Getting Started with

JMAP

the JSON Meta Application Protocol

okay, funny story

I pitched this talk:

Getting Started with

JMAP

the JSON Meta Application Protocol

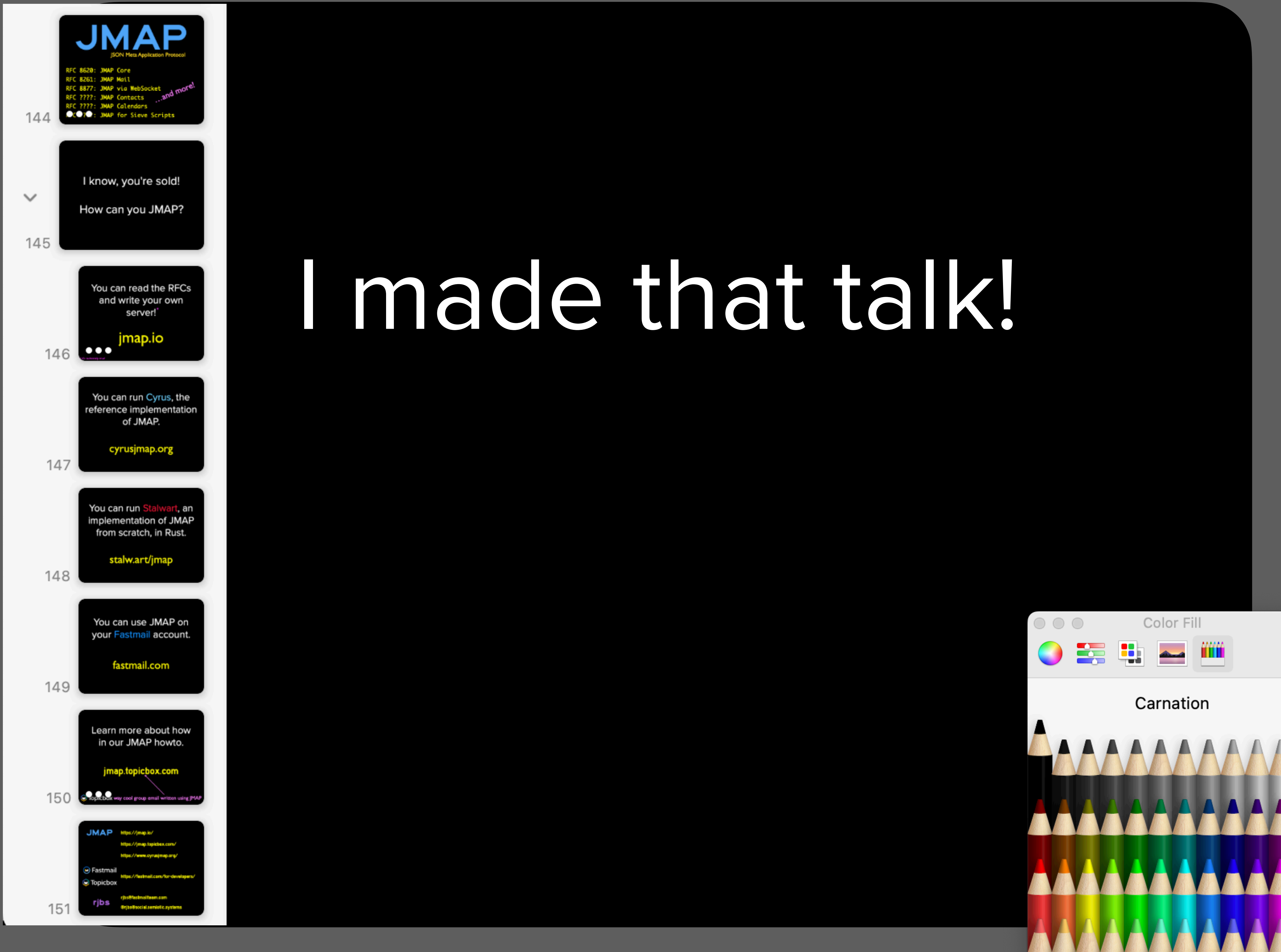
I pitched this talk:

Getting Started with

JMAP

the JSON Meta Application Protocol

- what is JMAP?
- how does it work?
- why is it so great?
- how do I get started
- what comes next?
- how does Fm use it?



144

JMAP

JSON Meta Application Protocol

```
RFC 8620: JMAP Core
RFC 8261: JMAP Mail
RFC 8877: JMAP via WebSocket
RFC ????: JMAP Contacts
RFC ????: JMAP Calendars
RFC ????: JMAP for Sieve Scripts
```

…and more!

145

I know, you're sold!

How can you JMAP?

146

You can read the RFCs and write your own server!

jmap.io

147

You can run Cyrus, the reference implementation of JMAP.

cyrusjmap.org

148

You can run Stalwart, an implementation of JMAP from scratch, in Rust.

stalw.art/jmap

149

You can use JMAP on your Fastmail account.

fastmail.com

150

Learn more about how in our JMAP howto.

jmap.topicbox.com

topicbox way cool group email written using JMAP

151

JMAP https://jmap.io/

https://jmap.topicbox.com/

https://www.cyrusjmap.org/

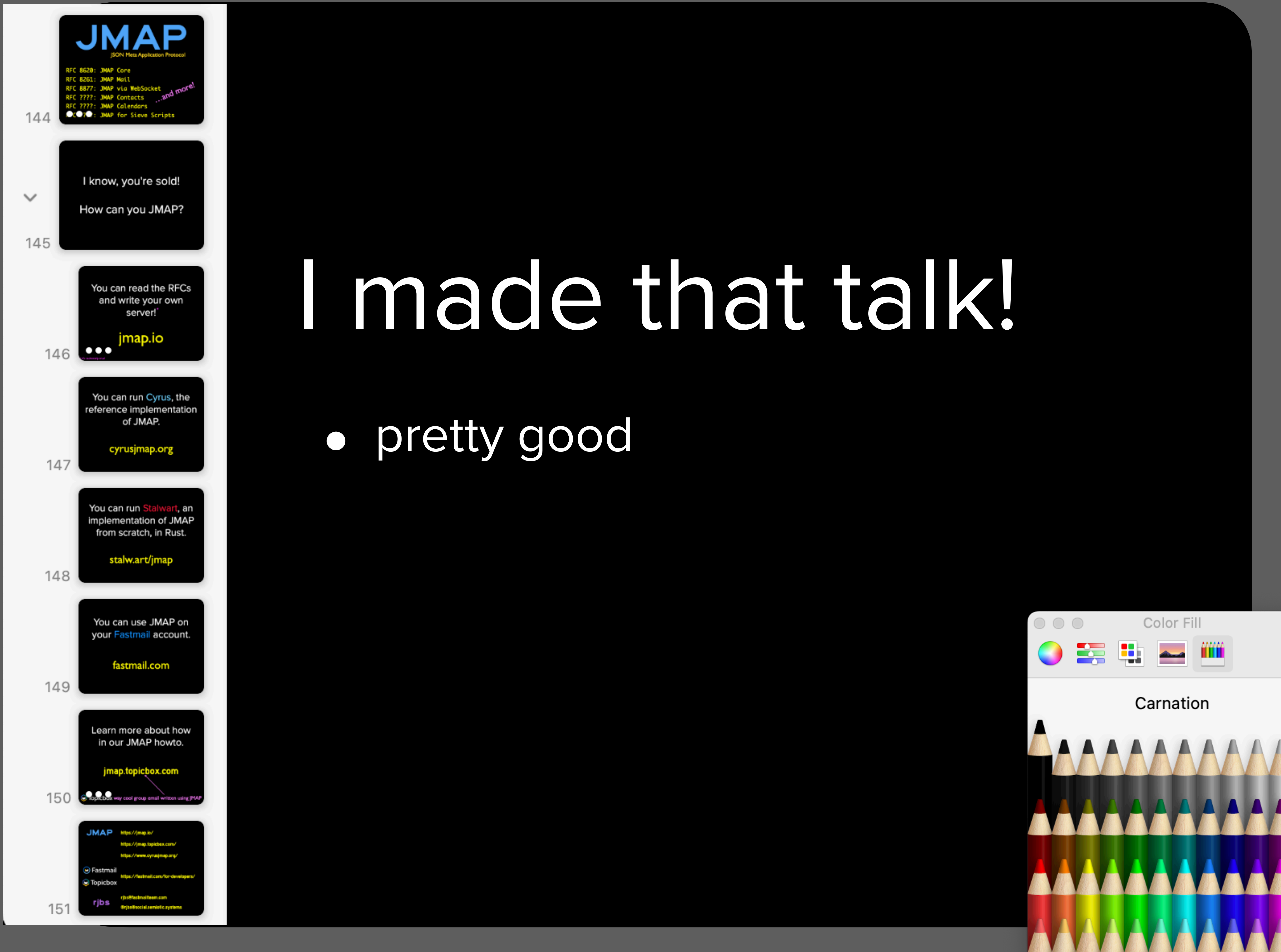
Fastmail

Topicbox https://fastmail.com/for-developers/

rjbs rjbs@fastmailteam.com

@rjbs@social.semiotic.systems

I made that talk!



I made that talk!

- pretty good

144
JMAP
JSON Meta Application Protocol

```
RFC 8620: JMAP Core
RFC 8261: JMAP Mail
RFC 8877: JMAP via WebSocket
RFC ????: JMAP Contacts
RFC ????: JMAP Calendars
RFC ????: JMAP for Sieve Scripts
```

...and more!

145
I know, you're sold!

How can you JMAP?

146
You can read the RFCs and write your own server!

jmap.io

147
You can run Cyrus, the reference implementation of JMAP.

cyrusjmap.org

148
You can run Stalwart, an implementation of JMAP from scratch, in Rust.

stalw.art/jmap

149
You can use JMAP on your Fastmail account.

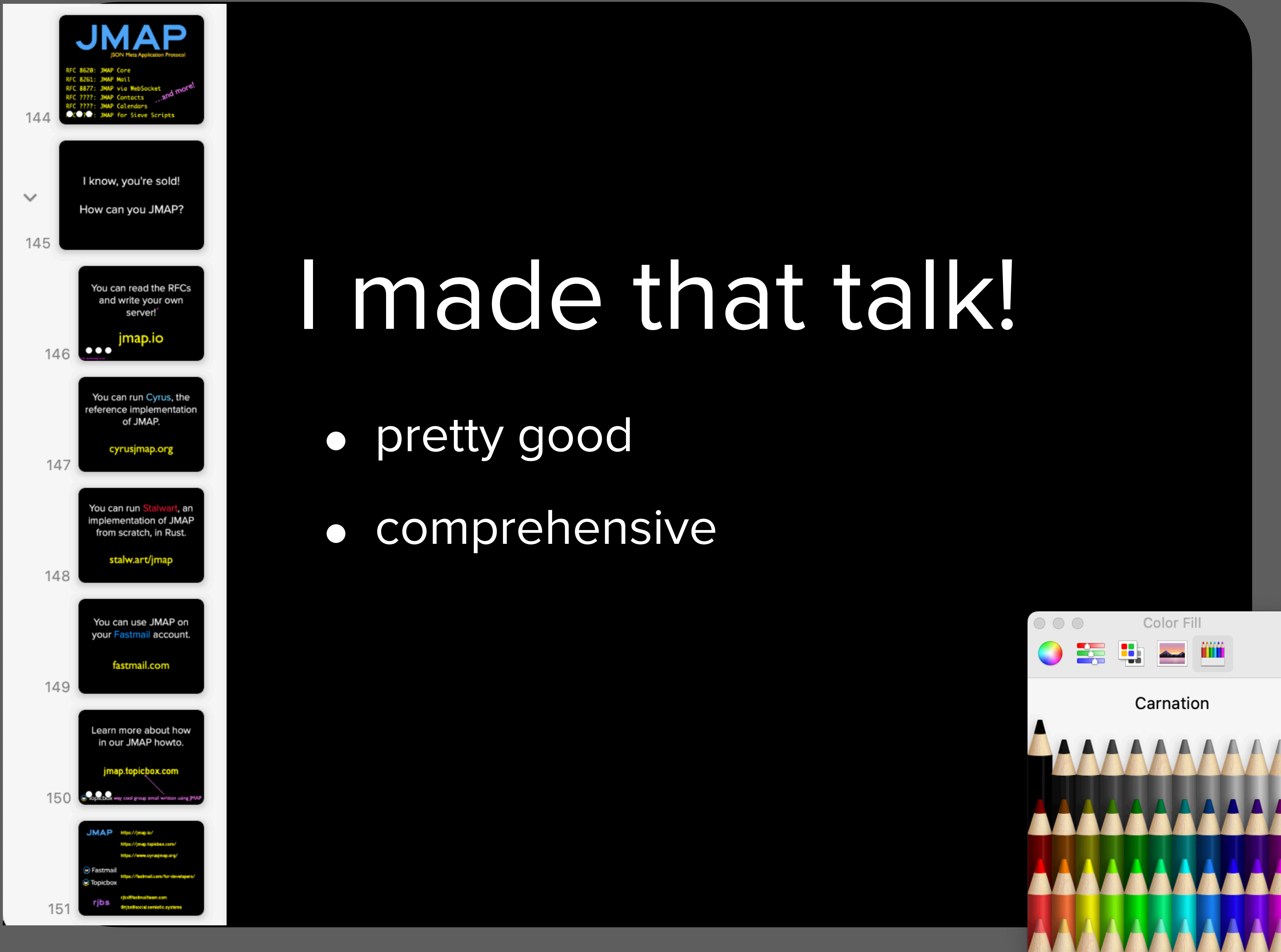
fastmail.com

150
Learn more about how in our JMAP howto.

jmap.topicbox.com

Topicbox way cool group email written using JMAP

151
JMAP https://jmap.io/
https://jmap.topicbox.com/
https://www.cyrusjmap.org/
Fastmail
Topicbox https://fastmail.com/for-developers/
rjbs rjbs@fastmailteam.com
@rjbs@social.semiotic.systems



I made that talk!

- pretty good
- comprehensive

144

JMAP

JSON Meta Application Protocol

```
RFC 8620: JMAP Core
RFC 8261: JMAP Mail
RFC 8877: JMAP via WebSocket
RFC ????: JMAP Contacts
RFC ????: JMAP Calendars
RFC ????: JMAP for Sieve Scripts
```

…and more!

145

I know, you're sold!

How can you JMAP?

146

You can read the RFCs and write your own server!

jmap.io

147

You can run Cyrus, the reference implementation of JMAP.

cyrusjmap.org

148

You can run Stalwart, an implementation of JMAP from scratch, in Rust.

stalw.art/jmap

149

You can use JMAP on your Fastmail account.

fastmail.com

150

Learn more about how in our JMAP howto.

jmap.topicbox.com

topicbox way cool group email written using JMAP

151

JMAP https://jmap.io/

https://jmap.topicbox.com/

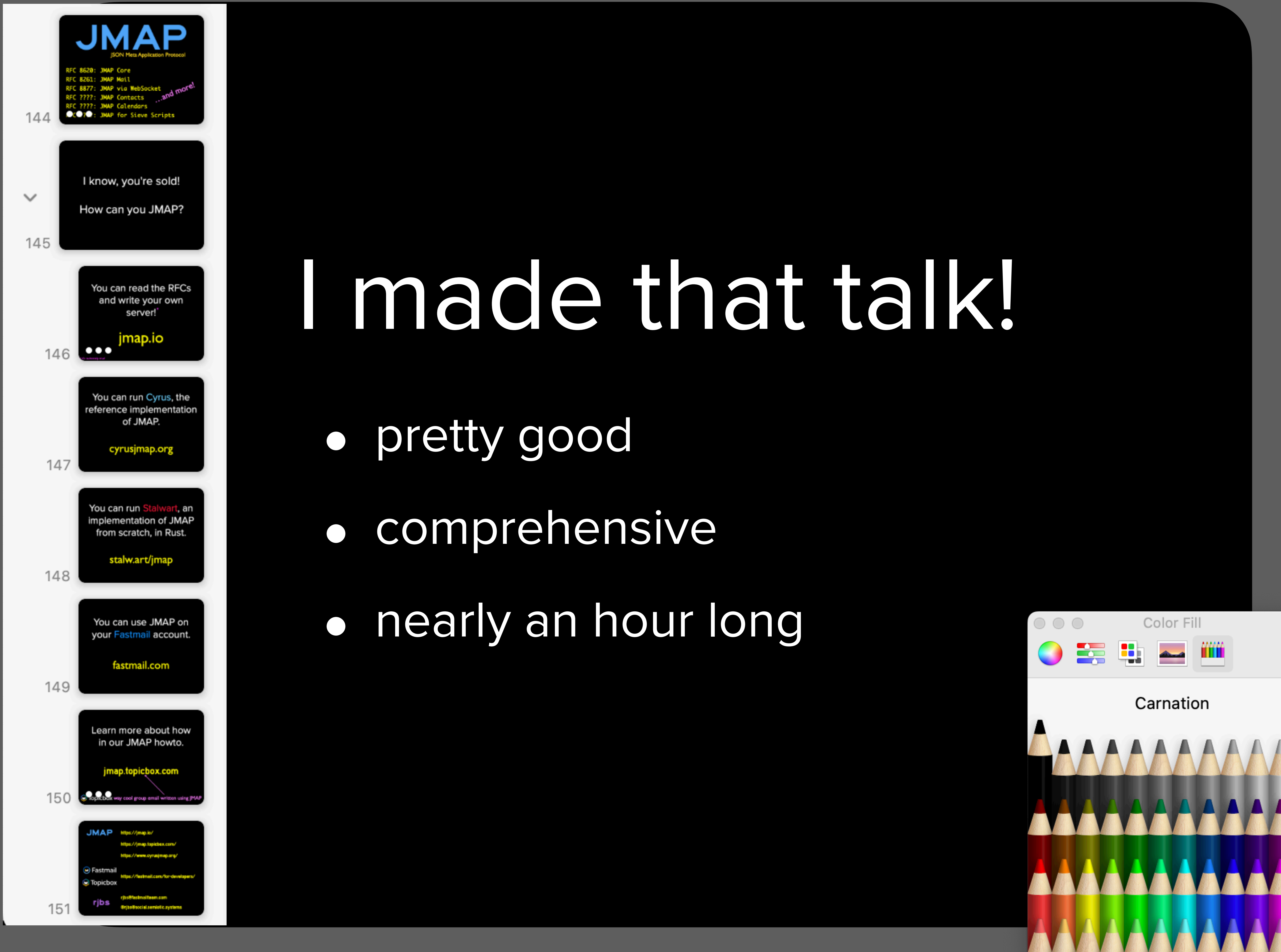
https://www.cyrusjmap.org/

Fastmail

Topicbox https://fastmail.com/for-developers/

rjbs rjbs@fastmailteam.com

@rjbs@social.semiotic.systems



I made that talk!

- pretty good
- comprehensive
- nearly an hour long

144

JMAP
JSON Meta Application Protocol

RFC 8620: JMAP Core
RFC 8261: JMAP Mail
RFC 8877: JMAP via WebSocket
RFC ????: JMAP Contacts
RFC ????: JMAP Calendars
RFC ????: JMAP for Sieve Scripts

…and more!

145

I know, you're sold!

How can you JMAP?

146

You can read the RFCs and write your own server!

jmap.io

147

You can run Cyrus, the reference implementation of JMAP.

cyrusjmap.org

148

You can run Stalwart, an implementation of JMAP from scratch, in Rust.

stalw.art/jmap

149

You can use JMAP on your Fastmail account.

fastmail.com

150

Learn more about how in our JMAP howto.

jmap.topicbox.com

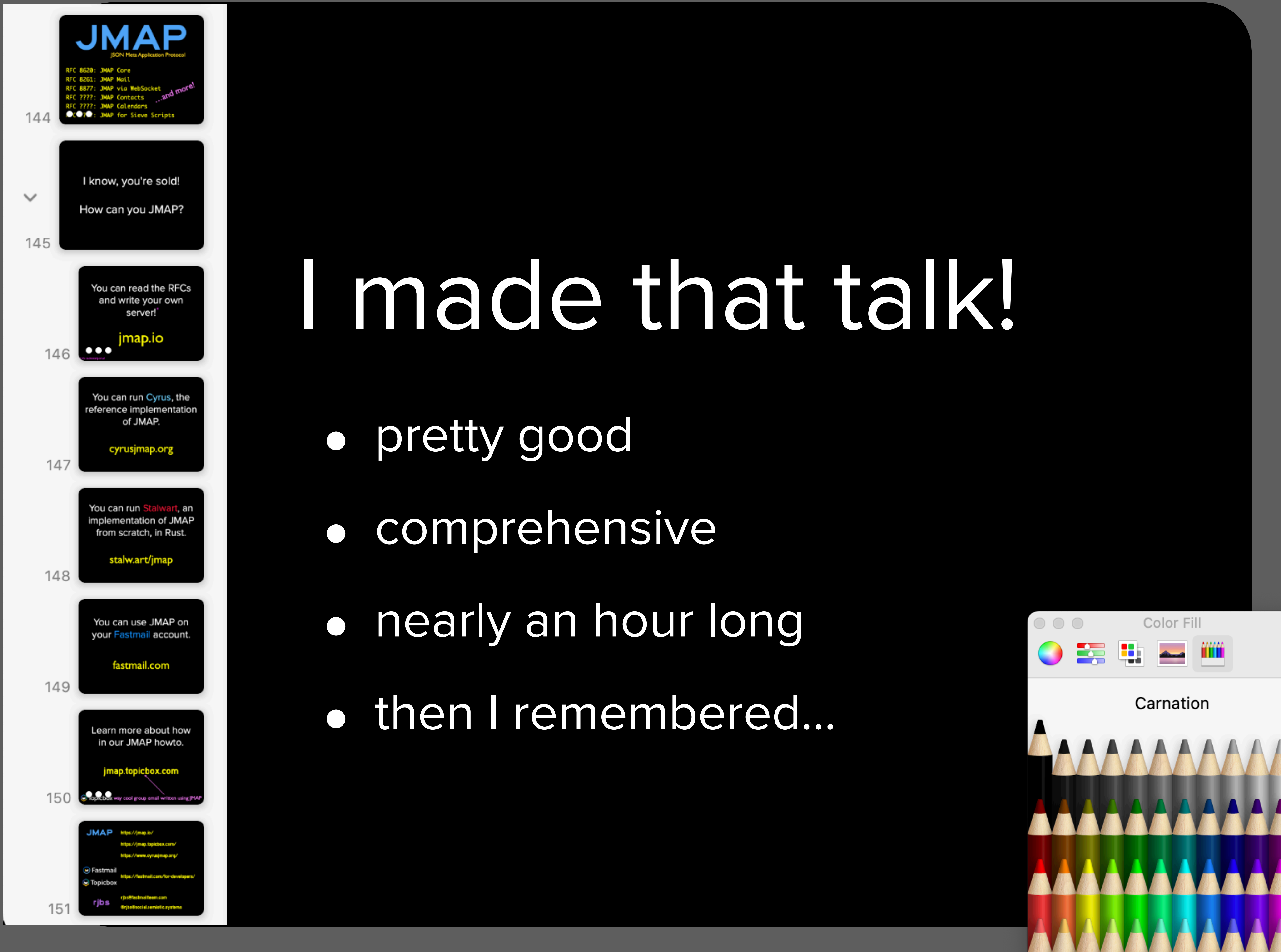
topicbox way cool group email written using JMAP

151

JMAP https://jmap.io/
https://jmap.topicbox.com/
https://www.cyrusjmap.org/

Fastmail
Topicbox https://fastmail.com/for-developers/

rjbs rjbs@fastmailteam.com
@rjbs@social.semiotic.systems



I made that talk!

- pretty good
- comprehensive
- nearly an hour long
- then I remembered…

144
JMAP
JSON Meta Application Protocol
RFC 8620: JMAP Core
RFC 8261: JMAP Mail
RFC 8877: JMAP via WebSocket
RFC ????: JMAP Contacts
RFC ????: JMAP Calendars
RFC ????: JMAP for Sieve Scripts
…and more!

145
I know, you're sold!

How can you JMAP?

146
You can read the RFCs and write your own server!

jmap.io

147
You can run Cyrus, the reference implementation of JMAP.

cyrusjmap.org

148
You can run Stalwart, an implementation of JMAP from scratch, in Rust.

stalw.art/jmap

149
You can use JMAP on your Fastmail account.

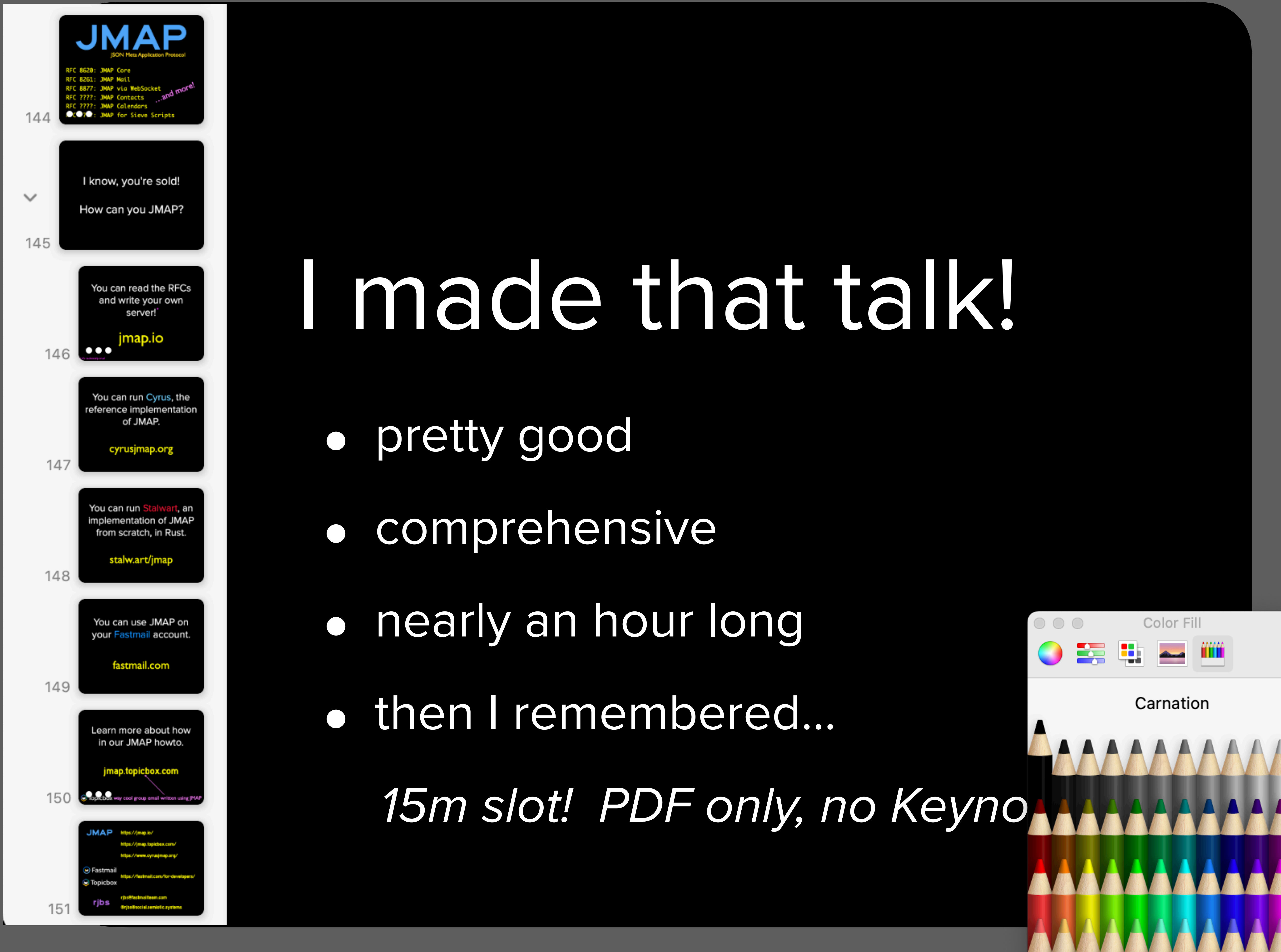
fastmail.com

150
Learn more about how in our JMAP howto.

jmap.topicbox.com

topicbox way cool group email written using JMAP

151
JMAP https://jmap.io/
https://jmap.topicbox.com/
https://www.cyrusjmap.org/
Fastmail
Topicbox https://fastmail.com/for-developers/
rjbs rjbs@fastmailteam.com
@rjbs@social.semiotic.systems



I made that talk!

- pretty good
- comprehensive
- nearly an hour long
- then I remembered…

15m slot! PDF only, no Keyno

144

JMAP

JSON Meta Application Protocol

RFC 8620: JMAP Core
RFC 8261: JMAP Mail
RFC 8877: JMAP via WebSocket
RFC ????: JMAP Contacts
RFC ????: JMAP Calendars
: JMAP for Sieve Scripts

…and more!

145

I know, you're sold!

How can you JMAP?

146

You can read the RFCs and write your own server!

jmap.io

147

You can run Cyrus, the reference implementation of JMAP.

cyrusjmap.org

148

You can run Stalwart, an implementation of JMAP from scratch, in Rust.

stalw.art/jmap

149

You can use JMAP on your Fastmail account.

fastmail.com

150

Learn more about how in our JMAP howto.

jmap.topicbox.com

topicbox way cool group email written using JMAP

151

JMAP https://jmap.io/
https://jmap.topicbox.com/
https://www.cyrusjmap.org/

Fastmail
Topicbox https://fastmail.com/for-developers/

rjbs rjbs@fastmailteam.com
@rjbs@social.semiotic.systems

so, I present…

JMAP

the absolute minimum.pdf

if you want to hear the
whole thing,
it can be arranged

if you want to hear the whole thing,

it can be arranged

Trappistes Rochefort
PRODUCED BY ABBAYE N.-D. DE SAINT-REMY - ROCHEFORT - BELGIUM
Trappistes Rochefort
8
BELGIAN ALE

Fastmail

making email better

Ricardo Signes

rjbs

📧 email, calendars, and contacts

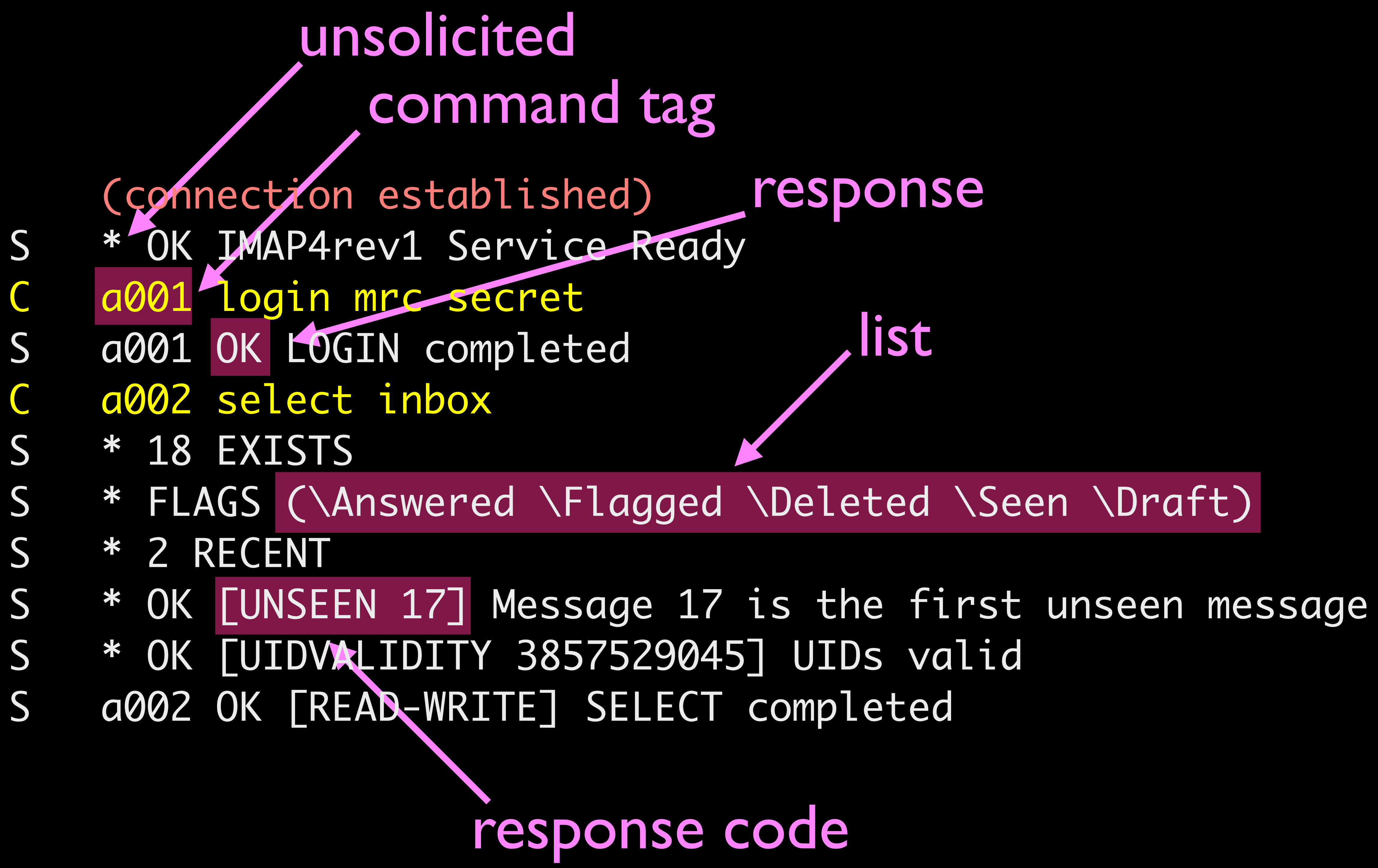
🦘 in Melbourne, AU

🥨 and Philadelphia, US

💙 we really love email

but first, IMAP

```
    (connection established)
S   * OK IMAP4rev1 Service Ready
C   a001 login mrc secret
S   a001 OK LOGIN completed
C   a002 select inbox
S   * 18 EXISTS
S   * FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
S   * 2 RECENT
S   * OK [UNSEEN 17] Message 17 is the first unseen message
S   * OK [UIDVALIDITY 3857529045] UIDs valid
S   a002 OK [READ-WRITE] SELECT completed
```



unsolicited

command tag

response

list

response code

```
    (connection established)
S   * OK IMAP4rev1 Service Ready
C   a001 login mrc secret
S   a001 OK LOGIN completed
C   a002 select inbox
S   * 18 EXISTS
S   * FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
S   * 2 RECENT
S   * OK [UNSEEN 17] Message 17 is the first unseen message
S   * OK [UIDVALIDITY 3857529045] UIDs valid
S   a002 OK [READ-WRITE] SELECT completed
```

(to say nothing of the payload)

```
Received: from mailmx.nyi.internal (localhost [127.0.0.1])
    by mx2.messagingengine.com (Authentication Milter) with ESMTP
    id 6E4A997388F.0E48A6A00C2;
    Thu, 25 Jan 2024 04:30:29 -0500
ARC-Seal: i=4; a=rsa-sha256; cv=pass; d=messagingengine.com; s=fm3; t=
    1706175029; b=hs+FUbwu+LcAUex9jy5wTnAL0SkZDfeFW9HlCmIGk4wdNsV6ZW
    4/HaLPUxFq7Auhg4Insp7Ocl7WVhQuhoftIwdm5zUSvsrFniB+RM5qxoesH0RKKY
    g43DZZTIeg4iSyj+Ipwi5dHB4/1595CmPBjrm0toGGjfg+YVC/Desew0kBKh4ulU
    V1IfLXCjLki8xJEy/DjLI+osVes4KUW5AlrFWAl+hPKOd1IVPsn05szc9uaiQOt9
    nibH4oDfECCruYrFEFH6MRU8d3rPaRQh1K7EQVdf6KmA4ZlU2NZF8TknFUefrql+
    /9rdrw1aYocMXnSFfxOrCxOKdc+iC/QO6tHw==
X-Received-x-me-csa: (Received x-me-csa header removed by
    mx2.messagingengine.com) none


------=_NextPart_000_0446_01DA4F79.5FD0FDA0
Content-Type: multipart/alternative;
 boundary="----=_NextPart_001_0447_01DA4F79.5FD0FDA0"
Content-Transfer-Encoding: 7bit
Content-ID: <17061750230.cBcEFDe57.528484@tb-mx1>


------=_NextPart_001_0447_01DA4F79.5FD0FDA0
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: quoted-printable
```

whatever you want to say
about HTTP+JSON

at least it isn't *that*

You probably all know how to use HTTP

You probably all know how to use HTTP

even if you don't know how it works

You probably all know how to use HTTP

even if you don't know how it works

and you probably *do* know how it works.

omitted here:

lots of slides about how weird IMAP is

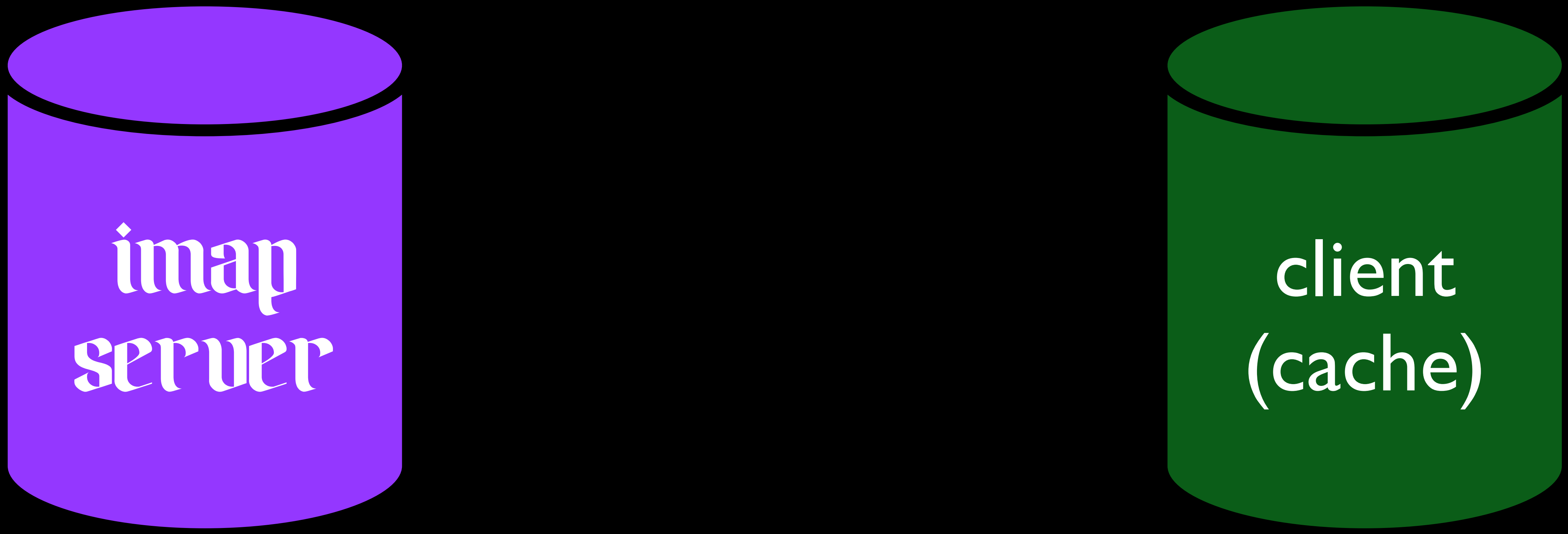
```
C   a004 fetch 12 body[header]
S   * 12 FETCH (BODY[HEADER] {342}
S   Date: Wed, 17 Jul 1996 02:23:25 -0700 (PDT)
S   From: Terry Gray <gray@cac.washington.edu>
S   Subject: IMAP4rev1 WG mtg summary and minutes
S   To: imap@cac.washington.edu
S   cc: minutes@CNRI.Reston.VA.US, John Klensin <KLENSIN@MIT.EDU>
S   Message-Id: <B27397-0100000@cac.washington.edu>
S   MIME-Version: 1.0
S   Content-Type: TEXT/PLAIN; CHARSET=US-ASCII
S
S   )
S   a004 OK FETCH completed
C   a005 store 12 +flags \deleted
S   * 12 FETCH (FLAGS (\Seen \Deleted))
S   a005 OK +FLAGS completed
```

```
C   a004 fetch 12 body[header]
S   * 12 FETCH (BODY[HEADER] {342}
S   Date: Wed, 17 Jul 1996 02:23:25 -0700 (PDT)
S   From: Terry Gray <gray@cac.washington.edu>
S   Subject: IMAP4rev1 WG mtg summary and minutes
S   To: imap@cac.washington.edu
S   cc: minutes@CNRI.Reston.VA.US, John Klensin <KLENSIN@MIT.EDU>
S   Message-Id: <B27397-0100000@cac.washington.edu>
S   MIME-Version: 1.0
S   Content-Type: TEXT/PLAIN; CHARSET=US-ASCII
S
S   )
S   a004 OK FETCH completed
C   a005 store 12 +flags \deleted
S   * 12 FETCH (FLAGS (\Seen \Deleted))
S   a005 OK +FLAGS completed
```

```
C    a004 fetch 12 body[header]
S    * 12 FETCH (BODY[HEADER] {342}
S    Date: Wed, 17 Jul 1996 02:23:25 -0700 (PDT)
S    From: Terry Gray <gray@cac.washington.edu>
S    Subject: IMAP4rev1 WG mtg summary and minutes
S    To: imap@cac.washington.edu
S    cc: minutes@CNRI.Reston.VA.US, John Klensin <KLENSIN@MIT.EDU>
S    Message-Id: <B27397-0100000@cac.washington.edu>
S    MIME-Version: 1.0
S    Content-Type: TEXT/PLAIN; CHARSET=US-ASCII
S
S    )
S    a004 OK FETCH completed
C    a005 store 12 +flags \deleted
S    * 12 FETCH (FLAGS (\Seen \Deleted))
S    a005 OK +FLAGS completed
```

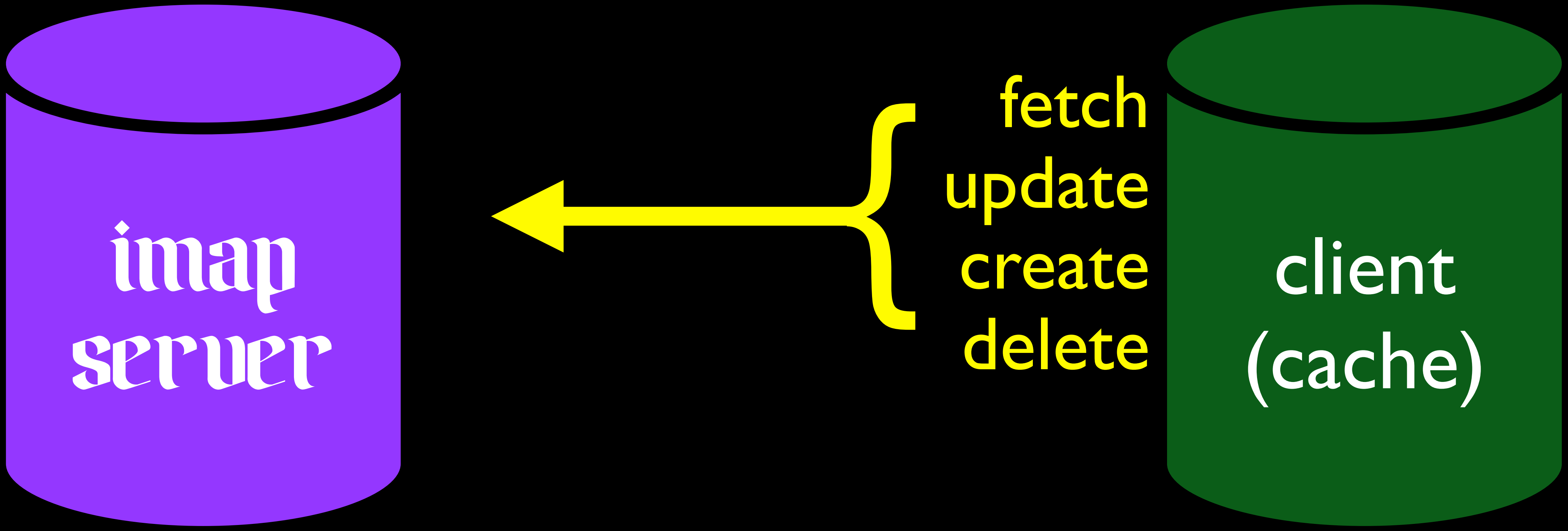
sent store, got fetch!

IMAP is a cache management protocol



imap server

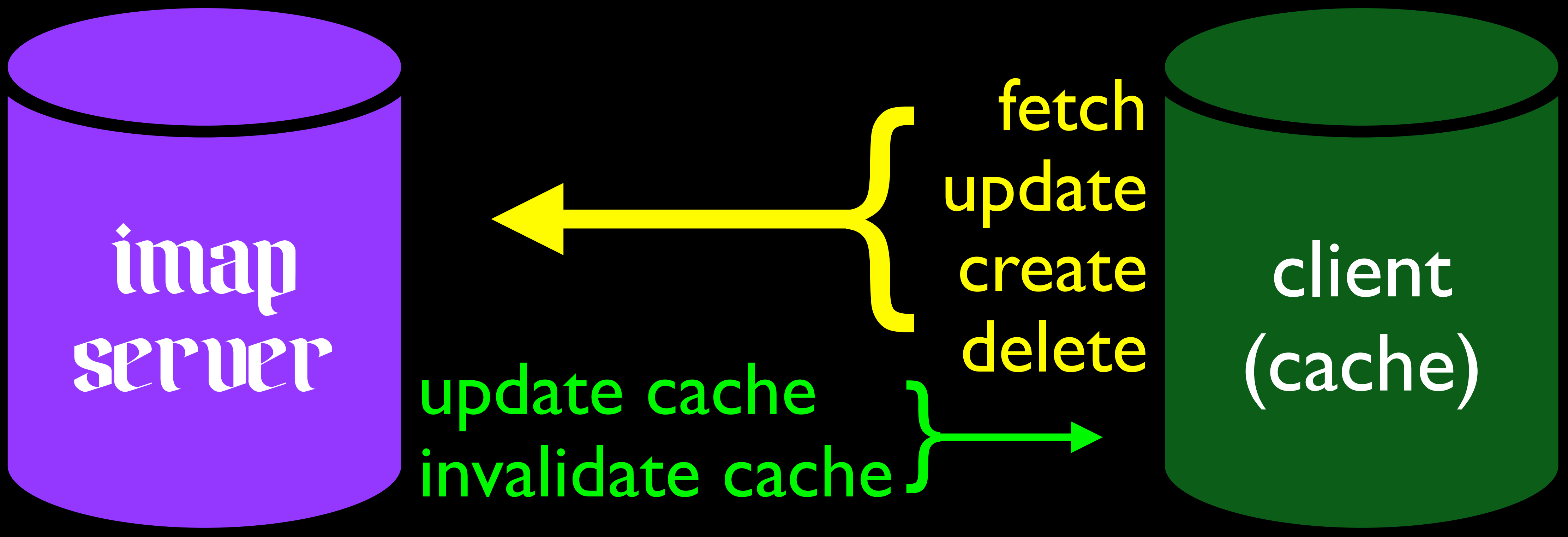
client (cache)

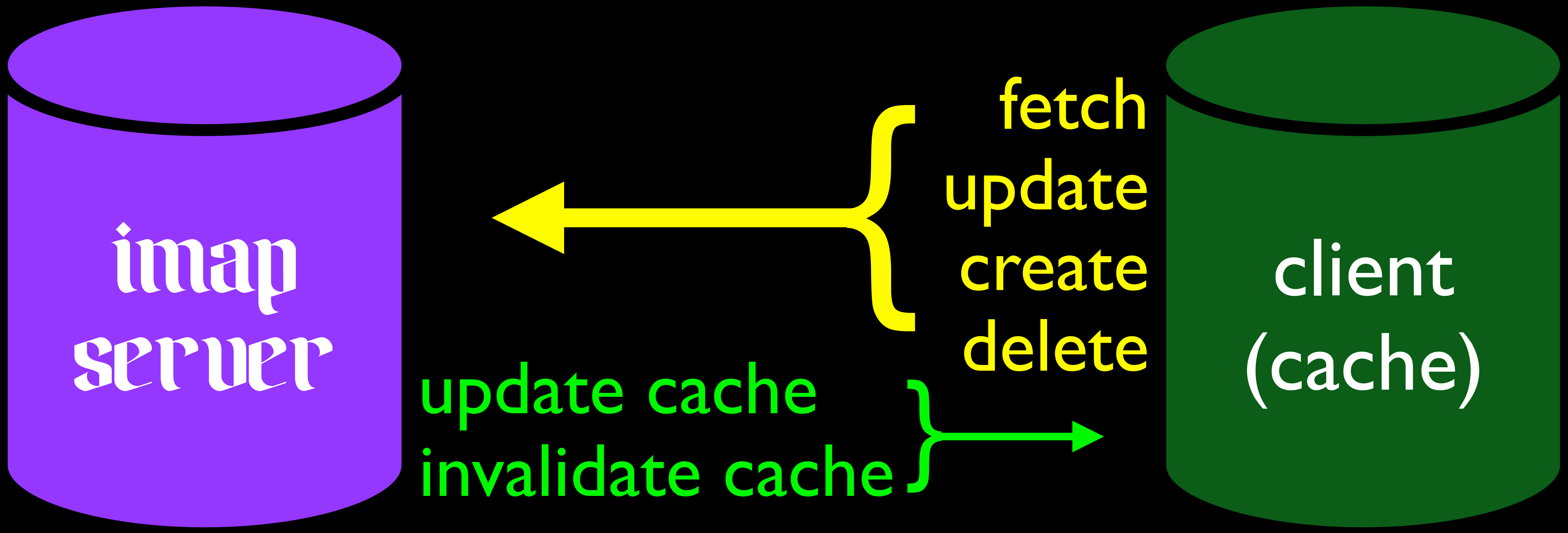


imap server

fetch
update
create
delete

client (cache)


imap server
fetch
update
create
delete
client (cache)
update cache
invalidate cache



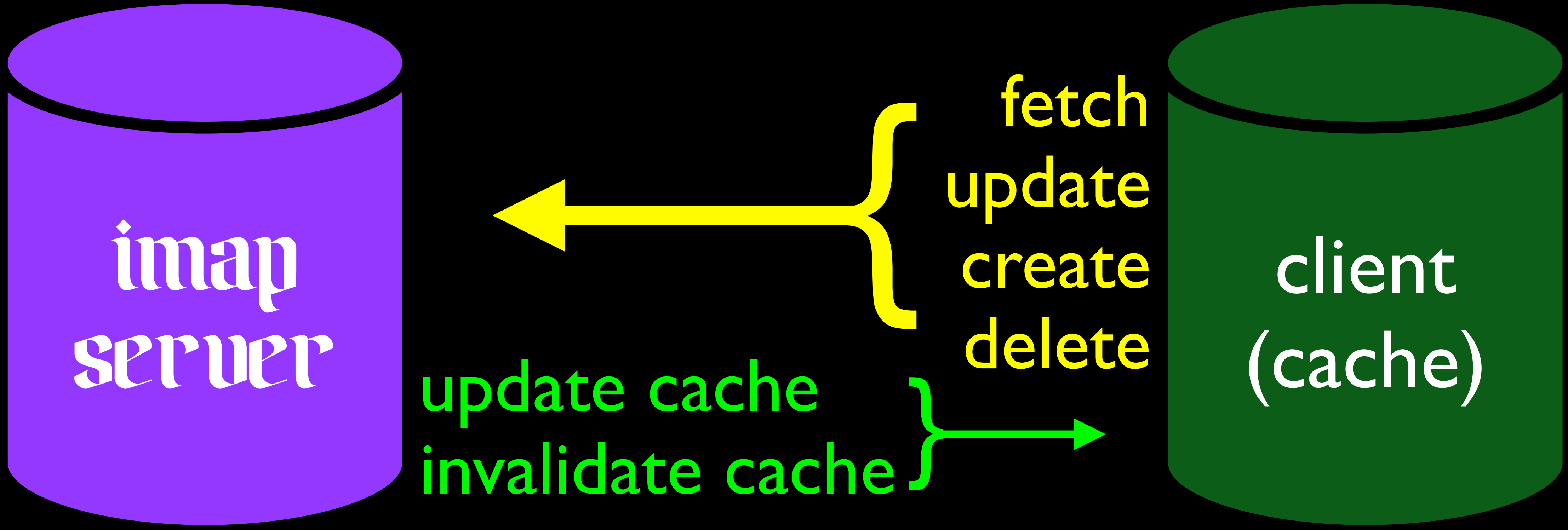
imap server

fetch
update
create
delete

client (cache)

update cache
invalidate cache

client means	client sends	server replies	server means



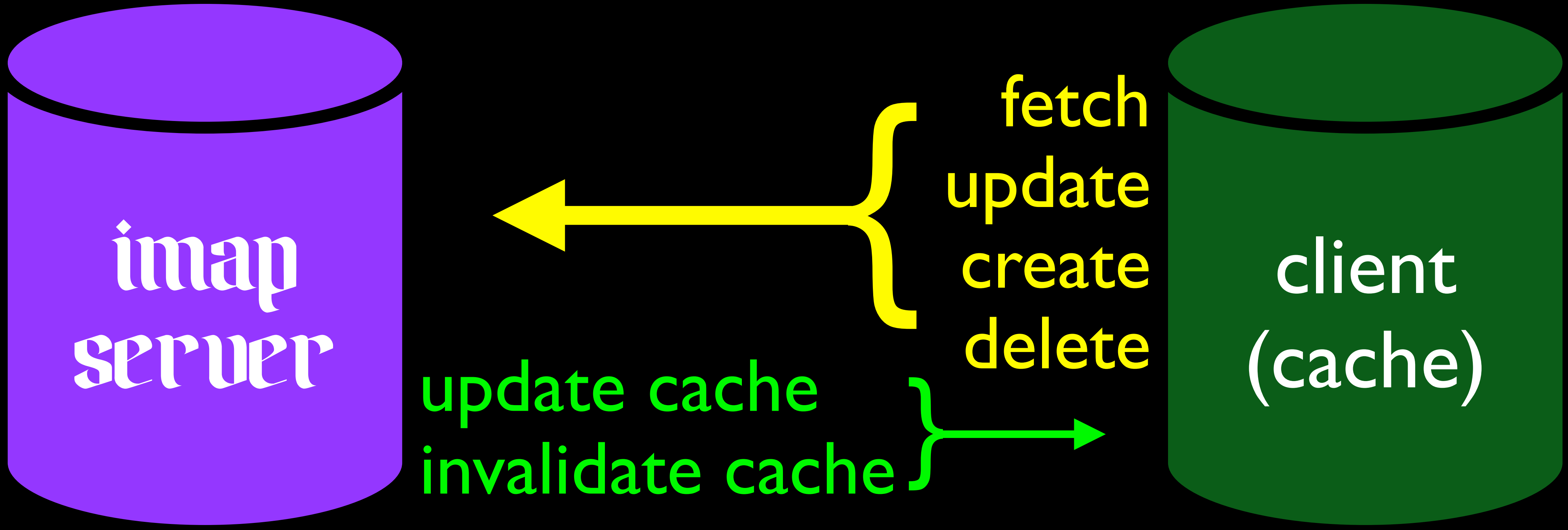
imap server

fetch
update
create
delete

update cache
invalidate cache

client (cache)

client means	client sends	server replies	server means
I want to work with the inbox.			



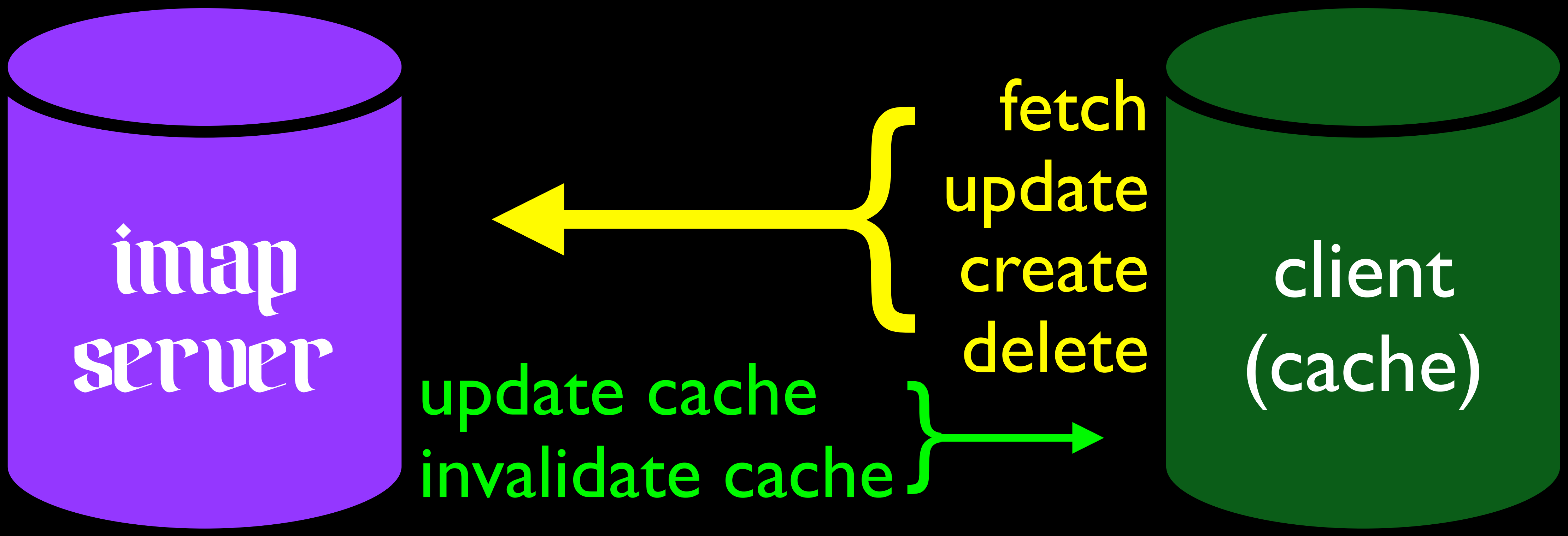
imap server

fetch
update
create
delete

update cache
invalidate cache

client (cache)

client means	client sends	server replies	server means
I want to work with the inbox.	SELECT INBOX		



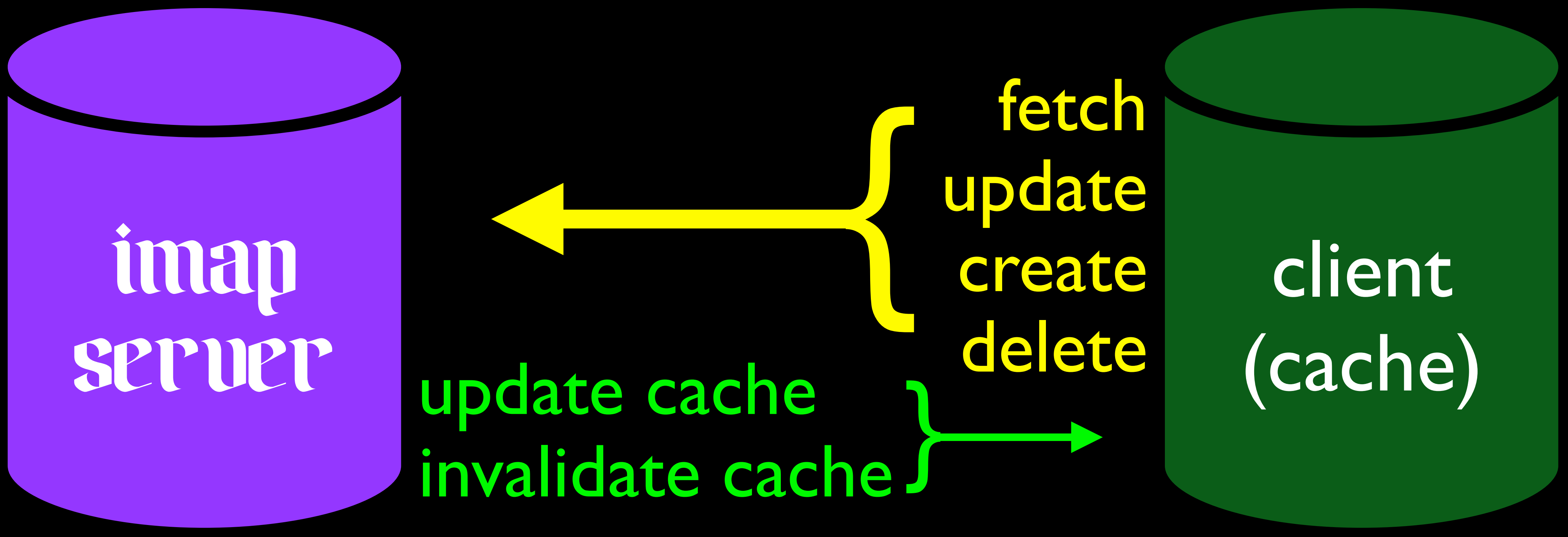
imap server

fetch
update
create
delete

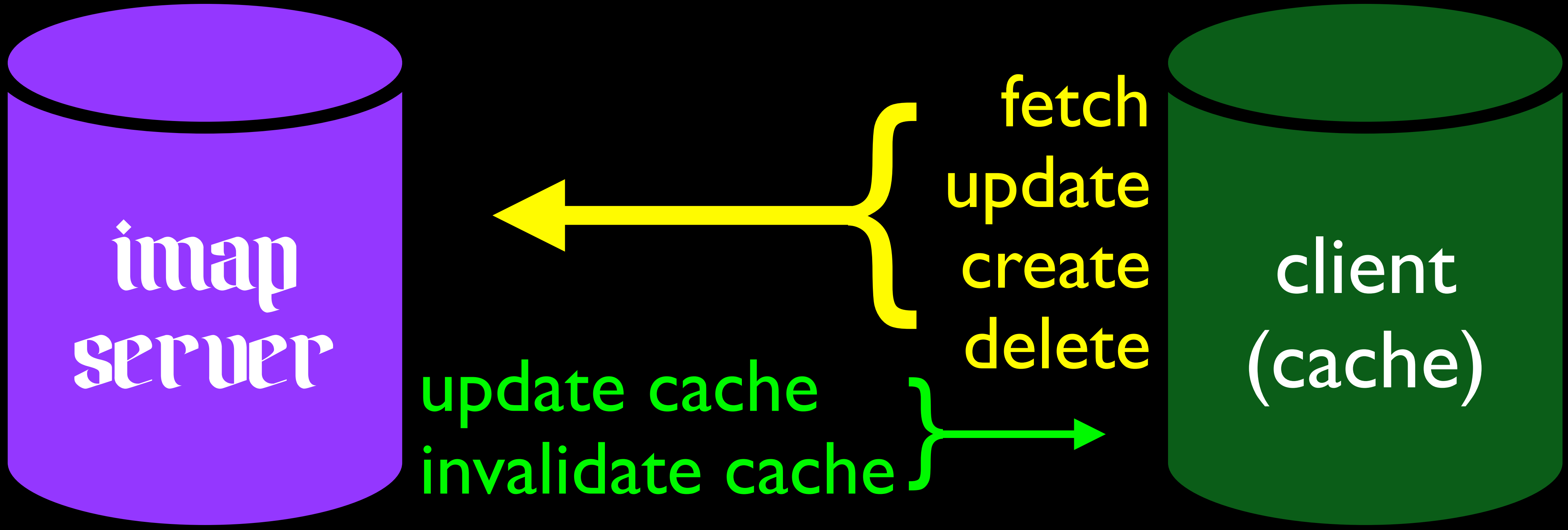
client (cache)

update cache
invalidate cache

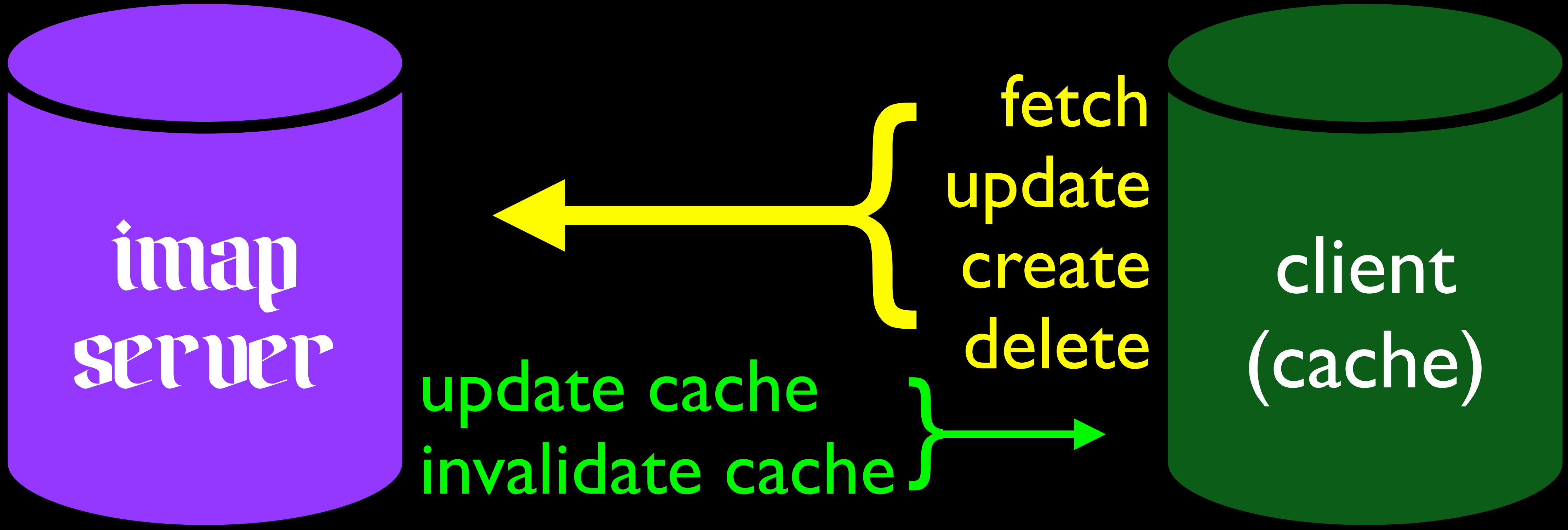
client means	client sends	server replies	server means
I want to work with the inbox.	SELECT INBOX	172 EXISTS FLAGS (…)	



client means	client sends	server replies	server means
I want to work with the inbox.	SELECT INBOX	172 EXISTS FLAGS (…)	Prep a cache. Here's how.



client means	client sends	server replies	server means
I want to work with the inbox.	SELECT INBOX	172 EXISTS FLAGS (…)	Prep a cache. Here's how.
I want to see new mail.			



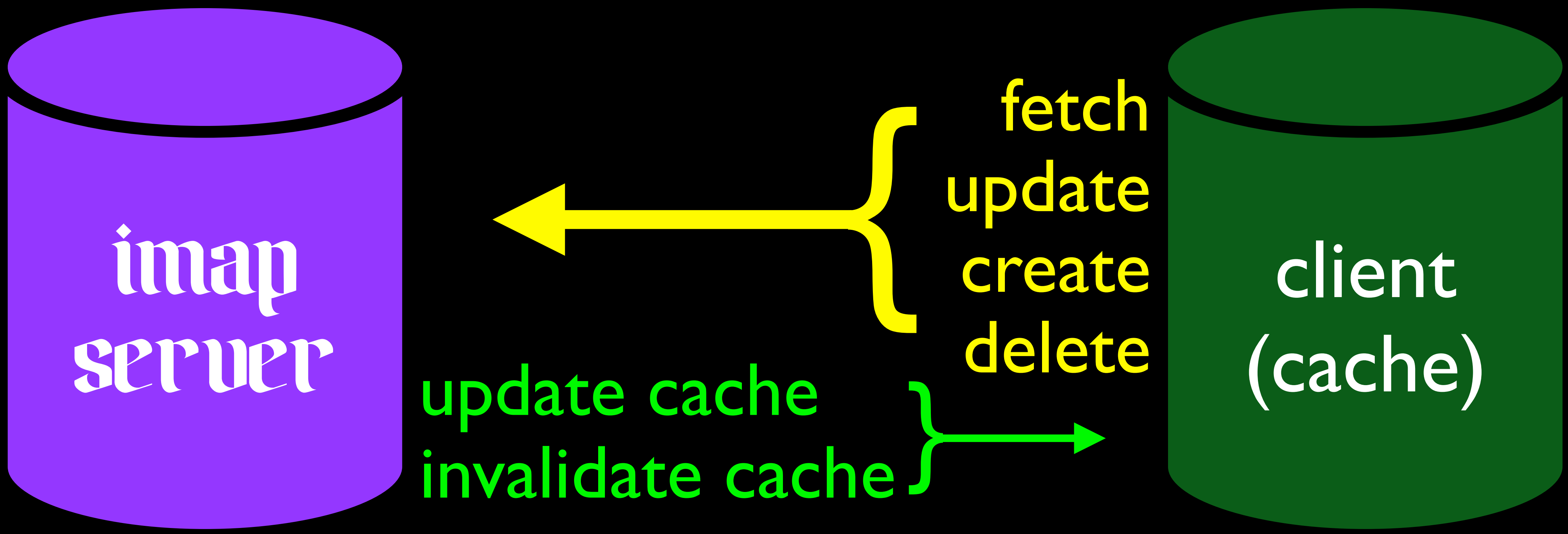
imap
server

fetch
update
create
delete

update cache
invalidate cache

client
(cache)

client means	client sends	server replies	server means
I want to work with the inbox.	SELECT INBOX	172 EXISTS FLAGS (…)	Prep a cache. Here's how.
I want to see new mail.	FETCH 12:* ENVELOPE		



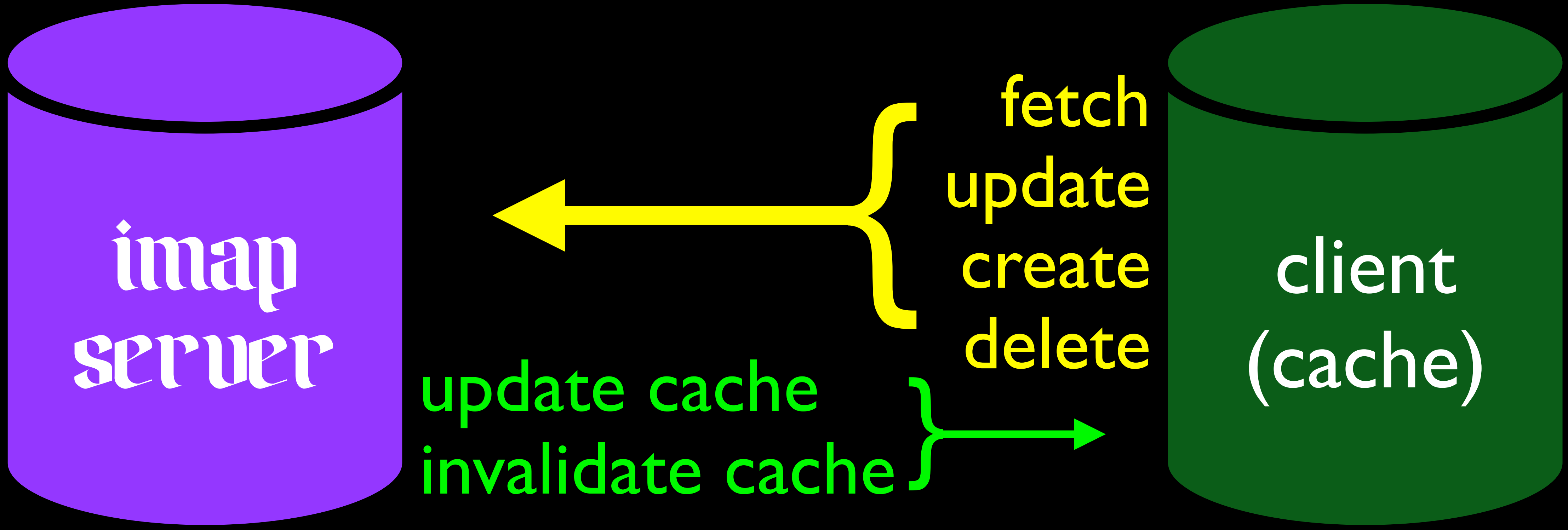
imap server

fetch
update
create
delete

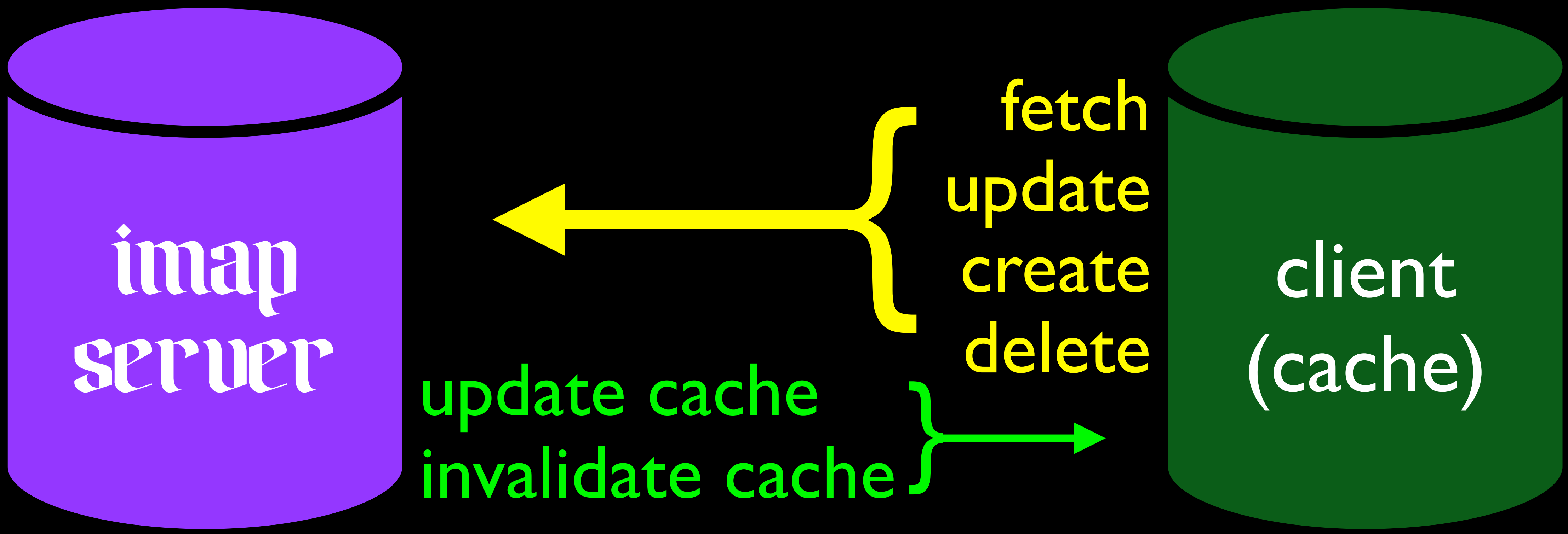
client (cache)

update cache
invalidate cache

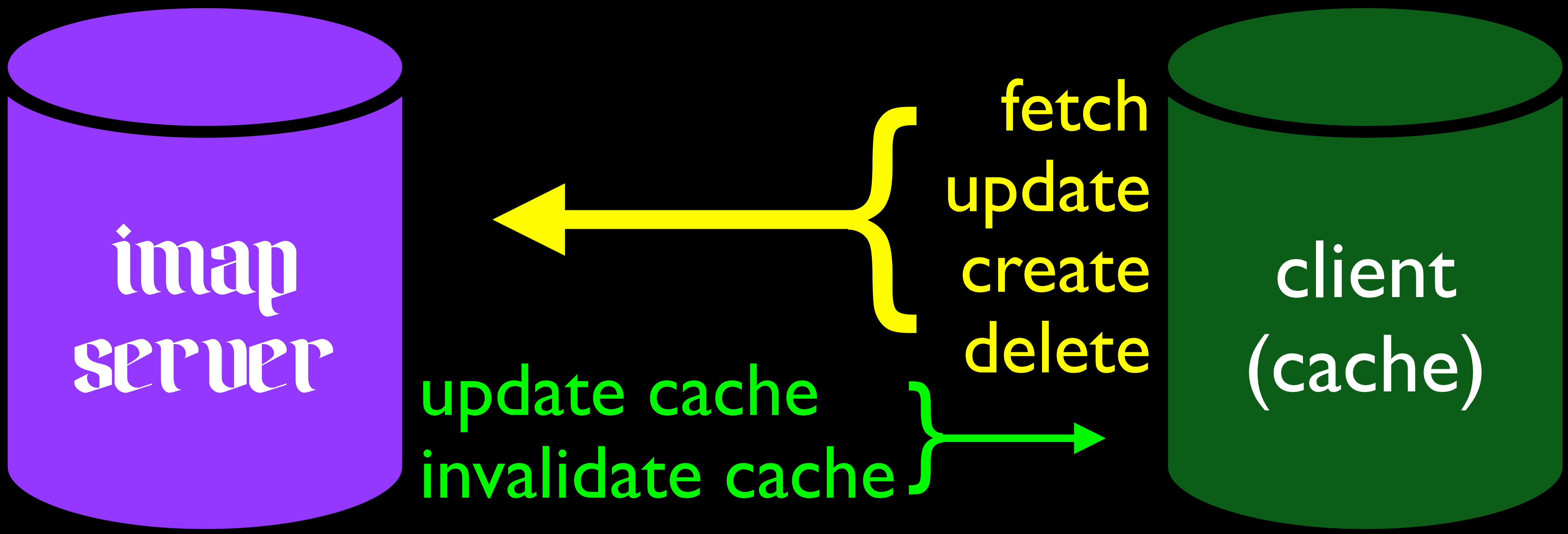
client means	client sends	server replies	server means
I want to work with the inbox.	SELECT INBOX	172 EXISTS FLAGS (…)	Prep a cache. Here's how.
I want to see new mail.	FETCH 12:* ENVELOPE	* 13 FETCH … * 14 FETCH …	



client means	client sends	server replies	server means
I want to work with the inbox.	SELECT INBOX	172 EXISTS FLAGS (…)	Prep a cache. Here's how.
I want to see new mail.	FETCH 12:* ENVELOPE	* 13 FETCH … * 14 FETCH …	Update your cache.



client means	client sends	server replies	server means
I want to work with the inbox.	SELECT INBOX	172 EXISTS FLAGS (…)	Prep a cache. Here's how.
I want to see new mail.	FETCH 12:* ENVELOPE	* 13 FETCH … * 14 FETCH …	Update your cache.
I want to mark this read.			



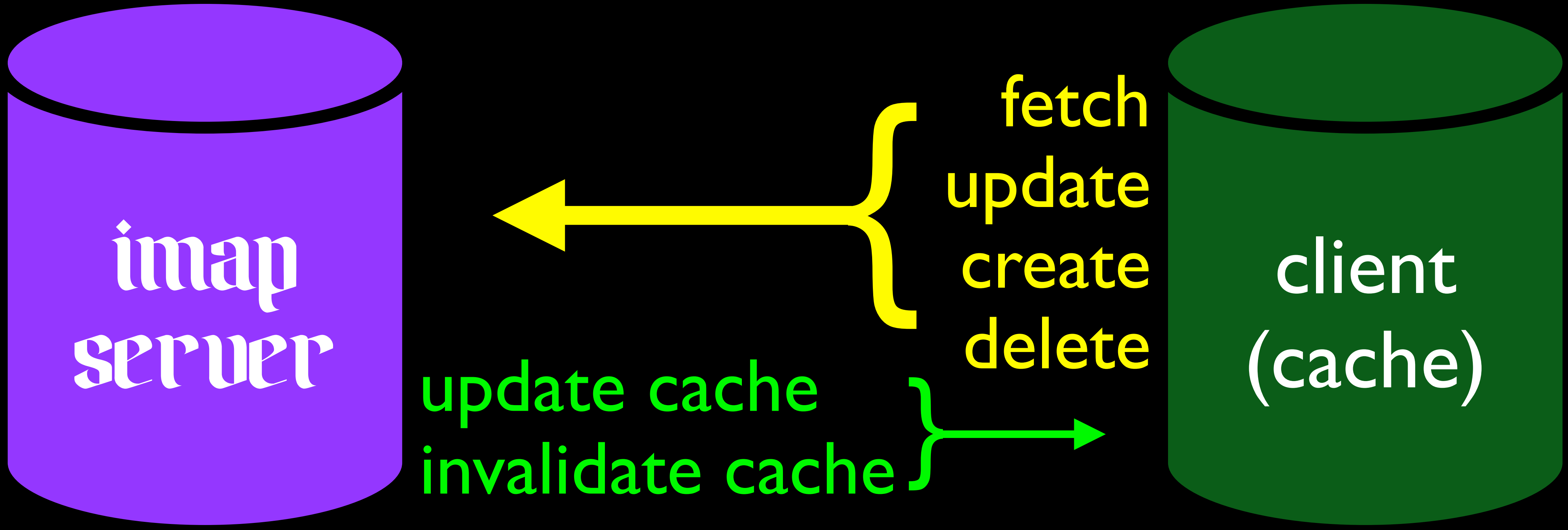
imap server

fetch
update
create
delete

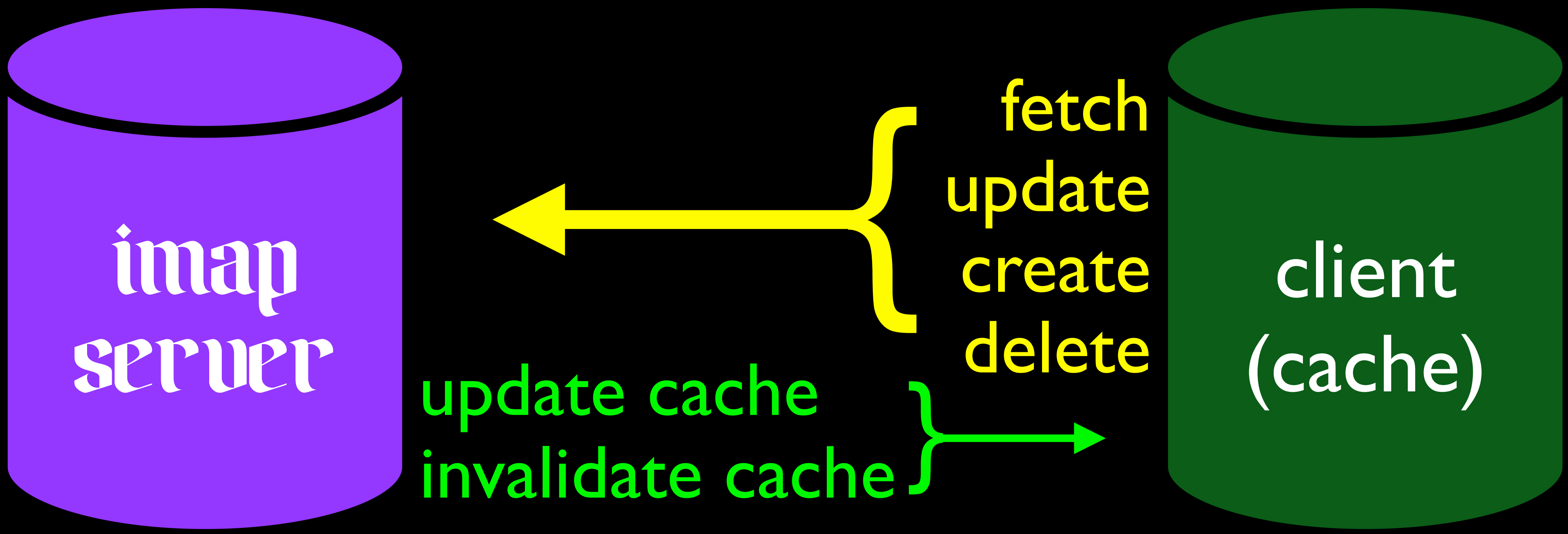
client (cache)

update cache
invalidate cache

client means	client sends	server replies	server means
I want to work with the inbox.	SELECT INBOX	172 EXISTS FLAGS (…)	Prep a cache. Here's how.
I want to see new mail.	FETCH 12:* ENVELOPE	* 13 FETCH … * 14 FETCH …	Update your cache.
I want to mark this read.	STORE 12 +FLAGS \Seen		



client means	client sends	server replies	server means
I want to work with the inbox.	SELECT INBOX	172 EXISTS FLAGS (…)	Prep a cache. Here's how.
I want to see new mail.	FETCH 12:* ENVELOPE	* 13 FETCH … * 14 FETCH …	Update your cache.
I want to mark this read.	STORE 12 +FLAGS \Seen	* 12 FETCH …	



imap server

fetch
update
create
delete

update cache
invalidate cache

client (cache)

client means	client sends	server replies	server means
I want to work with the inbox.	SELECT INBOX	172 EXISTS FLAGS (…)	Prep a cache. Here's how.
I want to see new mail.	FETCH 12:* ENVELOPE	* 13 FETCH … * 14 FETCH …	Update your cache.
I want to mark this read.	STORE 12 +FLAGS \Seen	* 12 FETCH …	Update your cache.

```
C   A03 SELECT INBOX (QRESYNC (67890007 123 41,43:211,214:541))
S   * OK [CLOSED]
S   * 314 EXISTS
S   * 15 RECENT
S   * OK [UIDVALIDITY 67890007] UIDVALIDITY
S   * OK [UIDNEXT 567] Predicted next UID
S   * OK [HIGHESTMODSEQ 130]
S   * OK [UNSEEN 7] There are some unseen messages in the mailbox
S   * FLAGS (\Answered \Flagged \Draft \Deleted \Seen)
S   * OK [PERMANENTFLAGS (\Answered \Flagged \Draft \Deleted \Seen \*)]
S   * VANISHED (EARLIER) 41,43:116,118,120:211,214:540
S   * 49 FETCH (UID 117 FLAGS (\Seen \Answered) MODSEQ (128))
S   * 50 FETCH (UID 119 FLAGS (\Draft $MDNSent) MODSEQ (129))
S   ...
S   * 100 FETCH (UID 541 FLAGS (\Seen $Forwarded) MODSEQ (130))
S   A03 OK [READ-WRITE] mailbox selected
```

```
C   A03 SELECT INBOX (QRESYNC (67890007 123 41,43:211,214:541))
S   * OK [CLOSED]
S   * 314 EXISTS
S   * 15 RECENT
S   * OK [UIDVALIDITY 67890007] UIDVALIDITY
S   * OK [UIDNEXT 567] Predicted next UID
S   * OK [HIGHESTMODSEQ 130]
S   * OK [UNSEEN 7] There are some unseen messages in the mailbox
S   * FLAGS (\Answered \Flagged \Draft \Deleted \Seen)
S   * OK [PERMANENTFLAGS (\Answered \Flagged \Draft \Deleted \Seen \*)]
S   * VANISHED (EARLIER) 41,43:116,118,120:211,214:540
S   * 49 FETCH (UID 117 FLAGS (\Seen \Answered) MODSEQ (128))
S   * 50 FETCH (UID 119 FLAGS (\Draft $MDNSent) MODSEQ (129))
S   ...
S   * 100 FETCH (UID 541 FLAGS (\Seen $Forwarded) MODSEQ (130))
S   A03 OK [READ-WRITE] mailbox selected
```

resync

```
C    A03 SELECT INBOX (QRESYNC (67890007 123 41,43:211,214:541))
S    * OK [CLOSED]
S    * 314 EXISTS
S    * 15 RECENT
S    * OK [UIDVALIDITY 67890007] UIDVALIDITY
S    * OK [UIDNEXT 567] Predicted next UID
S    * OK [HIGHESTMODSEQ 130]
S    * OK [UNSEEN 7] There are some unseen messages in the mailbox
S    * FLAGS (\Answered \Flagged \Draft \Deleted \Seen)
S    * OK [PERMANENTFLAGS (\Answered \Flagged \Draft \Deleted \Seen \*)]
S    * VANISHED (EARLIER) 41,43:116,118,120:211,214:540
S    * 49 FETCH (UID 117 FLAGS (\Seen \Answered) MODSEQ (128))
S    * 50 FETCH (UID 119 FLAGS (\Draft $MDNSent) MODSEQ (129))
S    ...
S    * 100 FETCH (UID 541 FLAGS (\Seen $Forwarded) MODSEQ (130))
S    A03 OK [READ-WRITE] mailbox selected
```

resync

client state

```
C   A03 SELECT INBOX (QRESYNC (67890007 123 41,43:211,214:541))
S   * OK [CLOSED]
S   * 314 EXISTS
S   * 15 RECENT
S   * OK [UIDVALIDITY 67890007] UIDVALIDITY
S   * OK [UIDNEXT 567] Predicted next UID
S   * OK [HIGHESTMODSEQ 130]
S   * OK [UNSEEN 7] There are some unseen messages in the mailbox
S   * FLAGS (\Answered \Flagged \Draft \Deleted \Seen)
S   * OK [PERMANENTFLAGS (\Answered \Flagged \Draft \Deleted \Seen \*)]
S   * VANISHED (EARLIER) 41,43:116,118,120:211,214:540
S   * 49 FETCH (UID 117 FLAGS (\Seen \Answered) MODSEQ (128))
S   * 50 FETCH (UID 119 FLAGS (\Draft $MDNSent) MODSEQ (129))
S   ...
S   * 100 FETCH (UID 541 FLAGS (\Seen $Forwarded) MODSEQ (130))
S   A03 OK [READ-WRITE] mailbox selected
```

resync

client state

server state

```
C    A03 SELECT INBOX (QRESYNC (67890007 123 41,43:211,214:541))
S    * OK [CLOSED]
S    * 314 EXISTS
S    * 15 RECENT
S    * OK [UIDVALIDITY 67890007] UIDVALIDITY
S    * OK [UIDNEXT 567] Predicted next UID
S    * OK [HIGHESTMODSEQ 130]
S    * OK [UNSEEN 7] There are some unseen messages in the mailbox
S    * FLAGS (\Answered \Flagged \Draft \Deleted \Seen)
S    * OK [PERMANENTFLAGS (\Answered \Flagged \Draft \Deleted \Seen \*)]
S    * VANISHED (EARLIER) 41,43:116,118,120:211,214:540
S    * 49 FETCH (UID 117 FLAGS (\Seen \Answered) MODSEQ (128))
S    * 50 FETCH (UID 119 FLAGS (\Draft $MDNSent) MODSEQ (129))
S    ...
S    * 100 FETCH (UID 541 FLAGS (\Seen $Forwarded) MODSEQ (130))
S    A03 OK [READ-WRITE] mailbox selected
```

resync

client state

server state

object state

```
C   A03 SELECT INBOX (QRESYNC (67890007 123 41,43:211,214:541))
S   * OK [CLOSED]
S   * 314 EXISTS
S   * 15 RECENT
S   * OK [UIDVALIDITY 67890007] UIDVALIDITY
S   * OK [UIDNEXT 567] Predicted next UID
S   * OK [HIGHESTMODSEQ 130]
S   * OK [UNSEEN 7] There are some unseen messages in the mailbox
S   * FLAGS (\Answered \Flagged \Draft \Deleted \Seen)
S   * OK [PERMANENTFLAGS (\Answered \Flagged \Draft \Deleted \Seen \*)]
S   * VANISHED (EARLIER) 41,43:116,118,120:211,214:540
S   * 49 FETCH (UID 117 FLAGS (\Seen \Answered) MODSEQ (128))
S   * 50 FETCH (UID 119 FLAGS (\Draft $MDNSent) MODSEQ (129))
S   ...
S   * 100 FETCH (UID 541 FLAGS (\Seen $Forwarded) MODSEQ (130))
S   A03 OK [READ-WRITE] mailbox selected
```

resync

client state

server state

object state

```
C   Give me updates to inbox since 123.
S   Here are many updates.  Apply these
    and you will be at 130.
```

Now you understand IMAP!

Now you understand IMAP!

So, who wants to implement it?

Right!

The good stuff is good,
but the bad stuff is
a total buzzkill.

👍 Good

👎 Bad

👍 Good

can re-sync

👎 Bad

👍 Good

can re-sync

domain-specific model

👎 Bad

👍 Good

can re-sync

domain-specific model

👎 Bad

horrible data format

👍 Good

can re-sync

domain-specific model

👎 Bad

horrible data format

weird-o transport layer

👍 Good

can re-sync

domain-specific model

👎 Bad

horrible data format

weird-o transport layer

no (or awful) commodity code

👍 Good

can re-sync

domain-specific model

👎 Bad

horrible data format

weird-o transport layer

no (or awful) commodity code

key features not in core

👍 Good

can re-sync

domain-specific model

👎 Bad

horrible data format

weird-o transport layer

no (or awful) commodity code

key features not in core

too many (s

JMAP

JSON Meta Application Protocol

request

```
POST /jmap HTTP/1.1
Host: api.fastmail.com
Content-Type: application/json

{
  ...
  "Email/get",
  { "ids": [ "1", "2", "3" ] },
  ...
}
```

response

```
Content-Type: application/json

{ ...
  "Email/get",
  { ...
    "list": [
      { ...
        "id": "1",
        "subject": "Hi, Brussels! 🇧🇪💙",
      }
    ],
  }, ...
}
```

response

```
{
  "id": "Mf713f8ce8838c8fb1176ad98",
  "messageId": [ "GBSALNHAGWNZBAVAWIFONB@signes.online" ],
  "blobId": "Gf713f8ce8838c8fb1176ad98e8c7adfc383b1eb2",
  "sentAt": "2020-09-23T21:50:06-08:00",
  "bodyValues": {
    "1": {
      "value": "Hi Rik, it's your mother...\n"
    }
  },
  "bodyStructure": {
    "type": "multipart/alternative"
  },
  "header:to:asAddresses": [
    {
      "email": "rjbs@rjbs.cloud",
      "name": "Ricardo Signes"
    }
  ],
  "replyTo": [
    {
      "email": "xwcznsl@domainsmadeeasy.com",
      "name": "DOMAIN REGISTRAR"
    }
  ]
}
```

response

```
{
  "id": "Mf713f8ce8838c8fb1176ad98",
  "messageId": [ "GBSALNHAGWNZBAVAWIFONB@signes.online" ],
  "blobId": "Gf713f8ce8838c8fb1176ad98e8c7adfc383b1eb2",
  "sentAt": "2020-09-23T21:50:06-08:00",
  "bodyValues": {
    "1": {
      "value": "Hi Rik, it's your mother...\n"
    }
  },
  "bodyStructure": {
    "type": "multipart/alternative"
  },
  "header:to:asAddresses": [
    {
      "email": "rjbs@rjbs.cloud",
      "name": "Ricardo Signes"
    }
  ],
  "replyTo": [
    {
      "email": "xwcznsl@domainsmadeeasy.com",
      "name": "DOMAIN REGISTRAR"
    }
  ]
}
```

sane date format!

You could stop here.

You could stop here.

Let's keep going!

response

```
{ ...
  "Email/get",
  { ...
    "state": "616",
    "list": [
      { ...
        "id": "1",
        "subject": "Hi, Brussels! 🇧🇪💙",
      }
    ],
  }, ...
}
```

request

```
{ ...
  "Email/changes",
  { ...
    "sinceState": "616",
  }, ...
}
```

request

```
{ ...
  "Email/changes",
  { ...
    "sinceState": "616",
  }, ...
}
```

response

```
{ ...
  "Email/changes",
  { ...
    "oldState": "616",
    "newState": "717",
    "created": [ "a", "b" ],
    "updated": [ "c" ],
    "destroyed": []
  }, ...
}
```

request

```
{ ...
  "Email/query",
  { ...
    "filter": [
      { ...
        "hasKeyword": "$flagged",
        "from": "rjbs",
      }
    ],
  }, ...
}
```

response

```
{ ...
  "Email/query",
  { ...
    "ids": [ "4", "8", "15", "16" ]
  }, ...
}
```

response

```
{ ...
  "Email/query",
  { ...
    "ids": [ "4", "8", "15", "16" ]
  }, ...
}
```

request

```
{ ...
  "Email/get",
  { ...
    "ids": [ "4", "8", "15", "16" ]
  }, ...
}
```

Brace yourself,
we need to look at
more IMAP.

```
C   a001 SEARCH FLAGGED FROM "rjbs"
S   * SEARCH 2 84 882
S   a001 OK SEARCH completed
C   a002 FETCH 2,84,882 FULL
S   * (lots of stuff)
```

```
C   a001 SEARCH FLAGGED FROM "rjbs"
    [ waiting ]
S   * SEARCH 2 84 882
S   a001 OK SEARCH completed
C   a002 FETCH 2,84,882 FULL
    [ waiting ]
S   * (lots of stuff)
```

response

```
{ ...
  "Email/query",
  { ...
    "ids": [ "4", "8", "15", "16" ]
  }, ...
}
```

request

```
{ ...
  "Email/get",
  { ...
    "ids": [ "4", "8", "15", "16" ]
  }, ...
}
```

```
Request  1: Email/query (filter)
Response 1: Email/query (1,2,3)
Request  2: Email/get   (1,2,3)
Response 2: Email/get   (1,2,3)
```

```
Request  1: Email/query (filter)
            [ waiting ]
Response 1: Email/query (1,2,3)
Request  2: Email/get   (1,2,3)
            [ waiting ]
Response 2: Email/get   (1,2,3)
```

request

```
{ ...
  [ "Email/query",
    { "filter": [...], ... },
    "a",
  ],
  [ "Email/get",
    { "#ids": { "resultOf": "a",
                "name": "Email/query",
                "path": "/ids" },
    "b",
  ],
}
```

Request

```
{ ...
  [ "Email/query",
    { "filter": [...], ... },
    "a",
  ],
  [ "Email/get",
    { "#ids": { "resultOf": "a",
                "name": "Email/query",
                "path": "/ids" },
    "b",
  ],
}
```

Request

```
{ ...
  [ "Email/query",
    { "filter": [...], ... },
    "a",
  ],
  [ "Email/get",
    { "#ids": { "resultOf": "a",
                "name": "Email/query",
                "path": "/ids" },
    "b",
  ],
}
```

```
Request  1: a: Email/query (filter)
            b: Email/get   (&a)
            [ waiting ]
Response 1: a: Email/query (1,2,3)
            b: Email/get   (1,2,3)
```

just a couple more things

request

```
{
  "using": [
    "urn:ietf:params:jmap:core",
    "urn:ietf:params:jmap:mail",
  ],
  "methodCalls": [
    [
      "Mailbox/query",
      {
        "filter"   : {  "role": "inbox" },
        "accountId": "rjbs@fastmailteam.com"
      },
      "r1"
    ],
    [
      "Mailbox/get",
      {
        "accountId": "rjbs@fastmailteam.com",
        "properties": [ "unreadThreads" ],
        "#ids": { "name": "Mailbox/query", "resultOf": "r1", "path": "/ids" }
      },
      "r2"
    ]
  ]
}
```

request

```
{
  "using": [
    "urn:ietf:params:jmap:core",
    "urn:ietf:params:jmap:mail",
  ],
  "methodCalls": [
    [
      "Mailbox/query",
      {
        "filter"   : {  "role": "inbox" },
        "accountId": "rjbs@fastmailteam.com"
      },
      "r1"
    ],
    [
      "Mailbox/get",
      {
        "accountId": "rjbs@fastmailteam.com",
        "properties": [ "unreadThreads" ],
        "#ids": { "name": "Mailbox/query", "resultOf": "r1", "path": "/ids" }
      },
      "r2"
    ]
  ]
}
```

```
{
  "using": [
    "urn:ietf:params:jmap:core",
    "urn:ietf:params:jmap:mail",
  ],
  "methodCalls": [
    [
      "Mailbox/query",
      {
        "filter"   : {  "role": "inbox" },
        "accountId": "rjbs@fastmailteam.com"
      },
      "r1"
    ],
    [
      "Mailbox/get",
      {
        "accountId": "rjbs@fastmailteam.com",
        "properties": [ "unreadThreads" ],
        "#ids": { "name": "Mailbox/query", "resultOf": "r1", "path": "/ids" }
      },
      "r2"
    ]
  ]
}
```

which APIs, and what version

```
"using": [
  "urn:ietf:params:jmap:core",
  "urn:ietf:params:jmap:mail",
]
```

```
"using": [
  "urn:ietf:params:jmap:core",
  "urn:ietf:params:jmap:mail",
  "https://cyrusimap.org/ns/jmap/contacts",
  "https://cyrusimap.org/ns/jmap/calendars"
]
```

```
{
  "using": [ ... ],
  "methodCalls": [
    [ "Mailbox/changes",       ... ],
    [ "Mailbox/get",           ... ],
    [ "Email/changes",         ... ],
    [ "Email/get",             ... ],
    [ "Contact/changes",       ... ],
    [ "Contact/get",           ... ],
    [ "CalendarEvent/changes", ... ],
    [ "CalendarEvent/get",     ... ],
 ]
}
```

```
"using": [
  "urn:ietf:params:jmap:core",
  "urn:ietf:params:jmap:mail",
  "https://cyrusimap.org/ns/jmap/contacts",
  "https://cyrusimap.org/ns/jmap/calendars",
  "https://marcolini.com/jmap/chocolat"
]
```

```
"using": [
  "urn:ietf:params:jmap:core",
  "urn:ietf:params:jmap:mail",
  "https://cyrusimap.org/ns/jmap/contacts",
  "https://cyrusimap.org/ns/jmap/calendars",
  "https://marcolini.com/jmap/chocolat"
]
```

Filters

```
"using": [
  "urn:ietf:params:jmap:core",
  "urn:ietf:params:jmap:mail",
  "https://cyrusimap.org/ns/jmap/contacts",
  "https://cyrusimap.org/ns/jmap/calendars",
  "https://marcolini.com/jmap/chocolat"
]
```

Filters

Preferences

```
"using": [
  "urn:ietf:params:jmap:core",
  "urn:ietf:params:jmap:mail",
  "https://cyrusimap.org/ns/jmap/contacts",
  "https://cyrusimap.org/ns/jmap/calendars",
  "https://marcolini.com/jmap/chocolat"
]
```

Filters

Preferences

Credentials

```
"using": [
  "urn:ietf:params:jmap:core",
  "urn:ietf:params:jmap:mail",
  "https://cyrusimap.org/ns/jmap/contacts",
  "https://cyrusimap.org/ns/jmap/calendars",
  "https://marcolini.com/jmap/chocolat"
]
```

Preferences

Filters

Credentials

DNS

```
"using": [
  "urn:ietf:params:jmap:core",
  "urn:ietf:params:jmap:mail",
  "https://cyrusimap.org/ns/jmap/contacts",
  "https://cyrusimap.org/ns/jmap/calendars",
  "https://marcolini.com/jmap/chocolat"
]
```

Preferences

Filters

Credentials

DNS

Files

```
"using": [
  "urn:ietf:params:jmap:core",
  "urn:ietf:params:jmap:mail",
  "https://cyrusimap.org/ns/jmap/contacts",
  "https://cyrusimap.org/ns/jmap/calendars",
  "https://marcolini.com/jmap/chocolat"
]
```

Filters

Preferences

Credentials

Billing

DNS

Files

```
"using": [
  "urn:ietf:params:jmap:core",
  "urn:ietf:params:jmap:mail",
  "https://cyrusimap.org/ns/jmap/contacts",
  "https://cyrusimap.org/ns/jmap/calendars",
  "https://marcolini.com/jmap/chocolat"
]
```

Filters

Preferences

Credentials

Billing

DNS

Files

...and a ton more

oh and also

```
GET /jmap/events HTTP/1.1
GET /jmap/events HTTP/1.1
```

```
event: state
data: {"Email": "818"}
```

```
GET /jmap/events HTTP/1.1
```

```
event: state
data: {"Email": "818"}

event: state
data: {"Email":"820", "Contact":"412"}
```

```
GET /jmap/events HTTP/1.1
```

```
event: state
data: {"Email": "818"}

event: state
data: {"Email":"820", "Contact":"412"}

event: state
data: {"Email":"833", "News": "69012"}
```

also, RFC 8030

aka Web Push

👍 Good

can re-sync

domain-specific model

👎 Bad

horrible data format

weird-o transport layer

no (or awful) commodity code

key features not in core

too many (s

👍 Good

can re-sync

domain-specific model

ubiquitous data format

standard transport layer

avoids round trips

one protocol, many jobs

real-time sync!

👎 Bad

~~horrible data format~~

~~weird-o transport layer~~

~~no (or awful) commodity code~~

~~key features not in core~~

~~too many (s~~

👍 Good

can re-sync

domain-specific model

ubiquitous data format

standard transport layer

avoids round trips

one protocol, many jobs

real-time sync!

👎 Bad

~~horrible data format~~

~~weird-o transport layer~~

~~no (or awful) commodity code~~

~~key features not in core~~

~~too many (s~~

not much adoption yet

👍 Good

can re-sync

domain-specific model

ubiquitous data format

standard transport layer

avoids round trips

one protocol, many jobs

real-time sync!

👎 Bad

~~horrible data format~~

~~weird-o transport layer~~

~~no (or awful) commodity code~~

~~key features not in core~~

~~too many (s~~

not much adoption yet

too many {s and "s

okay, what now?

first, read some simple code, which you can't run, just to get a look at it:

https://github.com/fastmail/JMAP-Samples

…then…

TIME TO
READ SOME RFCS!

wait, it's gonna be okay!

how to enjoy the specs

how to enjoy the specs

- first, get a basic sense of how the core methods in RFC 8620 work

how to enjoy the specs

- first, get a basic sense of how the core methods in RFC 8620 work
- that is: how to /get, how to /set, how to /query

how to enjoy the specs

- first, get a basic sense of how the core methods in RFC 8620 work
- that is: how to /get, how to /set, how to /query
- then learn the specific properties and quirks of specific data types: Mailbox, Email, &c.

RFC 8620

- request basics
- five value types
- */get
- */set
- */changes
- */query
- */queryChanges
- blobs
- push subscriptions
- event source

…but no syncable data types…

RFC 8621

- Mailbox/*
- Thread/*
- Email/*
- SearchSnippet/*

- Identity/*
- EmailSubmission/*
- VacationResponse/*

some RFC highlights

this is not an elaborate prank

Email/*

the "only" complicated part

request

```
[
  "Email/get",
  {
    "ids": [ "e1", "e2", "e3" ],
    "properties": [
      "from", "to", "subject",
      "preview",
      "mailboxIds",
    ]
  },
  "a",
]
```

response

```
[
  "Email/get",
  {
    "list": [
      { "id": "e1",
        "to": [ { "name": "Rik", "em...
        "from": [ { "name": "Neil", ...
        "subject": "Enjoy Belgium!",
        "preview": "Don't forget to ...
        "mailboxIds": { "m1": true },
      }
    ]
  },
  "a",
]
```

response

```
[
  "Email/get",
  {
    "list": [
      { "id": "e1",
        "to": [ { "name": "Rik", "em...
        "from": [ { "name": "Neil", ...
        "subject": "Enjoy Belgium!",
        "preview": "Don't forget to ...
        "mailboxIds": { "m1": true },
      }
    ]
  },
  "a",
]
```

response

```
[
  "Email/get",
  {
    "list": [
      { "id": "e1",
        "to": [ { "name": "Rik", "em...
        "from": [ { "name": "Neil", ...
        "subject": "Enjoy Belgium!",
        "preview": "Don't forget to ...
        "mailboxIds": { "m1": true },
      }
    ]
  },
  "a",
]
```

response

```
[
 "Email/get",
 {
  "list": [
   { "id": "e1",
    "to": [ { "name": "Rik", "em...
    "from": [ { "name": "Neil", ...
    "subject": "Enjoy Belgium!",
    "preview": "Don't forget to ...
    "mailboxIds": { "m1": true },
   }
  ]
 },
 "a",
]
```

response

```
[
 "Email/get",
 {
  "list": [
   { "id": "e1",
    "to": [ { "name": "Rik", "em...
    "from": [ { "name": "Neil", ...
    "subject": "Enjoy Belgium!",
    "preview": "Don't forget to ...
    "mailboxIds": { "m1": true },
   }
  ]
 },
 "a",
]
```

response

```
[
  "Email/get",
  {
    "list": [
      { "id": "e1",
        ...
        "mailboxIds": {
          "m1": true,
          "m2": true
        },
      }
    ]
  },
  "a",
]
```

response

```
[
  "Email/get",
  {
    "list": [
      { "id": "e1",
        ...
        "mailboxIds": {
          "m1": true,
          "m2": true
        },
      }
    ]
  },
  "a",
]
```

Email headers

RFC 8621 § 4.1.2

request

```
"properties": [
  "subject",
  "header:subject",
]
```

response

Email headers

RFC 8621 § 4.1.2

request

```
"properties": [
  "subject",
  "header:subject",
]
```

response

```
{ "id": "eX",
  "subject": "🇧🇪🧇",
  "header:subject": " =?UTF-8?…
]
```

Email headers

RFC 8621 § 4.1.2

Email headers

RFC 8621 § 4.1.2

```
"properties": [
```

Email headers

RFC 8621 § 4.1.2

```
"properties": [
  "subject",
```

Email headers

RFC 8621 § 4.1.2

```
"properties": [
  "subject",
  "header:subject",
```

Email headers

RFC 8621 § 4.1.2

```
"properties": [
  "subject",
  "header:subject",
  "header:subject:all",
```

Email headers

RFC 8621 § 4.1.2

```
"properties": [
  "subject",
  "header:subject",
  "header:subject:all",
  "header:subject:asText"
```

Email headers

RFC 8621 § 4.1.2

```
"properties": [
  "subject",
  "header:subject",
  "header:subject:all",
  "header:subject:asText"
  "header:subject:asText:all"
```

Email headers

RFC 8621 § 4.1.2

```
"properties": [
  "subject",
  "header:subject",
  "header:subject:all",
  "header:subject:asText"
  "header:subject:asText:all"
]
```

Email body

RFC 8621 § 4.1.4

```
"Email/get", { ...,
  "properties": [
    "blobId",
    "textBody",
    "htmlBody",
    "bodyValues"
  ],
  ...
```

Email body

RFC 8621 § 4.1.4

```
"Email/get", { ...,
  "properties": [
    "blobId",
    "textBody",
    "htmlBody",
    "bodyValues"
  ],
  ...
```

Email body

RFC 8621 § 4.1.4

```
"Email/get", { ...,
  "properties": [
    "blobId",
    "textBody",
    "htmlBody",
    "bodyValues"
  ],
  ...
```

→ This way lies MIME parsing!!

Email body

RFC 8621 § 4.1.4

```
"Email/get", { ...,
  "properties": [
    "blobId",
    "textBody",
    "htmlBody",
    "bodyValues"
  ],
  "fetchTextBodyValues": true,
  "fetchHtmlBodyValues": true,
  ...
```

```
{
  "bodyValues": {
    "1": {
      "value": "It's me, hi, I'm the body,...
    }, ...
  },
  "textBody": [ { "partId": "1", ... }, ... ]
}
```

JMAP makes the server be smart so the client can be stupid.

I know, you're sold!

How can you JMAP?

Lots of ways!

Lots of ways!

Ask for more later!
Trappistes Rochefort
Trappistes Rochefort
8
BELGIAN ALE

Fastest is to use JMAP on your Fastmail account.

https://fastmail.com/for-developers/

Learn more about how in our JMAP howto.

jmap.topicbox.com

way cool group email written using JMAP

Topicbox

Thank you!