Standard for Public Code version 0.7.1 checklist

☐ Code in the open

- ☐All source code for any policy in use (unless used for fraud detection) MUST be published and publicly accessible.
- ☐All source code for any software in use (unless used for fraud detection) MUST be published and publicly accessible.
- ☐The codebase MUST NOT contain sensitive information regarding users, their organization or third parties.
- ☐Any source code not currently in use (such as new versions, proposals or older versions) SHOULD be published.
- ☐Documenting which source code or policy underpins any specific interaction the general public may have with an organization is OPTIONAL.

☐ Bundle policy and source code

- ☐The codebase MUST include the policy that the source code is based on.
- ☐If a policy is based on source code, that source code MUST be included in the codebase, unless used for fraud detection.
- ☐Policy SHOULD be provided in machine readable and unambiguous formats.
- ☐Continuous integration tests SHOULD validate that the source code and the policy are executed coherently.

☐ Make the codebase reusable and portable

- ☐The codebase MUST be developed to be reusable in different contexts.
- ☐The codebase MUST be independent from any secret, undisclosed, proprietary or non-open licensed software or services for execution and understanding.
- ☐The codebase SHOULD be in use by multiple parties.
- ☐The roadmap SHOULD be influenced by the needs of multiple parties.
- ☐The development of the codebase SHOULD be a collaboration between multiple parties.
- ☐Configuration SHOULD be used to make source code adapt to context specific needs.
- ☐The codebase SHOULD be localizable.
- ☐Source code and its documentation SHOULD NOT contain situation-specific information.
- ☐Codebase modules SHOULD be documented in such a way as to enable reuse in codebases in other contexts.
- ☐The software SHOULD NOT require services or platforms available from only a single vendor.

☐ Welcome contributors

- ☐The codebase MUST allow anyone to submit suggestions for changes to the codebase.
- ☐The codebase MUST include contribution guidelines explaining what kinds of contributions are welcome and how contributors can get involved, for example in a `CONTRIBUTING` file.
- ☐The codebase MUST document the governance of the codebase, contributions and its community, for example in a `GOVERNANCE` file.
- ☐The contribution guidelines SHOULD document who is expected to cover the costs of reviewing contributions.
- ☐The codebase SHOULD advertise the committed engagement of involved organizations in the development and maintenance.
- ☐The codebase SHOULD have a publicly available roadmap.
- ☐The codebase SHOULD publish codebase activity statistics.
- ☐Including a code of conduct for contributors in the codebase is OPTIONAL.

☐ Make contributing easy

- ☐The codebase MUST have a public issue tracker that accepts suggestions from anyone.
- ☐The documentation MUST link to both the public issue tracker and submitted codebase changes, for example in a `README` file.
- ☐The codebase MUST have communication channels for users and developers, for example email lists.
- ☐There MUST be a way to report security issues for responsible disclosure over a closed channel.
- ☐The documentation MUST include instructions for how to report potentially security sensitive issues.

☐ Maintain version control

- ☐All files in the codebase MUST be version controlled.
- ☐All decisions MUST be documented in commit messages.
- ☐Every commit message MUST link to discussions and issues wherever possible.
- ☐The codebase SHOULD be maintained in a distributed version control system.
- ☐Contribution guidelines SHOULD require contributors to group relevant changes in commits.
- ☐Maintainers SHOULD mark released versions of the codebase, for example using revision tags or textual labels.
- ☐Contribution guidelines SHOULD encourage file formats where the changes within the files can be easily viewed and understood in the version control system.
- ☐It is OPTIONAL for contributors to sign their commits and provide an email address, so that future contributors are able to contact past contributors with questions about their work.

☐ Require review of contributions

- ☐All contributions that are accepted or committed to release versions of the codebase MUST be reviewed by another contributor.
- ☐Reviews MUST include source, policy, tests and documentation.
- ☐Reviewers MUST provide feedback on all decisions to not accept a contribution.
- ☐The review process SHOULD confirm that a contribution conforms to the standards, architecture and decisions set out in the codebase in order to pass review.
- ☐Reviews SHOULD include running both the software and the tests of the codebase.
- ☐Contributions SHOULD be reviewed by someone in a different context than the contributor.
- ☐Version control systems SHOULD NOT accept non-reviewed contributions in release versions.
- ☐Reviews SHOULD happen within two business days.
- ☐Performing reviews by multiple reviewers is OPTIONAL.

☐ Document codebase objectives

- ☐The codebase MUST contain documentation of its objectives, like a mission and goal statement, that is understandable by developers and designers so that they can use or contribute to the codebase.
- ☐Codebase documentation SHOULD clearly describe the connections between policy objectives and codebase objectives.
- ☐Documenting the objectives of the codebase for the general public is OPTIONAL.

☐ Document the code

- ☐All of the functionality of the codebase, policy as well as source code, MUST be described in language clearly understandable for those that understand the purpose of the codebase.

- ☐The documentation of the codebase MUST contain a description of how to install and run the software.
- ☐The documentation of the codebase MUST contain examples demonstrating the key functionality.
- ☐The documentation of the codebase SHOULD contain a high level description that is clearly understandable for a wide audience of stakeholders, like the general public and journalists.
- ☐The documentation of the codebase SHOULD contain a section describing how to install and run a standalone version of the source code, including, if necessary, a test dataset.
- ☐The documentation of the codebase SHOULD contain examples for all functionality.
- ☐The documentation SHOULD describe the key components or modules of the codebase and their relationships, for example as a high level architectural diagram.
- ☐There SHOULD be continuous integration tests for the quality of the documentation.
- ☐Including examples that make users want to immediately start using the codebase in the documentation of the codebase is OPTIONAL.

☐ Use plain English

- ☐All codebase documentation MUST be in English.
- ☐All source code MUST be in English, except where policy is machine interpreted as code.
- ☐All bundled policy not available in English MUST have an accompanying summary in English.
- ☐Any translation MUST be up to date with the English version and vice versa.
- ☐There SHOULD be no acronyms, abbreviations, puns or legal/non-English/domain specific terms in the codebase without an explanation preceding it or a link to an explanation.
- ☐Documentation SHOULD aim for a lower secondary education reading level, as recommended by the Web Content Accessibility Guidelines 2.
- ☐Providing a translation of any code, documentation or tests is OPTIONAL.

☐ Use open standards

- ☐For features of the codebase that facilitate the exchange of data the codebase MUST use an open standard that meets the Open Source Initiative Open Standard Requirements.
- ☐Any non-open standards used MUST be recorded clearly as such in the documentation.
- ☐Any standard chosen for use within the codebase MUST be listed in the documentation with a link to where it is available.
- ☐Any non-open standards chosen for use within the codebase MUST NOT hinder collaboration and reuse.
- ☐If no existing open standard is available, effort SHOULD be put into developing one.
- ☐Open standards that are machine testable SHOULD be preferred over open standards that are not.
- ☐Non-open standards that are machine testable SHOULD be preferred over non-open standards that are not.

☐ Use continuous integration

- ☐All functionality in the source code MUST have automated tests.
- ☐Contributions MUST pass all automated tests before they are admitted into the codebase.
- ☐The codebase MUST have guidelines explaining how to structure contributions.
- ☐The codebase MUST have active contributors who can review contributions.
- ☐Automated test results for contributions SHOULD be public.
- ☐The codebase guidelines SHOULD state that each contribution should focus on a single issue.
- ☐Source code test and documentation coverage SHOULD be monitored.
- ☐Testing policy and documentation for consistency with the source and vice versa is OPTIONAL.
- ☐Testing policy and documentation for style and broken links is OPTIONAL.

- ☐Testing the software by using examples in the documentation is OPTIONAL.

☐ Publish with an open license

- ☐All source code and documentation MUST be licensed such that it may be freely reusable, changeable and redistributable.
- ☐Software source code MUST be licensed under an OSI-approved or FSF Free/Libre license.
- ☐All source code MUST be published with a license file.
- ☐Contributors MUST NOT be required to transfer copyright of their contributions to the codebase.
- ☐All source code files in the codebase SHOULD include a copyright notice and a license header that are machine-readable.
- ☐Having multiple licenses for different types of source code and documentation is OPTIONAL.

☐ Make the codebase findable

- ☐The name of the codebase SHOULD be descriptive and free from acronyms, abbreviations, puns or organizational branding.
- ☐The codebase SHOULD have a short description that helps someone understand what the codebase is for or what it does.
- ☐Maintainers SHOULD submit the codebase to relevant software catalogs.
- ☐The codebase SHOULD have a website which describes the problem the codebase solves using the preferred jargon of different potential users of the codebase (including technologists, policy experts and managers).
- ☐The codebase SHOULD be findable using a search engine by codebase name.
- ☐The codebase SHOULD be findable using a search engine by describing the problem it solves in natural language.
- ☐The codebase SHOULD have a unique and persistent identifier where the entry mentions the major contributors, repository location and website.
- ☐The codebase SHOULD include a machine-readable metadata description, for example in a publiccode.yml file.
- ☐A dedicated domain name for the codebase is OPTIONAL.
- ☐Regular presentations at conferences by the community are OPTIONAL.

☐ Use a coherent style

- ☐The codebase MUST use a coding or writing style guide, either the codebase community's own or an existing one referred to in the codebase.
- ☐Contributions SHOULD pass automated tests on style.
- ☐The style guide SHOULD include expectations for inline comments and documentation for non-trivial sections.
- ☐Including expectations for understandable English in the style guide is OPTIONAL.

☐ Document codebase maturity

- ☐The codebase MUST be versioned.
- ☐The codebase MUST prominently document whether or not there are versions of the codebase that are ready to use.
- ☐Codebase versions that are ready to use MUST only depend on versions of other codebases that are also ready to use.
- ☐The codebase SHOULD contain a log of changes from version to version, for example in the `` `CHANGELOG` ``.
- ☐The method for assigning version identifiers SHOULD be documented.
- ☐It is OPTIONAL to use semantic versioning.