CCB Team Charter

The Mojaloop Change Control Board charter v1.0

Table of Contents

CCB Team Charter

1 Introduction

This document provides the team charter for the Change Control Board (CCB) and the Special Interest Groups (SIGs) that are part of the CCB, to facilitate the functioning, decision-making of the groups, by defining roles, responsibilities, guidelines regarding scope, processes for change proposals/requests, finalizing decisions, onboarding of new members, timelines, voting (in case consensus couldn't be reached) and such others.

1.1 Duration

The SIGs and CCB can propose changes to the charter and adopt them as agreed by the SIGs and ratified by the CCB. If no decision is made regarding the renewal (based on the voting process if needed) or if a change request to the charter is not approved, the same charter stays in effect. See section 4.4 for more information if consensus cannot be reached regarding changes to the charter.

2 Mission

The primary goals of the CCB and the SIGs are to:

1. Drive wider adoption of the Mojaloop Specification[1]
2. Improve clarity and help promote uniformity of implementations
3. Carefully balance considerations of existing implementations and new business needs for changes to keep the burden on existing implementations to a minimum while supporting changes needed to meet new business needs
4. Support changes needed to meet the business needs of adopters by keeping the standard relevant to defend against forking
5. Drive the costs lower for implementation and maintenance

Apart from these, the CCB also takes the responsibility to safeguard the **Mojaloop Family of API** specifications so that it upholds the Level One Principles. As part of maintenance, the CCB considers, debates critical feature requests and change requests generic enough to be part of the Specification itself and after deliberation decides whether to accept them or not. If agreed upon by the group, these changes are implemented and deployed as well, as per market relevance.

2.1 Background

As part of the Bill and Melinda Gates Foundation's (BMGF) **Financial Services for the poor** initiative, several projects have been initiated and pursued following the Level One Principles[2], to achieve its goals and objectives. Level One is a vision for a new digital payments platform that supports inclusive, interoperable, digital economies design principles to achieve this[3]. The Level One project aims to provide a set of tools and resources to implement a Level One platform in the form of Specifications, Standards and Open Source code. The requirements for a Level One aligned platform include the platform being Secure, Affordable, Convenient, Open and Robust.

One of these projects was the Product Development Partnership (PDP) project, formed with partner teams from Ericsson, Huawei, Mahindra Comviva and Telepin as subject matter experts with the effort overseen by the BMGF and supported by Alvarez & Marsal, ModusBox and Ripple. This project brought together leading software providers in the Telecom industry to develop a standard for money movement based on LevelOne principles.

2.2 Product Development Partnership (PDP)

The primary objective of the PDP project was to enable Interoperability for money transfers by providing an API Specification for Financial Service Providers (FSPs) to interact either bilaterally or through a common Switch.

The partner teams and organizations that were part of the PDP Project agreed to make the contributions made towards achieving this Specification public and that the Standard would be Open and free for all adopters. The full text of the joint statement by the PDP partners and the BMGF declaring the licensing and other details is in the Section 2.2.1.

As part of the project, a standard was designed, developed, implemented, tested for interoperability and finally adopted in three phases. The API that was produced as a result was branded the FSP Interoperability (**FSPIOP)** API.

[1] Mojaloop Specification is a combination of the Mojaloop Family of APIs – The FSPIOP API, Administration API, Third-party API, Settlement API and the set of Business documents that provide guidelines on Scheme Rules and choices: https://docs.mojaloop.io/mojaloop-business-docs

[2] Level One project - https://leveloneproject.org/

[3] From L1P Bootcamp presentation - https://k-learn.adb.org/system/files/materials/2017/11/201711-level-1-project-boot-camp.pdf

2.2.1 Joint Declaration and License[4]

This license is specific to the FSP Interoperability API and does not apply to the rest of the APIs that are part of the Mojaloop family of APIs (as long as they're not derivates of the FSPIOP API itself or re-use structures that are originally part of it).

The Bill & Melinda Gates Foundation believes that an economy that includes everyone, benefits everyone. In support of this goal, we asked leading mobile wallet technology providers Ericsson, Huawei, Mahindra-Comviva and Telepin to work together to create a set of APIs for interoperability within the digital financial services infrastructure. Together with consultants from Interledger and ModusBox, the group worked to produce the API currently referenced as the Mojaloop FSP Interoperability API.

The underlying charitable goal for the API is to spur innovation and access to digital products and services that serve the financially underserved with a focus on interoperability and strengthening and accelerating the availability of solutions that reflect the design principles of L1P as documented on www.leveloneproject.org. The contributors commit to making the relevant background technology which is provided to the API project necessary to implement the API in furtherance of the charitable goals available on a royalty-free basis.

2.2.2 Change Control Board (CCB) formation

After the *Build & Testing* phase of the PDP project, version 1.0 of the Specification document set has been publicly released on 13th March 2018. The PDP group then transitioned into a maintainer of the standard, while assuming the name "Change Control Board" (**CCB**), to act as a gatekeeper for the changes to the Specification and its general maintenance.

2.3 LevelOne project & Mojaloop

The Level One Project initiative by the BMGF was branded as **Mojaloop** towards the second half of the year 2017 and at this juncture, the **FSPIOP API** developed by the then PDP group has been adopted as the standard for the Mojaloop project.

After this initial API to support clearing of funds, several other APIs were introduced based on product and feature requirements. These include the **Administration** API, the **Settlement** API which are part of standard Mojaloop implementation. The **Third-party** API has been introduced more recently to support Payment Initiation Service Providers (PISP) which is under active development. There are other APIs for **Reporting**, **Cross border**/**currency** currently on the roadmap.

2.3.1 Mojaloop License

This license is the generic Mojaloop license that applies to the Mojaloop family of APIs except for the FSPIOP API.

[4] License on GitHub: https://github.com/mojaloop/mojaloop-specification/blob/master/LICENSE.md

3 Change Control Board

This section covers the aspects of membership, roles and responsibilities of the CCB and the SIGs and their general structure.

Considering the goals of the Level One initiative, the recommendation is to have representation from various industries or domains such as Switch Operators, System Integrators, Regulators, Financial Inclusion Advocates, Mobile Network Operators (MNOs), Payment Initiation Service Providers (PISPs) and such others. This is to have inputs from various sources and different perspectives to provide objective inputs to the CCB and enable making decisions for the betterment of the **Mojaloop Family of APIs** and to support wider adoption of the Mojaloop standard.

3.1 CCB membership

3.1.1 Initial authors

The initial group consisted of financial inclusion Advocates/Sponsors (BMGF), four partners (Leading Telecom software providers – Ericsson, Huawei, Mahindra Comviva, Telepin), who continued the partnership from earlier when the Specification was developed and evaluated.

Recognizing the contributions of the initial working group and Sponsors of the project, representatives from the BMGF and the four partners Ericsson, Huawei, Mahindra Comviva and Telepin will continue to be members of the FSPIOP SIG and cannot be voted out, until the companies themselves choose to leave.

3.2 Special Interest Groups (SIGs)

Each new initiative such as support for a new use case, feature or functionality which will result in the definition of one or more APIs, will define those APIs through a Special Interest Group. SIGs will be formed as a consequence of new functional initiatives being adopted by the Mojaloop community.

However, not all functional initiatives will require new APIs to be developed, and hence new SIGs formed. The decision on whether or not a new functional initiative requires a new API will be taken by the CCB, to which requests for new functionality will initially be directed. Since the CCB contains representatives of all the existing APIs, it is the natural forum in which to assign new functional areas to existing APIs, or to mandate the formation of a new API and hence a new SIG. The responsibility for forming the SIG itself, if the CCB decides that a new one is needed, will devolve to the group developing the functionality.

We envisage that, for new initiatives, a SIG will initially be formed from among the design and development teams working on the initiative, and that it will subsequently be expanded by recruits from organizations which have a particular interest in the area covered by the API(s). In the case of the existing FSP Interoperability API, a SIG was formed to manage changes to it as they are required.

As of June 2021, there are SIGs for the FSP Interoperability API, the third-party (or PISP) API and the Administration API.

3.2.1 SIG Membership

Any interested party should be able to register to join a SIG as a non-voting member. This should be an automated process which allows parties to self-register to express interest. The members of that particular SIG will then make a decision to have the registrant join as a non-voting member. After a period of *2 months*, there can be a motion brought in by one of the members of the SIG to promote the non-voting member to the voting group, based on their involvement, contributions and interest. Based on majority vote, they can then be made part of the particular SIG. Parties should be able to de-register. All registered members of a SIG should receive invitations to SIG meetings and notification of issues raised. A person may have more than one role in a SIG (for example, CCB Rapporteur, Core SIG member).

The following table shows the general composition of the SIGs.

Role / Description	Function / Responsibility
SIG Convenor	Convenor, Voting member
CCB Convenor	Observer
CCB Rapporteur	Observer
Core SIG member	Voting member
Registered SIG community member	Non-voting member
Community member	Attendee (based on request)

3.2.2 Implementers

As described in section 3.2, implementers with a particular interest in a particular API will be able to join the SIG which manages that API. We do not envisage that there should be a separate SIG for implementers, since our expectation is that the Product Management structure in the Mojaloop community will be the natural forum for discussions among implementers; that this will be capable of generating new functional requirements for the Mojaloop product; and that these will be communicated to the relevant CCB SIGs as well as to the Design Authority.

3.2.3 Convenors

Each SIG will have a formally designated convenor, who will be responsible for the following areas:

1. Definition of the SIG’s strategy and roadmap
2. Maintaining the documentation of the API(s)
3. Maintaining the Swagger / OpenAPI definition(s) of the API(s)
4. Convening meetings of interested parties to discuss the creation or modification of the API(s) which the SIG manages and documenting their decisions
5. The internal structure of the SIG: its decision-making processes, frequency of meetings, new-membership requests, general membership, etc.
6. Perform threat analysis and define minimum security, audit, compliance, and risk management measures that should be considered before going live
7. Managing relations between the SIG and the CCB

It will be the responsibility of each SIG to select a convenor. The current convenor of an SIG or CCB should help transition the on-boarding of the succeeding convenor as voted by the respective SIG or CCB.

3.2.4 Charter

Each SIG will inherit the charter of the overall CCB and will be required to formally adopt it before it can be recognized by the CCB. Any changes to the CCB charter will also apply to the SIGs.

3.3 CCB responsibilities

Following the formation of the Mojaloop Foundation, the Change Control Board (CCB) will report to the Technical Governance Board (TGB) of the Mojaloop Foundation. The CCB will be responsible for the following areas.

3.3.1 Best practice

The CCB will be responsible for defining best practice in API development and documentation, and for satisfying itself that SIGs have followed best practice in the development, modification and publication of their APIs.

3.3.2 Common documentation

Most APIs developed by SIGs will have common standards which must be followed by participants who use them. These standards will include, for instance:

1. The security requirements for messages based on the SIG's threat and risk level
2. The structure and content of headers
3. The command grammar of the interface

The CCB will be responsible for the documentation which describes these standards, and SIGs will be expected to include a reference to the CCB documentation in their API definitions.

Each SIG will have an area in the documentation GitHub repository which it can use to maintain its own code and documentation, together with ways to raise and resolve issues, create and merge branches, and so on.

3.3.3 Common data structures

We expect that, in several cases, different APIs will use common data objects. This will be the case if, for instance, both the Open API for FSP Interoperability and the PISP API implement the **GET */parties*** resource. In this case, we expect the following design pattern to be used.

Data objects will be defined by their Swagger definition, which will be held in a common repository. Different APIs which use a common data object will include these data objects in their own Swagger definitions. The CCB will also maintain written documentation of the data objects in the form of definitional tables, and SIGs will include these items in their own API definitions, together with references to the CCB definitions.

When there is a need to modify the structure of a common data object, we expect that this proposal will originate with a SIG, who will request the CCB to make the change. The CCB will be responsible for managing the change, coordinating discussions with the other SIGs who use the object, and for obtaining agreement from the SIGs to the proposed change. Whether overloading objects should be supported and, if so, how, is a matter for later discussion and definition.

3.3.4 Harmony and efficiency

There will be a need to ensure that, in a world where different groups are responding to their own individual development needs, they do not wind up solving what are essentially similar problems in different ways, leading to divergence among the APIs and unnecessary complication in the development and implementation of code to support the API definitions. In particular, it is important that actual and potential users of different APIs in the Mojaloop family have a clear and coherent overall view of what the individual APIs do and how they relate to each other. The CCB will:

1. Review all SIG API proposals, hopefully at an early stage
2. Identify areas where opportunities exist to harmonize work being undertaken by SIGs with work that has been, or is being, undertaken by other SIGs
3. Propose ways of aligning the efforts of the SIGs concerned such that individual requirements can be met in a context of overall efficiency; and negotiate their adoption by the SIGs concerned

3.3.5 Review and approval

The CCB is the final authority for approval of APIs. The grounds on which it may reject proposed APIs, or ask for them to be revised, will be restricted to the following:

1. Failure to follow best practice in Mojaloop APIs.
2. Failure to meet documentation standards.
3. Failure to follow harmony and efficiency recommendations.

3.3.6 Membership

All SIG convenors will be members of the CCB *ex officio*. Other members may be co-opted onto the CCB as required.

The CCB will have a convenor in the same way as SIGs do. It will be the CCB convenor's responsibility to organize and publicize meetings of the CCB, manage the agenda and record proceedings and decisions. The CCB will also have a Rapporteur, who will be responsible for managing relationships between the CCB and the other bodies in the community with which it interacts: for instance, the Design Authority (DA), the Product Management Group, the Technical Governance Board (TGB) and so on. The convenor and the rapporteur will be elected by the Mojaloop community (2022 onwards). The elections for the convenors and rapporteur are proposed to happen in December of each year so that they can take over at the beginning of the next year.

3.4 Structure of the SIGs and CCB

Outlined below, is the structure for the SIGs and the CCB.

3.4.1 Special Interest Groups

The following table shows the initial set of Special Interest Groups that have been defined as of September 2021. Even though the convenors of the SIGs change, the structure should be maintained subject to the preference of the CCB and the SIGs.

Name	Focus area
FSP Interoperability API	Core Mojaloop functionality: maintenance and querying of ALS, initiation & completion of funds transfers.
Third-party (PISP) API	Functionality relating to PISPs and to the DFSPs that interact with them: account linking and transfer initiation
Settlement API	Functionality to support the definition and execution of settlements in Mojaloop switches and the recording of liquidity cover provisions.
Administration API	Functionality to support the creation and maintenance of a functioning Mojaloop system: creating, modifying and suspending participants

3.4.2 Change Control Board

The following table shows the composition of the inaugural Change Control Board as of June 2021.

Role / Description	Function
TGB representative	Observer, Voting member
BMGF representative	Observer, Voting member
Mojaloop Foundation Community Manager	Observer, Voting member
Mojaloop Foundation - Product Strategy	Observer
Rapporteur	Voting member
Convenor, CCB	Convenor
Convenor, FSP Interoperability API SIG	Voting member
Convenor, Third-party (PISP) API SIG	Voting member
Convenor, Administration API SIG	Voting member

3.5 Financial inclusion advocates / sponsors as observers

As the initial sponsors of the PDP project and critical stakeholders in the success of the Mojaloop Specification, the Bill & Melinda Gates Foundation participates in the CCB as observers, ensuring smooth functionality to improve adoption of the APIs and achieve the Level One goals. They also ensure that any impediments to the adoption of the standard by the relevant parties are removed and risks mitigated by working with the technology/business partners and other stakeholders. These members also provide inputs to the team regarding the Standards and the Specification itself to safeguard the overall goals of the Level One initiative.

In addition, after the formation of the Mojaloop Foundation (MLF), the Community Manager of the MLF participates in the CCB as an observer (or another representative from the MLF as needed).

As the Mojaloop standard evolves, it is possible that more such members from different funding / supporting organizations or the Mojaloop community become part of the CCB.

3.6 Community members' participation

Community members with interest in specific topics can participate actively in conversations on issues (using issues or such on GitHub) and may also be invited to present issues at the SIG or CCB meetings.

The conversations in the CCB, SIGs are encouraged to happen in public to avoid repetition and for wider visibility. The CCB and SIGs have adopted "public working methods" which include using GitHub issues for all conversations, using open channels (such as Mojaloop Slack) for communication to ensure transparency and visibility. The CCB meetings happen over webinars, which are open to the public to join in and listen to as attendees and ask relevant questions. SIG meetings happen over other widely used conferencing tools and any interested community member should be able to join as a listener to these calls, facilitated by the SIG convenors.

Even though the voting on issues is limited to primary or alternate team members, the discussions, conversations regarding issues are open to the public as much as possible on GitHub. For more information on voting, refer to section 4.5. Interested members from the community are allowed to join and present on the CCB/SIG meetings/calls as well on a case-by-case basis based on interest and need.

3.7 Focused working groups

When a change request or proposal is made regarding a topic and if the CCB identifies the need for a separate working group instead of taking ownership of the item itself, to further analyze the issue and come up with a recommendation for the CCB, then the CCB recommends the formation of a working group which may involve interested parties and experts on the topic who may not be members of the CCB. However, at least one member of the CCB is to be part of the working group to provide updates and recommendations on both sides.

Once the final report in the form of a rejection or solution proposal or another recommendation is made regarding the specific topic for which a working group was constituted, it ceases to exist.

4 Team operations, functioning

This section describes the guidelines regarding the team operations and regular functioning of the SIGs and the CCB.

4.1 Tasks and action items

The SIGs, CCB meet regularly (weekly or bi-weekly) to discuss a list of issues that are tracked. This list includes change requests that are proposed (by partners, the community or interested parties) issues reported, feature requests and their corresponding solution proposals once accepted. This aligns with the overall goal of maintaining the Mojaloop API Specification.

The SIGs, CCB review proposals or change requests made or any clarifications requested and adds them to the tracking list, which are then discussed until they are resolved or deferred.

When a new feature or extension is identified, a sub-group or a separate stream that is affiliated with the CCB takes up the responsibility of making the change request (if not made earlier) and then a solution proposal (if the change request is accepted), which will be submitted to the corresponding SIG or CCB as well, for review and eventual adoption as needed.

4.2 Meetings, communication

The details of specific CCB, SIG team meetings such as frequency, timing and conferencing tools are managed by the convenors of the respective teams.

SIG, CCB related GitHub issues are tracked on Mojaloop GitHub repository (mojaloop-specification). Labels are used to identify which group they belong to and whether they are bugs, change requests, tracking issues or solution proposals.

Several group members also meet at related events such as Mojaloop Community events hosted by the Mojaloop Foundation and other specific events hosted by the BMGF or other interested parties.

4.2.1 Attendance

It is recommended that the SIG, CCB team members participate in at least 70% of the periodic virtual meetings and at least once a year in the onsite meetings.

Abstention will not be a mode of voting which may be seen as passive approval or disapproval of an issue. Items would only be approved or rejected by the team members present during a specific meeting during which voting is done. It will be announced in advance whenever an item is planned to be voted on. Voting using e-mail or GitHub comment is allowed in case a member cannot participate in a specific SIG/CCB meeting.

Considering the SIGs, CCB run on a weekly/bi-weekly schedule currently, if a CCB member misses four consecutive meetings without an alternate member joining from the same entity, a recommendation is made to the partner company to replace the existing members, both primary and alternate.

4.3 Decision making guidelines

As change requests, clarification requests, feature requests and issues are reported to the SIGs/CCB, these are discussed via GitHub Issues~~,~~ meetings as required until arguments are made from various perspectives.

As issues are reported, they are added to the issues and decision tracking list and are discussed at each of the SIG/CCB's scheduled meetings. If, the group agrees to make changes or fix issues with the specification, the changes are discussed and finalized after reviews. If, however, the group agrees that a specific issue is better taken-up at a later time, it can be deferred to a specified later date.

Regarding a change request or an issue, if there is no consensus among the members, the issue is held in status quo for a period of two weeks or a period decided by the team during which the discussions may continue, after which the decision goes for voting. See sections 4.4 and 4.5 regarding more details about this process.

4.4 Conflict resolution

If there is no consensus regarding any aspect of the SIG or CCB, whether it be operating procedures or voting mechanism or charter, the issue is to be brought up to the group either via emails or during a meeting.

The onus is on the convenor to follow-up the issue and call for additional meetings as necessary or take other steps to setup a discussion regarding the issue. Regarding matters not pertaining to the Specification but to operations or such, the voting process specified in section 4.5 can be leveraged, but to approve a decision or make a change related to operations (or Charter, etc.) more than 75% votes are needed for changes to be made, otherwise the *status quo* prevails.

4.5 Voting process

The aim of the CCB, like that of the other Mojaloop institutions, is to proceed by consensus. In the event that a unanimous decision on an issue cannot be reached, the issue will be put to a vote. For the CCB, those entitled to vote will be:

1. The convenors of SIGs, other than the convenor of the CCB itself.
2. The Mojaloop Foundation representative
3. The Mojaloop TGB representative

For SIGs, all the voting members of the SIG can vote on the issue including the SIG's convenor.

Before a vote is called for, a certain period ranging between a day to two weeks is allowed for further discussions after which the convenor sets a date for the voting process to happen.

For decisions related to the Specification, a simple majority is needed to approve the decision.

Regarding voting rights, each of the official SIG/CCB member gets a single vote, which can be cast by the primary member of the SIG/CCB from that partner entity. In case of the primary member's absence, the alternate member can cast the vote.

Voting against a particular proposal can be made either by the official members who are part of the conversation or by providing a reasoning accompanied with a "No" vote. A vote can be made by a member of the SIG/CCB over a channel such as e-mail if the person cannot participate in the meeting during which the voting happens.

When a change request is voted down or rejected, a detailed explanation that includes the reasoning and discussions regarding the topic is sent to the proposer.

4.6 Changes to the APIs, versioning, tooling

Once a decision is made on a change request to accept it and make changes to the Specification, a versioning process is to be followed as described in the below sections.

4.6.1 Adopting changes

The members of the SIGs and CCB are encouraged to adopt the changes made to the specification based on the market and Scheme needs. The members are free to support legacy standard as per their own guidelines but should be willing to accommodate partners who follow the new releases after the changes are accepted and published. However, the SIG/CCB itself doesn't prescribe a timeframe regarding adoption.

The members also agree not to develop parallel standards that are derivates of the Mojaloop family of APIs by forking the specifications and not bringing in the changes to the SIGs/CCB and keeping them separate.

4.6.2 Versioning

A running candidate version with approved changes is maintained as a current version, which is to be released when the SIG/CCB identifies the changes to be significant for a release.

The versioning includes a major, minor and patch version number. For example, Version 1.0.0 indicates ‘1’ being the major version ‘0’ being the minor version and ‘0’ being the patch version. For a given major version all the minor versions must be backwards compatible. For example, version 1.2.x should be backwards compatible with version 1.1.x, version 1.1.x should be compatible with 1.0.x and so on. A minor version change is to be used for new feature additions without any breaking changes. A ‘patch’ version increment can be used for minor changes or fixes such as changes in description or addition of new optional elements, which are fully compatible with the parent ‘minor version’. For example, version 1.1.1 can be used for the version where optional HTTP headers are added which is fully compatible with v1.1.0.

However, if there is a breaking change that is approved, that will be released as a major version with relevant release notes and documentation.

These releases and release notes are broadcast not just to the CCB and SIG teams but to all the stakeholders and the Mojaloop open-source community. The release mechanism and frequency depend on the changes made and, on the decisions made by the SIG/CCB regarding when to make a release.

4.6.3 Mojaloop Testing toolkit

This is an optional technology that can be used by the Mojaloop CCB and the SIGs. The Mojaloop Testing toolkit (TTK) is available for interested parties for self-testing and assessing the state of a deployment or a PISP or a Financial Service Provider. Moreover, the understanding is that the adopters or participants in a particular Scheme are recommended or bound by a Scheme rule to be evaluated using the Testing toolkit to ensure they conform to the **Mojaloop API Specifications**.

The goal is for this kit to be able to run against a deployment and then either provide results directly to an evaluator or which can be downloaded and submitted later. It would be possible for a trading partner or a Hub Operator to run this against a participant as well.

The objective of this test suite is to provide an indication of success or failure indicating the conformance or non-conformance of a particular FSP deployment (or codebase).

The automated testing framework should use an Open API 3.0 (or greater version) Swagger specification, several configuration details and then tests a given deployment (or codebase) and provides results. This makes maintenance of the testing tool-kit manageable and wouldn’t necessitate changes every time a change is made to the Specification.

There may be levels of testing such as a “self-testing” followed by a “test verification” using a third-party and then a “Scheme Level Testing” or “Certification” done by a trading partner or a Hub Operator.

5 Submitting Change Requests

The below sections address how new feature requests, changes and bugs (against APIs or specification) can be raised and the processes involved in submitting Change Requests and Solution Proposals to the CCB or SIGs.

5.1 Change Request Overview

A change request can be drafted using the Change Request template and submitted to the SIG or CCB as a GitHub issue or the issue can be proposed during a CCB/SIG meeting and then be documented.

The document mentioned above can be submitted by either a member of one of the SIGs or CCB or a community member. This person then will be the owner of the change request until it reaches a logical conclusion (accepted or deferred or rejected).

If members were to make changes for a customer, the recommendation is to propose the changes to the CCB for the integrity of the Specification and to maintain interoperability with other adopters of the **Mojaloop** family of **APIs** and the Mojaloop standard (subject to security and privacy concerns).

When a change request is made, the onus is on the owner of the issue to provide explanation, examples, demonstrations or such and make changes, as requested by the CCB/SIG team members to take the change request process further.

5.2 Solution Proposal Overview

Once a change request is accepted, it moves into a Solution Proposal phase. A solution proposal can be drafted using the Solution Proposal template and submitted to the SIG or CCB as a GitHub Issue in conjunction (where applicable) with a Pull Request that consists of the changes needed.

When a solution proposal is made, the onus is on the owner of the issue to provide explanation, examples, demonstrations or such and make changes, as requested by the CCB/SIG team members to take the solution proposal process further.

Once a solution proposal is approved, it is marked to be released with the next version of the API. (If the change is a non-breaking change, it can be published with just a minor version change, whereas if it includes a breaking change, a major version change is needed).

6 Initial adoption and membership

The version v1.0 of the CCB charter has been adopted by the Mojaloop CCB on the **22nd of September 2021**. Outlined below is the structure adopted for the SIGs and the CCB as of September 2021.

6.1.1 Special Interest Groups

The following table shows the initial set of Special Interest Groups that have been defined as of June 2021. Even though the individuals change, the structure should be maintained subject to the preference of the CCB and the SIGs.

Name	Focus area	Convenor
FSP Interoperability API	Core Mojaloop functionality: maintenance and querying of ALS, initiation and completion of funds transfers	**Henrik Karlsson**, Ericsson
PISP API	Functionality relating to PISPs and to the DFSPs that interact with them: account linking and transfer initiation	**Lewis Daly**, Crosslake
Settlement API	Functionality to support the definition and execution of settlements in Mojaloop switches and the recording of liquidity cover provisions.	The CCB & Administraton SIG agreed to have this API under the Administration SIG.
Cross-network API	Functionality to support routing of messages between Mojaloop networks and other networks, whether Mojaloop or non-Mojaloop. Support for FX conversion	The CCB & FSPIOP SIG have agreed to have this API under the FSPIOP SIG.
Administration API	Functionality to support the creation and maintenance of a functioning Mojaloop system: creating, modifying and suspending participants	**Miguel de Barros**, ModusBox

6.1.2 Change Control Board

The following table shows the composition of the Change Control Board as of June 2021.

Name	Function
Miller Abel	Observer
Matt Bohan	Observer
Simeon Oriko	Mojaloop Foundation Community Manager, Observer
Lesley-Ann Vaughan	Mojaloop Foundation - Product Strategy, Observer
Michael Richards	Rapporteur
Sam Kummary	Convenor, CCB
Henrik Karlsson	Convenor, FSP Interoperability API SIG
Lewis Daly	Convenor, Third-party (PISP) API SIG
Miguel de Barros	Convenor, Administration API SIG

7 Amendments

The following table outlines the changes made to the CCB charter.

Date	Change
2021-**Sep-22**	**Initial version** adopted by Mojaloop CCB and SIGs