push Engine (see §2.8)

BIND   Berkeley Internet Name Domain

CSS   Cascading Style Sheets

DNS   Domain Name System

FMC   Fixed and Mobile Convergence

GUI   Graphical User Interface

HTML   HyperText Markup Language

IMS   IP Multimedia Subsystem

IP   Internet Protocol

IPTV   Internet Protocol Television

MMOG   Massively Multiplayer Online Game

NGN   Next Generation Network

OSGi   Open Services Gateway Initiative

P2P   Peer-To-Peer

PHP   PHP: Hypertext Preprocessor

PNAI   Personal Network Administration Interface (see §2.2.3)

SIP   Session Initiation Protocol

T-Labs Deutsche Telekom Laboratories

VOD Video on Demand

VoIP Voice over IP

Index