Application Name	Mojaloop PISP
Application Version	1.0
Description	A review of the PISP application design (linking process) using the STRIDE threat model
Document Owner	Victor Akidiva
Participants	Victor Akidiva
Reviewer	Godfrey Kutumela

Objective

Review the PISP design based on current project status and provide security guidance on authentication, authorization, and compliance. The analysis will review the existing controls as proposed in the design and offer:

1. Validation over the adequacy of the proposed controls
2. Recommendations for any enhancements to the overall PISP control framework as envisioned in the design.

Assumptions

The following are some assumptions we used within our assessment:

1. Customers will use separate authentication credentials for the PISP application and the DFSP service they are subscribed to.

2. The Mobile Application used by PISP will adhere to best practice security recommendations to ensure user security
3. PISP onboarding process will follow the same process as DFSP onboarding process
4. The Web Application is publicly facing and accessible to the Internet from the DFSP side.

Recommendations for the service

The following are high-level recommendations for the Mojaloop - PISP application security:

1. The mobile/web application to be used by end-users should adhere to best practice standards for mobile and web applications i.e. OWASP top 20 a validation needs to be conducted (pentest) before the service is availed to the public. This is to be actioned by PISP / DFSP developers.
2. PISP users must have awareness not to use the same credentials for PISP service and their primary DFSP engagement depending on the relationship. This is to be actioned by PISP / DFSP developers.
3. Request consent process (web) needs to be airtight to prevent the PISP from generating rogue authentication URLs to be presented to the customer and harvest credentials via Man in the Middle Attacks. There needs to be a way for the end-user app to verify that the URL shown by PISP is the actual URL generated by DFSP if possible sending of this URL should bypass PISP. (Ref section 1.3.1 in PISP Linking Process). This is an existential risk to the process. This is to be actioned by Hub / PISP / DFSP developers.
4. As for authentication using the OTP process, ensure the OTPs are not transferable or reusable. This is to be actioned by DFSP developers.

5. Web Authentication URL will be publicly available hence needs to be protected by reverse proxy / WAF (Web Application Firewall) to protect it from web attacks. This is to be actioned by DFSP developers.
6. After grant consent, the user should receive an SMS (or any other appropriate alert) to notify them of a successful consent process with specified PISP. This will be handled in CRED-17 stage in the Credential registration process (Ref section 1.6 in Credential Registration).
7. The Special access token must be part of the credential generated by the switch for the PISP. This will link the credential to the user to the DFSP. There is a risk the PISP can generate a fake FIDO credential and challenge. (Ref section 1.6 in PISP Linking Process)

Commented [1]: Need to study how FIDO process will prevent PISP from intercepting challenge and payload.

External Dependencies

External dependencies are items external to the code of the application that may pose a threat to the application. These items are typically still within the organization's control, but possibly not within the control of the development team or broader network.

Dependency / Components	**Description**
Discovery Artifacts provided by the customer to PISP	Username, MSISDN (phone number), or email address
PISP API endpoint	Details of a selected endpoint to serve PISP requests (Linking, Discovery etc.)
DFSP API endpoint	Standard API endpoints from the DFSP

DFSP Web Authentication URL	Web URL for PISP users to verify their identity. The web application needs to be secured and developed according to best practices.
Mobile App SDK	SDK to be embedded in Mobile APPs that will provide Mobile App -> PISP -> Hub communication functionality via APIs. The mobile app needs to be secure and developed according to best practices. This development may not be in scope.
Auth Service	Authentication service that will host the FIDO capability. The Hub will run its own Auth service; however, onboarding DFSPs will either opt to use it or use their own authentication service.

Components and Trust Levels

Entry points define the interfaces through which potential attackers can interact with the application or supply them with data. For a potential attacker to attack an application, entry points must exist. Entry points in an application can be layered; for example, each web page in a web application may contain multiple entry points.

Name	Description	Trust Level
Mobile APP or SDK	This is the mobile app that the user will interact with to transact with the PISP.	This is outside our scope at the moment.
DFSP login page URL	Login page for all users for web authentication	(2) User with Valid Login Credentials (7) DFSP Web Server User Process (8) Database Read User (9) Database Read/Write User (16) Whitelisted IP
API Endpoint (PISP)	API endpoint that will	(2) User with Valid Login

Commented [2]: Check if necessary

	receive requests from PISP for processing	Credentials/Token (4) Hub User (5) Database Server Administrator (7) Web Server User Process (8) Database Read User (9) Database Read/Write User
API Endpoint (DFSP)	API endpoint that will receive requests from Hub for processing	(2) User with Valid Login Credentials (4) Hub User (5) Database Server Administrator (7) Web Server User Process (8) Database Read User (9) Database Read/Write User

Trust Levels

ID	Name	Description
1	Anonymous Hub user	Unauthenticated user to the Hub exposed applications from within Hub
2	Anonymous DFSP user	Unauthenticated user to the Web Application as exposed by the DFSP
3	Valid hub user	User with valid credentials within Hub
4	Valid DFSP user	User with valid credentials within DFSP
5	Hub Officers	User with valid credentials within Hub
6	Hub IT administrator - web application	User with valid credentials within Hub to manage Infrastructure and applications within the Hub
7	Hub IT Administrator - application server	User with valid credentials within Hub to manage applications
8	Hub IT administrator - database	User with valid credentials within Hub to manage the database server
9	Hub IT administrator - API gateway	User with valid credentials within Hub to manage Infrastructure - API gateway

10	Hub IT administrator - WSO2	User with valid credentials within Hub to manage Infrastructure - WSO2
11	Web server process user	Web server process
12	Database user read-only	Read-only DB user
13	Database user read/write	Read/Write DB user
14	Valid API DFSP token	User with valid API token to push API calls to Hub
15	Valid MTLS authenticated user	Authenticated DFSP using valid SSL certificates
16	Valid DFSP IP (Whitelisted)	Whitelisted DFSP IP

STRIDE Threat descriptions

Type	**Examples**	**Security Controls**
Spoofing (S)	Threat action aimed to illegally access and use another user's credentials, such as username and password.	Strong authentication
Tampering (T)	Threat action aimed to maliciously change/modify persistent data, such as continuous data in a database, and alter data in transit between two computers over an open network, such as the Internet.	Integrity
Repudiation (R)	Threat action aimed to perform illegal operations in a system that cannot trace the prohibited operations.	Non-repudiation
Information disclosure (I)	Threat action to read a file that one was not granted access to or read data in transit.	Confidentiality
Denial of service (D)	Threat aimed to deny access to valid users by making a web server temporarily unavailable or unusable.	Resilience and business continuity

Elevation of privilege (E)	Threat aimed to gain privileged access to resources to gain unauthorized access to information or compromise a system.	Authorization

Linking Analysis

SOURCE - Linking Source Process on Github

Linking is broken down into several separate phases:

1. **Pre-linking** - In this phase, a PISP asks what DFSPs are available to link with. Standard switch authentication between parties.
2. **Discovery phase** - In this phase, we ask the user for the identifier they use with the DFSP they intend to link with. This could be a username, MSISDN (phone number), or email address. The following are some risks with this process:
   a. Harvesting personal information by rogue PISPs
   b. Possible Man in the middle attacks on User->PISP communication
3. Request consent & Authentication - In this phase, a PISP attempts to establish trust between the three parties. Some notes:
   a. Users must verify the process, and this process is in place.
   b. Authentication - In this phase, a User proves their identity to their DFSP. The following are observations:
   c. Authentication via OTP and Web Channel
   d. Web channels will be exposed to the Internet by DFSP hence additional security required.
4. Grant consent - In this phase, a PISP proves to the DFSP that the User and PISP have established trust and, as a result, the DFSP confirms that mutual trust exists between the three parties.
   a. The design needs to show that secret issued to the user during the request for consent is passed to the DFSP in a secure manner
   b. We need to define the contents of the secret/token to determine its contents
   c. Separate transaction tokens to be generated once consent is completed.
5. Credential registration - In this phase, a user establishes the credential they'll use to consent to future transfers from the DFSP and initiated by the PISP.
   a. Credential registration happens between PISP and Mojaloop hub. There needs to be an authorization by the DFSP before the Hub completes credential registration.
   b. Auth Service notifies all parties of successful credential generation, and pending consent object is updated accordingly.

Commented [3]: Review how FIDO can help mitigate this

Commented [4]: Pending consent object + FIDO Challenge generated. Possible MITM by PISP on public key artifact generated by User device. Also how will we verify the public key did come from a user device.

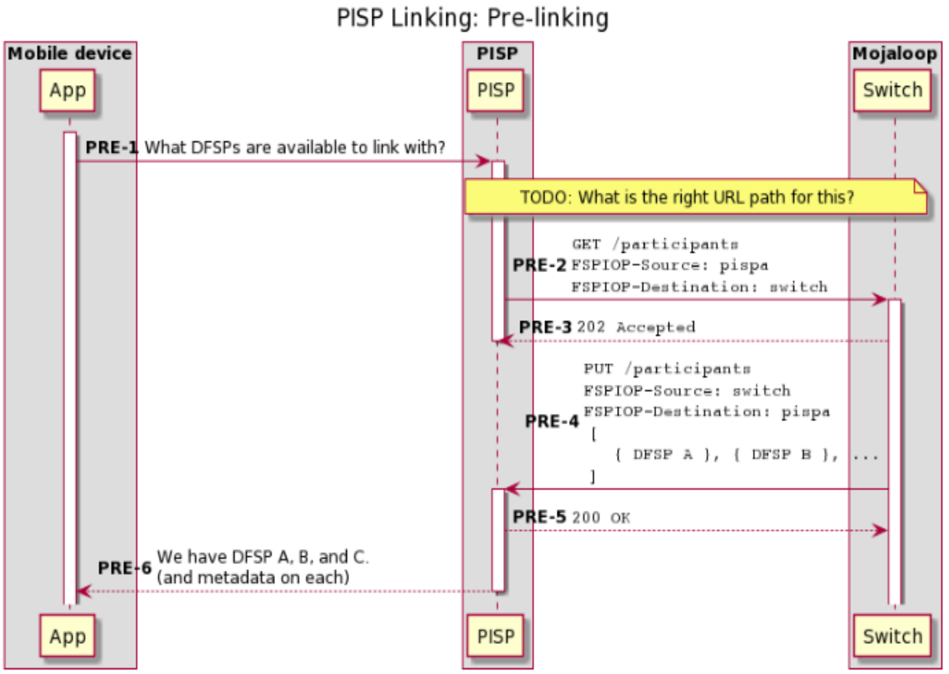
Linking Process STRIDE

The following table summarises how the linking process is affected by threats as viewed from a STRIDE perspective:

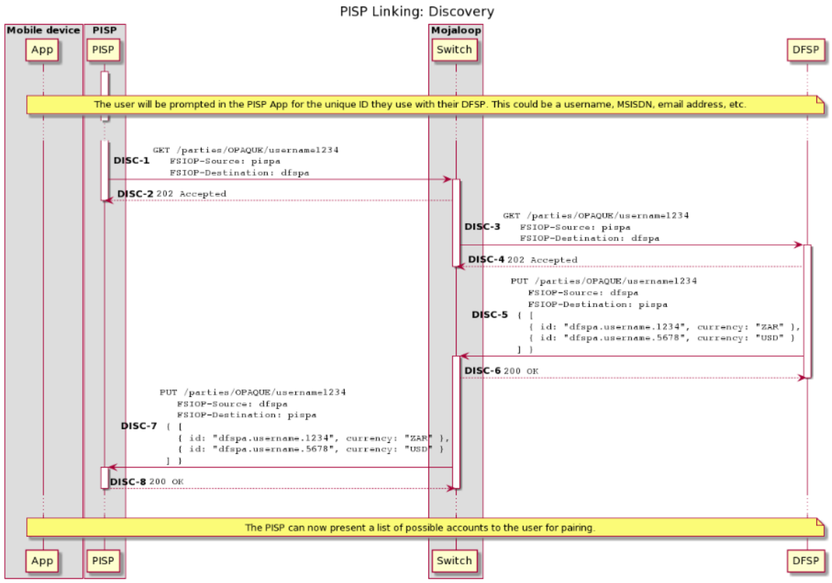
Component	S	T	R	I	D	E
Pre-linking	X	X				
Discovery	X	X		X		
Request consent	X	X	X	X	X	
Authentication - Web	X	X	X	X	X	X
Authentication - OTP/SMS				X		
Grant consent	X	X	X	X	X	X
Credential registration	X	X	X	X	X	X

Diagrams extracted from PISP design page on GitHub (PISP design details)

1. Pre-Linking


PISP Linking: Pre-linking
Mobile device
App
PISP
PISP
Mojaloop
Switch
PRE-1 What DFSPs are available to link with?
TODO: What is the right URL path for this?
PRE-2 GET /participants
FSPIOP-Source: pispa
FSPIOP-Destination: switch
PRE-3 202 Accepted
PRE-4 PUT /participants
FSPIOP-Source: switch
FSPIOP-Destination: pispa
[
{ DFSP A }, { DFSP B }, ...
]
PRE-5 200 OK
PRE-6 We have DFSP A, B, and C.
(and metadata on each)
App
PISP
Switch

2. Discovery



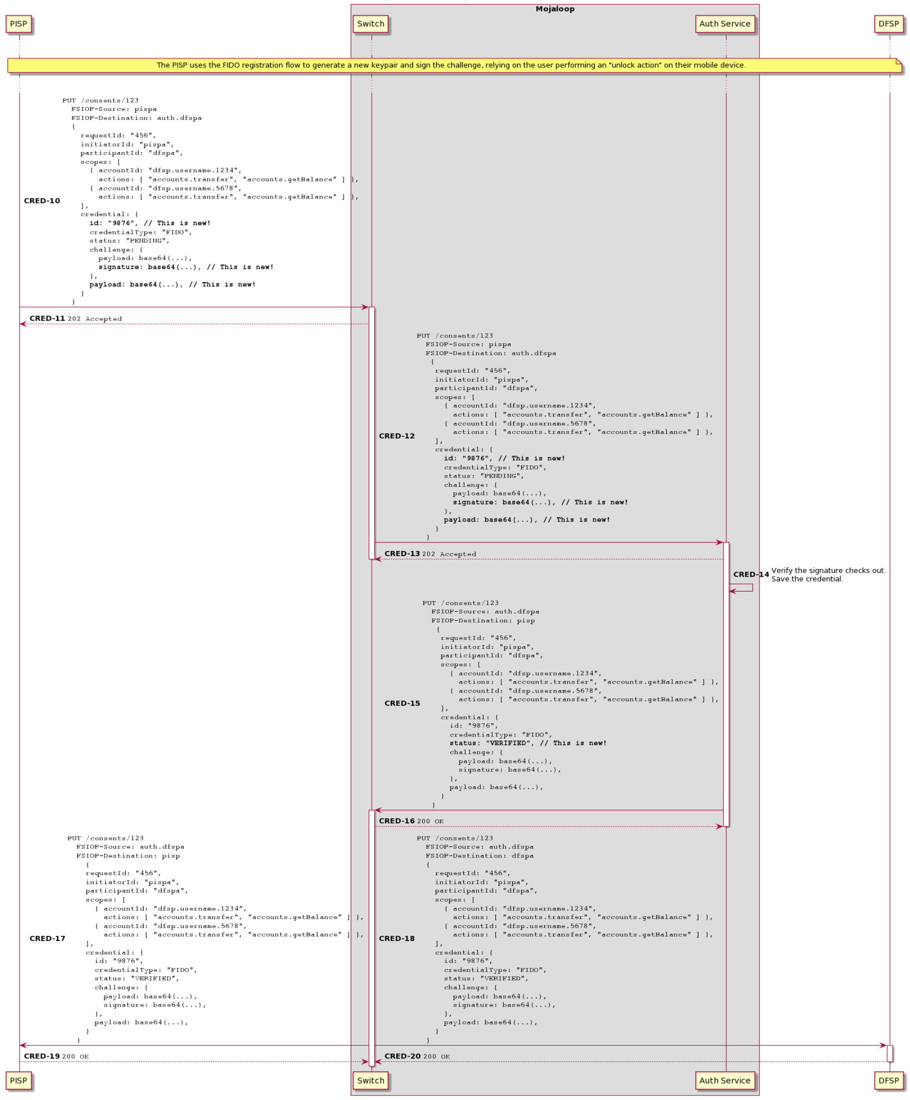
The user will be prompted in the PISP App for the unique ID they use with their DFSP. This could be a username, MSISDN, email address, etc.

```
DISC-1  GET /parties/OPAQUE/username1234
          FSIOP-Source: pispa
          FSIOP-Destination: dfspa
DISC-2  202 Accepted
DISC-3  GET /parties/OPAQUE/username1234
          FSIOP-Source: pispa
          FSIOP-Destination: dfspa
DISC-4  202 Accepted
DISC-5  PUT /parties/OPAQUE/username1234
          FSIOP-Source: dfspa
          FSIOP-Destination: pispa
        { [
          { id: "dfspa.username.1234", currency: "ZAR" },
          { id: "dfspa.username.5678", currency: "USD" }
        ] }
DISC-6  200 OK
DISC-7  PUT /parties/OPAQUE/username1234
          FSIOP-Source: dfspa
          FSIOP-Destination: pispa
        { [
          { id: "dfspa.username.1234", currency: "ZAR" },
          { id: "dfspa.username.5678", currency: "USD" }
        ] }
DISC-8  200 OK
```

The PISP can now present a list of possible accounts to the user for pairing.

Credential registration

PISP Linking: Credential registration (verification)


Mojaloop
PISP
Switch
Auth Service
DFSP
The PISP uses the FIDO registration flow to generate a new keypair and sign the challenge, relying on the user performing an "unlock action" on their mobile device.
PISP
Switch
Auth Service
DFSP

CRED-10

```
PUT /consents/123
  FSIOP-Source: pispa
  FSIOP-Destination: auth.dfspa
  {
    requestId: "456",
    initiatorId: "pispa",
    participantId: "dfspa",
    scopes: [
      { accountId: "dfsp.username.1234",
        actions: [ "accounts.transfer", "accounts.getBalance" ] },
      { accountId: "dfsp.username.5678",
        actions: [ "accounts.transfer", "accounts.getBalance" ] },
    ],
    credential: {
      id: "9876", // This is new!
      credentialType: "FIDO",
      status: "PENDING",
      challenge: {
        payload: base64(...),
        signature: base64(...), // This is new!
      },
      payload: base64(...), // This is new!
    }
  }
```

CRED-11 `202 Accepted`

CRED-12

```
PUT /consents/123
  FSIOP-Source: pispa
  FSIOP-Destination: auth.dfspa
   {
    requestId: "456",
    initiatorId: "pispa",
    participantId: "dfspa",
    scopes: [
      { accountId: "dfsp.username.1234",
        actions: [ "accounts.transfer", "accounts.getBalance" ] },
      { accountId: "dfsp.username.5678",
        actions: [ "accounts.transfer", "accounts.getBalance" ] },
    ],
    credential: {
      id: "9876", // This is new!
      credentialType: "FIDO",
      status: "PENDING",
      challenge: {
        payload: base64(...),
        signature: base64(...), // This is new!
      },
      payload: base64(...), // This is new!
    }
  }
```

CRED-13 `202 Accepted`

CRED-14 Verify the signature checks out.
Save the credential.

CRED-15

```
PUT /consents/123
  FSIOP-Source: auth.dfspa
  FSIOP-Destination: pisp
   {
    requestId: "456",
    initiatorId: "pispa",
    participantId: "dfspa",
    scopes: [
      { accountId: "dfsp.username.1234",
        actions: [ "accounts.transfer", "accounts.getBalance" ] },
      { accountId: "dfsp.username.5678",
        actions: [ "accounts.transfer", "accounts.getBalance" ] },
    ],
    credential: {
      id: "9876",
      credentialType: "FIDO",
      status: "VERIFIED", // This is new!
      challenge: {
        payload: base64(...),
        signature: base64(...),
      },
      payload: base64(...),
    }
  }
```

CRED-16 `200 OK`

CRED-17

```
PUT /consents/123
  FSIOP-Source: auth.dfspa
  FSIOP-Destination: pisp
    {
    requestId: "456",
    initiatorId: "pispa",
    participantId: "dfspa",
    scopes: [
      { accountId: "dfsp.username.1234",
        actions: [ "accounts.transfer", "accounts.getBalance" ] },
      { accountId: "dfsp.username.5678",
        actions: [ "accounts.transfer", "accounts.getBalance" ] },
    ],
    credential: {
      id: "9876",
      credentialType: "FIDO",
      status: "VERIFIED",
      challenge: {
        payload: base64(...),
        signature: base64(...),
      },
      payload: base64(...),
    }
  }
```

CRED-18

```
PUT /consents/123
  FSIOP-Source: auth.dfspa
  FSIOP-Destination: dfspa
  {
    requestId: "456",
    initiatorId: "pispa",
    participantId: "dfspa",
    scopes: [
      { accountId: "dfsp.username.1234",
        actions: [ "accounts.transfer", "accounts.getBalance" ] },
      { accountId: "dfsp.username.5678",
        actions: [ "accounts.transfer", "accounts.getBalance" ] },
    ],
    credential: {
      id: "9876",
      credentialType: "FIDO",
      status: "VERIFIED",
      challenge: {
        payload: base64(...),
        signature: base64(...),
      },
      payload: base64(...),
    }
  }
```

CRED-19 `200 OK`

CRED-20 `200 OK`

Security Requirements for Linking Process

Domain	Description	Risk	Proposed Solution in Current PISP Design	Comment / Additional Recommendation
Pre-Linking	PISP queries for available DFSPs	Minimal	Existing onboarding process business and technical requirements	Existing controls are adequate
Discovery	Using user attributes, PISP queries DFSPs that the user belongs to.	Spoofed mobile numbers can be used to map users to DFSPs for fraudulent purposes Rogue DFSPs may issue infinite queries with random mobile numbers Rogue PISPs may masquerade as DFSPs to the user Rogue PISPs may tamper with user selections. PISP may flood Hub with unlimited discovery calls causing DOS.	Authentication between DFSP and PISP via API calls Secure API Calls API Authentication and Authorization by Hub	Monitoring for unusual Discovery requests, i.e. Discovery not followed by consent. Throttling settings to ensure malicious traffic does not flood the Hub. Strict PISP onboarding process Consider GDPR requirements for PISP and Hub to protect user data.
Request Consent	Establish trust between user, PISP and DFSP	The spoofed secret used to initiate the consent process Rogue PISPs masquerade as DFSPs to collect user credentials via MITM	User to authorize consent process DFSP to issue the user with token/secret which is passed to PISP for presenting to	Secure the token Design token to contain self-verification properties and difficult to spoof/replicate

			DFSP API Authentication and Authorization by Hub OTP Authentication process OK Web Application Authentication has the risk of Man in the middle attack.	A separate token may be needed for each transaction. Send notification to end-user on successful consent process and show details of consented PISP. Protect Publicly exposed web application with a reverse proxy or Web Application Firewall.
Grant Consent / FIDO credential generation	Hub grants PISP consent and FIDO credential generated	The spoofed secret used to initiate the consent process Rogue PISPs masquerade as DFSPs to collect user credentials via MITM and submit spoofed FIDO artefacts.	API Authentication and Authorization by Hub	Need additional design diagram showing FIDO storage (where will FIDO credentials be stored) Need validation of public key generated by the user to ensure user-generated it. DFSP needs to approve the final consent process using a secret shared with User and passed to PISP. Define expiry for secret/token shared with end-user during the consent process.

Assets/ Components

The listing below indicates the various components and resources within the PISPs application and possible mapping to trust levels showing which actors can access what resources.

Name	Description	Trust Level
PISP Users	PISP users at Hub and DFSP.	(2) User with Valid Login Credentials (4) Hub User (5) Database Server Administrator (7) Web Server User Process (8) Database Read User (9) Database Read/Write User
Hub User login	PISP user within the Hub	(2) User with Valid Login Credentials (4) Hub User (5) Database Server Administrator (7) Web Server User Process (8) Database Read User (9) Database Read/Write User
DFSP User login	PISP user within the valid DFSP	(2) User with Valid Login Credentials (4) Hub user (5) Database Server Administrator (7) Web Server User Process (8) Database Read User

		(9) Database Read/Write User
Customer data	Any customer data sent to and from the PISP	(2) User with Valid Login Credentials (4) Hub user (5) Database Server Administrator (7) Web Server User Process (8) Database Read User (9) Database Read/Write User
Hub User data	Details about a PISP hub user profile	(2) User with Valid Login Credentials (4) Hub User (5) Database Server Administrator (7) Web Server User Process (8) Database Read User (9) Database Read/Write User
DFSP user data	Details about a DFSP user profile	(2) User with Valid Login Credentials (4) Hub user (5) Database Server Administrator (7) Web Server User Process (8) Database Read User (9) Database Read/Write User
DFSP data	Details about a DFSP entity profile	(2) User with Valid Login Credentials (4) Hub user (5) Database Server Administrator (7) Web Server User Process (8) Database Read User (9) Database Read/Write User
Web Server code execution	Permissions to execute code in webserver	(7) Web Server User Process (8) Database Read User (9) Database Read/Write User
Database read execution	Permissions to execute code in webserver	(7) Web Server User Process (8) Database Read User (9) Database Read User
Database Read/Write execution	Permissions to execute SQL code in a database server	(7) Web Server User Process (8) Database Read User (9) Database Read/Write User
API calls	Permission to execute API calls on an exposed endpoint	(14) Valid API DFSP Token (15) Valid MTLS authenticated user
User session	Data on user session	(2) User with Valid Login Credentials (4) Hub user (5) Database Server Administrator (7) Web Server User Process (8) Database Read User

		(9) Database Read/Write User
Create Users	Ability to create users - DFSP	(2) User with Valid Login Credentials (4) Hub user (5) Database Server Administrator (7) Web Server User Process (8) Database Read User (9) Database Read/Write User
Create users - Hub	Ability to create users - Hub	(2) User with Valid Login Credentials (4) Hub user (5) Database Server Administrator (7) Web Server User Process (8) Database Read User (9) Database Read/Write User
Access audit data - DFSP	Access to view audit log data	(2) User with Valid Login Credentials (4) Hub user (5) Database Server Administrator (7) Web Server User Process (8) Database Read User (9) Database Read/Write User
Access audit data - Hub	Access to view audit log data	(2) User with Valid Login Credentials (4) Hub user (5) Database Server Administrator (7) Web Server User Process (8) Database Read User (9) Database Read/Write User
Search DFSP information	Access to DFSP search functionality	(2) User with Valid Login Credentials (4) Hub user (5) Database Server Administrator (7) Web Server User Process (8) Database Read User (9) Database Read/Write User
Print report data	Access to DFSp print report	(2) User with Valid Login Credentials (4) Hub user (5) Database Server Administrator (7) Web Server User Process (8) Database Read User (9) Database Read/Write User
Endpoint URL / IP / Hostname	Access to published services	(16) Valid DFSP IP

Trust Levels

Trust levels indicate the various actors who can interact with the PISPs and the level of access they are allowed within the PISPs application. This information will be used to map out role-based access control for PISPs and its resources.

ID	Name	Description
1	Anonymous Hub user	Unauthenticated user to the PISP from within Hub
2	Anonymous DFSP user	Unauthenticated user to the PISP from within DFSP network
3	Valid hub user	User with valid credentials within Hub
4	Valid DFSP user	User with valid credentials within DFSP
5	Hub Officers	User with valid credentials within Hub
6	Hub IT administrator - web application	User with valid credentials within Hub to manage Infrastructure and applications
7	Hub IT Administrator - application server	User with valid credentials within Hub to manage applications
8	Hub IT administrator - database	User with valid credentials within Hub to manage the database server
9	Hub IT administrator - API gateway	User with valid credentials within Hub to manage Infrastructure - API gateway
10	Hub IT administrator - WSO2	User with valid credentials within Hub to manage Infrastructure - WSO2
11	Web server process user	Web server process
12	Database user read-only	Read-only DB user
13	Database user read/write	Read/Write DB user
14	Valid API DFSP token	User with valid API token to push API calls to Hub
15	Valid MTLS authenticated user	Authenticated DFSP using valid SSL certificates
16	Valid DFSP IP (Whitelisted)	Whitelisted DFSP IP

Additional Requirements for Understanding

1. Diagram showing External FIDO authentication
2. Details of how Mojaloop will generate and store FIDO credentials (using Vault?)
3. Validation process and structure of the secret generated by DFSP during the consent process.