Five Less Taught Topics in Data Communications
University of Washington – EE461 – Spring 2010
Lecture Notes

Document # Records-20100601
June 1, 2010

Slides and Lecture Notes Available on-line at:
http://mohsen.banan.1.byname.net/Records/20100601

Mohsen BANAN
E-mail: http://mohsen.banan.1.byname.net/ContactMe

Contents

List of Figures

Introductions

What Are We Here For?

- Who Am I?
- Why Am I Here?
- Why This Class?
- The Next Two Hours

Who Am I?

I am a data communications engineer.

29 years ago I was a graduate student here at the EE department at UW. Same exact place, but a different building that has now been replaced.

Internet was quite small in 1981.

Why Am Here?

Payman mentioned that he is teaching this class. I tool a look at the course material, suggested a few additional topics to cover and mentioned to him that I'd be happy to add some specific topics in a lecture. Payman suggested today.

I see you as raw material for the Internet engineering profession. Some of you will likely be involved in decisions that can shape tomorrow's Internet. As the early generation of Internet engineers, we have a particular responsibility to protect the integrity of this critical societal resource.

Internet is still in its infancy. The likes of you will likely be in position to shape it.

I want to get you thinking about some specific important topics that are often not discussed enough.

For the most part my goal is not to give you answers. I just want to steer your attention in some particular directions.

For the most part what I'll be talking about in not knowledge. I'll be talking about considerations for applying your knowledge.

Themes

- Example of Email/Messaging Through Out
- Multi-Dimensional Perspective (Not Just Engineering)
- Strategic (Not Tactical)
- Technological Inflection Points and Responsibilities of Engineering Profession
- Everything is online

Many of the topics that we are about to discuss are general and abstract and their discussion can benefit from concrete examples.

I'll be using email and Messaging Protocols as specific examples through out.

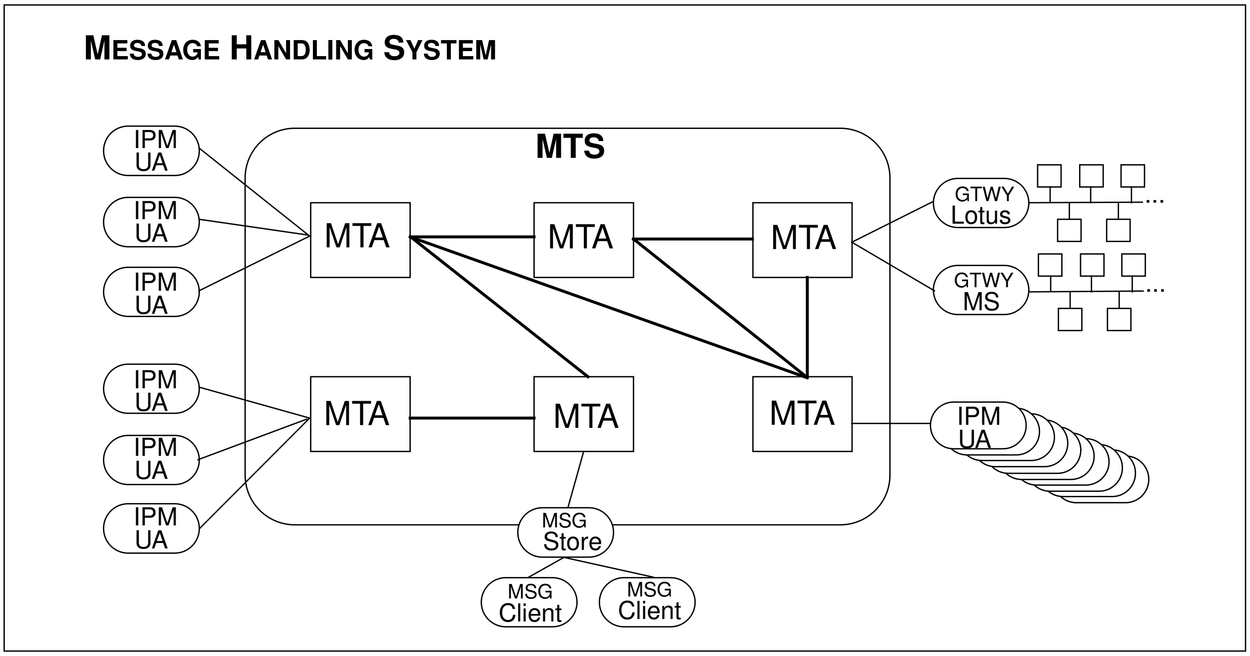

Figure 1: mhsModel

Messaging Model and Terminology

Figure 1 is taken from X.400 reference model.

The basic model for email (Inter-personal Message Handeling System (MHS)) mimics the postal service.

Some Basic MHS (email) Terminology Review

Some Basic MHS (email) Terminology Review

Since we are going to use email as the example through out, let's quickly review some basic terms.

- Message Handling System (MHS) – interpersonal, non-intrusive, either deliver of bounce
- Message Transfer Agent (MTA), Message Transfer System (MTS) – Examples: Sendmail, qmail, exim, MS-Exchange, ...
- Mail User Agent (MUA) – Examples: MS-Outlook, pine, Gnome's Evolution, Emacs' Gnus, ...
- WebMail – A Web Based MUA
- Message Delivery – email is put in your mailbox or pushed to MUA
- Message Submission – Sending email
- Message Retrieval – Example: imap

Part I

The End-To-End Argument

1 The End-To-End Argument

1.1 Relevant Literature

- End-to-end Arguments in System Design http://web.mit.edu/Saltzer/www/publications/endtoend/endtoend.pdf
- Rethinking the design of the Internet: The end to end arguments vs. the brave new world http://www.csd.uoc.gr/~hy558/papers/Rethinking_2001.pdf
- The Rise of the Middle and the Future of End-to-End. http://www.ietf.org/rfc/rfc3724.txt
- Tools for Conviviality - Ivan Illich https://clevercycles.com/tools_for_conviviality

[2] are also included in the References list in article format.

If you were to read one, then go for "Rethinking the design of the Internet".

Ivan Illich was a Catholic priest. His main focus is the right relationship between humans and their tools and society.

E2E Paper Key Point 1

The end to end arguments suggest that specific application-level functions usually cannot, and preferably should not, be built into the lower levels of the system – the core of the network.

Because, the function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the endpoints of the communications system. Therefore, providing that questioned function as a feature of the communications systems itself is not possible.

E2E Paper Key Point 2

Even if parts of an application-level function can potentially be implemented in the core of the network, the end to end arguments state that one should resist this approach if possible. Advantages of moving application-specific functions up out of the core include:

- The complexity of the core network is reduced, which reduces costs and facilitates future upgrades to the network.
- Generality in the network increases the chances that a new application can be added without having to change the core of the network.
- Applications do not have to depend on the successful implementation and operation of application-specific services in the network, which may increase their reliability.

E2E Paper Key Point 3

The most important benefit of the end to end arguments is that they preserve the flexibility, generality, and openness of the Internet. Movement to put more functions inside the network jeopardizes that generality and flexibility as well as historic patterns of innovation. A new principle evident already is that elements that implement functions that are invisible or hostile to the end to end application, in general, have to be "in" the network, because the application cannot be expected to include that intermediate element voluntarily.

1.2 Engineering Summary

- Generally speaking E2E is More Robust
- Dumb Pipe – Honest Pipe (Net Neutrality) – Intelligent Ends
- Care should be taken in identifying the ends
- Not at all absolute – Context of Problem Domain Trumps Principle
- Consider Business Dimensions
- Consider Other Than Business and Engineering Dimensions

1.3 Example: Interpersonal Messaging and E2E Argument

- Email (SMTP) two ends – for E2E purposes
- Texting (SMS) two ends – for E2E purposes
- SMTP MUA Diversity/Choice Vs Texting MUA Diversity/Choice
- SMTP MUA Privacy Vs Texting MUA Privacy
- SMTP Reliability Vs Texting Reliability
- SMTP Spam Vs Texting Spam
- SMTP User Autonomy Vs Texting User Autonomy
- SMTP Cost Vs Texting Cost

- Network-Address-Translators(NTAs) are kind of like diodes – one-way
- Was half of the E2E Argument lost when NATs came in?
- Walled Gardens – NATs and E2E

1.4 Illich's Concept of Convivial Tools

Illich's Concept of Convivial Tools

Tools are intrinsic to social relationships. An individual relates himself in action to his society through the use of tools that he actively masters, or by which he is passively acted upon.

To the degree that he masters his tools, he can invest the world with his meaning; to the degree that he is mastered by his tools, the shape of the tool determines his own self-image. Convivial tools are those which give each person who uses them the greatest opportunity to enrich the environment with the fruits of his or her vision. Industrial tools deny this possibility to those who use them and they allow their designers to determine the meaning and expectations of others. Most tools today cannot be used in a convivial fashion.

1.5 Was: E2E Argument – Now: The Autonomous Services Argument

- End-to-End system perspective is yesterday's topic
- System Perspective vs Service Perspective
- More and More Software is becoming Service
- Central Model vs (Autonomous+Federated Model)
- Central Service Model is profit driven and is counter to autonomy – We have no alternative to Facebook (other than not participating). It is a loosing game.

1.6 Facebook is Very Central and Very Much non E2E – What Are We To Do?

- Create a Viable Alternative Model
- Cultivate it Collaboratively
- I Call It: "The Autonomous Libre Services Argument"

1.7 The Autonomous Libre Services Argument

- Definition: An Internet Service is the capability of running Internet connected software
- Definition: Internet Service is Autonomous, when it is completely at your behest
  - To be Autonomous, it must be a LIBRE SERVICE (requirements of transparency at source code level + reproducibility results in copyleft)
  - To be Autonomous, service context needs to be PORTABLE. (e.g., take your entire Facebook context and move it to a different provider.)
  - To be Autonomous, the service provider must comply with NON-RETENTION of data and service termination at your behest
- Autonomous Services interact with other Autonomous Services directly. (End-to-End).
- Federated Services, operate on public information and on deligated Autonomous Service information segments.

Part II

Characteristics of Successful Protocols

2 Characteristics of Successful Protocols

2.1 Relevant Literature

- Lessons from History of Protocols: Comparative Case Studies http://www.freeprotocols.org/PLPC/100017
- What Makes For a Successful Protocol? http://www.faqs.org/rfcs/rfc5218.html

[1], [3] are also included in the References list in article format.

PLPC-100017 was written 8 years prior to RFC-5218. Some say that IAB is often a day late and a dollar short.

2.2 Mentioned Success Factors

- Positive Net Value (Meet a Real Need)
- Incremental Deployability
- Open Code Availability
- Freedom from Usage Restrictions (Patents)
- Open Specification Availability
- Open Maintenance Processes
- Good Technical Design
- Extensible
- No Hard Scalability Bound

2.3 A Protocol's Position in The Protocol Hour Glass

Figure **??** shows ...

2.4 Other Success Factors

- Leader's Execution Sophistication (Technology, Social, Business)
- Leader's Stamina
- Inclusion In Distros
- Initial Service Inclusion
- The Basic Nature of Protocol (Position in Hour Glass + ...)

Part III

Patents and Protocols

3 Patents and Protocols

3.1 Pointers to Relevant Literature

- The Free Protocols Foundation Policies and Procedures http://www.freeprotocols.org/PLPC/100201
- IETF Policy – http://www.ietf.org/ipr/
- IETF Page of Intellectual Property Rights Disclosures – https://datatracker.ietf.org/ipr/
- "The WAP Trap" – http://www.freeprotocols.org/PLPC/100014

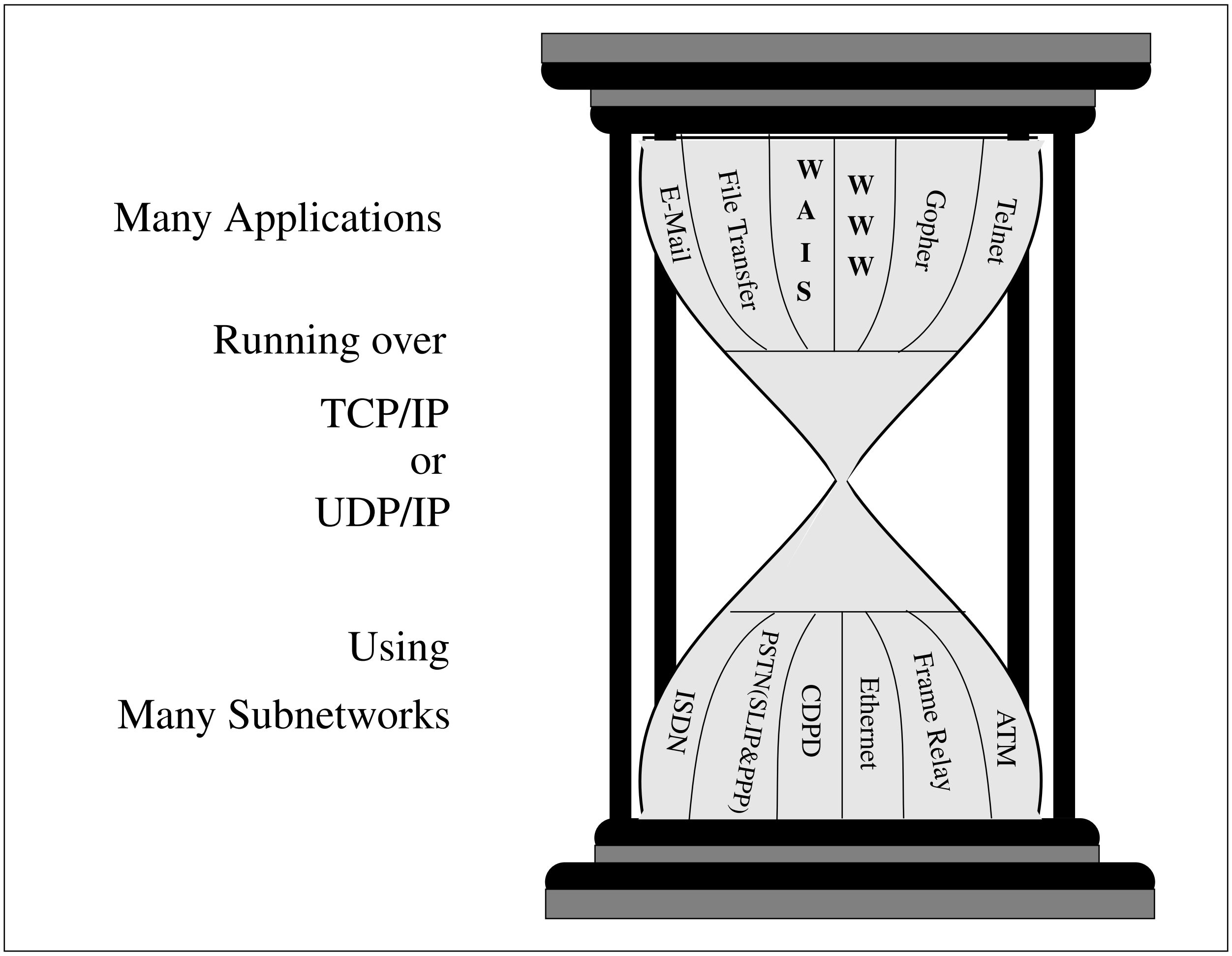

Figure 2: protocolHourGlass

3.2 What Are Protocols? – What Are Patents?

- What Are Protocols: Expected Behavior – Consider a sneeze
- What Are Patents: Exclusive rights to a process – Assignment of Ownership to Knowledge – Restriction on Applying Knowledge
- The Patent Controversy

3.3 How Patents Fit Into Protocols

How Patents Fit Into Protocols

Patents are applied to software, not to protocols. It is not possible to patent a protocol; in general only a process or an algorithm can be patented. However, a protocol may include a patented algorithm as an integral part of its specification. In this case, any software implementation of the protocol requires the use of patented software. That is, a patented process is an inherent part of the protocol.

Even if a protocol does not explicitly decree the use of a specific patented software process, it may still be the case that any practical implementation of the protocol requires the use of patented software components. The protocol could in principle be implemented in a way which avoids the use of patented software; in practice, however, the result would be a significantly inferior implementation, for example in terms of efficiency.

3.4 How Patents Fit Into Protocol Development Process

- Standard Processes (IETF/3GPP/WAP/W3C/...) don't all view patents and protocols the same way.
- WAP was Patented – See "The WAP Trap" – http://www.freeprotocols.org/PLPC/100014
- Standards track ietf protocols are intended to be and remain patent free
- Participation in Working Group, requires declaration of patents
- After Patent Declaration, usually comes a license available on fair, reasonable and non-discriminatory ("FRAND").
- One Big Silly Game!!!

3.5 The Patent Controversy

- I am against patents
- http://mohsen.banan.1.byname.net/PatentAssassin
- Stallman and Knuth are against patents
- Microsoft and Apple are in favor of patents
- The existence of the patent debate is understandable in the industry but not in academia
- Pursuit of knowledge is in conflict with ownership of knowledge

- "Nature of Poly-Existentials as the Basis For or Abolishment of The So-Called Intellectual Property Rights Regime" http://mohsen.banan.1.byname.net/PLPC/120033

Part IV

The Correct Ultimate Protocol Implementation

4 The Correct Ultimate Protocol Implementation

- Program design in the unix environment – Rob Pike gaul.org/files/program_design_in_the_unix_environment.ps
- qmail – http://cr.yp.to/qmail.html
- Dimensions of Correct – Be Conformant, Be Robust, Watch after my interests
- Dimensions of Ultimate – Superior to all peers

4.1 MTA Comparisons – qmail my choice of ultimate MTA

- qmail vs Sendmail vs Exchange vs Postfix vs Exim http://shearer.org/MTA_Comparison
- Linux Modules and Protocol Layers

4.2 The Robustness Principle

Be liberal in what you accept, and conservative in what you send

> Software should be written to deal with every conceivable error, no matter how unlikely; sooner or later a packet will come in with that particular combination of errors and attributes, and unless the software is prepared, chaos can ensue. In general, it is best to assume that the network is filled with malevolent entities that will send in packets designed to have the worst possible effect.

4.3 Unix Philosophy

- Write programs that do one thing and do it well.
- Write programs to work together.
- Write programs to handle text streams, because that is a universal interface.
- Small is beautiful.
- Allow the user to tailor the environment.
- The sum of the parts is greater than the whole.

4.4 The Right qmail Server Configuration

Figure **??** shows ...

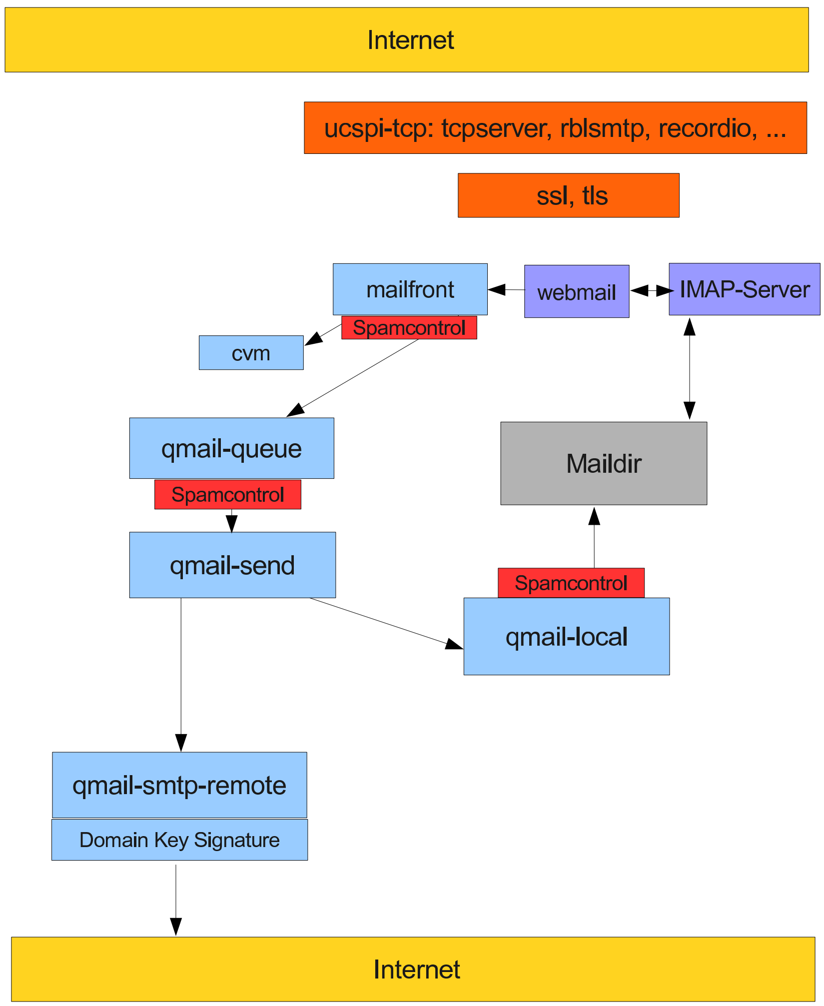

Figure 3: qmail-bystar-wellknown-sa

4.5 The Right qmail Autonomous Client Configuration

Figure **??** shows ...

4.6 qmail Server Configuration Expanded for Mobile Email

Figure **??** shows ...

4.7 qmail Autonomous Client Configuration Expanded for Mobile Email

Figure **??** shows ...

Part V

Some Acknowledged and Some Unacknowledged Internet Errors

5 Some Acknowledged and Some Unacknowledged Fundamental Internet Errors

5.1 Acknowledged: IP Address Exhaustion

- NATs as a side-effect – NATs are quite evil
- E2E deteriorated as a result
- IPv4 to IPv6 – Why has it taken so long?

5.2 Semi-Acknowledged: Longer Term Ramifications of Simple Protocols

- Roughly 90% of email traffic is spam (2009, 2010)
- A culture of patching (flexibility and extendability or patched errors?)
- Insecure protocols and insecure implementations has lead to a lofty Firewall, anti-malware, anti-spam and phishing protection software industry

5.3 Unrecognized: Domain Name Notation Is Backwards

mailto:desk@hommer.1.simposon.byname.net should have been mailto:net.byname.simposon.1.hommer@desk

- Numbers: Most significant digit is to the left
- Phone Numbers: From left to right – country code, area code, exchange, number

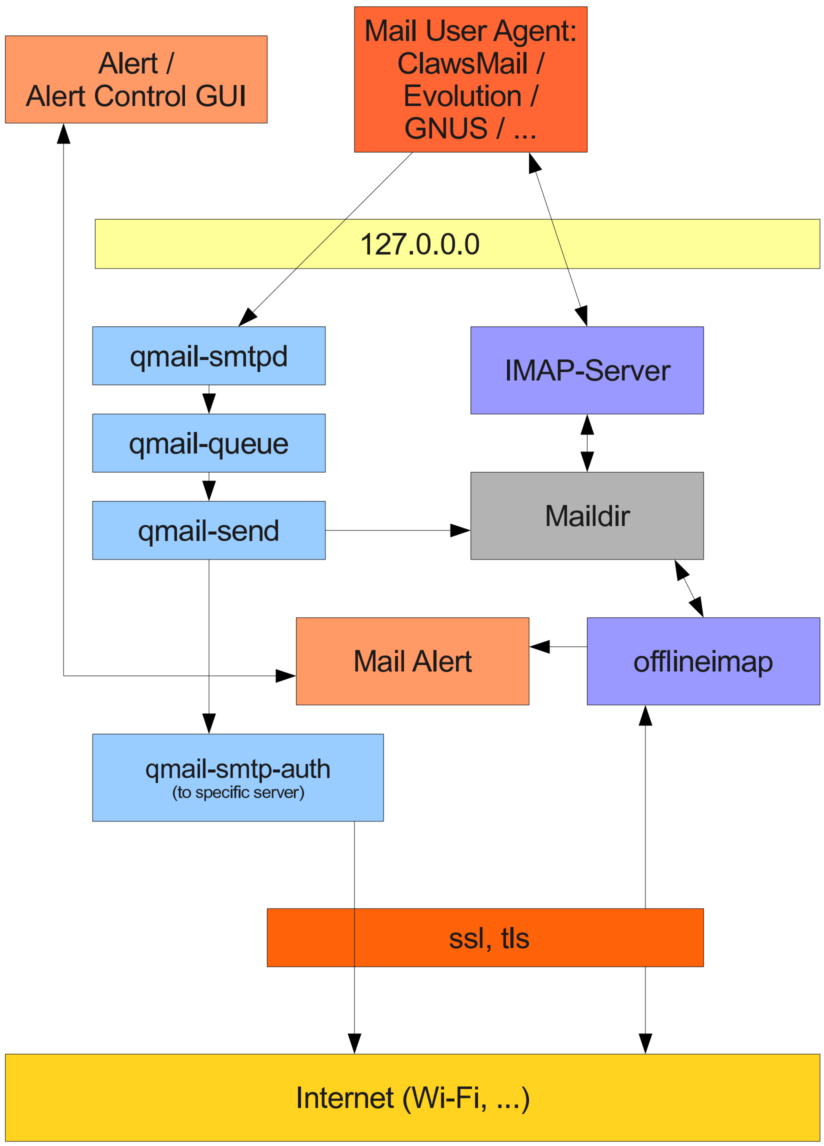

Figure 4: qmail-bystar-wellknown-ua

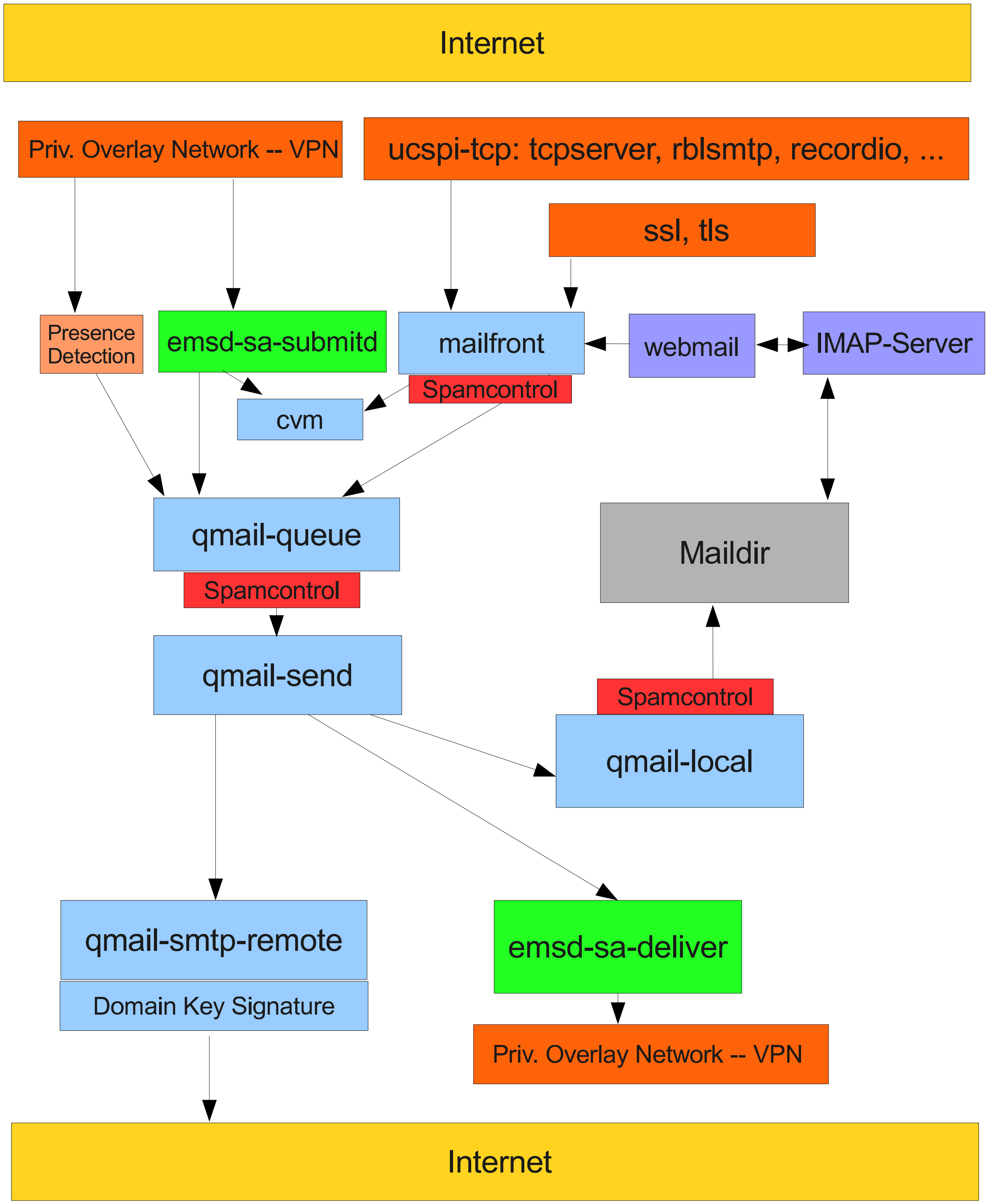

Figure 5: qmail-bystar-emsd-sa

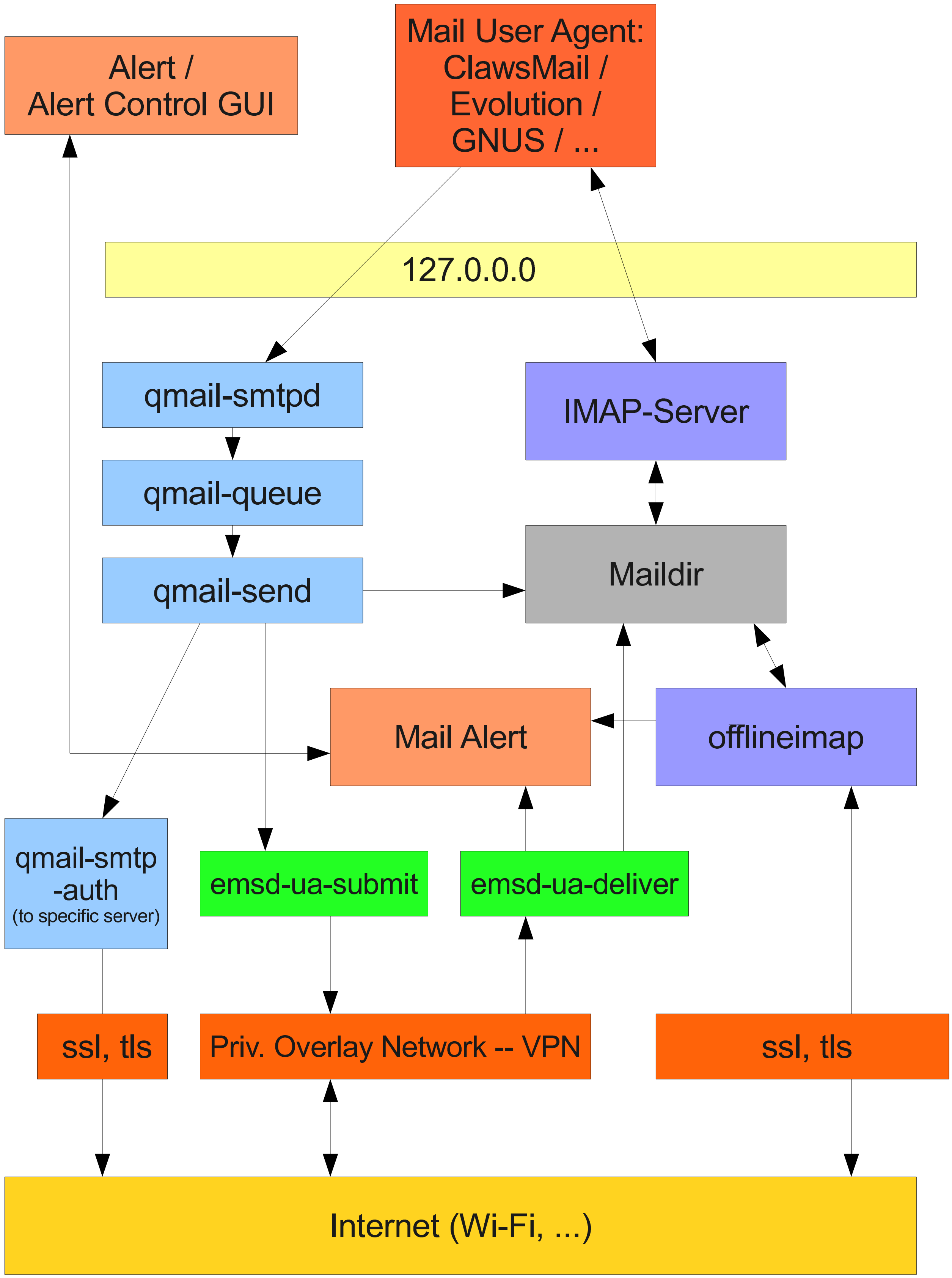

Figure 6: qmail-bystar-emsd-ua

- File System: General to specific from left to right – /var/qmail/control/virtualdomains
- Programming Languages: General to specific from left to right – employee.age=22

We can not complete urls. Actually that is quite important!

Part VI

Wrap Up

Wrap Up

Colophon

- Totally Libre and Copyleft
- No proprietary software used in preparation, presentation and communication of this information
- Slides prepared with beamer-latex
- Presented using Ubuntu-Debian-GNU-Linux and Maemo on PDA
- Served as an Autonomous Libre Service using Debian, Apache, Plone, ...

Questions/Comments/Discussion

References

[1] 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.

[2] J. Kempf, R. Austein, and IAB. The Rise of the Middle and the Future of End-to-End: Reflections on the Evolution of the Internet Architecture. RFC 3724 (Informational), March 2004.

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