

Flexibility Digital Infrastructure

Ofgem's System Use Case exercise.

Final version

Date: 19 January 2024

Author: ElectraLink

Summary

Our System Use Cases envision a hybrid system with an ecosystem of parties/platforms that are used to register users and to provide services for managing and aggregating assets (aggregator & FSP roles) and bringing them to flexibility buyers (marketplace platforms). They may develop their own data models and services; however, we see a need for the use of common services by a centralised, independent, and highly reliable party or system as the “system of reference” or “flexibility data backbone”. We refer to these services in the scenarios below as Common User Data services and Common Asset Data Services. These at the very minimum would have the elements of user and asset registration with enough data to manage de-duplication and may also involve partly asset - full “static” data for each asset (including portfolios).

A core principle in all our thinking here is that the foundational services delivered by BUC.2 & BUC.4 (and the underlying BUC.1) should be neutral as to how the other BUCs are delivered, i.e. that they should not restrict the technology or business model of the organisation(s) providing platforms for market operation, system operator functions, etc. Other core services/systems independent from the Common Data Services include Archival Data Service – metering data and suchlike, Contract registration, Product registration and other potential additional services that could be built on top of the basic infrastructure, such as Data transformation services and Device discovery service.

Thus, the foundational common services should provide an open API based on well-established technology that can be accessed by a variety of platforms providing these higher-level services. We also expect that there will need to be appropriate support to allow platform developers to build their platforms to integrate with these services (e.g. API documentation and support; API versioning and governance; test environments / sandboxes; etc). Note however that the foundational services may need to interact with some of the other BUCs, e.g. the Common Asset data register or User register service might interact with the Product Service (BUC.N) to determine what data an asset needs to provide in order to qualify for and deliver a given product; likewise there might need to be interaction with a Contracts service to manage questions of stackability of products across assets & portfolios. Some of these services may also be foundational aspects of the FDI.

Although we envision the SUCs of both BUC.2 and BUC.4 as conceptually similar architectures, we acknowledge them as different systems each with their own requirements, processes, and complexities that we strived to capture in the scenarios.

Whilst the systems that underpin the user and asset registers are different ones, their integration is directly embedded so they can communicate and exchange, link and verify data as close to real time. These systems can be operated by the same or different parties.

In developing the use cases here, we have assumed that messages are delivered between services by an underlying messaging layer, as represented by BUC.1. This layer needs to deliver the attributes such as reliability & availability, assured message delivery, security, non-repudiation, etc, appropriate to such a critical national infrastructure.

Narrative of the System Use Case

The SUC proposed describes how a set multiple parties/platforms can be used to register users and bringing them to flex buyers, but they all use and interface with a common central user register to assign unique IDs and manage de-duplication. In so doing, they ensure that their internal data can be translated to the common model embedded in these common services. (There is scope for a data translation service to support this translation, but that's probably a value-added service on top of the FDI rather than a core service).

A step that we consider essential in the design of a new FDI that enables this, and the other identified use cases is to define what we mean by user and classify users. The reason for this is the flexibility landscape is growingly complex and roles, such as the aggregator one, have been fragmented into multiple sub-actors that are placed in different stages of the flexibility end to end journey, and the interactions, data collection, management and several other aspects are distinct roles.

Parties recruit multiple users, integrate their assets and then bring them to market via other parties:

- E.g. FSP with app recruits users then works with large VPP to aggregate them into a portfolio for a VLP.
- E.g. Smart Home Tech company integrates multiple assets within the home (HEMS) and offers them to an FSP.
- E.g. User facing FSP with gamified and rewards business model recruits users then brings them to market via a large aggregator & VLP.

A key function that these intermediaries are undertaking is building and managing portfolios of assets, e.g. all the EV chargepoints on a feeder, to take to market. Assets may be part of multiple portfolios (by location, capability, etc). This allows aggregators and FSPs to “stack” them across multiple flex markets. e.g. an asset might be time-sliced across a local DSO flex service, national Balancing Mechanism bids & offers, and time-shifting arbitrage services for an energy supplier. In each case, the aggregator would be bidding a different portfolio to a different flex buyer, with a different mix of assets in each portfolio. It is also possible for assets to be represented by multiple aggregators in this model, e.g. with a different aggregator representing the asset for ESO and DSO services c.f. the aggregator managing it for the energy supplier. This means that providing clarity on which asset is represented by which “aggregator” to provide which services is an important function of the digital infrastructure, achieved through a combination of asset, product and contract registration services.

We describe in detail below the users/actors and their descriptions, we also include the most relevant for the scenarios in the table of actors and systems below.

1. Flex owner: any actor from a householder to a small wind turbine to a large I&C. It is the actor that literally generate the flexibility by altering their generation or consumption. In most cases, these users will provide flexibility services only through an aggregator, although it may be that this evolves to some actors providing flexibility directly to SOs without going through an aggregator.

Complexity of flex markets (e.g. energy suppliers relabel DFS) and multiplicity of apps, etc, makes it hard to track what they're currently signed up to. Especially as they may be signed up with multiple aggregators (e.g. with different aggregators representing them to different markets, as noted above; it's also likely that different assets in the home may be represented by different aggregators – EVs & chargepoints might be managed by the leasing company which underpins their purchase; heat pumps might be aggregated into a local heating portfolio by a heat-a-service specialist; etc.) Combined with the complexity of mapping of service names used by aggregators and FSPs onto the services being bought by SOs and with significant churn in domestic markets, this means that we have to expect that some users will lose track of who is representing their assets for which service,

and hence to accidentally sign up for conflicting services. The FDI needs to be able to create transparency onto these mappings.

Also, the ultimate source of information about asset upgrades, maintenance, changes of ownership, etc. And for linking assets to new MPANs when they move, etc. A key role of aggregators is to integrate owners and assets in a way that makes this info accessible to & interoperable with system services.

2. **Aggregator:** user facing, recruit users and manages the recruitment, portfolio building and settlement with users, creating portfolios of very small load assets. E.g. Loop, Hugo, equiwatt. Their speciality is in recruiting and communicating with customers and in integrating the assets they own into portfolios that can interoperate with energy system services. This is a very distinct capability c.f. the market-facing capability of FSPs.
Acts as agent for owner, bringing them into a portfolio that can be sold to flex markets.
3. **Flex service provider (FSP):** energy / flex market specialist, taking portfolios to market and optimising returns. Holds the direct relationship and interaction with SOs. Manage the largest form of unit/portfolio. – Flexitricity, Octopus, etc. FSPs manage the qualification and administrative processes required by the SOs, which can be quite burdensome (e.g. to become a VLP for BM). They also manage real time trading across the multiple flex markets / services, i.e. they are primarily market-facing entities.
4. **Flex planning:** S.O role to identify forward need for flex to support operations of network / system. Needs visibility onto availability of flex, historical reliability of delivery by assets (including portfolios of assets), etc. Defines the products that are bought / sold on flex markets. NB “products” also needs to be clearly defined – we expect to see something like generic product “templates” such as DSO Sustain or ESO Quick Reserve being instantiated as specific products such as “10MW of Sustain for the time period 5-7pm on weekdays for December & January”. It is these specific instances that are bought and sold through the markets.
5. **Flex procurement:** S.O role managing process of buying flex. Drives processes such as qualification of users and assets. If a user or owner has already been qualified by another SO, then may be able to simplify their own qualification process. (At best, they’ll trust the qualification by another SO. More likely they’ll want to do their own due diligence but will be able to use data obtained by the other SO. Either way, this reduces administrative overheads for users and their agents/aggregators & FSPs.)
6. **Flex market operator** (There are currently multiple markets – regional DSO markets, ESO markets for a variety of services, also scope for energy suppliers and traders to buy flex, secondary trading markets, etc. Even if these ultimately migrate to a single platform and flex facilitating party, they are likely to remain conceptually distinct markets).
7. **System operator** – SO role managing network & system state in real time, identifies need for flex (choosing from the products defined by flex planners) & invokes services.
8. **Flex dispatch** – drives actual dispatch of flex (e.g. translates need into specific FSPs / portfolios to dispatch). Translates decisions by the system operator into directions to specific FSPs, aggregators & assets to deliver service.

9. Special users (regulators, investors, other third parties with an interest to view what's going on).

NB parties might combine roles, e.g. some companies may choose to combine the Aggregator and FSP role, providing end-to-end service to flex owners. But others will prefer to specialize in order to maximise the value they add for a specific function.

BUC 2 & 4 currently take a system-led view, focusing on players close to the SO (e.g. FSP), implicitly assuming those players manage the wider circles. That's a reasonable, pragmatic view to create quick progress at this stage, but it may want to be more cognizant of the perspective of customers and other stakeholders in order to optimise service definitions to deliver benefits across the system (e.g. to facilitate optimum recruitment; to capture full data for analysis; to partition out permissions appropriately). Or at least create a path to evolve in these directions.

Use Case conditions	
Assumptions/Pre-requisites	
1	Seamless integration utilising the Data Sharing Infrastructure (Trust + Prepare + Share) outcomes defined in BUC.1 and BUC1.1.
2	Relevant data- and entity- assurance agreements are defined as part of BUC.1 and/or BUC.8 and are readily implementable by the system.
3	Information flows utilise a necessary common data standard and wider IT architecture support the functions, defined in BUC1.1.
4	Seamless integration to utilise common user registration outcomes in BUC.4.
5	Seamless integration to enable common pre-qualification outcomes in BUC.7.
6	Seamless integration to enable common TSO-DSO coordination outcomes in BUC.6.
7	Seamless integration with relevant common compliance tools in BUC.8
8	Asset details submitted to the system are accompanied with a mechanism for validating owner consents.
9	Asset details are validated according to a transparent and well-defined logic.
10	e.g. Approved BUC.2 platforms have BUC.4 system integration directly embedded and not require a UserID to be sent by BUC.4 system separately.
11	e.g. Simultaneous updates are resolved by a defined set of rules implemented by the CoordinationSystem_BUC.2

Actor name	Actor type ("system" or "business")	Actor description
Flex-owner	Individual/ Business	Any actor from a householder to a small wind turbine owner, to large I&C. It is the actor that literally produces the flexibility by altering their generation or consumption. In most cases, these actors will provide flexibility services only through an aggregator, although it may be that this evolves to some actors providing flexibility directly to SOs without going through an aggregator.
Aggregator	Business	Their speciality is in recruiting and communicating with customers and in integrating the assets they own into portfolios that can interoperate with energy system services. This is a very distinct capability c.f. the market-facing capability of FSPs.
Flexibility Service Provider (FSP)	Business	FSPs manage the qualification and administrative processes required by the SOs, which can be quite burdensome (e.g. become a VLP for BM). They also manage real time trading across the multiple flex markets / services, i.e. they are primarily market-facing entities.
Marketplace Platform (MPps)	Business	Marketplaces for trading flexibility services, existing platforms such as NODES, Flexible Power, Piclo, Electron, desX, etc.
Markets Operator (MO)	Business/system	Entities that facilitate the end-to-end flexibility service delivery. *Not to be confused with a Market.
System Operator (SO)	Business	Entities that buy flexibility to operate the power system e.g. ESO, DNO. In our examples below we have focused on the SO as a single entity, not exploring the different roles within the SO noted above. But ultimately these roles will need to be recognised to manage data access permissions effectively.
FDI-user	Business	Data producers and consumers verified within the FDI ecosystem, e.g. MO, SO, FSP and Special Users
Special Users	Business	Other entities verified within the FDI ecosystem such as the regulator, investors or third-party service provider with an interest in flexibility services.
Common_user_data_services	System	A new system, a lightweight registration service that focuses on de-duplication and assigned unique asset IDs. There is currently not an existing system as such, although there are existing systems and moving parts that can be brought together relatively fast to pivot into the required system. E.g.: The underlying messaging system of a national service. The AAR and CAR programs have received innovation funding and are very likely to be capable of providing this functionality. The AAR, CAR, FAIR solutions can be pivoted into systems. Primary user register service.

FO-FSP_common_register	System	A central register to check the allocation of Flexible Owner to FSP and de-duplicate users amongst FSPs. Second-level user register service.
------------------------	--------	--

Scenario

User verification and uploading commercial information:

1. First time User, FSP-1, has their organisation verified by the system (or systems).
 - a. Consider how FSP-1 will verify who they are to the system.

This is more of a detailed design question for a later stage, but the system **Common_user_data_service** can be either:

- The 1st point of contact for FSP/Aggregators. Potential greater independence.

Or

- It can be in the backend – call from MOs or MpPs calling to the single-user service to the **Common_user_data_service** and getting the ID back to then allocate to the FSP.

* The first route is probably the best one for FSPs, as they are expected to understand system processes. But for aggregators and end users, the second route might be more appropriate - they are not energy system specialists, so might want to delegate this function to FSP or MO. (More likely FSP than MO - if the aggregator or user tries to go direct to market but doesn't have the expertise to first get an ID, then the market might tell them to get an FSP to represent them).

- This is then sent to the **Common_user_data_service** where this fields of basic information are logged for a basic record and mainly to identify duplicates, but no more.
- **FSP-1** is then assigned a unique ID.

Depending on Market facilitator role resolution, **FSP-1** would go through **MO** – or accredited/selected delivery arm **MpP1** to provide flexibility services to them. Or **FSP-1** would use their selected **MpP** – **MO1*** to register to provide flexibility services to **SO-DSO1**. *

We' might suggest a 2-stage process. FSPs would be assigned a provisional ID when they provide information, then that ID is confirmed when the info has been validated (as we do with asset data, as discussed below). This lets the FSP take actions to begin registering portfolios and bidding for services while their details are checked.

2. The system issues a unique identifier (ID) associated with FSP-1's organisation.
 - a. Consider how issuing multiple unique IDs will be avoided if FSP-1 tries to register again.

By having a **Common_user_data_service**, issues of multiple IDs can be avoided. And as it undertakes checks before assigning an ID, duplication will be reduced. But it may not be entirely eliminated, especially if checks take time. That creates potential scope for a second level de-duplication service

that analyses the user register, and other data, in more detail and looks for duplicates. (This is to be further explored and fleshed in the more detailed design thinking).

b. If multiple systems are able to issue unique IDs, consider how IDs remain coordinated and unique across multiple systems.

n/a. As noted above, we're proposing a common single service. It may be backended by multiple registration and checking threads to create scalability and resilience, but it's inherently a single system. Plus the fallback offline deduplication service noted above to handle more subtle duplicates.

The key aspect would be for MO-MpPs not being able to assign IDs, only the **Common_user_data_services**.

NB FSPs and aggregators may have their own IDs, internal to their systems. They then need to translate them to the common system-wide ID allocated by the central service.

3. FSP-1 provides relevant commercial information using their unique ID.

a. Consider how FSP-1 will understand if their commercial information is relevant, compliant and necessary.

The requirements will need to be openly published, of course.

The register could also provide a checking service, where users could test their data before submitting it for registration. And there will probably be a testing service / sandbox, where users can test their system, APIs, etc.

4. FSP-1 configures relevant data-and entity- assurance agreements (defined in BUC.1/1.1 , BUC.8).

a. Consider how the assurance agreements can be operationalised, using FSP-1's defined user rights (see KPIs) and wider permissions logic, based on the unique IDs provided by the system.

This is a question to be defined in the detail designed in a later stage. It is a matter of standards agreed by DSI or relevant party and being applied/enforced by the market facilitator.

Our underlying assumption is that data definitions are defined in an automatable format, defining data structures and validation rules that can be enforced by registration and other services (e.g. the collection of asset data is driven by requirements of specific products the asset is bidding for, so there needs to be negotiation between the asset and the product service to get the data needed). But we have not defined the mechanism for doing this - we'd use standard schema already available on the market, rather than a bespoke internal capability.

Searchable directory:

5. FSP-1 searches directory of other users to identify MO-1, MO-2 and MO-3 IDs and express interest in market exploration.

- a. Consider how permissions-based secure messaging channels for user notifications could be enabled by the system.

Only the market operators running the market/services the FSP expressed interest in can – that's the association happens.

Implies that users have roles associated with their registration - they register as an FSP or an SO or etc. They could have multiple roles (e.g. an SO might act as FSP to another SO, as with CLASS). But the roles need to be recorded into the register.

Users are registered against roles, as above, so messaging service (Electralink's proposed messaging solution) can use this to inform what permissions need to be enforced.

Permissions may also be driven by assets — users are allowed to see updates for assets that they have a legitimate interest in (e.g. as they own or represent them or receive a service from them). This requires the asset register to record a list of all parties with a legitimate interest in them, so the messaging service can use that to decide what permissions need to be enforced.

This could also enable things like geofencing — notifications to everyone who owns or represents an asset in a specific location (e.g. that a DSO wants to buy services there). Users probably need to opt into the type of notification that want to receive when they register, with capability to update this subsequently.

Commercial interoperability across markets:

6. The system provides MO-1, MO-2, and MO-3 a means for accessing pre-authorised shared FSP-1 data used for initiating the Registration stage.

- a. Consider how controls on the commercial information MOs are authorised to access could be introduced.

Can be built into permissions discussed above.

- b. If multiple systems are able to issue unique IDs, consider how MOs know which system the FSP information is held on.

n/a

7. FSP-1 is notified of further action needed on their behalf to then initiate contractual agreements.

FSPs can sign up to the MOs of their interest by indicating it in the MPp of their preference.

Seamless integration with BUCs:

8. The system is able to seamlessly use and integrate all user unique ID outcomes (described in Steps 1-7) into the system used to deliver BUC.2 (Common Asset Registration) and wider BUCs.

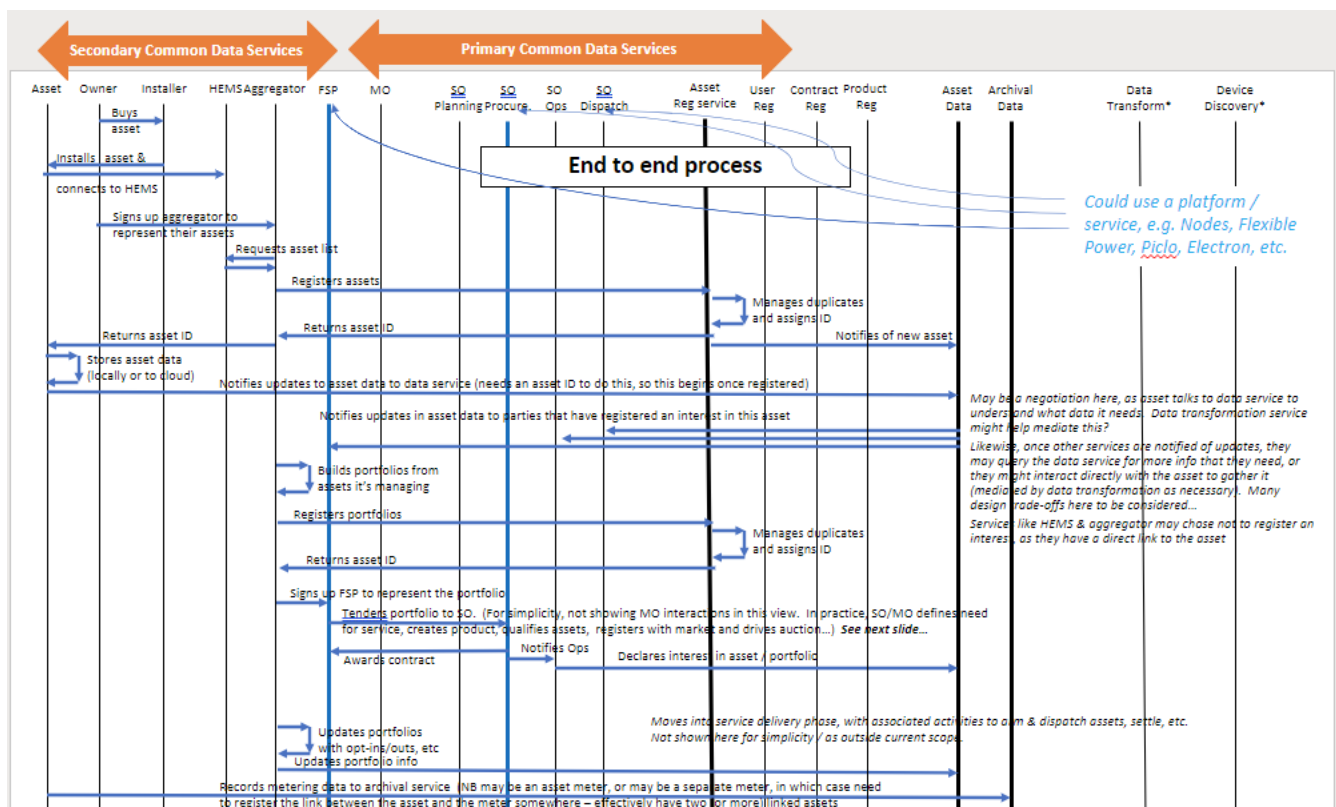
As described above, user IDs (owner, FSP, aggregator) need to be recorded against assets when they are registered / updated. All services can then use APIs onto the **Common_user_data_services** to get and update user info as necessary (and subject to permissions, as above).

9. We identify an additional step in the scenario where changes are notified to the relevant parties is missing. The functionality of bidirectional data exchanges so that FSPs can update important corporate information once instead of having to do it multiple times notifying multiple flex buyers (M.Os, S.Os and/or MpPs) for example changes in bank details, would be highly valuable for aggregators and FSPs.

* Important final note: In parallel to the steps in the process above, there is the relationship between FSP/aggregator and Flex Owner. Whilst this is a prerequisite for an FSP to bid for services in the market - flex owners and assets need to be registered with the FSP - but not for FSPs to be registered with the system. There is also a need to clarify how users and assets are qualified by aggregators:

- For users: will need some basic proof of identify, and address.
- For assets, needs proof that it exists and what it can do, e.g. by connecting to the assets and collecting data from them, and of their id, e.g. from serial numbers or some such. May also want a link to MPAN, locational info, etc.

At this point, there is also a need for a 'second level' user (and asset) check to ensure a user is not registered already with another FSP/aggregator for the same service. For this, we envision a system FO-FSP_common_register which offers a verification and deduping service. This system is currently being built and tested by ElectraLink in collaboration with FSPs.



BUC.2: Common Asset Registration

Narrative of the System Use Case

The SUC proposed is a Common Asset Data Services with a lightweight Asset Registration service that focuses on de-duplication and assigned unique asset IDs then a heavier Asset Data service to handle other data, with a publish/subscribe dissemination mechanism. The Asset Data service would be composed of an addressing/directory front end, then sharded services behind it doing the heavy work. We are suggesting sharing by asset location and type. The benefit of this structure is that you can re-arrange implementation of the sharding as the number of assets grows without affecting the endpoint presented to service users. It may make sense to market participants that other core data services are part of the foundational infrastructure to help make the data necessary to resolve stackability potential conflicts visible.

As well as this core data about assets and their ownership, capabilities, etc, there would also be value in a service that records archival / immutable data such as time series of meter readings need for operational and settlement metering, etc. This could be provided by a separate service to which the asset or its associated metering systems send the relevant data as it becomes available. As with other data, the level of data recorded could be negotiated between the asset (or the agent acting on its behalf) and the product service which defines what data is needed for that product. The asset would then provide this data to the Archival Data service shown on the sequence diagrams. That service would then notify subscribers with a legitimate interest in the data of its availability.

Use Case conditions	
Assumptions/Pre-requisites	
1	Seamless integration utilising the Data Sharing Infrastructure (Trust + Prepare + Share) outcomes defined in BUC.1 and BUC1.1.
2	Relevant data- and entity- assurance agreements are defined as part of BUC.1 and/or BUC.8 and are readily implementable by the system.
3	Information flows utilise a necessary common data standard and wider IT architecture support the functions, defined in BUC1.1.
4	Seamless integration to utilise common user registration outcomes in BUC.4.
5	Seamless integration to enable common pre-qualification outcomes in BUC.7.
6	Seamless integration to enable common TSO-DSO coordination outcomes in BUC.6.
7	Seamless integration with relevant common compliance tools in BUC.8
8	Asset details submitted to the system are accompanied with a mechanism for validating owner consents.
9	Asset details are validated according to a transparent and well-defined logic.
10	e.g. Approved BUC.2 platforms have BUC.4 system integration directly embedded and do not require a UserID to be sent by BUC.4 system separately.
11	e.g. Simultaneous updates are resolved by a defined set of rules implemented by the CoordinationSystem_BUC.2
12	

13	

Actor name	Actor type	Actor description
Flex-owner	Individual/ Business	Any actor from a householder to a small wind turbine owner, to large I&C. It is the actor that literally produces the flexibility by altering their generation or consumption. In most cases, these actors will provide flexibility services only through an aggregator, although it may be that this evolves to some actors providing flexibility directly to SOs without going through an aggregator.
Aggregator	Business	Their speciality is in recruiting and communicating with customers and in integrating the assets they own into portfolios that can interoperate with energy system services. This is a very distinct capability c.f. the market-facing capability of FSPs.
Flexibility Service Provider (FSP)	Business	FSPs manage the qualification and administrative processes required by the SOs, which can be quite burdensome (e.g. to become a VLF BM). They also manage real time trading across the multiple markets / services, i.e. they are primarily market-facing entities.
Marketplace Platform (MPps)	Business	Marketplaces for trading flexibility services, existing platforms such as NODES, Flexible Power, Piclo, Electron, desX, etc.
Markets Operator (MO)	Business/system	Entities that facilitate the end-to-end flexibility service delivery. *Not to be confused with a Market.
System Operator (SO)	Business	Entities that buy flexibility to operate the power system e.g. ESO, DNO. In our examples below we have focused on the SO as a single entity, not exploring the different roles within the SO noted above. But ultimately these roles will need to be recognised to manage data access permissions effectively.
FDI-user	Business	Data producers and consumers verified within the FDI ecosystem, e.g. MO, SO, FSP and Special Users
Special Users	Business	Other entities verified within the FDI ecosystem such as the regulated investors or third-party service provider with an interest in flexibility services.
Common_Asset_Data_Services	System	A new system, a lightweight registration service that focuses on de-duplication and assigned unique asset IDs, with potentially Asset data services, and the services required that could be seen as part of the foundational infrastructure to help make the data necessary to resolve stackability potential conflicts visible. There is currently not an existing system as such, although there are existing systems and moving parts that can be brought together relatively fast to pivot into the required system. E.g. The underlying messaging system of a national service.

		<p>The AAR and CAR programs have received innovation funding and very likely to be capable of providing this functionality. The AAR, CFAIR solutions can be pivoted into systems.</p> <p>Primary user register service.</p>
Archival_data_service	System	A service that records archival / immutable data such as time series meter readings need for operational and settlement metering, etc
Contract_registration_service	System	Core data system to support compliance, reliability, and conflict visibility.
Product_registration_service	System	Core data system to support compliance, reliability, and conflict visibility.
Data_transformation_service	System	Potential services that could be built on top of the common-foundational data services.
Device_discovery_service	System	Potential services that could be built on top of the common-foundational data services.
FO-FSP_common_register	System	<p>A central register to flag potential conflict on asset service delivery being registered with multiple FSPs. This requires registers of products and contracts, as it needs to identify situations such as "Asset-1 is part of Portfolio-1 that is delivering Product-1 to SO-1 under Contract-1. It is also part of Portfolio-2 that is delivering Product-2 to SO-2 under Contract-2. These contracts mean that there will be overlaps in service provision under the following circumstances. The product definitions mean that these services are not stackable under these circumstances.</p> <p>Second-level asset register service.</p>

* As indicated in the table above, there is a need for additional processes and services that can and should be built on top of the foundational infrastructure to ensure that the system is accounting for complexities that can massively impact operation, delivery, and settlement of flexibility services. A great example of this is the increasing discrepancies between meters and sub-meters.

Scenario

* Note: In parallel to the steps in the scenario, there is the relationship between FSP/aggregator and Flex Owner. Whilst this is a prerequisite for an FSP to bid for services in the market - flex owners and assets need to be registered with the FSP - but not for FSPs to be registered with the system

There is also a need to clarify how assets are qualified by aggregators:

- For assets, needs proof that it exists and what it can do, e.g. by connecting to the assets and collecting data from them, and of their id, e.g. from serial numbers or some such. May also want a link to MPAN, locational info, etc.

NB assets may have multiple aggregators / FSPs as noted above in BUC.2 scenario - this may or may not be allowed for different products. But FSPs need to be aware of this, so this is still a necessary query from a [Second-level_asset_register](#) or [FO-FSP_common_register](#)

At this point, there is also a need for a 'secondary' asset check to ensure a user is not registered already with another FSP/aggregator for the same service. For this, we envision a system **FO-FSP_common_register** which offers a verification and deduping service. This system is currently being built and tested by ElectraLink in collaboration with FSPs.

1) FSP-1 provides asset data to the system (or systems).

- a. How is the system is integrated with the system(s) necessary to deliver common user registration for FSP-1.?

Assets are owned and managed by users, so one attribute recorded against each asset will be a list of the users associated with it and their role in relationship to it (owner, aggregator representing it to the market, SO buying services from it, etc). The user and registration services are largely independent, but asset registration requires that whoever registers the asset and makes updates to it be a registered user.

Exploring some of the detailed considerations here:

The FSP-1 will register an asset/portfolio in the **Common_Asset_Data_Services** either through a single end point, or an end point that can be delivered in a decentralised way by multiple market platforms, depending on the resolution of the Flexibility Market Facilitator role.

An ultimate thing to think about when designing more in detail is which/how much data is allocated to the **Common_Asset_Data_Services** and what data falls within heavier data category. One of the main use cases for a **Common_Asset_Data_Services** is for de-duplication. We envision the de-duplication service as part of the asset data service and a key element will be to flag duplicates within each unit or portfolio and decide if reject or not based on current market stackability criteria. Thus, the data collected by this **Common_Asset_Data_Services** cannot be the minimal ID and capacity, because at this simplest state you can't recognise duplicates.

The **Common_Asset_Data_Services** then returns Asset ID (a portfolio is simply a type of asset – the ID will embed info as to what type of asset it identifies, so there will need to be a common taxonomy of asset types).

Detailed design question includes the extend of the detailed needed and whether it should be a cached view. Our underlying principle is that the ultimate store of information about an asset is the asset itself, or some close proxy such as the OEM's cloud. That will always have the best, most up-to-date info about the asset. The Asset Data service can therefore conceptually be seen as a cache of this data, making it easily accessible to other market participants, subject to permissions, etc. Hence the pub/sub model to disseminate changes to asset data.

Within this, the central service only contains the set of data necessary to identify the asset and qualify it for the services it is contracted to deliver/ bidding for. Other data about the asset may be held by aggregator, FSP, etc, if they need it.

Heavier asset data can be hold at FSP, aggregator level until the point where the asset or portfolio is allocated to a **product**, at which point it will need to go through another more rigorous check.

The actors involved, namely FSP-1, FSP_n, MO_n, MpP_n are able to interface with **Common_Asset_Data_Services** either by API integration or user interface as needed.

- b. Consider how the system can ‘signpost’ the necessary data requirements until pre-qualification, for a given MO.

The static data collected, stored and managed by the **Common_Asset_Data_Services** should be defined and there is currently already work being done in this area at Open Networks. Same as for BUC.2, it is discussed above.

Data requirements may also be embedded in product definitions (e.g. some products will require data on asset ramp rates, others won’t). The **Common_Asset_Data_Services** might therefore need “negotiate” with the product service to determine what data is needed from an asset in order to bid for a given product. This negotiation would be handled similarly to that for BUC.2, as mentioned above.

- c. Consider how interactions across multiple potential data access points (i.e. asset owners, installers) or databases (i.e. technology vendors, existing FSP or MO registries) will be supported.

The **Common_Asset_Data_Services** communicates with **MOs** system and with **FSPs** systems through standardised interoperable. Same as for BUC.2, it is discussed above.

2) The system validates the technical parameters for the data provided by FSP-1.

- a. Consider how validation using multiple trusted asset databases (e.g. OEM cloud platforms) will be handled.

We envision a system where data is captured based on the flex product being delivered into the **Common_Asset_Data_Services**. The **Common_Asset_Data_Services** initially believes whatever the FSP tells, marks the data as provisional and validation happens afterwards, after the asset/portfolio are linked into a product/service at which point the data is marked as confirmed. This would happen by going to archival data and other sources available.

The benefit of this approach is that data capture doesn’t need to be delayed by validation, so validation can be made as thorough as needed by FSPs. This is particularly beneficial for flexibility procurement in longer terms and particularly locationally influenced, where the flex recruitment happens after the requirement.

The cost is that data is provisional for some period until validation catches it.

- b. Consider how assets can demonstrate valid data (e.g. by virtue of existing participation in flexibility markets) and circumvent/expedite this step.
- c. Consider how validation of planned assets could be supported.

This is mostly covered in the 2.a. Validation would take place by going to archival data and other sources available. This will be heavily dependent on the product technical requirements, for which some are very

lightweight (e.g. DFS) and in some other cases very detailed and specific with technical tests required (e.g. dynamic containment).

- 3) The system validates the contractual parameters (i.e. right to operate for a given period) for the data provided by FSP-1.
 - a. Consider how the system could surface data needed to reconcile conflicting contractual claims by multiple FSPs to the asset and ensure only one operator for it at a given moment.

As discussed above, by identifying duplication of assets across flex services. Also, the discussion in the scenario about negotiation between product service and asset as to which data needs to be provided for a given product.

This would be ensured by the de-duplication service the **Common_Asset_Data_Services** would carry by also carrying a check on the units that make up for each portfolio.

- 4) The system registers the validated data to the dedicated asset record, ensuring each asset has a unique identifier.
 - a. Consider how the unique asset ID paradigm would be maintained should another Registered User, FSP-n, attempt to provide data for an existing validated asset.

This would be avoided by having only one system, the **Common_Asset_Data_Services** and not MOs or SOs, issuing unique IDs that would be first checked not only at biggest unit level (portfolio) but also backed by the **secondary check/ FO-FSP_common_register**.

If an asset is registered against multiple FSPs, e.g. for different flex services, then they can each provide data about it. In most cases, this won't create any conflict — FSPs will just update asset data relevant to their service. The **Common_Asset_Data_Services** can enforce that FSPs only update the data relevant to their services.

If it does create a conflict, e.g. one FSP incorrectly overwrites data provided by another FSP (for data common to both services), then the second FSP will see this when they are notified of the update via the pub/sub service. This will be relatively rare and could in the first instance be for the FSPs to resolve — only they know just what they are trying to do with the asset. (Please note that the asset is the ultimate reference point for these discussions).

- b. If multiple systems are used throughout steps 1-4, consider how 4a can reliably be achieved.

n/a. NB as mentioned above, the **Common_Asset_Data_Services** would only collect, store and manage static, slow changing data enough to identify duplications. Heavier, more detailed data of the assets would be collected, stored and held as per required per type of asset by **MpPs** and/or **FSPs** and/or **aggregators**.

- 5) The system confirms the registration of validated data with FSP-1.
 - a. Consider how FSP-n is then notified of attempted duplicated asset registration.

Pub/sub handles this - attempts at registering an asset are something that would change the asset data and so be notified to users that have an existing interest in the asset.

Asset record accessed by MOs:

- 6) Registered Users MO-1, MO-2 and MO-3 are able to access the registered asset data for use in their procurement systems.
 - a. Consider how FSP-1 could be notified if a given market operator, MO-n, accesses data of an asset that they operate.

The **Common_Asset_Data_Services** would notify interested parties that have subscribed to a portfolio (subject to security & privacy permissions). This could be handled by pub/sub. Need to consider the load of notifications this might create - only certain types of access need to be notifiable, and parties need to be able to opt in/out of each type of notification.

- b. If multiple systems are used throughout steps 1-5, consider how MO-n identifies and accesses the system that the registered data is held on.

Asset record updated by FSPs:

- 7) FSP-1 provides updated asset data to the system.
 - a. Consider how interactions across multiple potential data access points (i.e. asset owners, installers) or other databases (i.e. technology vendors, existing FSP or MO registries) will be supported.

FSP_n will interface with the **Common_Asset_Data_Services** to register primary information, as we have referred to, static data. MO_n and/or MpPs will primarily fill information with **Common_Asset_Data_Services** but could also interface directly with FSPs for additional data. They would all be talking to the common service **Common_Asset_Data_Services**, which would manage the permissions, conflicts updates, etc.

We are not addressing this step in too much beyond until there is further clarity on market operators figure and functions.

- b. Consider how the system would handle a situation where FSP-1 and FSP-n provide updated asset data simultaneously? Consider how the system would reconcile divergent asset data provided simultaneously?

Discussed above.

- c. If multiple systems are used for asset validation and registration, consider how the unique asset ID paradigm would be maintained if an asset was updated from a different access point than was originally registered from.

n/a

Unexpected system downtime:

8. The system faces unexpected downtime during a process such as validation and registration of FSP-n. d. Consider what measures need to be in place for data recovery and system resilience?

e. If multiple systems are being used and they have technical interdependencies, consider what additional features need to be in place.

Effectively preventing unexpected downtime in any system involves establishing a set of robust processes that make downtime a rare occurrence. Achieving high availability requires leveraging cloud architecture to decouple compute from storage, enabling independent scaling. Cloud services including load balancing, auto-scaling, and redundant systems across multiple availability zones ensure fault tolerance and avoid the issue of regional outages which improves overall system resilience. Additionally, implementing solutions to mitigate risks, such as Distributed Denial of Service (DDoS) attacks, helps fortify the system against external threats.

In tandem with the data architecture, automated monitoring systems play a crucial role in ensuring services operate within acceptable resource utilization ranges. Continuous monitoring allows for proactive detection of performance anomalies, triggering automated remediations like scaling during peak times or restarting failing components based on predefined patterns.

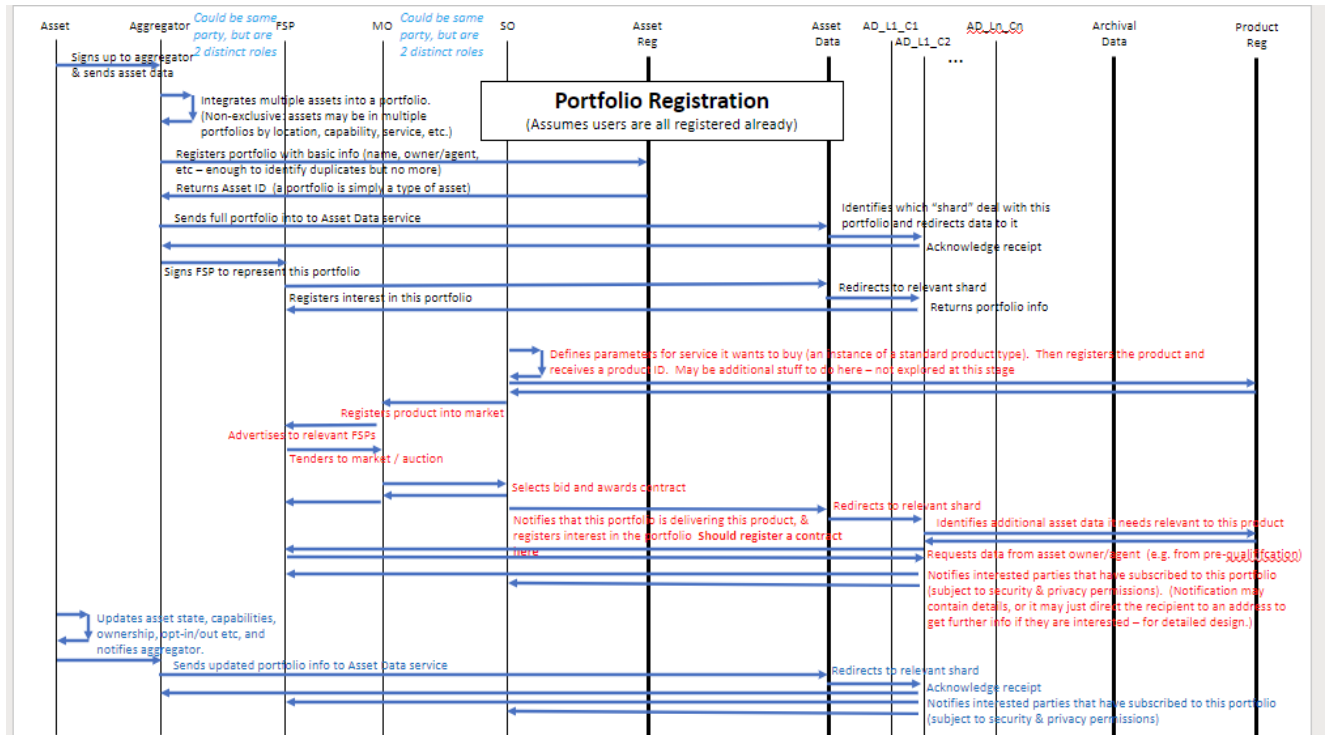
In addition to technical solutions, having a dedicated standby team which already operates similar critical infrastructure further enhances the resilience of a system. The team would enable proactive monitoring, rapid response, and efficient incident resolution. Specialists equipped to detect potential issues before they escalate, ensuring a swift and coordinated response to emerging challenges. The standby provides continuous optimisation, fine-tuning the system's performance, and addressing potential vulnerabilities in real-time. A dedicated team also contributes to ongoing training and knowledge transfer within the ensuring the service is fit for purpose over a series of years.

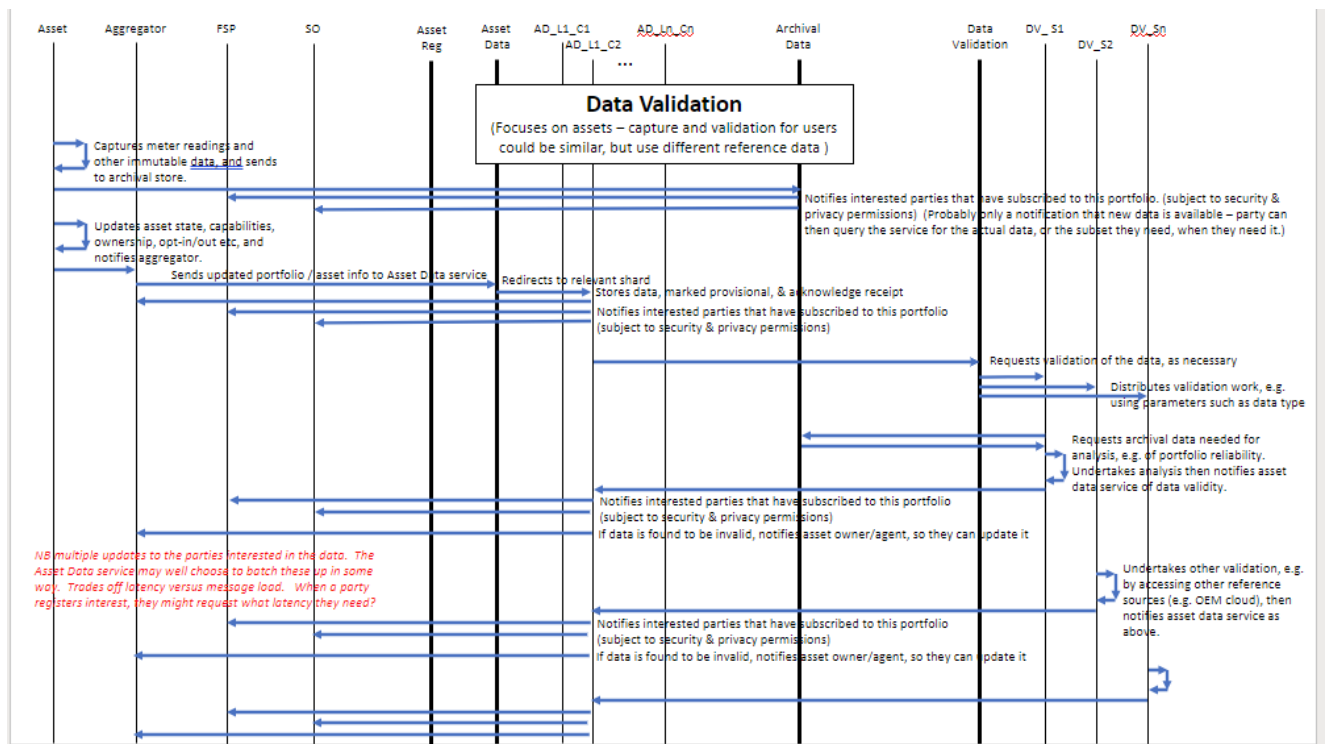
Given the complex nature of services involving multiple systems with technical interdependencies, adopting an event-driven architecture becomes advantageous. Utilizing a message queue and configuring channels based on event types enables the system to respond dynamically. This approach allows for message replay in case of system failures, ensuring data integrity and system stability over extended periods of downtime. The loose coupling of components in this event-driven model allows the system to maintain operation in the presence of failures in specific areas. Furthermore, by decoupling the message payload from the message, scalability is enhanced, contributing to the overall efficiency of the service.

Seamless integration with BUCs:

The system is able to seamlessly use and integrate all of user common asset registration outcomes (described in Steps 1-8) into the system used to deliver wider BUCs.

It is discussed above.





Non-functional requirements

Aiming to delve more into the non-functional requirements at a later stage, we flag at this stage the requirements we believe are critical for the successful delivery of a critical national service immediately and in the long run, which include:

Scalability – ability to handle growing transaction load.

Reliability & resilience

Extensibility – ability to add and adapt functions over time.

Security

Privacy

Usability

- Clean, versioned APIs, so service developers can access the register.
- User interface service, so asset owners and other parties can look into the register if necessary.

Appendix 1

Additional relevant information

Information layer:

NB information requirements can be a strong driver of functional service structure.

What data gets stored where?

- Reference point for asset & user is likely to be the entity itself (or a closely aligned cloud service, e.g. manufacturer cloud). No other place can provide up-to-date info with same level of assurance.
- But these entities may not be easily accessible – offline, behind firewalls, not set up for large query load, etc.
- Suggests a caching strategy

Asset lifecycle

- Assets will change over time – performance will degrade, they'll be maintained & upgraded & retired, ownership will change, location will shift, they'll be contracted with multiple aggregators & FSPs (serially and potentially in parallel as service stacking becomes more refined, or even as owners ask multiple parties to take them to market – joint agency model).
- This affects stuff that may be in the registers, e.g. asset ownership & capacity, either as it's needed in its own right or because it aids de-duplication. Even if not directly needed, it will be essential for planning, monitoring service delivery, etc, so it needs to be easily accessible for relatively high query loads.
- So the model needs to accommodate a way to capture and disseminate these changes.

Housekeeping functions

- Managing duplication. Aim to prevent assets being given multiple IDs. But it will almost certainly happen as assets evolve, shift between portfolios, etc. So need ability to match and de-duplicate too.
- Asset discovery. Assets will appear on the network without registering, either because there's no drive to register or through lags in recruitment processes. Do we want to maintain visibility of these assets?
- Others will emerge.

Contracts:

- Products are linked to assets and asset owners via contracts. Understanding these may be critical to understanding issues of duplication, service stacking, reliability of service delivery, etc.
- Contracts may themselves be multi-layer. SO writes a contract to buy flex from an FSP. FSP then writes a contract with one or more aggregators to create that flex. Each aggregator contracts with asset owners, probably via simplified / user-friendly T&Cs.
- The terms of each contract may vary, e.g. because commercial contracts from SO are not suitable to pass through to domestic asset owners, or because the aggregator and/or FSP can manage some risks at portfolio level rather than at individual asset level.