* My main question is why a lot of information that is related to the transfer between the Payer FSP and the Payee FSP needs to be shared to the PISP? The PISP should (at least in my mind) only care about presenting the end user quote for confirmation, and the result of the transaction to the end user. In the current design, it seems like private information like date of birth for both Payer and Payee is shared, internal IDs are shared, as well as internal FSP fees that are never meant to be shared to any party that is not involved in the actual transfer between the FSPs. Please explain to me what the PISP is supposed to do with that all that information? I'm not comfortable at all sending all this information to a PISP without proper understanding of why this information is required in the PISP. The PISP is not supposed to perform any kind of screening as far as I'm aware.
	+ *Answered in the comment section*
* As part of the project for designing the FSP Interoperability API, a glossary was developed which is located here <https://github.com/mojaloop/mojaloop-specification/blob/master/documents/Glossary.md>. The term that we decided to use for the financial service provider was FSP, not DFSP. If we should be consistent and follow that glossary, then please use FSP instead for the PISP API as well.
	+ *If DFSP is in the glossary as an alternative to FSP, I don’t understand why there’s a problem with using it.*
* There is a mix of American and British spelling, I would assume that we should use American spelling? This is a real problem in for example **GET /authorizations/**<*ID*>, where the correlation ID is called **authorisationId**. Another example is enquire vs inquire.
	+ *Fixed*
* If we should keep consistency with the FSP Interoperability API logical API service names, then please keep them relatively short. For example "Request permission to perform operations on one or more accounts owned by another participant." is more of a sentence than a logical name.
	+ *All API service names removed*
* Please do not allow unlimited number of elements (where Cardinality is set to for example 1..n), as an attacker may try to submit an extremely large number of elements, making the parser crash or cause a buffer overflow (however unlikely it might be in a scheme like this which is also using TLS). There is no purpose in allowing for example an unlimited number of scopes. Please decide a reasonable maximum number of elements. The same is also valid for any kind of string, this needs to have a limit in size.
	+ *Answered in comments*
* Please be consistent in naming, sometimes <ID> is used, in other instances <Id> is used.
	+ *Fixed*
* Many of the references to the FSP Interoperability API Definition in Section 6 seem to be off by 1, for example reference says 6.7.3.1 but where the actual Section number is 6.7.4.1.
	+ *Fixed (I think)*

**Specific comments**

Section 2.1.2: "This resource is an extension of the /authorizations resource described in Section 6.6 of Ref. 1 above"

* I don't understand the *extension* part here, to me it seems like a replacement? How should the FSP be able to support both as they have the same URI? The text also says that it is a new "A new POST service", this does not sound like an extension.
	+ *The resource is extended by having a service added. The PISP interface will have its own URI.*

Section 2.1.2.1.1: "This resource will allow the DFSP which owns an account from which the PISP has requested that funds be transferred to ask for authorisation from the PISP that its user has in fact confirmed that the request should be honoured, and that they have done so via two-factor authentication on the registered device."

* This is a very long sentence that is really hard to understand.
	+ *Rewritten*

Section 2.1.2.2.1: "When a PISP has shown the terms of a transfer to its customer and obtained the customer’s consent to or rejection of those terms, it will return this response to the DFSP."

* It is not a response that is returned, it is a callback.
	+ *Corrected*

Section 2.1.2.2.1: "The content of this callback will be as described in Section 6.6.3.1 of Ref. 1 above."

* Section 6.6.3.1 in FSP Interoperability API definition is not a callback, it is the **GET** request.
	+ *Corrected*
* Why is the logical API service name something other than in the FSP Interoperability API definition if it should be the same callback?
	+ *Removed*

Section 2.1.3.1.1: "The <ID> in the URI should contain the requestId that was assigned to the request by the PISP when it originated it."

* "... when it originated it." there are too many "it"s here to make this sentence easy to understand..
	+ *Corrected*

Section 2.1.3.1.2: "callbackUri"

* The Type name is Url instead of Uri. Please be consistent.
	+ *Corrected*
* Is this meant to be the full URI? I would prefer to have the host name static to decrease the likelihood of a man-in-the-middle attack. If the host name is static, then what can be changed by an attacker is just the path.
	+ *How is the DFSP to know which host to use?*

Section 2.1.3.1.2: "The HTTP request POST /consentRequests is used to request a DFSP to grant access for the sending PISP to one or more accounts owned by a customer of the DFSP who uses the PISP."

* Please define what a "sending PISP" means.
	+ *Reworded*

Section 2.1.3.2 "This section describes the callbacks that are used by the server under the resource/consentRequests."

* Missing whitespace..
	+ *Corrected*

Section 2.1.3.2.1: "The PUT /consentRequests/<ID> resource returns the current state of the permissions relating to a particular condition request. "

* "condition request"?
	+ *Corrected*

Section 2.1.4: "the /consents resource is used to negotiate a series of permissions between the PISP and the DFSP which owns the account(s) on behalf of which the PISP wants to transact."

* "the" should be "The".
	+ *Corrected*

Section 2.1.4.1.1: "The GET consents/<Id> resource allows a party to enquire after the status of a consent."

* Should be "GET /consents/<ID>"
	+ *Corrected*

Section 2.1.4.1.1: "The <Id> used in the query string of the request should be the consent request ID which was used to identify the consent when it was created."

* Which query string are you referring to here? There is no query string as part of the request that I can see?
	+ *Corrected*

Section 2.1.4.1.2 "The PUT /consents/<Id> resource is used to return information relating to the consent object whose consentId is given in the query."

* Which query are you referring to?
	+ *Corrected*

Section 2.1.5: "The PISP will be permitted to issue a PUT /parties response."

* Please explain how this would work. To me, it seems like the account holder should request an alias to be created for them at the FSP, preferably as part of the consent request when the PISP is given access to an account. When this alias is used by for example a merchant, the FSP routes the confirmation to the PISP.
	+ *Our current thinking on this is as follows. A PISP will be permitted to register identifiers which will direct DFSPs enquiring about the PISP’s customers to the PISP. When the PISP receives a GET /parties request from another party, it will respond to that request as if it were able to fulfil it directly. It will then request the DFSP to which this customer is linked to fulfil the transfer request.*

Section 2.1.5.2.1 "The PUT /parties resource will use the same form as the resource described in Section 6.3.4.1 of Ref. 1 above.
It should be noted, however, that the Party object returned from this resource has a different format from the Party object described in Section 7.4.11 of Ref. 1 above. The structure of this object is described in Section 2.2.1.21 below."

* I don't understand how this will work as they have exactly the same name. What is the purpose of the note as the only object that is in the **PUT /parties** callback is the Party? That means that the entire service is different, as there is only one element and that differs between the PISP and the FSP Interoperability API?
	+ *The purpose of the note is to specify that the data model is apparently the same as that described in Ref. 1, but that this apparently identical content hides a difference at a lower level.*

Section 2.1.6: "The services resource is a new resource which enables a participant to query for other participants who offer a particular service. The requester will issue a GET request, specifying the type of service for which information is required as part of the query string. The switch will respond with a list of the current DFSPs in the scheme which are registered as providing that service."

* Can you please give some examples of how this service is supposed to be used, so that it is easier to understand the purpose of it?
	+ *A PISP will use the service to obtain a list of those participants in the system which support account linking and transfer by proxy, since it may not be a requirement of the scheme that all participants should do so. A DFSP will use the service to obtain a list of FIDO servers which it can use to authenticate signatures received from a PISP. A DFSP may use the service to confirm that an account linking request has come from an entity which is registered with the scheme as a PISP.*

Section 2.1.7: "The PISP uses it to request the owner of the PISP’s customer’s account to transfer a specified amount from the customer’s account with the DFSP to a named recipient."

* Please use term Payee instead of recipient as recipient can mean both a recipient of message or recipient of money.
	+ *Corrected*

Section 2.1.7.2.1 "The PUT /thirdpartyRequests/transactions/<ID> resource will have the same content as the PUT /transactionRequests/<ID> resource described in Section 6.4.3.1 of Ref. 1 above."

* Please explain why the PISP needs to have the information that is sent in the **PUT /transactionRequests/**<*ID*>?
	+ *The PISP needs to know that the Payer DFSP a) has validated the request; b) is actioning it.*

Section 2.1.7.2.2 "The issuing PISP will expect a response to their request for a transfer which describes the finalised state of the requested transfer. This response will be given by a PATCH call on the /thirdpartyRequests/transactions/<ID> resource."

* This comment is related to the bullet above. As currently specified, there does not seem to be any purpose in the PUT callback that I can understand. Based on my (perhaps limited) understanding, the result of the transaction should come in the PUT callback after the transaction has been performed between the Payer and Payee FSP.
	+ *As described in the comment above, the payer DFSP may or may not accept the transfer request from the the PISP. This response does not describe the final state of the transaction (which is given in the PATCH); it confirms to the PISP that the DFSP has agreed to undertake the transfer on its behalf*

Section 2.1.7.2.2 "The content of this resource will be the same as the data model described in Table 23 of Ref. 1 above, in the section describing the PUT command on the /transfers resource shown in Section 6.7.3.1 of Ref. 1 above."

* Please explain to me why the PISP needs to know about the inner details of the transfer between the Payer and Payee FSP?
	+ *Because the PISP will need to confirm to its customer that the transfer requested by the customer has completed successfully (or, perhaps, failed.) It is an interested party.*

Section 2.1.8.1.2: "Challenge"

* Please do not allow a string of any size.
	+ *Dealt with in general response*

Section 2.1.9.3.1: "PUT /thirdPartyRequests/validations/<type>/<ID>/error"

* The "thirdPartyRequests" part seems a bit misplaced here, as this is not present in any other service as part of the resource.
	+ *Corrected*

Section 2.2.1.1: "AccountAddress"

* Can you please explain the purpose of the AccountAddress as part of this? What will it be used for? I would like to understand the purpose of this change in relation to PISP. I know that this has earlier been discussed as a potential part of transfers between schemes, but I don't really see the purpose for PISPs.
	+ *The PISP needs a unique identifier for each customer account which it can use to identify the accounts to which its customer wishes to grant access. This value, which is decided by the DFSP, is the value which the PISP uses in the* ***scopes*** *element in the consent request to identify to the DFSP which accounts are in question.*

Section 2.2.1.2: "It does not need to provide an address that makes the account identifiable outside thementity's domain."

* "thementity's"?
	+ *Corrected*

Section 2.2.1.2: "IMPORTANT: The policy for defining addresses and the life-cycle of these is at the discretion of the address space owner (the payee DFSP in this case)."

* I don't understand what the "the payee DFSP in this case"-part has to do with PISPs?
	+ *Typo. Corrected*

Section 2.2.1.8: "The AuthenticationValue data element contains a response returned by the recipient of an authorization request. It is described in Section 7.3.3 of Ref. 1 above, and is extended to support the new authentication type used for PISP."

* Are you proposing to extend the definition of the type in the FSP Interoperability API, or is this just using the same name which would be confusing?
	+ *Using the same name.*

Section 2.2.1.10: "Challenge"

* There is another Challenge in Section 2.1.8.1.2 (POST /thirdPartyRequests/verifications). Is that supposed to use this Challenge, or are different challenges?
	+ *It’s this challenge. Corrected.*

Section 2.2.1.10 "payload"

* Please do not allow a string without a maximum size.
	+ *Repeat issue*

Section 2.2.1.25 "This information is forwarded to the PISP by the Payer DFSP so that the PISP’s customer can make an informed consent to the transfer, and is forwarded to the FIDO server (if one is used) to confirm the bona fides of the authorisation received from the PISP."

* Please explain to me why the customer would be interested in all this information as showed in the table? This contains private information like the date of birth which should be used for screening purposes. Information like date of birth is **not** meant to be sent to a consumer, and I have no idea what the PISP is supposed to do with the information. Any kind of personal data that is transmitted must have an acceptable reason for it to be shared.
* The information showed in the table is also not the quote information, it seems to be a copy of the Party type.
	+ *Issue discussed separately*

Section 2.2.1.26: "The following shows a proposed revision of the Party data element to support the additional information required to support PISP interactions."

* Why is this information required to support PISP interactions? See comment above as this contains personal information which is not meant to be shared for other reasons than screening (date of birth). Why should the consumer or PISP be interested in seeing the merchantClassificationCode?
	+ *Issue discussed separately*

Section 2.2.1.28: "ScopeAction"

* The name "ScopeAction" sounds strange, either it is a scope, or it is an action?
	+ *It’s an action on an account, which is requested as part of a scope*

Section 2.2.1.29: "Scope"

* The name of the element is Scope (should be "scope"), but the type name is "ScopeAction"?
	+ *I think I corrected this. If I did, it was a cut-and-paste problem with the narrative.*

Section 2.2.1.31: "Transaction"

* Please explain to me why the PISP needs all these details regarding the transaction between the Payer FSP and the Payee FSP.
* For example, as an FSP I would not like the Payee FSP fee or Payee FSP commission to be sent to another party that is not involved in the actual transaction between the Payer FSP and Payee FSP (the PISP in this case). I would really not want to spread any personal information like date of birth which is part of the Party type.
	+ *Issue discussed separately*

Section 2.2.1.33: URL Regular expression

* The regular expression is not valid (the //). Additionally, the regular expression seems to allow me to enter any number of characters, meaning that an attacker can send an extremely long string, which can cause a parser crash or buffer overrun. Please don't allow an unlimited number of characters. Please investigate if you can find a better regular expression for validating the URL.
	+ *Corrected*

Section 2.2.2.6: "THIRDPARTYLINK"

* Please be consistent with existing names. Other types which contain more than one word (PERSONAL\_ID and ACCOUNT\_ID) has an underscore between the words, hence the name should be "THIRD\_PARTY\_LINK"
	+ *Corrected, and included as best practice*

Section 2.2.2.7: "BALANCEENQUIRY", "FUNDSTRANSFER"

* Please be consistent with existing enums. See above.
	+ *Corrected, and included as best practice*

Section 2.2.2.8: "3PDFSP"

* Why not just use some word that is easier to understand? You will likely not see any difference in performance compared to using a longer name. "3PDFSP" sounds like a character in Star Wars ;)
	+ *Corrected in the interest of world peace and harmony*