GIT, GITHUB, AND MARKDOWN FOR INTERNET DRAFTS

Mike Bishop

TODAY

Basic concepts

- Markdown
- Git
- I-D Template
- GitHub
- C-I Systems

Getting started with I-D Template and GitHub

- Local setup – each machine
- Repo setup – GitHub
- Repo setup – first time
- C-I Setup

Workflow Demo

TODAY

Basic concepts

- Markdown
- Git
- I-D Template
- GitHub
- C-I Systems

Getting started with I-D Template and GitHub

- Local setup – each machine
- Repo setup – GitHub
- Repo setup – first time
- C-I Setup

Workflow Demo

NOT TODAY

XML2RFC v3 (previous session)

Whether working groups should use GitHub for IETF work

Substantive discussion on GitHub Issue Tracker

GitHub versus other Git hosting services

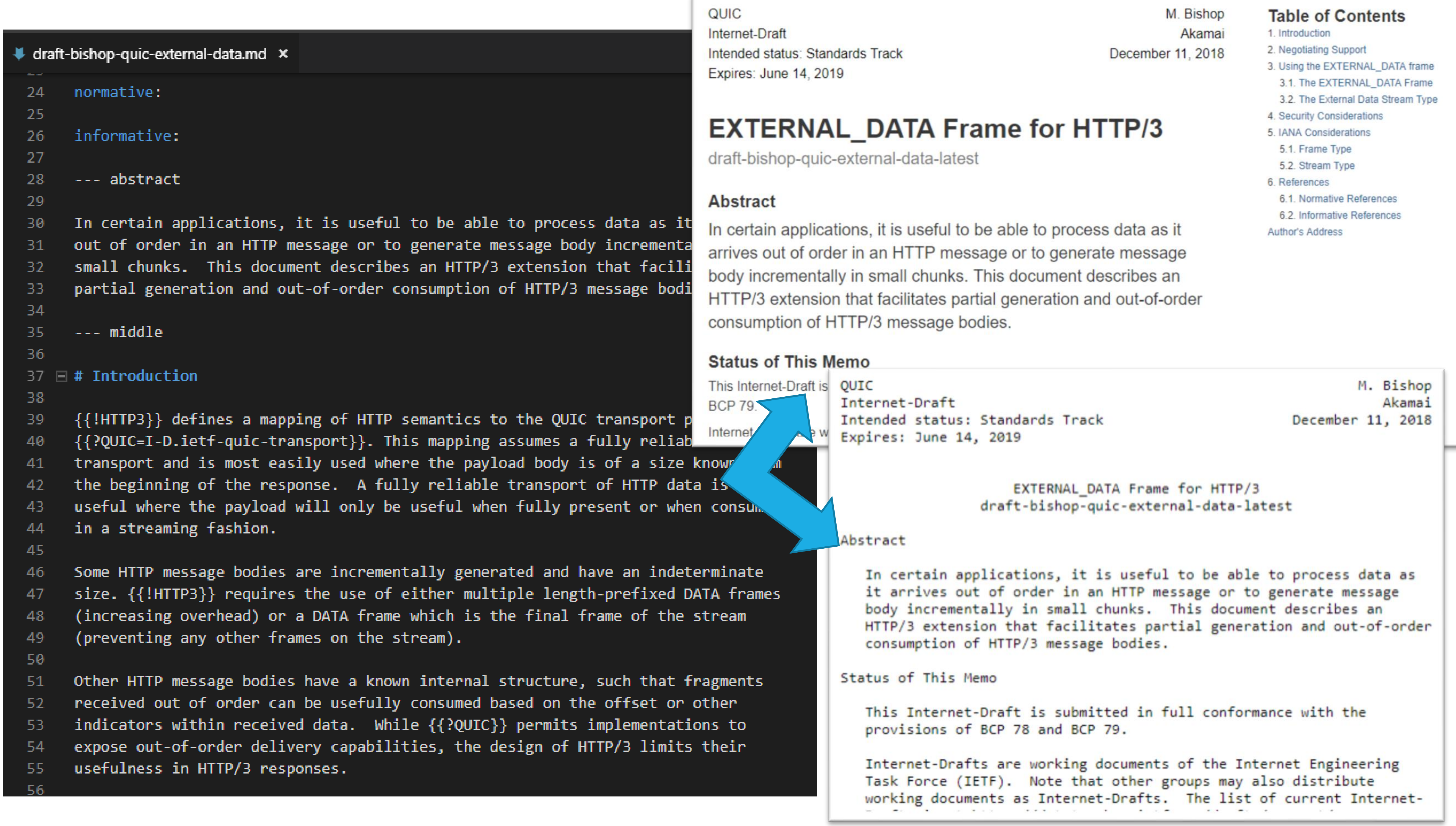
(Lack of) IPv6 support at GitHub

Correct pronunciation of “wugh”

Microsoft’s acquisition of GitHub

MARKDOWN

Certainly less cryptic than XML



draft-bishop-quic-external-data.md

```
normative:

informative:

--- abstract

In certain applications, it is useful to be able to process data as it
out of order in an HTTP message or to generate message body incrementa
small chunks.  This document describes an HTTP/3 extension that facili
partial generation and out-of-order consumption of HTTP/3 message bodi

--- middle

# Introduction

{{!HTTP3}} defines a mapping of HTTP semantics to the QUIC transport p
{{?QUIC=I-D.ietf-quic-transport}}. This mapping assumes a fully reliab
transport and is most easily used where the payload body is of a size known
the beginning of the response.  A fully reliable transport of HTTP data is
useful where the payload will only be useful when fully present or when consu
in a streaming fashion.

Some HTTP message bodies are incrementally generated and have an indeterminate
size. {{!HTTP3}} requires the use of either multiple length-prefixed DATA frames
(increasing overhead) or a DATA frame which is the final frame of the stream
(preventing any other frames on the stream).

Other HTTP message bodies have a known internal structure, such that fragments
received out of order can be usefully consumed based on the offset or other
indicators within received data.  While {{?QUIC}} permits implementations to
expose out-of-order delivery capabilities, the design of HTTP/3 limits their
usefulness in HTTP/3 responses.
```

QUIC | M. Bishop
Internet-Draft | Akamai
Intended status: Standards Track | December 11, 2018
Expires: June 14, 2019

EXTERNAL_DATA Frame for HTTP/3

draft-bishop-quic-external-data-latest

Abstract

In certain applications, it is useful to be able to process data as it arrives out of order in an HTTP message or to generate message body incrementally in small chunks. This document describes an HTTP/3 extension that facilitates partial generation and out-of-order consumption of HTTP/3 message bodies.

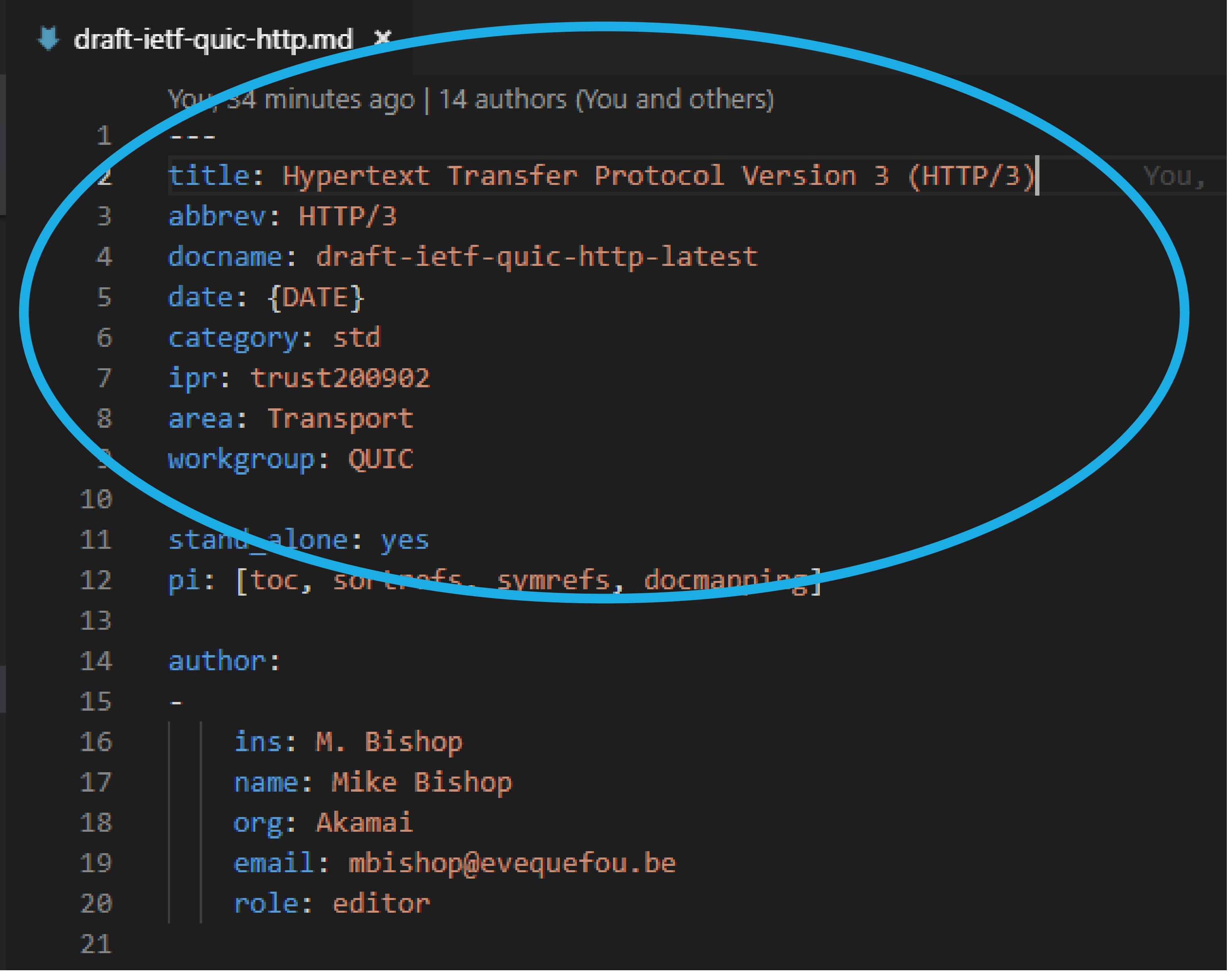
Status of This Memo

This Internet-Draft is ... BCP 79.

Internet...

Table of Contents

```
QUIC                                                          M. Bishop
Internet-Draft                                                   Akamai
Intended status: Standards Track                      December 11, 2018
Expires: June 14, 2019


                    EXTERNAL_DATA Frame for HTTP/3
                draft-bishop-quic-external-data-latest

Abstract

   In certain applications, it is useful to be able to process data as
   it arrives out of order in an HTTP message or to generate message
   body incrementally in small chunks.  This document describes an
   HTTP/3 extension that facilitates partial generation and out-of-order
   consumption of HTTP/3 message bodies.

Status of This Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF).  Note that other groups may also distribute
   working documents as Internet-Drafts.  The list of current Internet-
```


draft-ietf-quic-http.md ×

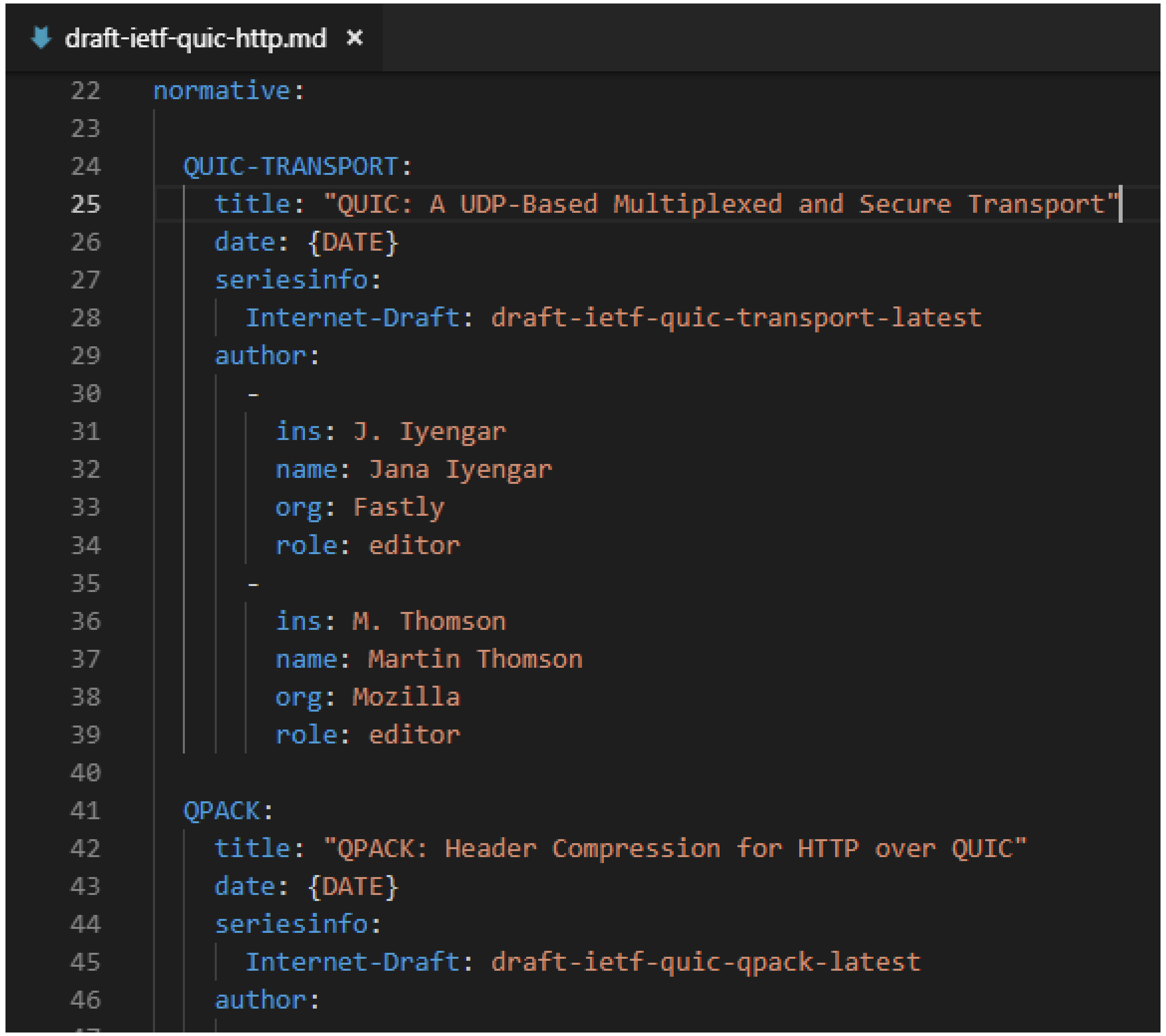
You, 34 minutes ago | 14 authors (You and others)

```
---
title: Hypertext Transfer Protocol Version 3 (HTTP/3)          You,
abbrev: HTTP/3
docname: draft-ietf-quic-http-latest
date: {DATE}
category: std
ipr: trust200902
area: Transport
workgroup: QUIC

stand_alone: yes
pi: [toc, sortrefs, symrefs, docmapping]

author:
-
    ins: M. Bishop
    name: Mike Bishop
    org: Akamai
    email: mbishop@evequefou.be
    role: editor

```

- Front matter describes the document
  - Used to generate boilerplate
- I-D Template tools require docname to end in "-latest"
  - Versions get taken care of later
- Grab another document and use it as a starting point

REFERENCES THREE WAYS

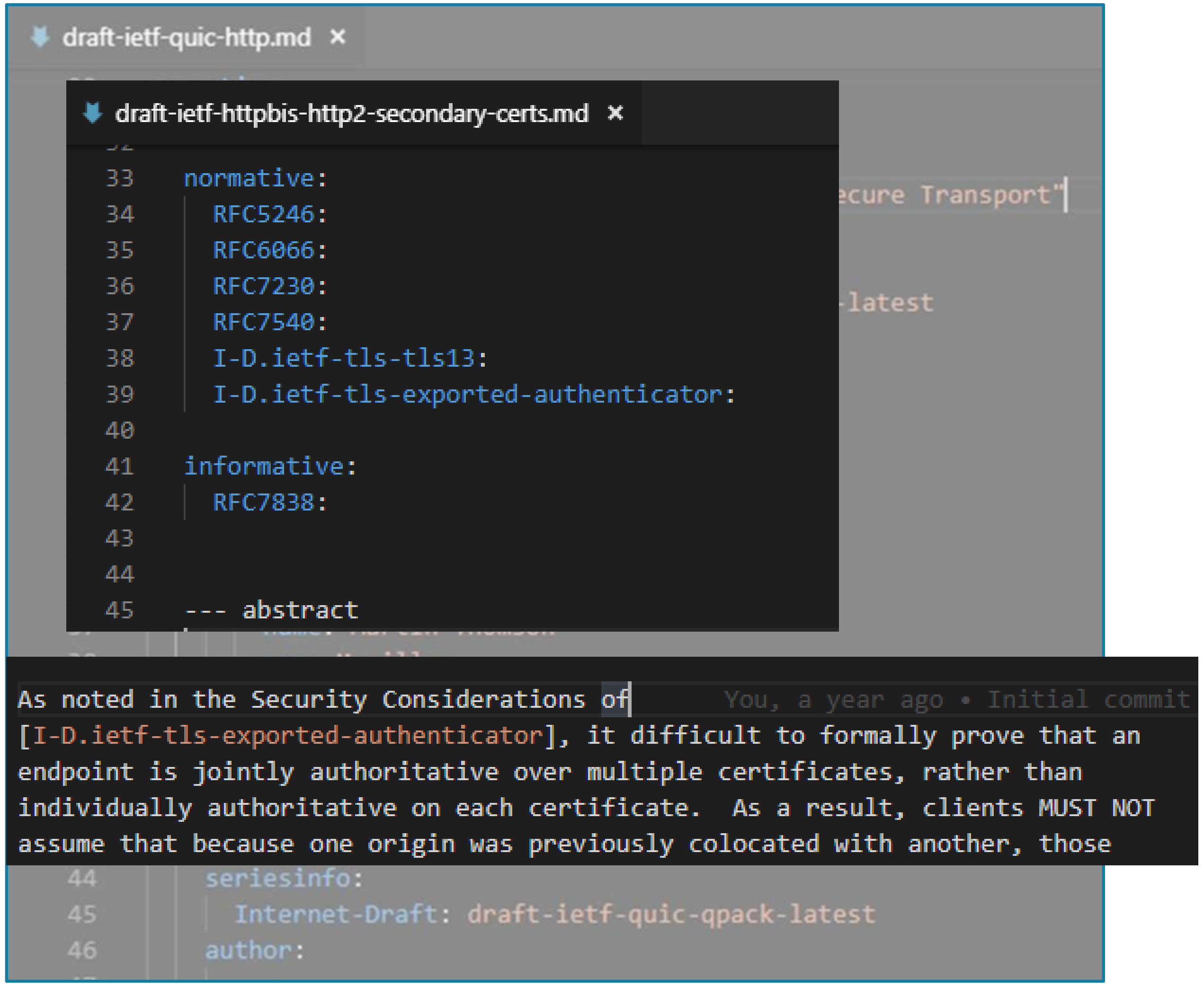
- Explicitly (format from xml2rfc)
  - Each document has `normative:` and `informative:` sections after the front-matter


draft-ietf-quic-http.md

```
normative:

  QUIC-TRANSPORT:
    title: "QUIC: A UDP-Based Multiplexed and Secure Transport"
    date: {DATE}
    seriesinfo:
      Internet-Draft: draft-ietf-quic-transport-latest
    author:
      -
        ins: J. Iyengar
        name: Jana Iyengar
        org: Fastly
        role: editor
      -
        ins: M. Thomson
        name: Martin Thomson
        org: Mozilla
        role: editor

  QPACK:
    title: "QPACK: Header Compression for HTTP over QUIC"
    date: {DATE}
    seriesinfo:
      Internet-Draft: draft-ietf-quic-qpack-latest
    author:
```

REFERENCES THREE WAYS

- Explicitly (format from xml2rfc)
  - Each document has `normative:` and `informative:` sections after the front-matter
- By standard identifier
  - Pulls from xml2rfc.ietf.org



draft-ietf-quic-http.md ×

draft-ietf-httpbis-http2-secondary-certs.md ×

```
normative:
  RFC5246:
  RFC6066:
  RFC7230:
  RFC7540:
  I-D.ietf-tls-tls13:
  I-D.ietf-tls-exported-authenticator:

informative:
  RFC7838:


--- abstract
```

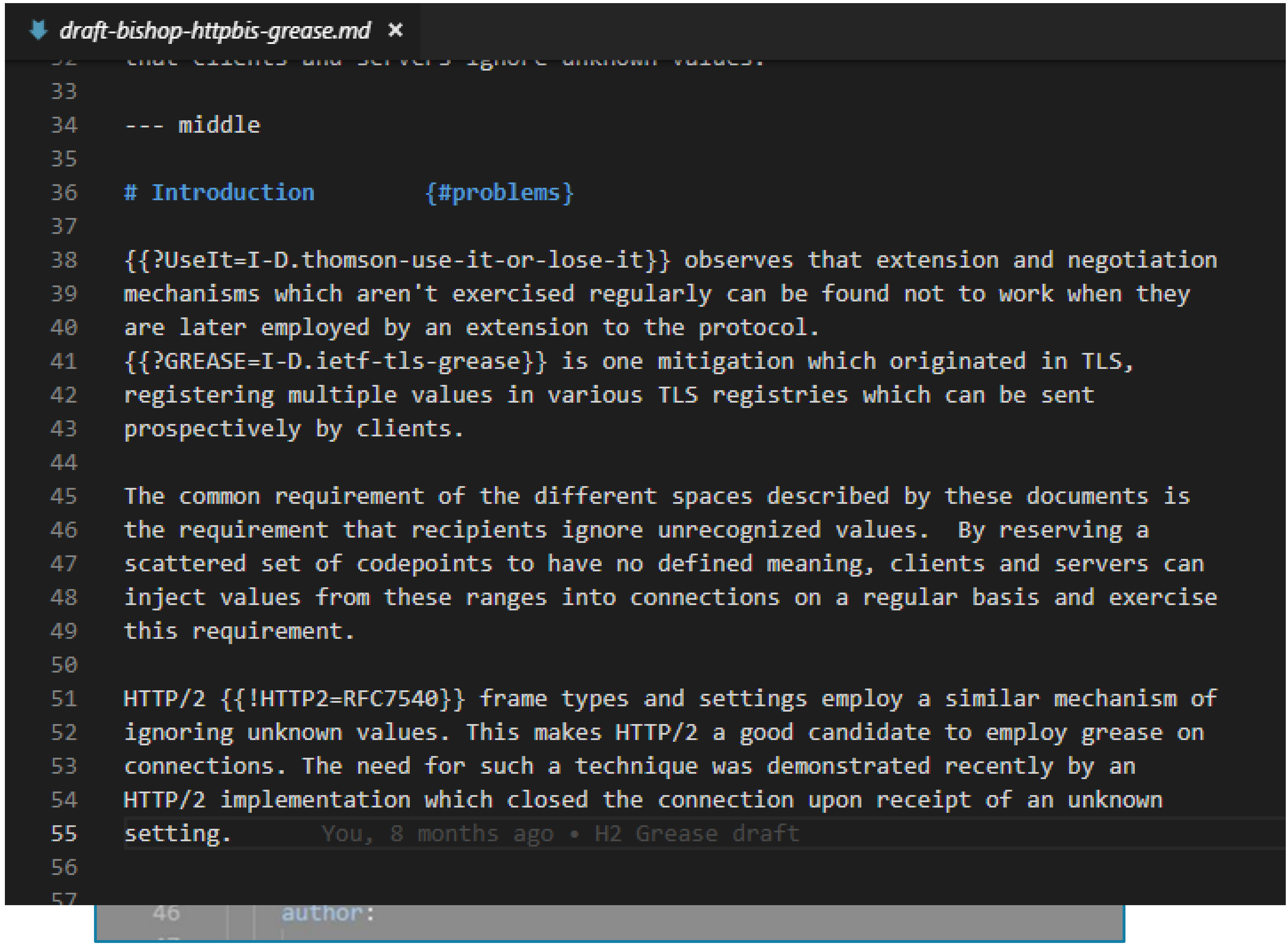
```
ecure Transport"
-latest
```

```
As noted in the Security Considerations of        You, a year ago • Initial commit
[I-D.ietf-tls-exported-authenticator], it difficult to formally prove that an
endpoint is jointly authoritative over multiple certificates, rather than
individually authoritative on each certificate.  As a result, clients MUST NOT
assume that because one origin was previously colocated with another, those
```

```
  seriesinfo:
    Internet-Draft: draft-ietf-quic-qpack-latest
  author:
```

REFERENCES THREE WAYS

- Explicitly (format from xml2rfc)
  - Each document has `normative:` and `informative:` sections after the front-matter
- By standard identifier
  - Pulls from xml2rfc.ietf.org
- Inline
  - Pulls in details by identifier
  - Permits renaming
    - `{{!displayName=reference}}` on first use
    - `{{!displayName}}` afterward
  - Normative/informative references indicated each time
    - `{{!normative}}`
    - `{{?informative}}`



```
--- middle

# Introduction        {#problems}

{{?UseIt=I-D.thomson-use-it-or-lose-it}} observes that extension and negotiation
mechanisms which aren't exercised regularly can be found not to work when they
are later employed by an extension to the protocol.
{{?GREASE=I-D.ietf-tls-grease}} is one mitigation which originated in TLS,
registering multiple values in various TLS registries which can be sent
prospectively by clients.

The common requirement of the different spaces described by these documents is
the requirement that recipients ignore unrecognized values.  By reserving a
scattered set of codepoints to have no defined meaning, clients and servers can
inject values from these ranges into connections on a regular basis and exercise
this requirement.

HTTP/2 {{!HTTP2=RFC7540}} frame types and settings employ a similar mechanism of
ignoring unknown values. This makes HTTP/2 a good candidate to employ grease on
connections. The need for such a technique was demonstrated recently by an
HTTP/2 implementation which closed the connection upon receipt of an unknown
setting.       You, 8 months ago • H2 Grease draft
```

DOCUMENT LAYOUT

```
--- abstract
```

(Text here)

```
--- middle
```

(Lots of text here)

```
--- back
```

(Appendix text here)

```
# Top-Level Heading {#first}
```

(Text here)

```
## Second-Level Heading {#second}
```

(Text here)

```
### Third-Level Heading
```

(Text here)

CROSS-REFERENCES

```
# Top-Level Heading {#first}
```

(Text here)

```
## Second-Level Heading {#second}
```

(Text here)

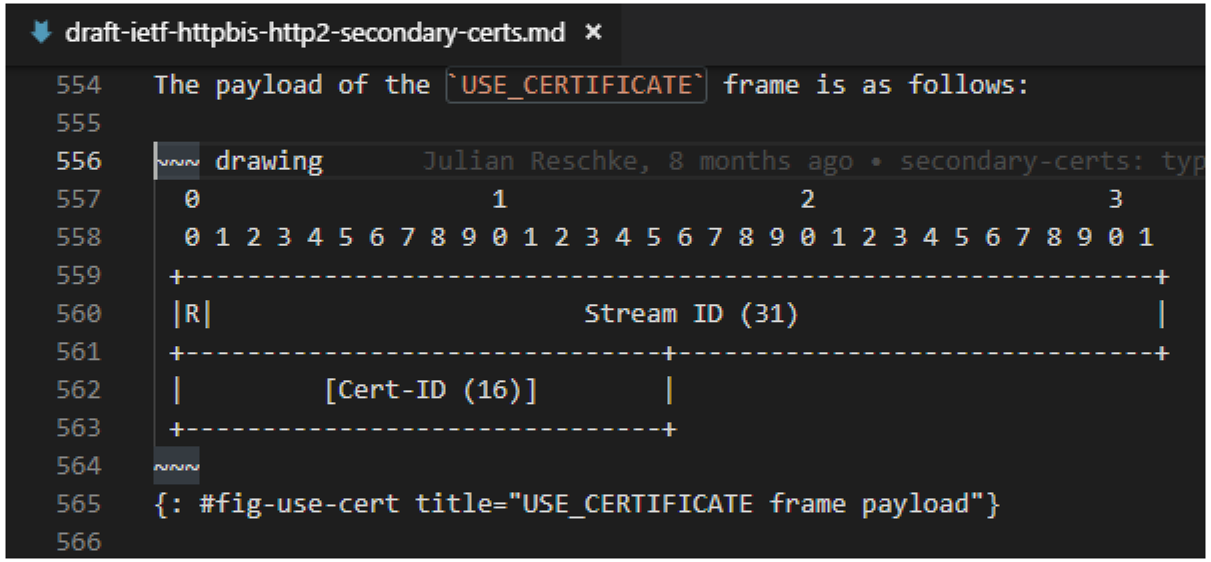
```
### Third-Level Heading
```

(Text here)

```
As discussed in {{second}}, the
thingadoodle is encoded following
the algorithm found in {{third-
level-heading}}.
```

OTHER COMMON ELEMENTS

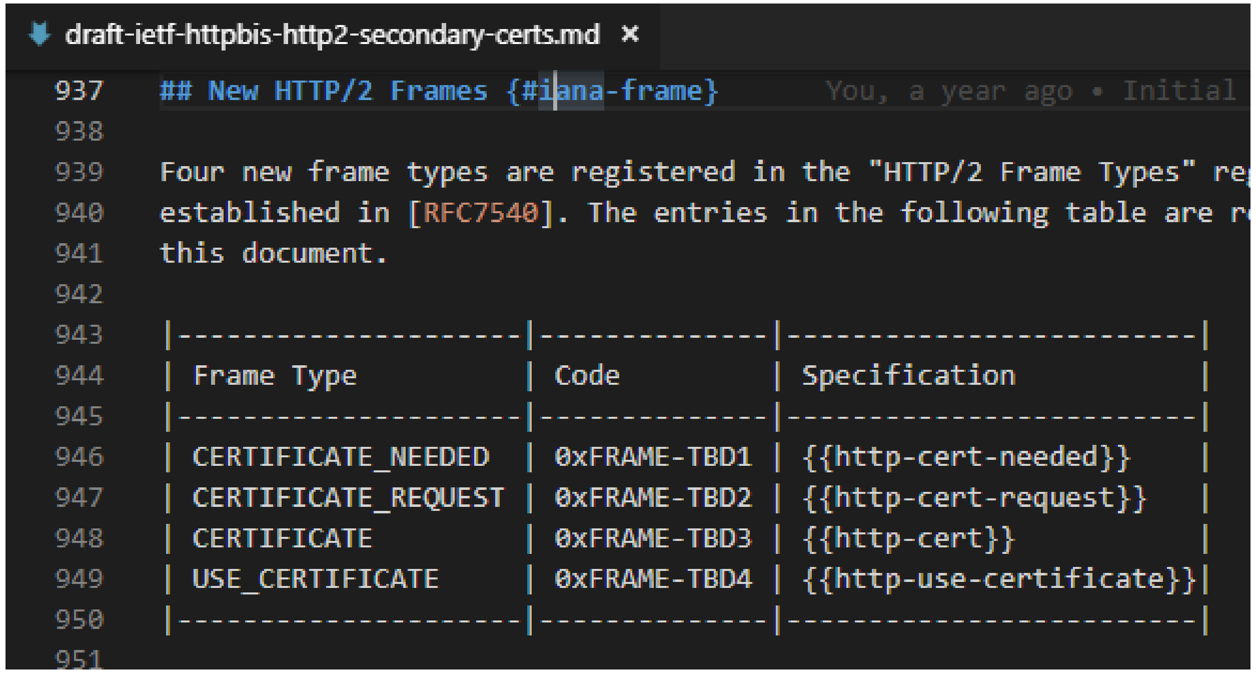
Drawings



draft-ietf-httpbis-http2-secondary-certs.md ×

```
The payload of the `USE_CERTIFICATE` frame is as follows:

~~~ drawing        Julian Reschke, 8 months ago • secondary-certs: typ
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-------------------------------------------------------------+
|R|                         Stream ID (31)                    |
+-------------------------------+-----------------------------+
|          [Cert-ID (16)]       |
+-------------------------------+
~~~
{: #fig-use-cert title="USE_CERTIFICATE frame payload"}

```

Tables



draft-ietf-httpbis-http2-secondary-certs.md ×

```
## New HTTP/2 Frames {#iana-frame}        You, a year ago • Initial

Four new frame types are registered in the "HTTP/2 Frame Types" re
established in [RFC7540]. The entries in the following table are r
this document.

|---------------------|--------------|-------------------------|
| Frame Type          | Code         | Specification           |
|---------------------|--------------|-------------------------|
| CERTIFICATE_NEEDED  | 0xFRAME-TBD1 | {{http-cert-needed}}    |
| CERTIFICATE_REQUEST | 0xFRAME-TBD2 | {{http-cert-request}}   |
| CERTIFICATE         | 0xFRAME-TBD3 | {{http-cert}}           |
| USE_CERTIFICATE     | 0xFRAME-TBD4 | {{http-use-certificate}}|
|---------------------|--------------|-------------------------|

```

GIT

Change Tracking

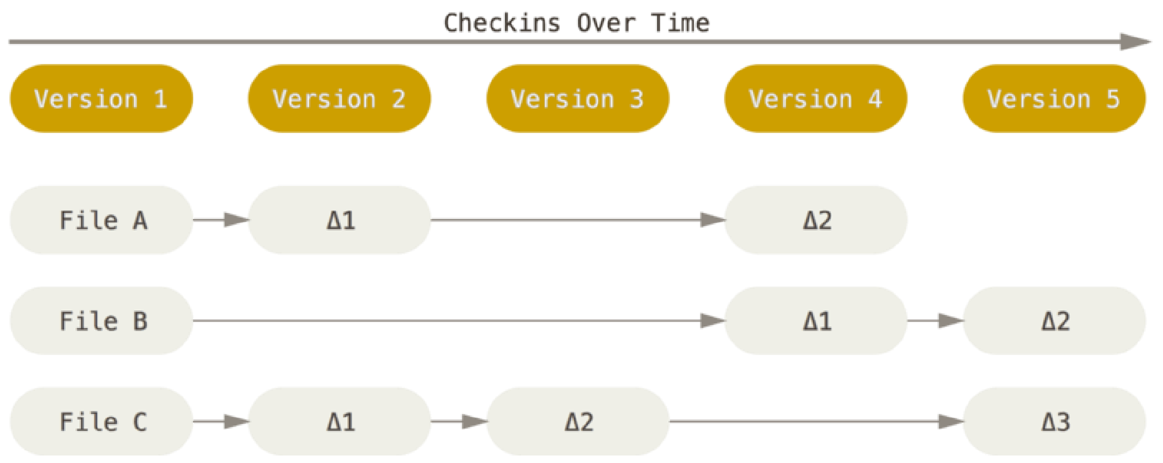
GIT BASICS

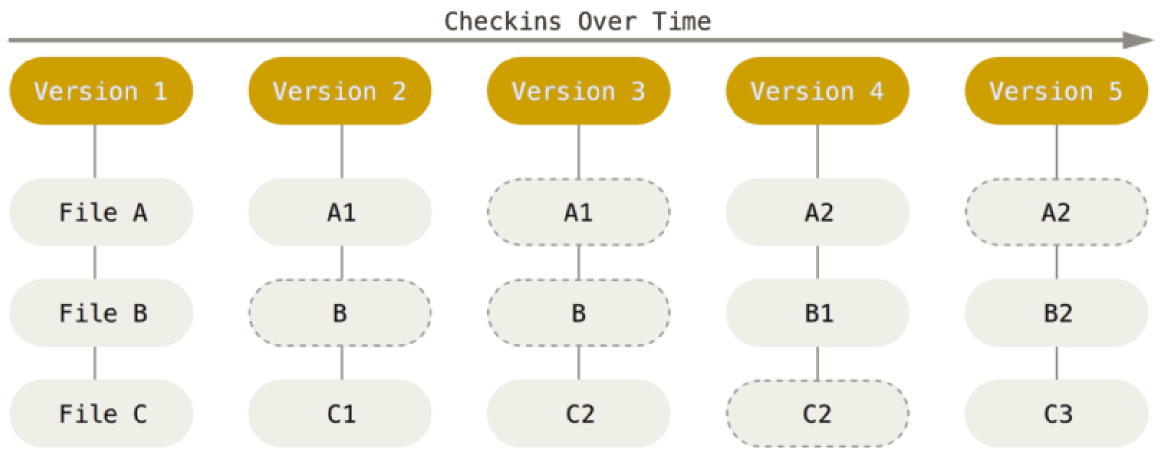
Git maintains a series of snapshots (“commits”)

Each snapshot of a file is stored by hash

Each commit is a collection of file snapshots to capture the current state of the repo

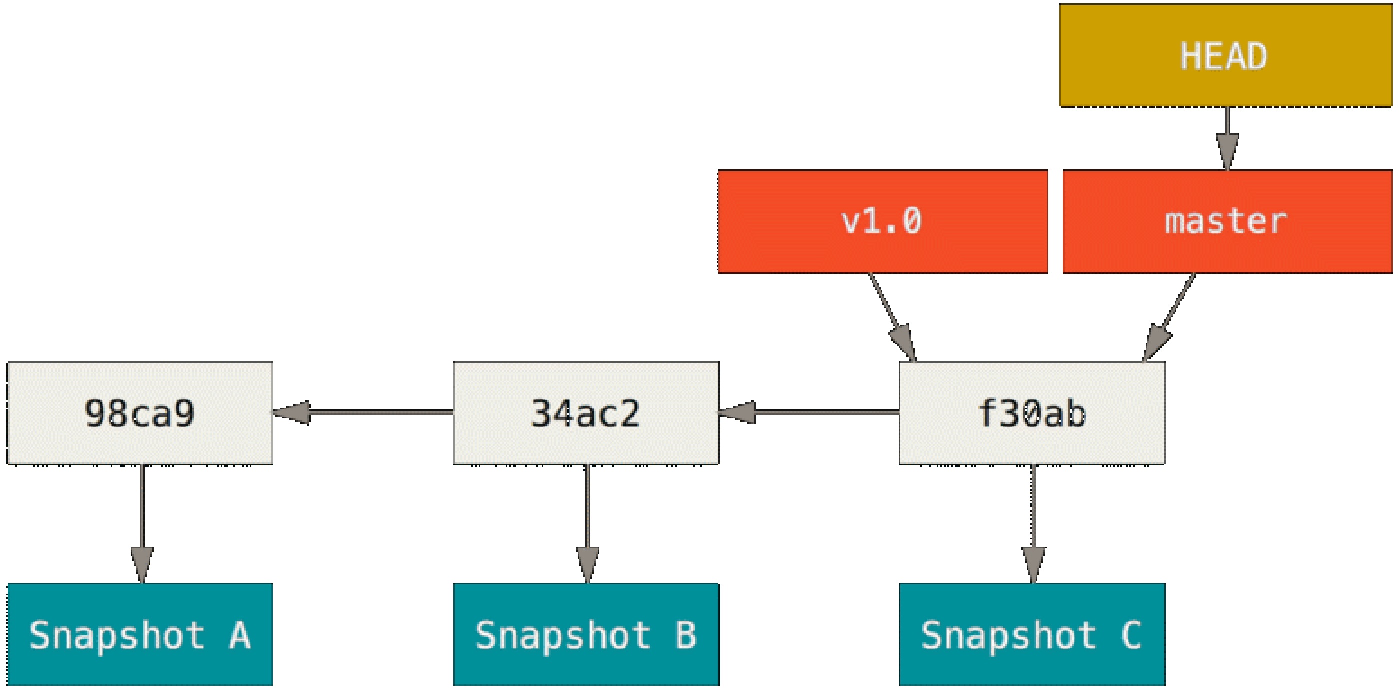
Each commit has 1+ parents to track history





Images from https://git-scm.com/, CC-BY-3.0

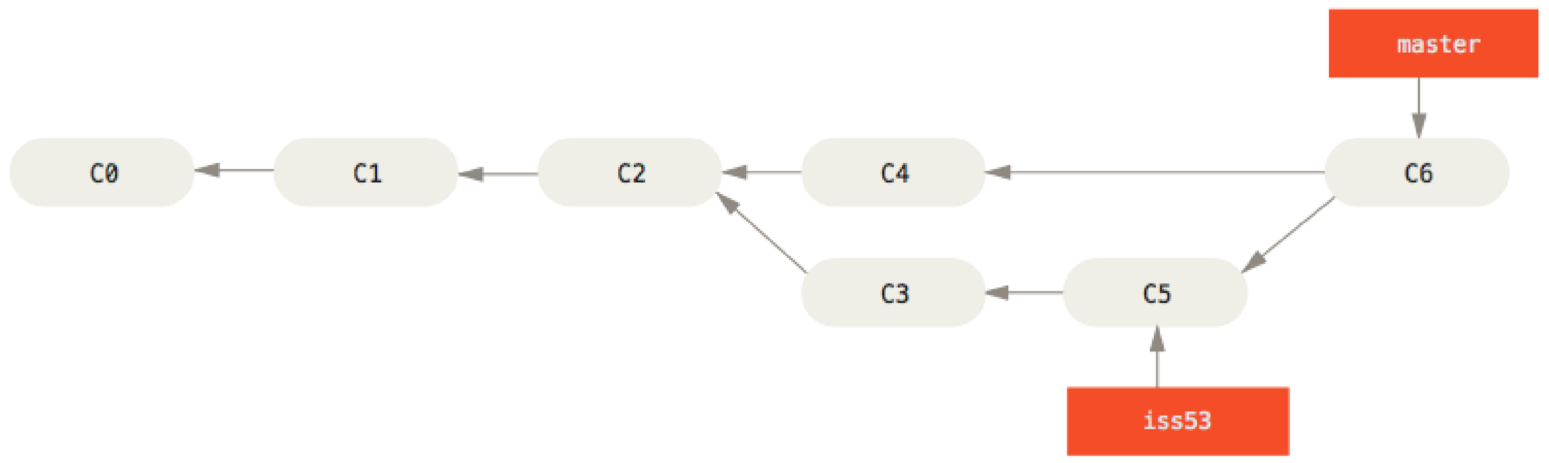
BRANCHES AND TAGS



Images from https://git-scm.com/, CC-BY-3.0

- Branches are cheap to create and disposable
- A “branch” is a pointer to a commit
  - Making new commits “on a branch” advances the pointer to the new commit
- HEAD is a pointer to the currently-selected branch
- A “tag” is also a pointer to a commit
  - …but it never advances

MERGING



Merges bring changes from one branch into another branch

Two ways to do this:

- Advance a branch down a continuous path of commits (“fast-forward”)
- Create a new commit that combines changes from two or more parent commits (“merge commit”)

REMOTES

clone:

- Suck down full copy of remote repo; remote named “origin” by default

push:

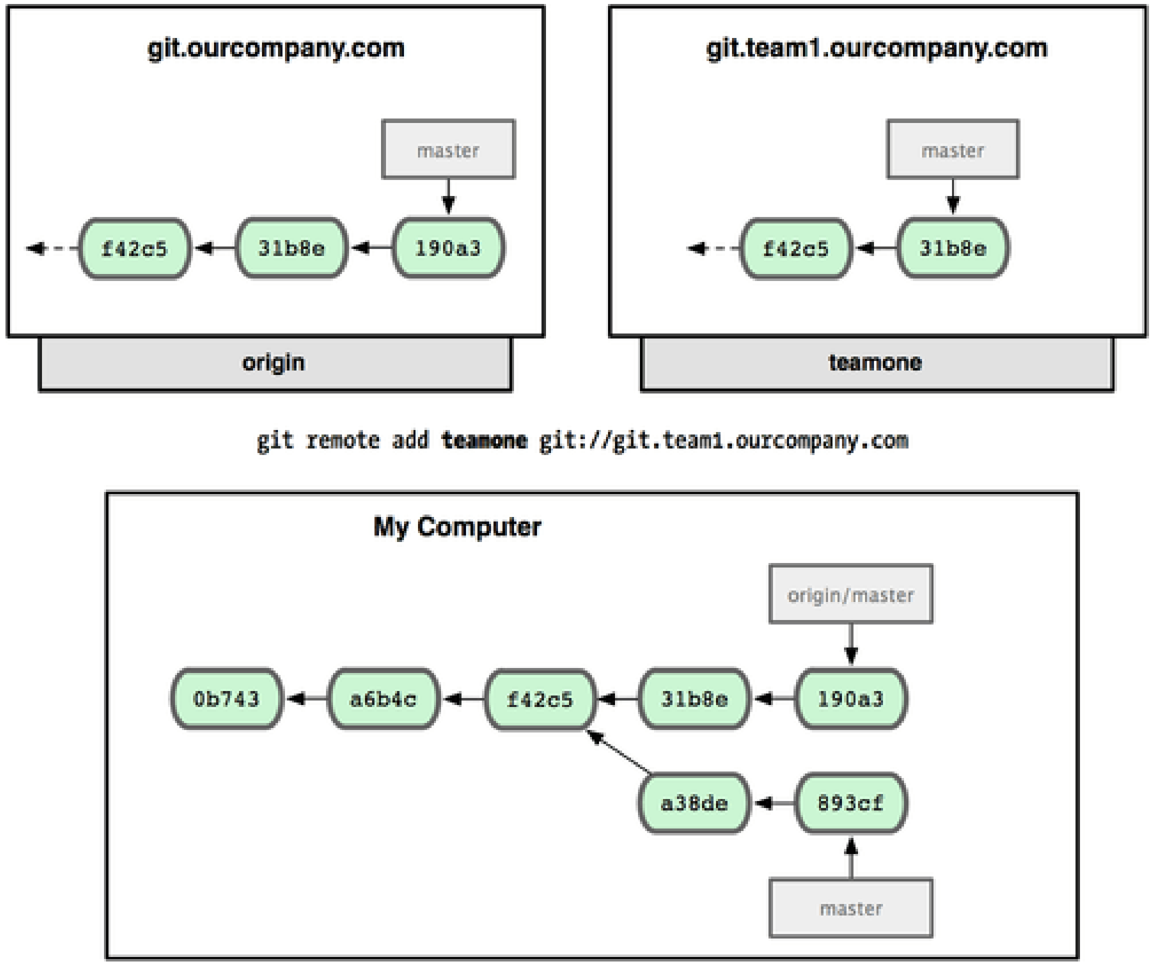
- Identify the missing ancestors of current commit
- Transfer only those commits
- Update remote branch to current commit

fetch:

- Pull any remote commits you don’t have
- Cache what commit remote branches point to

pull:

- Fetch remote repo
- Merge selected remote branch



```
git remote add teamone git://git.team1.ourcompany.com
```

OTHER USEFUL TERMS

Rebase

- Extracts the changes introduced by one or more commits
- Creates new commits that introduce the same changes from a different starting point

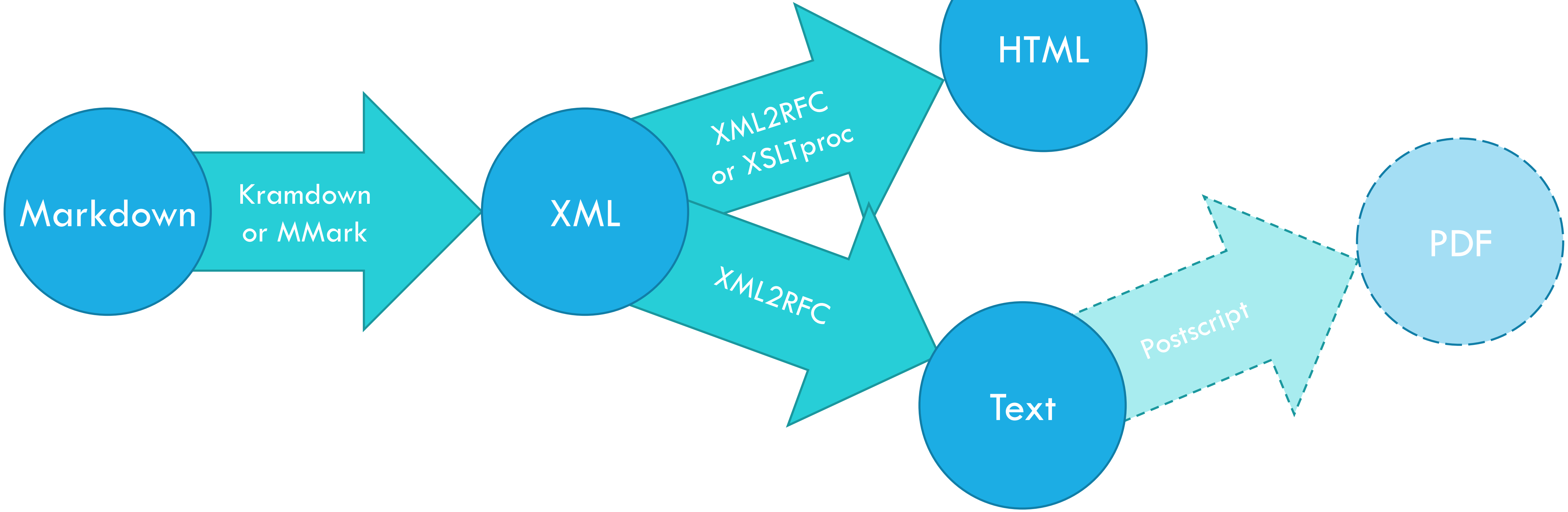
Squash

- Extracts the net set of changes introduced by a series of commits
- Creates a single new commit that introduces the same set of changes

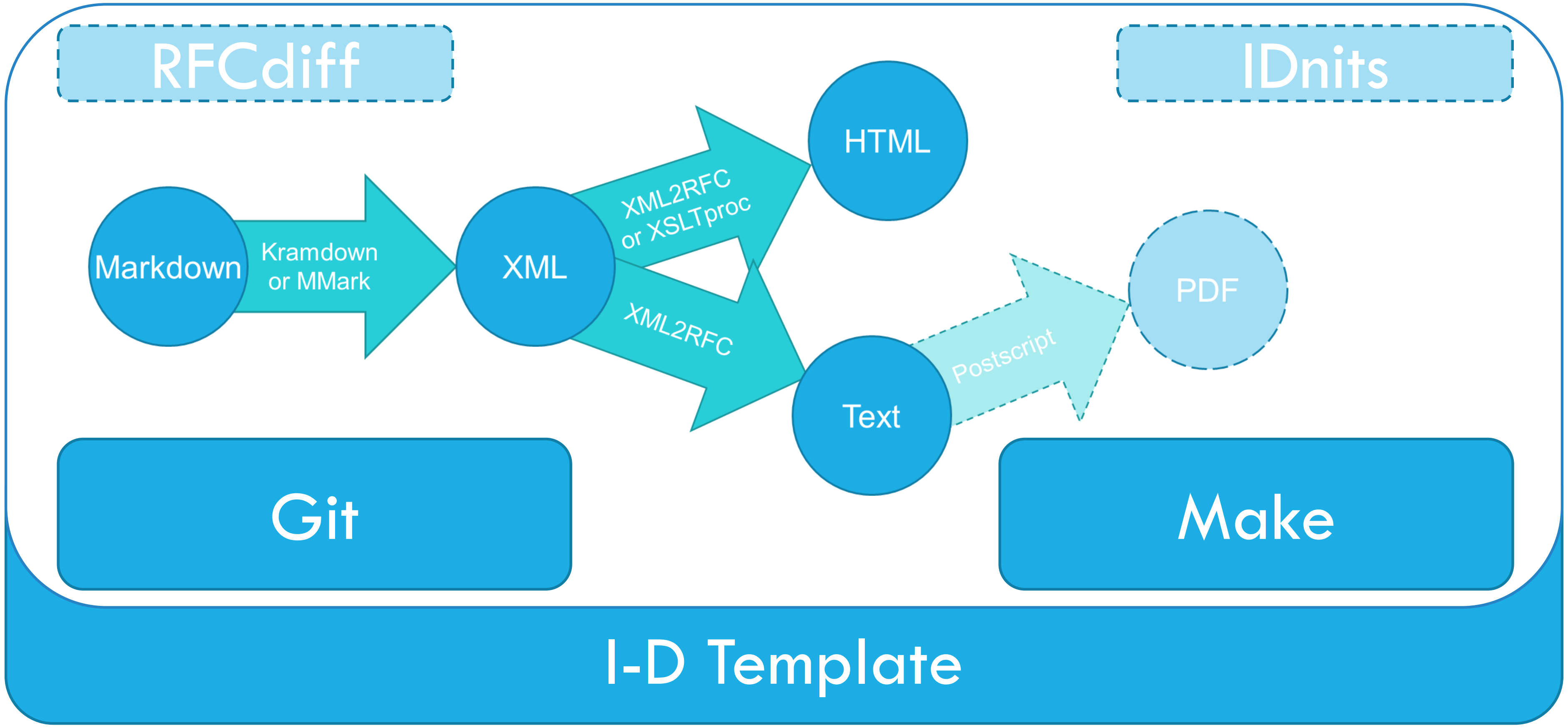
TOOLCHAIN

Turning Markdown into everything else

LOCAL TOOLING



LOCAL TOOLING



LIVE DEMO

Make your sacrifices to the Demo Gods now

GITHUB

Hosted Git and more

WHAT IS GITHUB?

Hosted git repository

Public repos are free

Private repos with 1-3 people are free

Private repos with 4+ people cost $

Enterprise-oriented on-premises costs $$$

Workflow tools

Issue tracker

Pull requests

Wiki

Automatic web pages

LIVE DEMO

Make your sacrifices to the Demo Gods now

SETUP SCRIPTS

https://github.com/richsalz/ietf-gh-scripts: Perl scripts that use the GitHub API to manage repos and documents

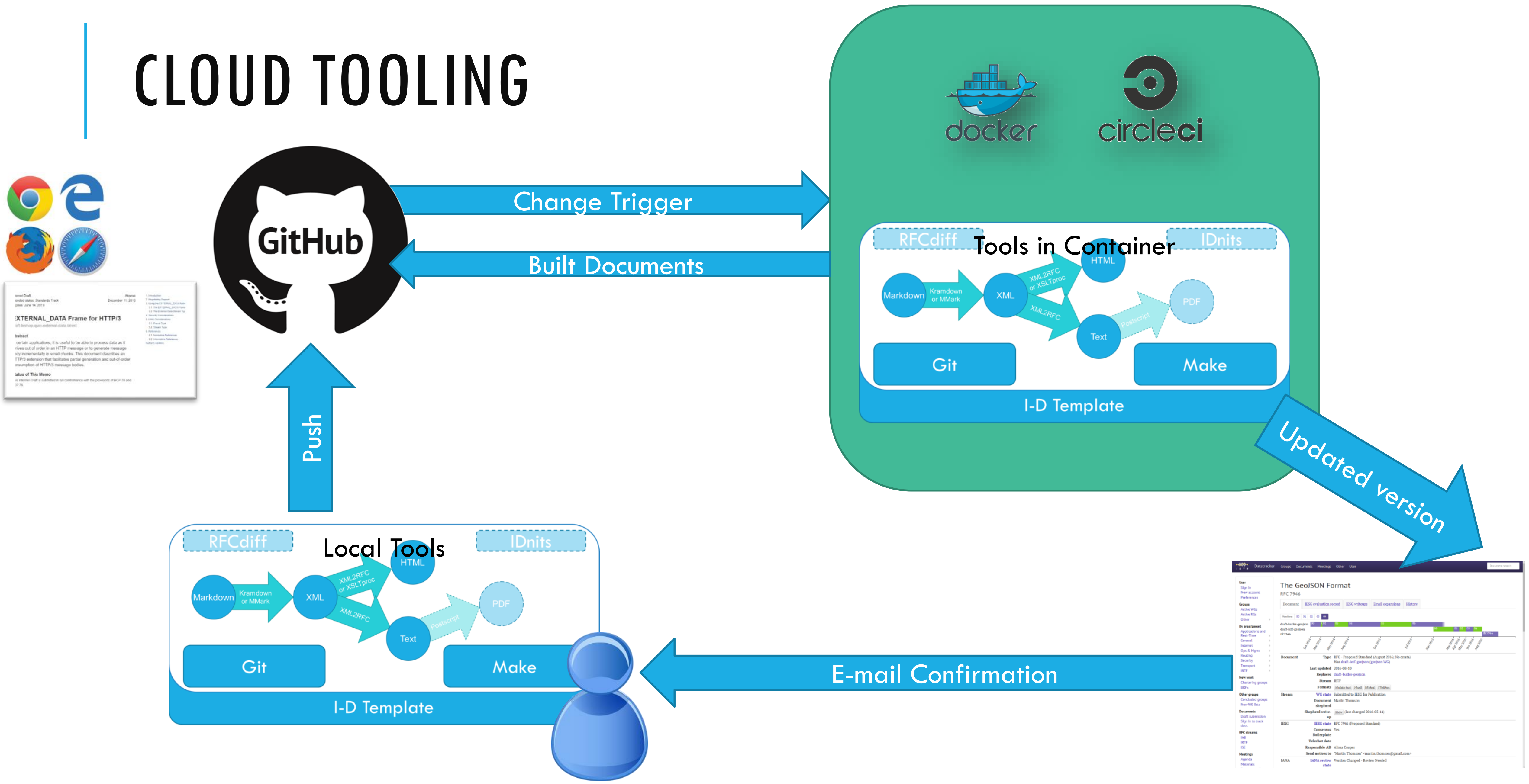
Currently supported:

- Add individual draft to new repo under individual account
- Create working group account
- Add draft to new repo under working group

Planned:

- Adopt individual draft into working group account

CLOUD TOOLING



LIVE DEMO

Make your sacrifices to the Demo Gods now

SETUP REVIEW

Local Tools

At a minimum, install:

- Git
- Make
- Xml2rfc (requires Python)
- kramdown-rfc2629 (requires Ruby)

Cloud Tools

On GitHub:

- Create repo
- Enable gh-pages
- Generate access token(s)
  - One for CircleCI (can share across repos)
  - One for git if using 2FA (per client machine)

On CircleCI:

- Follow repo
- Add access token to environment variables

IF THIS LOOKS HELPFUL….

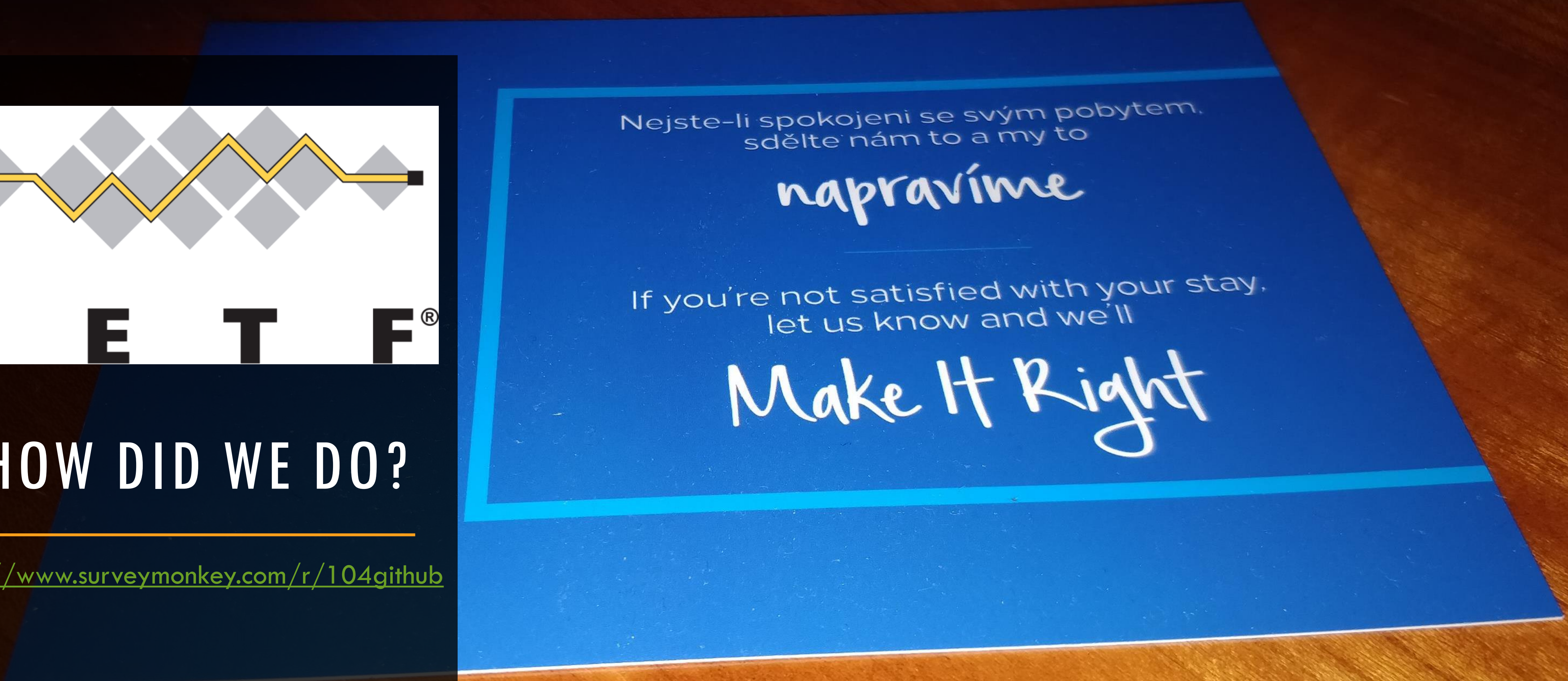
Buy this gentleman a drink!

COMMON TASK REVIEW

`make`	Attempt to build all the documents
`make update`	Updates the I-D Template
`git commit -am “Text”`	Commit all changes with commit message “Text”
`git checkout branch`	Switch to (existing) branch
`git checkout -b branch`	Switch to (new) branch
`git push -u origin branch`	Push new branch to GitHub
`git pull`	Pull changes from GitHub copy of this branch
`git tag -am “Doesn’t matter” draft-blah-blah-00`	Mark the current commit to be published as -00
`git push --tags`	Push new tags to GitHub; triggers draft submission

HOW DID WE DO?

https://www.surveymonkey.com/r/104github


Nejste-li spokojeni se svým pobytem, sdělte nám to a my to
napravíme
If you're not satisfied with your stay, let us know and we'll
Make It Right