Governance structure of an open-sourced matchbox:

Draft v0.6

1. Introduction

Here we define the governance structure of the open source software system matchbox. It is an application that facilitates peer-to-peer connections between federated nodes of the Matchmaker Exchange[1]. It is open-sourced with the BSD license. We propose the following series of definitions and processes to help streamline the development and integration processes and provide guidelines to help interested parties in deciding to contribute-to and/or using matchbox.

2. Executive summary:

There are four types of contributor categories. **Only the founding and voting members can vote** to add a feature, and the **votes carry the same weight**. **Founding members alone have the capability to veto** a vote result. **Silence or non-participation on a vote from either party results in a positive vote** being counted from them. Founding and voting members can request a delay on a vote in 2 week increments, upto a maximum delay of 2 months.

3. Contributor definitions:

Founding member	A contributor belonging to an organization that was one of the original implementors of matchbox.
Voting member	A contributor belonging to an organization that uses matchbox and has at least 40 patients.
General member	A contributor belonging to an organization that uses matchbox and has less than 40 patients.

[1] http://www.matchmakerexchange.org

Observer	Does not contribute to code base, but observes development and contributes in other capacities such as user feedback, logistics.

4. Voting rights:

I. i. **Only founding and voting members are authorised to vote** in the decision making process. While general members and observers can influence a vote via lobbying and presenting user-feedback, they cannot cast a vote.

II. ii. A vote of a founding member or a voting member, carry the same weight towards the decision. **Founding members reserve the right to veto** a decision that was voted into action if they feel it is orthogonal to the strategic direction and/or growth of matchbox.

5. Requirements for initiating an action on matchbox before a vote:

I. Each action must be sponsored by a single member (founding, voting, general) who champions for it on behalf of an observer or themselfs.

II. A single member can sponsor only 4 simultaneous open issues at a time.

III. Any action to be made to matchbox must have a corresponding GitHub issue number for tracking purposes.

IV. Actions can be defined by two *types*.

    A. Hotfix: an action that is urgent and must be released to production as soon as possible.

    B. Feature: An enhancement to matchbox that gives it new functionality.

V. All collected actions of the type “feature” are brought to discussion for a vote at a meeting held on the first Monday of every month.

VI. Actions that are deemed hotfixs can be brought to discussion or vote immediately via Github mechanisms.

VII. Once above is complete, the action must be put forward for voting. Only past a favorable positive decision can the action be initiated.

6. Voting process:

I. Voting is done via Github mechanisms.

II. A majority vote is considered a consensus, and the feature will be prioritized and queued for implementation.

III. Silence by a voting member is considered a positive/favorable vote.

IV. Founding and voting members can request a delay on a vote in 2 week increments, upto a maximum delay of 2 months.

  A. Other members can request a delay via their sponsor of the feature.

V. In a subsequent meeting, all positive vote action’s implementation priority is set, and is defined as urgent, medium and normal.

7. Implementation process:

I. The sponsor of the original action is responsible for functioning as the champion of the action, following and reporting on progress.

II. If possible, the sponsor can implement the action themselves, or lobby the support of other developers.

III. Actions must be conducted on their own respective branches. The branch name must be prefixed with *type,* a hyphen, and followed by the GitHub issue ID. For example: hotfix-10

IV. Once the action is implemented and deployed to the main code branch, the issue can be closed and the action considered complete.