MODUSBOX

Contributions from ModusBox to support the Community in DFSP Onboarding

mojaloop

MODUSBOX

ModusBox have been working with partners on the first implementations of Mojaloop systems.

This experience has thrown up a host of new insights into the practical difficulties of setting up a Mojaloop hub and onboarding DFSPs to a scheme.

As a consequence of these difficulties, ModusBox has been working on ways of easing, in general, the practical tasks of connecting many DFSPs to Mojaloop schemes.

MODUSBOX

Support for onboarding:

1. Standard Components
2. An example Scheme Adapter
3. A system to manage certificates and keys

MODUSBOX

Support for onboarding:

1. Standard Components

mojaloop

What problems are the Standard Components solving?

During commercial Mojaloop implementations...

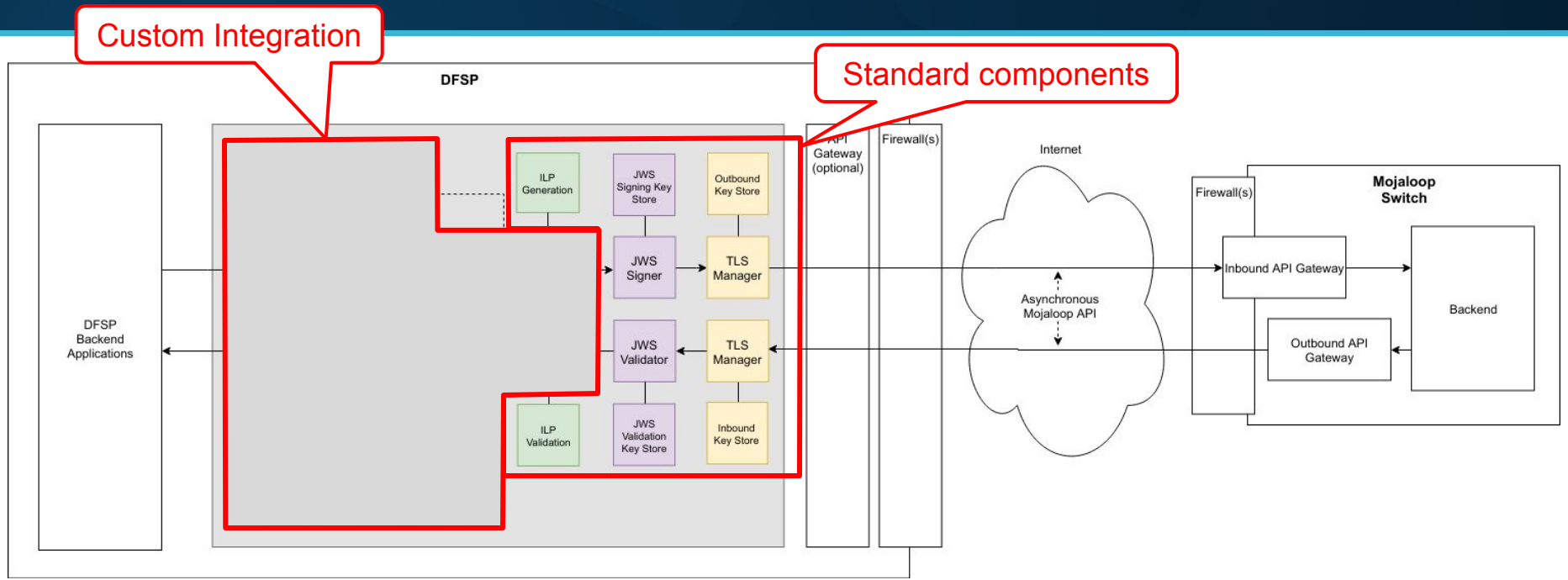
1. We encountered differing interpretations of some aspects of the Mojaloop API specification
    a. E.g. those relating to securing messages, leading to incompatibility between participants
2. These mismatches were only discovered when a participant integrated with the scheme
3. Errors discovered while establishing mojaloop compliant TLS, ILP and JWS led to considerable rework

These project pains led to extended timelines, raised costs and commercial risk for both switch operators and DFSPs.

How do the standard components help?

1. They implement complex operations needed by all participants
    - Real-world implementations
    - Comprehensively tested
2. Specification compliant security implementations out-of-the-box
    - Bidirectional, mutual x.509 authentication
    - Mojaloop spec compliant JWS
    - Interledger protocol packet signing and validation
3. Specification compliant HTTP headers
    - Mojaloop spec compliant headers and header processing out-of-the-box

Standard Component Architecture



mojaloop

MODUSBOX

Support for onboarding:

1. Standard Components
2. An example Scheme Adapter

What problem is the Scheme Adapter solving?

During commercial Mojaloop implementations we observed:

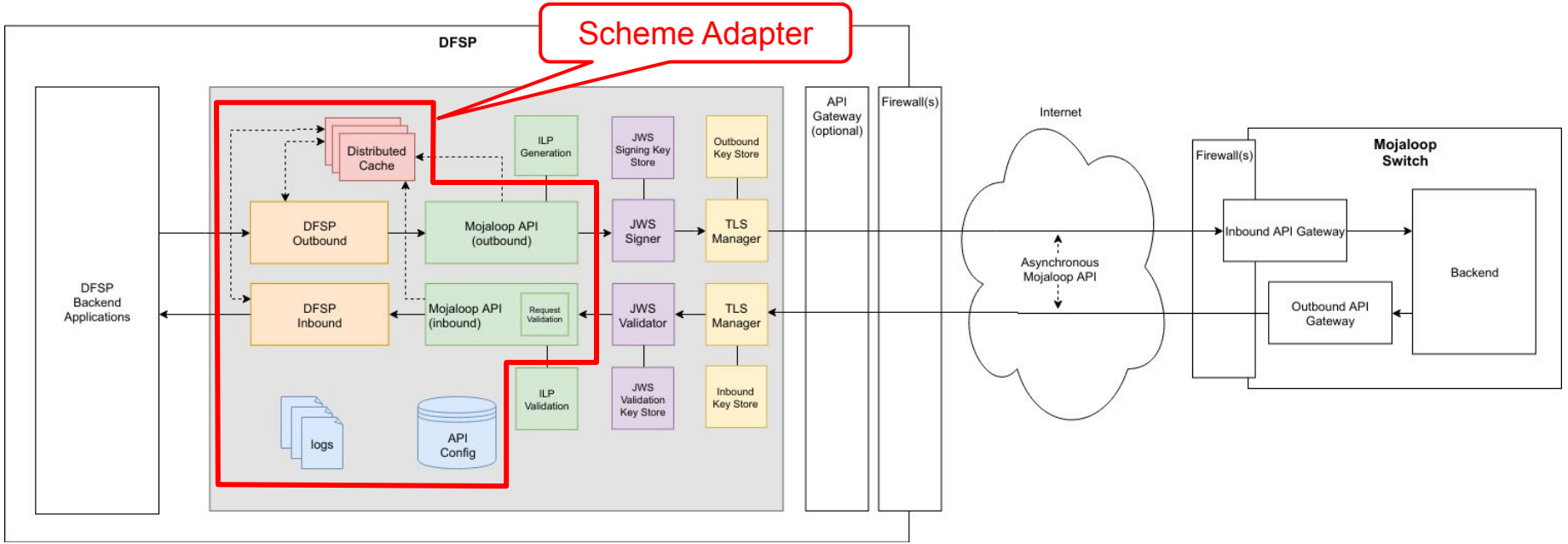
1. Multiple participants platforms are incompatible with native mojaloop API interface requirements.
2. Many problems onboarding participant platforms were discovered late in the integration cycle

These project pains led to extended timelines, raised costs and commercial risk for both switch operators and DFSPs.

How does the Scheme Adapter help?

1. Manages the complexities of interfacing using the Open API specification
2. Implements a configuration-based approach for defining scheme-specific ways of working
3. Uses standard components to reliably and resiliently perform complex operations
4. It makes it easier for DFSPs to encode the scheme-specific business rules by...
    - Aligning configuration options with decision points in business rules
    - Approaching direct representation of scheme operating guidelines

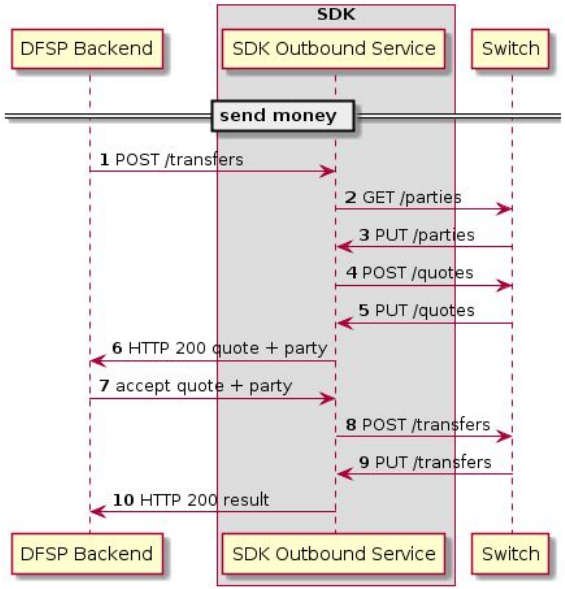
Scheme Adapter Architecture



mojaloop

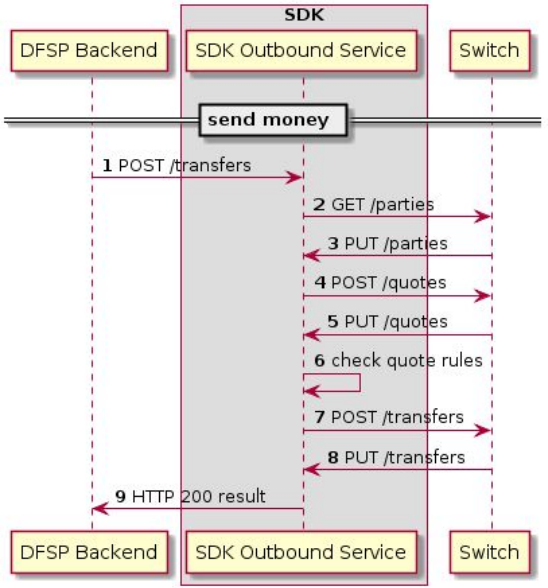
Synchronous Internal API Example: Sending funds...

Two stage



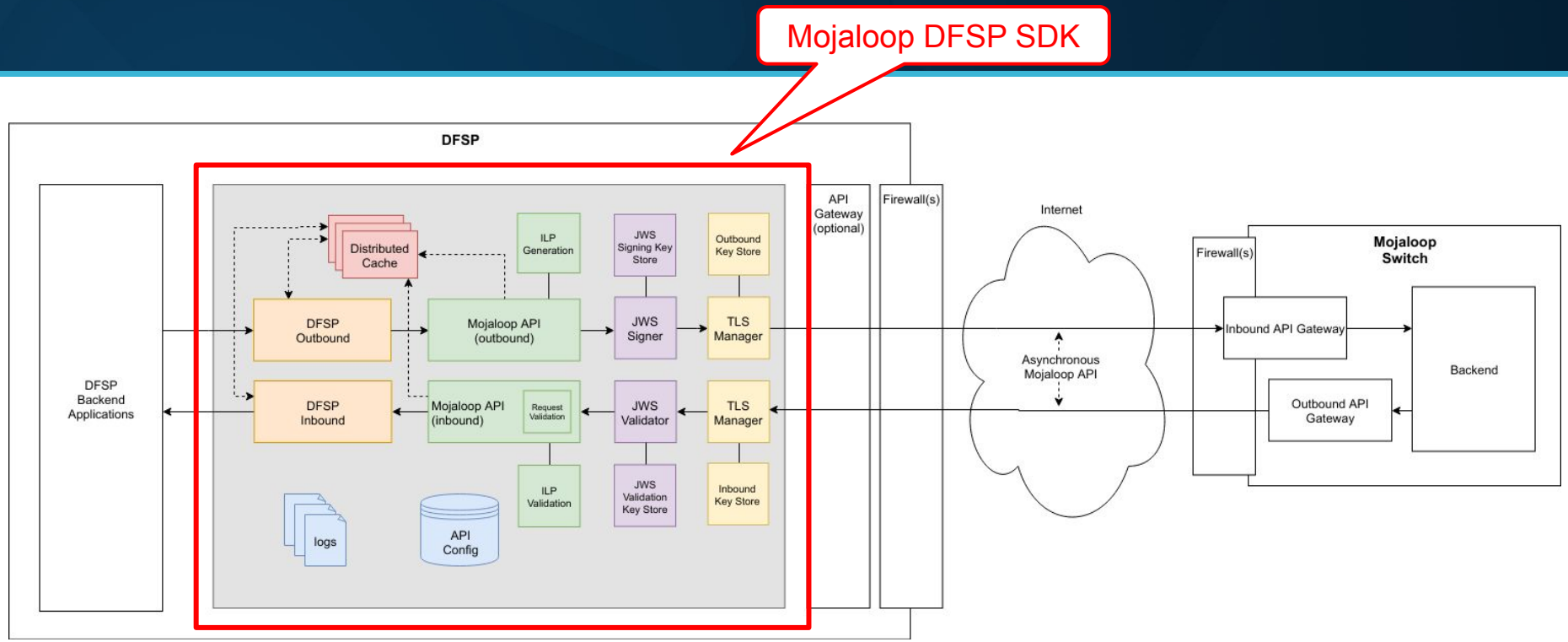
or...

single stage



mojaloop

Mojaloop DFSP SDK



mojaloop

MODUSBOX

Support for onboarding:

1. Standard Components
2. An example Scheme Adapter
3. A system to manage certificates and keys

Mojaloop PKI Admin Server

A Service that greatly reduces the overhead in sharing information, removing many manual errors in the creation, sharing and signing of signatures as well as facilitating the ongoing maintenance as signatures expire

What problem is this trying to solve?

During commercial Mojaloop implementations we observed:

1. Multiple requests for change of IP address whitelists - without an easy to follow audit trail
2. Multiple mistakes in the creation, signing and exchange of TLS certificates due to misinterpretation of configuration settings and manual processes
3. No method to easily distribute JWS certificates for DFSPs

These are project pains that lead to extended timelines, high cost and commercial risk for both switch operators and DFSPs.

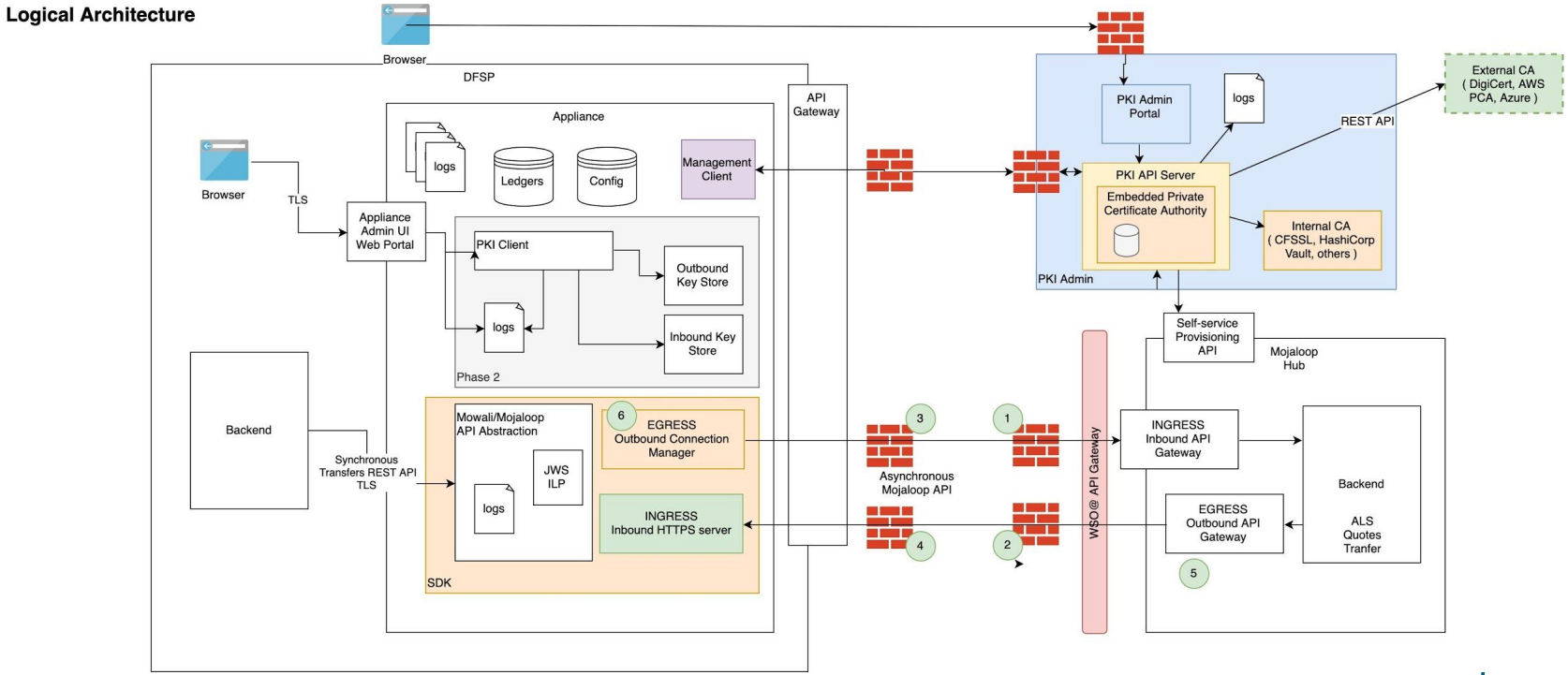
How does the PKI Admin Server help?

1. It greatly reduces the overhead in sharing information.
2. It automates the creation, sharing and signing of signatures, thereby removing multiple opportunities for error in manual processes.
3. It facilitates the ongoing maintenance of signatures by ensuring that best-practice expiry techniques are used, and that the renewal of expired signatures is managed without the need for manual intervention.

How does the PKI Admin Server help (continued)?

1. Reduces workflow requests
    - Copy and Paste of Key Data
    - Workflow, and feedback to all Partners of where requests are in the process
2. Audit Trail
    - Requests and activity logged and auditable
    - can be linked to Fraud and AML platform for Key Event tracking
3. Standardisation of Certificate Creation
    - Key elements configurable - to reduce entry error
    - Environment identified - to reduce chance of incorrect allocation
    - It could integrate with some external CA to create the certificates
4. Automation of JWS Certificate sharing and Testing
    - Process to distribute JWS Certificates from all DFSPs
    - Option to test working transfers with SDK

Long Term Architecture



mojaloop

Full API

TSP / PKI Admin

TSP / PKI Admin

Contact the developer

dfsp-inbound : DFSP Inbound PKI	Show/Hide	List Operations	Expand Operations
dfsp-network-config : DFSP Ingress and Egress endpoint configuration	Show/Hide	List Operations	Expand Operations
dfsp-outbound : DFSP Outbound PKI	Show/Hide	List Operations	Expand Operations
dfsp-pki : DFSP PKI certificates and CA	Show/Hide	List Operations	Expand Operations
hub-network-config : Hub Ingress and Egress endpoint configuration	Show/Hide	List Operations	Expand Operations
pki : Hub PKI Infrastructure setup	Show/Hide	List Operations	Expand Operations

Full API

dfsp-inbound : DFSP Inbound PKI Operations

TSP / PKI Admin

TSP / PKI Admin

Contact the developer

dfsp-inbound : DFSP Inbound PKI

Show/Hide | List Operations | Expand Operations

GET	/environments/{envId}/dfsps/{dfspId}/enrollments/inbound	Get a list of DFSP Inbound enrollments
POST	/environments/{envId}/dfsps/{dfspId}/enrollments/inbound	Create DFSP Inbound enrollment
GET	/environments/{envId}/dfsps/{dfspId}/enrollments/inbound/{enId}	Get a DFSP Inbound enrollment
POST	/environments/{envId}/dfsps/{dfspId}/enrollments/inbound/{enId}/sign	Sign and add the certificate to the enrollment
POST	/environments/{envId}/dfsps/{dfspId}/enrollments/inbound/{enId}/certificate	Sets the certificate enrollment

Full API

Pki : Hub PKI Infrastructure setup Operations

pki : Hub PKI Infrastructure setup

Show/Hide | List Operations | Expand Operations

GET	/environments	Returns all the environments
POST	/environments	Creates an environment on the PKI Admin
DELETE	/environments/{envId}	Deletes an environment and its data
GET	/environments/{envId}	Find an environment by its id
POST	/environments/{envId}/cas	Creates a CA for the environment
GET	/environments/{envId}/ca/rootCert	Returns the CA root certificate
GET	/environments/{envId}/dfsps	Returns a list with all the DFSPs in the environment
POST	/environments/{envId}/dfsps	Creates an entry to store DFSP related info

Full API

Dfsp-network-config Operations

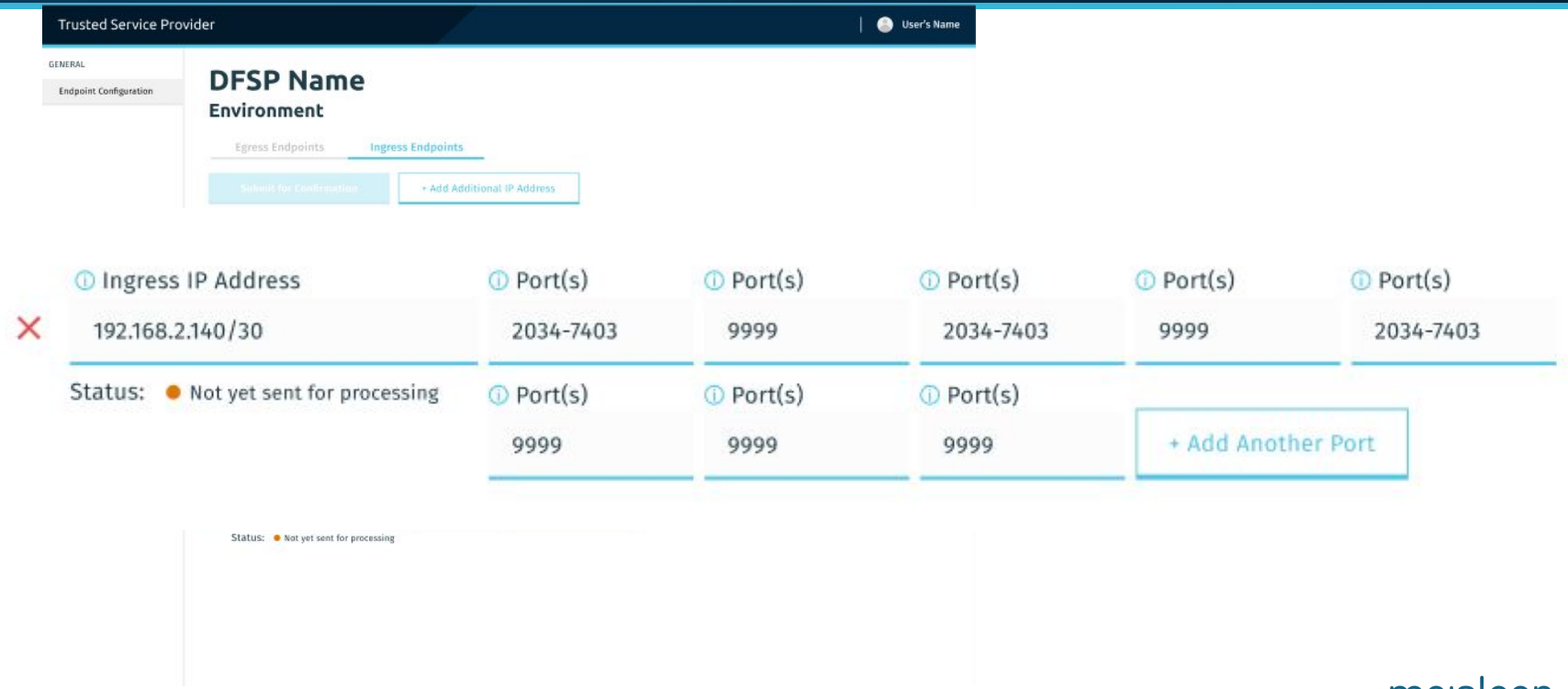
dfsp-network-config : DFSP - Ingress and Egress endpoint configuration

Show/Hide | List Operations | Expand Operations

GET	/environments/{envId}/dfsps/endpoints/unprocessed	Returns the unprocessed endpoint items
GET	/environments/{envId}/dfsps/{dfspId}/endpoints	Returns all DFSP endpoints
GET	/environments/{envId}/dfsps/{dfspId}/endpoints/unprocessed	Returns the unprocessed dfsp items
DELETE	/environments/{envId}/dfsps/{dfspId}/endpoints/{epId}	Delete an endpoint entry
GET	/environments/{envId}/dfsps/{dfspId}/endpoints/{epId}	Get an endpoint entry
PUT	/environments/{envId}/dfsps/{dfspId}/endpoints/{epId}	Update an endpoint entry
POST	/environments/{envId}/dfsps/{dfspId}/endpoints/{epId}/confirmation	Updates the endpoint as confirmed
GET	/environments/{envId}/dfsps/{dfspId}/endpoints/ingress/ips	Get the DFSP Ingress IPs
POST	/environments/{envId}/dfsps/{dfspId}/endpoints/ingress/ips	Adds a new IP entry to the DFSP Ingress endpoint
DELETE	/environments/{envId}/dfsps/{dfspId}/endpoints/ingress/ips/{epId}	Delete an endpoint entry
GET	/environments/{envId}/dfsps/{dfspId}/endpoints/ingress/ips/{epId}	Get an endpoint entry

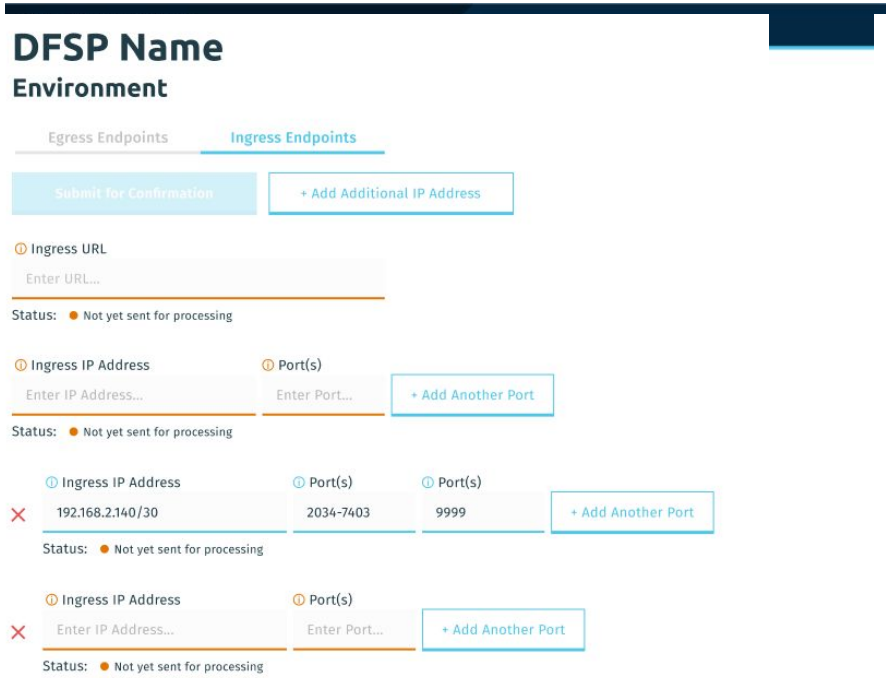
mojaloop

DFSP End Point Data Entry



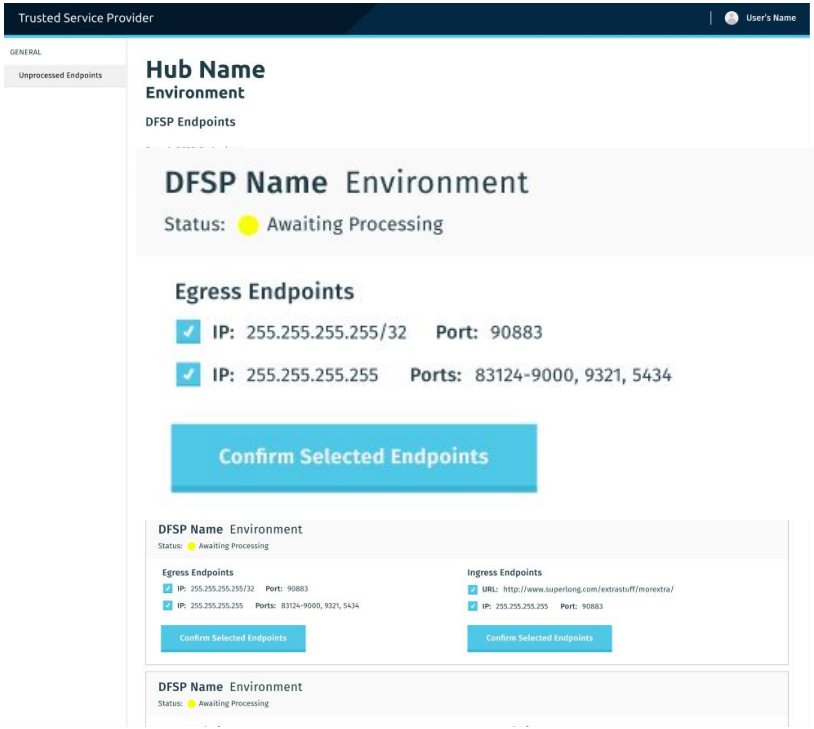
mojaloop

With End-Point Specific configuration options



mojaloop

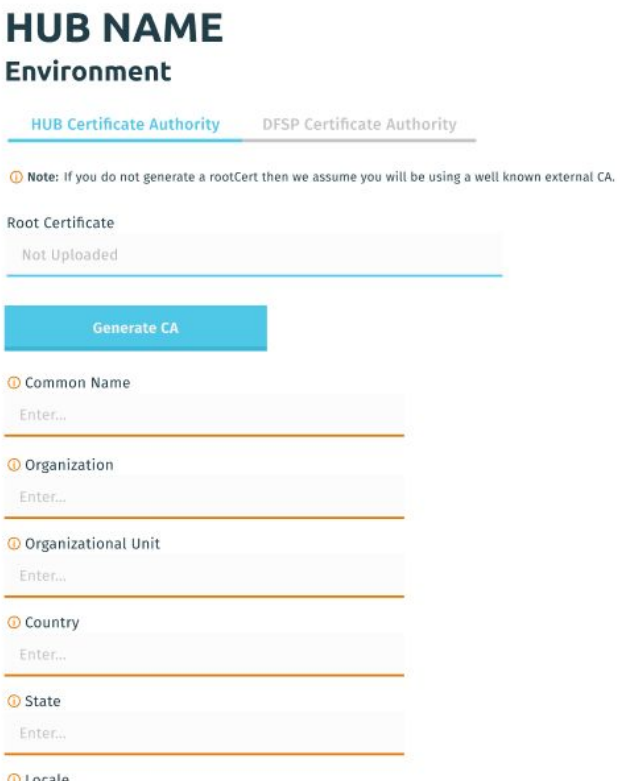
And clarity where the information is in the flow

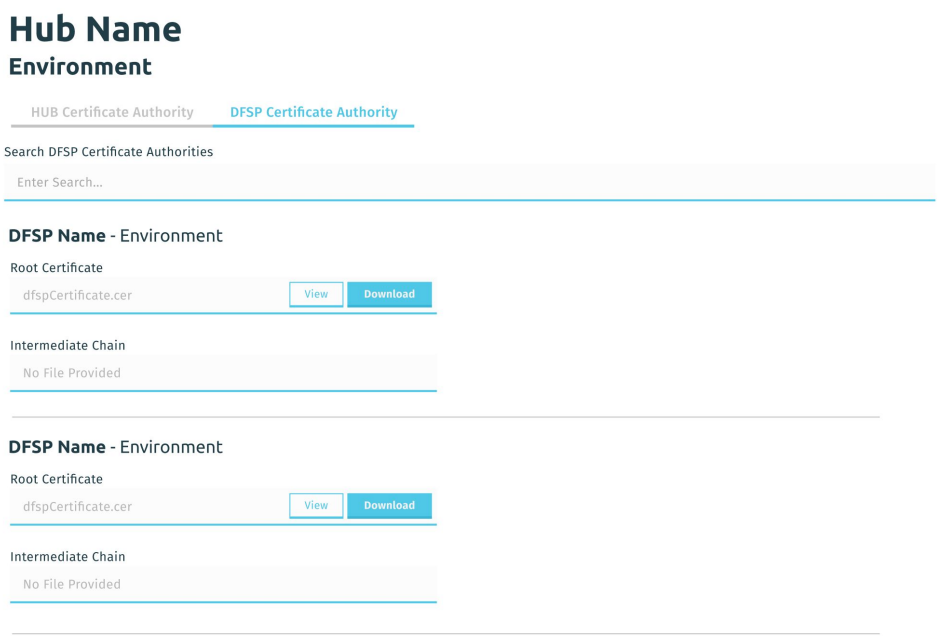


With an audit log to ensure clarity on what was done by whom and when

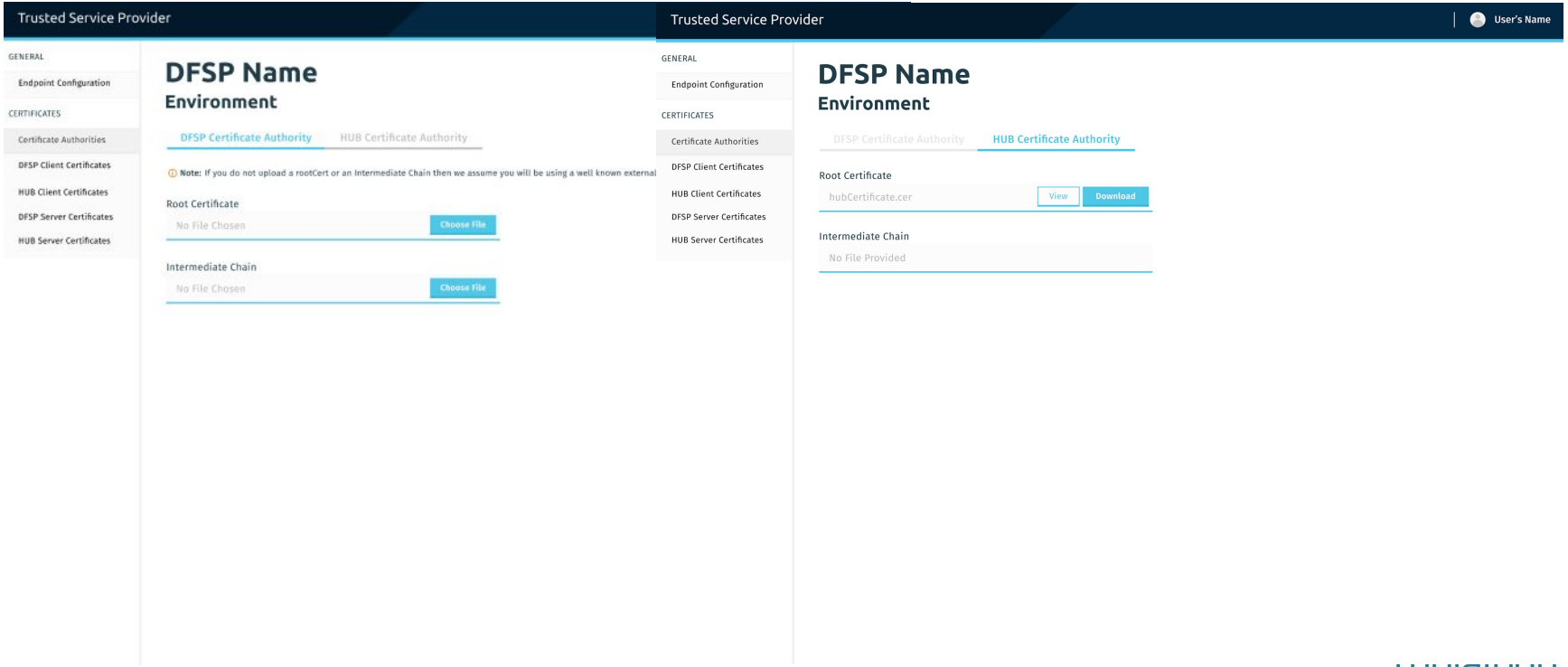
mojaloop

Certificate Authorities can be Self Signed or External

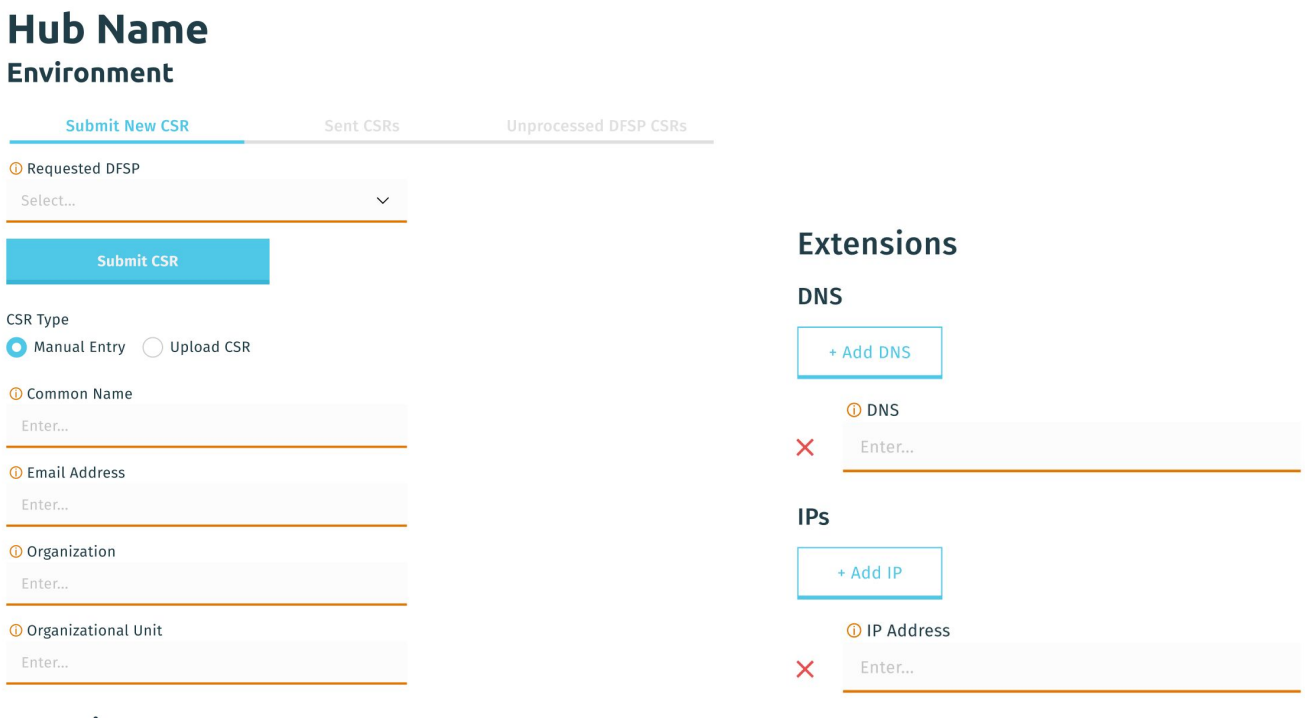




… also available for DFSP

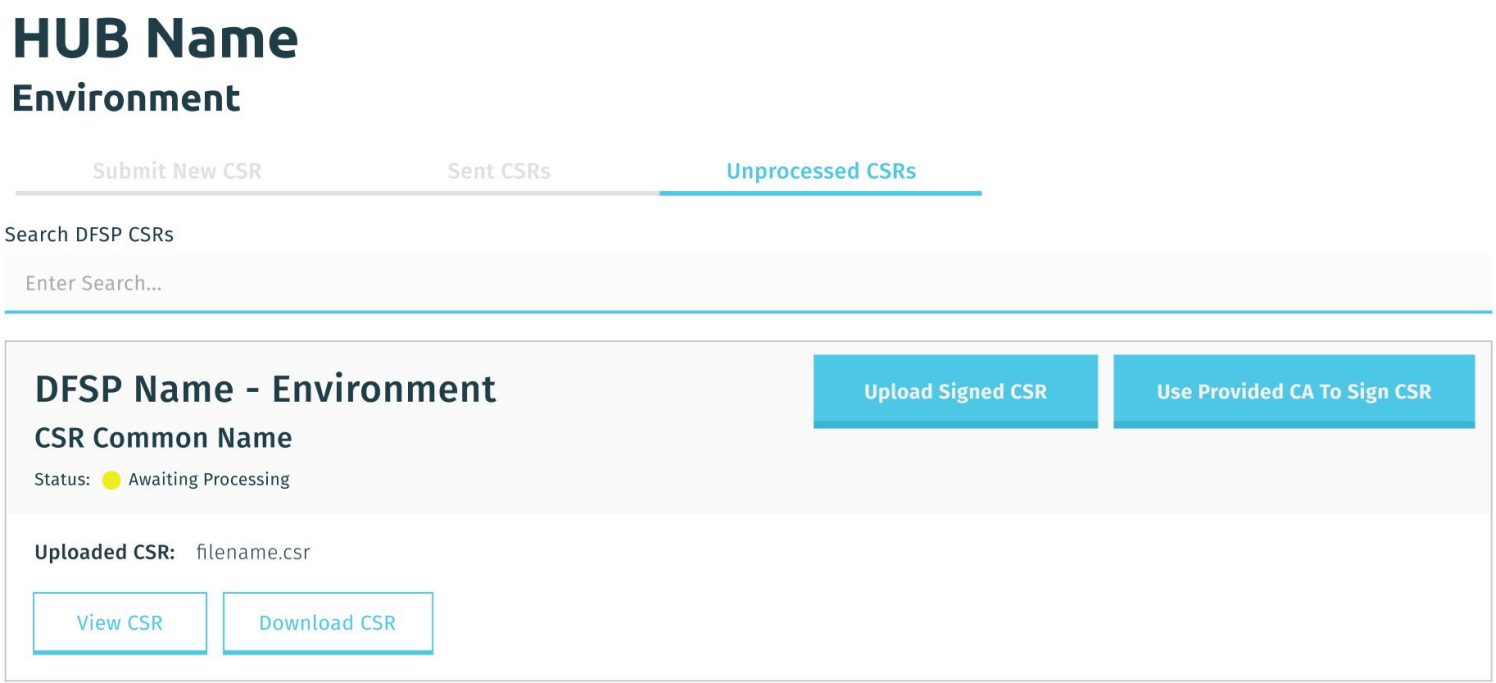


With Initiation of Certificate Signing Requests (CSRs)



mojaloop

CSR status Easily identified



mojaloop

And we are now working on the JWS certificate sharing

- Share DFSP JWS Certificate
- Receive other DFSP JWS Certificates
- When connected to SDK - send test transactions to DFSPs
- Automated Connection to receive new JWS certificates
- Revoking of JWS Certificates

MODUSBOX

Thank You

mojaloop