

Flexibility Digital Infrastructure System Use Case exercise

Piclo initial response

Jan 2024

(public version)

Piclo overall summary

Cons to centralised commercial & asset registers

- Data inaccuracy: A single asset registry would require users to periodically update data, but they may be participating in the market more frequently than they update the registry, rendering it inaccurate.
- Poor data completeness: A single asset registry would likely require users to update large amounts of data without knowing what system will use it, or why. It may miss information that some market stakeholders or systems require, because the users haven't submitted it.
- Project delivery: Infrastructure projects, particularly IT infrastructure, commonly overrun their time and financial budgets. Cost and time overruns are even more frequent for projects that have little or no antecedents, because past experience cannot be learned from. There are no other single asset registries in the world with a track record of operation, therefore, it is more likely that this project overruns.
- System resilience: A single centralised system is a single point of failure for the system, whereas relying on individual data sources spreads the risk.

'Just-in-time' data provision, through common API standards

Instead of a single, centralised register, we propose a decentralised system in which market participants share data to other participants and market owners, at the point of use.

1. SUC Template for BUC.4

Narrative of the System Use Case
Short description
<i>Written description of your SUC implementation of the BUC. Describe the SUC operation and what new/existing systems are involved and what system functions are used to deliver the BUC. Describe any aspects of the BUC narrative or KPIs or scenario that your SUC implementation does not meet. Optionally, please also include any overall architectural diagrams.</i>
Piclo response
<ul style="list-style-type: none">• Common user registration implies too much centralisation for high effort, and limited value• Different market operators require different commercial information• When a Market Operator verifies each entity, a unique ID and certificate must be generated• This can be used to register at other Market Operators which verify the same information• A common data exchange standard with specified fields (similar to Open Networks procurement process TWG outputs) makes data sharing viable, and should be determined in BUC. 1• A governance framework for Market Operators would mandate:<ul style="list-style-type: none">○ Unique IDs generated for each FSP, and linked to each market, preventing duplicate registration (for example Market A-1594)○ Requirement for market operators to send certificate of commercial information on request to FSP• In this diagram, the case of an FSP attempt to register multiple times is not considered. However, either all Market Operator could communicate new entries to a master record held by each Market Operator (fully distributed ledger) or request copies of any existing records from each Market Operator (distributed database).

2. SUC Template for BUC.2

Please use this template (based on [IEC standards](#)) to set out your SUC proposals which deliver the BUC narrative and KPIs, and address the scenario provided above. You may find the [PlantUML website](#) tool useful for making sequence diagrams (tutorial seen [here](#)), but diagrams created in Word/PowerPoint (or equivalent) are entirely acceptable.

Narrative of the System Use Case
Short description
<i>Written description of your SUC implementation of the BUC. Describe the SUC operation and what new/existing systems are involved and what system functions are used to deliver the BUC. Describe any aspects of the BUC narrative or KPIs or scenario that your SUC implementation does not meet. Optionally, please also include any overall architectural diagrams.</i>
<p>Piclo response:</p> <p>Currently, FSPs have a changing portfolio of growing assets - particularly aggregators, who are often managing assets on behalf of other end users.</p> <ul style="list-style-type: none">• Whilst assets themselves do not often change, their ownership and operational parameters can vary frequently based on the aggregator of the asset at that particular time.• Therefore, it is unrealistic to design a centralised system which requires assets to be updated only once.• Furthermore, different services require different technical parameters. Requiring FSPs to provide all information as required for all services is an unrealistic participation barrier.• Rather, asset registration should be a simple and efficient for all parties, and provide the right data at the right time, for the right user with appropriate permission.• It is the responsibility of the FSP to ensure accurate data at point of service, and the Market Operator to ensure accurate data at point of qualification. Penalties should be imposed on either party which does not uphold this responsibility.• FSPs are the data owners for these assets and wider market permissions to read and write asset information should not be granted. The problem this solves should have alternative solutions considered.• A decentralised system should exist with common data exchange standards specifying:<ul style="list-style-type: none">○ Universal formats and fields for each asset parameter○ Common API fields and formats• A governance framework for market operators should mandate:<ul style="list-style-type: none">○ Unique asset ID to be generated by each market operator○ Primacy rules to govern order of service priority○ Availability (notices, declaration, or other mechanism) enables conflicting services to determine whether conflicts exist <p>Challenges and tradeoffs</p> <ul style="list-style-type: none">• This assumes a protocol for fast, accurate retrieval of asset information between Market Operators, which would involve at least robust API development from all parties.• However, this decentralisation is more robust to the overall system, as it reduces a need for the FSP to consistently update central systems, and improves resilience.

- Availability is the mechanism by which service conflicts can be prevented. However, this does not specify the rules for resolving conflicts.
- This also assumes an extremely low latency of API connection between services. For services requiring a sub-second response time, it would not be feasible to enable conflicts to be resolved in this way.

Diagram(s) of the Use Case

Please include sequence diagram(s) working through the scenario steps to show how they are implemented in the SUC proposed.

Piclo response

[redacted]