Table of Contents

[1 Introduction 3](#_Toc137215538)

[1.1 Scope 3](#_Toc137215539)

[1.2 Conformance 3](#_Toc137215540)

[1.3 Terms and Definitions 3](#_Toc137215541)

[2 Architecture Overview 3](#_Toc137215542)

[2.1 Architecture Diagram 4](#_Toc137215543)

[3 Data Model 4](#_Toc137215544)

[4 Registry Functions 5](#_Toc137215545)

[4.1 Between Registrars and Registration Agents/Participants 6](#_Toc137215546)

[4.2 Between Registrars 6](#_Toc137215547)

[5 Registrar Management 6](#_Toc137215548)

[5.1 Registrar Identifiers 6](#_Toc137215549)

[6 Registry Architecture Options 6](#_Toc137215550)

[6.1 Evaluation Matrix 6](#_Toc137215551)

[6.2 Evaluation Conclusions 7](#_Toc137215552)

[7 Major and Minor Amendments 8](#_Toc137215553)

# Introduction

## Scope

This document outlines the requirements for the SML registry architecture. These requirements are based on the policies laid out in the [Exchange Framework Registration Policy v1.0](https://docs.google.com/document/u/0/d/1Eoo67tawuo9TtuYjjub2z4p3mCdIs_1J/edit) as they will be implemented by registrars in a multi-registrar environment.

## Conformance

The keywords ‘MUST’, ‘MUST NOT’, ‘REQUIRED’, ‘SHALL’, ‘SHALL NOT’, ‘SHOULD’, ‘SHOULD NOT’, ‘RECOMMENDED’, ‘MAY’, and ‘OPTIONAL’ in this specification are to be interpreted as described in RFC2119 and RFC 8174 when, and only when, they appear in all capitals, as shown here.

## Terms and Definitions

For the purpose of this specification, all terms shall have the definitions defined in section 2.3 of the E-invoice Exchange Framework – Approach to Managing a Federated Registry Services Model in a Four-Corner Network report found here: [https://businesspaymentscoalition.org/wp-content/uploads/bpc-e-delivery-netwo rk-validation-exercise-2020.pdf](https://businesspaymentscoalition.org/wp-content/uploads/bpc-e-delivery-netwo%20rk-validation-exercise-2020.pdf)

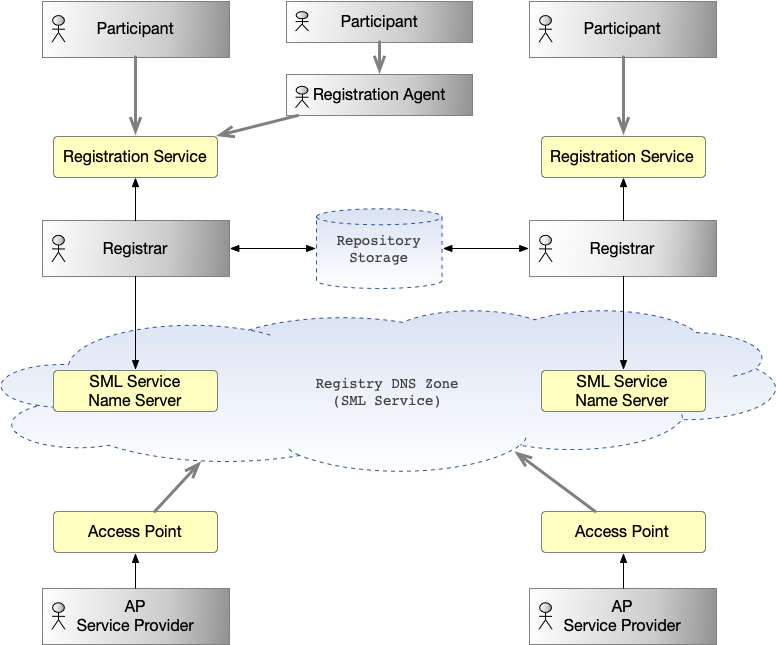
# Architecture Overview

In the registration framework depicted in section 2.1, participants either interface directly with a Registrar for Registration Services, or delegate the registration process to a registration agent who coordinates the registration with the registrar on behalf of the participant.

Registrars enter and maintain all participant data in the SML Registry, and facilitate participant discovery through the SML Service.

Definitions of each role in the registration process can be found in section 2.1.2 of the [Exchange Framework Registration Policy](https://docs.google.com/document/d/1Eoo67tawuo9TtuYjjub2z4p3mCdIs_1J/edit).

## Architecture Diagram



# Data Model

The following section represents the information elements included in a registration record. Note that all demographic reporting will be left out of the registry and done instead by surveying registrars.

| **Level** | **Element** | **Cardinality** | **Definition** |
| --- | --- | --- | --- |
|  | *RegistrationRecord* | 1..1 | Root of the registration record |
| L | ParticipantID | 1..1 | The Participant Identifier as specified in the BPC Identifier Policy specification. |
| L | Service Lookup Information | 0..n | Metadata to lookup the Service Metadata Publisher associated with the participant. |
| LL | AssociatedSMPURL | 1..1 | The URL of the participant’s Service Metadata Publisher. |
| LL | ServiceName | 1..1 | This field describes which type of service the record points to. The value of the service field MUST be an identifier  defined in a network service profile or specification. For example, the identifier for the OASIS SMP 2.0 service is defined in the BPC SMP Profile specification.  A participant MUST NOT be registered with more than one record having the same value in the service field.  The value of the service field MUST be treated as case insensitive. |
| L | Change Log Record | 1..n | Record of a change to a registry record. |
| LL | RegistrarID | 1..1 | A unique identifier for the registrar as described in section 5.1. |
| LL | ReferenceNumber | 1..1 | A unique identifier for the change, generated by the registrar. |
| LL | ChangeDescription | 1..1 | A short description of what has changed. |
| LL | ChangedDateTime | 1..1 | The date and time (UTC) on which the change occurred. |
| LL | ChangeSoliciterContactPerson | 1..1 | The name of the requestor (participant or registration agent) for the most recent change. |
| LL | RegistrationAgentName | 0..1 | The name of the agent who initiated the change request. |
| LL | AuthenticationTrustLevel | 1..1 | A description of the procedures used to verify that the participant owns the registered identifier. |

# Registry Functions

This section defines operations that must be possible between registration actors. A policy for how to execute each function is out of scope for this subcommittee. It is up to registrars themselves how to implement/execute these functions.

## Between Registrars and Registration Agents/Participants

* Registration of participant
* View of registrations
* Change/update registration
* Delete registration

## Between Registrars

* Creation of registration record
* Retrieval of a registration record
* Change/update registration record
* Delete registration record

# Registrar Management

The corporate will manage registrars on the network through the following functions:

* Provision a registrar in the network
* Management of registrar credentials
* Decommission registrar
* Monitor performance of the network
* This will depend on the selected technology and the policy on registrar expectations.

## Registrar Identifiers

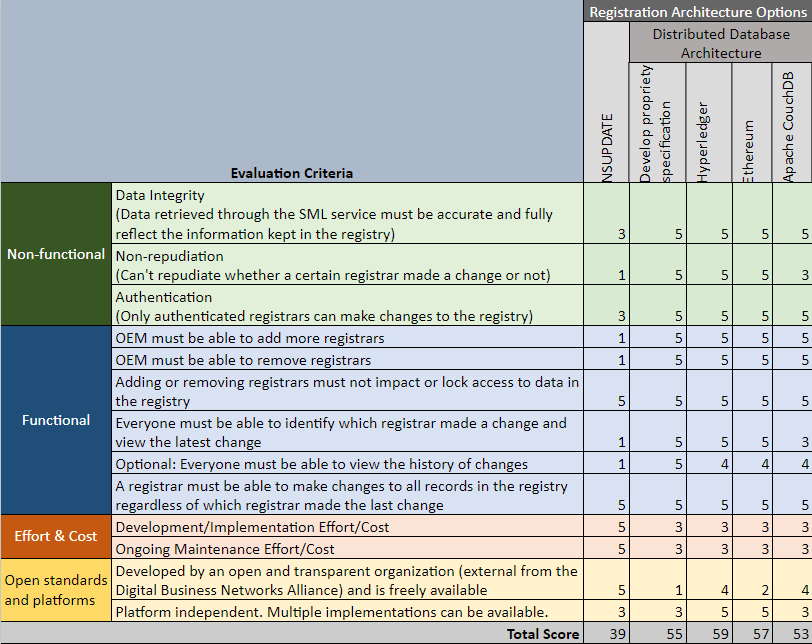
In order to be uniquely identified in the network, registrars must have an associated identifier. The definition of identifiers are outlined in the DBNA Identifier Policy and not specified in this document.

* RegistrarIDs MUST be a valid Business Identifier as defined in the DBNA Identifier Policy.
* A Registrar MUST only be identified with one RegistrarID.

# Registry Architecture Options

## Evaluation Matrix

The following matrix was used by the subcommittee to evaluate four architecture options against a set of predefined requirements. The options included NSUPDATE, which was used in the BPC validation exercise completed in 2021 and four distributed database architecture options, including Hyperledger, Ethereum, Apache CouchDB and a proprietary specification developed by the oversight entity.



## Evaluation Conclusions

The subcommittee eliminated NSUPDATE based on the evaluation but determined that the four distributed database options were all viable solutions, but do not represent all feasible technologies.

# Major and Minor Amendments

For purposes of Section 3.14 of the Bylaws of the Digital Business Networks Alliance, Inc. any amendment to this Technical Specifications Exhibit other than to correct typographical or clerical errors shall be considered a “Major Amendment.” Minor amendments to correct typographical or clerical errors may be approved by a committee established by the Board of Directors of the DBNAlliance. Major Amendments must be approved by the Full Members of DBNAlliance that qualify as Service Providers.