1 Identification and Significance of the Innovation

We are proposing 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 goal of the Libre Texting initiative is to provide 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 strategy to accomplish this has two distinct parts: the "model" part, and the technology part.

1.1 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.

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

1.2 The Technology

The technology part consists of the technological innovations 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.
- A new set of messaging protocols, providing 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. The proposed architecture is based on the concept of a Device-Resident End-MTA middleware module, as intermediary between the protocol software and the MUA.

Note that the Libre Texting initiative is not 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 and delivery of this functionality.

Though the model is the critical basis for Libre Texting, it is not the subject of this project. As a starting-point assumption, we take the viability of the Libre model for granted. This research proposal is focused entirely on the above technology components. Specifically, the key focus of the research is to determine feasibility of these technology solutions at large scale.

To sum all this up, the proposal is *to determine feasibility of the critical technological components required for practical implementation of a complete Texting service, based on the Libre model, on a multiplicity of devices, and at very large (planet-wide) scale*.

This, of course, has immense business consequences. Should this proposal prove feasible, we plan to develop the business dimension to profit from our unique leadership role.

2 Background and Phase I Technical Objectives

2.1 Background

The mobile messaging industry of today is a closed, proprietary construct. 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 wireless texting/messaging implementations (telephony SMS, and mobile email solutions such as BlackBerry) 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.

2.1.1 Requisite industry assets

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. Also, 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.

2.1.2 The opportunity

The opportunity is now also in place.

The existing Texting/Mobile-Email industry is dysfunctional and unstable in a number of respects. First, the industry is severely fragmented. In 2009 there are five major mobile messaging players: BlackBerry, AT&T + Apple iPhone, Sprint + Palm Pre, T-Mobile + Google Android, and 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. As additional industry players attempt to muscle their way into the lucrative Texting market, all indications are that this fragmentation will increase.

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 at work 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 gigantic 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 elusive 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.

We believe that the Libre Texting model is ideally constituted to emerge as the decisive point of stability and convergence. In contrast to the existing proprietary incumbents, and any quasi-Libre upstart, 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 in favor of the status quo.

Our goal is to establish Libre Texting as the convergence point and 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 four enabling technological innovations we describe, together with sophisticated engineering and business execution—all this can destroy the existing proprietary regime completely, preemptively stifle any quasi-Libre hijack attempt, and establish Libre Texting as the new industry standard.

The long-term forces towards convergence dictate an ultimate, winner-takes-all scenario. That winner can be us.

2.1.3 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.

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

Thus we are here talking about an email-type service, with a richer functionality than today's telephony SMS.

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

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

- Has an unconscious-carry form factor (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 proposal is that the device takes the form of a typical Wi-Fi-based Mobile Internet Device (MID).

Wi-Fi capability is now near-universal in mobile devices, so that this is by far the most common form of last-leg device connectivity. But Wi-Fi specifically is in no way a requirement, and any form of Internet connectivity will suffice.

In 2009 a large number of Libre Texting Devices are available in the marketplace, including Nokia 800/810, unlocked Android, unlocked iPhone, PocketPC, and others.

2.1.4 The user experience

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, 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. Using the mobile device, she accesses 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 access to the Overlay Network, and for access to her Libre Texting account.

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.

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.

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.

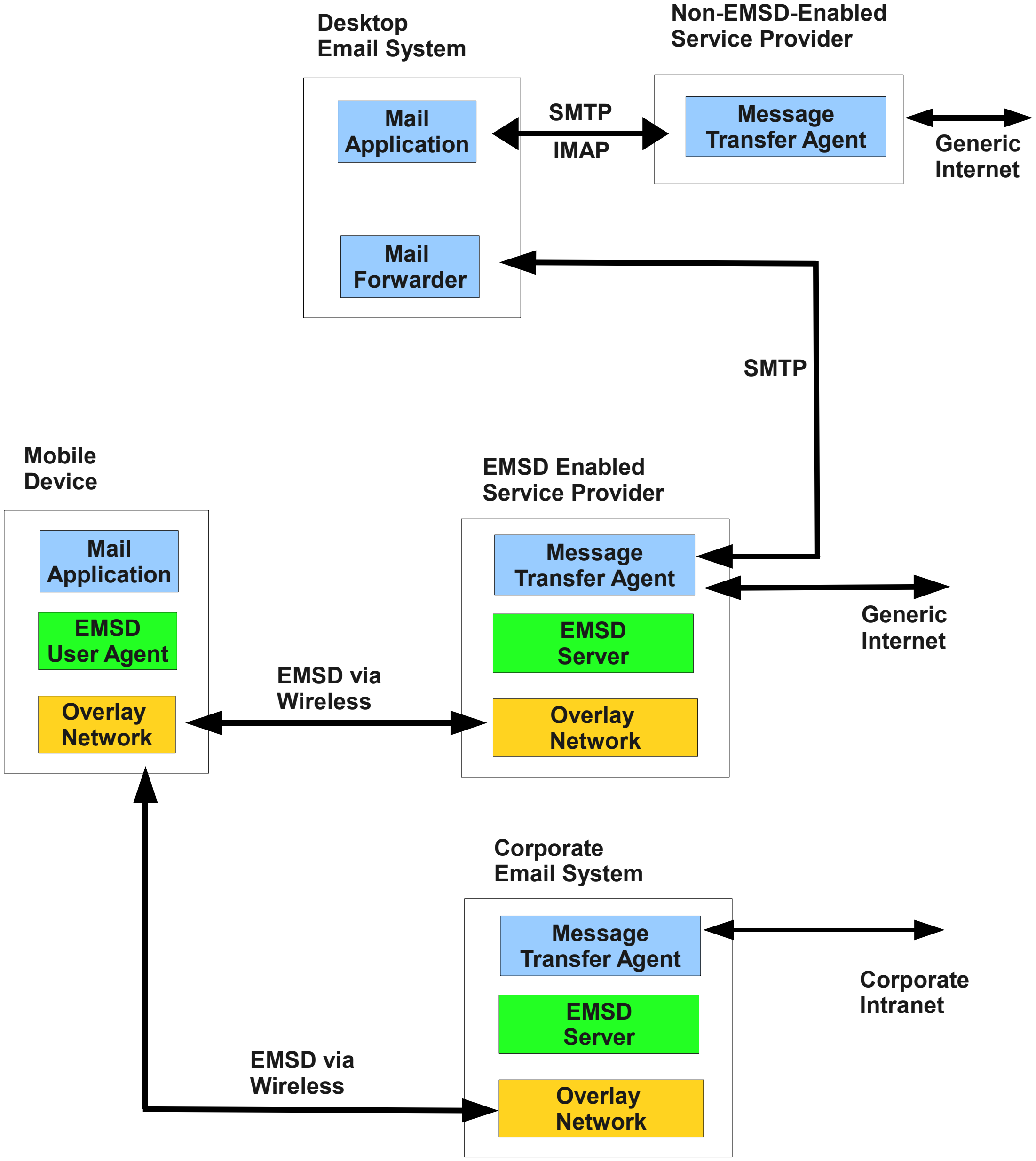

Figure 1: Libre Texting: Functional Operation

2.2 Technology

2.2.1 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. We propose use of an Overlay Network architecture for 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.

Our starting point choice 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 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.

We will validate feasibility of this proposed architecture in this Phase I program.

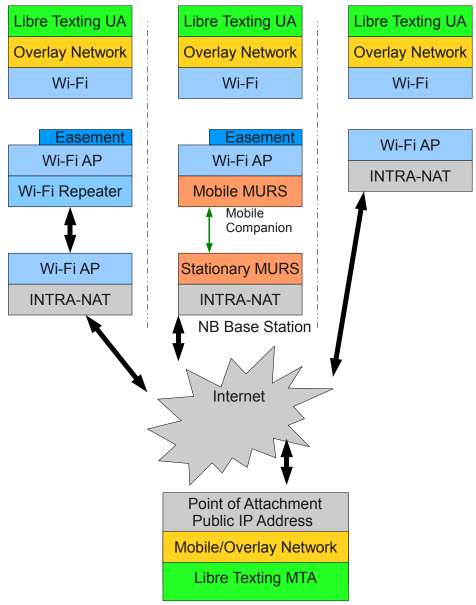

Figure 2: Wi-Fi last-leg Internet connectivity, and the Overlay Network Model

2.2.2 Libre Texting Protocols

The key component of our 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 [?] 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 [?] in 1997.

The Primary Investigator for this proposal (M. Banan) is the primary designer of the protocols, and the primary author of both RFCs.

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

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.

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* [**?**].

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.

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).

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.

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. Identifying a proper and pragmatic set of protocols (a profile) for Texting/Mobile Email and validating their collective interactions with EMSD is an important technical objective of this proposal.

2.2.3 Libre Texting Message Transfer Agent Software

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 proposed software architecture for integration of EMSD-SA into the qmail Mail Transfer Agent. This particular proposed 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 proposed in Figure 3 consists of the usual qmail architecture (shown in blue), with emsd-sa-submitd (shown at the top green) added as a module to process incoming EMSD traffic. The architecture of emsd-sa-submitd will be 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 will be initiated immediately the device becomes reachable.

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 the feasibility

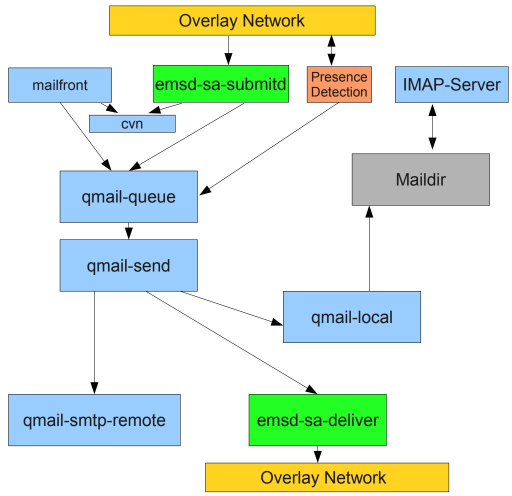

Figure 3: Software Architecture for Server Integration (qmail)

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

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

2.2.4 Libre Texting Mobile Software

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 (MUA). To facilitate this there must be minimal disruption of the existing MUA architecture.

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

In Figure 4 we show a software architecture for integration of EMSD-UA with qmail to create a Device-Resident End-MTA. On its external interface (grey and yellow – lower end), 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 will be used to optionally synchronize the device's mailstore/Maildir (grey) so that the user's inbox is locally available, even when there is no network connectivity.

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

A technological objective of Phase I is to validate correct operation of the Device-Resident End-MTA against a number of existing clients such as modest, Claws Mail, and Emacs GNUS.

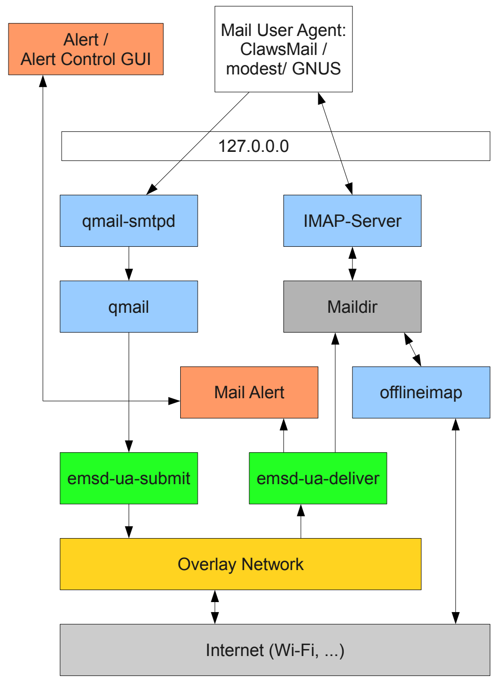

Figure 4: Software Architecture for Device Integration

3 Phase I Research Plan

The objective of the Phase I program is to determine feasibility and validity of all four components of our technological approach: (1) the proposed Overlay Network, (2) the EMSD protocols, (3) the proposed MTA software integration and propagation, and (4) the proposed Device-Resident End-MTA software architecture.

3.1 Research platforms

To carry out the feasibility study, proof of concept, and other tasks described in this proposal we need both server side and device platforms. The server side platform will be Debian GNU/Linux based. The device hardware platform will be Nokia 810 Internet Tablet. The device software platform will be Maemo.

Multi-Use Radio Service (MURS) public narrowband spectrum will be used for validation of Libre Texting over slow, high-latency networks. Verizon/Novatel's MiFi 2200 will be used for validation of Libre Texting over existing 3G networks.

We will publicly release the entire software for the Device-Resident End-MTA and the EMSD-MTA in source code form, subject to GPL, LGPL and AGPL licensing.

3.2 Research Task Areas

For timeline purposes, "Day 0" is when the project is funded, at which time work begins.

3.2.1 Overlay Network Research Tasks

The Phase I research program in this area is designed to validate our assumptions and increase our current understanding of the following aspects of use of the Overlay Network approach.

Scalability: Questions to be answered are: How many Mobile Texting Devices can a single server handle simultaneously? What are the bottlenecks? How reliable is OpenVPN under load? Should OpenVPN prove inadequate for large scale usage, what are other alternatives for building the overlay network? Are Mobile-IPv6 implementations viable?

Presence Detection: Questions to be answered are: How soon after establishment of a tunnel, can presence be determined outside of OpenVNP? What mechanisms for notification can be used? How best can presence information be communicated to the MTA or EMSD-SA?

We will use the server side research platforms to generate load on OpenVPN and conduct studies to answer the above questions.

These research tasks will be conducted by Dr. Hammoude. The work will start at Day 0 and be completed within 3 months.

3.2.2 MUA-EMSD Software Integration Research Tasks

The following aspects of integration of EMSD with Mail User Agent software on the Device will be investigated.

Questions to be answered are: Is the Device-Resident End-MTA Software Architecture shown in 4 practical for widespread deployment on Linux devices? What are practical software and protocol choices for on-demand mailbox synchronization?

We will use the device research platform to build a prototype Device-Resident End-MTA software package. By modelling and empirical verification we will establish the optimum software module configuration for the device. We will then verify correct operation against at least three Maemo-based email clients: (1) modest, Maemo's native Mail User Agent; (2) Claws Mail, a popular alternative; and (3) Emacs GNUS, a sophisticated mail client. Correct functioning of all features against all these MUAs will provide a significant measure of validation for our assumptions.

These research tasks will be conducted by the Principal Investigator (Mohsen Banan). This will be completed no later than 4 months after Day 0.

3.2.3 MTA-EMSD Software Integration Research Tasks

The following aspects of incorporation of EMSD into the existing MTA architectures will be investigated.

Optimum Software Integration Model: We hypothesize that it will be feasible to exclude message queuing and spooling functionality from the software implementation of EMSD, and instead provide this functionality via qmail. By modelling, prototyping and empirical verification we will establish feasibility of this integration approach. We will also identify and eliminate/minimize the causes of end-to-end EMSD + qmail submission and delivery delays.

Delivery Trigger Based on Presence Detection: We will investigate the feasibility of reusing the queuing and spooling system of qmail, in conjunction with lower-layer presence detection (Section 3), for immediate EMSD delivery.

This task will be conducted by the PI (M. Banan). The research will be completed at the end of Phase I.

3.2.4 EMSD and ESRO Protocols Research Tasks

Some aspects of EMSD and ESRO may require enhancements.

EMSD uses ASN.1 for encoding, and currently does not support language tags. We will investigate methods for providing generalized ways of incorporating recent Internet email localization enhancements to EMSD. will be investigated.

ESRO retransmission timer algorithms to minimize adverse link effects are not specified in the base protocol. We will investigate ESRO Congestion Control and related issues.

This task will be conducted by the PI (Mohsen Banan) and Dr. Payman Arabshahi, Consultant. This research task will be completed at the end of Phase I.

3.2.5 Libre Texting Security

Libre Texting is a seamless and consistent extension of the existing Internet email structure. Likewise, in order to provide true end-to-end security over both the wired and wireless Internet, the security mechanisms for Libre Texting must be an integral part of the overall Internet email structure.

The Phase I research program in the area of security is designed to identify protocol and network threats and validate the proposed security mechanisms. The Libre Texting paradigm fully supports the implementation of existing Internet email security mechanisms such as PGP or S/MIME. True end-to-end security can readily be implemented in the context of Libre Texting by means of these technologies. We will investigate mechanisms that can facilitate practical widespread usage of PGP and S/MIME with Libre Texting.

We will use the device side platform to validate the practicality of securing end-to-end email exchange.

These research tasks will be conducted by the PI (Mohsen Banan) and Dr. Payman Arabshahi, Consultant. It will be completed in 6 months.

3.3 Related Research and R&D

Early research and experimental work that supports the ideas and concepts upon which this proposal is based is summarized below.

pACT

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

Neda Communications, and the Principal Investigator (Mohsen Banan), played a major role in the development of the pACT system. In particular, the PI was the primary architect of the mobile messaging component of pACT. Previous to this the PI 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.

Independent of AT&T, Neda completed development of the protocols, and published them as RFC-2188, [**?**], (1998) and RFC-2524 (1999). As the primary author of these RFCs the Principal Investigator 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: (a) 3GPP TSG T WG2 SWG3 Messaging, (b) W3C Mulitmodal interaction Activity, (c) Open Mobile Alliance, (d) 3GPP2 TSG-X.

All these groups are dominated by existing vested interests and are therefore not likely 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 [**?**], [**?**], [**?**], but with respect to initial submission and final delivery of messages takes a more pragmatic and efficient approach.

The proposed EMSD approach addresses the key missing features (push-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) [**?**], but contains additional enhancements for optimization in a mobile setting. Push-IMAP was not included in the Lemonade Profile (RFC 4550) [**?**], and is only available as an internet-draft.

Though they are both based on IMAP, Yahoo Mail and MobileMe for iPhone do not use 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 proposed use of EMSD for Libre Texting has a number of advantages over the Push-IMAP approach. These include: superior efficiency [**?**], better Overlay Network NAT traversal rather than maintaining a persistent TCP connection, and native push-delivery instead of extending IMAP in a non-standard way.

4 Commercial Potential

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.

Regarding our own participation, we do not claim at this point to have all the answers, and we intend to further develop our business strategy in Phase II. But we have formulated our business approach in general terms. Our initial tactical revenue sources will be (1) Software Licensing and (2) Libre Texting Systems Integration. But our strategic long-term revenue source will come from providing broad-based large-scale Libre Texting services.

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 in excess of $150 billion dollars. Putting this in the context of our $150k Phase I SBIR funding application, we see that the market size is about one million times greater than our funding request. Furthermore, the texting market is a recurring revenue market.

Should this research validate the feasibility of our technical approach, the key challenges will be business and engineering execution at very large scale.

As leaders in the new Libre Texting paradigm, our long-term business execution strategy calls for participation in every major component of the Libre Texting technology chain.

4.1 Neda's Software Licensing Strategy

In Phase II we plan to develop a complete set of Libre Texting software available based on two sets of licenses: (1) Free Software Licenses, and (2) Neda Professional Software Licenses.

The dual licensing strategy is now generally well established in the open source arena, and our licensing strategy is not unusual in that sense. From a business perspective, the Gnu General Public License (GPL) is very restrictive. The GPL conflicts with the proprietary model and creates a demand for Neda Professional Software Licenses.

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.

4.2 Broad-Based Libre Texting Services Strategy

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.

We intend to further develop our Broad-Based Libre Texting Services Strategy in Phase II.

4.3 Competition: Protocols/Service/Software

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) [**?**] and Push-IMAP as potential alternatives to EMSD. However, we believe that the efficiency characteristics of EMSD [**?**], 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.

4.4 Company Information and Facilities

Neda Communications, Inc. is an Internet Application Services company. We provide consulting and Internet services to small-to-medium businesses (SMBs) and to individuals. We are a one-stop full-service shop—we maintain our own Data Center, and we provide a full suite of services for clients requiring any sort of Internet presence. Our revenues derive from the customary sources: consulting, website development, hosting, and subscriber service fees.

Neda has a core team of engineering and management personnel with extensive experience in the technical Internet and data communications fields. Among the team there are relationships going back many years, reflecting a long history of productive cooperation.

Of immediate relevance to this project is Neda's Data Center. The Data Center is a state-of-the-art facility built within a secure, fully detached building. About 700 square feet of concrete floor space is available for rack and shelf mount server assemblies and all required support equipment. In all respects (space, power, connectivity, etc.) the Neda Data Center will adequately meet the initial demands of our Libre Texting initiative.

5 Consultants and Subawards/Subcontracts

The services of Dr. Payman Arabshahi as an academic consultant have been secured as part of this effort.

Dr. Payman Arabshahi received his M.S. and PhD in Electrical Engineering from the University of Washington in 1994. He has served on the faculties of the University of Alabama in Huntsville and University of Washington. He is currently a senior scientist with the University of Washington's Applied Physics Laboratory, and on the graduate faculty of the Electrical Engineering Department at the UW. Dr. Arabshahi's expertise Layer 4 of our proposal (ESRO) are sought. His responsibilities and schedule in the proposed research are described in the "Phase I Research Plan"

6 Equivalent or Overlapping Proposals to Other Federal Agencies

NONE. This SBIR represents the first and only proposal of its kind to the US government.