

Telco Global API Alliance Community Organization Guidelines (Governance Structure)

This document describes the rules and governance of the project. It is meant to be followed by all the developers of the project and the Telco Global API community. Common terminology used in this governance document is defined below:

- **The Telco Global API Project:** The sum of all activities and sub-projects performed under the governance protocol defined in this document, concerning one or more repositories or the community.
- **Team members:** Members of the private [Telco Global API Team](#) Google group.
- **Maintainers:** Leaders of an individual project or part thereof ([MAINTAINERS.md](#)).
- **Sub-projects:** A single repository in the Telco Global API [GitHub organization](#) led by a voluntary group, open to anyone to participate.

Telco Global API Project is an independent open-source project, not controlled by any single company or organization. To emphasize this and clarify the governance structure (Community Organization Guidelines), the Telco Global Project will be conducted in the [Cloud Native Computing Foundation](#), which all contributors are members of.

For specific guidance on the practical steps for contributing to any API sub-project please see our [CONTRIBUTING.md](#) guide.

I. Principles

The Telco Global API community adheres to the following principles:

- Open: Telco Global API is open source, advertised on our website/ See repository guidelines and CLA, below.
- Welcoming and respectful: As a CNCF member project, we abide by the [CNCf Code of Conduct](#).
- Transparent and accessible: Changes to the Telco Global API organization, Telco Global API code repositories, and CNCF related activities (e.g. level, involvement, etc.) are done in public.
- Merit: Ideas and contributions are accepted according to their technical merit and alignment with project objectives, scope, and design principles.

II. Telco Global API Project

The development of open, global, and interoperable API solutions could benefit customers and developer ecosystems by giving them access to the operator capabilities of the networks that customers connect to. This allows applications to run consistently between telco networks and countries.

Kommentiert [KA(1): Will there be a TAPI Community?

Kommentiert [F12]: What's the difference between the Project and the Alliance. They seem to be used interchangeably. A single name would be better, or definitions for both

Kommentiert [KA(3R2): Good Point! I adjusted the document. It is "Project" instead of "Alliance" in the whole document.

Kommentiert [F14]: Suggest that this is removed as the first sentence is broad enough

Kommentiert [KA(5R4): Will be deleted.

Kommentiert [KA(6): If necessary, create a private Google Group for TAPI like the CNCF project "Telco Global API" did: <https://groups.google.com/forum/?pli=1#!forum/Telco-Global-API-team>

Kommentiert [KA(7): If necessary, create "maintainers.md" for TAPI

Kommentiert [KA(8): TAPI should be set up as an organization in Github instead of a single repository

Kommentiert [KA(9): Each undertaking to develop a Service API should be a project for itself and therefore will be a sub-project of TAPI. For every sub-project create a repository under TAPI.

Kommentiert [KA(10): True or not?

Kommentiert [F111]: Would it be worth adding 'or organization'? I'm not sure what the relationship with e.g., the CNCF or Linux Foundation, but Company doesn't cover them.

Kommentiert [KA(12R11): Sure!

Kommentiert [KA(13): Define process for contributing to TAPI subprojects.

Kommentiert [KA(14): Will we have an own website?

Kommentiert [KA(15): To adjust

Kommentiert [KA(16): Are we going to extend T-Mobile US CNCF Membership onto the Deutsche Telekom?

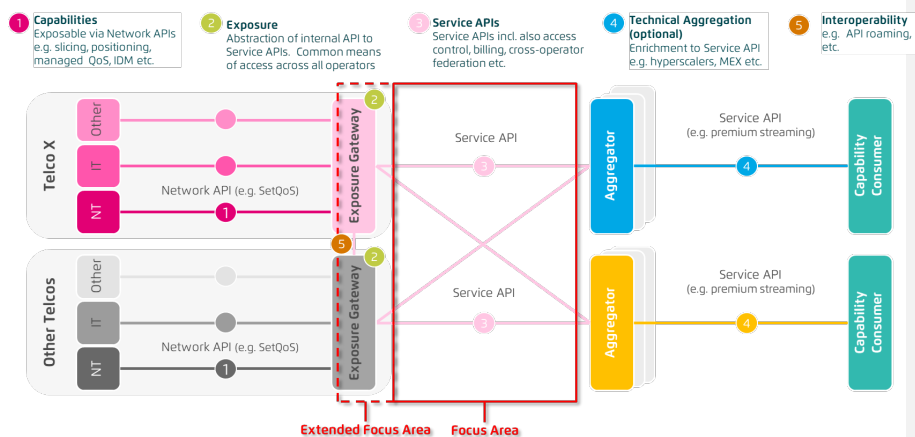
Kommentiert [F117]: If I understand this, sub-project work is always open, but committee work (may) not be. It could be worth making the distinction

Kommentiert [F118]: 'could' sounds very weak here. 'will' or at least 'intends to' would be better

Kommentiert [F119]: Probably best to remove 'and countries' as it's meaning is confusing (are countries networks that connect to Telcos?)

A new form of collaboration between various players including telcos, ISVs, device manufacturers and hyperscalers could address many of the challenges in scaling API services across heterogeneous operator architectures, advancing the connectivity industry towards a more robust and faster core network, encouraging the faster adoption of capabilities, resulting in value creation for the entire tech industry.

APIs grouped by services and bundled up into “Service APIs” could reduce the complexity of accessing operator capabilities for developers and enterprises.



Kommentiert [FI20]: Same comment as for 'could' above

A primary reason that the Telco Global API Project exists is as a forum for collaboration. Most of the work in the project should be done in the open. Community members and contributors must communicate in the open, ensure other community members and contributors can find notes of meetings, discussions, designs, and decisions, and periodically communicate a high-level summary of their contribution work to the community.

Kommentiert [FI21]: Should this be 'Telco Global API Alliance (or Project)' – The API is an API, not a forum!

Kommentiert [KA(22R21)]: Thanks! It was a „typo“. I added “Project”

Kommentiert [FI23]: This reduces what is stated in the Principles (Changes are done in public -> 'Most' of the work in the open)

Kommentiert [KA(24R23)]: In the principles it states that any changes are visible to the public. However this sentence here suggests that some parts of the project work of the project are done in an open source manner and other parts are not if we will a “private API group”. Tbd

Kommentiert [KA(25)]: Add definition

Kommentiert [FI26]: An org. structure diagram could be helpful, showing Project, steering, sub-projects, other committees, etc.

Kommentiert [KA(27R26)]: Yes, definitely.

III. Governance

Steering Committee

...

Committees

Some topics, such as Security or Code of Conduct, require discretion. Whereas sub-projects are led by voluntary groups which operate in the open and anyone can join. Whereas sub-projects do not have open membership and do not always operate in the open. The steering committee can form committees as needed, for a bounded or unbounded duration. Membership of a committee is decided by the steering committee, however, all committee members must be community members. A committee has a charter and a chair, and will report to the steering committee periodically, and to the community as defined by the committee's charter.

Kommentiert [KA(28)]: Edit

Kommentiert [FI29]: 'Community members' is not defined

IV. Project Structure

Sub-projects

The Telco Global API project is organized primarily into sub-projects. Each sub-project is comprised of members from multiple companies and organizations, with a common purpose of advancing the project with respect to a specific Service API topic, for example 'streaming' or 'localization'. Our goal is to enable distributed decision making and code ownership. This will be done by providing focused forums for getting work done, making decisions, and onboarding new contributors.

Areas covered by sub-projects may be vertically focused on particular Service API components or functions, cross-cutting/horizontal or spanning many/all functional areas of Service APIs. Some examples of each type are:

- Vertical: Micro Orchestration, Mediation, Transformation Traffic Management (API components)
- Horizontal: Scalability, Architecture, Lifecycle, Service

Sub-projects are documented in [subproject.yaml](#).

Each sub-project must have a MAINTAINERS.md file. A sub-project should have at least one maintainer. Ideally a sub-project is managed by two or more maintainers, depending on the size and scope of the sub-project. Here, the responsibilities must be clearly agreed upon between all of the maintainers. This includes co-ordinating who is responsible for which issues and pull requests.

Some sub-projects may have distinct (although sometimes overlapping) sets of contributors and maintainers, who act as the sub-project's technical leaders, organizers and facilitators.

Where a sub-project has a release process, access and documentation should be such that more than one person can perform a release. Releases should be announced on the [telco-global-api-users mailing list](#). Any new sub-projects should be first proposed on the [developers mailing list](#) following the voting procedures listed below. When a sub-project is no longer relevant, it should be moved to the 'telco-global-api-junkyard' GitHub organization.

Working Groups

Community rallying points are used to facilitate discussions/work regarding topics that are short-lived or that span multiple sub-projects.

Working groups are primarily used to facilitate topics of discussion that are in scope for the Telco Global API Project but that cross sub-project lines. If a subset of community members want to get together and discuss a topic, they can do so without forming a Working Group.

See [working group governance](#) for more details about forming and disbanding Working Groups.

Working groups are documented in [sigs.yaml](#).

V. Roles, Responsibilities and Requirements

Team members

Any members of the private Global API Team.

Team member status may be given to those who have made ongoing contributions to the Telco Global API project for at least 3 months. This is usually in the form of code submissions and

Kommentiert [KA(30): Random examples

Kommentiert [KA(31): Edit

Kommentiert [FI32]: The second half of this sentence doesn't really fit the first half. Would it be better to say "This will be done by providing focused forum for getting work....."?

Kommentiert [KA(33R32): Adjusted.

Kommentiert [KA(34): Edit

Kommentiert [KA(35): Note: What is API: Definition, Types, Specifications, Documentation: <https://www.altexsoft.com/blog/engineering/what-is-api-definition-types-specifications-documentation/>

Kommentiert [FI36]: Surely the sub-project must have at least one maintainer (otherwise you MUST have a MAINTAINER.md file, but it could be empty!)

Kommentiert [KA(37): Align on how a new release should be announced.

Kommentiert [KA(38): Clarify on whether we want to follow such a process with voting procedures.

Kommentiert [FI39R38]: There should be a dedicated section of the document that describes all of the communication methods/wikis/m/ls etc.

Kommentiert [KA(40R38): Correct, We still have to agree on how the communication will be coordinated.

Kommentiert [FI41]: Is 'project' or 'repo' actually meant here?

Kommentiert [FI42]: As this is charter, not a requirements document, this should use the form 'Community rallying point are used to facilitate...' There also needs to be a definition of what a rallying point is, how it is created, joined etc.

Kommentiert [KA(43R42): Thx!

Kommentiert [FI44]: Is he API itself or the project meant here?

Kommentiert [KA(45): edit

Kommentiert [KA(46): edit

Kommentiert [KA(47): Do we want a private Global API Team? Can anybody join or must the person be an employee of a hyperscaler or telco operator?

improvements and/or notable work on documentation but may also include other contributions such as organizing events or user support.

The current team members are:

- Nathan Rader (Deutsche Telekom)
- Elmar Mundt (Deutsche Telekom)
- Adrian Kisliuk (Detecon)
- Markus Kümmerle (Deutsche Telekom)
- Piotr Wesolowski (Deutsche Telekom)

Previous members:

- [None]

Please note that Telco Global API had received significant contributions from a number of unlisted individuals before this governance document was written, and thus formal team membership, was created.

Maintainers

Maintainers are first and foremost contributors that have shown they are committed to the long-term success of a sub-project. Contributors wanting to become maintainers are expected to be deeply involved in contributing code, pull request review, and triaging issues in the project for a minimum of three months.

Requirements

The process for becoming a maintainer is defined in the Telco Global API project charter.

The following requirements must be met by an individual wishing to become a maintainer for a sub-project:

- Deep understanding of the technical goals and direction of the sub-project
- Deep understanding of the technical domain of the sub-project
- Sustained contributions to the design and direction of the sub-project, demonstrated by performing all of the following:
 - Authoring and reviewing proposals
 - Initiating, contributing and resolving discussions (emails, GitHub issues, meetings)
 - Identifying subtle or complex issues in designs and implementation PRs
- Direct contributions to the subproject through implementation and / or review

Responsibilities

Maintainers lead one or more sub-project(s) or parts thereof and serve as a point of conflict resolution amongst the contributors and are the technical authority for a subproject (responsible for vision and direction and overall design, choose/approve change proposal (KEP) approvers, field technical escalations, etc.). They **MUST** have demonstrated both good judgement and responsibility towards the health of that sub-project. Sub-project Owners **MUST** set technical direction and make

Kommentiert [FI48]: Is sub-project meant here?

Kommentiert [KA(49R48): Correct. Thx!

Kommentiert [FI50]: Should there be a statement about an ongoing time / availability commitment from the maintainer?

Kommentiert [FI51]: Can you lead a part of a sub-project (sub-sub-project leader?:-)

Kommentiert [KA(52R51): 1/ There may be different types of maintainers ("reviewer" and "committer" as mentioned below) which would only be responsible for certain things in the sub-project.

2/ Also a maintainer can work on several sub projects at the same since a sub-project can have multiple maintainers. Maintainers can organize themselves and define what exactly they will be responsible for or working on in the sub-project in question.

Kommentiert [FI53]: Bulleted list, as above would be clearer here.

Kommentiert [FI54]: Probably should be moved to the list above

Kommentiert [KA(55R54): I think it wraps up the above mentioned points.

Kommentiert [FI56]: The term 'owner' is used several times, but the role is not defined.

Kommentiert [KA(57R56): Right. The text is copy pasted from the Kubernetes project. Sub-project owners in Kubernetes = Maintainer in Telco Global API

or approve design decisions for their sub-project - either directly, or through delegation of these responsibilities.

Maintainers are also intended to be organizers and facilitators, responsible for the continued operation of the sub-project and for communication and co-ordination with the other sub-projects, the Steering Committee, and the broader community. Ideally, maintainers are also team members, but exceptions are possible for suitable individuals that, for whatever reason, are not yet team members. Maintainers are granted commit rights to all sub-projects in the [GitHub organization](#).

Defined by: *owners* entry in subproject [OWNERS](#) files as defined by [sigs.yaml](#) *subproject.owners*

Maintainers should actively participate in Pull Request reviews and are expected to respond to assigned Pull Requests in a *reasonable* time frame, either providing insights, or assign the Pull Requests to other maintainers.

The following responsibilities must be met by the maintainer for a sub-project:

- Make and approve technical design decisions for the sub-project.
- Set the technical direction and priorities for the sub-project.
- Define milestones and releases.
- Mentor and guide approvers, reviewers, and contributors to the sub-project.
- Ensure continued health of sub-project
 - Adequate test coverage to confidently release
 - Tests are passing reliably (i.e. not flaky) and are fixed when they fail
- Ensure a healthy process for discussion and decision making is in place and is followed by the sub-project's members.
- Work with other sub-project owners to maintain the project's overall health and success holistically.

Types of maintainers

There are different types of maintainers (Reviewers and Committers) with different responsibilities, but all maintainers have 3 things in common:

1. They share responsibility in the project's success.
2. They have made a long-term, recurring time investment to improve the project.
3. They spend that time doing whatever needs to be done, not necessarily what is the most interesting or fun.

Reviewers

- A reviewer is a maintainer within the project. They share in reviewing issues and pull requests and their LGTM counts towards the required LGTM count to merge a code change into the project.
- Reviewers are part of the organization but do not have write access. Becoming a reviewer is a core aspect in the journey to becoming a committer.

Kommentiert [FI58]: Would ,operation and progress of' be better?

Kommentiert [KA(59)]: Sub-project owners in Kubernetes = Maintainer in Telco Global API

Kommentiert [KA(60)]: Expectations from Maintainers

Kommentiert [KA(61)]: Becoming a Maintainer/Adding Maintainers

Kommentiert [FI62]: A review can't really be defined as a maintainer, as they don't have commit access

Kommentiert [KA(63R62)]: In this definition of „reviewer“ (which is the main type of maintainer“ they do have commit access. As a reference you can check out the containerd project in CNCF: <https://github.com/containerd/project/blob/main/GOVERNANCE.md>

Committers

- A committer is a core maintainer who is responsible for the overall quality and stewardship of the project. They share the same reviewing responsibilities as reviewers but are also responsible for upholding the project bylaws as well as participating in project level votes.
- Committers are part of the organization with write access **to all repositories**. Committers are expected to remain actively involved in the project and participate in voting and discussing of proposed project level changes.

Security Advisors

- A security advisor is an advisory role in the project responsible for helping classify and advise on **embargoed** security disclosures. Security advisors are part of the organization without write access, but with read access to security disclosures and advisories before becoming public. Security advisors help maintain the integrity of the security review process and encourage responsible disclosure.
- Security advisors are individuals trusted by maintainers and representing significant users of the project. A reviewer may also be a security advisor, however, committers do not need this role as it is part of their regular duties. The security advisor duties are not part of the duties of being a reviewer. There is no expectation of advisors to become reviewers or participate in issue triage and code review.

Kommentiert [FI64]: Does repository == sub-project (i.e. is there one per sub-project)?

Kommentiert [KA(65R64)]: Yes, as mentioned on page 1 under "sub-project". Thanks for spotting this! I still have to make the adjustment in the doc.

Kommentiert [FI66]: Does being a maintainer on one project entitle you to commit to every other project?

Kommentiert [KA(67R66)]: A maintainer can work on several sub-projects at the same time. A maintainer might be a committer in sub-project A and a reviewer in sub-project B for example ... or be a reviewer in two sub-projects.

Kommentiert [KA(68)]: Do we need a Security Advisor?

Kommentiert [AK69]: The software **security** industry uses the term **Embargo** to describe the period of time that a **security** flaw is known privately, prior to a deadline, after which time the details become known to the public.

VI. Decision Making

Voting Principles

The Telco Global API project usually runs by informal consensus, however **sometimes a formal decision must be made**.

Depending on the subject matter, different methods of voting are used as laid out **below**.

For all votes, voting must be open for at least one week. The end date should be clearly stated in the call to vote. A vote may be called and closed early if enough votes have come in one way so that further votes cannot change the final decision.

In all cases, **all and only team members** are eligible to vote, with the sole exception of the forced removal of a team member, in which said member is not eligible to vote.

Discussion and votes on personnel matters (including but not limited to team membership and maintainership) are held in private on the **team mailing list**. All other discussion and votes are held in public on the **developer mailing list**.

For public discussions, anyone interested is encouraged to participate. Formal power to object or vote is limited to **team members**.

Consensus

The default decision making mechanism for the Telco Global API project is **lazy consensus**. This means that any decision on technical issues is considered supported by the **team** as long as **nobody objects**.

Kommentiert [FI70]: Grounds for this would be useful to avoid the process being abused.

Kommentiert [AK71]: review

Kommentiert [FI72]: Is 'team members' meant according to the definition on the first page? i.e. anyone that is a member of the Google group. Can all team members vote on every topic, even ones only relevant to other projects? How do you stop orgs. From signing up 50 team members on the day of an important vote? We've seen this happen.

Kommentiert [KA(73R72)]: Excellent point!

Kommentiert [FI74]: 'nobody objects' probably needs to be further extended to give some indication on what is considered valid grounds for objection (e.g., nobody objects based on substantiated technical grounds). This phrase probably needs work though as there may be other reasons that an objection can be justified.

Silence on any consensus decision is implicit agreement and equivalent to explicit agreement. Explicit agreement may be stated at will. Decisions may, but do not need to be called out and put up for decision on the [developers mailing list](#) at any time and by anyone.

Consensus decisions can never override or go against the spirit of an earlier explicit vote.

If any [team member](#) raises objections, the team members work together towards a solution that all involved can accept. This solution is again subject to lazy consensus.

In case no consensus can be found, but a decision one way or the other must be made, any [team member](#) may call a formal [majority vote](#).

Majority vote

Majority votes must be called explicitly in a separate thread on the appropriate mailing list. The subject must be prefixed with [VOTE]. In the body, the call to vote must state the proposal being voted on. It should reference any discussion leading up to this point.

Votes may take the form of a single proposal, with the option to vote yes or no, or the form of multiple alternatives.

A vote on a single proposal is considered successful if more vote in favor than against.

If there are multiple alternatives, members may vote for one or more alternatives, or vote “no” to object to all alternatives. It is not possible to cast an “abstain” vote. A vote on multiple alternatives is considered decided in favor of one alternative if it has received the most votes in favor, and a vote from more than half of those voting. Should no alternative reach this quorum, another vote on a reduced number of options may be called separately.

Supermajority vote

Supermajority votes must be called explicitly in a separate thread on the appropriate mailing list. The subject must be prefixed with [VOTE]. In the body, the call to vote must state the proposal being voted on. It should reference any discussion leading up to this point.

Votes may take the form of a single proposal, with the option to vote yes or no, or the form of multiple alternatives.

A vote on a single proposal is considered successful if at least two thirds of those eligible to vote vote in favor.

If there are multiple alternatives, members may vote for one or more alternative, or vote “no” to object to all alternatives. A vote on multiple alternatives is considered decided in favor of one alternative if it has received the most votes in favor, and a vote from at least two thirds of those eligible to vote. Should no alternative reach this quorum, another vote on a reduced number of options may be called separately.

Technical decisions

Technical decisions that only affect a single sub-project are made informally by the maintainer of this sub-project, and [lazy consensus](#) is assumed. Technical decisions that span multiple parts of the Telco Global API project should be discussed and made on the [Telco Global API developer mailing list](#).

Decisions are usually made by [lazy consensus](#). If no consensus can be reached, the matter may be resolved by [majority vote](#).

Kommentiert [F175]: I can see what this is trying to do, but it also means ‘if we make a mistake, we can’t undo it’

Kommentiert [F176]: I’m unsure about the use of majority voting mechanisms in this case, because it’s open to abuse (as described above) and it goes against the ‘meritocracy’ principal. If the maintainer/architect for a sub-project’s vote is worth the same as someone who’s just signed up, then all of the decision making becomes a numbers game.

Team Members

New [members](#) may be proposed by any existing member by email to [Telco Global API-team](#). It is highly desirable to reach consensus about acceptance of a new member. However, the proposal is ultimately voted on by a formal [supermajority vote](#).

If the new member proposal is accepted, the proposed team member should be contacted privately via email to confirm or deny their acceptance of team membership. This email will also be CC'd to [Telco Global API-team](#) for record-keeping purposes.

If they choose to accept, the [onboarding](#) procedure is followed.

Team members may retire at any time by emailing [the team](#).

Team members can be removed by [supermajority vote](#) on [the team mailing list](#). For this vote, the member in question is not eligible to vote and does not count towards the quorum. Any removal vote can cover only one single person.

Upon death of a member, they leave the team automatically.

In case a member leaves, the [offboarding](#) procedure is applied.

Changes in Governance

Changes in project governance (GOVERNANCE.md) could be initiated by opening a github PR. The PR should only be opened no earlier than 6 weeks before the end of the project lead's term. The PR should be kept open for no less than 4 weeks. The PR can only be merged follow the same voting process as in Changes in Project Lead.

Editorial changes

Editorial changes are changes which fix spelling or grammar, update work affiliation, or similar; they update style or reflect an outside and obvious reality. They do not change the intention or meaning of anything in this document. They must be made via PR and accepted via [lazy consensus](#).

Changes in Maintainership

Changes in maintainership have to be announced on the [developers mailing list](#). They are decided by [lazy consensus](#) and formalized by changing the MAINTAINERS.md file of the respective repository.

A maintainer or committer may resign by notifying the [team mailing list](#). A maintainer with no project activity for a year is considered to have resigned. Maintainers that wish to resign are encouraged to propose another team member to take over the project.

FAQ section

The FAQ section is informational and can be changed by [lazy consensus](#) unless challenged. If no consensus can be reached, the matter may be resolved by [majority vote](#).

Other matters

Any matter that needs a decision, including but not limited to financial matters, may be called to a vote by any member if they deem it necessary. For financial, private, or personnel matters, discussion and voting takes place on the [team mailing list](#), otherwise on the [developer mailing list](#).

Kommentiert [F177]: Is this an individual or organization? If it's individuals, then there is likely a scaling problem – supermajority votes needed to get them involved when there are likely hundreds of members.

Kommentiert [KA(78): Changes in Maintainership

VII. On- / Offboarding

The On- / Offboarding section is informational and can be changed by [lazy consensus](#) unless challenged. If no consensus can be reached, the matter may be resolved by [majority vote](#).

Onboarding

The new member is

- added to the list of [team members](#). Ideally by sending a PR of their own, at least approving said PR.
- announced on the [developers mailing list](#) by an existing team member. Ideally, the new member replies in this thread, acknowledging team membership.
- added to the [GitHub organization](#) as *Owner*.
- optionally added to the community, junkyard, and related organizations and repositories.
- added to the [team mailing list](#).
- added to the Telco Global API.io GSuite account with a user name of the new member's choice. (Most importantly, this comes with a <chosen-username>@Telco Global API.io email address and provides access to the team's GDrive and calendar. The new member should add the latter to their own calendar list.)
- announced to CNCF.
- given access to the shared password storage.

Offboarding

The ex-member is

- removed from the list of [team members](#). Ideally by sending a PR of their own, at least approving said PR. In case of forced removal, no approval is needed.
- removed from the [GitHub organization](#) and related organizations and repositories. Optionally, they can retain maintainership of one or more repositories if the [team](#) agrees.
- removed from the [team mailing list](#) and demoted to a normal member of the other mailing lists, i.e [developers](#), [users](#), and [announce](#).
- announced as removed to CNCF. We will explicitly request reconfirmation of removal from CNCF.
- removed from the shared password storage. All passwords, API tokens, etc. are cycled in due course (i.e. immediately in case of an involuntary departure but can be appropriately batched with other departures or routine cycling otherwise).
- removed from group accounts where applicable. Services with some variety of a group account include but are not restricted to Digital Ocean, DockerHub, GSuite, Netlify, Twitter (via Tweetdeck), Youtube.
- not allowed to call themselves an active team member any more, nor allowed to imply this to be the case.
- added to a list of previous members if they so choose.

If needed, we reserve the right to publicly announce removal.

VIII. Stepping down policy

Life priorities, interests, and passions can change. If you're a maintainer but feel you must remove yourself from the list, inform other maintainers that you intend to step down, and if possible, help find someone to pick up your work. At the very least, ensure your work can be continued where you left off.

After you've informed other maintainers, create a pull request to remove yourself from the MAINTAINERS file.

IX. Licensing

- Apache 2.0 license for API source codes
- Creative Commons license for documentation

X. FAQ

How do I propose a decision?

Send an email to [the developer mailing list](#) with your motion. If there is no objection within a reasonable amount of time, consider the decision made. If there are objections and no consensus can be found, a vote may be called by a team member.

How do I become a team member?

To become an official team member, you should make ongoing contributions to one or more project(s) for at least three months. At that point, a team member (typically a maintainer of the project) may propose you for membership. The discussion about this will be held in private, and you will be informed privately when a decision has been made. A possible, but not required, graduation path is to become a maintainer first.

Should the decision be in favor, your new membership will also be announced on the [developers mailing list](#).

How do I add a project?

As a team member, propose the new project on the [developer mailing list](#). If nobody objects, create the project in the GitHub organization. Add at least a README.md explaining the goal of the project, and a MAINTAINERS.md with the maintainers of the project (at this point, this probably means you).

How do I archive or remove a project?

Send an email to the [developer mailing list](#) proposing the retirement of a project. If nobody objects, move it to the Telco Global API-junkyard GitHub organization.

How do I remove an inactive maintainer?

A maintainer may resign by notifying the [team mailing list](#). A maintainer with no project activity for a year will be treated as if they had resigned. If there is an urgent need to replace a maintainer, discuss this on the [team mailing list](#).

How do I remove a team member?

Team members may resign by notifying the [team mailing list](#). If you think a team member should be removed against their will, propose this to the [team mailing list](#). Discussions will be held there in private.

--- -- -- --

Personal Notes on how to structure the governance of the Telco Global API Governance (not part of the final document):

<https://www.cncf.io/blog/2019/08/30/cncf-technical-principles-and-open-governance-success/>

- Define charter that specifies its scope (topics, subsystems, code repos and directories), responsibilities, areas of authority, how members and roles of authority/leadership are selected/granted, how decisions are made, and how conflicts are resolved.
- See the [SIG charter process](#) as example for details on how charters are managed. SIGs should be relatively free to customize or change how they operate, within some broad guidelines and constraints imposed by cross-SIG processes (e.g., the release process) and assets (e.g., the Kubernetes repo).
- A primary reason that SIGs exist is as forums for collaboration. Much work in a SIG should stay local within that SIG. However, SIGs must communicate in the open, ensure other SIGs and community members can find notes of meetings, discussions, designs, and decisions, and periodically communicate a high-level summary of the SIG's work to the community.
- See [sig governance](#) for more details about current SIG operating mechanics, such as mailing lists, meeting times, etc.
- More information:
[SIG Governance Requirements](#)
[SIG Lifecycle](#) - for a tactical checklist on creation and retirement

Some other subjects that I think need to be covered in this document

Diversity statement

Describes the intent of the community to promote inclusivity and equality across all walks of life.
<https://www.contributor-covenant.org/>

Conflict of Interest statement

Guidelines to mitigate or remove motivations or decision-making that may reflect the interests of one individual or organization exclusively.
https://opensource.org/conflict_of_interest_policy

IPR – I think there should be a statement about the expectations on the members regarding their existing IPR and any new IPR which is generated as part of this work.

Conflict management policy

Articulates processes and guidelines for managing conflict (social and technical) within communities.

Example at: <https://metabrainz.org/conflict-policy>

Probably would need some updating

Communications

How /where is project information stored (docs/presentations/minute/decisions, etc.), wikis etc.

Day-to-day communication (email lists, DM apps etc.

It would be useful to have a single section that details all of this.

Finances

There probably should be some text regarding how financial matters will be dealt with. I guess that this would say that the Telco Global API Project does not have any budget and will not meet any costs – they are the responsibility of the contributors or their employer etc. and will not be remunerated. This is going to be particularly important when you start going through the standardization process (3GPP?).