



# **RPC workshop on using GitHub during AUTH48**

**IETF 114**

# Note Well

This is a reminder of IETF policies in effect on various topics such as patents or code of conduct. It is only meant to point you in the right direction. Exceptions may apply. The IETF's patent policy and the definition of an IETF "contribution" and "participation" are set forth in BCP 79; please read it carefully.

As a reminder:

- By participating in the IETF, you agree to follow IETF processes and policies.
- If you are aware that any IETF contribution is covered by patents or patent applications that are owned or controlled by you or your sponsor, you must disclose that fact, or not participate in the discussion.
- As a participant in or attendee to any IETF activity you acknowledge that written, audio, video, and photographic records of meetings may be made public.
- Personal information that you provide to IETF will be handled in accordance with the IETF Privacy Statement.
- As a participant or attendee, you agree to work respectfully with other participants; please contact the ombudsteam (<https://www.ietf.org/contact/ombudsteam/>) if you have questions or concerns about this.

Definitive information is in the documents listed below and other IETF BCPs. For advice, please talk to WG chairs or ADs:

- [BCP 9](#) (Internet Standards Process)
- [BCP 25](#) (Working Group processes)
- [BCP 25](#) (Anti-Harassment Procedures)
- [BCP 54](#) (Code of Conduct)
- [BCP 78](#) (Copyright)
- [BCP 79](#) (Patents, Participation)
- <https://www.ietf.org/privacy-policy/>(Privacy Policy)



# Agenda

- Note Well
- Agenda Bashing
- Overview of Editing Process
- Previous Experiments



## Agenda, continued

- Ideas from notes.ietf.org:  
<https://notes.ietf.org/s-wo8xHGRFGkMB6VFQF5fQ?both#>
  - What benefits and features are we seeking?
  - Technology
  - Source files and source control
  - Can the process be offered to all authors?
  - What should pull requests and diffs look like?



## Overview of the Editing Process

- EDIT state
  - Document is converted to v3 if need be
  - Updates to XML are made (cleanup of lists, tables, artwork, etc.)
  - Document is copy edited – all kinds of edits are made during this phase (e.g., fixing a xref and also asking about the meaning of a passage)
- RFC-EDITOR state
  - Second pass focusing on IANA Considerations, validation of code components, questions to the authors
- AUTH48 state
  - Edits are complete
  - Document and diff files are made available
  - Questions are submitted to the authors

We are focusing on AUTH48 in this workshop. The use of GitHub in EDIT and RFC-EDITOR states is not in scope.



## Previous GitHub Experiments

- RFC 8446 – 156 pages, 58 AUTH48 questions
- RFC 8829 – 115 pages, 78 AUTH48 questions
- RFC 9069 – 15 pages, 12 AUTH48 questions
- RFC 9131 – 17 pages, 17 AUTH48 questions
- RFC 9245 – 11 pages, 6 AUTH48 questions

[https://www.rfc-editor.org/rpc/wiki/doku.php?id=github\\_auth48\\_experiments](https://www.rfc-editor.org/rpc/wiki/doku.php?id=github_auth48_experiments)



## RFC 8446

- Authors created an AUTH48 repo  
<<https://github.com/tlswg/tls13-rfc>>
- Workflow:
  - Authors worked through the AUTH48 questions and made updates to the XML file in the repo.
  - Editor followed the commits and made the same updates to the RPC copy of the XML file and posted the files per the usual AUTH48 process.
  - Author final approval via email per current AUTH48 process
- GitHub features used: PRs, plain commits, watch



## RFC 8829

- Existing document repo was used and a new branch for AUTH48 created <<https://github.com/rtcweb-wg/jsep>>.
- The authors created an issue from each question.
- Workflow:
  - The authors provided feedback on each issue.
  - Concluded issues were labeled and assigned to the editor.
  - Editor created branches, edited the XML, and submitted PRs to close the issues.
  - Author final approval via email per current AUTH48 process
- GitHub features used: Issues, PRs, labels, @mentions, assignees





## RFCs 9069 and 9131

- The editor created an AUTH48 repo and invited the authors as collaborators:
  - <https://github.com/rfc-editor/rfc9069-AUTH48>
  - <https://github.com/rfc-editor/rfc9131-AUTH48>
- The editor created an issue for each author question.
- Workflow:
  - Both editor and authors created branches, edited the XML, and submitted PRs to close issues.
  - Author final approval via email per current AUTH48 process
- GitHub features used: repo template, issues, PRs, labels, @mentions



## RFC 9245

- Initially, editor created a repo, but the author requested that the draft repo be forked <<https://github.com/rfc-editor/draft-eggert-bcp45bis>>
- Suggested updates made as PRs.
- Open questions or comments were made into issues.
- Workflow:
  - Edits were made in markdown.
  - Both editor and authors created branches and submitted PRs to close issues.
  - Markdown was converted to v3 XML at the end.
  - Author final approval via email per current AUTH48 process
- GitHub features used: repo template (initially), issues, PRs

---

**Let's talk**



## What benefits and features are we seeking?

- Tracking content changes?
  - Capturing what change was made when by who and why
  - Auditable? By whom and for how long?
- Distributed authoring?
- Issue tracking?
- Pull requests?



## Technology

- git vs GitHub vs GitLab vs Google Docs
- Any other candidates?
- Would need to support long-term archiving
  - How would it work with email archives?
- Issue tracking
  - Exportable issues--taking RFC Editor questions out of the XML and exporting them to the author's issue tracker
    - ietf-comments script for converting RFC Editor questions into GitHub issues (<https://github.com/mnot/ietf-comments>)



## Source files and source control

- Source control
  - Author controlled or RPC controlled?
- Relationship between source draft and the RFC-to-be
  - Should the author's repo be cloned or forked?
  - Should the RPC start with a fresh repo?



## Can this process be offered to all authors?

- What are the criteria for going forward?
  - Stable process?
  - Identified benefits over original process?
  - Required author skill level? Should training be offered?
- What if it only a subset of authors can use and benefit from the process?
  - What percentage of authors helped would justify costs and resources?



## What should pull requests and diffs look like?

- Although the RPC makes multiple passes through a document, these passes may not (probably won't) align with short, crisp PRs.
- Given that, can we organize PRs by type?





## Wrap up



**Thank you!**