

Network Operators and Open Source Software Study

Pilot Project Report

Leslie Daigle

December 2020

This document is one subsection of the Network Operators and Open Source Software project report. Full report available from <https://possie.techark.org/>

Pilot Project Report

Pilot Project — executive summary

Would a platform for collaborative Open Source Software projects be useful and engaging for network operators? The POSSIE project included a pilot project to explore the issue.

The project focused on one target software tool to develop, collaboratively, in open source software, and provided an organizational framework in which to progress the work. The primary question was: did this framework (an example of what a standing OSS platform could offer) provide the context necessary to engage operators? And if not, why not? Results were mixed — the framework was necessary to achieve progress, but more is needed.

In more detail: The RS-OSS Pilot Project

The Routing Security Open Source Software Pilot Project worked with a group of network operators already engaged in discussions with each other on the MANRS (Mutually Agreed Norms for Routing Security¹). In the MANRS discussions, a proposal had surfaced for creating a tool to check router configurations against the requirements of MANRS.

The RS-OSS project focused on the suggestion that had been surfaced, provided the organization and support to schedule meetings to discuss requirements, progress and next steps, and share information from the project (Slack channel, organizing and providing meeting notes, ensure a GitHub repository, etc).

There was clear interest in the project, as evinced by the participation in the scheduled meetings, as well as the original proponent's success in finding employer-supported time to code the initial prototype implementation of a solution to the identified problem.²

This prototype was greeted with enthusiasm by other participants, as evidence that progress could be made in the problem space. However, the pilot project did not

¹ <https://www.manrs.org/>

² <https://github.com/manrs-tools/MANRS-validator>

succeed in moving beyond this phase — the prototyper had chosen a particular OSS network testing tool³ on which to base the implementation, and that tool might or might not be the appropriate one to expand the code to address a larger set of practical applications of the tool. The original developer didn't have time to delve further into the project, and other project participants didn't have the wherewithal to pick up where he left off. Other, external, developers expressed interest in the project, but wanted to use it as an application of their own preferred tool⁴ (necessitating a complete rewrite of the original, and targeting a slightly different set of requirements).

In short — the pilot project was a clear illustration of the principle that, absent a consistent base of coding resources available to incrementally progress the implementation along collaboratively agreed directions, no amount of discussion would move the project forward.

Key learnings from the RS-OSS pilot project

The pilot project component of the overall POSSIE project, which focused on taking an existing set of collaborating network operators with an identified tool need, revealed one important lesson: for success of collaborative open source software projects, an active organizer is necessary, but not sufficient. A successful project must also have identified coding resources (either committed from participants, or provided by the platform) in order to make progress.

The results of the concurrent open source software survey also brought to light a second point to consider: the selected pilot project (a software tool) was not operationally-critical to participating network operators, which inevitably reduced the availability of participants and resources.

That's not to say that the pilot project was without effect. It started from some observations from network operators, on a shared mailing list, that were literally going nowhere until the pilot project arrived to provide some organizational focus to the effort. Some code was written, and a prototype created.

³ Robot Framework — <https://robotframework.org/>

⁴ Batfish — <https://www.batfish.org/>

RS-OSS Project notes and detailed timeline

March, 2019

One MANRS participant commented on a MANRS mailing list that he wasn't finding good OSS tools for his needs — abandonware, mostly.

A different MANRS participant suggested to the MANRS list that a router configuration validator tool could be built, OSS — there was zero follow up on the list.

April, 2019

Discussions began, out of band, to see about following up the validator tool idea.

The validator tool proponent pulled together a project plan (statement of work, possible validator tests to do)

- Vision: <https://docs.google.com/document/d/1dT7iNUHvKg1l7IT9KQz5bnIDyuWM6dWxIII-Jmc14C8/edit?usp=sharing>
- Validation tests: https://docs.google.com/spreadsheets/d/1W64T-tyVBbzSXjMwy4iOWPGjxGHmOkVyLUuHBL_Ct_A/edit?usp=sharing

Discussions (e-mail) with potential proponents of OSS, from the MANRS list. Most were supportive of the idea, but expect to have zero cycles (of their own, or resources they manage).

June, 2019

Some discussions in the corridors at NANOG. Other base tools were proposed and discussed.

July, 2019

The main proponent did some work on the tool, as part of his dayjob external development.

The project organizer put out a call for participation in a meeting, to the MANRS list. The aim is to get operator help in developing the specs and scoping for the project, or even possibly the existing OSS tools to do the work, before we start talking about serious coding resources.

Late July 2019

Had a call with the main proponent and folks attracted from the MANRS mailing list. 15 - 20 participants. Key takeaways were:

- This has got momentum because a key contributor has forged ahead in a very organized way, with problem statement, specific tests, and methodology
- The organizer's value-add was to collect people together — the main proponent's mail to the MANRS mailing list was inadequate for attracting attention.
- The focus people had was the mission of MANRS — make more networks secure. I.e., not for their own interests, but for getting others to step up and figure themselves out.

Following the call, and peoples' expressed interests, a GitHub repo space was set up in the MANRS project, for the main proponent's code. A Slack workspace was established.

Early August 2019

The main proponent worked on getting corporate legal release for putting code in the GitHub repository.

September 26, 2019

The source code was posted on the MANRS GitHub repo.

A third party shared a Mikrotik config dump so that the main proponents could continue the development.

The project organizer felt that relying uniquely on the original proponent was not a great way to get more people involved in the development. Floated "reveal" of the source code call.

November 2019

Had a call. Made a plan.

December 2019

Had a call. Made a plan, with a new tool (Batfish)

January 2020

Had smaller call. Batfish proponents all over this. Dump the Robot Framework stuff. But, no cycles. Anywhere.

Final conclusions

Ad hoc coding resources tend to be driven by their own motivations and timelines. Which can be okay, but doesn't lead to a coherent outcome.

To be successful, a collaborative open source project (or platform) must:

- Provide organizational resources
- Ensure there are dedicated coding resources for the project
 - Participants' availability and focus will wax and wane through any project. To stay focused, there has to be a stable development environment
- Focus on projects that are identified as important by the network operators themselves
 - Operationally important tools
 - Open standard reference implementations

Full report available at <https://possie.techark.org>

Leslie Daigle

December 2020

Full report available from <https://possie.techark.org/>