

NXD

OPEN SOURCE DESKTOP WHITEPAPER

*From Linux Desktop to Virtual Desktop:
How AJAX and HTML5 Change Perspectives*

Jean-Paul Smets-Solanes



Draft Version Prepared by Nexedi for FCA
Improvements and contributions welcome: info@nexedi.com
Official FCA version to be released after review by members



Document Signature Table

Name	Function Company	Signature	Date
<i>Prepared by:</i> Jean-Paul Smets	CEO Nexedi		2011-01-20
<i>Approved by:</i> Jean-Paul Smets	CEO Nexedi		2011-01-20
<i>Released by:</i> Jean-Paul Smets	CEO Nexedi		2011-01-20



Distribution List

- Smets, J-P. Nexedi SA ■
- Brettnacher, T. Nexedi SA
- Tahara, Y. Nexedi KK
- Monerat, R. Nexedi STL
- Woelfel, K. Nexedi GmbH

Table of Contents

1 Introduction.....	7
1.1 Purpose and Scope.....	7
1.2 Document Overview.....	7
1.3 References.....	8
1.4 Abbreviations.....	9
2 Executive Summary.....	11
2.1 Make your own Evaluation and Share your experience.....	13
2.2 Early Conclusions.....	13
3 Reference Scenarii.....	15
3.1 Use Cases.....	15
3.2 Abstract Strategies.....	17
4 Real World Case Studies.....	19
4.1 Case 1: Financial Institution (Open Source Applications).....	19
4.2 Case 2a: French Parliament (Linux Desktop).....	20
4.3 Case 2b: Nexedi (Linux Desktop).....	20
4.4 Case 3: Data Publica (Gadget Desktop).....	21
4.5 Case 4: Generalitat de Catalunya (WebOS Desktop).....	21
5 Cost Comparison.....	23
5.1 Base Parameters.....	23
5.2 Cost of Proprietary Licenses.....	25
5.3 Distribution, Security and User management.....	26
5.4 Other Costs.....	27
5.5 Summary: Raw Cost vs. Net Cost.....	27
5.6 Variations.....	29
6 Desktop Application Development.....	32
6.1 Old School: Graphic Toolkits.....	32
6.2 New School: Javascript, HTML5, CSS3.....	33
6.3 Online IDEs.....	34

7 Conclusion.....	36
7.1 Enterprise Application Convergence and the Web Desktop.....	37
8 Appendix 1 - Open Source Virtual Desktop Products.....	39
8.1 Gadget Portals.....	39
8.2 WebOS.....	42
8.3 Remote Desktop Access.....	43
9 Appendix 2 - Open Source and Free AJAX Applications	46
9.1 Illustration.....	46
9.2 Imaging.....	47
9.3 Word Processing.....	47
9.4 Spreadsheets.....	48
9.5 Calendar.....	48
9.6 Mail Reader.....	49
9.7 WebOS Desktops.....	49
9.8 IDE.....	49
9.9 Languages.....	50
9.10 Others.....	51
10 Appendix 3 - Web Interoperability.....	53
10.1 Common Access to Files.....	53
10.2 Copy and Paste.....	54
10.3 Drag and Drop.....	54
10.4 Printing.....	54
10.5 Foreign Web Application inside WebOS.....	55
10.6 Multiple Gadget Providers.....	55
10.7 Single Login.....	56

1 Introduction

1.1 Purpose and Scope

Nexedi is the creator of ERP5 (www.erp5.com), an open source ERP, CRM and KM used by governments, multinational companies and SMBs. Nexedi provides ERP5 as SaaS through the TioLive platform (www.tiolive.com). As part of its Research & Development activities, Nexedi analyzes technology trends which may impact the future environment in which ERP5 will be used. As a member of the Free Cloud Alliance (FCA), Nexedi contributes to the advance of a Cloud Computing economy based on Freedom and Open Source.

In this whitepaper, we analyzed the trend in relation to the Desktop and its applications. Trends in relation to the Desktop can impact both the environment in which business applications such as ERP5 are used and the technologies which will be used in future business applications. This whitepaper tried to analyze real world case studies and not only abstract features because we believe that comparing features is never sufficient to compare the value of technologies. Only real world experience can tell what a technology is worth. Moreover, we tried to conduct our analysis in a way which is independent of any integration services which could be provided in relation with ERP5.

After peer review by FCA members, the whitepaper will be amended and updated, then released by FCA officially.

1.2 Document Overview

This document has 7 chapters:

- This chapter (Chapter 1) defines the goals, the content, the references and abbreviations.
- Chapter 2 is an executive summary of our whitepaper.
- Chapter 3 defines reference scenarii for the user of Desktop applications in a company and the abstract strategies which can be considered on the Desktop.
- Chapter 4 provides real world case studies of the implementation of the abstract strategies defined at Chapter 3.
- Chapter 5 explains the costing model which was developed as part of the whitepaper and compares the cost of abstract strategies defined at Chapter 3.
- Chapter 6 provides an overview of the mature and new development tools for the Desktop in relation with the strategies defined at Chapter 3.
- Chapter 7 contains the conclusions of our whitepaper.

The document has 3 appendices:

- Appendix 1: a list of open source applications for the virtual desktop.
- Appendix 2: a list of open source desktop applications based on AJAX technologies.
- Appendix 3: a list of interoperability issues in Web based Desktop environments and the solutions which are considered.

1.3 References

The section provides the lists of the applicable abbreviations.

1.3.1. Applicable Documents

ID	Title / Reference	Document Number	Issue/Rev.
AD-1	Open Source Cloud Computing – Free Cloud Alliance http://www.freecloudalliance.org/fca-Home/fca-Free.Cloud.Alliance?format=pdf	fca-Free.Cloud.Alliance	1

1.3.2.Referenced Documents

ID	Title / Reference	Document Number	Issue. Rev.
RD-1	The Widget Landscape - http://www.w3.org/TR/wid-gets-land/	N/A	Q1 2008
RD-2	Netvibes - http://www.netvibes.com/en	N/A	N/A
RD-3	Igoogle - http://www.google.com/ig	N/A	N/A
RD-4	YouOS - http://en.wikipedia.org/wiki/YouOS	N/A	N/A
RD-5	EyeOS - http://www.eyeos.org/	N/A	N/A
RD-6	Exo WebOS - http://wiki.exoplatform.org/xwiki/bin/view/WebOS/	N/A	N/A
RD-7	Onlineformapro OpenOffice http://www.onlineformapro.com/boutique.php?cat=18	N/A	N/A
RD-8	Qooxdoo - http://qooxdoo.org/	N/A	N/A
RD-9	ExtJS - http://www.sencha.com/products/js/	N/A	N/A
RD-10	jQueryUI - http://jqueryui.com/	N/A	N/A
RD-11	AmpleSDK - http://www.amplesdk.com/	N/A	N/A
RD-12	Ajax.org - http://ui.ajax.org	N/A	N/A
RD-13	ZK - http://www.zkoss.org/	N/A	N/A
RD-14	GWT - http://code.google.com/webtoolkit/	N/A	N/A
RD-15	YUI - http://developer.yahoo.com/yui/	N/A	N/A
RD-16	Pyjamas - http://pyjs.org/	N/A	N/A
RD-17	Sproutcore - http://www.sproutcore.com/	N/A	N/A
RD-18	Capuccino - http://cappuccino.org/	N/A	N/A
RD-19	PhoneGAP - http://www.phonegap.com/	N/A	N/A

1.4 Abbreviations

2 The section provides the lists of the applicable abbreviations.

ID	Abbreviation	Title	Description
AB-3	ERP	Enterprise Resource Planning	Enterprise resource planning (ERP) is a company-wide computer software system used to manage and coordinate all the resources, information, and functions of a business from shared data stores.[]
AB-4	CRM	Customer Relationship Management	Customer relationship management (CRM) consists of the processes a company uses to track and organize its contacts with its current and prospective customers. CRM software is used to support these processes; information about customers and customer interactions can be entered, stored and accessed by employees in different company departments. Typical CRM goals are to improve services provided to customers, and to use customer contact information for targeted marketing.
AB-5	DMS	Document Management System	A document management system (DMS) is a computer system (or set of computer programs) used to track and store electronic documents and/or images of paper documents. The term has some overlap with the concepts of content management systems. It is often viewed as a component of enterprise content management (ECM) systems and related to digital asset management, document imaging, workflow systems and records management systems.

ID	Abbreviation	Title	Description
AB-6	CMS	Content Management System	A content management system (CMS) such as a document management system (DMS) is a computer application used to manage work flow needed to collaboratively create, edit, review, index, search, publish and archive various kinds of digital media and electronic text.
AB-7	SaaS	Software as a Service	Software as a Service (SaaS, typically pronounced 'sass') is a model of software deployment whereby a provider licenses an application to customers for use as a service on demand. SaaS software vendors may host the application on their own web servers or download the application to the consumer device, disabling it after use or after the on-demand contract expires. The on-demand function may be handled internally to share licenses within a firm or by a third-party application service provider (ASP) sharing licenses between firms
AB-8	OLAP	Online Analytical Processing	Online analytical processing is an approach to quickly answer multi-dimensional analytical queries. OLAP is part of the broader category of business intelligence, which also encompasses relational reporting and data mining. The typical applications of OLAP are in business reporting for sales, marketing, management reporting, business process management (BPM), budgeting and forecasting, financial reporting and similar areas. The term OLAP was created as a slight modification of the traditional database term OLTP (Online Transaction Processing)

1 Executive Summary

GNU/Linux and open source software are now the dominant underlying technology of IT, for embedded device such as routers and mobiles phones, or for servers of Web and Enterprise applications. However, one area of IT has seen little progress in open source adoption for the last 10 years: the Desktop. A company such as Nexedi, where all employee use a GNU/Linux desktop with open source productivity, communication and graphic design software still remains an exception.

The disappointing adoption of open source Desktop also happens within open source communities themselves. In France, a country ranked by Red Hat study¹² most active in the world for open source, the open source independent integration market is dominated by 4 medium size companies. Until 2007, executives of those 4 companies were all using a proprietary desktop for daily activities, either based on Windows or MacOS. Nowadays, 50% of them has migrated to GNU/Linux desktop and open source email infrastructure. One of them however is still using MacOS and proprietary email infrastructure provided by Google.

Name	Company and Position	OS	Email Software	Date
Company A	COO	Windows	Thunderbird	2009
Company B	COO	Windows	Thunderbird	2009
		GNU/Linux	Thunderbird	2010
Company C	CEO	Windows	Thunderbird	2007

Name	Company and Position	OS	Email Software	Date
		GNU/Linux	Thunderbird	2010
Company D	President	MacOS	Google GMail	2009

Table 1. Desktop environment used by executives of leading Open Source companies in France at given date

The Paris Libre and Open Source Software (PLOSS) association is France's largest open source professional community in open source. The ratio of users of Linux desktop among members during PLOSS board meetings is not much higher than 10%, based on 2009 samplings. The adoption of Google proprietary email infrastructure is growing among PLOSS members at the expense of open source email infrastructure. This adoption is in striking contrast with the early days of open source business in France in 1996, when most if not all CEOs of open source companies were using GNU/Linux desktop and open source email infrastructure.

The situation found in France is not exceptional. Similar ratio can be observed in Japan or US, where open source still remains a market for server applications. We were recently reported that during a meeting of a Linux User Group in Connecticut (USA), no member was using GNU/Linux on his or her laptop.

At the same time, use of Linux desktop is growing, notably in government, education and for mobile applications, especially in countries such as Brazil and Russia. The Brazilian government ordered recently 1.5 million GNU/Linux PC based on Mandriva³ distribution. Yet, some true advocates of open source within Brazilian public administration, people who unquestionably helped Brazil to progress towards technology independence through open source, still use Windows as their default Desktop operating system. Whenever a majority of leaders of the very companies and organizations which shape the open source market do not use for themselves open source desktop, or do not use for themselves the tools which they advocate to their clients, there is little hope that GNU/Linux based open source desktop will be adopted quickly and widely by the market. The adoption of open source desktop will just take longer than expected in the 90s or require a different path than the one which was originally imagined

Rather than providing yet another comparison between Linux Desktop and

Windows Desktop or MacOS, this whitepaper analyzes practical alternatives which exist on the Desktop in the context of increasing adoption of cloud computing based collaboration tools and increasing adoption of mobility tools such as tablet PCs. Would it be for example possible to use an Android based Archos 101⁴ or a MeeGo based Wetable as a desktop PC for office applications, as a development platform for PaaS applications?



Figure 1. Could tablets be the future of open source Desktop ?

1.1 Make your own Evaluation and Share your experience

We try in this whitepaper to provide tools so that everyone can define an open source Desktop strategy suitable to his or her organization. Real word case studies analyzed in Chapter 4 show the possible adequation between a given approach (ex. open source on Windows) and an environment (ex. Banking industry). The spreadsheet provided at Chapter 5 is based on figures which must be adapted to each environment and can help defining the most cost efficient approach. A list of development tools is provided in appendix as an alternative to traditional desktop development libraries such as MDF, Cocoa, Qt, GTK, etc.

This whitepaper is going to evolve and we would really like you to include your contributions in order to raise the awareness on the benefits of open source Desktop. If you find interesting results or business cases, if you think that major tools are missing, please feel free to contact us with your own results (info@nexedi.com).

1.2 Early Conclusions

For the reader in a hurry, here are the conclusions.

A Web based Virtual Desktop - a.k.a. WebOS or Gadget Portal - is, together with a NetPC, 25% more cost efficient than a traditional GNU/Linux desktop which itself is 25% more cost efficient than proprietary desktop on a PC. The most cost-efficient desktop strategy nowadays is based on the Web rather than based on Linux or Open Source alone. Modern desktop strategies, which combine a Web based Virtual Desktop and legacy approaches based on Windows, Linux or MacOS are easier to implement than a full migration to GNU/Linux because they provide users with new tools without forcing them to abandon existing ones immediately.

Thanks to WebKit, a cross-platform library now used by nearly everyone in the industry except Microsoft, development environments for desktop and mobile applications are moving towards Web based technologies: HTML5, Javascript and CSS. All new developments of desktop applications, even those which include multimedia and 3D, should therefore adopt HTML5 based technologies, and leave away legacy toolkits such as Qt, GTK, Java Swing, Cocoa, MFC, etc.

Convergence between e-business, content management, business applications (ERP, CRM, etc.) and now desktop and mobile applications means that every enterprise application now becomes part of so-called corporate "Intranet". Relevant strategies must be considered by software publishers in order to take advantage of the new Web based Virtual Desktop paradigm.

2 Reference Scenarii

In order to compare the different approaches for the desktop, let us first define some reference scenarii, both in terms of use cases and in terms of abstract strategies.

2.1 Use Cases

We consider a company 1000 employee. Each employee uses a Desktop PC to do part or all of the tasks listed in the table bellow.

Category	Tasks
Group-ware	Access Web based applications provided on the Intranet Access public Web sites Read and write emails Chat with colleagues Manage their personal calendar Manage their personal tasks Share a calendar of events with colleagues Read documents published on the Intranet Play videos published on the Intranet Make phone calls Receive phone calls
Productivity	Author short documents such as letters, memos Author long documents such as audit report with references and indices Author simple spreadsheets with a few tabs, hundred of lines, simple formulae and not external data sources Author complex spreadsheets with dozens of tabs, thousands of lines, macros and external data sources Author text based presentations Author presentations with animations and illustrations
Multimedia	Author illustrations Author or modify pictures Author or modify videos

Category	Tasks
Special-ized	Use specialized proprietary Desktop software which version exists for Linux Use specialized proprietary Desktop software which only exists for Windows Use specialized proprietary Desktop software which only exists for MacOS

Some tasks need to be better defined, especially the concept of specialized proprietary software. For example, in many companies, business intelligence is implemented using sophisticated proprietary software such as “SAS Institute” analytical software. Such proprietary software has versions for Linux and Windows. There is no open source equivalent yet of this software with the same scope, efficiency and advanced user interface, although there are quite a few great open source business intelligence software which even surpass “SAS Institute” on some aspects (ex. PALO, Rapid-I, R-Project, etc.). It can thus be meaningless to force satisfied users of “SAS Institute” to migrate to open source business intelligence which is not going to make them satisfied. “SAS Institute” was chosen here only as an example which has the specificity of being available on Linux platform. We are not stating here that open source business intelligence is not relevant - it actually is very much - but that it is irrelevant and counter-productive in some cases to turn satisfied users into unsatisfied users.

Unlike “SAS Institute”, most specialized proprietary software are not available for Linux. For example, Desktop software used for the exchange of banking data is often available only for Windows. Desktop software for multimedia creation, such as post-production and compositing, can be available only for MacOS. Sometimes, hardware dongles such as USB keys are used to prevent installing such software on a virtual machine or on a Windows terminal server. There is therefore no other option in some cases than to run Windows or MacOS to benefit from such software.

2.1.1. Other Parameters

In order to simplify our study, we have deliberately ignored parameters which are closely related to the Desktop environment: software distribution, user management and security.

Software distribution consists of automating the upgrade of the operating system and the installation of Desktop software through network based tools. Many tools exist for that purpose. Some are open source (ex. LRS, Pulse, GLPI, etc.) whereas some are proprietary (ex. Landesk, Zenworks).

User management consists of managing the database of users and defining access rights for every user. This can be handled using operating system agnostic approaches, usually based on an LDAP directory (ex. MDS), or tightly coupled with the choice of Desktop OS (ex. Active Directory).

Security consists of protecting Desktop PCs from viruses and intrusions. Both proprietary and open source solutions exist to implement security policies in a company. Open Source Desktop OS are usually safer than Windows, both for technical and social reasons.

Software distribution, user management and security are usually much less costly to implement in a fully open source environment. However, the exact cost difference is difficult to determine and subject to controversy. Also, many companies already purchased software for software distribution, user management and security. Migrating their desktop environment will not write off those costs, at least in the short term. We have therefore chosen to ignore them in our base calculations and to provide a cost parameter as an option in the spreadsheet introduced at Chapter 5.

2.2 Abstract Strategies

We are going to compare 5 strategies: full proprietary, open source on Windows, Linux Desktop, Gadget Desktop and WebOS Desktop.

The "Full proprietary" strategy consists of implementing so-called "Industry Standard" Desktop software, usually provided by Microsoft and Adobe. The operating system is usually Windows and, for some activities such as multimedia creation, MacOS.

The "Open Source" strategy consists of implementing leading open source desktop applications on a proprietary Desktop OS, Windows or MacOS. Proprietary specialized applications are installed in the same way as in the "Full proprietary" strategy, ie. locally.

The "Linux Desktop" strategy consists of implementing leading open source desktop applications on Linux and to provide access to proprietary specialized applications installed on a terminal server through a kind of remote desktop protocol (RDP, VNC). However, this strategy can not be universal, since some multimedia applications require too much bandwidth to be used through RDP, and

some proprietary applications can not be used on a terminal server because of hardware protection. A certain share of users will thus keep on using Windows or MacOS on the Desktop.

The "Gadget Desktop" strategy is a big shift. It could also be named "the Web is the Desktop" or "Web based Virtual Desktop". However, there are more than one way to use the Web as a Desktop. One way is based on the concept of portal with gadgets, shown to the user as soon as he or she turns on his PC. This idea was introduced by Netvibes [RD-2], copied by iGoogle [RD-3] then by numerous open source frameworks. Word processor, spreadsheet, mail, etc. become web applications. Just like with the Linux desktop, proprietary specialized applications are accessed through some kind of remote desktop protocol, usually embedded into a Java applet or Flash. Just like with the Linux desktop, it can not be a universal strategy and a share of users will keep on local installation of software on Windows or MacOS.

The "WebOS Desktop" is the other form of a strategy in which "the Web is the Desktop". In this case, what is presented to the user through a browser is a screen which looks like a regular desktop. YouOS [RD-4] was probably the first WebOS, mostly implemented as an proof of concept. It was later copied by numerous other companies, including by EyeOS [RD-5] and Exo [RD-6] open source projects. Nowadays, most if not all Javascript frameworks provide their own "WebOS" sample implementation and the number of WebOS hosted services has exploded⁵. Just like with the Linux desktop, proprietary specialized applications are accessed through some kind of remote desktop protocol, usually embedded into a Java applet or Flash. Just like with the Linux desktop, this strategy can not be a universal. A share of users will keep on local installation of software on Windows or MacOS.

The table bellow provides typical software used on the Desktop in each abstract strategy.

	Proprietary	Open Source	Linux	Gadget	WebOS
OS	Windows	Windows	Linux	Any	Any
Browser	Internet Explorer	Firefox or Chrome	Firefox or Chrome	Firefox or Chrome	Firefox or Chrome
Productivity	MS Office	OpenOffice	OpenOffice	Gadget Portal	WebOS

	Proprietary	Open Source	Linux	Gadget	WebOS
Group-ware	Outlook	Thunderbird	Thunderbird	Gadget Portal	WebOS
Multi-media	Photoshop Illustrator Premiere PageMager	Photoshop Illustrator Premiere PageMager	Gimp Inkscape KDElive Scribus	Gadget Portal	WebOS

Table 2. Typical software used on the Desktop for each abstract strategy

Software is also required on the server side for Web based strategies. Typical examples of software are provided bellow.

	Gadget	WebOS
Open Source	EyeOS 2.X Exo Platform UNG Feng Office Group Office	EyeOS 1.x EyeOS 2.X Exo Platform
Proprietary	Google Apps ZOHO	MyBoo Jooce

Table 3. Typical software used on the server site for Web base Desktop strategies

3 Real World Case Studies

We introduce in this chapter real world business cases for each of the abstract strategies presented at the previous chapter. Rather than assessing the economic impact precisely, which anyway would be different in the case of your company and which as already changed between the time this paper was written and the time you read it, our goal here is to explain the motivations for each strategy.

Business cases provided in this chapter need to be more detailed and extended. However, this information is either difficult to obtained or covered by NDA. The first release of this whitepaper could not include some information protected by NDA. Please feel free to contribute better business cases if you know any - info@nexedi.com

3.1 Case 1: Financial Institution (Open Source Applications)

A large international financial institution decided in 2004 to move away from proprietary software to open source, including on the desktop. This institution consists of 2000 employee in 8 countries and about 30 offices, interconnected through a private Wide Area Network (WAN).

The first step consisted in replacing Microsoft Office, Lotus Note and Internet Explorer with OpenOffice, Thunderbird and Firefox, yet keeping a Windows Desktop based network. In order to create a strong incentive for employee to make this change, 2 measures were taken. First, employee who would adopt new open source software would be provided a new Desktop PC with a large Flat LCD panel. At that time, Flat LCD panels were still considered at that time as

unusual in many organizations. Second, online training for OpenOffice was provided to all employee using e-learning tools provided by OnlineForma Pro [RD-7].

It took about 2 years to complete the migration from proprietary desktop software to open source desktop software. Some difficulties came from certain proprietary Web applications, such as Oracle applications, which at the time would only run on Internet Explorer. This has changed now and Oracle applications operate in Firefox. Yet, from time to time, old versions of Internet Explorer could possibly be required to run certain Web based applications. Another application which could not be replaced with open source alternative is SAS Institute analytical software. Statisticians rejected alternative solutions such as R, the programming language used by most academic statisticians and by the French National Statistics Institute (INSEE). SAS Institute is thus still used.

The savings generated by the use of open source software on the desktop were tremendous and have been used to invest in the development of new applications based on Web based open source frameworks in order to replace existing desktop proprietary applications provided by third party vendors at a price considered as “too high” by this organization. The savings generated on the desktop have already been more than doubled by the replacement of 30% of those applications.

Due to the requirement to keep on running certain proprietary applications on Windows, the next step in the roadmap - ie. moving to the Linux Desktop - was never taken. Instead, the institution is now considering the Virtual Desktop option.

3.2 Case 2a: French Parliament (Linux Desktop)

French Parliament has finalized in 2009 the implementation of Linux Desktop on more than 1000 seats. This implementation is now successful⁶ and documented⁷. French Military Police has conducted similar implementation during the same period.

3.3 Case 2b: Nexedi (Linux Desktop)

Nexedi uses GNU/Linux desktop (Mandriva Linux) and enforces a policy “No Proprietary Software Allowed Here”. OpenOffice is used for productivity. Users have the choice for groupware between Web based email or any IMAP compatible mailer. More than 5 different IMAP mailers are used in the company, some of which are command line based and others are GUI based. Graphic design is achieved using Inkscape, Gimp, Scribus and KDenlive. VOIP uses Twinkle. Everything else is implemented using ERP5 open source ERP/CRM/KM.

Nexedi clients generally use Windows Desktop, proprietary productivity (MS Office) and proprietary groupware (ex. MS Outlook). No problems of interoperability have been experienced over the last 9 years of operation, with very few exceptions.

A Windows based PC was purchased 3 years ago because it was cheaper to purchase a PC than an official Windows license. This PC was used to file a report for the European Commission using a specialized proprietary software imposed by the European Commission. This PC is now used from time to time for tasks such as upgrading the operating system of mobile phones, testing interoperability. A Mac was also purchase to develop mobile applications for the iPhone since the iPhone SDK requires to use a Mac.

Three proprietary applications are used as an exception to the “No Proprietary Software Allowed Here”. The first one is Adobe Flash, which is required to implement multiplatform webmeeting and access many web sites. Open Source implementations of Flash are not yet compatible enough. The second application is Oracle JDK because many Java applications are not yet compatible with OpenJDK, the open source partial implementation of Oracle JDK framework. The third application is Skype because open source VOIP software such as Twinkle is not yet compatible with non routable networks of Nexedi clients.

The Nexedi case is the proof that the “No Proprietary Software Allowed Here” can be implemented in a company, on the Desktop as well as for accounting, ERP, CRM, multimedia. It is both cost efficient, easier to manage thanks to the Linux distribution built-in software deployment tools, and reduces security related problems. The savings generated by this approach are equivalent to 5 man year of R&D activities.

3.4 Case 3: Data Publica (Gadget Desktop)

Data Publica (www.data-publica.com) is a startup company which focuses on Open Data. Just like many startup software company, it has decided to adopt Google's closes source, proprietary solution “Google Docs” for email infrastructure and for office document management. Similar cases exist in Non Profit Organization with offices in more than 30 countries.

Sadly, we do not have yet a good example of Gadget Desktop based on open source software. Please feel free to provide one.

3.5 Case 4: Generalitat de Catalunya (WebOS Desktop)

EyeOS has implemented WebOS Desktop for the Catalonia regional government⁸. Specifically, the Governance Department, which includes the city of Barcelona and serves over 7.2 million people. This project, created for the Catalan Government uses an eyeos-based solution that organizes more than 400 facilities of 9 different types and through different eyeos applications. Every one from those 400 centers is currently using eyeos for their internal communication, but also to manage a wide variety of resources with the central administration.

4 Cost Comparison

This Chapter provides a cost comparison between the different Desktop alternatives. We have compared the four reference cases provided in Chapter 3. We explain here how the model was built and how it can be tweaked to the needs of your company.

4.1 Base Parameters

All calculations depend on 6 parameters: number of users, number of users of complex productivity software, number of users of specialized software, number of user of multimedia creation software, yearly cost of a NetPC and yearly cost of a standard PC.

The number of users here is the number of physical desktop environments provided in the company. We do not take into consideration here the case in which a single PC is used by multiple users. In our example, we have 1000 users, 1000 seats, 1000 PCs.

The number of users of complex productivity software is the number of users in the company who need to create long documents (ex. 100 page sales offer) or complex spreadsheets (ex. consolidated accounts). We consider in here that only Microsoft Office or OpenOffice can meet this requirement for now. However, this could change in the not so distant future with products such as PALO Suite which provides an Web based online spreadsheet capable of handling millions of lines, and with yet-to-be released online text editors capable of handling documents of more than 200 pages with cross-references and indices. In our example, we have 100 users of complex productivity.

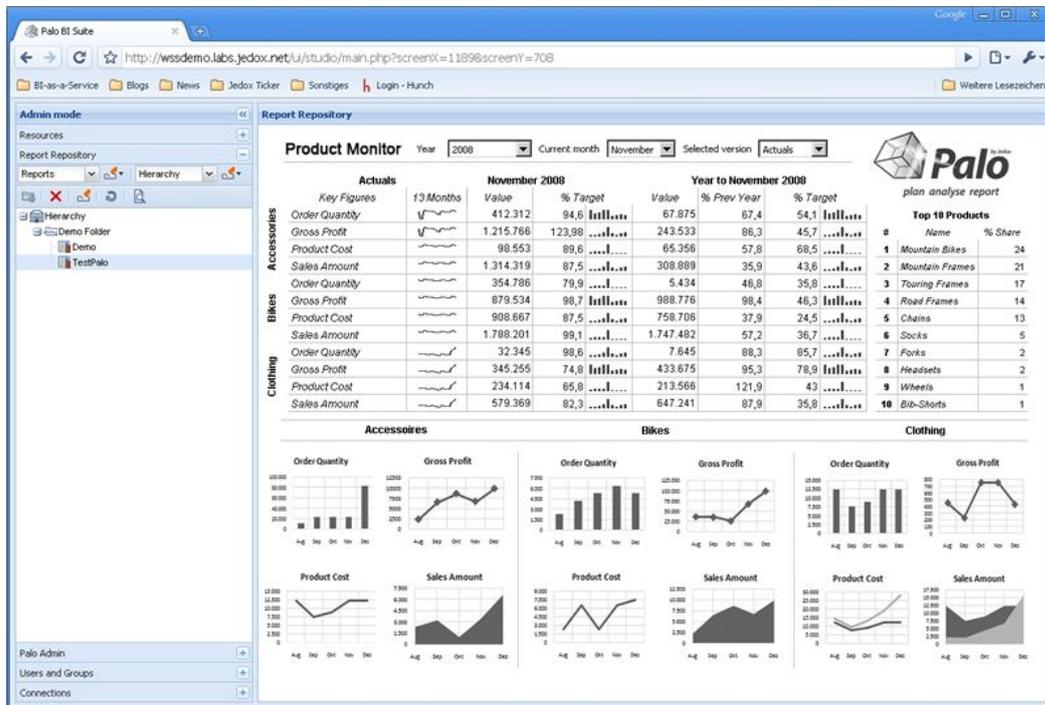


Figure 2. PALO Suite provides online spreadsheet and reporting better than a desktop spreadsheet

The number of users of specialized software is the number of users who need access to business applications to interface with their bank, or to technical applications to calculate the structure of a building, or to applications to prepare forms for an EU project. Such applications are very often Windows only. In our example we have 100 users of specialized software.

The number of users of multimedia creation software is, roughly, the number of users of Adobe applications used in the context of graphic design, music creation or video post-production. Such users are usually MacOS based. In our example we have 10 multimedia authors.

The total number of users is therefore equal to the number of users of complex productivity software plus the number of users of specialized software plus the number of users of multimedia creation software plus all the other users who only need to access the documents created by those three groups, and only need to do simple tasks such as small documents, small spreadsheets, email, access web applications, make phone call, etc.

The yearly cost of a PC is equal, here, to the price of the PC divided by 2. We consider here that PCs are changed every other year. So, in our example, we use

a value of 300 EUR for a laptop PC which has been purchased 600 EUR.

The yearly cost of a NetPC is equal to the price of the NetPC (ex. 220 EUR for a very nice Samsung NC10) with additional external LCD screen (ex. 100 EUR for a 19 inch screen) and keyboard. Server costs are then added. A dual CPU, quad-core server with 24 GB RAM costs less than 2500 EUR nowadays (Super-Micro brand for example) and is perfect to support 20 to 40 users. The total yealy cost is thus in out example about 200 EUR.

4.2 Cost of Proprietary Licenses

In the "Proprietary" scenario sheet, we define the yearly costs of various proprietary software. For example, we consider that proprietary operating system costs in average 20 EUR per year and per PC. This cost takes into account upgrades from one version to another and pre-installed versions.

Such costs are usually well known in large companies which benefit from corporate licenses with Microsoft which provide a kind of subscription to a portfolio of applications at a price which is either global or dependent on the number of seats.

For small companies, license costs is more difficult to assess. The operating system is considered as "Free" since it comes with the hardware and it is very hard to get it reimbursed. Operating system is seldom upgraded, since new hardware is usually purchased for a price which is cheaper than the upgrade itself. Consider for example the price of a NetPC and compare it withe the official price of Windows 7. The NetPC with Windows 7 can be cheaper than Windows 7 alone.

Other costs really depend on the prices granted by Microsoft, Adobe, IBM, etc. for their Desktop applications. The values we use derive from information – usually secret – which we could gather in large companies. For example, a 50 EUR / year cost for MS Office is much lower than what most small companies pay, but sometimes higher than what some large companies pay (ex. 10 EUR / year). Some organizations such as governments, hardware manufacturers or universities can even access Microsoft software portfolio for ridiculously low cost, close to zero, whenever there is a fear that they could move to open source.

The cost of multimedia creation software represents the cost for all Adobe cre-

ation software suite (Photoshop, Illustrator, CS3, Pagemaker, etc.). It is often implemented on MacOS hardware with large screen and powerful specs, which significantly increase the yearly hardware cost. We have taken into account the costs of such software because it will still take a few years before they become available as Web based applications. We have also considered in some scenarii that equivalent applications exist on the Linux desktop for multimedia creation.

The cost of specialized software depends on the environment and on the company. We did not take it into account in our evaluation because, whatever the scenario, the impact on the total cost will remain the same. If the specialized software is multi-platform (ex. "SAS Institute"), then the cost of using SAS will depend on the number of users of SAS in the company but not on the choice of desktop technology. And if the specialized software is not multi-platform (ex. SAGE Line 100), this will only impact the number of Windows desktop which must remain in the company in order to use SAGE. Moreover, specialized software is moving increasingly to the Web or to Web based servers. We have therefore set this cost to zero in the spreadsheet. However, this parameter can be adjusted (ex. To 1000 EUR) in the "Proprietary" sheet in order to exhibit a total cost of ownership (TCO) in the "Summary" sheet.

4.3 Distribution, Security and User management

In addition, on every sheet, a parameter named "Distribution, security and user management" can be adjusted to represent the external costs related to the distribution of software application, security such as anti-virus and intrusion detection, and user management such as user directories. This includes the license costs of implementing products such as Pulse⁹, Landesk, Zenworks, ec. It also includes the license costs of anti-virus (ex. Symantec, ClamAV, etc.), user directories (ex. MS Active Directory, MDS¹⁰). It also includes the cost of salaries of staff dedicated to the installation and upgrade of desktop PCs. It does not take into account however the cost of servers which are used to provide the service, because this cost does not depend linearly on the number of users.

The yearly cost of "Distribution, security and user management" must be adjusted for each company. In the "Proprietary" sheet we have set it to 100 EUR, which is a rather arbitrary and low value for a Windows environment. In the

⁹ <http://www.mandriva.com/enterprise/fr/produits/pulse-2-la-solution-open-source-de-gestion-de-parcs-heterogenes>
¹⁰ <http://mds.mandriva.org/>

"Open Source" sheet, we have include some savings which can be obtained through the use of open source directory, open source anti-virus and open source software distribution. In the "Linux" sheet, we have considered even higher savings due to easier software distribution and security using GNU/Linux. In the "Gadget and WebOS" sheet, we have considered that NetPCs do not need "Distribution, security and user management" and that Windows based PCs are managed using the same approach as in the "Proprietary" sheet.

4.4 Other Costs

The use of different operating systems or different approaches for the Desktop in a company can lead to costs related to heterogeneity of document formats. However, as long as a company has a good document management system, such as ERP5's Document Management System (DMS), documents created by one group of users in one format (ex. ODT) become automatically available to all other groups in any format (ex. DOC, HTML, PDF), thanks to the universal conversion engines present in the document management system.

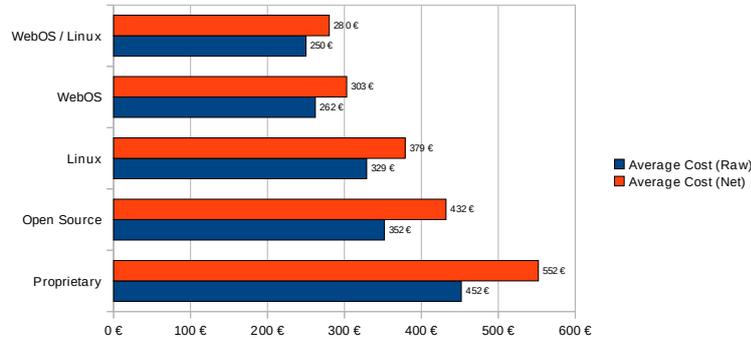
We therefore consider that heterogeneity costs, which used to exist 15 years ago at a time when MacOS, Unix and Windows networks could not easily interconnect, do not exist any longer in the scenarii we have defined, thanks to the widespread adoption of the Web, to the ubiquitous use of Internet standards, and to near-perfect interoperability between Linux, MacOS and Windows. Modern information systems are now designed to handle heterogeneity without extra costs rather than trust a single software provider for everything.

Our point of view could be of course challenged, by changing the value of "Other Costs" in our model and looking at the results.

4.5 Summary: Raw Cost vs. Net Cost

All costs are summarized in the first sheet. The first line shows the average raw cost per user, taking into account the hardware and license costs only. The second line shows the average cost per user, taking in addition the cost of "distribution, security and user management" and other costs such as heterogeneity.

	Proprietary	Open Source	Linux	WebOS	WebOS / Linux
Average Cost (Raw)	452 €	352 €	329 €	262 €	250 €
Average Cost (Net)	552 €	432 €	379 €	303 €	280 €
Gain (Raw)	0%	22%	27%	42%	45%
Gain (Net)	0%	22%	31%	45%	49%



Users	1000
Complex Users	100
Specialized Users	100
Multimedia Authors	10
NetPC Cost (incl. Server)	200 €
PC Cost	300 €

Figure 3. Costs of Desktop strategies

The raw savings which can be obtained by using open source on proprietary OS is about 22%. Migrating to the Linux desktop extends raw savings to 27%. Adopting the "WebOS & Gadget" approach extends raw savings to 42% or even more, by combining this with the "Open Source" approach.

The net savings which can be obtained by using open source on proprietary OS is about 22%. Migrating to the Linux desktop extends raw savings to 31%. Adopting the "WebOS & Gadget" approach extends net savings to 45% or even more, by combining this with the "Open Source" approach.

The small difference, from a raw savings point of view, between Linux Desktop and open source on Windows probably explains why few companies have migrated to the Linux Desktop. The cost of Windows licenses in a GNU/Linux Desktop environment are somehow higher because Windows is often installed using so-called Terminal Services on a central Windows Server. In addition, purchasing hardware without pre-installed operating system has become nearly impossible in Europe, Japan and US, where governments are not doing their job to prevent abuse of dominant position on the Desktop PC market. As a result, Linux Desktop means usually paying the Windows license twice: once on the laptop and once of the Windows Server which provides Terminal Services. Few companies can follow the approach of Nexedi described in Chapter 4, where a single Windows PC is shared by all staff who need to use at most once a year. The situ-

ation might be different in countries such as Brazil where pre-installed Linux PCs amount for 10% (incl. 5% Mandriva) of the market thanks to mass adoption in education and strong government policies.

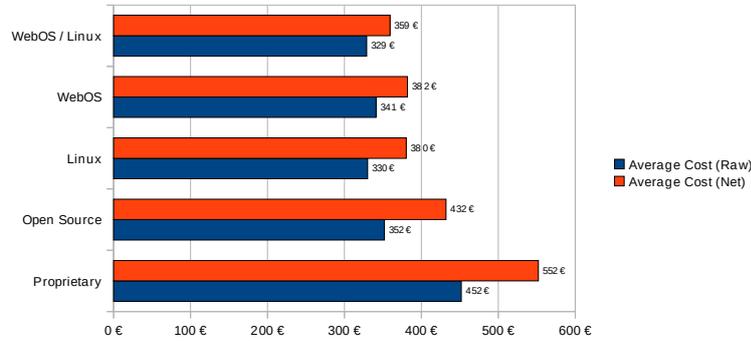
If we consider the net savings, Linux Desktop becomes much more appealing. However, many IT managers are afraid of heterogeneity costs and do not believe in the fact that management of networks of Linux based PCs is much more cost efficient than management of networks of Windows based PCs. They have therefore been reluctant in the last 10 years to try and save an additional 10% with Linux desktop. Yet, Linux Desktop is a reality and can save a lot of time and money as we have seen in the previous chapter.

The savings which the "WebOS & Gadget" approach provides are tremendous. Thanks to the lower cost of NetPCs, a massive 42% raw savings can be reached, compared to 22% with open source. Net savings are as high as 45%. Clearly, this is the way to go nowadays, either in combination with Windows desktop, or in combination with Linux desktop. Adoption of Linux desktop provide an extra 4% savings.

4.6 Variations

Part of the savings of the "WebOS & Gadget" approach relate to the choice of a 200 EUR NetPC (400 EUR incl. Extra) instead of a standard business laptop (600 EUR). What happens if we use a standard business laptop in this scenario?

	Proprietary	Open Source	Linux	WebOS	WebOS / Linux
Average Cost (Raw)	452 €	352 €	330 €	341 €	329 €
Average Cost (Net)	552 €	432 €	380 €	382 €	359 €
Gain (Raw)	0%	22%	27%	25%	27%
Gain (Net)	0%	22%	31%	31%	35%



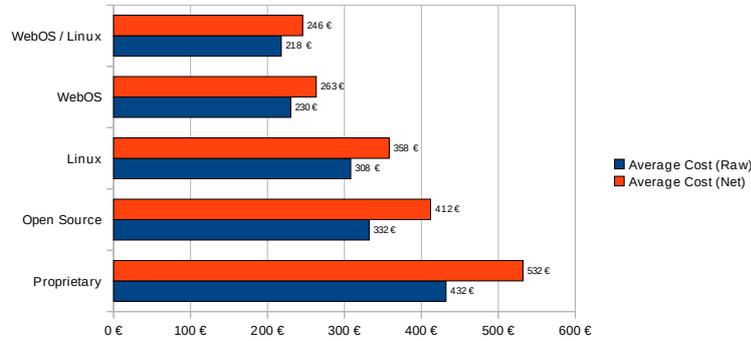
Users	1000
Complex Users	100
Specialized Users	100
Multimedia Authors	10
NetPC Cost (incl. Server)	300 €
PC Cost	300 €

Figure 4. Summary of costs if standard PCs are used instead of NetPCs

Net savings (31%) then become equivalent to the Linux desktop scenario. Raw savings (25%) are a bit lower than the Linux desktop scenario. This shows that the "WebOS & Gadget" is worth considering, even without a partial migration to NetPC. This also shows that savings provided by the "WebOS & Gadget" are immediate.

The choice of GNU/Linux Desktop can be more efficient for a software development company. Let us check what happens to our model if we have no specialized software which requires windows, and thus no heterogeneity. The net savings provided by Linux desktop reach 33%, compared to 23% for Open Source on Windows. The use of WebOS approach provides in addition extra net savings (37%) by reducing management costs of the network.

	Proprietary	Open Source	Linux	WebOS	WebOS / Linux
Average Cost (Raw)	432 €	332 €	308 €	230 €	218 €
Average Cost (Net)	532 €	412 €	358 €	263 €	246 €
Gain (Raw)	0%	23%	29%	47%	50%
Gain (Net)	0%	23%	33%	50%	54%

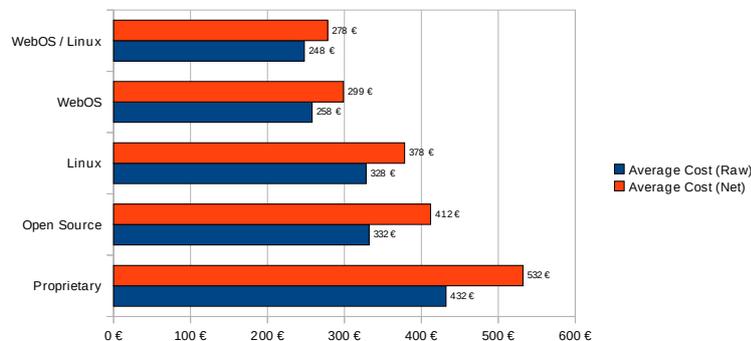


Users	1000
Complex Users	100
Specialized Users	0
Multimedia Authors	10
NetPC Cost (incl. Server)	200 €
PC Cost	300 €

Figure 5. Summary of costs if no specialized software is needed

Let us consider now that the cost of Windows is 0, since there is no way to be reimbursed. In this case, the Linux Desktop scenario provides nearly no raw savings (24%) compared to the Open Source scenario (23%). However, the WebOS scenario is still worth considering thanks to the raw savings provided by the use of NetPC.

	Proprietary	Open Source	Linux	WebOS	WebOS / Linux
Average Cost (Raw)	432 €	332 €	328 €	258 €	248 €
Average Cost (Net)	532 €	412 €	378 €	299 €	278 €
Gain (Raw)	0%	23%	24%	40%	43%
Gain (Net)	0%	23%	29%	44%	48%



Users	1000
Complex Users	100
Specialized Users	100
Multimedia Authors	10
NetPC Cost (incl. Server)	200 €
PC Cost	300 €

Figure 6. Summary of costs if no Windows license is "same as zero"

5 Desktop Application Development

A Desktop strategy should take into account the development of specialized desktop applications in the company. This Chapter provides an overview of the options available for the development of desktop applications.

5.1 Old School: Graphic Toolkits

Until a few years ago, development of desktop applications required to select a development toolkit and a development language to build an application which could be used exclusively on the desktop. A plethora of development toolkits currently exist for that purpose.

The choice for developers has often been to decide between proprietary toolkits designed for a single platform (ex. MFC, Cocoa, Visual Basic), proprietary multi-platform toolkits (ex. 4th Dimension, Rebol Command, Oracle Java) and multi-platform open source toolkits (ex. Qt, WxWidgets, GTK, OpenJDK, etc.).

Without any doubts, we can claim that nowadays, multi-platform open source toolkits are perfectly suitable for the development of about any desktop application. Qt¹¹, an open source toolkit now owned by Nokia, can be programmed using C++, Python, Perl, Ruby and 10 other languages. It has been used to develop business applications, mobile applications, multimedia applications for video processing and even real time 3D applications.

Java has also proven to be a nice multi-platform toolkit for the development of desktop applications, although compatibility issues remain between the proprietary version now owned by Oracle and the open source versions known as

¹¹ http://en.wikipedia.org/wiki/Qt_%28framework%29
tmpau1xbjz_.odt

OpenJDK, especially for Multimedia and rendering.

5.2 New School: Javascript, HTML5, CSS3

A technology, which emerged from the KDE community and which was originally implemented using Qt before being ported to other graphic toolkits by Apple has completely changed the perspective of Desktop application development: WebKit. Together with the improvements of Javascript, HTML and CSS, traditional toolkits such as Qt or Java have become in our opinion totally irrelevant for the development of modern desktop applications for the enterprise.

This fact may be very shocking to most developers, and even to ourselves, especially considering the known deficiencies of the Javascript language and its reputation for poor performance. However, this is no longer the case. Thanks to the application of just-in-time compilation techniques, Javascript interpreters are now fast enough to implement... 2D animation software with Ajax Animator¹², native 3D graphics¹³ or through extensions such as O3D¹⁴ and Canvas3D,¹⁵ application server with POW¹⁶.

The kind of frenetic development of new widgets and concepts, which 10 years ago happened through the graphic toolkits such as Qt, GTK, Cocoa, etc. has now moved to the HTML5 world. A plethora of Javascript frameworks provides the same level of development environment as what has been found until now with graphic toolkits. Have a look at Tibco, a famous proprietary software company in the banking industry, and their General Interface¹⁷, a complete IDE written in and for Javascript. Take a look at Qooxdoo [RD-8], ExtJS [RD-9], JQueryUI [RD-10], AmpleSDK [RD-11], Ajax.org [RD-12], ZK [RD-13], GWT [RD-14], YUI [RD-15], Pyjamas [RD-16], Sproutcore [RD-17], Capuccino [RD-18]. They all provide toolkits which go far beyond than Qt or GTK in terms of richness, community, easyness.

And if you still believe that traditional toolkits are required for multimedia and 3D, have a look at Titanium¹⁸. This toolkit uses and extended version of WebKit which integrates all the extensions of HTML5: OpenGL, video, etc. Titanium can

¹² <http://osflash.org/ajaximator>
¹³ http://www.nihilogic.dk/labs/canvas3dtexture_0.2/
¹⁴ <http://code.google.com/intl/fr-FR/apis/o3d/>
¹⁵ <http://www.c3df.org/>
¹⁶ http://davidkellogg.com/wiki/Main_Page
¹⁷ <http://www.generalinterface.org/>
¹⁸ <http://www.appcelerator.com/>

be used to develop multimedia rich desktop applications and mobile applications in HTML5 and Javascript, which look like a desktop application and act as a desktop application.

What all this means is that WebKit, and to some extent Gecko, have become the universal graphic toolkit of modern desktop application development. WebKit unifies web applications, which now look exactly the same as a desktop application, and desktop applications, which are now based on pure Web technology. A Web framework such as Pyjamas Desktop¹⁹ can be used to develop pure web applications. And PhoneGAP [RD-19] now provides cross platform development of mobile applications using... HTML and Javascript.

HTML5 and CSS3 have become the universal graphic language which supersedes GDI (Windows), Carbon (MacOS) and X11 (Linux). WebKit is its most widely adopted implementation, on the desktop and on mobile phones, followed by Firefox and Opera. Development of new desktop applications should therefore focus from now on HTML5 based technologies: faster to implement, easier to distribute.

5.3 Online IDEs

Development environments are not an exception in this evolution of the Desktop towards the Web. The same phenomenon applies to development environments. Using a local PC to develop applications is no longer the only way to achieve good results results quickly, and could even become the least efficient way very soon.

Many open source developers now use... proprietary desktop such as Windows and MacOS. Although no precise figures exist at the time of this whitepaper, the ratio of open soure developers using proprietary desktop, instead of decreasing, has very likely increased in the last 10 years. Such developers still need a Linux platform to develop, and very often use a virtual machine to run GNU/Linux inside their proprietary OS, thanks to the now wide availability of high performance virtual machines (QEMU, KVM, VirtualBox, VMWare). One of the nice features provided by virtual machines is that they act as a sandbox and can simplify development on multiple projects in parallel, each of which with different Linux distribution and different version of core libraries. One could therefore ar-

gue that using hosted virtual machines (ex. NiftyName, GANDI, Rackspace) and NetPCs actually provides a development environment which is as good if not better than the one of a personal laptop thanks to the built-in backup and collaborative nature of a hosted environment.

At the same time, Web based development, which was introduced as early as in 1998 by open source Zope²⁰ project, is gaining higher acceptance. The time when developers considered that a development environment which is not text and local file based could not be a true development environment, has passed, even if such belief remains strong among open source developers. Platforms such as RhoHub²¹ give a good example of what an online Web based development environment looks like. To a smaller extent, Github²² is the proof that providing online development services can drastically simplify the life of corporate developers. The nice aspect of online Web based development environment is that they reduce the setup time of a new project, by providing everything to start working and removing the need for installation and configuration of software. Another interesting aspect is that they help develop in a collaborative way. Last but not least, online development environments come with a plethora of tools, extensions and interfaces to simplify development. Online mobile application development environments provide for example testing connectors for about any mobile phone. Some online development environments provide a way to distribute unit tests execution over a cluster of virtual machines in order to get results faster. All that would be impossible on a local laptop²³.

The biggest proof that Online development is now widely accepted is after all Salesforce.com²⁴. Salesforce.com not only shows that online development can be useful and successful, generating turnover for thousands developers worldwide. It also shows that consultants can now act as developers of business applications, thanks to the introduction of graphic development environments which drastically lower the learning curve to create custom business applications.

So, no doubt: the future of desktop application development is online, collaborative and HTML based.

²⁰ <http://www.zope.org>

²¹ <http://rhomobile.com/products/rhohub/>

²² <http://github.com/>

²³ <http://sebastien-arbogast.com/2009/02/03/we-dont-need-an-online-ide/>

²⁴ <http://www.salesforce.com/platform/>

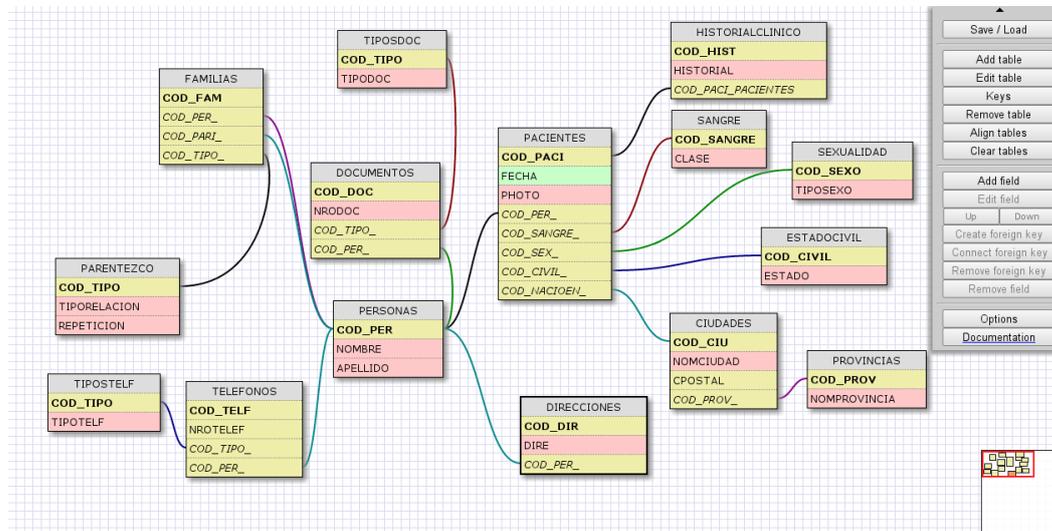


Figure 7. wwwsql designer, a Javascript only SQL schema editor with support for CUBRID database

6 Conclusion

We initially started to write this whitepaper after we found that the vast majority of CEOs of open source companies in France, the most active country for open source according to Red Hat, use... proprietary desktops. We also discovered that some migrations projects to Linux desktop in large companies failed simply because CEOs of large companies, who often meet high rank Microsoft executives, seldom believe that migration is worth the effort and fear conflict with a powerful company.

What we found by writing this whitepaper is that the raw savings provided by migrating to the Linux desktop are small, especially whenever pre-installed Linux PC are more expensive than pre-installed Windows PC due to the lack of political will to enforce competition in Europe, US and Japan. The net savings provided by the Linux desktop are significant but they are difficult to assess, especially considering the fear of heterogeneity which is present in IT services of large companies, and which is often overestimated. Linux desktop will thus likely remain an option only for clever and fearless small companies, which can greatly benefit from it. It could also remain an option in countries with a political will to enforce free competition on the Desktop market, such as Brazil or Russia.

Nexedi as a company, will keep on using Linux desktop both on its laptops and NetPCs, because GNU/Linux saves money and makes our staff more efficient. At the same time, Nexedi will refrain from promoting the Linux desktop to its clients in Europe, US and Japan. Instead, we shall promote another approach: the Web Desktop. By leveraging NetPCs and modern HTML / Javascript technologies, the Web Desktop provides up to 49% cost savings compared to a traditional desktop environment and 28% savings compared to a traditional GNU/Linux Desktop. The Web Desktop can be implemented without any change to the

current Desktop environments of large companies, without raising fears of heterogeneity.

6.1 Enterprise Application Convergence and the Web Desktop

The Web Desktop is also in line with the current convergence between business applications : ERP, CRM, e-business, content management, KM. Modern ERPs such as ERP5 are Web based and include document management, CRM, e-commerce, e-procurement, etc. CRM platforms such as Salesforce.com provide numerous extensions beyond pure CRM and are moving towards ERP. E-commerce applications such as Magento provide backoffice features similar to those of an ERP. Content managements systems such as Drupal provide workflows and e-commerce extension. KM platforms, in particular search engines such as Exalead, pretend that they can replace an ERP.

This convergence can also be found in the ever wider meaning of the word "Intranet". "Intranet" now implies for some opinion leaders that all applications in a company should be available "through the Intranet". "Intranet" is now used as a generic term to cover all Web based applications in a company, not only the internal communication media.

The outcome of enterprise application convergence is still unknown. An ERP such as ERP5, with a clean, simple data model and a cloud computing friendly database technology which does not require data migration has a clear advantage over e-commerce platforms or content management platforms which require a lot of of tables and data migration management. However, ERP5 still has a lot of progress to achieve in terms of user interface and faster learning curve. Yet, ERP5 with its built-in Web and Gadget frameworks, with its "Through The Web" development interface, is a great platform for the Web Desktop, either alone or integrated to another platforms.

With the increasing acceptance of Web based enterprise applications, the notion of Desktop as we know it today, is being silently superseded by a Web based Virtual Desktop. Thanks to powerful Javascript toolkits, development of desktop applications such as a spreadsheet, cash-register, video player, illustration package has now become easier than ever using Web technologies. Web based desktop applications, unlike applications developed with legacy toolkits, perfectly in-

tegrate in the new approach of the Web based Virtual Desktop.

Adopting the Web Desktop, though Firefox, through WebKit, using Linux, Windows or MacOS, is thus our key recommendation for a Desktop strategy for the next 10 years, in conjunction with the increased use of NetPCs. All Desktop and mobile application development should move now to HTML5 and possibly Javascript, either through native Javascript toolkits (ex. ExtJS, PhoneGAP, JQuery, Qooxdoo, AmpleSDK, etc.) or using Javascript translators (ex. ZK, GWT, Pyjamas). Existing Web applications should adopt new user interfaces, taking inspiration in particular from EyeOS, the mosy widely spread open source Web Desktop, which now combines in the same environment, the concept of Web OS and of Gagdet Portal.

7 Appendix 1 - Open Source Virtual Desktop Products

This Appendix provides a list of solutions for Virtual Desktop. Gadget portals, WebOS, remote virtual desktops are listed. Solutions are sorted by alphabetical order. We provide no recommendation here. If you need a recommendation, please contact us (info@nexedi.com).

7.1 Gadget Portals

Gadget portals are as old as NetVibes (2005). They are a now a mature approach to the Web based Virtual Desktop. They are used to provide users with a dashboard of tasks and informations with direct access to applications through one click.

7.1.1.Exo - www.exoplatform.com

Exo combines document management, portal and WebOS in a single platform. It is programmed using the Java language. It uses its own Javascript library. It includes groupware applications such as email reader. Exo is implemented on the proprietary Oracle Java platform. Although version 2.x is open source, version 3.x seems to be proprietary.

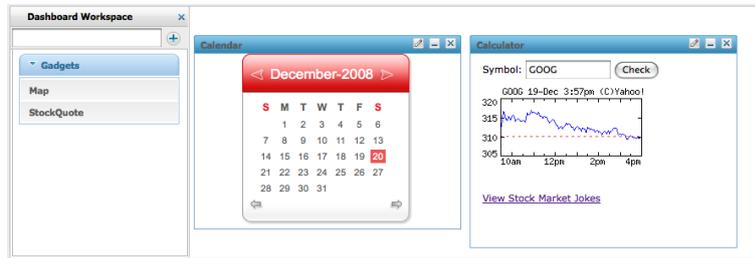


Figure 8. Exoplatform dashboard and gadgets

7.1.2. EyeOS 2.x – www.eyeos.com

EyeOS now includes a gadget portal mode in addition to the WebOS mode. EyeOS is programmed using the PHP language. It now relies on the Qooxdoo Javascript library. It includes numerous applications and in particular a spreadsheet, a calendar, email reader, word processor, instant messaging. Collaboration features to author documents in group are now built-in applications. EyeOS is implemented using open source LAMP platform.

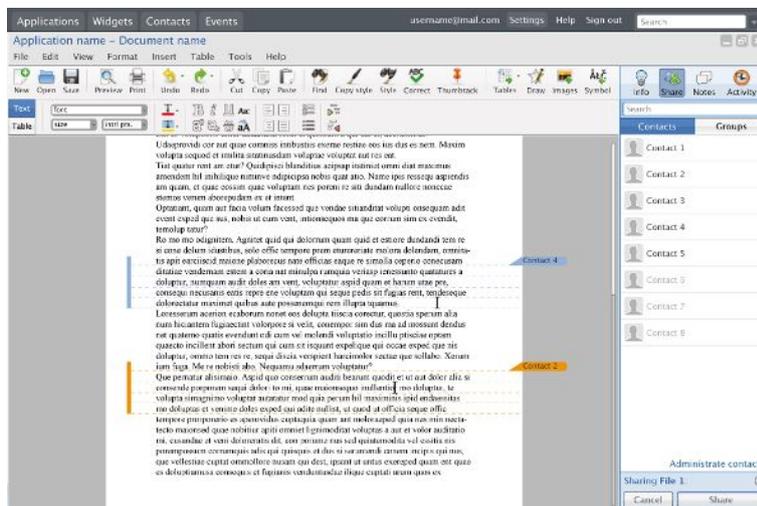


Figure 9. EyeOS spreadsheet screenshot with collaboration toolbar in action

7.1.3. Feng Office – www.fengoffice.com

Feng Office, previously known as OpenGoo, is an open source portal platform which has been provided in SaaS mode to thousands SMBs worldwide. It provides groupware such as email, calendar, tasks, document authoring. It is developed using the PHP language. It relies on its own Javascript library. It is imple-

mented on an open source LAMP platform.

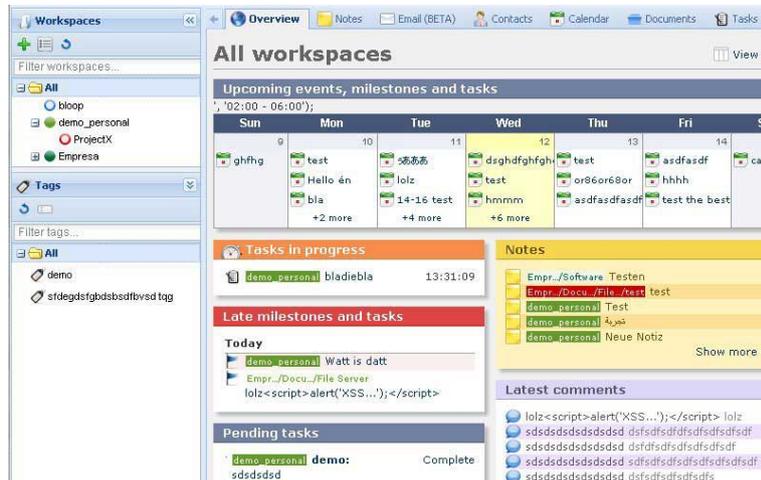


Figure 10. FengOffice Screenshot

7.1.4. Group Office – www.group-office.com

Group Office is a groupware solution with primary focus on email, calendar and file sharing. Proprietary extensions provide CRM and synchronization with iPhone and other platforms. It is provided in downloadable mode or in SaaS mode (like Feng Office). It is developed using the PHP language. It relies on the ExtJS Javascript library. It is implemented on an open source LAMP platform.

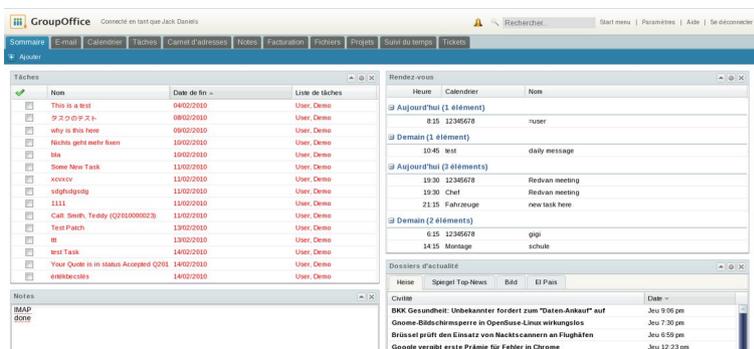


Figure 11. Group Office Portal Screenshot

7.1.5. UNG – www.freecloudalliance.org/ung-Home.Page

UNG is an open source implementation of an online office suite with text pro-

cessing, spreadsheet and illustration. It is based on community frameworks. UNG development is currently related to ERP5 but will be refactored in 2011 to support any document management backend. Calendaring and email are being implemented and will be released in Q2 2010.

7.2 WebOS

WebOS are as old as YouOS²⁵ (2005). They have not been adopted by Google unlike Gadget Portals. However, they seem to be very well accepted for Enterprise applications. The latest version of EyeOS also seems to show that Gadget Portals and WebOS are actually the same concept presented through different skins.

7.2.1.Exo - www.exoplatform.com

Exo combines document management, portal and WebOS in a single platform. It is programmed using the Java language. It uses its own Javascript library. It includes groupware applications such as email reader. Exo is implemented on the proprietary Oracle Java platform.



Figure 12.Exoplatform WebOS

7.2.2.EyeOS 2.x - www.eyeos.com

EyeOS is one of the first open source WebOS. EyeOS is programmed using the PHP language. It now relies on the Qooxdoo Javascript library. It includes nu-

²⁵ <http://en.wikipedia.org/wiki/YouOS>
tmpau1xbjz_.odt

merous applications and in particular a spreadsheet, a calendar, email reader, word processor, instant messaging. Collaboration features to author documents in group are now built-in applications. EyeOS is implemented using open source LAMP platform.

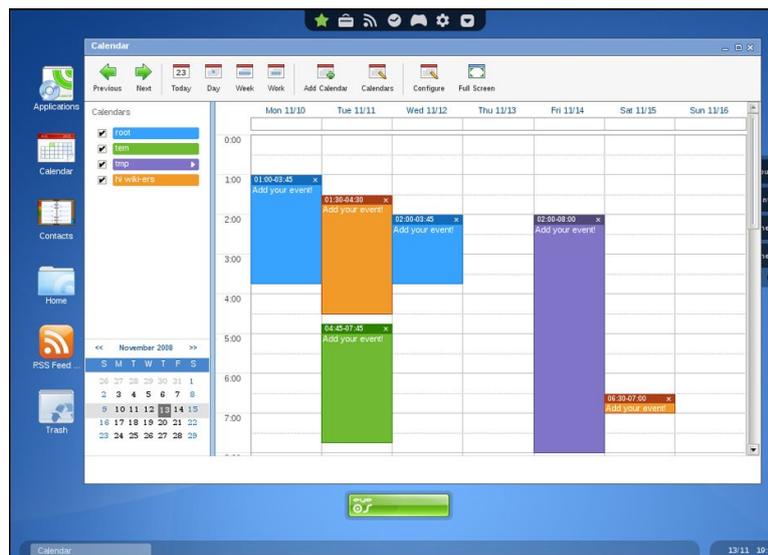


Figure 13. EyeOS WebOS screenshot

7.2.3. Lucid Desktop - www.lucid-desktop.org

Lucid Desktop is the Web Desktop application of the Dojo foundation.

7.3 Remote Desktop Access

Migration to the Virtual Desktop still requires access to legacy applications running on Windows, Mac OS or GNU/Linux. Various technologies exist for that purpose.

7.3.1. Ulteo - www.ulteo.com

Ulteo Virtual Desktop is a Virtual Desktop solution for legacy application available on Windows and GNU/Linux. Thanks to a Java applet, Ulteo provides through a standard Web browser such as Firefox access to any application running on a central Windows server or on a central Linux server.

It is a very interesting solution to keep compatible with existing software run-

ning on Windows or Linux, or to extend the software library of Web based Virtual Desktop with legacy software.

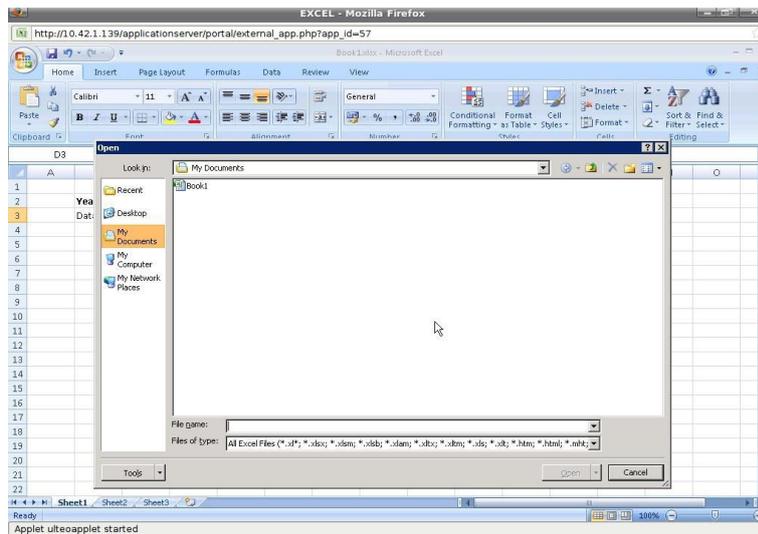


Figure 14. Excel running as a virtual desktop application inside Firefox browser

7.3.2. FlashLight VNC - www.wizhelp.com/flashlight-vnc

Flashlight-VNC is a VNC viewer and player. It can be used to access a VNC server and to play recorded VNC sessions. It can be useful for a single user to access, through a Web browser, a legacy application running on a legacy PC with a legacy Windows version. Other applications include sharing recorded screen captures for e-learning purpose or for technical support.

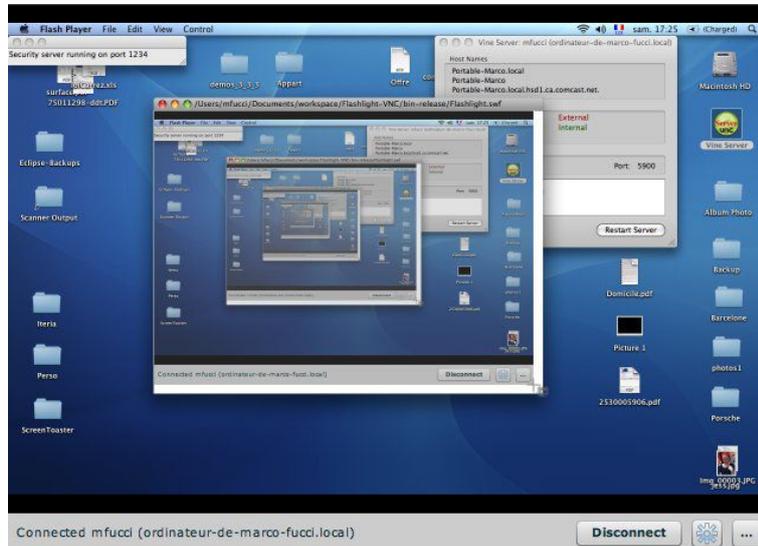


Figure 15. Flash-Light VNC screenshot

7.3.3. Other VNC Implementations

More VNC implementations are now available, not to mention RDP. The most recent implementations of VNC seem to use pure HTML / Javascript instead of a Flash client of a Java applet. It is the case of EduVNC (www.eduvnc.com) which is based on A-VNC (code.google.com/p/a-vnc). It is also the case of AjaxVNC (code.google.com/p/a-vnc). Together with approaches developed in solutions such as RemoteVNC (remotevnc.org), it is very likely that the future versions of VNC will no longer need a plugin (Flash, Java) and will support remote control of one desktop by another solely through a Web browser.

8 Appendix 2 - Open Source and Free AJAX Applications

This Appendix provides a list of Javascript frameworks and extensions for web desktop. Solutions are sorted by alphabetical order. We provide no recommendation here. If you need a recommendation, please contact us (info@nexedi.com).

8.1 Illustration

Some of the applications listed bellow are references on www.canvasdemos.com, a Web site which shows powerful applications of the Canvas component of modern Web browsers. Other applications come from a blog²⁶.

Application	Li-brary	URL	Li-cense	Features
Hanov Web Draw	N/A	www.hanovsolutions.com/webdraw	Un-known	Vector based drawing of sketch illustrations
Sketchpad	N/A	mugtug.com/sketchpad	Un-known	A demo application which shows that making an equivalent of Photoshop of GIMP as an AJAX application is possible
Physic Demo	N/A	physicsketch.appspot.com	Un-known	A Stunning demo which shows how far Javascript can be used nowadays
Ajax Anima-tor	N/A	antimatter15.com/ajaxanimator/build	GPL	A usable Flash movie creation environment for simple animated illustrations
InfoVis Tool-kit	N/A	thejit.org	BSD	Data visualization and graph library
Zoto ²⁷	N/A	code.google.com/p/zoto-server	BSD	Photo sharing system with many AJAX features
yFile	N/A	www.yworks.com	Proprietary	Graph viewer

²⁶ <http://stackoverflow.com/questions/1230651/graph-drawing-for-the-web-2-0>
²⁷ http://www.readwriteweb.com/archives/ajax_photo_sharing_site_zoto.php

Application	Li-brary	URL	Li-cense	Features
GWTUML	N/A	code.google.com/p/gwtuml	LGPL	UML designer
svg-edit	JQuery	code.google.com/p/svg-edit/	Apache	Full features SVG editor
Flowchart	Ajax.org	ui.ajax.org/#demos/elements.flowchart	LGPL	Flow chart editor

Table 4. AJAX Illustration Applications

8.2 Imaging

Thanks to Canvas HTML page element, it is now possible to build sophisticated image processing applications in pure Javascript.

Application	Li-brary	URL	Li-cense	Features
Ajax Image Editor	ExtJS	www.ajax-image-editor.com	GPL	AIE (Ajax Image Editor) is a rich internet application for manipulating images online. It's works with Mozilla Firefox and Internet Explorer.
Pixastic	JQuery	www.pixastic.com	MPL	Pixastic is a JavaScript library which allows you to perform a variety of operations, filters and fancy effects on images using just a bit of JavaScript. Pixastic is also an experimental online photo editor that incorporates some of the same techniques used in the library to create an image manipulation application.

8.3 Word Processing

An increasing number of Rich Text Editors²⁸ has been created with Javascript.

²⁸ 10 jQuery and non-JQuery javascript Rich Text Editors
<http://www.queness.com/post/212/10-jquery-and-non-jquery-javascript-rich-text-editors>
tmpau1xbjz_.odt cc by-nc-nd

Application	Li-brary	URL	Li- cense	Features
CKEditor	N/A	ckeditor.com/	LGPL	CKEditor is a text editor to be used inside web pages. It's a WYSIWYG editor, which means that the text being edited on it looks as similar as possible to the results users have when publishing it. It brings to the web common editing features found on desktop editing applications like Microsoft Word and OpenOffice.
TinyMCE	N/A	tinymce.moxiecode.com	LGPL	TinyMCE is a platform independent web based Javascript HTML WYSIWYG editor control released as Open Source under LGPL by Moxiecode Systems AB.
Xinha	N/A	trac.xinha.org/wiki/WikiStart	BSD	Xinha is a powerful WYSIWYG HTML editor component that works in all current browsers. Its configurability and extensibility make it easy to build just the right editor for multiple purposes, from a restricted mini-editor for one database field to a full-fledged website editor. Its liberal, BSD licence makes it an ideal candidate for integration into any kind of project.
Wymeditor	jQuery	www.wymeditor.org/promote/	GPL MIT	WYMeditor is a web-based WYSIWYM (What You See Is What You Mean) XHTML editor (not WYSIWYG). WYMeditor's main concept is to leave details of the document's visual layout, and to concentrate on its structure and meaning, while trying to give the user as much comfort as possible (at least as WYSIWYG editors).
JQRTE	jQuery	jqframework.com/jqrte	MIT	Rich Text Editor integrated with jQuery and jQuery plugin, provide a free web 2.0 wysiwg rich text editor! Start using it to integrate with your site now

8.4 Spreadsheets

PALO is the most sophisticated open source web spreadsheet but other players now also provide interesting client-side only spreadsheets.

Application	Li-brary	URL	Li-cense	Features
PALO Web	ExtJS	www.jedox.com/en/products/Palo-Suite/palo-web.html	GPL or proprietary	Palo is a memory resident multidimensional (OLAP or MOLAP) database server and typically used as a Business Intelligence tool for Controlling and Budgeting purposes with Spreadsheet software (incl. its own native web based spreadsheet) acting as the user interface.
ZK	ZK	www.zkoss.org/product/zk-spreadsheet.dsp	GPL	A simple spreadsheet application based on ZK framework.
SocialCalc	N/A	www.socialtext.com/products/spreadsheets.php		An online spreadsheet integrated with Socialtext and created orginally by Dan Bricklin, the inventor of Visicalc
Sheet	JQuery	www.visop-dev.com/jquerysheet.html	LGPL	Query.sheet gives you all sorts of possibilities when it comes to giving your web application a spreadsheet style interface with MS Excel style calculations.
Simple Spreadsheet	Qooxdoo	sourceforge.net/projects/simple-spread		A simple spreadsheet implemented with Qooxdoo

8.5 Calendar

Open Source AJAX calendar applications how now reached a state where they provide similar features with proprietary online calendaring software such as Google Calendar.

Application	Li-brary	URL	Li-cense	Features
FullCalendar	JQuery	arshaw.com/fullcalendar/	MIT or GPL	FullCalendar is a jQuery plugin that provides a full-sized, drag & drop calendar like the one below. It uses AJAX to fetch events on-the-fly for each month and is easily configured to use your own feed format (an extension is provided for Google Calendar).
wdCalendar	JQuery	www.web-delicious.com/jquery-plugins/#calendar	LGPL	wdCalendar is a jquery based google calendar clone. It cover most google calendar features.
Ext Calendar Pro	ExtJS	ext.ensible.com/products/calendar	GPL	Ext Calendar is a calendar application for ExtJS
ZK Calendar	ZK	www.zkoss.org/product/zkcalendar.dsp	GPL	ZK Calendar is a Google Calendar like Ajax component that integrates rich and intuitive scheduling functionality into enterprise web applications using pure Java.
Ext Scheduler	ExtJS	ext-scheduler.com	Proprietary	A scheduler application (ie. a calendar for business applications and project managemeny)

8.6 Mail Reader

Email readers are integrated in WebOS and collaboration suites, rather than standalone.

Application	Li-brary	URL	Li-cense	Features
Group Office	ExtJS	www.group-office.com	AGPL and proprietary	A complete collaboration suite which includes an email client.
EyeMail	Qooxdoo	wiki.eyeos.org/EyeMail	AGPL	EyeOS include an email client compatible with POP and IMAP

8.7 WebOS Desktops

Nearly every Javascript framework includes its sample WebOS Desktop, although few are you in practice.

Application	Li-brary	URL	Li-cense	Features
SonSpring JQuery Desktop	JQuery	sonspring.com/journal/jquery-desktop desktop.sonspring.com	GPL MIT	
Leopard Desktop with JQuery using jqDock	JQuery	net.tutsplus.com/tutorials/javascript-ajax/leopard-desktop-in-jquery-using-jqdock		A tutorial to teach you how to use jQuery to create a completely coded Dashboard, just like Leopard!
Andy Sharman	JQuery	desktop.udjamaflip.com/ www.udjamaflip.com/joomla/53-free-joomla-admin-template-jquery-desktop.html	GPL	A desktop template which is used for the administration mode of Joomla
MochaUI	Mootools	mochau.org/demo/	MIT	MochaUI is a web applications user interface library built on the Mootools JavaScript framework. It includes a Web Desktop and Portal framework.

8.8 IDE

Online IDEs are getting increasingly popular due to the collaborative features they provide²⁹.

Application	Library	URL	License	Features
Sednaspaces	N/A	www.sednaspaces.com		SednaSpace is a browser based software development platform for the technical and non-technical people to develop web based software products/projects having BPM, Workflow and SOA.
Cloud9	Ajax.org	www.cloud9ide.com	LGPL	Cloud9 IDE is aiming to be the IDE for Javascript developers.
Etherpad	N/A	code.google.com/p/etherpad	Apache	Web based collaborative text editor
EditArea	N/A	www.cdolivet.com/index.php?page=editArea sourceforge.net/projects/editarea/	Unknown	EditArea, a free javascript editor for source code. This editor is designed to edit source code files in a textarea.
CodeRun	N/A	www.coderun.com/ide	Unknown	Web based IDE which includes JQuery Support
IDE One	N/A	ideone.com	Unknown	Run code in more than 50 languages online
CodePad	N/A	codepad.org	Unknown	Run code in more than 10 languages online
Bespin	N/A	bespin.mozilla.com		HTML5 based Web based code editor part of Mozilla Labs projects ³⁰
WWW SQL Designer	N/A	code.google.com/p/wwwsqldesigner		A web based SQL designer tool
EyeDesigner	N/A	blog.eyeos.org/en/2009/07/14/eyedesigner-1-0-released/		Form designer application for EyeOS
Ymacs	DynarchLIB	www.ymacs.org/demo	BSD	A javascript implementation of a subset of Emacs
CodeMirror	N/A	marijn.haverbeke.nl/codemirror	ZLib-style	A text editor which can be used to create Web IDEs
CodePress	N/A	codepress.sourceforge.net	LGPL	CodePress is web-based source code editor with syntax highlighting written in JavaScript that colors text in real time while it's being typed in the browser.
JSVI	N/A	gpl.internetconnection.net/vi	GPL	A VI clone written in Javascript
MDK Editor	N/A	www.mdk-photo.com/editor		An XML/CSS oriented text editor
ecoder	N/A	ecoder.quintalinda.com	GPL	ecoder is an open-source web-based code editor, with real-time colour syntax highlighting, which allows multiple documents to be edited directly
Amy Editor	N/A	www.amyeditor.com github.com/aprilchild		A Collaborative Text and Source Code Editor for Developers. Includes an IDE and a MacOS like windowing interface. Can also be used as an embedded editor, with quite sophisticated features.
NotaPad	N/A	www.nicolarizzo.com/games-room		A web based IDE, still in development, based on my CodeTextArea widget

Application	Library	URL	License	Features
Code Text Area	N/A	code.google.com/p/code-textarea	New BSD License	A web based editor for developers
Qooxdoo playground	Qooxdoo	demo.qooxdoo.org/current/playground		A Web based javascript console which can be used to experiment with Qooxdoo components
Foxmemo	Qooxdoo	sourceforge.net/projects/foxmemo		Form development IDE

8.9 Languages

Javascript has now become a popular runtime for various programming languages. It could become the universal runtime for object languages, a long time goal in computer science.

Application	Language	URL	License	Features
Skulpt	JQuery	www.skulpt.org		Skulpt is an entirely in-browser implementation of Python.
CoffeeScript	CoffeeScript	jashkenas.github.com/coffee-script		CoffeeScript is a little language that compiles into JavaScript. Underneath all of those embarrassing braces and semicolons, JavaScript has always had a gorgeous object model at its heart. CoffeeScript is an attempt to expose the good parts of JavaScript in a simple way.
emscripten	Python Lua LLVM	code.google.com/p/emscripten		Emscripten is an LLVM-to-JavaScript compiler. It takes LLVM bitcode (which can be generated from C/C++, using llvm-gcc or clang, or any other language that can be converted into LLVM) and compiles that into JavaScript, which can be run on the web (or anywhere else JavaScript can run).
StratifiedJS	StratifiedJS	onilabs.com/stratifiedjs		StratifiedJS extends the JavaScript language with a small number of keywords for concurrent programming. It allows you to express asynchronous control flow in a straightforward, structured sequential style.
haskellin-javascript	Haskell	github.com/johang88/haskellin-javascript		A haskell interpreter in javascript
oia	Io	github.com/stevedekorte/oia		A port of the Io programming language to Javascript.

Application	Language	URL	License	Features
O'Browser	CAML	www.pps.jussieu.fr/~canou/obrowser/tutorial/		Javascript bytecode interpreter of the CAML language
Pyjs	PYJS	pyjs.org		Python Javascript Compiler
GWT	Java	code.google.com/webtoolkit/overview.html		Java to Javascript Compiler

8.10 Others

The following links are not yet classified:

- Speed of JS Frameworks³¹
- A Brief Look at Online IDEs³²
- Open Source Online IDE³³
- The Most Complete AJAX Framework and JavaScript Libraries List(124+)³⁴
- Comparison of JavaScript-based source code editors³⁵
- Six Revisions: 10 Impressive JavaScript Animation Frameworks³⁶
- Six Revisions: 10 Promising JavaScript Frameworks³⁷
- Six Revisions: Graphing/Charting Data on Web Pages: JavaScript Solutions³⁸
- Javascript framework usage among top websites³⁹
- Web Resource Depot⁴⁰ has a lot of information about Web & AJAX frameworks
- ajaxline.com Best AJAX Text Editors⁴¹
- Wikipedia Online rich-text editor⁴²
- mashable.com 6 Embeddable HTML Editors for Your Website Compared⁴³
- Languages Come to Javascript VMs: CoffeeScript 1.0, StratifiedJS, C/C++ with Emscripten, Python⁴⁴

31 <http://dante.dojtoolkit.org/taskspeed/report/charts.html?exclude=defaultbrowser0>
32 <http://geekswithblogs.net/yowhann/archive/2007/03/18/109153.aspx>
33 <http://stackoverflow.com/questions/2040108/opensource-online-ide/2040117>
34 <http://ntt.cc/2008/02/13/the-most-complete-ajax-framework-and-javascript-libraries-list.html>
35 http://en.wikipedia.org/wiki/Comparison_of_JavaScript-based_source_code_editors
36 <http://sixrevisions.com/javascript/10-impressive-javascript-animation-frameworks/>
37 http://sixrevisions.com/javascript/promising_javascript_frameworks/
38 http://sixrevisions.com/javascript/graph_chart_plot_data_javascript/
39 <http://royal.pingdom.com/2008/06/11/javascript-framework-usage-among-top-websites/>
40 <http://www.webresourcesdepot.com/>
41 <http://www.ajaxline.com/best-ajax-text-editors>
42 http://en.wikipedia.org/wiki/Online_rich-text_editor
43 <http://mashable.com/2008/11/03/embeddable-html-editors/>
44 <http://www.infoq.com/news/2010/12/languages-on-javascript>

9 Appendix 3 - Web Interoperability

There are many differences between a desktop environment (ex. KDE, GNOME) and a window manager (ex. TWM). A Desktop environment usually provides common APIs for all application to implement copy and paste, drag and drop, file access, printer access, sound sharing, etc.

Rich Javascript frameworks (ex. ExtJS, Qooxdoo, GWT) try to provide a kind of unified API for those features. Virtual Desktop environments such as EyeOS go further by defining all the APIs of a desktop environment in Javascript. However, in a real world environment, one may want to use PALO Suite spreadsheet (ExtJS based) together with EyeOS word processor, within the same virtual desktop or on two different tabs of the browser. This raises concerns related to interoperability of Web based Virtual Desktop environments

9.1 Common Access to Files

PALO Suite implements its own file manager, which is not the same as the file manager of EyeOS or the file manager of Group-Office. There is not central place to keep all files, which is quite inconvenient.

9.1.1.Solution

The “Resilience” project has proposed to create a Javascript library, JIO, to unify the access to diverse storage facilities directly from the browser. Thanks to a component architecture, access to new types of storage does not require to rewrite Javascript applications. The approach is similar to kio (KDE) or gvfs (GNOME).

9.2 Copy and Paste

Copying and pasting inside PALO is possible. Copying and pasting inside EyeOS seems somehow possible. Copying and pasting a spreadsheet designed with PALO Suite into EyeOS word processor does not seem to be possible. Copying and pasting inside PALO which itself is hosted inside an EyeOS virtual desktop seems impossible.

Copying and pasting images works fairly well but is limited to bitmaps, which results in loss of quality and aliasing.

9.2.1.Solution

HTML5 drag-and-drop API⁴⁵ could be considered as a way to support a more interoperable implementation of copy and paste. Another approach could consist in mapping copy and paste keys and use a text representation of information which is copied and pasted.

9.3 Drag and Drop

Drag and Drop from PALO to PALO does not seem to be possible. Drag and Drop from EyeOS to EyeOS does not seem to be possible. Drag and Drop from PALO into EyeOS word processor does not seem to be possible.

9.3.1.Solution

HTML5 provides a drag-and-drop API.

9.4 Printing

Printing from EyeOS uses the printing capabilities of the local PC. There is no printing in PALO.

9.4.1.Solution

HTML5 provides a print API. However, this print API relies on the guest operating system. Another approach consists of implementing a library for printing,

Jprinter. Such a library could even act as a native IPP client, providing more control to select printers and define printing options.

9.5 Foreign Web Application inside WebOS

A WebOS should be capable of providing a unified interface to all applications, both its own and those provided by a third party server. For example, one may want to use PALO Suite spreadsheet instead of WebOS built-in spreadsheet, by clicking on an icon. The result should look like a spreadsheet and not like a spreadsheet inside a browser.

Currently, running PALO as part of EyeOS is possible but many little issues happen in relation with keyboard handling.

9.5.1.Solution

Javascript frameworks could adopt common theming technologies, such as those of JSueryUI's theme roller. Each javascript framework would try to adapt non native theming information into its native theming representation. This way, different applications using different frameworks would share the same appearance.

Another solution is based on isolation, which is also a concept which the “Resilience” project is going to study. Through appropriate isolation, security risks should be reduced. Also, keyboard mapping can be different from one isolation container to another. Last, different containers may use different credentials.

9.6 Multiple Gadget Providers

A gadget based portal should be capable of displaying information provided by multiple providers. For example, one may want to view ERP5 Gadgets inside Group Office. This is possible using the iframe tag but it is not always sufficient⁴⁶.

9.6.1.Solution

Google promotes OpenSocial which is in reality more an “OpenGadget” based

on Google Gadget API⁴⁷ standard than anything else⁴⁸.

IBM has a product called “IBM Mashup Center”⁴⁹ which tries to integrate so-called iWidget⁵⁰ with OpenSocial gadgets.

OpenAjax provides “OpenAjax Hub 2.0”⁵¹ to handle security risks related to the combinatin of multiple gadgets in the same browser. It is based on the OpenAjax Gadget specification⁵².

9.7 Single Login

All application part of a single Web based Virtual Desktop should share the same authentication infrastructure so that it is not necessary to login multiple times, once for PALO, once for ERP5, etc.

Ideally, multiple login should be possible at the same time, by running multiple browser tabs or windows.

9.7.1.Solution

Modern Web solutions for authentication are based on OpenID and/or OAuth. With OpenID, a single login can be used for multiple applications. With OAuth, one application can use private data from another application without requiring to copy user and passwords from one application to another.

⁴⁷ <http://code.google.com/intl/fr/apis/gadgets/>

⁴⁸ <http://blogs.zdnet.com/social/?p=386>

⁴⁹ <http://www.ibm.com/developerworks/lotus/library/mashups-opensocial/index.html>

⁵⁰ <http://download.boulder.ibm.com/ibmdl/pub/software/dw/lotus/mashups/developer/iwidget-spec-v2.pdf>

⁵¹ <http://www.openajax.org/whitepapers/Introducing%20OpenAjax%20Hub%202.0%20and%20Secure%20Mashups.php>

⁵² http://www.openajax.org/member/wiki/OpenAjax_Metadata_1.0_Specification