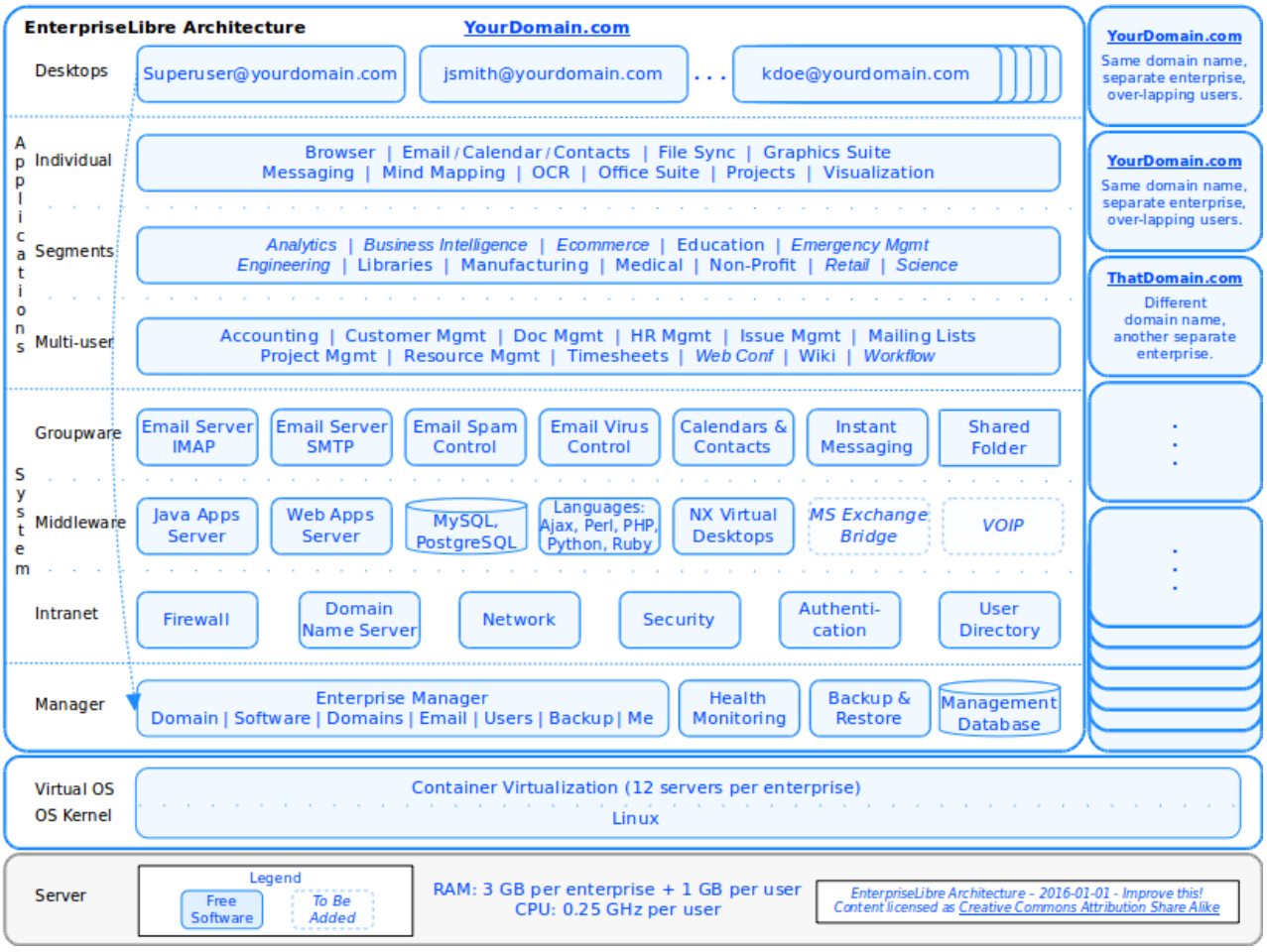
EnterpriseLibre:
A Virtual Organization For Everyone

William Stewart, Ph.D.
Cirrus Computing System Architect

2016-08-01

Video Demonstration: https://youtu.be/rv9IIx1vkPQ



1. Executive Summary

> *I really worry about everything going to the cloud – you don't own anything.* – Steve Wozniak; Apple Co-founder, 2012.

EnterpriseLibre packages the best open source software into a complete enterprise that replaces all proprietary software - system, desktops, 28 apps so far - for almost any small or medium size enterprise. The whole system can be managed by anyone from a simple GUI.

Only open source software can provide this. It's mathematically impossible with proprietary software. The *O(N^2)* increase in complexity with the number of components is just the start. Lock-in is the end. EnterpriseLibre proves that open source can deliver an integrated, full-stack, enterprise solution, with automation and robustness impossible with proprietary code.

And in a fractured online world, it's uniquely secure, as secure as a hosted solution can be. Why? First, when software is onsite, protected by a building, a room, perhaps even a locked cabinet, it starts as secure as it can be. All twelve of the EnterpriseLibre virtual servers in an enterprise are co-located, usually on the same server. They are all on a dedicated secure network, again all contained in a small physical space. All applications and databases are inside those virtual servers. So when John in San Francisco sends an email, instant message, or file to Jane in London, it moves data only a few centimeters – at most.

Then to provide distributed access, open up just one port and protocol – 22 and NX – and encrypt the connection to enable access to the computer desktops, also running on that central server. You now have the most secure possible onsite solution, with access over the Internet on the functional equivalent to long, encrypted monitor cords.

This gives you onsite power – real computer desktops, full-featured desktop applications, Intranet integration – and secure distributed access as well. Even though a desktop might look like it is running on a local device, it is just an encrypted picture. There is no way to attack any internals. For example, when John and Jane work on their enterprise Wiki, no matter where they are in the world, there is no SQL injection attack vector. The Wiki, although it appears to be running locally, does not exist on the rest of the Internet. (To start with, unless you click a button to open it.) John and Jane access a flawless image of the Wiki, through an encrypted NX tunnel, open to access only by their authenticated human fingers.

The *only* secure hosted architecture is centralized, with an encrypted desktop. Multiple Web apps and multi-tenancy put genuine security permanently out of reach. Only the virtual desktop can provide full functionality and maximum security at the same time. EnterpriseLibre shows the combination is achievable, at enterprise scale.

And now anyone can use it. A simple GUI enables anyone to manage their enterprise with an ease never available before, from domain to desktops. And this release has gone meta: a pre-built hosted enterprise that can manage itself, *and* create and manage others. This enables anyone to host an enterprise solution locally, and create and provide them for others.

The architecture is clean, from virtual servers up, so runs on almost any local hardware, or hosted service including Amazon and Google cloud. It's near optimally efficient, inexpensive and green, able to add geographic liquidity to carbon markets. (Cirrus Computing uses one data-center that runs on all green power). And it's production proven, supporting several customers for several years, hardened, and ready for wider use. EnterpriseLibre makes it

easy to get open source software to the *majority* of the world's users – small and medium-size enterprises.

Key Links

5 min video demonstration: https://youtu.be/rv9IIx1vkPQ

GitHub source:
https://github.com/CirrusComputing/EnterpriseLibre

Contents

Graphics & Tables

2. Scope

> *It may be the devil or it may be the Lord, but you're gonna have to serve somebody.* – Bob Dylan, 1979.

This section orients EnterpriseLibre by defining the customer, objective, requirements, and alternatives.

2.1 Customer

The customer is anyone that needs software for a team of two or more.

2.1.1 Team Needs

They need the best solution for two or more people working together, which means a dedicated, secure network, servers, storage, system software, desktops, a complete set of horizontal applications, and vertical segment applications as required.

2.1.2 Nothing But Net

Their users could be located anywhere – in one room, several buildings, many countries, home offices, traveling – and using any kind of device, connected to the Internet.

2.1.3 Not Technical

Like most people, they want the benefits of software, however are experts in something else, so likely don't have any technical expertise.

2.2 Objective

The objective is to provide a complete solution to everyone everywhere.

2.2.1 Complete Solution

Integrate the best open source software at all levels of the stack into a complete solution providing the fullest possible software value.

2.2.2 Universal Availability

Make the solution available to everyone, everywhere.

3. Requirements

> *If you try sometimes, well you just might find, you get what you need.*
> – Rolling Stones, 1969.

This section describes the customer's requirements, the needs of the typical person looking for software to support a team, baselined from experience, research, and interviews, and refined with feedback from Cirrus Computing customers, the largest with 17 or more desktops for more than three years now.

Introductory paragraphs are descriptive. All other sentences containing the word "must" are requirements. There are four areas, nine categories, and 27 primary requirements. Two requirements, Single-Sign-On and Open Source Software, are needed twice. Nine blocks stand in the way, four more than once, listed in each section in alphabetical order for consistency.

The 27 primary requirements are summarized below. With one exception, the current release of EnterpriseLibre meets them all :-). A cross-reference of requirements and blocks to the architecture principles that resolve them can be found in *Section 7 – Traceability*.

Usability

- Ease Of Use
  - Virtual Desktop
  - Application Simplicity
  - Single-Sign-On
- No Maintenance
  - Full-Time Availability
  - Hosted Convenience

Security

- Confidentiality & Integrity
  - Single Tenant
  - Secure Network
  - Secure Channels
  - Physical Protection
- Mobile Privacy
  - End-To-End Encryption
  - Single-Sign-On
  - Two-Password Login
- Certifiable Assurance
  - Proven Design
  - Hardened Components
  - Open Source Software

Capability

- Full Functionality
  - Integrated Solutions
  - Best Components
  - Any Application
- One-Click Management
  - Enterprise Automation
  - Scalability Automation
  - User Automation

Sustainability

- Long-Term Use
  - Open Source Software
  - On-site Or Hosted
  - Natural Evolution
- Green I.T.
  - Computer Recycling
  - Energy Efficiency
  - Carbon Offsets

3.1 Usability Requirements

> Software must be fun, so everyone will use it.

Most people are experts in something other than software. To get everyone on a team to use software, it must be fun – easy to use, and always available.

3.1.1 Ease Of Use

> Software must be easy, so everyone *can* use it.

The main requirements and blocks for ease of use follow.

3.1.1.1 Virtual Desktop

- Users must be able to access their full computer desktops over the Internet from any device ("virtual desktop"), so they can work with all their software and files anywhere.
- Virtual desktops must be accessible from any of the main platforms – Android, iOS, Linux, Mac OS, MS Windows – to be usable by anyone, anywhere..
- Virtual desktops must perform with near real-time usability, i.e. less than **1/5'th** of a second perceptual delay[1] on any network in the world with at least downstream bandwidth of **128 Kbps** (0.128 Mbps) and upstream bandwidth of **28 Kbps** (0.028 Mbps), and a latency of **200 ms** (0.2 seconds) or less between user and solution.[2]

3.1.1.2 Application Simplicity

- Applications must provide all capability with each version through one optimized user interface, with finer controls implemented through modules, roles, and permissions.
- Application functionality must be driven by user needs, with enhancements prioritized by value for effort, and new features introducing as little change as possible.
- Applications must use standard formats and protocols wherever available, and include an easy way to export information in an open format for use with other applications.

3.1.1.3 Single-Sign-On

- Multi-user applications must transparently authenticate users with secure single-sign-on from their desktop, starting up already logged in to their account.

3.1.1.4 BLOCKS

[1] Kjetil R., Ragnhild E., Carsten G.; *Can gamers detect cloud delay?*; Westerdals Oslo School of Arts, Simula Research Laboratory; IEEE 978-1-4799-6882-4/14.

[2] With removal of extra X11 calls, NX software performs magically well up to 400 ms latency.

The main factors blocking fulfillment of the requirement follow.

3.1.1.4.1 Proprietary Problems

Proprietary software introduces three problems blocking value: technical limitations, conflict of interest, and licensing complexity. The first two are described here, the third in *Section 3.2.1.4* under Full Solutions, the requirement it impacts the most.

The first problem, *Technical Limitations*, arises from the proprietary source – the inability to read or change the code. That makes it very difficult to provide great usability with a complex, powerful enterprise solution. For example, it is impossible to integrate a proprietary component with single-sign-on if it doesn't provide a standard interface, and of course workarounds defeat the security. For more than a small number of apps, unless they are all from the same vendor, it is very difficult to implement and maintain single-sign-on over time. There is no way to modify the code if needed, or even read it to figure out how to best use it.

The second problem is Conflict *of Interest*, since the complexity of software makes it easier for vendors to increase revenue at customer expense, often decreasing usability. If some software was a car, every release would change the design and location of the lights, doors, windows, instruments, gear shift, pedals, and steering wheel. When the users just wanted it to be faster and more comfortable. This continuous change is driven by a search for revenue well after the application has passed through most of it's innovation cycle. Widely known examples of needless software change include each Microsoft Windows desktop (how do I find a file again?) and MS Office suite (if it's easy, make it more difficult).

On the other hand, some proprietary software rarely changes, even though there is a good user base and widely desired enhancements, if it has a monopoly position but little revenue potential, since the vendor has no interest in assigning development resources to make the update. For example, Microsoft Outlook did not add the ability to search any contact field until many years after it was available in FLOSS apps like Evolution and Thunderbird.

At the application level, if the first priority is functionality, the second is often limiting usability by generating financial transactions to remove blocks, such as increasing the allowed user count, or enabling a visible but greyed-out feature. These conflict of interest problems become part of the proprietary software itself, of its design and how it is built and even tested. This negative-value work might once have been required to keep a captive development team to support proprietary software, however today the real cost remains very high, and significantly limits its use and benefits.

3.1.1.4.2 Virtual Desktop Stasis

The blazing performance of the NX virtual desktop protocol has been available for Linux since 2003, when Gian Filippo Pinzari and NoMachine developed and released the core technology as FLOSS, keeping them open source through V3.*, a great benefit. FLOSS NX servers based on this work include FreeNX, neatx from Google, Remmina, SPICE from RedHat, and x2go. Good FLOSS NX clients include OpenNX, qtnx, and those from Remmina, Spice, and x2go. The current version of EnterpriseLibre uses server and clients from X2Go.

The main block to wider use for many years has been simply that the main alternative, the higher bandwidth graphics-based VNC, has been easier to use and good enough, especially for single-user use. So, without a scalable multi-user FLOSS requirement, most NX use has been in larger enterprises using the highly performant NoMachine software, with more use recently of X2Go.

With support from Amazon and others, virtual desktops have been growing in use over the past few years.

3.1.1.4.3 Web 2.0 Diversion

The great Web 2.0 diversion – the multi-decade attempt to to defeat the laws of physics by replacing the operating system with the web browser – started well-meaningly, with Netscape Navigator in 1994. Navigator was the 15'th browser, however built by Marc Andreessen and much of the team that developed the first popular browser, Mosaic. Navigator and Mosaic were purposely designed with great GUI separation, so the same software ran on Linux, Mac, and Windows, making them the first popular "cross-platform" application. This raised the gleam in many an eye of replacing MS Windows... from above. The scale of the prize blinded them to the technical difficulty, and two decades later the search continues from many attempts to better Ajax to Google's Chrome OS.

However, mainly the idea turned into Web applications, briefly called "Web 2.0", ending up with Google Docs and Salesforce and Zoho, where hosting and mobile access outweighed the interface deficiencies of the Web browser. Large advances in local browser capability have been achieved with Ajax, HTML5, Java, JavaScript, and even PHP. However, the problem remains physical: instructions for a process have to run on one side or the other. We can move them from the server, however at some point just end up running everything on the local device again – OS, middle-ware, and application – losing the advantages of the hosted solution, and ability to meet many other customer requirements.

The insurmountable advantage of the virtual desktop will always be how much faster it is to run an instruction on a server compared to the time it takes light to travel from the client to the server. For example, the time between instructions on a chip is vanishingly small, so use the full cycle-time, for a medium-power **3 GHz** chip about **3.33 x 10^-10** seconds. On the other hand, the minimum time for an Internet packet to go halfway round the earth is **0.0668 seconds**, or **200,519,027** times longer. If the access device is only a kilometer from the hosted solution, a packet can make the trip in only **3.3 x 10^-6 seconds**, but that is still **10,007** times longer than time between instructions if all on the server. And this advantage is before instruction pipelining, and the continual improvement in hardware performance.

In other words, we can take round-trips out of the abstracted GUI layer, but not the bottom process layer. We can take duplicate calls out of X11 and send the process results anywhere in the world with near real-time performance. However, to take out "duplicate calls" in the actual process itself, running on a remote server, from a local device, is not not computationally possible. The best guidance is always Torvald's Principle – don't try to outperform the Linux kernel.

Since the GUI is separate, running independently in parallel, it can distribute the process results as they become available. This design can easily meet the relatively undemanding requirement of human perception, since the maximum - minimum of **0.067 seconds** between any hosted solution and user on our planet is well within human acceptability. And that's before NX does its optimization magic. The best solution – keep all process and data in one place, and distribute the interface as lightly as possible – is now workable everywhere.[3]

Therefore, while it's impossible to re-architect fifty years of software to fit over HTTP, there's no need either. The tipping point in appreciation the virtual desktop leapfrogs Web apps can likely be traced to their addition to Amazon Web Services in 2013. Use of both RDP and NX

[3] As the (proprietary) NoMachine V4 shows, you can get very close to a comparable NX client in a browser, with significantly more CPU and bandwidth, however, a real "zero-client" virtual desktop.

have since continued to grow.

3.1.2 No Maintenance

> Software must be completely self-maintaining,
> so it can be used by anyone, anywhere, anytime.

The main requirements and blocks for no maintenance follow.

3.1.2.1 Full-Time Availability

- Other than applications where innovation is faster and there's more tolerance for bugs, software must run full-time without problems, and if any new issue is ever discovered, it must be resolved for all current and future users, for always increasing reliability.

3.1.2.2 Hosted Convenience

- Software must be hostable in a version with transparent management of all infrastructure, so anyone, anywhere can obtain the full benefits with only an Internet connection, and can easily move their solution if ever desired or needed.

- Hosted solutions must have the same functionality as the onsite version, and be accessible from any of the main client platforms – Android, iOS, Linux, Mac OS, MS Windows – to be usable by anyone, anywhere.

3.1.2.3 BLOCKS

The main factors blocking fulfillment of the requirement follow.

3.1.2.3.1 Custom Software

Any custom software needs more maintenance as it evolves to address functionality and security issues, perhaps eventually over a period of years becoming polished and hardened.

FLOSS solutions have much less need of expert attention, since they avoid licensing issues and have fewer security problems. However custom integrations still require more maintenance for the same reasons, a need for time and use to evolve naturally. While easier than before, establishing a relationship with an expert to build and manage a custom FLOSS enterprise solution, even when hosted, is a high enough barrier it has significantly limited wider use by small and medium size teams. The hosted solution solves part of the problem. An integrated FLOSS enterprise that gets monotonically better over time would solve the rest.

3.1.2.3.2 FLOSS Skepticism

Especially early, there was skepticism in the FLOSS community about the benefits of hosted solution, since most of the offerings were proprietary, multi-tenant, stored data who knew where, and only added more locks to existing lock-in.[4]

[4] *Free software is a means to an end, and the end is freedom. What good is it if free software gets a "wider audience" if it doesn't mean freedom for that "audience"?* – Richard Stallman; Personal email to W. Stewart; 2011-02-24.

However, with different inputs – all FLOSS, single-tenant architecture, trusted providers – hosted solutions can be quite helpful: a solution to the largest barrier to wider use of FLOSS, need for local infrastructure and assistance. With virtual desktops, the barrier becomes a channel, making almost any FLOSS that runs on Linux instantly accessible to anyone, anywhere in the world. As per the Objective, EnterpriseLibre is designed, especially in its hosted versions, to be so *conveniently* accessible the intrinsic value of FLOSS can become obvious world-wide more quickly, helping increase ecosystem for the benefit of all.

3.1.2.3.3 Local Infrastructure

While acquiring and maintaining local hardware – servers, storage, backup – is easier all the time, it still requires significant time and resources. One modern appliance server can easily support up to **100 users**, however must still be "production" quality: high-performance, reliable, energy optimized, and more expensive, and it will need some kind of manual setup and ongoing system maintenance.

These are high barriers, so many small and medium-size teams often make do with email, a few MS Office apps, maybe a couple Web apps. Even when they are larger, the maintenance overhead for local infrastructure continues to be a constant brake on the software capability they use and refresh over time.

3.1.2.3.4 Proprietary Problems

All the proprietary software problems described elsewhere block the No Maintenance requirement.

The *Technical Limitations* problem (*Section 3.1.1.4*) makes it very difficult to provide a completely self-maintaining solution at scale, since proprietary code make it unfeasibly difficult to integrate many components to run bug-free over time.

The *Conflict of Interest* problem (*Section 3.1.1.4*) can drive maintenance when vendors have an interest in an ongoing services contract, either direct or through accredited providers.

The *Licensing Complexity* problem (*Section 3.2.1.4*) makes it effectively impossible to remove all maintenance, since the larger the enterprise and more powerful the solution, the more frequently licensing will need updating.

In practice, the resources and complexity required to solve these problems, by establishing an ongoing relationship with support provider, are a high barrier that significantly limit powerful solutions from reaching many teams that could benefit from them.

3.2 Capability Requirements

> Software must provide as much help as technically possible to reduce effort, improve results, and provide new capabilities.

A powerful enterprise software solution that integrates the best components, and can be managed at the click of a button, multiplies the effectiveness of a team several fold.

3.2.1 Full Functionality

> Software must provide full functionality solutions that integrate the best components, and make available any other software users desire.

The main requirements and blocks for full functionality follow.

3.2.1.1 Integrated Solutions

- Since no one component can provide all capability well, the best enterprise solution must include the most capable software in each category, then provide integrated value that is more than the sum of its parts.
- Enterprise solutions must automatically synchronize all needed information between components, such as domain names, name and email servers, email addresses, user names, and others, to increase accuracy, efficiency, and security.

3.2.1.2 Best Components

- All software (other than applications where user choice and innovation are the primary drivers) must be the standard in its category, or, if there is not yet consensus, a leader by user share, to provide the best sustainable capability – if users want it, it will come.

3.2.1.3 Any Application

- All users must have access to all standard individual-use applications with their virtual desktop, and all other multi-user, segment, and custom applications must be available.
- Solutions must be able to host any software that runs over IP, if the Superuser has enabled access external to desktops, including but not limited to HTTPS, IMAP, OWA, SMTP, TCP, UDP, to be capable of supporting any current or future Internet application.
- All users must have at least minimum access accounts with all applications, so everyone can gain the benefits of team-wide use.

3.2.1.4 BLOCKS

The main factors blocking fulfillment of the requirement follow.

3.2.1.4.1 Proprietary Problems

The first two proprietary software problems described in *Section 3.1.1.4*, Technical Limitations and Conflict of Interest, greatly restrict the capability of enterprise solutions. Particularly Technical Limitations, since synchronization of information can be impossible if a proprietary component doesn't provide an API, as when QuickBooks notoriously encrypted their database. These difficulties can be almost as bad when the component modifies or "enhances" the standard. Sometimes a component has an "undocumented" API which changes unpredictably with each update. And an enterprise solution can require close to *O(N^2)* interfaces if every application needs to share data with all others, until it becomes preferable to develop a central component that translates to an internal format and handles all communication, e.g. HL7 for medical messaging[5].

However, the third problem introduced by proprietary software, *Licensing Complexity*, is the worst, since it becomes the show-stopper with more than a few components, which is the point of an integrated enterprise solution. And the problem can be quantified: there are no solutions built with proprietary software approaching the capability of EnterpriseLibre because an O(*N^2*) increase in licensing complexity with the number of components *N* quickly makes the time and cost impractical for very low values of *N*.

Technical limitations might be solved, but licensing blocks are forever, and multiply with the number of components, interfaces, and solution life-cycle. Every point that requires a licensing interaction throughout development, integration, testing, deployment, and maintenance, even just to install an old component for support testing, brings action to a stop, which is relatively speaking an infinite decrease in momentum. If both a technical and financial contact are required, the delays multiply. What should be an automated, zero-time action can grow to minutes, days, and even weeks, causing schedule and budget to balloon.

And the more capable a solution gets, the more this happens. An enterprise solution can require dozens of iterations for integration and testing, sometimes needing new builds several times a day. With more components, connections, dependencies, points of failure, and impacts from change, the complexity growth is not exactly the same as the number of edges *E = (N*(N-1))/2* in a graph of *N* nodes, or *O(N^2)*, however it's close enough. And a truly complete solution like EnterpriseLibre uses two dozen best-in-category components at the system layers alone, before adding multi-user enterprise applications.

So, in practice, the capability of enterprise solutions built with proprietary software is limited by math. Every new component costs more to add, and the resources consumed just by management of the complexity rises. These multiplicative impacts can be seen in almost any large government or public enterprise multi-component proprietary solution once normalized as a per user cost. Almost all this complexity can be resolved, factored away, with a clean architecture, good design, and proven components - the precise power of software – if it were not for the manual licensing blocks introduced by proprietary code.

3.2.1.4.2 Web 2.0 Diversion

The fracture of software across the Web makes all capability requirements more difficult to provide, and often impossible, for example as seen in attempts to find a single-sign-on solution for Web apps with the same security as the standard desktop implementation.

And whenever multiple usernames and passwords are required, especially if they have

[5] Team experience indicates transition to a central component to reduce connections from *N* to *N^2* happens around *N* ~ 4, when a distributed architecture needs 4 * (4 – 1) / 2 = 6 connections, whereas a central architecture requires only 4, and increases accuracy and decreases maintenance.

different change cycles, some users will find the complexity just too high, and save them in their local Web browser, leaving them on every local device they use.

3.2.2 One-Click Management

> Software must be manageable by anyone, transparently handling all underlying configuration, so it can respond immediately when needed.

The main requirements and blocks for one-click management follow.

3.2.2.1 Enterprise Automation

- Enterprise solutions must provide a single, easy interface where anyone can manage all system settings, including domain name, name server, email servers, email aliases, backup services, main timezone, and reboot of desktops and whole solution.

- Enterprise solutions must provide a single, easy interface where anyone can manage their security settings, including reset of user passwords, whether 2-factor authentication is required for specific users or the whole team, which applications and services are accessible outside virtual desktops (including incoming and outgoing email), use of security protocols such as HTTPS, and ability to change domain names of any externally accessible service with instant disabling of access by the old names.[6]

3.2.2.2 Scalability Automation

- Enterprise solutions must provide a single, easy interface where anyone can manage user accounts over the life-cycle, including: addition given just their desired email address (e.g. "jsmith") causing automatic creation of their virtual desktop, email, calendars, application accounts, and emailed notification of their login information, in less than five minutes; suspending an account including virtual desktop and all software and files with instant effect; and resuming an account with instant effect.

- Enterprise solutions must provide a single, easy interface where anyone can manage their applications over the life-cycle, adding and removing software with near instant response, and transparent handling of all aspects of installation and subsequent maintenance of system information such as domain names and email servers and user accounts as they change.

3.2.2.3 User Automation

- Enterprise solutions must provide a single interface where any user can easily manage their personal user information, including first and last names, system username, email address, password, notification email, and requirement for two-password login (if user level change has been enabled by the Superuser), .

3.2.2.4 BLOCKS

The main factors blocking fulfillment of the requirement follow.

[6] Instant disabling of old email domain names is the only requirement EnterpriseLibre doesn't fully meet, since there is an IP address workaround to be addressed in an enhancement.

3.2.2.4.1 Proprietary Problems

Vendor specific management solutions (e.g. for Microsoft environments), and standards based monitoring software can help manage proprietary software at scale, however the proprietary software and licensing barriers make it unfeasibly complex to provide one application that interfaces with all components and automatically implements the one-click functionality described in the requirements.

So, in practice, the need to manage software components one-by-one multiplies the time required several times, which causes many small and medium-size teams to significantly decrease the amount of software they use, and the number of users they provide access, just to limit the management overhead.

Proprietary software services also benefit financially from the sole provider premium, for example where only the application vendor can effectively provide consulting, or switching the support provider would cost too much in disruption. Large vendors like Microsoft formalize certifications for management of their products, in effect absorbing external IT personnel to help lock-in use of their software.

3.2.2.4.2 Web 2.0 Diversion

Many Web applications are popular in large part because they don't need local infrastructure, and are simple to manage, individually. However, a complete solution needs more than one application, from more than one vendor, and so requires some kind of manual setup (e.g. all users probably don't get all apps), and maintenance as users and other information changes. In practice, the fracturing across the Web makes it unfeasibly complex to provide a single, sustainable GUI with the kind of enterprise capability described in the requirements.

3.3 Security Requirements

> Software must be hermetically secure, invisible outside the team,
> and usable from anywhere with complete assurance.

Genuine security increases user confidence and team benefits, and is increasingly required for meaningful certification and accreditation for government, organization, and professional use.

3.3.1 Confidentiality & Integrity

> Software must be electronically and physically protected
> from access by anyone outside the team.

The main requirements and blocks for confidentiality & integrity follow.

3.3.1.1 Single Tenant

- Enterprise solutions must provide dedicated, independent software for each customer at all levels of the stack, so process and data are completely separate from all others.

3.3.1.2 Secure Network

- Enterprise solutions must be protected by a dedicated firewall, configured by default to block all communications not specifically enabled.
- Enterprise solutions must include a dedicated name server, configured by default in silent mode so only valid domain names return an IP address.
- Users must be able to easily control whether applications and services are accessible outside the virtual desktops, and be able to change their domain name with instant effect, e.g. from "wiki.mydomain.com" to "secret.mydomain.com".
- Delays must be added to unsuccessful login attempts to prevent automated attacks, and password resets must send a link for the user to choose their own new password.

3.3.1.3 Secure Channels

- Enterprise solutions must provide secure channels for outside experts to provide assistance with applications, providing direct access while maintaining security[7].

3.3.1.4 Physical Protection

- Enterprise solutions must be protected against unauthorized physical access, and hosted software must offer customers the option of controlling the locks on their cabinets, and adding any other levels of physical security desired.
- Hosted software must be safeguarded against damage from fire and water, and have

[7] Cirrus Computing customers have needed secure channels for assistance with CRM and ERP.

secondary power and network connections.

- Complete backups must be taken at least daily and a secure copy kept off-site, and, in case of disaster, be recoverable on a new server in any location in less than one hour.

3.3.1.5 BLOCKS

The main factors blocking fulfillment of the requirement follow.

3.3.1.5.1 Local Infrastructure

Local solutions unavoidably include the requirement for local physical protection, which can be a significant cost for small and medium size teams, and sometimes impossible – for example installation of fire suppression systems in a leased office. Even maintenance of a reliable, secure off-site backup can be hard to ensure over time.

3.3.1.5.2 Multi-tenant Illusion

The illusion that multi-tenancy – complexifying an application to support more than one customer – is needed for online solutions started with Salesforce in 1999, providers of the first popular Web app, for customer relationship management (CRM). Copies of support software for each user would have cost too much. So development to support multiple customers in one application was overall less expensive, even though inefficient and took more hardware.

Salesforce received a patent on multi-tenancy, and have promoted it ever since as the secret sauce for hosted software, thus a powerful barrier to competition.[8] Even though the savings are only licensing, caused by proprietary software in the first place, somehow many became convinced the advantage transfered to processing, that Salesforce had figured out how to make the application layer faster than the operating system (*Section 5.1.6 – Virtualization*).

For much the same licensing reasons, all mainstream hosted apps from Google, Microsoft, Salesforce, Zoho, etc., are multi-tenant, massive monolithic systems serving many customers. And, to be fair, it would not be cost-effective to provide a full enterprise solution like EnterpriseLibre with proprietary software.[9] However, when the top of the architecture isn't the Internet best practice – a domain name – to which many applications are attached, but the reverse, inevitably a lot of custom code will be required with an inevitable reduction in security. And all separation between customers is hidden from view, application to storage, baseline and updates, a very high risk, you-have-to-trust-me situation.

3.3.1.5.3 Web 2.0 Diversion

Enterprise solutions cobbled out of multiple web apps put security completely outside the customer's control, usually making uncertain even location of their data. Any other protections the customer may wish to add, such as locked cabinets or storage encryption, are similarly impossible if their solution is fractured into pieces across the Web.

3.3.2 Mobile Privacy

[8] *Put crudely, if you can run everything on a single software instance, you only have to buy one software license.* – Wikipedia Multitenancy, 2015-05-02.

[9] *Savings... can be eclipsed by the difficulty of scaling... Development... is more complex... Security testing is more stringent owing to... multiple customers data... co-mingled.* – Ibid.

> Software must be securely accessible with any network connection, so users can work anywhere with privacy.

The main requirements and blocks for mobile privacy follow.

3.3.2.1 End-To-End Encryption

- Mobile use must be protected by strong end-to-end encryption for initial authentication and all communications, so users can securely connect with any network anywhere.

3.3.2.2 Single-Sign-On

- Once a user is logged into their virtual desktop, the enterprise solutions must single-sign-on to all multi-user applications, so only one login and password are needed.

3.3.2.3 Two-Password Login

- Accounts for the whole team or specific users must be configurable to require two-password login, requiring entry of a random PIN sent to their notification address following regular login to confirm the identity.

3.3.2.4 BLOCKS

The main factors blocking fulfillment of the requirement follow.

3.3.2.4.1 Network Vulnerability

Any hosted software without good encryption for all communications is vulnerable on every Internet connection, since anyone with access to each segment of the Internet the traffic passes over can intercepting usernames, passwords, email, and other communications.

If a wireless connection is unencrypted, everyone nearby can read the traffic. However, even if encrypted, the network administrator and anyone else with system access to the router can decrypt the traffic. Either way, anyone with access to any *other* segment of the Internet the traffic traverses can read everything on that section. Therefore, encryption must be under the users control, end-to-end, 100% secure, or reliable assurance cannot be provided.

3.3.2.4.2 Web 2.0 Diversion

Web apps are only as secure as the browser. And if there is are several, each with a separate login, the chances for interception increases, and usability difficulty increases use of local software and files. Multiple Web apps also make two-password login impossible, since there is no single authentication point. And without a virtual desktop sitting underneath the Web apps, multiple copies of files will inevitably end up on multiple devices.

3.3.3 Certifiable Assurance

> Software must be independently reviewable, buildable, and testable, to provide meaningful security certification and accreditation assurance.

The main requirements and blocks for certifiable assurance follow.[10]

3.3.3.1 Proven Design

- Enterprise solutions must use proven designs and standards with the widest use over many years, so there is the greatest likelihood security deficiencies have been fixed.

3.3.3.2 Hardened Components

- Enterprise solutions must use hardened software components with widest use over many years, so there is the greatest likelihood security deficiencies have been fixed.

3.3.3.3 Open Source Software

- Enterprise solutions must use all open source software with the widest use, so most security issues have already been resolved, it can all be reviewed by anyone, and is independently buildable and testable from source into the running binary version.

- Server virtualization must be managed by an open source software operating system, above the hardware so it can be reviewed and updated if needed, and below the system and application levels so the security separation has kernel-level assurance.

3.3.3.4 BLOCKS

The main factors blocking fulfillment of the requirement follow.

3.3.3.4.1 Custom Software

Unless required for reasons such as performance, custom design and software adds several times the complexity, and several times the potential for vulnerabilities that aren't discovered for a long time – by the good guys anyway. And every update of custom software requires much more review to maintain any security assurance. In practice, any custom element significantly limits the level of security certification any solution can meaningfully obtain.

3.3.3.4.2 Proprietary Problems

Proprietary software can hide vulnerabilities for a long time, since the code cannot be reviewed by anyone outside a small group of company developers. Unpatched vulnerabilities can be exploited for considerable periods since the good guys don't know about them (e.g. the many viruses exploiting Internet Explorer). There can also be back-doors inserted by the vendor or governments (e.g. the hard drive firmware breach of 2015).

And from the perspective of a security reviewer, the vulnerability of a solution with N proprietary components each with V% vulnerability rises at the rate of 1-((1-V)^N). For example, with five components each with a 10% vulnerability, the overall solution vulnerability is **41%**, and if they have a 20% vulnerability, the overall vulnerability rises to **67%**. Thirty components with a 10% vulnerability combine for a solution vulnerability of **96%**.

While considerable effort has been invested by governments and other organizations over

[10] See *Section 6* for compliance with Government of Canada safeguarding of PROTECTED B information.

decades to develop assurance processes, unfortunately meaningful assurance for proprietary software remains more difficult than ever to obtain at practical time and cost.

3.3.3.4.3 Web 2.0 Diversion

Web apps put meaningful security assurance out of reach, with all the problems blocking Confidentiality and Integrity and Mobile Privacy making certification or accreditation to formal standards for serious use effectively impossible.

3.4 Sustainability Requirements

> Software must be sustainable for users and their environment.

A team's software and the processes and information they create are their most important non-human assets, so they need to know they will be able to count on them over time.

3.4.1 Long-Term Use

> Users must be able to count on their software long-term.

The main requirements and blocks for long-term use follow.

3.4.1.1 Open Source Software

- Enterprise solutions must use all open source software, so customers can build their own systems and maintain them themselves without lock-in of any kind.

3.4.1.2 Onsite Or Hosted

- Enterprise solutions must be capable of running onsite on any commonly available type of computer (assuming sufficient resources).
- Enterprise solutions must be capable of running on any common hosted platform, including AWS, KVM providers, Mesos, and OpenStack.

3.4.1.3 Natural Evolution

- Enterprise solution architecture, design, and implementation must evolve naturally with continuing innovation in all layers of the software stack, and be able to update capability with minimum complexity and impacts on the existing software.

3.4.1.4 BLOCKS

The main factors blocking fulfillment of the requirement follow.

3.4.1.4.1 Hosted Lock-In

The lock-in with proprietary hosted software is nearly absolute. The vendor can unilaterally change the functionality, remove any part or all, and increase the per user cost as the customer grows. It is usually difficult to impossible to setup the same solution locally. There is almost never an option to obtain all the source code, and, when there is an escrow arrangement, it rarely keeps up with changes, and quickly becomes obsolete.

3.4.1.4.2 Local Infrastructure

Whether proprietary or open source software, if a solution has distributed molecules –

desktops, laptops, tablets, smartphones – the challenge is unavoidably significant, and will always require significant resources. Open Source software solutions are easier to manage over time, with longer support for older hardware, however by nature of managing many physical objects, require constant attention to provide long-term use. It's a perfect case for a distributed approach. With dozens of different kinds of devices, changing rapidly, only giving up completely and letting users pick any access device and platform they wish, then making the desktop accessible over the Internet, can work.

Any hardware-level dependencies are the ultimate lock-in risk. If a server, disk, or software component can only be obtained from one company, with forced purchases to maintain compatibility with other components, a bad situation can only get worse. Even a computer chip little-endian big-endian change can be resolved with a Linux kernel rebuild. Anything more than that puts long-term use at very high risk.

3.4.1.4.3 Proprietary Problems

When proprietary software is onsite, license locks of several kinds can make it unusable in many ways: for new users, on new computers, while mobile, communicating with other software because of a required cascading upgrade, or simply after a period of time like it rusts out.

Even in larger organizations with more resources, the licensing and management complexity of maintaining a standard proprietary environment – from operating system to applications – increases non-linearly with users, devices, and time. Even in the largest organizations, giving all users all applications still costs too much, and then provided sometimes has concurrent user caps that effectively amount to the same limitations.

3.4.2 Green I.T.

> Software must have minimum effect on the environment.

The main requirements and blocks for green I.T. follow.

3.4.2.1 Computer Recycling

- Enterprise solutions must be usable from any kind of old computer, possibly with a Linux install as a simple Internet access device, to greatly decrease the costs of client acquisition, replacement, and upgrade.

3.4.2.2 Energy Efficiency

- Enterprise solutions must be designed and built to be as efficiency optimized, requiring the least CPU processing, to use as little electricity as possible.[11]

- Hosted solutions must enable easy load-balancing so the hoster can reduce infrastructure and energy use over local installations by several times.

3.4.2.3 Carbon Offsets

[11] Carbon trading markets, used very successfully to resolve North American acid rain, are opposed by Pope Francis. Ted Williams hit .401. Pope Francis is hitting .999. Who are we to judge?

- Hosted solutions must be easily run in or moved to locations powered by green electricity, and carbon offsets used by customers any other location, to help add geographic liquidity to the carbon trading market.

3.4.2.4 BLOCKS

The main factors blocking fulfillment of the requirement follow.

3.4.2.4.1 Local Infrastructure

For more information on how local infrastructure blocks green I.T., please see the paper Open Source Cloud Computing: The Greenest IT.

3.4.2.4.2 Proprietary Problems

For more information on how proprietary problems blocks green I.T., please see the paper referenced above.

3.4.2.4.3 Web 2.0 Diversion

For more information on how the Web 2.0 diversion blocks green I.T., please see the paper referenced above.

4. Architecture Principles

> *There's something happening here, what it is ain't exactly clear.*
> – Buffalo Springfield, For What It's Worth, 1966.

This section describes how four architecture principles help EnterpriseLibre meet the requirements with the best solution.

The principles support each other, and have already facilitated new capability. For example, much of the Enterprise Manager evolved naturally, as the rest of the solution made it apparent the new functionality was easily possible. Similarly, Venn solutions (*Section 5.4*) emerged spontaneously from the existing capability without new software development.

Architecture diagrams for the three main layers can be found in *Section 5 – Solution Design*. A cross-reference of how the architecture principles help meet the customer requirements and resolve the blocks can be found in *Section 7 – Traceability*.

4.1 Intranet Foundation

> EnterpriseLibre is built on an Intranet foundation,
> integrating the best software at each level of the stack.

Since shortly after discovery of the Web server in 1990[12], the standard solution for any team has been a dedicated enterprise Intranet. The strength is the simplicity: a secure network protecting all software and data, accessible just by the team, running standard IP.

This section describes how the Intranet foundation helps support the customer requirements. A complete cross-reference can be found in *Section 7 - Traceability*.

4.1.1 Usability Benefits

How the Intranet foundation supports the usability requirements.

4.1.1.1 Ease Of Use

How the Intranet foundation supports the ease of use requirements.

4.1.1.1.1 Virtual Desktop

The standard Intranet foundation assumes a desktop computer for every user as the default interface, providing the best practice already built-in, making it easy to then simply "lengthen the monitor cord" with NX to enable access over the Internet.

[12] Cisco, which helped build the NSFNET in the 1980's, led promotion of the idea of an Intranet for every organization in the 1990's. They made much of their own Intranet available externally, so anyone could download documents in minutes that previously would've taken days to get a smudged fax. Everyone that visited a Cisco website or FTP server wanted an Intranet for their own organization. Catalysed by release of the first Apache Web server in 1995, the Cambrian explosion of the Intranet began.

4.1.1.1.2 Single-Sign-On

The Intranet foundation provides the standard design for single-sign-on, with a central user directory, secure authentication to the desktop, then transparent and unforgeable logins to each multi-user application within the desktop. No custom designs or code required.

4.1.1.2 No Maintenance

How the Intranet foundation supports the no maintenance requirements.

4.1.1.2.1 Full-Time Availability

Since the Intranet is the standard model, there is no need for custom development, and EnterpriseLibre can build on proven design and components across the stack, providing the hardened foundation that enables delivery of full-time availability for the integrated solution.

4.1.2 Capability Benefits

How the Intranet foundation supports the capability requirements.

4.1.2.1 Full Functionality

How the Intranet foundation supports the full functionality requirements.

4.1.2.1.1 Integrated Solutions

The Intranet foundation is the standard way to provide integrated enterprise solutions, since the secure network enables standard communications between all components, there are no security concerns, and everything is fast. At the system layers, EnterpriseLibre integrates about two dozen long-time hardened FLOSS components into an unprecedentedly powerful turn-key enterprise Intranet. Adding applications to this standard foundation, and building more integrated value, is now as straight-forward as it's going to be.

4.1.2.2 One-Click Management

How the Intranet foundation supports the one-click management requirements.

4.1.2.2.1 Enterprise Automation

The Intranet foundation provides all the parts for powerful enterprise automation. All the standard software, interfaces, and protocols are available – virtual servers, name server and email servers, user directory, databases, applications, desktops, all protected inside a dedicated IP network. In other words, there is full enterprise functionality to automate. Open integration can take it from there.

Similarly, the Intranet foundation provides a complete set of pieces for security automation. For example, the Enterprise Manager controls which apps and services are accessible outside the desktops using the standard firewall component, sets HTTPS requirements using standard Apache configuration, and mandates two-password login by adding a check-point after the standard desktop login. No custom components needed, all the standard parts are available.

4.1.2.2.2 User Automation

As enterprise automation, the Intranet foundation provides all the pieces for powerful user automation. The standard components – user directory, application databases, email servers, in the secure IP network – are available to the Enterprise Manager so it can change names, passwords, notification addresses, two-password settings, and email addresses across the enterprise, with one click.

4.1.2.2.3 Scalability Automation

As enterprise automation, the Intranet foundation provides all the parts for powerful scalability automation. All the standard components – virtual servers, name servers, email servers, user directories, applications, database systems, virtual private networks – are available to the Enterprise Manager and System Manager to manage in the usual way. No custom designs or components are required.

4.1.3 Security Benefits

How the Intranet foundation supports the security requirements.

4.1.3.1 Confidentiality & Integrity

How the Intranet foundation supports the confidentiality & integrity requirements.

4.1.3.1.1 Single Tenant

The Intranet is fundamentally a single tenant architecture, with one network protecting one customer's software and data, including as many dedicated components as needed. That is, a single-tenant enterprise containing single-tenant applications. This makes it straightforward to pace the domain name as the top architectural concept, with its own name server, email system, applications, desktops, and all other software, safeguarded by a dedicated firewall. And with each customer having a modular, dedicated solution, even in the hosted model they can customize any part of their enterprise without affecting any other customer.

4.1.3.1.2 Secure Network

The Intranet basically *is* a secure network, one of its main purposes, protecting all the customer's software and data, and enabling easy and secure communications between users and applications inside the Intranet. Each EnterpriseLibre Intranet has it's own domain name server configured as per the requirements in silent mode and able to change application and service domain names instantly, and it's own firewall configured as per the requirements in default blocking mode.

4.1.3.1.3 Physical Protection

The Intranet foundation makes physical protection straight-forward, onsite or hosted, since the twelve virtual servers in the Intranet can run physically collocated on any architecture. For example, EnterpriseLibre could run on twelve different hardware servers, or on one appliance server, or on an IaaS such as AWS, KVM, Mesos, or OpenStack (see *Section 5 – Solution Design*). Wherever it is, it's easy to collocate and physically protect it.

As described in *Section 5.1.8 – Hardware Server*, the most costly resource for EnterpriseLibre is RAM, requiring ~**1 GB** per user, with CPU secondary. Therefore, one local appliance server could easily support up to **100** users, and, in the hosted solution, one production server could easily support at least **250** users. Entire servers – and racks if needed – can be dedicated to

one customer if desired. Customers can control the locks on their cabinets if they wish. Onsite or hosted, the Intranet foundation makes collocation and physical protection as easy as it is going to be.

4.1.3.2 Mobile Privacy

How the Intranet foundation supports the mobile privacy requirements.

4.1.3.2.1 Single-Sign-On

See *Section 4.1.1.1.2* for how the Intranet foundation supports single-sign-on.

4.1.3.3 Certifiable Assurance

How the Intranet foundation supports the certifiable assurance requirements.

4.1.3.3.1 Proven Design

The Intranet foundation is the longest proven design for enterprise solutions. EnterpriseLibre is able to use all the security designs, protocols, and best practices improved over decades. Any required mapping to technical, government, and organization security standards is straight-forward. It is orders of magnitude easier for security experts to understand and analyze than any custom design, so that much easier to obtain meaningful levels of certification assurance.

4.1.4 Sustainability Benefits

How the Intranet foundation supports the sustainability requirements.

4.1.4.1 Long-Term Use

How the Intranet foundation supports the long-term use requirements.

4.1.4.1.1 Natural Evolution

The Intranet foundation is the most widely used design, and so has the most mature software most likely to follow standards, play nicely with other software, and evolve most naturally with improvements in technology and customer requirements. It is particularly well suited to support innovation at the network layer, hosting, integrating, and communicating with new applications and services that use the capabilities of the Internet.

4.1.4.2 Green I.T.

How the Intranet foundation supports the Green I.T. requirements.

4.1.4.2.1 Energy Efficiency

The Intranet foundation supports energy efficiency with a design that assembles the best components across the stack, each optimized for a specific purpose, dedicated to a single customer and their process and data.

4.2 Open Integration

> EnterpriseLibre uses open integration to provide the most value with the least life-cycle cost.

Open integration resolves much of the complexity and cost of the development and life-cycle of enterprise solutions using the following principles:

- Open Source Software. Use FLOSS wherever possible, to enable review of the code to help integration and assurance, add functionality if required, and solve the *O(N^2)* licensing complexity block. Contribute all changes back to the baseline for consideration for inclusion and long-term enhancement and support.

- Standards. Use independent standards to minimize development and maintenance, and maximize interoperability.[13] Standards interfaces mean orders of magnitude less development, and make integration of powerful enterprise solutions feasible.

- Life-Cycle Cost. After filtering by capability, select all designs and components by least cost over the life-cycle. Any functionality can be updated in software. However, life-cycle cost drains resources that could be used for other value forever. When a solution has a low life-cycle cost, the capability continually gets *better* over time.

- Least Development. Create the least possible new software to provide the integrated solution value. Since user functionality is all that matters, wherever a requirement can be met with existing components, the advantages of not having to create, test, document, and maintain new software are, relatively speaking, infinite.

This section describes how open integration helps support the customer requirements. A complete cross-reference can be found in *Section 7 – Traceability*.

4.2.1 Usability Benefits

How open integration supports the usability requirements.

4.2.1.1 Ease Of Use

How open integration supports the ease of use requirements.

4.2.1.1.1 Virtual Desktop

The NX protocol is perhaps one of the best examples ever of use of open integration, simply improving the performance of an already well-defined standard, X11, for use in a lower bandwidth, higher latency environment, and so requiring no changes to any other layer. Many of the advantages of EnterpriseLibre flow from this clean integration of the NX protocol.

[13] "Open integration" could be dated to the first *X/Open Portability Guide (POSIX)* published July 1985. RPC's for app communications were developed in the 1980's, on Sun Solaris, the first popular Unix. RDBMS were blocky, however convenient standard backplanes. In the early 1990's, the Internet Protocol, in competition with OSI for mindshare, was called the "de facto" standard. And so on.

4.2.1.1.2 Application Simplicity

Most open source software applications are easier to use, undistorted by other forces. The interfaces are simple and sometimes elegant. Standard formats and protocols are the norm, including of course for data export – for example LibreOffice reads and writes a wide range of formats from RTF to Microsoft docx, and has long provided a convenient built-in PDF export. Enhancements are usually prioritized by user preference, and unnecessary user interface changes avoided. And of course there is always one version, with one optimized interface.

4.2.1.1.3 Single-Sign-On

Single-sign-in is a great example of an increasingly widely adopted open integration standard. As usual, Microsoft technology has divided technology, however in the FLOSS world more and more multi-user applications come with single-sign-on authentication capability built-in, and where not available can of course be added. The parts all fit together well.

Since this design and associated software are standard and simple, they are also used for authentication to any Web applications the Superuser enables accessible outside the virtual desktops, further increasing security and reducing code and complexity.

4.2.1.2 No Maintenance

> *No code is better than no code.* – A. S. Martprogrammer, early.

How open integration supports the no maintenance requirements.

4.2.1.2.1 Full-Time Availability

Minimizing life-cycle cost requires minimizing maintenance to the lowest level, which requires increasing availability to the highest level. All four open integration principles help. Open source software is used the longest, open to many eyes and improvements, and usually more reliable. Standards are used wherever available. After functionality, designs and software are selected by least life-cycle cost. And new code is avoided, with possibly new problems.

EnterpriseLibre builds these principles into a complete FLOSS enterprise, providing a reference for continuous reliability improvement at the Intranet level so successfully experienced at the component level. It is already very stable: some customers have run for years without any issues. Software can't be proved bug-free, however the open source software design enables any reliability issues found to be fixed and incorporated in the baseline, providing assurance the solution availability will only continue to increase over time.

4.2.1.2.2 Hosted Convenience

Virtual desktops enable the hosted version, and open integration make virtual desktops easy by simply extending the existing X11 over the Internet with NX.

4.2.2 Capability Benefits

How open integration supports the capability requirements.

4.2.2.1 Full Functionality

How open integration supports the full functionality requirements.

4.2.2.1.1 Integrated Solutions

The main purpose of open integration is to provide more than the sum of the parts. A lot of this integrated value can be seen with one use case: logging into a virtual desktop, starting the document management app with single-sign-on, double-clicking on a document to automatically open LibreOffice, saving a change to the document management system, and colleagues automatically receiving email notifications the document has been updated.

Wherever possible, information is automatically synchronized to remove effort, improve accuracy, and increase capability. Whenever a domain name or email server changes, all needed configuration is performed across the Intranet. When a user changes their email address, changes are automatically made enterprise wide. Notifications are sent automatically when a colleague updates a shared file in the document management system. If desired, direct connections between application databases, for example to exchange data between CRM and financial apps, can be easily and securely implemented.

4.2.2.1.2 Best Components

There is only one choice for each component from each proprietary vendor. And no one vendor can provide components for all needs. However, there is almost always more than one FLOSS software component in each category. And, over time, usually a consensus standard has developed with more than half the user share. EnterpriseLibre integrates more than three dozen of these standard, long-time leading FLOSS components across the stack.

4.2.2.1.3 Any Application

Open integration makes inclusion of any FLOSS application straight-forward, with standard components, interfaces, and protocols all in place, requiring minimal customization or complexity. And FLOSS provides a breadth of capability no one vendor can match, including Microsoft or Google. EnterpriseLibre currently includes the following applications, with other good FLOSS options listed in *Table B-2*.

- Individual Apps. Every desktop includes the twelve standard FLOSS apps for individual use: Browser, Desktop Publishing, Email & Calendar, File Sync, Graphics Bitmap, Graphics Vector, Messaging, Mind Mapping, Office Suite, OCR, Project Scheduling, and Visualization.

- Multi-User Apps. Includes one-click availability of the standard or leading app in ten horizontal multi-user categories: Accounting, Customer Management, Document Management, HR Management, Issue Management, Mailing Lists, Project Management, Resource Management, Timesheets, Wiki.

- Segment Apps. Includes one-click availability of five apps in four segments: Education, Manufacturing, Medical, and Non-Profit.

It's also *easiest* with open source to provide every user with every app – the least code, least maintenance, no special cases from licensing distortions. Other rights and permissions for multi-user apps are managed by the application Admin as usual, set to the Superuser to start.

4.2.2.2 One-Click Management

> *POSIX does not address... system administration. How do you add users? How do you back up the file system? How do you install a package?*
> – Donald Lewine; POSIX Programmer's Guide; O'Reilly; 1994.

How open integration supports the one-click management requirements.

4.2.2.2.1 Enterprise Automation

The Intranet foundation provides the enterprise parts, and then open integration makes the automation practical, providing the Enterprise Manager to implement any configuration of domain name, name server, email server, application domain names, backup services, and other settings the Superuser wishes to specify with a single click.

The Enterprise Manager can change an application domain name from "crm.mydomain.com" to "contacts.mydomain.com" by directly updating the name server database. When the Superuser sets up another backup service, to store their data on external storage just in case, the Enterprise Manager updates the backup system, all FLOSS, and within the hosted solution, directly. Configuration across the enterprise can be done automatically, with one-click.

Similarly, open integration makes one-click automation of enterprise security possible. When the Superuser enables external access to applications, and changes their domain names, the Enterprise Manager directly updates the firewall, name server, and Web server as needed. All protocols and formats are standard, all the code is free. No custom complexity needed.

4.2.2.2.2 User Automation

As with enterprise automation, the Intranet foundation provides the pieces, and open integration makes user automation practical. The Enterprise Manager can practically interact with all the components needed to change a name or email address across the whole enterprise. All the software is open source, and follows open standards. If software can do it, it can be done.

4.2.2.2.3 Scalability Automation

As for enterprise automation, the Intranet foundation provides the pieces, and open integration makes scalability automation practical. The Enterprise Manager, given just a new user's email address, can practically reach out, using open source software and standards, to automatically create a new virtual desktop, email and calendars, accounts with all team applications, and send a notification email, in less than five minutes. Communication is possible; data can be exchanged. Similarly, open integration makes practical the ability to suspend and resume entire user accounts, virtual desktops to email, with one-click.

Open integration also makes practical one-click automation of the life-cycle of multi-user applications. Using open standards and software, the Enterprise Manager can transparently handle installation, configuration of system information like domain names and email servers, and creation of accounts for current users. If source needs to be read or a tweak required to enable the one-click end result, it can be done.

Open integration also enables automated management of multiple enterprises (*Section 5.3 – System Manager*). From a starting point of about nine hours, it takes about **45** minutes to automatically build a complete enterprise Intranet, securely, error-free. (This enterprise creation capability is working user self-serve from the Cirrus Computing website, connected to Paypal, however not public at time of writing.) Similarly, whole enterprises – users, virtual desktops, applications – can be suspended and resumed instantly with a click. Enterprises can be moved between servers, around the world if desired, in less than an hour. Without artificial limitations, entire enterprises can be managed as though weightless, which they are.

4.2.3 Security Benefits

> *Hiding source code does inhibit the ability of third parties to respond to vulnerabilities ... but this is obviously not a security advantage.*
> – Chief Information Officer, US Department of Defense, 2015-08-29.

How open integration supports the security requirements.

4.2.3.1 Confidentiality & Integrity

How open integration supports the confidentiality and integrity requirements.

4.2.3.1.1 Single-Tenant

Open source software eliminates all the artificial technical and licensing blocks driving the regression to the monolithic multi-tenant architecture. Each EnterpriseLibre enterprise provides each customer with their own servers, networks, desktops, applications, databases, and all other software. In other words, each software component is dedicated to a "single tenant", and there is no cost penalty for doing so, and a great efficiency advantage.

The open integration supports single-tenancy in a second way as well, by using nearly zero overhead container virtualization (*Section 5.1.6*). This enables EnterpriseLibre to provide twelve dedicated virtual servers for each customer enterprise, providing compartmented security with almost no overhead. When hosted, customers can run their entire enterprises securely on one optimized server today up to at least 1,000 users, with almost no virtualization cost, making multi-tenancy unnecessary at every level of the stack.

4.2.3.1.2 Secure Network

Open integration enables provision to each customer of an encrypted network with a dedicated name server and protected by a dedicated firewall. Using open standards and software, the Enterprise Manager can change the name of any of service available outside the desktops, such as "wiki.mydomain.com" to "secret.mydomain.com", with instant effect.[14]

4.2.3.1.3 Secure Channels

Open integration helps provide direct access to an application, perhaps for outside expert consulting assistance, by connecting existing pieces into a very secure channel. First, a virtual desktop is created, then read and write access to the application folders and code is

[14] Each domain name label complies with RFC 1035, with a maximum of 63 characters, letters, digits, or the hyphen, must start with a letter, and must end with a letter or digit. Therefore, there are 26 x 37^61 x 36 = 1.83x10^197 options. Since there are about 10^80 atoms in the universe, there are 10^117 more choices for a label than atoms in the universe.

provided within the desktop. Any tools such as PHPMyAdmin can be run safely. Root access to the virtual server hosting the app can be provided. Any files needed can be emailed to their temporary team address, or they can connect Evolution to an existing address. Any very large files like database archives can be uploaded with their built-in Syncthing app.

With this open integration design, the risk is not technical, but rather that the consultant themselves is malicious: purposely tries to bypass the security to gain access to other parts of the system, and then somehow manages to do so. The risk is no longer that leaving open a special security hole could get breached by any number of unquestionably ill-intentioned human and automated agents across the world. The secure channel is assembled from existing pieces of the secure solution, provided to just one person, and hermetic security of the solution remains uncompromised.

4.2.3.2 Mobile Privacy

> *Some of the world's most popular online services have been enlisted as partners in the NSA's mass surveillance programs. Technology companies are being pressured by governments around the world.*
> – Edward Snowden, NY Times, 2015-06-05.

How open integration supports the mobile privacy requirements.

4.2.3.2.1 End-To-End Encryption

Open integration enables straight-forward use of the proven OpenSSH software for the virtual desktop connections, providing end-to-end protection for authentication and all subsequent communications, supporting mobile privacy with a strongly hardened component.

4.2.3.2.2 Single-Sign-On

Open source software is strong on standards, so multi-user applications usually provide an interface for single-sign-on. Each EnterpriseLibre solution has it's own dedicated user directory using OpenLDAP, a long hardened FLOSS cornerstone.

User passwords have no hard-coded restrictions, such as requirement for capital letters, since these policies are customer choices and can be easily added as enhancements to the Enterprise Manager. However, as foundation, they can have up to **14** of the usual alphanumeric characters, for a total of **42^14** different options. Any high-frequency attacks are blocked by delay mechanisms, so only human mistakes are allowed.

4.2.3.2.3 Two-Password Login

Two-password login is provided by integrating the open source software component PAM_OBC into the standard desktop authentication process, providing a single check-point after the usual desktop login for entry of the random PIN. The Superuser can require two-password login for the entire team or specific users, and if left open to user's discretion then they can turn it on or off themselves from the Me tab of their Enterprise Manager.

4.2.3.3 Certifiable Assurance

> *I certainly wouldn't characterize [Edward Snowden] as a hero... You won't find much admiration from me... specific techniques [governments] use become unavailable if they're discussed in detail.* – Bill Gates, 2014.

How open integration supports the certifiable assurance requirements.

4.2.3.3.1 Proven Design

> *Continuous and broad peer-review, enabled by publicly available source code, improves software reliability and security through the identification and elimination of defects that might otherwise go unrecognized.*
> – Chief Information Officer, US Department of Defense, 2015-08-29.

EnterpriseLibre integrates proven design at all levels of the stack, including single-sign-on, default firewall blocking, public-key cryptography, two-password login, single desktop, separation of GUI, processing, and data layers, and other elements.

The EnterpriseLibre integration software also follows proven design practices. For example, the Enterprise Manager and System Manager separate user interface, processing, and data. Which is one reason they use a Gnome GUI instead of an HTML page, to securely surface only the information needed.

Of course, EnterpriseLibre is FLOSS and so can be modified however desired. However, users of the baseline solution can be assured that even the Superuser cannot break the extraordinary security of a running enterprise :-).

4.2.3.3.2 Hardened Components

Open Integration reuses hardened components as the only answer to scale. Since EnterpriseLibre is based on the standard Intranet, there is good availability of FLOSS components across the stack hardened over many years. For example, WikiMedia has been steadily improved since 2000, Apache since 1995, Firefox as Mozilla since 1993, Linux since 1991, and LibreOffice as StarWriter since 1989.

4.2.3.3.3 Open Source Software

> *OSS better meets Saltzer & Schroeder's Open design principle ('the protection mechanism must not depend on attacker ignorance'). This is not merely theoretical.* – Chief Information Officer, US Department of Defense, 2015-08-29.

Open source is more secure for several reasons.[15] The foundation operating system Linux provides a great deal of the advantage, since it has the Unix architecture, designed by AT&T Bell Lab scientists with security embedded in its most basic design. Problems with widely used FLOSS are found and fixed, since "sunshine kills bacteria". The source can be reviewed

[15] *In the 1980s I was in the IEEE committee... standardizing interface specs for a Unix-like system... someone gave it the name "IEEEIX"... It seemed to me... like a shriek of terror... everyone would call it "Unix". That would have boosted AT&T, the GNU Project's rival... so... came up with "POSIX"... barely in time, the committee adopted it.* – Richard Stallman, The origin of the name POSIX, 2011.

by anyone. And the customer can rebuild their enterprise from source at any time, providing assurance the running version is free of any inserted vulnerabilities. It is not only secure, but also can be reviewed, analyzed, and practically certified as secure. Those with responsibility for safeguarding the information it processes can use it with meaningful assurance.

At the virtualization level, use of the open source Linux container technology provides two kinds of assurance. First, it's in the operating system, instead of the chip or proprietary code, and so can be reviewed. And second, it is managed in the most secure software in the stack. If the Linux kernel has a security bug, we have a bigger problem, otherwise the customer can be confident virtualization is as secure as it can be, other than dedication of entire servers, which is also possible and facilitated by the zero-license costs of the FLOSS design.

4.2.4 Sustainability Benefits

> *Since PDP-11 Unix became operational in February, 1971, over 600 installations have been put into service.* – Dennis Ritchie, Ken Thompson; The UNIX Time-Sharing System; Jul-Aug 1978.

How open integration supports the sustainability requirements.

4.2.4.1 Long-Term Use

> *Writing these programs is not so easy if you rely solely upon the manufacturer's documentation... A function might... add several new features peculiar to that computer system or operating system variant.*
> – Donald Lewine, POSIX Programmer's Guide, O'Reilly; 1994.

How open integration supports the long-term use requirements.

4.2.4.1.1 Open Source Software

> *Because open source software features open code, more programmers are able to view the code, create new functionality, and fix bugs... the same natural way that science has developed over time.*
> – Chen Yang; Taoism of Open Source; 2007.

FLOSS provides the ability to own and update the source code, an ability – even if never exercised – mandatory for meaningful long-term software assurance. And it will always be accessible at an acceptable cost over time, for new users and if the team grows.

Since EnterpriseLibre is all FLOSS, users can be confident it exists for them, now and in the future. All EnterpriseLibre software built by Cirrus Computing is FLOSS licensed under the GPL, including the System Manager and Enterprise Manager. The customer can be assured all the software will continue to be available, including all enhancements, over time.

4.2.4.1.2 Onsite Or Hosted

> *Think of free speech, not free beer.* – Richard Stallman; 2015-04-22.

Open integration of a standard Intranet makes it easy to move an EnterpriseLibre enterprise between onsite hosting and any kind of hardware or IaaS hosted service.

Sometimes onsite is the best option, in a company data-center, local building, or home office. With its open source virtual server foundation, EnterpriseLibre is easily installable on a local appliance server. With continuing hardware improvement, teams up to a few dozen users will soon be able to run EnterpriseLibre on a single CPU.

On the other hand, hosting is often the best option for convenience, cost, and security. With open integration across the stack, EnterpriseLibre can run on almost any hosted hardware server or online IaaS that can run a Linux image – almost all of them. an EnterpriseLibre Intranet including virtual desktops has been tested and runs great on Amazon Web Service, KVM hosting services, and OpenVZ services, and would work on others such as Mesos, OpenStack, etc. (see *Section 5 – Solution Design*).

However, whether onsite or hosted, genuine long-term use also requires the ability to change. Local hardware can break and replacement be too expensive, or the team can become more distributed. Data-centres may come and go, or the customer may just wish to switch. The ability to move an Intranet enterprise between onsite and hosting without lock-in to any architecture or vendor, even if rarely exercised, provides the genuine long-term assurance users need. As long as the customer has a data backup, they can always recreate their open source solution, locally or hosted, and keep their team going.

4.2.4.1.3 Natural Evolution

> *Programmers must... isolate nonportable code from portable-code, such that even hardware-dependent features are easily identified.*
> – Donald Lewine; POSIX Programmer's Guide; O'Reilly; 1994.

Open integration of best-in-category FLOSS components, using standards everywhere, and minimal development, gives EnterpriseLibre the simplest possible architecture, supporting evolution and enhancement with the least complexity. New software can be added and components changed or updated with minimal impact to the rest of the solution.

4.2.4.2 Green I.T.

> *With Solaris 10 zones there is no overhead whatsoever and a potential to run literally hundreds of virtual servers per CPU.* – Jack Meoff, 2005.

How open integration supports the green I.T. requirements.

4.2.4.2.1 Energy Efficiency

Open integration supports efficiency in general with its focus on least life-cycle cost, and in particular by use of FLOSS, which is usually highly tuned to be as efficient as possible, particularly at the system layers. Leading examples of high-performing FLOSS include the Apache web server, Linux kernel, and PostgreSQL database.

4.3 Virtual Desktops

> EnterpriseLibre uses the standard computer desktop user interface, delivered securely over the Internet to any device.

The desktop has been the standard computer GUI for more than three decades. Many ways have been tried to reduce the inconvenience of the clients, including thin-client terminals, docking stations, laptops, netbooks, and the various sizes of tablets and smartphones.[16]

This section describes how virtual desktops help support the customer requirements. A complete cross-reference can be found in *Section 7 - Traceability*.

4.3.1 Usability Benefits

How virtual desktops support the usability requirements.

4.3.1.1 Ease Of Use

How virtual desktops support the ease of use requirements.

4.3.1.1.1 Virtual Desktop

> *There are not many reasons to use NX just to access a Solaris box on your LAN. We aim to provide the same functionalities over the Internet and in a secure and scalable way, something plain X + XDMCP cannot offer.*
> – Gian Pinzari; LTSP post; 2003-03-31.

The computer desktop is the best GUI we have yet developed. All the common variants – Gnome, KDE, MacOS, Unity, Windows – provide the same usability, with differences on, relatively speaking, details. As long as there is a background, folders, files, applications, copy-and-paste and drag-and-drop, running on one OS and chip, the essentials are present.

The NX protocol simply "extends the monitor cord" by communicating any X11 based desktop over the Internet in near real-time around the globe. The main reason for the blazing performance is that X11 is text-based, so can be optimized more efficiently than any graphics based method, such as the higher bandwidth VNC. The second reason is it takes light only **0.067** seconds (at **299,792** km/s) to go or half-way around the earth (**20,038** km), which is the minimum maximum time for an Internet packet to travel from any hosted server to any access device on the planet, a speed it's fast approaching. That's a bit less than a **14'th** of a second, about three times fast than the requirement for acceptable virtual desktop usability on networks with less than **200** ms latency. Light is fast, or the planet is small.

Virtual desktops have another counter-intuitive advantage, often running applications faster than if they were locally installed. This is because users increasingly access the Internet from devices that are smaller, lighter, and less powerful than an office computer, so running software on a more powerful server and then sending the GUI over the Internet takes less

[16] When holographic projection is easy, client hardware will likely become a detail, and GUI innovation will continue with the "deskspace". EnterpriseLibre will be ready :-).

time, in total, than running the software locally.

In practice, EnterpriseLibre virtual desktops using x2go use only about **125 Kbps** (0.125 Mbps) down and **~25 Kbps** (0.025 kbps) up, making bandwidth a non-issue around the world. Virtual desktops hosted in Ottawa and Montreal data-centers have been tested and found to perform with near real-time performance on fixed and wireless networks across Canada and the USA, and from Frankfurt, Dubai, and Delhi. The farthest user, in Melbourne Australia, reported there was perceptible delay however "acceptable" performance.

The clients run fine on old computers, and have saved Cirrus Computing's largest customer significant capital expense. They have also been tried and work fine on an Odroid C1, and would work well on a Raspberry PI or any similar device.

4.3.1.1.2 Application Simplicity

Virtual desktops provide access to real, full-feature applications, powered by a real operating system and full UI, with much better usability than Web apps. For example, every EnterpriseLibre desktop includes the highly interactive apps FreeMind, Gimp, Inkscape, LibreOffice, ProjectLibre, Scribus, and VUE. Even the Evolution email, contacts, calendar application is recognizable by non-technical users as designed for them, with big buttons and lots of space.

4.3.1.1.3 Single-Sign-On

Single-sign-on is designed for the desktop, so straightforward to implement wide area with virtual desktops, since the only part that makes them virtual is NX, at a different layer in the architecture. This enables clean implementation for each application without affecting the security of the standard single-sign-on design and implementation.

4.3.1.2 No Maintenance

How virtual desktops support the no maintenance requirements.

4.3.1.2.1 Full-Time Availability

The virtual desktops enable the hosted version, which enables the deployment scale to ensure that any problem encountered with one customer is never encountered by any other customer. This provides steadily increasing reliability at the FLOSS Intranet enterprise level that individual components have experienced. Even with the relatively limited experience with Cirrus Computing customers over a few years, EnterpriseLibre has already become better, now effectively 100% reliable, a rare experience with integrated enterprise solutions.

4.3.1.2.2 Hosted Convenience

Virtual desktops enable the hosted version to have all the same usability and capability as the onsite version.

4.3.2 Capability Benefits

How virtual desktops support the capability requirements.

4.3.2.1 Full Functionality

How virtual desktops support the full functionality requirements.

4.3.2.1.1 Integrated Solutions

Virtual desktops are the most capable GUI yet developed for interaction with a computer. With the power of a full operating system underneath, running on the same chip, it will always be more capable than any Web browser technology.

And it provides the most natural portal for access to all the software and data in an enterprise solution. From the Shared Files folder to single-sign-on to multi-user applications, the virtual desktop enables the whole integrated solution to be accessed from one place.

4.3.2.1.2 Any Application

Virtual desktops enables easy access to almost any FLOSS application that runs on Linux. EnterpriseLibre currently provides built-in access to twelve individual apps with each desktop, ten common multi-user horizontal use apps, and five segment apps. With the EnterpriseLibre platform in place, addition of almost any application is straightforward from here.

(For future investigation, since NX supports Windows RDP, any application that runs on a Windows server could be transparently included in an EnterpriseLibre virtual desktop as a custom add-on. Wine also continues to improve, and could also be used to embrace and replace Windows based software.)

4.3.3 Security Benefits

How virtual desktops support the security requirements.

4.3.3.1 Confidentiality & Integrity

How virtual desktops support the confidentiality & integrity requirements.

4.3.3.1.1 Secure Channels

Direct access to applications for assistance by experts, e.g. with CRM or ERP, is much more secure when protected by virtual desktops, especially when active just for the purpose and period needed. An consultant first logs in to their secure desktop with the usual protections, including two-password authentication if mandated by the Superuser, and then all further work is fully encrypted, including access to the application code, data, and even dedicated server if needed. From within the secure desktop, use of any tools and utilities like PHPMyAdmin is fully protected. And there is no reason to leave data and files lying around on a local computer. Providing the channel from within the virtual desktop enables the hermetic security of the enterprise to be maintained.

4.3.3.1.2 Physical Protection

Virtual desktops enable the hosted version, and, for most small and medium size organizations, hosted protected software in a controlled access room in a managed data-center in a trusted jurisdiction is more secure than onsite in their office or home.

4.3.3.2 Mobile Privacy

How virtual desktops support the mobile privacy requirements.

4.3.3.2.1 End-To-End Encryption

Virtual desktop communications are encrypted end-by-end by the long-time standard OpenSSH software. X2Go software integrated into EnterpriseLibre uses a key of **1024 bits**. Each solution uses an OpenSSL RSA **2048 bit** SHA1 certificate for IMAP, SMTP, and related services.

4.3.3.2.2 Single-Sign-On

See *Section 4.3.1.1.3* above for a description of how virtual desktops help support single-sign-on.

4.3.3.2.3 Two-Password Login

Virtual desktops make it straight-forward to provide two-password login by providing a standard point for entry of a second PIN after the regular desktop login.

4.3.3.3 Certifiable Assurance

How virtual desktops support the certifiable assurance requirements.

4.3.3.3.1 Proven Design

By going using a long proven interface, not getting lost in the great Web 2.0 diversion, virtual desktops avoid all the new design, code, and complexity that makes certifiable assurance so difficult for all custom software.

4.3.4 Sustainability Benefits

How virtual desktops support the sustainability requirements.

4.3.4.1 Long Term Assurance

How virtual desktops support the long-term use requirements.

4.3.4.1.1 Onsite Or Hosted

The virtual desktops enable EnterpriseLibre to provide all the usability, capability, and security of an onsite enterprise, onsite or online. The solution could be hosted either on a server in the basement of a small business owner, the headquarters of a non-profit, a large enterprise existing data-center, or hosted service like AWS (tested and works fine). Wherever hosted, the virtual desktops provide the same capability to the users. The customer has the assurance they will always be able to find a hosting solution over the long-term.

4.3.4.2 Green I.T.

How virtual desktops support the green I.T. requirements.

4.3.4.2.1 Energy Efficiency

The virtual desktops enable hosting, which provides great energy efficiency by load-balancing a smaller number of optimized low-power servers. See the paper *Open Source Cloud*

Computing: The Greenest IT for more information.

4.3.4.2.2 Computer Recycling

Display of virtual desktops take very little CPU, so computers at least ten years old and often older work fine as clients. For security, old Windows computers can be secured to run just the virtual desktop, requiring even less power. Even the fan runs on the lowest setting!

Old computers without a current Windows license can be easily wiped clean, installed with a favorite brand of Linux, and used with security assurance and little local power. The significant environmental impact and expense of new computers can be reduced several fold, year after year.

4.3.4.2.3 Carbon Offsets

Virtual desktops enable anyone, anywhere to use a carbon efficient enterprise solution, powered by renewable electricity. For example, Cirrus Computing uses a data-center in Montreal powered by almost all green hydro-electric power[17]. When a hosted solution is used by a customer in a high-carbon location, virtual desktops make the cost available for carbon offsets, adding geographic liquidity to the market.

[17] Hydro-electric projects provide green power once built, however the environmental and human impacts of construction must be taken into consideration in every case.

4.4 Optimum Efficiency

EnterpriseLibre is as efficient as possible, using the least resources.

The more efficient software is, the more can be provided. And optimum efficiency, especially for an integrated solution, naturally provides simplicity, reliability, and other benefits.

This section describes how optimum efficiency helps support the customer requirements. A complete cross-reference can be found in *Section 7 – Traceability*.

4.4.1 Usability Benefits

How optimum efficiency supports the usability requirements.

4.4.1.1 No Maintenance

How optimum efficiency supports the no maintenance requirements.

4.4.1.1.1 Full-Time Availability

Outages have not only direct costs, but also compounding costs from the uncertainty, every moment someone worries about an outage happening, and how long it will be. With full-time availability, this cost is zero. Full time availability on an annual basis is worth pursuing – a kind of ultimate customer confidence tipping point.

This release of EnterpriseLibre includes significant enhancements and may not be 100% shaken out. However, the foundation is well proven. Three customers have used hosted solutions for several years. The largest has used at least **17** virtual desktops for more than three years, with only two system problems: shortage of email server space, and email server RAM, both fixed. They occasionally send support emails, usually requesting a reminder on how to create a desktop start link on an old computer, since an even older one broke.

4.4.1.1.2 Hosted Convenience

Other things being equal, hosted solutions are much more efficient than onsite, since they run on optimized production servers, and can use load-balancing to reduce total requirements across customers, users, and even time-zones. Therefore, even if not needed by other factors, optimum efficiency requires EnterpriseLibre to be available in a hosted option, achieved with the Intranet foundation, virtual desktops, and clean architectural separation from lower-levels at the virtual server so it can run on anything from metal to an IaaS such as AWS, KVM, Mesos, or OpenStack.

4.4.2 Capability Benefits

How optimum efficiency supports the capability requirements.

4.4.2.1 Full Functionality

How optimum efficiency supports the full functionality requirements.

4.4.2.1.1 Integrated Solutions

The more efficient a solution is, the lower the cost, the more capability can be provided for the same price. The technical efficiency of EnterpriseLibre is one of the key factors enabling an unprecedentedly high level of capability to be provided at such an accessible price.

4.4.3 Security Benefits

How optimum efficiency supports the security requirements.

4.4.3.1 Confidentiality & Integrity

How optimum efficiency supports the confidentiality & integrity requirements.

4.4.3.1.1 Single Tenant

Optimum efficiency requires a single-tenant design, where any logic separating multiple customers in a solution is handled by the fastest layer, the operating system kernel, always orders of magnitude faster than any higher software layer (*Section 5.1.6 - Virtualization*). At the same time, fortunately the kernel is also the most secure layer for virtualization, even more than hardware, since any flaws can be more easily reviewed, discovered, and repaired.

4.4.3.2 Certifiable Assurance

How optimum efficiency supports the certifiable assurance requirements.

4.4.3.2.1 Proven Design

Optimum efficiency requires use of proven design that has had the time for use and improvement, which also means time for security flaws to be found and fixed. EnterpriseLibre leverages the best, proven design at every layer of the stack, from the dedicated Intranet to the authentication systems, security designs that have been greatly strengthened over time.

4.4.3.2.2 Hardened Components

Optimum efficiency requires use hardened components, since only real-life use over a wide range of use cases can provide the experience required for high-performance tuning. EnterpriseLibre includes hardened components at every level, with efficiency likely close to single-digit percentages of optimal in some cases, e.g. the Linux kernel.

4.4.3.2.3 Open Source Software

Optimum efficiency requires an all FLOSS design, since the best technical performance requires software undistorted by proprietary compromises, and with the wider use and development input only possible when anyone can review the code. The performance of the Linux kernel and OpenVZ container virtualization are the foundation for most of EnterpriseLibre's efficiency and low cost in the hosted model. However, much of the rest of the software has also been widely used and optimized over many years, seen for example in the scalable performance of the Apache Web server and PostgreSQL database.

4.4.4 Sustainability Benefits

> *An open system architecture can reduce cost risk by increasing interoperability, portability, and information sharing.*
> – Joint Logistics Commanders, US Department of Defense, 1996.

How optimum efficiency supports the sustainability requirements requirements.

4.4.4.1 Long-Term Use

How optimum efficiency supports the long-term use requirements.

4.4.4.1.1 Open Source Software

See *Section 4.4.3.2.3* above for a description of how pursuit of optimum efficiency supports use of FLOSS.

4.4.4.2 Green I.T.

How optimum efficiency supports the green I.T. requirements.

4.4.4.2.1 Energy Efficiency

> Q: What's the difference between a scientist and an engineer?
> A: The engineer cares about cost.

Efficiency in large part means energy efficiency, the main environmental impact of software, and the main cost to most hosters – more than space, hardware, or bandwidth. Therefore, EnterpriseLibre has been designed for optimum efficiency at all points, as described in this section.

Perhaps most significantly for energy efficiency, the integration of container technology, as described in *Section 5.1.6 – Virtualization*, was made first for efficiency reasons, since it had so much less overhead than other approaches. This efficiency, translating directly into energy efficiency, is what makes the critical difference in enabling offer of the hosted version so cost-effectively.

5. Solution Design

> *There's more to the picture, than meets the eye.* – Neil Young, Hey Hey, My My, 1979.

This section describes the three levels of the system design: the enterprise itself, the Hosted Manager providing easy configuration, and the System Manager providing easy management of multiple hosted enterprises.

The fourth section describes Venn solutions, an unexpected recently emergent capability to create multiple overlapping solutions at one domain name, providing interesting benefits for larger enterprises.

All software and documentation is available from the CirrusOpen.org Wiki.

5.1 Enterprise

This section describes the EnterpriseLibre design, a complete solution integrated from the best FLOSS components at all levels of the Intranet stack.

5.1.1 Architecture Diagram

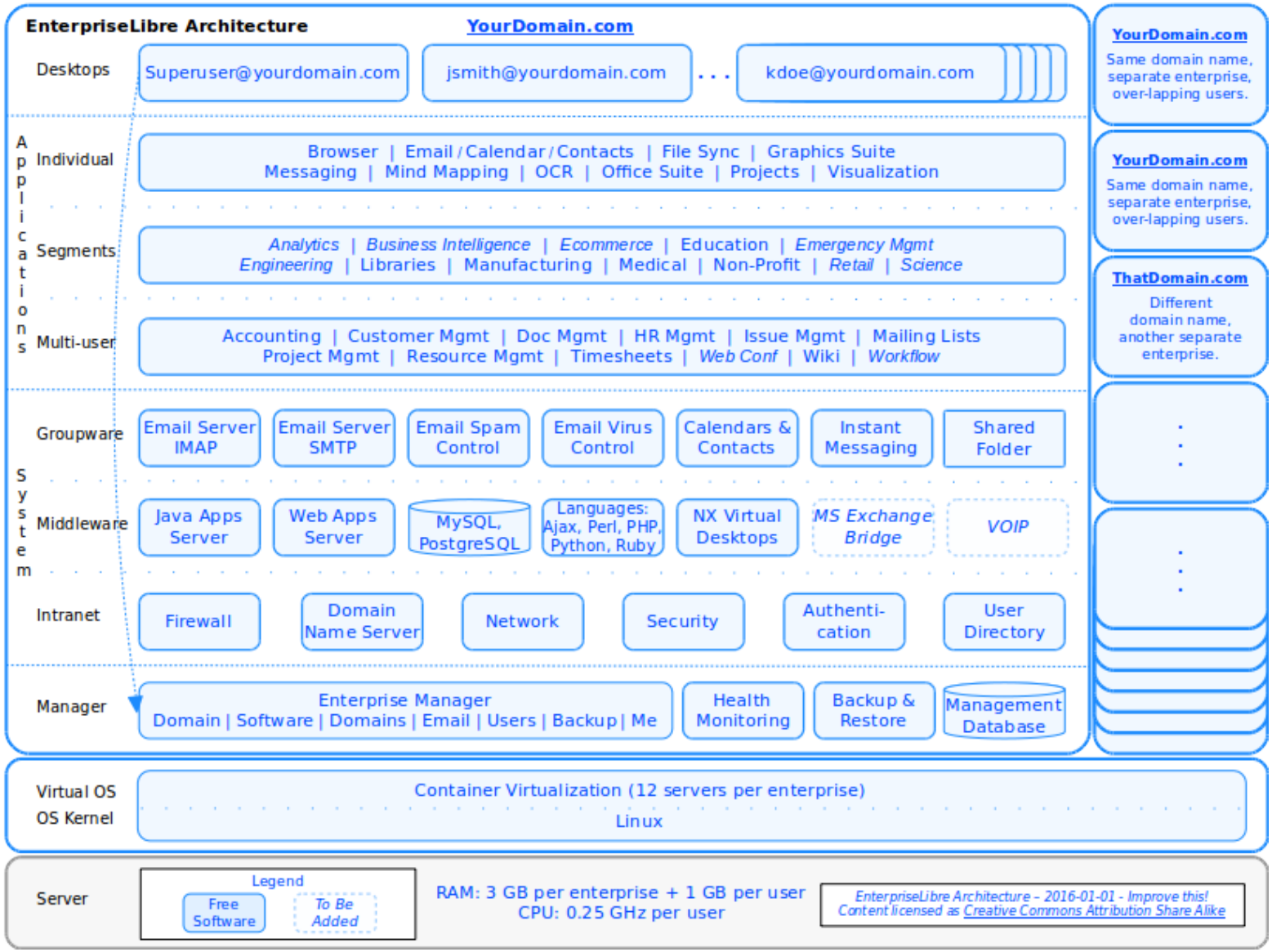
The EnterpriseLibre enterprise integrates the best open source software into a single solution that can evolve however users desire. In the same way a Linux distribution integrates operating system components around a kernel to create something more than the parts, EnterpriseLibre integrates the components of the Intranet stack into a complete hosted enterprise, dedicated to a single customer, organized around their domain name.

Each Intranet is built with twelve Linux virtual servers running on their own secure network, integrating more than two dozen hardened FLOSS components into an Intranet platform. On top, twelve standard FLOSS individual user applications are included with every desktop, and ten leading multi-user apps and five segment apps (so far) can be added with one click. The Intranet has been tuned by long use to work together as a one reliable enterprise solution.

Perhaps the architecture feature that stands out most is the single-tenant modularity – all dedicated software for each customer. This replication could be viewed from a hoster's point of view as inefficient, but it's the customer that counts. From the customer's point of view, nothing else makes sense. The great increase in requirements satisfaction with a dedicated, single-tenant enterprise solution is worth a great deal, and, as described in *Section 5.1.6 – Virtualization*, implemented with near zero overhead cost.

While separate code, each Intranet *does* benefit from improvements in others. For example, the current release has much increased stability compared to the first version due to better virtualization management which completely solves resource shortage crashes, learned from experience with one customer. Further improvements discovered with one solution can be incorporated in the baseline for all future solutions, and provided as an optional update to all current customers. The enterprise Intranet, as a single solution, will get better over time.

The following architecture diagram shows the layers and components of the enterprise, along with the operating system and virtualization layers on which it runs. The virtualization solution enables multiple solutions to run securely on the same hardware.



Graphic 5-1: EnterpriseLibre Architecture

5.1.2 Desktops

The desktops are currently Ubuntu Linux, built on Debian, using the Gnome user interface. Ubuntu has been the leading Linux desktop since 2006, shortly after Mark Shuttleworth decided to prove a Linux desktop could be built that was easy enough to be usable by non-technical people. As such a visible component, and with a critical mass of users, it has been a major driver for the growth and awareness of FLOSS around the world. As the enterprise has embraced Ubuntu, they have also worked on tablet and smartphone versions for a complete offering, helping further increase the open source software profile at the same time.

EnterpriseLibre should also support a "desktop of choice". This could be most easily be done on creation, with changes and updates as a second step. Even user-level choice is possible where application compatibility is achievable, i.e. the EnterpriseLibre architecture can manage the life-cycle of both users and applications on different user desktops without significant added code complexity, which might be more achievable than it first appears. The next evolution of the EnterpriseLibre architecture will provide more generalization of interfaces to enable easier addition of the widest number of servers, desktops, and applications.

5.1.3 Applications

The three categories of applications - Individual, Multi-user, and Segment - are described in the following sections. In most cases EnterpriseLibre includes the best and acknowledged leading software in the category, filtered from hundreds of other options that can be added later, many documented at Freeopensourcesoftware.org/index.php?title=Applications.

5.1.3.1 Individual

EnterpriseLibre includes twelve standard individual use applications with each virtual desktop, an unprecedentedly powerful toolbox not available from any one proprietary vendor:

- Browser. Includes Firefox as default, tracing roots to the original Netscape code in 1994, confidential browsing within the virtual desktop no matter what Internet connection is used. Chrome to be added, held up by a system incompatibility.
- Email / Calendar / Contacts. The venerable Evolution, originating with Ximian from 2000, once the "most widely used FLOSS in the world", better than Outlook and easier to use, still improving. Thunderbird, preferred by some technical users, to be added.
- File Sync. The standard FLOSS folder and file sync application Sync Thing.
- Graphics Bitmap. The long-time standard open source app for bitmap graphics Gimp.
- Graphics Publishing. The long-time FLOSS standard desktop publishing app Scribus.
- Graphics Vector. The long-time standard open source vector graphics app InkScape.
- Office Suite. Includes long time LibreOffice tracing back to StarWriter in 1989, now with five powerful applications, and good Microsoft Office interoperability. (Could add OpenOffice if the one-way license flow issue is addressed).
- Messaging. The standard Pidgin app, secure messaging completely internal to the organization from the desktops.

- Mind Mapping. The standard FreeMind app. The extended app FreePlane is on the near-term Enhancement List.
- Optical Character Recognition. The best FLOSS OCR app Tesseract.
- Project Scheduling. The standard FLOSS Gantt scheduling software ProjectLibre, based on long established OpenProj, with Microsoft Project format
- Visualization. The unprecedentedly powerful VUE software provides a wide range of graphics creation and sophisticated information management capabilities.

5.1.3.2 Multi-User

Multi-user applications are used by a team to manage some kind of central information. EnterpriseLibre includes standard FLOSS apps in ten multi-user categories, capability unmatched by any single company, including Google and Microsoft. When the Superuser directs installation, the app is automatically integrated with the customer's enterprise Intranet, providing single-sign-on for all users with first login, and kept automatically updated with changes to users and system information.

In most cases EnterpriseLibre includes the leading software in the category. Two additions are required to complete the multi-user layer – Web Conferencing and Workflow – with good FLOSS options identified in *Appendix B - Software Listing*.

5.1.3.2.1 Collaboration

The current release includes the following collaboration applications:

- Customer Relationship Management. vTiger, originally a simpler fork of the most established FLOSS application SugarCRM, under consideration for addition when a sustainable open source software community of users and developers is identified.
- Document Management. Nuxeo, originally a simpler fork of the most established FLOSS application Alfresco, inclusion on the near-term Enhancement List.
- Issue Management. The simplest of the ticket management systems Trac.
- Mailing Lists. The standard GNU Mailman software.
- Wiki. The software that runs Wikipedia, MediaWiki. Many other kinds of Wiki software such as TWiki can be easily added as options (see Software Listing in Appendix).

5.1.3.2.2 Enterprise

The current release includes the following enterprise applications:

- HR Management. One of the first FLOSS applications to organize personnel information, skills, searches, and related items, OrangeHRM.
- Resource Management. The oldest and most established application for management of any kind of resource Booked, built on the previous phpScheduleIt.

5.1.3.2.3 Financial

The current release includes the following financial applications:

- Accounting. Longest established and widely used FLOSS accounting app SQL-Ledger.
- Timesheets. Longest established and widely used FLOSS app TimesheetNG, built on much beloved model of simplicity Timesheet.php.

5.1.3.2.4 Project Management

The current release includes the following project management applications:

- Project Management. The most popular PM collaboration app, Redmine.

5.1.3.3 Segments

Different segments use specialized applications tailored for their specific domains. EnterpriseLibre currently includes standard FLOSS apps in four segment categories. This is the least complete layer; however, with the rest of the solution complete, it is easy now to add other segment applications wherever good options exist.

5.1.3.3.1 Education

The current release includes the long-time standard FLOSS education application:

- Learning Management. The first and most established application, Moodle.

5.1.3.3.2 Manufacturing

The current release includes a leading FLOSS manufacturing applications:

- Enterprise Resource Planning. One of the most established ERP applications Odoo, based on the previous OpenERP. Several other excellent open source software ERP applications can also be added. With USB pass-through now available with the virtual desktop, integration with local devices is also possible.

5.1.3.3.3 Medical

The current release includes a leading FLOSS medical applications:

- Electronic Medical Records. One of the best applications for single practitioners or multi-physician groups, OSCAR, including an integrated drug interaction database.

There are several other good FLOSS applications options to be added to the medical category, which benefits greatly from the security and hot backup capability of EnterpriseLibre.

5.1.3.3.4 Non-Profit

The current release includes two of the leading FLOSS applications for non-profits:

- Association Management. The first and most established FLOSS suite, CiviCRM.
- Church Management. The first FLOSS suite, ChurchInfo, built on ChurchDB.

5.1.3.3.5 Other

Other segments with good FLOSS applications that can be easily added include Business Intelligence, Data Analytics, Ecommerce, Emergency Management, Engineering, Libraries, and Science, with software options identified in *Table B-2: Software Listing (Options)*.

5.1.4 System Software

> *Software that has widespread use often has lower risk, since there are often good reasons for its widespread use. Examples of OSS that are in widespread use include:*
>
> - *Apache - Web server*
> - *Mozilla Firefox - Web browser*
> - *Mozilla Thunderbird, Evolution - Email client*
> - *OpenOffice.org - Office document suite*
> - *OpenSSH - Secure Shell*
> - *OpenSSL - SSL/cryptographic library implementation*
> - *bind - DNS server*
> - *Postfix, Sendmail - Mail servers*
> - *gcc - Compiler suite*
> - *GNAT - Ada compiler suite (technically this is part of gcc)*
> - *perl, Python, PHP - Scripting languages*
> - *Samba - Windows - Unix/Linux interoperability*
> - *Mailman - mailing list manager*
> - *MySQL and PostgreSQL - Relational Database System*
> - *GIMP - Bitmap graphics editor*
> - *MediaWiki – Wiki*
>
> – Chief Information Officer, US Department of Defense, 2015-08-29.

This section describes the EnterpriseLibre system software, the components that sit between the operating system and everything else and keep the enterprise running smoothly.

5.1.4.1 Groupware

The current release integrates the following groupware:

- Email IMAP. Includes the long-time FLOSS standard Dovecot, lightweight, and fast, with strong authentication and IMAP feature support.
- Email SMTP. Postfix, the open source software mail transfer agent (MTA) that routes and delivers electronic mail is well integrated within EnterpriseLibre. Its extensions feature deep content inspection in the mail queue, mail authentication with DKIM, SPF or other protocols and SMTP level access policies such as grey-listing and rate control.
- Spam Control. EnterpriseLibre makes use of the long-time FLOSS spam filter DSPAM, using adaptive filtering to adjust to individual user accounts.
- Virus Control. Long time standard ClamAV is used to detect malicious attachments in email, mainly to help protect Microsoft Windows users.
- Calendars & Contacts. The leading groupware FLOSS app SOGo supports many clients

and protocols including CalDAV, CardDAV, GroupDAV, and Microsoft ActiveSync.

- Shared Folder. A dedicated folder for sharing files of any size very quickly, from images to documents to database archives, provided by the Linux operating system.

5.1.4.2 Middleware

The current release integrates the following middleware:

- Apps Server Web. The world's most widely-user web server software Apache, getting better since 1995, supports the enterprise's web apps.

- Apps Server Java. JBoss, the leading open source software enterprise Java application server, supports the enterprise's Java apps.

- Databases. Includes the two standard database management systems MySQL and PostgreSQL, providing databases for the enterprise's applications as needed.

- Languages. Includes server-side programming languages Ajax, Perl, PHP, Python, and Ruby for applications that use Apache compiled modules.

- Virtual Desktop Server. Includes the hardened and scalable X2Go server.

5.1.4.3 Intranet

The current release integrates the following Intranet software:

- Firewall. Uses long-time FLOSS software Shorewall, which uses the Netfilter (iptables / ipchains) system built into the Linux kernel, controlling traffic with rules defined at a higher level to make management of complex configuration schemas easier.

- DNS Server. Of course, every Intranet includes the long-time FLOSS standard Domain Name System (DNS) software BIND.

- Network. Uses long-time standard OpenVPN for secure connections in routed or bridged configurations and to connect multiple data-centers in one system.

- Security System. Uses OpenSSH to encrypt all traffic. Uses OpenSSL to implement the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS) protocols.

- Authentication. Long-time standard Kerberos provides secure network authentication so nodes can prove their identity to one another in a secure manner. Also uses Linux Pluggable Authentication Modules (PAM) to provide dynamic authentication support.

- User Directory. Includes the long-time standard OpenLDAP implementation of the Lightweight Directory Access Protocol (LDAP).

5.1.5 Enterprise Manager (Goto 5.2)

The Enterprise Manager provides a simple GUI where any non-technical Superuser can easily manage any aspect of their enterprise, including the domain name, name and email servers, users, applications, services, backups, and related settings, described in *Section 5.2*.

5.1.6 Virtualization

EnterpriseLibre is built on OpenVZ, the first container technology, providing full Linux-in-Linux capability, with a standard, portable design independent of hardware or IaaS. This architecture means any underlay that can run twelve Linux servers works fine for an EnterpriseLibre installation. One could deploy the solution on twelve different hardware servers in different locations, or run them all on one server or chip. The ability to run Linux servers *somewhere* will be around for a very long time. And as described in *Section 5.3 – System Manager*, it halves the operational cost if you are offering the solution as a hoster.

Software is the right-place for virtualization, and the kernel is the right layer. Hardware level virtualization has several efficiency limitations, and molecular lock-in.[18] And higher-level software virtualization has the same problem as multi-tenancy – trying to outperform the kernel, an impossible goal, since for every action – switching between different customer's process, accessing different customer's RAM – the operating system can make the switch between customers with orders of magnitude less code, and therefore less time.

Containers perform with the least overhead, and are the cleanest virtualization answer, providing as many Linux operating systems as desired, all transparently managed by the kernel. And when customers are hosted on large, shared hosted servers for efficiency and cost savings, they can be assured the separation goes beyond the application, or even process, to entire virtual operating systems. LXC and other container technologies can be added (relatively) easily when ready, with the same security as OpenVZ where they provide the same Linux-in-Linux capability, and more if approved by Mr. Torvalds.

Containers enable virtualization scalability with almost no overhead, providing a standards, performance, and security win-win-win. In this case, the centre holds best.

The following table summarizes the relationships between the EnterpriseLibre virtual servers.

[18] Hardware virtualization has three efficiency problems: (a) Stasis – achievement of an all-vendor standard, even if doable (lol), would lock into hardware a technology that would be impossible to update and slow to improve; (b) Diversion – every penny spent on chip virtualization is not spent on performance of the chip itself; (c) Scale – some virtualization decisions must be made in the chip and cannot be changed, limiting the performance of some use cases, such as large numbers of virtual servers that containers can support with near linear performance, significantly contributing to EnterpriseLibre efficiency in both the single customer and hosted model with load-balancing.

\ <---> \	DNS	Fire	Auth	User Dir	DB	Email	Web	Java	XMPP	Group	NX	Back
DNS						SMTP:25						
Firewall	TCP:53 UDP:53					SMTP:25					SSH:22	
Authentication	TCP:53 UDP:53					SMTP:25						
User Directory	TCP:53 UDP:53					SMTP:25						
Database	TCP:53 UDP:53					SMTP:25						
Email	TCP:53 UDP:53		UDP:88	TCP:636								
Web Server	TCP:53 UDP:53	HTTP:80	UDP:88	TCP:636	TCP:3306 TCP:5432	SMTP:25 SMTP: 10026		HTTP:80 HTTP:443				
Java Server	TCP:53 UDP:53		UDP:88	TCP:636	TCP:3306 TCP:5432	SMTP:25						
XMPP	TCP:53 UDP:53		UDP:88	TCP:636	TCP:3306 TCP:5432	SMTP:25						
Groupware	TCP:53 UDP:53		UDP:88	TCP:636	TCP:3306 TCP:5432	SMTP:25						
NX Desktop	TCP:53 UDP:53		UDP:88	TCP:636	TCP:3306 TCP:5432	IMAP:993 SMTP:587	HTTP:80		TCP:5222	HTTP:80		
Backup	TCP:53 UDP:53 SSH:22 SCP:22	SSH:22 SCP:22	SSH:22 SCP:22	SSH:22 SCP:22	SSH:22 SCP:22	SSH:22 SCP:22 SSHFS:22 SMTP:25	SSH:22 SCP:22	SSH:22 SCP:22		SSH:22 SCP:22	SSH:22 SCP:22 SSHFS:22	

Table 5-1: Virtual Server Inter-communications

5.1.7 Operating System

Linux, the OS kernel that runs the rest of the software, is designed with the Unix architecture created by Dennis Ritchie and Ken Thompson in 1971, built with free software tools provided by Richard Stallman's GNU Project started in 1983, and started by Linus Torvalds in 1991.

Mr. Torvalds has shepherded it ever since, helping create the world's most valuable software, stratospherically when weighted by brevity, and continuing to get better. Supported by a majority of the world's technical companies, ubiquitous from Android to Raspberries, it has far the best standardization, capability, security, and sustainability, and may be the greatest challenge to entropy humans have yet developed.

EnterpriseLibre uses Ubuntu Linux, from Debian, for all servers, twelve for each Intranet enterprise. The current release uses Long Term Support (LTS) 10.04, and needs updating. This baseline provides a complete, production-ready enterprise solution, so updating any component is now a much easier, targeted effort, producing an even better, still production-ready solution. Updates of servers is a near-term next step.

Since the EnterpriseLibre foundation is the Linux virtual server, it can be moved between local installations and hosting relatively easily. Linux runs on almost any server or online IaaS, and clean separation at the server level enables the Linux kernel to do what it does best, manage CPU, RAM, and I/O with extreme efficiency. When hosted, large, powerful servers can balance the load across users, customers, and even time-zones, so less hardware is needed.

5.1.8 Hardware Server

Almost any server that can run Linux can run EnterpriseLibre. Additional specifications are provided below, collected mostly from experience with Cirrus Computing customers.

- RAM. The critical constraint is RAM, with full provisioning averaging about **1 GB** per user, assuming all users are logged in at all times, a perhaps initially surprisingly low number, however accurately reflective of the use of a general user base. Each enterprise's system and management software uses less than **4 GB**. Therefore, a server's RAM requirements can be sized, as a requirement if buying, as a loading max if hosting, by the formula:

  $$RAM = 4 \times Enterprises + Users \quad \text{(GB)}$$

  So for example, an appliance server for one enterprise and ten users needs at least $4 \times 1 + 10 =$ **14 GB** of RAM.[19] An organization anticipating five enterprises and 100 users needs $(4 \times 5 + 100) =$ **120 GB** of RAM, which is possible today with a single appliance server. Most rack mounted Cirrus Computing servers have **96 GB** RAM.

- CPU. Most software requires little CPU, so large servers can easily time-share many users with great stability. Cirrus Computing testing shows an active user requires (as an approximate but single metric) under **0.25 GHz** CPU. With server performance improving along several dimensions, this approximate measure indicates a modern server with **eight 4-core 3 GHz** chips can support around **384** users. CPU is a secondary constraint, or, in other words, good processing performance is not costly.

[19] For less than 25 users it would be a good best practice in general, not just for EnterpriseLibre, to add 8 to 16 GB RAM just to make sure the server can handle any spikes in a small pool.

- I/O. Particularly on large servers, after memory and processing are satisfied, input / output (I/O) capacity can be key to performance, and Cirrus Computing found a couple situations where it became the bottleneck resource. Software tuning can help, however I/O is the third priority constraint and not a cost driver, so the best approach is to ensure the server used has more than enough capacity for the load it will process.

- Storage. With storage technology improving rapidly, ten TB's and more can be put on board a standard 1U server, the current EnterpriseLibre architecture. Cirrus Computing customer experience shows use grows over time of course, however most growth is in email and attachments, averaging less than **1 GB** a user a year. However, to support nearly unlimited storage capacities, an enhancement is planned to enable addition of FLOSS network attached storage – in the same rack, one rack over, or more distant.

5.2 Enterprise Manager

This section describes the Enterprise Manager, giving any user one-click control over their entire enterprise from one, easy user interface.

5.2.1 Architecture Diagram

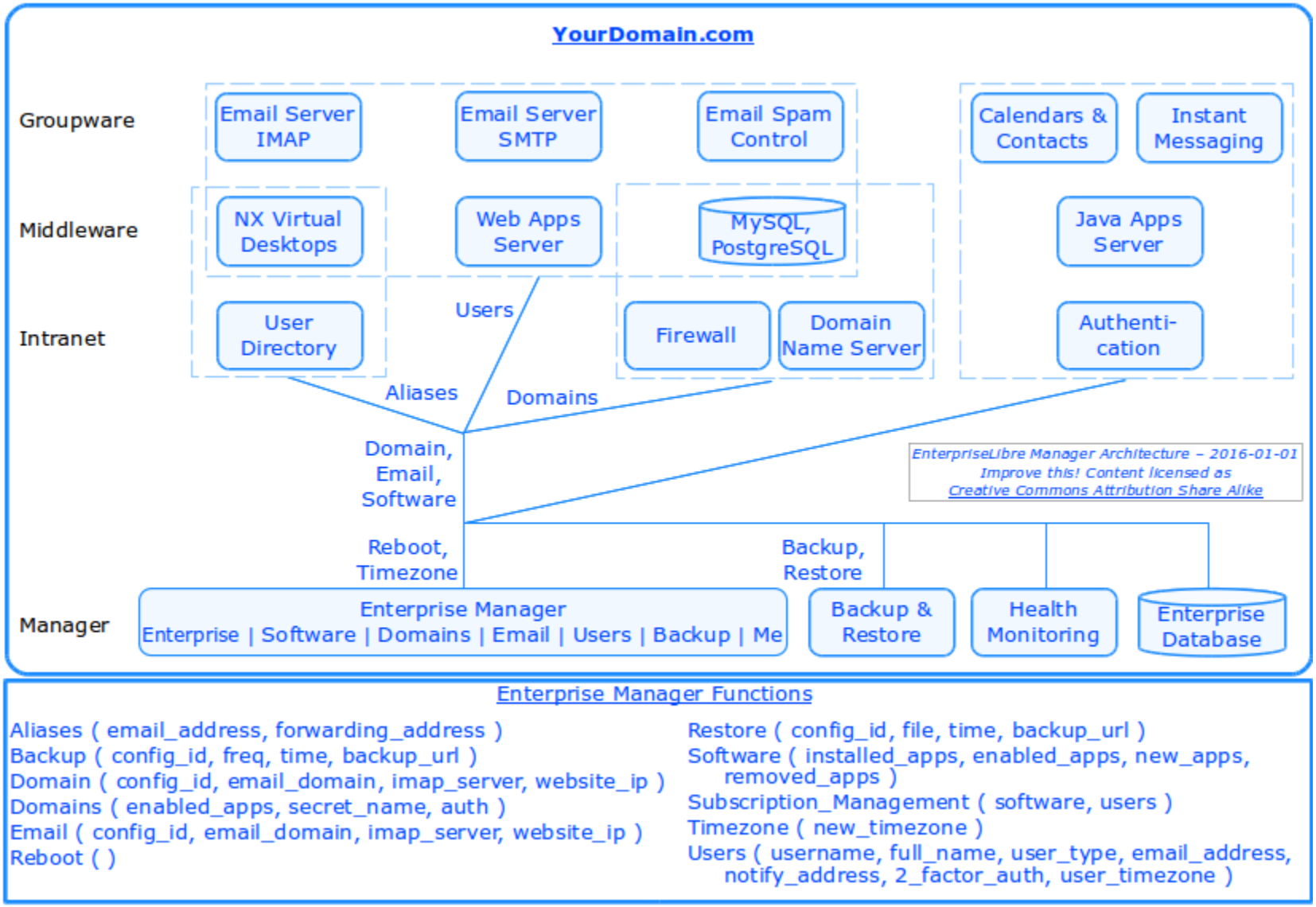
From first setup to ongoing life-cycle management, the Enterprise Manager coordinates all the software in an Intranet enterprise however the user wishes.

For first use, the Enterprise Manager enables a team to add the solution without changing any current software, by connecting it to their existing name and email servers. They can then create virtual desktops for users with the same email addresses they have now, providing instant hosted capability that keeps all their email in sync like any client application, except this client is an entire virtual desktop attached to a complete enterprise solution of its own.

Then, whenever they wish, the Superuser can click a button to switch over their email service, and EnterpriseLibre will automatically transfer all their user's existing email to the solution, and manage the email going forward. With the Domains tab, the Superuser can make the email server names identical, so anyone using “imap.mydomain.com” won't need to change a thing. Then they can start using EnerpriseLibre applications and stop using old systems at whatever rate they desire, with almost no transition cost.

Some parts of the Enterprise Manager, backup, and monitoring are currently centralized, and will be fully modularized in a near-term update, so each enterprise coordinates data with the System Manager only as required.

The following architecture diagram shows the Enterprise Manager, components, functions, and relationships between them.



Graphic 5-2: Enterprise Manager Architecture

5.2.2 Backup

The backup system uses the standard FLOSS app Duplicity to create incremental, space-efficient archives, managed in almost any location using a wide range of protocols. GnuPG encrypts the archives, so they are secured in storage and while being transferred.

5.2.3 Monitoring

Enterprises are supported by the monitoring system Nagios, a widely used FLOSS standard for monitoring networks, hardware, and software of all types. Alerts can be set for when there are problems, and when they're resolved.

5.2.4 System DNS

When using their solution's name server, system name servers are also configured as gateways, so users can point their domain name at the registrar to these unchanging gateways, instead of automating changes to glue records, and having to make manual changes at the registrar for every domain change. For example, the Cirrus Computing system "virtualorgs.net" uses the name unchanging gateway servers "ns1.eseri.com" and "ns2.eseri.com", passing all communications (unmonitored) through to the enterprise, no matter their current domain name.

For those that wish to remove this system gateway, an enhancement is planned to automate glue record changes, so an enterprise can use "ns1.mydomain.com" directly, and even change it dynamically to "secret789.mydomain.com", of course with the trade-off of having to make a manual change at the registrar each time. For those using an onsite solution, the same effect can be achieved with the current EnterpriseLibre by setting these system name servers to be the same as the organization name servers (to be available from the System Manager).

5.2.5 Payment Interface

The payment interface provides automated support for Superuser changes to user desktops, applications, or email accounts, the items that most directly affect cost and make the most sense for sustainable pricing. The current software implements immediately any change that falls within the current subscription cost, so if two applications are suspended and a new one selected at the same price, installation proceeds immediately. If an update requires a larger subscription cost, the payment interface handles the transaction, after which the Enterprise Manager implements all queued changes.

5.2.6 Database

The database stores all required management data for each enterprise, including anchor domain name, current domain name, email server, service domain names and settings, software, users, and related information. The standard PostgreSQL database is used.

5.2.7 User Interface

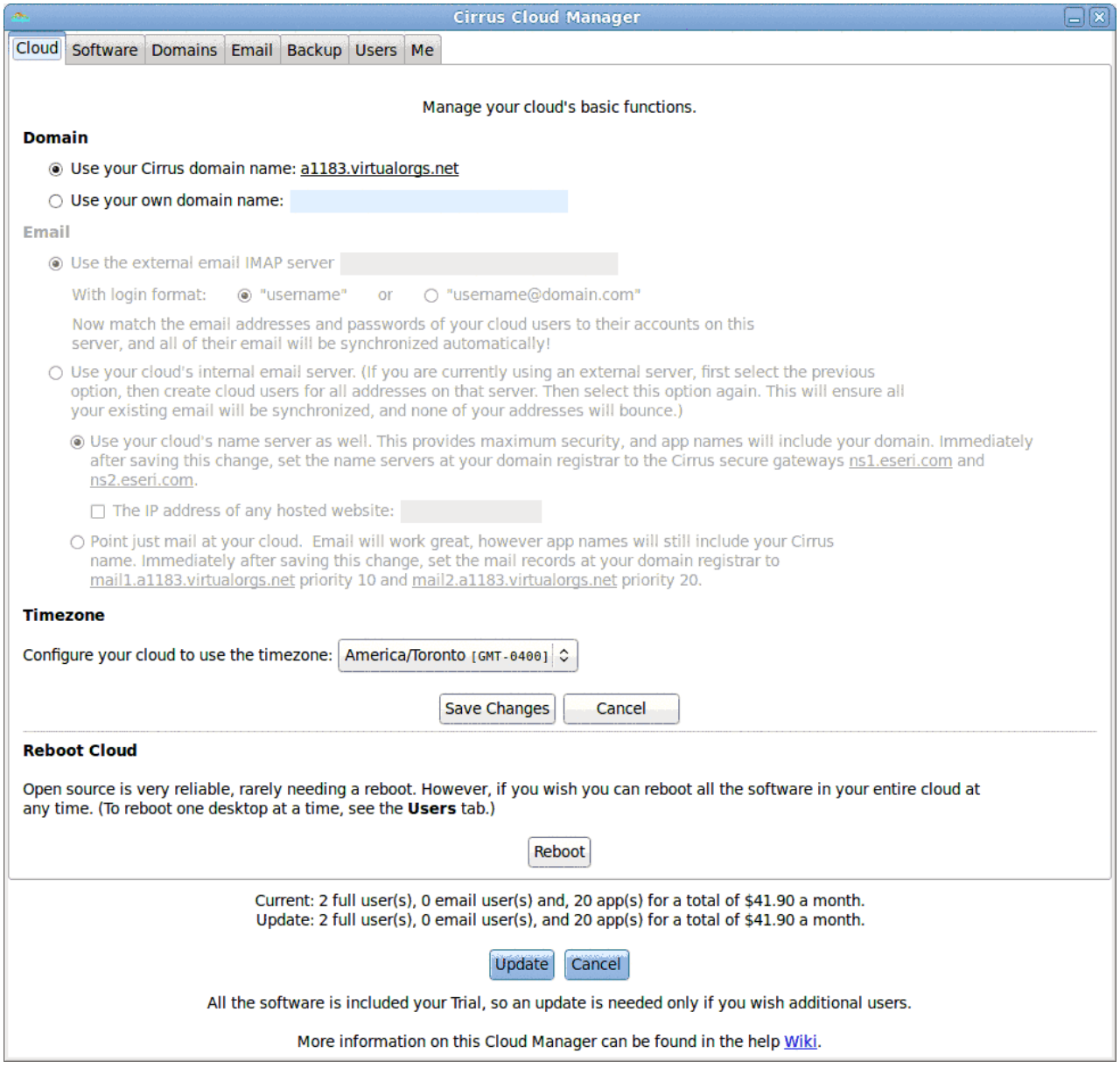
The Enterprise Manager enables anyone to perform enterprise configuration of their entire enterprise with an ease never available before, providing all the capability specified in the requirements for One-Click Management.

5.2.7.1 Enterprise Tab

This tab gives the Superuser the ability to easily configure the solution's domain and email:

1. Either use the solution's anchor domain, "a12345.virtualorgs.net" or;
2. Use your own domain name, and then either:
    a. Connect to an existing external email server.
    b. Use your internal email server, and internal name server (most secure).
    c. Use your internal email server, however keep your external name server.

The Superuser can also change the enterprise timezone, and reboot the entire solution.

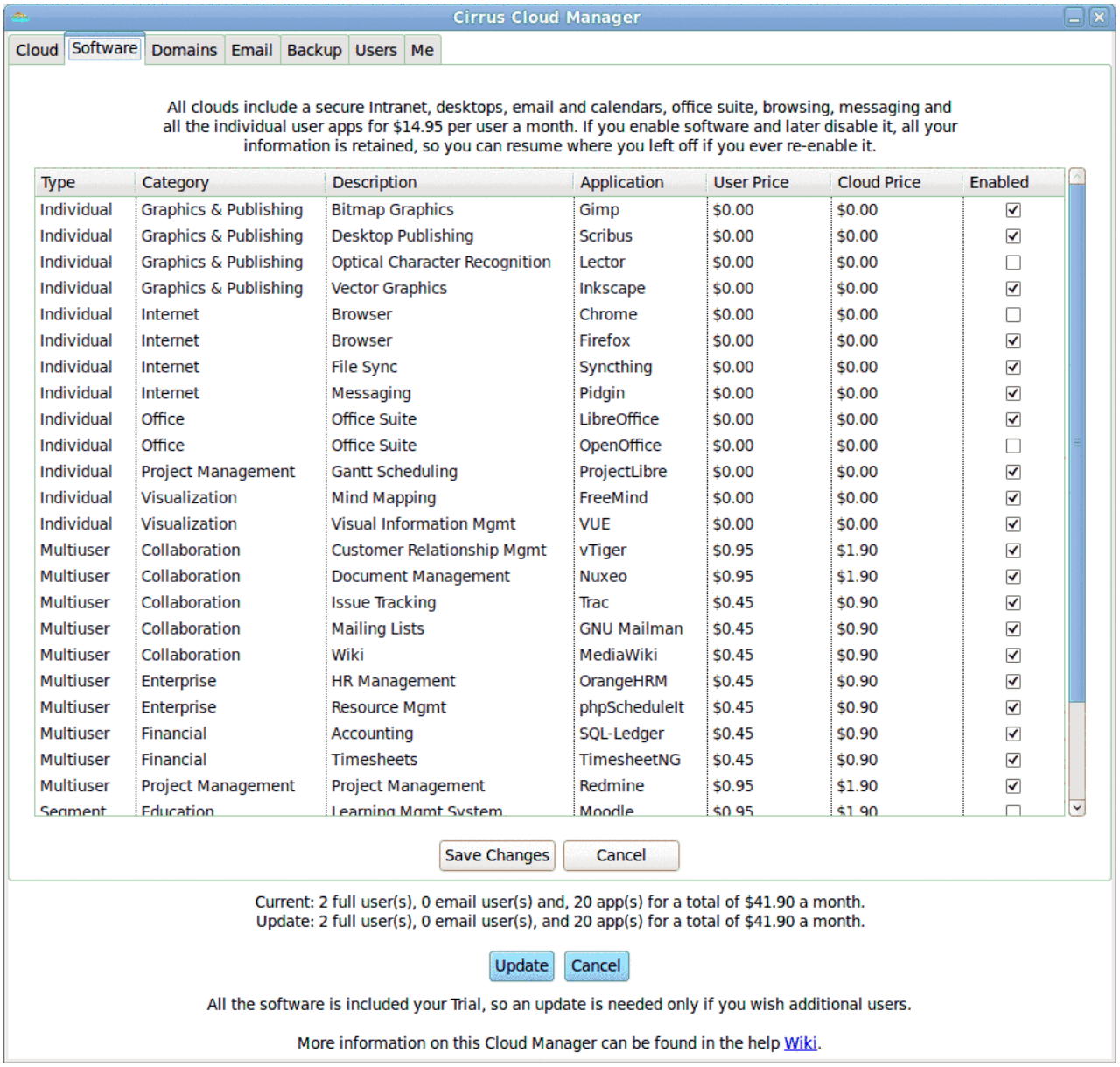


Graphic 5-3: Enterprise GUI

5.2.7.2 Software Tab

All solutions include the enterprise foundation, and every desktop includes a dozen standard FLOSS individual use apps. This tab enables easy addition of multi-user software, that only some teams want.

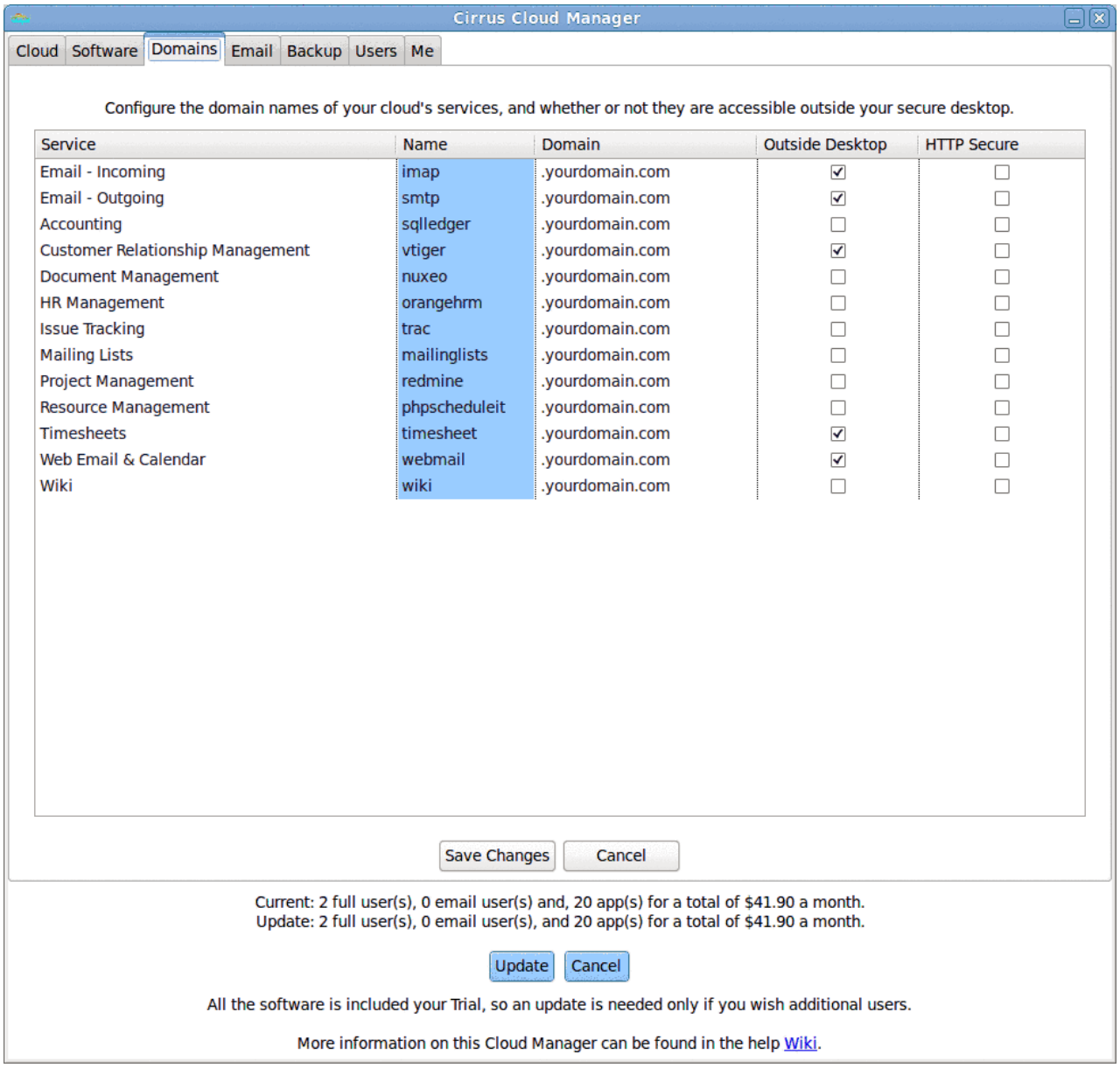
The manager automates all installation, configuration with system information such as domains and email server, and setup of accounts for all users with single-sign-on. If an application is disabled and then re-enabled later, all information is retained so users can resume where they left off.



Graphic 5-4: Software GUI

5.2.7.3 Domains Tab

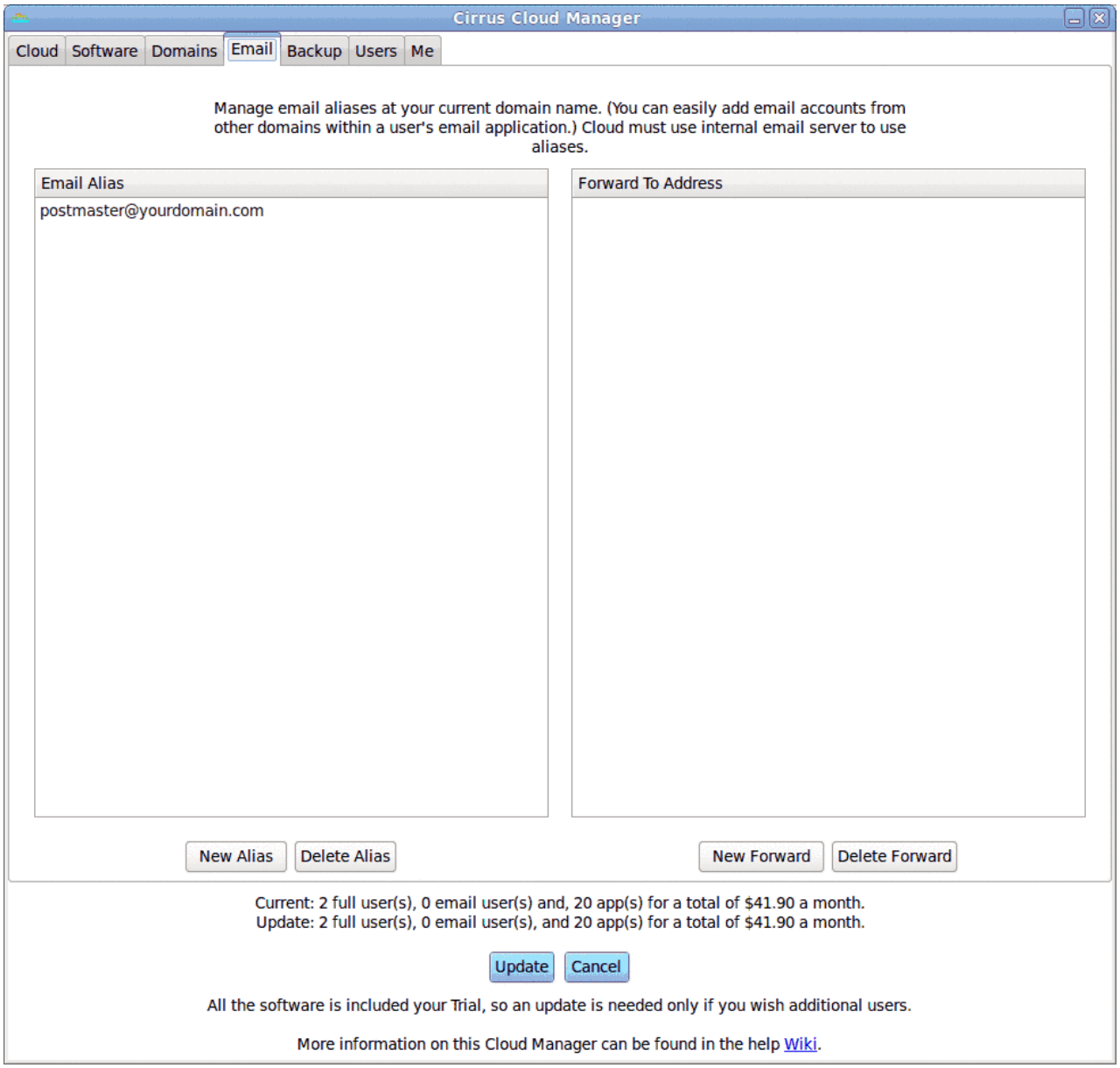
When using the anchor domain, or customer domain name and internal name server, this tab gives the Superuser the ability to configure the names of the solution's apps and services, their accessibility outside the secure desktops, and security settings. If the Superuser changes a domain name from "wiki.yourdomain.com" to "secret.mydomain.com", access by the old name is instantly disabled, and because the enterprise name server is configured in silent mode, the new name must be known for access to continue.



Graphic 5-5: Domains GUI

5.2.7.4 Email Tab

When using the internal email server, this tab gives the Superuser the ability to easily manage email aliases and forwards at the current domain name, perhaps the most common customer system requirement.

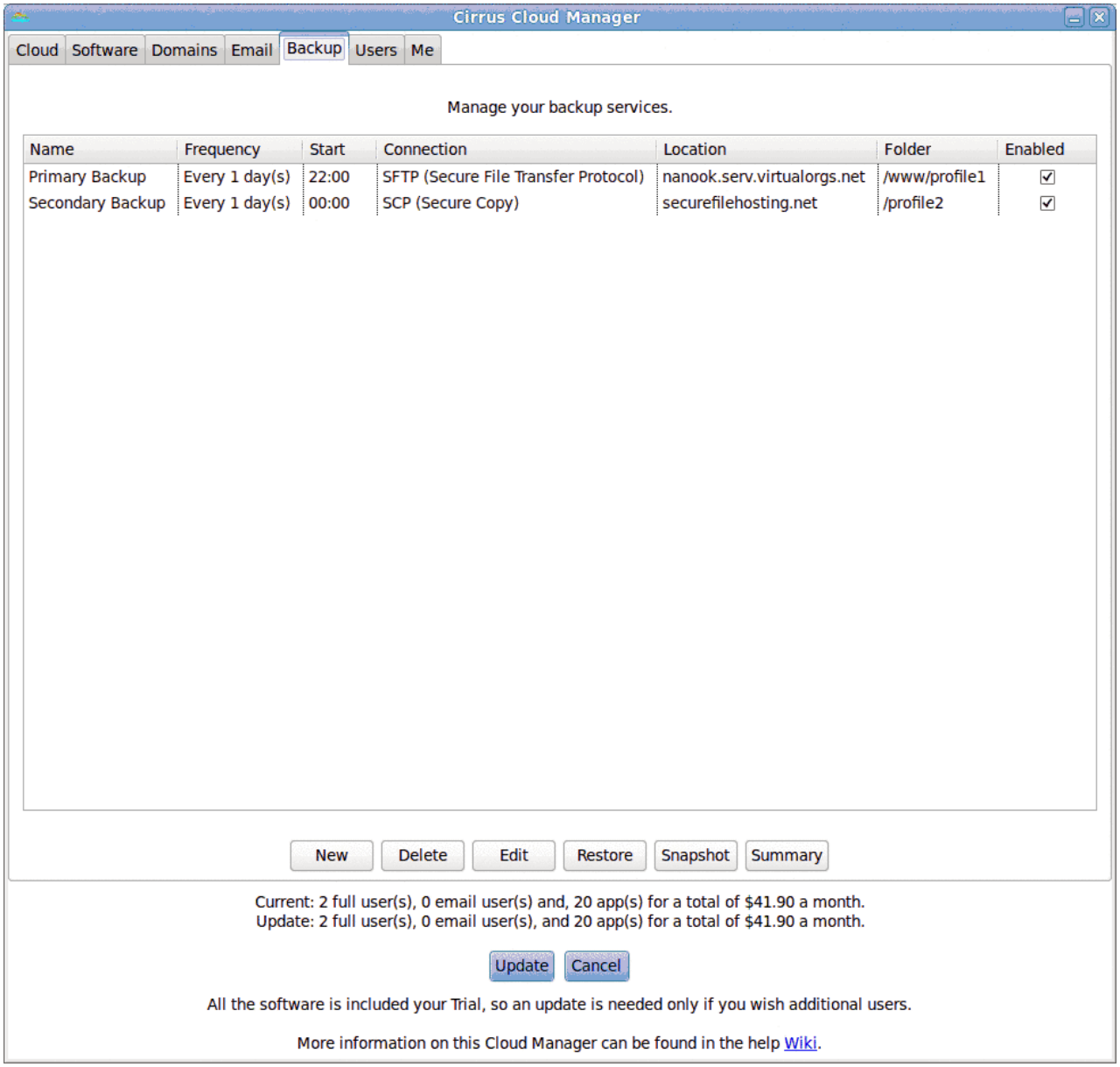


Graphic 5-6: Email GUI

5.2.7.5 Backup Tab

This tab gives the Superuser the ability to easily configure backup services, controlling their team's fundamental security and sustainability. Incremental, space-efficient, secure backups can be managed in almost any location using a wide range of protocols, such as Amazon S3, FTP, SCP, etc. The Superuser can restore any file (without being able to read user files). Users can restore their own files from their individual Enterprise Manager.

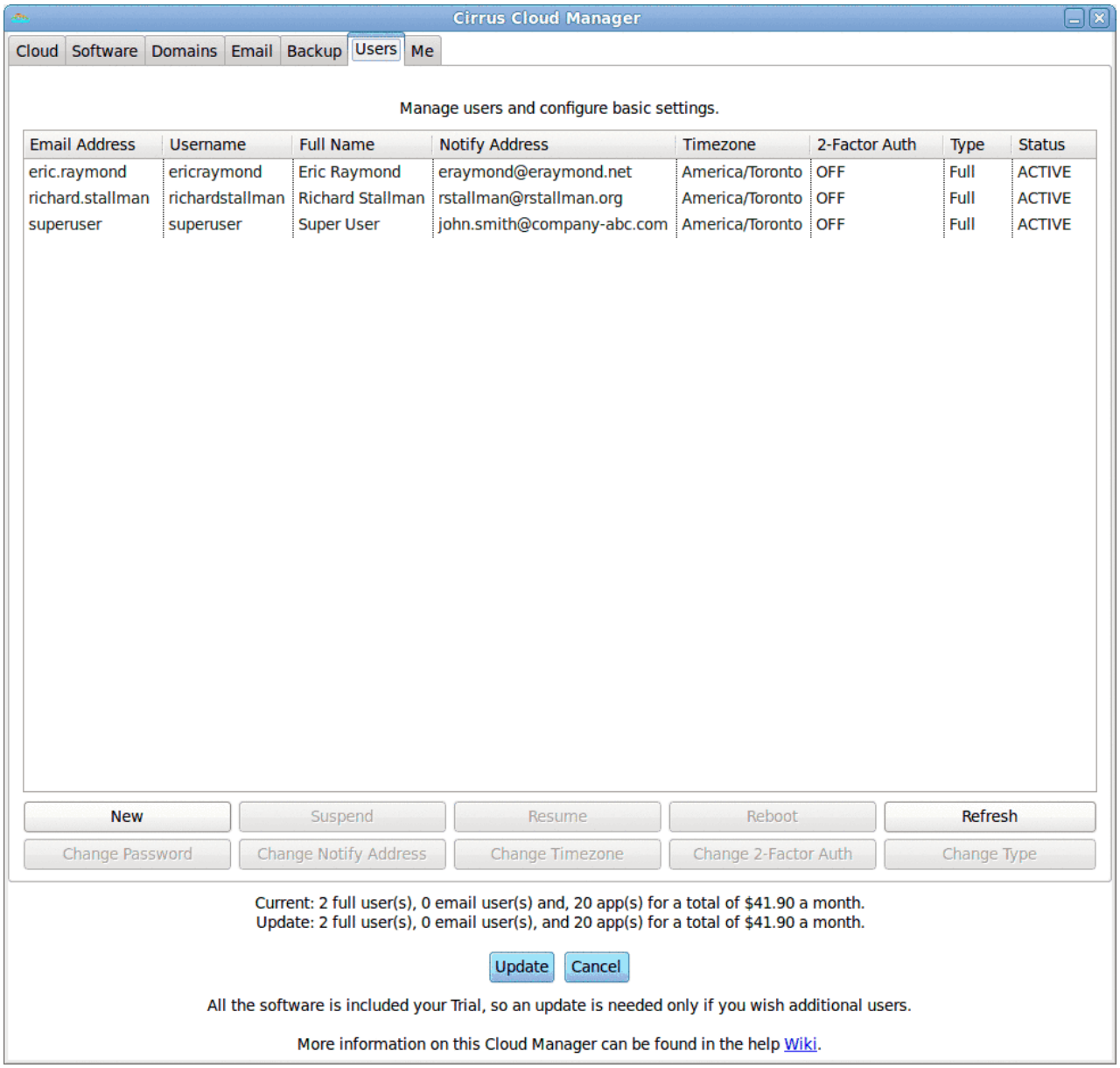
The schedule can start at any time, and repeat any number of hours, days, weeks, or months. Currently, the Primary backup cannot be changed since it is the first line protection, however it can be disabled if really desired. New backup services can be created without cost, since there has not been enough use to know if it will have significant load; if it does, it can be included in the simple subscription pricing using the payment interface.



Graphic 5-7: Backup GUI

5.2.7.6 Users Tab

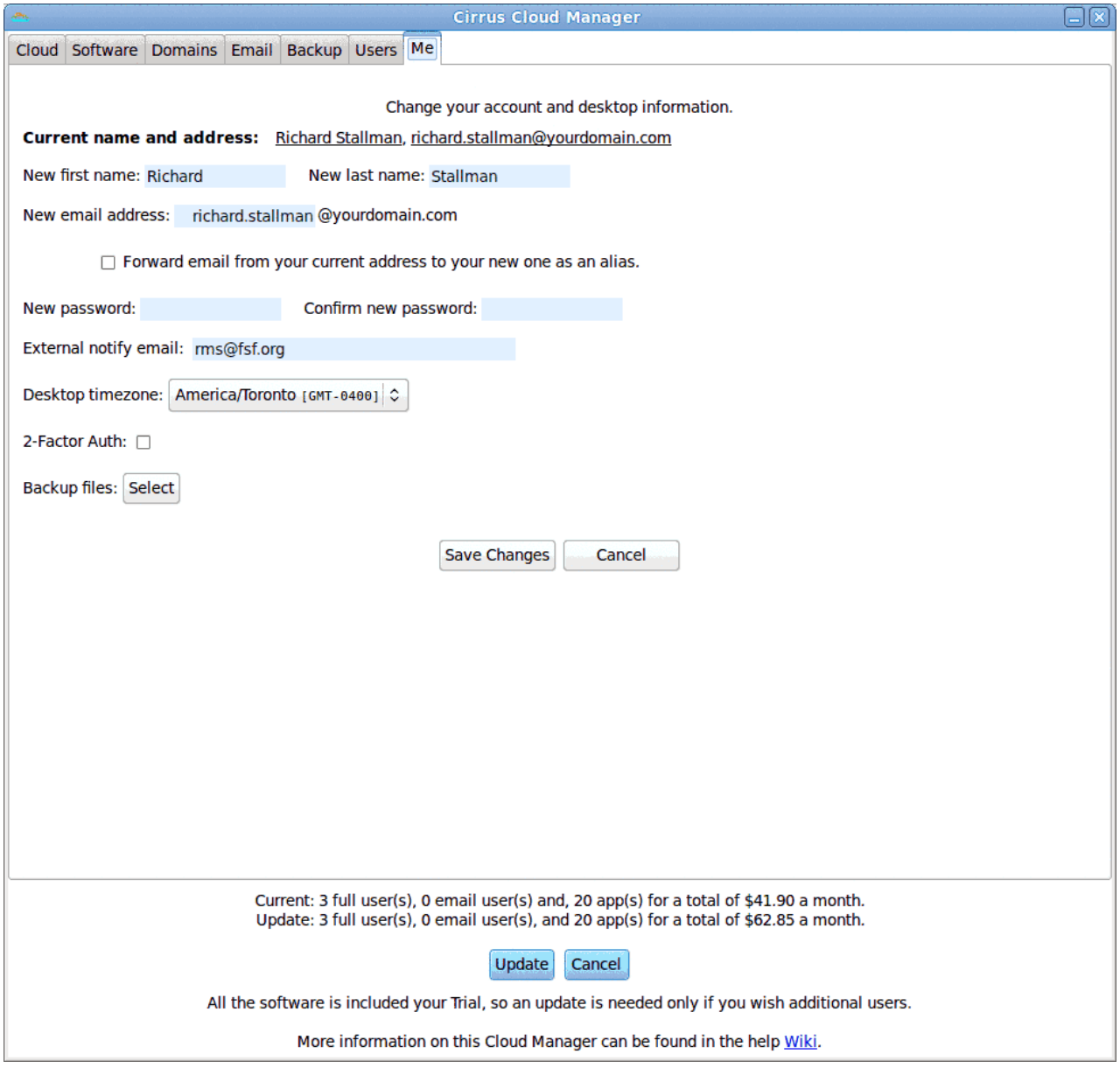
This tab gives the Superuser the ability to add, suspend, and resume user accounts with one click, including virtual desktop, email and calendars, and applications including single-sign-on. The Superuser can also mandate use of two-password login, change an account's notify address and timezone, reboot the virtual desktop, initiate change of password, and change the type between full desktop and email address only.



Graphic 5-8: Users GUI

5.2.7.7 Me Tab

This tab gives the users the ability to make changes to their own account, including name, email address (with option to forward old one), password, notify address, timezone, and two-password requirement (if not mandated by Superuser). The user can also restore any of their own files and folders by selecting from the available backups.



Graphic 5-9: Me GUI

5.3 System Manager

> *The more widely used FLOSS is, the sooner we obtain a cosmic win^win.*
> – W. Stewart, 2013.

This section describes management of a system of more than one enterprise, from several on a single server to thousands on hundreds of servers. All the key functionality is working, and most of it is linked to the easy System Manager GUI.

5.3.1 Architecture Diagram

Analysis shows there is a reason Amazon Web Services are so profitable: if you need servers full-time, they cost about twice as much as outsourcing the data-centers and owning the hardware. Data-centers are an unchanging, very low-cost commodity service, cabinets and all. Server specification and configuration is completely dependent on the need.

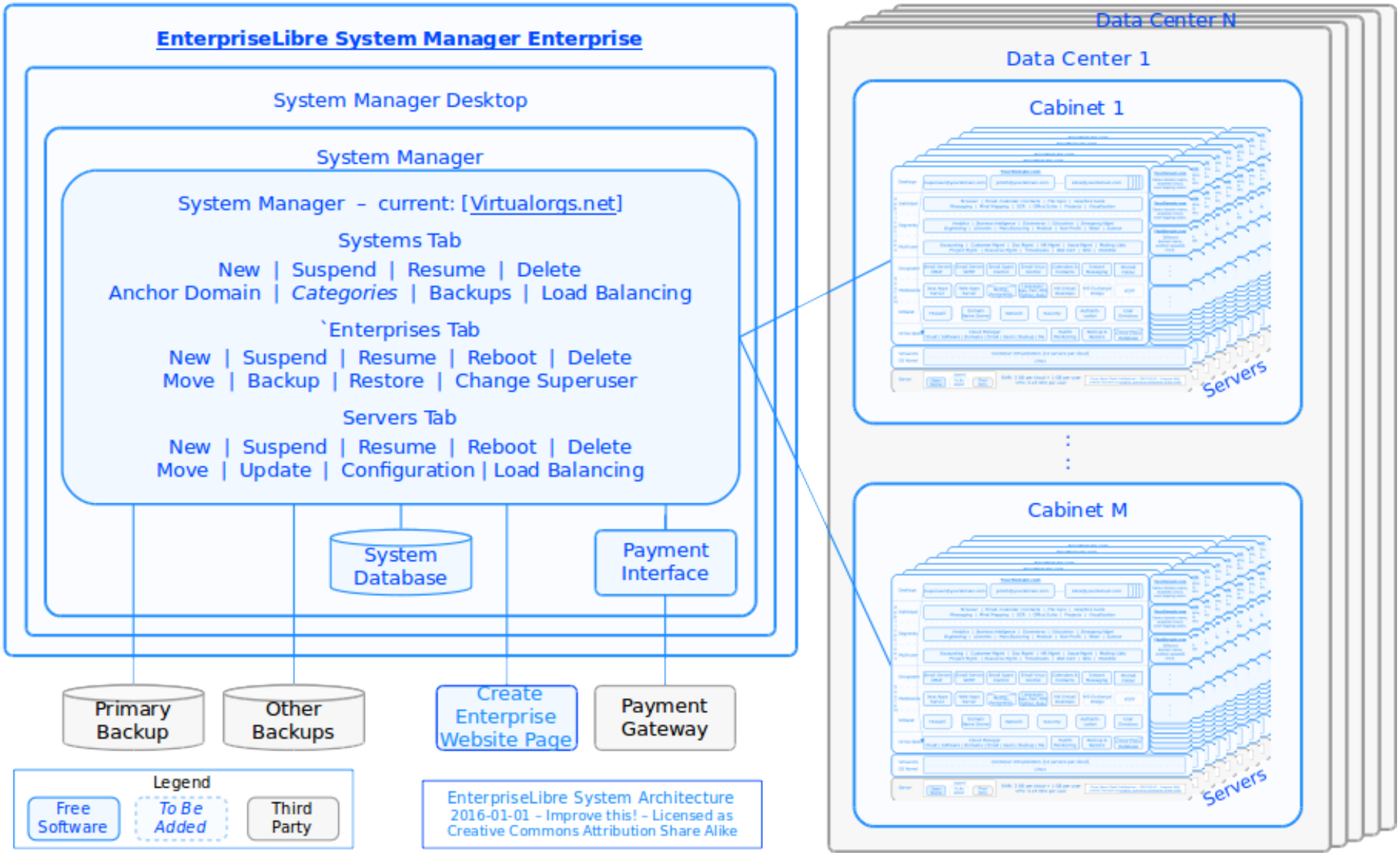
Therefore, EnterpriseLibre is designed to run on a single onsite appliance server, or hundreds of servers anywhere with all remote management, making any EnterpriseLibre hoster's scalability challenge a simple matter of having servers shipped to data-centers as needed. It runs fine on any hosted IaaS, however does not require one, and does not need the approximately doubling of hosting cost that extra layer brings.[20]

This architecture, together with near zero overhead container virtualization, single-tenancy optimum CPU efficiency, and the load-balancing described in *Section 5.3.6*, enable provision of EnterpriseLibre very inexpensively compared to other hosted offerings.

Each system is defined by an "anchor domain name", an Internet home where enterprises live when not configured for some other domain. The main Cirrus Computing system uses the anchor domain "virtualorgs.net", and each enterprise has a unique third-level anchor domain like "a11235.virtualorgs.net".

The following diagram shows the top-level system architecture with the data-centers, cabinets, servers, management software and other components, and the relationships between them.

[20] In practice, we also found that none of the off-the-shelf IaaS virtual servers had the right size for EnterpriseLibre, as described in Section 5.1.8, and so were even more expensive due to waste of resources in excess of requirement in order to obtain the critical resource of RAM.



Graphic 5-10: System Manager Architecture

5.3.2 Data-Center

A multi-enterprise system can be located all in one or across several data-centers. The main Cirrus Computing system currently links five production servers in two data-centers in different provinces, and a third backup location.

The normal configuration in most third-party data-centers is servers in locked cabinets. Some security policies may require keys to be controlled by the user. Cabinets in one system can be located in multiple data-centers. While it has not been tested at load, the current System Manager could probably easily scale to **100 servers** with **1,000 users** each for **100K users** in one system, with hardware performance alone steadily increasing the possible scale.

An entire system can be installed on much smaller servers as well, as a single "appliance", where the "data-center is effectively the local office.

5.3.3 Networks

All networks are encrypted, using standard open source software public key cryptography virtual private network (VPN) protection. Virtual desktop communications are protected by OpenSSH, with the NX channel using a private key of **1024** bits. Each enterprise uses an OpenSSL RSA **2048** bit SHA1 certificate for IMAP, SMTP, and related network communications.

5.3.4 Payment Interface

The current EnterpriseLibre includes user self-serve enterprise creation at any time: a solution is requested, payment made, built in about an hour, and an email sent to the Superuser when complete. The payment interface currently supports PayPal, and can be adapted for other providers. A web page enabling this self-serve enterprise creation currently works from the website, however is not yet public for scalability reasons.

5.3.5 Database

The database stores all required management data for each enterprise, including anchor and configured domain name, name and email server, service domain names, software, users, and related information. The standard PostgreSQL database is used.

5.3.6 Load Balancing

> *Big numbers make life easy... at around room temperature, gas molecules move with speeds ranging from zero to several thousands meters per second. Despite this variability, the average speed of a gas molecule... shows only tiny fluctuations.* – Michael de Podesta, Understanding the Properties of Matter, 2002.

EnterpriseLibre supports scalability of multiple Intranets, servers, and data-centers. However, just Intranets and servers need be considered to manage load-balancing and obtain maximum performance with minimum resource use, as described in the following sections.

5.3.6.1 Intranets

Load balancing requires management of the critical cost-driving resource RAM, swap space, and virtual server parameters. The current EnterpriseLibre includes the following resource management lessons, learned from customer experience. to resolve the main problem of out of memory (OOM) RAM shortages, with almost no swapping or need for support, thereby meeting the requirements for No Maintenance:

1. RAM. To guarantee the analysis, assume all users use their desktop at all times. Fill servers to a maximum defined by the formula from *Section 5.1.8 – Hardware Server*, *(4 x Enterprises + Users)*. And size servers with enough RAM to ensure individual peaks are absorbed, in practice needing just a few dozen users, and easily achieved with **96 GB** RAM or more at less cost each year.

2. Buffer. Next, leave a 10% buffer to reduce the probability of swapping even more, already low due to the unrealistic sizing assumption of a 100% usage rate. For example, on a **256 GB** RAM server, reserve **25.6 GB** leaving **230.4 GB** for Intranet enterprises. Then scale solves the problem, since the larger the server, the less chance the buffer will ever be needed. That is, if the buffer on a **256 GB** RAM server is used by some amount X% of the time, on a **1024 GB** RAM server it will be used that amount much less than X% of the time. The performance becomes much more predictable – less like the weather, more like the tides.

3. Swap. Next, provide a swap file at the Linux kernel level only, sized to be "more than enough" since disk is inexpensive, for example using the formula ***(128 + 2 * RAM)***. So for a **128 GB** RAM server configure **384 GB** disk swap, and for a **1.0 TB** RAM server configure **2,128 GB** (2.125 TB) disk swap. Greater optimization at this point is counter-productive, since, while RAM is the cost driver, in absolute terms **1 GB** per user is inexpensive enough at scale that savings with more disk swap are not worth any performance cost to the user. With this design, swap is designed not to be used, rather only as an inexpensive backstop to guarantee there will never be an OOM crash, and even the worst cases of runaway RAM problems can only cause a slow down.

4. Virtual Servers. With this design solving resource management, additional controls at the container level are unnecessary. Which is good, since they were found to be practically unmanageable. With twelve virtual servers in each enterprise, each with varying levels of RAM, process, thread, and other requirements, any customized limits, alerting, and management were found to be infeasible, i.e. not cost-effective for a hoster to manage at scale. However, as the solution reliability and availability stabilized, virtual server resource monitoring also became unnecessary, so all container level management was removed, leaving it to the best positioned layer: the kernel. In particular, this means there are no container level memory limits, so near zero chance for OOM crashes. For example, a virtual server on a **512 GB** RAM server has **51.8 GB** of RAM buffer then another **640 GB** unused disk swap to use up first.[21]

This design has close to optimum resource efficiency, i.e. least wasted resources and cost, since a 10% buffer can be reduced by only single-digit percentages ;-) The performance is also close to optimal, since swapping is reduced to effectively zero on larger and more powerful servers in smaller space, which remains the continuing trend.

[21] In an enhancement, alerts will be sent to the Superuser if a virtual server ever uses twice the expected requirement, since it is then likely there is a genuine problem. For example, the Java apps server typically takes 2 GB RAM maximum, so an alert will be sent if it ever reaches 4 GB. The large hardware server can handle the load fine, and performance will be maintained, however the Superuser is notified, and can choose to reboot the virtual server or entire enterprise if desired.

5.3.6.2 Servers

While large servers can resolve most scalability issues, a hoster needs to be able to support more than one server. In other words, a customer using an appliance server likely only uses the System and Enterprises tabs of the System Manager, however a hoster also needs the Server tab, and a way to cost-effective load-balance enterprises across them.

The two steps to server load-balancing are selection of the server for initial enterprise creation, and moving enterprises when one becomes too full. Static load-balancing is automated, while dynamic load-balancing is currently a manual procedure, however simple as described below:

1. Static. Memory is the scarce resource, the most expensive to provide and therefore the limiting parameter. Therefore, when more than one server is available, the current static load balancing algorithm chooses the one for new enterprises with the most available RAM, using the formula *(4 x Enterprises + Users)* from *Section 5.1.8 – Hardware Server* to calculate the current load for each server. The System Manager GUI will include the ability to change these parameters, from the defaults of **4 GB** per enterprise and **1 GB** per user, for each system and individual server if desired to reflect different use cases.

2. Dynamic. When a server becomes too full, for example when it starts to swap more than X% of the time, an existing enterprise must be moved from the server to another. Currently this is a manual decision and procedure, however can easily be automated with a variety of algorithms in the future. In the meantime, analysis indicates manual moves are required infrequently enough at scale they are not a material cost for a hoster.

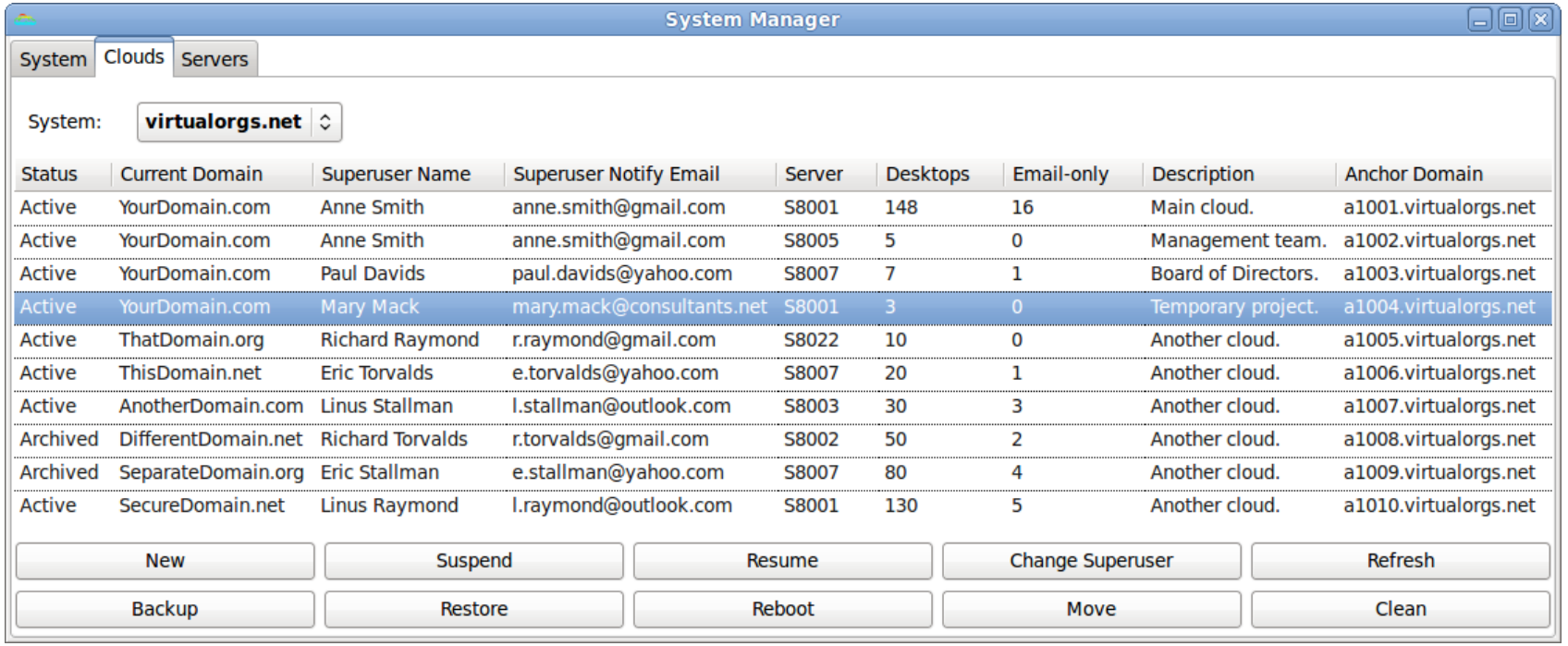
Enterprise load-balancing provides opportunity for significant savings, since the more users and more diverse the population, the more stable the performance. With hundreds of users on one server, statistical smoothness becomes the norm, since the thing operating systems are best at is time-sharing. With a server sized with hundreds of GB RAM and dozens of CPU cores, the chance of a resource issue Linux can't handle becomes vanishingly small. The reserves can be reduced, and load-balancing even by timezone across multiple servers can save additional infrastructure, cost, and environmental impact.

EnterpriseLibre has the pieces in place to (relatively) easily add these kinds of system-level dynamic load-balancing for automated management of large-scale enterprise populations. First level estimates indicate per-user operational costs around **$1.00 per month** are attainable, with hardware improvements halving the cost every few years.

5.3.7 User Interface

The System Manager application provides life-cycle management for systems of enterprises and servers from one simple user interface. This is the last, top-level component to be finalized in the bottom-up EnterpriseLibre build, and some work is still continuing, as noted in the text.

The drop-down at the top will enable selection of the current system from an alphabetical list, which will then change the contents of the Enterprises and Servers windows appropriately.



Graphic 5-11: System Manager GUI

5.3.7.1 Systems Tab

This tab enables life-cycle management of systems and editing of their settings, and is currently in development. The functionality is truly top-level: mainly updates to information recorded about the systems in the database.

- New. Create a new system, and set the anchor domain, default load balancing numbers, and default backup location. (At this top level this just requires creation of a new record in the Systems table).
- Suspend. No further configuration possible until resumed. Can only suspend a system if it has no live enterprises.
- Resume. Re-enable configuration and use of a system.
- Delete. Complete delete a system (archive data with time-stamp).
- Anchor Domain. Edit the anchor domain name across all enterprises in that System.
- Categories. Edit the labels used to reserve servers for enterprises in matching categories, such as "Medical", "EU", "Company ABC", etc.
- Backups. Edit the default backup location for enterprises in that System.
- Load Balancing. Edit the load-balancing parameters for enterprises in that System, with the default set to **4 GB** per enterprise and **1 GB** per user, and 100% for usage rate.

5.3.7.2 Enterprises Tab

This tab enables life-cycle management of enterprises and editing of settings. All functions are working, and, except for Backup, Restore, and Change Superuser, are hooked to the GUI.

- New. Create a new enterprise, given just the notification email address of the Superuser. The server is chosen automatically, however can be specified. Assembly of a new enterprise now takes under an hour.
- Suspend. The enterprise is shut down.
- Resume. Enterprise is restarted.
- Reboot. Restart all the virtual servers in the enterprise (copy of Superuser capability).
- Delete. Complete delete an enterprise (archive data with time-stamp).
- Move. Move from one server to another, including from one data-center to another if desired This is the manual step for dynamic load-balancing.
- Backup. Snapshot backup the enterprise. Backup copies only the data, very efficiently for space and bandwidth. Experience indicates backups require about **1.5 GB** per user.

- Restore. Restore an enterprise's data from a backup.
- Change Superuser. Change the Superuser notification contact data.

5.3.7.3 Servers Tab

This tab enables life-cycle management of servers and systems, and is currently in development. The primary function is New, which installs a new server with an EnterpriseLibre configuration, and is currently largely automated, however run manually.

- New. Add a new server to a system, installing the EnterpriseLibre foundation over the network, and set any custom categories, load-balancing, or backup settings to override any system defaults.
- Suspend. No further use or configuration possible until resumed. Can only suspend a server if it has no running enterprises.
- Resume. Bring server back online, re-enable configuration and use.
- Reboot. Reboot entire server from kernel level.
- Delete. Complete delete a server (archive data with time-stamp).
- Move. Batch move of all enterprises on a server to another server.
- Update. Update the EnterpriseLibre system software from the kernel up.
- Configuration. Change the settings for available RAM, CPU, storage, and categories.
- Categories. Edit any categories used to reserve the server for certain types of enterprises.
- Load Balancing. Edit the load-balancing parameters for that specific server to over-ride any system settings.

5.4 Venn Solutions

There is an interesting architecture layer that has recently emerged between the enterprise and system, at the domain name level, where enterprises can overlap. Just as open integration enables automated construction of whole Intranet enterprises at the click of a button, the open architecture of the Intranets themselves make possible assembly of new structures from multiple Intranets with an ease never possible or even imagined before.

For example, Yourdomain.com could create five enterprises, separate but with overlapping users – one for the full team, one for management, one for creative, one for finance, and one for the board of directors. The CEO could have three desktops – team, management, board – doing different work, using different document management systems, syncing files they wish with Syncthing. However, their email and calendars can be automatically kept in sync in all the desktops with the following configuration:

- External Servers. When enterprises are added to existing systems without any changes – the zero cost approach - all enterprises are set (with their Enterprise Manager) to point at the existing email server, and email for any user is kept automatically synced as usual by each enterprise. Logically, each virtual desktop is just a fancy email client for that user.

- Internal Servers. When the organization decides to use their enterprise's own name and email servers for better security and possibly cost, one must be primary, so EnterpriseLibre only allows one enterprise at a time within a given system have the authoritative name server for a domain name. Superusers for other enterprises at the same domain then set their name and email servers to point to the primary enterprise. As long as the primary enterprise has an account for every user, any number of separate and overlapping enterprises can be created in any combination however desired.[22]

The Venn solution capability become apparent with the most recent Enterprise Manager update, and has not been fully explored. However, it appears some (relatively) simple System Manager enhancements could add significant levels of compartmented security for organizations with separate but linked teams, with different levels of sharing of information. For example, medical organizations such as hospitals can have several departments that need to separately manage much of their information, share some information freely between departments, and block most communications externally for privacy reasons.

The open architecture and one-click automation enables easy experimentation with Venn solutions as the need and interest arise. Additional experience with an appropriately structured customer, such as medical group, distributed non-profit, government entity, or similar organization will help find the areas of greatest intersecting value.

[22] An enhancement is planned so users do not need an account with the primary enterprise, however can still have email at the same domain name, enabling completely non-overlapping enterprises.

6. Security Matters

> *I'm speaking to you from Silicon Valley, where some of the most prominent and successful companies have built their businesses by lulling their customers into complacency about their personal information.*
>
> – Tim Cook; Apple CEO; 2015-06-01.

This section provides the EnterpriseLibre security policy, describes the security architecture, provides an example of compliance for safeguarding Government of Canada PROTECTED B information, and references other standards it meets for securing electronic medical records.

6.1 Security Policy

The EnterpriseLibre security policy is to provide mobile access with all the security advantages of a centralized architecture, providing a dedicated, single-tenant solution safeguarded in one electronically and physically protected space, accessed with a single encrypted virtual interface, to enable certification and use for the most sensitive purposes.

6.2 Security Architecture

> *DoD security depends on OSS. Anyone can review it, including for the possibility of malicious code. Making source code available to the public significantly aids defenders.*
>
> – Chief Information Officer, US Department of Defense, 2015-08-29.

The EnterpriseLibre architecture builds on existing security knowledge and best practices by architecting an enterprise solution protected with seven layers:

1. Secure Building. All current Cirrus Computing data-centers are located in Canada, with the usual industrial-level security protections at the building level.

2. Secure Room. All computing infrastructure is in a secure room limited to authorized personnel by access controls such as card readers and audit logs.

3. Dedicated Hardware. All servers and storage are dedicated to Cirrus Computing customers, and under configuration management so security can be assured from the kernel on metal up. When large hosted servers are shared by more than one customer for efficiency and cost benefits, they maintain kernel-level separation. If desired, entire servers, cabinets, and even racks can be easily dedicated to a single customer.

4. Dedicated Software. When hosted on large, shared servers for efficiency and cost benefits, each customer still has dedicated software from **twelve** virtual servers up, including a couple dozen integrated components at the system and middleware layers, with their own dedicated applications on top.

5. Encryption. Virtual desktops provide access to the whole system so software and files don't need to be stored on local devices, and use strong encryption end-to-end.

6. Authentication. For stronger authentication, the Superuser can mandate two-password login, requiring entry of a random PIN after regular login. After login, single-sign-on provides secure access to multi-user applications within the desktops.

7. Open Source Software. EnterpriseLibre uses all open source that has undergone open review and security hardening, and to provide assurance there is no malware hidden in the binary.

With this architecture, all communications within a team – instant messages, email, even use of team applications like a Wiki – all happen within the enterprise, which means within a server **1.719"** tall. No matter where the users are, one on a virtual desktop in Montreal and another in Madrid, the information stays within that single server. That's pretty secure.

6.3 Government of Canada Security Example

This section provides an example of how the strengths of the EnterpriseLibre security architecture make it straightforward to certify, even in hosted mode, for a real-world requirement, electronic processing of Government of Canada PROTECTED B information, defined in the PWGSC Industrial Security Manual [MSI-ISM] as follows:

> *PROTECTED B: Information and assets that, when compromised, could reasonably be presumed to cause serious injury, i.e. social insurance number, criminal or financial information, medical information.*

6.3.1 Personal Information Protection & Electronic Documents Act

EnterpriseLibre meets the requirements of Canadian federal legislation, the *Personal Information Protection and Electronic Documents Act* [PIPEDA]. Schedule 1 provides most of the requirements, Section 4.3.4 defines medical information as sensitive, and Section 4.7 describes safeguards. EnterpriseLibre meets and exceeds the requirements of PIPEDA with an enterprise solution that can be readily accredited to safeguard PROTECTED B information.

In addition, within each single-tenant enterprise, higher levels of privacy at the data-level can be implemented by any customer using a full range of mandatory and discretionary access controls with applications. For example, many teams start by creating folders and documents in their document management application for control of access, change, and archiving by groups and individuals. Underneath all the multi-user applications there is the security of the twelve Linux servers, the Intranet foundation, and likely the PostgreSQL or MySQL database.

6.3.2 Operational Security Standard on Physical Security

The EnterpriseLibre solutions hosted by Cirrus Computing comply with the requirement of the Treasury Board of Canada standard *Operational Security Standard on Physical Security* [OPSPS] to safeguard information in a Security Zone:

> *Security Zone - is an area to which access is limited to authorized personnel and to authorized and properly-escorted visitors; it must be indicated by a recognizable perimeter and monitored continuously, i.e., 24 hours a day and 7 days a week. Example: an area where secret information is processed or stored.*

6.3.3 TBS Management of Information Technology Security

EnterpriseLibre complies with the Treasury Board of Canada Secretariat, *Management of Information Technology Security* [MITS] standard to encrypt PROTECTED B information in transmission:

> *... departments must encrypt protected B information before transmitting it across the Internet... Public Key Infrastructure (PKI) is one way that departments can fulfill requirements for authentication, confidentiality, integrity and non-repudiation.*

The X2Go software integrated into this release uses a **1024 bit** key by default. Longer keys and other algorithms are easily added. Each enterprise uses an OpenSSL RSA **2048 bit** SHA1 certificate for IMAP, SMTP, and related Internet communications.

User passwords can have up to **14** of the usual characters. High-frequency attacks are blocked by delays, so only human mistakes are possible. The Superuser can mandate stronger authentication with a Two-Password requirement for all users, or individual accounts.

6.3.4 Physical Protection of Computer Servers

EnterpriseLibre solutions hosted by Cirrus Computing comply with the RCMP standard *Physical Protection of Computer Servers* [G1-031] for systems to be in a locked sever room:

> *Physical Security Requirements for Server Rooms (Protected B): Locked server room OR Lock up the servers located in an Operations Zone or higher OR Secure server room.*
>
> *Locate the server in a separate room and control access to this room. Limit access to those individuals having an operational or job-related need; The room should be built with walls that extend from the floor slab to the underside of the floor/roof slab above; ... Control of access can be achieved though a variety of means, such as mechanical keyed locks or electronic card readers.*

EnterpriseLibre also satisfies the higher-level requirement for a "Secure Server Room". Additional safeguards can be easily provided with locked cabinets and customer controlled keys, standard availability in most data-centers. Even stronger protections, such as storage encryption and similar measures, can be easily added at the cabinet level to provide almost any level of assurance needed, even when hosted in a third-party data-center, an EnterpriseLibre capability unique among all current hosted systems.

6.3.5 Approved Cryptographic Algorithms

EnterpriseLibre complies with the requirement of the Canadian Security Communications Commission standard *Approved Cryptographic Algorithms* [ITSA-11E] to use secure encryption:

> "*CSEC approves the use of the following algorithms for the encryption of Protected information with the limitations outlined below: AES (128, 192, 256 bits);*
>
> *CSEC approves the use of the following algorithms for the establishment of encryption keys: RSA (Rivest, Shamir, Adleman), other algorithms based on exponentiation in finite fields (e.g., Diffie-Hellman, MQV).*"

See *Section 6.3.3* for compliance information.

6.4 Medical Systems Security Compliance

EnterpriseLibre is also compliant with the requirements for safeguarding the privacy of health information, such as electronic medical records, listed in the following sections:

A.1 Security – Electronic Medical Records.

A.3 Security – Government of Ontario Health Information.

7. Traceability Cross-References

> *A tapestry to feel and see, impossible to hold.* – Carole King, Tapestry, 1971.

This section links together the rest of the document, tracing how the requirements, blocks, and architecture principles inter-relate. The fourth section compares requirements compliance with alternative solutions.

7.1 Requirements To Architecture Principles

The following table traces customer requirements to supporting architecture principles, summarizing how EnterpriseLibre meets the users needs. More information is provided in *Section 3 – Customer Requirements* and *Section 4 – Architecture Principles*.

Requirement \ Principle		Intranet Foundation	Open Integration	Virtual Desktops	Optimum Efficiency
	64	*14*	*24*	*17*	*9*
Usability	*15*	*3*	*5*	*5*	*2*
• Ease Of Use	*8*	*2*	*3*	*3*	*0*
– Virtual Desktop	*3*	+	+	+	
– Application Simplicity	*2*		+	+	
– Single-Sign-On	*3*	+	+	+	
• No Maintenance	*7*	*1*	*2*	*2*	*2*
– Full-Time Availability	*4*	+	+	+	+
– Hosted Convenience	*3*		+	+	+
Capability	*13*	*4*	*6*	*2*	*1*
• Full Functionality	*7*	*1*	*3*	*2*	*1*
– Integrated Solutions	*4*	+	+	+	+
– Best Components	*1*		+		
– Any Application	*2*		+	+	
• One-Click Management	*6*	*3*	*3*	*0*	*0*
– Enterprise Automation	*2*	+	+		
– User Automation	*2*	+	+		
– Scalability Automation	*2*	+	+		
Security	*24*	*5*	*9*	*6*	*4*
• Confidentiality & Integrity	*9*	*3*	*3*	*2*	*1*
– Single Tenant	*3*	+	+		+
– Secure Network	*2*	+	+		
– Secure Channels	*2*		+	+	
– Physical Protection	*2*	+		+	

Requirement \ Principle		Intranet Foundation	Open Integration	Virtual Desktops	Optimum Efficiency
• Mobile Privacy	7	1	3	3	0
– End-To-End Encryption	2		+	+	
– Single-Sign-On	3	+	+	+	
– Two-Password Login	2		+	+	
• Certifiable Assurance	8	1	3	1	3
– Proven Design	4	+	+	+	+
– Hardened Components	2		+		+
– Open Source Software	2		+		+
Sustainability	12	2	4	4	2
• Long-Term Use	6	1	3	1	1
– Open Source Software	2		+		+
– Onsite Or Hosted	2		+	+	
– Natural Evolution	2	+	+		
• Green I.T.	6	1	1	3	1
– Energy Efficiency	4	+	+	+	+
– Computer Recycling	1			+	
– Carbon Offsets	1			+	

Table 2-1: Trace: Requirements To Principles

7.2 Blocks To Requirements

Nine blocks stand in the way of the requirements, four more than once. The following table traces blocks to requirements, listed in order of largest number of impacts. More information is provided in *Section 3 – Customer Requirements*.

Block \ Requirement		Usability		Capability		Security			Sustainability	
		Ease Of Use	No Maint.	Full Soln's	1-Click Mgmt.	Conf. & Integ.	Mobile Privacy	Cert. Assur.	Long-Term	Green I.T.
	25	3	4	2	2	3	2	3	3	3
Proprietary Problems	7	+	+	+	+			+	+	+
Web 2.0 Diversion	7	+		+	+	+	+	+		+
Local Infrastructure	4		+			+			+	+
Custom Software	2		+					+		
Hosted Lock-In	1								+	
FLOSS Skepticism	1		+							
Multi-tenant Illusion	1					+				
Network Vulnerability	1						+			
Virtual Desktop Stasis	1	+								

Table 2-2: Trace: Blocks To Requirements

A weighted impact by magnitude could tune this analysis. For example, while Hosted Lock-In impacts only one requirement, it's also absolute, so should always be avoided. However, this analysis by breadth alone is quite informative: the two blocks with by far the greatest impact are Proprietary Problems and the great Web 2.0 Diversion.

7.3 Blocks To Architecture Principles

The following table traces requirements blocks to the architecture principles that resolve them, listed in the same order of impact as in the *Blocks to Requirements* trace table. Since the resolution here is straight-forward, a brief description is provided within the table, leaving *Section 4 – Architecture Principles* to focus on meeting the requirements themselves.

Block \ Principle		Intranet Foundation	Open Integration	Virtual Desktops	Optimum Efficiency
	19	*4*	*4*	*6*	*5*
Proprietary Problems	*2*		Reduces *O(N^2)* to ~ *O(N^1.1)*		Use more efficient FLOSS
Web 2.0 Diversion	*3*	Provides real solution		One desktop, full applications	Need central, single-tenant
Local Infrastructure	*2*			On-site capability when hosted	Hosted ready, no HW lock-in
Custom Software	*3*	Good SW ready for every need	Proven FLOSS for everything		Use tuned, standard FLOSS
Hosted Lock-In	*2*		Onsite ready, hosted portable	Onsite capable, hosted portable	
FLOSS Skepticism	*3*	Standard, all FLOSS design	EnterpriseLibre all FLOSS	Global access to any FLOSS	
Multi-tenant Illusion	*2*	Dedicated for each customer			Dedicated performance
Network Vulnerability	*1*			One encrypted channel	
Virtual Desktop Stasis	*1*			Multi-user FLOSS Intranet	

Table 2-3: Trace: Blocks To Principles

7.4 Requirements To Alternative Solutions

The following table shows how leading alternatives to EnterpriseLibre meet the requirements of a typical small to medium size organization. These are not equal alternatives, since no proprietary solution can approach the usability, capability, security, and sustainability of a complete open source software Intranet enterprise. But they are the best available.

Requirement \ Solution	Enterprise Libre	Microsoft Onsite	Microsoft Online	Google Online	Web 2.0: SalesForce, Netsuite, Zoho, etc.
	27	*12*	*11*	*11*	*4*

Requirement \ Solution	Enterprise Libre	Microsoft Onsite	Microsoft Online	Google Online	Web 2.0: SalesForce, Netsuite, Zoho, etc.
Usability	*5*	*2*	*4*	*4*	*2*
• Ease Of Use	*3*	*2*	*2*	*2*	*0*
– Virtual Desktop	Y				
– Application Simplicity	Y	Y	Y	Y	
– Single-Sign-On	Y	Y	Y	Y	
• No Maintenance	*2*		*2*	*2*	*2*
– Full-Time Availability	Y		Y	Y	Y
– Hosted Convenience	Y		Y	Y	Y
Capability	*6*	*1*	*1*	*1*	*0*
• Full Functionality	*3*	*1*	*1*	*1*	*0*
– Integrated Solutions	Y	Y	Y	Y	
– Best Components	Y				
– Any Application	Y				
• One-Click Management	*3*	*0*	*0*	*0*	*0*
– Enterprise Automation	Y				
– User Automation	Y				
– Scalability Automation	Y				
Security	*10*	*8*	*4*	*4*	*1*
• Confidentiality & Integrity	*4*	*3*	*0*	*0*	*0*
– Single Tenant	Y	Y			
– Secure Network	Y	Y			
– Secure Channels	Y				
– Physical Protection	Y	Y			
• Mobile Privacy	*3*	*3*	*3*	*3*	*0*
– End-To-End Encryption	Y	Y	Y	Y	Y
– Single-Sign-On	Y	Y	Y	Y	
– Two-Password Login	Y	Y	Y	Y	
• Certifiable Assurance	*3*	*2*	*1*	*1*	*1*
– Proven Design	Y	Y			
– Components	Y	Y	Y	Y	Y
– Open Source Software	Y				
Sustainability	*6*	*1*	*2*	*2*	*1*
• Long-Term Use	*3*	*1*	*1*	*0*	*0*
– Open Source Software	Y				
– Onsite Or Hosted	Y	Y	Y		
– Natural Evolution	Y				
• Green I.T.	*3*	*0*	*1*	*2*	*1*

Requirement \ Solution	Enterprise Libre	Microsoft Onsite	Microsoft Online	Google Online	Web 2.0: SalesForce, Netsuite, Zoho, etc.
– Energy Efficiency	Y		Y	Y	
– Computer Recycling	Y			Y	Y
– Carbon Offsets	Y				

Appendix A - References

This section collects references made in the document.

A.1 Security: Electronic Medical Records

Standards for securing electronically processed medical records:

- College of Physicians and Surgeons of Ontario; *Medical Records*; CPSO.
- Canadian Medical Protective Association; *Electronic Records Handbook*; CMPA.
- Ontario MD; *Vendor Collaborative Networks*; OMD.

A.2 Security: Government of Canada

Canadian Government standards for securing medical and PROTECTED B information:

- Health Canada; *Contractor Security Requirements*; TPD/BGTD FAQ.
- GOC; *Personal Information Protection and Electronic Documents Act*; PIPEDA.
- RCMP; *Physical Protection of Computer Servers*; G1-031.
- PWGSC; *Industrial Security Program*; ISP.
- CSEC; *Approved Cryptographic Algorithms*; ITSA-11E.
- Treasury Board Secretariat ; *Management of Information Technology Security*; MITS.
- Treasury Board Secretariat; *Operational Security Standard on Physical Security*; OPSPS.
- Public Works and Government Services; *Industrial Security Manual*; MSI-ISM.

A.3 Security: Government of Ontario Health Information

Ontario Government requirements for protecting the privacy of health information:

- GOO; *Personal Health Information Protection Act*; PHIPA.
  (Also see: GOC; *Declaration of PHIPA as substantially similar to PIPEDA*; PHIPIP.)

A.4 Software Sources

The main sources surveyed for FLOSS to integrate into EnterpriseLibre:

- Apache Software Foundation – Started with the Apache web server, and now supports several foundation FLOSS components.
- GitHub - Provides GIT configuration management for increasing amounts of FLOSS.
- Mozilla.org – Supports the Firefox web browser and Thunderbird email client.

- The Document Foundation – Supports the standard FLOSS office suite LibreOffice, tracing roots through OpenOffice to StarWriter in 1985.
- Savannah – Home for development, distribution, and maintenance of GNU software.
- SourceForge.net – Long-time free hosting for FLOSS.
- Openhub – Formerly Ohloh, has a good FLOSS rating system and other metrics.

Surveyed secondary sources:

- CPAN -- Directory of free Perl scripts.
- IceWALKERS -- Linux software.
- NASA / Catalog – Several different types of FLOSS.
- Tigris.org – FLOSS software engineering tools.
- Med FLOSS – Compilation of open source software for health care.

Surveyed directories:

- DMOZ – Open Source
- EU Open Source Software Inventory
- FSF/UNESCO Free Software Directory
- Yahoo – Computers_and_Internet/Software/Open_Source/

Appendix B - Software Development Environment

The software is maintained with a simple set of FLOSS, summarized below.

- EnterpriseLibre. Cirrus Computing has always used an EnterpriseLibre solution to develop the solution, especially useful for a distributed team. Secure access to the rest of the development environment is easily provided from within the virtual desktops.

- Languages. Most of the EnterpriseLibre software in the System Manager and Enterprise Manager is written in Perl, bash, and python. No standardization or other restrictions limit the code used, with best tool for the purpose chosen as required.

- Configuration Management. The code is managed in the distributed revision control system Git, developed by L. Torvalds, and particularly strong for distributed development.

- Documentation. Documents like architecture diagrams, user interface designs, enhancement list, and others are mostly developed in LibreOffice documents, and archived and version controlled by the Nuxeo document management system. Outside of the code, software design and other information is mostly managed in the Cirrus Computing Wiki, using MediaWiki, providing exceptional ease of use and highly used.

- Support Tickets. The standard Trac software is used to document support tickets, a simple system for targeted purposes.

- Mailing List. The standard GNU Mailman software, developed by R. Stallman, is used for the Support mailing list (and each solution's mailing list software).

Appendix C - Software Listing

The first table lists the software in EnterpriseLibre, ordered by layer as described in *Section 5 – Solution Design*. Potential additional categories and FLOSS options are provided in a follow-on table. References reviewed in selection of these components are listed in *A.4 – Software Sources*. All the EnterpriseLibre software can be downloaded from the CirrusOpen.org Wiki.

Category	Component	Software	Notes
Desktops	Desktop	Ubuntu	Standard FLOSS desktop.
Applications: Individual	Browser	Firefox	Chrome to be added.
	Email / Calendar / Contacts	Evolution	Standard FLOSS. Thunderbird to be added.
	File Sync	Syncthing	Standard FLOSS.
	Graphics Bitmap	Gimp	Standard FLOSS.
	Graphics Publishing	Scribus	Standard FLOSS.
	Graphics Vector	InkScape	Standard FLOSS.
	Messaging	Pidgin	Standard FLOSS.
	Mind Mapping	FreeMind	Standard FLOSS. Newer FreePlane to be added.
	Office Suite	LibreOffice	Standard FLOSS.
	Optical Character Recognition	Tesseract	Standard FLOSS.
	Projects	ProjectLibre	Standard FLOSS.
	Visualization	VUE	Standard FLOSS.
Applications: Segments	Education	Moodle	Standard FLOSS.
	Manufacturing	Oodo	Renamed OpenERP.
	Medical	OSCAR	Many good FLOSS medical & dental apps to be added.
	Non-Profit	CiviCRM, ChurchInfo	Standard FLOSS.
Applications: Multi-user	Accounting	SQL-Ledger	Standard FLOSS.
	Customer Relationship Management	vTiger	Standard FLOSS. SugarCRM to be added.
	Document Management	Nuxeo	Standard FLOSS. Alfresco to be added.
	HR Management	OrangeHRM	Standard FLOSS.
	Issue Management	Trac	Standard FLOSS (outside SW development).
	Mailing Lists	GNU Mailman	Standard FLOSS.
	Project Management	RedMine	Standard FLOSS.
	Resource	Booked	Standard FLOSS.

Category	Component	Software	Notes
	Management		
	Timesheets	TimesheetNG	Standard FLOSS.
	Wiki	MediaWiki	To be added: DocuWiki, Foswiki, Tiki, TWiki.
System: Groupware	Email IMAP	Dovecot	Standard FLOSS.
	Email SMTP	Postfix	Standard FLOSS.
	SPAM Control	DSPAM	Standard FLOSS.
	Virus Control	ClamAV	Standard FLOSS.
	Calendars & Contacts	SOGo	Standard FLOSS.
	Shared Folder	Linux	Standard FLOSS.
System: Middleware	Apps Server Web	Apache	Standard FLOSS.
	Apps Server Java	JBoss	Standard FLOSS.
	Databases	MySQL, PostgreSQL	Standard FLOSS.
	Languages	Ajax, Perl, PHP, Python, Ruby	Standard FLOSS.
	NX Server	X2Go	Other FLOSS: FreeNX, neatx, Remmina, SPICE, x2go. Also originator, closed in V4, NoMachine.
System: Intranet	Firewall	Shorewall	Standard FLOSS.
	DNS Server	BIND	Standard FLOSS.
	Network	OpenVPN	Standard FLOSS.
	Security	OpenSSL, OpenSSH	Standard FLOSS.
	Authentication	Kerberos, PAM	Standard FLOSS.
	User Directory	OpenLDAP	Standard FLOSS.
EnterpriseLibre	Enterprise Manager	EnterpriseLibre	Full FLOSS Intranet.
	Health Monitoring	Nagios	Standard FLOSS.
	Backup & Restore	Duplicity	Standard FLOSS.
Operating System	Virtualization	OpenVZ	Most scalable and secure container solution.
	Linux	Ubuntu	Ubuntu, server compatibility with desktop.

Table B-1: Software Listing

The following table lists categories for potential addition to EnterpriseLibre where good FLOSS is known to exist. All options are draft only, not a complete list.

Category	Component	Draft Options	Notes
Applications: Segments	Analytics	Hadoop, Spark, Storm	Where the need is not large, e.g. Cirrus Computing received a query, could not pursue at the time, to provide up to 200 desktops with local Hadoops on demand for training.

Category	Component	Draft Options	Notes
	Business Intelligence	Pentaho	Could have multi-app bundles with some useful automated integration.
	Ecommerce	OFBiz	Could have useful automated integration, e.g. to Shopify, Drupal, WordPress, etc.
	Emergency Management	OpenRelief, TicketsCAD, Vesuvius	EnterpriseLibre helps as a solution with rapid scalability, mobile accessibility.
	Engineering	BRL-CAD, Code Aster, LibreCAD	Many other good components can be added for specific kinds of engineering. This category is so large it was left to an enhancement.
	Libraries	Evergreen, Koha	Standard FLOSS.
	Science	GNU Octave, SageMatch, SciLab	Standard FLOSS. Many other good options for specific branches of science, a large enough category to be considered as an enhancement.
Applications: Multi-user	Meetings	Openmeetings	Standard FLOSS.
	Web Conferencing	BigBlueButton	Standard FLOSS.
	Workflow	Bonita BPM, ProcessMaker	Standard FLOSS.
System: Groupware	MS Exchange Bridge	Davmail	Standard FLOSS. Installed for a customer using MS Exchange OWA, works very well, to added in a near-term enhancement.
System: Middleware	Fax System	HylaFax	Standard FLOSS. There is still some need for a fax system, routing to and from desktops.
	VOIP System	Asterisk	Standard FLOSS. Good opportunity for useful automated integration.

Table B-2: Software Listing (Options)

Appendix D - Software Team

Cirrus Computing was formed in 2008 by a group of software engineers in Ottawa, Canada to bootstrap a better FLOSS solution, with help over the years from the following team.

Integration Lead – Nimesh Jethwa

At university, Nimesh moved virtual servers between subnets for fun, then built the first full Enterprise Manager, with one-click configuration of domains and email servers, applications, and backup, then reused the code to build the System Manager by induction.

Email & Enterprise Manager – Lukas Kamps

Lukas helped integrate the first email system, built the first Enterprise Manager GUI for automated user creation, helped integrate the Intranet databases, and provided the most important resources for early meetings – a kitchen table and wireless connection.

Java & Website – Sergei Konov

Juggling Java, language and system, Sergei built our first secure enterprise desktop login system, linking website authentication to an easy pre-configured virtual desktop start link, downloadable with or without password. He did most of his programming while riding a bike.

OS & Server – Richard Leir

Rick improved the performance and reliability of almost all the software in EnterpriseLibre, from server configuration to life-cycle solution and user management, and led stress testing and hardening of the multi-server creation load-balancing that enables hoster scalability.

OS & Virtualization – Karoly Molnar

Karoly designed the EnterpriseLibre near optimally efficient kernel virtualization architecture, then used it to build the multi-server, single-tenant Intranet, led secure integration of the virtual desktop, and helped build the network, email, and life-cycle desktop automation.

System Architect – William Stewart

Bill architected EnterpriseLibre to use FLOSS to provide integrated solutions without lock-in or *O(N^2)* barriers, with virtual desktops so a hosted version could provide almost any FLOSS to anyone world-wide, with the same usability, capability, and security as an onsite Intranet.

Software & Automation – Greg Wolgemuth

Greg helped build EnterpriseLibre while going to university on the side, leading the hosted automated creation, life-cycle virtual desktop automation, single-sign-on integration, and helping with almost every part of the first complete Intranet and management solution.