Mojaloop 3PPI Enablement PI-18

3PPI team

Mojaloop PI-18 community event

Agenda

1. PI-17 overview for 3PPI
2. 3PPI features current status
3. API snippets update
4. 3PPI roadmap

PI-17 overview for 3PPI

PI-17 Contributors

- Chetan Manjeshwar
- Cian Brassil
- Georgi Logodazhki
- JJ Geewax
- Kevin Leyow
- Kuan Yen Heng
- Lewis Daly
- Michael Richards
- Miguel de Barros
- Nizam Anuar
- Sam Kummary
- Vijay Guthi

Background: 3PPI phase-1 outcomes

1. Designs and scaffolding for 3PPI flows, demonstrable with mocked services
2. Define characteristics of 3PPIs and FSPs that support 3p functionality and modify the core mojaloop system (APIs first) to support these definitions
3. Draft Open API definition **v0.1** to support Account Linking, Third-party transfers (including sub functions for authentication such as GET /authorizations), subject to the approval of the Mojaloop CCB.
4. Happy path PoC demonstration for 3p linking and transfers
5. Create mocked services including those needed for account linking, transfers with CI/CD pipelines setup
6. Modify the Mojaloop SDK to support extensions for 3p functionality
7. A working Mojaloop lab environment capable of demonstrating a third-party transfer between two DFSPs initiated by a PISP

Background: 3PPI phase-2 outcomes

1. Full coverage of error scenarios in third-party focused flows (linking, transfers)
2. Updating the 3P services and demos to match the 3p API draft **v0.1** and use real FIDO credentials
3. Introducing a "CONSENT" type to the Account
4. Lookup Service for determining which PISP / AuthService is responsible for a given Consent resource
5. Updating the third-party scheme adapter to function for both FSPs and 3PPIs
6. Building integrations between SDK scheme adapter and third-party scheme adapter code bases
7. Integrating third-party scheme adapter code to verify signatures using the Auth Service API methods
8. Creating integration, end-to-end tests for the 3p flows - account linking and transfers

Useful links - 3PPI

1. Mojaloop sandbox/demo for exploring 3p functionality: https://sandbox.mojaloop.io/
2. 3PPI demo-1 (15:50): https://www.youtube.com/watch?v=RvLoP4Tj8q8
3. 3PPI overview: https://github.com/mojaloop/pisp-project
4. Third-party API spec: https://github.com/mojaloop/mojaloop-specification/tree/master/thirdparty-api
5. API snippets (Mojaloop APIs): https://github.com/mojaloop/api-snippets
6. PISP project issues: https://github.com/mojaloop/project/issues#workspaces/mojaloop-project-59edee71d1407922110cf083/board?labels=oss-pisp&repos=116650553

3PPI Goals phase-3 [PI-17, 18]

Confidence Vote: 3.5

Goal	**Mojaloop 3PPI services adopted by 1 3PPI implementer & 1 FSP (in testing / QA)**	
Key Epics Objectives	1. Productionize and Profile Third Party Components. IaC support for 3PPI services **3.** Google Standard Payment API Adapter [Open Source] developed and released	**2.** Write a DFSP Integration Guide (similar to https://sandbox.mojaloop.io/guides/overlay/g2p-3ppi-account-linking.html); **4.** Payment Manager support for DFSPs supporting PISPs **5. Bonus**: 1) 3P transaction request with ISO 20022. 2) Documentation: starter kits, extend use case support building on base functionality. 3) Documentation extensions: Business processes, list of responsibilities.
Not Doing now but important next & Risk / Issues	**Future Roadmap:** 1. Spec changes to the GSP Adapter (will be retrofitted).	**Risks / Issues:** 1. Dependency on Payment Manager, IaC (less on on IaC but PM) 2. Community contributors and resourcing 3. Dependency on 3PPI API changes to support an example 3PPI implementer 4. To get this deployed live, we are dependent on implementers and DFSPs (their internal processes)
Success Defined How?	Stretch : Is there 1 implementer who has deployed the 3PPI Features, with 1 3PPI and 1 DFSP in test? If so, then we have been successful.	Minimum : End-to-end testing of 3PPI features with support for FSP, PISP instances, 3P services on the Hub and PISP services along with GSP adapter

3PPI features current status

ML 3PPI features current status

1. 3PPI API v1.0 published ✅
2. 3PPI services initial implementation & end-to-end tests ✅
3. 3PPI services integrated with Mojaloop releases (v13.1.0) ✅
4. Productionize and profile 3PPI services ✅
5. Design for Google standard Payments (GSP) adapter ✅
6. GSP API adapter implementation ⚙️

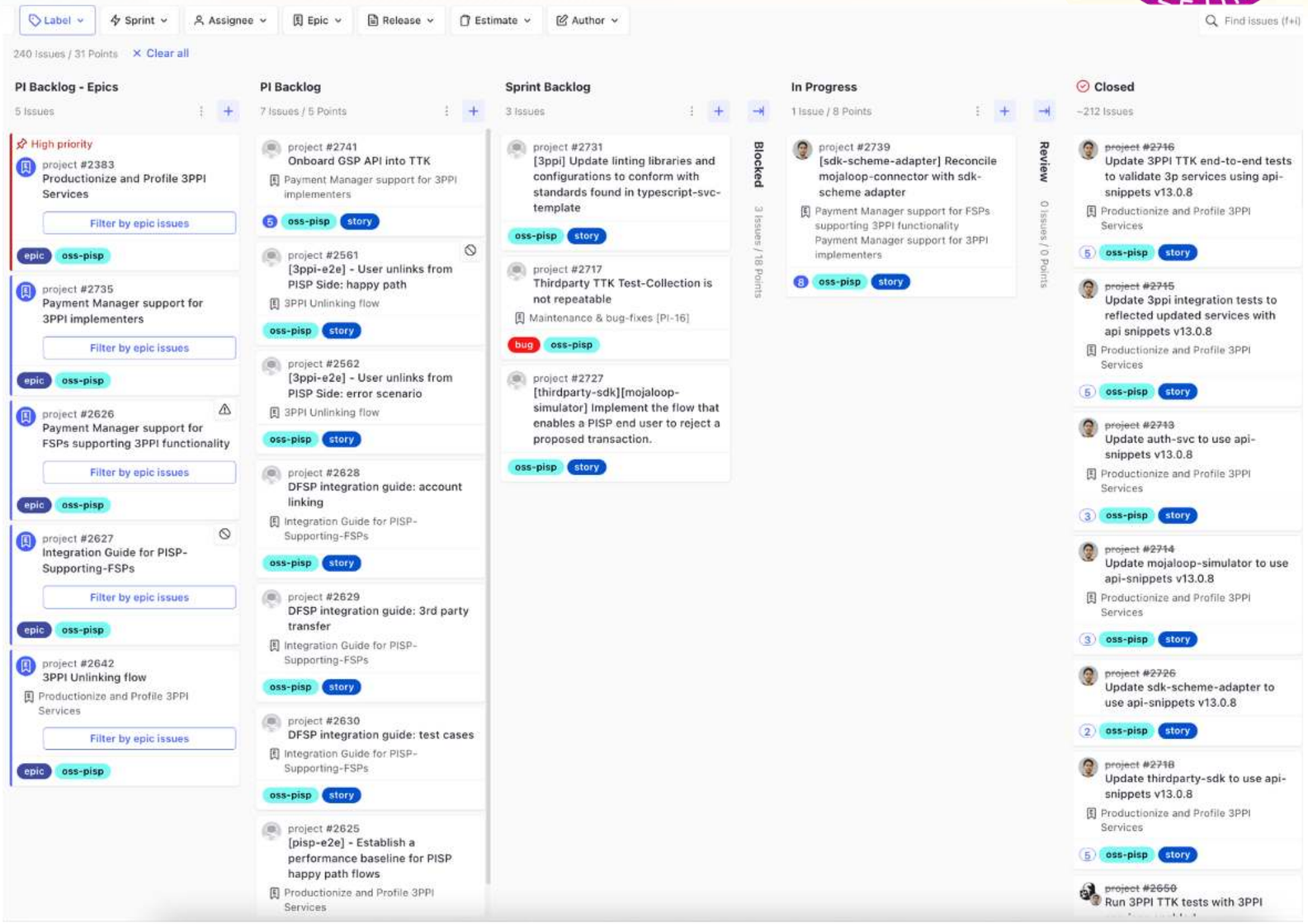
3PPI PI-17 overview

1. 3PPI phase-3 Goals
   a. Productionize Mojaloop 3PPI services
      i. Validate 3PPI helm charts, deployments ✅
      ii. Validate 3PPI services TTK tests and run against deployed 3PPI services ✅
      iii. Ensure API, documentation, implementation are in sync (3PPI API v1.0) ✅
   b. Develop GSP adapter for 3PPI implementers ⚙️
      i. Change request “refactoring 3PPI interface” approved ✅
      ii. Solution proposal in review ⚙️
      iii. Technical debt on Payment Manager, sdk-scheme-adapter for 3P functionality ✅
      iv. Core connector for 3P implementers
      v. Core connector for FSP implementers (Generic version)
   c. Usage guides, scheme templates for 3PPI usage ⚙️

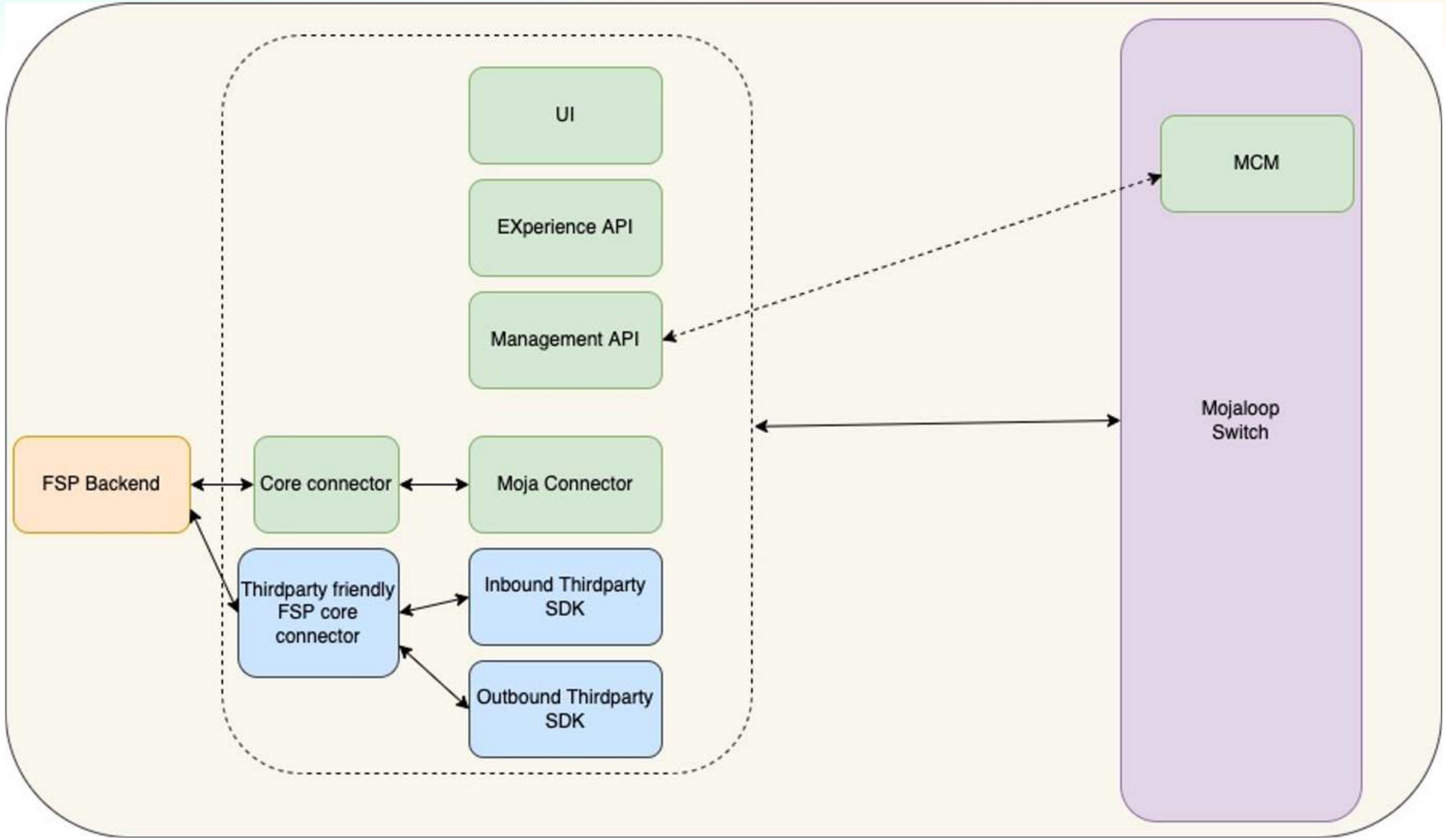
3p end-to-end tests video

Workstream Progress

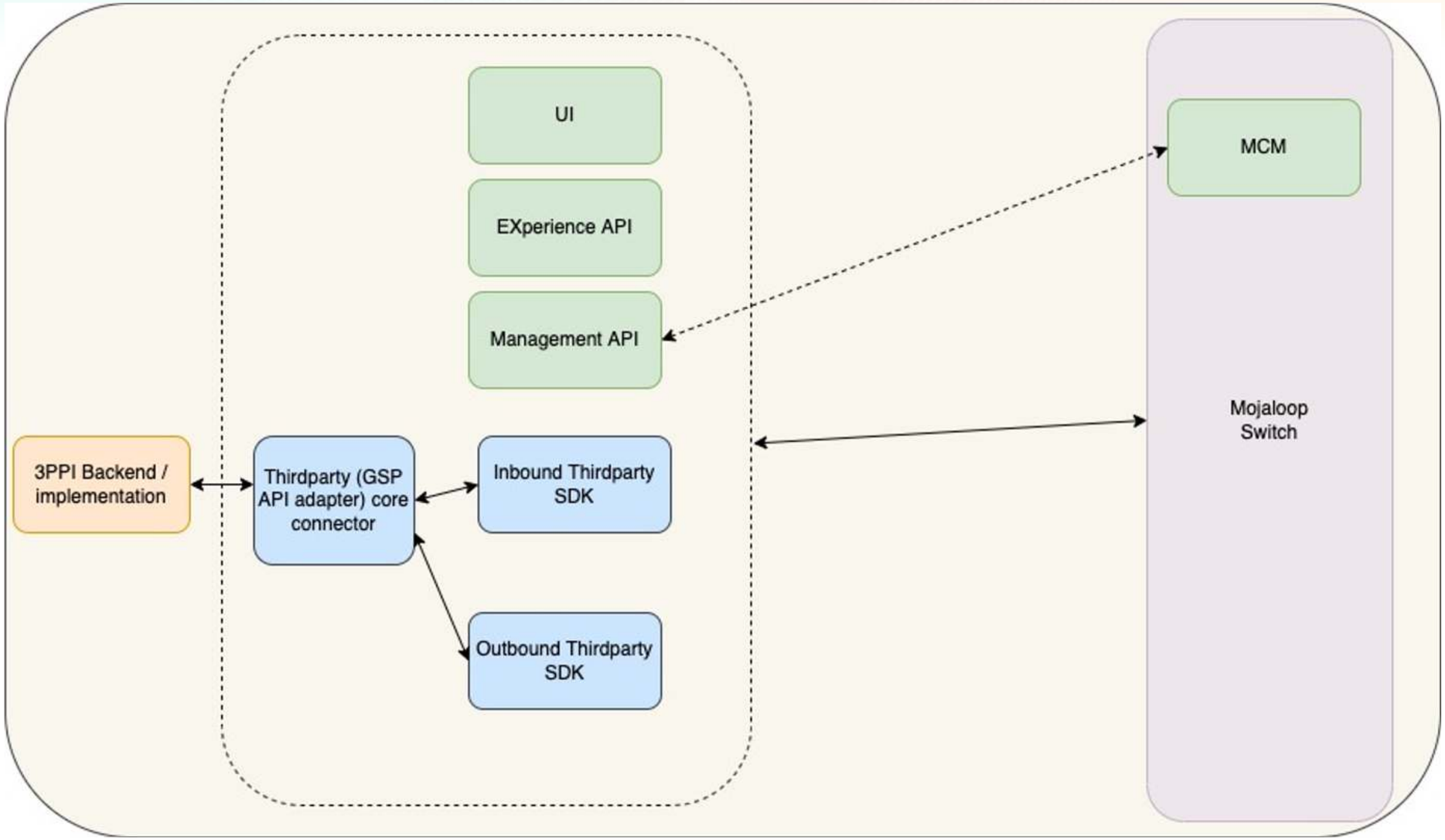
- Bugs fixed
- API discrepancies addressed
- 3PPI charts validated
- 3PPI TTK end-to-end tests validated
- Prerequisites for 3PPI adapter development addressed
- Development of adapters for 3p functionality in progress



3PPI flows: FSP - PM - Mojaloop



3PPI flows: 3PPI - PM - Mojaloop



API snippets update

What are API Snippets?

Reusable yaml definition code snippets & autogenerated Typescript interfaces

1. it is the set of basic Mojaloop Data Transfer Object Interfaces - YAML & Typescript
2. one source of truth - common type system
3. compact template definitions files for microservices
4. tooling for Typescript interfaces generation
5. Swagger UI online browser: https://mojaloop.github.io/api-snippets/

The Old Way

For example, if we wanted to add a new Currency code

1. Navigate to an existing specification: https://github.com/mojaloop/mojaloop-specification

```
2075        pattern: ^[0-9a-f]{8}-[0-9a-f]{4}-[1-5][0-9a-f]{3}-[89ab][0-9a-f]{3}-[0-9a-f]{12}$
2076        description: Identifier that correlates all messages of the same sequence. The API data type UUID (Universally Unique
2077      Currency:
2078        title: Currency
2079        description: The currency codes defined in [ISO 4217](https://www.iso.org/iso-4217-currency-codes.html) as three-lette
2080        type: string
2081        minLength: 3
2082        maxLength: 3
2083        enum:
2084          - AED
2085          - AFN
2086          - ALL
```

1. Open a pull request and add a new Currency Code
2. Copy and paste in every service where we refer to Currency… at least 10 times. Hope you don't make any mistakes in 1000's of lines of Yaml specification
3. Code review 10+ times

The New Way

For example, if we wanted to add a new Currency code

1. Navigate to the Currency.yaml definition in api-snippets

```yaml
title: Currency
description: >-
  The currency codes defined in [ISO
  4217](https://www.iso.org/iso-4217-currency-codes.html) as three-letter
  alphabetic codes are used as the standard naming representation for
  currencies.
type: string
minLength: 3
maxLength: 3
enum:
  - AED
```

1. Pull Request + Code Review
2. A new version is built automatically (if we wish)
3. For each service, update the package.json to the latest api-snippets:

1. `npm install`

```
"@mojaloop/api-snippets": "^13.0.9",
"@mojaloop/central-services-error-handling": "11.3.0"
```

3PPI roadmap

PI-18 3PPI Goals [To be confirmed during roadmap planning]

Confidence Vote: 3.5

Goal	**Mojaloop 3PPI services adopted by 1 3PPI implementer & 1 FSP (in testing / QA)**	
Key Epics Objectives	1. Google Standard Payment API Adapter [Open Source] developed and released (PM included in 3p flows) **3.** Payment Manager support (3p friendly core connector) for DFSPs supporting 3PPIs	**2.** Write a DFSP Integration Guide (similar to https://sandbox.mojaloop.io/guides/overlay/g2p-3ppi-account-linking.html); **5. Bonus**: 1) 3P transaction request with ISO 20022. 2) Documentation: starter kits, extend use case support building on base functionality. 3) Documentation extensions: Business processes, list of responsibilities.
Not Doing now but important next & Risk / Issues	**Future Roadmap:** 1. Spec changes to the GSP Adapter (will be retrofitted).	**Risks / Issues:** 1. Dependency on Payment Manager, IaC (less on on IaC but PM) 2. Dependency on 3PPI API changes to support an example 3PPI implementer 3. To get this deployed live, we are dependent on implementers and DFSPs (their internal processes)
Success Defined How?	Stretch : Is there 1 implementer who has deployed the 3PPI Features, with 1 3PPI and 1 DFSP in test? If so, then we have been successful.	Minimum : End-to-end testing of 3PPI features with support for FSP, PISP instances, 3P services on the Hub and PISP services along with GSP adapter

Thank you!

Mojaloop sandbox for exploring 3p functionality: https://sandbox.mojaloop.io/

3PPI demo-1 (15:50 mark): https://www.youtube.com/watch?v=RvLoP4Tj8q8
ML 3PPI demo-2 (34:09 mark): https://www.youtube.com/watch?v=SGBCm4WDBsE&t=10s