POLICY ISSUES PAPER – CONTROL SHEET

Title of Paper	***Delivery Strategy – Systems Integration Strategy***		
DA Issue Ref		Date:	6 October 2016
Issue Owner (Accountable)	*James Crump*		
Author of Paper (Responsible)	*David Liversidge*		
Status of Paper	4 – Draft for EDAG Review 5 – Final Recommendation to DA		
Timing	*Planned review cycle for Blueprint Phase Product:* *User Group – 27 September 2016 (for review);* *EDAG – 13 October 2016 (for review);* *Programme Board – 26 October 2016 (for information)* *Design Authority – 27 October 2016 (for approval).*		
Dependencies	*Inbound dependency on the development of solution architecture shortlist*		

Circulation	Workstream Leaders / Design Team / User Group / EDAG /DA Huddle / Website

Issue	*Transition from current switching arrangements to new switching arrangements.*		
Impacts Domestic?	Yes	Impacts Non-Dom?	Yes
Policy Objective (and reference to ToM v2)			
Previous Positions on this/related Issues	*N/A*		
Summary of Recommendations			

Internal and External Engagement	
Business Process Design	
Regulatory Design	
Delivery Strategy	Author
Commercial Strategy	
DIAT	
Legal	
Other Ofgem Teams	

Meetings at which this paper has been discussed	
Workstream Leaders	
User Group	
EDAG	To be discussed at 13 October meeting.
Ofgem Design Authority	To be discussed at 27 October meeting.

POLICY ISSUES PAPER – CONTENT

Purpose and approach to this Systems Integration Strategy

Purpose of this Systems Integration Strategy

1. This product describes the System Integration strategy for the new Switching Arrangements which will enable gas and electricity consumers achieve faster, more reliable switching. This product has been produced to comply with the Product Description issued by Ofgem (Appendix 1) in August 2016.

2. The main objectives for this system integration strategy within a Switching Programme context are:

   - Defining the purpose, aim, objectives, scope, risks and dependencies relevant to system integration
   - Defining system integration activities in a wider delivery context and key relationships between these and other programme activities, with a particular focus on Design Management and Testing
   - Identifying key roles for system integration during Design, Build and Test (DBT) and any preparation required ahead of this phase, including relationships with other programme roles such as Design Authority and Test Execution and Management
   - Assessing options for responsible bodies that could fulfil these roles within the broader governance and assurance arrangements
   - Defining appropriate approaches to system integration, taking into account best practice whilst recognising the remaining areas of uncertainty due to ongoing definition of the programme;
   - Defining the key system integration documentation and deliverables that would be expected as the programme progresses through its planned phases

3. The integration of implemented system or solution elements is generally performed according to a predefined strategy. The definition of the integration strategy is dependent on the architecture of the system and relies on the way this has been designed. The strategy is then enacted through an integration plan that defines:

   - The minimum configuration/readiness of expected system components or sub-systems;
   - Their order of assembly and aggregation in order to support efficient subsequent testing and other verification validation actions (e.g. trials);
   - The approaches to be adopted to check, evaluate and de-risk interfaces, including the provision of the necessary capabilities in the integration environment to support this process.

4. The process of creating a system integration strategy usually starts with the selected testing strategy and needs to be closely aligned to it. The system integration strategy sets the overall approach to system integration, the main activities and associated roles and responsibilities. It is vital that a system integration strategy is established early in the programme to ensure that an appropriate level of system integration activity is planned for throughout the programme. This planning should be consistent with the assessed integration risks and the options for undertaking system integration should then be examined and assessed; including a 'do nothing' option.

Approach

5. We have taken the following steps to develop this system integration strategy:

   i. Due Diligence investigation of applicable best practice from Software and Systems Engineering, IT Service Management and Project and Programme Management together with specific lessons learned from relevant recent programmes such as SMIP and Nexus;

   ii. Tailoring of the best practice and lessons learned to the particular circumstances and predicted risks applicable to implementation of the new End to End Switching arrangements;

   iii. Iterative development and evaluation of the system integration strategy in line with the programme Target Operating Model ensuring coherence with related work packages as they develop (e.g. Solution Architecture, Governance & Assurance and Testing Strategy); and

   iv. Discussion with stakeholders and subject matter experts from DCC, Ofgem and the wider energy retail industry, including formal review through the governance structure for the Blueprint phase[1].

Timeline and Dependencies

6. This system integration strategy forms part of the Ofgem Delivery Strategy workstream within the Blueprint phase of the Switching programme. It will be subject to a Request for Information (RfI) as part of Design Baseline 1 (DB1).

7. Following the RfI, the programme will develop detailed design specifications for the chosen solution architecture and its operational requirements, and further develop commercial, regulatory and delivery proposals as part of the Detailed Level Specification (DLS) phase. Following the DLS phase, regulatory changes will be enacted and the Data Communications Company (DCC) will procure a provider(s) of a Central Registration Service (CRS) for the specified solution.

8. The Blueprint phase contains other workstreams which have interdependencies with the design of a system integration strategy. Without proper understanding and management, these dependencies represent a risk to the effectiveness of the proposed

[1] Design Team, User Group, EDAG and Design Authority

system integration strategy. This is an ongoing process and this strategy will require further iteration to both reflect and inform these interdependent areas as the programme progresses towards delivery (see Next Steps section). Inbound dependencies will have a direct effect on the design of a system integration strategy, and in turn the output arising from the system integration strategy will affect a number of other programme areas.

9. The key dependencies within the programme workstreams are captured in Table 1 below.

Work stream/ package	**Type**	**What is affected**	**Impact and how it will be addressed**
BP Design – solution architecture, data and business process models	In	The final solution architecture and its detailed specification will dictate how the overall system will be decomposed into sub-systems and components for implementation across the various parties involved	The final solution architecture will dictate the: • number of sub-systems • scale of change in constituent legacy systems • complexity of interfaces (internal and external) • scale of E2E ecosystem and therefore number of E2E processes affecting multiple parties to integrate • risks and therefore level of assurance/testing required The final solution design will dictate that nature and complexity of the interfaces between system components and market participants and hence the risks to be managed system integration. This strategy has been developed based on the range of solution options currently being explored (the 'short list') and assumes that a 'new capability' option will be chosen as opposed to the 'do nothing' or 'do minimum' options. If a do nothing or minimum change option is selected, then this will radically alter the basis of the programme and this strategy will need to be revisited.
Delivery/ Transition	In	The number of parties needing to be ready at 'go live'	A 'big bang' would mean 'all' users would need to be ready at the planned go live. Transition based on sub-groups (e.g. fuel type) would mean that not all users would need to be ready at the same time. With the large number of parties involved and the inherently higher risks involved, the integration effort (peak resource) through DBT will be more significant for a 'big bang' release to ensure readiness, although the duration of the effort is likely to be shorter

Delivery/ Transition	In	The number of service 'releases' that need to be integrated and tested	If transition has multiple functional releases, each of these would go through a full integration and test cycle with regression back to previous release. This does not affect the design of the system integration strategy but will affect the cost and time and risk profile of system integration which needs to be balanced against the reduced risk to the programme timeline inherent in a single release approach
Delivery/ Testing	In/Out	The Testing risks and likelihood of issues and defects will be reduced by an effective system integration strategy/the strategy for testing will influence and mirror the integration approach	System Integration is inextricably linked to Testing. It complements testing by more proactively managing the design and build activity across all parties and identifying and resolving issues before the formal testing phases. It also aims to ensure improved co-ordination of design, build and testing across all parties, improving readiness. A well-defined and executed system integration strategy will significantly de-risk testing and hence reduce timescales and costs. System integration and testing together need to cost-effectively identify and mitigate issues and defects prior to release noting a low risk appetite for the new arrangements
Delivery/ Testing	In/Out	The system integration roles and responsibilities will affect the roles and responsibilities for testing and vice versa	Testing Roles and responsibilities are discussed and outlined in the testing strategy. These roles and responsibilities assess options for who is best placed to undertake (manage and execute) the cross-party test phases. These options include a specialist integrator as recommended via this system integration strategy. Final option for SI roles and responsibilities decided in this strategy needs to align with the final options decided in the testing strategy.
Delivery/ Testing	In/Out	The use of optional/ informal testing phases (pre-SIT and pre-UIT) is inextricably linked to the system integration strategy.	As recommended in the testing strategy, discretionary/ optional test phases should be used to de-risk the main (formal/non-discretionary) testing. As per the 'Agile principles' discussed in section 2, these optional test phases are just one form of de-risking technique for the DBT phase which can include other approaches; e.g. prototyping, design walk-throughs, etc. This range of design de-risk techniques needs to be considered holistically in this SI strategy aligned to the integration and design risks which will not be fully understood until the solution architecture and detailed specification is complete.

Delivery/ Governance & Assurance	In/Out	System integration needs to form a coherent part of the overall Design Management arrangements within the Governance & Assurance regime	Issue, Defect, Change and Configuration Management processes and associated governance need to be effective to deal with the volume of change expected and the multi-party environment. 'Best for the programme' decisions need to be made by an empowered body informed and supported by the system integration role. Management of configurations across design baselines, integration and test environments and procedures, etc. needs to be well controlled.
Delivery/ Governance & Assurance	In/ Out	System Integration Roles and Responsibilities (in)/ Results and assurance findings (out)	Integration, together with Testing, provides assurance as the programme progresses through DBT. However, the roles for system integration need to be considered alongside other programme roles to ensure they are aligned and coherent, and responsibilities for system integration should be clear and appropriate within the wider governance arrangements
Delivery/ Governance & Assurance	In	The arrangements for providing assurance during DBT will affect SI	Integration, as for other functions, will need to be subject to the agreed programme Assurance regime, which may involve a mix of self-assurance and independent assurance methods.
Regulatory Design	Out	Code modifications and licence changes arising from the Switching Programme	Regulatory architecture for the switching programme must provide appropriate incentives to ensure that parties prepare for and contribute to system integration activities on a timely basis as defined in this strategy and the SI Management Plan. Furthermore, the SI strategy and plan will identify specific SI l roles and responsibilities that will need to be reflected in Codes/Licences. For example, if DCC is to act as SI (or procure SI), this will have to be supported in its licence.
Commercial	Out	Procurement of the CRS and related services	Procurement and cost decisions relating to the CRS and other related services will need to consider the requirement for system integration. If the option for an independent specialist system integrator is chosen, then this service will need to be separately procured via DCC or another route.

[Table 1 – System Integration Dependencies within the Switching Programme]

Background to System Integration

10. Many of the requirements of the new End to End (E2E) switching arrangements will only be satisfied through the synergistic interaction of the component parts of that system. To implement a complex system of this nature, the whole solution or system has to be broken down into sub-systems, modules and components (e.g. hardware, software and firmware). These are then designed, built, integrated and tested in controlled circumstances, progressively building the system from the components/modules into sub-system elements and finally the overall system. The new switching system has to support new/updated business functions and processes, so it also has to be tested with the intended business processes and any interactions with wider services and systems applicable to the operating environment. If multiple releases are planned, then this process may be repeated iteratively for each release requiring careful configuration control of design baselines, and associated design, build and test environments across all parties involved.

11. The process of translating the business requirements required by the users or sponsors of the Switching system, through a process of logical and physical architecture definition into detailed technical specifications for each system component/module and the interfaces between these, is rarely if ever perfect and ambiguities and inaccuracies will be inherent in this process. If individual parties, as providers of parts of the whole system, are then simply left to their own devices to design, build and test their parts of the system, there is a high likelihood that when they integrate and interface these together issues will be found; e.g. due to different interpretations of ambiguous interface specifications.

12. If left unchecked and undetected until an advanced point in the programme implementation phase (i.e. the formal integration and interface testing phases), it becomes expensive and time consuming to resolve these issues through design changes (e.g. the interface specifications) as this will require potentially some or all parties having to re-work (design, build and test) their component parts before repeating the formal integration and interface testing phase where the issue was discovered.

13. Successful achievement of the outcomes of the Switching Programme will be dependent on all parties delivering their part of the new arrangements together with effective integration of the component parts to achieve the overall system and service level requirements to time, cost and quality. Implementation of the new E2E switching arrangements will be challenging and there are a number of key programme risks that can only be mitigated through an effective system integration approach.

14. Effective system integration can significantly reduce the number and severity of issues and defects that arise late during formal testing or in early life that should have been identified and mitigated earlier when cost and time for rectification is much less. For example, proactive sharing of detailed design interpretations of interface specifications across parties and prototyping of high risk interfaces will enable rapid analysis and rectification of issues during design. In view of this, an effective system integration strategy needs to be developed for the programme which should be aligned with the testing strategy.

Purpose, Aim and Objectives of System Integration

15. System integration consists of a process that "iteratively combines implemented system elements to form complete or partial system configurations in order to build a product or service. It is used recursively for successive levels of the system hierarchy." (ISO/IEC 15288 2015, 68). The system integration process can be extended to any kind of product-based system or service-based system.

16. The purpose of system integration is to plan for and then manage and co-ordinate the activities and resources required to ensure that the component parts of the whole switching system are designed, built assembled, tested and accepted in controlled and progressive way that ensures that the requirements at the whole system/whole solution level are satisfied and that any cross-interface and cross-party design issues and defects are identified and resolved as early as possible.

17. The aim of system integration is to prepare the whole system (i.e. the new E2E switching arrangements as defined below) for final validation and transition into live operations. System integration consists of progressively assembling combinations or 'aggregates' of implemented elements that compose the whole system as architected during design, and to check correctness of static and dynamic aspects of the interfaces between the implemented elements.

18. The objectives of system integration for Switching can be summarised as below:

    - Ensuring that design, build and test activity is effectively managed and controlled across all parties to understand and then mitigate the integration risks in a timely and cost-effective way
    - Progressively assembling the implemented elements (components/modules and their interfaces) to make sure that they are compatible with each other.
    - Demonstrating that the aggregation of implemented elements perform the expected functions and meet measures of performance/effectiveness.
    - Detecting issues and defects related to design and integration activities by submitting the aggregates to informal and formal testing and validation activities

19. Like all aspects of delivery, this strategy has given consideration to the financial and time cost of performing the system integration function compared to the time, cost and performance benefits it can bring to the programme.

Scope of System Integration

20. The activity/work scope of system integration tailored to the specific risks and features of the Switching programme is discussed in the next section. The physical scope of system integration for Switching will in turn depend on the scope of the solution to be implemented for the new End to End (E2E) switching arrangements. This E2E solution defines the 'system boundary' for the purposes of integration and will include the key internal and external interfaces within and between the component parts of the system being developed by the various parties involved. The scope of this 'whole system' will be

determined by the output of the Blueprint and Detailed Level Specification phases which will specify:

- The scope of end-to-end business process model and the chosen solution (logical and physical) architecture, including system components, their functionality and their hierarchy (both CRS and other Industry parties' systems and processes)
- The internal interfaces between the system components and any external interfaces with relevant wider energy retail market systems and processes (e.g. balancing and settlement, network charging, etc.)[2]
- Functional requirements (e.g. messaging formats, protocols)
- Non-functional requirements (e.g. Availability, Reliability)
- Service management (operational) requirements (e.g. Incident Management, Help Desks)

Best Practice and Lessons Learned Applicable to Systems Integration

21. As mentioned in the Approach section above, due diligence of applicable best practice and lessons learned was undertaken. This is contained at Appendix 2 and has been used to develop this strategy. The applicable areas of best practice and guidance examined are summarised in Table 2 below.

		Domain/Discipline				
		Systems Engineering	Software Engineering	IT Service Management	Project Management	Programme Management
Best Practice	Lead professional bodies	INCOSE	BCS (Chartered Institute for IT) Axelos (formerly OGC)		APM & PMI, Axelos, DSDM (for Agile DSDM)	
Best Practice	Best Practice Guidance / Frameworks	INCOSE Body of Knowledge		ITIL (+ SIAM)	Prince 2 & APM BoK	MSP APM Body of Knowledge
			Agile			
Best Practice	Aspects applicable to Testing and System Integration	Integration, Verification & Validation		ITIL Service Transition: Validation & Testing	Solution and Scope Mgmt (APM) Quality (Prince 2)	Assurance Quality
Standards	Higher level/ umbrella standards	ISO15288, ISO9000 and ISO9001. IEEE730		ISO20000	None?	Management of Portfolios
Standards	Life Cycle Delivery Models	ISO12207 (System & Software Lifecycle Processes) – 'V model'		Service Life Cycle	Waterfall, Agile, V model, Prince 2 Processes	Transformational Flow
Standards	Testing and Integration Specific Standards	IEEE1012 & ISO15026 (pt2 – Assurance Case)	ISO29119, IEEE829, IEEE1008, IEEE1028	None	None	None

[2] This will be determined by the output from the Business Process Design workstream which has identified a short list of solution architectures to deliver the new switching arrangements and will then go through a process of final selection once information has been received from Industry via the RfI process

[Table 2 – Sources of Applicable Best Practice and Guidance Relevant to System Integration of Switching]

22. In addition to examination of best practice and standards, which is itself drawn from multiple lessons learned across many programmes, a number of highly relevant recent projects were examined for Lessons Learned, including the on-going Smart Metering Implementation Programme (SMIP) and project Nexus. The key areas of best practice, standards and lessons learned relevant to this system integration strategy are summarised below with further detail at Appendix 2.

23. Best Practice and Standards. System and Software Engineering include within its lifecycle processes a function called Integration. The INCOSE System Engineering Body of Knowledge expands on this to define some of the main activities, approaches, techniques and outputs applicable to system integration. These are reflected in Sections 3 and 5 below, tailored to the Switching programme context.

24. The International Standard ISO15288 (Part 1, 2015) defines Integration as:

> *The purpose of the Integration process is to synthesize a set of system elements into a realized system (product or service) that satisfies system requirements, architecture, and design. This process assembles the implemented system elements. Interfaces are identified and activated to enable interoperation of the system elements as intended. This process integrates the enabling systems with the system-of-interest to facilitate interoperation.*
>
> *NOTE 1 For a given level of the system hierarchy, this process iteratively combines implemented system elements to form complete or partial system configurations in order to build a product or service. It is used recursively for successive levels of the system hierarchy.*
>
> *NOTE 2 The interfaces are defined by the Architecture Definition and Design Definition processes. This process coordinates with these other processes and checks to make sure the interface definitions are adequate and that they take into account the integration needs.*

25. Service Integration and Management (SIAM) represents another applicable area of best practice. SIAM is a rapidly developing area of service management, and one that is closely associated with multiple disciplines including IT service management, enterprise architecture, organisational change management, quality management and risk management. ITIL processes and capabilities provide a strong foundation for implementing SIAM because they cover the lifecycle of IT services, and the terminology is recognised and understood by most IT suppliers. ITIL, therefore, provides a strong platform and common language with which a SIAM team can manage and work with a full spectrum of suppliers.

26. SIAM draws on other sources of best practice as well in specific domain areas, for instance COBIT5 and ISO/IEC 38500. Whilst ITIL provides high level guidance on many aspects of IT management relevant to SIAM it does not currently do so in the context of

a multi-supplier eco-system. Axelos, the current owners of ITIL have announced that during 2015 they will be publishing white papers on SIAM.

27. However, Service Integration should not be confused with System Integration; the former focuses more on how the various suppliers of the service components are managed and brought together and managed whist the latter focuses more on how the physical components of the system/solution are brought together, verified (tested) and validated. **In the multi-party delivery environment applicable to switching, where the end-to-end switching service will only be realised through the effective integration of the CRS plus changes to other parties systems and implementation of new/modified business processes, then aspects of both Service and System Integration are applicable to successful realisation of the Switching programme.** This is discussed further in Section 3.

28. **Lessons Learned (SMIP and Nexus).** SMIP does have a system integrator function and role which is undertaken by the Data Services Provider (DSP) under DCC assurance. However, this role was not adequately defined and resourced in the DSP contract and potentially conflicted with the primary delivery roles of the DSP (i.e. it was not independent and separate). This SI role only extended to the DCC designed and delivered services and not the 'whole solution'; i.e. it did not extend to the associated design activity of the DCC user base and hence 'both sides' of the interfaces with the DCC delivered services. As a result, a number of interface and integration issues emerged during Systems Integration Testing (SIT).

29. The difficulties experienced in SIT highlighted that the service provided by the Systems Integrator did not meet expectations and was not adequately scaled to handle the size and complexity of the integration challenge. DCC and the DSP responded quickly and deployed additional unplanned resources to overcome the challenges, conducted a full external review of SI activity and initiated a project to implement the recommendations of the review3.

30. Nexus did not have a dedicated systems integration role and many of the current issues can be traced to a lack of proactive planning and management of cross-party integration and testing and associated risks with a lack of clear responsibility and accountability for these roles agreed at the outset. This resulted in many issues emerging late in testing (market trial) with many parties not being ready and hence designs not being stable. The recent assurance review undertaken by PwC states: "The issues that have contributed to this [lack of progress in testing] position have included **'blocking' defects that have halted, and in some cases, continue to halt the efficient execution of end-to-end test scenarios [and] the complex coordination of testing across participants for certain scenarios such as Change of Ownership ('CoO').**

[3] 'DCC Annual Service Report: Performance Year 2015/16' available https://www.smartdcc.co.uk/media/400423/asr_py2015-16_-_submitted_-_published.pdf

31. **Agile.** Although not specifically a best practice approach for systems integration, a number of Agile principles and techniques align closely with the purpose and objectives of system integration; that is to plan and organise the DBT activity in such a way as to understand and then mitigate risks at whole system/whole solution level, with a focus on delivering the end customer requirements and priorities whilst embracing and managing change effectively. As recommended by the most recent Government report into public IT programmes4, there are a number of underpinning Agile principles and techniques that can be applied to the Switching Programme. Specifically:

Collaboration and Team Working. For example, cross-party sharing of design information as designs progress, particularly at the interface points, and joint resolution of emerging issues (similar to the use of SMIP Design Forums);

Prototyping/incremental development. For example, early drops of build and test information ahead of main test phases to de-risk (e.g. SMIP GBCS Interface Testing for Industry);

Change Management. An effective means of managing issues, change and configuration during DBT that spans party boundaries with incentives to respond and resolve issues quickly**;**

Iterative release of capability into live environment. This will be considered by Transition. However, even if a 'Big Bang' release is opted for in terms of participants, iterative release of functionality based on prioritisation of requirements aligned to customer benefits should be considered to progressively move towards next day switching.

Key Risks Applicable to System Integration

32. As outlined above, this system integration strategy has been tailored from applicable best practice and lessons learned to suit the particular risks for the programme that can be mitigated by an effective testing regime. The key risks identified are summarised in Table 3 below together with the proposed mitigations where effective testing forms part of those mitigations.

Programme/ System Level Risk	**Mitigation Approaches**
Imperfect definition of component and interface specifications during Blueprint and DLS phases	Effective sharing and collaboration across design teams throughout DBT A central body to continually clarify, mediate and arbitrate on design issues Need to prototype and de-risk designs and interfaces as early as possible

[4] 'System Error – Fixing the Flaws in Government IT", Institute for Government 2012

Complex, multi-party environment leading to federated, dispersed design, build and test of components and sub-systems controlled through a variety of regulatory instruments with no central visibility and oversight	Need for centralised planning and co-ordination of all DBT activity to ensure continued alignment Need to align and help ensure readiness of all parties Need central body to develop and implement cross-party design, build and test infrastructure and control its configuration Need for clear, centralised issue, change and configuration management processes
System partitioning based on organisational rather than 'ideal' boundaries and hence complex interfaces between parties	Need for continual review of interface specifications and their practical design interpretation across multiple parties Need to develop means for early de-risking of design and testing of interfaces
Delivery in an environment that is undergoing significant change	Need proactive and effective change management dealing with internal and external change
The need to control time, cost and quality at the whole system/whole programme level	Need for clear and responsive decision making based on 'what's best for system and programme' Effective Issue, Change and Configuration control with ability to trade-off within time, cost, quality at whole programme/whole system level

[Table 3 – Programme Risks Applicable to System Integration]

Analysis: System Integration Activities in a Switching Programme Context

System Decomposition

33. A complex solution or system such as that being designed for Switching will need to be broken down into a number of sub-system elements, modules and components to be designed, built and tested individually before being brought together progressively, integrated and tested into higher level 'aggregates' building up to the full (end to end) solution or system. There is 'good practice' design guidance that describes how to decompose a system adopting the following principles: (ref?)

    - Reduction in the number, type and complexity of the physical interfaces
    - Testability (i.e. system modules/components that can be tested separately)
    - Compatibility of technologies
    - Ease/practicality of implementation (design, build and test) of components
    - Optimisation of resources shared between sub-system elements
    - Modularity (i.e. elements have low interdependence)
    - Openness (i.e. physical interfaces are non-proprietary and can be openly published to all)
    - Resilience (i.e. components support reliability, maintainability and ease of update/upgrade)

34. However, there are constraints specific to the Switching Programme that will limit the ability to decompose the system in a way which fully embraces this best practice. For example, the new solution design that will underpin the new E2E switching arrangements cannot be delivered by a single organisation or provider, so the process of decomposing the whole system into lower level sub-system elements, components and modules cannot solely be done on a 'risk' basis. That is, it cannot always be divided along boundary points, or interfaces that reduce the complexity and number of interfaces and allow for future flexibility and ease of upgrade.

35. Also, with all the new switching solution design options being considered, there will be some existing legacy components, modules or sub-system elements that will need to be used or modified (e.g. DES and ECOES) or replaced by new services being developed as well as external processes and systems to interface with as part of the wider market arrangements. These may prevent or constrain selection of appropriate and compatible technologies and the use standard, proven and/or open interfaces for all aspects of the design.

36. So, even if the best practice in system decomposition is adopted, it is likely that the the design of the physical solution architecture and the detailed specification process for Switching will result in the whole solution or system required to support the new end to

end switching process being distributed across a number of parties and comprising a mixture of modified legacy and new components. This will result in a range of sub-optimal and complex interfaces that will need to be effectively implemented to deliver the specified system level requirements.

System Integration

37. The system integration process is used to systematically assemble the higher-level system from lower-level ones (modules or components) that have been implemented (designed, built and individually tested). Depending on the design risk, System integration often begins with analysis and simulations (e.g., various types of prototypes) and progresses through increasingly more realistic systems and system components until the final product or service is achieved.

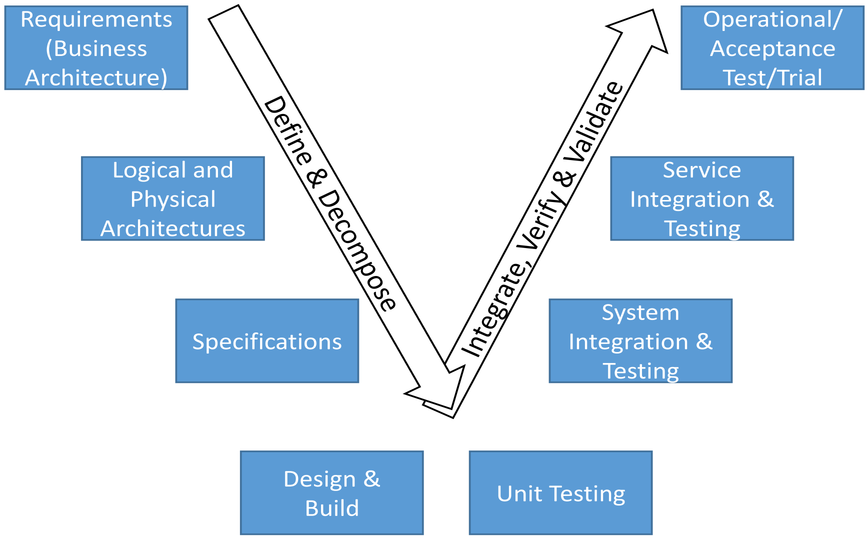
38. System integration is based on the notion of an **aggregate** - a subset of the whole system made up of several **elements** (implemented system **components/modules** and their **physical interfaces**) on which a set of Verification and Validation (V&V) actions is applied (e.g. inspections, walk-throughs, testing, etc.). Each aggregate is characterized by a **configuration** which specifies the implemented elements to be physically assembled and their **configuration status**.

39. To perform V&V actions, a V&V configuration that includes the aggregate plus V&V tools and environments is required. The V&V tools are classed as **enabling products** and can be simulators/emulators (simulating implemented elements), stubs, etc.

System Integration and the 'V' Model

40. According to the V Model, which is being adopted for Switching implementation, system definition (left hand side of the V) is done by successive levels of **decomposition**; each level corresponds to the physical architecture of systems and system elements. The **implementation** of the system (right hand side of V) takes the opposite approach of **composition** (i.e., a level by level approach). At each level, integration is done on the basis of the physical architecture defined during system definition.

41. Integration can therefore be positioned within the right hand side of the V model, aligned with the appropriate testing strategy. See Figure 1 below:



[Figure 1 - System Integration and the V Model]

42. In the context of Switching, the integration activity is required to join together, physically link, the implemented components to form sub-system elements. Each implemented component is first individually tested (via pre-Integration Testing) prior to entering integration. Integration then adds the appropriate level of cross-party/cross-interface testing activity to the assembly activity, leading up to full end-to-end testing and final validation. The final validation performs any agreed operational acceptance tests that are required to authorise transition for use of the system in the live environment.

Service Integration

43. Sourcing from multiple IT suppliers allows an organisation to maintain in-house technical teams or large single source suppliers, and become more adaptable by taking advantage of competitive marketplace behaviours which incentivise cost reduction and leverage innovation. The use of multiple best of breed suppliers can incur large management overhead costs and lead to difficulty in integrating and managing end to end (e2e) services. Service Integration and Management (SIAM) thinking, as summarised above, has developed to aid that management challenge.

44. The term SIAM was predated by the term Service Integration which has been in use since at least 2009. It should not be confused with the term System integration. Unlike ITIL (for IT service management) and PRINCE2 (for project management) mentioned in Appendix 2, it is not a best practice framework and does not yet have an established body of knowledge. Instead it is largely implemented in line with proprietary models developed by large IT service providers to meet requirements developed by third party advisors. In UK government it is seen as a way for large governmental IT organisations to better manage and control multi-sourced operations, by compiling (and then sharing between themselves) their best practices and their most successful management methods. SIAM services are defined in the Government Service Design Manual with

multiple providers offering SIAM services through the G-Cloud. This defines the key features of an effective SIAM model as follows:

- being able to define different service requirements for critical and non-critical services (for example, some commodity services may require online service support or service desk only, whereas mission critical IT systems will require a more integrated service model)
- a performance regime that ensures organisations don't pay for services they can't or don't use
- explicit **service integration arrangements that focus on service performance, usability and availability from a user perspective**, not just from a supplier's commercial perspective
- skills and capabilities that support transitioning to, and managing services in a new commodity-based environment
- **a focus on open standards and interoperability** to support workflow, performance management and service management, billing and payment

45. Within a switching programme context, the end-to-end switching solution will be realised through multiple providers: some of these directly procured by DCC to develop and deliver the CRS; other parties providing legacy system and services (e.g. DES and ECOES – modified or unmodified) and other Industry parties (suppliers, network operators and gas transporters) modifying their systems to implement their part of the end to end switching solution and hence provide a fast, reliable switching sere vice to consumers.

46. Given this context, elements of service integration are applicable to Switching in addition to systems integration from a physical hardware/software solution perspective.

System Integration Relationship with Other Delivery Activities

47. The diagram at Appendix 3 provides an overview of the key activities required in the DBT phase of Switching and positions system integration activity to illustrate where it sits in relation to the closely related activities of testing and design management/control as well as some of the wider DBT activities. These key relationships are discussed briefly below.

48. **Testing**. As discussed above, system integration follows and aligns with the testing strategy. However, it can also be seen from the discussion above that it goes beyond just testing and includes the proactive management and de-risking of the bringing together of the system elements (i.e. the system components/modules and their interfaces) prior to their formal testing through the defined test phases. System integration and testing are inextricably linked and both the respective strategies and plans should be aligned and complimentary in how they identify, plan for and mitigate design risks and ultimately allow a fit for purpose system to be delivered with traceability back to the original user requirements.

49. Given this, it is recommended that the system integration and testing strategies should be managed together as linked strategies in later iterations, as well as the respective management plans when these are produced in the DLS phase. It is also recommended that the responsible body selected to undertake system integration roles (see below) should also take on the responsibility for test execution and management for the cross-party test phases; whether or not a specialist system integrator is used. This aligns with the recommendations made in the testing strategy.

50. **Design Management.** System integration activity can reveal issues or non-conformances ahead of and in addition to those identified through testing, particularly with interface specifications. These issues and defects may require modifications of the design of the system. Modifying the design is not usually part of the system integration process but there is a clear linkage with the design management processes of issue and defect investigation and resolution, change evaluation and management and configuration management and control The body that is responsible for undertaking system integration (and testing) is usually well placed to provide direct support to these design management processes; for example investigating issues and defects and recommending resolutions for approval (usually to a Design Authority-type function in a major programme or operational service). If the body undertaking system integration is seen to be independent from the individual parties and service providers delivering solution components, then this greatly assists in its ability to determine 'best for programme' solutions to issues and defects and gain agreement to these. This factor is taken into account in the assessment of the system integration responsibility options in section 4.

51. Where a design change is proposed to resolve an issue or defect, it is vital that any impacts on other areas of the system are fully considered (especially at the interface points) and that the configurations of design baselines, documentation and testing environments, etc. are carefully controlled and aligned across all parties. This is one of the reasons that issues are so costly and time-consuming to rectify during formal testing hence the importance of considering methods to reveal and fix any issues ahead of formal testing; i.e. during design and build. As for defect investigation and resolution, the body responsible for system integration is usually well placed to evaluate design changes and control associated configurations.

52. It is therefore recommended that the body responsible for system integration provides direct support to design management processes; particularly in respect of issues and defects that require cross-party (cross-interface) resolutions. The Governance and Assurance workpackage is developing these design management arrangements so, as identified in Table 1, this strategy needs to be aligned with the final governance and assurance arrangements once they are agreed.

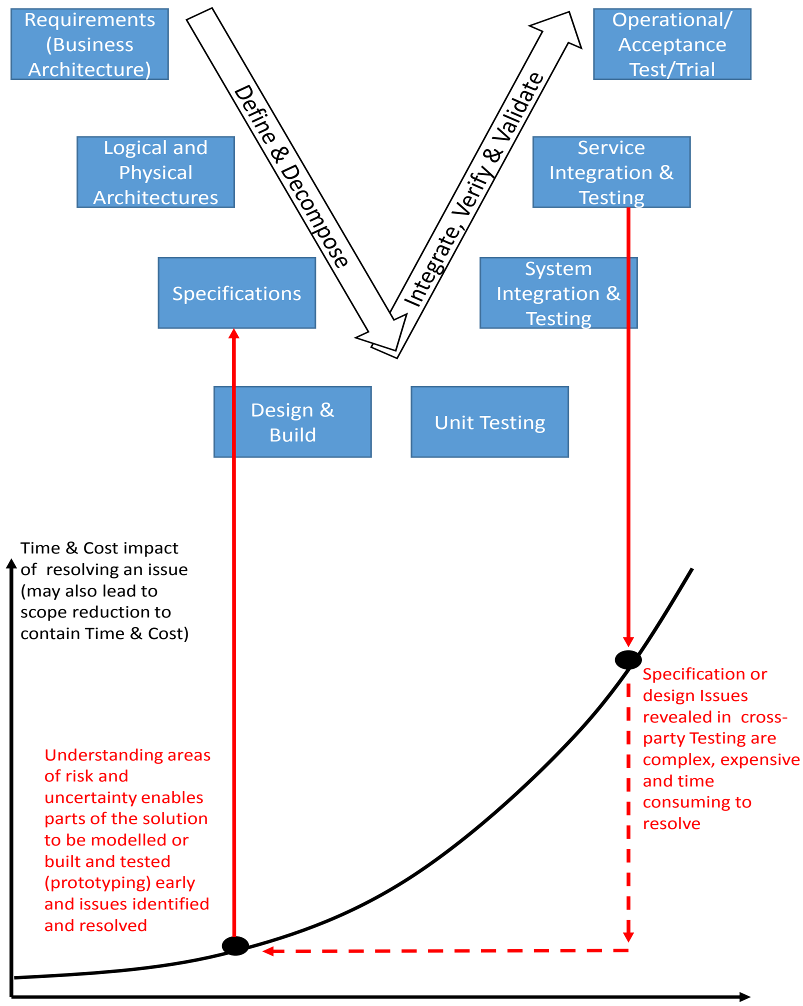
System Integration Added Value

53. In the context of switching programme delivery, it might not be clear what value that a discrete system integration function adds over and above, say, just allowing the physical architecture components to be designed, built and tested by the various parties and

providers and then subjecting these to the testing regime as defined in the testing strategy. There are two key areas of added value that an effective and appropriately tailored system integration strategy brings to the switching programme:

Activity and Skills Gaps. Without an expert system integration function as outlined in this section, adopting the approaches and techniques discussed in section 5, there could be a lack of focus on how best to progressively deliver and realise the whole system requirements from the perspective of understanding and mitigating interface and integration risks. System integration is widely recognised as a key skill and activity for the realisation of complex systems and an experienced system integration body, that is seen as independent from individual parties and providers, will bring: techniques and approaches appropriate to early identification and mitigation of integration risks; close co-ordination and management of DBT activity and readiness; and an ability to foster collaboration across the various parties and providers during DBT.

Early Design Issue Identification and Resolution. As detailed in Table 3, the design definition and decomposition process in imperfect leading to inaccuracies and ambiguities in design specifications. The impacts of these are particularly acute at the interface points between system components and sub-system elements delivered by different parties where different interpretations of specifications can lead to design issues and defects Without a system integration function, it is likely that these issues would remain undetected until the formal, cross-party test phases where the cost and time penalty of investigation and resolution is very high. With careful analysis and planning, these interface and integration risks can be understood during DLS and appropriate techniques deployed during DBT to mitigate these; such as early prototyping, design walk-throughs, iterative testing using emulators, test stubs, etc. Given that the cost of a design change increases rapidly in the later programme phases, there can be a large net programme cost and time benefit from deploying these de-risking techniques as illustrated in Figure 2 below.



[Figure 2 – Cost of Change and Early Identification and Resolution Opportunities]

System Integration Activity Summary

54. Based on the analysis and discussion above, it is recommended that the activities that should be undertaken as part of a system integration function for Switching include the following:

- **Prepare for Integration**
    - Identify and define check points for the correct operation and integrity of the interfaces and the selected system functions.
    - Define the integration strategy. Integration should be performed according to a predefined integration strategy that sequences the order for assembling the implemented system elements based on the priorities of the system requirements and architecture definition focusing on the interfaces, while minimizing integration time, cost, and risks.
    - This strategy should align with the planned testing strategy and targeted on mitigating the predicted risks to a level in line with the risk appetite for the

programme. There should be clear delineation between system integration and design management functions.

- **Establish the integration plan** (this activity is carried out concurrently to the detailed level design activity and aligned with the testing strategy) that defines:
    - Finalise/refine the optimised integration strategy for the design and its risks: order of aggregates assembly using appropriate integration approaches (see section 5).
    - Identify quality gates / checkpoints to progress through integration with clear acceptance criteria (to be aligned with testing strategy)
    - Define the V&V activities to be undertaken for the purpose of integration. This can include formal test phases and informal/de-risking activities such as prototyping.
    - Define the configurations of the aggregates to be assembled and tested and associated configurations of integration and testing 'means' (emulators, stubs, environments, etc.)
    - Define the integration means and testing means required (dedicated enabling products) that may include integration procedures, integration tools, testing tools (simulators, emulators stubs/caps, test benches, etc.), test data and test procedures.
- **Obtain the integration means** and testing means as defined in the integration plan and testing plan; the acquisition of the means can be done through various ways such as procurement, development and reuse; usually the acquisition of the complete set of means is a mix of these methods.
- **Oversee and assure readiness** of each party for integration and testing. Monitor design and build across the multiple parties and service providers and foster collaboration and team-working to ensure design information is shared, particularly at the interfaces, and ambiguities and inaccuracies in the specifications are identified early and resolved.
- **Take delivery** of each implemented solution component/element:
    - Check the delivered configuration, conformance of implemented elements, compatibility of interfaces, and ensure the presence of any mandatory documentation.
- **Assemble the implemented elements/components** into aggregates:
    - Gather the implemented elements to be assembled, the integration means (tools, assembly procedures), and the testing means (testing environments, emulators, stubs, procedures).
    - Connect the implemented elements to constitute aggregates in the order prescribed by the integration plan using the assembly procedures and assembly tools.
    - Add or connect the testing tools to the aggregates as predefined.

- **Verify (test) each aggregate [in line with the Testing Strategy]**:
    - Check the aggregate is correctly assembled according to established procedures.
    - Perform the agreed testing process and procedures (see testing strategy and plan) and check that the aggregate shows the right design properties/specified requirements.
- **Manage the results of integration:**
    - Record integration and test results, investigate and report any issues and non-conformances encountered. These could be due to the integration and testing strategy, the integration and testing enabling systems, execution of the integration or incorrect system or element definition. Where these lead to corrective actions or requirement changes, support investigation and evaluation of these changes from a whole system, integration and test perspective.
    - Assess progress and results against agreed acceptance criteria; take decision-making through programme governance
    - Maintain traceability of the integrated system elements to the integration strategy, system architecture, design, and system requirements including interface requirements and definitions that are necessary for integration.
    - Provide key information items that have been selected for baselines. The Configuration Management process is used to establish and maintain configuration items and baselines. The system integration process identifies candidates for the baseline, and then provides the information items to CM.

System Integration Roles and Responsibilities

System Integration Roles

55. The roles for system integration in terms of management and execution are defined by the scope of activities captured in the previous section. In terms of approval and acceptance, and assurance, of system integration, this should align with the overall governance and assurance arrangements being developed separately as identified in the dependencies table. The roles for system integration are summarised below together with the phases of the programme during which they should take place.

- Developing and maintaining an Integration Strategy and associated Management Plan at whole system/whole programme level (ideally combined with the Testing Strategy and Plan); **from Blueprint through to DBT**
- Developing and implementing a range of appropriate integration approaches and techniques to mitigate the integration/interface risks identified during DLS; **developed during DLS, implemented during DBT**
- Collecting and screening/validating design information (from multiple parties); **commences during DLS and continues through DBT for design baselines and change proposals**
- Managing/overseeing/assuring design, build and test of components/ sub-system elements by individual parties and hence ensuring their 'readiness' for integration and testing; **during DBT**
- Building components/sub-systems into logical 'aggregates' (and management and execution of testing at cross-party levels); **during DBT**
- Development and provision of the 'means' of system integration (and testing); e.g. sandpits, test environments, tools, stubs, emulators, racks, etc.) aligned with system integration (and testing) strategies and plans; **ideally starts prior to DBT to ensure resources are in place to support DBT**
- Managing the configuration of integration and test environments and build standards consistently (aligned with the testing strategy); **throughout DBT**
- Managing and investigating cross--party defects/issues; support to associated change evaluation and configuration management at system level (align with Governance & Assurance arrangements); **throughout DBT**
- Contributing to management/mitigation of technical risk at whole system level (align with Governance & Assurance arrangements); **ideally starting in DLS and continuing through DBT**
- Ensuring accurate and up-to-date and shared DBT knowledge across all parties; e.g. of integration and test plans, build standards, design and interface information, configuration and build standards, issues and changes and their status, etc.; **throughout DBT**

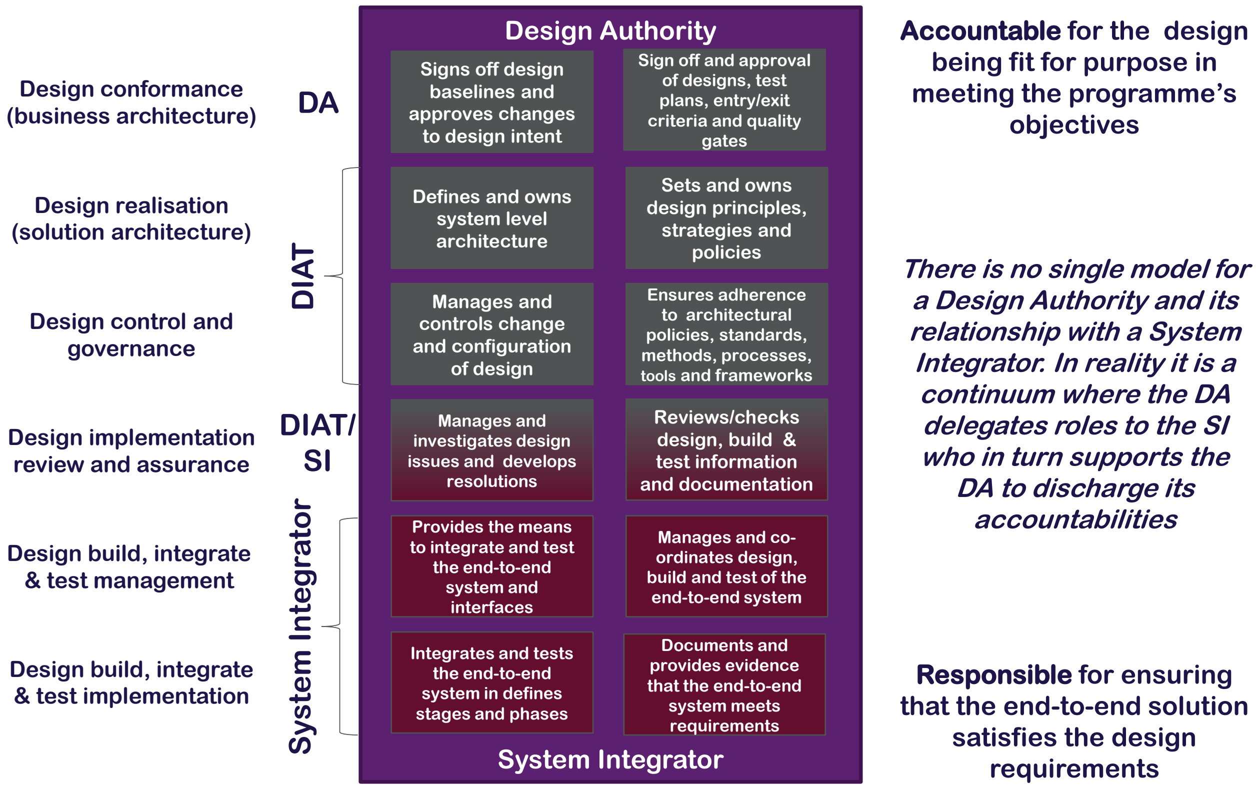
System Integration Relationship with Other Delivery Roles

56. As discussed in the previous section from an activity perspective, system integration is closely linked with testing and design management. In terms of the roles associated with testing, as detailed in the testing strategy, it has already been recommended that the roles of test execution and management for cross-party testing should be part of the system integration roles and these are already reflected in the recommended set of roles above.

57. In terms of design management, the associated roles for system integration need to be understood in relation to the roles of the Design Authority (DA). Within the context of the switching programme, the roles and responsibilities for the DA for the DBT phase are still being developed as part of the governance and assurance workpackage aligned to the top level governance arrangements being determined by the programme board. For the purposes of this version of the system integration strategy, the following roles and responsibilities are assumed for the DA during DBT based on best practice and the experience of SMIP:

    - To ensure that the solution design is 'fit for purpose', and to make or approve changes to the design intent
    - To increase the likelihood and predictability of success whilst reducing the probability and cost of non-conformance and duplication.
    - To ensure that the solution design adheres to a common set of [design] principles and that delivery remains focussed on the strategic goals of the programme.
    - To develop and impose business control and governance over projects and programmes from an architecture and design perspective.
    - To define and enforce adherence to the architecture policies, standards, methodologies, processes, tools and frameworks.

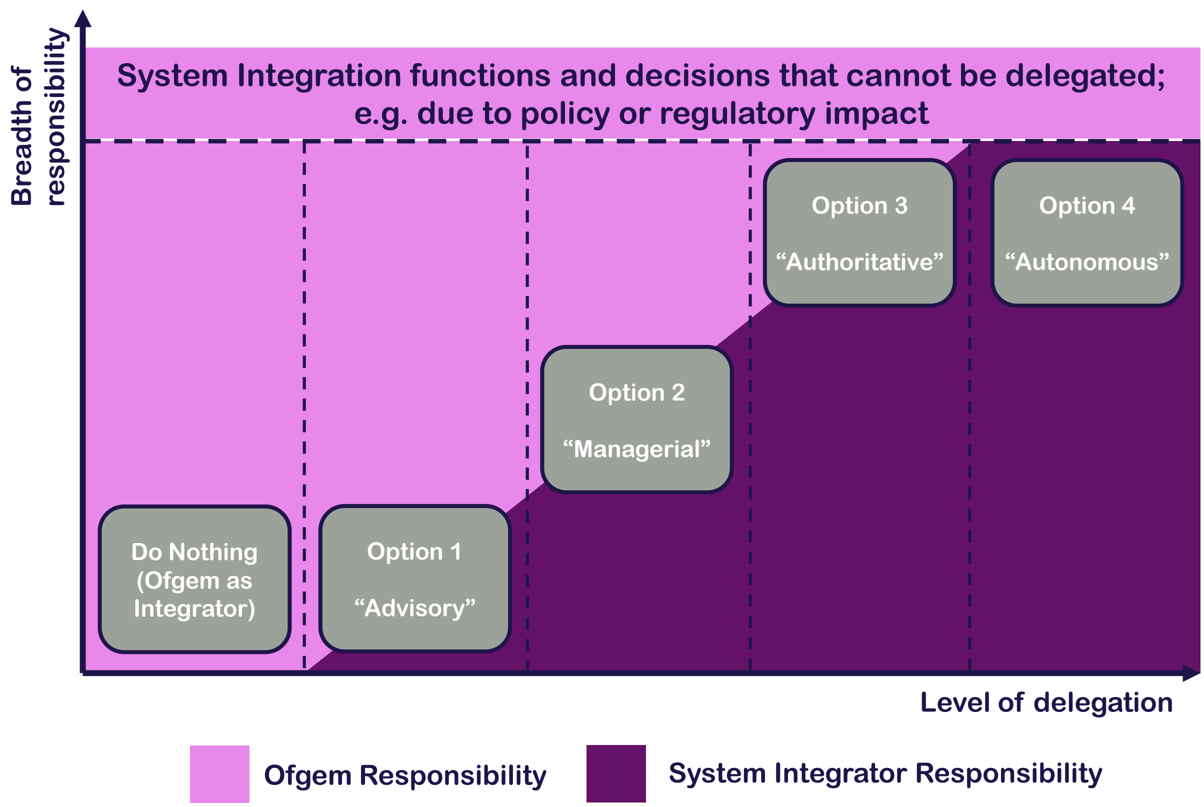
58. On this basis, the system integration roles can be considered as providing direct support to the DA roles, particularly in terms of ensuring a fit for purpose solution and increasing the likelihood of success through reducing the probability and cost of non-conformance. The DA would remain ultimately accountable for the areas above, with the body responsible for system integration responsible for a range of activities in support of these. A possible split between the roles of the DA and SI is illustrated in Figure 3 below, showing where the current Switching Programme DA and DIAT constructs overlay this. It is recommended that further work be undertaken to refine the system integration roles once the wider governance and assurance arrangements for DBT are agreed; in particular in respect of the DA.



[Figure 3 – System Integration Roles in Support of Design Authority Roles]

Systems Integration Responsibilities and Associated Options

59. There are options available for the responsible body that could undertake the roles defined above. In the base case (counterfactual or 'do nothing), the 'proactive' system integration activities and roles defined above that go beyond those defined in the testing strategy would not get done (i.e. would not be planned and resourced) and the impacts of integration and interface risks not being proactively identified and managed would impact the programme late in the testing phase. Beyond the 'do nothing' option, there exists the option to appoint a specialist system integrator body to perform the integration roles on behalf of the programme 'authority'. Within this option, there are a range of sub-options for the specialist system integrator that reflect the degree of empowerment this body could be given. These sub-options are illustrated in Figure 4 below with further definition in Table 4.



[Figure 4 – Ofgem/System Integrator Balance]

Option 1: Advisory	Option 2: Managerial	Option 3: Authoritative	Option 4: Autonomous
Independent, Trusted Expert Information gathering and sharing Intermediary role Highlights	Independent, Trusted Expert Information gathering and sharing Responsible for Design, Build and	Manages Knowledge & Information Oversees/assures component DBT Responsible for all Design, Build and Test Management at	Decides on system component specs Assures parties DBT plans and processes Oversees component DBT and agrees readiness for SIT

issues/risks and develops potential resolutions 'early' Does not directly manage any DBT activities No decision making authority – feeds into DA and other Governance bodies	Test Management at System (x-party) level Co-ordinates issue and risk resolution Co-ordinates Change and Configuration Management process May have some limited decision making powers	System (x-party) level Resolves issues and risks (within delegations) Authorises Change and manages Configuration (within delegations)	Responsible for all Design, Build and Test Management at System (x-party) level Resolves issues and risks, authorises change and manages configuration unless it impacts on a 'key user requirement'

[Table 4 – Specialist System Integrator Options]

60. The options in Table 4 above for a specialist system integrator to support Ofgem as the overall programme and design authority have initially been assessed together with the 'do nothing' option using a range of applicable factors, including: Delivery (Cost, Time and Quality); the ability to mitigate/address the risks and dependencies identified in Tables 1 and 3; and alignment with relevant Design Principles (particularly, Reliability, Competition, Robustness, Cost/Benefit and Implementation). Table 5 below records the result of that options assessment using the scoring mechanism as explained at Appendix 4.

Responsibility Option	**Cost**	**Time**	**Quality**	**Risk Reduction**	**Alignment with Design Principles**
Do Nothing – System Integration responsibility falls to CRS Provider (for CRS only) with no dedicated, experienced integrator at whole system, whole solution level	Likely to be significant net time impacts due to integration and interface issues requiring resolution late in programme testing	Likely to be significant net time impacts due to integration and interface issues requiring resolution late in programme testing	Likely to require scope reductions to contain time and cost leading to performance being traded-out	Limited proactive identification and management of detailed design integration and interface risks and deployment of associated risk reduction techniques such as prototyping	No 'new' body required in governance – may be simpler and some parties will be more comfortable without 'another decision maker'. Does not support design robustness, flexibility, reliability or ease/speed/risk of implementation
Advisory System Integrator	Net costs are uncertain – an advisory SI would require modest investment but its value-	As for cost.	Does not reflect best practice and lessons learned; the SI role needs appropriate	Integration risks would be better understood and issues identified early, but	New 'role' to be accommodated within governance may add complexity. However, this should help

	add would depend on the various parties implementing its advice. Without incentives and any authority, this may often conflict with their individual delivery priorities		empowerment from the programme/ design authority as part of the overall governance regime	resourcing and implementing the means to mitigate and resolve them may have to be separately negotiated on a case by case basis between all parties	influence design robustness, flexibility and reliability and partially improve ease/speed/risk of implementation
Managerial System Integrator	Integration and testing would be better planned and co-ordinated across parties, and issues identified with resolutions proposed but still relies on cross-party agreement to implement noting benefits may not fall where costs lie	As for cost	Does not fully reflect best practice and lessons learned; the SI role needs appropriate empowerment from the programme/ design authority as part of the overall governance regime	Risks and issues would be identified with mitigations and resolutions developed. However, implementation may still get caught in bottleneck if all decisions have to be referred up to higher authority	New 'role' to be accommodated within governance may add complexity. However, this should help influence design robustness, flexibility and reliability and significantly improve ease/speed/risk of implementation
Authoritative System Integrator	Appropriate levels of delegation would mean that many lower level issues and risks could be resolved early, minimising what needs to be escalated. Although some of these may require earlier investment, the SI would gain agreement on the basis of net cost-	As for cost.	Aligns with best practice and lessons learned.	Some risk mitigation measures and issue resolutions may need to be escalated if agreement cannot be reached or they exceed certain thresholds	Independent and authoritative body aligned to Ofgem in its role as DA. Roles and Responsibilities of SI in relation to DA and wider Governance would need to be clarified to avoid ambiguity and confusion

	benefit.				
Autonomous System Integrator	To make an SI accountable for delivery, it would need all the necessary levers to control parties and providers. Without these, the SI is likely to add significant risk contingency	May lead to conflicts requiring lengthy resolution of parties disagree with SI approaches and its proposed resolutions	There is no evidence to suggest that a SI needs to be given full autonomy over design decisions to discharge its roles – the DA ususally remains ultimately accountable and needs to be seen as so by all parties	In theory, SI has power to implement all techniques and approaches it sees as appropriate to mitigate integration risks and to decide on issue and defect resolutions across parties.	Cuts across individual license responsibilities in this cross-party delivery environment. Would be difficult to implement outside of the CRS provision given that the SI could not own and control 'the whole supply chain'.

[Table 5 – Assessment of System Integration Responsibility Options for Switching]

61. Based on this initial assessment, it is recommended that an independent, specialist system integrator body be appointed to plan, manage and undertake system integration roles and approaches in line with this this strategy (i.e. reject the 'do nothing' option). The system integrator role should include Test Execution and Management of the cross-party Test Phases. This system integrator role should provide direct support to the Design Authority and the SI and DA roles should be fully aligned and complimentary.

62. Based on this initial assessment, it is recommended that a 'Managerial' or 'Authoritative' model is adopted for the System Integrator. However, the precise roles of the system integrator, and the degree of delegation it is given, should be finalised within this strategy once the integration risks are better understood and the testing and governance and assurance arrangements are finalised.

Other considerations

63. If a specialist system integrator is used as recommended, then there are further considerations of who this should be, and when and how they are brought on board. In terms of the 'who and how', and assuming that the skills and resources do not exist today in Ofgem, then these would either need to be recruited or procured (either by Ofgem or DCC on their behalf).

64. In terms of 'when', ideally (and especially for Options 3 and 4) this should be during DLS phase when the system integrator can influence the design and interface specifications and partitioning of functionality across the parties and component systems and develop the integration strategy and plans in parallel. In any event, they would need time to prepare for DBT: putting in place design forums, knowledge management arrangements,

defect & change & configuration management procedures, testing plans, procuring the integration and testing means, etc.

Systems Integration Approaches, Deliverables and Documentation

System Integration Approaches

65. Detailed approaches for system integration should be developed in the DLS phase as part of the System Integration Management plan summarised in the deliverables and documentation below. These detailed SI approaches can only be developed once the final solution architecture is chosen and the detailed level specification is developed. A range of general SI approaches are summarised in Table 6 below. These should be tailored to the final solution and detailed integration risks as they mature during DLS. In addition to these generic approaches, the detailed SI approach developed during DLS should embody applicable Agile principles as summarised above, such as prototyping, and these should be aligned with and complementary to any testing de-risk approaches.

66. Usually, a mixed integration approach is selected as a trade-off between the different approaches listed in Table 6 below, allowing optimisation of work and adaptation of the process to the system under development. This should take into account the realisation time of the implemented components, their delivery scheduled order, their level of complexity, the technical risks, the availability of integration tools and environments, cost, deadlines, specific personnel capability, etc.

Integration Approach	Description
Global Integration	Also known as *big-bang integration*; all the delivered/implemented elements/components are assembled in only one step. • This technique is simple and does not require simulating/emulating the implemented elements not being available at that time. • Difficult to detect and localize faults; interface faults are detected late. • Should be reserved for simple systems, with few interactions and few implemented elements without technological risks.
Integration "with the Stream"	The delivered/implemented elements/components are assembled as they become available. • Allows starting the integration quickly. • Complex to implement because of the necessity to simulate the implemented elements not yet available. Impossible to control the end-to-end "functional chains"; consequently, global tests are postponed very late in the schedule. • Should be reserved for well-known and controlled systems without technological risks.
Incremental Integration	In a predefined order, one or a very few implemented elements are added to an already integrated increment of implemented elements. • Fast localization of faults: a new fault is usually localized in lately integrated implemented elements or dependent of a faulty interface. • Require simulators/emulators for absent implemented elements. Requires more test cases, as each implemented element addition requires the testing of the new configuration and regression testing. • Applicable to any type of architecture.

Subsets Integration	Implemented elements are assembled by subsets, and then subsets are assembled together (a subset is an aggregate); could also be called "functional chains integration". • Time saving due to parallel integration of subsets; delivery of partial products is possible. Requires less means and fewer test cases than integration by increments. • Subsets shall be defined during the design. • Applicable to architectures composed of sub-systems.
Top-Down Integration	Implemented elements or aggregates are integrated in their activation or utilization order. • Availability of a skeleton and early detection of architectural faults, definition of test cases close to reality, and the re-use of test data sets possible. • Many stubs/caps need to be created; difficult to define test cases of the leaf-implemented elements (lowest level). • Mainly used in software domain. Start from the implemented element of higher level; implemented elements of lower level are added until leaf-implemented elements.
Bottom-Up Integration	Implemented elements or aggregates are integrated in the opposite order of their activation or utilization. • Easy definition of test cases - early detection of faults (usually localized in the leaf-implemented elements); reduce the number of simulators to be used. An aggregate can be a sub-system. • Test cases shall be redefined for each step, drivers are difficult to define and realize, implemented elements of lower levels are "over-tested", and does not allow to quickly detecting the architectural faults. • Mainly used in software domain and in any kind of system.
Criterion Driven Integration	The most critical implemented elements compared to the selected criterion are first integrated (dependability, complexity, technological innovation, etc.). Criteria are generally related to risks. • Allow testing early and intensively critical implemented elements; early verification of design choices. • Test cases and test data sets are difficult to define.

[Table 6 – System Integration Approaches]

Systems Integration Deliverables and Documentation

67. A range of deliverables and documentation will need to be developed both leading up to and during the DBT phase of the programme. The key deliverables and documents are described in Table 7 below.

Document/Deliverable	Purpose/Scope	Who Produces	When
System Integration Strategy	To define the overall objectives and roles and responsibilities	Ofgem (accountable) supported by DCC (responsible)	By end of Blueprint, but refined during DLS
Programme Level System Integration Management Plan	To define the detailed approaches to be adopted for system	Ofgem (accountable) supported by	By end of DLS. Note: SI roles and responsibilities will

	integration aligned to detailed design and risks, including detailed roles and responsibilities	DCC (responsible)	need to be defined in Licences and Codes via Enactment (out dependency)
Individual Integration Plans (aligned with or combined with Test Plans)	These cover how the overall Integration (and Test) management plans will be implemented by individual parties including the bodies responsible for cross-party integration and testing	Relevant parties responsible for Design, Build, Integration and Testing	A minimum time (to be specified) after start of DBT
Integration (and Testing) Means: emulators/ simulators, Test Data, Stubs, Environments, Labs and Tools	These are software and hardware products developed to enable connection/ assembly and execution of representative tests on partial system elements and the whole system as designed and built	System Integrator (or body responsible for system integration and testing)	Set up prior to and maintained for duration of integration and testing. Should consider re-use where possible. Some items may need to be retained for enduring business
Integration Reports (Readiness, Progress and Completion)	Required to authorise readiness for integration (and testing), monitoring of progress and report issues and their resolution	All parties in line with respective Integration (and testing) plans	Periodically and to support key milestones, checkpoints, etc.

[Table 7 – Main System Integration Deliverables and Documentation for Switching]

68. As the programme progresses into the DLS phase, the strategy must be further refined to reflect the evolving solution design and testing plans developed to describe how this strategy will be put in place and enacted across the parties involved. This is discussed further below under 'next steps'.

Next Steps, Conclusions and Recommendations

Next steps – Requirements for the DLS and Enactment Phases

69. As mentioned previously, there are many interdependencies between system integration and other areas of the programme, notably Design Management, the Solution Architecture, Testing, Governance & Assurance and the Transition Strategy. This version of the system integration strategy has taken into account those areas as far as they have been defined noting that there are still a range of options left open in each of these areas and other areas of uncertainty. Given this, it is recommended that the system integration strategy continues to be refined and matured during the DLS phase as these related areas develop and mature. It is further recommended that the Testing Strategy is merged with the system integration strategy, or that testing becomes a sub-strategy to system integration.

70. A strategy needs a clear management plan to allow that strategy to be effectively enacted by all the parties involved in a controlled and consistent way. It is therefore recommended that an overall System Integration Management Plan be developed during the DLS phase as per Table 7. It is further recommended that the Testing Management Plan is merged with, or at least becomes a sub-plan of, the System Integration Management Plan.

Conclusions and Recommendations

71. A well designed system integration strategy that is enacted effectively by an overall suitably empowered body responsible for integration at the 'whole system, whole programme level' has the potential to significantly de-risk the later stages of DBT and hence save time and cost. This is achieved by: effective co-ordination and oversight of DBT activity across all parties; early identification and mitigation of integration risks using appropriate approaches and techniques; providing greater assurance that all parties and the parts of the system they are responsible for are ready for planned integration and testing; and effective management and execution of integration and testing across parties.

72. The new switching arrangements are 'more than the sum of the parts' and, within the multi-party delivery environment relevant to the new switching arrangements, it is vital that individual parties are managed and co-ordinated 'like they are one team' to ensure that design, build, integration and testing are undertaken in line with over-arching strategies and plans that address the key challenges and risks for the programme.

73. It is recommended that:

    System integration becomes a recognised part of the overall Delivery Strategy, closely coupled with Testing and Governance & Assurance, but with key interdependencies also with other parts of the programme such as Transition and the Solution Architecture.

Consideration should be given to managing the system integration and testing strategies together as linked strategies in later iterations, as well as the respective management plans when these are produced in the DLS phase.

The responsible body selected to undertake system integration should take on the responsibility for test execution and management for cross-party test phases; whether or not a specialist system integrator is used. This aligns with the recommendations made in the testing strategy.

The body given responsibility for system integration should provide direct support to design management processes; particularly in respect of issues and defects that require cross-party (cross-interface) resolutions. The Governance and Assurance workpackage is developing these design management arrangements so this strategy needs to be aligned with the final governance and assurance arrangements once they are agreed.

The following roles, as a minimum, should be undertaken by the body responsible for system integration:

- Developing and maintaining an Integration Strategy and Management Plan
- Developing and implementing a range of appropriate integration approaches and techniques to mitigate the integration/interface risks identified during DLS
- Collecting and screening/validating design information (from multiple parties)
- Managing/overseeing/assuring design, build and test of components/ sub-system elements by individual parties and hence ensuring their 'readiness' for integration and testing
- Building components/sub-systems into logical 'aggregates' (and management and execution of testing at cross-party levels)
- Development and provision of the 'means' of system integration (and testing)
- Managing the configuration of integration and test environments and build standards consistently
- Managing and investigating cross--party defects/issues; support to associated change evaluation and configuration management at system level
- Contributing to management/mitigation of technical risk at whole system level
- Ensuring accurate and up-to-date and shared DBT knowledge across all parties

It is recommended that further work be undertaken to refine the system integration roles to align with the testing roles (when finalised) and with the wider governance and assurance arrangements for DBT once agreed; in particular in respect of the complimentary DA roles.

An independent, specialist system integrator body should be appointed to plan, manage and undertake system integration roles and approaches in line with this this strategy (i.e. reject the ‘do nothing’ option). This system integrator would ideally be appointed during DLS.

The system integrator role should include Test Execution and Management of the cross-party Test Phases. This system integrator role should provide direct support to the Design Authority and the SI and DA roles should be fully aligned and complimentary.

Based on the initial assessment, it is recommended that a ‘Managerial’ or ‘Authoritative’ model is adopted for the System Integrator. However, the precise roles of the system integrator, and the degree of delegation it is given, should be finalised within this strategy once the integration risks are better understood and the testing and governance and assurance arrangements are finalised.

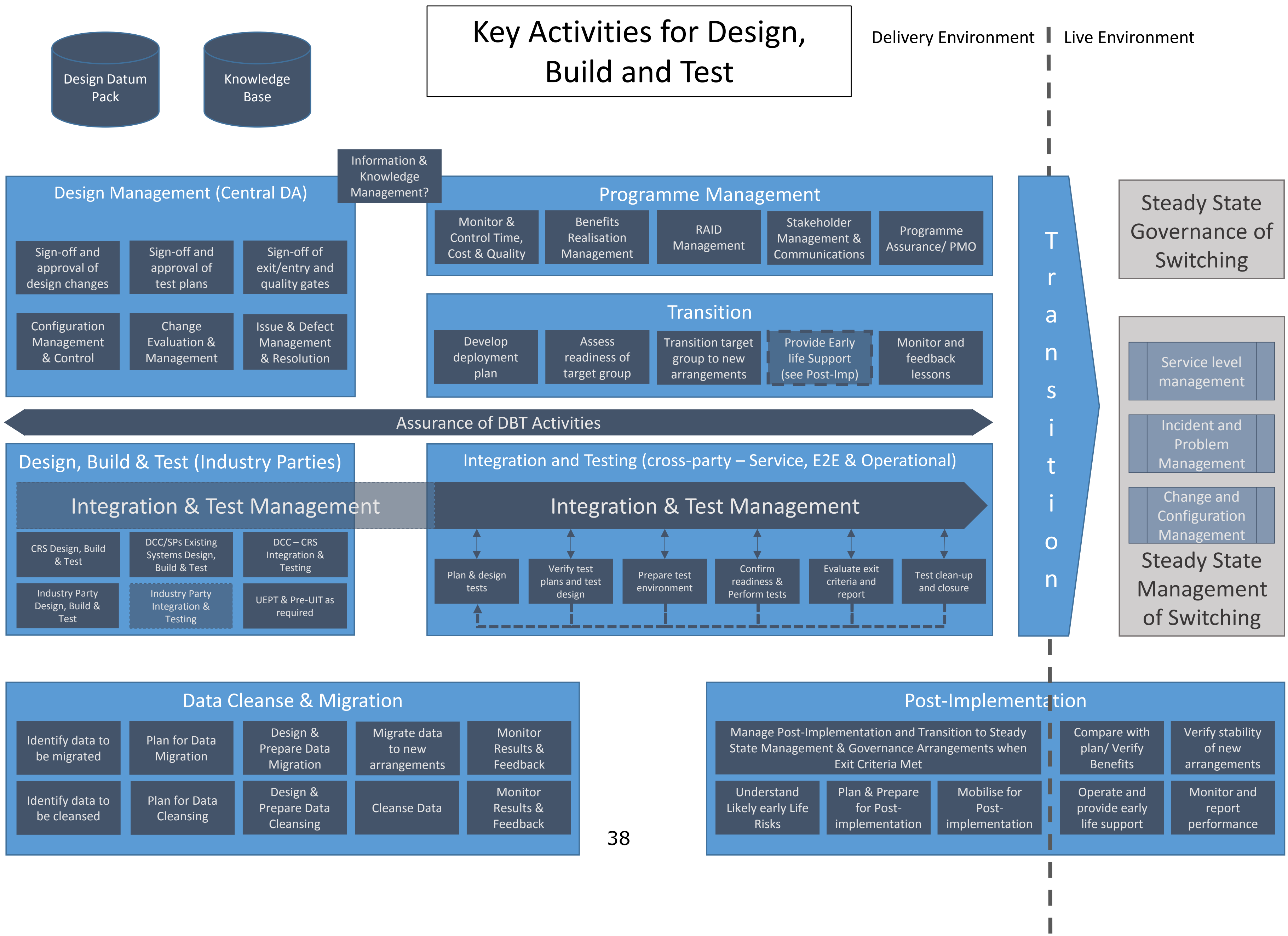
Further work should undertaken to ensure effective ‘design management’ arrangements are developed for delivery of the new switching arrangements, including Issue and Defect Resolution and Change and Configuration Management. These could be separate, part of the wider Governance and Assurance arrangements or a part of the Systems Integration strategy.

Detailed approaches for system integration should be developed in the DLS phase as part of the System Integration Management Plan once the final solution architecture is chosen and the detailed level specification is developed. These should be tailored to the final solution and detailed integration risks as they mature during DLS. In addition to these integration approaches, the use of applicable Agile principles and techniques should also be considered, such as prototyping, and these should be aligned with and complementary to any testing de-risk approaches.

The system integration strategy should continue to be refined and matured during the DLS phase as the interrelated areas develop and mature.

An overall System Integration Management Plan be developed during the DLS phase.

Appendix 1 – Diagram summarising key activities for DBT



Appendix 2 – Option Assessment Scoring Matrix

The following matrix has been used to assess the options based on a Red (R), Amber (A), Yellow (Y), Green (G) scoring against each of the factors relevant to the particular option being evaluated.

Evaluation Category	**Cost (Net)**	**Time (Net)**	**Quality (of Test outcomes)**	**Risk Reduction potential**	**Alignment with Design Principles**
Key to scoring (Red, Amber, Yellow, Green)	G = Likely to provide an overall cost-benefit Y – Likely to be cost neutral or costs are not significant A – Significant costs with benefits uncertain R – Not cost-effective; net cost overall	G = Likely to provide an overall time benefit Y – Likely to be neutral on time or schedule increases are not significant A – Significant time penalty with benefits uncertain R – Not time-effective; net delay overall	G – makes a significant positive contribution to achievement of outcomes Y – Makes a positive contribution to outcomes but with uncertainly A – May impact negatively on required outcomes R – likely to undermine required outcomes	G – significantly mitigates one or more of the key risks identified Y – May mitigates one or more risks but contribution uncertain A – Unlikely to mitigate any of the key risks identified R – may increase the level or risk exposure	G – significantly supports one or more design principles Y – Significantly supports one or more design principles but may have some small negative impacts A – May support one or more design principles but negatively affects others R – Potentially at odds with one or more design principles

Appendix 3 – Summary of Best Practice and Lessons Learned Applicable to Systems Integration for Switching

See separate document

Appendix 4 – Product Description

<table>
<tr><td>Product Title:</td><td>System Integration Strategy</td></tr>
<tr><td>Format / Presentation:</td><td>System Integration Strategy document and slide pack presentation</td></tr>
<tr><td>High level Summary/ Product Purpose:</td><td>The System Integration (SI) strategy will describe the approach to ensuring the effective integration of the component parts of the new Switching arrangements to be designed, built and tested by the multiple parties involved in the gas and electricity retail markets.

Current switching arrangements involve multiple parties exchanging data between them in a set of well-defined but complex interactions to effect a change of supplier for gas and/or electricity. The new switching arrangements seek to harmonise arrangements across the gas and electricity markets and implement faster, more reliable switching based on revised industry rules and the provision of a Central Registration Service (CRS).

The delivery of the new switching arrangements will require the design, build and test of the CRS plus a range of changes to be made to the various participants' existing systems and processes. Depending on the final solution chosen, this will significantly change the nature and complexity of the interfaces between the various parties (including the CRS Provider) to effect a successful switch, together with interfaces to other systems and participants external to the boundary of the new switching arrangements.

Irrespective of the scope of the CRS within the final chosen architecture, responsibility for implementation of the new switching arrangements will span multiple parties. There is therefore a need to consider how the overall integration aspects will be managed to mitigate integration risks and reduce cost and time impacts from integration problems arising late in the implementation phase.</td></tr>
<tr><td>Composition:</td><td>
<table>
<tr><th>Section</th><th>Description</th></tr>
<tr><td>Introduction and background</td><td>Narrative providing background relating to (for example):
Switching Programme
The Integration 'challenge' or 'problem'
The value of effective System Integration</td></tr>
<tr><td>Purpose</td><td>Explanation of the purpose of the SI Strategy
Explain the key integration risks and how these could be mitigated through the SI strategy</td></tr>
<tr><td>System Integration Functions and Roles</td><td>Narrative setting out the System Integration functions and roles that need to be performed to mitigate the integration risks relevant to Switching
What are the SI functions and roles applicable to Switching and how do these relate to 'Interface Management'?
What would be a proportionate level of SI effort (resources, skills, experience) to mitigate the</td></tr>
</table>
</td></tr>
</table>

	identified integration risks? Timeframe: When would the SI functions/roles need to be performed and how would these fit in with the DLS, Enactment and DBT phases, including Post-Implementation Relationship with other functions/roles being examined for Governance and Assurance within the programme; e.g. Design Authority, Programme Director, etc. How the SI functions/roles are affected by the likely solution options and related delivery and transition strategies; e.g. multiple releases Mapping of the SI functions and roles to current proposals across Delivery and other workstreams to understand any 'gaps'
System Integration Responsibilities and Options	Narrative setting out: Who will be responsible and accountable for managing system integration within current proposals for governance and assurance (noting any gaps identified above) Identify any options to address the gaps identified and provide a proportionate level of SI expertise as identified in the previous section – e.g. just enhance proposed DA and other governance functions or whether the use of a Specialist System Integrator will add value
Option Evaluation	Narrative setting out the options – covering whether a specialist System Integrator would add value compared to current proposed governance & assurance (& regulatory) mechanisms and, if so, the extent of its role, who could undertake the role, how and when would they be brought on board, etc.: Whether a specialist SI adds value compared to other options The extent of its delegated powers How and when it would be procured/brought on board A key element of the option evaluation is understanding how a specialist System Integrator would work within the proposed governance structure for DLS, Enactment and Delivery and whether current structures and organisations will have the skills and empowerment required to manage integration and mitigate the associated risks effectively.

<table>
<tr><td></td><td><table><tr><td>Next Steps</td><td>Explain the next steps to take forward the proposed SI strategy though DLS, Enactment, Delivery and Post-Implementation
Development of overall SI Plan in DLS
How Integration issues will influence detailed design
Detailed roles and responsibilities for recommended option from evaluation above and how this changes through programme phase (DLS, Enactment, Delivery and Post-Implementation)</td></tr></table></td></tr>
<tr><td>Inbound dependencies
(What feeds into product):</td><td>Business Process Design – Solution Architecture
Delivery Strategy – Governance & Assurance
Delivery Strategy – Testing Strategy
Delivery Strategy – Transition Strategy</td></tr>
<tr><td>Outbound dependencies (What product will feed into):</td><td>Delivery Strategy – Governance & Assurance
Delivery Strategy – Testing Strategy
Business process Design – Solution Architecture
Commercial/Procurement
Regulation</td></tr>
</table>

RESOURCES AND ROLES

Product Approver (Accountable):	DA		
Product owner (Responsible): **Supported by:**	DCC, David Liversidge Delivery Strategy Design Team		
Delivery from:	**July 2016**	**Due date:**	**October 2016**

REVIEW AND APPROVAL

<table>
<tr><td rowspan="3">Reviewers (Consult/Inform):</td><td colspan="3"><u>During production</u>
Consult:
Delivery Strategy Design Team and User Group, EDAG
Inform:
DCC, other interdependent workstreams/workpackages as required</td></tr>
<tr><td colspan="3"><u>During assurance and formal consultation</u></td></tr>
<tr><td colspan="3"><u>During finalisation</u></td></tr>
<tr><td>Planned iterations</td><td>V0.1</td><td>Draft SI strategy document and slide pack submitted to Ofgem and Delivery Strategy DT for review and scrutiny</td><td>End August 2016</td></tr>
</table>

<table>
<tr><td rowspan="2"></td><td>V1.0</td><td colspan="2">Updated versions taking into account Ofgem/DT feedback and ready for review through UG, EDAG, DA</td><td>End September 2016</td></tr>
<tr><td>V1.1</td><td colspan="2">Baseline SI Strategy taking into account review feedback and Ofgem direction</td><td>End October 2016</td></tr>
<tr><td>Approver</td><td colspan="2">DA</td><td>Approval date:</td><td>October 2016</td></tr>
<tr><td>Acceptance Criteria:</td><td colspan="4">
<table>
<tr><td>Quality Criteria</td><td>Tolerance</td><td>Quality Method</td></tr>
<tr><td>Sets out clear mitigations for key System Integration issues and risks together with most appropriate roles and responsibilities for managing these</td><td>Not applicable</td><td>Review by Ofgem and Switching Programme governance (Design Team, User Group, EDAG)</td></tr>
<tr><td>Addresses any significant concerns raised by stakeholders</td><td>Not applicable</td><td>To satisfaction of Ofgem</td></tr>
</table>
</td></tr>
<tr><td>Date of final version:</td><td colspan="4">October 2016</td></tr>
</table>