Libre Services:
A non-proprietary model for delivery of Internet services

Document # PLPC-100101
Version 1.0
March 26, 2006

Available on-line at:
http://www.freeprotocols.org/PLPC/100101

Mohsen Banan
http://mohsen.banan.1.byname.net/ContactMe

Andrew Hammoude
http://andrew.hammoude.1.byname.net/ContactMe

Free Protocols Foundation
3610 164th Place SE
Bellevue, WA 98008-5807
Phone: (425) 644-8026
Web: http://www.freeprotocols.org

A component of the *Libre Service Manifesto*

This article is one of a broader series articles that we refer to collectively as the *Libre Services Manifesto*. Together these articles describe every aspect of the Libre Services model.

The complete collection of articles is available at:
http://www.FreeProtocols.org/PLPC/100105

Contents

List of Figures

Executive Summary

The Internet has created an enormous new offshoot of the software industry: the **Internet services** industry. This industry has become a key medium, not just for day-to-day communications and productivity, but also for the expression of information and ideas.

However, this vitally important new industry exists entirely in the form of the traditional proprietary software model. Within the general software arena, the free software movement is well established as an alternative to proprietary software. But as yet, the free software movement has no formal presence within the Internet services domain.

We are a group of engineers with a vision for the future of Internet services. We believe that the free software movement as we see it today is just the beginning, and the next major evolutionary phase of free software is its strong emergence into the Internet services arena.

We believe that the intellectual property ownership mechanisms of patents, copyright and trade secrecy, as they exist today, have almost no legitimacy at all within this arena. At bottom these ownership mechanisms are business constructs, intended to provide competitive advantage in a commercial context. That they do. But they do so at great cost to the broader society. Some of the societal costs are obvious; others are more subtle and indirect. But the costs are real, and very far-reaching. They include:

- A crippling of the software engineering profession. These ownership mechanisms cut directly across the engineering *freedom of action* that is the foundation of the software development process.
- The loss to the public of the technical benefits of unrestricted engineering development.
- In the case of Internet services based on commercial providers, the compromising of a number of important civil liberties, including personal privacy, freedom of information, and freedom of speech.
- A severe distortion of the competitive business environment.
- Eventual loss of governance of the Internet to purely commercial interests.

Instead of the proprietary software model, we are advocates of the free software movement, in which software is treated as a communal resource, freely available for reuse by anyone. Our ultimate vision is a completely open software industry, in which *all computing and communications is based entirely on free software*.

We are proposing a radically new, completely non-proprietary model for the delivery of Internet services. We call this the **Libre Services** model.

Libre Services are an extension of the principles of free software into the Internet services domain. They are Internet services that may be freely copied and reused by anyone. They are a communal resource, not owned by anyone, freely available for use by society at large. Any company, organization or individual can reproduce and host any Libre Service, either for their own use, or for commercial or non-commercial delivery to others. The Libre Services model exists in relationship to the proprietary Internet services model of AOL, MSN and Yahoo, in an analogous way to how GNU/Linux exists in relation to Microsoft Windows.

This is a radical departure from the existing commercial model, with societal benefits that are equally radical and far-reaching. The Libre Services model provides a range of critical *freedoms* that are entirely absent from the proprietary model:

- The *freedom* of the engineering community to engage in unrestricted creative development, building new and better Internet services for the benefit of the public.
- The *freedom* of any group or community to operate their own Libre Services, according to whatever principles they see fit. Since they are no longer subject to the actions of a commercial service provider, this guarantees a range of critical civil liberties: privacy, protection against government monitoring, freedom of information, freedom of ideas, freedom of speech.
- The *freedom* of the business community to participate in the Internet services industry, without any intellectual property barriers standing in the way. Libre Services transform the closed industry of today into a truly open industry, creating major new business opportunities and industry growth.

Libre Services are the right way to deliver Internet services to the user. Our goal is to establish Libre Services as a non-proprietary alternative to the existing proprietary services industry.

In this article we describe the Libre Services concept, and how we intend to turn it into a reality. A key component of our bootstrapping strategy is a **project-based** model for collaborative participation. We have defined a set of independent, self-contained projects required to move this initiative forward. This allows efficient, coordinated collaboration on multiple bootstrapping tasks in parallel.

1 Introduction

1.1 The software development process

Software development is an inherently cumulative and collaborative process. It is cumulative in the sense that new software is created by assembling existing software constructs into ever more complex and powerful constructs. And it is collaborative in the sense that it is readily amenable to joint, collective development by large numbers of organizations and individuals.

For this reason software has a unique capability to undergo rapid and complex evolutionary growth. This is nowhere better demonstrated than by the extraordinary growth and vitality of the Internet itself.

However, this evolutionary capability depends critically on *freedom of action*. It depends on the freedom of the software engineering community to reuse existing software constructs, and engage in collaborative development, without restriction. Any restrictions placed on this freedom inhibit the growth potential of the software and Internet industries.

By contrast, the conventional business model is based on asset ownership, and denial of that ownership to competing companies. In the case of the software industry, asset ownership is effected by means of a proprietary software model, in which software ownership is maintained by means of patents, copyright, and trade secrecy.

However, these ownership mechanisms cut directly across the essential *freedom of action* that gives software its unique evolutionary capabilities. All three mechanisms explicitly deny access to existing software constructs. They prevent software reuse and collaborative development, and therefore inhibit the natural software development process.

The proprietary software model is in fundamental conflict with the nature of software itself.

1.2 The free software movement

Twenty years ago Richard Stallman understood this very well, and he and others formulated the principles of **free software**, a completely non-proprietary software model [1]. Under this model software is a communal resource, freely available to the entire software development community without any restrictive ownership mechanisms.[1]

In 1985 Stallman and others founded the Free Software Foundation [2], an organization dedicated to the promotion of free software. They did the necessary intellectual work to formalize the principles of free software, created written materials to define and promote the free software concept, and established a framework for collaborative development of free software projects.

This early work led eventually to the creation of GNU/Linux, the foundational software for the entire free software movement [3].

Twenty years later, this movement is mature and robust. It is fully proven as a viable development model

[1]Throughout this document whenever we say "free software" we are referring to *freedom of action*, not *zero monetary cost*. Or to use the stock clarification: we mean free as in "free speech" not "free beer."

that can equal or exceed the capabilities of the proprietary model.

1.3 Our philosophy

We believe that the intellectual property ownership mechanisms of patents, copyright and trade secrecy, as they exist today, have virtually no legitimacy at all within the digital domain.

Many of our laws and practices serve to balance rights between potentially conflicting constituencies. As originally conceived, and as practiced within the material domain, these intellectual property mechanisms may well serve this purpose. But within the digital domain, they do not.

These ownership mechanisms confer unquestionable competitive advantage on their owner. But they do so at unacceptable cost to society at large. Patents, copyright and trade secrecy explicitly prevent the cumulative and collaborative development processes that give software its unique potential. As a result society is denied the full realization of this potential.

We believe that the proprietary model is the wrong basis for the software industry. Instead, we are advocates of the free software model, in which software is treated as a communal resource, subject to complete freedom of action by anyone.

The new conventions of non-material capitalism

We further believe that the free software movement as we see it today is only the beginning. It is the first manifestation of a much bigger cultural shift: a shedding of the traditional conventions of material capitalism, and the adoption of a new set of conventions based on non-material capitalism. Western capitalistic societies are rooted in the historical conventions and institutions of material products and materially-based services. In the digital domain these conventions appear in the form of the proprietary software model.

But in the non-material world, there is a better way of doing things. The power of free software derives from a relinquishing of the traditional intellectual property conventions. Instead, free software is based on a set of principles that allow powerful generative forces to come into play. Thus traditional copyright is rewritten in the form of copyleft; ownership of software via patents is relinquished in favor of patent-free protocols and software; self-interested software hoarding via trade secrecy is relinquished in favor of a convention of openness and sharing.

The result is a culture of creative freedom and collaboration, based on collective pooling of resources. Twenty years after the fact the premise appears very simple: in the digital domain there is more to be gained by collective pooling than by individual ownership.

We believe that these principles apply, not just in the digital domain, but throughout the non-material domain in general. We believe that these principles have equivalent power and can bring equivalent benefits in many fields throughout the sciences and humanities. We invite other professions to look critically at the free software movement, and consider applying its principles to their own field of endeavor.

But one thing at a time. The next natural extension of free software is its extension into the domain of Internet services.

2 The Subscriber Services industry

The Internet has created an enormous new offshoot of the software industry: the **Internet services** industry[2]. By Internet services, or Subscriber Services, we mean *any service that is provided to a user via the Internet*. Some examples are:

- **Communications**. E-mail, mobile messaging, instant messaging, discussion groups, fax delivery, phone service (VOIP), voice messaging, web-based address book, web-based calendar, etc.
- **Web presence**. Personal website, blog, photo gallery, video gallery, etc.
- **Information delivery**. News, weather, stock reports, traffic reports, maps, directions, images, directories, catalogs, etc.
- **Search**. Google, AltaVista, Ask Jeeves, etc.
- **Transactions**. Services in which third parties are matched up for some type of transaction. Job listings, housing listings, buying/selling, auctions, personal ads, ridesharing, etc.
- **Business services**. Services provided over the Internet to businesses, such as those provided by Salesforce.com etc.

And many other types of service. The above list is far from complete, nor is our simple categorization of services in any way definitive. Subscriber Services is a new industry, not yet 20 years old, and still undergoing a process of disorganized self-definition. Thus far evolution of the industry has been driven by a multitude of *ad hoc* commercial initiatives, and it remains in a state of cheerful chaos.

Nevertheless, the scope of what we are describing is extremely large. In effect, it is the entire global Internet itself.

2.1 The industry today

Though still undergoing chaotic evolution, the Subscriber Services industry is well established. In 2006, generalized Internet services are provided by several large providers such as AOL, MSN and Yahoo. These major providers deliver a broad range of services to their subscribers, including most of the examples given above. Google is also aggressively moving into the generalized services arena, rapidly augmenting its core search service with a variety of additional services.

In addition to the major providers of generalized services, there is a wide variety of providers of specialized services, such as classified advertising (Craigslist), auctions (Ebay), airfare and vacation booking

[2]Note on terminology: the term “Internet services” is potentially ambiguous. The term “Internet Service Provider” (ISP) is commonly used to mean a company that provides basic Internet access, i.e. connectivity at Layer 3 in the OSI model. The term “Application Service Provider” (ASP) is used to mean a company that delivers end-user services such as e-mail, i.e. functionality at Layer 7 in the OSI model. Throughout this document when we say “Internet services” or “Subscriber Services” we mean services at application level.

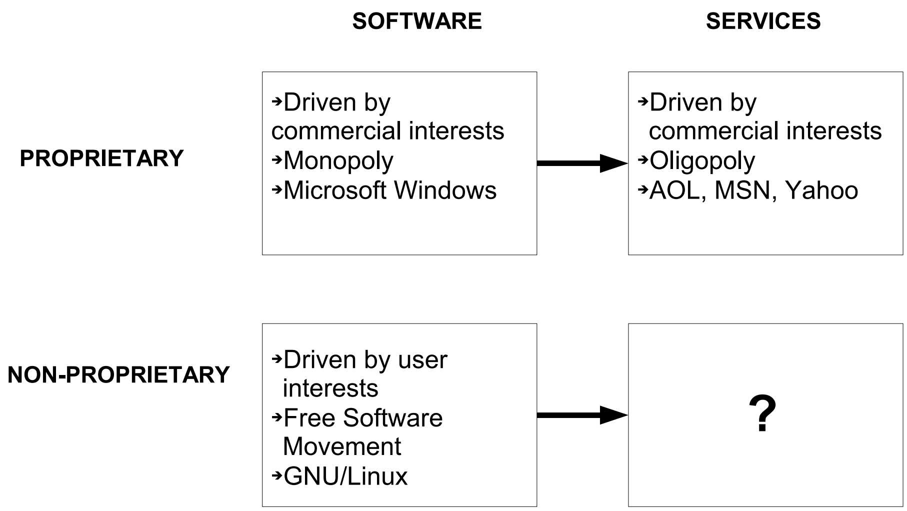

Figure 1: Proprietary and non-propietary presence

(Expedia), job listings (Monster.com), dating (Match.com), car trading (AutoTrader.com), and numerous others. We can expect that over time the large general service providers will provide many of these specialized services too, so that the industry will consolidate into a small number of dominant providers.

In addition to the variety of new services that the Internet has enabled, a fundamental change is occurring in the way traditional software applications are being provided to the user. Before the appearance of the Internet services industry, software applications were either run locally, on the user's own PC, or perhaps on a remote server on a local area network.

But now traditional software applications are migrating towards a service-based implementation. Many user applications that hitherto have been implemented as local stand-alone applications, are now being implemented as Internet services, in which the user is provided with the same or similar functionality via the Internet.

This represents a fundamental shift in the focus of the software industry. The focus is moving away from *software as a product*, and towards *functionality as a service*. This trend is just now becoming widely recognized within the industry, and there remains confusion about its exact nature and implications. This general confusion is reflected in the multiplicity of terms used to refer to this trend, such as "software as a service," "information technology as a service," and "transformation of software into services."

2.2 Domination of the proprietary model

All these developments—both the evolution of the Internet services industry, and the migration of traditional software into a service-based implementation—are taking place almost entirely in a proprietary context. Even Google, despite any pretensions it may have to the moral high ground, is based on proprietary software, heavily defended by patents, copyright and trade secrecy.

Figure 1 shows the presence of the proprietary and non-proprietary models in both the software and services arenas. Within the general software arena, the non-proprietary model has been fully formalized in the form of the free software movement. The principles of free software have been clearly articulated, and formally codified in the form of the GPL and other open-source licenses.

In addition, the Free Software Foundation and other organizations exist to provide leadership and advocacy, and to serve as rallying points for participation. As a result of all this, the free software movement exists as a viable alternative to the proprietary model.

However, a corresponding set of non-proprietary formalizations does not exist within the services domain. Today, virtually all Internet services are provided under the traditional proprietary software model, and as yet the free software movement has no formal presence in this domain. A set of defining principles has not been established, nor is there any leadership or rallying point for participation in the services domain, corresponding to the leadership of the Free Software Foundation in the software domain.

As a result of this there does not exist a non-proprietary alternative to the proprietary services model.

2.3 The problem: Governance by commercial interests

Meanwhile the proprietary services industry continues to grow rapidly.

It also continues to undergo a process of consolidation. In the services arena, as in the general software arena, there are strong forces of convergence towards a small number of dominant providers, and eventually a monopoly. In the more mature software industry this has resulted in the Microsoft Windows monopoly, now with no proprietary competition at all.

The services industry is already undergoing a similar process of convergence, and this may continue until there is a monopoly in this arena too. Microsoft is a particular concern in this regard. With its dominant position in both user environments (in the form of Windows), and services (in the form of MSN.com), Microsoft can create a level of integration between its own proprietary user environment and its own proprietary service that cannot be matched by any of the other major players. Based on its operating system monopoly it can exercise dominance of the services arena in a way that the other providers cannot, and can eventually achieve a monopoly position there too.

This trend towards a commercial oligopoly or monopoly presents some societal concerns.

The Internet is a global public resource, and as such requires representation and advocacy for the public interest. Such representation includes maintaining technical standards and protocols, protecting civil liberties, and preventing unfair business practices.

Standards and protocols

In the technical arena of standards and protocols, public representation has historically been provided by standards organizations within the Internet industry, such as the Internet Engineering Task Force (IETF). But service providers are under no obligation to use public protocols, and are free to deliver services using their own proprietary protocols. And if a commercial provider is large enough, in terms of number of subscribers, it becomes a *de facto* authority within the industry, able to establish its own protocols as an industry standard.

The history of Instant Messaging is a good example. Clearly the right answer for Instant Messaging is a set of non-proprietary public protocols, allowing global interoperability and open participation by all service providers.

But this is not what happened. Even though IRC (Internet Relay Chat) already existed as an open protocol for Instant Messaging, the major providers disregarded this entirely. Instead, each defined its own set of proprietary protocols, in an attempt to take complete ownership of Instant Messaging functionality and deny this functionality to competing providers. The result was that each provider created its own island of subscribers, unable to communicate with the subscribers on the competing islands.

Only when the resulting dysfunctionality became unacceptable to their subscribers did the major providers address the issue. But not by adopting an open protocol—rather, by implementing interoperability gateways amongst one another. The major providers now have cross connectivity for Instant Messaging, but the underlying protocols remain proprietary, and smaller providers are marginalized and at a competitive disadvantage. In effect the major providers have created a cartel to take complete ownership of Instant Messaging.

The presence of the free software movement in Instant Messaging remains limited to a small island of IRC users, who are able to communicate with the proprietary subscribers via interoperability gateways provided by the proprietary service providers. The major providers thus continue to control the Instant Messaging arena, with the free software community provided for as a courtesy.

Thus in the technical arena such public representation as exists is ineffective. The large commercial providers are able to establish their own standards and protocols, and engineering standards organizations such as the IETF have become largely irrelevant.

Civil liberties

In the area of civil liberties there is no formal public representation at all, and protection of civil liberties remains entirely in the hands of the commercial service providers.

If the commercial providers could be relied upon to act in the public interest, this would be of no great concern. But such is not the case. The commercial providers are under no obligation to protect the public welfare. Their sole mandate is to pursue their own commercial interests, and these may be highly detrimental to the broader interests of society.

Two recent events demonstrate this very well. In the first event, Microsoft recently shut down the Internet journal of a Chinese dissident, who had been expressing political views that his government found objectionable. In the second event, Google recently agreed to censor its search results in China, expunging web content that the government considers objectionable. Google will base its censorship decisions on guidance provided by Chinese government officials. So much for the moral high ground.

Both of these actions were taken at the request of the Chinese government, and both companies complied because this was in their commercial interests: they did so in return for access to the Chinese market. This was a *quid pro quo* arrangement, in which a political favor was exchanged for a commercial one.

Despite the simplistic justifications offered by their respective public relations departments, the fact is that Microsoft is silencing freedom of speech, and Google is degrading freedom of information. These are clear trespasses against basic civil liberties.

These are stark illustrations of how commercial interests can be highly injurious to the human condition. Any doubt we may have about the hazards of entrusting Internet governance to the commercial providers is surely dispelled by these examples.

2.4 The solution: Free software presence in the services domain

In the general software arena, the free software movement and GNU/Linux play an essential role in providing a non-proprietary alternative to the Windows monopoly. It is imperative that a similar non-proprietary alternative be established for the services industry. Without such an alternative, Internet governance will remain largely in the hands of commercial interests.

All the basic principles of the free software movement carry over into the services domain. In particular, Internet services are amenable to similar cumulative and collaborative development mechanisms to software. And as in the case of software, these mechanisms depend critically on the appropriate *freedom of action*.

But in order for these mechanisms to be replicated in the services arena, the proper formalization is required. This formalization must include: a coherent model for non-proprietary services, a definition of the concept and principles, creation of industry-wide awareness, a framework for collaborative development, leadership, and a rallying point for participation. In short, an equivalent movement to the free software movement is required in the Internet services arena.

3 The Libre Services model

We are proposing a radically new, completely non-proprietary model for the delivery of Internet services. We call this the **Libre Services** model.

Libre Services are an extension of the principles of free software into the domain of Internet services. Free software allows complete freedom of action: it may be copied and reused without restriction. Libre Services provide equivalent freedom of action: they are Internet services that may be copied, modified, reproduced, extended, and redistributed *in their entirety*. Libre Services are:

- Implemented entirely in free software
- Based entirely on patent-free protocols
- Reproducible as a complete service by anyone

They are a communal resource, not owned by anyone, freely available for use by society at large. Any company, organization or individual can reproduce and host any Libre Service, and deliver the service to

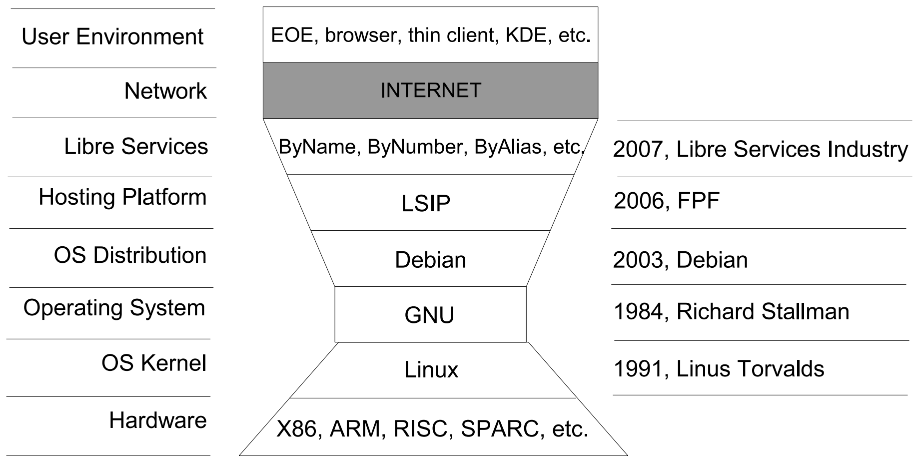

Figure 2: Context for Libre Services

others. Or any group of individuals can host the service for themselves, thus acting as their own service provider.

The Libre Services model exists in relationship to the proprietary Subscriber Services model of AOL, MSN, Yahoo and Google, in an analogous way to how GNU/Linux exists in relation to Microsoft Windows. Both Libre Services and GNU/Linux are open and free models, and both provide the essential *freedom of action* that is absent from the closed model.

3.1 Technological context

Figure 2 shows how Libre Services fit into the overall free software context. The left side of the figure shows the general technological requirements, and the center of the figure shows how these are realized today. It should be noted that this realization represents the general industry environment, and specific implementation choices we have made, at the time of writing in early 2006. This realization is open to future change and evolution.

The figure is not unlike the common "hourglass" representation of the OSI (Open Systems Interconnection) and Internet architectures, in which a high degree of heterogeneity at the upper and lower levels is bridged by a common set of protocols in the stem of the hourglass.

Figure 2 has a similar shape, but here we are showing how a high degree of heterogeneity among hardware platforms and user environments is bridged by a unifying set of software components. At the bottom of the figure is the hardware level, consisting of the large number of hardware platforms and architectures available for client side and server side computing.

GNU/Linux

In the open-source software world, the key enabling component is the GNU/Linux operating system, providing a complete environment for open-source software development. Linux, the operating system kernel, is the unifying interface for running GNU on a wide variety of hardware platforms.

(To place things in historical perspective, the GNU Project was founded in 1984. The earliest version of the Linux kernel was released in 1991.)

Debian

An important development in the evolution of the open-source software movement was the appearance of GNU/Linux distributions, or complete GNU/Linux software packages, assembled together for easy installation and use. The first of these appeared in 1992, soon after the first release of the Linux kernel.

Distributions play an essential role within the GNU/Linux framework. The integration of the various GNU/Linux components into a usable system is not a simple process. Distributions eliminate the need for a developer to locate, download, compile, install and integrate a large number of necessary components into a working GNU/Linux system. Instead, the complexities of system construction are handled transparently by the distribution software.

Our initial Libre Services implementations are based on the Debian distribution of GNU/Linux [4]. Debian was founded in 1993, and has emerged as the most practical and reliable distribution for software engineering development. Equally important, Debian fully conforms to the philosophy of the free software movement. The Debian project is guided by the *Debian Social Contract* [5], an explicit statement of the philosophy and guiding principles of Debian.

Libre Services Integration Platform

The Libre Services Integration Platform (LSIP) is a generalized framework for developing Libre Services. All our initial implementations are based on LSIP.

LSIP is a set of tools, policies and conventions for services development and deployment. It provides a uniform, disciplined environment for transformation of software into services, integration, and service aggregation. It allows efficient integration of free software components into coherent services.

LSIP is the key technological component of Libre Services. It is the component that makes generalized, large-scale services development practical and efficient.

User environments

Libre Services are implemented as server-side software entities, typically running on servers at the service provider's premises, remote from the user. The user interacts with the service using his or her own computer, and may do so using any of a wide variety of user environments, such as EOE (Emacs Office

Environment), a web browser, thin client, or KDE (K Desktop Environment). These are shown at the top level of Figure 2. The link between the user environment and the service is the Internet itself.

Since Libre Services are completely open, there are no proprietary restrictions on which user environment can be used to interact with the services. The Libre Services model allows *any* user environment to interact with *any* service using *any* hardware platform.

3.2 Benefits to society

Libre Services bring the cumulative and collaborative development characteristics of free software into the services arena. They are open to completely unrestricted, large-scale collaborative development, and therefore have an ability to undergo complex evolutionary growth that cannot be matched by the proprietary model. In terms of richness of functionality, they have the ability to surpass the proprietary model completely.

In addition to straightforward end-user functionality, Libre Services also provide a number broader societal benefits.

Engineered for the user, not for business

In both the free and proprietary models, software is created by engineers. But the motives driving the engineering effort are entirely different under the two models.

In the proprietary model, engineering acts at the behest of business. And the prime directive of business is to make profit. Though corporations like to proclaim that their number one concern is the customer, this is a fiction. The welfare of the customer is of concern to the corporation only insofar as it relates to profit; beyond that it is meaningless. Under the proprietary model, service providers can and frequently do act directly against the interests of the user.

But in the free software model, engineering does not take place within a business framework. Instead it is a collaborative effort, undertaken by many organizations and individuals in a variety of diverse environments. Therefore *the dependence of engineering on business imperatives is severed.* The engineering effort no longer takes place at the behest of business. Instead it is driven by fundamental, constructive engineering motives: the desire of the software engineering community to create applications of real value to the user.

The resultant software is therefore fully aligned with the usage, requirements and interests of the user. It is built to benefit the user, not to benefit business.

Civil liberties: Services operated by the user, for the user

In the proprietary services model, the service provider and the user are two separate and distinct entities. All policy decisions regarding operation of the service are made by the provider, with little input from the user.

In particular, the user is entirely subject to the provider's actions regarding civil liberties issues such as privacy, censorship, and freedom of speech. The provider's actions are taken based on commercial considerations, and these actions may constitute serious violations of the user's civil liberties.

As we have noted, commercial providers can silence dissent and enforce censorship in order to gain access to foreign markets. As we will note in the next section, they can also cooperate in government intrusion into their users' personal and private affairs. Commercial providers have also cooperated with enforcement of censorship and freedom of speech restrictions by monitoring web logs and bulletin boards, erasing banned content, and reporting offenders to government authorities.

Under the Libre Services model, however, any group or community of people can host the service cooperatively for themselves, and operate it according to whatever policies they see fit. The Libre Services model thus breaks the separation between the provider and the user—they can now be one and the same.

Libre Services can be operated by the user, for the user. The civil liberties of the user are thereby assured.

Privacy and security

In the proprietary services model, user activity can be monitored without the user's knowledge or consent. There are two forms of monitoring that present societal concerns:

- Monitoring by commercial entities
- Monitoring by government agencies

In the case of commercial monitoring, any aspect of a user's activities can be recorded and reviewed by the service provider. This includes the content of incoming and outgoing e-mail, search queries, websites visited, products and services purchased—indeed, any service usage that is technologically available to the provider can be monitored, without the knowledge or consent of the user.

This form of monitoring is much less of an issue in the case of Libre Services because, as we have noted, the service is designed to benefit the user, not a commercial entity. Since the service is not created for commercial benefit, there is no great incentive to include commercial monitoring capability within the service.

But to the extent that commercial monitoring remains a concern, the Libre Services model can provide complete guarantees of privacy. In the case of proprietary services, based on closed software, monitoring can take place because the community of users has *no way of knowing what the software is actually doing*. But in the case of Libre Services, the complete openness of the software permits verification and authentication that the service is completely free from all monitoring activity. The community of users is able to know exactly what the software is doing, and that it is doing no more and no less than they wish it to do.

Much more worrisome than commercial monitoring is monitoring by the government. National governments may have very broad powers to monitor their citizens' usage of Internet services. In the USA, an agency with sufficient authorization can compel a service provider to disclose all available information about a user, and cooperate in monitoring all communications and other service usage, without the user's knowledge or consent. The FBI's controversial Carnivore system, for example, is designed to capture all e-mail traffic for a particular targeted individual. Post-9/11, the necessary authorization can be provided simply by association, at several levels of remove, with someone the government considers to be a person of interest for national security reasons.

In the proprietary services model, covert government monitoring is possible because the user has *no way of knowing what the service provider is doing*. In particular, the provider is under no obligation to disclose government monitoring to the user. But in the case of Libre Services, any individual or organization can prevent covert monitoring by running the service for themselves, rather than leaving it in the hands of a third-party provider.

In addition, by eliminating the separation between the provider and the user, the Libre Services model makes current monitoring practices impractical. Under the proprietary model, a government agency conducts monitoring activity by directing its compliance demands against a well-defined commercial service provider. But under the Libre Services model the oligopoly of commercial service providers has disappeared, to be replaced by numerous private self-providers.

The government can still come knocking and demand access to a user's information. But it must now direct its compliance demands against a multiplicity of individual persons and organizations. And it can no longer do this without the user's knowledge.

Service stability and continuity

In both the free and the proprietary worlds, software applications and services can be discontinued. The provider of the application or service can go out of business, or may decide to discontinue supporting the application. In either case the user may be left with an investment in an "orphaned" application. But the dynamics of how this occurs, and the effects on the user, are very different under the two models.

In the free software world, application extinction occurs because of migration of the community of users away from the application towards other, better applications. Extinction occurs because of a process of user-driven convergence, based on the genuine merits of alternative solutions.

In the proprietary world, applications are left orphaned not by the actions of *the users*, but by the actions of *the provider*. And these actions may be taken for reasons that have little to do with the actual merits of the application, but may be based on purely business considerations.

Because of these differing dynamics, application orphaning is a gradual and organic process in the free software world, whereas in the proprietary world it can occur suddenly and without warning.

Thus in the free software world, continuity of applications and services is much less of an issue than in the proprietary world. Applications persist based on their merits, and where they do not persist, this is to the ultimate betterment of the industry and the user.

But to the extent that service continuity is of concern to the user, the Libre Services model provides guarantees of continuity that are completely absent from the proprietary model. First, since the services are a communal resource, the user is not tied to any particular service provider. The effect of the Libre Services model is to *decouple the service functionality from the service provider*. If a provider goes out of business or discontinues providing the service, a user can simply go to an alternative provider, and be assured of receiving a functionally identical service.

The same consideration applies to the availability of technical support for the service. Again, since the service is a communal resource, the user is not tied to any particular vendor for technical support. Under the Libre Services model, technical support remains readily available for as long as the service itself remains available.

Finally, under the rather implausible scenario in which an entire Libre Service inexplicably disappears, but an organization remains fully committed to the orphaned service, the organization still has recourse. Since Libre Services are implemented entirely in free software, the organization has guaranteed perpetual access to the software. If necessary, the organization can reproduce and operate the entire service for itself.

Complex integration of user environments with services

As indicated in Figure 2, Internet services work by communication over the Internet between a client application running in the user environment, and a corresponding server application running within the service.

In the proprietary model, a particular service is tied to certain specific user environments. The service can be accessed only via one or two user environments, typically a web browser, and possibly also a dedicated client application provided by the service provider.

Under the Libre Services model, however, there are no proprietary limitations placed on integration between the User Environment layer and the Libre Services layer in Figure 2. Since the service is completely transparent, *the dependence of the service on any particular user environment is severed*. Thus any user environment can be integrated with any Libre Service.

Furthermore, a much more complex level of integration is possible. In particular, free user environments (i.e. user environments based on free software) can be integrated with Libre Services. And since both the client and server sides of the service are now completely transparent, this permits a highly complex level of integration between the two. This allows the development of Internet services with a power and versatility that far exceeds what exists today.

For these reasons we believe that the free software movement as we see it today is just the beginning. Today, free software exists at operating system and application level. The Libre Services model brings the power of free software to the Services level, the User Environment level, and the integration between these two levels. The result will be a complete transformation of the Internet services industry.

3.3 Benefits to service providers

The Libre Services model also brings important benefits to the providers of Internet services.

A truly open industry

Under the proprietary model the Internet services industry is dominated and controlled by a few large providers. These dominant players actively stifle competition by means of restrictive business practices, such as the use of proprietary protocols, highly aggressive patent assertion, and other practices based on ownership and control of intellectual assets.

Under the Libre Services model, however, there are no intellectual property barriers to business entry, and any company that wishes to host services can do so. This has major business consequences. The effect of the Libre Services model is to open the entire services industry to free market entry. This will result in unrestricted engineering collaboration and business competition, and will catalyze enormous industry growth. Libre Services will transform the closed industry of today into a truly open industry, in which all participants can compete on a level playing field.

Collaborative development

In the proprietary model, small service providers can also be marginalized on the basis of service quality and functionality. A small proprietary service provider cannot compete with the resources of the large providers in their ability to develop new and better functionality.

Libre Services, however, are based on the large-scale cumulative and collaborative development mechanisms of free software. Any development contribution made by any engineer, anywhere, becomes immediately available to the entire constituency of service providers. In effect, the Libre Services model permits global pooling of engineering development resources.

This provides a level of cooperative development capability that far exceeds the resources of even the largest proprietary provider. Eventually the Libre Services model can surpass the proprietary model entirely in terms of service quality and functionality.

Industry representation

As we saw in the case of Instant Messaging, small proprietary providers can also be marginalized in terms of representation in industry decision-making, for example in establishing technical standards and protocols. This is because a service provider's voice in such decision-making is based ultimately on the provider's size, in terms of number of subscribers. The major commercial providers are thus able to exert a dominating influence over industry standards and policies.

The Libre Services movement, however, provides a unified voice of advocacy for all its subscribers. In effect the Libre Services model permits global pooling of the entire community of Libre Service subscribers as a single constituency for industry representation.

3.4 Our goal: Creation of the Libre Services industry

Libre Services are the right way to deliver Internet services to the world. As well as providing a number of vitally important societal benefits, they are also the proper basis for a healthy, thriving and egalitarian services industry.

Our goal is nothing less than the creation of an entirely new industry: the Libre Services industry. Much as others established the free software movement twenty years ago, we are establishing the Libre Services movement today.

3.5 The need for a movement

Possibly the free software movement might have come into existence on its own, in some spontaneous organic way, even without the actions of the Free Software Foundation. We only get to experience one history, so we will never know. But at the very least the Free Software Foundation greatly expedited this process by explicitly formalizing the principles of free software. And quite possibly, without the Free Software Foundation or some other entity taking this initiative, the free software movement might never have come into existence as a coherent movement at all. In any event, 20 years later, the free software movement now exists as a viable alternative to proprietary software. Society is surely better off for having the choice.

Similar speculations can be made about the Libre Services movement. One could question whether there is any need for anything so grandiose as a "movement," with a "blueprint" and a "manifesto." Possibly it too might arise spontaneously, without requiring any explicit formalization. Possibly everything we are trying to achieve is destined to occur anyway, as a natural consequence of the already established free software movement. One could argue that services are just another form of software, and the existing free software movement is already sufficient to create its own non-proprietary presence in the services arena.

It is true that in a purely reductionist sense, services are just another form of software. But from a holistic viewpoint they are no more "just" an extension of software than biochemistry is "just" an extension of chemistry. Internet services have a richness and complexity beyond that of general software, and they have a set of dynamics that are not apparent within the general software arena.

Technical development challenges

The creation of a coherent service is a complex process, presenting its own set of technical challenges. This process involves the transformation of software components into a service-oriented implementation, integration of software components, and service aggregation.

The existing technical conventions of free software are not well adapted to these requirements. These requirements have been partially met by the appearance of hosting platforms, but these have been implemented on a specialized, *ad hoc* basis, and the integration of software into services remains inefficient and costly.

In the longer term a much more general framework is required. This framework must include a coherent set of tools, policies and conventions for efficient integration of free software components into services, and for consistent service aggregation. The Libre Services Integration Platform (LSIP) we have developed is one such framework.

Collaborative development challenges

In addition to these technical issues, managing collaborative development on services presents another set of challenges. For example, collaboration on services is much more vulnerable to software division (the tendency of open-source projects to split into rival projects) than collaboration on individual software components. For this and other reasons services require a set of collaborative policies and methodologies that are significantly different from the general free software model.

Motivation for development

Furthermore, the motivations driving development of free software and Libre Services are quite different. The initial "consumer" of free software was the engineering community itself—the community of software engineers who recognized the need for non-proprietary software tools such as editors, compilers, debuggers, etc. Free software was developed by engineers for engineers, and so the benefits of free software translated directly into the necessary action to create it.

By contrast, the "consumer" of services is the end-user, and a completely different set of motivations and dynamics must come into play to create Libre Services. Though the benefits of Libre Services are very real and very far-reaching, it is not clear that these benefits translate directly into forces and action to create them. Under these circumstances an explicit movement is required to provide the necessary motivation.

3.6 The time is now

In the end, the best anyone can say is that there is uncertainty about the future. One could be complacent, take no action, and accept whatever default history will hand us. But the stakes are very high. We believe it behooves us to take positive action to ensure the future, rather than complacently assume that history's default will be the one we desire.

The time to do this is now. All the necessary requisites for Libre Services now exist. The key enabling components are the Linux kernel, the GNU operating system, and the Debian distribution. These components are now sufficiently complete and mature to make Libre Services a reality.

The window of opportunity for this is unknown. It is possible that the proprietary services may become so entrenched that they become impossible to dislodge. The number of proprietary subscribers may become so large and so tightly bound to their service provider, that a different services model can no longer gain credibility. In this case, the opportunity to establish Libre Services, if not acted upon quickly, may be lost.

We would like to ensure that, 10 years from now, society will have a non-proprietary alternative to the proprietary Internet services. As in the case of free software, society will surely be better off for having the choice.

4 Libre Services: From concept to reality

Figure 3 provides an overview of the major stages of development needed to establish Libre Services as a reality, and the principal deliverables needed at each stage. The major requirements are:

- **Conceptual definition**. The first requirement is a clear definition and explication of the concept. The necessary intellectual work must be done to define the concept fully, and written materials created to

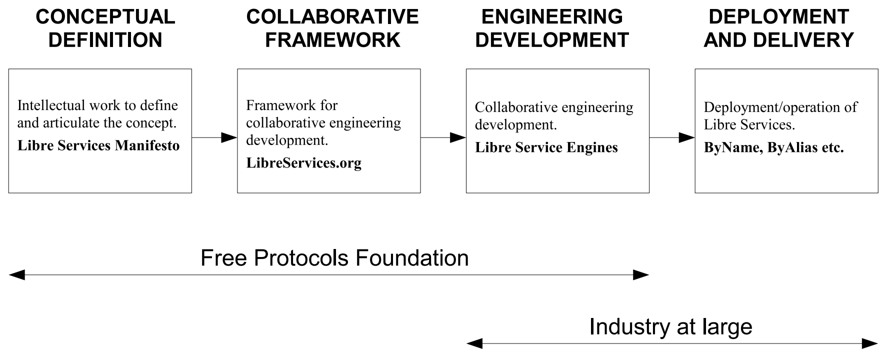

Figure 3: From concept to reality

communicate it to others.

The principal required deliverable is the *Libre Services Manifesto*, a comprehensive description of the Libre Services model.

- **Collaborative framework**. The next requirement is a framework for collaborative engineering development. Open-source software development does not take place in a vacuum—it requires a proper framework to proceed effectively. This framework must include a central software repository, and automated mechanisms to allow developers to retrieve software components from the repository, then resubmit modified software back into the repository. The framework must also define the policies and procedures to be followed for orderly collaboration, and provide various other resources for developers.

  The required deliverable is LibreServices.org, a website and forum providing all the necessary resources and functionality.

- **Engineering development**. Next it is necessary to do the engineering work to create Libre Services software. This is done by the open-source development community within the industry at large.

  The required deliverables are a set of software components and Libre Service Engines.

- **Deployment and delivery**. Finally it is necessary to deploy and deliver services to individual subscribers. This is done by service providers within the industry at large.

  The required deliverables are a set of deployed, usable, first-generation Libre Services.

4.1 Transformation of software into services

Figure 4 provides an overview of the technical processes required to transform free software components into a working service for delivery to users.

In addition to free software components, two other important elements are required to create a service: an integration portal, and LSIP. These three elements are integrated together to create a **Libre Service Engine**.

The Libre Service Engine is a key technological component of the Libre Services model. It is a complete, fully integrated package of service features and capabilities, ready for deployment and delivery by a

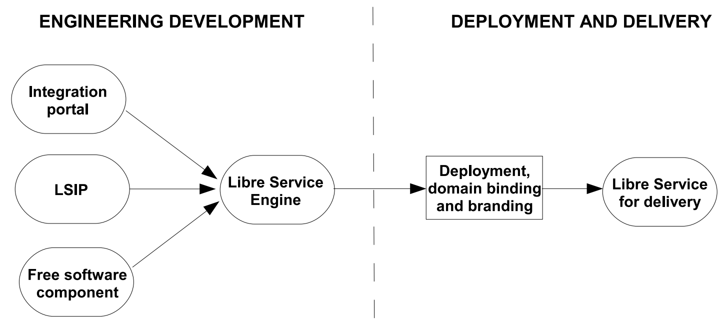

Figure 4: Transformation of free software into Libre Services

service provider. It is part of the definition of a Libre Service that such an engine exist, ready for deployment without requiring any further software integration work.

It is the responsibility of a service provider to create a deployed service based on the service engine. This includes binding the service to a specific Internet domain, branding the service with the provider's trade name, and providing the technical and business structures required to deliver the service and support users.

The vertical line in Figure 4 represents an important conceptual point of demarcation in the Libre Services model. It is the division between industry-general engineering activity, and provider-specific deployment activity. Engineering development is a collective activity, undertaken by the industry as a whole. The resulting assets are industry-generic, available for use by all industry participants. Deployment and delivery, on the other hand, is undertaken by a commercial or non-commercial service provider. The resulting deployed service is associated with and carries the brand name of a specific provider.

Thus LSIP and all service engines are industry-generic resources. ByName, a deployed service, is associated with a specific service provider, in this case Neda.

In the Libre Services model any engineering development, taking place at any point within the industry, becomes immediately available to the entire industry. This means that differentiation and competition among service providers no longer takes place on the basis of service functionality. Instead it must now take place purely on the basis of deployment and delivery characteristics such as service cost, reliability, and customer service.

4.2 Freedom in principle vs. freedom in practice

Both free software and Libre Services offer total transparency and freedom of action. In the case of free software, anyone can copy the software, run it on their own computer, modify it, and redistribute it to others. Libre Services provide an analogous set of freedoms. In the case of Libre Services, anyone can reproduce the entire service, operate the service on their own server, modify the service, and redistribute it to others.

These freedoms are real, and in principle can be exercised by anyone. In practice, however, exercising these freedoms requires specialized skills and expertise. For someone to modify a free software program they must first set up the necessary programming environment, including an editor, compiler, linker and various other tools. They must have the necessary technical skills to do this, and they must also have the necessary programming skills to do the software modifications.

Modifying a Libre Service requires even greater expertise. In addition to all the skills required for free software programming, someone wishing to do this must also have the necessary technical skills to set up and operate their own server. Exercising programming freedom in the context of Libre Services is therefore more technically demanding than for free software, and may not be practical for an individual programmer.

In practice therefore, the freedoms associated with free software and Libre Services can be exercised only by a minority of people with the necessary expertise. The majority of people within society at large have neither the know-how nor the inclination to do hands-on software programming, and so in practical terms cannot exercise these freedoms at all.

Nevertheless, all of society benefits from the existence of these freedoms, because the results are available to all. In the case of free software, a community of free software programmers exists, who do actual hands-on software programming. The result is new and better software applications, which can then be used by anyone. In addition, free software development is done in a open democratic manner, by members of society itself, rather than being under the control of a specific private party, as is the case for proprietary software.

Society benefits from Libre Services in much the same way, though there are some important differences. In the case of Libre Services there will also be a community of services developers, with the technical expertise to do actual services development.

But for Libre Services this development will more typically be undertaken as a group activity rather than an individual activity. Many free software programs are stand-alone applications, running on an individual's own computer, and here software development is something that can reasonably and practically by done by the individual programmer. Libre Services, on the other hand, are normally operated by and provided to groups of people, rather than individuals. Thus in the Libre Services domain, the analogous thing to the individual user is the collective group of users, who wish to operate their own Libre Services and/or do services development.

But as with free software, the results of this development are then available to all of society. And as with free software, this development is done in a open democratic manner, rather than being under the control of a proprietary service provider.

Another difference between the free software and Libre Services models is how the end results are utilized by society at large. Among other things, free software frequently takes the form of a stand-alone application, suitable for use by an individual user. In this case the beneficiary is the individual user who can run the program.

But a Libre Service is not usually something that is run by an individual user. Instead, Libre Services are utilized by society at large via two different sets of beneficiaries. As a service, there is both a provider and a user of the service. In general these need not be the same entity, and both are beneficiaries of the Libre Services model. The results of Libre Services development are available for deployment by any service provider—commercial or non-commercial; public, corporate, or private. And the resulting deployed services are then available to the community of end users of the services.

Thus in the case of both free software and Libre Services, all of society benefits from a set of freedoms available in principle to all, but exercised in practice by few.

4.3 Local software vs. network service

As we noted in Section 2.1, there is a major trend towards providing functionality as a remote service, rather than as a local application. The commercial service providers are actively promoting this trend, since this directly increases consumption of their own services. As a result, network-based services are becoming a pervasive model for how people access computing resources and functionality. However it may not necessarily be in the user's best interests to access functionality remotely, when equivalent functionality can be provided locally.

Some types of computing functionality are not practical for a user to run locally, and such functionality must be provided as a network service. An obvious example is an SMTP e-mail server. Technically one could run this as a local application, but it is far more convenient and cost effective to implement this functionality on a centralized server, shared by multiple users. Another example would be search—it is not feasible for an individual user to run an Internet search service locally. And many other examples. Certain types of functionality are inherently service-oriented, and in practical terms must be implemented in the form of a network service.

On the other hand there are some software components and functionality that must be run locally. At these two extremes there is no discussion to be had—inherently service-oriented functionality must be provided as a service, and inherently local functionality must be provided locally.

But intermediate between these two extremes there is a large class of software components that can reasonably be implemented either as a local program or a remote service, and here the merits of the two approaches are open to discussion.

Certainly, the network-based computing model provides a number of important advantages. These include:

- Reduced total cost of ownership
- Ease of access to specialized software
- Ease of maintenance
- Cost effectiveness of shared computing resources

However, there are also a number of major disadvantages to accessing computing resources remotely, when equivalent functionality can be provided locally. These are:

- **Privacy and confidentiality**. In the case of a proprietary software application versus a proprietary service, users' privacy is far better protected if their personal information (address book, calendar, etc.) is maintained locally rather than remotely. As we have noted elsewhere, commercial service providers are under no obligation to protect users' privacy and civil liberties, and have already shown their willingness to compromise both of these.

This consideration is less of an issue in the case of local free software program versus a Libre Service, since the Libre Services model provides far greater protection of users' privacy and civil liberties than the proprietary model.

- **Integration**. The greater practicality of deep integration between applications, when these are run locally.
- **Control**. Running software locally provides users with a greater degree of control over their computing environment. Users can set up and configure their local computing environment according to their own choices and preferences. For example users have choice over which specific software component they will use for a particular type of functionality, or what revision they will use. When the functionality is provided as a service, these choices must be left to the service provider.
- **Programming freedom**. A further dimension of user control is the ability to make hands-on software modifications—i.e. exercise the programming freedom guaranteed by free software and Libre Services. In the case of a local free software program, it is possible for a user with the necessary technical skills to set up a programming environment and modify the software.

  In the case of the same functionality running as a service, however, it may not be feasible for the user to modify the service. If the user is part of a group of people who are running and maintaining their own service, then the user can indeed modify the service, just as an individual user can modify a local program. But if the user is an individual subscriber to a service being run by a second-party service provider, the user cannot modify the service.

  Therefore an additional advantage to running software locally rather than remotely, is that those with the necessary technical skills can modify the software.

  But note that this consideration applies only to those with the skills and inclination to do hands-on software modifications. For the great majority of non-technical users, this difference between local programs and remote services does not apply.

There are thus a number of reasons to maintain computing functionality locally rather than remotely. In general, the partitioning of functionality between the local free software environment and the remote Libre Service should be based on consideration of all the relevant issues.

Other considerations

The above discussion focusses on providing functionality on a user's desktop rather than as a network service. But a user may wish to access her services when she is away from her home desktop, for example at an Internet cafe. Under these circumstances the user may want to have a full suite of computing functionality provided entirely by the service.

It must also be borne in mind that the Libre Services model has a very significant business dimension. It fully supports a model in which services are provided in a commercial, for-profit context, and in this context commercial Libre Service providers must be responsive to their market.

Regardless of the merits of maintaining local functionality, users of the service may wish to access their computing resources remotely. If Libre Service providers are to compete against the proprietary service providers they must fully cater to the demands of their users in this respect.

5 Separation of responsibility: FPF and Neda

5.1 Areas of responsibility

As shown in Figure 3, the major requirements for establishing Libre Services fall under different areas of responsibility.

The resources to be created in the conceptual definition and collaborative framework arenas are general industry enablers, and are sufficiently bounded in scope to be created by a single organization. Therefore major responsibility for creation of these resources will be taken by the Free Protocols Foundation (FPF).

The scope of the required engineering and deployment work, on the other hand, is too large for any one company or organization acting alone, and must be undertaken by the industry at large.

Engineering development

The engineering development work is a communal activity, and can be undertaken by any individual or organization. In the free software model, software developed by any entity is released and licensed under the General Public License (GPL) or other open-source license [6], and so becomes immediately available to the entire open-source development community. In this collaborative environment it is not of any great significance who or which organization creates any particular component of the Libre Services software. The body of Libre Services software is not owned by anyone, in any restrictive sense.

At the outset the FPF is playing a major role in building the necessary engineering resources. We are actively developing LSIP and a set of starting-point service engines. As others begin to participate we expect that this work will evolve into a distributed industry-wide effort, with the FPF playing an appropriate coordination role.

Deployment and delivery

Deployment and delivery of Libre Services is an essential requirement for their adoption. If Libre Services are to come into widespread usage, they must be operated and supported for the end user. In practical terms, this is something that must be done by service providers in a commercial context. Somewhere along the line, there must be a business model that supports the delivery of Libre Services to individual users.

However, as a business activity, this falls outside the responsibility of the FPF. The necessary deployment work must therefore take place in an entirely separate context.

5.2 Complementary roles of FPF and Neda

We have established a separation of responsibility to address this issue. Responsibility for moving this initiative forward will be divided between two separate entities:

- The **Free Protocols Foundation** is responsible for creating all the assets required in the conceptual definition and collaborative framework arenas. This consists of the work to articulate the concept and create the necessary development framework.

The FPF is also taking responsibility for creating momentum in the engineering development arena. We are doing the necessary work to create a set of starting-point engineering resources, and we are establishing a framework to enable collective participation by others.

- **Neda Communications, Inc**. is responsible for creating the necessary deliverables and demonstrating proof-of-concept in the deployment arena. This consists of the work to deploy and operate a set of usable, first-generation Libre Services.

The Free Protocols Foundation and Neda Communications thus play complementary roles in moving this initiative forward.

5.3 Conflict of interest

The relationship between the FPF and Neda is not unlike that between a professional association and a member of that profession; for example the American Bar Association and a particular law firm. The FPF is a voice of advocacy for the Libre Services industry as a whole. It has a moral authority, and its ultimate mandate is to serve the public interest.

Neda Communications, on the other hand, is merely one company that conducts engineering and business operations within this industry. At the outset it is playing a unique leadership role, but eventually we hope it will be just one of many companies delivering Libre Services to users.

But what is highly unusual about this, is that the Libre Services industry does not yet exist; and the enabling framework for the industry, and the first company within the industry, are both being established at the same time. Furthermore, these two structures are being established by the same persons. The authors of this paper are directors of the FPF, and also employees of Neda. Not only are we moving this initiative forward under the FPF, we are also doing the necessary deployment work under Neda.

This creates a conflict of interest issue. In the long run, it is not possible for the same persons to manage both the FPF, and a commercial entity within the Libre Services industry. At some point these responsibilities must be separated entirely.

But for the moment this conflict cannot be avoided, since we cannot divide ourselves in two, either as individuals or as a team.

In the short run, however, we believe that this conflict of interest is manageable. This is because in every capacity—we as individuals, the FPF as an organization, and Neda as an organization—we share the same unifying philosophy. In each capacity we are fully committed to the principles of the free software movement, and in each capacity we share the same ultimate vision: completely open software and Internet services industries, in which *all computing and communications is based entirely on free software*.

In particular, although Neda is a for-profit company, in all its engineering and business practices it fully conforms to the philosophy and principles of the FPF. All protocols developed by Neda are patent-free; all software developed by Neda is free software; all services developed by Neda are Libre Services. The potential conflict of interest is thus greatly mitigated.

It must also be emphasized that, since Libre Services are completely non-proprietary, there are no intellectual property barriers to participation in this industry. Neda enjoys no proprietary advantages whatsoever, and any company that wishes to participate is free to do so. What we are creating is a truly open industry, in which all participants must compete on a truly level playing field. And the ultimate beneficiary of this is society at large.

6 Libre Services: Bootstrapping an industry

We are proposing an entirely new model for delivery of Internet services. This is an ambitious initiative, and will require the participation of many others to turn into a reality.

So far what we have described has been largely theoretical. We have described a new concept, and outlined the major technical requirements to implement it. But to make all this real, a lot more than this is required. An explicit strategy is needed to bootstrap Libre Services into existence.

The goal of our bootstrapping strategy is to create *deployed, usable, first-generation Libre Services*. This means that all the requirements of Figure 3 must be fully addressed.

The FPF is taking primary responsibility for the first two requirements in Figure 3: the conceptual definition, and creation of the collaborative framework. We are creating the *Libre Services Manifesto* to articulate and promote the concept, and we are building the required development framework at LibreServices.org.

But the scope of the next two requirements—the necessary engineering and deployment work—is far too large to be undertaken by any single organization or group of people acting alone. The scope of work required to build a real Libre Services industry is enormous, and can only be accomplished as a collective enterprise by many organizations and individuals.

The key to enabling this collective effort is a coherent basis for participation.

6.1 A project-based model for participation

We have established a coherent basis and model for collaborative bootstrapping of the Libre Services industry. This consists of two major components:

- We have done the initial development work to create a set of starting-point, reference software components. These include LSIP, and a coordinated family of Libre Service Engines.
- We have established a project-based model for collaborative participation.

We have defined a set of projects, representing the next stages of work required to move this initiative forward. Each project is largely independent and self-contained, and ready to be undertaken by an interested group or organization immediately. This project-based model allows efficient, coordinated collaboration on multiple bootstrapping tasks in parallel.

Each project is defined in the form of a Project Document, providing a complete specification for the project. The complete list of projects and Project Documents is provided in a separate article. See the article titled *Libre Services: Projects for bootstrapping*.

In this project-based model the role of the FPF is largely one of coordination and support. We will take responsibility for defining projects, creating and maintaining the Project Documents, and seeking out project sponsors.

6.2 Deployment and delivery

As a commercial activity, deployment and delivery falls outside the scope of the FPF. We therefore rely on independent service providers to do the necessary deployment work. At the outset Neda will take responsibility for this. The primary role of Neda is to demonstrate engineering and business proof-of-concept by deploying usable first-generation services.

As the bootstrapping process gains momentum, we hope and expect that other service providers will also deploy the services.

6.3 An invitation to participate

We invite and encourage others to join us in this ambitious initiative. We invite active participation by all relevant constituencies:

- **Engineering**. To do the work to build usable, first-generation Libre Services.
- **Business**. To deploy and deliver first-generation services in a commercial context.
- **Grantmaking foundations**. To provide non-profit sponsorship and funding for Libre Services bootstrapping projects.
- **The investment community**. To finance commercial deployment of Libre Services.
- **The academic community**. To analyze and criticize the concept.
- **The media**. To publicize the concept and educate the public.

The Libre Services Forum at LibreServices.org provides a variety of resources to assist organizations and individuals who wish to participate.

The forum hosts a number of mailing lists to facilitate various forms of participation. These include a general interest mailing list, a mailing list oriented towards engineering development, and a mailing list oriented towards business development. These provide a means for organizations and individuals to announce their participation, seek out partners, and coordinate cooperative effort.

Immediate mission-critical tasks are the creation of the *Libre Services Manifesto*, and the engineering work to create usable first-generation services. Those who are interested in promoting the Libre Services model can assist us by contributing additional material to the *Manifesto*, revising existing material, and translating *Manifesto* articles into foreign languages.

Organizations and individual programmers who wish to participate in the engineering effort can do so through the software repository and development resources at LibreServices.org. All FPF-sponsored Libre Services software is available at LibreServices.org, licensed under the GNU General Public License (GPL).

7 Starting point for bootstrapping

Bootstrapping of Libre Services is not starting from zero. As part of our framework for collaboration we have created an initial set of starting-point engineering and deployment assets.

We have done substantial engineering development work to create a set of reference Libre Services software components. These are available for immediate use as the basis for collaborative engineering development.

Under Neda we have also done the initial work to demonstrate deployment proof-of-concept.

7.1 Engineering development

Libre Services Integration Platform

We have completed initial development of LSIP, the Libre Services Integration Platform. LSIP is the basis for efficient services development, and a key component of the Libre Services model. It consists of a uniform set of tools, policies and conventions for integration of software into services. LSIP is now sufficiently complete and mature for use as a general industry resource.

Libre Service Engines

We have done the intellectual work to define the requirements for a coordinated set of services, allowing highly generalized interactions among each other. We have identified the key abstractions that must be represented within such a set, including such things as individual persons, businesses, physical locations, and events. We have then designed a family of services to represent these abstractions, and to allow rich and complex interactions among them. The result is a coherent and powerful model for generalized Internet services.

Based on this general conceptual architecture we have created an initial set of starting-point Libre Service Engines. Thus far we have created service engines to provide the following functionality:

- A service for named individual persons.
- A service for individual persons referred to by a numerical ID, and allowing usage via numeric devices.

- A service for individual persons referred to by an alias.
- A service for preserving the memory of deceased individual persons.

We are in the process of creating service engines based on the other abstractions and usage models in our general conceptual architecture. These include services based on the generalized abstraction of business entities, physical locations, and events; services for publication of information; and services allowing complex interaction among the various types of abstracted entities.

Everything we have built—the LSIP development platform, and all the starting-point service engines—is available as free software licensed under the GPL. These are intended to be reference implementations, freely available for examination, evaluation and reuse by the software engineering community.

7.2 Deployment and delivery

Under Neda we have deployed an initial set of working Libre Services based on the starting-point service engines. The first of these is the **ByName** service. ByName provides a basic set of Internet services for the individual user, including a personal domain, personal website, e-mail, mobile messaging, integrated support for mobility, and a few other capabilities. It is the world's first Libre Service!

Libre Services are thus not merely an abstract concept—they are a real construct that exists today.

Thus far we have deployed the following services:

- **ByName**. As basic set of services for the individual user.
  http://www.ByName.com
- **ByNumber**. A similar set of services to ByName, but based on a number assigned to the user instead of the user's name. This allows access to the services using numeric devices such as telephone keypads.
  http://www.ByNumber.com
- **ByAlias**. A similar set of services to ByName, but allowing the use of an alias instead of the user's real name.
  http://www.ByAlias.com
- **ByMemory**. A set of services for preserving the memory of deceased persons. These include features for creating memorials and biographies, and for creating and maintaining shared genealogies.
  http://www.ByMemory.net

These initial services are in varying stages of development, in some cases providing only very basic capabilities. But they are sufficient to demonstrate end-to-end proof of the Libre Services concept. They show that it is possible to deliver real services to an end-user, using nothing but free software.

Acknowledgements

We would like to thank the following persons for their assistance in the preparation of this paper: Steven Caro, Richard Stallman.

References

[1] Free Software Movement. A large body of writing and references describing the philosophy of the Free Software Movement is available at: http://www.gnu.org/philosophy/philosophy.html

[2] Free Software Foundation. An organization dedicated to the promotion and support of free software. http://www.fsf.org/

[3] GNU Operating System. GNU/Linux is the foundational software for the entire free software movement. http://www.gnu.org/

[4] Debian. A distribution of the GNU/Linux operating system and other software packages. The initial starting-point Libre Services are based on the Debian distribution. http://www.debian.org/

[5] *Debian Social Contract*. A statement of the philosophy of the Debian project, and a set of commitments made by the developers towards the free software community. http://www.debian.org/social_contract

[6] *GNU General Public License*. The free software license under which GNU/Linux and many other free software projects are licensed. http://www.gnu.org/copyleft/gpl.html