

Questions for Selecting Software Partners

Simply Secure
collab@simplysecure.org

A note on this document

This is not a quiz to judge the worthiness of projects or the people who work on them. Many questions have no right answer, and are included to help Simply Secure foster a diverse portfolio of projects aligned with its core goals.

1 Initial questions

These first-order questions will help us select which projects to look into more deeply.

1.1 Open-source licensing

Simply Secure is committed to improving the design and usability of open-sourced tools; what percentage of the deployed or already-developed source code has been released openly?

1.2 Secure communications

Simply Secure is currently focused on secure communications (including functionality such as chatting, emailing, messaging, file sharing, or other data transfer from one person to another). What communication functionality does the software support? What makes it secure (e.g., does it support end-to-end encryption)?

1.3 Impact

Simply Secure is interested in improving the design and usability of tools that have a high probability of having a significant real-world impact. With that in mind, how many active users does the software have? If that statistic is not available, how many downloads has the software had in the past month and in the past year? Or, how does the team measure real-world impact of the software?

2 Deeper Questions

Some of these questions can be answered from public information; others may need to be answered in collaboration with the software team.

2.1 Software features

Is it end-user facing, or designed to run behind the scenes? If it is designed to run behind the scenes, how does it support end-user-facing software)?

Of its available communication and data-sharing functions, which among them support end-to-end encryption?

For each communication and data-sharing function that supports end-to-end encryption, is it supported exclusively, by default, or on an opt-in basis?

2.2 Software users and support

What type(s) of users is the tool targeted at (e.g., tech-savvy activists, investigative journalists, people living under censorship, everyone)?

What natural language(s) does the interface support (e.g., English, Chinese, French, Persian, Spanish)?

If known, with what type(s) of users or in what region(s) of the world is the tool popular?

Is there documentation for using the software? If so, please provide a link.

Are there forums, mailing lists, or other sources users can turn to in learning how to use the software? If so, please provide a link.

2.3 Transparency and open-source licensing

Under what license(s) is the software open-sourced?

Where and how is the code published (e.g., on GitHub, Sourceforge, Google Code)? Please provide a link.

In what programming language(s) is the software written?

Does the project accept code contributions from new, external volunteers?

What percentage of code is accepted into the project in this way?

Is there documentation for building and contributing source code to the software? If so, please provide a link.

Does the development team publish meeting notes, design documents, or otherwise open its decision-making process to people outside the team? If so, how?

2.4 Development practices

How frequently are new releases pushed? How are releases planned and scheduled?

What is the teams development roadmap for the near and medium term?

What tools, methods, and activities does the team engage in to assure the quality of the software?

Has the software undergone an external security audit or review? If so, when? Please link to the resultant report, if possible.

Have there been reports of vulnerabilities, breaches, or backdoors found in the software? If so, how did the team respond?

2.5 Team

How many software engineers have contributed to the codebase in the past month, the past year, and since the beginning of the project?

How many contributors are core members of the team?

Are core members of the team paid for their work on the project, or do they volunteer their time?

If members are paid, what is the teams source of income (e.g., grants, revenue from selling the software or a pro version of it, consulting side-work)?

Are there non-engineers affiliated with the project, or engineers who dedicate some of their time to non-coding work? If so, what tasks do they focus on (e.g., writing grants, user support, marketing, project management, visual or interaction design)?

3 Software team engagement

These questions should be answered in collaboration with the software team.

3.1 Design and usability work

Simply Secure seeks to partner with projects that are eager and equipped to accept feedback on the user experience of their tool, but that are also able to productively weigh (and potentially push back against) design tradeoffs that may compromise the softwares security guarantees.

Does the software team have a sense of the major user pain points with the software as it stands today? If so, what are the top issues, and how were they determined?

What work has gone into improving the user experience of the software recently? What has been the impact?

What (if any) work is the team currently doing to improve the user experience of the software?

Has the team worked with visual or interaction designers and/or user-experience researchers to improve the tool? If so, who, and what resulted from the collaboration?

Have members of the team worked with visual or interaction designers and/or user-experience researchers on other projects? If so, please tell us a bit about how that collaboration went.

3.2 Engagement logistics

Simply Secure hopes to find partners who are excited to engage with us, who have the resources to make changes, and who are willing to commit to a transparent collaboration.

How broad is the interest within the team to collaborate with Simply Secure (1-2 people, the whole core team, the whole team plus a noisy majority of the volunteers, etc.)?

Which member(s) of the team would be Simply Secures primary point(s) of contact? Would they be responsible for implementing changes identified by the evaluation and design process?

How much time can the engineers who are working to implement the changes identified by the evaluation and design process dedicate to this work?

If Simply Secure were to perform a design and usability evaluation of the software, would the team be willing to commit to having the results made public? Please list any reservations, concerns, or constraints.

If Simply Secure were to perform a security evaluation of the software, would the team be willing to commit to having the results made public? Please list any reservations, concerns, or constraints.

If Simply Secure were to engage with the team in designing new software interactions or on other development activities, would the team be willing to commit to having the results made public? Please list any reservations, concerns, or constraints.

Does the team have any other reservations, concerns, or constraints around transparency?