



# E-Mail

## SMTP and Postfix



SMTP

Email

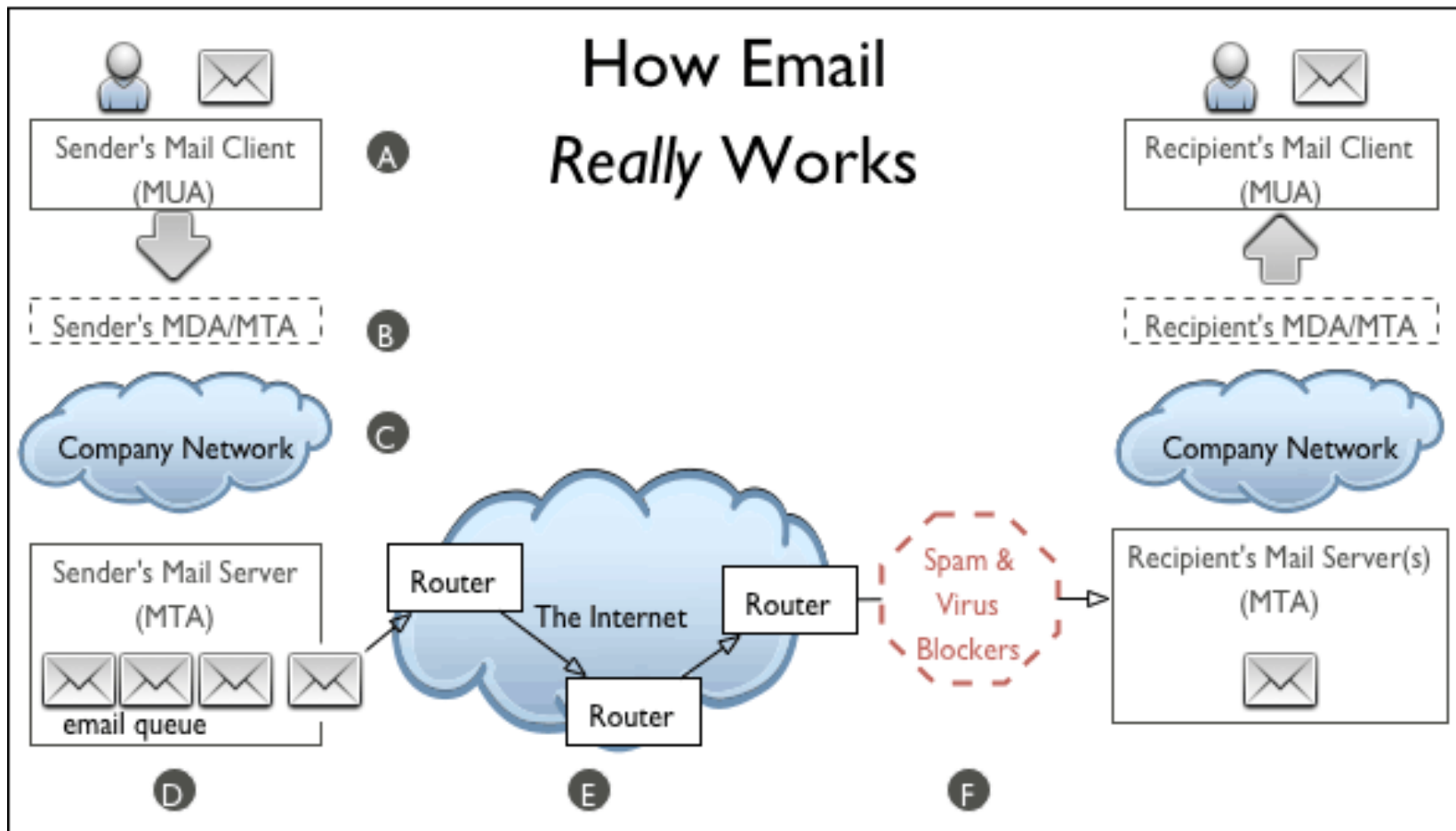
# Scope

- How Email Appears to Work
- How Email Really Works
- Mail User Agent (MUA)
- Message Format
- Mail Delivery Agent (MDA)/ Mail Transfer Agent (MTA)
- Firewalls, Spam and Virus Filters

# How Email Appears To Work

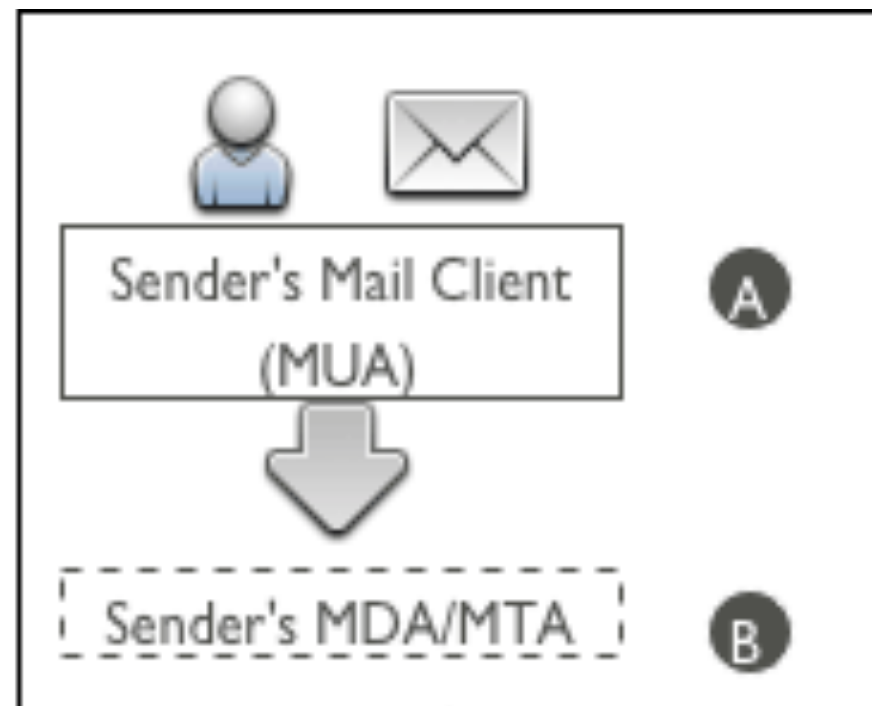


# How Email Really Works



# Mail User Agent (MUA)

- Application the originating sender uses to compose and read email
  - **Pine, MH, Elm, mutt, mail, Eudora, Marcel, Mailstrom,**
  - **Thunderbird, Pegasus, Express, Netscape, Outlook, ...**
- You can have multiple MUAs on one system - end user choice



# Message Format

- **Envelope**
  - Routing information for the "postman"
- **Message Header**
  - Sender
  - Recipients (simple, lists, copies, blind copies)
  - Other fields of control (date, subject)
- **Message Body**
  - Free text
  - Structured document (i.e.: MIME)

# Message Format

From: Philip Hazel <ph10@cus.cam.ac.uk>

To: Julius Caesar <julius@ancient-rome.net>

Cc: Mark Anthony <MarkA@cleo.co.uk>

Subject: How Internet mail works

Julius,

I'm going to be running a course on ...

- Format was originally defined by RFC 822 in 1982
- Now superseded by RFC 2822
- Message consists of
  - Header lines
  - A blank line
  - Body lines



# Message Format

Embedded MUA uses interprocess call to send to MTA

Freestanding MUA uses SMTP to send mail

Headers added by the MUA before sending

From: Philip Hazel <ph10@cus.cam.ac.uk>

To: Julius Caesar <julius@ancient-rome.net>

cc: Mark Anthony <MarkA@cleo.co.uk>

~~Subject: How Internet mail works~~

Date: Fri, 10 May 2002 11:29:24 +0100 (BST)

Message-ID: <Pine.SOL.3.96.990117111343.19032A-100000@taurus.cus.cam.ac.uk>

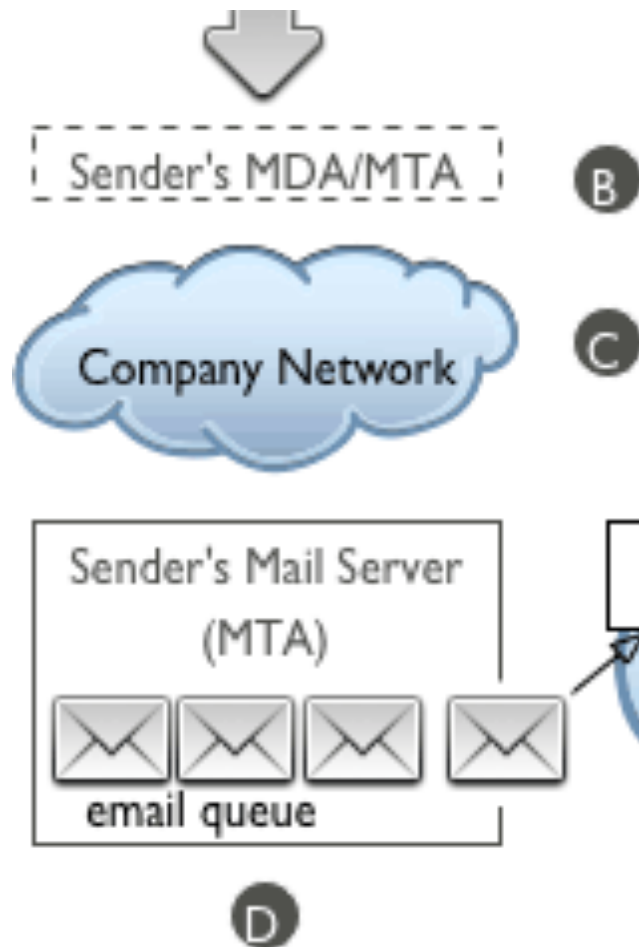
MIME-Version: 1.0

Content-Type: TEXT/PLAIN; charset=US-ASCII

Julius,

I'm going to be running a course on ...

# Mail Delivery Agent (MDA) / Mail Transfer Agent (MTA)



- MDA/MTA **accepts the email, then routes it to local mailboxes or forwards** it if it isn't locally addressed
- An email **can encounter a network cloud within a large company** or ISP, or the largest network cloud in existence: the Internet.

# Mail Delivery Agent (MDA) / Mail Transfer Agent (MTA)

Headers added by MTAs

```
Received: from taurus.cus.cam.ac.uk  
        ([192.168.34.54] ident=exim)  
        by mauve.csi.cam.ac.uk with essmtp  
        (Exim 4.00) id 101qxX-00011X-00;  
        Fri, 10 May 2002 11:50:39 +0100
```

```
Received: from ph10 (helo=localhost)  
        by taurus.cus.cam.ac.uk with local-smtp  
        (Exim 4.10) id 101qin-0005PB-00;  
        Fri, 10 May 2002 11:50:25 +0100
```

From: Philip Hazel <ph10@cus.cam.ac.uk>  
To: Julius Caesar <julius@ancient-rome.net>  
cc: Mark Anthony <MarkA@cleo.co.uk>

...

# Message in transit

- A message is transmitted with an *envelope* :  
MAIL FROM:<ph10@cus.cam.ac.uk>  
RCPT TO:<julius@ancient-rome.net>
- The envelope is separate from the RFC 2822 message
- Envelope (RFC 2821) fields need not be the same as the header (RFC 2822) fields
- MTAs are (mainly) concerned with envelopes  
Just like the Post Office...
- Error (“bounce”) messages have null senders  
MAIL FROM:<>

# An SMTP Session Example

220 server.bluepipe.net ESMTP Postfix

**HELO macbook.catpipe.net**

250 server.bluepipe.net

**MAIL From: <regnauld@x0.dk>**

250 2.1.0 Ok

**RCPT To: <regnauld@nsrc.org>**

250 2.1.5 Ok

**DATA**

354 End data with <CR><LF>.<CR><LF>

**Subject: hello**

.

250 2.0.0 Ok: queued as 41A8B4F5C94

**QUIT**

221 2.0.0 Bye

# SMTP: response codes

- 1xx:positive preliminary answer (action to be continued in subsequent command)
- 2xx:positive response indicating that processing has been carried out as requested
- 3xx:positive partial response: the client must give more data for processing to continue
- 4xx:negative answer, processing is refused, but the command can be tried again later
- 5xx:negative answer, processing cannot be carried out

# ESMTP

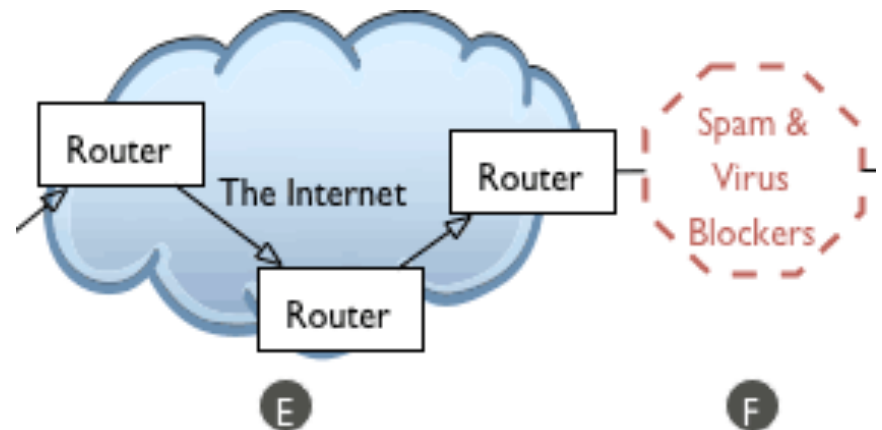
```
220 server.bluepipe.net
ESMTP Postfix
EHLO macbook.catpipe.net
250-server.bluepipe.net
250-PIPELINING
250-SIZE 104857600
250-VERFY
250-ETRN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250-BINARYMIME
250 CHUNKING
MAIL From: <regnauld@x0.dk>
```

...

- Defined in RFC 1651 and following
  - Adds new functionality
    - Transport of 8bit MIME messages
    - Maximum message size limit
    - Function limitation (EXPN, VRFY, ...)
    - Other extensions (pipelining, private extensions)
- The welcome message for ESMTP is EHLO (instead of HELO). In case of a negative answer, the client must revert to the old protocol.

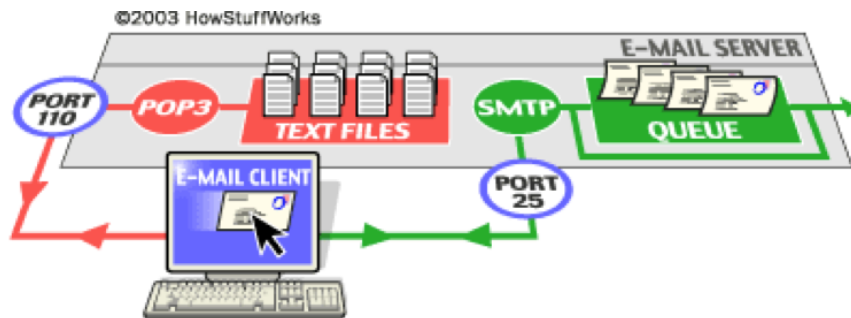
# Network Cloud

- large company network or ISP, or the largest network cloud in existence: the Internet.
- may encompass a **multitude of mail servers, DNS servers, routers, lions, tigers, bears (wolves!)** and other devices and services
- devices may be **protected by firewalls, spam filters and malware detection software** that may **bounce or even delete an email**





# Email Queue



- The email **enters an email queue with other outgoing email messages** .
- If there is a high volume of mail in the queue—**either because there are many messages or the messages are unusually large, or both** —
- the **message will be delayed in the queue until the MTA processes the messages ahead of it** .
- Transient failures will cause mail to stay in the queue until they are fixed for a configurable period of time:
- Permanent failures will cause the MTA to create a bounce message (from mailer-daemon) that gets sent to the original sender specified in the envelope **UNLESS** the sender field there is empty (<>)

image from <http://computer.howstuffworks.com/e-mail-messaging/email3.htm>

# MTA to MTA Transfer

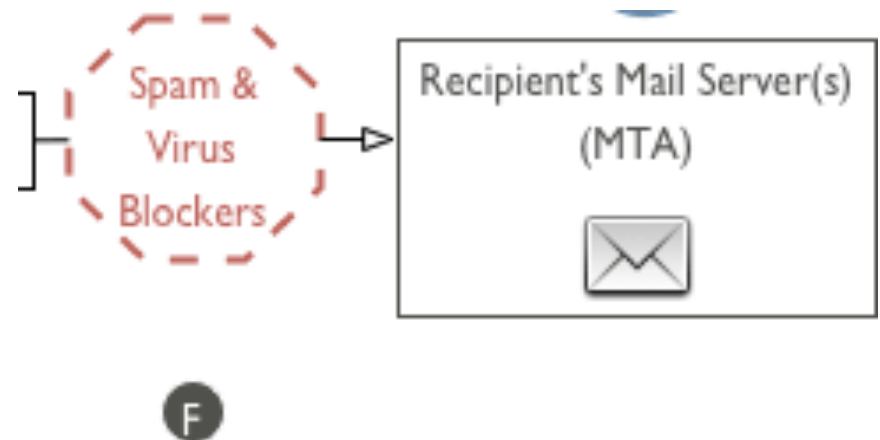
- Email **clears the queue**, enters the Internet network cloud, **where it is routed along a host-to-host** chain of servers
- The sending MTA **handles all aspects of mail delivery until the message has been either accepted or rejected** by the receiving MTA
- Each MTA needs to **"stop and ask directions"** from the **DNS in order to identify the next MTA in the delivery chain**
- Exact route **depends partly on server availability and mostly on which MTA can be found to accept email** for the domain specified in the address
- **ABUSE:** *Some spammers specify any part of the path, deliberately routing their message through a series of relay servers in an attempt to obscure the true origin of the message.*

# DNS resolution and transfer process

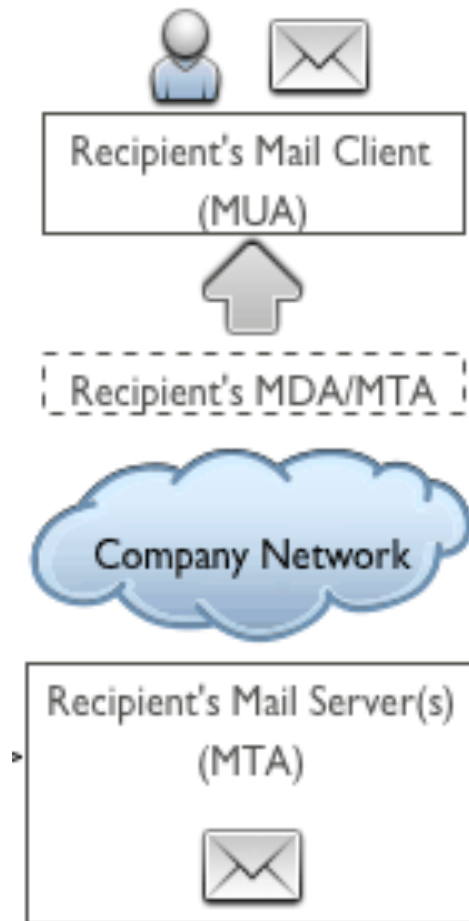
- **To find the recipient's IP address and mailbox**, the MTA must **drill down through the DNS system**, which consists of a set of servers distributed across the Internet beginning with the root nameservers
  - root servers refer requests for a given domain to the root nameservers that handle requests for that tld
    - *MTA can bypass this step because it has already knows which domain nameservers handle requests for these .tlds e.g. telecom.ma*
  - asks the appropriate DNS server which Mail Exchange (MX) servers have knowledge of the subdomain or local host in the email address
  - DNS server responds with an MX record: a prioritized list of MX servers for this domain
  - To the DNS server, the server that accepts messages is an MX server. When is transferring messages, it is called an MTA.
  - MTA contacts the MX servers on the MX record in order of priority until it finds the designated host for that address domain
  - **sending MTA asks if the host accepts messages for the recipient's username at that domain (i.e., username@domain.tld) and transfers the message**

# Firewalls, spam, and virus filters

- An email encountering a firewall may be **tested by spam and virus filters** before it is allowed to pass inside the firewall
- filters test to see **if the message qualifies as spam or malware**
- If the message contains **malware, the file is usually quarantined and the sender is notified**
- If the message is identified as spam, **it will probably be deleted without notifying the sender.**



# Delivery



- If the message makes it past the filters:
  - The MTA calls a local MDA to deliver the mail to the correct mailbox, where it will sit until it is retrieved by the recipient's MUA

# Bibliography: RFCs

- RFC 2821, 2822,
- RFC 1122, 1123: prerequisites for machines connected to the Internet
- RFC 1651: extensions to the SMTP protocol
- RFC 1653: SIZE extension
- RFC 1830: transporting large messages containing binaries
- MIME RFCs...



Postfix MTA

**Email**

# Short History

- Originally developed in the late 90s at IBM by Wietse Venema, author of security software (SATAN, TCPwrappers, ...), as "IBM Secure Mailer"
- Place under an Open Source license, and renamed "Postfix"
- Intended as a replacement for then insecure mail systems, such as Sendmail



# Design goals

- Safety
- Robustness
- Performance
- Modularity
- Compatibility

# Safety

- Postfix makes it very hard to lose mails – many checks to ensure that mail has been written to disk or delivered
- Back off mechanisms in case of repeated failure

# Security

- Collection of daemons working together
- Doesn't use environment for communication
- Very paranoid about input checking, all allocation is dynamic (avoiding buffer overflows)
- chroot support out of the box for almost all processes & daemons
- No data is *ever* exchanged directly between processes – all is done via IPC, and files on disk
- Conservative resource usage

# Performance

- Designed to be fast from the ground up
- Also behaves well with neighbors, doesn't flood them with mail, and instead uses a throughput adaptation
- Will not block delivery for a message if one recipient domain fails

# Modular

- One program, one function
- All programs controlled from "master.cf"
- Many small programs working together, with limited privileges
- Compatible with Sendmail's /etc/aliases and .forward conventions

# Features

- Virtual domains – domains and users are completely independent of system (UNIX) users
- Aliases – sendmail compatible
- Rewriting – senders, recipients, globally
- RBL support (Realtime Blackhole Lists) support
- Content filtering using pipes, SMTP or milter
- Support for arbitrary mail manipulation with policy services (custom programs talking to postfix)

# Features

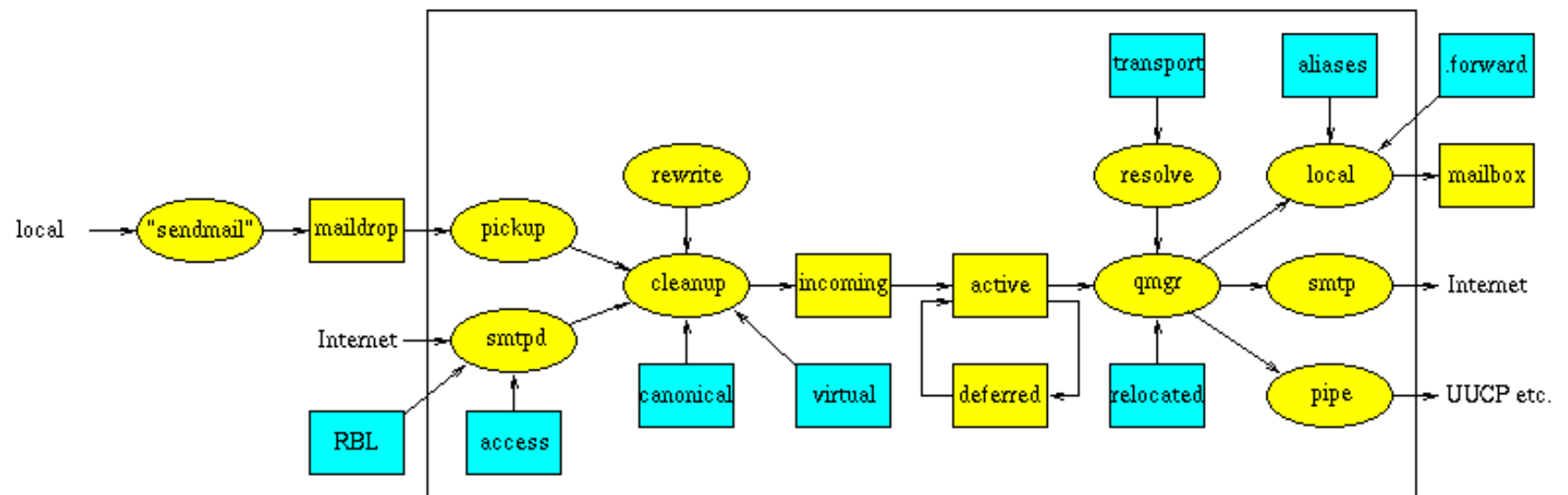
- Restriction classes
  - Conditional filtering
- Sender or recipient address verification (test email addresses before accepting mail from them)
- TLS support

# Core concept: maps

- In postfix, everything is looked up in a map (table)
- Maps can be in many formats or use many data sources:
  - hash/btree
  - regexp/PCRE
  - CIDR
  - NIS
  - LDAP, \*SQL (user defined queries)



# Architecture



# Basic Postfix configuration

## Two primary configuration files

- `main.cf`

- Main configuration file where all the subsystems are configured (smtp, smtpd, cleanup, routing, ...)

- `master.cf`

- File controlling how the "master" process of postfix will launch all the necessary postfix daemons to perform mail routing, on-demand

# Other configuration files

- Reside in "maps" mentioned earlier
- Tables containing values and conditions, referred to from main.cf, controlling all aspects such as:
  - Virtual and local domains
  - Routing rules
  - Access control
  - Rewriting
  - ...

# Configuration: postconf command

- postconf – used to view and edit configuration parameters
  - For changing the configuration, it is usually done vi editing "main.cf" directly

# Some basic main.cf

```
# what domains do I accept mail for (user@...)
mydestination = $myhostname, localhost, \
    hervey.ws.nsrc.org

# who do I send mail as ?
myorigin = $mydomain

# what clients do I consider local (and trust them)
mynetworks = 127.0.0.0/8 192.168.1.0/24

# Send all outgoing mail to this server
relayhost = mail.example.com

# Aliases
alias_maps = hash:/etc/aliases
```

## Some basic main.cf (cont'd)

- **in the file `/etc/aliases`:**

**root:** sysadm

phil: regnauld@nsrc.org

# Virtual domains

- Allows having multiple mail domains on one machine
- They can be *completely* different than your own hostname/domainname
- Example:
  - **in main.cf:**  
`virtual_maps = $cf/virtual-domains`
  - **in virtual-domains file:**

<code>superdomain.com</code>	<code>VIRTUAL</code>
<code>phil@superdomain.com</code>	<code>phil@nsrc.org, pr@eu.org</code>
<code>@superdmain.com</code>	<code>sysadm@localhost</code>

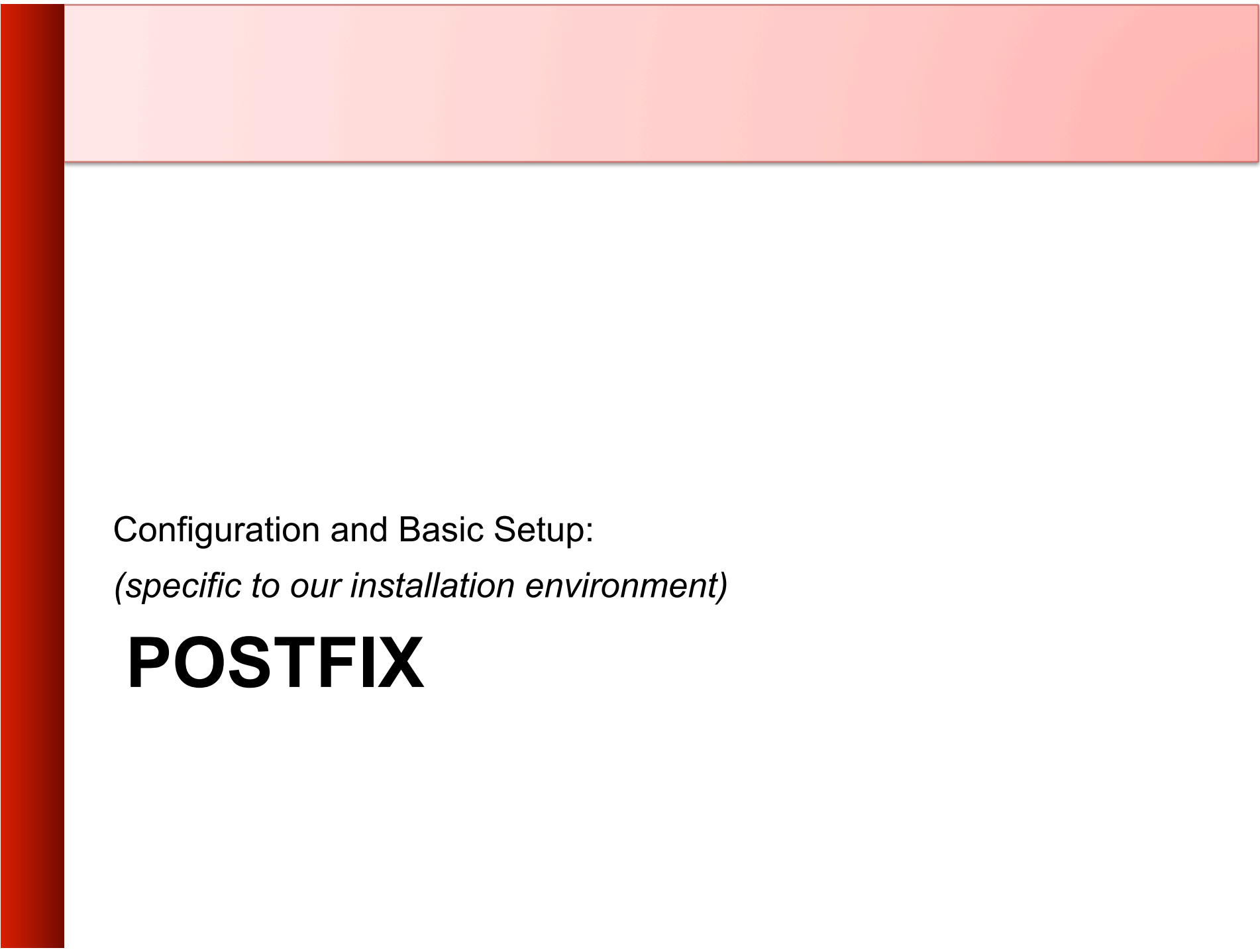
# Controlling postfix

- postfix start – start the postfix system
- postfix stop – stop the postfix system
- postfix check – verify the configuration
- newaliases – rebuild the local aliases
- mailq – show the mails in the  
queue currently being processed



# Bibliography: References

- Links
  - <http://www.postfix.org/>
  - <http://www.ijs.si/software/amavisd/>
- Books:
  - "Postfix", Richard Blum, ed. Sams (1st ed. May 15, 2001), 624 p., ISBN: 0672321149
  - "The book of Postfix", Ralf Hildebrandt, Patrick Koetter ed. No Starch Press (October 2003), 328 p., ISBN: 1593270011



Configuration and Basic Setup:  
*(specific to our installation environment)*

# POSTFIX

# installing

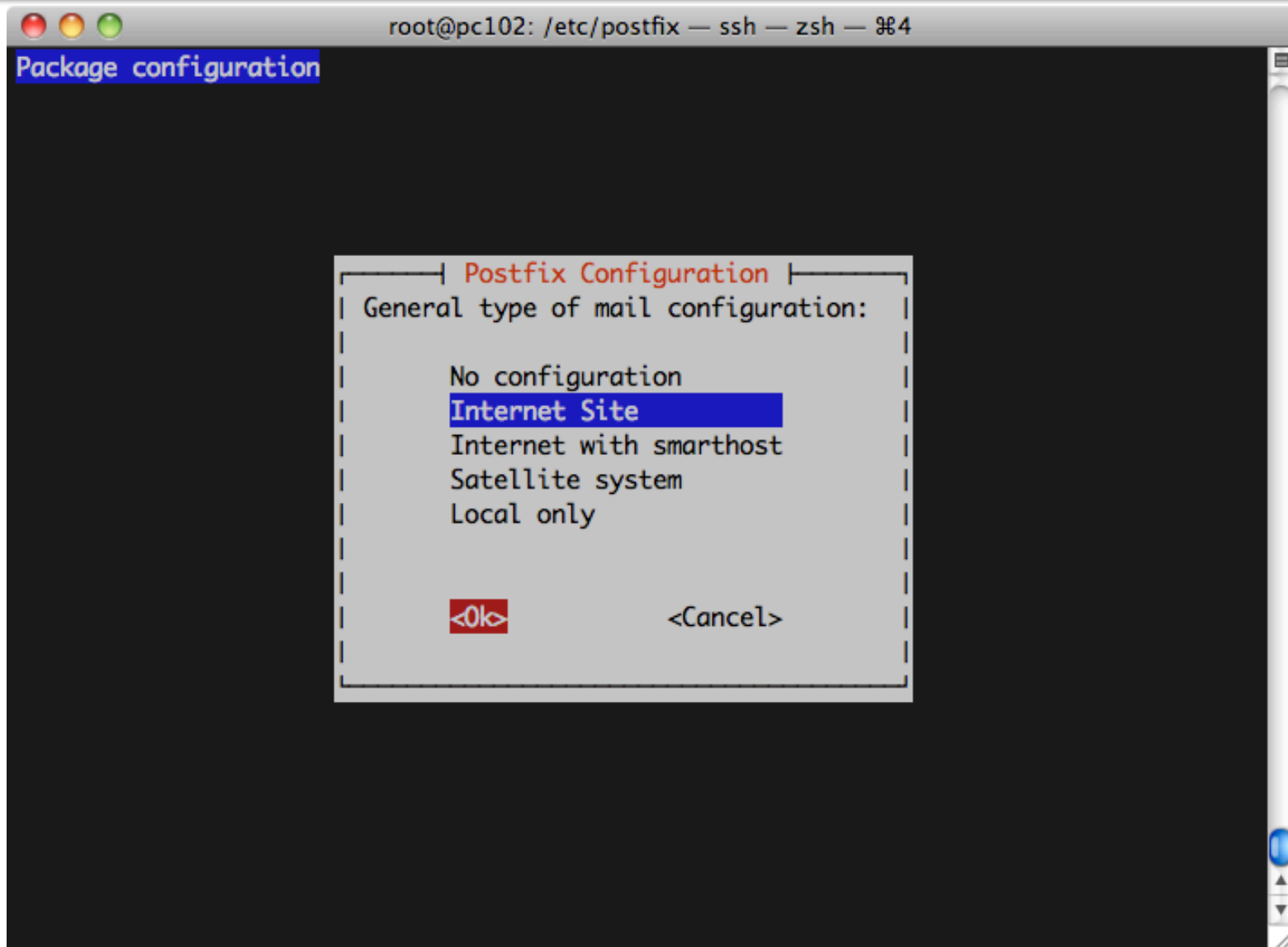
- It is already installed, otherwise you'd have to do:

```
# apt-get install postfix
```

- In our case we need to do:

```
# dpkg-reconfigure -phigh postfix
```

# Type of Mail configuration



# FQDN

## Postfix Configuration

The "mail name" is the domain name used to "qualify" `_ALL_` mail addresses without a domain name. This includes mail to and from `<root>`: please do not make your machine send out mail from `root@example.org` unless `root@example.org` has told you to.

This name will also be used by other programs. It should be the single, fully qualified domain name (FQDN).

Thus, if a mail address on the local host is `foo@example.org`, the correct value for this option would be `example.org`.

System mail name:

`pc102.ws.nsrc.org`

`<Ok>`

`<Cancel>`

# myhostname

- our virtual hosting solution will create a hostname of gold.ws.nsrc.org. we need to change this as well as set the 'destinations' this host accepts mail for.
- Edit /etc/postfix/main.cf and set:

```
myhostname = pc31.ws.nsrc.org  
mydestination = localhost, $myhostname \  
                hervey.ws.nsrc.org
```

- Restart postfix:  
**# service postfix restart**

# Install a basic MUA

```
# apt-get install alpine
```

```
$ alpine
```

(command keys in alpine are case insensitive). Press 'e' to exit the counter then follow the menu 'i' to go to the inbox, etc.

Try sending an email to [sysadm@pcX.ws.nsrc.org](mailto:sysadm@pcX.ws.nsrc.org) as well as any other domain that should be working in class.

Since we did not delegate from nsrc.org mail to yahoo.com may bounce but should work to a gmail.com address. Returning mail won't find its way here.



Extras

email



# Bits and pieces we can't cover

- Adding SSL to SMTP as well as SMTP AUTH
- POP3
- IMAP
- Webmail
- SSL to POP3 and IMAP
- Configuration of other MUAs