Libre Texting: A Collaborative Initiative Based On A Reference Implementation

Neda Communications, Inc.

http://www.neda.com

Mohsen BANAN

http://mohsen.banan.1.byname.net/contact

Document # PLPC-110015
May 3, 2010

Available on-line at:
http://www.neda.com/PLPC/110015

Contents

List of Figures

Part I

Summary, Roadmap and Purposes

1 Summary

This article describes **Libre Texting**, a new incarnation of the Mobile Email/Wireless Texting (henceforth just “Texting”) medium, based on the principles and ideology of Free Software. Today the Texting industry is dominated by the proprietary, walled-garden model. The Libre Texting model provides equivalent functionality to existing proprietary Texting solutions such as BlackBerry, but (a) based entirely on patent-free protocols and free software, (b) using any mobile Internet device, and (c) wherever any type of wireless Internet connectivity is available.

The Libre Texting model we describe comprises five major elements: (1) the inherent propagative power of the Free Software and end-to-end models, (2) an Overlay Network architecture, (3) a new set of messaging protocols, (4) a novel software architecture for Message Transfer Agent integration, and (5) a new software architecture for multiform device integration, based on Device-Resident End-MTA middleware.

The key to technical implementation of Texting is the right set of protocols. Existing email protocols are inadequate in several respects—they lack push-mode delivery, and they also lack the bandwidth and power efficiency necessary for wireless applications. The proposed new protocol profile addresses and corrects these shortcomings.

Libre Texting is highly synergistic with current industry dynamics, including: a burgeoning recognition of the power of Free Software, ready availability of Linux-based mobile devices, and widespread Wi-Fi availability.

Libre Texting is a transformative technology. It allows the current walled-garden hegemony to be broken, creating a new point of convergence as the standard. This has major engineering, business and societal consequences, with tangible benefits to the end user. For example, greatly expanded range of choice—the user can now make independent selection of each component of her Texting setup, including the user interface, the Mail User Agent, the device, and other desired components (forwarders, synchronizers, filters etc.). The Libre Texting technology unbundles the Texting application, opening every point of the Texting technology chain to competition and best-in-class component selection. Furthermore, the Libre model opens the Texting industry to the powerful free software generative dynamics.

In terms of business, the Libre Texting initiative represents a radical shift of the Texting industry to the *non-proprietary, for-profit* quadrant, causing a major industry reconfiguration, with significant winners and losers. The losers are the existing vested proprietary interests, whose economic hegemony vanishes. But the winners are the many more companies who can now enter the Texting market—and the end user who benefits from the resulting competition.

Regarding broader societal consequences, the Libre model provides assurances of transparency, privacy and freedom of speech—assurances that are absent under the proprietary model.

2 Roadmap

This document is one of a series of documents that together describe every aspect of Libre Texting, Libre Services and By* concepts. These are:

- *Libre Texting: An Industry Initiative* [5]. Describes Libre Texting from a conceptual and technical standpoint and creates a collaborative development framework.
  http://www.neda.com/PLPC/110015

- *ByStar Libre Texting Proposal for ISPs/ASPs and Handset Manufacturers* [4]. Describes the step that Handset Manufacturers and Internet Service Providers and Internet Application Service Providers can take to incorporate Libre Testing in their existing products and services.
  http://www.neda.com/PLPC/110017

- *Libre Services: A non-proprietary model for delivery of Internet Services* [8]. Provides a complete description of the Libre Services model.
  http://www.freeprotocols.org/PLPC/100101

- *The By* Concept: A Unified Model for Internet Services* [9]. Provides a complete description of the By* unified services model.
  http://www.neda.com/PLPC/110001

- *The By* Family of Libre Services: The future of the Internet Services industry* [10]. Neda's Open Business Plan. Describes the business dimensions of the By* Libre Services.
  http://www.neda.com/StrategicVision/BusinessPlan

It is recommended that these documents be read together.

The ByStar Libre Texting is part of a bigger picture—it is a particular technological and business component within the much larger ByStar Libre Services initiative.

2.1 Audiences

This document is intended to address a varied set of audiences.

All should read Part I and Part II.
Engineers interested in technical details should focus on Part III, Part IV and Part V.
Software developers are likely to be most interested in Part V.
Business oriented readers (VCs, investors) should focus on Part VI.

3 Purposes

We have chosen to frame our work in the non-proprietary for-profit quadrant, as a result this documents addresses both societal and economic and business concerns.

This document has a number of purposes including:

- Create a framework that facilitates collaboration of engineers to bring about Texting based on the Internet end-to-end principle.
- Provide an alternative to traditional Texting for those concerned with a number of critical freedoms and civil liberties, including privacy and freedom of speech.
- Provide a basis to receive funding from organizations concerned with societal benefits that Libre Texting brings about.
- Provide a basis to allow participation of investors in the business context of our work

3.1 Invitation to Participate

Every aspect of this initiative is Libre—all assets are web published and available as a communal public resource. We invite you to evaluate our analysis and documents, and participate in this new texting model in your own context.

- As a mobile messaging user, when selecting a mobile device and messaging service we invite you to look beyond simple functionality, and consider the model in which the service is offered. The Libre model is inherently aligned with the interests of the user, fully guaranteeing your privacy and civil liberties.
- As a technical or management professional within todays proprietary mobile messaging industry, we invite you to prepare for the future. The walled garden model has been a wild ride, but now its over, and the most actively-thinking network providers know when its time to switch horses.
- For active participants in the free software movement, we invite you to join us in advancing the Libre Texting initiative. To facilitate participation we have established a project-based model for collaborative action, described in the article titled, Libre Services: Projects for Bootstrapping. This document presents Libre Texting and other Libre projects that can benefit from collaborative work and enhancement.

Finally, if you think the Libre Texting initiative has merit, we ask you to help spread the word. Please feel free to distribute this article to others who may be interested, either *in toto*, or in the form of the following link to its website location: http://www.neda.com/PLPC/110015

Part II
Concept

4 Introduction

In this article we describe a new incarnation of the Mobile Email/Wireless Texting medium (henceforth just "Texting"), based on the ideological principles of Free Software. We refer to this new incarnation as the **Libre Texting** model. Texting is already well established as a communications medium. But today the Texting industry exists in the form of a proprietary, walled-garden model, controlled by a small number of powerful proprietary commercial interests. The Libre Texting model provides equivalent functionality to existing proprietary Texting solutions such as BlackBerry, but:

- In a completely non-proprietary form
- Using *any* mobile Internet device
- Wherever *any* type of wireless Internet connectivity is available.
- At very large (planet-wide) scale

The Libre Texting initiative has two distinct parts: what we call the "model" part, and the technology part.

4.1 The Model

The model part refers to the technocratic context within which the Texting functionality exists and is delivered. The proposed model is completely non-proprietary, or **Libre**. This means that the Texting service is based exclusively on patent-free protocols, implemented exclusively in free software, and conforms fully to the Internet end-to-end principle.

- **Patent-free protocols**. A patent-free protocol is one for which there are either (a) no known software patent restrictions, or (b) if software patent restrictions are known to exist, then non-restrictive usage rights have been obtained from the patent-holder. [7]
- **Free Software**. The term "Free Software" refers to *freedom of action*, not *zero monetary cost*. Or to use the stock clarification: this means free as in "free speech" not "free beer."
- **Internet end-to-end model**. In this model the Internet serves merely as a passive communication pipe, allowing direct and unrestricted communication between client and server.

The upshot of all this is that Libre Texting does not carry any restrictive limitations on its dissemination, implementation, or usage by anyone.

4.2 The Technology

The technology part consists of the technological components required for practical, large-scale implementation of Libre Texting. The critical enabling technology consists of the following four components, acting together in close integration:

- An Overlay Network architecture for end-to-end communication, permitting NAT traversal, and push-mode delivery.
- A new set of messaging protocols, providing native push-mode delivery, wide-area narrowband efficiency, and scalability.

- A novel software architecture for smooth integration into existing Message Transfer Agents (MTAs). This is necessary for ready industry adoption and integration into multiple MTAs.
- A novel software architecture for uniform integration with multiple existing open platform devices and Mail User Agents (MUAs). The architecture is based on the concept of a Device-Resident End-MTA middleware module, acting as intermediary between the protocol software and the MUA.

Note that the Libre Texting initiative is not primarily about new or enhanced messaging functionality. In terms of capability, Libre Texting provides essentially equivalent functionality to existing Mobile Messaging/Texting solutions such as the proprietary BlackBerry system. Rather, it is about a radically new model for ownership, development and delivery of this functionality.

This article describes the Libre Texting model, and the technology required for practical implementation of this model, on a multiplicity of devices, and at very large (planet-wide) scale.

5 Background

The mobile messaging industry of today is a closed, proprietary construct. Existing wireless texting/messaging implementations (telephony SMS, and mobile email solutions such as BlackBerry) are walled-garden implementations, closely protected by patents, copyright and trade secrecy. The wireless phone companies and/or their business partners own and control every component of the messaging service, including the device, the protocols, the software and the network.

In addition to their proprietary nature, existing solutions violate the Internet end-to-end principle by implementing centrally controlled, service-provider store-and-forward components as a function "within" the network. This is in contrast to the Internet email architecture, which is end-to-end.

5.1 Industry Enablers

Until quite recently, implementation of a Texting solution outside these walled-garden environments has been blocked by absence of the necessary non-proprietary components, such as open devices and public wireless spectrum. But now a completely non-proprietary, end-to-end Libre Texting solution is technically possible. This is enabled by a number of industry developments:

- Public spectrum Wi-Fi is now ubiquitous and has become the standard technology for final-leg device connectivity. In many locations Wi-Fi is available for direct, single-leg connectivity between the mobile device and the open Internet. This coverage can be expected to spread, eventually resulting in near-universal Wi-Fi Internet access.
- In situations or locations where direct Wi-Fi Internet connectivity is not available, a number of wide-area networks now exist to provide second-to-last-leg, wide-area wireless connectivity.
- Mature and sophisticated Linux-based PDAs are readily available as generic open mobile devices.
- Open, patent-free protocols exist for efficient wireless messaging.
- Device and server implementations of the protocols exist in the form of free software.
- The eventual transition to IPv6 will allow restoration of the true mobile Internet end-to-end model, delivering mobile messaging capability without any form of built-in dependence on the service provider.

Thus all the necessary industry assets are now in place to implement a completely Libre Texting service. Every component of the service can be implemented in an open form, without any closed or proprietary or dependencies. This includes the device, the protocols, the software, and access to the wireless network.

5.2 Industry Context

The Libre Texting model is also in line with major cultural and strategic dynamics now prevailing throughout the industry.

The existing Texting/Mobile-Email industry is dysfunctional and unstable in a number of respects. First, the industry is severely fragmented. In 2010 there are five major mobile messaging players: (1) BlackBerry, (2) AT&T + Apple iPhone, (3) Sprint + Palm Pre, (4) T-Mobile + Google Android, and (5) Microsoft PocketPC. These all provide essentially the same functionality. Yet these are isolated islands of functionality, based on different devices, different protocols, and different Mail User Agents, none of which are cross-compatible. A systems integrator or user cannot mix and match among these components. And as additional industry players attempt to muscle their way into the enormous Texting market, there is every indication that this fragmentation will increase, at least in the short term.

But in the long term, this situation is untenable. As a global communications medium, there are strong forces of convergence towards a single dominant solution. Sooner or later the industry must and will coalesce around a unified Texting solution, providing across-the-board, industry-wide standardization.

In addition, there are strong strategic forces pressing for change within the industry. Today the industry is a proprietary hegemony, from which small business players are excluded. Also excluded are some very large players, exceedingly covetous of the lucrative Texting market. Thus there is great pressure to break the walled-garden regime, by powerful forces outside the walled garden.

Given all this, a major industry shift of some sort is inevitable. Whether acting individually or in concert, there is strong motivation among the industry players to preemptively fabricate, and lay claim to, the eventual point of convergence. The preemptive fabrications may take many forms, but are likely to be quasi-Libre constructs, superficially resembling the true Libre solution we propose.

The Libre Texting model is ideally constituted to emerge as the decisive point of stability and convergence within this environment. In contrast to the proprietary incumbents, and any quasi-Libre pretender, Libre Texting is not constrained by any form of proprietary ownership mechanism such as patents or restrictive copyright. This is the fundamental generative power of the free/Libre model. It is this power that can cause Libre Texting to displace the existing proprietary regime, in the face of ferocious attempts by very powerful vested interests to defeat it.

Furthermore, there is a growing cultural awareness of the tremendous power of free and open source software (FOSS), and a dawning recognition that this is slowly engulfing the proprietary software industry. Thus there is a cultural tropism towards acceptance of the Libre model in both ideological and practical terms.

Our goal is to establish Libre Texting as the point of convergence and the common standard for operation of all devices, and all message transfer services, worldwide. We believe Libre Texting is the right solution at the right time. The inherent generative power of the Libre model, together with the current industry instabilities, together with the enabling technological components we describe, coupled with the right sort of engineering and business execution—all this can destroy the existing proprietary regime completely, outstrip any quasi-Libre hijack attempt, and establish Libre Texting as the new industry standard.

It is also worth noting that the shift to Libre Texting is evolutionary in nature. While the model and its effects are revolutionary, deployment and adoption of Libre Texting requires no abrupt technological discontinuity—its deployment and growth can proceed gradually and organically. Libre Texting can initially be introduced into the existing proprietary environment as quietly as inserting a thermometer.

5.3 Definitions

A consistent terminology for wireless texting/messaging has not yet been established, and terms such as "texting," "wireless messaging," and "mobile email" are often used interchangeably, and with different meanings. “Texting” is often used in the context of telephony SMS, and “messaging” is often used in the context of mobile email, but this usage is by no means universal. The best we can do is define our own terms clearly.

5.3.1 Texting

Throughout this document we are talking about an email-type service, with a richer functionality than today’s telephony SMS. Specifically, we use the term **Texting** to mean a mobile messaging service that:

- Supports the unconscious carry, always on, model for device usage
- Supports immediate (push-mode) delivery and alert for right-now messaging
- Is a functional extension of Internet email, oriented to short text messages

5.3.2 Libre Texting

In terms of model, we now formally define **Libre Texting** as a Texting service that:

- Is based exclusively on patent-free protocols
- Is based exclusively on 100% free software
- Is delivered as a Libre Service
- Conforms fully to the Internet end-to-end model

5.3.3 Libre Texting Device

We also define a **Libre Texting Device** as a device that:

- Has an unconscious-carry form factor (e.g. shirt or pants pocket)
- Has text format input and output capability
- Has Wi-Fi for last-leg connectivity, or other form of wireless Internet connectivity
- Is an open platform—i.e. permits unrestricted software addition and configuration

Note that there is no implication that the device is necessarily a traditional data-enabled mobile phone. To the contrary, our baseline assumption throughout this document is that the device takes the form of a typical Wi-Fi-based Mobile Internet Device (MID).

Note also that Wi-Fi specifically is not a necessary requirement. Wi-Fi capability is now near-universal in mobile devices, so that this is in fact the most common form of last-leg device connectivity. But any form of wireless Internet connectivity will suffice.

In 2010 a large number of Libre Texting Devices are available in the marketplace, including Nokia n800/n810, Nokia n900, unlocked Android, unlocked iPhone, PocketPC, and others.

Of course any netbook is also entirely capable of supporting Libre Texting, though it lacks the unconscious-carry form factor. But our primary focus is on the more unconsciously portable devices as defined above.

5.4 Initial Setup

We assume the following initial conditions as a starting point: (1) the user already has or will independently acquire a Libre Texting Device as defined above, and (2) the user has final-leg Wi-Fi (or other wireless) connectivity to the Internet.

Setting up Libre Texting service starts with provisioning the user with the necessary accounts and access credentials. She starts by accessing the generic Libre Texting website (from any point of Internet access), and creating a new account for herself. At the time of account creation, the following items are created for her:

- A Libre Texting account, and account credentials (username and password)
- Overlay Network access credentials (username and password)
- A unique Overlay Network static IP address

Next, we set up her mobile device with the necessary software, and configure the device with her account credentials. Now specifically using the mobile device, she manually logs in to her Libre Texting website account and invokes the device setup procedure. The setup queries her for the device make and model number, then following proper confirmation the setup automatically downloads the free Libre Texting software to her device, and configures it with her credentials for automatic access to the Overlay Network, and automatic access to her Libre Texting account.

She is now good to go. She now has complete Libre Texting capability from her device, including immediate push-mode delivery.

5.5 Functional Operation

Figure 1 shows how Libre Texting delivers functionality to the user. The EMSD-specific components (EMSD User Agent and EMSD Server) are shown in green in the figure. The Overlay Network at the bottom of the stack (shown in yellow) is used whenever EMSD connectivity is needed.

Under one scenario, Libre Texting service can be provided by an independent email service provider (the box labeled "EMSD Enabled Service Provider" in the center of the figure). Under Libre Texting this role can be played by *any* Message Center operator—for example, by any one of the large number of existing ISP companies. All that is required for an ISP or other Message Center operator to become a provider of EMSD-based mobile messaging services, is for them to install the necessary EMSD Message Center software.

Anyone with access to the Internet can now exchange emails with the mobile user. Emails addressed to the mobile account are fielded by the email service provider from the generic Internet using standard Internet protocols, then delivered to the mobile device using the EMSD protocols.

The Message Transfer System may include a number of EMSD Server Agents (EMSD-SAs). Each EMSD-SA may have any number of EMSD User Agents (EMSD-UAs) with which it communicates.

To send a message, the user composes the message then submits it to the EMSD service provider via the EMSD protocols. The service provider then sends the message to its destination using standard Internet email protocols.

Users typically wish their mobile messaging capability to function as a wireless extension of an existing land-based email account. For example, the user may wish the mobile device to act as an extension of a home or office desktop mail application, as shown at the top of Figure 1. This functionality is provided by installing the appropriate mail forwarding software on the desktop system. This software integrates with the desktop mail application, and allows messages to be selectively forwarded to the mobile device on the basis of user-defined email filters. Properly qualified emails are forwarded to the EMSD service provider using standard email protocols, then delivered to the mobile device using the EMSD protocols.

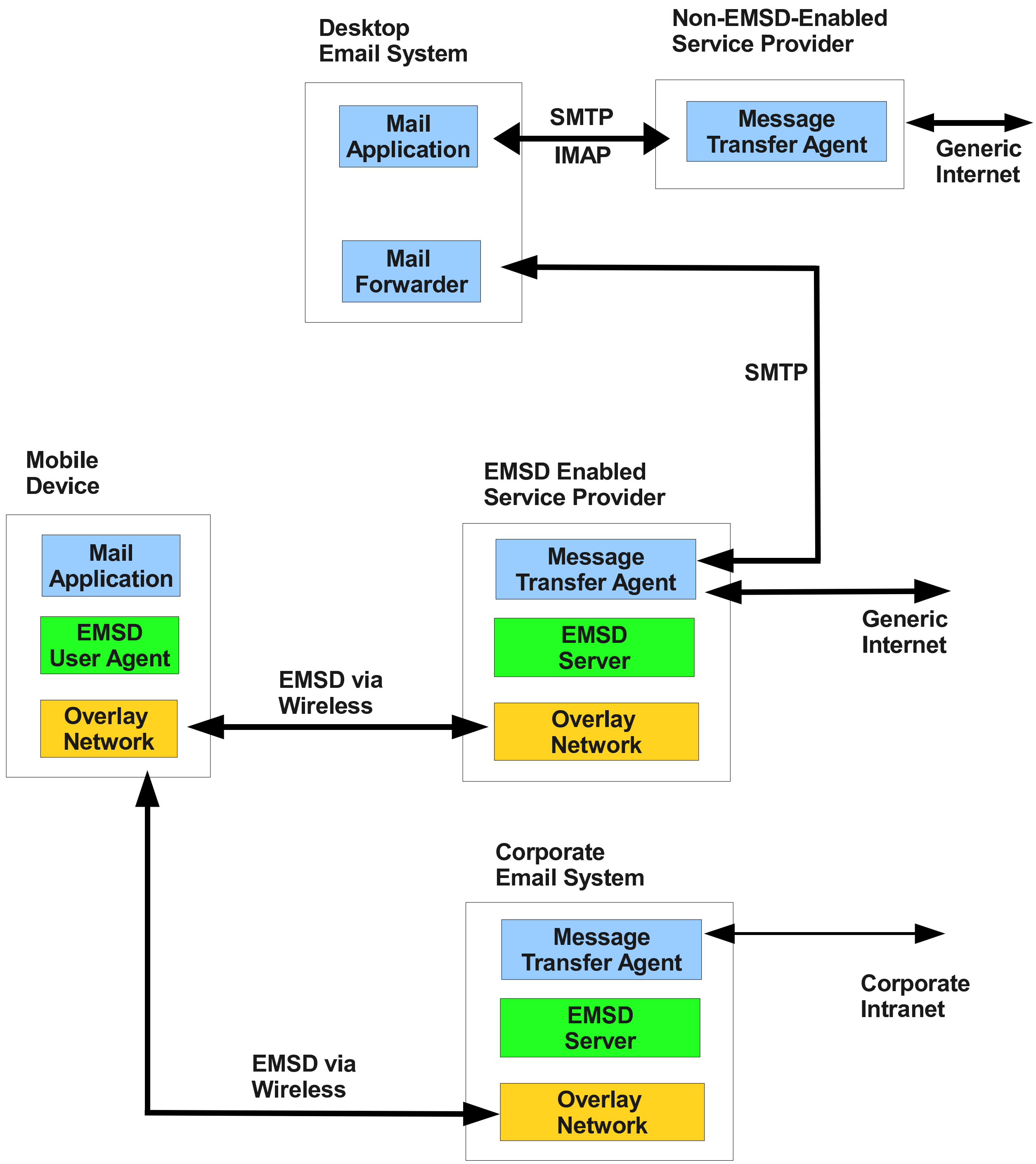

Figure 1: Libre Texting: Functional Operation

Note that the user does not install EMSD software on his desktop; rather he installs forwarder software—the desktop knows nothing of EMSD.

When the user submits a message from the mobile device, the EMSD service provider sends the message to its destination as usual, and in addition it can send a notification to the desktop mail application, to maintain mailbox synchronization between the handheld device and the desktop system.

Under a different scenario, Libre Texting capability can be part of a corporate email system, as shown at the bottom of the figure (the box labeled "Corporate Email System"). This functionality is provided by installing the appropriate EMSD software in the corporate Message Center.

Part III
Technology

6 Technological Components

6.1 Overlay Network Wireless Access

By definition, the Libre Texting Device has some form of wireless Internet connectivity. The most common situation is where the device has Wi-Fi connectivity behind Network Address Translation (NAT), with a leased dynamic private IP address.

Regardless of the wireless modality, it must be extended to achieve NAT traversal, for mobility and presence detection (needed for push-mode operation), and for security enhancement. Our reference implementation uses an Overlay Network architecture to accomplish this. The Overlay Network resides on top of the existing network structure, and provides the following requirements:

- A static IP address (independent of the user's physical location and point of attachment within the local Wi-Fi network)
- Unfiltered bidirectional data flow for Libre Texting traffic (NAT traversal)
- Layer 3 authentication and data confidentiality

The device first establishes initial Internet connectivity, typically behind a NAT. Based on that initial connectivity the device then signs on to the Overlay Network based on its unique static IP address, and is now end-to-end, two-way connected. Thus the device is always reachable by the server. Reachability while mobile is in effect accomplished by the static IP address.

The Overlay Network architecture also gives us immediate presence detection, allowing prompt message push delivery. In other words, the moment the device is reachable, the MTA becomes aware of the device. By means of the Overlay Network we become in effect a network operator, without owning a physical network.

We are using the term Overlay Network in a broad generic sense. The actual overlay capability can be provided by any of the following technologies:

- Virtual Private Network (based on OpenVPN)
- Public Mobile IPv6
- Private Mobile IPv6

All these are viable candidates, and possible future evolutionary directions for Libre Texting. A key consideration for selecting one or more of these for support is ready availability of free software for implementation of these technologies on Libre Texting Devices.

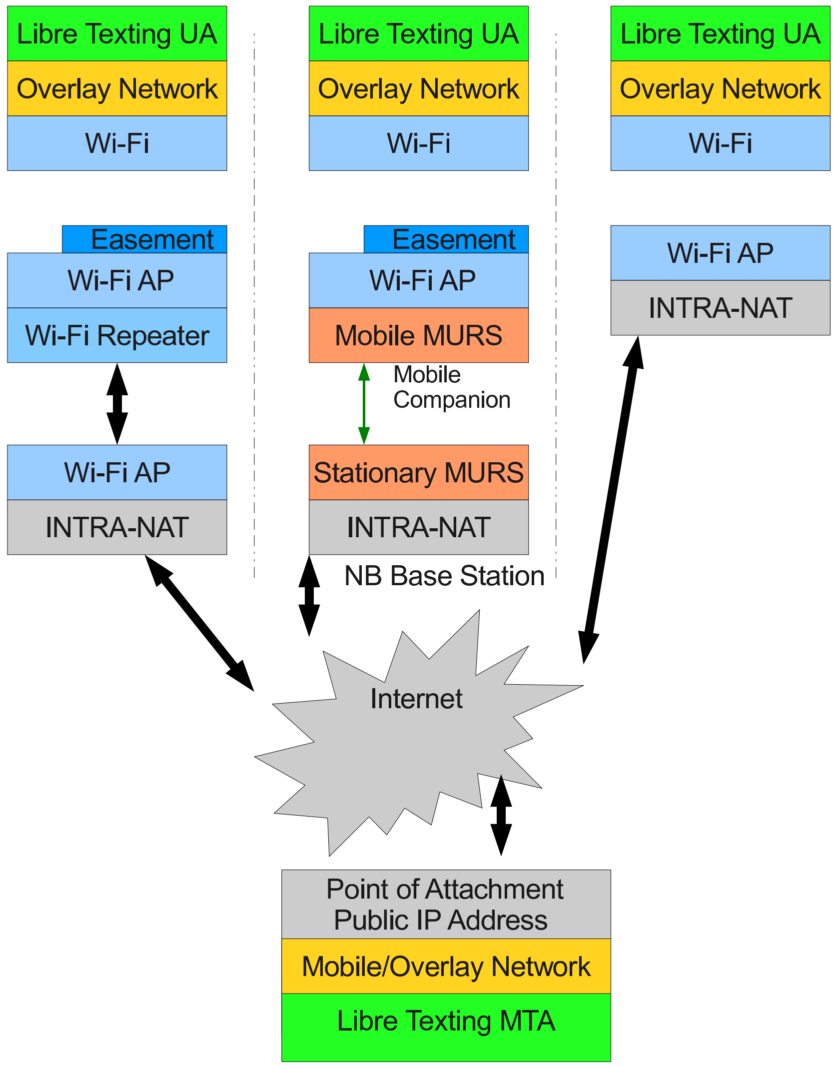

Figure 2: Overlay Network with Wi-Fi last-leg Internet connectivity

The starting point choice for our reference implementation is OpenVPN. OpenVPN is a widely used, free and open source virtual private network (VPN) program for server-to-multiclient encrypted tunnels between MTAs and Mobile Texting Devices. This is entirely adequate to our purposes, and has widely available free software implementations.

Note that we exclude public and private mobile IPv4 from the list of candidate technologies. We exclude public IPv4 because of exhaustion of the address space. We exclude private mobile IPv4 because it is essentially equivalent to Virtual Private Network, and because software for Mobile-IPv4 is not widely available for mobile device platforms.

Figure 2 shows three major scenarios for connection between the device and the open Internet, and extension of this connectivity by the Overlay Network. The right of the figure shows the simplest situation, where the user has direct Wi-Fi Internet connectivity.

The center of the figure shows a situation where a wide-area network (either broadband or narrowband) is used for second-to-last leg connectivity. For example via public spectrum MURS, or using a mobile wide-area Wi-Fi hotspot appliance such as MiFi for use with the cellular network.

The left of the figure shows a situation where Internet connectivity is via multiple concatenated Wi-Fi links. This often happens when a user with Wi-Fi access chooses to share that access by providing an "easement" to others who may have been previously restricted.

6.2 Libre Texting Protocols

The key component of our reference implementation is a set of mobile messaging protocols called the EMSD protocols. The EMSD protocols fully satisfy the necessary technical requirements for mobile messaging. They provide the efficiency required for wireless applications, and support push-mode message delivery, an essential requirement for the expected immediacy of Texting.

The EMSD protocol, titled:

Neda's Efficient Mail Submission and Delivery (EMSD) Protocol Specification Version 1.3

was published as Internet RFC-2524 [12] in 1999.

The EMSD protocol provides reliable connectionless mail submission and delivery services on top of ESRO. The ESRO protocol, titled:

AT&T/Neda's Efficient Short Remote Operations (ESRO) Protocol Specification Version 1.2

was published as RFC-2188 [13] in 1997.

Existing Internet email protocols are not suitable for Texting/Mobile Email, because they fall short in two major respects. First, they lack the necessary efficiency characteristics. Wide-area narrowband wireless networks are severely constrained by bandwidth limitations, and mobile devices are constrained by limitations such as display size, battery capacity, and memory size. These constraints place an extremely high premium on the efficiency of data transfer. Existing Internet protocols such as SMTP, IMAP and POP do not provide the required efficiency.

Second, existing Internet email protocols do not properly support the push mode of delivery. For more detailed discussion of the shortcomings of existing protocols, see the article *EMSD: The LEAP E-Mail Component* [2].

EMSD narrowly focuses on submission and delivery of short mail messages with a clear emphasis on efficiency. It is designed specifically with wireless network usage in mind. EMSD is designed to be a natural enhancement to the mainstream Internet mail protocols, when efficiency in mail submission and mail delivery are important.

Efficiency

SMTP is the main mail transport mechanism used throughout the Internet. However, SMTP is highly inefficient for the transfer of short messages. SMTP is inefficient both in terms of the number of transmissions, and in terms of the number of bytes transmitted. Even when fully optimized with PIPELINING [case?], SMTP remains significantly inefficient.

Table 1 shows the number of N-PDUs exchanged for the transfer of a short Internet email when using SMTP, SMTP with PIPELINING, QMTP, and EMSD. The names used for identifying the PDUs are informal names.

The submission of a short message using SMTP requires 15 transmissions. The submission of a short message with SMTP and PIPELINING requires 9 transmissions. The submission of a short message with EMSD (EMSD-P and ESRO) typically requires only 3 transmissions.

The key design requirement of EMSD is efficiency. Because of the threefold (at least) gain in efficiency, this justifies the deviation from the SMTP model.

	SMTP	SMTP + Pipelining	QMTP, QMQP	EMSD
Client:	SYN	SYN	SYN	Submit.Req
Server:	SYN ok	SYN ok	SYN	Submit.Resp
Client:	HELLO	HELLO	message	ack
Server:	ok	PIPELINING	accept close	
Client:	MAIL	MAIL RCPT DATA	close	
Server:	ok	ok		
Client:	RCPT	message QUIT		
Server:	ok	accept ok close		
Client:	DATA	close		
Server:	ok			
Client:	message			
Server:	accept			
Client:	QUIT			
Server:	ok close			
Client:	close			

Table 1: Comparison of EMSD to other protocols

For mail submission and delivery of short messages EMSD is up to 5 times more efficient than SMTP, both in terms of the number of packets transmitted, and in terms of number of bytes transmitted. Even with PIPELINING and other possible optimizations of SMTP, EMSD is still up to 3 times more efficient than SMTP, both in terms of the number of packets transmitted, and in terms of number of bytes transmitted.

Various efficiency studies comparing EMSD with SMTP, POP and IMAP are available. A detailed theoretical and empirical comparison of SMTP and EMSD is available in *Efficiency of EMSD* [1].

Reliability

In order to provide the same level of reliability that the existing email protocols provide for short messages, it is clear that a reliable underlying service is needed. UDP by itself is clearly not adequate.

Use of TCP however, involves three phases:

1. Connection Establishment
2. Data Transfer
3. Disconnect

The reliable transfer of a short message using TCP involves a minimum of five transmissions, as is the case with QMTP.

The ESRO protocol provides reliable connectionless remote operation services on top of UDP with minimum overhead. ESRO supports segmentation and reassembly, concatenation and separation.

The reliable transfer of a short message using ESRO involves 3 transmissions, as is the case with EMSD-P.

In order to minimize the number of bytes transfered, efficient encoding mechanisms are needed. By selecting ASN.1 as the notation used for expressing the EMSD information objects, EMSD has the flexibility of using the most efficient encoding rules, such as Packed Encoding Rules (PER).

Protocol Function	SMTP	IMAP	POP	EMSD
Submission	XX			XXX
Delivery	XXX			XXX
Relay (Routing)	XXX			
Retrieval		XXX	XXX	XX
Mailbox Access		XXX	X	
Mailbox Sync.		XXX		

Table 2: Messaging Protocol Functionality

Relationship of EMSD to other mail protocols

EMSD is designed to be a companion to existing Internet mail protocols. It is designed to fit within the many protocols already in use for messaging.

The various Internet mail protocols provide different sets of capabilities for mail processing.

Table 2 summarizes the capabilities of SMTP, IMAP, POP and EMSD in different areas of functionality. The number of X's in each cell indicate how well the function is supported by these protocols.

The table clearly shows that combinations of these protocols can be used to complement one other in providing rich functionality to the user.

For example, a user interested in highly mobile messaging functionality can use EMSD for the submission and delivery of time-critical and important messages, and use IMAP for comprehensive access to his/her mailbox.

From the very beginning, the Internet email architecture was not monolithic. Rather, it takes the form of a set of specialized protocols working together.

6.3 Message Transfer Agent software integration

An important consideration is that Libre Texting must fit naturally into the existing structure of the Internet email service. Libre Texting may be disruptive in terms of model, but it must not be disruptive in terms of technological implementation. To facilitate ready industry adoption, the Libre Texting protocols must fit in a straightforward way into existing Message Transfer Agents.

The great majority of Internet email traffic is currently handled by the following MTAs: qmail, Sendmail, Microsoft Exchange, Postfix, Exim.

Figure 3 shows the software architecture used for integration of EMSD-SA into the qmail Mail Transfer Agent. This particular architecture is specific to Linux and qmail, but we expect that integration with the other major MTAs can be accomplished based on a similar scheme.

The configuration in Figure 3 consists of the usual qmail architecture (shown in blue), with emsd-sa-submitd (shown at the top in green) added as a module to process incoming EMSD traffic. The architecture of emsd-sa-submitd is similar to that of mailfront, which is a modern modular replacement for qmail-smtpd. This architecture permits sharing of the Credential Validation Module (cvm), and the Credentials, between emsd-sa-submitd and mailfront.

Based on the presence detection information acquired from the Overlay Network, EMSD deliveries are initiated immediately the device becomes reachable.

[MB to fix up] The novel and experimental (unproven) part of this proposed architecture is the linkage between the Presence Detection module (shown in orange) and qmail-queue. Apart from its main input, output and timers, qmail-queue is not event driven. An important technological objective is to investigate

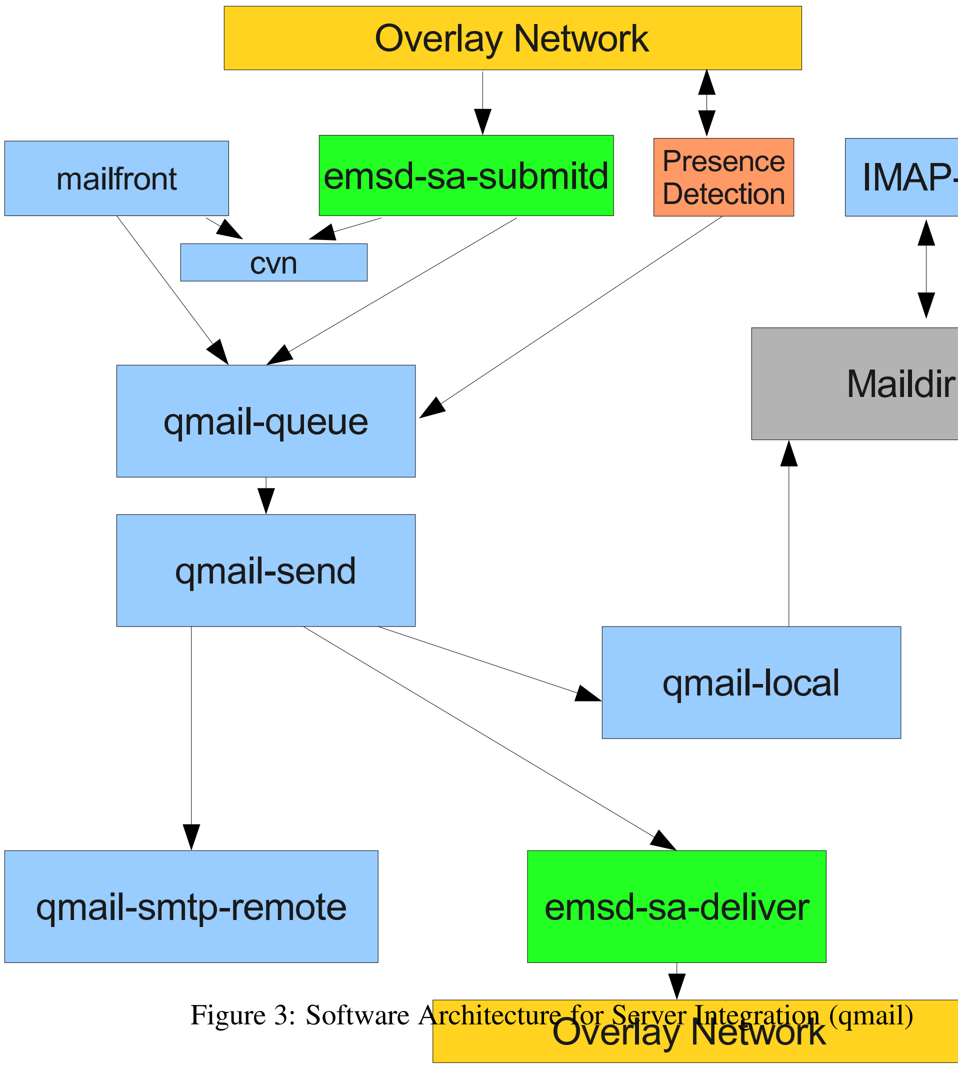

Figure 3: Software Architecture for Server Integration (qmail)

the feasibility of reusing or modifying the queuing and spooling system of qmail, in conjunction with lower-layer presence detection for immediate EMSD delivery.

Since all software shown in Figure 3 is free/Libre Software, EMSD-enabled MTAs can be deployed by anyone wishing to implement Libre Texting.

6.4 Device software integration

Towards our goal of widespread industry adoption, the Libre Texting protocols must fit in a straightforward way into many end user devices, running a variety of Mail User Agents (MUAs). To facilitate this there must be minimal disruption of the existing MUA architecture.

We use an architecture based on the concept of a **Device-Resident End-MTA** middleware module, acting as intermediary between the protocol software and the MUA.

Figure 4 shows the software architecture for integration of EMSD-UA with qmail to create a Device-Resident End-MTA. On its external interface (shown in grey and yellow at the bottom of the figure), the Device-Resident End-MTA interacts with the Internet at large using EMSD, SMTP, and IMAP. On its internal interface (local loop-back interface; address 127.0.0.1) the Device-Resident End-MTA interacts with the MUA based on SMTP and IMAP. Thus the MUA need have no awareness of EMSD at all. This architecture is quite general and can be used on almost all platforms. In this model, the MUA is always configured for the 127.0.0.1 interface for the SMTP gateway, and the IMAP server. The Device-Resident End-MTA is then configured with the real external server information.

offlineimap is used to optionally synchronize the device's mailstore/Maildir (shown in grey) so that the user's inbox is locally available, even when there is no network connectivity.

Though this architecture is based on qmail, the resulting Device-Resident End-MTA package is quite general, and can be installed in all Linux PDA platforms, and very likely other platforms too.

Note that because all software shown in Figure 4 is free/Libre software, the Device-Resident End-MTA can be made available on any Linux-based device without any restrictions.

Part IV
Libre Texting Components & Choices

7 Libre Texting Components & Choices

As an open solution, Libre Texting unbundles the Texting technology chain. Figure 5 shows the major technology components, and some of the choices presently available. In general, these components are now decoupled, allowing independent competition and selection for each component.

Some technical dependencies exist between Wi-Fi Routers and Networks (e.g. Novatel MiFi, tied to the 3G cellular networks), but as time goes on we can expect these components to become increasingly decoupled also, as Routers become general-purpose multi-network wireless companion devices.

8 Libre Texting benefits

Libre Texting provides major benefits for the industry and the end user.

As an open solution, Libre Texting has the effect of unbundling the Texting application, exposing every point of the Texting technology chain to free market entry and competition. Furthermore, the Libre model opens the Texting industry to the powerful generative dynamics of the free software development model.

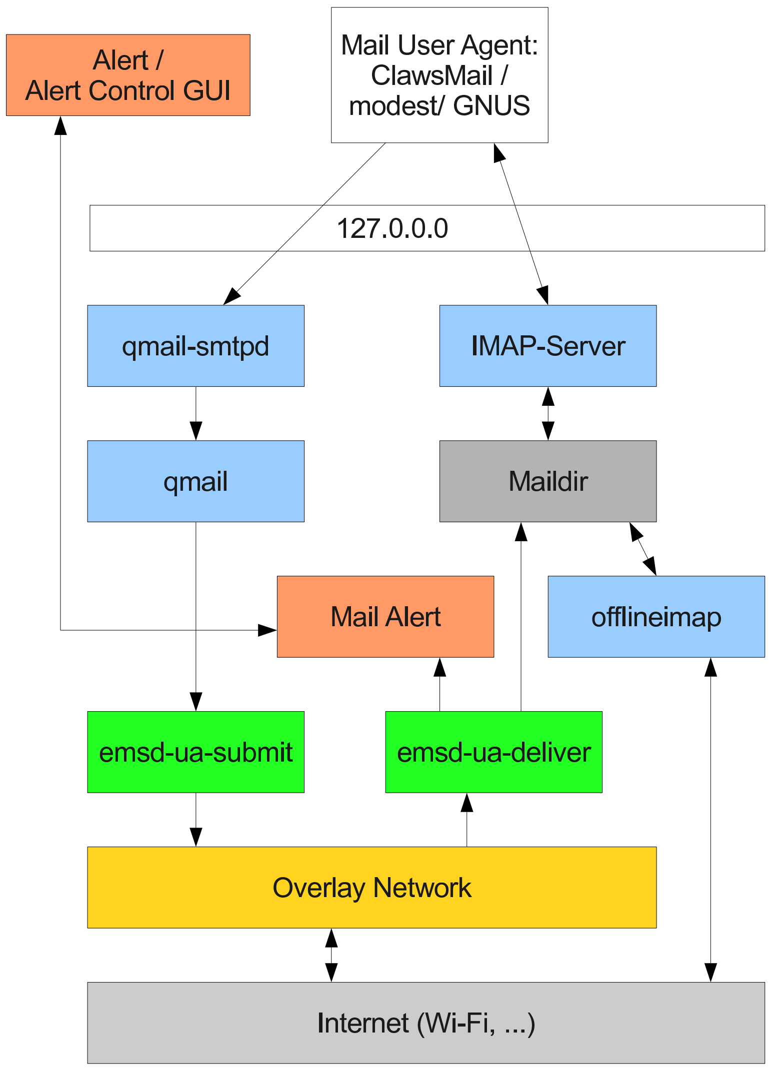

Figure 4: Software Architecture for Device Integration

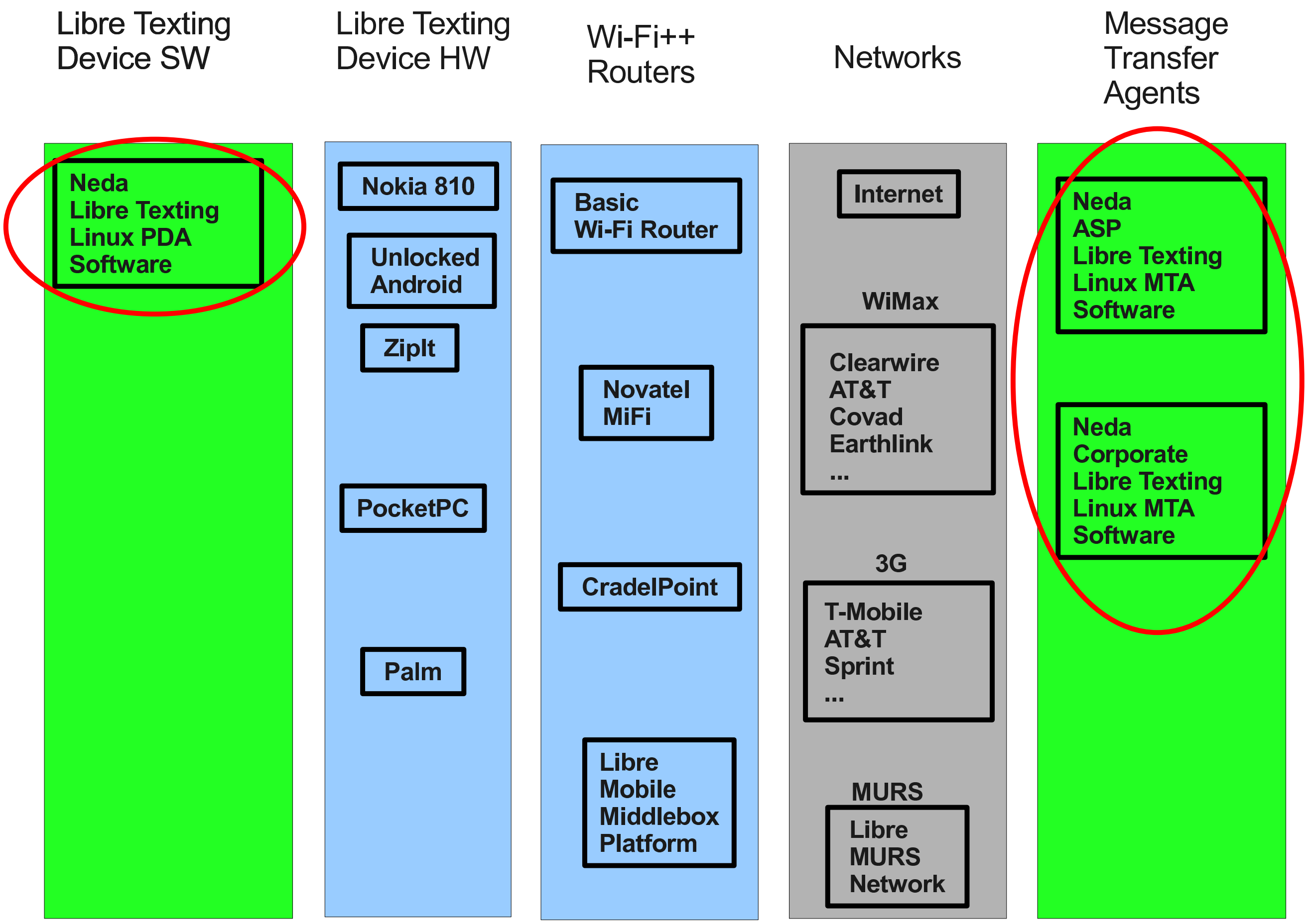

Figure 5: Libre Texting: Components and Choices

This results in greatly increased business opportunities, technical innovation, and unrestricted industry growth. These dynamics promote constantly increasing richness of features and functionality, in ways we can imagine today, and ways we cannot. One entire branch of technical innovation we can readily imagine consists of a new class of applications residing on top of Libre Texting. For example: caller ID information can be pushed directly to the mobile user, for a telephone call received on the land line.

All this will directly benefit the end user, who will have greatly expanded range of choice, increased features and functionality, mix-and-match component selection—and all this, almost certainly, at more competitive prices.

The Libre model also provides assurances of a number of critical freedoms and civil liberties, including privacy and freedom of speech. These liberties are certainly not assured under the proprietary model.

9 History and Related Work

Here we describe some previous and on-going work related to the present initiative.

9.1 personal Air Communications Technology (pACT)

In 1995 AT&T Wireless Services began development of a wireless messaging system called *personal Air Communications Te*
The spectrum for pACT was Narrowband PCS, and the lower layers of pACT were based on CDPD technology [3], the first nationwide native mobile IP network. The pACT messaging protocols were designed for efficient IP-based mobile messaging/texting. In effect, pACT was functionally equivalent to the popular mobile email solutions of today such as BlackBerry.

Neda Communications played a major role in the development of the pACT system. In particular, Neda engineering staff were the primary architects of the mobile messaging component of pACT. Previous to this these same engineering staff had played an active role in the development of CDPD.

AT&T spent about $500M on the development of pACT, including $160M for the purchase of nationwide Narrowband PCS licenses. But then in March 1997 AT&T abandoned the pACT wireless messaging project entirely, and elected not to maintain or further pursue any of the pACT technology.

But recognizing the significance and importance of this, Neda completed development of the protocols independent of AT&T, and published them as RFC-2188 (1998) [13], and RFC-2524 (1999). The primary author of these RFCs has made patent-free declarations for both protocols through the Free Protocols Foundation.

Lemonade

Since 2003 the Lemonade working group at the IETF has been attempting to provide a set of enhancements and profiles of Internet email submission, transport, and retrieval protocols to facilitate operation on platforms with constrained resources, or via communications links with high latency or limited bandwidth. A primary goal of this work is to ensure that those profiles and enhancements continue to interoperate with the existing Internet email protocols in use on the Internet, so that these environments and more traditional Internet users have access to a seamless service.

Various other groups are also active in this area, including:

- 3GPP TSG T WG2 SWG3 Messaging,
- W3C Mulitmodal interaction Activity,
- Open Mobile Alliance,
- 3GPP2 TSG-X.

All these groups are dominated by existing vested interests and are therefore unlikely to be ready or willing to accept the Libre model in full.

Libre Texting as presented in this proposal is not in conflict with Lemonade [17] [16] [14], but with respect to initial submission and final delivery of messages takes a more pragmatic and efficient approach.

The EMSD protocols address the key missing features (push-mode delivery and efficiency) needed for mobile email, while Lemonade attempts to address various peripheral features.

Push-IMAP

Push-IMAP (also known as P-IMAP, or Push extensions for Internet Message Access Protocol) is based on IMAPv4 Rev1 (RFC 3501) [15], but contains additional enhancements for optimization in a mobile setting. Push-IMAP was not included in the Lemonade Profile (RFC 4550) [17], and is only available as an internet-draft.

Though they are both based on IMAP, neither Yahoo Mail nor MobileMe for iPhone uses a standard form of Push-IMAP. Yahoo Mail uses a proprietary extension to the IMAP protocol, and Apple's MobileMe uses a server within Apple that maintains a persistent IP connection to each iPhone, which allows push email.

We believe that our use of EMSD for Libre Texting has a number of advantages over the Push-IMAP approach. These include: superior efficiency [1], NAT traversal by means of an Overlay Network rather than maintaining a persistent TCP connection, and native push-mode delivery instead of extending IMAP in a non-standard way.

Part V
Reference Implementation

10 Reference Implementation

10.1 Overview

In order to make the concepts described in previous sections real, we are developing a complete set of Libre Texting software and Libre Texting Services to deliver Libre Texting functionality to the end user as a starting point.

The work on the Reference Implementation is on-going. At this time, use of the reference implementation is experimental.

Libre Texting is implemented in the broader context of Libre Service. The broader Libre Services Integration Platform (LSIP) is heavily used. The Libre Texting Reference Implementation uses the ByStar Libre Services for initial service provisioning.

All of the Libre Texting implementation is based on pure Libre Software. The primary platform for all Libre Texting software is Debian GNU/Linux.

All needed software is available through bysource.org and bybinary.org.

The reference implementation is intended to include no restrictions on use or development. For the reference implementation, we only use public spectrum. The entire software is copyleft. All used protocols are patent-free.

10.1.1 L7 Libre Texting Overview

When wireless connectivity to the Internet is available, then all that is needed is Layer 7 (above Layer 3) capabilities.

The chosen components for the Reference Implementation are:

Mobile Device Hardware: Nokia 810 and Nokia 900 For the initial Reference Implementation we confine the supported hardware as those that can best run Debian GNU/Linux distribution. As of this writing the following devices are supported: Nokia n900, Nokia n810 and Asus.

Device Software See **??** for details.

MTA Software See **??** for details.

VPN Based Mobility: openVPN OpenVPN is a widely used, free and open source virtual private network (VPN) program for server-to-multiclient encrypted tunnels between MTAs and Mobile Texting Devices.

Libre Textting Service: ByStar See 13.4 for details.

10.1.2 L3 Collaborative Wireless Network Overview

In order to address the need for widespread conectivity, as an optional part of the reference implementation we are building a collaborative wireless network based on public spectrum.

The chosen components for the Reference Implementation are:

Public Spectrum: MURS and WiFi Multi-Use Radio Service (MURS) is a two-way radio service.

MURS Collaborative Network Infrastructure:

MURS Base Station (Collaborative Propagation):

Mobile MURS/WiFi Router:

CalAmp Dataradio router/modem/radios has been used successfully.

10.2 Usage

The reference implementation is intended to be immediately usable by anyone.

Getting started on use of Libre Texting is as simple as following these steps:

10.2.1 Service Usage

- Choose a Device (e.g., Nokia 810) – Or buy an integrated device
- Get A ByName Account
  - Go to http://www.byname.com
  - Get Yourself an Account
- Download Your customized preconfigured software into your device
- Get Internet connectivity and start using your device

10.3 Libre Software Components

10.3.1 Neda EMSD-UA Stack

Neda EMSD-UA is dual licensed as GPL and also as Neda commercial license. Those interested in obtaining the Neda EMSD-UA software in source form can obtain it at http://www.bysource.org/

As a debian package Neda EMSD-UA is available at: http://www/bybinary.org

10.3.2 Neda EMSD-SA Stack

Neda EMSD-SA is dual licensed as AGPL and also as Neda commercial license. Those interested in obtaining the Neda EMSD-UA software in source form can obtain it at http://www.bysource.org/

As a debian package Neda EMSD-SA is available at: http://www/bybinary.org

10.3.3 Device MUA – ByStar Emacs Gnus

The architecture of a Device-Resident-MTA permits incorporation of any Mail User Agent (MUA) on the device.

In order to create a very rich environment for experimentation with relevant MUA features and integration, we have chosen Emacs-Gnus.

Additional packages that make use of Emacs-Gnus possible for texting can be obtained at http://www.bysource.org/

10.4 Device Software Integration

Software components needed for integration on the device include:

- Maemo (Debian GNU/Linux derivative)
- Neda Libre Services Integration Platform
- emacs23, Gnus, EOE
- autoAP
- openvpn
- NEDA-EMSD-UA
- qmail, courier-imap, maildrop, offlineIMAP
- ssl, tls
- mailAlert, mplayer
- MessageScreeningUI

10.5 MTA Software Integration

Software components needed for integration on the MTA include:

- Debian GNU/Linux
- Neda Libre Services Integration Platform

- openvpn
- NEDA-EMSD-SA
- qmail, mailfront, cvm, courier-imap, maildrop
- apache
- squirrelmail
- ssl/tls/https for the above

10.6 Mobile MURS/WiFi Router Software Integration

Hardware

In order to build an MURS/WiFi Router you need the following hardware components.

- WiFi with the Atheros Chipset
- CalAmp DataRadio MURS

Software

In order to build an MURS/WiFi Router you need the following software components.

- Debian GNU/Linux
- Neda Libre Services Integration Platform
- Master Mode WiFi Driver
- hostap-utils.deb hostapd.deb
- dhcpserver
- ferm (iphosts nat)
- router mode

10.7 MURS Basestation Software Integration

Hardware

In order to build an MURS/WiFi Router you need the following hardware components.

- CalAmp DataRadio MURS

Software

In order to build an MURS/WiFi Router you need the following software components.

- Debian GNU/Linux
- Neda Libre Services Integration Platform
- Master Mode WiFi Driver
- hostap-utils.deb hostapd.deb
- dhcpserver
- ferm (iphosts nat)
- router mode

10.8 Spectrum

For the reference implementation, we only use public spectrum.

10.9 MURS Spectrum

In the United States, the Multi-Use Radio Service (MURS) is a two-way radio service consisting of five frequencies in the VHF spectrum regulated by Title 47 of the CFR (Code of Federal Regulations) Part 95. Established by the U.S. Federal Communications Commission in the fall of 2000, MURS created a radio service allowing for ”licensed by rule” or ”permitted by part” (Part 95) operation (rather than permitted by individual license), with a power limit of 2 watts, four times that of FRS radio.

MURS comprises the following five frequencies:

```
Frequency  Authorized bandwidth
151.820 MHz  11.25 kHz
151.880 MHz  11.25 kHz
151.940 MHz  11.25 kHz
154.570 MHz  20.00 kHz
154.600 MHz  20.00 kHz
```

These are very narrow bands and the efficiencies provided by EMSD are therefore critical.

Part VI
Libre Texting Open Business Plan

11 Business Consequences

The Libre Texting model has immense business consequences.

Libre Texting represents a radical shift of the Texting industry to the *non-proprietary, for-profit quadrant*. This can be expected to cause a major industry reconfiguration, with significant winners and losers. The losers are the existing vested proprietary interests, whose economic hegemony vanishes. But the winners are the many more companies who can now enter the Texting market—and the end-user who benefits from the resulting competition.

12 Business Plan Reader Assumptions

We assume appropriate background knowledge on the part of the reviewers. In particular, we assume:

- An understanding of the fundamental dynamics of large-scale, protocol-based industries such as email and texting. In particular, an understanding of the very strong forces of convergence towards a unified set of protocols for industry-wide interoperability.
- An understanding of the tremendous power of the non-proprietary model, as exemplified by free software and other non-proprietary constructs.
- An understanding of the particular mechanisms of business operation in the *non-proprietary, for-profit* quadrant. Note that the Venture Capitalist philosophy and belief system is fully wedded to proprietary ownership of assets as a fundamental business paradigm. This belief system is in total conflict with the notion of a non-proprietary business construct.
- An understanding of the walled garden model and its characteristics. In particular, an understanding of the characteristics of the existing, closed and proprietary wireless service provider model. These include a stifling of engineering creativity, a distortion of the competitive business environment, and perhaps most important, a hazard to basic civil liberties such as privacy, freedom of information, and freedom of speech.

  And so conversely, an understanding of the essential societal benefits of replacing the proprietary walled garden model with the non-proprietary end-to-end model.

All these are critical underpinning concepts, essential for proper understanding of this proposal. In addition there is of course also the need for domain-specific knowledge of Mobile Messaging, Internet Email Protocols, and GNU/Linux software architecture.

13 Making Libre Texting Widespread

The Libre Texting model can be thought of as comprising two distinct elements: the adoption component, and the commercialization component. The adoption component is about making Libre Texting widespread. The commercialization component is about profiting from this. In this section we discuss the adoption dynamics; in the next section we discuss the commercialization mechanisms.

An essential component of the Libre Texting model is the logic and dynamics of how it will become widespread.

Today's mobile messaging landscape consists of ferocious competition among a multiplicity of solutions, all residing wholly within the confines of the proprietary ideological context.

Our proposal stands separate and distinct from all that. In terms of functionality, Libre Texting provides nothing new. It does not fall in the address-a-functional-need category. It provides the same functionality, but under a model that is vastly more potent than the proprietary model. This potency rests ultimately on the tremendous propagatory power of the non-proprietary model, which *removes all barriers and frictions from development, deployment, and usage at every point within the messaging framework*.

Every element of the Libre Texting model—the protocol design, the proposed software architecture, and the service delivery model—has been designed with a critical goal in mind: to enable its widespread propagation. Libre Texting has been endowed with all the necessary characteristics for it to emerge as the global Mobile Messaging industry standard, in use planet-wide, to the exclusion of all proprietary messaging solutions. The key dynamics to achieve this are:

- Eliminate all restrictions to deployment, participation and usage of Libre Texting at each point within the mobile messaging technology chain. The Libre Texting solution must spin within a frictionless bearing.

- Make the Libre Texting solution fully compatible with the existing messaging infrastructure.

Libre Texting includes the following components to achieve this:

- Completely open and patent-free protocols
- Free software for devices
- Free software for MTAs (Message Transfer Agents)
- A network architecture compatible with the existing messaging infrastructure
- An initial, in-place, easy-to-enroll Libre Texting service

13.1 Open and patent-free protocols

The Libre Texting protocols are patent-free, so there are no restrictions on their deployment and usage by anyone. Any company, organization or individual can implement the protocols without incurring licensing fees or other financial obligation.

As well as being patent-free, the Libre Texting protocols are also totally unrestricted. They have been published as Internet RFCs, thus ensuring that they remain freely and permanently available within the public domain.

Historical precedents

All this is in accord with the fundamental dynamics and history of the data communications industry. The Libre Texting proposal exists within a particular historical context. And part of this context is a number of historical industry events, that are essential to understanding the how's and why's of Libre Texting.

A particularly apposite historical event is the evolution of email, and the dynamics of that evolution. In 1992 or thereabouts, the then-fragmented email industry began to converge on a set of non-proprietary protocols. At that time the email landscape was dominated by a number of large proprietary players, most notably IBM's PROFS system, DEC's All-in-One, Lotus Notes, Microsoft Mail and various X.400 products. All proprietary, and all component-wise non-interoperable. These were able to interconnect through various gateways, but the fragmentary, proprietary, competitive and non-interoperable industry landscape proved fatal. In a matter of two years or so, all email solutions converged on SMTP/POP/IMAP, and all proprietary solutions disappeared. The convergence point was non-proprietary, and it was not driven by big business. SMTP (Internet email) provided no new capabilities and met no new need. But it reshaped the email medium. Note the word: reshaped. Internet email is now the global electronic mail standard, in use planet-wide, to the exclusion of all other electronic mail solutions.

Today we are presented with a a starkly equivalent landscape in the mobile messaging arena, and precisely the same fundamental dynamics are at work. We have SMS, Blackberry, Apple's iPhone mobile email, Palm, and multiple others. All proprietary, and all component-wise non-interoperable. As in 1992, these are able to interconnect through various gateways. But the fragmentary, proprietary, competitive and non-interoperable mobile messaging landscape is as fatal today as the wired messaging landscape was in 1992, and for exactly the same reasons.

This and other historical events are part of a pattern, which is itself a manifestation of a set of dynamics that have repeatedly driven convergence of major data communication services towards a unified protocol. These forces of convergence lead to a winner-takes-all dynamic, just like SMTP/POP/IMAP, as the service matures.

We have studied the history and characteristics of succesful protocols, and we understand it. And we have created the Libre Texting model based on that understanding. In 2000 we wrote a white paper titled "Lessons from History: Comparitive Protocol Case Studies," [6], where we identify and analyse the critical enabling characteristics of convergence-point protocols.

In 2008, the IAB (Internet Architecture Board) published RFC-5218, [18], titled, "What Makes for a Successful Protocol?" The IAB document essentially repeats the same material we wrote in our 2000 paper.

Knowledge of the historical context and an intuitive understanding of the underlying dynamics is a prerequisite for understanding the logic and viability of Libre Texting. A reviewer without this understanding is not qualified to assess this proposal.

Our proposal is about convergence on a set of non-proprietary end-to-end protocols for mobile messaging, in precise analogy with the 1992 history. The proposed convergence point is non-proprietary, and creates a gigantic business opportunity for those shaping the convergence.

13.2 Free software for devices

Implementation of Libre Texting at the device end is based on a Device-Resident End-MTA architecture. This is an important component of our design-for-propagation principle. This architecture allows Libre Texting capability to be implemented as a straightforward add-on to existing Mail User Agents (MUAs). There is no disruption to the existing MUA landscape at all, so that best-of-breed MUAs can be used for Libre Texting without no modification.

The Device-Resident End-MTA package resulting from this proposal is quite general, and can be installed in all Linux PDA platforms, and very likely other platforms as well.

13.3 Free software for MTAs (Message Transfer Agents)

The same design-for-propagation principle applies at the message center end. The software architecture for integration of Libre Texting into existing Message Transfer Agents (MTAs) involves inclusion of minimal new software, allowing straightforward integration of Libre Texting into the existing messaging infrastructure.

The Libre Texting MTA package resulting from this proposal is also quite general, and immediately applicable to many existing mail servers.

13.4 Starting point Libre Texting service: part of ByStar services

The final piece required for widespread usage is an initial service to deliver Libre Texting functionality to the end user. We have our own service in place to address this requirement: the ByStar family of services [9], providing Libre Texting as a standard feature. As part of our strategy to promote unrestricted and widespread usage of the service, the ByStar services will initially be deployed under a no-cost model.

The ByStar services are in fact much broader in scope than mobile messaging, providing a comprehensive set of services for individuals and businesses. In particular they provide Libre Texting as part of an integrated suite of messaging capabilities, providing various messaging forms and access methods, both wired and mobile. Hereafter we will call the messaging component of ByStar the **ByStar Libre Texting service**.

14 Business Plan

Libre Texting represents a radical shift of the Texting industry to the *non-proprietary, for-profit quadrant*, causing a major industry reconfiguration, with significant winners and losers. The losers are the existing vested proprietary interests, whose economic hegemony vanishes. But the winners are the many more companies who can now enter the Texting market—and the end-user who benefits from the resulting competition.

Clearly, the commercial potential of this is immense, and certainly not limited to Neda. This can impact, positively, thousands of jobs throughout the industry.

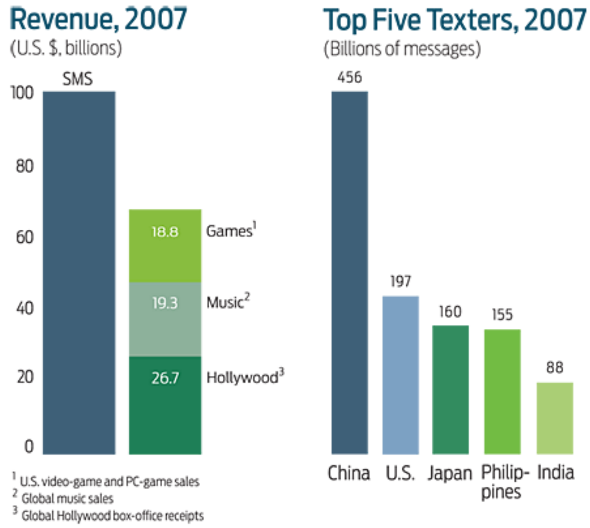

Figure 6: 2007 Texting Numbers

We have already formulated a coherent business plan for our own participation. Deployment of our broad-based Libre Texting services will take place within the context of our existing ByStar Libre Services [9], as part of the existing ByStar Business Plan [10]. This is an Open Business Plan that specifically addresses the dynamics and mechanisms of business operation within the *non-proprietary, for-profit quadrant*.

As an established communications modality, the magnitude of the Texting market is already well characterized. For example see the article titled *thx 4 the revnu* by Steven Cherry in the October 2008 issue of IEEE Spectrum, available at: http://www.spectrum.ieee.org/oct08/6817. This article provides relevant analysis and statistics on Texting usage and market size.

In 2008 the Texting/Mobile Email market was over $150 billion dollars. Putting this in the context of a $150k funding, we see that the market size is about one million times greater than that initial funding. Furthermore, the texting market is a recurring revenue market.

Figure 6 shows the top five texters in 2007 (Sources: Ovum, Quantifica). The growth of texting usage in the U.S. and world wide is not showing any slowdowns

14.1 Revenue sources

Our major revenue streams are described in the following sections. The first two are transient and tactical in nature. The third is the truly enormous business opportunity, and our ultimate strategic goal.

14.1.1 Software licensing (transient, tactical)

We will develop a complete set of Libre Texting software based on two sets of licenses: (1) Free Software Licenses, and (2) Neda Professional Software Licenses.

This dual licensing strategy is well established in the open-source arena. From a business perspective, the Gnu General Public License (GPL) is very restrictive. The GPL generally bars usage under the proprietary model, thus creating a demand for Neda Professional Software Licenses.

14.1.2 Professional and consulting services (transient, tactical)

We anticipate that our Software Licensing revenues will be accompanied by Systems Integration consulting revenues. Neda has a long track record in the Consulting and Systems Integration arena.

14.1.3 Broad-based Libre Texting services (ultimate, strategic)

The ultimate long-term revenue source is of course the Libre Texting services business itself. Our unique leadership role gives us a number of advantages in this arena including: (a) first-mover position in the Libre Texting industry, (b) name recognition as the leader of this initiative, and (c) a highly favorable marketing opportunity in the form of Libre vs Proprietary ideological conflict.

The assets we have built over the past several years, in particular our Data Center and our existing Internet Application Services, leaves us well positioned to realistically target becoming a large-scale Libre Texting service provider.

14.2 Path to commercialization

The proportions of revenues deriving from the three major sources above will shift radically over time. Initially, the majority of Neda's revenues will derive from software licensing and professional services. Though we will provide support for Libre Texting services from the beginning, we do not expect this to be a significant revenue source at the outset.

However, as the industry matures, support for the Libre Texting service (and beyond that, the broader ByStar services) will emerge as the dominant revenue stream, and will eventually eclipse all others. The Libre Texting and ByStar segment of the industry represents the ultimate, major, profit-making opportunity for Neda. The path to commercialization consists of the following steps:

Making Libre Texting widespread (Linux PDAs): We have previously described the general principles of our strategy for making Libre Texting widespread. The execution steps for specific device platforms and operating systems include:

- Include device-side Libre Texting software as available with the following distributions: Maemo 5, Maemo 4, Ubuntu, Debian (and perhaps Android). Note that Maemo 5, Maemo 4, Ubuntu and Debian all use the .deb packaging, and that standard entry into the Debian distribution propagates to other distributions.
- Include MTA Libre Texting software as available with the following distributions: Debian, Ubuntu, Redhat, Centos.
- Promote and support usage among the following MID/phone/netbook communities: Nokia n900, Nokia n810, Android, Asus, and other Debian/Ubuntu based netbooks.
- Support ByStar Libre Texting for the above. Note that standard entry into the distribution of device-side Libre Texting software directs usage towards our ByStar Libre Texting service by default.

This stage of execution generates minimal revenues. However the incremental cost of building and maintaining these software products and services is minimal for Neda—we have much of this in place already.

Note that because of our non-proprietary model, we do not need to partner with any existing players to initiate and promote Libre Texting usage.

Support for ISP/ASP deployment (professional services, plus hosting revenues): Once usage of Linux Mobile Internet Devices for Libre Texting is well established, we will promote inclusion of Libre Texting into existing services of ISPs and ASPs. Details of our business development strategy in this arena is presented in a separate document [11].

Our revenue sources at this stage will consist of consulting services involved in deploying Libre Texting services within the ISP and ASP operating environment. In certain cases we may host the service for ISPs/ASPs in our data center, providing hosting as a further revenue source for us.

Software for proprietary devices (licensing, plus professional services): Once usage of GNU/Linux Mobile Internet Devices for Libre Texting is well established and multiple Libre Texting service providers are in place, we will next focus on enabling Libre Texting implementation on devices with proprietary operating systems.

Devices based on proprietary operating systems such as Windows CE, iPhone, Palm OS, Epoc, etc. provide a licensing revenue source for Neda since they cannot use the General Public License. For these devices we offer commercial Neda Professional Software Licenses. This Software Licensing revenue source will typically be accompanied by revenues from systems integration and other professional services.

Subscription-based, ByStar Libre Texting services: As usage of our ByStar Libre Texting services grows, more of our revenues will be based on recurring subscription service usage model. In terms of revenue characteristics, this is similar to RIM's Blackberry model. This is an ultimate, strategic revenue source.

Advertizing-based ByStar Libre Texting services: As our Libre device software becomes increasingly widespread and reaches larger scale, it then becomes cost-effective to point users by default to the advertizing supported branch of ByStar Libre Texting Services. In terms of revenue characteristics, this is similar to Microsoft's hotmail service.

14.3 Competition: protocols, software, & services

The nature of competition within the Libre context is very different from the proprietary context.

Within the Libre context, it is not possible to maintain sustainable advantage on the basis of proprietary ownership, nor is it possible to create advantage on the basis of functional service differentiation from any other Libre Texting service provider. Any technical enhancement becomes instantly available to all providers throughout the entire Libre environment.

Instead, competition within the Libre environment becomes a matter of which protocols, software implementations and services are used to implement and deliver the service.

With regard to protocols/profiles, we recognize Lemonade (RFC-4550) [17] [16] [14] and Push-IMAP ([15] plus its internet-draft) as potential alternatives to EMSD. However, we believe that the efficiency characteristics of EMSD [1], which are not matched by these IETF-proposed protocols, will prove decisive. The efficiency of EMSD is better suited to Libre Texting, particularly in the case of narrowband wide-area networks.

With regard to free software protocol implementations, alternative and/or overlapping software capabilities are inherently non-competitive, and freely available for integration in our own Neda Libre Texting implementation. Furthermore, our own implementations are accompanied by a coherent business model, which is not the case for most other FOSS projects.

With regard to Application Service Providers (ASPs), we expect that large proprietary services such as Google, MSN and Yahoo will be reluctant and slow to adopt the Libre Texting model, because of their existing business relationships and investment in proprietary solutions.

References Cited

[1] " Mohsen BANAN ". " emsd efficiency ". Permanent Libre Published Content "100003", Autonomously Self-Published, "October" 1996. http://www.freeprotocols.org/PLPC/100003.

[2] " Mohsen BANAN ". " emsd: The leap e-mail component ". Permanent Libre Published Content "100026", Autonomously Self-Published, "August" 2000. http://www.freeprotocols.org/PLPC/100026.

[3] ” Mohsen BANAN ”. ” internetwork mobility – the cdpd approach ”. Permanent Libre Published Content ”120021”, Autonomously Self-Published, ”December” 2007. http://mohsen.banan.1.byname.net/PLPC/120021.

[4] ” Mohsen BANAN ”. ” bystar libre texting proposal for isps/asps and handset manufacturers ”. Permanent Libre Published Content ”110017”, Autonomously Self-Published, ”June” 2009. http://www.neda.com/PLPC/110017.

[5] ” Mohsen BANAN ”. ” libre texting: A collaborative initiative and a reference implementation ”. Permanent Libre Published Content ”110015”, Autonomously Self-Published, ”June” 2009. http://www.neda.com/PLPC/110015.

[6] Andrew Hammoude ” ” Mohsen BANAN. ” lessons from history: Comparitive case studies ”. Permanent Libre Published Content ”100017”, Autonomously Self-Published, ”August” 2000. http://www.freeprotocols.org/PLPC/100017.

[7] Andrew Hammoude ” ” Mohsen BANAN. ” the free protocols foundation policies and procedures ”. Permanent Libre Published Content ”100201”, Autonomously Self-Published, ”March” 2000. http://www.freeprotocols.org/PLPC/100201.

[8] Andrew Hammoude ” ” Mohsen BANAN. ” libre services a non-proprietary model for delivery of internet services ”. Permanent Libre Published Content ”100101”, Autonomously Self-Published, ”March” 2006. http://www.freeprotocols.org/PLPC/100101.

[9] Andrew Hammoude ” ” Mohsen BANAN. ” the by* concept: A unified model for internet services ”. Permanent Libre Published Content ”110001”, Autonomously Self-Published, ”January” 2007. http://www.neda.com/PLPC/110001.

[10] Andrew Hammoude ” ” Neda Communications, Inc. ” the by* family of libre services: The future of the internet services industry an open business plan ”. Permanent Libre Published Content ”110002”, Autonomously Self-Published, ”March” 2009. http://www.neda.com/StrategicVision/BusinessPlan.

[11] Mohsen BANAN Andrew Hammoude ” ” Neda Communications, Inc. ” the by * family of libre services for network service providers: Generic proposal a strategy for rapid entry into the internet application services market a proposal ”. Permanent Libre Published Content ”110005”, Autonomously Self-Published, ”September” 2007. http://www.neda.com/PLPC/110005.

[12] M. Banan. Neda’s Efficient Mail Submission and Delivery (EMSD) Protocol Specification Version 1.3. RFC 2524 (Informational), February 1999.

[13] M. Banan, M. Taylor, and J. Cheng. AT&T/Neda’s Efficient Short Remote Operations (ESRO) Protocol Specification Version 1.2. RFC 2188 (Informational), September 1997.

[14] E. Burger and G. Parsons. LEMONADE Architecture - Supporting Open Mobile Alliance (OMA) Mobile Email (MEM) Using Internet Mail. RFC 5442 (Informational), March 2009.

[15] M. Crispin. INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1. RFC 3501 (Proposed Standard), March 2003. Updated by RFCs 4466, 4469, 4551, 5032, 5182.

[16] R. Gellens. Deployment Considerations for Lemonade-Compliant Mobile Email. RFC 5383 (Best Current Practice), October 2008.

[17] S. Maes and A. Melnikov. Internet Email to Support Diverse Service Environments (Lemonade) Profile. RFC 4550 (Proposed Standard), June 2006.

[18] D. Thaler and B. Aboba. What Makes For a Successful Protocol? RFC 5218 (Informational), July 2008.