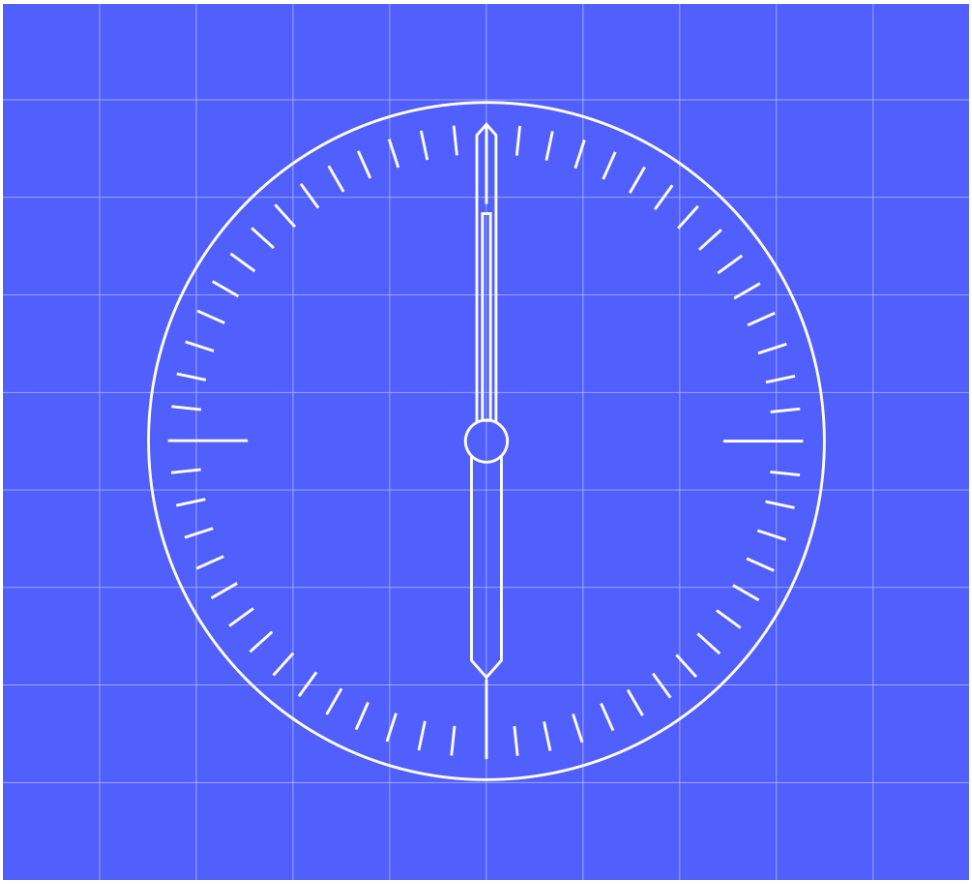
MHHS PROGRAMME
Industry-led, Elexon facilitated

# Interface Code of Connection



| Document owner | Document number | Version | Status: | Date |
|---|---|---|---|---|
| **MHHS DAG** | **MHHS-DEL1197** | **Version 1.3** | **Approved** | **22nd November 2023** |

Deleted: 1

Deleted: 15

Deleted: September

# Contents

## Table of Figures



## Table List

## Change Record

| Date | Author | Version | Change Detail |
|---|---|---|---|
| 4th July 2022 | MHHS Design Team | 0.1 | Initial Draft |
| 11th July 2022 | MHHS Design Team | 0.2 | Update post review with design team |
| 27th March 2023 | MHHS Design Team | 0.3 | Updated to incorporate DIP service provider input |
| 13th April 2023 | MHHS Design Team | 0.4 | Updated to incorporate feedback from Avanade |
| 02nd May 2023 | MHHS Design Team | 0.5 | Updated to incorporate feedback from internal review |
| 06th June 2023 | MHHS Design Team | 0.6 | Added section 3.5.2.6, updated tables in sections 8.1.3 and 8.1.4 to incorporate feedback from industry consultation. |
| 19th June 2023 | KG | 0.7 | Update – All sections post industry consultation |
| 03rd July 2023 | KG | 0.8 | Update – All sections post industry consultation |
| 12th July 2023 | KG | 1.0 | Approved |
| 15th September 2023 | KG | 1.1 | Updated references<br>Updated PKI roles<br>Added Section 8.8 API Keys<br>Added Section 12 – Appendix A |
| | KG | 1.2 | Corrected spelling and grammar<br>Updated section 8.3, 8.4, 8.5 and all reference to PKI roles. |
| 07th November 2023 | KG | 1.3 | Updated post SDWG review, added PKI role mapping to section 5.4.3 |

## Reviewers

| Reviewer | Role |
|---|---|
| | |
| | |

## References

| Document/Link | Publisher | Published | Additional Information |
|---|---|---|---|
| **Cyber Assessment Framework (CAF)** | NCSC | V3.0 | https://www.ncsc.gov.uk/collection/caf |
| **[1] MHHSP-DES138- Interface Catalogue** | MHHS | V 5.1 | June 2023 |
| **[2] MHHS-E2E001 - End-to-End Solution Architecture** | MHHS | V3.1a | March 2023 |

| | | | |
|---|---|---|---|
| **[3] MHHS-DIP002 Functional & Non-Functional Requirements** | MHHS | V2.2 | August 2022 |
| **[4] MHHS-DEL1364 – DIP PKI Policy** | MHHS | V1.0 | May 2023 |
| **[5] MHHS-DIP005 - End to End Security requirements** | MHHS | V1.4 | September 2022 |
| **[6] MHHS-DIP004 - Cyber Security Connection Guidance** | MHHS | V1.2 | November 2022 |
| **[7] MHHS-DEL-1387- DIP PKI Certificate Profiles** | MHHS | V1.0 | July 2023 |

## Terminology

| Term | Description |
|---|---|
| AGS | API Gateway Service |
| API | Application Programming Interface |
| BSC | Balancing and Settlements Code |
| CRL | Certificate Revocation List |
| DCA | DIP Certificate Authority |
| DCP | DIP Connection Provider – Non Market Participants connecting to the such as; 3rd Parties, Software providers, Academia etc. |
| DIP | Data Integration Platform |
| DIP Connection Provider | 3rd Party Organisation providing services on behalf of a Market Participant. |
| DIP Manager | The Function responsible for the enduring service / operational running of the DIP and supporting services. |
| DIP Service User | A registered user of the DIP Portal (Market Participants & 3rd Party organisations) |
| DIP Service User Administrator | A DIP Service User who has been appointed the role of Administrator within the DIP for their registered organisation. |
| DoS | Denial of Service |
| DPIA | Data Protection Impact Assessment |
| EDA | Event Driven Architecture |
| HTTP | Hyper Text Transfer Protocol |
| HTTPs | Hyper Text Transfer Protocol Secure |
| IP | Internet Protocol |
| JSOn | JavaScript Object Notation |
| MHHS | Market Wide Half Hourly Settlements |
| MP | Market Participant |
| OWASP | Open Web Application Security Project |
| PKI | Public Key Infrastructure |
| REC | Retail Energy Code |
| REST | REpresentational State Transfer |
| SIT | System Integration testing |
| TLS | Transport Layer Security |
| UIT | Unit Integration Testing |
| UKAS | United Kingdom Accreditation Service |
| URL | Uniform Resource Locator |

# 1 Introduction

## 1.1 Purpose

This is the Code of Connection document for the DIP Service Interface, defining the interface usage requirements and responsibilities for Market Participants to securely exchange information. It complements the **[2] MHHS-E2E001 - End-to-End Solution Architecture** document, prepared for connecting participants by the MHHS Programme.

## 1.2 Scope

The scope of this Code of Connection is to define the operational context and constraints in which the DIP Interface operates to provide consistent performance including:

- Interaction with multiple environments, i.e., SIT, UIT, PRE-PRODUCTION, PRODUCTION;
- Operational and Security responsibilities.

The Code of Connection will identify configurable parameters that will be reviewed periodically to cater for changing demand and capacity forecasts. This code of connection also defines the operational procedures required by Market Participants and DIP Connection Providers to securely exchange metering and consumption information with the DIP using Public Key Infrastructure (PKI) Certificates.

## 1.3 Document Structure

This document is structured as follows;

Section 1 **Introduction;** this section;

Section 2 **DIP Security Requirements** outlines the high-level security requirements and responsibilities;

Section 3 **Public Key Infrastructure** provides an explanation of, and guidance on the use and management of Public Key Certificates and associated keys;

Section 4 **DIP Interface Description** provides an overview of the interface;

Section 5 **Participant Engagement** details the processes to be followed and information to be provided by Parties when registering with the DIP service and requesting DIP certificates from the DIP Certificate Authority (DCA);

Section 6 **Managing DIP Certificates** describes the processes and procedures for distributing key cryptographic key material, including CSRs and Certificates;

Section 7 **Transport Layer Security (mTLS)** details the processes for generation, distribution, use and management of TLS keys and Certificates;

Section 8 **Message Signing** details the processes for generation, distribution, use and management of JSON message signing keys and Certificates;

Section 9 **DIP User Portal** provides an overview of the user portal;

Section 10 **Non-Functional Considerations** contains a description of the volume profile relating to capacity, bandwidth, and message rates over periods of time measurement, and applicable throughput controls. It also introduces resilience and system outage considerations.

Section 11 **Frequently Asked Questions**

Section 12 **Appendix A – Supplementary Information**

# 2 DIP Security Requirements

The security requirements for the DIP Messaging Interface are intended to minimise the risks to the DIP infrastructure and its users.

The DIP itself will be certified against ISO/IEC 27001[1] and is subject to certification using a UKAS-certified auditing body.

Therefore, to protect the DIP and the wider ecosystem it is important for each connecting organisation to reliably identify and appropriately manage all risks associated with the infrastructure supporting their interface. This section details the security responsibilities, best practices and requirements that should be applied to the DIP Interface.

## 2.1 Information Assurance Responsibilities

DIP Services Users (Market Participants and DIP Connection Providers) must be prepared to provide assurance of compliance with the DIP Manager as part of the on-boarding process and as requested as part of any Audit that may be conducted by the DIP Manager (or their duly appointed agent). Governance and processes covering the DIP Service User assurance will be contained in the DIP Subsidiary Documents being prepared under Issue 101.

The level of evidence requested will vary depending on the type of DIP Service User. For example, it is expected that Market Participants and DIP Connection Providers (collectively referred to as DIP Service Users in this document), will already follow best practice standards and frameworks for information security. Compliance evidence can include, but is not limited to, any or all the following:

- Evidence of the application of security best practice as prescribed by NCSC CAF – if not ISO/IEC 27001 or equivalent accredited – covering the interconnected systems for the MHHS interface;
- Statements of Applicability demonstrating that all applicable ISO/IEC 27001 controls have been applied, where appropriate (where an organisation is ISO 27001 certified);
- Evidence of compliance with other recognised security frameworks such as Cyber Essentials or Cyber Essentials Plus;
- Written confirmation that the Code of Connection (this document) has been read and understood.

## 2.2 Determination of Security Standards

Risks to information assets and services are normally identified by determining:

- The vulnerabilities within or around the system or service (including interfaces to it);
- The threats that can exploit those vulnerabilities;
- The impact of any resulting compromises.

To protect the DIP ecosystem, all DIP Service Users will be expected to carry out regular risk assessments of their systems and interfaces, even if not contractually obliged to do so. Details of how to conduct risk assessments can be found in well-established assessment models, such as ISO/IEC 27005 or HMG's Information Assurance Standards 1 & 2. (See **[6] HHS D004 - Cyber Security Connection Guidance for further information**).

## 2.3 Security Requirements

The security requirements in this Code of Connection are based on NCSC CAF recommendations that organisations adopt a recognised baseline of security controls such as those prescribed in ISO 27001 (or equivalent, e.g., NIST 800-53), supplemented with a set of enhanced level controls upon which the DIP Service Users (Market Participants and DIP Connection Providers) are reliant to protect the DIP Interface. These enhanced level controls require that DIP Service Users should:

[1] It is not a requirement for DIP Service Users to be certified against ISO/IEC 27001.

- Ensure they have a valid Data Protection Impact Assessment (DPIA) for the data they send or receive from the DIP.
- Perform a self-assessment – Interface Code of Connection compliance status annually. See **[6] MHHS DIP004 – Cyber Security Code of Connection** for further details of requirements.
- Retain all audit logs of basic user activities (e.g., logon, logoff, failed attempts) and security events for all information systems and services that interact with the DIP, within legal constraints, for a minimum of 3 months of live data and 12 month archived.
- Have a logical network schematic of the information systems and services in scope that interact with the DIP, and include:
    - Services and functionality;
    - Gateway/boundaries functionality.
- Ensure that the edge routers and switches in the Data Centres / Cloud are physically / logically secured with direct access only being granted to staff who have a demonstrable and approved need for access;
- Each organisation agrees that it will use its own time source for time synchronisation with its organisation and that time does not need to be synchronised across organisational boundaries. In the event of artefacts such as incidents and logs being required to be examined forensically, it is agreed to accept 'DIP defined time' (time.windows.com) as the master for any incidents and in log files.

The messaging security requirements and how and where they are satisfied can be seen in the table below:

| Objective | Mechanism | Provided By |
|---|---|---|
| Confidentiality | Encryption in transit | Transport Layer Security |
| Data Integrity | Message Authentication Code (MAC) | Transport Layer Security |
| | Hash Function | Message Signing |
| | Digital Signature | Message Signing |
| Data Origin Authentication | Message Authentication Code (MAC) | Transport Layer Security |
| | Digital Signature | Message Signing |
| Non-Repudiation[2] | Digital Signature | Message Signing |

Table 1- Security Services & Mechanisms.

Hence, messages sent to and received from the DIP will be secured by:

- Mutual TLS (mTLS) for transport layer security and authentication
    - Transport Layer Security TLS 1.3 where possible TLS 1.2 as a minimum
- Message signing for non-repudiation
- mTLS for strong security authentication / authorization at the API.
- API key (ingest only) for rate limiting, usage tracking.

[2] Non-Repudiation is not a specific security requirement but can be realised using a digital signature.

The implementation of both Transport Layer Security and JSON Message Signing require the use of Public-Key Cryptography which depends on the use of Digital Certificates. This is described below.

# 3 Public Key Infrastructure

This section provides an overview of Public Key Infrastructure (PKI) and details the PKI requirements and solutions as they relate to the DIP. It explains the Digital Certificate requirements for Parties and the interaction between the PKI components and processes, enabling Parties to register, request, install and manage the certificates necessary to secure the DIP ecosystem.

## 3.1 Overview

Implementing the mechanisms required to secure the flow of DIP messages with TLS and digital signatures requires Digital Certificates. The Digital Certificates are used to secure communications using public key cryptography schemes.

A Public Key Infrastructure (PKI) is a collection of policies, procedures and technology needed to manage Digital Certificates in a public key cryptography scheme.

## 3.2 Digital Certificates

The objective of a public key cryptography scheme is trust. A Digital Certificate is an electronic signature from a trusted third party that guarantees the validity and authenticity of a public key.

The Digital Certificate binds an entity, being an institution, a person, a computer program, a web address etc., to its public key.

The most common type of Certificate is the one compliant with the X.509 standard, which allows the encoding of a party's identifying details in its structure.

The most common trust model used to validate the validity and authenticity of a public key is a Certificate Authority and this is the trust model used in the DIP System PKI.

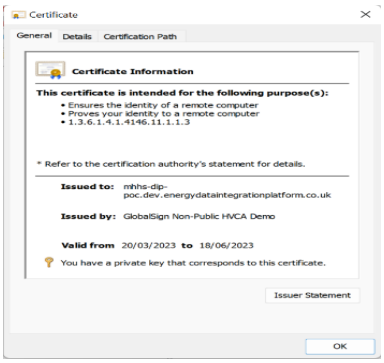



Figure 1 - Digital Certificate.

The DCA issues certificates compliant to the X.509 standard.

## 3.3 What is meant by a Public Key?

Authentication and message integrity are important concepts in secure communications and are vital security services for Market-wide Half-Hourly Settlement.

Authentication requires that entities who exchange messages are assured of the identity that created a specific message. For a message to have “integrity” it should not have been modified during transmission (or at least if it has, you should have a way to find out).

For example, you will want to be sure you are communicating with a real third party (such as the DIP) rather than an impersonator. Or if you have received a message from the DIP, you will want to be sure that it has not been tampered with by anyone else during transmission.

Traditional authentication mechanisms rely on digital signature schemes that, as the name suggests, allow a party to digitally sign its messages. Digital signatures also provide guarantees as to the integrity of the digitally signed message.

Digital signature schemes require the DIP and each DIP Service User to hold two cryptographically connected keys: a public key that is made widely available and acts as authentication anchor, and a private key that is used to produce digital signatures on messages. Recipients of digitally signed messages can verify the origin and integrity of a received message by verifying that the attached signature is valid using the public key of the assumed sender.

The unique mathematical relationship between the keys (which is called a trapdoor one-way function) is such that the private key can be used to produce a signature on a message that only the corresponding public key can verify.

It is the public key that is embedded in a Digital Certificate, and the security of public key cryptography relies on ensuring that private keys are kept secure.

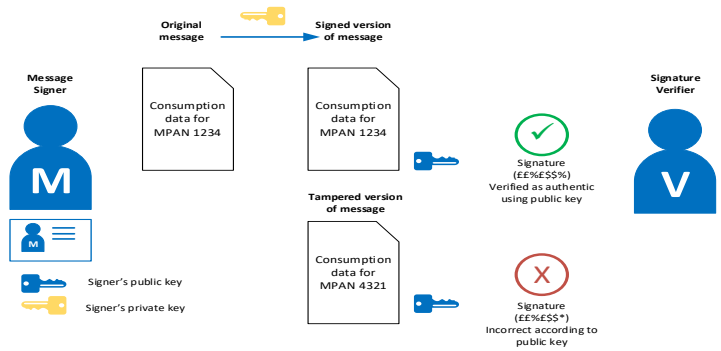



Figure 2- Signing and Verifying Signatures.

## 3.4 Certificate Authority

### 3.4.1 Certificate Policy

The Certificate Authority (or CA - synonymous with an Issuing Authority – or IA) is a trusted third party specialised in issuing and managing Digital Certificates. The DIP Certificate Authority (DCA) is managed by GlobalSign. The DIP adheres to the DIP Certificate Policy document which is a named set of rules that indicates the applicability of a Certificate to a particular community and/or class of application with common security requirements, i.e., the DIP Service. The DIP PKI Certificate Policy document [4] **MHHS DEL1210 - DIP PKI Policy** is structured in accordance with the guidelines in IETF RFC 3647, with appropriate modifications, deletions, and references to other documentation as appropriate.

The DCA (GlobalSign) has been integrated into the DIP User Portal to issue digital Certificates to authorised DIP Service Users. These Certificates are digitally signed by the DCA (GlobalSign) and bind DIP Service Users with their public keys. As a result, if you trust the DCA (and know its public key), you can trust that the specific party’s public key included in the Certificate is genuine.

### 3.4.2 Certificate Usage

Certificate usage is defined by a Certificate Profile. Certificate Profiles are approved and issued by GlobalSign. The DCA will ensure that DIP Certificates are issued only:

1) To Eligible Subscribers; and
2) Are used only for the purposes of:
   a) The creation, sending, receiving, and processing of communication within the DIP environments.
      i) Symmetric key generation (Digital Signature, Key Agreement);
      ii) TLS Communication (Digital Signature, Key Agreement, TLS Web Client Authentication, TLS Web Server Authentication); and
      iii) Authentication and Non-Repudiation (Digital Signature, Non-Repudiation, Key Encipherment, Data Encipherment, Key Agreement, TLS Web Client Authentication. TLS Web Server Authentication).

Certificates are publicly available, as they do not include either the DIP Service User or the DCA's private keys. As such they can be used as anchors of trust for authenticating messages originating from different parties.

CAs also have a Certificate, which they make widely available. This allows the consumers of identities issued by a given CA to verify them by checking that the Certificate could only have been generated by the holder of the corresponding private key (the CA).

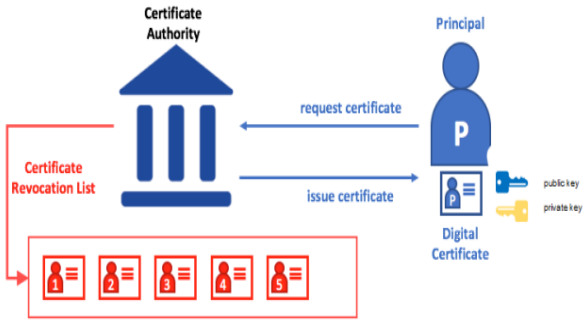



Figure 3 - Certificate Authority.

## 3.5 Determining Trust and Validity

To verify whether a Digital Certificate for an end-entity can be trusted and is valid requires the party doing the validation (or Relying Party) to validate as follows:

### 3.5.1 Chain of Trust

The standard trust model used in the DIP holds that if the Digital Certificate is digitally signed by the DCA, then it can be trusted by the relying party (see section 5.7 of this Code of Connection). To do this the relying party needs to verify the DCAs Digital Signature and that certain contents of the Digital Certificate are also correct (for example that the Certificate has not expired and that the key usage is for Digital Signatures).

To verify the DCAs Digital Signature, the relying party must have access to the DCAs public key, which itself will be contained in another Certificate signed by a higher-level (Intermediate) or top-level (Root) CA. All Certificates that comprise the 'Certificate Chain' will be required to complete the validation process (also known as 'walking the chain'). All Certificates in the Certificate Chain will be published in a repository accessible by the DIP Community[3] by the DIP Service Provider.

[3] This will be hosted by the DIP Service Provider and accessed through a User Portal.

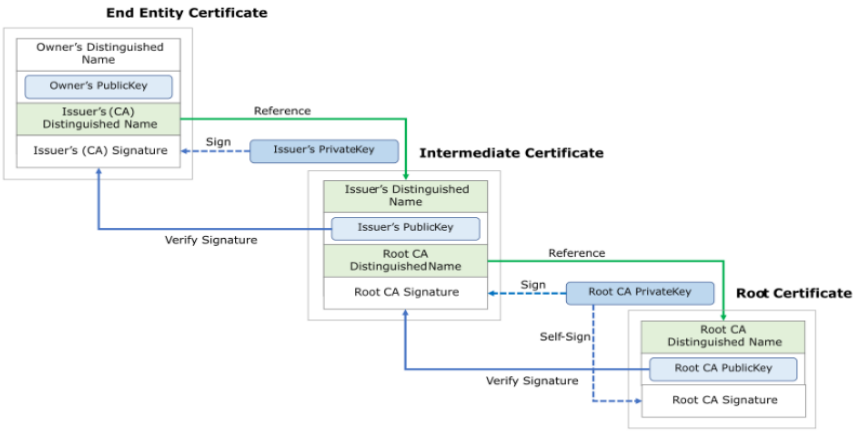



Figure 4 - Certificate Chain.

### 3.5.2 Certificate Revocation Lists

#### 3.5.2.1 *Overview*

A Certificate Revocation List (CRL) is a list of references to Certificates that a CA knows to be revoked for one reason or another (much like a list of stolen credit cards).

When any party (known as a relying party) wants to verify another party's identity, it first checks the issuing CA's CRL to make sure that the Certificate has not been revoked. A relying party does not have to check the CRL, but if they do not, they run the risk of accepting a compromised identity.

A DIP Service User shall ensure that it submits a Certificate Revocation Request in relation to a Certificate in accordance with the DIP Manager:

- Immediately upon becoming aware that the Certificate has been compromised, or is suspected of having been compromised; or
- Immediately upon ceasing to be an Eligible Subscriber in respect of that Certificate.

#### 3.5.2.2 *Circumstances for Revocation*

A Certificate must be revoked:

- When any of the information in the Certificate is known or suspected to be inaccurate;
- Upon suspected or known compromise of the Private Key associated with the Certificate;
- Upon suspected or known compromise of the media holding the Private Key associated with the Certificate.

#### 3.5.2.3 *Who can request Revocation?*

See section 6Managing DIP Certificates

#### 3.5.2.4 *Procedure for Revocation Request*

See section 6 Managing DIP Certificates

#### 3.5.2.5 *Revocation Checking Requirement for Relying Parties*

Relying Parties are those entities that are using a Certificate to authenticate another Certificate Subscriber named in the Certificate. In the context of the DIP, every Certificate Subscriber is also a Relying Party at some point, for example when they are setting up secure channels with TLS or verifying the Digital Signatures on received messages.

All Relying Parties must check the Revocation Status for each Certificate upon which they rely. This is achieved using the Certificate Revocation List that is published at the location defined in the CRL Distribution Point field in every Certificate.

Additional information about CRLs can be found in the **[4] MHHS DEL1210 – DIP PKI Policy.**

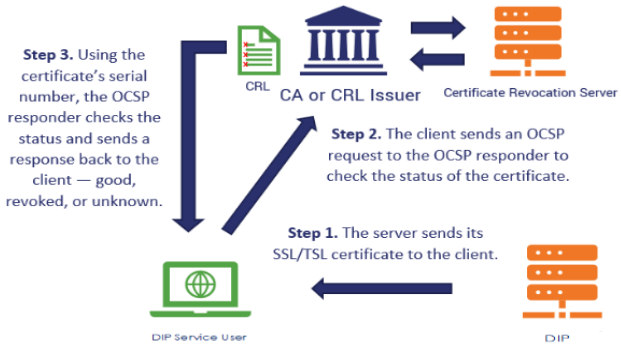



Figure 5 – Certificate Revocation List

#### 3.5.2.6 *CRLCache recommendations.*

Where a DIP Service Users system supports configurable CRL cache parameters, the recommendation would be to set the cache value to no more than 24 hours.

## 3.6 Key Management

### 3.6.1 Overview

Secure procedures for the handling of cryptographic keys are vital for Cryptosystems. Since Public-Key Cryptography depends on the secrecy of the Private Key, the expectation is that each DIP Service User should ensure that key material is protected appropriately.

For most organisations, this will involve putting in place a boundary protection device, such as a firewall, that can enforce the controls specified. The device, or Policy Enforcement Point (PEP), must have logical access to a key store for TLS authentication and be able to create or import keys in a secure fashion such as via a Cryptographic Module. What provides the PEP capability, and whether there is a Cryptographic Module in operation and the nature of the Local Public Key Store is at the discretion of the DIP Service User and subject to their own assessment of risk, subject to Code Bodies conditions as appropriate.

Each party shall raise a Major Security Incident as soon as possible notifying the DIP Manager once they become aware of compromise or suspected compromise of any private key material associated with a Certificate. This may result in the Certificate being revoked.

Where a Certificate is revoked by the DCA, the Certificate will be added to its appropriate X509 Certificate Revocation List, which shall be readily accessible.

Should a Certificate be revoked, the associated public and private keys will no longer be trusted. It is the responsibility of each DIP Service User to regularly poll the CRL to pick up any Certificate and Certificate Revocation information in a timely fashion. (The recommendation is to use the OCSP property on the certificate received to verify the status of the certificate)

# 4 DIP Interface Description

## 4.1 DIP Landscape

The DIP landscape will be a distributed network of services and roles that requires constant communication of data for operational purposes. The Event Driven Architecture (EDA) is an architectural pattern used for the production, management and consumption of data messages/events that enables the creation of a responsive/reactive, asynchronous, non- blocking/concurrent and de-coupled systems topology. Hence this is the chosen architectural pattern used to support the MHHS TOM.

There are approximately 27 Registration Services that will need to maintain operational data integrity and consistency across approximately 35 Data Services, 85 Metering Services, 17 DNO's and 60 Suppliers. In addition, the Data Services must provide an approximately total of 32 million daily (14 billion annual) consumption events to Central Settlement.

At the core of this architecture is the Data Integration Platform (DIP) which is responsible for brokering the communication between all industry participants operating under the TOM.

Figure 6 – Direct (Internet) Access below shows the DIP Interface(s) in the context of the components that it directly interfaces with:

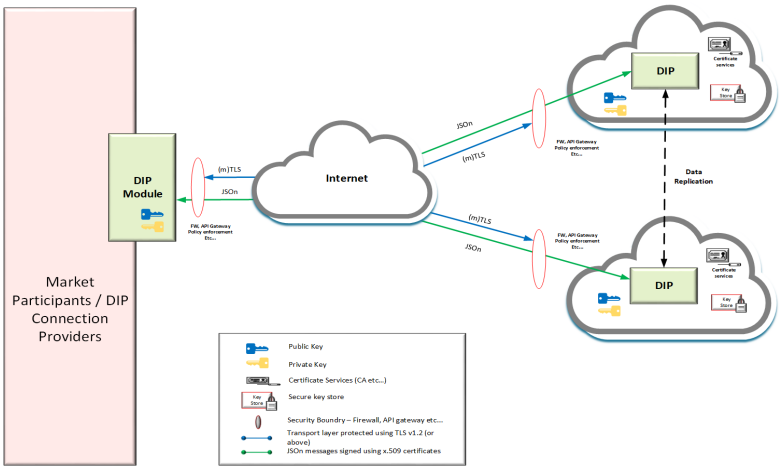



Figure 6 – Direct (Internet) Access

Inbound messages are sent to the DIP by calling pre-determined URLs (dependant on environment), whilst the outbound messages from the DIP are delivered to a URL (webhook[4]) of the recipient's choice.

The DIP is hosted on Microsoft's Azure public cloud. To maintain a geographically redundant service, the "Cloud Provider" execution venue takes advantage of Microsoft's UK South and UK North regions. Within the UK South region the DIP will be deployed across 3 availability zones and for UK North the DIP will deployed into 1 availability zone. This configuration provides the capability to failover seamlessly between physically separate availability zones and also between regions.

The DIP platform is protected against catastrophic failures pertaining to the hosted components that make up the entirety of the service. (All DIP components) The granular monitoring of each of the applications and underlying

[4] Webhooks are URLs which are event-based, meaning that when a specific event occurs in the DIP, a message can be delivered to a pre-defined URL.

infrastructure is paramount to providing a continuous, truly highly available platform. In the event of a catastrophic failure there would be no impact to the DIP Service user.

The Microsoft Azure platform will deliver a set of efficient, scalable, and dependable Internet Gateway (IGW) servers which will be backed by a scalable bank of firewalls that will secure the DIP application and its corresponding infrastructure. These firewalls will be configured to handle the mTLS connections for all incoming and outgoing JSON messages, thus offering policy-based security for the DIP program and its supporting system. The IGWs will be configured to manage the mTLS links for all incoming and outgoing messages. (It should be noted that only TLS v1.2 or higher will be supported.)

mTLS and digital signing certificates will be issued by the DIP Certificate Authority (DCA) See section 5 and section 6 for further details on PKI Roles and Certificates.

Behind the IGW the DIP uses Azure API management to create and maintain interfaces into the DIP via the Internet. Azure API management will be used to standardise inbound interfaces within the DIP solution.

At the other end of the internet connection will be the DIP Service User's Environment. The recommendation is for the DIP Service User Organisation to include both a Policy Enforcement Point (PEP) (i.e., firewall) and one or more API Gateways depending on their capacity and resilience requirements or alternative security controls which meet the minimum security requirements set out in section 2.3

In Figure 6 – Direct (Internet) Access above both Active and Non Active Market Participant connectivity can be achieved.

All certificates will be issued from the DIP Certificate Authority (DCA)

#### 4.1.1 Active Market Participant

A Market Participant who will be directly interfacing with DIP without the services of a DIP Connection provider.

One dual purpose Certificate is required:

- A Certificate approved for TLS (mTLS) to secure the channel between their PEP and the DIP
- A Certificate approved for Message Signing to sign messages that are sent to the DIP.

A certificate authorised for both mTLS and digital signing can be requested or separate certificates for mTLS and digital signing can be requested from the DIP User Portal and owned by the DIP Service User Organisation, i.e., the message originator. See Section 6 for further details.

All parties are advised to implement mTLS connection pooling both to support their own Non-Functional Requirements (NFRs) and to reduce the centralised and accumulated impact of potentially high traffic volumes on the DIP system.

#### 4.1.2 Non-Active Market Participant

A Market Participant who will be utilising the services of a DIP Connection Provider.

- The DIP Connection Provider will request their TLS certificate from the DIP.
- Following on from Organisation Vetting and Registration with GlobalSign a Market participant can nominate a DIP Connection Provider to perform the role of a Certificate Admin. The Certificate Admin will be granted privileges to login to the DIP User Portal and manage certificates on behalf of the Market Participant.
- Where a Market Participant chooses not to nominate Certificate Admin key materials need to be transferred securely to the DIP Connection Provider. See section 6.2 Private Keys for further details.

Deleted: A

A Non-Active Market Participant only requires a PKI certificate approved for message signing only.

Where a Non-Active Market Participant chooses to use multiple DIP Connection Providers a unique PKI certificate will be required for each DIP Connection Provider.

## 4.2 DIP Connections

Connections between the DIP and either Market Participants or DIP Connection Providers across the internet are required to be subject to cryptographic protection that ensures the authenticity, integrity, and confidentiality of the connection. Hence, all communications with the DIP shall be via a secure communications channel. Individual messages will flow over this secure communications channel.

The communications channel will consist of an encrypted and authenticated session between the DIP and DIP Service User Policy Enforcement Point (PEP). Policy Enforcement Points will typically be boundary firewalls that can negotiate the security parameters required to set-up a secure TLS session.

The TLS session **must** be configured for mutual authentication (mTLS), such that each entity authenticates the other during the handshake protocol.

### 4.2.1 Message Validation

The DIP performs message validation in two phases, a synchronous check at the API that are communicated directly back to the Sender, and then further asynchronous checks and addressing that are carried out once the message has been initially received. The rationale behind the split, is that the initial API is intended to be as lightweight as possible thereby increasing message submission speed.

Where no error condition is encountered no event/message acknowledgement other than the initial API response is sent back to the Sender.

The messages have specific security properties that must be satisfied. These properties are:

- **Integrity of the message** – This property is to ensure that the message received is as it was when sent and has not been accidentally or maliciously changed in transit;
- **Data origin (message) authentication** – Because of the threat of a message or information being sent being impersonated by an unauthorised party, data origin authentication (sometimes called message authentication) is the assurance that a given entity is the original source of the received data. The message is authenticated (signed) using a cryptographic mechanism (in this case a cryptographic key). Where required each message is individually authenticated to allow the recipient to verify the claimed identifier of the message;
- **Mutual Authentication** - Both parties engaged in a communication session authenticate each other such that both are assured of the identity and authenticity of the other;
- **Confidentiality** – Confidentiality is a critical security property because data privacy is a constraint for the MHHS Programme due to legislation such as the General Data Protection Regulation (GDPR).

All synchronous interactions must satisfy both data origin authentication, data integrity, confidentiality and non-repudiation. Consequently, all synchronous interactions should be rejected if:

- They are not from an authentic source; or
- They have been changed in transit; or
- They are not aligned to the JSON schema - defined in *https://app.swaggerhub.com/organizations/MHHSPROGRAMME*.

## 4.3 Interfacing with the DIP

The technical interface specification is defined by the **[1] MHHSP-DES138-Interface Catalogue**. It defines the MHHS interface at a logical level to enable Market Participants and DIP Connection Providers to update their current systems to meet its requirements. It contains instructions on how messages are to be composed, the structure of the messages and any specific data validation to be performed. (API definition can be found here *https://app.swaggerhub.com/organizations/MHHSPROGRAMME*)

### 4.3.1 Configurable Parameters

The solution has been designed so that operational service limits can be enforced for:

- Message ingress
  - Maximum payload size See **[2] MHHS-E2E001 - End-to-End Solution Architecture**
- Message Egress
  - Maximum number of events and
  - Maximum payload size See **[2] MHHS-E2E001 - End-to-End Solution Architecture**;
- Back off and re-try values (see a suggested message retry strategy below);

Which is supported by dedicated error handling (details of which will be found in the See **[2] MHHS-E2E001 - End-to-End Solution Architecture**

Message retention periods currently 2 year default **[3] MHHS-DIP002 - Functional & Non-Functional Requirements**.

### 4.3.2 Back off and re-try values Strategy

Participants are expected to adopt a retry with exponential back off (up to a configurable maximum wait time) if there is a failure in connecting to the DIP. See **[7] MHHS-E2E002 - End-to-End Functional & Non-Functional Requirements** –requirement E2E0105.

# 5 Participant Engagement

## 5.1 Overview

Below is an overview of the on-boarding process DIP Service Users will follow: (Still under review)

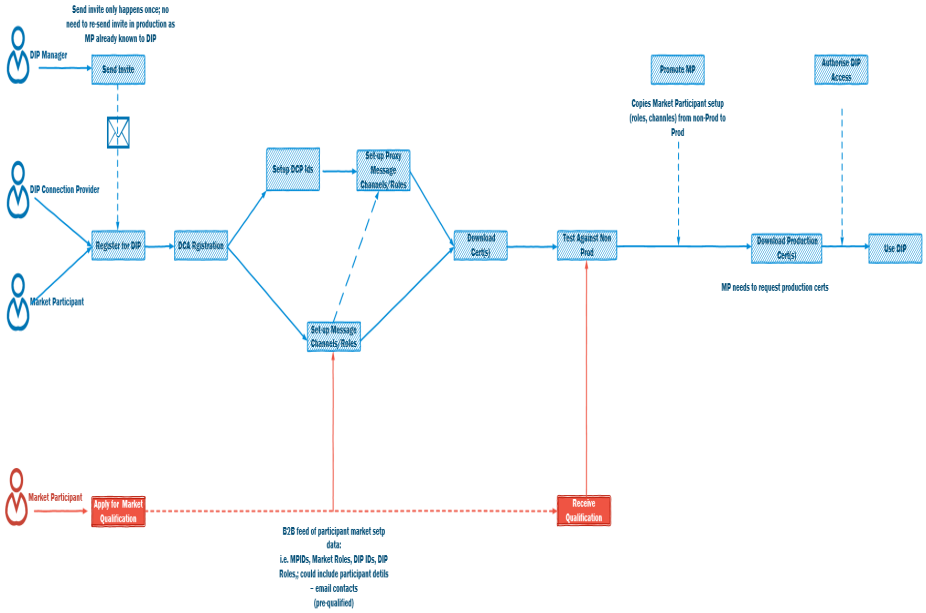



Figure 7 – On-Boarding Steps

1. Register for the DIP i.e. receive invitation from DIP manager and follow steps in section 5.4
    a. In parallel Market Participants need to pursue Market Qualification.
2. DCA registration See section 5.4
3. Setup message channel configuration (Guidance to be provided).

Download non-production certificates (see section 6

4. Managing DIP Certificates for further information).
5. Test in non-production environment as part of the qualification process.
6. Achieve Market Qualification and promotion to production DIP.

Download production certificates (see section 6)

7. Managing DIP Certificates for further information).
8. Use production DIP.

## 5.2 DIP On-boarding

All DIP Service Users (Market Participants and DIP Connection Providers) who require access to the DIP must have completed the on-boarding process before they can fully utilise the DIP. Market Participants will also need to have completed market qualification for the market roles they want to perform with the MHHS TOM before they can operate in the production DIP environment. Access to non-production environments is possible before qualification is complete, in fact is important for Market Participants to access these non-production environment as part of the qualification process is exhibiting correct DIP use.

The DIP Manager will invite DIP Service Users to join the DIP:

- An email with details of how to access the DIP portal will be sent to the DIP Service Users registered email
- On accessing the DIP User Portal the User Admin will verify company information and assign their first Certificate Admin. This first Certificate Admin must be a permanent employee of the Market Participant and will be required to register with the DCA before digital certificates can be obtained. See section 12.1 Organisational vetting and registration process flow.

See section 5 Participant Engagement and section 6

- Managing DIP Certificates for further details on digital certificates (PKI Certificates)

A prerequisite to using digital certificates in the DIP is that ALL DIP Service Users must have passed organisational vetting and domain vetting by the DCA (See section 5.4.4) once the vetting process is complete, DIP Service Users can login to the DIP User Portal and request Certificates.

- Certificates will be issued using the DCA.

The DIP Manager will not routinely assign market roles to Market Participant. Market Roles are assigned when the Market Participant has completed the registration process with their code body.

## 5.3 DIP Certificate Governance

DIP Certificates are governed by the DIP Manager.

## 5.4 Registration

Please refer to the DIP Manager for further details on the process for new and existing Market Participants (To be defined through the wider qualification working group)

### 5.4.1 DIP Service User Administrator – Mandatory Role

Recommendation is for a minimum of 2 DIP User Administrators. (These can also act as the Certificate Admin).

The DIP User Administrator will be responsible for managing PKI role Certificate Admins within the DIP User Portal on behalf of their organisation.

Note: The DIP Service User Administrator will also be responsible for the management of additional DIP roles such as Certificate Admin, Message Admin, and Analytics Admin.

### 5.4.2 DIP Service User PKI Representatives

#### 5.4.2.1 *How many individuals should be assigned PKI Roles*

The DIP User Organisation should consider how many users in total are nominated into Certificate Admins for managing DIP certificates. There should be a suitable number of individuals with assigned Certificate Admins to manage certificates in the event of sickness, absence, holidays etc.

- Without a valid certificate the DIP will reject messages.

Formatted: Strikethrough

Deleted: is

#### 5.4.2.2 *PKI Role Governance*

The governance of additional Certificate Admin roles is at the discretion of the DIP User Organisation. The DIP Manager will not request any employee validation letters for these roles.

Note: DCA Certificates cannot be issued to the DIP User Organisation until organisational vetting and registration has been completed by the first Certificate Admin.

#### 5.4.2.3 *Certificate Admin – Mandatory Role*

The Certificate Admin is an individual or individuals who have the knowledge and capability to manage Certificates and the associated processes (initial Certificate signing requests, replacements, revocation requests etc.) on behalf of the DIP User Organisation, who ultimately are the Certificate-owning organisation.

- The level of seniority of the first Certificate Admin (Permanent employee) is at the discretion of the registering organisation. The first Certificate Admin has the authority to make decisions for or on behalf of the organisation and will agree to the relevant terms and conditions of the DIP;
- Ensure the certificate admin is aware of Certificate activities such as Certificate Signing Requests (CSRs) or Certificate Revocation Request (CRRs)
- A DIP User Organisation using a Certificate Admin who is an employee of a DIP Connection Provider is ultimately responsible for ensuring the Certificate Admin complies with all relevant aspects of certificate management as set out in this document. In all cases, the DIP User Organisation is responsible for compliance with the DIP Manager.
- Additional Certificate Admin's may be employees of a different organisation. In this case it is expected that there will be a commercial agreement between the Third Party and the Market Participant.

### 5.4.3 Roles Privilege Table

**Formatted:** English (US)

**Formatted:** Strikethrough

**Moved (insertion) [2]**

**Deleted:** TC

**Formatted:** Indent: Left: 0.63 cm, No bullets or numbering

**Formatted:** Not Strikethrough, Not Highlight

**Moved (insertion) [1]**

**Deleted:** TC

**Moved up [2]:** The TC is an individual or individuals who have the knowledge and capability to manage Certificates and the associated processes (initial Certificate signing requests, replacements, revocation requests etc.) on behalf of the DIP User Organisation, who ultimately are the Certificate-owning organisation;¶

**Moved up [1]:** A DIP User Organisation using a TC who is an employee of a DIP Connection Provider is ultimately responsible for ensuring the TC complies with all relevant aspects of certificate management as set out in this document. In all cases, the DIP User Organisation is responsible for compliance with the DIP Manager;¶
A TC, as noted above, may be an employee of a different organisation. In this case it is expected that there will be a commercial agreement between the Third Party and the Market Participant. ¶

**Formatted:** Highlight

**Formatted:** Strikethrough

| Privilege | Org. DIP Service User Administrator | Certificate Admin | Previous PKI Role mapping | DIP Manager |
|---|---|---|---|---|
| Invite Certificate Admin to undertake organisational vetting and registration. | X | | | |
| Manage (add/remove) Certificate Admin roles within the DIP User Portal. | X | | | |
| Organisation Vetting and Registration | | First Certificate Admin | Nominating Officer | |
| Submit a CSR (Certificate Signing Request) | | X | SRO, ARO, TC | |
| Access PKI Certificate from DIP User Portal | | X | SRO, ARO, TC | |
| Access IA Trust Chain (public) Certificates from DIP User Portal | | X | SRO, ARO, TC | |
| Submit a CRR (Certificate Revocation Request) | | X | SRO, ARO, TC | |
| Submit a Certificate rekey request | | X | SRO, ARO, TC | |
| Get notified from the DIP when PKI Cert. nearing expiry (90, 60, 30, 1 days) | X | X | SRO, ARO, TC | |
| Get notified whenever a change of Certificate Admin | X | X | SRO, ARO, TC | X |
| Get notified on a lifecycle status change of CSR | | X | SRO, ARO, TC | X |
| Get notified on a lifecycle status change of a CRR | | X | SRO, ARO, TC | X |
| Get notified on a lifecycle status change of a Certificate rekey | | X | SRO, ARO, TC | X |
| Get notified when PKI Certificate available for download | | X | SRO, ARO, TC | |
| Able to download certificate from DIP User Portal | | X | SRO, ARO, TC | |

Table 2- Roles Privilege Table

Deleted: Org. SRO

Formatted: Font: 8 pt, Highlight

#### 5.4.4 Vetting and registration process

Before any DIP Service User can request certificates in the DIP, the Certificate Admin from the DIP Service User Organisation must complete the DCA registration and the DIP Service User Organisation must be successfully vetted by the DCA to confirm authenticity. See section 12.1 for further information.

##### 5.4.4.1 *DNS Domain creation/verification*

A prerequisite to requesting a certificate is to create and validate the DIP users Organisations DNS domain with the DCA, a DIP Service User who has been granted the role of Certificate Admin can register the DIP User Organisation DNS domain in the DIP Portal. See section 12.1.2 for further information.

Full details of the Public Key Infrastructure requirements can be found in section 3 of this code of connection document and in the **[4] MHHS DEL1210 – PKI Policy.**

### 5.5 What DIP Certificates Do I need?

DIP service Users will generally have two sets of certificates: one for non-Production environments and one for Production. Where Market Participants utilise the services of DIP Connection Providers then the responsibility of the management of the certificates is split: the Market Participant owns the certificate required for digital signing and will need to authorise the DIP connection provider to use this certificate on their behalf and the DIP connection Provider will need their own digital certificate for securing the mTLS connection.

| Party | TLS Certificate | Digital Signature Certificate | Notes |
|---|---|---|---|
| **Market Participant (active)** | Yes | Yes | The DIP Digital Signature Certificate is made available to allow Parties to verify signed messages from the DIP. A certificate authorised for both mTLS and digital signing can be used or separate certificates for mTLS and digital signing can be requested. |
| **Market Participant (non-active)** | No | Yes – see notes | These parties do not themselves connect to the DIP instead a DCP acts on their behalf. They give authority to DCPs to sign messages using their organisational certificates. |
| **DIP Connection Provider**[5] | Yes | No – see notes | DIP Connection providers supporting multiple clients need only a single connection between their service and the DIP, secured by their own TLS Certificate.<br>Messages must be signed using a certificate provided by the DIP Market Participant who they are representing |

Table 3 - Digital Certificate Requirements by Party Type

The following section describes how certificates are managed in 3 different scenarios:

[5] DIP Connection providers typically receive messages from subscribers in one format and transform to JSON format on behalf of the subscriber before forwarding on to the DIP.`


Figure 8 – Certificate Management Scenarios

Scenario 1 – Market Participant A direct to DIP as DIP Service User.

- Market Participant A manages their own connection to the DIP using the same multi use certificate for TLS and Digital signing.
- One Certificate for non-production environments
- One Certificate for production environment.

Scenario 2 - Market Participant B uses DIP connection Provider X for connection to DIP

- Market Participant B authorizes DIP connection Provider X to sign messages on their behalf using Market Participant B Digital certificate (this certificate does not allow mTLS)
- DIP connection Provider X uses their own digital certificate for mTLS.

Scenario 3 - Market Participant C uses multiple DIP connection Providers.

- Market Participant C requests multiple digital certificates: one for DIP connection Provider X and one for DIP Connection Provider Y.

**Deleted:** B

- Market Participant C authorizes both DIP Connection provider X and DIP connection Provider Y to sign messages on their behalf.
- DIP connection Provider X uses their own digital certificate for mTLS.
- DIP connection Provider Y uses their own digital certificate for mTLS.

Deleted: B

Hence, for Market Participants directly connecting to the DIP this will be a single certificate issued for both mutual TLS and digital signing.

DIP Connection providers may not use Certificates for different purposes. For example, a Digital Signature Certificate may not be used for TLS, and vice versa.

Certificates bind to an environment as well as an identity. This means that Market Participants and DIP Connection Providers may not re-use Certificates across different environments (Pre-Production and Production). For example, Market Participants and DIP Connection Providers cannot use their PRE-PROD Certificates to secure their PROD environment.

## 5.6 Certificate Subscriber Obligations

### 5.6.1 Overview

DIP Service Users have certain certificate subscriber obligations which become effective immediately on receipt of a signed Certificate from the DCA/Issuing Authority. Certificate Subscribers must understand and accept these obligations before applying for, accepting, or using a Certificate.

Once issued, and before the issued certificate is used, the accuracy of the information contained within must be reviewed.

### 5.6.2 Obligations

By applying for a Certificate (and submitting information requested as part of an application), Market Participants are requesting that the Issuing Authority issue a Certificate to them, or a DIP Connection Provider acting on their behalf; and that they are expressing agreement to the terms of terms and conditions of Certificate use governed by the DIP Manager as may be amended from time to time.

The DIP Service Users warrant that:

- They will abide by the terms of the DIP Manager and use the Certificate and any associated services only in accordance with the rules contained therein;
- The information that they provide in support of their application for a Certificate is true and accurate;
- Their company or other organisation holds such rights as necessary to any trademarks or other such information submitted during the application for a Certificate;
- They will keep any secret information such as passwords, passphrases, PINs, private keys, or other personal secrets that they use in obtaining authenticated access to the DIP PKI services, such as Certificate Repositories;
- They will protect such secret information from access by anyone or anything until it is securely destroyed or deleted;
- No unauthorised person has, or has ever had, access to the secret information that they use in obtaining authenticated access to the DIP PKI services;
- They shall immediately inform the DCA /Issuing Authority (GlobalSign) should such secret information be compromised or be suspected of being compromised;
- They acknowledge that the DCA / Issuing Authority (GlobalSign) at its sole discretion may conduct revocation of any Certificate issued by it.

Parties also acknowledge that certain information provided by them at the time of enrolment will be embedded in a Certificate and may be published in a directory of Certificates and Revocation Information where required for the purpose of operating the digital certification services. They consent to the disclosure of this information for these purposes and understand that they have the right to correct any information as required.

For a device or application, the individual responsible for the device or application must accept these responsibilities. For those entities who hold Certificates and act on behalf of Subscribers, the Subscriber must ensure all responsibilities are met.

**WARNING:** If a DIP Service Users Private Key is compromised, unauthorised persons could decrypt or sign messages with the key and commit the Subscriber to unauthorised obligations.

## 5.7 Relying Party Obligations

### 5.7.1 Overview

Relying Parties are those entities that are using a Certificate to authenticate another Certificate Subscriber named in the Certificate. In the context of the DIP, every Certificate Subscriber is also a Relying Party at some point in the process of exchanging messages with the DIP and DIP Service Users.

### 5.7.2 Obligations

Relying Parties are advised to postpone any transaction which places reliance on or by any Certificate issued by the Issuing Authority until they are satisfied of the trustworthiness of the Certificate and of the measures to be taken to support their Reliance on the Certificate. This applies when they:

- Submit a query in search of a Certificate;
- Verify a Digital Signature created with a Private Key corresponding to a Public Key contained in a Certificate;
- Rely on a Certificate or on the information contained within a Certificate in support of a transaction;
- Otherwise rely on or use any information or services provided by the DIP Issuing Authority Certification Services.

Parties accept and warrant that their use of Certificate Status Information and their Reliance on or use of any Certificate or the information embedded in it will be governed by the DIP Manager.

The DIP Manager defines specific obligations that parties must meet before Relying on a Certificate or any information embedded in the Certificate. If parties rely on or use a Certificate (including the information embedded in the Certificate) without first meeting these obligations, their reliance on or use of that Certificate or the information will be inconsistent with the DIP Manager and will be considered unreasonable and in derogation of their duty of care under the Common Law of England and Wales.

DIP Manager Provisions defining terms of use of and Reliance on Certificates, together with Certificate Status Information are provided by the Issuing Authority. Relying Parties must therefore accept that sufficient access to information is provided to ensure that they can make an informed decision as to the extent to which they will choose to rely upon or use a Certificate or the information embedded in it. They further accept that they are responsible for deciding whether to rely on a Certificate or information embedded in it.

Subscribers of Certificates issued by the DCA /Issuing Authority (GlobalSign) must inform the Issuing Authority or its nominated representative (see relevant contact details in 6.1), should any passwords, passphrases, PINs, private keys, or other personal secrets used in obtaining authenticated access to PKI Services be compromised or be suspected of being compromised.

On receipt of such a notice, the DCA / Issuing Authority (GlobalSign) will Revoke the compromised Certificate and publish notice of such Revocation for the benefit of Relying Parties. However, Relying Parties acknowledge the possibility of theft or other form of compromise of a Private Key corresponding to a Public Key contained in a Certificate which may not be detected by the Certificate Subscriber, and of the possibility of use of a stolen or compromised Key to authenticate an organisation or to forge a Digital Signature to a document.

# 6 Managing DIP Certificates

## 6.1 Introduction

This section describes in more detail the process for obtaining DIP PKI Certificates as well as the main roles and functions of the PKI service. It details the processes to be followed and information to be provided by DIP Service User when requesting DIP PKI Certificates from the DIP Certificate Authority (DCA).

DIP PKI Certificates will be available in readiness for SIT testing. The DIP PKI Certificate processes will be managed using the DIP User Portal where Certificates will also be distributed. The **Central Point of Contact** for all matters relating to DIP PKI Certificates is:

- bsc.change@elexon.co.uk.

DIP Service Users are also advised to review the **[4] MHHS DEL1210 – PKI Policy.**

DIP Service Users will be responsible for managing and securing the certificates they use to communicate with the DIP, there are four actions in the management of certificate:

- Issuing of certificates
- Revocation of certificates
- Renewal of a certificate – prior to expiry
- Reissue of a certificate

A prerequisite to these actions is that the DIP Service User has passed organisational vetting and registration.

Certificates will be issued from the DCA.

The certificates issued by the DCA are currently valid for 398 days which equates to 1 year and a month overlap.

### 6.1.1 Certificate Issuance

This following sections describes in more detail the process for obtaining DIP Certificates as well as the main roles and functions of the DIP service. See section 5.4 for further information on the processes and roles involved in managing certificates.

On successful verification of a PKCS #10 Certificate request the DCA will generate a Public-Key Certificate for the DIP Service User's Public Key and place that Certificate within a publicly accessible repository.

### 6.1.2 Certificate signing requests

The DIP Service User (Certificate Admin) can submit a request for a new certificate by following the process below:

To request a new certificate the DIP Service User (Certificate Admin) will use the DIP User Portal to provide a Certificate Signing Request (CSR), the signing will be fulfilled by GlobalSign.

- DIP Certificate Admin can only request certificates through the DIP portal.

Once signed, the certificate is fulfilled and therefore considered as a certificate towards the market participant's quota.

The certificate request completion only works on the server/service where the CSR was generated, should it be completed elsewhere then it will not complete.

| Name | Description |
|---|---|
| Common name | This value will contain a prefix for the environment and the domain which they are requesting a certificate for. The prefixes will be as follows:<br>• energydip-nonprod – All Non Production environments<br>• energydip-prod – Production environment<br>For example, the following value could be specified:<br>• *energydip-nonprod.marketparticipant.co.uk*<br>• *energydip-prod.marketparticipant.co.uk* |
| Organisation name | The name of the organisation as specified during Organisational vetting. |
| City | The city of the organisation as specified during Organisational vetting. |
| State | The state of the organisation as specified during Organisational vetting. |
| Country | The country of the organisation as specified during Organisational vetting. |

Table 4- Certificate Request Retails.

#### 6.1.3 Certificate revocation

A certificate may need to be revoked for several reasons. (See **[4] MHHS-DEL1364 – DIP PKI Policy** section 5.7 for full detail)

An approved Certificate Admin can revoke certificates using the DIP User Portal following the process below:

- From within the portal, the DIP Service User (Certificate Admin) navigates to the certificates page, the DIP Service User will be shown their current certificates.
- Under the certificate actions option, they can choose Revoke.
  - To revoke a certificate a reason for revocation must be entered selected from a list of possible reasons:
- On submission of the reason, the DIP portal will request the certificate is revoked by the DCA.
- The DIP portal will inform the DIP Service User (See section 5.4.3) that the certificate is successfully revoked.

Once revoked the certificate will no longer be valid when calling the DIP as either the mTLS or message signing certificate. During the process of mTLS or message signing the Online Certificate Status Protocol (OCSP) is called. The OCSP is a property of the certificate and is an endpoint that specifies the certificate status (valid/revoked).

| Reason | Description |
|---|---|
| Key compromise | If the DIP Connection Providers key has been lost, permanently deleted or if an unauthorized entity has been able to take possession of the key, the certificate must first be revoked before being recreated from scratch with a new key. |
| Cessation of operation | If the service user ceases to operate, the certificate must be revoked. This reason can only be used by the DIP Manager. |
| Affiliation changes | This is when a key employee leaves the DIP Service User Organisation. A key employee is an employee that has access to the certificate and associated keys. |
| Certificate superseded. | If a new certificate has been produced for any reason, the old certificate will be superseded and will require revocation |
| Withdrawal of privilege | The DIP Service User is no longer allowed to access the DIP; therefore, their certificate should be revoked. |

**Deleted:** therefore

| | |
|---|---|
| Removal from CRL | If a certificate is accidentally revoked for any reason and should not be on the Certificate Revocation List (CRL), that certificate will need to be removed from the CRL. This will be a very rare occurrence. |

Table 5- Certificate Revocation Details.

See **[4] MHHS-DEL1364 - PKI Policy section** 5.7 for further information on certificate revocation.

#### 6.1.4 Certificate renewal

Prior to expiry a Certificate Admin should generate a new CSR and get it signed via the DIP User Portal, the process for this is the same as 6.1.1 Certificate Issuance.

As all requests for signing come through the DIP portal, the portal will notify the DIP Service User that a certificate is about to expire and therefore that they should generate a new CSR and get it signed via the DIP portal. See 5.4.3 Roles Privilege Table for further details of which DIP Service Users will be notified.

Note: Renewing a certificate does not invalidate the current certificate. The current certificate will remain active for the remainder of the validity period allowing a grace period for seamless transfer.

#### 6.1.5 Certificate renewal notifications

Notifications of certificate expiry will be sent to the DIP Service User Administrator at the following intervals.

- 90 days prior to the certificate expiring
- 60 days prior to the certificate expiring
- 30 days prior to the certificate expiring
- 1 day prior to the certificate expiring.

The new certificate will start from the date the Certificate Signing Request has been completed and not the date the current certificate expires.

#### 6.1.6 Certificate rekey.

If you'd like a copy of your certificate, for example are you installing on multiple servers or devices? Additionally, If you encounter a private key error and cannot fully install your Client Digital Certificate, you can simply reissue your certificate.

An approved Certificate Admin can submit a request to reissue a certificate by following the process below:

## 6.2 Private Keys

The size of Issuing Authority and any supporting CA-Keys shall be not less than 4096-bit modulus for RSA.

The size of Subscribers' Private Keys shall be not less than 2048-bit modulus for RSA.

DIP Service Users are responsible for the Back-Up of their own keys.

DIP Service Users, who are natural persons, must be authenticated to their cryptographic module before the activation of the Private Key. This authentication may be in the form of a PIN, pass-phrase password, or other activation data. When deactivated, Private Keys must not be exposed in plaintext form.

Deleted: n

Deleted: intervals;

Deleted: completed

Deleted: rekey

Deleted: Additionally

Deleted: Users who are natural persons

Unless unavoidable, private keys should never be transmitted. That said, there is a requirement for DIP Connection Providers to hold the private keys of subscribed Market Participants.

Where private keys must be moved, the advice is to export and transport private keys using a PKCS #12 file. A PKCS #12 file - also known as PFX - is a single, password protected Certificate archive that contains the entire certificate chain plus the matching private key.

Window servers have a utility through the MMC that allows the export of an installed TLS server Certificate along with its corresponding private key to a PFX file.

For Linux-based servers you can use OpenSSL to manage Certificates and keys including creating various file bundles including PKCS #12 archives.

For other types of devices, the instructions will vary depending on the type of device you are using. Please consult your documentation.

The PKCS #12 file encryption key should never be used for another transfer, even for a different private key.

### 6.3 Encryption Secrets/Password Guidance

When creating encryption secrets or passwords to protect key material, Parties are referred to NCSCs guidance on password strategies, and specifically the use of password managers and random words and phrases:

- https://www.ncsc.gov.uk/blog-post/three-random-words-or-thinkrandom-0

# 7 Transport Layer Security (mTLS)

Mutual TLS (mTLS) is a method for mutual authentication where both the client (Service User) and the DIP confirm they trust each other by verifying certificates sent as part of establishing the connection.

- mTLS will be used on both the ingress data API and egress webhook.

**Deleted:** mTLS) is

#### 7.1.1 TLS Requirements and Configuration

The secure mTLS session will be based on the Transport Layer Security (TLS) v1.2 protocol standard (TLS v1.2 as a minimum) and will make use of authentication using PKCS #3 Ephemeral Diffie Hellman key exchange to generate a shared secret (TLS-RSA) with AES-128-GCM-SHA256 for communications encryption.

If this Authentication step fails, an “HTTP 403 Unauthorized” error will be returned to the DIP Service User. The error codes are referenced in the Interface Specification **[1] MHHSP-DES138-Interface Catalogue**

#### 7.1.2 TLS Key Generation and Certificate Signing Requests (CSRs)

The DIP SP issues RSA Certificates for TLS – in line with the NCSC guidance – with the following parameters:

- 4096-bit RSA with SHA256

Public/Private key pairs must be created following successful completion of a DIP Service User’s registration process. Each DIP Service User will be responsible for the generation of TLS keys and CSRs used by their message service interface.

Each DIP Service User will need to issue an associated PKCS #10 Certificate Signing Request’s (CSR) to the DCA to obtain a Public-Key Certificate.

No DIP Service User may make a Certificate Signing Request which contains:

- Any information that constitutes a trademark unless it is the holder of the Intellectual Property Rights in relation to that trademark; or
- Any confidential information which would be contained in a Certificate issued in response to that Certificate Signing Request.

Where any CSR fails to satisfy the requirements set out in Section 6.1.2 of the Code of Connection, the DCA:

- Shall reject it and refuse to issue the Certificate which was the subject of the Certificate Signing Request; and
- May give notice to the Party which made the Certificate Signing Request of the reasons for its rejection.

Where any Certificate Signing Request satisfies the requirements set out in Section 6 of the Code of Connection, the DCA shall issue the Certificate which was the subject of the Certificate signing Request. See section 12.3 Mutual TLS (mTLS) process flow for further details.

# 8 Message Signing

Authentication within the DIP is based on two form factors:

- The first is covered by the establishment of a secure communications channel provided by mTLS, as described above.
- The second is the authentication of individual messages which is achieved through the application of a digital signature.

Applying a digital signature to a message also adds a layer of data integrity assurance.

Digital signatures are applied to hashes of JSON messages (sometimes called message digests) and are used to detect unauthorised modifications to data, as well as to authenticate the identity of the signatory. In addition, the recipient of signed data can use the digital signature as evidence in demonstrating to a third-party that the signature was, in fact, generated by the claimed signatory.

This section outlines the processes for generation, distribution, use and storage of message signing keys and certificates.

## 8.1 Message Configuration Requirements

For most messages sent to or from the DIP there is a requirement to apply message signatures (the exception being some HTTP responses / acknowledgements which do not need to be signed). Receiving applications must verify the message signatures, effectively performing authentication and authorisation of the message sender and confirming that the message has not been tampered with in transit. Verification of each message should be performed using the digital signature Certificate relevant to the environment being used (PRE-PRODUCTION / PROD).

On receipt of a message and where JSON schema validation is successful, the recipient will authenticate the message by verifying the message signature which is required on all messages. This Digital Signature uses an RSA 256 key.

Messages that fail signature verification will not be processed by the DIP. In this instance, the DIP Service User will be advised of authentication failure as part of the HTTP Response or Acknowledgment generated and sent to the requesting DIP Service User, along with an appropriate Response Code.

## 8.2 Message definition

Message structures are specified through the definition of a JSON schema. The definition of the schema to be used for the DIP can be found in the latest [1] **MHHSP-DES138-Interface Catalogue**.

## 8.3 Message signatures

A DIP message signature represents digitally signed content using JSON data structures and base64 encoding. A message contains the following values:

| Header Value | Format | Details |
|---|---|---|
| X-DIP-Signature | BASE64 encoded.<br>UTF-8 Character set.<br>Padding RSA PKCS-1 | Ensure BASE64 encoding of the message signature |
| X-DIP-Signature-Date | Plain Text | The message signature date |
| X-DIP-Signature-Certificate | BASE64 encoded.<br>UTF-8 Character set. | Ensure BASE64 encoding of public key certificate of the sender (private key used to generate signature) |

Deleted: above;

Deleted: a

Deleted: url

Formatted Table

Deleted: The

Deleted: Base64

Deleted: ¶
Padding RSA PKCS-1

| | | |
|---|---|---|
| X-DIP-Content-Hash | BASE64 encoded.<br>UTF-8 Character set. | SHA256 Hash of JSON Message Body.<br>BASE64 encoding of the SHA256 Hash (using UTF-8 Character set). |

Table 6- RFC 7515 Logical Values.

Deleted:

Deleted: ¶
Padding RSA PKCS-1

Deleted: Base64 URL encoded JSON payload using UTF-8 Character encoding

Formatted: Body Text

## 8.4 Signing Messages

To create a message signature, perform the following steps:

| Step | Action |
|---|---|
| 1 | Generate the content of the message, this is the message body. |
| 2 | Create a SHA256 hash of the message body. BASE64 encode the output from the SHA256 hash (this is the SHA256 Content Hash). |
| 3 | Generate an ISO8601 timestamp in the format: YYYY-MM-DDTHH:MM:SS.XXX:Z. This is the Signature Date. |
| 4 | Concatenate the Signature String in the following format:<br>▪ Verb;Destination;Signature Date;SHA256 Content Hash<br>***Note:***<br>• Verb = Uppercase only<br>• Destination = Lowercase only |
| 5 | Create a SHA256 hash of the Signature String (this is the SHA256 Signature string). |
| 6 | Sign the SHA256 Signature string using the private key of the signing certificate. This is the Signed Signature. Utilise RSA PKCS1 signature padding. |
| 7 | Create BASE64 encoded string of the Signed Signature. |
| 8 | Add to the request the header: X-DIP-Signature with the value of the BASE64 encoded Signed Signature. |
| 9 | Add to the request the header: X-DIP-Signature-Date the value of the Signature date. |
| 10 | Add to the request the header: X-DIP-Signature-Certificate with the BASE64 encoded signing certificate (this will contain only the public key). |
| 11 | Add to the request the header: X-DIP-Content-Hash with the BASE64 encoded SHA256 Content Hash |

Table 7- Signing Messages.

Formatted Table

Formatted: Font: (Default) Calibri

Formatted: Font: Bold, Italic

Formatted: List Bullet 2, Space After: 0 pt, Add space between paragraphs of the same style, Line spacing: single, No bullets or numbering

Formatted: List Bullet 2, Space After: 0 pt, Add space between paragraphs of the same style, Line spacing: single, Bulleted + Level: 1 + Aligned at: 0.63 cm + Indent at: 1.27 cm

Deleted: Step ... [1]

## 8.5 Verifying Signatures

When validating a message signature, the following steps MUST be performed. If any of the listed steps fails, then the signed content MUST be rejected:

| Step | Action |
|---|---|
| 1 | Verify the certificate received (X-DIP-Signature-Certificate) is trusted and valid. |
| 2 | Take the value of the header: X-DIP-Signature and decode using BASE64. This is the BASE64 Decoded Signature. |
| 3 | Build a comparison string: |

| | |
|---|---|
| | • Verb - from HTTP Request received.<br>• Destination - from HTTP Request (the url of the endpoint the message has been received on).<br>• Signature Date - ISO8601 timestamp in the format: YYYY-MM-DDTHH:MM:SS.XXX:Z (retrieved from the X-DIP-Signature-Date).<br>• Content Hash - Create a SHA256 hash of the JSON Message body received. BASE64 encode the output from the SHA256 hash. For any request with an empty body (GET, DELETE requests) please assume an empty JSON body – {} |
| 4 | Create a SHA256 hash of the comparison string. This is the SHA256 hashed comparison string. |
| 5 | Verify the SHA256 hashed comparison string with the BASE64 Decoded Signature, using the public key of the certificate received (X-DIP-Signature-Certificate) |

Table 8- Verifying Message Signatures.

Deleted: Step ... [2]

## 8.6 Signature Key Generation and Certificate Signing Requests (CSRs)

The DCA issues Certificates for Digital Signatures (in line with NCSC guidance) with the following parameters:

- RSA 4096
- ~~ECDSA 256 with SHA256 on the P-256 curve~~

Public/Private key pairs must be created following successful completion of the registration process. Each registered organisation will be responsible for the generation of mTLS keys and Certificate Signing Requests (CSR) used by their message service interface.

A DIP Service User or DIP Connection Provider who is a PKI role holder (Certificate Admin) will need to create an associated PKCS #10 CSR and upload the CSR to the DIP portal, where the CSR will be used to generate a signed public key of the certificate, signed by the DCA.

The embedded spreadsheet in section 8.7 contains the definitive profile for the DIP Certificates. Parties must ensure that they adhere to this profile when creating CSRs. The certificates must be built or configured as indicated in the profile to ensure they work correctly, and CSRs are not rejected.

The DIP Service User granted appropriate PKI permissions (See Table 5.4.3 Roles Privilege Table) should use the distribution mechanisms for CSRs and Certificates outlined in Section 6 Managing DIP Certificates

Tools for public/private key pair and CSR generation will depend on the local environment. Representative guidance is available in the following links:

| Tool/Platform | Guidance |
|---|---|
| OpenSSL | https://wiki.openssl.org/index.php/Command_Line_Utilities |
| Microsoft Azure Key Vault | https://docs.microsoft.com/en-us/azure/key-vault/certificates/create-certificate-signing-request |
| AWS | https://docs.aws.amazon.com/cloudhsm/latest/userguide/ssl-offload-windows-create-csr-and-certificate.html |

Table 9- CSR Guidance.

## 8.7 Certificate Profile

The **[7] MHHS-DEL-1387- DIP PKI Certificate Profiles** contains the definitive profile for the DIP PKI Certificates. DIP Service Users must ensure that they adhere to this profile when creating CSRs. The Certificates must be built or configured as indicated in the profile to ensure they work correctly, and CSRs are not rejected:

## 8.8 API Keys

API keys are used within the DIP to enforce rate limiting, quota management and analytics this functionality is performed by Azure API Management (APIM).

- API Keys will be allocated at Market Participant Level and not MPID or Role Level.

The API key will need to be provided in the message header field X-API-KEY of all messages.

| Header Value | Details |
|---|---|
| X-API-KEY | Unique key(s) assigned to all DIP User Organisations.<br><br>See swagger definition for further details.<br>https://swagger.io/docs/specification/authentication/api-keys/ |

### 8.8.1 Obtaining API Keys (s).

The *DIP User Portal* will allow signed in users to navigate directly to this part of the APIM Portal to obtain the keys. The APIM portal uses the same sign in credentials as used for the DIP User Portal (SSO)

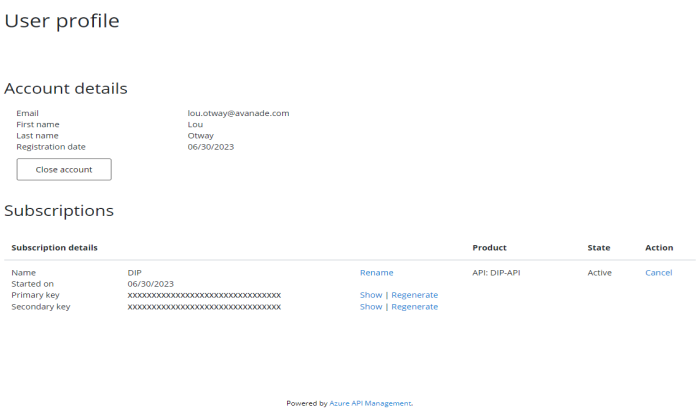



Figure 9- APIM Portal

### 8.8.2 How many API Keys(s) do I need?

2 keys are provided by default (Primary / Secondary) thus allowing API Key(s) to be rotated independently on demand.
Note:

- There is no automatic key rotation.

### 8.8.3 API Key Rotation

API Keys should be rotated at least once a year or immediately after any suspected compromise of an API Key.

- New API keys can be generated in the APIM portal using the “Regenerate” option shown in Figure 9- APIM Portal
- Primary and Secondary Keys can be regenerated independently to ensure service continuity.

Note:

- When a key is regenerated, the old key will immediately cease to work.
- The User Organisation will need to ensure the alternate key has been implemented prior to regenerating an API key.

Deleted: regenerated

### 8.8.4 What happens if I use the wrong key?

If an invalid API key is used a 401 (unauthorised) error will be returned.

### 8.8.5 API Key(s) and Dip Connection Providers

Where a Market Participant has chosen to use a DCP, the Market Participant will be responsible for providing the API Keys(S) to the DCP in a secure manner. See the NCSC guidance for protecting private keys.

# 9 DIP User Portal

The DIP User Portal is a website allowing DIP Service Users access to reporting on audit and monitoring data, manage users and certificates, and perform DIP functions such as replaying messages. It also integrates with the Service Management system where users may raise and track service incidents.

The DIP User Portal will make use of standard TLS to ensure the connection between the client (the user's browser) and the server (the DIP User Portal) is secured.

Users will authenticate with the DIP portal using Azure Active Directory (Azure AD). The following describes the process for a user logging in first time:

- DIP Service User representatives are invited by the DIP manager or their DIP Service User Administrator
- The user receives an email welcoming them to the DIP. Within this email is a redemption link.
- The user clicks upon the link, and they are taken to the DIP User Portal where they are asked to sign-in with their company credentials.
- Upon successful sign-in the user is asked to configure Multi-Factor Authentication (MFA). This involves configuring one or more second factor authentication mechanisms (the first factor being the password):
    - Microsoft authenticator app
    - Mobile phone (text or phone call)
    - Google Authenticator (Chrome extension) (https://chrome.google.com/webstore/detail/authenticator/bhghoamapcdpbohphigoooaddinpkbai)
- The user will then review and accept the terms of use for the DIP User Portal.
- Once accepted the user will be able to access the DIP User Portal

During subsequent logins, the user will have to complete an MFA challenge if this is the first time, they have logged in within 24 hours. They will not have to accept the terms of use again unless the terms are updated.

## 9.1 DIP Roles

### 9.1.1 User Admin

This role within the DIP Portal will be used to register users, assign roles, and create constituent DIP IDs for Market Participants.

Deleted: ID's

### 9.1.2 Message Admin

This role within the DIP User Portal will be used to set up Message Channels and view / replay messages.

### 9.1.3 Certificate Admin

This role within the DIP User Portal will be used to complete the GlobalSign onboarding and can generate / revoke and re-issue certificates.

Deleted: onbaording

Deleted: revokeand

### 9.1.4 Analytics Admin

This role within the DIP User Portal can only review reports.

# 10 Non-Functional Considerations

This section contains a description of the volume profile relating to capacity, bandwidth, and message rates over periods of time measurement, and applicable throughput controls and limits including suppression of “message storms”.

## 10.1 Capacity Management

In the event of Market Participants or DIP Connection Providers acting on behalf of Market Participants who exhibit extreme patterns of data submission then the DIP Manager will be able to implement controls that will limit API access to offending parties.

# 11 Frequently Asked Questions

| Question | Response |
|---|---|
| We already have an SRO/ARO for other services and want to use the same individuals. Can we? | Yes. The same individuals can be nominated for the roles of Certificate Admin in the DIP |
| We are a small organisation and do not have a Head of HR. What should we do? | A Certificate Admin who is a permanent employee of the registering organisation can complete vetting and registration.<br>The certificate admin must ensure they have authority to raise changes against their Organisations Domain Name System (DNS) which is a mandatory requirement to complete the process. |
| When will we be able to request our Certificates? | On completion of Organisational vetting and registration. |
| Is the appointment of a Certificate Admin mandatory? | **Yes**, The Certificate Admin can be an employee of the Market Participant or a 3rd party of the Organisations' choice, such as a DIP Connection Provider. The Market Participant should ensure the 3rd Party has and maintains appropriate security and controls for the management and security of their Private Key and all supporting Certificate requirements. |
| Can we appoint multiple Certificate Admins? | Yes. Some larger organisations, such as the DNOs, may need to split out their organisations based on Market Roles. Or to provide additional cover. |
| We are a DIP Connection Provider that have a group of individuals performing this function for several Market Participants. Can we use a common email to create CSRs and to send/receive Certificates? | **No.** Certificate Admins must use their own unique ID for logging into the DIP User Portal.<br>A Certificate Admin can be nominated by any DIP User Organisations.<br>• For example, a user from a DCP could have been nominated by multiple Market Participant Organisations in to the role of Certificate Admin. |
| How long will it take before we receive confirmation? | If all information in the form is correct, you will be invited to join the DIP by the DIP Manager immediately thereafter. |
| Do all Certificate Admins need to be aware of all activity surrounding the Certificates? | The Certificate Admins, on behalf of the organisation, are responsible for the certs. It is up to the organisation to establish an effective communication process where he/she is kept aware of all activity associated with the certificates, including in the case where there will be a change of PKI Role. |
| Who can request Certificates? | Any one of the Certificate Admins. As such it is important to establish good communication such that duplication of cert requests do not occur. |
| Who do we contact if there is a problem or disagreement? | In the first instance, you should contact the DIP Manager. |

Table 10- FAQs

Deleted: example

# 12 Appendix A – Supplementary Information

## 12.1 Organisational vetting and registration process flow.

### 12.1.1 Vetting

The diagram below shows the process flow for undertaking vetting and registration with GlobalSign via the DIP User Portal.

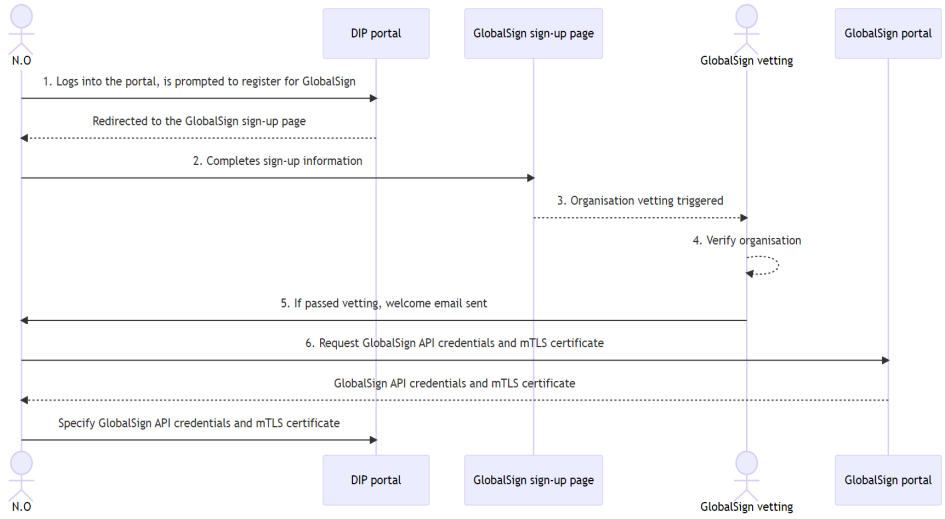



Figure 10 - Organisational vetting process

The following describes the flow:

- The DIP Service User Administrator is invited to join the DIP, the DIP Service User Administrator signs into the DIP User Portal and is prompted to add a Certificate Admin.
- The portal links to the DCA signup form that is specific to the DIP and captures the following data:
  - The organisation details will include the registered address and VAT number.
  - The contact details for the account admin and billing admin. These can be the same person.
- Organisation vetting is triggered.
- This involves verification by the DCA and could result in a phone call where the agent calls the specified number and confirms details with the responsible party.
- If organisational vetting is passed, a new account will be created for the organisation. The account is specific to the organisation of the DIP Service User and will be created irrespective of whether they are already a customer of the DCA.

**Deleted:** he

**Deleted:** details which

**Deleted:** a responsible

### 12.1.2 Registration

A prerequisite to requesting a certificate is to create and validate a domain, a DIP Service User who has been granted the role of Certificate Admin can register the Organisations' domain, the process is outlined below:

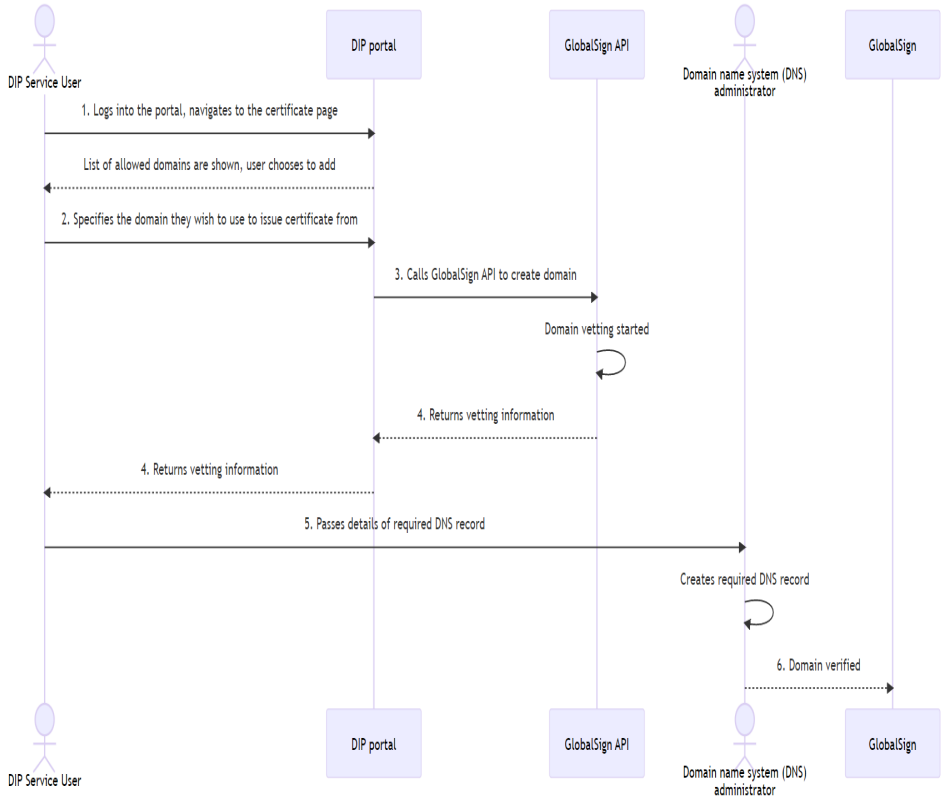



Figure 11 - Domain vetting process

The following describes the flow:

- The Certificate Admin logs into the DIP portal and navigates to the manage certificate screen.
- A list of domains that can be used to issue certificates against is shown. The Certificate Admin specifies the domain name.
    - This triggers the domain validation process. The Certificate Admin will need to ensure the following steps are undertaken:
        - Request a random string is generated and returned in the response to the domain creation request.
        - This string must be added as a TXT record on the DNS for the relevant domain.
        - This is returned to the DIP Service User Administrator
- The Certificate Admin passes the details of the required TXT record to the DNS administrator of the registered organisation, this is person who manages the DNS for the specified domain. They create the DNS record.

**Deleted:** against,

- The DCA picks up the record and the domain is verified.

## 12.2 PKI Certificate Registration process and timescales

| Step | Explanation | Expected Time To Complete | Who |
|---|---|---|---|
| 1 | Login to the DIP User portal and navigate to the certificate management screen. | Immediately available once the user has redeemed the invitation to access the DIP. | Certificate Admin |
| 2 | Complete the information requested to register with GlobalSign. | Immediately available for Nominating Office post login to the DIP | Certificate Admin |
| 3 | GlobalSign will contact the head office of the registering organisation to verify the Nominating Officer is who they say they are (this is Organisational Vetting) | Within 3 days of registration with GlobalSign. | GlobalSign |
| 4 | Once verified access to GlobalSign portal provisioned and user notified | Within an hour of Organisation Vetting completion | GlobalSign |
| 5 | Create GlobalSign API credentials and authentication certificate | Dependant on registering organisation (internal task) | Certificate Admin |
| 6 | Login to DIP User portal and upload GlobalSign API credentials and authentication certificate | Dependant on registering organisation (internal task) | Certificate Admin |
| 7 | Register the DNS domain within DIP User portal | Dependant on registering organisation (internal task) | Certificate Admin |
| 8 | Create TXT record as specified in step 7, within DNS system for domain | Dependant on registering organisation (internal task) | Certificate Admin |
| 9 | Verify the domain within the DIP User portal (checks for TXT record). This is domain vetting | Dependant on registering organisation (internal task) | Certificate Admin |
| 10 | The DIP Service User Administrator can now add the PKI roles to their nominated users. (These are the names of the Certificate Admin provided when registering with the DIP) | Dependant on registering organisation (internal task) | DIP Service User Administrator |
| 11 | Click to create certificate within the DIP User portal certificate management section. Select mTLS, signing or mTLS & signing certificate (dependant on MPO type) and verified domain (if more than 1). Expected certificate name displayed and copied. | Dependant on registering organisation (internal task) | Certificate Admin |
| 12 | Generate CSR with the specified certificate name on the same domain. This generates the private key and unsigned public key | Dependant on registering organisation (internal task) | Certificate Admin |
| 13 | Paste CSR text into portal, portal verifies CSR and sends to GlobalSign on behalf of the MPO (using the credentials and certificate specified in step 6) | Dependant on registering organisation (internal task) | Certificate Admin |

**Formatted:** Highlight

**Formatted:** Strikethrough

**Deleted:** the SRO

| | | | |
|---|---|---|---|
| 14 | GlobalSign signs the CSR, this is the signed public key. | Within 10 seconds of request | GlobalSign |
| 15 | Certificate Admin able to view the signed public key from the DIP User portal (string of text) | Dependant on registering organisation (internal task) | Certificate Admin |
| 16 | Create a .cer file and paste the public signed key into the file | Dependant on registering organisation (internal task) | Certificate Admin |
| 17 | Fulfil the certificate using the .cer file on the server/service used to generate the CSR (step 11). This completes the key pair | Dependant on registering organisation (internal task) | Certificate Admin |

## 12.3 Mutual TLS (mTLS) process flows

### 12.3.1 mTLS Ingress (API).

The below sequence diagram details the process flow for establishing mTLS between the DIP Service User and the DIP:

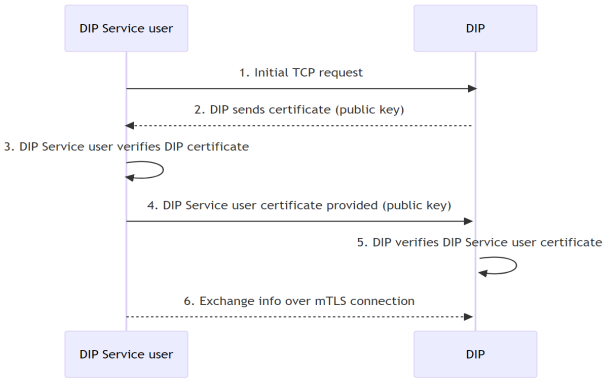



Figure 12- mTLS ingress process flow

1. The DIP Service User makes the handshake TCP request to the DIP ingress endpoint.
2. The DIP returns its certificate, this is the public key.
3. The DIP Service User verifies the properties of the certificate sent by the DIP:
    a) The domain name of the issuing Certificate Authority (CA)
    b) The domain name of the cert
    c) The certificate chain.
    d) The expiry date of the certificate
    e) Whether the certificate has been revoked
4. The DIP Service User sends its certificate, public key.
5. The DIP will verify the properties of the certificate sent by the DIP Service User:
    a) The domain name of the issuing Certificate Authority (CA)

b) The subject name of the certificate.

c) The certificate chain.

d) The expiry date of the certificate

e) Whether the certificate has been revoked

6. The DIP sends messages to the DIP Service User (encrypting the traffic using the provided public cert, The DIP Service User decrypts using its private key.
7. The DIP Service User sends messages to the DIP encrypting the traffic using the provided DIP public cert, DIP decrypts using its private key.
8. mTLS has been achieved.

### 12.3.2 mTLS Egress (webhook)

The below sequence diagram details the process flow for establishing mTLS between the DIP and DIP Service Users registered URL (webhook).

Deleted: registred

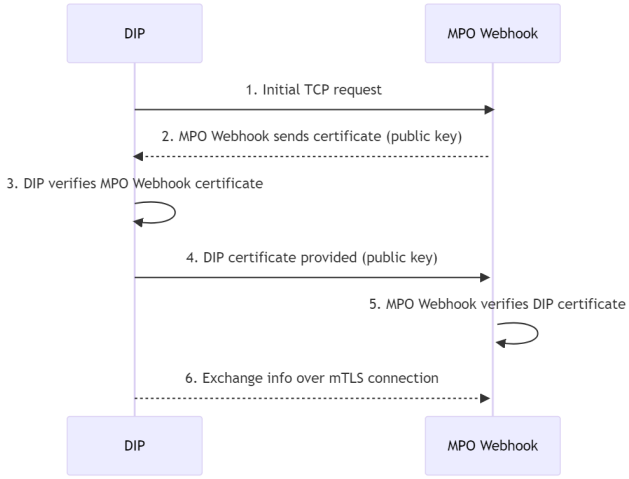



Figure 13- mTLS egress process flow

1. The DIP Service makes the handshake TCP request to the DIP Service Users registered URL (webhook).
2. The DIP Service User returns its certificate, this is the public key.
3. The DIP Service verifies the properties of the certificate sent by the DIP Service User:
    a) The domain name of the issuing Certificate Authority (CA)
    b) The domain name of the cert
    c) The certificate chain.
    d) The expiry date of the certificate

e) Whether the certificate has been revoked

4. The DIP Service sends its certificate, public key.
5. The DIP Service User will verify the properties of the certificate sent by the DIP Service:
    a) The domain name of the issuing Certificate Authority (CA)
    b) The subject name of the certificate.
    c) The certificate chain.
    d) The expiry date of the certificate
    e) Whether the certificate has been revoked
6. The DIP Service User sends messages to the DIP Service (encrypting the traffic using the provided public cert, DIP Service decrypts using its private key.
7. The DIP Service sends messages to the DIP Service User encrypting the traffic using the provided DIP Service Users public cert, DIP Service User decrypts using its private key.
8. mTLS has been achieved.

## 12.4 Message signing

Message signing is the process to confirm that message contents haven't been tampered with, and that the message is from the sender, it is a non-repudiation control. Message signing will be required for messages sent to the DIP from all DIP Service Users (data ingress endpoint) and messages sent out from the DIP (data egress endpoints), responses to API calls will not be signed (e.g. if the DIP Service User sends a message in an incorrect format then a 400 bad request response would be received, this response would not be signed).

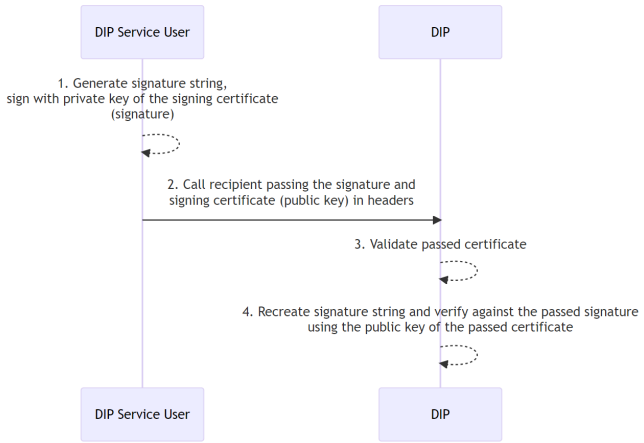



Figure 14 - Message signing ingress flow

1. The DIP Service User generates the signature string and encrypts using the private key of their client certificate, this is the signature.
2. The DIP Service User passes the signature as a header in their request to the DIP ingress API. The public key of the client certificate is also passed as a header in the request.
3. The signature is decrypted using the public key cert sent as part of the request, this is the decrypted signature.

4. The DIP validates the decrypted signature. If passed than the signature is valid.

## 12.5 Certificate Signing Request process flow.

Deleted: flow

To request a new certificate the Certificate Admin will use the DIP User Portal to provide a Certificate Signing Request (CSR), the signing will be fulfilled by GlobalSign.

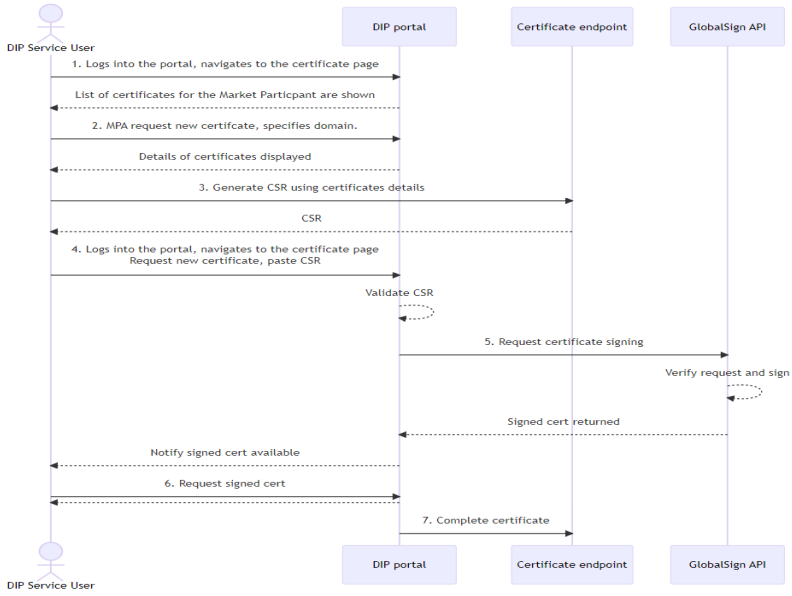



Figure 15 - Certificate signing process flow.

The following describes the flow:

- From within the DIP User Portal, the Certificate Admin navigates to the certificates page where a list of certificates is shown.
- The Certificate Admin clicks to request a new certificate, selects a registered domain, details of the certificate to request are shown:
- The Certificate Admin uses the details to create the CSR using the tools detailed in Table 9- CSR Guidance. Generating the CSR creates the private key of the certificate and the unsigned public key.
- Once the CSR has been generated, the Certificate Admin will paste it into the DIP portal. The CSR will be validated to ensure that the values specified match the details of the certificate (shown to the Certificate Admin previously).
- The Certificate Admin downloads the signed certificate value, then completes the certificate request on the same server that they used to request the certificate from. This step will sign the public key and complete the process.

## 12.6 Example code

**DIP Certificate Authority (DCA) – Root Certificate**

Each DIP Server User Organisation will need to download the DCA root certificate. This will be available directly from GlobalSign and can be found here https://support.globalsign.com/ca-certificates/root-certificates/globalsign-root-certificates

**Note:** In the example code (Certificate Validation), the DCA root certificate was first converted into .cer file and the text output used in the code.

**Note:** This is example code written in Microsoft C#. This code will ***NOT*** be supported and is for reference purposes only.

#### 12.6.1 Message Signing.

```
using System.Collections;
using System.Globalization;
using System.Security.Cryptography;
using System.Security.Cryptography.X509Certificates;
using System.Text;

namespace Util
{
   public class ContentSignatureHelper
   {
      /// <summary>
      /// The signature header
      /// </summary>
      public const string signatureHeader = "X-DIP-Signature";

      /// <summary>
      /// The signature date header
      /// </summary>
      public const string signatureDateHeader = "X-DIP-Signature-Date";


      /// <summary>
      /// The signature content hash header
      /// </summary>
      public const string signatureContentHashHeader = "X-DIP-Content-Hash";

      /// <summary>
      /// The signature certificate header
      /// </summary>
      public const string signatureCertificateHeader = "X-DIP-Signature-Certificate";


      /// <summary>
      /// Creates the signature headers required for the DIP based on the parameters passed
      /// </summary>
      /// <param name="messageContent"></param>
      /// <param name="verb"></param>
      /// <param name="recipientUrl"></param>
```

```
/// <param name="signingCertificate"></param>
/// <returns></returns>
/// <exception cref="ArgumentNullException"></exception>
/// <exception cref="ArgumentException"></exception>
public static IDictionary<string, string> GenerateSignatureHeaders(string messageContent,
                                                    HttpMethod verb,
                                                    Uri recipientUrl,
                                                    X509Certificate2 signingCertificate)
{
   if (messageContent == null || string.IsNullOrEmpty(messageContent))
   {
      throw new ArgumentNullException(messageContent);
   }

   if (signingCertificate == null)
   {
      throw new ArgumentNullException("signingCertificate");
   }

   if (!signingCertificate.HasPrivateKey)
   {
      throw new ArgumentException("Passed signingCertificate does not have a private key to sign with");
   }

   var contentHash = GetContentHash(messageContent);

   // Signature date with ISO 8601 format
   string signatureDate = DateTime.UtcNow.ToString("o", CultureInfo.InvariantCulture);

   // Signature string
   string signatureString = $"{verb};{recipientUrl};{signatureDate};{contentHash}";

   // Sign the signature string
   var signedString = GenerateSignature(signingCertificate, signatureString);

   var headers = new Dictionary<string, string>
   {
      { signatureHeader, signedString },
      { signatureDateHeader, signatureDate },
      { signatureContentHashHeader, contentHash },
      { signatureCertificateHeader, Convert.ToBase64String(signingCertificate.Export(X509ContentType.Cert))}
   };

   return headers;
}

/// <summary>
/// Gets the content hash
/// </summary>
/// <param name="content"></param>
/// <returns></returns>
/// <exception cref="ArgumentNullException"></exception>
```

```
    public static string GetContentHash(string content)
    {
      if (string.IsNullOrEmpty(content))
      {
        throw new ArgumentNullException("content");
      }

      using SHA256 sHA = SHA256.Create();
      return Convert.ToBase64String(sHA.ComputeHash(Encoding.UTF8.GetBytes(content)));
    }

    private static string GenerateSignature(X509Certificate2 cert, string signatureString)
    {
      byte[] signingSignature = Array.Empty<byte>();

      RSA rsa = cert.GetRSAPrivateKey();
      signingSignature = rsa.SignHash(SHA256.HashData(Encoding.UTF8.GetBytes(signatureString)),
HashAlgorithmName.SHA256, RSASignaturePadding.Pkcs1);

      // Base64Url encryption of the signed signature
      return Convert.ToBase64String(signingSignature);
    }
  }
}
```

### 12.6.2 Certificate validation

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Cryptography.X509Certificates;
using System.Text;
using System.Threading.Tasks;

namespace Util
{
  public class CertificateHelper
  {
    /// <summary>
    /// Certificate used to validate the certificate returned by the recipient has been issued by our CA
    /// </summary>
    private static X509Certificate2Collection _rootCACertificates;

    /// <summary>
    /// Validates the chain of an cert against the trusted root
    /// </summary>
    /// <param name="certificate">(X509Certificate2) The leaf certificate</param>
    /// <param name="chain">(X509Chain) The chain of the leaf certificate</param>
    /// <param name="issuingCA">(string) Cert of the issuing CA</param>
    /// <returns>(bool) Whether the certificate is valid and has a valid chain</returns>
    /// <exception cref="Exception">Chain status (validation) errors</exception>
```

```
        public static bool ValidateChain(X509Certificate2 certificate, X509Chain chain, string issuingCA, bool
checkRevocationStatus = true)
        {
            // If the issuing CA cert is null get it from the local store
            if (_rootCACertificates == null || _rootCACertificates.Count == 0)
            {
                _rootCACertificates = new X509Certificate2Collection
                {
                    new X509Certificate2(rawData: Encoding.UTF8.GetBytes(issuingCA))
                };
            }

            // Rather than use the system trust CAs use our trusted root
            chain.ChainPolicy.CustomTrustStore.Clear();
            chain.ChainPolicy.TrustMode = X509ChainTrustMode.CustomRootTrust;
            chain.ChainPolicy.CustomTrustStore.AddRange(_rootCACertificates);

            // Will check the status of all certs in the chain using OCSP or CRL depending on what is available
            chain.ChainPolicy.RevocationFlag = X509RevocationFlag.EntireChain;
            // Means the check will use a cache if possible
            chain.ChainPolicy.RevocationMode = X509RevocationMode.Offline;
            chain.ChainPolicy.VerificationTime = DateTime.Now;

            if (checkRevocationStatus)
            {
                chain.ChainPolicy.VerificationFlags = X509VerificationFlags.NoFlag;
            }
            else
            {
                chain.ChainPolicy.VerificationFlags = X509VerificationFlags.IgnoreEndRevocationUnknown;
            }

            // Build the certificate chain, performing the revocation check
            bool valid = chain.Build(certificate);

            if (!valid)
            {
                var chainStatus = chain.ChainStatus.Select(s => s.StatusInformation).ToArray();
                throw new Exception($"Chain status invalid the following flags have been identified: {String.Join(" ",
chainStatus)}");
            };

            return valid;
        }

        /// <summary>
        /// Validate a certificate in terms of whether it has expired and has the correct subject prefix name
        /// </summary>
        /// <param name="certificate"></param>
        /// <returns></returns>
        public static bool ValidateCertificate(X509Certificate2 certificate, string environmentPrefix = "")
        {
            DateTime now = DateTime.Now;
```

```
        if (now < certificate.NotBefore || now > certificate.NotAfter)
        {
            throw new Exception($"mTLS certificate timestamp is not valid");
        }

        // If the environment prefix is null or empty or the subject contain the environment prefix then continue
        if (!certificate.SubjectName.Name.Contains(environmentPrefix, StringComparison.InvariantCultureIgnoreCase))
        {
            throw new Exception($"mTLS certificate subject name {certificate.SubjectName.Name} does not contain the
environment prefix of {environmentPrefix}");
        }

        return true;
    }
  }
}
```

Avanade used a call back in egress to trigger this:

```
 /// <summary>
        /// Validates the returned mTLS cert from the server
        /// The validity of a cert on a (public) url can be ascertained here: https://globalsign.ssllabs.c
om/
        /// </summary>
        /// <param name="sender">Server that responded</param>
        /// <param name="certificate">The certificate sent</param>
        /// <param name="chain">The chain of the certificate sent</param>
        /// <param name="sslPolicyErrors">Any SSL errors when calling the server</param>
        /// <returns>(bool) Whether the certificate is valid</returns>
        private bool ServerCertificateValidationCallback(object sender, X509Certificate2 certificate, X509
Chain chain, SslPolicyErrors sslPolicyErrors)
        {
            try
            {
                // Verifies that the FQDN matches the domain name of the certificate
                if (sslPolicyErrors != SslPolicyErrors.None && sslPolicyErrors != SslPolicyErrors.RemoteCe
rtificateChainErrors)
                {
                    var chainStatus = chain.ChainStatus.Select(s => s.StatusInformation).ToArray();
                    throw new Exception($"Certificate SslPolicy errors: {sslPolicyErrors}. Chain status:
{String.Join(" ", chainStatus)}");
                }

                if (CertificateHelper.ValidateCertificate(certificate, _environmentPrefix))
                {
                    return CertificateHelper.ValidateChain(certificate, chain, Environment.GetEnvironmentV
ariable("IssuingCA"),checkRevocationStatus: Convert.ToBoolean(Environment.GetEnvironmentVariable("CheckRev
ocationStatus")));
                }

                return false;
            }
            catch (Exception ex)
            {
```

```
                _logger.LogError($"The following error was found in recipient certificate validation: {ex.
Message}");
                return false;
            }
        }
    }
```

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

Page 31: [1] Deleted Kevan Gleeson (MHHSProgramme) 25/10/2023 22:09:00

Page 32: [2] Deleted Kevan Gleeson (MHHSProgramme) 25/10/2023 22:09:00