

## White Paper

# Linux / OpenSource VoIP.

---

How to turn the Linux/OpenSource threat to Siemens' VoIP business into an opportunity.

Jose C. Lacal  
Senior Engineer  
Data Services  
Siemens Information and Communication Networks

+01 (561) 955-3081  
jose.lacal@icn.siemens.com

Updated 12/23/98 14:00 hr.

***NOTICE: This is NOT an official Siemens document. This document reflects the author's opinions only. There is no implicit or explicit assumption of Siemens' endorsement of the contents of this document. The author is solely responsible for the material in this document.***

---

# Table of Contents

1. Revision History.....	5
2. Glossary.....	6
3. Summary.....	7
4. Linux / OpenSource Backgrounder.....	9
Technical merits.....	9
OpenSource's benefits.....	9
Stability.....	9
Lower cost of ownership.....	10
Speed of releases.....	10
"Security by transparency".....	10
No built-in hardware obsolesce.....	10
The business case.....	11
Own your own destiny.....	11
Create products for "markets of one".....	12
Faster development cycles.....	12
Lower overhead, lower cost, longer life.....	12
Closeness to the customer.....	13
Broader market potential.....	13
Open interfaces to the end customer.....	13
Where is Linux / OpenSource going?.....	13
Additional information.....	13
5. Linux/OpenSource Threat.....	14
Commoditize VoIP knowledge.....	14
Evaporate price points.....	14

De-proprietarize hardware.....	14
Competitors rushing to OpenSource (Ericcson).....	14
Investors funding OpenSource start-ups.....	15
Half-life of proprietary knowledge is collapsing.....	15
Partnerships are not working out for Siemens.....	15
6. Opportunities for Siemens.....	16
Integration knowledge.....	16
Large-scale deployments.....	17
OpenSource needs an Andersen Consulting.....	17
Brand-name recognition (credibility).....	17
7. Linux/OpenSource Business Models.....	18
Support Sellers.....	18
Loss Leader.....	19
Widget Frosting.....	19
Accessorizing.....	20
IBM's business model.....	20
8. Plan of Action.....	21
Appendix A: Summary of Ericcson's Erlang operating system.....	22
Open-source Erlang - White Paper.....	22
Erlang Overview.....	22
Features:.....	22
Components.....	23
Tools and Libraries.....	23
Appendix B: Summary of Ericcson's High-Availability Software.....	25
What is Eddie?.....	25
IP Migration Application.....	25
Load-balancing DNS Server.....	25

Intelligent HTTP Gateway.....	25
Content Replication Application.....	26
Appendix C: Voxzilla.org Project.....	27
Aims of the Voxilla H.323 Project.....	27
How this project got started.....	27
Who is Equivalence?.....	27
Suggestions for the first program.....	27
And what then. ....	28

## 1. Revision History.

10/14/98	JCL	First draft
12/23/98	JCL	First public release

Filename: OpenSource VoIP threat.doc

This document will be permanently evolving. Please consider it in "permanent draft" status.

Areas marked <TBFO> need to be filled out.

Please send any comments about this document to [jose.lacal@icn.siemens.com](mailto:jose.lacal@icn.siemens.com)

## 2. Glossary.

This is a list of acronyms used throughout this document and their definition.

Closed Source Software (CSS)	Refers to software where the user does not get access to the source code of the software. The user is at the mercy of the software developer when it comes to patches, customization, and further enhancements to the code. Microsoft's software is the perfect example of CSS.
Linux	Linux is a free Unix-type operating system originally created by Linus Torvalds with the assistance of developers around the world. Linux is an independent POSIX implementation and includes true multitasking, virtual memory, shared libraries, demand loading, proper memory management, TCP/IP networking, and other features consistent with Unix-type systems. Developed under the GNU General Public License, the source code for Linux is freely available to everyone. More information available at <a href="http://www.linux.org">http://www.linux.org</a>
OpenSource	The basic idea behind OpenSource is very simple. When programmers on the Internet can read, redistribute, and modify the source for a piece of software, it evolves. People improve it, people adapt it, and people fix bugs. In addition, this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing. The OpenSource community has demonstrated that this rapid evolutionary process produces better software than the traditional closed model, in which only a very few programmers can see source and everybody else must blindly use an opaque block of bits. More information available at <a href="http://www.opensource.org">http://www.opensource.org</a>
OS	Operating System. The basic piece of software that allows computers to interpret instructions from both software applications and operator's input. Microsoft Windows and Linux are examples of operating systems.

### 3. Summary.

This paper urges Siemens to open-source the API to the InterXpress product line. Siemens should allow anybody to write against the Siemens VoIP API and have anybody's equipment (even toasters) talk to a Siemens' VoIP equipment. That can only expand the pie for everybody.

Also, Siemens should **publicly** and **financially** support all OpenSource VoIP efforts, such as Voxilla. With a very small amount of money (compared to what a "traditional" development effort costs), Siemens could buy itself a very powerful contingency plan in case any of our multiple "partnerships" break down.

I am of the opinion that Siemens has already lost the race to the VoIP market with our "traditional" development and partnering approach. Our partners (3Com, NetSpeak) take Siemens for a ride, and we end up with the short end of the stick.

At the same time, in the Linux/OpenSource space, Siemens could be THE leading VoIP supplier / force. That is an opportunity of a lifetime. In the traditional arena, we will always be laggards. For once, Once, ONCE, let's make this elephant (Siemens) take a leap forward instead of slurring our way through business life.

1998 has been the year of Linux/OpenSource. OpenSource is the 'Intel inside' of the next generation of personal computers. Just as Intel and Microsoft are inside most PCs today, sendmail, Apache, BIND, Perl and Linux are inside the most powerful web sites.

InterActive Week, on its December 18<sup>th</sup> edition, available at <http://www.zdnet.com/intweek/stories/news/0,4164,2177618,00.htm> reports:

"About 510,000 copies of Linux were shipped by Linux distributors this year - an increase of 211 percent over a year ago - making Linux the fastest growing server operating system, according to preliminary International Data Corp. figures. As a point of comparison, Microsoft Corp. shipped 1.2 million copies of Windows NT, making it the leader in number of units shipped."

Keep in mind that, since Linux can be installed multiple times without the need to buy additional licenses, the real number of Linux installations is several times the number of copies shipped.

The long-term economic model is not about the golden eggs (the Intellectual Property) laid last year. It is about the golden goose and tomorrow's golden eggs. The shelf life of eggs these days is shrinking dramatically, and the value of an egg that no one knows about is tiny. Give the eggs away as a testament to the value of the goose and a prediction of eggs to come.

Siemens must ride the coattails of the nascent OpenSource community and its established channels such as RedHat. OpenSource has a certain buzz that is greater than its real customer-closing value, but this buzz is getting hot. Moving aggressively towards OpenSource can make Siemens a category killer for the VoIP application server market.

The level of growing investment in Linux -- both from established VC firms and from proven high-tech companies -- demonstrates that Linux is crossing the corporate line from being a utopian technical system to a solid business decision.

As a reference, IBM has traditionally kept close guard over its proprietary software creations and is seen as a jealous defender of its patents. However, like a growing number of companies, it now sees possible benefits to opening up its technology. Far from giving up on its commercial goals, IBM stands to benefit indirectly from solving security issues due to its large and rapidly growing investment in e-commerce. IBM sees this "giving away" of OpenSource software as an indirect revenue generator.

Please refer to sections **6. Opportunities for Siemens** and **7. Linux/OpenSource Business Models** for a compilation of business opportunities the Linux/OpenSource movement offers Siemens.

## 4. Linux / OpenSource Backgrounder.

This section provides a quick summary of background information about Linux / OpenSource.

OpenSource is software (such as Linux, Apache, BIND, sendmail, Netscape Navigator, DJGPP, Python, Tcl/Tk, Perl, etc.) that you can use and distribute freely, and it always comes with the source code for you to modify to suit your needs. More information available at <http://www.opensource.org>

1998 has been the year of Linux / OpenSource. Moreover, 1999 is the year of Microsoft's NT Server 5.0 fiasco. OpenSource is the 'Intel inside' of the next generation of personal computers. Just like Intel and Microsoft are inside most PCs today, sendmail, Apache, BIND, Perl and Linux are inside the most powerful web sites.

Netscape's jumping into the OpenSource market has had a tremendously positive effect on other commercial software developers. IBM, Oracle, Sybase, Informix and a host of other software and hardware vendors have embraced the Linux / OpenSource movement.

The public's perception of OpenSource has changed in the last few months, much faster than anybody could have anticipated. The reality has not: OpenSource is a good alternative for enterprises and consumers alike.

### Technical merits

These are some of the technical highlights of Linux:

- Built from the ground up to comply with open standards (TCP/IP, Posix, etc.)
- Source code available to allow infinite customization
- Minimalistic approach: compile into the kernel only the required services / features
- No run-time or per-user license costs
- Runs on very low-end hardware (386, 4 Mb RAM, 100 MB hard disk)
- Large, dedicated and knowledgeable development team scattered around the world
- State-of-the-art 64-bit OS with source code at a very low cost
- Highly scalable: from 386 PCs to SMP Pentium IIs, Alpha and Sparc chips
- Designed to maximize available hardware resources

It is relevant to mention that Sun Microsystems has announced that, effective immediately, they will ship Linux (along with the traditional Solaris OS) with all of their high-end UltraSparc machines, based on overwhelming customer demand.

### OpenSource's benefits

These are some of the benefits of using software developed under the OpenSource philosophy.

#### Stability

No one in the computer industry was surprised when Steve Ballmer, Microsoft Vice-President, admitted that in the rush to beat Netscape, Microsoft had cut corners in quality control. (So had Netscape -- we are not playing favorites here.) Based on anecdotal

evidence, many people have problems with Windows crashing for no particularly good reason, often causing people to lose work. Sometimes it is just an annoyance, but other times it can be quite serious. While Windows 95 and in particular Windows NT are far more stable than Windows 3.1x, they still fail with alarming regularity. There are documented cases of Linux servers running for over a year at a time without a system-halting crash.

#### Lower cost of ownership

Commercial operating systems, especially server operating systems, like Windows NT Server and Novell IntraNetwork, can cost more than US\$400 for a single copy and limited number of user licenses. Even the more modest US\$100 for Windows 95 can be a problem for people in developing countries, as well as students and others. The price for commercial systems almost never includes development tools, which cost even more.

Linux, on the other hand, includes a complete Network OS along with free C, C++, and FORTRAN compilers. Also included are other development tools, and everything is usually available for less than US\$50, with the right to install the software on an unlimited number of PCs.

#### Speed of releases

Major upgrades to CSS operating systems come very slowly. Windows 95 came out three years after Windows 3.11; Windows 98 was released three years after Windows 95. Other systems are similar. The major distributions of Linux, on the other hand, are releasing major updates every few months or so.

Minor upgrades also take longer to acknowledge and fix in commercial systems. Recent problems with Netscape Navigator and Microsoft Internet Explorer are good examples of this. In the case of the "denial of service" TCP/IP bug, a Linux patch was posted for it mere hours after the problem was isolated. Anyone who is technically capable can fix the bugs, too, merely by changing the code in question and recompiling.

#### "Security by transparency"

One of the biggest problems with "commercial" software is that users can not get into the code and verify if there are security holes in the code. All "commercial" software vendors use the "security by obscurity" security model: "we don't tell you what's in the code, trust us it's secure." Then, when somebody finds a flaw, the vendor takes its merry time before plugging the security hole.

With Linux/OpenSource software, we follow the "security by transparency" model: all of the code is out there in the open, and anybody can try to break it. If somebody does, that same person will report it and immediately work on a fix.

#### No built-in hardware obsolescence

Linux has superb support for older hardware. In fact, it is often easier to find hardware support for 486-era hardware than the latest gizmos to hit the market. This is because the driver authors need time to write and test the drivers, and some vendors are not exactly forthcoming with the information required to write a driver.

What this means for the user is that they can protect their investment in hardware. When old hardware is rendered obsolete by the latest version of Windows (or MacOS...we try to be fair), it can most likely still run enough of Linux to be perfectly useful. There is no reason for Linux users to try to make hardware obsolete - and every reason for them to provide support for older hardware.

## The business case

The Linux / OpenSource community is ready to talk the language of business: money. Richard Stallman, of GNU and Free Software Foundation fame, spoke for the entire Linux / OpenSource community when he said: "Making money with OpenSource software is good. There's nothing wrong with commercial software, as long as it is OpenSource."

From: <http://www.opensource.org/for-suits.html>

"The OpenSource model has a lot to offer the business world. It is a way to build open standards as actual software, rather than paper documents. It is a way that many companies and individuals can collaborate on a product that none of them could achieve alone. It is the rapid bug-fixes and the changes that the user asks for, done to the user's own schedule.

"The OpenSource model also means increased security; because code is in the public view it will be exposed to extreme scrutiny, with problems being found and fixed instead of being kept secret until the wrong person discovers them. And last but not least, it is a way that the little guys can get together and have a good chance at beating a monopoly.

"Of all these benefits, the most fundamental is increased reliability. And if that's too abstract for you, you should think about how closed sources make the Year 2000 problem worse and why they might very well kill your business.

"...The foundation of the business case for OpenSource is high reliability. OpenSource software is peer-reviewed software; it is more reliable than closed, proprietary software. Mature OpenSource code is as bulletproof as software ever gets.

"This is a radical idea to many businesspeople. Many have a belief that OpenSource software is necessarily not 'professional', that it is shoddily made and more prone to fail than closed software.

"The Internet's infrastructure makes the best possible refutation. Consider DNS, sendmail, the various open-source TCP/IP stacks and utility suites, and the open-source scripting languages such as Perl that are behind most 'live' content on the Web. These are the running gears of the Internet."

## Own your own destiny

By using Linux and OpenSource software development components, Siemens will take control of its own destiny. The concept is that \*any\* OS currently uses is tweaked to its creator's desires. With Linux, STN can tweak it to its OWN desires / needs, instead of relying on Microsoft / Sun / HP to provide STN with whatever STN wants.

On the other hand, if Siemens developers customize the Linux kernel, it is more likely that the documentation coming out of internal Siemens people will be more accurate and detailed than the documentation from a "commercial" vendor. The vendor wants to hide as much information from its OS' internals as possible.

Linux is based on open standards (TCP/IP, C and C++) and is in the process of becoming Unix 98 compliant. Therefore, all development done by Siemens in / for Linux will be portable across multiple platforms with a minimum of adjustments.

Create products for "markets of one"

Linux allows Siemens to pursue new business opportunities at substantially lower price points. To the point of building infinitely-customizable products, even for a single customer.

Faster development cycles

Siemens could also "contract" with several Linux/OpenSource developers around the world to work for Siemens from their homes and to develop software that Siemens has first crack at. Why? There are some wonderful developers out there, throughout the world, that are currently releasing their software to the Net.

That software is orders of magnitude better than what "commercial" developers put out. For a very small amount of money, Siemens could "hire" the best programmers on the Internet. The code will still be released to the Net, but Siemens will have both a copyright on the software as well as the right to use it internally.

Examples of struggling software developers out there whom could work for Siemens for US\$5,000.00 per year:

- GNOME  
GNOME is intended to be a free and complete set of user friendly applications and desktop tools, similar to CDE and KDE but based entirely on free software.  
<http://www.gnome.org>
- Linux Router Project  
A networking centric mini-distribution of Linux. LRP is small enough to fit on a single 1.44MB floppy disk, and makes building and maintaining routers, terminal servers, and typically embedded networking systems next to trivial. Throw in a 486 PC with 08 MB RAM, 01 1.44 Floppy, no HD and 02 NICs and you have got yourself a router. No Cisco router required.  
<http://www.linuxrouter.org>

Lower overhead, lower cost, longer life

Using Linux/OpenSource software allows Siemens to save money by replacing proprietary OSes (NT Server, Solaris, SCO, Netware, etc.) with high per-unit licensing fees.

Siemens should look into Linux as a replacement for both Solaris and Digital Unix on the ISP packages Siemens sells. There will be tremendous benefits from doing so:

- Cheaper packages, allowing Siemens to sell better to price-conscious customers
- Faster bug fixes than the "commercial" vendors are able to offer
- Infinite customization since Siemens has access to the source code
- New features available without paying additional licensing fees
- 100% Internet-standard implementation, non-proprietary
- Rock solid performance, with documented uptimes of years

- Sys admins in underdeveloped countries are used to Linux anyway
- Complete packages with all services bundled in for a low fee: sendmail, Apache, IMAP4, POP3, LDAP, TACACS, ftp, newsgroups, DNS, secure commerce, etc.

Take a look at Red Hat's new Secure Webserver, which allows you to create a site to conduct secure electronic commerce (128 bits). That, plus a PostgreSQL database engine in the back, and you have got a highly customizable eCommerce platform. All for US\$99.95 and with full source code included.

More info at <http://www.redhat.com/product.phtml/WB2000>

Closeness to the customer

OpenSource does not mean better programmers, you get better users. OpenSource empowers end users to help maintain high-quality software.

Broader market potential

<TBFO>

Open interfaces to the end customer

<TBFO>

## Where is Linux / OpenSource going?

Let the numbers speak for themselves:

InterActive Week, on its December 18<sup>th</sup> edition, available at <http://www.zdnet.com/intweek/stories/news/0,4164,2177618,00.htm> reports:

"About 510,000 copies of Linux were shipped by Linux distributors this year - an increase of 211 percent over a year ago - making Linux the fastest growing server operating system, according to preliminary International Data Corp. figures. As a point of comparison, Microsoft Corp. shipped 1.2 million copies of Windows NT, making it the leader in number of units shipped."

Keep in mind that, since Linux can be installed multiple times without the need to buy additional licenses, the real number of Linux installations is several times the number of copies shipped.

## Additional information

You can find more information about Linux / OpenSource at the following websites:

<http://www.linux.org>  
<http://www.opensource.org>  
<http://www.linuxtoday.com>  
<http://www.linuxworld.com>  
<http://www.kernel.org>  
<http://www.linuxresources.com>

## 5. Linux/OpenSource Threat.

This is an analysis of the threats the Linux/OpenSource philosophy presents to Siemens' VoIP strategy.

### Commoditize VoIP knowledge

The value of VoIP knowledge is decreasing exponentially by the month. With the inclusion of VoIP capabilities in both Windows 98 and NT Server 5.0 (aka Windows 2000), plus the availability of an OpenSource-based H.323 stack, VoIP knowledge is going to become a commodity in the next 12-18 months.

### Evaporate price points

VoIP technology is on the same price/performance slope as the computer industry. That means prices drop constantly and relentlessly.

Glossy business plans made on the assumption of telecom-class pricing strategies are ridiculously optimistic. Assuming less than a 50% per year price erosion is criminally misguided.

### De-proprietarize hardware

The wide availability of source code afforded by Linux/OpenSource products makes proprietary hardware a thing of the past.

Specifically, I predict that vendor-specific VoIP hardware (3Com, Vienna, Nortel, Natural MicroSystems, etc.) will be replaced in the next 12 months by generic, off-the-shelf components that can be installed on off-the-shelf PCs running off-the-shelf VoIP applications under Linux. Therefore, creating very powerful, cheap, customizable VoIP systems.

For a historical precedent, look at hardware drivers prior to the introduction of Microsoft Windows 3.1 (in the good old DOS times). Prior to Win 3.x, each software vendor had to write drivers for each possible hardware element available in the marketplace. WordPerfect, for example, was very popular since they had an extensive library of printer drivers that was the best in the industry. When Win 3.x was introduced, that competitive advantage evaporated, since the printer manufacturers had to write **one** driver, for Win 3.x, and then **any program** would be able to access that printer via Win 3.x's Print Manager.

### Competitors rushing to OpenSource (Ericsson)

Ericsson's OpenSource initiatives:

<http://www.erlang.org>

Erlang is a programming language designed at the Ericsson Computer Science Laboratory. Open-source Erlang is being released to help encourage the spread of Erlang outside Ericsson.

We are releasing free of charge:

- The entire source code of the current Erlang system.
- The entire source code for Mnesia a distributed Database Management System, appropriate for telecommunications applications and other Erlang applications with need of continuous operation and soft real-time properties.
- Extensive libraries of code for building robust fault-tolerant distributed applications

All the above software has been battle tested in a number of Ericsson products.

Please refer to **Appendix A: Summary of Ericsson's Erlang operating system** below for more details.

Ericsson is obviously trying to position their software as the reference language for high-end, Telecom-quality products.

Why would anybody develop using closed-source languages if there is a good OpenSource alternative?

## Investors funding OpenSource start-ups.

Will 1999 be the year of the open source startup? It may seem unlikely that dropping the words open source into a business plan will have the same wallet-grabbing effect on investors that the word Internet did two years ago. And yet, lately a number of venture capitalists, prominent individual investors, and major corporations have made significant investments in companies that release freely redistributable software along with the source code.

This fall Intel, Netscape, and VC (venture capital) firm Greylock Management made waves in the open source world with an undisclosed investment in Linux distribution vendor Red Hat Software. In addition, OpenSource companies VA Research, Cobalt Networks Inc., and Sendmail Inc. have all received investment money in the last year.

## Half-life of proprietary knowledge is collapsing

<TBFO>

"The paradox of software economics:" Economics is the allocation of scarce resources. But, OpenSource is based on abundance, you more you give the more you have.

## Partnerships are not working out for Siemens

<TBFO>

## 6. Opportunities for Siemens.

I can see a tremendous opportunity for Siemens in the VoIP arena.

NOTE: The material below was lifted from <http://www.zope.com/Information/BusinessDecision> and adapted to Siemens' situation.

How can going OpenSource increase the value of Siemens' VoIP offerings? Here is what I see:

- Going OpenSource will increase Siemens' user base. A wider brand awareness and a stronger identity as a leading-edge company leads to more consulting work and to an increased valuation of Siemens' intellectual assets.
- OpenSource gives rock solid, battle-tested, bulletproof software on more platforms and with more capabilities than closed source software, thus increasing the value of Siemens' VoIP consulting.
- Fostering a Siemens-centered VoIP community creates an army of messengers, which is pretty effective marketing.
- This (VoIP) is not the last innovation Siemens will make.
- In the status quo, the value of packaging the software as a product would approach zero, as Siemens currently has zero market penetration. What is the value of a killer product with few users? The cost to compete in the established VoIP market is proving itself prohibitive.
- OpenSource makes the value of Siemens' ideas more apparent; thus, the perceived value of the company is apparent.
- Our architecture is "safer" for consulting customers. With thousands of people using it, the software is far less marginal. The customer is able to fix things themselves or reasonably find someone to do it for him or her. Finally, the software will "exist forever".
- Dramatically increasing the base of users and sites using it gives Siemens' products and services a tremendous boost in "legitimacy".
- The long-term economic model is not about the golden eggs (the intellectual property) laid last year. It is about the golden goose and tomorrow's golden eggs. The shelf life of eggs these days is shrinking dramatically, and the value of an egg that no one knows about is tiny. Give the eggs away as a testament to the value of the goose and a prediction of eggs to come.
- The community can work with us to dramatically increase the pace of innovation and responsiveness to new technical trends, such as Wireless VoIP and such.
- Ride the coattails of the nascent OpenSource community and its established channels such as RedHat. OSS has a certain buzz that is greater than its real customer-closing value, but this buzz is getting hot. Moving aggressively towards OpenSource can make Siemens a category killer for the VoIP application server market segment.

More specifically, these are the major opportunity areas for Siemens within the Linux/OpenSource VoIP arena.

### Integration knowledge

Siemens has a tremendous amount of knowledge in the areas of:

- Voice
- Switching
- VoIP development

- PBX
- Embedded design
- Manufacturing / assembly
- Distribution
- Marketing

Large-scale deployments

<TBFO>

OpenSource needs an Andersen Consulting

"The OpenSource community needs an IBM-like company support infrastructure."

Brand-name recognition (credibility)

## 7. Linux/OpenSource Business Models.

The most commonly asked question at this point is: "How do you make money in the 'free' OpenSource marketplace?" That is the wrong question. Not every software company makes any money selling software, to begin with. Look at Corel, WordPerfect, Ashton-Tate, Borland, etc.

Also, please remember: software was free before the 1980's. Therefore, really, the OpenSource is just going forward to the past.

There are at least four known business models for making money with OpenSource software. We will explore each business model in more detail below.

- **Support Sellers**  
In this model, you (effectively) give away the software product, but sell distribution, branding, and after-sale service. This is what (for example) Red Hat and Cygnus are doing.
- **Loss Leader**  
In this model, you give away open-source as a loss-leader and market positioned for closed software. This is what Netscape is doing.
- **Widget Frosting**  
In this model, a hardware company (for which software is a necessary adjunct but strictly a cost rather than profit center) goes OpenSource in order to get better drivers and interface tools cheaper.
- **Accessorizing**  
Selling accessories -- books, compatible hardware, complete systems with open-source software pre-installed. It is easy to trivialize this (open-source T-shirts, coffee mugs, Linux penguin dolls) but at least the books and hardware underlay some clear successes: O'Reilly Associates, SSC, and VA Research are among them.

These are the traditional business models in the OpenSource community:

### Support Sellers

"Seventy-five percent of our revenue is support," said Todd Andersen, vice president of marketing at SuSE Inc.

"There is a huge base of customers who need corporate-class support for OpenSource software."

There is a symbiosis relationship between free software (OpenSource) and making profits:

- Loss leaders (free software) bring people into the fold.
- The more money the company makes charging for the commercial version of the software, the more resources the company can devote to improving the OpenSource version of the software.

- Having a web-based resource center attracts users.
- The OpenSource approach accelerates the adoption of new technologies ("early adopters").
- OpenSource projects need commercial support.

Hybrid projects can be successful.

OpenSource is a world where the best ideas win: "one person, with a good idea, can reach millions of users." It is a culture of open discussion, where more smart minds create simpler, more powerful software.

"The law of large numbers:" if an OpenSource software package has enough users, it will be:

- highly functional
- reliable
- reasonably easy to use
- moderately well documented

On the other hand, corporate users want:

- Certainty: guaranteed support
- Services: training, consulting
- Domain-specific extensions: solutions customized to the company's needs

The OpenSource movement, by itself, has NO staying power. There has to be profits for vendors to stay in business, grow, and contribute back to the community.

## Loss Leader

Siemens can give away its VoIP Intellectual Property and still create a significant revenue stream by selling consulting and customization services.

## Widget Frosting

Instead of paying royalties to 3Com and NetSpeak, Siemens can use OpenSource VoIP software as the core of its offerings. That will

## Accessorizing

<TBFO>

## IBM's business model

IBM has 20 people devoted, full-time, to the Apache and other OpenSource projects. Most internal and external web servers at IBM run Apache.

The business model IBM pursues with OpenSource software is as follows:

- Use OpenSource code at the core (Apache)
- Build IBM's value-added products on top (Websphere)
- Provide service contracts to support both Apache and Websphere
- And build proprietary solutions on top of Apache

How has IBM measured the success of its use of OpenSource software thus far?

- Increased market presence
- Increase in service contracts (24% per year growth in service area)
- Taken skills acquired with the OpenSource development projects and spread them around the company.
- Injected the "Internet time" mentality into the company.
- "The tide rises all boats:" the more support the OpenSource movement receives from corporations like IBM, the more business opportunities will be created for all players.

IBM's corporate management is monitoring the success of the internal Apache project: if they succeed, the OpenSource movement will be taken more seriously inside IBM.

## 8. Plan of Action.

This is

- There's an on-going project to develop a Linux-based, OpenSource H.323 stack.
- That will be the foundation for a whole sleuth of OpenSource-based Telecom products: H.323 to PSTN gateways, NetMeeting-like apps, etc.
- Do not underestimate / brush off the potential of the Linux community to severely affect the VoIP marketplace. Linux has come out from nowhere to become the only serious threat to NT.
- Bottom line: in less than 12 months, there will be a very inexpensive full OpenSource, Linux-based VoIP platform available out there. That will put tremendous price pressure on all other commercial offerings.

Based on my previously stated opinion to Pedro Colaco's VoIP strategy paper:

Based on the experience of the Internet Phone since 1995, none of your priorities is what the IP telephony market wants. My priorities would be: an open interface (see below), fast turnarounds (include new features fast and furious), and flexibility (go with the market, instead of trying to dictate to the market).

If you ask me, I would give away Siemens' interface (API) to the InterXpress product line. I would allow anybody to write against the Siemens API and have their equipment (even toasters) talk to a Siemens' IX equipment. That can only expand the pie for everybody.

My unsolicited suggestion would be for Siemens to \*publicly\* and \*financially\* support the efforts of the Voxilla people. With a very small amount of money (compared to what the Development staff costs the IBU), Siemens could buy itself a very powerful contingency plan in case any of our multiple "partnerships" break down.

## Appendix A: Summary of Ericsson's Erlang operating system.

From: [http://www.erlang.org/white\\_paper.html](http://www.erlang.org/white_paper.html)

### Open-source Erlang - White Paper

This is a brief introduction to Erlang for programmers. Erlang was originally developed in Linux, and currently supports most UNIX variants as well as Windows NT Server.

#### Erlang Overview

Erlang is a programming language which has many features more commonly associated with an operating system than with a programming language: concurrent processes, scheduling, memory management, distribution, networking, etc.

The initial OpenSource Erlang release contains the implementation of Erlang, as well as a large part of Ericsson's middleware for building distributed high-availability systems.

#### Features:

**Concurrency** - Erlang has extremely lightweight processes whose memory requirements can vary dynamically. Processes have no shared memory and communicate by asynchronous message passing. Erlang supports applications with very large numbers of concurrent processes. No requirements for concurrency are placed on the host operating system.

**Distribution** - Erlang is designed to be run in a distributed environment. An Erlang virtual machine is called an Erlang node. A distributed Erlang system is a network of Erlang nodes (typically one per processor). An Erlang node can create parallel processes running on other nodes, which perhaps use other operating systems. Processes residing on different nodes communicate in exactly the same way as processes residing on the same node.

**Robustness** - Erlang has various error detection primitives, which can be used to structure fault-tolerant systems. For example, processes can monitor the status and activities of other processes, even if these processes are executing on other nodes. Processes in a distributed system can be configured to fail-over to other nodes in case of failures and automatically migrate back to recovered nodes.

**Soft real-time** - Erlang supports programming "soft" real-time systems, which require response times in the order of milliseconds. Long garbage collection delays in such systems are unacceptable, so Erlang uses incremental garbage collection techniques.

**Hot code upgrade** - Many systems cannot be stopped for software maintenance. Erlang allows program code to be changed in a running system. Old code can be phased out and replaced by new code. During the transition, both old code and new code can coexist. It is thus possible to install bug fixes and upgrades in a running system without disturbing its operation.

**Incremental code loading** - Users can control in detail how code is loaded. In embedded systems, all code is usually loaded at boot time. In development systems, code is loaded when it is needed, even when the system is running. If testing uncovers bugs, only the buggy code need be replaced.

**External interfaces** - Erlang processes communicate with the outside world using the same message passing mechanism as used between Erlang processes. This mechanism is used for communication with the host operating system and for interaction with programs written in other languages. If required for reasons of efficiency, a special version of this concept allows e.g. C programs to be directly linked into the Erlang runtime system.

## Components

OpenSource Erlang comes with several standalone components that can be used as building blocks when developing applications. These components understands Erlang's systems messages (load, unload, start, stop, restart, change code).

**Inets** - HTTP 1.0 server and FTP client.

**Mnesia** - Distributed real-time database for Erlang. Supports RAM-replication as well as disk storage, allows dynamic schema changes, allows arbitrarily complex data structures to be stored. Mnesia is very fast since it runs in the same address space as the applications that use it - this is possible since both Mnesia and the applications are written in Erlang. Mnesia is a nice example of the power of Erlang: in how many languages could you write a fully-featured industrial-strength distributed DBMS in less than 20,000 lines of code?

**Orber** - CORBA v2.0 Object Request Broker (ORB).

**SNMP** - Extensible SNMP v1/v2 agent and MIB compiler.

## Tools and Libraries

OpenSource Erlang comes with a library of useful tools:

**Appmon** - Graphical monitoring of process groups (locally and on remote nodes).

**ASN.1** - Compile-time and runtime package which supports the ASN.1 Basic Notation and the encoding rules BER and PER.

**Compiler** - Erlang compiler.

**Debugger** - Graphical Erlang debugger.

**ERTS** - Erlang runtime system, including the virtual machine, the garbage collector, and the port mapper daemon.

**GS** - Library for writing graphical user interfaces.

**IC** - Compiler from OMG's Interface Definition Language (IDL) to Erlang and C.

**Jive** - Support for calling Java from Erlang and vice versa.

**Kernel** - C code necessary to run the Erlang system: Erlang built-in functions (BIFs); code, boot and name servers; networking and distribution support; loaders, linkers and loggers; OS and file system interfaces.

**MESH** - Measurement handling, including an SNMP MIB.

**Mnemosyne** - Optional query language for Mnesia.

**Mnesia Session** - Foreign languages interface to Mnesia defined in IDL, providing Mnesia access via the IIOP and erl\_interface protocols.

**OS monitor (OS\_MON)** - Monitoring of CPU, disk and memory utilization, including SNMP v1/v2 MIBs. Interfaces to Solaris syslogd and Windows NT event log.

**Parse tools** - LALR-1 parser generator for Erlang (yecc), similar to yacc. Yecc takes a BNF grammar definition as input, and produces Erlang code for a parser as output. Yecc is used to generate the Erlang parser.

**PMan** - Tool for tracing and viewing the state of Erlang processes (locally or on remote nodes).

**SASL** - Progress/error/crash report handling, report browsing, release handling, overload regulation.

**Stdlib** - Libraries for: input/output; incore and disk-based table storage (ETS and DETS); graphs, dictionaries, lists, strings, sets, queues; regular expressions; math. Erlang interpreter, tokenizer, parser, lint and pretty-printer. Generic frameworks for fault-tolerant servers, event handlers, state machines, and process supervisors.

**Table visualizer** - Tool for viewing ETS and Mnesia tables.

**Toolbar** - Simplifies access to the Erlang Tools.

**Tools** - Coverage analyser, profiler, text-based tracer, Emacs mode, Emacs TAGS file generator, make utility, call graph utility.

**Xref** - Provides statistics on cross referencing between modules and functions.

## Appendix B: Summary of Ericsson's High-Availability Software.

The Eddie OpenSource Project is focused on building High-Availability Server Farms.

### What is Eddie?

Eddie is an OpenSource project and a set of high availability cluster applications to be used when building robust and scalable server farms. Eddie is an OpenSource initiative initiated by Ericsson and currently consists of four components, i.e.

- IP Migration Application
- Load-balancing DNS Server
- Intelligent HTTP Gateway
- Content Replication Application

The first three components have been released for Linux as Beta packages. They have been developed and tested on a LAN running six standard RedHat 5.0 hosts (PII, 96MB RAM, 100Mb/s Ethernet). The plan is to remove the beta label and to release a Solaris version real soon. Stay tuned for an NT version.

All Eddie configuration is done using Apache style configuration files and a set of O&M commands.

Eddie has been written using Erlang, a functional language developed by Ericsson. Erlang itself has recently been released as OpenSource.

### IP Migration Application

The IP Migration Application is an all-purpose cluster application making it possible to setup overlapping fail-over sets using any mix of servers. All aspects of the clustering are configurable and call-back scripts can be written to customize start, stop, monitoring and alarm handling. Our goal is to do a better job than currently available cluster solutions.

### Load-balancing DNS Server

The Load-balancing DNS Server is plug-in compatible with BIND. The DNS server uses additional load information to direct clients to servers most suitable (low load and alive). Work is underway to add an Adaptive Response Time Analysis making it possible to direct clients to servers in the "vicinity", in terms of response time values and smart heuristics.

### Intelligent HTTP Gateway

The Intelligent HTTP Gateway is a peer, which sits on top of back-end HTTP servers such as the Apache Web server. The gateway schedules requests to the least loaded back-ends on the LAN using adaptable load-balancing algorithms.

It also analyzes incoming requests, directing certain types of traffic to certain types of hosts using configurable scheduling patterns, e.g. CGI-script requests can be directed to

host A and requests for static HTML pages to host B. The gateway also features advanced admission-control mechanisms.

### Content Replication Application

The Content Replication Application is work in progress. The goal is to build a fully replicated and distributed LAN/WAN file system with robustness as the main objective. Stay tuned.

## Appendix C: Voxzilla.org Project.

From their website, currently unavailable at <http://www.voxilla.org/> we see the following info:

### Aims of the Voxilla H.323 Project

The aim of the Voxilla Linux H.323 project is to facilitate the creation a H.323 protocol implementation by coordinating the efforts of interested individuals or companies.

All source code contributions to the project must meet the requirements of OpenSource.

Once this implementation is available, it can form the basis of a family of OpenSource Internet conferencing and telephony products.

### How this project got started

Equivalence had been contemplating starting an OpenSource H.323 project for a few months, when we saw the item in slashdot.org about Rich Bodo's Voxilla Project.

So we contacted Rich straight away and asked him if he would be interested in our contributions. He said "Yes!!".

### Who is Equivalence?

Equivalence is a company owned by Craig Southeren and Robert Jongbloed. Between us, we have been developing commercial software for Unix, Windows, and Macintosh for more than ten years. All of our products are available for both Unix and Windows, and use our own multi-platform C++ code library.

Among our products is a proxy server for the H.323 protocol. Development of this product required the development of an ASN to C++ processor, and becoming familiar with the H.323 protocol suite.

Equivalence will be making the source code of our C++ multi-platform code library available under an OpenSource license. We will also contribute the Linux/X86 binary of our ASN preprocessor, and will provide source code for a bootstrap H.323 application that should get the ball rolling.

We will contribute more code as time goes on, and try to act as a coordinator for any additional efforts.

### Suggestions for the first program

Any H.323 application needs a protocol stack. Not suprisingly, we feel that the initial aim of this project should be to produce such a stack.

As a suggestion, we propose working towards a H.323 "parrot". This simple application should receive a H.323 call, record any voice data until the call is terminated, and then call back and playback the recording.

This will require implementing most of the H.323 protocol stack, including all of the call originating/answering PDUs, as well as capturing and replaying audio data in real time.

And what then.

Once a protocol stack exists, anything is possible! Possible applications include:

- H.323 to PSTN gateway
- NetMeeting clone (GNU/Meeting?)
- Video gateways

Any ideas?