Flexibility Service System Interoperability

Review of options around APIs and Standards for the Dispatch of Flexibility Services

Open Networks
October 2024 | Version 1.0

DOCUMENT CONTROL

Authorities

Version	Issue Date	Authorisation	Comments
1	08/10/2024		

Related documents

Reference 1	*Flexibility Services Interoperability : D3 Evaluation Matrix, October 2023*

Change history

Version	Description
1.0	First distributed version

TABLE OF CONTENTS

Contents

Introduction

"Build vs Buy"

Introduction

About ENA

Energy Networks Association represents the companies which operate the electricity wires, gas pipes and energy system in the UK and Ireland.

We help our members meet the challenge of delivering electricity and gas to communities across the UK and Ireland safely, sustainably and reliably.

Our members include every major electricity and gas network operator in the UK and Ireland, independent operators, National Grid ESO which operates the electricity system in Great Britain and National Grid which operates the gas system in Great Britain. Our affiliate membership also includes companies with an interest in energy, including Heathrow Airport and Network Rail.

We help our members to:

- Create smart grids, ensuring our networks are prepared for more renewable generation than ever before, decentralised sources of energy, more electric vehicles and heat pumps. Learn more about our Open Networks programme.
- Create the world's first zero-carbon gas grid, by speeding up the switch from natural gas to hydrogen. Learn more about our Gas Goes Green programme.
- Innovate. We're supporting over £450m of innovation investment to support customers, connections and more.
- Be safe. We bring our industry together to improve safety and reduce workforce and public injury.
- Manage our networks. We support our members manage, create and maintain a vast array of electricity codes, standards and regulations which supports the day-to-day operation of our energy networks.

Together, the energy networks are keeping your energy flowing, supporting our economy through jobs and investment and preparing for a net zero future.

About Open Networks

Britain's energy landscape is changing, and new smart technologies are changing the way we interact with the energy system. Our Open Networks programme is transforming the way our energy networks operate. New smart technologies are challenging the traditional way we generate, consume and manage electricity, and the energy networks are making sure that these changes benefit everyone.

ENA's Open Networks programme is key to enabling the delivery of Net Zero by:

- opening local flexibility markets to demand response, renewable energy and new low-carbon technology and removing barriers to participation
- opening data to allow these flexible resources to identify the best locations to invest
- delivering efficiencies between the network companies to plan and operate secure efficient networks

We're helping transition to a smart, flexible system that connects large-scale energy generation right down to the solar panels and electric vehicles installed in homes, businesses and communities right across the country. This is often referred to as the smart grid.

The Open Networks programme has brought together the nine electricity grid operators in the UK and Ireland to work together to standardise customer experiences and align processes to make connecting to the networks as easy as possible and bring record amounts of renewable distributed energy resources, like wind and solar panels, to the local electricity grid.

The pace of change Open Networks is delivering is unprecedented in the industry, and to make sure the transformation of the networks becomes a reality, we have created three workstreams under Open Networks to progress the delivery of the smart grid.

2023 Open Networks programme Workstreams

- Network Operation
- Market Development
- Planning and Network Development

Our members and associates

Membership of Energy Networks Association is open to all owners and operators of energy networks in the UK.

- Companies which operate smaller networks or are licence holders in the islands around the UK and Ireland can be associates of ENA too. This gives them access to the expertise and knowledge available through ENA.
- Companies and organisations with an interest in the UK transmission and distribution market are now able to directly benefit from the work of ENA through associate status.

ENA members

BUUK infrastructure	Cadent Your Gas Network	electricity north west Bringing energy to your door	ESB NETWORKS	nationalgrid
ESO	Northern Gas Networks	Northern Ireland Electricity Networks	NORTHERN POWERGRID	Scottish & Southern Electricity Networks
SGN Your gas. Our network.	SP ENERGY NETWORKS	UK Power Networks Delivering your electricity	WALES&WEST UTILITIES	

ENA associates

- Chubu
- EEA
- Guernsey Electricity Ltd
- Heathrow Airport
- Jersey Electricity
- Manx Electricity Authority
- Network Rail
- TEPCO

Assessment of Flexibility Interoperation

This document accompanies a spreadsheet to enable objective assessment and evaluation of options around APIs and standards for dispatch of flexibility services, as part of Deliverable D3. This work culminates delivery of a sequence of three deliverables to ENA, building upon extensive stakeholder engagement activity, feeding into a holistic gap analysis and minimum requirements capture activity. Based on specific arising requirements from ENA, an annexe to the minimum requirements document was also included, discussing security considerations in more detail.

In considering and carrying out the analysis of these options, considering the evaluation criteria, it is important to note that, at present, given the lack of a security architecture and defined solution architecture, it will not be possible to give each option a complete score. This is because, as has been previously illustrated through D2 (based on requirements elicited from D1), there are several dependencies which need to be satisfied to permit objective assessment of flexibility dispatch options holistically.

In taking feedback from DSOs based on D2, there is a clear identified requirement from the industry to ensure that there is greater detail around the non-API and non-standard based areas of flexibility dispatch. This is because such components sit within a wider framework for an interoperable system, and wider business requirements will drive technical decisions around protocols, standards, and interfaces.

We deliberately use the term “system” below, to recognise that a flexibility dispatch platform is one important component of a wider flexibility ecosystem, comprising an interoperable technical system to enable DSO to FSP communications across the full lifecycle of flexibility business processes. D2 has set out the requirements identified from the gap analysis of D1, around wider business process integration and interoperability. It is important to note that no currently available standard meets the requirements here, nor integrates with the (as-yet undefined) wider flexibility ecosystem.

Considerations in selection of an API standard

Using the following high-level categories of requirements, the following should be kept in mind when considering and scoring options (as described later in this document). These consider the evaluation criteria set out by the ENA’s 2023 scoping proposal paper.

Performance

To evaluate whether options deliver suitable performance, a set of detailed non-functional requirements will need to be developed. It is not possible to define blanket performance criteria for a standard or API alone, since the performance criteria need to be defined based on the architecture of an implementation. It would be possible to define blanket system-level performance criteria, which would encompass the full end-to-end implementation.

Based on our requirements capture and gap analysis, we did not identify any significant performance requirements which are likely to pose an issue to a modern implementation of an API. Specifically, flexibility services are dispatched ahead-of-time (with day-ahead an aspiration for some DSOs, and week-ahead being the rough approach used by a common platform used today). This reduces the sensitivity of the overall system (end-to-end) to performance issues.

Despite this, there are 2 scenarios where performance requirements will emerge, in translating from an API or standard into an implementation.

The first scenario is where, as set out in the Flexibility Systems Interoperability tasking for this work, there could be significant changes to the flexibility market in subsequent years, in order to adapt to market changes, or the introduction of new services or operating models. It is conceivable at this stage that, in future, flexibility services could be dispatched closer to real-time, and this would introduce additional system-level performance requirements.

Performance requirements could also arise to support integration with other business processes in the wider flexibility ecosystem. For instance, if flexibility service providers were to (in future) be required to provide telemetry or status updates on their delivery of flexibility services in real-time, even following a day-ahead dispatch, there would be a performance requirement for delivery of this information to the dispatching system operator promptly and with an agreed latency.

The second scenario where a performance requirement may emerge is in the implementation of a communications protocol. As part of implementation of most internet-based protocols, there will be stateful or bidirectional communications (such as handshakes and key exchanges). These will introduce performance requirements, to ensure that these operations are completed before either side of the interface "times out" the connection (in the same way that on a slow internet connection, a web page may fail to load due to the server timing out waiting on the client). In addition, if the resulting system-level implementation features stateful messaging exchanges, there will generally need to be a state time-out, in order to allow the server to "forget" about previous stateful exchanges to avoid resource exhaustion and vulnerability to denial-of-service attacks. This will not be a primary performance requirement driver, but it illustrates how at system-level the performance of communications links will ultimately have hidden requirements.

We anticipate that FSPs and DSOs will see some performance requirements around efficiency of communications, especially where FSPs are offering assets for dispatch over metered network connections (such as public mobile networks), although this is likely to be a downstream consideration for the FSP or aggregator's own internal communications from their own dispatch system to assets, since a DSO would not be directly dispatching a field-deployed asset over the mobile network.

Open Standard

The openness and freedom for FSPs and others to implement whatever is selected for flexibility dispatch is a key requirement that has been captured. FSPs are generally keen for an open standard, as the key properties of openness reduce legal risk for them (i.e. intellectual property rights and royalties being required for implementing the standard), as well as reducing barriers to development.

From a commercial perspective, open standards are more likely to yield better value for bill-payers, through efficiencies in implementations (such as open-source implementations), and are likely to reduce the likelihood of a "single point of failure" or "single source" vendor option for an implementation of a flexibility dispatch system.

We have considered in the evaluation criteria that openness should also encompass the governance and accountability around the evolution of any relevant standards, on the basis that a key property of an open standard is that it can evolve to meet the UK's future energy system requirements going forwards, and hopefully avert the requirement to carry out a costly change in future. A key property of an open standard includes openness of participation, and we thus highlight this as a key requirement. One easy way to determine this is to look at relevant standards and ascertain whether they are led and governed through a recognised Standards Development Organisation (SDO), such as IEEE, IEC, BSI, ISO or similar – these organisations generally have publicly accessible policies around participation, access, governance, and intellectual property rights and patent disclosures.

At this point, we must note and emphasise that a standard is a written specification of how the world should work. It is a technical document, and a standard in and of itself is not an implementation. A standard can define any other aspect of a wider system – for example, there are standards around business processes (ISO 9001, ISO27001), as well as standards for protocols (RFC standards for web protocols such as HTTP, TLS, etc.), and architectures. As such, when considering a standard, or any other kind of specification, it is important to consider the following key properties:

1) That the standard or specification is subject to change control, and governance around the issuance of new versions, to ensure that versions incorporating changes retain agreed properties around forward and backwards compatibility (as set out separately below as a requirement), as well as any other necessary requirements that are identified.

   The standards development process should ensure that versioned standards are only issued for suitable standards, such that if an implementer adopts a newer version, this will not disrupt existing capabilities or ability to interoperate with a DSO, for example.

2) That the precise scope of the standard or specification is understood, specifically what is not included in the standard. For example, CIM is a popular standard for an information model to communicate structured information through, but it is not in itself a transport protocol, API, or architecture for a wider cross-organisation IT interface for dispatch commands to be sent over the public internet.

   In the context of this work, and the example set out earlier, any given standard is likely to encompass a limited subset of the necessary functionality to build a full flexibility dispatch system.

Interoperability

The practicalities of delivering real interoperability and interchangeability should be kept in mind. This includes how testing can be carried out, both to validate implementations against a referenced “gold standard”, as well as the integration tests required before an integration is declared ready for production use. In addition, there are likely to be testing requirements which will arise in order to validate correct security behaviours and properties (including both positive and negative test cases, such as rejecting invalid or spoofed messages, and properly acknowledging valid messages).

To deliver true interoperability at API level, such tests will need to be accessible to implementers, as well as sufficient for system operators to have confidence that an implementation which can pass interoperability tests is sufficiently tested to integrate with their dispatch platform. Otherwise, it is highly likely that standards bifurcation will be observed, with each system operator having their own different validation process and expectations around integration testing before an FSP can interconnect.

As part of this, both system operators and service providers will seek access to a “known good” implementation of the standard or API, in order to validate their own implementation against this. Ideally, interoperability tests will be sufficiently robust that if an FSP interface passes the tests, it will work with a system operator with minimal further testing. This is akin to “type approval” and interoperability testing that is sometimes carried out, with a sufficiently robust “type approval” testing specification ensuring that approved systems will all interoperate correctly. A lower-cost method of delivering this would be to establish an “interoperability lab”, perhaps using sandbox environments, to allow market participants to test their implementations against each deployed implementation they will need to interoperate with.

Scalability

Given the potential for the flexibility services market to evolve in future, there are no firm limits on the upper scale that these systems may need to operate at. This will likely be driven to some extent by market factors (such as the prevalence of aggregation, and the likelihood of smaller providers directly offering flexibility services to system operators). A standard or API should not pre-judge how the market will evolve, and should therefore be sufficiently scalable to allow for a large number of market participants, and avoid the technology or standard from becoming a constraint to market access, as this would likely be considered a significant barrier by regulators and policy-makers.

Enabling horizontal scalability (i.e. the ability for implementers to add more server capacity to deliver their services through addition of more physical or virtual servers) is one approach to delivering scalability to a solution.

Alongside this, however, there are also more practical scalability consideration which, while not necessarily mandatory in a standard, require consideration for practicality's sake – the process of onboarding a new FSP to a system operators' dispatch system would need to also scale, and this is likely to include exchange of cryptographic key material or other credentials. That means this process would need to take place securely. In addition, other entities interacting with this entity would need to be able to access that key material, in order to communicate with it securely. In the event that future changes in the flexibility ecosystem introduce new flows of communications in the context of dispatch, such as with ESO or a market facilitator, scalability would also need to be considered in the context of programmatically facilitating introductions of FSPs to these entities, in order to avoid the overall system becoming unworkable with a large number of FSPs in the market.

Security

Security should always be a core consideration of any system connected to the energy sector, especially those which are interconnected with CNI systems in system operators' networks. Including Flexibility Dispatch Systems, which will inevitably need to interface with CNI systems. The best practice from NCSC is that secure-by-design should be the default approach for building new systems – security should be a consideration in the earliest design phase of a project, and implemented by default as part of the whole system, rather than as an after-thought layer over the top of another non-secure system. Similarly, the principle of "defence in depth" means that systems should be designed to not rely on any one single point of failure for their security – meaning that robust, layered security mitigates against the risks of any one component failing to deliver the expected security properties.

Practically, this means that there will be a requirement for a specific and robust threat model, which will introduce wider security requirements on the architecture of the overall solution. Since it is not possible to design security to secure an abstract system, it is necessary to evolve the security threat model alongside the architecture and a specific technical proposal. The requirements in this deliverable have set out those which can be envisaged without other downstream decisions – for example around transport protocols, non-repudiation of long-term data held at-rest, and security properties of wire protocols.

Significantly however, these requirements will need to be augmented in the context of a holistic threat model and architecture, in order to design an approach to identity and access management, and access control and authentication.

Other considerations around security included around the ability for participants to patch and upgrade (while maintaining availability and uptime to the extent required) – this may introduce other architectural requirements around high availability and redundancy in deployments, depending on the decisions taken here.

Consultation with stakeholders highlighted the importance in strong governance around security postures, threat models, and requirements, on the basis that each system operator has their own approach to security, and will need to do their own work to evaluate and consider the best approach to integration of a new system into their existing IT estate.

In the event that a flexibility dispatch system is considered to have implications on CNI, and therefore national security implications, the security threat posture of the wider system and integrations will need to be carefully considered, to ensure that an appropriate protective posture is adopted. This may include considerations such as around isolation from internet-connected services, use of architectures for protocol-gapping and sandboxing, personnel security and expectations on third party providers, and other approaches to reducing aggregated risk.

There will also be implementational considerations (which are not usually within the scope of a technical standard, but would rather be in the scope of a standard for governance and accountability), such as around restrictions and expectations on vendors having remote access to production systems (in particular from overseas locations). Similarly, robust governance and accountability will be required around any expectations and flow-down security requirements onto FSPs and others in the ecosystem, to ensure that these do not pose unfair barriers to entry, but also to ensure that they are adhered to, and that system operators are able to effectively manage the cyber-security risks to their own businesses which arise from establishing interconnections with third parties.

Maintainability

This category considers considerations around the feasibility of market participants running and operating the necessary infrastructure to participate in the market effectively, and ensuring that the burden is not unduly onerous on them (which would discourage market participation and reduce choice in the flexibility market).

Similarly, a system is inherently more maintainable if it is modular, to permit integrations with other business systems directly, rather than introducing a range of requirements for extra manual processes to transfer information between systems.

Maintainability also includes the ability to install security patches and updates to improve the security and functionality of the API and interfaces, as well as to patch the underlying infrastructure that supports the applications. Another key stakeholder input around maintainability was the ability for different participants in the market to evolve their participation at different paces, so that FSPs only need to introduce API version upgrades where they see commercial value and benefit in implementing an upgrade for enhanced functionality or feature.

Finally, in this category, we also consider the ability for an API standard to evolve to meet changes in the market and expectations from regulators and policy-makers over time, as once a standard is in place, the update process effectively becomes part of the maintenance lifecycle of the systems deployed to support implementations.

Platform Independence

The ability for a standard or API to be platform independent is also important – this was identified in consultation with stakeholders to be an important consideration, to ensure that FSPs are free to create their own implementations, or purchase third party ones, and build the necessary integrations without a requirement to defer to a single platform vendor that would gain a preferential market position and be a single-source supplier.

The ability for a standard to run independently of any one vendor or platform also helps to avoid creation of single points of failure or dominant market players (which could create challenges in achieving value for bill-payers).

In addition, effective platform independence also ensures that interoperability is maintained, by avoiding the potential for non-standardised interfaces to gain significant adoption and uptake as “de-facto” standards, which would again prevent the true interoperability which is sought by the ENA in their process of attempting to develop an open and interoperable dispatch API standard.

Backwards and Forwards Compatibility

By backwards compatibility, we mean the ability for (e.g.) an FSP to implement an older version of a given holistic system, and (where appropriate and not a security issue) not upgrade it if they don’t gain advantages from newer features. This requires decoupling of technology from platforms (so that they are installing security updates on operating systems, hardware, hypervisors etc.), but allows them to avoid (e.g.) totally re-writing their own internal platform interfaces against the dispatch system every time there is an update.

Backwards compatibility ensures that the wider system behaves in the presence of an older component. This also includes a requirement for graceful degradation, and ensuring that where a message is not fully processed due to unknown fields/values being seen, this is handled appropriately, and the sender is aware of this. This means there should be a means to communicate the right behaviour in the event of an error (i.e. criticalities on fields – criticality of abort, alert, ignore, etc.) – this way the message sender can indicate how important it is that an implementation understands a given field or message.

Forwards compatibility is effectively the opposite – ensuring that the wider system can support newer components being introduced without causing breakage. This means not introducing mandatory new fields without a major new version, and handling graceful fallback scenarios where one side of the link only supports an older version.

It is important to note that security requirements mean it is unlikely that any component of this system will be “fit and forget” – rather this is to minimise the extra workload and burden on FSPs and DSOs, but there will still be a requirement to promptly install security patches and generally keep platforms and system secure.

“Build vs Buy”

A key question arising from this work is around whether or not there is a viable option for an industry-wide API-based dispatch system, or whether the option of developing a bespoke dispatch API for the UK industry is required. In this section, the considerations needed to evaluate these options are set out.

In the IT industry, this is commonly referred to as the “build vs buy” question – the trade-off between developing a new solution against requirements, as opposed to the use of an existing solution. For avoidance of doubt, the phrase “buy” is used without implication of there being a cost to access an external standard, and is used to reflect the option of using an existing available product or standard/API.

This section aims to present the advantages and disadvantages of each approach. Firstly, focusing on the selection and adoption of a standard, and later from the perspective of an API.

True open standards may limit options

The option of “buy” isn’t truly available when an open standard is sought

There is not a straightforward “buy” option for an open standard, since the key properties of an open standard are around its governance, evolution and development. If one were available to “buy”, it would then require the same governance to be added over the top as a bespoke standard would require.

Adoption of an existing standard avoids this, but then prevents the UK from adopting the changes it seeks to make, unless it can gain wider acceptance of those changes. This would delay the pace of innovation for the flexibility market, and be a difficult-to-defend decision, in the face of questions from industry about why changes cannot be made. In essence, adopting an external standard and accepting all upstream changes to that standard would mean the UK had no real influence in this, unless UK market participants chose to involve themselves in this process (presumably at their own cost).

Given that many FSPs are already cautious around the time commitments required in participating in governance for UK-based activities, there is unlikely to be significant interest from implementers willing to do this. In addition, there are parallels in the telecoms sector, where the UK has a significant shortfall in standards activity, and limited input and influence as a result.

Existing standards development organisations can help to set up a committee and collaboration area, with existing policies and procedures, but do not themselves develop the standard – volunteers on standards committees develop standards. Therefore, for any evolution of an existing standard, or creation of a new standard, the same governance requirements will be in place. In the IT sector, large blue-chip companies dedicate multiple FTEs of staff solely and entirely to participation in standards development, because they recognise that this work drives innovation and enables products to reach the market and make that market bigger through interoperability reducing barriers to use. For example, at the time of writing [1], there were 535 voting members of the IEEE 802.11 group which develops the Wi-Fi standard, with a further 136 in the process of becoming a voting member.

Regardless of whether an existing or new standard is adopted and evolved, the bulk of the work of governance and stewardship of standards development would be the same. There is therefore a strong argument to be made for ensuring that whatever standard is adopted has the correct scope and functionality to meet what the UK energy market needs, on the basis that the most significant overhead (governance and standards development) would be the same either way – therefore the option is available to develop a bespoke initial open standard, based on the most appropriate external references.

It is important to note that existing commercial proprietary implementations of APIs and systems are not likely to meet the requirements for an open, freely implementable standard. Firstly, that is an implementation, rather than a standard. Even if it were suitably documented, there would not (by default) be rights for third parties to freely implement the API, and there would be commercial and intellectual property licensing considerations that would affect both FSPs and system operators. The ENA has set out the importance of the adoption of truly open standards, and therefore we consider this to be a key requirement for

Stakeholder feedback seeks to avoid re-work

Industry doesn’t like reinventing the wheel, but doesn’t want to have to throw away work

In stakeholder discussions and workshops, there was a general view that there is no requirement to reinvent the wheel, and that there may be existing viable solutions. No FSP/aggregator stakeholders had experience with such solutions however, indicating that the maturity of existing standards/APIs was not necessarily at system-ready level.

In addition, while FSPs and aggregators generally supported the idea of evolving and iterating a system, to deliver early value and incremental uplifts (in line with the “agile” spirit of software development, as opposed to a “waterfall” approach), they also were keen to emphasise that while they want a flexibility dispatch protocol quickly, they also don’t want to have to change their own implementation as this will add costs to their own businesses.

This creates a risk that if an unsuitable API/protocol is adopted in the early days of flexibility dispatch interoperability, it could result in a significant long-term requirement to support this, if FSPs and aggregators are reluctant to move to an upgraded version. This will introduce a requirement for system-level governance, in order to balance considerations such as costs to FSPs, and the security of the overall system (i.e. through deprecating and disabling legacy insecure options).

This also introduces conflicting and contradictory requirements. Larger FSPs were generally more willing to integrate against the adopted standard without further details, and to follow revisions to that standard, while recognising that they would prefer to see iterative change and evolutions to a standard, rather than a need to completely start again and lose the investment they have made in implementation. Smaller FSPs were generally more concerned by ensuring that the costs to them over the longer term were manageable and that supporting API integration was not a major consideration for them in their operations. In addition, many participants explained a preference for standards to be accompanied by real implementations and "sandbox" examples to support testing and validation, due to the inherent ambiguities present in written standards. This recommendation has formed a key part of the requirements and gaps which were highlighted in D1 and D2, and we reiterate the importance of ensuring that the necessary ecosystem surrounding a standard is understood before decisions are taken – the industry has made clear the value of being able to test against a real implementation, rather than develop against an abstract specification.

This likely sets a sequential logical dependency for there to be a functional simulated "system operator-side" API implementation (along with the wider holistic system including security and design considerations), to allow them to develop an integration against that API, and validate their handling of messages with real tests. It is worth noting that most standards will not have this available, and that this would create a requirement for the Open Networks programme to consider and implement such work, irrespective of choosing to adopt an existing standard, or create a new one.

We believe that this summarises the general sentiment and challenge in the trade-off between adopting an existing standard. It will not be possible to ensure that every stakeholder is happy, since some of the requirements conflict – a desire to see agile evolution and incremental changes to a standard will never be easily reconciled against a desire to "get it right first time". Within some technical guide rails and governance to verify the solution and proposal holistically at each step of development though, it should be possible to avoid breaking changes that would mandate changes by every implementing FSP to permit the system to function.

There will also likely be a range of requirements to validate conformance and interoperability with versions of the standard, which should also be considered from a governance perspective, since there will be a requirement to develop appropriate and robust tests to validate that FSP implementations are not vulnerable or poorly architected/designed, and that they properly respond to and process dispatch messages. Where dispatch is integrated to a wider flexibility market and ecosystem, this will introduce a hard requirement for wider holistic assessment of implementations against the wider ecosystem, and this will require greater clarity on the decisions and approaches being taken in the wider ecosystem.

This also introduces a consideration where there is an unavoidable requirement for FSPs and other implementers to introduce changes in order to adapt to the UK's flexibility ecosystem evolving in line with HMG and Ofgem policy. Given that the need to support future unspecified changes in the wider market is required, there may be an argument to be made to set the expectation on FSPs that, within reasonable governance, there will inherently be changes made to standards, specifications and expectations (such as around security), but with transparency and accountability around how these decisions are taken, working in partnership with HMG and Ofgem.

Future market evolution considerations

Standards and APIs codify future market structures and dynamics

One factor which should be considered by Open Networks Programme stakeholders is the extent to which selection of an external standard or API will effectively define or dictate the future evolution of the UK's flexibility services market. We do not believe that it is the intention of the Open Networks Programme or Ofgem to allow the UK's flexibility service market structure to be dictated by the technical evolution of a standard or API. Therefore, we would highlight that a key consideration around the "build vs buy" decision is the extent to which the UK needs the ability to develop its own flexibility dispatch ecosystem over time.

A key finding from the stakeholder engagement activity was that FSP and aggregator stakeholders were generally very keen to see integration across the whole of the dispatch business process ecosystem, rather than the isolated development of solutions in the different Open Networks Programme flexibility services implementation areas (registration, procurement, market interactions, operational planning, scheduling, dispatch and settlement). The more that UK-specific practices need to be reflected in these areas, the more likely it is that a UK-specific standard and API may be required, in order to codify the functionality and market behaviours used in the UK market. An example of this includes the potential future market facilitator role, under consideration by Ofgem – the creation of a market facilitator is likely to introduce additional communications requirements on parties in the flexibility ecosystem. Since the specifics of this role are likely to be specific to the UK energy market, these requirements are likely to mean that the UK would need to implement its own approach to information exchange with the market facilitator. This could be delivered as an extension to an existing standard or API, whereby the market facilitator would be able to communicate with API users. These API fields and functions would inherently not be available in any initial standard or API implementation, as they are as yet undefined and unknown. Introducing these would therefore be likely to require a breaking change, as well as reconfiguration of API users' systems to recognise and communicate with the market facilitator.

One risk of implementing a solution which does not meet the needs of FSPs and aggregators is that it results in limited adoption or uptake, or reduces the willingness of potential market participants to participate in the distribution flexibility market, in favour of the ESO flexibility market, or balancing mechanism. Some flexibility resources are already dispatched through the Flexible Power platform, as a result of pilots and trials that stakeholders have discussed, although adoption has been far from universal, and it was clear from FSP and aggregator feedback that what they seek from Open Networks is a unified and coherent integrated flexibility ecosystem.

There is a significant risk that, without properly considering the whole market structure when selecting standards, each area of the flexibility ecosystem will end up bifurcated on a different non-interoperable platforms, interfaces, standards, or approaches, resulting in a series of different technology platforms, and effectively delivering the status quo that is already available today – i.e. disconnected dispatch and procurement platforms operating independently. Clear feedback from FSPs and DSOs alike was received during stakeholder engagement – the supply-side of the flexibility market seeks an integrated and coherent flexibility ecosystem that reduces costs and streamlines market access, with as few barriers to participation and entry as possible.

Standards development and governance

Existing standards bring governance, but that also increases overheads

We make the following recommendations around governance and delivery, as well as accountability, in line with the specification for D3. While existing standards come with existing governance frameworks and committees, these will generally be internationally focused, and will not be likely to meet the needs of the UK as it embarks upon its net-zero transition. It is likely that blaming an external standards development process for being slow would not be acceptable to stakeholders, not least FSPs and HMG/Ofgem.

For this reason, it is worth noting and acknowledging that there is likely to be a requirement for standards governance around whatever is adopted (as well as wider governance around requirements that sit outside of an API specification) for wider ecosystem integration, interoperability, testing, security and architectural requirements. Without agreeing these before FSPs and others attempt to implement a standard or API, it is likely that the evolution will be perceived to be major changes to the standard, since to most outside participants in the flexibility ecosystem, they seek a holistic solution they can implement, rather than merely a standard.

Therefore we would recommend that when considering next steps on this work, beyond merely selecting a standard, that significant consideration is given to also recognising that FSPs and others in the industry seek a specification and design for a holistic implementation (which also considers and interacts with other parts of the flexibility ecosystem in due course), as opposed to simply the specification for an API – in and of itself this would not be implementable, as the wider security, architecture and implementation factors would be required for an FSP to understand and begin to consider their development. Absent architecture and implementation clarity around security and design, it would not be possible for an FSP to implement the standard, and it would likely be a distraction for the ecosystem to present the selection of a partial API specification as the solution to flexibility services dispatch.

Summary of Findings

As part of this work, we explored various different potential technologies for dispatching flexibility, including OpenADR, CIM, UMEI. UMEI was an EU funded research project bringing together various DSOs, marketplace providers, and dispatch platforms to demonstrate a proof of concept. This project worked well and demonstrated feasibility, however, as a result the system is not fully fleshed out, and it lacks the surrounding enduring ecosystem and enabling architecture and design. It would provide a strong foundation to build upon, however, differences between the EU and UK energy markets may result in it being easier to build from scratch using the learnings from this project mixed with the UK specific requirements.

The Common Interface Model (CIM) has already seen success in UK projects, in particular in the Ofgem Long Term Development Statement (LTDS) work. This system works by defining agreed 'profiles' or data structures for energy operators to exchange highly detailed information about their network. CIM could operate well as a similar data model for flexibility services, however, the required profiles would need to be defined, and for a full system the supporting architecture, security and testing would similarly need to be defined. CIM is, in essence, a standard for defining a standardised data model.

OpenADR (Open Automated Demand Response), is an existing solution which would meet many of the requirements 'out of the box', as it features an architecture, a passable data model, built in security and a recognised testing suite and certification program. OpenADR also benefits from an existing governance board (the OpenADR Alliance), with the ability for FSP and DSOs alike to join. There is also likely to be downstream familiarity with OpenADR for use in market-driven control of domestic energy resources, as the standard was developed for communicating signals to devices to allow them to reduce their usage during periods of high demand.

After the TWG fully understands their wider requirements, as discussed in D2, they will be in a position to evaluate standards, and whether any available standard meets the requirements. At present, we believe that OpenADR is the closest to a usable standard, although that it would not meet the requirements at present, and would require modifications to be made. There is therefore a decision for the TWG to take once its requirements are better understood, as to whether to adopt an existing API standard (and then build the enabling and supporting infrastructure and changes to it separately), or build a new option. If the option to adopt an existing standard is taken, there would need to be consideration of how to manage where the UK deviates from the existing implementation, and whether this would necessitate formation of a new standard, and if this is permitted by the authors of the upstream standard.

Evaluation Methodology

To ensure that effort on this activity was focused on reviewing options which presented credible properties that made them viable as options for dispatch API standards, a short-listing process was used. This converged on 3 options which could be technically evaluated against the requirements set out in the evaluation spreadsheet – CIM, OpenADR, and UMEI. Some standards which were explored and identified were not shortlisted – these are set out below, alongside the rationale for this, bearing in mind that this work was not an exhaustive evaluation of all possible standards. Similarly, legacy standards which are not API-based or meeting the basic outline criteria of this piece of work have not been considered (i.e. DNP3 and similar).

Standards Not Shortlisted	
Standard or Proposal	**Rationale to not shortlist**
OASIS Energy Interoperation Common Transactive Services (CTS)	• Standard reached a draft v1.0 in April 2022, with a feedback deadline of June 2022, and has not been ratified since, with draft containing many TODO's. • While standard implements a wide marketplace specification, the semantics used to represent "transactions" (i.e. their concept of an invoked tender/contract) do not align with the messages required to carry out dispatch. • The transaction support does not allow for cancellation or modification of transactions (i.e. to implement cease or variation instructions as required by ENA). Nor is there any concept of declaration of availability. Transaction support only includes price and quantity fields; therefore is not communicating what is being dispatched, and would not meet needs of a dispatch protocol.
Energy Flexibility Interface (EFI)	• Is a protocol for direct control of end-user smart devices, seeking manufacturers to implement EFI on those devices, to enable the in-home devices to work with other protocols such as OpenADR and similar. • May be useful for aggregators and FSPs for their "to device" communications. • Protocol is heavily designed around end device control, and announcing device capabilities. It is stated (Section 1.2) as a solution for interoperability in communications between a consumer device and demand-side management system.
IEEE 2030.5	• The relevant portion of IEEE 2030.5 (i.e. demand response and load control) is effectively a SCADA control-type protocol for direct asset control, rather than dispatch. Messages are therefore end-device oriented, focusing on parameters like temperature offsets for thermostats. • The protocol contains no concepts for reference to procured services, and instead focuses on the actual control events for end devices.

USEF	• The USEF Foundation is no longer active as of June 2021, although the materials it produced are available online. • USEF defines a wider flexibility ecosystem in their standards, which may deliver on other areas of interest such as issuing flexibility requests (i.e. market engagement). • The operate phase of the standard is the one which pertains to flexibility dispatch, but the written standard does not provide sufficient details to allow the standard to be implemented or validated (i.e. the schemas for operate phase messages such as "FlexOrder" were not defined or available in the XML schema).
Inter-Control Center Communications Protocol (ICCP) [IEC 60870-6]	• ICCP is a widely used tele-control protocol for communications between control centres. It offers basic control requests (switching, trips, raise/lower, etc.) and code execution. • ICCP offers no security, and assumes it is provided by network layers – this would present challenges in a flexibility market with a wide range of participants.

Scoring Table Explanation

Based on the gap analysis (D1), requirements identification (D2) and comparison work (D3 spreadsheet), the following key recommendations are made. Note that, per the agreed scope of this work, no specific recommendation is made as to any particular technical standard. Rather, a framework for evaluation of technical standards has been proposed.

To maximise the capability for this to be used as an objective comparison tool, a relatively straightforward "breadth-first" approach to comparison has been proposed. Based on some pre-qualification questions that determine the scope of a given standard or API that can be evaluated, a series of 9 evaluation categories are then explored:

- Open standards
- Interoperability
- Scalability
- Security
- Maintainability
- Platform independence
- Backwards compatibility
- Forwards compatibility
- Governance

The evaluation criteria set out in each section have been derived from the stakeholder consultation exercise, and gaps and requirements identified in D1 and D2. For each requirement, a MoSCoW categorisation has been proposed (with rationale included as a comment on that field of the Spreadsheet). For simplicity, any "won't" requirements have been inverted to be positive "must" requirements, since generally any negative requirements from stakeholders were also expressed in a positive "must" sense. This also assists in evaluation, by giving three simple scores – the "Must", "Should" and "Could" score.

This methodology ensures that no one criterion gains an excessive weight compared with other criteria, and reflects the breadth of requirements and desirables that have been identified. It is possible to tune the evaluation approach by altering the MoSCoW classification of any given criterion on the spreadsheet to change evaluation criteria (ensuring that this change is replicated to any other tabs of the spreadsheet being used for comparison).

The "actual" column should be filled in for each evaluation criterion - by reviewing a given standard, and for each evaluation criterion setting a "yes" or "no" value against the actual column (by entering "Y" or "N"), a score is calculated for Must, Should and Could criteria.

After determining whether each standard meets all of the given criteria, a score will be shown, which is based on the percentage alignment that a given standard has against the Must, Should and Could criteria. At this point, consideration can also be given as to the relative prioritisation of must, should and could criteria against each other – a holistic assessment can then be carried out for each of the standards to understand alignment and gaps which exist in standards.

Since many of the requirements identified have system-level and architectural level implications, it is possible to select which of the categories of evaluation criteria are covered by a standard, and focus on those requirements. Where categories are removed from evaluation, it is important to note that these requirements will need to be delivered through other work, standardisation, architecture or implementation design, and this will need to sit outside of the scope of a given standard, requiring further work to be carried out. This can also be used as a form of high-level comparative gap analysis, to understand which standards consider architecture, testing, security, etc., since some give no consideration to these factors.

A justification for each initial MoSCoW classification of each criterion is included as a comment in each cell for column D, providing a brief explanation of why the level suggested was chosen. The MoSCoW classification can be changed as set out above if the TWG wishes to adjust their evaluation criteria.

Cost Efficiency and Ease of Implementation

Given the areas of uncertainty highlighted throughout the gap analysis and minimum requirements deliverables, it is not feasible to carry out a comparative evaluation of costs, since we believe that OpenADR is the only evaluated API standard that could be modified to meet the requirements of a flexibility dispatch standard for the UK energy market.

Since there would appear to be a requirement to make a range of modifications to OpenADR however, this is likely to entail the same overheads and cost burdens as running the governance process for a new standard. The requirement for governance around an API standard is significant however, and it is worth noting the scale of governance on other international standards as set out earlier in this document.

Stakeholders expressed a level of willingness to support and participate in open governance activities, but wanted to see clear benefit for themselves for participating, particularly for smaller market participants. There would also be a requirement for a core level of technical capability to lead on development work of the standard, and likely a requirement for a secretariat to manage administration and communications of the group. It is important to reiterate that while there are many existing standards development organisations, they do not develop the actual standard – they provide the framework in which participants (from the sector) can carry out the development of the standard.

There would likely be some cost savings if this new standard was able to leverage existing work on OpenADR for a dispatch API, although the extent of these cost savings is likely to be minimal in comparison with the wider costs of managing and operating a standard in the longer term. As such, there is likely a decision to be taken as to whether or not to develop a standard from OpenADR as a baseline, or to develop from scratch, based on requirements.

While there may be some minor advantages in ease of implementation for FSPs by using an existing standard, given we do not believe OpenADR (as-is) currently meets the requirements for the UK energy market, and therefore the costs of implementation are likely to be broadly similar to implementing a new standard, as implementers would need to carefully ensure that any variations for the UK version were in their own implementation.

Recommendations

For ENA to continue to progress this work beyond the scope of the current engagement, we make the following 5 high-level recommendations. As agreed in the scope of this work, we do not make any specific technical recommendation around standards or API choices.

1. We recommend that the TWG actively engage with Ofgem and DESNZ in order to build a **shared and common understanding around the limitations of selecting an API and standard, and the significant work required beyond this to deliver an implementation**, due to critical areas which sit outside of the scope of an API and standard, and which are essential to delivery of a viable and implementable solution.

   This could be justified based on the clear findings from stakeholder engagement in this work (as set out in D1 and D2) seeking wider cross-functional integration across the wider flexibility ecosystem, and therefore the cross-workstream dependencies sitting in ON, as well as the need to potentially develop architecture, implementation, testing and security threat models for API integration. This should also help to build a wider understanding among regulatory and policy stakeholders, including on the dependencies on external factors before having sufficient information to definitively select an API standard.

2. We recommend that the TWG **consider whether it has sufficient information at this stage to make a fully informed selection of an API or standard**, or whether it should make a preliminary recommendation that is subject to change on the basis of future findings, and areas of future uncertainty (as discussed below).

   in the event that there are still unanswered questions around architecture, security requirements, and similar, on the basis that the API or standard sits within these outer frameworks that may still need to be defined and agreed in consultation with industry and relevant expert stakeholders around topics like security architecture and risks.

3. We recommend that TWG to carry out a holistic prioritisation and **determine exactly what is in scope of the dispatch API work**. The wider the scope, the less likely an existing API/standard will deliver what is required.

   This recommendation is made, since there is considerable body of evidence to show that IT projects likely to deliver successfully have carefully managed, defined and understood scope, with sufficient technical knowledge present in the requisite in-scope areas and enabling technology to effectively manage the project [2]. There is a risk that without a very carefully managed scope (and clearly defined boundaries for the first version), that expectations and demands on it will evolve, requiring progress to be reset. In particular, scope around relationships with other ON workstreams should be set clearly, given stakeholder feedback on the value of a joined-up and aligned flexibility ecosystem beyond merely dispatch.

4. We recommend that the TWG conduct a **technical analysis across the whole flexibility landscape**, to determine whether the advantages gained from selecting an API and standard that already exists

(and adapting as required going forwards through creation of a UK-specific standard or API) outweigh the potential opportunities of starting from a clean slate standard or API (informed by existing standards and APIs).

This is worth considering, in order to ensure a defensible decision is made to either adopt an existing standard, or create a new one, given the likely overlap in governance and standard development requirements in either case. In the event that future changes in the wider flexibility ecosystem (such as the introduction of a market facilitator) introduce new requirements for information to be exchanged, this may necessitate specification of new interfaces and data structures to communicate dispatch-related information that would not be encompassed in a direct DSO-dispatch standard.

5. We recommend that the TWG **split the scoped future work into a series of logical (but linked and dependency-managed) focus areas**, in order to reduce the delivery risk of a single large piece of work that cannot be sub-divided. We recommend that there is an overall architecture and security focus group, a governance and accountability focus group, and a versioning and interoperability focus group. There may be further requirements for specific energy domain expertise as well around scoped areas of work, such as flexibility dispatch and procurement processes.

   This would support delivering the technical knowledge requisite to de-risk delivery of the TWG's future work, and also sub-divide the focus of work, such that the "ask" both on individual members, as well as wider flexibility market participants, is more focused and topical, to encourage and enable wider participation in the work and demonstrate to policymakers the involvement of the wider ecosystem in formation of the underlying solution.

In making these recommendations, we have had regard to the feedback from stakeholders, including FSPs, aggregators, technology providers, DSOs, ESO, and others. We have also taken into consideration the priorities set out by the TWG for Deliverable 3's specification – in particular noting the importance set out around agility in governance to enable changes to the flexibility market to be incorporated into the selected API standard, and the potential for new services or operating models to emerge in future.

Key areas of uncertainty which may inform requirements

We believe that the three key areas of uncertainty that are likely to materially inform further requirements that may impact on selection of an API (through either introducing data model requirements, or API and protocol requirements) would include:

- Decisions around the **introduction of a market facilitator** role by Ofgem, and whether this would interface with the dispatch API.

- **Primacy rules and how those are implemented** and codified in an API/standard, and whether this introduces any ESO to FSP communications, or requires a DSO to relay communications from ESO to an FSP, and how this would be handled by an FSP.

- **Architectural and security design considerations** for a viable and deployable solution, which meets the needs of stakeholders, including identity management, cryptographic key management, and key exchange. These may introduce other requirements which will not be identified in an API standard, but which may have downstream impacts on one, such as a need for a centrally managed public key infrastructure (PKI) to sit across the wider ecosystem.

In addition, a range of gaps were set out in D1, many of which may also reflect areas of uncertainty that could materially inform further requirements for a dispatch API standard.

Enclosed as annexes to this report are the final versions of Deliverable D1 and D2, the interim reports covering a holistic gap analysis for the wider flexibility ecosystem, the minimum requirements analysis, and an annexe on cyber security considerations.

References

[1] IEEE P802.11 - Working Group Public Members List, Available https://www.ieee802.org/11/members.htm

[2] Darryl Carlton, ‘Lack of technical knowledge in leadership is a key reason why so many IT projects fail’, The Conversation, 10/9/18, Available https://theconversation.com/lack-of-technical-knowledge-in-leadership-is-a-key-reason-why-so-many-it-projects-fail-101889.

Appendix 1: D1 Interoperability Gaps

Executive Summary

This document forms the first full deliverable of the PNDC/ENA ON Flexibility Systems Interoperability project. As electrical consumption patterns shift, and additional loads are added to the network the procurement of power from local energy resources, such as batteries, is a major tool to enable net-zero. These local resources can respond to local requirements, and avoids the need to provide costly network enforcement through more intelligent utilisation and distribution of existing resources. The ENA Open Networks (ON) project is focused on enabling such changes, and is broken down into various different working groups to tackle discrete issues. The primary objective of the ENA interoperable Dispatch Working group was to select a common interface for all flexibility service providers (FSPs) to provide flexible power in local constrained areas to electrical network operators using a common interface.

A common API was determined to be a requirement. This was to prevent eco-system fragmentation, and reduce barriers to entry for FSPs. If FSPs were to implement discrete solutions for each regional network operator would reduce the commercial incentive to provide solutions and increase costs for bill payers. To ensure each of the technical requirements for such a common interface was captured, PNDC were commissioned to engage with various stakeholders in the growing flexibility service ecosystem to understand their requirements. This process would also establish any barriers and identify any gaps in both technical implementation and practical understanding.

Through this process PNDC engaged with 50+ stakeholders, from various different parts of the ecosystem. This led to a wide range of experiences, desires, and requirements. FSPs were broadly keen to have a simple and easy to use common interface, with no major migration as there was insufficient return on investment (ROI) for them to justify significant and recurring software developer resource. From a technical perspective their requirements, were fairly minimal, preferring rapid operation and stability over advanced functionality initially. The challenge this presents is potential future innovation would be stunted if initial basic operation is codified with no clear means of progression. Therefore, once ambitious FSPs have realized the value in the market and would like to progress they would be stuck on legacy versions. As a result, one of the major 'hidden' requirements is extendibility.

A second major requirement was that of data portability, network operators would require to transition their portfolio of FSPs to a new platform to prevent vendor lock-in. As without interchangeability, FSPs would have to re-register all of their assets each time a system is changed. This would create a poor user experience for FSPs and require consistent technical resource reducing the ROI.

The major practical gap which was identified was the implicit presumption that a common API can be defined down to message structure permits "interchangeability" needs to be carefully re-visited – interoperability and interchangeability are two different properties, and API selection is likely to be one of the last steps in a process of aligning a common business process for flexibility procurement, dispatch and settlement across multiple DSOs, since an API needs to be designed to support the implementation of that business process.

Introduction

As part of the wider ENA Open Networks Project, the interoperable dispatch working group had the objective of developing an interoperable common API for dispatch of local energy resources to meet local constraints. The advantage of a consistent communication interface is that is reduces the barrier to entry for FSPs, deceases costs for bill payers, and grows the overall eco-system. Accordingly, as part of this workstream PNDC was contracted to explore any interoperability gaps, both technical and practical and engage with various stakeholders to establish technical requirements.

One relevant factor when considering this current work is **the technical distinction between a standard and an API**.

An Application Program Interface (API) is a specific technical interface definition used for applications or systems to communicate across. It should be well-defined, such that others can implement it. It is possible, however, for an API to be subject to intellectual property or licensing restrictions. This is the case for a proprietary software API – where access to the API documentation, definitions and interface specifications may be governed by an Non-disclosure agreement (NDA) or other commercial licensing arrangement. Different stakeholders had different perspectives on what an API was with respect to a platform, and an API versus an architecture, as well as an API versus a standard.

An API is fundamentally a way for two or more different pieces of computer software to communicate. An API may be defined through documentation or a specification, which is a technical document setting out the parameters required to interoperate with other systems implementing that API. An API can provide a practical way to allow independent systems with different architectures to communicate, by creating a well-defined abstraction. The term API is often used to refer both to the technical specification of communications, as well as a specific implementation of that technical specification. It is important to note that for an API to be interoperable (i.e. allowing others to implement it), there must be a technical specification and documentation around it. Best practice is for this specification and documentation to be versioned, and subject to change control and governance, to ensure that appropriate integration and interoperability tests are defined correctly for each version.

In contrast, a standard, as generally considered by standards development organisations (SDOs) to be technical specifications and procedures, which enable interoperability across consistent protocols which can be universally understood and adopted. [A1:3] A standard could, for example, include the technical definition and specification of an API, but the standard would not cover the implementation itself – the standard would define how implementations should act and behave.

The process of technical standards development is one of broad consensus and mutual agreement. This means that the standards development process moves relatively slowly, in order to ensure that all stakeholders have an opportunity to have their views heard, and input to the process. Rapid evolution would likely hinder this. To ensure that a standard gains adoption and uptake (since there is generally no specific obligation to adopt or implement a standard) over other alternative proprietary solutions, the process of developing standards needs to be inclusive and open to be credible.

Standards Development Organisations (such as the IEEE, IEC, ETSI/3GPP etc.) have significant rules and procedures designed to ensure that standards are developed transparently and openly, in order to avoid negative effects – such as dominance [A1:4] and disclosure of affiliations [A1:5], alongside detailed governance procedures [A1:6].

Decisions such as this are likely to lead to a need for formal defensible decision-making and governance around such decisions, in order to ensure that all stakeholders' needs and interests are given due consideration.

By way of example to indicate how this could go “wrongly”, a proprietary API specification could hypothetically be adopted as a result of this process, due to meeting most of the technical requirements. The vendor of the proprietary solution would therefore gain a dominant position in the market for the DSO-side of that solution, and would be likely to gain significant market-share and revenue as a result of this. As a downstream consequence of this, they could potentially also increase their pricing and render their commercial terms for access to API specifications more onerous. FSPs wishing to participate in the market would therefore experience a new commercial barrier to participation through licensing this specification or API details. Were the API subject to patent protection, there may be licensing liabilities for those implementing it, through utilising the underlying patented methods.

There may also be secondary benefit afforded to one vendor as a result of such a selection, such as if an incumbent vendor is the sole source of test harness software or validation/emulation software, to support those developing integrations with an API.

As a result, we have identified a gap around the requirements for governance for a specification vs an API, and how that decision would be taken, in order to ensure that considerations such as IPR protection, licensing, pricing and commercial models, and generally ensuring ease of implementation for a diverse range of FSPs are taken into account. These considerations are taken into account in formal standards development processes through IPR policies, and we therefore highlight as a gap the governance around these decision-making processes.

More generally, we believe that in a "worst case" scenario, factors around licensing and IPR of a proprietary API specification could be "market restricting" and limit access for FSPs and aggregators to participate in the UK market, with likely detriment to system operators and bill-payers.

In addition, there will need to be clear governance and accountability around decisions which are taken, that may result in costs being borne by a market participant (either DSO or FSP/aggregators), since these will ultimately be borne by bill-payers. For example, if a DSO changes platform vendor, this could introduce re-validation and conformance testing being required by multiple FSPs. This change may be to the benefit of the DSO, but to the detriment of the FSP, in such a situation. Similarly, FSPs may seek longer availability of older versions of APIs to reduce integration and software development costs, resulting in higher costs to DSO/DNOsin the support and maintenance costs of their platform. A principle captured in feedback from a DSO was that it is important to ensure that where costs or implications of a decision are borne by one entity, that entity has influence over that, and is not effectively cross-subsidising other market participants by reducing their costs. This highlights the importance of a governance and accountability layer.

Stakeholder Engagement

PNDC conducted an in-person workshop to foster initial engagement from a variety of stakeholders, for a high-level understanding of the challenges, requirements and considerations for FSPs to engage in the flexibility market. This was followed by a series of direct interview style engagements with stakeholders to better understand their requirements/considerations/concerns. This format was selected as it allowed us to best understand stakeholders, "pain points", and the best solution to alleviate these. The stakeholders included distribution network operators (DNOs), Distribution System Operators (DSOs), Standard Development Organizations (SDOs), National Grid Electricity System Operator (NGESO), dispatch service providers, subject matter experts, aggregators, industry groups, government, consumer representatives, equipment vendors, established market providers and new entrants. This elicited a series of views of requirements. For example, FSPs were generally positive about the move towards a common interface, however, did not want such a system to be overlay onerous.

Type of Stakeholder	Number of Engagements
Subject Matter Expert	3
Flexible Service Provider	13
Dispatch Provider	6
DSO/DNO	19
Industry Groups	1
Manufacturer	1
Market Provider	2
Standard Development Organization	3
Government/civic	2
Total	50

The majority of the system requirements can be grouped into either FSP or DSO/DNO focussed requirements. This was either directly from FSPs/DNOs or through associated parties, for example, SDOs were able to provide helpful feedback surrounding existing standards, dispatch providers were able to provide useful insight into existing implementations. These highlighted suggestions were then validated against the FSPs/DNOs to establish if this would be requirement or not.

From these engagements, a series of key insights were gained:

Flexibility Service Providers

- FSPs have a strong preference for deploying a solution now, which can then be iterated on, rather than waiting to develop “the perfect solution”.
- FSPs were mixed on if a “confidence parameter” would be required or not, but had no strong feelings on the matter.
- FSPs would prefer a common digital life-cycle engagement between all DSOs, including Tendering, PQQ, Contracts, dispatch and settlement.
- FSPs agreed that developing consistency in the market and wider eco-system would be extremely beneficial to grow liquidity.
- FSPs had no strong feelings on including a unique asset identifier.
- FSPs had a preference to employing modern technologies, (HTTP REST vs XML SOAP), given the more established ecosystem of developers.

- FSPs highlighted the importance of trial sandboxes, where they can explore and experiment with the interface, even over supporting documentation as documentation can be interpreted differently.
- FSPs wished to have greater visibility of potential commercial opportunities.
- FSPs would look forward to stable APIs for automation but also email notifications for information.
- FSPs recognised the importance of Cyber Security, but mostly considered this a platform issue.
- FSPs generally sought iteration on a design, albeit without breaking backwards compatibility – discussions tended to focus on examples of versioned systems, and the potential to support different versions of an API for a longer period of time, with implementers able to upgrade when they felt it was worthwhile.
- FSPs generally don't mind what the dispatch platform is employed, provided it is consistent, has longevity, and relatively simple to deploy.

DNO/DSOs

- DSO/DNOs wished to have clear control/accountability for platform interactions and issues.
- DSO/DNOs want a dispatch system which can integrate with existing/external systems, i.e. network data.
- DSO/DNOs wish to have market data, dispatch and settlement able to be linked up through a single platform.
- DSO/DNOs wish to have data portability between dispatch platforms, and noted the need to integrate with a range of internal systems.
- DSO/DNOs would like, and value, a simple user interface (UI) for any system like this.

Interoperability Gaps

Previous ENA TWG work identified that the deployment of a common API would increase market engagement, however, the complexity, scope and interoperability of that common "API" varies considerably. The most pronounced gap we noticed was the varying understanding and interpretation of the system wide requirements. Notably, the distinction between a standard and an implementation, while our focus remains on exploring the requirements for a standard. Understanding what organisations may desire from an implementation perspective, provides underlaying requirements for the standard which that implementation should be built against. In addition, given that many participants were not focusing on this distinction themselves, it was necessary to discuss perceived requirements from an implementation perspective, in order to understand where gaps lay in how a standard for interoperability could be defined.

These gaps have been broken down below into the six flexibility platform tasks defined by Ofgem: Coordination, Flexibility Procurement, Dispatch and Control, Platform Transaction Settlement, Platforms Market Services and Analytics and Feedback. Finally, we have captured some additional gaps which did not directly fit into these categories.

These gaps distil the more discursive engagement with providers, and capture the more nuanced discussions and differences which came up in the different stakeholder engagement sessions, since differing or deviating views on how things should or might work highlight potential gaps that must be bridged before a truly interoperable system can be delivered and implemented successfully.

Ofgem Flex Area	Number of Gaps Identified
Coordination	10
Flexibility Procurement	7
Dispatch & Control	4
Platform Transaction Settlement	2
Platforms Market Services	3
Analytics & Feedback	2

Coordination

- There is a **gap in clearly defining the distinction between a "standard" and an "implementation"**. Currently DSO/DNOs are using implementations of semi-proprietary APIs “implementations” from specific vendors (for example, Piclo for market engagement, and Smarter Grid Solutions for dispatch).

  With a view to the ENA’s stated desired outcomes, along with wider energy market principles around open market access, reducing barriers to new providers, and ensuring bill-payer value, a good solution would ideally be a more formal open “standard”, where different commercial vendors could implement their own solution and be procured through a competitive tender process, and be “hot-swapped” as required with consumer data transitioning between services. This would all be possible with a “standard”.

  To reach the end-state of a de jure standard, there are certain properties required around openness, interoperability and governance. It is likely to not be a usable outcome for ENA TWG for this process to select an off-the-shelf "product", since this would incorporate one vendor's Intellectual Property Rights (IPR), and present challenges for others to implement it, and give that vendor a significant commercial advantage over other providers in this space, indefinitely (for as long as that API remains the chosen API).

- There is a gap in the **technical process required to agree on the specification or definition of the (technology-agnostic) minimum viable information** required for a DSO/DNOs to adequately define what they wish to a) procure; b) dispatch; and c) settlement.

  To deliver a new API, or use an existing API, this will need to be standardised, to create an initial version that offers sufficient functionality. Consideration should be given to the level of technical detail required today, versus in future, to allow for future expansion.

  Forming complete and holistic requirements to enable interoperability to be delivered through implementation of a specification or standard is likely to require this, and this is likely to require, to some extent, a formal consultative/deliberative process. One of the opportunities presented by this would be the potential opportunity to surface information through interchange specifications which support FSPs in delivering innovative services in the future. As such, the definition of a set of minimum parameters and properties of flexibility services/contracts would be necessary. There is likely to be existing markets work in other ENA ONP groups which would set out these specifications, and which could be referred to.

  It is worth considering from a governance perspective the extent to which the information exchanged through requirements communication and availability responses may pre-determine the types of services able to be made available, and the ways to uplift this going forwards. By way of example, one DSO/DNOs described that their basic information used to agree a service was: units, time, volume and service direction.

  Some DSO/DNOs expressed a view in general that more information and visibility/understanding of services offered, and capacity to deliver them, was advantageous, and that they felt it may be beneficial for them to have access to this kind of information from FSPs on an ongoing basis. Other DSO/DNOs were less concerned and took a more regimented approach to this, with the view that if services were not delivered when dispatched, this would be handled through contractual provisions for non-delivery.

- There is a gap to be defined around **governance of the specification/standard of an API, and its longevity** - some future services not envisaged today are likely to require information to be exchanged which an API is not likely to contain if designed today. The gap is around how the standard evolves, and how adoption is encouraged, and who is ultimately able to decide to cease accepting older versions - this may be needed by DSO/DNOs to improve security or resilience (for example), but downstream FSPs/aggregators may not wish to adopt a new version if there are significant costs to rolling out software, or it would require new hardware to be deployed at edge devices.

  Similarly, if an API is to evolve over time, there is a gap around how the governance for this works – stakeholders highlighted the concern whereby one group (e.g. DSOs) ends up paying for the operation and development of a platform into the longer term, while another group (e.g. FSPs) could end up unwittingly increasing costs on the platform operators, such as through expecting long-term support for older APIs, or the ability to switch between different API call versions on a regular basis.

  This can be mitigated through robust governance and formal deprecation policies which are agreed by all users and stakeholders, but this does create an interoperability gap to be tracked, around how/whether such decisions are made, and the impact that this may have on FSPs or aggregators in requiring them to invest in developing upgraded API implementations and having these re-reviewed.

- There is a gap in **determining suitable high-level design principles for an API or specification, and the governance process around this**, particularly relating to cyber security, energy security and resilience, and where API structure may in some way be designed around commercial or deal structure factors, or create an implicit bias around some of these. For example, more technically complex or onerous design principles are likely to present a higher barrier to participation of smaller providers, especially FSPs that are less technically savvy.

  This could be mitigated through FSPs providing services through aggregators, but this is likely to also add a layer of extra profit margin to services procured, or reduce the level of competition and dynamism in the market. Significant quantities of aggregation may also reduce or eliminate effective competition “at-scale”, were an aggregator to invest in acquiring and aggregating existing capacity using investment funds, with a view to exploiting areas of long-term constraint. Having a relatively low barrier to implementation (with open access standards and methods of testing/validating integrations) would help to counter this.

- There is a (current) gap around **alignment between market engagement and procurement/dispatch channels** - this is where stakeholders feel there would be benefit in introducing a cohesive system. Today most DSO/DNOscarry out market engagement and procurement separately from dispatch. This gap appears to be on the critical path of many participants on both sides of the supply chain, and currently means most players are exploring or implementing split systems.

  Flexibility providers generally sought alignment around this, which raises a number of technical gaps around system design - would this require the market engagement API to be run as Critical National Infrastructure (CNI), if it was linked to the dispatch system? It would be possible to potentially run these as discrete systems, implementing portions of a commonly agreed API, but this would require more detailed architecture design, since implementing a single API split across multiple end systems will introduce security implications.

- There is a gap around **determining whether existing NGESO APIs, such as** Platform for Ancillary Services (PAS) **could be used for flexibility,** or not, thus avoiding a new API being required, but being considered alongside the potential commercial implications of this (such as cross-market price arbitrage).

  Some FSP stakeholders had basic familiarity with the balancing mechanism APIs, and questioned whether or not this would be a viable route for API-based dispatch, in order to avoid the creation of another, potentially overlapping, standard.

  There are likely to be wider questions at market regulation level, similar to those being undertaken by Ofgem at present through their formal statutory consultation [A1:1], if the same services were being offered through both the balancing mechanism, and a flexibility market.

- It is important to note that decisions about API usage should sit separately from market regulation concerns, but there could be implications more generally if it was a desired market outcome for providers to be able to offer capacity through both markets, and this may itself introduce requirements in flexibility and dispatch APIs (such as if there was a requirement to disclose capacity that was offered through other mechanisms, etc.)

- There is a **gap in scope around determining how far down in the architecture an API should reach** - should an API be specified out to the aggregator/FSP but not beyond? Or should an API reach down to home level? This will need to be decided, in order to scope and architect an API, and the associated security design. BSI PAS 1878/1879 have defined a flexible communication interface for residential consumer energy resources, however, the clear boundary is yet to be defined.

  One potential solution which FSPs were broadly in favour of was having a “protocol break” at the FSP or aggregator level – this would allow an FSP to interact with the DSO market and dispatch processes through a given ENA TWG API, then exert control or indirect control through price signalling over a more “domestic” protocol, such as OpenADR.

  There are advantages of a “protocol break” (i.e. a different API to homes than DSOs) since it's easier for bigger players to update their API integration code with a DSO in future, and it is hard to upgrade software on end consumer devices. In addition, this may create a more open and dynamic market where FSPs and aggregators can innovate at pace without requiring changes from DSOs.

- There is a potential **gap around the requirement for coordination between ESO and DNO/DSO prevent ambiguity over who is dispatching which assets** in response to which events. This would link into existing work on primacy, but stakeholder awareness of this work was limited, and accordingly further investigation may be required, and a deeper understanding of obligations for DSO/DNOs to coordinate with ESO before dispatching flexibility. This gap was highlighted in discussions with a DSO on the island of Ireland, which operates a different market and network structure, where they may coordinate with ESO to dispatch flexibility services. Nonetheless, if there is were to be a situation where there is ambiguity over whether a service is dispatched by DSO or ESO, this should be explored at API design stage, in order to prevent ambiguity from the perspective of an FSP over whether they have received all necessary authorisations to provide their flexibility service.

- There is a gap over **governance and accountability around platforms and infrastructure**. For example, if there is a shared centralised dispatch platform, owned/operated by a third-party organisation, who is responsible in the event a failure, and accordingly DSO/DNOs being unable to provide the requested power. There is both the short term (i.e. keeping the power on) and the longer-term (financial compensation to other market participants) implications to consider.

- In discussions with DSOs, some highlighted that where significant quantities of flexibility services were being procured in aggregation, these may start to reach thresholds of criticality, where extra security and resilience would be desired.

- For scalability, there is a requirement to provide **automatic linkage between Bid, Dispatch, and settlement orders**, as the existing manual process is unscalable. A number of FSPs highlighted the relatively long period between dispatch and settlement and ultimately payment.

Flexibility Procurement

- There is a **gap around defining a taxonomy of necessary data fields for a flexibility service,** if it were to need to be marketed as such and defined through an API. This may not be needed initially, but for a true offer/bid API, it is likely that a taxonomy would be beneficial to support understanding in an unambiguous way what is being offered in a machine-parseable form.

- There is likely to be a requirement to generate a minimum set of data fields required, as well as a process with robust governance to agree on what additional information is required, or optional. This is likely to impact on the services which can be offered in future by FSPs, or procured by DSOs, as the API would then become the constraining factor in information shared for flexibility information exchange, and deviating away from this would lose many of the benefits of interoperability sought by the ONP.

- The process and measure for determining how the taxonomy of information evolves will also be significant.

- There is a **gap around the nature of information communicated and available to customers of flexibility services** (i.e. DSOs), and how this would/should be implemented in an API or wider interoperable flexibility system – specifically around the nature of flexibility services offered, and the assets backing them up. This is likely to interact with wider market considerations around demand reduction-based flexibility, as opposed to short-term generation or feed-in-based flexibility. For example, a DSO discussed a scenario where they contracted an FSP to deliver a quantity of flexibility services. Upon further technical review, it became apparent to the DSO that the FSP would likely only be able to deliver a small fraction of the contracted capacity.

- Based on our stakeholder discussions, at present there are generally no contractual penalties for non-delivery of flexibility services, in order to open and expand the market to wider participation. In order to deliver a functioning and well-organised marketplace, it is likely that this gap needs to be addressed, such that there is greater clarity for flexibility customers as to the nature of dispatch, and likely availability, of flexibility services. For example, there may need to be a taxonomy of types of flexibility services available (e.g. demand turn-down, battery feed-in, generator turn-up, etc.), and dispatch types (e.g. directly controlled asset, indirectly controlled asset with acknowledgement by telephone, or price-signal based indirect consumer market signals).

- There is a **gap around the "extra" information that FSPs and consumers of an API are keen to have access to**, in order to determine what information is required to innovate "over the top" of what is available. This is likely to be determined by other ENA working groups, but the outcome will be required in order to enable an interoperable API specification which can deliver suitable information to permit a functional flexibility market.

- There is likely to be a gap in **clarification of the scope of interoperable APIs**, and whether this includes a “universal” API, where every electrical dispatch service in the country is brought under a common platform/API, from residential 3kW to ESO 1.5 GW. Or are separate levels/APIs employed at each level? The wider the scope of an API, the wider the requirements set. The cyber security considerations around an ESO-level flexibility dispatch system are likely to be far more significant than very small-scale dispatch of localised assets in the kW range, and some stakeholders expressed a view that flexibility dispatch would need to interconnect with ESO, and many stakeholders would be keen on a single linear system across all magnitudes of flexibility services

  There is also a **technical gap requiring top-level system design around the current proposals**, and how these would interact with transmission level assets. This could be assets connected to ESO, or to TOs. Currently, there is a lack of clarity around this requirement.

  If a different API is required between Transmission & Distribution, that will introduce a potential barrier to adoption, although on the other hand, if Transmission has significantly higher technical security requirements for those seeking to connect, this may present a barrier to adoption by smaller providers to the distribution market.

- There is also a wider question around **how at-scale aggregation of flexibility resources may behave in relation to existing DSO thresholds of concern** – some DSO stakeholders highlighted that they prefer to have direct control to large generators, to provide resilience outside of internet-based communications channels, and ensure that they are interacting with DSO-provided disconnection equipment at the site. This would prove problematic in a situation where assets were customer-owned and behind the meter. In the event that an aggregation provider has aggregated sufficient capacity to exceed thresholds of comfort for indirect dispatch in a DSO system however, and heightened security and resilience requirements would apply, this may introduce higher technical security requirements anyway – at which point it likely would make sense to adopt the same security measures and align Transmission and Distribution technical security requirements, to reduce the number of divergent sets of requirements applicable to FSPs.

Dispatch & Control

- There is a **gap around separation of responsibilities,** and how this would be implemented in an API-based dispatch and procurement system. DSO stakeholders expressed a view there may be a threshold at which point their requirements would change - potentially when one flex provider is contributing > 300 MW of flexibility services. This may introduce security requirements which would be likely to have an impact on the API or those implementing it, or create a point at which users would need to implement a second (more complex) API for higher security. This could present a barrier to market participation beyond 300 MW, and instead impact on market availability, or result in salami-slicing of the same flexibility capacity under different trading entities, masking the aggregation of risk of > 300 MW if the requirements are onerous. Non-transparent de-aggregation of inter-dependent flexibility capacity is likely to introduce security and resilience concerns for DSOs, and in stakeholder discussions, this was generally recognised to be important, but not something currently being managed.

  These requirements will need to be determined. There is also likely a gap around how identification of correlated/linked/double-accounted flexibility is delivered in the market. For example:

  a) If a flexibility provider offers aggregation of under 300 MW, but this is itself derived from a source of aggregation (say an EV charger maker) that itself is offering > 300 MW, is this detectable? Should this be something a DSO can become aware of via an API?

  b) If a flex provider offers aggregation that is correlated (i.e. offering flex services unwittingly from a car maker, and an EV charger), can this be detected and prevented, in order to understand the realistic available flexibility capacity?

  c) If an aggregator is aggregating downstream flex services offered by consumers, is there a way to gain visibility of the dependency chain - for example, if dispatch signal to customers is via an app-based push message, this places a significant level of dispatch through Google GCM and Apple APNS, which are likely to exceed 300 MW. Such a dispatch operation is therefore less likely to succeed during certain situations (i.e. cloud service outages) [A1:2].

- In discussions about dispatch requirements and perspectives, there was a clear **gap/risk around familiarity bias** – interviewees are inherently more able to discuss things they are more familiar with, rather than systems which yield the right solutions. There has been much less technical implementation of protocols and standards than envisaged at the outset of this work, particularly among FSPs – many are waiting for an API to be available before implementing it, as they recognise it makes little sense to integrate with an API that will be replaced in future. This means that requirements capture is inherently higher-level, as there are fewer lessons to be learned from past implementations. This was also reflected in the stakeholder workshops, where FSPs raised good questions about market structures and dynamics, rather than specific technical considerations around dispatch of services – highlighting that different stakeholders have different expectations in terms of an interoperable API.

- There is a gap in determining if a flexibility API should focus on domestic units aggregated together to form larger FSPs, or interact directly with domestic level assets. This question has generally arisen from FSPs and aggregators, keen to understand future market dynamics. It is likely that an API and standard can be designed to support both use-cases, without making a specific decision here.

- There is a gap between the safety and security of direct control over electrical assets, and the requirements for market-based flexibility services. Some DSO Cyber Security representatives highlighted the desire to have direct SCADA type inter-connection with larger FSPs beyond a certain threshold. However, this would only work for traditional "geographically consolidated" facilities, rather than distributed flexibility resources (i.e. aggregated assets). If there is a requirement here, this should be considered in relation to how the market is formed and structured, to avoid unintended consequences or imbalances. This gives rise to a gap around **whether and how security measures for grid stability would interact with general prohibitions on DSO/DNOs controlling assets "behind the meter"**. Where significant quantities of flexibility services are offered, this could create a situation (with a commensurate gap around an interoperable flexibility system) where, say, 300 MW of flexibility service is indirectly dispatched, but 300 MW of generation was directly dispatched with SCADA-type control.

Platform Transaction Settlement

- There is a gap around **how the multiple relevant flexibility market functions come together through APIs**. It is likely that market participants will require the ability to reconcile settlements back to dispatches, and dispatches back to orders. This is likely to benefit from thinking upfront between the different processes in the flexibility market - as getting this right at the outset is likely to make life easier for all market participants (i.e. knowing clearly and unambiguously which flex contract was dispatched), and support matching that through to settlement. This may introduce a requirement for a common/agreed "primary/foreign key" between systems to provide robust linkage.

- There is a gap around h**ow an API would facilitate rapid reconciliation for prompt payment** - which is a slow process today. FSPs and aggregators were keen to see this speed up, as it improves their cashflow, and this is likely to introduce cross-functional requirements into the settlement and metering processes. Improving the speed and accuracy of measurement data would help speed up this overall process.

Platforms Market Services

- There was a gap noted that a consistent digital journey, from tendering, PQQ, qualification, bidding, dispatch and settlement should be employed to reduce barriers to entry, and support FSPs in providing services across the whole market. However, this is currently being actioned by other ENA workstreams. Therefore this gap sits around the governance and harmonization of the output of other workstreams, and whether sufficient information will be available on the outputs of those workstreams to derive robust requirements for interoperability through an API for flexibility procurement and dispatch.

- One likely manifestation of this gap towards a requirement will be around **creating a single harmonised business process flow across all DSOs**, to allow a single API and standard to be usable – if each DSO implements a different business process, a common API will be unable to encompass the majority of this process, meaning that much of the critical process would have to be implemented outside of the automatable API. If there is not a critical mass of functionality in a standardised interoperable API, there will be little or no effective interoperability from the perspective of a market participant.

- Considering the SGAM model (IEC SRD 63200), it appears important for there to be suitable stage-gate reviews to work through the SGAM interoperability levels, working from the top down, as the lower layers codify and encode the higher layers in decisions and design patterns implemented. This would mean that, based on the SGAM interoperability model, the communication layer (L4) would require agreement and consensus around the business layer (L1), the function layer (L2), and the information layer (L3).

- There is a gap in the **meaning and definition of interoperability**, as applied to this current work, which is particularly apparent around platform market services, where the DSO is likely to be the customer of a specific platform, potentially implementing a given API. From the DSO perspective, the desired property may well be platform interchangeability (which is a superset of interoperability, and a higher bar to achieve), in order to provide viable alternatives with data portability and substitutability at product level. The specific requirements here should be identified, as they may go beyond mere API-level and data structure interoperability, towards system-level interchangeability if the goal is to prevent single-sourcing. Similarly, this may be necessary in order to ensure that platforms exhibit the same behaviours across different DSO/DNOsthat each deploy different implementations of a standard or API.

Analytics & Feedback

- There is a gap around **whether** (and the method through which, if it is chosen to) **market participants should signal confidence or certainty in their messages**, without otherwise constraining the commercial factors or structures of deals, since we assume that an API specification should not constrain the form or nature of commercial agreements or offer structures agreed with FSPs. In discussions with FSPs and DSOs, this was considered to be useful, although not universally, as some took the view that this would be handled contractually as non-delivery of flexibility services.

- There is a **gap around how data portability is implemented with respect to an interoperable API**, given that true interoperability in a multi-DSO system is likely to involve multiple platform implementations for the API, and FSPs and aggregators are likely to therefore need to interact with more than one of these. Maximising data portability and interchange format alignment is likely to make it easier for FSPs and aggregators to look at and learn from analytics and feedback. Similarly, DSO/DNOsare likely to also benefit from data portability, making it easier to change platform, as well as compare learnings and best practice.

Other Gaps Identified

In the course of carrying out stakeholder engagement, a number of other areas of gaps were identified. These are captured and explored in this section of the report.

CYBER SECURITY

The topic of security arose in a number of interviews, primarily with DSOs, who have a requirement to protect the security of UK CNI, which includes the energy system. It is important to note that the Ofgem flexibility platform tasks do not reference security. It is clear that there are a range of gaps around cyber security and the wider security and resilience posture.

This gap is documented in more detail in the separate document, titled "**ENA TWG Cyber Security Considerations**", which will be provided as part of Deliverable 2.

There is a **technical gap around how the technical security requirements would be defined for an API, and the governance around this**. This current piece of work is clearly not focused on answering such questions, but there are clearly some considerations around this from different stakeholders, as well as a range of challenges and gaps in how stakeholders discuss and manage security - many focusing on the compliance side of security (i.e. ISO27001, CE+), which are easier for larger organisations to deliver to, but which deliver limited technical assurance, versus those who were discussing specific requirements around contractual requirements for connections.

There is a **gap in clarity around requirements for security properties sought** - there are both transport layer, and data structure layer requirements. For example, one DSO highlighted the goal of non-repudiation of messages, which would require signatures on messages above the transport layer (since transport layer messages would not be usable for non-repudiation after receipt). In addition, there is a **general gap around the security architecture requirements and design**.

GOVERNANCE & ACCOUNTABILITY

While the Ofgem Flexibility Platform Task taxonomy[A1:3] considers a coordination function, a number of high-level discussions with stakeholders have highlighted **gaps around the wider governance requirements of a flexibility and dispatch** API or standard. This section is closely linked with the following section exploring the distinction between standards and APIs. DSO stakeholders have also specifically highlighted the requirement for accountability, specifically with regard to security and robustness of a flexibility platform.

It is likely that many aspects of this will fall far outside of the scope of the current ENA TWG's remit, but the approaches taken in addressing these factors are likely to impact on the design and implementation of a flexibility and dispatch API, as well as impacting on technical requirements.

In addition, there are a number of security and resilience-related factors around platform outages, where there will be a **layer of governance required in order to handle inter-organisational dynamics**, such as where third-party or public-cloud hosted platforms are used for APIs, or by FSPs to host their own client to the API.

Finally, there will also be **significant governance requirements in order to agree and determine an appropriate security model and threat model**, and agree an appropriate solution architecture (and act as steward for its longer-term evolution). This is similar to the governance questions highlighted in the following section exploring the governance aspects of standards development, in relation to API design and implementation.

Further Considerations for Interoperability

As a series of further considerations for interoperability, there may be gaps or requirements arising from the following areas:

- The implicit presumption that a common API can be defined down to message structure permits "interchangeability" needs to be carefully re-visited – interoperability and interchangeability are two different properties, and API selection is likely to be one of the last steps in a process of aligning a common business process for flexibility procurement, dispatch and settlement across multiple DSOs, since an API needs to be designed to support the implementation of that business process.

- The need for implementers (i.e. FSPs/aggregators) to be able to test and develop their API integrations against a non-production reference implementation of the flexibility API, including both current and future API versions.
- The need for conformance and interoperability testing, and governance around when updated software implementations require re-testing, or whether such testing is carried out by implementers and self-declared.
- The extent to which DSO/DNOsare willing to allow self-declaration of interoperability and conformance testing by FSPs/aggregators, versus requiring formal independent third party testing.
- The need for conformance and interoperability test suites to be well-defined, robust, and sufficient – from a security best practice perspective, a test suite would ideally cover positive testing (i.e. testing a range of valid inputs/outcomes), negative testing (i.e. testing a range of invalid inputs/outcomes), and fuzz testing (i.e. applying malformed and nonsensical inputs and checking for adequate rejection).
- The potential for in-band signalling through an API about future deprecation or breaking changes.
- The potential for an extensible API to introduce or support optional fields, which can be implemented by FSPs and users over time, without breaking support for existing implementations, and the types of testing that this would require.
- The extent to which an API/standard implementation may require changes to carefully-governed and managed CNI systems from DSOs, and add lead times to delivery of changes.
- The general cadence of change to an API or standard, the resourcing required to deliver this, and the extent to which FSPs and aggregators are willing to contribute time towards such endeavours, versus seeking something more minimal – some FSPs were willing to consider being involved in such an activity, but were concerned about time commitments and distracting them from their core business.
- Even if a standard for extensible information encapsulation were to be agreed, there would be a range of gaps and work required in order to determine what fields should be present in which messages, and ensuring that end-to-end functionality is delivered. It is important to note that from an interoperability perspective, interoperability requires aligned business processes, as well as technical alignment of information formats, data structures, and protocol flows.
- Interoperability can be defined at different levels of abstraction, and this will be important when setting requirements and communicating with stakeholders – there are also requirements around data portability which may not be fully encapsulated in standards or API specifications. For example, requiring FSPs to re-register and sign up to onboarding on new platforms after DSO/DNOs change platform vendor could present barriers to participation, and disruption. Data portability could help to address this, but this would require a more detailed and in-depth specification of internal information structures (going beyond merely an API), to enable that portability across vendors.
- The challenge faced by ENA ONP here is double-sided – generally one entity operates a platform, and offers an API which others consume. In this case, there are multiple DSO/DNOs(who will likely seek to separately procure implementations of an agreed API from multiple vendors), as well as multiple FSPs (who will likely implement or purchase an existing software integration with the agreed API). Since there will be diversity of provision on both sides of this marketplace, this creates potentially complex validation and interoperability testing (i.e. where a DSO may seek interoperability testing to have been carried out against their own platform), which may introduce barriers to participation for smaller FSPs.
- It is also important to note, when considering interoperability, that an API, as a manifestation of business processes, needs to consider not only the data structures themselves (i.e. the format for

serialising/exchanging data, akin to "nouns" and "adjectives" in vocabulary), but also the workflows and functions (i.e. the "verbs" and "adverbs") used. The workflows and functions need to match and enable the implementation of a business process – it is not merely sufficient to agree on data formats/structures, as this will form an incomplete API specification, which will not be able to be implemented and used to deliver flexibility services. From an SGAM (smart Grid Architecture Model) perspective, the verbs of an API implement the function layer (L2), the information contained within messages is the information layer (L3), and the actual architecture and implementation is the communication layer (L4). Each layer ultimately cascades requirements downwards, codifying the higher layers of the architecture.

Barriers to Implementation

From FSPs perspectives, the barriers to implementation were less technical and more commercial. Stakeholders, engaged or aware of the Open Networks Programme, were concerned around the longevity of any particular technical solution currently employed, as this workstream may replace the implemented solution. This resulted in them holding back until a solution is agreed. The potential market is then further fragmented into larger scale providers, who can engage in more commercially rewarding alternative auxiliary services markets, such as the balancing mechanism (BM). Smaller scale providers, who would otherwise represent the prime market participant, struggle make the business case internally to go through the cost and time of integrating with different Application Programming Interfaces (APIs) in different DSO/DNO regions, particularly for day-ahead dispatch, as existing non-automated solutions are "good enough". This may continue to be suitable for FSPs, however, it rapidly becomes unscalable for Distribution System Operator/ Distributed Network Operator (DSO/DSO). This creates an asymmetry in demands in the market.

Moreover, in order to prevent vendor lock-in, and associated potential monopolistic tendencies, DSO/DNOs need to be able to competitively tender for a suitable dispatch platform, and switch platforms as and when required. However, with lack of data portability, FSPs may require to re-register all their assets with the new platform. Further discouraging initial engagement particularly around platform renewal periods. Finally, all these smaller issues are compounded by the inherent lack of long-term certainty in the market, as reinforcement may still be employed. This is compared against large scale generators in the capacity market who could be assured a price per MWh for the lifetime of the asset.

These barriers are primarily commercial rather than any existing platform being inherently too technically complex, or difficult to implement, as these issues could be overcome with sufficient return on investment. However, as the primary objective of flexibility services is to avoid network reinforcement. It is important to make flexibility a viable alternative, which is only possible with sufficient market liquidity.

From a DSO/DNO perspective, the main barrier to implementation are cyber security considerations, and coordination. Integration with the rest of the eco-system, and supporting network infrastructure. For example, in a real-time flexibility system, real-time/live analytical data will be required from the physical copper infrastructure in the ground to enable real-time responsiveness, and the lowest cost to consumers for a dynamic type product. Sustain, on the other hand, could be procured and deployed more easily due to the more static requirements.

References

[A1:1] https://www.ofgem.gov.uk/publications/ofgem-launches-consultation-balancing-mechanism-reforms-protect-consumers

[A1:2] https://www.reuters.com/technology/amazon-says-multiple-cloud-services-down-users-2023-06-13/

[A1:3] https://www.ofgem.gov.uk/sites/default/files/docs/2019/09/ofgem_fi_flexibility_platforms_in_electricity_markets.pdf

[A1:4] https://standards.ieee.org/beyond-standards/what-are-standards-why-are-they-important/

[A1:5] https://standards.ieee.org/faqs/dominance/

[A1:6] https://standards.ieee.org/faqs/affiliation/

[A1:7] https://standards.ieee.org/about/policies/

Appendix 2 : D2 Minimum Requirements

Executive Summary

This document outlines the minimum technical requirements a flexibility service dispatch interface would require in order to provide a minimum viable product (MVP). This information is intended to inform the process of selection or development of a common standard. This would enable an interoperable ecosystem of flexibility service providers (FSPs), vendors, and DNOs/DSOs to engage in reliable and automated flexibility services.

We strongly recommend that the ENA Open Networks Flexibility Services Technical Working Group (TWG) take into consideration the contents of this document alongside Deliverable 1 (Interoperability Gaps), which sets out the background context to this report, derived from a series of stakeholder engagement exercises used to inform a gap analysis for the information required to inform activities selecting or developing standards or a protocol for the dispatch of flexibility services.

Deliverable 3 (Suitability of Existing Protocols & Implementations) will set out a more detailed set of recommendations and selection criteria for a world incorporating interoperable flexibility service dispatch, alongside a wider interoperable flexibility service ecosystem and marketplace.

The report first explores the required minimum architectural decisions which need to be made for a communication standard. This determines the required components, and the associated communication parameters, to share data around the different system components. As this particular workstream focuses on dispatch we highlighted different potential architectures for a dispatch system to demonstrate how the requirements vary depending on the architecture.

This was followed up by an analysis of the existing TWG key service parameters in a dispatch system, and considerations to format and encoding schemes. This highlighted a small number of recommendations including adding additional key service parameters of, currency, type of energy source and globally unique identifiers to aid with coordination between all the various components in the wider ecosystem.

This was followed by a section outlining the requirements for cyber security considerations, and by request from the TWG, we have included alongside this document some early (pre-release) technical recommendations from our lead security architect, who has experience in designing and developing security solutions for Government and Critical National Infrastructure sectors.

Finally, the report details the minimum testing requirements for an interoperable standard, including positive/negative testing, fuzzing and Interoperability/integration testing, and to enable implementers to easily determine where interop issues may occur. Moreover, we provided recommendation of creating a sandbox testing environment for reducing barriers to adoption and uptake by FSPs, and for them to build and test their implementations against.

Each of these four areas – architecture, communication content, security and testing regimes, combined would provide the minimum requirements for any large-scale interoperable system.

Architectural Requirements

The first stage in developing any system is understanding and agreeing on a common view of the architecture – that is to say, understanding the different components which have to communicate, and at fundamental level, how they will communicate. As this workstream specifically focuses on dispatch, we first explore the simplest version of this communication, as shown in Figure 1. The operator will send dispatch messages to the Flexibility Service Provider (FSP), and the FSP will respond with a physical change in electrical generation or consumption to match the operator's request. Operator in this context is defined as either a distribution network operator (DNO), Distribution System Operator (DSO) or Future System Operator (FSO). The content of these messages can then be defined, to include things such as start time, duration and requested capacity. The format, units and interpretation of these messages then needs to be agreed between the FSPs and the Operators to enable interoperable communication between both parties.

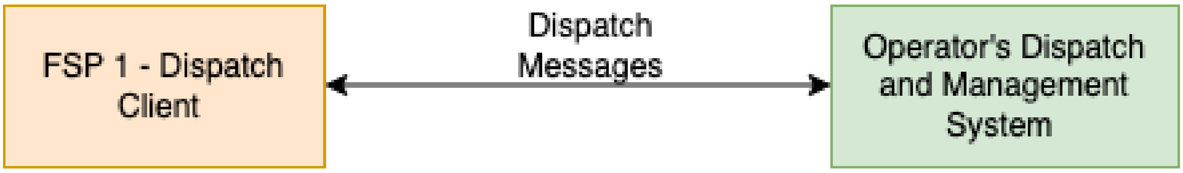

Figure 1: Simple Dispatch Architectural Diagram

Even at this fundamental level (as shown in Figure 1), there is an immediate fundamental technical implementation decision required – **should the FSP or the operator's dispatch and management system be the "server" in a traditional REST-based client/server Application Programming Interface (API) architecture?**

Answering this robustly will be a key API requirement before other architecture decisions take place, and will impact on the evaluation of potential architectures, APIs and standards. This will drive other technical and architectural requirements, but indicates the kind of fundamental decisions which need to be taken before selecting protocols, standards or APIs.

Conventionally in API design, the requester of a service would be the client in a client/server architecture, with the provider of the service being the server – this means that a long-running server service can listen proactively for an incoming request, and trigger an interrupt to service the request when it is received, act upon the request, and send a response back to the initiator of the request in real-time to acknowledge and confirm the status of the request. This is how an HTTP client works – the user's browser makes a request of the web server – the browser is the client, making a request of the server. Following this logic, an FSP would run a server, and the DNO would be the client in this architecture.

This raises several design and security considerations which should be explored as part of making this decision.

If the network operator were to act as the server, then an FSP needs to either poll the server periodically to look for a new dispatch request, carry out TCP long-polling, regular short-lived heartbeat requests to look for pending dispatches, or establish, through a protocol such as WebSockets or similar, a route for the server (DSO) to initiate communication with the client (FSP). This would be counter-intuitive compared with conventional logic.

Conversely, if the FSP was to act as the server, they would need to take responsibility for implementation and enforcement of a number of security measures (i.e. TLS verification of communications and similar), and would need to operate an exposed attack surface over the internet. There would be a need for DSOs to enrol the appropriate FSP URLs into their dispatch clients, and handle error states at the DSO end where a dispatch request failed due to a DNS or IP routing failure. The FSP would also need to ensure that their external attack surface (including their dispatch service API) was patched and maintained, since it would be exposed outwardly, and to ensure that they only respond to correct requests from the authorised network operator. This would place significant business logic responsibility for validation and authorization on the FSP to implement correctly, and would be likely to present effective or practical barriers in FSPs entering the market, with this level of responsibility on them to host and operate a resilient API like this, in the face of denial of service attacks and similar, which may be directed towards them. Many FSPs may also struggle to access and afford the skilled personnel needed to maintain and secure such a service on an ongoing basis.

It is therefore a **requirement that the ENA TWG and stakeholders determine the directionality of any API, taking into account security architecture and design, as well as the means through which an FSP receives dispatch instructions.** This should take into account the wider architecture of a flexibility system, in order to create a cohesive and consistent approach to system architecture.

As demonstrated by API directionality, even this simple example (Figure 1) of two components communicating can change considerably depending on the architectural decisions. Accordingly, three potential operating models of this simple FSP to operator interaction are detailed below. The first of these architectural design patterns, as shown in Figure 2, would create a single centralised national common dispatch and management system. Every FSP and electrical operator wishing to engage in the flexibility market would connect to this common system and use it as an intermediary. The advantage of this design is that it creates a common system for all operators and FSPs, overcoming the currently fragmented nature of DNO regions, and reducing the barrier to entry for FSPs to grow their portfolios new regions. It would also be easier for new entrants as commonality would grow the overall ecosystem and enable a new network of supporting companies to assist smaller FSPs to engage. This would reduce integration and interoperability testing requirements, as every market participant knows the (one) implementation they need to be able to work with.

The disadvantage of this model is the governance, security and accountably of this shared system. A shared governance model would allow for joint advancement of functionality and features; however, this would likely be constrained by requiring group consensus. From an accountably perspective, clear legal agreements would have to be agreed between operators and providers of this common platform as to determine who is responsible if operational failure resulted in loss of power for consumers. Operational uptime also poses some consideration as in the event of an operational outage all regions would be affected, thereby, creating a disproportionate issue and additional operational reserves would have to be held than if this was segmented.

This would also create a single dispatch platform across all DNOs, which would be a significant and attractive target for foreign state actors. Such a platform would, from a security perspective, needs to be designed to resist attack by the most determined and capable foreign intelligence services and state-aligned attackers, given its centralised nature across all distribution network operators, and thus the potential for use to cause cascaded impact across the wider UK energy system as a means of a high-profile central attack on the UK's energy networks, without the adversary crossing into directly attacking OT systems.

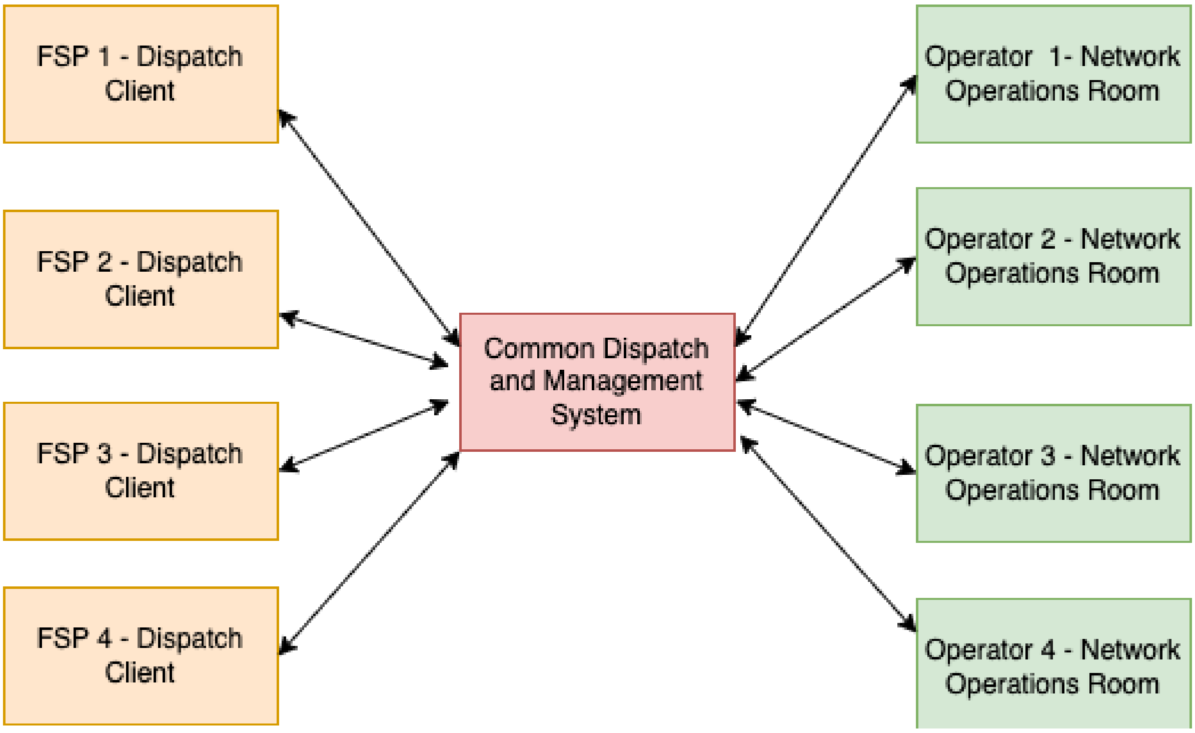
FSP 1 - Dispatch Client
FSP 2 - Dispatch Client
FSP 3 - Dispatch Client
FSP 4 - Dispatch Client
Common Dispatch and Management System
Operator 1- Network Operations Room
Operator 2 - Network Operations Room
Operator 3 - Network Operations Room
Operator 4 - Network Operations Room

Figure 2 - Common Dispatch Management Systems

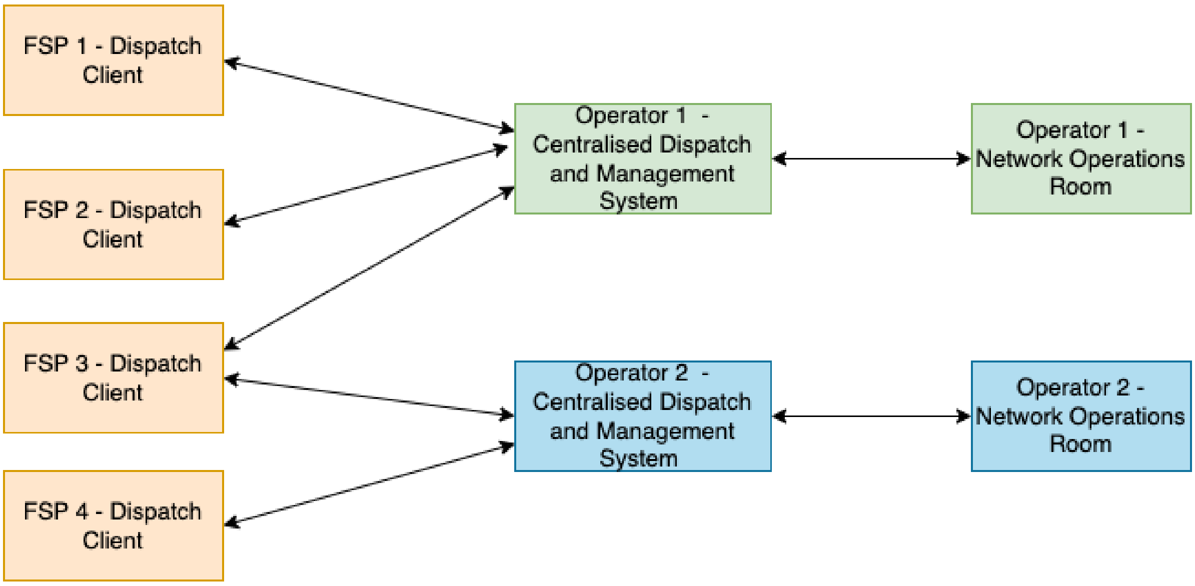
FSP 1 - Dispatch Client
FSP 2 - Dispatch Client
FSP 3 - Dispatch Client
FSP 4 - Dispatch Client
Operator 1 - Centralised Dispatch and Management System
Operator 2 - Centralised Dispatch and Management System
Operator 1 - Network Operations Room
Operator 2 - Network Operations Room

Figure 3: Region Specific Dispatch Systems

In this second potential model, as shown in Figure 3, each operator would control and deploy their own instance of a dispatch system. This system would follow a common standard so that all the functionality and capabilities, would be common throughout the industry to ensure interoperability and prevent isolation. Through a competitive tendering process these systems could be created by a common vendor or disperse vendors. Provided all the platforms followed a common standard operators would be able to switch systems and migrate FSPs data over to a new system to prevent vendor lock-In, with minimal interruption for FSPs.

The advantage of this model is the operators would have additional oversight and control over the dispatch platform, including maintaining and monitoring based on their individual constraints. The disadvantage is that FSPs operating in multiple DNO regions would have to interact with multiple platforms, and potentially support and maintain multiple parallel versions with different operators. For example, operator 1 could update to the latest standard version/security patch, before operator 2 FSPs. Accordingly, it would be best practice for FSPs to fragment their systems based on operator region to avoid incompatibility issues, however, this creates additional overhead for FSPs. This would remain the case even if backwards & forwards compatibility were implemented, as there would remain the need to integration test and validate against each version and implementation of a specification.

One consideration of this model is that, as shown with FSP 3 in Figure 3, it would, absent other rules or restrictions, be possible for an FSP to use a single instance of a dispatch client to communicate with multiple network operators. This could create an aggregation risk, whereby a single point of contact in the FSP network is interacting with multiple network operators. This should be considered carefully from an architecture and security perspective, both as a way to potentially ease the technical and operational burden on FSPs, as well as a potential cyber risk aggregation point, where a successful attack could gain a foothold in multiple network operators' flexibility supply chain.

Finally, each operator could deploy a provider-sharded dispatch management system, as shown in Figure 4. This model would provide a 1:1 relationship between each FSP and a Dispatch Frontend Instance (DFI). Each DFI would map to a Centralized Dispatch and Management System (CDMS) which would manage all the dispatch operations. Once a FSP is onboarded as a flexibility provider to a network operator, they would be granted access to a unique dispatch front-end instance, which itself is connected to the respective network operator's CDMS. In essence, in this architecture, there is a "factory" concept, where the Frontend Instance Factory is instructed to create a new instance of a DFI for each new FSP. The FSP would then connect to this instance of the DFI, and the CDMS will be able to issue instructions through the DFI to the FSP. The Network Operations Room (with oversight over the overall electrical network) would detect a requirement for local generation and instruct the CDMS to meet these requirements. The CDMS would then decide on the optimal combination of FSPs, and issue dispatch instructions to each of the required FSPs through the matching DFI.

In such an architecture, the level of isolation and segmentation between DFIs could be determined by an operator's security posture – some may be software-isolated slices, while other more critical ones could be deployed in hardware-isolated environments.

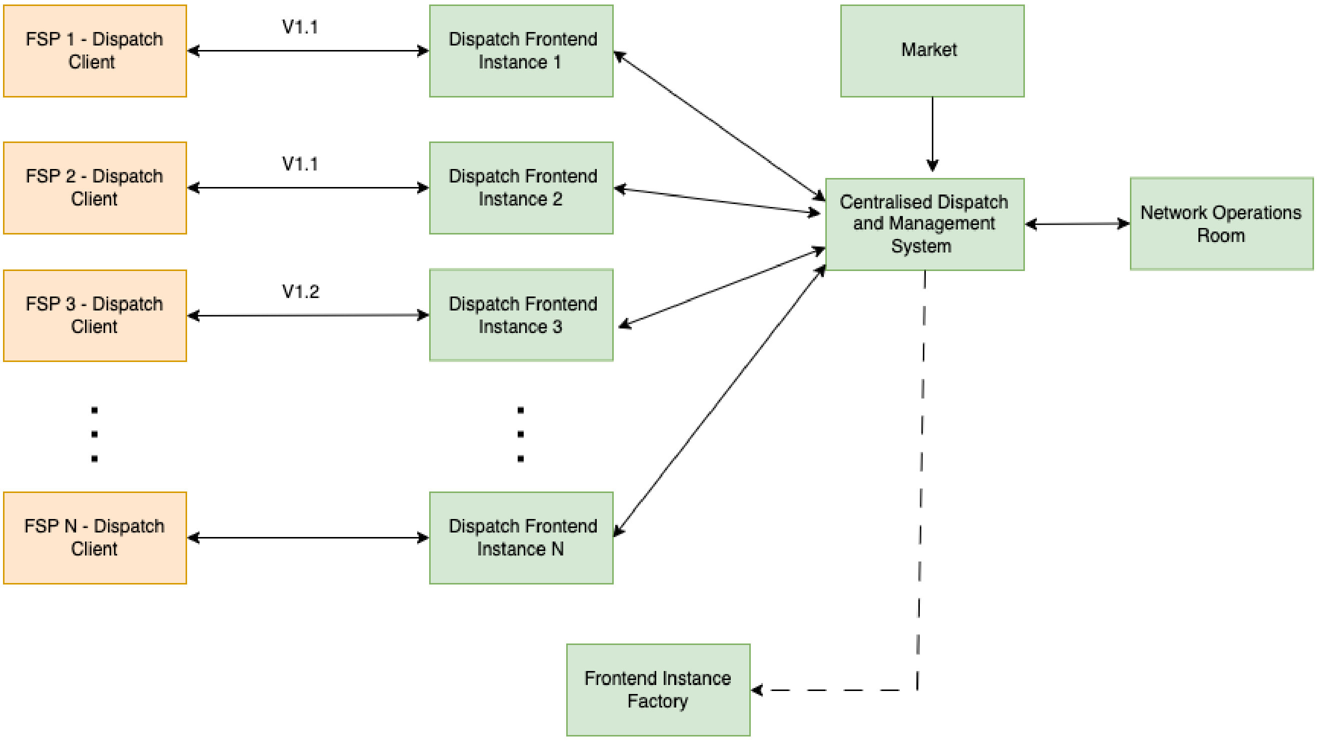
FSP 1 - Dispatch Client
V1.1
Dispatch Frontend Instance 1
Market
FSP 2 - Dispatch Client
V1.1
Dispatch Frontend Instance 2
Centralised Dispatch and Management System
Network Operations Room
FSP 3 - Dispatch Client
V1.2
Dispatch Frontend Instance 3
FSP N - Dispatch Client
Dispatch Frontend Instance N
Frontend Instance Factory

Figure 4 – A Single Operator's Sharded Dispatch System

This segmentation of different providers increases cyber security protection by isolating potential exposure to operators, and limiting the potential for a compromised component to cause wider impact on the system.

The separation of the frontend and the CDMS into separate components would also enable decoupled version updates to take place, as this split moves the update requirements from FSP <-> CDMS, to FSP <-> Frontend <-> CDMS, therefore, each of these links can be updated independently. This reduces the barrier to entry for smaller FSPs and would improve longevity, provided the Frontend <-> CDMS link maintained backwards compatibility. Early FSP adopters could continue to engage on Version 1.1, and remain on that version without any changes while the rest of the eco-system grows around them to V2, V3, V4 etc. Additionally, as the Frontend <-> CDMS link is entirely within the control of the operator it would be easier to update as required, as FSPs wouldn't need to be involved. As new FSPs come online they would be required to connect in at the latest version, and when FSPs require a new function, they would be able to update in their own time, but this would provide a slow but steady progression forward without limiting innovation or leaving parties behind. The disadvantage of this design it is more complex for operators to implement and maintain.

This concept can be extended to include multiple network operators, as shown in Figure 5. The FSPs in the middle of the diagram each connect to different DFIs using different version numbers to enable FSPs to engage in the most markets possible while limiting “re-inventing the wheel”. There are various models demonstrated in Figure 5, where FSPs may, without any other rules, choose to implement. Firstly, FSP 1 operates two client systems each of which connect into a different operator. FSP 2 on the other hand uses a single client to connect to two different systems as they are both operating on the same version number. FSP 3, operates two identical client systems to engage with operators’ systems but have employed segmentation between the systems for additional security. FSP N, operates three segmented clients, however, uses commonality between the two V1.2 systems to link to both the ESO and an operator, enabling them to engage in multiple markets seamlessly.

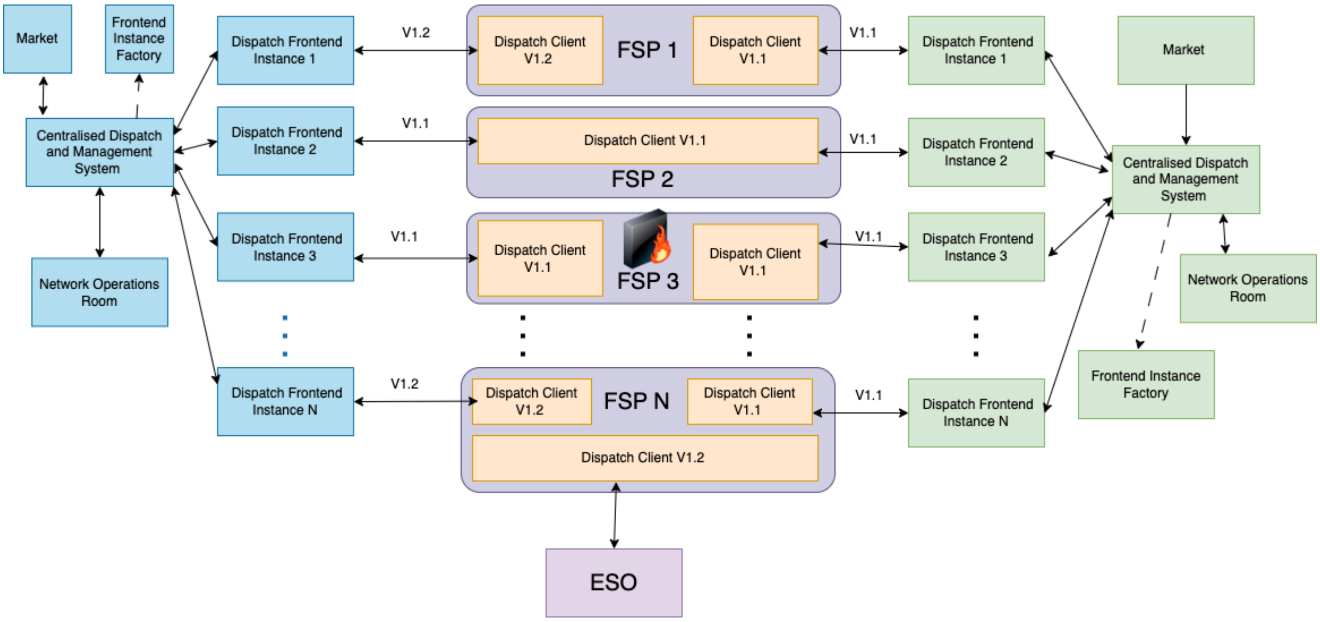
Market
Frontend Instance Factory
Centralised Dispatch and Management System
Network Operations Room
Dispatch Frontend Instance 1
Dispatch Frontend Instance 2
Dispatch Frontend Instance 3
Dispatch Frontend Instance N
V1.2
V1.1
V1.1
V1.2
Dispatch Client V1.2
FSP 1
Dispatch Client V1.1
Dispatch Client V1.1
FSP 2
Dispatch Client V1.1
FSP 3
Dispatch Client V1.1
Dispatch Client V1.2
FSP N
Dispatch Client V1.1
Dispatch Client V1.2
V1.1
V1.1
V1.1
V1.1
Dispatch Frontend Instance 1
Dispatch Frontend Instance 2
Dispatch Frontend Instance 3
Dispatch Frontend Instance N
Market
Centralised Dispatch and Management System
Network Operations Room
Frontend Instance Factory
ESO

Figure 5: Provider-Sharded Architecture with Multiple Operators

While the minimum required message content between the FSP and the Dispatch system remains consistent within all the architectures, each of these designs choices would result in a unique set of minimum architectural requirements, and accordingly would affect the overall design of the architecture. Additionally, as set out previously, the wider technical architectural decisions around REST API implementation also need to be considered – such as whether FSPs are indeed dispatch clients, or whether FSPs implement the REST API server to be consumed by a DSO client.

Moreover, as additional external systems are added the minimum message content requirements may vary to provide linkages. An example, of this is shown in Figure 6, which outlines a fuller whole eco-system view of flexibility services including Ofgem’s proposed flexibility service marketplace module, settlement and monitoring modules. Combined these modules present an architecture including the three core functions of flexibility services – market engagement, dispatch and settlement, all of which would need to be considered when designing the minimum requirements for a dispatch system. As each of these external systems interact with dispatch components in some way, and therefore, require passing data effectively.

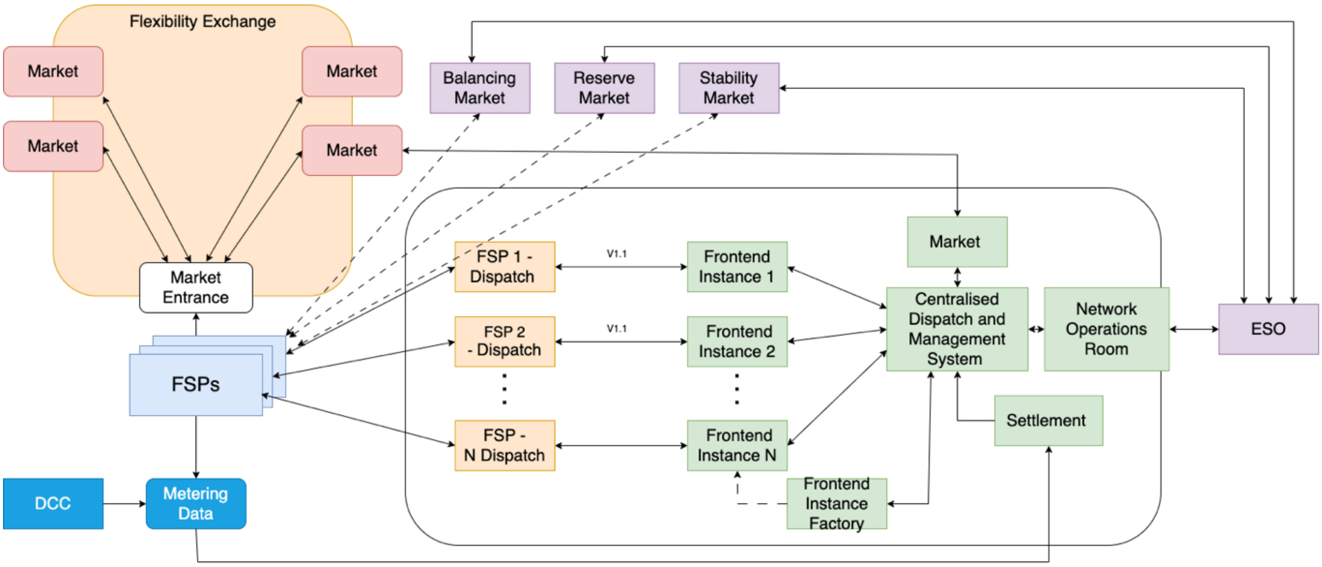

Figure 6 6 – Market, Dispatch and Settlement Functionary Combined

The most common way of linking these systems together would be employing globally unique message identifiers which would link to an entire lifecycle for each event. For example, when an operator would tender for capacity in a local area, the FSP would respond to this tender. Accepted offers would transition to the dispatch management system, which would dispatch as required. The FSPs demand response would be measured against the requested capacity variation, and compared to the original request and the appropriate payment would be made. To achieve this smoothly it would be required to have a unique identifier to link these stages. There is therefore a **requirement to identify appropriate primary key and foreign key “references”, and establish through the protocol which entity defines these, and their uniqueness constraints (global energy-system level uniqueness, vs DSO-unique, vs FSP-unique).** In addition, this must be designed to trade off convenience with maintaining a single source of truth, and what the failure mode would be, in the event that a DSO or FSP is unable to access their full IT estate at a given time.

By way of example, we would propose that a dispatch request should be executed with reference to a specific contract identifier (and that this contract identifier should be name-spaced by the DSO issuing it, with the DSO setting the identifier). From a practical perspective, however, dispatching against a contract requires a lookup process at both ends, in order to reconcile the action - for the DSO, based on the asset or service they require, what the correct contract reference to dispatch is, and for the FSP, to translate from a contract reference to an asset to be dispatched. This would therefore suggest that, while from a normative data structure and business process perspective, a dispatch operation is carried out against a contract (as the subject of the API operation), the information communicated should also contain the DSO-defined asset identifier being dispatched. This introduces a layer of redundancy and resilience – the FSP could dispatch without access to their own record of all contracts, in the event they had an outage affecting their contract store. Similarly, a DSO could issue a manual dispatch through knowledge of the identifier they were seeking to dispatch, without access to their contract records. This introduces a second-order layer of complexity around reconciliation and validation – in the event that a dispatch against a contract does not match the contracted flexibility asset, the standard or protocol will need to define how this situation should be handled, and whether the dispatch should be rejected. There would also be a third-order layer of complexity around handling reconciliation and settlement where a

FSP was unable to validate the asset being dispatched against the contract, and a discrepancy is later identified – such a process is likely to be relatively manual, or rely on after-the-fact manual reconciliation by an independent system on past dispatches and contracts.

All these architectural design choices, and their associated cyber security considerations, will need to be made to determine the minimum requirements. It is worth considering, as electrical network operators would have greater responsibility for the cyber security and governance of their flexibility platforms, than their flexibility service providers, this will likely have to be security biased on the operator. Accordingly, by request of the ENA TWG, we enclose, **alongside this requirements document, a technical preview of a component of D3, encompassing technical security requirements and guidance towards the secure architecture and implementation of Critical National Infrastructure (CNI) grade flexibility services**. This is subject to review and update based in the final deliverable of this programme of work, and should be considered to introduce a number of additional requirements to ensure the security of the design of any API, standard or specification, as well as around the wider architecture and threat posture. We suggest it is an initial **requirement that a security threat model and posture be agreed for the whole flexibility dispatch ecosystem**, such that this threat model can be used to assess whether particular solutions provide appropriate solutions.

While the focus of this document is on the dispatch of flexibility services, it is also important to consider the wider consequences of this, including the longer-term implications of design and architecture decisions, which may impact the security posture of downstream suppliers and providers of flexibility services. In particular, there are a number of questions and challenges around thresholds of significance and criticality, which should be carefully explored on an individual electrical network basis, taking into account the downstream communications and dispatch protocols implemented. These are likely to result in thresholds or limits at which point alternative measures or steps need to be implemented by FSPs – consideration should be given to how these thresholds can be communicated to API participants (perhaps via an API endpoint) for transparency.

The above also demonstrates how **architectural choices and decisions** (e.g. around whether one or multiple instances of a flexibility dispatch platform are run) will **vary the technical requirements specification required for fields in data structures** which will be communicated in a standardised data interchange format (e.g. uniqueness constraints being global, per-DSO, per-FSP, or per-DSO-per-FSP), as well as the approach to selection of values used to link data structures.

Communications and Data Representation Requirements

The WS1A-P3 work in 2022 outlined a series of key service parameters for the different phases of dispatch. The full detail for these parameters is available in the ENA document ON22 WS1A-P3 Key Service Parameters [A2:1].

The structure of these messages followed a life-cycle approach of a dispatch service, ranging from Declaration of availability, Acceptance of offered services, Scheduling of services, Instruction of services, Cease instruction, Variation of dispatched services, Status monitoring of services and cancelation of dispatch request. Each of these life cycle stages is then further broken down by the required messages, and their associated parameters. This method effectively considers and captures the requirements at each stage of the life cycle.

From our review of the gap analysis carried out for D1, through stakeholder interviews, the key service parameters outlined in the ON22-WS1A-P3 Key Service Parameters document appears to be sufficient in isolation, albeit subject to some of the notes we have observed on how service parameters are NOT a protocol or API spec, and there is a design process to go from this to anything implementable.

Many of the fields feature an optional parameter along with an example of the data type and a verbal description of intended functionality. We would recommend adding an additional parameter for version number in every message, so that all messages transmitted to the API are versioned. We would also highlight the usage of Megawatts (MW) as the default base measurement unit, this is currently in-line with parallel work undertake by the settlement group. However, this is deployed based on the assumption of ESO selecting to implement this dispatch standard. If ESO, decide to not implement this standard, Kilowatts (kW) may be a more suitable measurement unit.

This represents an outstanding issue for required clarity, if the ESO/FSO are planning on implementing this flexibility service communication standard to attempt to create a universal communication standard for all DNOs/DSOs and the FSO. This understanding will define the required message data content, and accordingly the minimum viable requirements. However, we have identified a number of requirements/considerations which we believe should be incorporated regardless of the ESO/FSO's decision.

In each of the proposed dispatch messages: Request for services, Declaration of availability, Acceptance of offered services, Scheduling of services, Instruction of services, Cease instruction, Variation, Status, Post-action and Cancellation. We would recommend that each message should contain a unique message identifier, which should be unique to the issuing party, and the issuing party should be identified within each message. This helps to avoid duplicate messages being processed multiple times.

Additionally, under the "declaration of availability" phase of dispatch we would additionally recommend including four additional fields.

- Type of energy source (hydro, battery, demand turndown etc) and a unique identifier for each class of asset, which can be used group together to prevent a critical dependency on a single manufacturer or software.
- An optional field of the predicted amount of Green House Gases (CHG) generated as a future differentiation unit.
- A representation of the currency of service provider's offered price, using standard three letter ISO symbols, e.g. GBP, EUR. Consideration should be given to the unit of currency, and potential for avoiding use of floating point and decimal values by using the lowest available unit of currency.
- An optional confidence interval denoting how confident the FSP is that they will be able to provide the advertised service, noting that this will require governance and design to prevent everyone declaring 100% confidence – this could be a future enhancement and would not be a baseline requirement.

Under the "Instruction of services" phase of dispatch, we would additionally suggest including:

- An asset group ID to tie the previously proposed group of asset to the instruction of services, or include an additional message for the FSP to respond in real time the current make-up of the package they're actioning to start generation or curtailment, taking into account the resilience vs security trade-off of having this information aggregated.

Even with these requirements, there will still be **other implementation requirements to consider at-scale across other messages, where requirements may not yet be apparent** – unique message identifiers may be best implemented as message counters, which deliver replay protection and message sequencing and ordering, without requiring verifiers to maintain a significant cache of previously seen messages, in order to carry out an iterative search for a given message identifier. While this is not necessarily a protocol requirement, it is a more practical implementation consideration that should be taken into account when designing or selecting a protocol.

For each value in any message, it will be a **requirement to stipulate the exact representation format and serialisation format of values.** Where units are used, these should be conveyed (ideally in the name of a field), in order to avoid all values becoming strings (and complicating parsing). For example, if pricing units is set to MWh and energy capacity is set to 50 MWh, this adds a level of complexity that needs unravelled, since pricing units is defined under ENA's taxonomy to be optional. Consideration should also be given as to whether the above is truly a minimum requirement, or whether the concept of setting out service units (delta) and service volume separately will add unnecessary complexity at this stage.

For example, this firstly assumes that all service procurement is delta-based. If this is a suitable long-term assumption, this is likely to be acceptable. In particular, this may preclude this dispatch protocol from being used for other purposes in the network in future where absolute dispatch signalling for DG or curtailment may be desired. Secondly, this makes use of a dynamically defined service units field, which will require a way to communicate this. This could be by string or an enumerated type, but adding a new unit would require every implementation to understand the new unit. It is likely to be easier, for a minimum requirement, to determine the lowest permitted unit of service dispatch, and to implement this for the first version. This is the same method taken by the Bitcoin blockchain, which considers all transactions as an integer number of "satoshis", rather than attempting to handle decimal or fractional quantities. In this context, if the lowest dispatchable quantity was 1 kWh, then a field of "Service_Volume_kWh" could be defined, avoiding the need for initial implementations to support the complexity of variable units.

Data Structures and Encoding Schemes

One of the most important aspects of any standard or API is the proper definition of data structures and representations – this ultimately defines what information can be represented and communicated through an API, between network operators and flexibility service providers. Data structures and representations must be agreed in advance, so that different implementations can understand messages from each other, in whatever format they are encoded or encapsulated in. As a simple example, an asset identification number might be considered an integer by one party, and a string by another party. This could create a breaking interoperability failure if one party added a non-numerical character to the value. Similarly, this could cause a failure if the integer number represented exceeded a 32 or 64 bit integer field value. Therefore, it is a fundamental **requirement that all data types and structures are defined and documented unambiguously and, in an implementation-neutral way**, at wire-format level (rather than programming language level). This enables FSPs and network operators to understand the expected format and implement it in whatever language they choose.

To give two specific examples where the importance of this is especially pronounced, consider a decimal number, and a date/time field:

Where decimal numbers are communicated, it is important to consider how they are encoded and represented – a common means of representation of a decimal number is IEEE754-2019 for decimal floating-point numbers. This standard defines 32, 64 and 128-bit decimal floating point numbers [A2:2]. IEEE floating point representations are widely used, but do not give numerically precise decimal representations – while the standard defines rules for rounding, there are a number of edge cases in implementations, where floating point representations may differ from expected decimal representations.

For example, in some implementations of floating point addition:

```
0.1 + 0.2 = 0.30000000000000004
```

While this is not a material difference under most units likely to be used in a dispatch or settlement system, floating point numbers are likely to yield minor differences from expected values, and comparison operations must be carefully implemented to avoid failures in reconciliation of values [A2:3]. While this may appear trivial, this could become problematic if network operators or FSPs were to implement their own business logic in different ways, resulting in boolean comparisons failing on edge cases.

For example, a dispatch request for 0.1 MW by a DNO might be erroneously rejected by a FSP which was working in floating point numbers, and was only able to offer 0.099999997 MW – this is less than the dispatch request, so the request cannot be fulfilled by strict interpretation of these numbers. This leads to a r**equirement for robust definition of the correct format and encoding scheme for all values** communicated through an API or specification of an API, in order that there is no ambiguity. This also extends to the correct format and encoding scheme being used for internal representations of such values, or a robust testing regime, such that any relevant internal business logic that would impact on the reliability of the flexibility dispatch or response system also uses a suitable representation, to prevent unintended downstream consequences of format selection. It is important to note that, generally, standards and APIs do not define requirements on external functions and how they operate – the correct functionality should be tested using adversarially-selected field values through end-to-end interoperability testing (as described later), to ensure that inter-connected systems behave correctly when presented with data fields.

In various places where dates/times are specified, it is **essential to specify the exact format to be used for dates and times, including time zones**. This is critical as date and time inconsistencies and misinterpretations have historically caused numerous interoperability issues. This is due to the various different date time formats employed, e.g. “10/09/23” (as a string) could represent an arbitrary string, or be decoded to represent the 10th of September 2023, or the 9th of October 2023, or the 10th of September 2123, depending on regional settings and assumptions around this ambiguous date. These issues permeate into programming languages and supporting libraries. Similarly, operating across time zones or seasonal time changes, e.g., British Summer Time (BST) and Greenwich Mean Time (GMT) can cause coordination issues. For example, if a dispatch order is setup within GMT, but executed in BST the dispatch could be an hour out of sync. A non-time zone aware implementation may naively assume that all times are represented in local time or UTC, which would be incompatible with a system correctly indicating a time zone.

APIs have traditionally solved this issue by employing either Unix/Epoch Time or ISO 8601 time. Unix/Epoch is a 32-bit integer representation of the number of seconds since January 1, 1970 (midnight UTC/GMT), and is displayed as a large integer number, for example, 1694698011. This gives second level precision but is not easily human readable. ISO-8601 on the other hand follows a human readable format of year-month-day, followed by the time and any time zone offset, for example, 2023-09-14T13:21:52Z. A disadvantage of this it is slightly more verbose, but it is human-readable. This verbosity often does not benefit inter-machine communication as being human readable is not usually a requirement. It may however provide a useful redundancy for dates and times to be readable in messages, especially where they are relied on in contractual or other disputes.

Many APIs would implement Unix/Epoch Time. However, a constraint of the most common 32-bit implementations of the current Unix/Epoch system is that it does not support dates past 19th January 2038, thereby, potentially limiting longevity for a long-term standard. It would be possible to use a larger integer value, but this would need to be carefully tested for compatibility in implementations, to ensure interoperability in future.

Another consideration is that certain programming languages do not natively support the parsing of ISO 8601, and external libraries may be required. Once a standard format is agreed, dates and times should always be stored and transmitted in an agreed format (including time-zone representation), to ensure temporal coordination and non-ambiguity. From a future-proofing perspective, it may be wise to **consider a requirement for all times to be expressed in UTC, in order to reduce complexity of integrating with assets in another time-zone in future** – which could happen in the island of Ireland’s energy market in the event that the

Republic of Ireland were to adopt any hypothetical EU changes to daylight saving time or a harmonised union-wide time zone. Similarly, ensuring clarity around timing of dispatch would be necessary in the event that international interconnectors were implemented using this dispatch API in future. Either way, **this must be appropriately tested, to ensure that, even where it is assumed that all dates/times are represented in UTC, they behave appropriately where a request is issued with a local time or alternative time zone**.

Security Requirements

At this point, while it is clear that security requirements will be important to implementation of an API such as this, without clear decisions around wider solution and system architecture, as well as a holistic threat model and security posture design, it would be remiss to set out specific security requirements at this time – the requirement at this stage is for a threat model and security posture to be agreed, around which an architecture can be designed, and an API/protocol implemented, according to that architecture and threat model.

Nonetheless, as set out above, an early version of part of D3, covering security, has been incorporated as an annex to this document.

Testing Requirements

Another significant consideration for the implementation and successful growth of a standard/common interface is the ability to reliably ensure when two parties are unable to correctly interoperate understanding where the issue lies. This can occur where two parties have both implemented their interpretation of a standard, but the interpretations differ. FSPs were aware of this phenomenon to such an extent they highlighted they would even prefer a sandbox testing instance to documentation.

This is due to the fact their developers can implement against a sandbox test environment, confirm correct operation, and then seamlessly transition over to the production environment, whereas documentation can be interpreted differently. A specification document is harder to robustly test against, especially where interpretations of wording are required in order to validate behaviours against the specification. Many technical standards use “test vectors” of pre-defined inputs and outputs to provide ways to validate behaviours at algorithm level, but for a system or interoperability level test, a reference implementation is generally required.

Therefore, to enable testing, we propose that it is a necessary **requirement for there to be, at the point of selection of a standard or development of an API/specification, a plan to create an agreed “reference implementation” of both client and server**. This is so that a developer can test the behaviour of either client/server implementation against a “known-good” reference implementation of the protocol.

One approach to delivery of a reference implementation is the creation of a “sandbox” environment – this is likely to be a second **requirement, upon network operators, to make available to the flexibility ecosystem, a non-production sandbox environment**, allowing them to test their own implementation against a near-complete replica instance of what the network operator themselves deploys.

This test sandbox system should a mirror the production system as much as possible, and ideally offer a low barrier to entry web user interface, so that users can make a step wise progression, from traditional methods of dispatch, to web based, to automated. This would grow ecosystem confidence and understandability.

From a network operator’s perspective, there will also be a **requirement for any API or standard to be sufficiently well-defined to enable exhaustive informed testing to take place**. This means that, for each value or function, the permitted and disallowed inputs and outputs should be suitable well-defined to enable a compliance test suite to be run for inputs and outputs. In addition, there will be a **requirement for a sufficiently**

well-defined functional test suite, which tests business logic flows and message sequences (i.e. to ensure an invalid dispatch message is rejected, to ensure that an implementation acknowledges the correct dispatch).

There is also a **requirement for a suitable set of security specifications that codify the behaviours assumed by the network operator, within a wider security model**. For example, this may contribute to functional testing by defining tests which ensure that unsigned, invalidly signed or repeated/replayed dispatch requests are rejected by the flexibility service provider's client.

To specify good practice for testing, this should encompass (for each of the areas identified above, i.e. for data structures, for functional invocations, and for the overall system):

- **Positive testing –** testing of an implementation under normal circumstances, to ensure that correct inputs and stimulus result in expected behaviours, and that all valid inputs are accepted.
- **Negative testing –** testing of an implementation under abnormal and boundary circumstances, to ensure that invalid inputs are appropriately rejected, and do not result in incorrect actions being triggered.
- **Fuzzing-based testing** – a kind of testing often associated with security testing, where randomly generated values are injected into fields, and valid or invalid messages are replayed out-of-sequence, in order to ensure that implementations are robust to unpredicted errors.
- **Interoperability/integration testing –** a set of testing carried out pairwise between two implementations as they would be deployed. This would be between a given client implementation, and a given network operator's implementation, in order to validate that, once an implementation passes the other tests, it also interoperates correctly at system-level with a network operator's sandbox implementation.

Integration test suites will validate that the implementation under-test correctly interoperates with simulated versions of the rest of the eco-system. This will include reliability tests to ensure the correct response occurs, even when the system is under high load or experiencing unexpected input. Integration testing is important to be carried out in representative environments, with the same security measures and protocols in place as would be in place in a deployed environment.

Prior to a large-scale deployment these test suites would have to be developed or acquired. Such a process is likely to need to take place through a standards development approach, working with network operators and flexibility service providers, in order to appropriately define such tests. To avoid "false starts" and insufficient tests being defined, this process should seek input from those with experience in standards and test development. As part of this process, there will be **a requirement for the wider design of any API/protocol/standard to be fully-defined, including any stateful behaviours and state machine design, messages, error states, and expected responses**.

As such, there is a **requirement to establish appropriate governance around test suite development, evolution, and implementation**, since implementation and definition of tests will effectively set entry criteria to the flexibility market.

It is worth noting that one of the major learnings from OpenADR 1.0, leading towards OpenADR 2.0, was the need for test tools, plans and a certification programme [A2:2].

This gives rise to the following technical requirements, which we believe would be the **minimum required in order to enable verifiable and demonstrable interoperability**, and permit for a governed process to facilitate system interconnectivity:

- A set of “positive test vectors”, provided as stimulus to the “system under test” (which could be either a DNO server, or a FSP client), to provide a set of
- Sufficient definition of the necessary outputs or exposed state of the “system under test” in order to verify the outputs or behaviours in response to the test vectors provided as stimulus for the test.
- A set of “negative test vectors” (which is likely to evolve over time), to provide a set of invalid inputs, and the expected error state and output behaviour to be triggered in the event that the “system under test” receives such a message.
- A defined “interoperability test suite”, which can be automated to ensure that different client and server implementations correctly correspond and interact, with the correct business process and high-level logic flows implemented – if any messages or API endpoints are stateful rather than idempotent, such state should be exhaustively tested here (i.e. by attempting to stop services not yet dispatched, to dispatch after stopping a given service, etc.)

Conclusions

This document has discussed some of the minimum technical requirements of a flexibility service dispatch interface. These are written with a view towards helping the ENA TWG to understand and consider design decisions around APIs, specifications, and standards. With the goal of creating an interoperable ecosystem of flexibility service providers, solution vendors, and network operators, that provides a reliable and automated flexibility dispatch system, these minimum requirements and associated discussion should help the ENA TWG to understand the necessary next steps on this project.

This document should be read alongside Deliverable 1 (Interoperability Gaps), which set out background context and stakeholder input.

Many of the minimum requirements identified in this document were architectural in nature, but would be fundamental to the design, implementation, and indeed specification of any kind of API. This reiterates the importance of having a high-level design and vision signed off by relevant stakeholders – for example, one of the minimum requirements identified is to determine the direction of an API – i.e. whether the FSP is a client or server in an API ecosystem. Decisions like this have a number of implications either way, and neither approach is obviously the right answer – there are trade-offs with each decision. This document sets out, for the minimum requirements identified, some of these considerations.

This document also reviewed the ON22-WS1A-P3 Key Service Parameters, and noted these followed a good structure and considered the context of the whole life cycle of a dispatch system. The parameters included sufficient high-level detail to enable a standalone dispatch system, however, we have recommended adding a currency field, an energy source field, sufficient unique key linking identifiers to enable the dispatch system to interoperate in the broader ecosystem. We have also suggested the possible inclusion of a greenhouse gases emissions field as another deciding factoring when sourcing generation. We have further suggested the inclusion of version numbers for the messages, and employing specific data formats and representations to remove ambiguity.

Complementing this document, we have included an advance part of Deliverable 3, exploring security requirements, given the ENA TWG’s request for as much early visibility as possible of information like this.

Finally, we recommended the inclusion of sufficient test and validation infrastructure as part of the standard, to ensure that participants can resolve any issues, expediently and ensure uptake.

References

[A2:1] ENA, ON22 WS1A-P3 Key Service Parameters, 31 May 2022, Available https://www.energynetworks.org/assets/images/Resource%20library/ON22-WS1A-P3%20Key%20Service%20Parameters%20(31%20May%202022).zip?1694991619

[A2:2] ANSI/IEEE Std 754-2019, Available: https://754r.ucbtest.org/background

[A2:3] Comparison, Available: https://floating-point-gui.de/errors/comparison/

[A2:4] OpenADR, Presentations to the OpenADR Seminar Austin, Available: https://www.openadr.org/assets/docs/openadr%20plma%20seminar_final.pdf

Appendix 3: D2 Annex on Cyber Security Considerations

In the course of carrying out stakeholder engagement, requirements solicitation and gap analysis, some cyber security or security architecture related considerations have come up, and some discussions around security have arisen in the course of the work to date.

While the current scope of work does not include a full technical security design for a multi-organisation API, this annex captures some of the initial requirements and considerations which have arisen during the course of the existing work, and discussions that have taken place to date. It also sets out some best practice principles for the design of a security architecture, and (while outside of the scope of current deliverables) some of the rationale for importance of this process.

This annex is therefore not a complete requirements specification or security model, but rather a summary of security-related outputs that have come up so far. One limitation of this is that a limited number of stakeholders had specific detailed thoughts on security, and input here was generally very high-level (i.e. recognising the need for security, or the importance of security). Input was also given in the abstract and general form by participants, and without a robust system architecture, design, and security threat model, is unlikely to provide a holistic set of security design requirements.

Security systems and architectures are best not the product of "design by committee", since they require trade-offs to be made according to design principles, which holistically must come together to deliver the desired security properties. In a multi-stakeholder environment, good governance around burden sharing and clarity of responsibility and accountability is essential. NCSC has some public cloud security guidance setting out the cloud security shared responsibility model [A1] – while this is not necessarily directly applicable here, its existence helps to explain the importance of the concept of parties sharing responsibility, rather than trying to push responsibility onto others. At architecture level, there is a need for a cohesive and comprehensive view of the security properties required, in order to ensure a robust solution is developed. Similarly, the concept of the shared responsibility model sets out the kind of governance view that may be required to ensure that FSPs and network operators are clear on security roles, responsibilities and expectations.

In addition, there is a recent move by most Western cyber security agencies to drive a move towards "Security-by-Design and -Default" [A3:2]. This is supported by UK NCSC. At this point of consideration of a dispatch and flexibility API, the principles of security by design and default should be given serious consideration from the outset, rather than applying security as an overlay to another solution.

Information Provenance

Some stakeholders expressed a view that there would be provenance requirements in the information transmitted through any dispatch API. Security properties of non-repudiation (i.e. the ability to prevent a party claiming they did not authorise/send a message) and replay resistance (i.e. the ability to prevent a recipient presenting a previous authorisation and claiming it was new) were suggested as requirements.

As an initial analysis of this, we believe this would introduce specific requirements:

- Timestamping of messages
- Relative ordering of messages
- Replay resistance (potentially achieved via reliable/trusted timestamping or message sequence numbering)
- Enduring message digital signatures (encompassing the timestamp/sequence number and contents of message)
- Transport layer security to prevent tampering with messages in transit

- Clear and robust security design to ensure unambiguous definitions of correctness, avoiding undefined or ambiguous states and values in protocol messages
- Versioned APIs and messages (potentially down to versioning of fields), in order to facilitate rejection of unsupported versions of messages
- Use of fields such as HTTP user agent to provide DSOs with visibility of security risk exposure by client software consuming the API
- A concept of criticality (or mandatory comprehension) per-field, such that an implementer of an API can evaluate, from a message, whether it has the capabilities to properly parse the message with its API version understanding
- A general principle, where possible, of idempotency in the API, to avoid duplicated requests causing changes in state, and to attempt to avoid creation of "state machines" in client/server implementation logic where possible
- Clear and robust definition of responsibilities and accountability in API documentation, to make clear which party (or parties) are responsible for carrying validating and verifying fields, and how to respond in the event of a validation logic failure.
- Robust definitions of types for every field, to prevent type-confusion attacks or comparison failures
- Use of a secure and well-reviewed serialisation format for encapsulation of API messages, in order to minimise risk of buffer overflows in parsing of API messages through having clear and well-defined field lengths, rather than relying on variable field lengths and complex serialisation formats.

A complete security architecture and design will be required, in order to develop specific technical requirements that could validate specific solutions' applicability here. We do not believe at this point that "off the shelf" protocols will deliver these capabilities, since the industry standard (HTTPS) is focused on protection of information "in-transit", rather than longer-term information provenance at-rest. While there are technical solutions and standards available for the signing of data, these signatures generally do not extend to timestamps or message sequencing.

Given the potential for a bad actor to abuse flexibility for commercial gain, by attempting to seek settlement for a false dispatch, there is likely to be a requirement for linkage of settlement requests against dispatches and contracts, to facilitate robust and unambiguous reconciliation processes.

There are likely to be learnings possible here from financial services and markets which facilitate trading (such as around transaction ordering and timestamping), as well as the balancing mechanism market.

Any kind of message signing platform or wider "identity-based" security system will introduce technical requirements around a PKI or similar registry to act as a root of trust, and authorise and distribute trusted and validated public keys, and handle revocation or disabling of compromised keys, and manage re-issuance in future.

General Security and Resilience Posture

Several discussions with stakeholders have touched on some gaps around wider security posture of a flexibility and dispatch API. The concept of thresholds for security requirements have come up, including thresholds at which traditional generation would face additional security and communications requirements, based on their connected capacity.

From a network perspective, there are likely to be 3 key areas of concern:

1. **Risk of cyber attack on DSO dispatch systems (which would sit as a point of common interconnection), originating from an API participant**.
   A DSO's dispatch system is likely to be relied upon for critical dispatch, and is also (depending on architecture) likely to present a common attack surface that is linked to multiple FSPs. This presents a potentially interesting target for an attacker, since a single point in a DSO's system may present an

opportunity to cause disruption or outages.

2. **Risk of being unable to dispatch when required, due to API outages, technical failures or communications issues**.
   This could include wider telecoms outages, as well as cloud provider or datacentre outages (making the FSP's connection to the DSO inoperable), or upstream supply chain outages. There will be times where the inability to dispatch an FSP could be detected (i.e. the FSP is no longer sending heartbeat signals), and times where this will not be detectable. As an example for the latter case, if an aggregator responds to dispatch by issuing a push message or SMS to end customers, and their SMS provider fails to deliver messages, or Firebase Cloud Messaging (FCM) had an outage, they may be unable to dispatch when required, due to wider technical issues. Similarly, if they rely on other third party APIs (such as to electric vehicle charging points), and those APIs are unavailable, they may be unable to deliver services on dispatch.

3. **Risk of misunderstanding nature or diversity of flexibility service offered, and dispatch proving ineffective for other reasons**.
   This could include unintended correlation or overlap of flexibility services, as well as a lack of visibility of the level of criticality (or connected capacity) of dispatchable flexibility services. This may not always be visible to the DSO, since an FSP could potentially have control over a significant number of geographically dispersed (or proximate) assets, and the full extent of this may not be visible to the DSO.
   There could be situations where multiple FSPs or aggregators are unwittingly re-selling the same underlying asset flexibility to the DSO, resulting in risks of non-delivery when dispatched in a period of network constraint – for example, where flexibility services are offered via an OEM, as well as an installer, as well as an operator, all relying on control of the one asset.
   There may also be downstream aggregation of risk, where an FSP is highly dependent on a single provider (such as one EV manufacturer's API), and indeed multiple different FSPs may themselves rely on that provider, or infrastructure shared by that provider.

There are clearly other risks and concerns, although these 3 have come up in discussions and appear most pertinent to DSO dispatch APIs.

Security considerations also need to be considered holistically, balanced with the potential for security requirements to also introduce costs of compliance, or barriers to entry, for FSPs and aggregators.

In general, new-entrants and smaller providers are less ready for complex and high-touch security practices or formal audit mechanisms like ISO27001 or SOC-2 etc. Smaller providers often have confidence in their technical security practices, but are unlikely to have formal certifications in place. Smaller providers generally favour moving at pace, and iterative development and change. Similarly, it is important to recognise that even the largest organisations (with significant cyber security resources and certifications) can make security-fatal errors in their implementation of critical security functions [A3:12], and therefore the presence of certifications and governance should not be replacements for demonstrable technical security measures.

There could be risks to the success of wider flexibility if the requirements for dispatch API interoperability were unduly onerous, or this may make it more difficult for DSOs to interact with smaller providers. Higher security requirements may well be justifiable (and could lead to aggregators who do adhere to higher security practices adding value into the chain), but could also reduce the diversity of flexibility services available to DSOs.

Communications Protocols

Before selecting technology or protocol options, it is important to design a communications system with an accepted and agreed lifespan. It is also critical that, before selecting or specifying requirements for communications security, a wider security architecture is developed – protocol requirements cannot be set without a wider architecture, as the overall security of the system will rely on the protocol used, as well as wider design parameters. Attempting to do one without the other will result in a piecemeal approach to security that does not deliver a secure system.

> *For example, assuming that a communications protocol was designed without a full security architecture of the overall system, high-quality industry standard techniques like HTTPS and TLS 1.3 could be adopted. Despite this, there would still be exploitable security gaps, since these only deliver transport-layer protections, and without a wider set of design around internet isolation, would likely result in DSO dispatch systems exposing a significant attack surface to the internet. In addition, basic use of HTTPS and TLS 1.3 without a specific PKI and AAA architecture would result in creating a safe public website, but not a safe private website. Similarly, without an architecture considering system isolation and logging + monitoring, it is likely that there would be significant gaps, even using a suitable protocol.*

The longer a time period that a system is designed to be secure for, the more consideration must be given to making use of the most modern equipment, as well as the non-technical drivers and levers to ensure that participants in the dispatch API are adopting updated standards and technology.

For example, for a system that needs to be secure for 5 to 10 years, consideration should be given to quantum-safe cryptography [A3:3]. This does not mean that it must be used today, but rather that it should be an expected and understood requirement by all participants that there may be a requirement to introduce it in the lifespan of currently available products, and that participants should seek appropriate contractual commitments for this from downstream suppliers. Introducing algorithm agility for cryptography in a standard can be a good first step towards planning for future changes to security measures implemented in a protocol, although as was encountered in the lifespan of TLS on the internet, it is important to also ensure that older implementations can appropriately cope with seeing optional advertised support for more modern security features they are unaware of. Conversely, it is also critical to ensure that downgrade attacks cannot be used to lower the level of effective security through use of algorithm agility to negotiate back to older versions – a versioned API, where higher versions use newer cryptography, and older versions are deprecated and removed, is one approach to resolving this in a more straightforward manner.

Standard approaches such as PKI (public key infrastructure) [A3:4] are likely to provide effective and viable methods for identity management, but these will require robust governance and security measures around creation of certificates and the business processes around issuance and revocation of certificates, to ensure only authorised organisations and individuals at those organisations can request certificates.

Security design patterns for the whole system surrounding communication protocols should be considered, including avoiding any “single line of contact” to the internet, and other pieces of best practice guidance from NCSC. In modelling and understanding risks, the NCSC Criticalities Process [A3:5] may prove helpful as a methodology.

There are also a number of potentially relevant design patterns from NCSC which may prove useful in architecting and modelling suitable designs for import of data [A3:6], export of data [A3:7], alongside more general guidance around common security anti-patterns [A3:8] of bad practice that are commonly seen.

Given the inherent links between a flexibility/dispatch API, and the UK’s critical national infrastructure, it is likely that the infrastructure around such an API would be an attractive and visible target to nation-state attackers, and that they would attempt to gain a foot-hold in it for future exploitation, or exertion of influence. There are also potential economic benefits to market participants, to the detriment of others, were they able to exploit functionality of such an API in a difficult-to-detect manner. The likelihood and incentives/motivations for attack would make such a system at high risk of attack.

NIS Regulation Implications

Designation of Operator of Essential Service Status

At present, flexibility services would not be covered by the current criteria and thresholds as set out by the Secretary of State for BEIS (since replaced by DESNZ) and Ofgem as NIS joint competent authorities for the energy sector, since flexibility services would generally not fall under the criteria unless they were also a large generation site or similar.

NCSC is a national technical authority in cyber-security, and does not hold specific regulatory responsibilities, or the ability to endorse any OES' specific activities.

NIS Regulation Obligations

The NIS Regulations require operators of essential services (if designated as an OES) to take appropriate and proportionate technical and organisation measures to manage the risks posed to the security of the network and information systems on which their essential service relies, per regulation 10(1) of NIS; and to take appropriate and proportionate measures to prevent and minimise the impact of incidents affecting the security of the network and information systems used for the provision of an essential service, with a view to ensuring the continuity of those services.

This means from a NIS perspective that DSO covered by NIS should consider their NIS compliance when incorporating flexibility into their network, and in particular consider the security of the network and information system that their essential service relies on, which is likely to include their flexibility service interface, as well as any hosted provider or aggregator whose systems they rely on to deliver an essential service.

Given the threshold for electricity distribution covering a loss of supply incident to 50,000 customers for more than 3 minutes, DSOs would need to evaluate their current and potential future use and reliance on FSPs, and the extent to which they may be exposed to incidents exceeding NIS reportable thresholds in the event of a flexibility failure, or a need to take emergency actions which could cause a similar impact.

NIS Digital Service Providers

There is an additional route through which a flexibility platform could be deemed a NIS-covered service, regulated by the ICO as competent authority, were it to be deemed to be an online marketplace.

Per ICO guidance [A9], an online marketplace provided to external customers (individuals or organisations) by an entity that is not a small or micro-business is a "Relevant Digital Service Provider" (RDSP) under NIS.

There is a requirement for RDSPs to register proactively with the ICO.

Online marketplaces are considered to be:

> *digital services that allow individuals or traders to conclude sales or service contracts with traders, either on their own website or by means of providing services to traders' websites. Online retailers that sell directly to individuals on their own behalf are not covered.*

It is likely that one or more aspects of the properties of a flexibility marketplace or platform could be construed to be an RDSP, on the grounds that a digital service (i.e. the flexibility marketplace, platform, or system) allows traders (FSPs and aggregators) to conclude sales or service contracts (i.e. flexibility contracts) with traders (DSOs and flexibility service buyers), on their own website (i.e. the flexibility platform).

RDSPs are covered by regular NIS regulations (Part 4, and Regulation 12) [A3:10] as well as a separate "DSP Regulation" [A3:11]. ICO is clear in their guidance that when measures are implemented to manage risks, implementers are allowed to consider the state of the art when evaluating available measures, however they are not allowed to consider costs of implementation in this evaluation. This means that **security measures implemented for NIS compliance must be taken without regard to costs of implementation**.

On this basis, it is highly likely that a flexibility marketplace or platform would be considered to be an RDSP, and therefore subject to NIS regulations as a digital marketplace under the ICO's supervision as competent authority. This would require the digital platform to be secured appropriately to NIS standards.

References

[A3:1] https://www.ncsc.gov.uk/collection/cloud/understanding-cloud-services/cloud-security-shared-responsibility-model

[A3:2] https://www.cisa.gov/resources-tools/resources/secure-by-design-and-default

[A3:3] https://www.ncsc.gov.uk/whitepaper/preparing-for-quantum-safe-cryptography

[A3:4] https://www.ncsc.gov.uk/collection/in-house-public-key-infrastructure/pki-principles

[A3:5] https://www.ncsc.gov.uk/files/Criticalities-and-CNI-Knowledge-Base-Industry-Flyer.pdf

[A3:6] https://www.ncsc.gov.uk/guidance/pattern-safely-importing-data

[A3:7] https://www.ncsc.gov.uk/guidance/design-pattern-safely-exporting-data

[A3:8] https://www.ncsc.gov.uk/whitepaper/security-architecture-anti-patterns

[A3:9] https://ico.org.uk/for-organisations/the-guide-to-nis/digital-service-providers/

[A3:10] https://ico.org.uk/for-organisations/the-guide-to-nis/security-requirements/

[A3:11] https://www.legislation.gov.uk/eur/2018/151/contents

[A3:12] https://msrc.microsoft.com/blog/2023/09/results-of-major-technical-investigations-for-storm-0558-key-acquisition/

Visit our website to find out more about Open Networks

ena
energynetworks
association

Energy Networks Association
4 More London Riverside
London SE1 2AU
t. +44 (0)20 7706 5100

w. energynetworks.org
@EnergyNetworks

Energy Networks Association Limited is a company registered in England & Wales No. 04832301
Registered office: 4 More London Riverside, London, SE1 2AU

The voice of the networks