

p≡p-mail-tool

Using p≡p for email with ordinary Mail User Agents

A status report with some personal opinions

Luca Saiu

positron@pep-project.org

p≡p Foundation

2023-06-26

Updated version, last changed on 2023-10-19.

*These slides are **not** being kept up to date with software changes: as of 2023-10 p≡p-mail-tool already has **many new features not mentioned here**.*



My purpose

p≡p-mail-tool is a new software I wrote to make p≡p-for-email support **more Mail User Agents** with **little effort**.

Many hackers are particular in their preferences for mail software: by supporting software popular among technical people we can get mindshare.

This new software is also useful for us to develop and debug the p≡p Engine.



My way of using mail

I am old

- email is my favourite mean of communication (personal messages other than IM)
- asynchronous, allows for long, carefully-written messages

I care about free software

I follow many GNU projects (and others)

I am a sentimental person

I keep every message since 2005

(I even keep spam)

I have a lot of messages, multiple mailboxes, complex classification

rules, filters, etc. I will not abandon email for instant messaging



My way of using mail

I am old

- email is my favourite mean of communication (or second favourite after Usenet)
- asynchronous, allows for long, carefully-written messages

I care about free software

I follow many GNU projects (and others)

I am a sentimental person

I keep every message since 2005

I keep every message since 2005

I even keep spam

I use a lot of messages, multiple mailboxes, complex classification

I use a lot of messages, multiple mailboxes, complex classification



My way of using mail

I am old

- email is my favourite mean of communication (or second favourite after Usenet)
- asynchronous, allows for long, carefully-written messages

I care about free software

I follow many GNU projects (and others)

I am a sentimental person

I keep every message since 2005

I keep every message since 2005

I keep every message since 2005

I keep every message since 2005

I keep every message since 2005

I keep every message since 2005



My way of using mail

I am old

- email is my favourite mean of communication (or second favourite after Usenet)
- asynchronous, allows for long, carefully-written messages

I care about free software

- I follow many GNU project mailing lists (and others)

I am a sentimental person

I keep every message since 2005

(even those spam)

I have a lot of mailboxes and lots of mailboxes, complex classification

I have a lot of mailboxes and lots of mailboxes, complex classification

I have a lot of mailboxes and lots of mailboxes, complex classification



My way of using mail

I am old

- email is my favourite mean of communication (or second favourite after Usenet)
- asynchronous, allows for long, carefully-written messages

I care about free software

- I follow many GNU project mailing lists (and others)

I am a sentimental person

I keep everything since 2005

I keep everything since 2005

I keep everything since 2005

I keep everything since 2005

I keep everything since 2005

I keep everything since 2005



My way of using mail

I am old

- email is my favourite mean of communication (or second favourite after Usenet)
- asynchronous, allows for long, carefully-written messages

I care about free software

- I follow many GNU project mailing lists (and others)

I am a sentimental person

- I keep every message since 2005
 - ...because in 2005 I had a hard disk crash
 - lost my mail archive between 1997 and 2005!
- (I even keep spam)
- millions of messages; multiple mailboxes, complex classification

Mail is important: I will not tolerate an unusable or inconvenient setup.



My way of using mail

I am old

- email is my favourite mean of communication (or second favourite after Usenet)
- asynchronous, allows for long, carefully-written messages

I care about free software

- I follow many GNU project mailing lists (and others)

I am a sentimental person

- I keep every message since 2005
 - ...because in 2005 I had a hard disk crash
 - lost my mail archive between 1997 and 2005!
- (I even keep spam)
- millions of messages; multiple mailboxes, complex classification

Mail is important: I will not tolerate an unusable or inconvenient setup.



My way of using mail

I am old

- email is my favourite mean of communication (or second favourite after Usenet)
- asynchronous, allows for long, carefully-written messages

I care about free software

- I follow many GNU project mailing lists (and others)

I am a sentimental person

- I keep every message since 2005
 - ...because in 2005 I had a hard disk crash
 - lost my mail archive between 1997 and 2005!
- (I even keep spam)
- millions of messages; multiple mailboxes, complex classification

Mail is important: I will not tolerate an unusable or inconvenient setup.



My way of using mail

I am old

- email is my favourite mean of communication (or second favourite after Usenet)
- asynchronous, allows for long, carefully-written messages

I care about free software

- I follow many GNU project mailing lists (and others)

I am a sentimental person

- I keep every message since 2005
 - ... because in 2005 I had a hard disk crash
 - lost my mail archive between 1997 and 2005!
- (I even keep spam)
- millions of messages; multiple mailboxes, complex classification

Mail is important: I will not tolerate an unusable or inconvenient setup.



My way of using mail

I am old

- email is my favourite mean of communication (or second favourite after Usenet)
- asynchronous, allows for long, carefully-written messages

I care about free software

- I follow many GNU project mailing lists (and others)

I am a sentimental person

- I keep every message since 2005
 - ... because in 2005 I had a hard disk crash
 - lost my mail archive between 1997 and 2005!
- (I even keep spam)
- millions of messages; multiple mailboxes, complex classification

Mail is important: I will not tolerate an unusable or inconvenient setup.



Am I an unusual case?

I use the **command line**.

Linguistic interfaces are superior to graphical interfaces for human interaction

From <https://www.gnu.org/philosophy/why-free-software.html>: "The GNU project promotes the notion of language as an interface - a medium of the abstract."

See <https://www.gnu.org/philosophy/why-free-software.html> for more on this.

See <https://www.gnu.org/philosophy/why-free-software.html> for more on this.

Even more than the Unix command line I use the **GNU Emacs** <https://gnu.org/s/emacs> text editor (Emacs is a work of genius)

- essentially a **Lisp environment** [https://en.wikipedia.org/wiki/Lisp_\(programming_language\)](https://en.wikipedia.org/wiki/Lisp_(programming_language)), written in C (the best work of genius)
- all the editor functions written in Lisp
- scriptable, user-extensible
 - entire applications written on top of Emacs!

I use Emacs to write this page, using <https://www.gnu.org/philosophy/why-free-software.html>, running on top of Linux, as my main Linux desktop.



Am I an unusual case?

I use the **command line**.

Linguistic interfaces are superior to graphical interfaces for human interaction

[I can defend this statement. See my GHM 2022 presentation "In defence of language as an interface – a statement of the obvious":

- Slides: <https://ageinghacker.net/talks/language-slides--saiu--ghm-2022--2022-10-01.pdf>
- Video recording: <https://audio-video.gnu.org/video/ghm2022/2022-10--language--saiu--ghm.webm>

Even more than the Unix command line I use the **GNU Emacs** <https://gnu.org/s/emacs> text editor (Emacs is a work of genius)

- essentially a **Lisp environment** [https://en.wikipedia.org/wiki/Lisp_\(programming_language\)](https://en.wikipedia.org/wiki/Lisp_(programming_language)), written in C (Emacs is a work of genius)
- all the editor functions written in Lisp
- scriptable, user-extensible
 - entire applications written on top of Emacs!

(Emacs is also the best Emacs configuration ever, <https://ageinghacker.net/emacs/>, running on top of Emacs, as my Emacs user report)



Am I an unusual case?

I use the **command line**.

Linguistic interfaces are superior to graphical interfaces for human interaction

[I can defend this statement. See my GHM 2022 presentation "In defence of language as an interface – a statement of the obvious":

- Slides: <https://ageinghacker.net/talks/language-slides--saiu--ghm-2022--2022-10-01.pdf>
- Video recording: <https://audio-video.gnu.org/video/ghm2022/2022-10--language--saiu--ghm.webm>

Even more than the Unix command line I use the **GNU Emacs** <https://gnu.org/s/emacs> text editor

- essentially a **Lisp environment** [https://en.wikipedia.org/wiki/Lisp_\(programming_language\)](https://en.wikipedia.org/wiki/Lisp_(programming_language)), written in C
- all the editor functions written in Lisp
- scriptable, user-extensible
 - entire applications written on top of Emacs!

https://www.gnu.org/software/emacs/manual/html_node/emacs/emacs.html



Am I an unusual case?

I use the **command line**.

Linguistic interfaces are superior to graphical interfaces for human interaction

[I can defend this statement. See my GHM 2022 presentation "In defence of language as an interface – a statement of the obvious":

- Slides: <https://ageinghacker.net/talks/language-slides--saiu--ghm-2022--2022-10-01.pdf>
- Video recording: <https://audio-video.gnu.org/video/ghm2022/2022-10--language--saiu--ghm.webm>

Even more than the Unix command line I use the **GNU Emacs** <https://gnu.org/s/emacs> text editor (Emacs is a work of beauty)

- essentially a **Lisp environment** [https://en.wikipedia.org/wiki/Lisp_\(programming_language\)](https://en.wikipedia.org/wiki/Lisp_(programming_language)), written in C
- all the editor functions written in Lisp
- scriptable, user-extensible
 - entire applications written on top of Emacs!

<https://ageinghacker.net/2022/09/20/emacs-is-a-work-of-beauty/>



Am I an unusual case?

I use the **command line**.

Linguistic interfaces are superior to graphical interfaces for human interaction

[I can defend this statement. See my GHM 2022 presentation “In defence of language as an interface – a statement of the obvious”:

- Slides: <https://ageinhacker.net/talks/language-slides--saiu--ghm-2022--2022-10-01.pdf>
- Video recording: <https://audio-video.gnu.org/video/ghm2022/2022-10--language--saiu--ghm.webm>

Even more than the Unix command line I use the **GNU Emacs** <https://gnu.org/s/emacs> text editor (Emacs is a work of beauty)

- essentially a **Lisp environment** [https://en.wikipedia.org/wiki/Lisp_\(programming_language\)](https://en.wikipedia.org/wiki/Lisp_(programming_language)), written in C (Lisp is a work of genius)
- all the editor functions written in Lisp
- scriptable, user-extensible
 - entire applications written on top of Emacs!
 - I use **Gnus** (by Lars Magne Ingebrigsten; <https://gnus.org>), running on top of Emacs, as my Mail User Agent



Am I an unusual case?

I use the **command line**.

Linguistic interfaces are superior to graphical interfaces for human interaction

[I can defend this statement. See my GHM 2022 presentation "In defence of language as an interface – a statement of the obvious":

- Slides: <https://ageinhacker.net/talks/language-slides--saiu--ghm-2022--2022-10-01.pdf>
- Video recording: <https://audio-video.gnu.org/video/ghm2022/2022-10--language--saiu--ghm.webm>

Even more than the Unix command line I use the **GNU Emacs** <https://gnu.org/s/emacs> text editor (Emacs is a work of beauty)

- essentially a **Lisp environment** [https://en.wikipedia.org/wiki/Lisp_\(programming_language\)](https://en.wikipedia.org/wiki/Lisp_(programming_language)), written in C (Lisp is a work of genius)
- all the editor functions written in Lisp
- scriptable, user-extensible
 - entire applications written on top of Emacs!
 - I use **Gnus** (by Lars Magne Iñebrigsten; <https://gnus.org>), running on top of Emacs, as my Mail User Agent



Am I an unusual case?

I use the **command line**.

Linguistic interfaces are superior to graphical interfaces for human interaction

[I can defend this statement. See my GHM 2022 presentation “In defence of language as an interface – a statement of the obvious”:

- Slides: <https://ageinhacker.net/talks/language-slides--saiu--ghm-2022--2022-10-01.pdf>
- Video recording: <https://audio-video.gnu.org/video/ghm2022/2022-10--language--saiu--ghm.webm>

Even more than the Unix command line I use the **GNU Emacs** <https://gnu.org/s/emacs> text editor (Emacs is a work of beauty)

- essentially a **Lisp environment** [https://en.wikipedia.org/wiki/Lisp_\(programming_language\)](https://en.wikipedia.org/wiki/Lisp_(programming_language)), written in C (Lisp is a work of genius)
- all the editor functions written in Lisp
- scriptable, user-extensible
 - entire applications written on top of Emacs!
 - I use **Gnus** (by Lars Magne Iñebrigsten; <https://gnus.org>), running on top of Emacs, as my Mail User Agent



Am I an unusual case?

I use the **command line**.

Linguistic interfaces are superior to graphical interfaces for human interaction

[I can defend this statement. See my GHM 2022 presentation “In defence of language as an interface – a statement of the obvious”:

- Slides: <https://ageinhacker.net/talks/language-slides--saiu--ghm-2022--2022-10-01.pdf>
- Video recording: <https://audio-video.gnu.org/video/ghm2022/2022-10--language--saiu--ghm.webm>

Even more than the Unix command line I use the **GNU Emacs** <https://gnu.org/s/emacs> text editor (Emacs is a work of beauty)

- essentially a **Lisp environment** [https://en.wikipedia.org/wiki/Lisp_\(programming_language\)](https://en.wikipedia.org/wiki/Lisp_(programming_language)), written in C (Lisp is a work of **genius**)
- all the editor functions written in Lisp
- scriptable, user-extensible
 - entire applications written on top of Emacs!
 - I use **Gnus** (by Lars Magne Iñebrigsten; <https://gnus.org>), running on top of Emacs, as my Mail User Agent



Am I an unusual case?

[Gnus: quick demo]

[For myself: pmt-demo-0. Do not use nnimap+pEp:tests-incoming]



Am I an unusual case?

Am I an unusual case?

- No, just a hacker [in the GNU sense];
- other hackers have different but equally strong preferences. Hackers want expressivity and control.
- (maybe, thinking again, I am unusual: I also have to test the p≡p Engine)



Am I an unusual case?

Am I an unusual case?

- **No**, just a **hacker** [in the GNU sense];
- other hackers have different but equally strong preferences. Hackers want expressivity and control.
- (maybe, thinking again, I am unusual: I also have to test the p≡p Engine)



Am I an unusual case?

Am I an unusual case?

- **No**, just **a hacker** [in the GNU sense];
- other hackers have different but equally **strong preferences**. Hackers want expressivity and control.
- (maybe, thinking again, I am unusual: I also have to **test the p≡p Engine**)



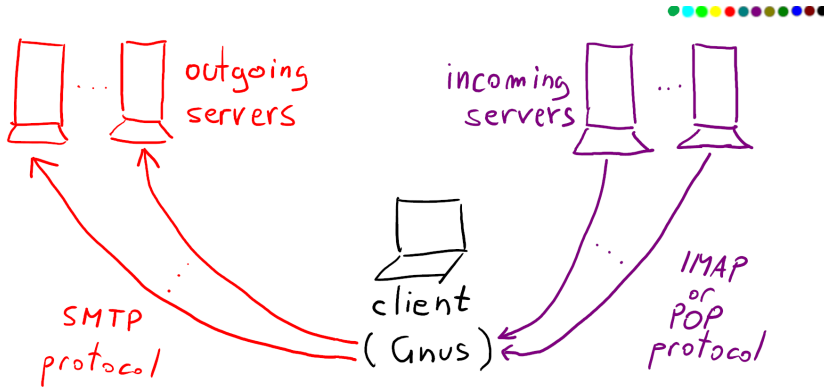
Am I an unusual case?

Am I an unusual case?

- **No**, just **a hacker** [in the GNU sense];
- other hackers have different but equally **strong preferences**. Hackers want expressivity and control.
- (maybe, thinking again, I am unusual: I also have to **test the p≡p Engine**)



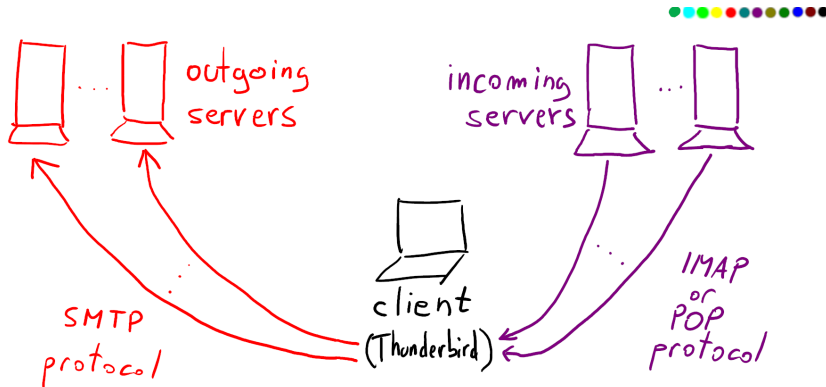
Email: servers and clients (remote case only)



The **outgoing servers** and **incoming servers** may run on the same server machine but speak different protocols. The client is a user application: Thunderbird, Gnus...



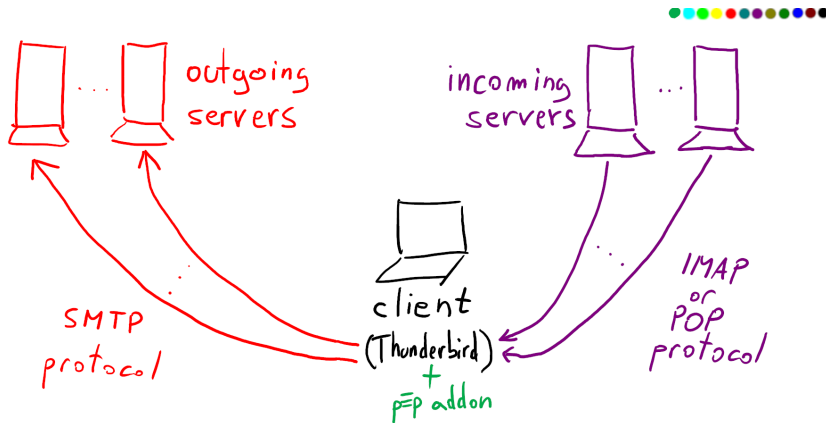
Email: servers and clients (remote case only) with a p≡p plugin (not p≡p-mail-tool)



This is how "traditional" p≡p plugins, **not** p≡p-mail-tool, work. The p≡p plugin (for Thunderbird or anything else) is complex and only works for one client program.



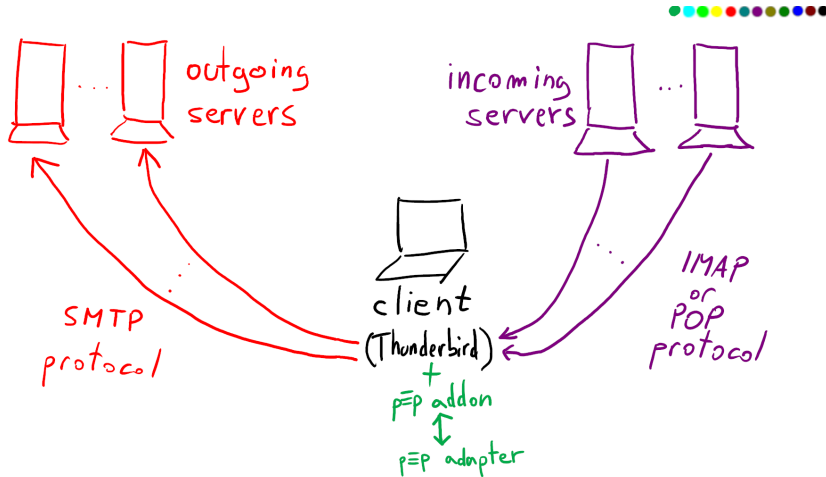
Email: servers and clients (remote case only) with a p≡p plugin (not p≡p-mail-tool)



This is how "traditional" p≡p plugins, **not** p≡p-mail-tool, work. The p≡p plugin (for Thunderbird or anything else) is complex and only works for one client program.



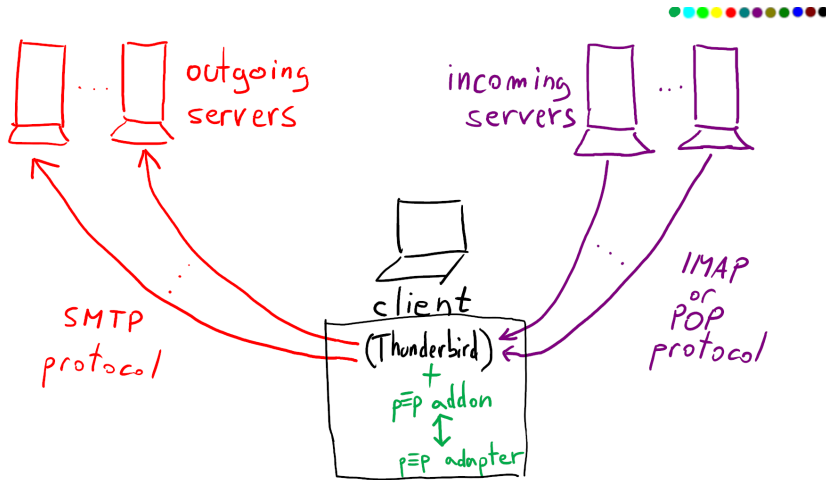
Email: servers and clients (remote case only) with a p≡p plugin (not p≡p-mail-tool)



This is how "traditional" p≡p plugins, **not** p≡p-mail-tool, work. The p≡p plugin (for Thunderbird or anything else) is complex and only works for one client program.



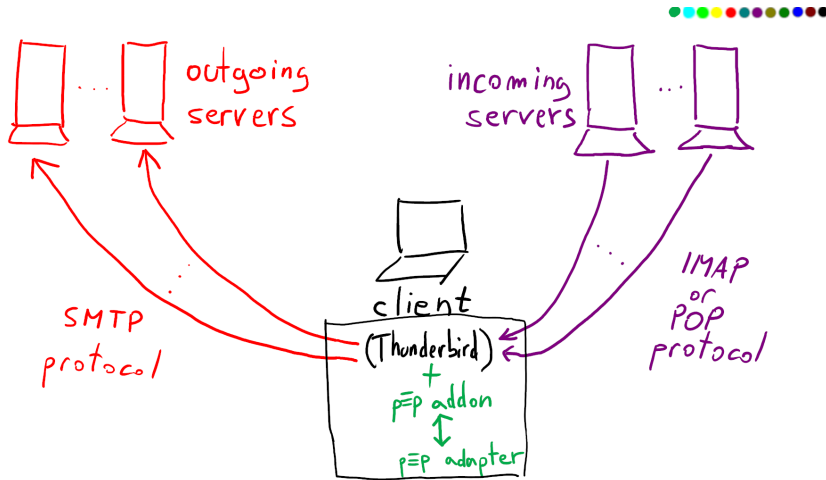
Email: servers and clients (remote case only) with a p≡p plugin (not p≡p-mail-tool)



This is how "traditional" p≡p plugins, **not** p≡p-mail-tool, work. The p≡p plugin (for Thunderbird or anything else) is **complex** and only works for **one client program**.



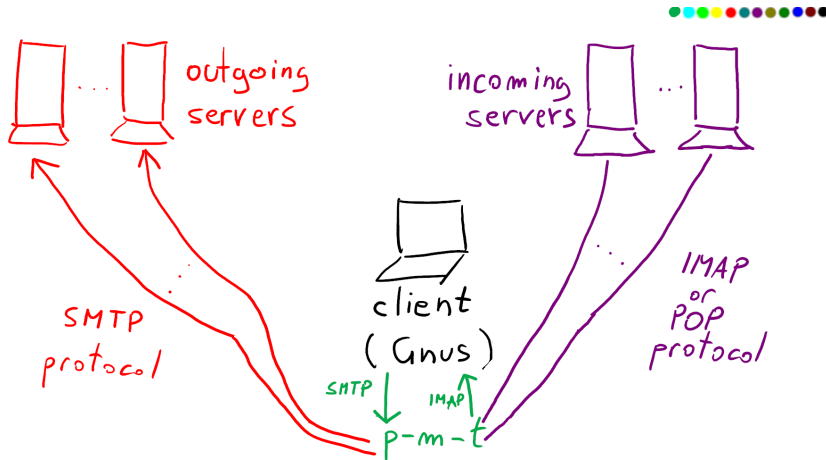
Email: servers and clients (remote case only) with a p≡p plugin (not p≡p-mail-tool)



This is how "traditional" p≡p plugins, **not** p≡p-mail-tool, work. The p≡p plugin (for Thunderbird or anything else) is **complex** and only works for **one client program**.



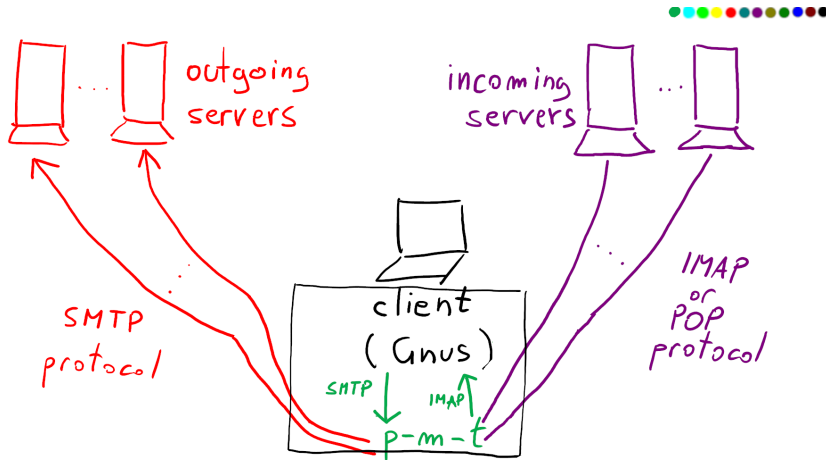
A simpler and more modular solution: p≡p-mail-tool



p≡p-mail-tool locally serves unencrypted messages over IMAP and SMTP. Easier!
(This diagram is simplified: what I show here as "p-m-t" is actually more complex...)



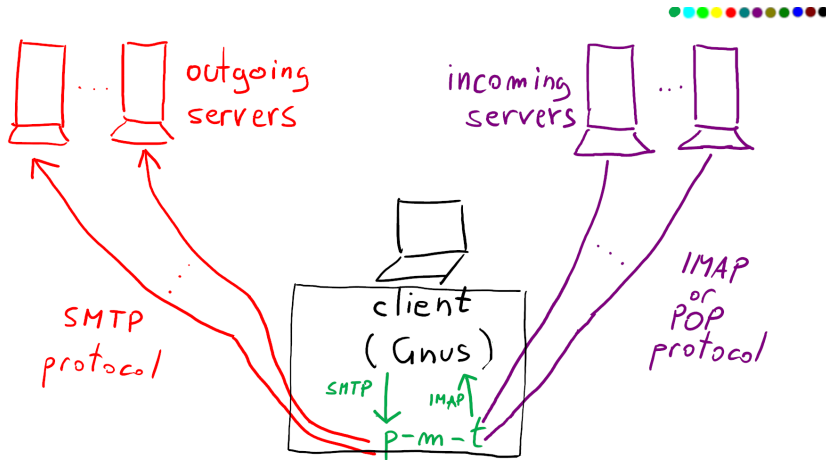
A simpler and more modular solution: p≡p-mail-tool



p≡p-mail-tool locally serves unencrypted messages over IMAP and SMTP. Easier!
(This diagram is simplified: what I show here as "p-m-t" is actually more complex...)



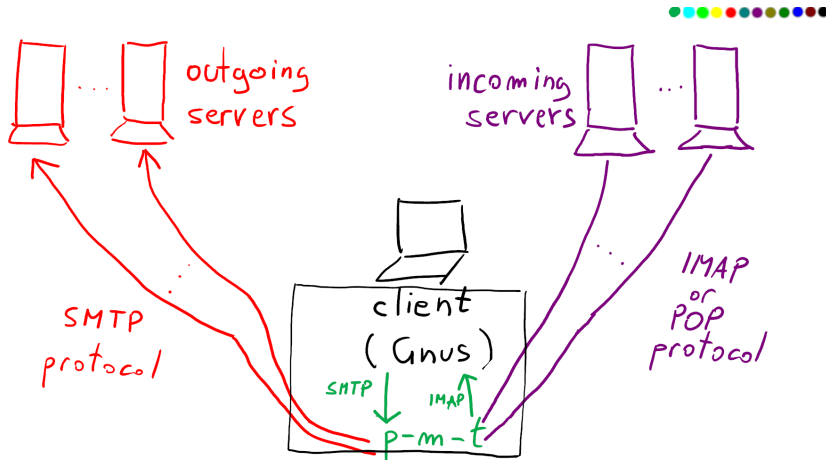
A simpler and more modular solution: p≡p-mail-tool



p≡p-mail-tool locally serves unencrypted messages over IMAP and SMTP. **Easier!**

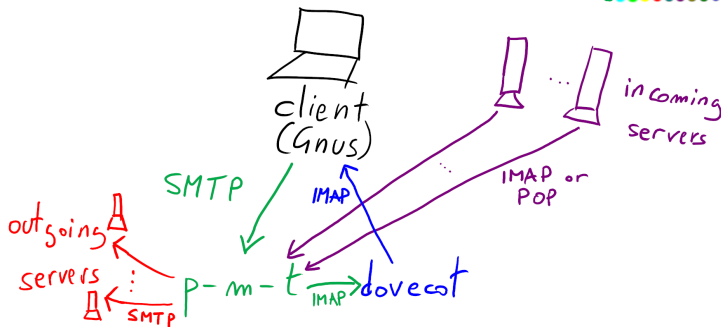
(This diagram is simplified: what I show here as "p-m-t" is actually more complex...)

A simpler and more modular solution: p≡p-mail-tool



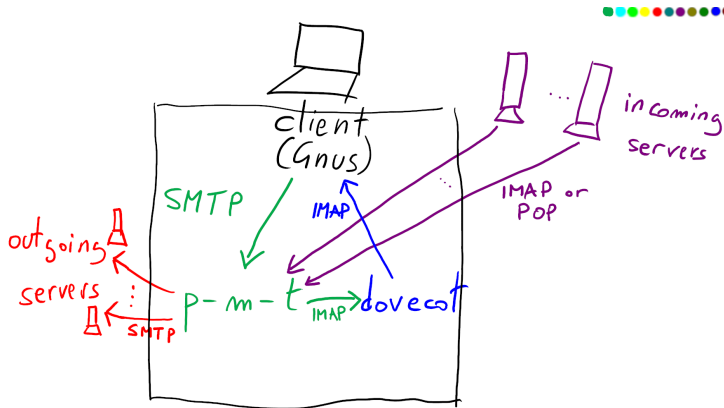
p≡p-mail-tool locally serves unencrypted messages over IMAP and SMTP. **Easier!**
(This diagram is simplified: what I show here as "p-m-t" is actually more complex. . .)





I can include a Python SMTP server (`aiosmtpd`) in `p≡p-mail-tool`. The IMAP side is much more critical and difficult to rewrite, so I am reusing `Dovecot` <https://dovecot.org>, a good IMAP server.

The local mail server must be fast and reliable, and speak IMAP



I can include a Python SMTP server (`aiosmtpd`) in `p≡p-mail-tool`. The IMAP side is much more critical and difficult to rewrite, so I am reusing `Dovecot` <https://dovecot.org>, a good IMAP server.

[2023-07 update: now Dovecot is **automatically set up, started, stopped** by `p≡p-mail-tool`: it is still used as shown in this figure, but the user does not see any of the complexity. **No need for the user to write a Dovecot configuration file!**]



Gnus and p≡p-mail-tool

[Demo: Gnus and p≡p-mail-tool]

[For myself: pmt-demo-0 again with nnimap+pEp:tests-incoming]



The client experience

Are we missing something important when using the Mail User Agent?

- Yes: when composing a message we need to see its **outgoing colour**.

→ A crude solution: using p≡p-mail-tool

→ A better solution: modifying p≡p-mail-tool



The client experience

Are we missing something important when using the Mail User Agent?

- **Yes:** when composing a message we need to see its **outgoing colour**.
 - A crude solution: using p≡p-mail-tool from the command line.
[Demo: p≡p-mail-tool from the command line]
 - A better solution: invoking p≡p-mail-tool from the Mail User Agent, as a subprocess.
[Demo: Gnus with p≡p-mail-tool]



The client experience

Are we missing something important when using the Mail User Agent?

- **Yes:** when composing a message we need to see its **outgoing colour**.
 - A crude solution: using p≡p-mail-tool **from the command line**.

```
[Demo: pEp-mail-tool --outgoing-rating-stdin]
```

- A better solution: invoking p≡p-mail-tool from the Mail User Agent, as a subprocess.

```
[Demo: Gnus with ...]
```



The client experience

Are we missing something important when using the Mail User Agent?

- **Yes:** when composing a message we need to see its **outgoing colour**.
 - A crude solution: using p≡p-mail-tool **from the command line**.

[Demo: pEp-mail-tool --outgoing-rating-stdin]

- A better solution: invoking p≡p-mail-tool from the Mail User Agent, as a subprocess.

[Demo: Gnus with ...]



The client experience

Are we missing something important when using the Mail User Agent?

- **Yes:** when composing a message we need to see its **outgoing colour**.
 - A crude solution: using p≡p-mail-tool **from the command line**.
- A better solution: invoking p≡p-mail-tool **from the Mail User Agent, as a subprocess**.

[Demo: pEp-mail-tool --outgoing-rating-stdin]

[Demo: Gnus with p≡p-mail-tool minor mode]

[For myself: get-demo-1, then quickly get-demo-2 and get-demo-3]



The client experience

Are we missing something important when using the Mail User Agent?

- **Yes:** when composing a message we need to see its **outgoing colour**.
 - A crude solution: using p≡p-mail-tool **from the command line**.
[Demo: pEp-mail-tool --outgoing-rating-stdin]
 - A better solution: invoking p≡p-mail-tool **from the Mail User Agent, as a subprocess**.
[Demo: Gnus with **p≡p-mail-tool minor mode**]

[For myself: pmt-demo-1, then quickly pmt-demo-2 and pmt-demo-3.]



Status

<https://codeberg.org/pEp-foundation/pEp-mail-tool>

p≡p-mail-tool seems very solid and is ready to be played with. The code is simple enough Python (~ 1300 LoC [2023-10 update: ~ 3000 LoC]), building on the p≡p Python adapter. The minor mode for Gnus is clean Emacs Lisp.

I welcome:

- bug reports
- code contributions (support for other Mail User Agents)
- help from some Python expert with aiosmtpd authentication [2023-09 update: no longer needed, *trust setting* now implemented]

Limitations:

- no *trust setting* [2023-07 update: no longer true, *trust setting* is implemented now]
- no *keyreset* [2023-07: no longer true, *key reset* is also implemented now]
- no *sync* (not so important, because...)
 - ... **trusted (local) server only** (a significant limitation for some)

Please try it. [2023-10: several important new features already added!]



Status

<https://codeberg.org/pEp-foundation/pEp-mail-tool>

p≡p-mail-tool seems very solid and is ready to be played with. The code is simple enough Python (~ 1300 LoC [2023-10 update: ~ 3000 LoC]), building on the p≡p Python adapter. The minor mode for Gnus is clean Emacs Lisp.

I welcome:

- bug reports
- code contributions (support for other Mail User Agents)
- help from some Python expert with aiosmtpd authentication [2023-09 update: no longer needed, *trust setting* now implemented]

Limitations:

- no *trust setting* [2023-07 update: no longer true, *trust setting* is implemented now]
- no *keyreset* [2023-07: no longer true, *key reset* is also implemented now]
- no *sync* (not so important, because...)
 - ... *trusted (local) server only* (a significant limitation for some)

Please try it. [2023-10: several important new features already added!]



Status

<https://codeberg.org/pEp-foundation/pEp-mail-tool>

p≡p-mail-tool seems very solid and is ready to be played with. The code is simple enough Python (~ 1300 LoC [2023-10 update: ~ 3000 LoC]), building on the p≡p Python adapter. The minor mode for Gnus is clean Emacs Lisp.

Limitations:

I welcome:

- bug reports
- code contributions (support for other Mail User Agents)
- help from some Python expert with aiosmtpd authentication [2023-09 update: no longer needed, *trust setting* now implemented]

- no *trust setting* [2023-07 update: no longer true, *trust setting* is implemented now]
- no *keyreset* [2023-07: no longer true, *key reset* is also implemented now]
- no *sync* (not so important, because...)
 - ... *trusted (local) server only* (a significant limitation for some)

Please try it. [2023-10: several important new features already added!]



Status

<https://codeberg.org/pEp-foundation/pEp-mail-tool>

p≡p-mail-tool seems very solid and is ready to be played with. The code is simple enough Python (~ 1300 LoC [2023-10 update: ~ 3000 LoC]), building on the p≡p Python adapter. The minor mode for Gnus is clean Emacs Lisp.

Limitations:

I welcome:

- bug reports
- code contributions (support for other Mail User Agents)
- help from some Python expert with aiosmtpd authentication [2023-09 update: no longer needed, *trust setting* now implemented]

- no trust setting [2023-07 update: no longer true, *trust setting* is implemented now]
- no *keyreset* [2023-07: no longer true, *key reset* is also implemented now]
- no *sync* (not so important, because...)
 - ... *trusted (local) server only* (a significant limitation for some)

Please try it. [2023-10: several important new features already added!]



Status

<https://codeberg.org/pEp-foundation/pEp-mail-tool>

p≡p-mail-tool seems very solid and is ready to be played with. The code is simple enough Python (~ 1300 LoC [2023-10 update: ~ 3000 LoC]), building on the p≡p Python adapter. The minor mode for Gnus is clean Emacs Lisp.

Limitations:

I welcome:

- bug reports
- code contributions (support for other Mail User Agents)
- help from some Python expert with aiosmtpd authentication [2023-09 update: no longer needed, *trust setting* now implemented]

- no trust setting [2023-07 update: no longer true, *trust setting* is implemented now]
- no *keyreset* [2023-07: no longer true, *key reset* is also implemented now]
- no *sync* (not so important, because...)
 - ... *trusted (local) server only* (a significant limitation for some)

Please try it. [2023-10: several important new features already added!]



Status

<https://codeberg.org/pEp-foundation/pEp-mail-tool>

p≡p-mail-tool seems very solid and is ready to be played with. The code is simple enough Python (~ 1300 LoC [2023-10 update: ~ 3000 LoC]), building on the p≡p Python adapter. The minor mode for Gnus is clean Emacs Lisp.

Limitations:

I welcome:

- bug reports
- code contributions (support for other Mail User Agents)
- help from some Python expert with aiosmtpd authentication [2023-09 update: no longer needed, *trust setting* now implemented]

- no trust setting [2023-07 update: no longer true, *trust setting* is implemented now]
- no *keyreset* [2023-07: no longer true, *key reset* is also implemented now]
- no *sync* (not so important, because...)
 - ... **trusted (local) server only** (a **significant limitation** for some)

Please try it. [2023-10: several important new features already added!]



Status

<https://codeberg.org/pEp-foundation/pEp-mail-tool>

p≡p-mail-tool seems very solid and is ready to be played with. The code is simple enough Python (~ 1300 LoC [2023-10 update: ~ 3000 LoC]), building on the p≡p Python adapter. The minor mode for Gnus is clean Emacs Lisp.

Limitations:

I welcome:

- bug reports
- code contributions (support for other Mail User Agents)
- help from some Python expert with aiosmtpd authentication [2023-09 update: no longer needed, *trust setting* now implemented]

- no trust setting [2023-07 update: no longer true, *trust setting* is implemented now]
- no *keyreset* [2023-07: no longer true, *key reset* is also implemented now]
- no *sync* (not so important, because...)
 - ... **trusted (local) server only** (a **significant limitation** for some)

Please try it. [2023-10: several important new features already added!]



Any questions?

Thanks



Bibliography I

