

Multiple implementations as ingredient for success

Job Snijders job@bsd.nl

OPSAREA, IETF126

July 23rd, 2026

Vienna, Austria

What is it we are doing here at IETF?

Publishing specifications, we do it for ...

- ▶ ... for fame and glory?
- ▶ ... for interoperable systems?

This talk is about both! ;-)

What *are* independent implementations?

Separate, distinct software implementations of the same standard or specification that are created and maintained by different organizations or developers.

Practically speaking

An example: two certificate signers and two verifiers (4 in total).

A single implementation need not completely cover a given spec, but the totality of implementations should cover everything in that specification.

So ... I'll just prompt two AI models?

Not really, that would miss the point.

If Bob reads a spec and first writes software using the `vi` editor and then again using `emacs` editor, it obviously is not independent implementation, right?

Don't approach this looking for legal circumvention or loopholes. Look at the underlying principles.

Why implement before publication?

- ▶ Saves time
- ▶ Saves money
- ▶ Better quality specifications
- ▶ pre-publication: publish with confidence
- ▶ post-publication: implement with confidence

... how is this possible?

How does this upfront investment help the process?

- ▶ Confirmation the spec is readable & *understandable by others than yourself*. (And if AI also is used, confirmation the spec was a good prompt!)
- ▶ **Catch bugs and design flaws that a single implementation might miss or hide.**
- ▶ Avoid having to do a -bis RFC, this saves the WG, directorate reviewers, IESG, and RFC Editor tons of time!
- ▶ *Brings people together*: pairs designers, implementers, operators, and other stakeholders.
- ▶ Natural gatekeeping: *unimplementable ideas don't progress*

Implementation required in charters: IDR

*“The WG maintains a long-standing policy that any protocol specification (excluding YANG modules) must have **at least two independent implementations** prior to advancing to publication.”*

Implementation required in charters: SSHM

*“Protocol documents should not be submitted to the IESG for publication before they have **at least two demonstrably interoperable implementations.**”*

Implementation required in charters: SIDROPS

*“Given the importance of routing security to the overall stability of the Internet, the WG will not submit protocol specifications for publication to the IESG before demonstrating **at least two interoperable implementations**. See RFC 5657 (part of BCP 9) for guidance on what implementation reports should contain and BCP 205 for guidance on how to raise awareness of running code.”*

Implementation required in charters: PLANTS

*“The Working Group will not submit specifications for publication to the IESG before demonstrating **two inter-operable implementations.**”*

What can you do?

- ▶ Evaluate existing charter, try to remember if there have been 'incidents' in the past caused by lack of running code.
- ▶ When forming working groups, consider adding a requirement for implementations to the charter.
- ▶ When reviewing specifications, ask for an Implementation Status section (see [RFC7942](#))

But what about ...

- ▶ *Our company doesn't implement before RFC publication.*
OK! Hopefully some others do! But are you sure you can't do a proof-of-concept?
- ▶ *Our company doesn't do open source*
No problem! A requirement for implementation has nothing to do with the distribution license.
- ▶ *My idea is BCP or INFO or STD, does any of this apply to me?*
Focus on the principle: is there an implementable aspect to your internet-draft? Is interoperability a goal?
- ▶ *I can't progress without IANA codepoints*
Understood, this is why Early IANA Allocation ([RFC7120](#)) and First-Come-First-Serve registry policies are super useful!

Ask me anything

Open discussion