# Environment Management Plan

**Version: 1.0**
**Author: Switching Operator**
**Classification: DCC Controlled**

## Table of Contents

DCC Controlled

## List of Tables

## List of Figures

# Document control

## Revision history

| Revision date | Summary of changes | Version number |
|---|---|---|
| 01-09-2023 | Initial Draft | 0.1 |
| 15-09-2023 | Content added to reflect discussions at the RTS/DCC/NC workshop (held on 12/09/23) and comments received from RTS on 12/09/23. | 0.2 |
| 26-09-2023 | Table 1 Test Data Management RACI Model has been added following discussion at the RTS/DCC/NC workshop held on 25/09/23. | 0.3 |
| 05-10-2023 | Updated following feedback from the REC Code Manager received on 03/10/23 | 0.4 |
| 16-10-2023 | Final version for review following uplift to reflect discussion and agreements at the RTS/DCC/NC workshop on 12/10/23 | 0.5 |
| 09-11-2023 | Final version following uplift to reflect discussion and agreements at the RTS/DCC/NC meeting on 09/11/23 | 0.6 |
| 19-04-2024 | Baselined at v1.0 following the circulation of v0.6 as part of the R0070 consultation alongside the final legal text for approval – noting that no changes were proposed as part of this process. | 1.0 |

**The document was issued to the following representatives at each review cycle:**

**Code Manager:** *Beth Brown, Gemma Dixon, Mark Pearce.*

**DCC**: *Kit Ho; Robbie McMillan, Edgar Mokwunyei, Mark Riley, Marlene Gibson-Skinner, Gordon Wallace.*

## References

| ID | Title |
|---|---|
| [1] | Schedule 25 - Central Switching Service |
| [2] | Schedule 26 - Switching Service Management |

# 1. Executive Summary

The Environment Management Plan describes the environments to be made available to provide ongoing support for testing within the Switching ecosystem. The general usage of each of these environments is covered, as well as roles and responsibilities for REC Parties in the provision and maintenance of these environments.

The Switching Operator is responsible for ensuring that the Central Switching Service (CSS) Provider makes the required CSS environments available. Switching Data Service Providers are required to provide suitable environments to support REC Change Proposals. No obligations are placed on CSS Users (other than Switching Data Service Providers) with regards to provision of environments.

Switching environments are required to support the following types of testing:

- Testing of CSS Operational Releases
- Testing of REC Change Proposals, referred to throughout this document as testing of Major Releases
- External Testing, either of new market entrants or additional testing for Maintenance of Qualification of parties as a result of a significant change either to CSS or to a party system
- Test Tools are made available by the Switching Operator to support the ongoing testing of Operational and Major Releases, as well as to support External Testing. Where a party is required to undertake CSS Connectivity Testing, these arrangements are described.
- Where Switching Data Service Providers and other CSS Users are testing their own internal changes, it is expected that these are tested against Simulators unless agreed with RECCo, as part of the Maintenance of Qualification or as specified in a REC Change Proposal.

## 1.1. Scope

This document describes:

- The environments to be made available by REC parties, including the Central Switching Service Provider and Switching Data Service Providers, to support ongoing testing requirements.
- The general usage of those environment to support External Testing and the testing of Operational and Major Releases.
- The roles and responsibilities relating to the provision and management of the test environments.
- The process to be undertaken to allow REC Parties to connect to the Central Switching Service environments.
- The simulators made available by the Switching Operator to support specific testing requirements.
- The process for demonstrating connectivity to the Central Switching Service environments.
- The general principles expected to be applied by REC Parties in regard to release management and control within the test environments.

# 2. Roles and Responsibilities

Please refer to the REC Main Body and the Schedules 25 and 26 to understand your Party obligations under the REC.

## 2.1. Role of the Switching Operator

The Switching Operator is responsible for:

- Ensuring that the CSS environments (Registration, Address and Reporting Services) as described in this document are available to support the testing requirements.
- The management of environments to support External Testing.
- The management of any required connection by parties to CSS environments.
- The provision of Simulators, maintained to the latest release, required to support External Testing and the testing of Operational and Major Releases.

## 2.2. Role of the Central Switching Service Provider

The Central Switching Service (CSS) Provider is responsible for:

- The provision and maintenance of the CSS (Registration, Address, and Reporting Services) environments in accordance with this document.
- Support for External Testing and the testing of Operational and Major Releases.

## 2.3. Role of the Switching Data Service Providers

The Switching Data Service Providers are responsible for:

- The provision and maintenance of Switching environments in accordance with this document and the Test Data Management Plan.

## 2.4. Role of other CSS Users

No obligations are placed on CSS Users (other than Switching Data Service Providers) with regards to provision of environments. Where other CSS Users may need to undertake testing associated with a Major Release, the environment requirements for this will be articulated as part of the relevant REC Change Proposal.

## 2.5. Responsibilities of Parties

This section details the responsibilities of the REC Parties in relation to the environments and the key activities which are to be undertaken.

The tables follow the RACI model matrix used for identifying roles and responsibilities:

- R = Party who is Responsible for carrying out the activity.
- A = Party who is ultimately Accountable to ensure that the activity is carried out appropriately.
- C = Parties who are Consulted for input or advice.
- I = Parties who are Informed of an activity for reference purposes

| Activity | Switching Operator | CSS Provider | REC Code Manager | Switching Data Service Providers | CSS Users (other than SDSPs) |
|---|---|---|---|---|---|
| The provision, maintenance, and deployment of the CSS environments (Registration, Address and Reporting Services) in accordance with this Environment Management Plan | A | R | I | I | I |
| The provision, maintenance, and deployment by the SDSP of at least one environment capable of integration with CSS | I | I | I | A, R | I |
| The provision and maintenance of Simulators in accordance with this Environment Management Plan | A, R | I | I | I | I |
| Planning, coordination, and management of Operational Releases to the CSS environments | A, R | R | I | I | I |
| Planning, coordination, and management of Major Releases to the CSS environments | A, R | R | I | I | I |
| Planning, coordination, and management of Major Releases to the SDSP environments *<br>* Note that a REC Change Proposal may result in a party other than REC Code Manager leading on the management of a change e.g., an SDSP | C | C | A, R | A, R | I |
| Provide and maintain the database for use with the Test Data Tool | A | R | I | I | I |
| Manage and track CSS Production environment availability | A, R | R | I | I | I |

| | | | | | |
|---|---|---|---|---|---|
| Manage and track CSS test environment availability and usage, ensuring contention is minimised | A, R | R | I | I | I |
| Manage and track SDSP Production environment availability | I | I | I | A, R | I |
| Manage and track SDSP test environment availability and usage, ensuring contention is minimised. | I | I | I | A, R | I |
| Where an SDSP requires to connect to a CSS environment, the DCC Test Simulator High Level Process referenced in Section 5.1.1 Requesting a Physical Connection is complied with | I | I | I | A, R | I |

**Table 1 Environment Management RACI Model**

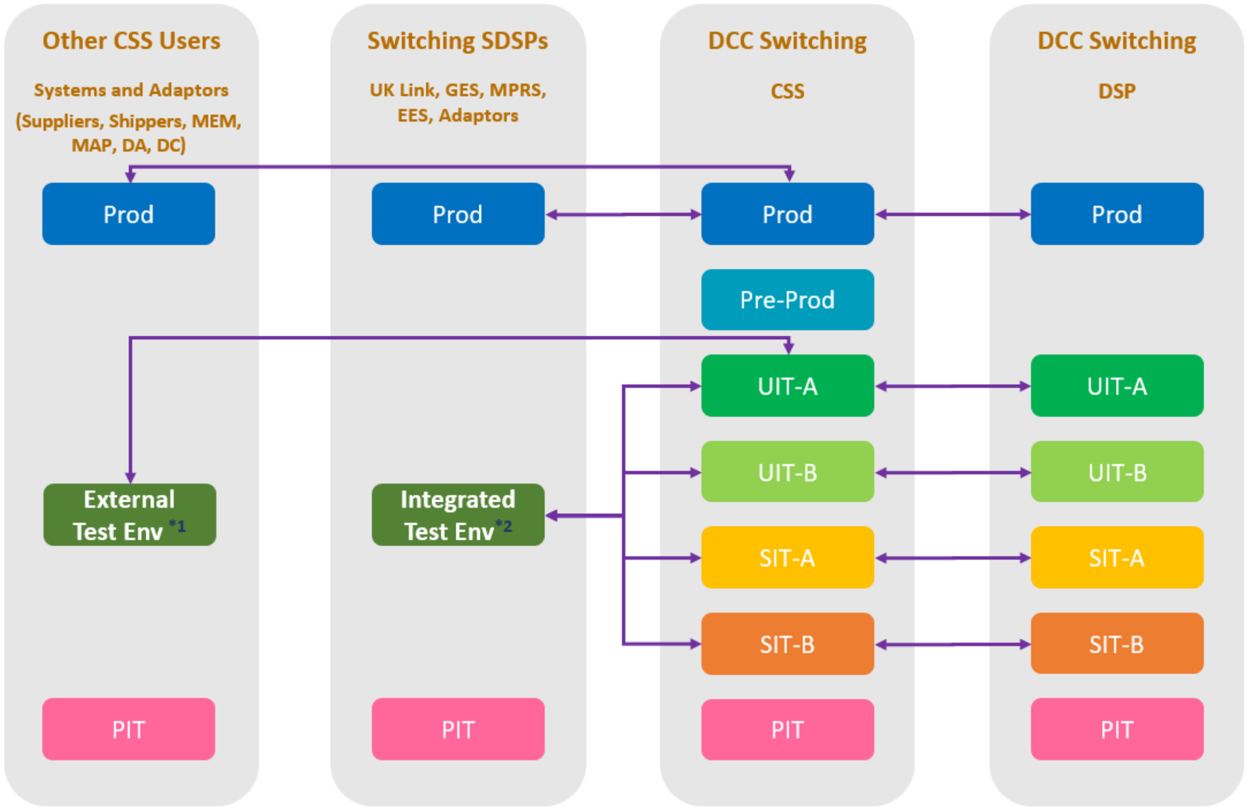
# 3. Environments

This section describes:

- The environments to be provided by the CSS Provider and Switching Data Service Providers
- The general usage of those test environments
- The standard Path to Live for Operational and Major Releases

## 3.1. Environments Overview

Figure 1: below represents the CSS Provider environments and their associated DCC Smart Metering and Switching Data Service Provider environments.



[*1] Other CSS Parties are only obligated to have a PROD environment but will need a test environment for Market Qualification and are expected to have a PIT environment for internal testing.

[*2] SDPS are obligated to have at least one integrated test environment. This can only be connected to one CSS test environment at any time.

**Figure 1: Switching Environments**

## 3.2. Switching Environments

### 3.2.1. PIT environments

REC Parties who connect to CSS are expected, but not obliged, to have a Pre-Integration Test (PIT) environment, where the party can test their functional and non-functional requirements. PIT environments are always internal to the party and entirely within the control of that party, though this may include the environments of any CSS Interface Provider.

### 3.2.2. A-Stream environments

The CSS A-Stream environments are provided for Production support and External Testing and are normally maintained at or near to the Production release levels.

#### 3.2.2.1. SIT-A

The primary purpose of the SIT-A CSS environment is to allow the integration testing of Operational Releases of the CSS (Registration Service, Address Service, and Reporting Service).

The testing of CSS Operational Releases is managed by the Switching Operator.

Although the SIT-A CSS environment is capable of external connection, Operational Releases are normally tested against Simulators. The SIT Simulator is used by the Switching Operator to act as GRDA, ERDA, Energy Suppliers, Supplier Agents, EES, GES, DCCSM, and ECOS.

#### 3.2.2.2. UIT-A

The primary purpose of the UIT-A environment is to support External Testing, as described in the CSS External Testing Guidance document. External Testing is:

- The qualification of new market entrants to connect to the Production CSS; or
- Additional testing for Maintenance of Qualification of REC Parties to connect to the Production CSS where required by REC Performance Assurance.
- External Testing is conducted with Simulators acting as all parties other than the market participant under test.

### 3.2.3. B-Stream environments

The CSS B-Stream environments are provided for the testing of Major Releases and these environments are therefore usually one or more releases in advance of Production levels.

#### 3.2.3.1. SIT-B

The primary purpose of the SIT-B CSS environment is to allow the integration testing of Major Releases of the CSS (Registration Service, Address Service, and Reporting Service) and any REC Party systems within the scope of the release.

The SIT-B CSS environment is capable of external connection. The components of the release, and its impact on REC parties, will determine whether true integration testing of the release is required. For any REC party not required to be a participant in the testing of the Major Release, the SIT Simulator will be used by the Switching Operator. This may include any of GRDA, ERDA, Energy Suppliers, Supplier Agents, EES, GES, DCCSM, and ECOS.

#### 3.2.3.2. UIT-B

The primary purpose of the UIT-B CSS environment is to allow the integration testing of Major Releases of the CSS and any REC party systems within the scope of the release, in particular where those REC parties are CSS Users other than Switching Data Service Providers.

The UIT-B CSS environment is capable of external connection. The components of the release, and its impact on REC parties, will determine whether true integration testing of the release is required. For any REC party not required to be a participant in the testing of the Major Release, the SIT Simulator will be used by the Switching Operator. This may include any of GRDA, ERDA, Energy Suppliers, Supplier Agents, EES, GES, DCCSM, and ECOS.

### 3.2.4. PreProd

The primary purpose of the PreProd CSS environment for the execution of non-functional testing, in particular testing that the CSS can meet its non-functional requirements. Non-functional testing of CSS is not routinely undertaken as part of standard delivery of Operational and Major Releases. Therefore, the PreProd environment is not on the standard path to live for Operational or Major Releases. All testing in PreProd is conducted against Simulators.

### 3.2.5. Production

All REC Parties will have a Production environment.

## 3.3. Path to Live

This section describes the standard Path to Live for Operational and Major Releases.

### 3.3.1. Operational Releases

Operational Releases are tested in the A-Stream prior to deployment to Production. Path to Live is PIT, SIT-A and Prod. The Operational Release is then deployed to other environments, including the B-Stream and PreProd where the code is merged with any in-flight major releases.

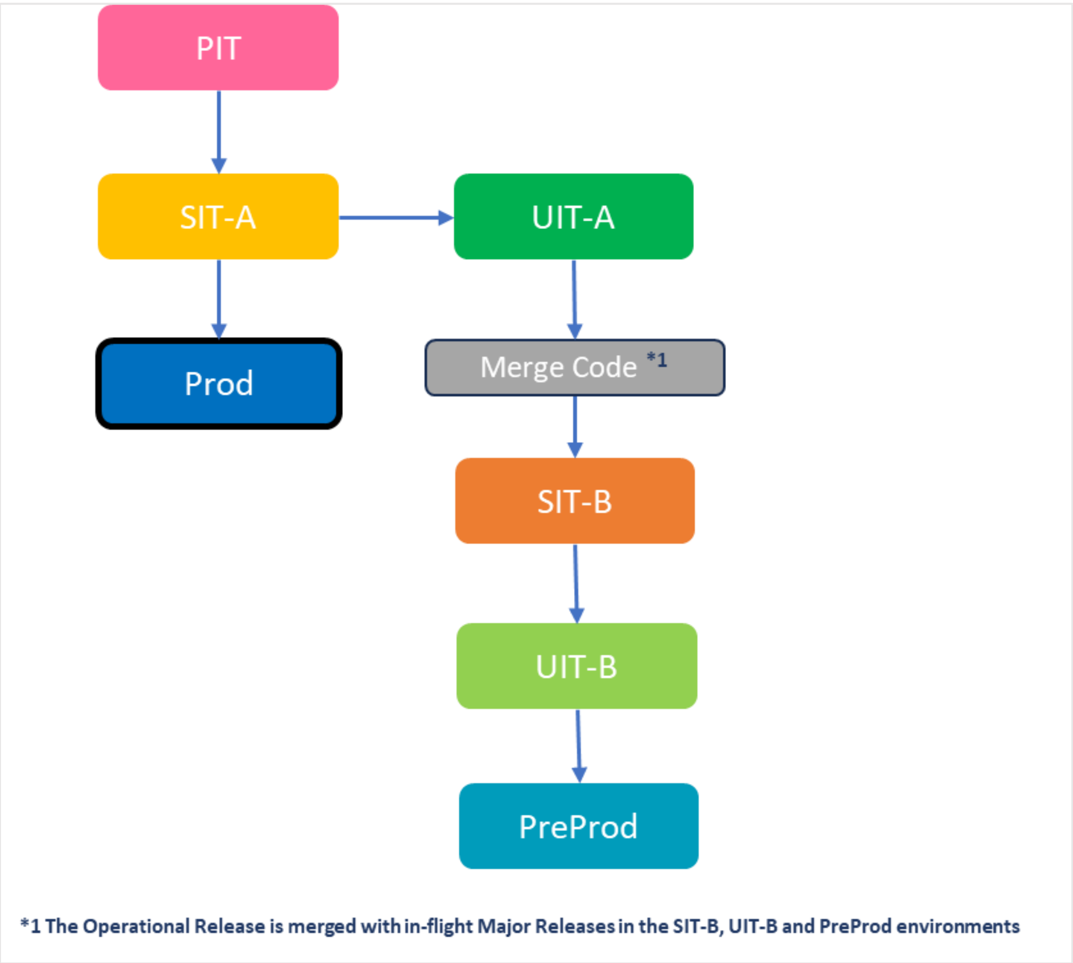

PIT
SIT-A
UIT-A
Prod
Merge Code *1
SIT-B
UIT-B
PreProd
*1 The Operational Release is merged with in-flight Major Releases in the SIT-B, UIT-B and PreProd environments

**Figure 2 Path to Live for Operational Releases**

### 3.3.2. Major Releases

Major Releases are tested in the B-Stream, and then promoted through the A-Stream prior to deployment to Production. PreProd is not normally on the Path to Live for CSS Releases.

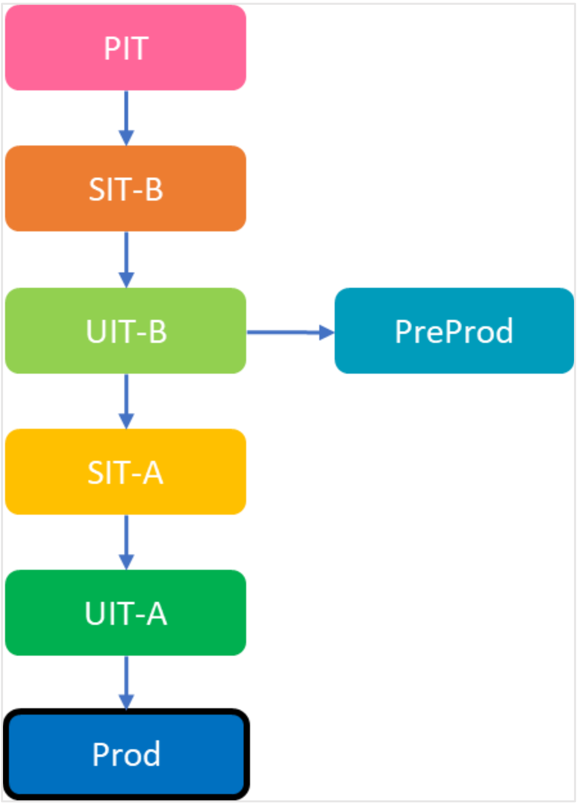


**Figure 3 Path to Live for Major Releases**

## 3.4. Test Data Management

Environment usage is closely linked with the approach to Test Data Management described in the Test Data Management Plan.

## 3.5. Security Management

The SIT, UIT and PreProd environments may all contain copies of Production data and are therefore subject to the security principles described in the Schedule 25 – Central Switching Service.

# 4. Tooling

This section describes the tools put in place by the Switching Operator to support testing within the environments.

## 4.1. Simulators

The section describes the use of the simulators and how they are be used. This will include:

- CSS PIT Simulator
- SIT Simulator
- Counterparty Simulator
- Performance Test Tool

The default configuration for all environments is to use simulators, which focus on supporting functional testing.

It is expected that the use of physical connections will be a low occurrence in A-Stream. The exception to this is for market participants conducting External Testing in UIT-A. In the B-Stream, the configuration will be driven by the requirements of the Major Release under test.

A simulated and physical connection for a Party cannot be supported simultaneously in any environment.

### 4.1.1. CSS PIT Simulator

The CSS PIT Simulator is provided to CSS Users to simulate interactions with the CSS in their local (non-connected) environments. It is expected that CSS users utilise the CSS PIT Simulator in their Development environments to test their internal changes, rather than connecting to a test environment, unless driven by a REC Change Proposal.

The CSS PIT Simulator will simulate the actions of CSS by responding to received messages and replying to messages replicating the CSS functionality.

The CSS PIT Simulator enables parties to develop their applications and ensure that the messages they send to CSS conform to the CSS Physical Interface Design. This allows them to receive messages into their systems in the format that CSS will provide.

The CSS PIT Simulator is available for both Windows and Linux platforms for all parties via GitHub at: https://github.com/SwitchingOne/CSS-Simulator/
The supporting User Guide and Configuration guide can also be found in this location.

CSS Users that require access to this GitHub location should email the following, providing the email address of the user requiring access: DCCSwitchingSI@netcompany.com

The nominated user will be able to download, install and configure within their own environment.

### 4.1.2. SIT Simulator

The SIT Simulator is provided by the Switching Operator to simulate CSS User interactions in the SIT and UIT environments. Multiple instances of this simulator will be deployed and managed by the Switching Operator to replicate different types of users. The SIT Simulator is hosted within the Switching Operator's Microsoft Azure environment, to enable centralised administration.

The Switching Operator may use the SIT Simulator in the SIT and UIT environments to simulate:

- Gaining and losing Energy Suppliers – this will facilitate execution of test scenarios that are initiated by the gaining Energy Supplier and providing the 'intelligence' to respond as a losing Supplier as required.
- Switching Data Service Providers to ensure CSS generates the correct messages for these systems, and to allow the initiation of test scenarios initiated by the SDSPs.
- Supplier Agents e.g., DA, DC, MEM, Shippers, MAP, to ensure CSS generates the correct messages for these systems.

The Switching Operator is responsible for configuring the SIT Simulator Tool to send messages back to the CSS environment. Direct access to the SIT Simulator is not required for CSS Users.

### 4.1.3. Counterparty Simulator

The Counterparty Simulator may be used as a second energy supplier i.e., counterparty, when testing switches as a gaining or losing supplier. Alternatively, depending on the scale of testing, it may be decided that the Switching Operator will perform the action of the Counterparty manually.

The Counterparty Simulator is hosted within the Switching Operator's Microsoft Azure environment to enable centralised administration and is available via GitHub (https://github.com/SwitchingOne/CSS-Simulator/).

Use of the Counterparty Simulator will normally only be required for External Testing, where a market participant is required to undertake tests as both a gaining and losing supplier as part of their qualification to connect to CSS.

Parties will interact only at the commencement of the messaging sequence, doing so via a specified endpoint. This endpoint will be provided by the System Operator as part of External Testing preparation.

### 4.1.4. Performance Test Tool

The Performance Test Tool is used by the Switching Operator to demonstrate that the CSS meets its non-functional requirements. Non-functional testing of CSS is not routinely undertaken as part of standard delivery of Operational and Major Releases.

Where it is determined that a change could impact the overall non-functional requirements, as a result of either a Major or Operational Release, the Switching Operator may execute non-functional tests of CSS using the Performance Test Tool.

The Performance Test Tool has the following capabilities:

- The injection of a representative message load that meets the non-functional requirements (switches and initial registrations).
- The ability to mock all external end points to CSS and return the expected responses.
- The ability to reconcile all input and output messages to prove correct system behaviour.
- The Performance Test Tool supports Peak (and Peak of Peaks) Gate Closure tests as well as Soak Test and Resilience Tests.

## 4.2. Test Data Tool

The Test Data Tool is used by the Switching Operator to manage the identification, allocation, and distribution of Test Data to REC Parties. Further information on the Test Data Tool can be found in the Test Data Management Plan.

# 5. Environment Connectivity

## 5.1. Physical Connections

It is expected that CSS Users make a physical connection to a test CSS environment only when requested to do so, as a result of a REC Change Proposal or to undertake External Testing. Making a physical connection and undertaking testing without this authorisation may:

- Cause further misalignment of data between CSS and other CSS User systems;
- Impact ongoing testing within the CSS environments where a simulated connection is assumed to be in place.

There is no provision for the support of 'sandbox' type testing for CSS Users.

### 5.1.1. Requesting a Physical Connection

If requested to set up a physical connection, the CSS User should: raise a Service Request as per the **DCC Test Simulator High Level Process.** The Service Request should be raised 4 weeks ahead of any intention to connect physically to an environment and should contain with the following information:

- The CSS environment to which they wish to physically connect.
- Start and end dates for the physical connection.
- Purpose of connection.

The Switching Operator will then engage with the CSS Provider to allow for the CSS User to connect physically to the environment, including the following steps:

- Webhook Registration
- Connectivity Testing (if required)

Once the date for connection expires, the Switching Operator will reconnect the end point back to the relevant simulator.

If the requested party requires to either disconnect earlier than the supplied date, or to extend the physically connected period they must inform the Switching Operator through the same Service Request process.

### 5.1.2. Webhook Registration

Where a CSS user requires to receive events from physical CSS environment, they will need to register to the events which are allocated to them as set out in the Schedule 25 – Central Switching Service. The registration of webhooks should only be completed in a controlled manner through the Switching Operator, to ensure that any other testing that is occurring is not interfered with.

## 5.2. Connectivity Methods

This section provides background on the types of connection in place for connectivity to the CSS environments.

### 5.2.1. ExpressRoute

An ExpressRoute extends on-premises networks into the Microsoft cloud over a private connection facilitated by a connectivity provider. ExpressRoute connections do not go over the public Internet. This allows ExpressRoute connections to offer more reliability, faster speeds, consistent latencies, and higher security than typical connections over the Internet.

However, this necessitates a direct network connection between a party's estate and the CSS and therefore technology is implemented by both parties and maintenance of that connection needs to be supported.

To allow for the connectivity between the DCC Smart Metering Environment and CSS a dual ExpressRoute is provisioned to connect to the Gamma Network.

The connection to CSS from the environment utilises the Smart Metering Key Infrastructure (SMKI) and DCC Key Infrastructure (DCCKI) to provide authenticity and where applicable, integrity and non-repudiation of all communications with the DSP Interfaces.

### 5.2.2. Internet

The default connection method for parties to connect to CSS is the internet. Where a party has decided to use this connection method, the bandwidth available would be determined by the purchased speed and restrictions set by their ISP and is the responsibility of the party, and chosen provider, to maintain.

The maintenance of the network connection would not be part of the connecting parties' responsibility and no additional technology would be required to implement or maintain the connection, although parties should ensure that the service level agreement provided by the internet service provider affords a level of service and support required to maintain connectivity to a sufficient level.

It is recommended - but not mandated - that any party using the Internet as the connectivity method to the CSS also procures a secondary internet connection as a backup.

## 5.3. Networking Considerations

### 5.3.1. Domain Name System (DNS)

The exposed service name must be resolvable for both systems connecting over the internet and those services that are connecting over the ExpressRoute gateway.

To allow for connectivity back to the connecting parties' Webhook, CSS will also need to resolve the service name of the connecting party on either connecting party.

Where both CSS and connecting parties connect over the internet, the service name and boundary IP address should be placed into the Internet DNS zone by the owning party. All systems connecting over the internet will then need to be able to resolve this address from the Internet DNS server.

Where systems connect over the ExpressRoute Gateway, connecting parties will be given access over Port 53 to the DNS server within the CSS solution to resolve the IP address of the service name. The CSS will then also query the connecting parties' DNS server to resolve their corresponding service name.

### 5.3.2. Firewalls

Where firewalls are required to be opened between a connecting party and the CSS, it is the responsibly of that party to ensure the opening of any firewalls, in line with the details captured within the connectivity details for the relevant testing conducted.

### 5.3.3. Routing

Parties are responsible for any routing to allow their systems to connect to the API Management Address being enacted in line with the details captured within the connectivity details for the relevant Testing conducted.

### 5.3.4. Public Key Infrastructure (PKI)

CSS provides an application interface to enable the exchange of registration and address information between the CSS and CSS Service Users. There are various network access options to CSS, details of which can be found in the Schedule 25 – Central Switching Service. . Common to all network access options is the requirement for two specific elements of security to secure messages flowing across the system:

- Transport Layer Security (TLS 1.3 where possible TLS 1.2 as a minimum); and
- JSON message signing.

Both TLS and JWS are cryptographic mechanisms that use public-key cryptography to implement specific security services. Implementing TLS and JWS relies on Public Key Infrastructure (PKI) – the technology that supports the management of public keys.

## 5.4. Connectivity Testing

Where a party needs to connect physically to an environment this will normally be completed as a Business as Usual (BAU) process and will not be subject to a connectivity test. However, the REC Code Manager may require a Connectivity Test to be conducted, for example due to a major change such as re-platforming of a parties' technology stack, or as part of External Testing for a new market participant. In the event that a Connectivity Test is required, then the following steps may be undertaken and reported back as part of the formal change process.

### 5.4.1. Phase 1 - Initial Environment Configuration

In the initial environment Connectivity Test, the following must be completed to allow for the integration of Parties in the following phases:

- DNS Service Names have been added to Internet DNS service.
- DNS Service have been added to the providing DNS service for Private Networks.
- Both connecting parties have provisioned and implemented sufficient technology to support Synchronous messaging including any network security.
- The CSS and connecting parties have installed the TLS communication certificate for the CSS Service Name.
- Firewalls opened to allow connectivity to the specified Service Names and Ports as documented.

### 5.4.2. Phase 2 - Network Connectivity

The following steps must be completed to ensure that communication between parties has been successfully put in place.

#### 5.4.2.1. DNS check

This will ensure that DNS records have been correctly entered and are resolvable from all environments:

- NSlookup from the CSS to the Internet DNS server for all Service Names to be connected over the Internet.
- NSlookup from the CSS to the appropriate Private DNS servers for all Service Names.
- NSlookup from all connecting parties for the appropriate Test Phase's CSS Service Name.

#### 5.4.2.2. CSS Certificate check

This will ensure that the CSS TLS Certificate is installed, and the connecting party can connect to the service: Connection on 443 from all connecting parties for the appropriate Test Phase's CSS Service Name.

### 5.4.3. Phase 3 – TLS and PKI Certificate and Webhook Registration

This test is to confirm that the test participant has obtained TLS and PKI certificates, installed on their system, and successfully registered webhooks with CSS.

### 5.4.4. Phase 4 – Data and Message Exchange with CSS

#### 5.4.4.1. Phase 4 - Message Exchange

This will ensure that all parties can exchange messages:

- Both connecting parties have provisioned and implemented sufficient technology to support Asynchronous messaging including any network security.
- The connecting party has followed the registration details set out in the PKI work instructions including the loading of the private certificate locally and supplying the public certificate to the CSS, as defined in the Schedule 25 – Central Switching Service.
- The CSS Provider has set-up appropriate users and roles and associated the users with the supplied certificate.
- The connecting party registers their webhook with CSS.
- The connecting party sends a message to the CSS and receives a successful Synchronous and Asynchronous message in response.

# 6. Management and Control

## 6.1. Release Management

This section describes the processes put in place to ensure that environments are updated with the appropriate version of code and configuration prior to the use of any environment.

The CSS Provider will provision and support 2 separate testing streams, Stream-A and Stream-B to support both Operational and Major changes. These environments will be updated with the latest production code base and configuration plus any relevant release under test.

Changes to software can be necessitated by Defects or from Change Proposals to any system under the responsibility of any of the CSS Provider, the Switching Data Service Providers or CSS Users. The deployment and path to live of these upgrades will fall under the responsibility of that party. However, where these upgrades change the configuration of the interfaces connected to the CSS, they should inform REC Performance Assurance to assess whether additional testing for Maintenance of Qualification is required.

It is expected that all REC parties operate robust industry standard change and release management processes, ensuring that the Switching ecosystem is not placed at risk.

The Switching Operator will maintain a Maintenance and Major Change release schedule on the Switching Portal, thus providing clear time in the plan for when software can be released by parties in a co-ordinated manner.

## 6.2. Environment Planning

The Switching Operator will maintain an overall plan showing the use of the CSS environments. This will allow the Switching Operator to manage contention between environments and manage the needs of parties where required. It will also allow the Switching Operators, as part of the REC Change control process, to provide a view on environment usage and when Major Releases can be tested and promoted into Production.

The Switching Operator will also ensure that when planning the usage of the environments, the plans and schedules for Smart Metering environments are also taken into consideration.

The Switching Operator will create and maintain an overall Environments Plan showing the future schedule of releases and changes, to assist in both planning and reporting on potential contentions of use.

## 6.3. Environment Reporting

The Switching Operator can provide relevant reporting on environment scheduling and availability.

**----END OF DOCUMENT----**