

Workgroup: Network Working Group

Internet-Draft:

draft-hallambaker-mesh-protocol

Published: 13 January 2021

Intended Status: Informational

Expires: 17 July 2021

Authors: P. M. Hallam-Baker

ThresholdSecrets.com

Mathematical Mesh 3.0 Part V: Protocol Reference

Abstract

The Mathematical Mesh 'The Mesh' is an end-to-end secure infrastructure that facilitates the exchange of configuration and credential data between multiple user devices. The core protocols of the Mesh are described with examples of common use cases and reference data.

[Note to Readers]

Discussion of this draft takes place on the MATHMESH mailing list (mathmesh@ietf.org), which is archived at https://mailarchive.ietf.org/arch/search/?email_list=mathmesh.

This document is also available online at <http://mathmesh.com/Documents/draft-hallambaker-mesh-protocol.html>.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 17 July 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

- [1. Introduction](#)
- [2. Definitions](#)
 - [2.1. Requirements Language](#)
 - [2.2. Defined Terms](#)
 - [2.3. Related Specifications](#)
 - [2.4. Implementation Status](#)
- [3. Mesh Protocols](#)
- [4. Mesh Service](#)
 - [4.1. Data Model](#)
 - [4.2. Partitioning](#)
- [5. Protocol Bindings](#)
 - [5.1. DNS Web Service Discovery](#)
 - [5.2. Web Service Protocol Binding](#)
 - [5.2.1. Transport Security](#)
 - [5.2.2. HTTP Message Binding](#)
 - [5.2.3. Request](#)
 - [5.2.4. Response](#)
 - [5.3. DARE Message Encapsulation](#)
 - [5.3.1. Null Authentication](#)
 - [5.3.2. Device Authentication](#)
 - [5.3.3. Profile Authentication](#)
 - [5.3.4. Ticket Authentication](#)
 - [5.4. Payload Encoding](#)
 - [5.5. Error handling and response codes](#)
- [6. Mesh Service Transactions](#)
 - [6.1. Service Description](#)
 - [6.2. Account Creation](#)
 - [6.2.1. Bind User Account](#)
 - [6.2.2. Bind Group Account](#)
 - [6.2.3. Unbind Account](#)
 - [6.3. Persistence Store Management](#)
 - [6.3.1. Status](#)
 - [6.3.2. Download](#)
 - [6.3.3. Conflict Detection](#)
 - [6.3.4. Filtering](#)
 - [6.3.5. Transact](#)
 - [6.4. Messaging](#)
 - [6.5. Publication](#)
 - [6.5.1. Claim Transaction](#)
 - [6.5.2. PollClaim Transaction](#)

- [6.6. Cryptographic](#)
 - [6.6.1. Generate Key Shares](#)
 - [6.6.2. Key Agreement](#)
 - [6.6.3. Sign](#)
- [7. Message Interactions](#)
 - [7.1. Message PIN Interaction](#)
 - [7.1.1. Registration](#)
 - [7.1.2. Authentication](#)
 - [7.1.3. Validation](#)
 - [7.1.4. Example](#)
 - [7.2. Completion Interaction](#)
 - [7.3. Contact Exchange Interaction](#)
 - [7.3.1. Remote](#)
 - [7.3.2. PIN](#)
 - [7.3.3. EARL](#)
 - [7.4. Group Invitation](#)
 - [7.5. Confirmation Interaction](#)
- [8. Device Connection Interactions](#)
 - [8.1. Dynamic QR Code \(PIN\) Authenticated](#)
 - [8.1.1. Phase 1:](#)
 - [8.1.2. Phase 2:](#)
 - [8.1.3. Phase 3:](#)
 - [8.1.4. Phase 4](#)
 - [8.2. Witness Authenticated](#)
 - [8.2.1. Phase 1:](#)
 - [8.2.2. Phase 2:](#)
 - [8.2.3. Phase 3:](#)
 - [8.2.4. Phase 4](#)
 - [8.3. Static QR Code Authenticated](#)
 - [8.3.1. Phase 1](#)
 - [8.3.2. Phase 2](#)
 - [8.3.3. Phase 3](#)
 - [8.3.4. Phase 4](#)
- [9. Protocol Schema](#)
 - [9.1. Request Messages](#)
 - [9.1.1. Message: MeshRequest](#)
 - [9.1.2. Message: MeshRequestUser](#)
 - [9.2. Response Messages](#)
 - [9.2.1. Message: MeshResponse](#)
 - [9.3. Imported Objects](#)
 - [9.4. Common Structures](#)
 - [9.4.1. Structure: KeyValue](#)
 - [9.4.2. Structure: ConstraintsSelect](#)
 - [9.4.3. Structure: ConstraintsData](#)
 - [9.4.4. Structure: PolicyAccount](#)
 - [9.4.5. Structure: ContainerStatus](#)
 - [9.4.6. Structure: ContainerUpdate](#)
 - [9.5. Transaction: Hello](#)
 - [9.5.1. Message: MeshHelloResponse](#)

- [9.6. Transaction: BindAccount](#)
 - [9.6.1. Message: BindRequest](#)
 - [9.6.2. Message: BindResponse](#)
- [9.7. Transaction: UnbindAccount](#)
 - [9.7.1. Message: UnbindRequest](#)
 - [9.7.2. Message: UnbindResponse](#)
- [9.8. Transaction: Connect](#)
 - [9.8.1. Message: ConnectRequest](#)
 - [9.8.2. Message: ConnectResponse](#)
- [9.9. Transaction: Complete](#)
 - [9.9.1. Message: CompleteRequest](#)
 - [9.9.2. Message: CompleteResponse](#)
- [9.10. Transaction: Status](#)
 - [9.10.1. Message: StatusRequest](#)
 - [9.10.2. Message: StatusResponse](#)
- [9.11. Transaction: Download](#)
 - [9.11.1. Message: DownloadRequest](#)
 - [9.11.2. Message: DownloadResponse](#)
- [9.12. Transaction: Transact](#)
 - [9.12.1. Message: TransactRequest](#)
 - [9.12.2. Message: TransactResponse](#)
 - [9.12.3. Structure: EntryResponse](#)
- [9.13. Transaction: Post](#)
 - [9.13.1. Message: PostRequest](#)
 - [9.13.2. Message: PostResponse](#)
- [9.14. Transaction: Claim](#)
 - [9.14.1. Message: ClaimRequest](#)
 - [9.14.2. Message: ClaimResponse](#)
- [9.15. Transaction: PollClaim](#)
 - [9.15.1. Message: PollClaimRequest](#)
 - [9.15.2. Message: PollClaimResponse](#)
 - [9.15.3. Structure: CryptographicOperation](#)
 - [9.15.4. Structure: CryptographicOperationSign](#)
 - [9.15.5. Structure: CryptographicOperationKeyAgreement](#)
 - [9.15.6. Structure: CryptographicOperationGenerate](#)
 - [9.15.7. Structure: CryptographicOperationShare](#)
 - [9.15.8. Structure: CryptographicResult](#)
 - [9.15.9. Structure: CryptographicResultKeyAgreement](#)
 - [9.15.10. Structure: CryptographicResultShare](#)
- [9.16. Transaction: Operate](#)
 - [9.16.1. Message: OperateRequest](#)
 - [9.16.2. Message: OperateResponse](#)
- [10. Security Considerations](#)
- [11. IANA Considerations](#)
- [12. Acknowledgements](#)
- [13. Normative References](#)
- [14. Informative References](#)

1. Introduction

This document describes the Mesh Service protocol supported by Mesh Services, an account-based protocol that facilitates exchange of data between devices connected to a Mesh profile and between Mesh accounts.

Mesh Service Accounts support the following services:

- *Provides the master persistence store for the Catalogs and Spools associated with the account.
- *Enables synchronization of Catalogs and Spools with connected devices.
- *Enforces access control on inbound Mesh Messages from other users and other Mesh Services.
- *Authenticates outbound Mesh Messages, certifying that they comply with abuse mitigation policies.

A Mesh Profile **MAY** be bound to multiple Mesh Service Accounts at the same time but only one Mesh Service Account is considered to be authoritative at a time. Users may add or remove Mesh Service Accounts and change the account designated as authoritative at any time.

The Mesh Services are build from a very small set of primitives which provide a surprisingly extensive set of capabilities. These primitives are:

Hello Describes the features and options provided by the service and provides a 'null' transaction which **MAY** be used to establish an authentication ticket without performing any action,

CreateAccount, DeleteAccount Manage the creation and deletion of accounts at the service.

Status, Download, Upload Support synchronization of Mesh containers between the service (Master) and the connected devices (Replicas).

Connect Initiate the process of connecting a device to a Mesh profile from the device itself.

Post Request that a Mesh Message be transferred to one or more Mesh Accounts.

Although these functions could in principle be used to replace many if not most existing Internet application protocols, the principal

value of any communication protocol lies in the size of the audience it allows them to communicate with. Thus, while the Mesh Messaging service is designed to support efficient and reliable transfer of messages ranging in size from a few bytes to multiple terabytes, the near-term applications of these services will be to applications that are not adequately supported by existing protocols if at all.

2. Definitions

This section presents the related specifications and standard, the terms that are used as terms of art within the documents and the terms used as requirements language.

2.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [\[RFC2119\]](#).

2.2. Defined Terms

The terms of art used in this document are described in the *Mesh Architecture Guide* [\[draft-hallambaker-mesh-architecture\]](#).

2.3. Related Specifications

The architecture of the Mathematical Mesh is described in the *Mesh Architecture Guide* [\[draft-hallambaker-mesh-architecture\]](#). The Mesh documentation set and related specifications are described in this document.

2.4. Implementation Status

The implementation status of the reference code base is described in the companion document [\[draft-hallambaker-mesh-developer\]](#).

3. Mesh Protocols

The Mesh specifies two separate types of protocol interactions:

Mesh Service Protocol A synchronous protocol supporting interactions between devices and a Mesh Service Host and between Mesh Service hosts.

Mesh Messaging Protocol An asynchronous protocol that supports interactions between devices connected to the same account and between accounts.

The Mesh Messaging Protocol uses the Mesh Service Protocol as transport. The Mesh Service Protocol in turn is supported by either the HTTPS binding over TCP or by the Mesh Datagram binding over UDP.

Mesh Messaging Protocol	
Mesh Service Protocol	
HTTPS	Mesh Datagram
TLS	
TCP	UDP

Figure 1: Protocol Layering

Mesh Services **MUST** support the HTTPS binding and **MAY** support the Mesh Datagram binding.

4. Mesh Service

A Mesh Service is a minimally trusted service. In particular a user does not need to trust a Mesh service to protect the confidentiality or integrity of most data stored in the account catalogs and spools.

Unless the use of the Mesh Service is highly restricted, a user does need to trust the Mesh Service in certain respects:

Data Loss A service could refuse to respond to requests to download data.

Integrity (Stale Data) The use of Merkle Trees limits but does not eliminate the ability of a Mesh Service to respond to requests with stale data.

Messaging A service could reject requests to post messages to or accept messages from other mesh users.

This risk is a necessary consequence of the fact that the Mesh Service Provider is accountable to other Mesh Service Providers for abuse originating from their service.

Traffic analysis A Mesh Service has knowledge of the number of Mesh Messages being sent and received by its users and the addresses to which they are being sent to or received from.

The need to trust the Mesh Service in these respects is mitigated by accountability and the user's ability to change Mesh Service providers at any time they choose with minimal inconvenience.

It is possible that some of these risks will be reduced in future versions of the Mesh Service Protocol but it is highly unlikely that these can be eliminated entirely without compromising practicality or efficiency.

4.1. Data Model

The design of the Mesh Service model followed a quasi-formal approach in which the system was reduced to schemas which could in principle be rendered in a formal development method but without construction of proofs.

Like the contents of Mesh Accounts, a Mesh Service may be represented by a collection of catalogs and spools, for example:

Account Catalog Contains the account entries.

Incident Spool Reports of potential abuse

Backup of the service **MAY** be implemented using the same container synchronization mechanism used to synchronize account catalogs and spools.

4.2. Partitioning

Mesh Services supporting a large number of accounts or large activity volume **MAY** partition the account catalog between one or more hosts using the usual tiered service model in which a front-end server receives traffic for any account hosted at the server and routes the request to the back-end service that provides the persistence store for that account.

In addition, the Mesh Service Protocol supports a 'direct connection' partitioning model in which devices are given a DNS name which **MAY** allow for direct connection to the persistence host or to a front-end service offering service that is in some way specific to that account.

5. Protocol Bindings

Mesh Service transactions are mapped to an underlying messaging and transport protocol. The following binding

Mesh Services **MUST** support the Web Service binding specified in this document and **MAY** support the UDP binding currently in development.

5.1. DNS Web Service Discovery

The DNS Web Service discovery mechanism is used to discover Mesh Services regardless of the protocol binding .The service name, DNS prefix and and .well-known service suffix are specified as follows:

*Service Name: mmm

*DNS Prefix: _mmm._tcp

*Well Known service suffix: /.well-known/mmm

5.2. Web Service Protocol Binding

The Web Service Protocol binding makes use of the most widely deployed and used protocols:

*Discovery: DNS Service discovery

*Transport: TLS

*Application: HTTP

*Presentation: DARE Message

*Encoding: JSON, JSON-B

The chief limitations of the Web Service Protocol Binding are that the use of TCP based transport results in unsatisfactory latency for some applications and that the HTTP application layer only serves to allow a host to support multiple services on the same TCP/IP port.

5.2.1. Transport Security

Mesh Services **MUST** offer TLS transport and **MAY** offer non TLS transport. MESH clients **SHOULD** use TLS transport when connecting to a MESH service.

TLS version 1.3 [[RFC8446](#)] or higher **MUST** be supported. Client authentication **SHOULD NOT** be used.

5.2.2. HTTP Message Binding

All messages are exchanged as HTTP POST transactions. Support for and use of HTTP/1.1 [[RFC7230](#)] is **REQUIRED**. Services **MAY** support HTTP/2.

In contrast to other approaches to the design of Web Services, the only use made of the HTTP transport is to distinguish between different services on the same host using the Host header and .well-

known convention and for message framing. No use is made of the URI request line to identify commands, nor are the caching or proxy capabilities of HTTP made use of.

5.2.3. Request

The HTTP request **MAY** contain any valid HTTP header specified in [\[RFC7230\]](#).

Request Line URI /well-known/<service> (unless overridden using a TXT path attribute)

Request Line Method POST

Host: Header <domain>

Content-Encoding As specified in section yy below.

Content-Type As specified in section zz below.

Content-Length or Transfer-Encoding As specified in [\[RFC7230\]](#).

Payload The content payload as specified in section XX below.

[No dump of the binding yet]

~~~~

### 5.2.4. Response

The response **MAY** contain any HTTP response header but since JWB services do not make use of HTTP caching and messages are not intended to be modified by HTTP intermediaries, only a limited number of headers have significance:

**Response Code** The HTTP response code. This is processed as described in section zz below.

**Content-Type** As specified in section zz below.

**Content-Length or Transfer-Encoding** As specified in [\[RFC7230\]](#).

**Cache-Control** Since the only valid HTTP method for a JWB request is POST, JWB responses are not cacheable. The use of the cache-control header is therefore unnecessary. However, experience suggests that reviewers find it easier to understand protocol

specifications if they are reminded of the fact that caching is neither supported nor desired.

[No dump of the binding yet]

~~~~~

5.3. DARE Message Encapsulation

The payload of the HTTP requests and responses is a DARE Message whose payload contains the Mesh Service request or response.

The DARE Message encapsulation is used to authenticate the request or response data. The form of the authentication depending on the credentials available to the sender at the time the request is made.

Mesh Service **MUST** support the use of Mutually Authenticated Key Exchange [[draft-hallambaker-mesh-security](#)] to establish the Master Key used for authentication of requests and responses.

Requests and Responses **MUST** be authenticated. Requests and Responses **MUST** be encrypted if the transport is not encrypted and **MAY** be encrypted otherwise.

5.3.1. Null Authentication

Null Authentication **MAY** be used to make a Hello Request.

The Null Authentication mechanism **MUST NOT** be used for any Mesh Service request or response other than a Hello request.

Since the Mutually Authenticated key exchange requires both parties to know the public key of the other, it is not possible for a client to authenticate itself to the service until it has obtained the service public key. One means by which the client **MAY** obtain the service public key is by requesting the service return the credential in a Hello transaction.

5.3.2. Device Authentication

Device Authentication is used in two circumstances

- *When requesting creation of an account

- *When a device is requesting connection to a profile.

5.3.3. Profile Authentication

Profile Authentication has the same form as Device Authentication except that the client provides its Device Connection Assertion as part of the request:

5.3.4. Ticket Authentication

Ticket Authentication is used after a device has obtained an authentication ticket from a service. The ticket is returned in the response to a previous Profile Authentication exchange.

5.4. Payload Encoding

The Dare Message payload of a Hello request **MUST** be encoded in JSON encoding. The payload of all other requests **MUST** be in either JSON encoding or one of the encodings advertised as being accepted in a Hello response from the Service. Services **MUST** accept JSON encoding and **MAY** support the JSON-B or JSON-C encodings as specified in this document. Services **MUST** generate a response that is compatible with the DARE Message Content-Type specified in the request.

JSON was originally developed to provide a serialization format for the JavaScript programming language [[ECMA-262](#)]. While this approach is generally applicable to the type systems of scripting programming languages, it is less well matched to the richer type systems of modern object oriented programming languages such as Java and C#.

Working within a subset of the capabilities of JSON allows a Web Service protocol to be accessed with equal ease from either platform type. The following capabilities of JSON are avoided:

The ability to use arbitrary strings as field names.

The use of JSON objects to define maps directly

The following data field types are used:

Integer Integer values are encoded as JSON number values.

String Text strings are encoded as JSON text strings.

Boolean Boolean values are encoded as JSON 'false', 'true' or 'null' tokens according to value.

Sequence Sequences of data items that are encoded as JSON arrays

Object of known type Objects whose type is known to the receiver are encoded as JSON objects

Object of variable type

Objects whose type is not known to the receiver are encoded as JSON objects containing a single field whose name describes the type of the object value and whose value contains the value.

Binary Data Byte sequences are converted to BASE64-url encoding [[RFC4648](#)] and encoded as JSON string values.

Date Time Date Time values are converted to Internet time format as described in [[RFC3339](#)] and encoded as JSON string values.

5.5. Error handling and response codes

It is possible for an error to occur at any of the three layers in the Web Service binding:

Service Layer

HTTP Layer

Transport Layer

Services **SHOULD** always attempt to return error codes at the highest level possible. However, it is clearly impossible for a connection that is refused at the Transport layer to return an error code at the HTTP layer. It is however possible for a HTTP layer error response to contain a content body.

In the case that a response contains both a HTTP response code and a well-formed payload containing a response, the payload response **SHALL** have precedence.

6. Mesh Service Transactions

6.1. Service Description

The Hello transaction is used to determine the features supported by the service and obtain the service credentials

The request payload:

```
{  
  "HelloRequest":{}}
```

The response payload:

```
{
  "MeshHelloResponse": {
    "Status": 201,
    "Version": {
      "Major": 3,
      "Minor": 0,
      "Encodings": [ {
        "ID": ["application/json"]
      } ],
    },
    "EnvelopedProfileService": [ {
      "EnvelopeId": "MA36-TUJL-QRZJ-3M3L-SRBQ-BRYQ-W2YM",
      "dig": "S512",
      "ContentMetaData": "ewogICJVbm1xdWVJZCI6ICJNQTM2LV RVSkwtUV
JaSi0zTTNMLVNSQLtEtQlJZUS1XMllNIiwKICAiT Wzc2FnZVR5cGU iOiAiUHJvZm1
sZVNlc nZpY2UiLAogICJjdHkiOiAiYXBwbGljYXRpb24vbW1 tLT29iamVjdCIsCiAg
IkNyZW F0ZWQiOiAiMjAyMS0wMS0xM1QxNjo zODoxN1oifQ"} ,
      "ewogICJQcm9maWxlU2Vydm l j ZSI6IHsKICAgICJQcm9maWxlU2lnbmF0dX
JlIjog ewogICAgIC AiVWRmIjogIk1BMzYtVFVKTC1RUlpKL TNNM0wtU1JCUS1CUll
RLVcyWU0i LAogICAgIC AiUHVibGljUGFyYW1ldGVycy I6IHsKICAgICAgIC AiUHV i
bGlj s2V5RUNESCI6IHsKICAgICAgIC AgIC Jjc nYiOiAiRWQ0NDgi LAogICAgIC Ag I
CAg IlB1YmxpYyI6ICJHwnNNaWl0c l NMc3Zpbj U5SUZvTW hqcHJJYTRxNUFTTVZZuk
FNVGdlUUZT M05lUWxDU1hkCiAgZmZzbzJNaHM5RD c2ZDJFX1lSWWFtZl l BIn19fSw
KICAg ICJTZXJ2aWNlRW5jcnlwdGlvbi I6IHsKICAgICAg IlVk ZiI6ICJNRE5PLVFF
VzMtTz VHTy1JNUL aLVNXTEstUENaMy1WQ1QzIiw KICAgICAg IlB1YmxpY1 BhcmFtZ
XRlcnMi OiB7CiAgICAgIC Ag IlB1YmxpY0tleUVDREgi OiB7CiAgICAgIC Ag ICAi Y3
J2Ijog Ilgl g0NDgi LAogICAgIC Ag IC Ag IlB1YmxpYy I6IClwY2szb0lGWjZ0eEUz ajJ
HRU xXZzha Q1BXaFRHaWtRUWZqeFUxUHhGbVZBcz RmLV d6MW05CiAgMkg2alFxwmJq
ZmVKLUV ickpetzJ0TVFBIn19fX19",
      {
        "signatures": [ {
          "alg": "S512",
          "kid": "MA36-TUJL-QRZJ-3M3L-SRBQ-BRYQ-W2YM",
          "signature": "MeeZPEkxMp nwnFicq eau_cw3m82x8n-ft5Gbp0ka
U53PTSHilXrnbJozvpM9kIFK5dr_xFVAcQsAoH0KS9NciI2CHvMYuZgmOX7Yj8Ru
bBFpWQAblssgVp5WDTJB BKHNhy9ahysun5far-CSdh6MBEA"} ] ,
        "PayloadDigest": "7rouYS Gk4aJyF1lFml3TtsjI e_4VLmbAzgm1STim
5-8z0gBV-R1n8-qXNUolXqQWo05cc8hBXuZdfz6KfZdmwA"} ] ,
      "EnvelopedProfileHost": [ {
        "EnvelopeId": "MAFN-RXI0-YV7S-ZIU2-WXL0-S3ZO-DUU6",
        "dig": "S512",
        "ContentMetaData": "ewogICJVbm1xdWVJZCI6ICJNQZ0LVJYSU8twV
Y3Uy1aSVUyLVdYTE8t UzNaTy1EVVU2IiwKICA iTWzc2FnZVR5cGU iOiAiUHJvZm1
sZUhvc 3Qi LAogICJjdHkiOiAiYXBwbGljYXRpb24vbW1 tLT29iamVjdCIsCiAgIkNy
ZW F0ZW QiOiAiMjAyMS0wMS0xM1QxNjo zODoxN1oifQ"} ,
        "ewogICJQcm9maWxlSG9zdCI6IHsKICAgICJQcm9maWxlU2lnbmF0dXJlIj
ogewog ICAqIC AiVWRmIjogIk1BRk4tUlhlJTY1ZVjdTLVpJVTItv1hMTy1TM1pPLUR
```

```
VVTYiLAogICAgICAIUHVibGljUGFyYW1ldGVycyI6IHsKICAgICAgICAIUHVibGlj
S2V5RUNESCi6IHsKICAgICAgICAgICJjcnYiOiAiRWQ0NDgiLAogICAgICAgICAgI
lB1YmXpYyI6ICJjRWwhGR2FLd3B1Y1F1N0lGemhlWE1TOUVrT3p2dTY5SWNxeWNjcm
Q3RkRmDG9tdkU2S044CiAgVXJhUDcZblJ4TDhMOGEwdjdF0FlBQTRBin19fSwKICA
gICJLZXlBdXR0ZW50aWNhdGlvbI6IHsKICAgICAgICAgICAgICAgICAgICAgICAgICAg
NjVYNY1MN1k3LVMzREUtN09MNy1PRk5JIiwKICAgICAgICAgICAgICAgICAgICAgICAg
nMiOiB7CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIC
ogIlg0NDgiLAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIC
JNnNx0FQyOVRuTWctcV84a0NibGFMeVc3bGNFdEJ6NWFTCiAgZ2p2RDFNVm04MElz
V0JmRk5MTER2NFFBin19fX19",
{
  "signatures": [{
    "alg": "S512",
    "kid": "MAFN-RXIO-YV7S-ZIU2-WXLO-S3ZO-DUU6",
    "signature": "BuUF07hpykuHiwuM0i1P4aCMIHBHmubtKdskHfBR
NrHGV0Va3Lcljyp5XaXp1KDhrygv8l9XcJiAL2Vy2xTHghzfu29XLC4p59RxhlxP6
nlWP0Lw0pUER9v9v9BEss1ULDJYw7yc-mEf0h5sygAntx8A"}
  ],
  "PayloadDigest": "b0kDrRdW4SGwdJ0_YaJoeS1JhczC7o-ftjVwr3sC
yLo7-dHhyB-cKe3Fksrg31v_p2R4q6IhUwY0wFrgFEF1nA"}
}]}
```

6.2. Account Creation

6.2.1. Bind User Account

A User Account is bound to a Mesh Service by completing a BindAccount transaction with the service.

The BindRequest message specifies the account address and ProfileUser of the account to be serviced.

The BindAccount transaction is unique in that it can fail to complete for reasons that are outside the scope of the Mesh specifications. Creation of an account might require payment to be made or authentication of the user's credentials. It is thus quite normal for the result of a CreateRequest to be the account being created in an 'on hold' state which can only be changed out of band.

If the request is at least partially successful, a BindResponse message is returned. In the case of partial success, a description of the request status and link to a Web page providing further details **MAY** be returned.

The request payload:

```
{
  "HelloRequest": {}}
```

The response payload:

```
{
  "MeshHelloResponse": {
    "Status": 201,
    "Version": {
      "Major": 3,
      "Minor": 0,
      "Encodings": [ {
        "ID": ["application/json"]
      } ],
    },
    "EnvelopedProfileService": [ {
      "EnvelopeId": "MA36-TUJL-QRZJ-3M3L-SRBQ-BRYQ-W2YM",
      "dig": "S512",
      "ContentMetaData": "ewogICJVbm1xdWVJZCI6ICJNQTM2LV RVSkwtUV
JaSi0zTTNMLVNSQLtEtQlJZUS1XMllNIiwKICAiT Wzc2FnZVR5cGU iOiAiUHJvZm1
sZVNlc nZpY2UiLAogICJjdHkiOiAiYXBwbGljYXRpb24vbW1 tLT29iamVjdCIsCiAg
IkNyZW F0ZWQiOiAiMjAyMS0wMS0xM1QxNjo zODoxN1oifQ"}],
      "ewogICJQcm9maWxlU2Vydm l j ZSI6IHsKICAgICJQcm9maWxlU2lnbmF0dX
JlIjogewogICAgICAI VWRmIjogIk1BMzYtVFVKTC1RUlpKL TNNM0wtU1JCUS1CUll
RLVcyWU0i LAogICAgICAI UHVibGljUGFyYW1ldGVycy I6IHsKICAgICAgICAI UHV i
bGlj s2V5RUNESCI6IHsKICAgICAgICAgIC Jjc nYiOiAiRWQ0NDgi LAogICAgICAgI
CAgIlB1YmxpYyI6ICJHwnNNaWlOc l NMc3ZpbjU5SUZvTW hqcHJJYTRxNUFTTVZZuk
FNVGdlUUZT M05lUWxDU1hkCiAgZmZzbzJNaHM5RDc2ZD JFX1lSWWFtZl l BIn19fSw
KICAgICJTZXJ2aWNlRW5jcnlwdGlvbiI6IHsKICAgICAgIlVk ZiI6ICJNRE5PLVFF
VzMtTzVHTy1JNULaLVNXTEstUENaMy1WQ1QzIiwKICAgICAgIlB1YmxpY1 BhcmFtZ
XRlcnMiOiB7CiAgICAgICAgIlB1YmxpY0tleUVDREgiOiB7CiAgICAgICAgICAI y3
J2IjogIl g0NDgi LAogICAgICAgICAg IlB1YmxpYyI6IClwY2szb0lGWjZ0eEUzajJ
HRU x XZzhaQ1BXaFRHaWtRUWZqeFUxUHhGbVZBczRmLV d6MW05CiAgMkg2alFxwmJq
ZmVKLUVickpETzJ0TVFBIn19fX19",
      {
        "signatures": [ {
          "alg": "S512",
          "kid": "MA36-TUJL-QRZJ-3M3L-SRBQ-BRYQ-W2YM",
          "signature": "MeeZPEkxMp nwnFicq eau_cw3m82x8n-ft5Gbp0ka
U53PTSHilXrnbJozvpM9kIFK5dr_xFVA cQsAoH0KS9NciI2CHvMYuZgmOX7Yj8Ru
bBFpwQABlssgVp5WDTJB BKNHny9ahysun5far-CSdh6MBEA"} ] ,
        "PayloadDigest": "7rouYS Gk4aJyF1lFml3TtsjI e_4VLmbAzgm1STim
5-8z0gBV-R1n8-qXNUolXqQWo05cc8hBX uZdfz6KfZdmwA"} ] ,
      "EnvelopedProfileHost": [ {
        "EnvelopeId": "MAFN-RXI0-YV7S-ZIU2-WXL0-S3ZO-DUU6",
        "dig": "S512",
        "ContentMetaData": "ewogICJVbm1xdWVJZCI6ICJNQZ0LVJYSU8twV
Y3Uy1aSVUyLVdYTE8tUzNaTy1EVVU2IiwKICA iTWzc2FnZVR5cGU iOiAiUHJvZm1
sZUhvc3 QiLAogICJjdHkiOiAiYXBwbGljYXRpb24vbW1 tLT29iamVjdCIsCiAgIkNy
ZW F0ZWQiOiAiMjAyMS0wMS0xM1QxNjo zODoxN1oifQ"}],
        "ewogICJQcm9maWxlSG9zdCI6IHsKICAgICJQcm9maWxlU2lnbmF0dXJlIj
ogewogICAgICAI VWRmIjogIk1BRk4tUlhlJTY1ZVjdTLVpJVTItv1hMTy1TM1pPLUR
```

```
VVTYiLAogICAgICAIUHVibGljUGFyYW1ldGVycyI6IHsKICAgICAgICAIUHVibGlj
S2V5RUNESCI6IHsKICAgICAgICAgICJjcnYiOiAiRWQ0NDgiLAogICAgICAgICAgI
lB1YmxpYyI6ICJjRWwhGR2FLd3B1Y1FIN0lGemhlWE1TOUVrT3p2dTY5SWNxeWNjcm
Q3RkRmDG9tdkU2S044CiAgVXJhUDcZblJ4TDhMOGEwdjdF0FlBQTRBin19fSwKICA
gICJLZXlBdXR0ZW50aWNhdGlvbI6IHsKICAgICAgICAgICAgICAgICAgICAgICAgIC
NjVYNY1MN1k3LVMzREUtN09MNY1PRk5JIiwKICAgICAgICAgICAgICAgICAgICAgIC
nMiOiB7CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
ogIlG0NDgiLAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
JNnNx0FQyOVRuTWctcV84a0NibGFMeVc3bGNFdEJ6NWFTCiAgZ2p2RDFNVm04MElz
V0JmRk5MTER2NFFBin19fX19",
{
  "signatures":[{
    "alg":"S512",
    "kid":"MAFN-RXIO-YV7S-ZIU2-WXLO-S3ZO-DUU6",
    "signature":"BuUF07hpykuHiwuM0i1P4aCMIHBHmubtKdskHfBR
NrHGV0Va3Lcljyp5XaXp1KDhrygv8l9XcJiAL2Vy2xTHghzfu29XLC4p59RxhlxP6
nlWP0Lw0pUER9v9v9BEss1ULDJYw7yc-mEf0h5sygAnTx8A"}
  ],
  "PayloadDigest":"b0kDrRdW4SGwdJ0_YaJoeS1JhczC7o-ftjVWr3sC
yLo7-dHhyB-cKe3Fksrg31v_p2R4q6IhUwY0wFrgFEF1nA"}
]}}
```

[Future: Consider converting this to a Messaging flow.]

6.2.2. Bind Group Account

Mesh Group Accounts are created in the same manner as user accounts except that a ProfileGroup is specified.

The request payload:

6.2.3. Unbind Account

An account registration is deleted using the `UnbindAccount` transaction.

The request payload:

The response payload:

6.3. Persistence Store Management

All the state associated with a Mesh profile is stored as a sequence of DARE Messages in a Dare Container. The Mesh Service holding the master copy of the persistence stores and the devices connected to the profile containing complete copies (replicas) or partial copies (redactions).

Thus, the only primitive needed to achieve synchronization of the profile state are those required for synchronization of a DARE Container. These steps are:

- *Obtain the status of the catalogs and spools associated with the account.
- *Download catalog and spool updates
- *Upload catalog updates.

To ensure a satisfactory user experience, Mesh Messages are intentionally limited in size to 64 KB or less, thus ensuring that an application can retrieve the most recent 100 messages almost instantaneously on a high bandwidth connection and without undue delay on a slower one.

6.3.1. Status

The status transaction returns the status of the containers the device is authorized to access for the specified account together with the updated Device Connection Entry if this has been modified since the entry presented to authenticate the request was issued.

The request payload:

```
{  
  "StatusRequest":{}}
```

The response payload:

[illegible]

```

        "PayloadDigest": "fb_iksIe0dM4IWIWZjmKlYQSF-XttjIAg8Bww4tJ
jp0E0P9bxX42pkNorLfHQ8XyD8x9IHT-FKh-_1hLJNAUzA"}
    ],
    "ContainerStatus": [{
        "Container": "MMM_Inbound",
        "Index": 3},
    {
        "Container": "MMM_Outbound",
        "Index": 1},
    {
        "Container": "MMM_Local",
        "Index": 2},
    {
        "Container": "MMM_Access",
        "Index": 1},
    {
        "Container": "MMM_Credential",
        "Index": 3},
    {
        "Container": "MMM_Device",
        "Index": 3},
    {
        "Container": "MMM_Contact",
        "Index": 2},
    {
        "Container": "MMM_Application",
        "Index": 1},
    {
        "Container": "MMM_Bookmark",
        "Index": 2},
    {
        "Container": "MMM_Task",
        "Index": 2}
    ]}]

```

6.3.2. Download

The download transaction returns a collection of entries from one or more containers associated with the profile.

Optional filtering criteria **MAY** be specified to only return objects matching specific criteria and/or only return certain parts of the selected messages.

The service **MAY** limit the number of entries returned in an individual response for performance reasons.

Obsolete example 1

6.3.3. Conflict Detection

Clients **SHOULD** check to determine if updates to a container conflict with pending updates on the device waiting to be uploaded. For example, if a contact that the user modified on the device attempting to synchronize was subsequently deleted.

The means of resolving such conflicts is not in the scope of this specification.

6.3.4. Filtering

Clients may request container updates be filtered to redact catalog entries that have been updated or deleted or spool entries that have been read, deleted or were received before a certain date.

6.3.5. Transact

The transact transaction appends envelopes to one or more stores. The operation is atomic, that is either all the changes specified will be made to the stores or none will. This ensures that simultaneous attempts to update a store do not result in race conditions.

Each update to a catalog or container specifies the expected container index and apex digest. This provides a strong guarantee of consistency. The service **MUST** verify each update to check that the Merkle Tree values specified are consistent with the store entries and that the signature on the apex value (if specified) is valid and correct.

Services **MAY** impose limits on the size and number of additions performed in response to a TransactRequest message to ensure that processing time does not degrade performance for other users.

Obsolete example 2

6.4. Messaging

Mesh Messaging is an asynchronous messaging service that allows exchange of information between devices connected to a Mesh account and between Mesh users.

To enable effective abuse mitigation, Mesh Messaging enforces a four corner communication model in which all outbound and inbound messages pass through a Mesh Service which accredits and authorizes the messages on the user's behalf.

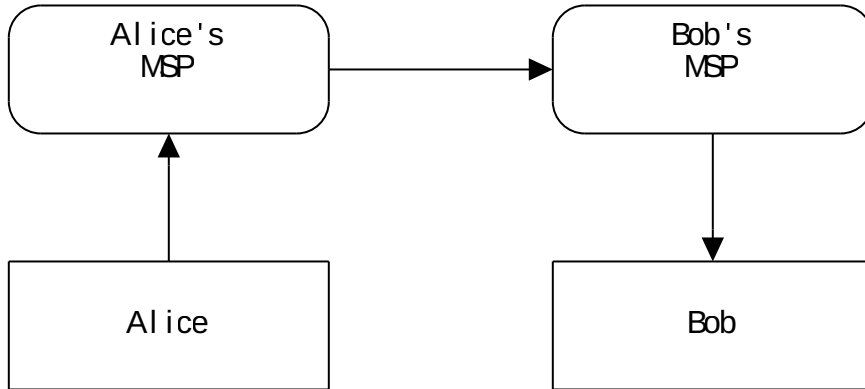


Figure 2:

The Post transaction is used for client-service and service-service messaging transactions.

Client-Service (Post Transaction)

To send a message, the client creates the Mesh Message structure, encapsulates it in a DARE Message and forwards this to its service using a Post transaction.

The Post transaction is authenticated to the service by device using the usual means of profile or ticket authentication.

The DARE Message **MUST** be signed under a device signature key accredited by a Device Connection Assertion provided in the message signature block.

The request payload:

```
{
  "ConnectRequest": {
    "EnvelopedRequestConnection": [ {
      "EnvelopeId": "MD3N-INKQ-MLJM-WYE2-642P-2VM4-V4TL",
      "ContentMetaData": "ewogICJYbmIxdWVJZCI6ICJORDJELTJCmkyTmkVDUy1NWDRKLUJMNkstWURWUy03T0JBIIiwKICAiTWVzc2FnZVR5cGUiOiAiUmVxdWVzdENvbml5Y3Rpb24iLAogICJjdHkiOiAiYXBwbGljYXRpb24vbW1tLT29iamVjdCIsCiAgIkNyZWFOZWQiOiAiMjAyMS0wMS0xM1QXNj0zODoyMFoifQ"}],
      "ewogICJSZXF1ZXNQ29ubmVjdGlvbiI6IHsKICAgICJNZXNzYWdlSWQiOiAiTkQyRC0yQjJGLTJFQ1MtTVg0Si1CTDZLLVlEVlMtN09CQSIsCiAgICAIqXV0aGVudGllYXRlZERhdGEiOiBbewogICAgICAgICJFbnZlbG9wZULkIjogIk1DRFotSVpGWS1QV1QzLTM2VVAtNEw2US01UU1KLVLXV1kiLAogICAgICAgICJkaWciOiAiAiUzUXMiIsCiAgICAgICAgIkNvbnRlbnRNZXRhRGF0YSI6ICJld29nSUNKVmJtbHhkV1ZKWkNJNklDSk5RMfJhTFVsYVJsa3RVRRmRVTXkwCiAgek5sVlFMVFJNTmxFdE5WRk5TaFAvJFkwkpdpd0tJQ0FPvFdwemMyRm5aVlI1Y0dVaU9pQWlVSEp2Wm1sc1oKICBVUmxbWxqWlNjc0NpQwdJbU4wZVNJNklDSmhjSEJzYVdOAGRhbHZiaTl0YlclcwdmIySnFaV04wSwl3S0lDQogIGlRM0pswVhSbFpDSTZJQ0l5TURJeExUQXhmVEV6VkRFMk9qTTRPakl3V2lKOSJ9LAogICAgICAIzXdvdZ0lDSlFjbTltYVd4bFJHVjJhV05sSWpwvZ2V3b2dJQ0FnSWxCeWIyWgogIHBIr1ZUYVdkdVlYUjFjbVVPt21CN0NpQwdJQ0FnSUNKVlpHWWlPaUFpVFVORVdpMUpxA1paTFZCWFZETXRNCiAgelpwVUMwMFRFWlJMVfZSVFVvdFdWZFhXU0lzQ2lBZ0lDQwdJQ0pRZFdkc2FXTlFZWEPoYldWMFPySnPJam8KICBnZXdvZ0lDQwdJQ0FnSUNKUWRXSnnNhV05MWlhSRlEWUklJam9nZXdvZ0lDQwdJQ0FnSUNBZ0ltTnlkaUk2SQogIENKRlpEUTBPQ0lzQ2lBZ0lDQwdJQ0FnSUNBaVVIVmliR2xqSWpvZ0lqZFBZMnRyUwtSMVVYWXRVUkZVkvVrCiAGMWRXVXRWM0YUwmxSMFJTHRHR3h6VKdowMrZDJMvmxvt1MxNU1wwLBVBd14VvdZS0lDQnlSa3A1TVhwdlEKICYUktUVkpVv1hsTlQwdEXUVGXpVlVFawZYMTlMQW9nSUNBZ0lrSmhjMlZGYm10eWVYQjBhvzl1SWpvZ2V3bwogIGdJQ0FnSUNBaVZXUm1Jam9nSwsxRVQxTxRXVU5IVHkxSVRqVlpMVXRZVEZvdE5rWldUuzFKU1RKRExVazBSCiAga1lpTEFvZ0lDQwdJQ0FPvUhWaWJHbGpVR0Z5WvcxbGRHVnljeUk2SUhzS0lDQwdJQ0FnSUNBaVVIVmliR2wkICBquzJWNVJVTKVTQ0k2SUhzS0lDQwdJQ0FnSUNBZ0lDSmpjb1lpT2lBaVdEUTBPQ0lzQ2lBZ0lDQwdJQ0FnSQogIENBaVVIVmliR2xqSWpvZ0lscFZVM3Bmtlc5c2FuwnJJamhzVlZsd2FwUnNkbkgGUnpJNFFuQm9TRVU1Y0U5CiAgT2FqSXdpWG96U0hoRWVRNBXBUamsWU25NS0lDQjJMVWhTVGt3MFFFXbDFWRXRVWw1abFJ6TkZTWEOwVVFVfaWyKICYMTlMQW9nSUNBZ0lrSmhjMlZCFhShb1pXNTBhV05oZEsdsmJpSTZJSHNLSUNBZ0lDQwdJbFZRwm1JNklDSgogIE5RMDVSTFRkRk1rOHRWVWhWUKMxS1JWRTNMVE5VUTBzdFIwRlZSeTAyVEZks0lpd0tJQ0FnSUNBZ0lsQjFZCiAgbXhwWTFCaGNtRnRaWFJsY25NaU9pQjdDaUFnSUNBZ0lDQwdJbEIXww14cFkdGxlVVZEukVnaU9pQjdDaUEKICBnSUNBZ0lDQwdJQ0FPWTNKMKlqb2dJbGcwTkrNaUxBb2dJQ0FnSUNBZ0lDQwdJbEIXww14cFl5STZJQ0puUwogIDB0QlJtOW10Mkp5TnpSBwFHTnRiWGRmZUVZeGVVZEZWV3B0UkVKTVJXDHbkWFJGwjI5WGNrcENjamRmZwt4CiAgQ1MyMDJDauFNuLZKWVJtwmliR1JHU1RWullqRmhSMVZUVGxwd2NIRkJJbjE5ZlN3S0lDQwdJQ0pDWvhObFUKICYayBg5ibUYwZFhkbElqb2dld29nSUNBZ0lDQWlwV1JtSWpvZ0lrMUVRMDh0VGtmZMFPmu1Va3BhTFVKQldVdwogIHSa1pDU2kxSFZFMvpMVE5TVERRaUxBb2dJQ0FnSUNBaVVIVmliR2xqVUDGeVlXMwxkR1Z5Y3lJNklIc0tJCiAgQ0FnSUNBZ0lDQWlVSFZpykdsalMyvjVSU5FU0NJNklIc0tJQ0FnSUNBZ0lDQwdJQ0ppqY25ZaU9pQWlSV1EKICAwTkRnaUxBb2dJQ0FnSUNBZ0lDQwdJbEIXww14cFl5STZJQ0p4T0dkwmJFUTFibMM0UTFoMmFsaGtNR2gxWQogIDFWMLnsRTBwbEZWTOMxNLIwZE9iRTVSVEVklYymUNWMPvFZFROa1MxbDBDAUFnyjBwTLRYZEhTMnh5YVdVCiAgNE1YaFdZakY0U1VadFowZEJJBbjE5Zlqx0SIsCiAgICAgIHsKICAgICAgICAIa2lnbmF0dXJlcyl6IFt7CiAgIC
```

The response payload:

The response payload:

```
{
  "ConnectResponse":{
    "Status":201,
    "StatusDescription":"Operation completed successfully",
    "EnvelopedAcknowledgeConnection":[{
      "EnvelopeId":"MATU-PAPK-JHQ5-BBES-PFX7-UCRN-OLWQ",
      "ContentMetaData":"ewogICJYbmlxdWVJZCI6ICJXR0FaLUJYMcEtN0
c3Mi1KNVUzLVNHR1QtVUizVy1UVjRMIiwKICAiTWVzc2FnZVR5cGUiOiAiQWNrbm9
3bGVkZ2VDb25uZWNOaw9uIiwKICAiY3R5IjogImFwcGxpY2F0aw9uL21tbS9vYmpl
Y3QiLAogICJDCmVhdGVkIjogIjIwMjEtMDEtMTNUMTY6Mzg6MjBaIn0",
      "SequenceInfo":{
        "Index":1,
        "TreePosition":0},
      "Received":"2021-01-13T16:38:20Z"},
    "ewogICJBY2tub3dsZWRnZUNvbm5lY3Rpb24iOiB7CiAgICAiTWVzc2FnZU
lkIjogIldHQVotQlgyQS03RzcyLUo1VTMtU0dHVC1VQjNXLVRWNEwiLAogICAgIkv
udmVsb3BlZFJlcXVlc3RDb25uZWNOaw9uIjogW3sKICAgICAgICAiRW52ZWxvcGVJ
ZCI6ICJNRDNOLU1OS1EtTUXKTS1XWUUYLTy0MlAtMlZNNC1WNFRMIiwKICAgICAgI
CAiQ29udGVudE1ldGFZEYXRhIjogImV3b2dJQ0pWYm1seGRXVkpaQ0k2SUNKT1JESk
VMVEpDTWtZdE1rVkrVeTEKICBOV0RSS0xVSk10a3N0V1VSV1V5MDNUMEpCSWl3S0l
DQWlUV1Z6YzJGblpWUjVjR1VpT2lBaVvtVnhkV1Z6ZAogIEV0dmJtNwZM1JwYjI0
aUxBb2dJQ0pqZEhraU9pQWlZWEl3YkdsallYUnBiMjR2YlxdEwyOWlhbVZqZENJC
iAgc0NpQWdJa055WldGMFpXUWlPaUFpTWpBeU1TMhdNUzB4TTFReE5qb3pPRG95TU
ZvawZRIIn0sCiAgICAgICJld29nSUNKU1pYRjFawe4wUTI5dWJtVmpkR2x2Ym1JNk1
Ic0tJQ0FnSUNKCiAgTlpYtnpZV2RsU1dRaU9pQWlUa1F5UkMweVFqSkdMVEpGUTFN
dFRWZzBTaTFDVERaTEwBwE10TjA5Q1EKICBTSXNDaUFnSUNBaVFYVjBhR1Z1Z
EdsallYUmxarVJoZEdFaU9pQmJld29nSUNBZ0lDQWdJQ0pGYm5abGJHOqogIHdaVW
xrSwpvZ0lrMURSRm90U1ZWR1dTVMVFWMVF6TFRNMLZWQXRORXcyVVMwMVMVMUTMVMx
YVjFraUxBb2dJCiAgQ0FnSUNBZ0lDSmthV2NpT2lBaVV6VXhNaUlzQ2lBZ0lDQWdJ
Q0FnSwT0dmJuUmxiBlJOWlhSaFJHRjBZU0kKICA2SUNKbGQyOW5TVU5LVm1KdGJlA
GtWMVpLV2t0Sk5rbERTazVSTUZKaFRGVnNZVkpzYTNSVlJtUlZUWgt3QwogIGlBZ2
VrNXNWbEZNVkZKTlRteEZkRTRVXUms1VGFURmFWakZrV2tscGQWdEpRMEZwVkJkV2V
tTXlSbTVhVmxiJCiAgMVkwZGFhVUdsVlNfcDJXbTFzYzFvS0lDQlZVbXhrYld4
cVdsTkpjME5wUVdkSmJVNHdaVkJkTmtSRFMKICBtaGpTRUp6WVZkT2FHUkhiSFppY
VRsMFlsY3dkbU1U25GYVYwNHdTV2wzUzBsRFFRb2dJR2xSTTBwc1dWaogIFNiRn
BEU1RaSlEwbDVUVVJKZUV4VVFYaE1WRVY2VmtSRk1rOXFUVFJQYWtsM1YybEtPU0o
5TEFvZ0lDQWdJCiAgQ0FpwlhkdlowbERTbEZqYlRsdFlwZDRiRkpIVmpKaFYwNXNT
V3B2WjJWM2IyZEpRMEZuU1d4Q2VXSXlXZ28KICBnSuhCaVixWlVZVmRrZFZswVvqR
mpiVlZwVDJsQ04wTnBRV2RKUTBGblNVTKtWbHBIV1dsUGFVRnBWRlZPUgogIFZkcE
1VcFhhMXBhVEZaQ1dGwKVUWFJQ02lBZ2VscFdwVU13TUZSRVdsSk1WRlpTVkZWdmR
GZFdaRmhYVTBsCiAge1EyBEJaMGxEUVdkSlEwcFJaRmRLYzJGWFRsRlpxRXBvWwXk
V01GcFlTbnBKYW04S0lDQm5aWGR2WjBsRFEKICBZEprMEZuU1VOS1VXU1htbk5oV
jA1TVdsahNSbEV3VWtsSmFtOW5aWGR2WjBsRFFXZEprMEZuU1V0QlowbAogIHRUbm
xrYVVRm1NRb2dJRu5LUmxwRVVUQlBRMGx6UTJsQlowbERRV2RKUTBGblNVTKJhVlZ
JVm1saVIyeHFTCiAgV3B2WjBscVpGQlpNb1J5VVd0U01WVl1XWFJaVlVrelZrVnJD
aUFnTVdsWfZYUldNMFl3VW14U01GS1RNSFIKICBoUjNoNlZrZG9xbVJyWkRKTvZte
HZUMU14TlUxv1dsQlZiRGw0VlZkwlMwbERRbmXTYTNBMVRwaHdkbEVLSQogIENBeV
VrdFVwa3BWVjFoc1RsUXdkRXhVVKd4UFZsVkJhV1pZTVRstVFXOW5TVU5CWjBsc1N
taGpNbFpHwW0xCiAgT2VXVllRakJoVnpsMVNXcHZaMlYzYndvZ0lHZEprMEZuU1V0
```

QmFwWlhVbTFKYW05b1NXc3hSVlF4VFhSWFYKICBVNUlWSGt4U1ZScVZscE1WWFJaV
kVadmRFNXJXbGRVXpGS1UxUktSRXhWYXpCU0NpQWdhMwxwVEVGdlowbAogIERRV2
RKUTBGcFZVaFdhV0pIYkdWVlIwWjVXVMN4YkdSSFZubGplVWsyU1Voe1MwbERRV2R
KUTBGblNVTKJhCiAgVlZJVm1saVIyd0tJQ0JxVXpKV05WSlZUa1ZUUTBrM1NVaHpT
MGxEUVdkSlEwRm5TVU5CWjBsRFNtcGpibGwKICBwVDJsQmFWZEvvVEJQUTBse1EyB
EJaMGxEUVdkSlEwRm5TUW9nSUV0QmFWVklWbWxpUjJ4cVNXCZaMGxzYwogIEZaV
k0zQm1UbGM1YzJGdVduSmphbWh6Vmxac2QyRldVbk5rYmtwR1VucEpORkZ1UW05VFJ
WVTFZMFU1Q2lBCiAgZ1QyRnFTWGRQV0c5NlUwaG9SV1JWt1hCVWFtc3dVMjVuUzBs
RFFqSk1WV2hUVkd0M01GRlhiREZXUlhSVlclKICBXMWFiRko2VGtaVfDFMhdWRlZGY
VdZS0lDQl1NVGxNUVc5b1NVTKJaMGxyU21oak1sWkNaRmhTYjFwWE5UQgogIGhWMD
VvWkVkc2RtSnBTVPfKPU0h0TFNVTKJaMGxEUVdkSmJGwnJXbWxKTmtsRFNnb2dJRTV
STURWU1RGUmtSCiAgazFyT0hSV1ZXaFdvA014UzFKV1JUTk1WRTVWVRCemRGSXdS
bFpTZVRBeVZFwmtTMGxwZDB0SlEwRm5TVU4KICBCWjBsc1FqRlpDaUFnYlhod1dUR
kNhR050Um5SYvdGSnNZMjV0YVU5cFFqZERhVUZuU1V0QlowbERRV2RKYgogIEVJeF
dXMTRjRmt3ZEd4bFZWwKvVa1ZuYVU5cFFqZERhVUvLSUNCblNVTKJaMGxEUVdkSlE
wRnBXVE5LTwtScIAgcWIyZEpIR2N3VGtSbmFVeEJiMmRKUTBGblNVTKJaMGxEUVdk
SmJFSXhXVzE0Y0ZsNVNUWkpRMHB1VXdVZ0kKICBEQk9RbEp0T1cxT01rcDVUbnBTY
ldGSFRuUmlXR1JtWlVWwVHVlZaRVpXVjNCMFVrVktUVkpYZEHca1dGSgogIEdXak
k1V0d0cmNFTmphbVJtWld0NENpQWdRMU15TURKRGFVRm5VbFpLV1ZKdFdtbG1SMUp
IVTFsv1VsbHFSCiAgbWhTTVZaVVZHeFdkMk5JUmtKSmJqRTVabE4zUzBsRFFXZEpR
MHBEV1ZoT2JGVUtJQ0F5Ykc1aWJVWXdarMgKICBLyKvscWIyZGxkMjluU1V0Qlowb
ERRV2xXVjFKdFNXCZaMGxyTVVWUk1EaDBWR3R6TUZWcE1VMVZhM0JoVaogIEZWS1
FsZFZkd29nSUhSU2ExcERVmmt4U0ZaRk1WcE1WRTVUVkVSUmFVeEJiMmRKUTBGblN
VTkjhVlZJVm1ScIAGAyiEhFWWRHZVZsWE1XEGtSMVo1WTNsSk5rbEljMHRKQ2lB
Z1EwRm5TVU5CWjBsRFFXbFZTRlpwWtKc2EKICBstXlWalZTVlU1RlUwTkP0a2xJY
zB0SlEwRm5TVU5CWjBsRFFXZEpRMHBxWTI1WmFVOXBRV2xTVjFFS0lDQogIHdUa1
JuYVv4QmIyZEpRMEZuU1V0QlowbERRV2RKYkVJeFdXMTRjRmw1U1RaSlEwcDRUMGR
rV21KRlVURmliCiAgBU0wVVRGb01tRnNhR3R0UjJneFdRb2dJREZXTWx0c1JUQldi
RVpXVDBNeE5sSXdaRTlpU1RWU1ZFVmtRbFkKICB5TVV0V01WcEzaRlJvYTFNeGJEQ
kRhVUZuWpCd1RsUllaRWhUTW5oNVlWZFZDaUFnTkUxWWFGZFpha1kwVQogIDFWYw
RGb3daRUpKYmpFNVpsZ3hPU0lzQ2lBZ0lDQWdJSHNLSUNBZ0lDQWdJQ0FpYzJsbmJ
tRjBkWepsY3lJCiAgNklGdDdDaUFnSUNBZ0lDQWdJQ0FnSUNKaGJHY2lPaUFpVXpV
eE1pSXNDaUFnSUNBZ0lDQWdJQ0FnSUNKcmEKICBXUWlPaUFpVFVORVdpMUpXa1paT
FZCWFZETXRNe1pwVUMwMFREWlJMVfZSVFVvdFdWZFHxU0lzQ2lBZ0lDQogIGdJQ0
FnSUNBZ0lDSnphv2R1WVhSMWntVWlPaUFpV1ZKb1dHdE1ZblYwU1ZGS1pqRlJWRWx
zTlY5aFEyMW1WcIAGMGhDVERksSFFtWldUR1J1TTJaU1oyNDRXb1k0VkhCV2JRb2dJ
R0Z1Y1d0R1VtSmhTWfptU1hkRE56Wmp1SHAKICA0VjJFemFrRnpRVnB0Um1waVNHU
npZbEp3wnpGw1NsWTRhR0ZrVlRGV09FZETMWHAWVUdoTGRtTjBDaUFnUwogIGx0bV
UzcEhSVWh5U1d0a1NFdG10bnB1YjI1TGIzUmZSWEJwUkRkaFRtZDBlVmgyUlZOM1R
VRWlMvJbZQ2lBCiAgZ0lDQWdJQ0FnSwxCaGVXehZZV1JFYVdkbGMzUWlPaUFpV1VF
d2VuaGFNRfo1V1RCWFMxYzRjRmhtWwxsUmQKICAwB3lSamQ2T0hZM09WTmxIMEZIU
lVKcU5FNDJkVvJCQ2lBZ1VrRXhibEJHTTFaNFJYZFNNMmRMVldjNVUwVQogIDV0bV
Z5ZUHRMWEyNXBZaz1tWkdGallUZHVwbEVpZ1Ywc0NpQWdJQ0FpUTJ4cFpXNTBUBt1
1WTJVaU9pQWlPCiAgVVGZUZaUk5tdGxNWEZIVVvaeFpuSTNRemN4UVNjc0NpQWdJ
Q0FpUVd0amIzVnVkrUZrWkhKbGMzTWlPaUEKICBpWvd4cFkyVkJaWghoYlhCc1pTN
WpiMjBpZlgwI10sCiAgICAiU2VydmVyTm9uY2Ui0iAiYXRMTFRfWFRDYlVQdm5iR2
lGtKi1USIsCiAgICAiV2l0bmVzcyI6ICJXR0FaLUJYMKEtN0c3Mi1KNVUZLVNHr1Q
tVUIzVy1UVjRMIn19",

{}

[illegible]

Service-Service (Post Transaction)

The Mesh Service receiving the message from the user's device **MAY** attempt immediate retransmission or queue it to be sent at a future time. Mesh Services **SHOULD** forward messages without undue delay.

The Post transaction forwarding the message to the destination service carries the same payload as the original request but is authenticated by the service forwarding it. This authentication **MAY** be by means of either profile or ticket authentication.

Missing example 18

Denial of Service Mitigation

Services **SHOULD** implement Denial of Service mitigation strategies including limiting the maximum time taken to complete a transaction and refusing connections from clients that engage in patterns of behavior consistent with abuse.

The limitation in message size allows Mesh Services to aggressively time out connections that take too long to complete a transaction. A Mesh Service that hosted on a 10Mb/s link should be able to transfer 20 messages a second. If the service is taking more than 5 seconds to complete a transaction, either the source or the destination service is overloaded or the message itself is an attack.

Imposing hard constraints on Mesh Service performance requires deployments to scale and apply resources appropriately. If a service is attempting to transfer 100 messages simultaneously and 40% are taking 4 seconds or more, this indicates that the number of simultaneous transfers being attempted should be reduced. Contrawise, if 90% are completinin less than a second, the number of threads allocated to sending outbound messages might be increased.

Access Control

The inbound service **MUST** subject inbound messages to Access Control according to the credentials presented in the DARE Message payload.

After verifying the signature and checking that the key is properly accredited in accordance with site policy, the service applies authorization controls taking account of:

*The accreditation of the sender

- *The accreditation of the transmitting Service
- *The type of Mesh Message being sent
- *User policy as specified in their Contact Catalog
- *Site policy.

Service-Client (Synchronization)

The final recipient receives the message by synchronizing their device. The message received will be appended to the inbound spool.

6.5. Publication

The Publication mechanism allows content to be published through a Mesh Account and retrieved by means of the EARL mechanism described in Uniform Data Fingerprint [[draft-hallambaker-mesh-udf](#)]. This mechanism is used in certain flows supported by the Mesh Device Connection and Contact Exchange functions.

Content is published by appending an entry to an account's Publication spool. The content may then be retrieved by issuing a claim to the account specifying the publication identifier that is authenticated under the value specified in the EARL.

Use of the Publication spool to post content necessarily requires that the content be smaller than the maximum message size imposed by the Mesh Service so that it can be uploaded to the service by means of a Transact transaction.

Publication of large data items will require modification of the protocol to support use of a detached message body. Transfer of a detached message body is outside the scope of this document.

6.5.1. Claim Transaction

The claim transaction is used to post a claim to a document published by means of an EARL. The claim interaction is used in the Static QR Code connection interaction but **MAY** be used for other purposes as required by Mesh applications.

A claim is made by sending a ClaimRequest message to the service to which the publication is posted. The service responds with a ClaimResponse message specifying the success or failure of the claim.

A device is preconfigured during manufacture and a Device Description published to the EARL:

The client claiming the publication creates a claim message specifying the resource being claimed and the address of the Mesh account making the claim.

```
{
  "MessageClaim":{
    "MessageId":"NAB5-YJPI-PTQM-LDCJ-73YB-YRMJ-EJTZ",
    "Sender":"maker@example.com",
    "Recipient":"maker@example.com",
    "PublicationId":"EBQA-UU6A-52ZG-L2KJ-2NZY-GZOP-KQ6B",
    "ServiceAuthenticate":"ABDM-BH6K-2J3F-S5PU-G30A-6UDM-S5CW",
    "DeviceAuthenticate":"AB0A-4PKL-7A5C-EW3R-7NNS-EOS4-XX5C"}}}
```

The message is signed by the claimant to make a RequestClaim to the service:

```
{
  "ClaimRequest":{
    "EnvelopedMessageClaim":[{
      "EnvelopeId":"MAT7-SJFB-SVYG-7UK6-ZAEP-PS27-E7LE",
      "dig":"S512",
      "ContentMetaData":"ewogICJvbm1xdWVJZCI6ICJ0QUI1LVlKUEktUFRRTS1MRENKLTczWUItWVJNSi1FS1RaIiwKICAiTWVzc2FnZVR5cGUiOiAiTWVzc2FnZUNSsYWltIiwKICAiY3R5IjogImFwcGxpY2F0aW9uL21tbS9vYmplY3QiLAogICJDbmVhdGVkIjogIjIwMjEtMTU0MTY2MzY2NDVhbn0",
      "ewogICJNZXNzYWdlQ2xhaW0iOiB7CiAgICAiTWVzc2FnZUlkIjogIk5BQjUtWUpQSS1QVFFNLUXEQ0otNzNZQi1ZUk1KLUVKVFOiLAogICAgIiNlbnRlciI6ICJtYWtlckBleGFTcGx1LmNvbSIsCiAgICAiUmVjaXBpZW50IjogIm1ha2VyQGV4YW1wbGUuY29tIiwKICAgICJqdWJsawNhdGlvbklkIjogIkVUUUetVVU2QS01MlphLUwyS0otMk5aWS1HWk9QLUtrNkIiLAogICAgIiNlcnZpY2VBdXR0ZW50aW9uIiwKICAgICJETS1CSZLLTJkM0YtUzVQVS1HM09BLTZVRE0tUzVhbn0",
      {
        "signatures":[{
          "alg":"S512",
          "kid":"MCVM-VUHL-UTSB-SV5M-VRD5-P5XF-QAEV",
          "signature":"kN-cd4dUkTGT6gjm3MVnuRPCHWYtGmqi2IbhaBmbagc3D6hmJgNTbGApAuVofNkPqc_jBhSpU-WA5NoDvH-e8dn0lUsMLxLUyFvs8VSnsPwFvdpMzypm4woMdnQV-7QofoqU7DBAMlJBjRlSiG8mdR4A"}
        ],
        "PayloadDigest":"pmdjNV2iacjFv2I_5rPfwbCHRV2G2xAaVvw1hb84J1-on4WiTf8GkSuEwM0hk__qH7qk19nroBGJYEszo-e-CsQ"}
      ]}
  ]}
```

The publication is found and the claim is accepted, the publication is returned in the response.

```
{
  "ClaimResponse":{
    "Status":201,
    "StatusDescription":"Operation completed successfully",
    "CatalogedPublication":{
      "Id":"EBQA-UU6A-52ZG-L2KJ-2NZY-GZOP-KQ6B",
      "Authenticator":"EBGU-ETMA-3DE7-CAHG-BVVE-QARV-4LVT-LKKT-IXR4
-FTJQ-JLWD-4RWV-TFFH-E",
      "EnvelopedData":[{
        "enc":"A256CBC",
        "kid":"EBQL-HMNU-PPID-5DNC-0A36-4Q4R-05RJ",
        "Salt":"YsZp_N4nuW0Av_NwsHZqqQ",
        "recipients":[{
          "kid":"EBQA-UU6A-52ZG-L2KJ-2NZY-GZOP-KQ6B",
          "wmk":"IZxxQwSP394qBo43PA2UuajT-fsFs-FDes1A9A0xxUw9
q_o44vw9Ig"}
        ]},
      "c-BCcKJKlzhEvj23CuFDw7XNNl28Vp6G8u0nU7786yumJMW9qNddCtT1
l2Ui6YxFqrczuEl56Mfr9dsRj66u0R4rzn0hsa9YOCj_9qkQmFQaVqbEqDLAGIc5y
WBD-VhUpPu1m0t4fPqIdB5eZ0d8jnuvJAFIaGqgtH5C8Dh55fCuKbwS3t3F0G6UgE
H8IDqg4Qh4jFA5UhmP7HfsQGHld0TroLmL_5raTmsLfLZ1ziULc-7Y7TL0MRL42a7
xvzlSmDGV0QQYvnhC0e_gH6h0Xjezu0eaI79FerzeZgeKHv309IO8D4apW4eg-65B
zn0-enj5XKjXmbtQ4jMlAD_iauowS6dwARK2ULV3YsXUGTSn_fwZFj8jMWXeMC0P8
5mZWzdBKHCy9V0cJvG-hqdsHt2y0eqU0eHlNZtKX1rjc5a1d_CopL9yi5JgZ-YJbK
XjahTUEwb88YGLkKvdk1j-S0M8DixJ1Q6w3FfrPZdi8FqSkGwfunIktPIr7i_yXsb
syZXFe00NHLdoSxV18-UMBmsUMphajm91MRFYK_JQK37My6Ra0VQkh8TA1CeYmLM6
Ur_i0eH2Q8-DEpvUhyr3Sv70PZdB3LbrKCFqNS0m_kquJtC0BXCp5ye62vFf9iJGh
icC04xRJngfg_0eWMK96wWFeLujMF_C238rs3TuL3dtK2m76rLfLtK76wrN8CwegJ
OSf0kUVwzTwzjsJR770PbeHJrCsA00XCkn_uZE81cJC1ivagAvyiuUpX6jBW-agwL
xGKGdsmYKA6kbpZLzAHIKD4x4fEI_EDF-bM-_0JNkTsiZ9XSkd0h8Vrid8_hU6BkP
meeHWWdpTpH7kv-FUFRaLe9MiXyR6ebqhtQWhnV9VXsV4EmC625owxyvt9nt5eRQa
9rpc_wb5HkqGrhXxHCns61awBZA5JmtHW3VsyEXOUAYFJjha_PKuitqzRIG2-yzxQ
1NBmFlxK8Q57tLaJGQYbEP05CgbutTZMmN8oJtm_82TZJJAsECX7gGVq-cUjeNky0
Zx3RF4PQwC76YJQF4TudVTw5VoGE7YU61ZY7Kj-cpWoxnny3-U-NTti4jm0gYU8XD
jk95bQhXe1ttXsqAt43NFs1zsQKJ348offS-5g52wPqs0wSroqruP77MyqyFx0Flq
hPHdNKQmTujikfDisqym3QRDwJYctcSM4ifHzVIGkXICKj-DeST9c-f5wDKWrB2m
sSJE1w8xDJctv_0tSfKgPb7nwhxzy7Ettjg0601CETKLpyrBQg3uH62sfDPleA5Uw
lpD2PF-GN3B10vJgm34tazM4U9CyNGhkoSxt_BxofedfgzLJTbw0zXGZPJLp6NRim
hAr4_pjmaeqTwJQjVY-2q2jsz2aY_5X65r4XS-A0I-paUhebXrj3sEiVaSa67KGNJ
rLoLLwr3EqfY90Pyus1ylh3kd13nEMeRMlPsxqwrjyuHiVKzVwfAEq-0e0HaT9VlV
XyfnLoCj60L0tNhYng154gq4Bj-8nuR2SZVPoRxVxOnM3yEgMlgNe6HLSUVAwen1S
NLh0EK8axBv9NfTEALVRxwAk10oDFz64SwgimIzgJJP402iFt60r_7ksZ1li7AjhC
t-GmjVYELwNP0UVEErHXAIZCt88TwV10SSFVhsV0rfA1eqbjNFqz1sLJjrF-osUKI
xGicbeb0-yhj5TCxtzWKbbp4FmfwrG0cdRS7m3qQlJB5_m2bR6P6qEYekuNxSxcC0
DWEn-9WfL0-hNnI03m0FvEYFSMN6MpBwyiTPsnDepbbZfD3yJMUUzs3IIaQwI3Wf
OUEicFB_Mc16KRE1IndUSOqjtdUGNJ37UhBvEfzYmUzWHFKHnUVvvMFjVtD1kh_F8
psgrYwg6pu1KS6cXttLzYNpu1i3_4uzsbksZLF-S5M-wx4xy9858YazjJ-bRUDg4a
TQ_CHKtYV3m4tr06lgCAa4aVt32kzPBve9pt-YxxMT7gsGEszmRphZ1kq4miaMnr-
8zjQn7CKq853v6K1HzCsJA8avjUDWb5mfqvI1u9hxrDZ866guMatMgqnI_jU78VqE
```

```
_a3r17C8KfH8mH8fy1w5VMKqCs1CRJ0zCLgv8_5YdcUDcn1I7ytMdagFXN-gc5a0E
RBSpr9sxcwdGX5p9b40C0Ci1oEza0sfUoNHYPM8WHyQHJFPjcLsOH60S1ydtZKta
nAFtBPs_PLNm0aPylqAGMn-rTV14yy68A9tdBgKqZMCZzDaHHYzYdNS25EaYFT-G7
5KiiF0ARAGMIwAVM5WcEfMbFq-SZXsxKDxMxfXJ00mPi09xBx2BXwRdAN5Qr1BZHi
kIRvs5FT-PFPuE3MyYpWAbhlc6PPBmam0ma-L04S8jkFPMjpfQ33JT00Iytrz2I_
qhQq27wAGy0ZeILdXbJCMFzr0Ms_kB6ZVYGnteSsrk04Qu9rQsb3JnV0uKnqwHh0E
pTx2Uus6__JdWv8JUizt5tPo13u8ynI0kCxnAPRXDhBHbHD8nTtNYkNB3cpczToQh
NmLe8-BN7tqZ3anErBOMR6N39B0sANJyWrom_pWEt52ZYsaqEz75I-GFZFxfst-M6
NDywxXRrcJDXVmpSnGCxAb9IrFR07aQqn5fh0ATok3onsTyz2ArshZsDFxLi2sNR
fRct1ZUpge_ryyQu0ex4ItwnAtsJIw8pTnTBkzoDUAmA43VVeAC_pr9QPAY_4B0c8
gvdL87fuYbyr_L_VRhTzp3XMMjfAbKjPXdIkNiRKvkbGr9Py0b1riUwd8uLffzvds
w9QQiV7GuP3_AwwIOu0ijk0ig62Ewb7bGA1K-_Zyka5PCDjAMEwKVSbI-Av7-dSu-
bu28TgGk8T0f3RNDQR7drp-zCCT7Kbz5CtAUR8SkB7USqU0Vng7Ygo5d61Fufx-jz
dvZhaTgcbcd69qtxKqXDb9uhGCyKBLP5sDjx-aHX-q7WzFs8UptsD9JwsUM1HJHB3
klH8xloE9zNtmTpq9j7GCj4RvmWfswHkEfDDGxqipiXFixpYowMp2tu2WxGt1d80A
xUMkY"
    ]}}}
```

The device waiting to be connected uses the PollClaim transaction to receive notification of a claim having been posted.

6.5.2. PollClaim Transaction

The PollClaim transaction is used to discover if a claim has been posted to a published document.

When an authenticated, authorized request is made, the service responds with the latest claim posted to the publication.

The device in the example above periodically polls the service to which the device description is published to find if a claim has been registered.

The PollClaimRequest contains the account to which the document is published and the publication ID:

```
{
  "PollClaimRequest":{
    "PublicationId":"EBQA-UU6A-52ZG-L2KJ-2NZY-GZOP-KQ6B",
    "TargetAccountAddress":"maker@example.com"}}
```

The response returns the latest claim made as signed message:

```

{
  "PollClaimResponse":{
    "Status":201,
    "StatusDescription":"Operation completed successfully",
    "EnvelopedMessage":[{
      "PayloadDigest":"pmdjNV2iacjFv2I_5rPfwbCHRV2G2xAaVvW1hb84
J1-oN4WiTf8GkSuEwM0hk__qH7qk19nroBGJYEsz0e-CsQ",
      "EnvelopeId":"MDRR-M6ZK-YCM4-CIZB-PT7L-K5NX-2MOS",
      "dig":"S512",
      "signatures":[{
        "alg":"S512",
        "kid":"MCVM-VUHL-UTSB-SV5M-VRD5-P5XF-QAEV",
        "signature":"kN-cd4dUkTGT6gjm3MVnuRPCHWYtGmqi2IbhaBmb
agc3D6hmJgNTbGAPAuVofNkPqc_jBhSpU-WA5NoDvH-e8dn0lUsMLxLUyFvs8VSns
PwFvdpMzyp4woMdnQV-7QofoqU7DBAMlJBjRLSiG8mdR4A"}
      ],
      "ContentMetaData":"ewogICJvbm1xdWVJZCI6ICJ0QUI1LVlKUEktUF
RRTS1MRENKLTczWUItWVJNSi1FSlRaIiwKICAiTWVzc2FnZVR5cGUiOiAiTWVzc2F
nZUNsYWltIiwKICAiY3R5IjogImFwcGxpY2F0aW9uL21tbS9vYmplY3QiLAogICJD
cmVhdGVkIjogIjIwMjEtMTNUMTY6Mzg6NDVaIn0",
      "SequenceInfo":{
        "Index":1,
        "TreePosition":0},
      "Received":"2021-01-13T16:38:45Z"},
      "ewogICJNZXNzYWdlQ2xhaw0iOiB7CiAgICAiTWVzc2FnZUlkiJogIk5BQj
UtWUpQSS1QVFFNLUXEQ0otNzNZQ11ZUk1KLUVKVFOiLAogICAgIlNlbnRlciI6ICJ
tYWtlckBleGFTcGx1LmNvbSIsCiAgICAiUmVjaXBpZW50IjogIm1ha2VyQGV4YW1w
bGUuY29tIiwKICAgICJqdWJsawNhdGlvbklkiJogIkVCUUEtVWU2QS01MlphLUwyS
0otMk5aWS1HWk9QLUtRNkIiLAogICAgIlNlcnZpY2VBdXR0ZW50aWNhdGUiOiAiQU
JETS1CSDZLLTJkM0YtUzVQVS1HM09BLTZVRE0tUzVDVyIsCiAgICAiRGV2aWNLQXV
0aGVudGljYXRlIjogIkFCT0EtNFBLTC03QTVDLUVXM1ItN050Uy1FT1M0LVhYNUMi
fX0",
    {}
  ]}]

```

6.6. Cryptographic

The Operate transaction is used to perform one or more cryptographic operations using private key material recorded in the Threshold Catalog. Such operations typically represent one part of a threshold key operation divided between the service and a device connected to an account.

As with all operations involving the Threshold catalog, the request **MUST** meet the authentication criteria specified by the catalog entry. These typically include the request being authenticated by a specific key.

6.6.1. Generate Key Shares

`CryptographicOperationShare` is used to request that a private key held by the service to be divided into two or more key shares. One key share is then encrypted under the encryption key of the service and the others are encrypted under public keys specified in the request. These parameters are returned in a `CryptographicResultShare`.

The request payload:

```
{
  "TransactRequest": {
    "Accounts": ["bob@example.com"],
    "Outbound": [
      {
        "EnvelopeId": "MAJX-UL35-3S2N-HIT4-3LND-YN12-QWFP",
        "enc": "A256CBC",
        "kid": "EBQI-LVNF-YZQ5-YNGA-CB36-SPXN-BHSX",
        "Salt": "mSGcWKixJS60ikGr1lZ8Wg",
        "recipients": [
          {
            "kid": "MBLV-2ICB-OATF-HHQH-OTA7-E6GC-TH76",
            "epk": {
              "PublicKeyECDH": {
                "crv": "X448",
                "Public": "4Jrht04zIboZtGL0oWQiBh6AUyVQiR1PZJZA_yOPOqow6feU73nsjZETICBkcVLjtMVdlTm7LzmA"}},
              "wmk": "rIgRDmiPg73rzanZcO49MbBs2GgS-Ab9gHkkkt-bzoZa
a7MFtU347Q"}
            ],
            "ContentMetaData": "ewogICJVbm1xdWVJZCI6ICJ0QktFLVZSVDct
TUVaVC00SU9WLTZLT1ItVDNGTy1aUk1FIiwKICAiTWVzc2FnZVR5cGUiOiAiR3Jvd
XBjbnZpdGF0aW9uIiwKICAiY3R5IjogImFwcGxpY2F0aW9uL21tbS9vYmplY3QiLA
ogICJDCmVhdGVkIjogIjIwMjEtMTNUMTY2Mzg6MjlaIn0"}},
            "NZ7vOnWpbnesaFkr4PNOCacqFqemmIPbZhehX6t1GkP3CpUnMws_LWZZ
-2T_tL14_IoTW57466pSHCEpABY1unesf0xtFXcw6kXD9Cfj-JxkrsjmrWRAp7LC-
CQcPnuhtNmUJ0ygUJSH1gTnr5p7amQSdtT2LHsIZ6pMpgPsm3swc0iTb1WmPyVeue
IUChyNfGk_y3BNBo4I4B3nLMLNHIMvrdgFS-dGSfbq1RKRaZLVPxxCOn7ioqigBM8
pwP7yzeU9z_NpEqhxlGSmEKK30C9PAHQDdLM05xIaEw6dz6WET_pRPzU8IhYEPsN1
Un6i4Z9sp1cIto18wyKvrZWojBtE2JKdXgdKuBVW83ppIUUV-p0GhYa_6GR6rS4eQ-
SExyX3GaFktVoDTK6rNtlj0a8_Q9-3A03nb1yzbyPtgi0pQ0Uhk-7lJHKhcN_jY-4
ACHVbfUwn5GNseMc-79QvgL8F0RkdLdLTTsijixm2oattZukSAiyLTSwQ1HIMB1g
lHXj-Bga_4oQDuoEUoqzXKwUbXIsH9pdvqGK3Pd8X_2j2BLfmlefGZILwigzCzXaN
KZmR4Jt2H4mwWtk2t1-2k7Rmh8p26cMe4aBQGqQmWIpDoERcmuI22Sn3tfSJXBQkV
s8wnp8Riztj5zPMCGfWxHAI3nUPGVrwJgmyTWaHPLewUSqHzbie5uM-qytqbR6JZw
_FvkIg8RbCfggxoaJ98q7LXDQ63xUYbhlSeM8CrGcCcUPHEanfKxsDwBFX0GHNqre
T8fGjV3P_6w00WTe8fYD3_dazywf4ZV_8k0g0dm6Fdnq--mb2SeYLikt45fF0osxr
P6cFGakXMazrWdaEW031qUEczzip_APgxgZ-2M0V53eE2rjigDejIwpDVDHZZL7Rc
C40J-wS7sstLOGithbWLnaJOB5SP-Q5-psoq8a2-R-PO5BHOjZ1K_CXDmM4lVhmsF
E4bw3NAqV0CDubZRg8HiqY9I3w-iuQsAXjUft55ndhamknVPPZYl20-Q0ej6CIMl9
OigomwWx-Fs5vrxM2R_9vMs4uhREfcQCH3Vybn0S6hUctod7v-1iwHg0LG3uiI4Vp
XmG3P1fcgUzIhPfbUDS4kFHktc4IQysTjDg0-c3iYugDRvYwpoyEmjgZlLT353_2U
QJNB3B0r4wKDPjuW85wyRQPkHRR8QVit5Tc6RGw6e2EV7N4pJKaje-BE2vo1jE_zz
_rQohgjPsgRv7k0w9FLwxcFakXhCq1gJdHlRmonixkjkHZtfMnNFkztOY2LCG3cHy
bDLn5KAI1jAhiCeMe36haFNWUy3R0b7TXviGLSg6-0AQR2p965-kvMurXh2cfbJwL
fT6QUTQKp8Qj0oDsHKWAFJbcIYW72PiQ4yVYNQiEzGESckMs31AmdEtaikW7__tGa
sZeM4irzzUZb_UAuIltI3lnUrbLwNt2G1nu9-WP1_T3PEPHTwy2cw6Byi9wmtVFMA
rrsZXMM6CwfoYFAhw1BeuL-Em-FtRTsqawVbNaRr1AkB70cum8m8hrrepBVneNSXQ
U6SVJKCEI3QNrTcmj0BT6PkM6z51ahu8LBFg0lYwhLM6qqY0osP3VN-UDftcL23ia
Ln2MWUmf1GkH1myLDy0ABWYcQPSa8pkHnllv tpENMB0tj8lkPunSjLbdiSI7P4pb
```

2Z0oG7lQEw1l1K5NG2mZzzY4d90QivirxBaHL-ucID0jCvjNj7XeU1qsqg61IXScd
STIFirKu7DWceJ-t30FBEhxQ9NioV9LmuPEDkfKsnJg3Jqhj6TK7-v7FibtmEA16W
uK-duapE2QWDLXed_J-91MLrYiLwc02EjHnp5l3BZ61WBxqmrG4HfeUu7hGRhCBxF
o7pMQwclb4s3XdxzcZpkHm4-ZsorpWkzwPKzSw3Bpag0ISmqyORPBUIDodiGDZtDM
GDUMEEnvlSPtp2kVhJYUbkawng0t59qriLaZefH4J1Sats3yJkqWzyVqaEFmN0u6K
HQdJPGdu9ZZc8g6rzSUPzKgbp1IXuKcHbWyg98n1-ofqZuBc8ELzeux00s9Vja7wc
nA2Q-PK56G5LrMKIQAztqavwz8ggsEeAzkvgPUIIDNhE9zKTZYR4FN0pC0Fh-0Tbu
6E5CfADPrLTT9GJrPOJKKbq9siXtn0xoJnvcT10Zp7GDHi9HN7esi0m00LoRHELvY
COYQGVzey6puHu2TZdwilS70MhFn2EBVPHLS5iQn208_Rm7Mg5KhBwjkgV8MWI_Cn
L965GATILbXZHiNhIajTND-D9aJ2BonA-0ljeUeAdFzuAfb9RPgu5yX03oX7Zqdx
Gz1HuQErNPvd5XU-2_xKnEK8djQB4fNV4V0UpaFrURTFcd4xX4MKSaJkPuU3s22v
zoxfl6iwhf3NlnWp0_xVPrDYQfUs9Zez3FADjMXj4ZY-Xh7nnqDU_TuHGJs00GBVM
77LuG-7WIOu0exjPV96we82YVMh8N1GYdr0W78nRwRysLQI-ryB_1Jp1pDIFUIh0
UsW0IFqCDuF79hjz18820ApTd3tFI-E2GAHyc3HTAVCpzNGBAj5EjIXWxlkxgfg8o
NjmrDCR1JjuuFILCtQwTb9dHty0ZiDZxdU8-5aTybcBvGgNw03fQp4ywF1XEH9w1s
Sv8U-31ms0ZKTyb6kGP3-VUD2EcMq0MlkvReFA6ayaw9j1snnyzqG0w29LGJEdhPo
h-ti0-tDf3_DNwmNORx8pK0C_xHWTQpSqmnasIFQVgcqNbMbtcwGgtNJZ1ZhEtVFj
An-tDKmbxlXt7zUpCfBZg0C2n804TzvpRI90mMwkKC3x8JfEdmkt0xn2dUbIOS8pH
LttErF01NVXox_006nW_Sgj0jdmbRv0fYdR_uqbxZM5dAmfRzvE7Xc6ZoecydxafJ
RH3fbMF3109KmA0yWExInjM4rQ1dLkfbcd011-VGt235BdGkZEGkAbME15j12Mv3b
2NYmwu4AU0j16VQuFBfV-GICHMgdsX23N0LEvhVQZbFr9E10MJ_zjW5lNElCdjlHL
dvV8tPbhi9ehBxxa8Tuq7e4ivgM70BFeYjcPnUcHz0_9Lp2wyuGXMiAaUAdldsgaj
r8iygilwK74CLEsAm0WUzP5PrIDIBeLHEVh8sGEeLCRFmVs_2uW46x1j4wC00-x3z
AHe1txYYv3rsUwBawPDVMW-gd33npWbaVjGcY1WF70a0jX9_2GtJNFbXYF2WTNbz
gc-2R6fRfa1jgIT6YVS-t47c4b_oHprXm9nQwG7xLcFqQxq1WkMLe6MEWzywn8gTc
IATvbQ0vB1ETcrMZXVLkCGny7PwvVVQUqxo7egotMB92MtL0kd9XADzGCTwLByFv3
olFb4Q7xu2Zxxpg0dwlHq52_H3SsHhDeanbuJ2mG69_CfbzTC1piZrQ1Wx7LJuRTV
K83Ag-ooBb-659GJGHYAo_oNponYHFG4c7W9DgoUk2HE-t7whx19Wy1AWrCG1ldu2
KVz6wF9J7hi_0t4jsLNBxdGtrNs-W6DvP3uw0Cwph4k1oLVX4kCj3hVqa8bHT5lQz
pT_e5let_kRXjzaZE1uBkSlGdDckqJ5-_riWTbzAGXR2tAhB1TOFzvJ1dYcM0xPdA
hNLiaALHNcTDROcEW-4kisC0taDwDBNPH>IfudImYHG4D6JC3HCpsLMLbXwY7tUj8
VT8F2499UAuB-rK6W6eeH1wTpXjog0Cxc3znrypyHDepUBZj8GxfmWqMJtPRXQWtp
fKCVHQ0aKpPh88-VfhgB2LI0EKtRUHdPasR3deK9AR5wZM8amg00A7WB2nJ9pNe5V
JFWAlRpUYc9Bb6LRdIz8lz3M8SuyD4jA0pItw770muTv-vgIvlyzoL5nYWy4N1RJy
2eITl-UYjv_9cFYJJUS8K6ghYiV2vvzu-WkbJjLSRzgvSehFe8em7vqKLEE6TTRb2
rLgX0mMdwvLm26wRxXoAfmFwr7MYeznIPooCLE00SusX4te1JIIId1tMqg7xPqHmT8
dVjRBk9J8i00C6wgugGq0TPU3xUU5E5TK7rkyZRbuoSB9d6jyke7qgr1t2oqoAI0F
I4HZir1d6AZ6MQXlW-4llcqB0ivfSqiGxDPviq6whzuMgwdejpj8pHFpM_4krTR7k
G57mrSDkit2eTmVFvSmQnYxfpdUC1DFEKTGFG92dkSLD-ooPw-eBFwokmHgrP6MjX
JQ_-lim0w_qh1ku8tpA02r6j6NMAdXvGFajIR9g-ADRKiHwURcSxCfNlHfrIAi9BF
wv7F0sq_Cun22JqvFvNwVS9STu9ow"

] } }

The response payload:

```
{
  "TransactResponse":{
    "Status":201,
    "StatusDescription":"Operation completed successfully"}}}
```

6.6.2. Key Agreement

CryptographicOperationKeyAgreement is used to request a threshold key agreement operation on a specified public key.

The request payload:

```
{
  "DownloadRequest":{
    "Select":[{
      "Container":"MMM_Inbound",
      "IndexMin":2,
      "IndexMax":3}
    ]}}
```

The response payload:

```
{
  "DownloadResponse": {
    "Status": 201,
    "StatusDescription": "Operation completed successfully",
    "Updates": [ {
      "Container": "MMM_Inbound",
      "Envelopes": [ [ {
        "EnvelopeId": "MAJX-UL35-3S2N-HIT4-3LND-YNI2-QWFP",
        "enc": "A256CBC",
        "Salt": "mSGcWKixJS60ikGr1lZ8Wg",
        "recipients": [ {
          "kid": "MBLV-2ICB-OATF-HHQH-OTA7-E6GC-TH76",
          "epk": {
            "PublicKeyECDH": {
              "crv": "X448",
              "Public": "4Jrht04zIboZtGL0oWQiBh6AUyVQiR1PZ
JZA_yOP0qow6feU73nsjZETIcBkcVLjtMvd1Tm7LzmA"}},
            "wmk": "rIgRDmiPg73rzanzC049MbBs2GgS-Ab9gHkkkt-b
zoZaa7MFtU347Q"}
          ],
          "ContentMetaData": "ewogICJVbm1xdWVJZCI6ICJOktFLVZS
VDctTUUVaVC00SU9WLTLT1ItVDNGTy1aUk1FIiwKICAiTWVzc2FnZVR5cGU0iAiR
3JvdXBjb2ZpdGF0aW9uIiwKICAiY3R5IjogImFwcGxpY2F0aW9uL21tbS9vYmplY3
QiLAogICJDcmVhdGVkIjogIjIwMjEtMTNUMTY2MDZmZjIjLn0",
          "SequenceInfo": {
            "Index": 2,
            "TreePosition": 426,
            "Received": "2021-01-13T16:38:29Z"},
          "NZ7vOnWpbnesaFKr4PNOCacqFqemmIPbZhehX6tlGkP3CpUnMws_
LWZZ-2T_tL14_IoTW57466pSHCePABY1unesf0xtFXcw6kXD9Cfj-JxkrsjmrWRAP
7LC-CQcPnuhtNmUJ0ygUJSH1gTnr5p7amQSdtT2LHSIZ6pmpgPsm3swc0itB1wMpY
VeueIUChynfgk_y3BNBo4I4B3nLMLNHIMvrdfGS-dGSfbq1RKRaZLVPxxC0n7ioqi
gBM8pwP7yeU9z_NpEqhxlgSmEKK30C9PAHQDdLM05xIaEw6dz6WET_pRPzU8IhYE
PsN1Un6i4Z9sp1ciTo18wyKvrZwojbtE2JKdXgdKuBVW83ppIUV-pOGhYa_6GR6RS
4eQ-SEXYX3GaFktVoDTK6rNtljOa8_Q9-3AO3nb1yzbyPtgi0pQOUhk-7ljHKhcN_
jY-4ACHVbfUwn5GNseMc-79QvgL8F0RkdLDLTtsijxm2oattZukSAiyLTSwQ1HI
MB1glHXj-Bga_4oQDUoEUoqzXKwUbXIsh9pdvqGK3Pd8X_2j2BLfmlefGZILwigzC
zXaNKZMr4Jt2H4mwWtk2t1-2k7Rmh8p26cMe4aBBQGqQMwIpDoERcmuI22Sn3tfSJX
BQkvVs8wnp8Riztj5zPMCGFWxHAI3nUPGVrwJgmyTwahPlEWUSqHzbie5uM-qytqbR
6JZW_FVkIg8RbCfkgxoAJ98q7LXDQ63xUYbhLSem8CrGcCcUPHEanfKxsDwBFX0GH
NqreT8fGjV3P_6w0OWTe8fYD3_dazywf4ZV_8k0g0dm6Fdng--mb2SeYLikt45ffO
osxrP6cFGakXMazrWDaEW031qUEczzip_APgxgZ-2M0V53eE2rjigDejIwpDVDHZZ
L7RcC40J-wS7sstLOGithbWLnaJOB5SP-Q5-ps0q8a2-R-P05BHojZ1K_CXDmM4lv
hmsFE4bw3NAqVOCDubZRg8HiqY9I3w-iuqsAXjuft55ndhamknVPPZYl20-Q0ej6C
IMl90igomwWx-Fs5vrXm2R_9vMs4uhREfcQC83Vybn0S6hUctod7v-1iwHgOLG3ui
I4VpXmG3P1fcgUZihPfBUds4kFHkktc4IQysTjdgo-c3iYugDRvYwpoyEmjgzLLT35
3_2UQJNB3B0r4wkDPjuW85wyRQPkhRR8Qvit5Tc6RGW6e2EV7N4pJKaje-BE2vo1j
E_zz_rQohgjPsgRV7k0w9FLwxcfAkXhCq1gJdHLRmonixkjKHZtfMnNFkzt0Y2LCG
3cHybDLn5KAII1jAhiCeMe36haFNWUy3ROb7TXviGLSq6-0AQr2p965-kvMuRXh2cf
```

bJwLft6QUTQKp8Qj0oDsHKWAFJbcIYW72PiQ4yVYNQiEzGEsckMs31AmdEtaikW7_
_tGasZeM4irzzUZb_UAuILTI3lnUrbLwNt2G1nu9-WP1_T3PEPHTWy2cW6Byi9wmt
VFMArrsZXMM6CWfoYFAhw1BeuL-Em-FtRTsqawVbNaRr1AkB70cum8m8hrrepBVne
NSxQU6SVJKCEI3QNrtcmj0BT6PkM6z51ahu8LBFg0lYwhLM6qqYoosP3VN-UDftcL
23iaLn2MWUMf1GkH1myLDy0AbWYcqPSa8pkHnllv_tpENmB0tj8lkPunSjLbdiSI7
P4ph2Z0oG7lQEw1l1K5NG2mZzzY4d90QivirxBaHL-ucID0jCvjNj7XeU1qsqg61I
XScdSTIFIrKu7DWceJ-t30FBEhxQ9NioV9LmuPEDkfKsnJg3Jqhj6TK7-v7FibtmE
A16WuK-duapE2QWDLXed_J-91MLrYiLwc02EjHnp5l3BZ61WBxqmrG4HfeUu7hGRh
CBxfo7pMQwclb4s3XdxzcZpkHm4-ZsorpXWkzwPKzSw3Bpag0ISmqyORPBUIDodiGD
ZtDMGDUMEENv1SPTp2kVhJYuBkaWng0t59griLaZefH4J1SAts3yJkqWzyVqaEFmN
0u6KHQdJPGdu9Zzc8g6rzSUPzKgbp1IXuKcHbWyg98nl-ofqZuBc8ELzeux00s9Vj
a7wcnA2Q-PK56G5LrMKIQAZTqavwz8ggsEeAzkvgPUIIDNhE9zKTZYR4FN0pC0Fh-
OTbu6E5CfADPrLTT9GJrPOJKKbq9siXtn0xoJnvcT10Zp7GDHi9HN7esi0m00LoRH
ELvYCOYQGVzey6puHu2TZdwiLS70MhFn2EBVPHLS5iQn208_Rm7Mg5KhBwjKGV8MW
I_CnL965GATILbXZHiNhIajTND-D9aJ2BonA-0lJeUeAdFzuAfb9RPgu5yX03oX7Z
qdxTGz1HuQErNPv1d5XU-2_xKnEK8djQB4fNV4V0UpaFrURTFcd4xX4MKSaJkPuU3
s22vzoxfl6iWhf3NlnWp0_xVPrDYQfUs9Zez3FADjMXj4ZY-Xh7nnqDU_TUhGJs00
GBVM77LuG-7WIOu10exjPV96we82YVMh8N1GYdr0W78nRwRysLQI-ryB_1Jp1pDIF
UIh0UsW0IFqCDuF79hjz18820ApTd3tFI-E2GAHyc3HTAVCpzNGBAj5EjIXWxlKxg
fq8onJmrDCR1JjuuFILCtQwTb9dHty0ZiDZxdU8-5aTybcBvGgNw03fQp4ywF1XEH
9w1sSv8U-31ms0ZKTyb6kGP3-VUD2EcMq0MlkvReFA6ayaw9j1snnyzqG0w29LGJE
dhPoh-ti0-tDf3_DNwmNORx8pK0C_xHWTQpSqmnasIFQVgcqNbMbtcwGGtNJZ1ZhE
tVFjAn-tDKmbx1Xt7zUpCfBZg0C2n804TzvpRI90mMwK3x8JfEdmkt0xn2dUbIO
S8pHLttErF01NVXox_006nW_Sgj0jdmbRv0fYdR_uqbxZM5dAmfRzvE7Xc6Zoecyd
xaFJRH3fbMF3109KmAOyWExInjM4rQ1dLkfbcD01l-VGt235BdGkZEGkAbME15j12
Mv3b2NYmWu4AU0j16VQuFbFv-GICHMgdsX23N0LEvHvQZbFr9E10MJ_zjW5lNE1Cd
j1HLdv8tPbhi9ehBxxa8Tuq7e4ivgm70BFeYjcPnUchZ0_9Lp2wyuGXMiAaUAdld
sgajr8iygilwK74CLEsAm0WUzP5PrIDIBeLHEVh8sGEeLCRFmVs_2uW46x1j4wC00
-x3zAHe1txYYv3rsUwBawPDVMW-gd33npWbaVjGcY1WF70a0jX9_2GtJNFbXYF2WT
NbzbGc-2R6fRfa1jgIT6YVS-t47c4b_oHPRxM9nQwG7xLcFqQxq1WkMLE6MEWzywn
8gTcIATvbQ0vB1ETcrnZXVLkCGny7PwvVQUqxo7egotMB92MtL0kd9XADzGCTwLB
yFv3o1Fb4Q7xu2Zxxpg0dw1Hq52_H3SsHhDeanbuJ2mG69_CfbzTC1piZrQ1Wx7LJ
uRTVK83Ag-ooBb-659GJGhYao_oNponYHFG4c7W9DgoUk2HE-t7whx19Wy1AWrCG1
ldu2KVz6wF9J7hi_0t4jsLNBxdGtrNs-w6DvP3uw0Cwph4k1oLVX4kCj3hVqa8bHT
5lQzpT_e5let_kRXjzaZE1uBkSlGdDckqJ5-_riWTbzAGXR2tAhB1TOFzvJ1dYcM0
xPdAhNLiaALHNcTDroCEW-4kisC0taDwDBNPH>IfudImYHG4D6JC3HCpsLMLbXwY7
tUj8VT8F2499UAUB-rK6W6eeH1wTpXjog0Cxc3znrypyHDepUBZj8GxfmWqMJtPRX
QWtpfKCVHQ0aKpPh88-VfhgB2LI0EKtRUHDpasR3deK9AR5wZM8amg00A7WB2nJ9p
Ne5VJFWAlRpUYc9Bb6LRdIz8lZ3M8SuyD4jA0pItw770muTv-vgIv1YzoL5nYWy4N
1RJy2eITl-UYjv_9cFYJJUS8K6ghYiV2vvzu-WkbJjLSRzgvSehFe8em7vqKLEE6T
TRb2rLgX0mMdwvLm26wRxXoAfMfWr7MYeznIPooCLE00SusX4te1JIIId1tMqg7xPq
HmT8dvJRBk9J8i0OC6wgugGq0TPU3xUU5E5TK7rkYZRbuoSB9d6jyke7qgr1t2oqo
AI0FI4HZir1d6AZ6MQXlW-4llcqB0ivfSqiGxDPviq6whzuMgwdejpj8pHFpM_4kr
TR7kG57mrSDkit2eTmVFvSmQnYxfpdUC1DFEKTpgF92dkSLD-ooPw-eBFwokmHgrP
6MjXJQ_-lim0w_qh1ku8tpA02r6j6NMAdXvGFajIR9g-ADRKiHwURcSxCFN1HfrIA
i9BFwv7F0sq_Cun22JqVfVnWVS9STu9ow",

{
}

```
    }}  
  }}
```

6.6.3. Sign

Threshold signature is not currently supported.

7. Message Interactions

Message interactions are asynchronous interactions that occur between devices connected to the same account or between accounts.

All messages are signed by the sender and encrypted under the encryption key of the recipient if this is known to the sender.

7.1. Message PIN Interaction

The Message PIN Interaction is used to register and validate PIN codes used to authenticate certain transactions. This interaction allows a PIN code issued by one device to be consumed by another allowing for greater convenience in managing devices or contact exchange.

For example, Alice might delegate the PIN code issue privilege to her mobile device without delegating the administration privilege to that device. This would allow Alice to use her mobile device to initiate the connection of a large number of devices to her Mesh as her house is being built and approve them later using her administrative device.

Use of the Message PIN interaction is optional. An application that issues a PIN code to authenticate a message **MAY** store the PIN value within the application without persisting it to external storage.

Derivation of the SaltedPin, MessageId and Witness values from their respective inputs is described in the Schema Reference [[draft-hallambaker-mesh-schema](#)].

7.1.1. Registration

To register a PIN code to an Account, a device:

- *Generates the PIN code value
- *Calculates the SaltedPin value for the specified Action
- *Calculates the PinId binding the specified SaltedPin to the Account.

*Creates and signs MessagePin containing the SaltedPin , Action and Account values with the MessageId value PinId.

*Appends the MessagePin value to the Administration Spool of the Account.

Note that this construction provides limited protection against forgery attacks by a party with access to the MessagePin. A party with such access can use it to construct the witness value required to authenticate a request.

PIN Code values consist of an opaque sequence of octets represented as a UDF nonce value. Codes are presented in canonical UDF form, i.e. Base32 encoding separated into groups of 4 characters. The PIN value is converted to binary form for calculation of the SaltedPin, thus ensuring that the canonical form of the PIN value is used.

7.1.2. Authentication

The PIN Code value is passed out of band to a user who will enter it into a device to authenticate a request made to the issuer.

A request that **MAY** be validated by means of a PIN is a subclass of MessagePinValidated and contains the following fields:

AuthenticatedData A DARE Envelope containing the data that is authenticated.

ClientNonce A nonce value used to prevent certain replay attacks.

PinId Digest value binding the SaltedPin to the Account.

PinWitness Witness value calculated as KDF (Device.UDF + AccountAddress, ClientNonce)

The device uses the PIN code and Action identifier corresponding to the desired request to calculate the SaltedPin value in the same manner as during registration. This value is then used to calculate the PinId and PinWitness values.

7.1.3. Validation

The PIN code is validated by performing the steps of:

*Calculating the SaltedPin value from the PIN code and Action

*Calculating PinId from SaltedPin and Account

*Retrieving a MessagePin from the Administration spool with the MessageId PinId.

*Calculating the PinWitness value from SaltedPin, ClientNonce and AuthenticatedData and checking this matches the value specified in the message.

*Performing the requested action.

*Posting a Complete message to the Administration Spool of the Account marking the PIN code as used.

This process can fail at multiple points resulting in different error results:

PinInvalid No PIN code is specified, the Pin code indicates an unsupported algorithm or the calculated PinWitness does not match the one specified by the request.

PinUsed The PIN code has been used previously.

PinExpired The PIN code is no longer valid.

Note that in the case that an attempt is made to reuse a PIN, it is not automatically the case that the first use of the PIN was the one that was valid and only the second attempt was invalid.

Implementations **SHOULD** alert the user to the attempted re-use so that this possibility can be considered and appropriate action taken.

7.1.4. Example

Alice connects a device using a QR code presented by her administrative device.

The administration device creates a PIN code and records it to the Local spool:

```
{
  "MessagePin":{
    "MessageId":"AAAU-6HVM-7AA6-TNEG-PFUS-IXZ3-BMYL",
    "Account":"alice@example.com",
    "Expires":"2021-01-14T16:38:30Z",
    "Automatic":true,
    "SaltedPin":"AANT-HAQW-KDUY-GUMW-YFY2-YWB5-23YC",
    "Action":"Device"}}

```

This pin value is used to authenticate the connection request from the device:

```
{
  "RequestConnection": {
    "MessageId": "NBPS-TE2K-5BQZ-3HWA-PJNU-WYOX-DAMW",
    "AuthenticatedData": [
      {
        "EnvelopeId": "MCNR-5XZX-CCZC-XTMJ-RYP2-PACV-HCBM",
        "dig": "S512",
        "ContentMetaData": "ewogICJvbm1xdWVJZCI6ICJNQ05SLTVYWlgtQ0NaQy1YVE1KLlVJZUDItUEFDVi1IQ0JNIIiwKICAIWVZc2FnZVR5cGUiOiAiUHJvZm1sZURldm1jZSIsCiAgImN0eSI6ICJhcHBsawNhdGlvbi9tbW0vb2JqZWNOIiwKICAIQ3JlYXRlZCI6IClYMDIxLTAxLTEzVDE2OjM4OjMwWiJ9"}],
        "ewogICJQcm9maWxlRGV2aWNlIjogewogICAgIlByb2ZpbGVTaWduYXR1cmUiOiB7CiAgICAgICJvZGYiOiAiTUNOUi01WfPYLUNDWkMtWFRNSi1SWVAYLVBbQ1YtSENCTSiIsCiAgICAgICJQdWJsawNWQXJhbWV0ZXJzIjogewogICAgICAgICJQdWJsawNLZXlFQ0RIIjogewogICAgICAgICAgImNydiI6ICJFZDQ0OClSciAgICAgICAgICAIuHVibGljIjogInRpdWhzU3Y3dnRaaGZBSGRNUjJUNncyMzg3bG9CZF9lMFhpe19xaDNNQUlwZ2QyVW9FdZQKICBqYzFuaXJhY3NERUJjS0xYRGJpOxVxeUEifX19LAogICAgIk1hc2VfBmNyeXB0aw9uIjogewogICAgICAIuVWRmIjogIk1EUUUtQ09CUy1KUDVDLTl3UEYtSkEzVS1DNk5WLUdKSEMIaogICAgICAIuHVibGljUGFyYW1ldGVycyI6IHsKICAgICAgICAIuHVibGljS2V5RUNESCi6IHsKICAgICAgICAgICJjcnYiOiAiWDQ0OClSciAgICAgICAgICAIuHVibGljIjogIm1DQy00eHVLVHFBSGhudF9pZ2pERjBPemg3RFRo1dUNVcjE4Z0NtZ0hCS0dqNDVaTEh0dkKICBKaEJCvG1LQmt6b3drNV93dEo3MU4wcUEifX19LAogICAgIk1hc2VDbXR0ZW50awNhdGlvbiI6IHsKICAgICAgICIlVkJzIi6ICJNQ05LLTMzSE4tREVJUC1aTVQ0LTJBWDctQ1dUU101TVk0IiwKICAgICAgICIlB1Ym1YmxpY1BhcmFtZXRLcnMiOiB7CiAgICAgICAgICIlB1Ym1YmxpY0tleUVDREgiOiB7CiAgICAgICAgICAIY3J2IjogIlg0NDgiLAogICAgICAgICAgICIlB1Ym1YmxpYyI6ICJBUHVuM01nN0xoMEpia3hkNC1teWNKY3NfZWpMQ2F0YXkyR3NpSFJGS09uRnMxd1dXNmZzCiAgUm91b3ctcks4SjVkdmlDZjFnYlFUQUlBIn19fSwKICAgICJCYXNlU2lnbmF0dXJlIjogewogICAgICAIuVWRmIjogIk1ENkQtMlJRUI1SSU1WLUw3RzMtNDZQSy03Q10eLTdCWE0iLAogICAgICAIuHVibGljUGFyYW1ldGVycyI6IHsKICAgICAgICAIuHVibGljS2V5RUNESCi6IHsKICAgICAgICAgICAgICJjcnYiOiAiRWQ0NDgiLAogICAgICAgICAgICIlB1Ym1YmxpYyI6ICJCSHRuN3d3Wk1xZGF1VFc2M0M0blJwDdhKSHV5d2Z5aFdzd0sxYUp1WTJMWEEzbnpxdFByCiAgMFkwWEpzbzNhTXFqajVJRhdQVG1CZzJBIn19fX19",
        {
          "signatures": [
            {
              "alg": "S512",
              "kid": "MCNR-5XZX-CCZC-XTMJ-RYP2-PACV-HCBM",
              "signature": "NznNSRNVFwr5gUmYxqWmMRBZFBGM-AhdfrXgtEoPNoPNm8-5-rSl3b1RzCSycA6iXIoe0FkXibCAKGo301nVzV1Wy5qFu34pVXUH4oxuH3ac957_UgTtGjypbRa10xqpENlFWbfrMVCiluHa4Jyd1REA"}],
              "PayloadDigest": "jUV5RGz2PDyvbgrOndJlCBEP1YV53I5bpEJKKnyz-xC1BImrlcI3x7m0UCEFx_anC-tjsrl3CqbxKjCN8gPcQ"}],
              "ClientNonce": "quIpPcx1_Kgkww-XMAQeAQ",
              "PinId": "AAAU-6HVM-7AA6-TNEG-PFUS-IXZ3-BMYL",
              "PinWitness": "5zFBoTgLeXwSgJc68gRNB5zKuoJtWr1aP9fZYTUUpm26AGQDYgLAScoVVq60oNs-9YNQlCUduoa9yDtBVj-dKg",
              "AccountAddress": "alice@example.com"}]
    }
  }
}
```

The administration device can now use the PIN Identifier to retrieve the MessagePIN from the Local spool and use it to verify the request.

7.2. Completion Interaction

Completion messages are dummy messages that are added to a Mesh Spool to mark a change the status of messages previously posted. Any message that is in the inbound spool and has not been erased or redacted **MAY** be marked as read, unread or deleted. Any message in the outbound spool **MAY** be marked as sent, received or deleted.

Services **MAY** erase or redact messages in accordance with local site policy. Since messages are not removed from the spool on being marked deleted, they may be undeleted by marking them as read or unread. Marking a message deleted **MAY** make it more likely that the message will be removed if the sequence is subsequently purged.

After using the PIN code to authenticate connection of a device in the previous example, the corresponding MessagePin is marked as having been used by appending a completion message to the Local spool.

```
{
  "MessageComplete":{
    "MessageId":"NCAI-WEPR-GYOU-BSFM-5U3V-ASJN-RHPC",
    "References":[{
      "MessageId":"AAAU-6HVM-7AA6-TNEG-PFUS-IXZ3-BMYL",
      "ResponseId":"MADI-J7SC-HSPL-URE0-Z5NC-CWK6-KZIB",
      "Relationship":"Closed"}
    ]}}
}
```

The completion message is added to the spool in the same upload transaction that adds the device to the device catalog. This ensures that both operations occur or neither occurs.

7.3. Contact Exchange Interaction

The contact exchange interaction is used to support unilateral or mutual exchange of contact information. Contact exchange has three functions in the Mesh:

- *To exchange public key information to allow encryption of messages sent to and verification of signatures on messages sent from the contact subject.

*To exchange contact information allowing use of other communication protocols (e.g. telephone, SMS, xmpp, SMTP, OpenPGP, S/MIME, etc).

*To request that the recipient grant privileges to accept certain types of messages from the contact subject.

Registration of the subject's contact information in a registry service eliminates the need for the first of these functions but not the other two. To prevent abuse, every Mesh Message is subject to access control and a Mesh service will only accept a message from a sender if there is an entry in the Threshold Catalog of the account that expressly permits delivery of messages of the specified type that are authenticated by an authorized signature key.

The communication of unsolicited information afforded by the contact exchange interaction is deliberately limited so that a majority of users can accept contact exchange requests without prior authorization. It is however likely that some users will receive a considerable volume of requests forcing them to require contact requests be authorized through some form of third party accreditation.

7.3.1. Remote

The Remote Contact Exchange transaction consists of a sequence of MessageContact messages sent from the initiator to the responder, responder to the initiator, etc. While there is in principle no limit on the number of messages exchanged, most exchanges will be completed in three exchanges or less:

Initiator to Responder Contains Initiator contact data without authentication context from the exchange.

Responder to Initiator (optional) Contains Responder contact data authenticated under a PIN challenge presented in the previous message.

Initiator to Responder (optional) Contains Initiator contact data authenticated under a PIN challenge presented in the previous message.

Each message provides the recipient with additional information which **MAY** motivate the recipient to provide additional contact information to the sender.

[illegible]

```
{
  "signatures": [{
    "alg": "S512",
    "kid": "MDXT-5MND-ZJMG-NQSR-QYUK-NVHA-INAU",
    "signature": "hqs91kZaFLMQ9eEa5VeXc2Aj1wN5z3B9hklwgcH_
xBeBccw2r12m1NaoNFMh_5uKFCXxc8zd65KAu87UwIjagWF3G8aevnu_-ku28-90G
A8IwD9_MuuMZcgEsFz5F_YdwfXCH5L0SfaZEMdsLAKxUTwA"}
  ],
  "PayloadDigest": "Z4CKnUtWFCRlouUzqjT-IwCR6BBT98AhMuyoh1B3
_M7VKTU2jPaCskGsNwFv3rpY96G9U0306FtX5W0A6mguJA"}
],
"Reply": true,
"Subject": "alice@example.com",
"PIN": "ADHF-R2A7-SV2S-XUBQ-C7CU-HA6Y-JBVA"}}
```

The Mesh Contact Exchange transaction does not provide for validation of the contact information beyond the binding to the Mesh Account Address used to perform the exchange.

7.3.2. PIN

Contact exchange requests **MAY** be authenticated by a PIN code. Initial contact exchange requests **SHOULD** include a PIN code value that can be used to authenticate a response (if given). PIN codes **MAY** also be exchanged out of band.

A MessageContact authenticated by means of a PIN code is authenticated as described in the PIN Interaction section above.

7.3.3. EARL

A MessageContact message **MAY** be published as an EARL. This allows contact data to be presented to the recipient on a printed document such as a business card in machine readable format such as a QR code.

7.4. Group Invitation

The GroupInvitation interaction is used to invite a recipient to join a Mesh Group. The interaction is essentially a form of contact exchange except that a sender **SHOULD NOT** send group invitations unless there is an existing relationship. Thus the 'first trust' issues intrinsic to the contact exchange interaction do not apply.

The message specifies the group name and the contact entry for the group. The contact entry includes the CapabilityDecryptService used to decrypt messages sent to the group when combined with information provided by the threshold service for the group.

Receipt of a GroupInvitation message does not require a response.

```

{
  "GroupInvitation":{
    "MessageId":"NBKE-VRT7-MEZT-4IOV-6KOR-T3FO-ZRME",
    "Sender":"alice@example.com",
    "Recipient":"bob@example.com",
    "Contact":{
      "ContactPerson":{
        "Anchors":[{
          "Udf":"MBND-NF5R-BE0Z-EHDY-PXM3-XTDU-HK5T",
          "Validation":"Self"}
        ],
        "NetworkAddresses":[{
          "Address":"groupw@example.com",
          "EnvelopedProfileAccount":[{
            "EnvelopeId":"MBND-NF5R-BE0Z-EHDY-PXM3-XTDU-HK5T",
            "dig":"S512",
            "ContentMetaData":"ewogICJvbm1xdWVJZCI6ICJNQk5ELU
5GNVItQkVPWi1FSERZLVBYTTMTWFREVS1ISzVUIiWkICAgITWVzc2FnZVR5cGUiOiA
iUHJvZm1sZUdyb3VwIiwKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
iAogICJDcmVhdGVkIjogIjIwMjEtMDEtMTNUMTY6Mzg6MjhaIn0"},
            "ewogICJQcm9maWxlR3JvdXAiOiB7CiAgICAgICAgICAgICAgICAgICAgIC
5hdHVyZSI6IHSKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
tWFREVS1ISzVUIiWkICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIC
CAGICAgICJQdWJsawMiOiAiTkdxV01jYUI4a1Y5WjJQ0FVyd2lpYUFEdVhDUMhbbG
VSX0NDZFB1eURXZnBTTVBGM0IxZgogIG1faVd1NTdFbDR0RGR0Uk8wR3ZISjFPQSJ
9fX0sCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIC
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIC
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIC
2IjogIlg0NDgiLAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
bGdra2dmVWhCeXpibW5rWkw1cFNoNkFGd2t1UkNOTThMUVVnCiAgbEgwOEYtRi14U
WNDS3M1TndDZ0x6ejBBIn19fSwKICAgICAgICAgICAgICAgICAgICAgICAgICAgIC
ogewogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
KTVgiLAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
S2V5RUNESCI6IHSKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIC
1ueDF0aGpsNHU5R29rCiAgLWRPUHdmeXpOUTEtVnJSdUU2RG8wR01BIn19fX19",
          {
            "signatures":[{
              "alg":"S512",
              "kid":"MBND-NF5R-BE0Z-EHDY-PXM3-XTDU-HK5T",
              "signature":"pXHiPRR5N9QZieJEiaoxW8J6bn0p8arf
1LtY4U6LROXZpW9ru2eb-7vA4-qDm6d11gUED0Kyhp-AP-zPo1BeiRo7kFuUU6xcz
M5zCIjnDEQXJuXiLF1FMUfcL9V9l9bAeaPiy7VETgdEtzobLAaFQxIA"}
            ],
            "PayloadDigest":"SFn06ZCeXo0y3hSACwFeQmGB6ahSrVMD
sRDwERcy6lPnAtyPjtP6ht2dVv89ZxFOVBON1q-UKkhjZxTtrnsirg"}
          ],

```

```

    "Protocols":[{
      "Protocol":"mmm"}
    ],
    "Capabilities":[{
      "CapabilityDecryptPartial":{
        "Id":"MB44-I7TT-A3JE-N5LK-SKTY-WJD6-URGZ",
        "KeyData":{
          "Udf":"NC6A-BU3A-VQE7-KXXQ-F352-E5AG-KZW4",
          "PrivateParameters":{
            "PrivateKeyECDH":{
              "crv":"X448z",
              "Private":"_dcwVDGmDmeEIMuBSyGM_a-BbQZiWi
QrmvcktCKDn6lXkkat6XV37x59EeEqTaBnqj2zmLWb7ws"}}},
          "SubjectId":"MB44-I7TT-A3JE-N5LK-SKTY-WJD6-URGZ",
          "ServiceId":"NC6A-BU3A-VQE7-KXXQ-F352-E5AG-KZW4",
          "ServiceAddress":"groupw@example.com"}}
      ]}
    ]}}}}

```

7.5. Confirmation Interaction

The confirmation interaction consists of a RequestConfirmation message from the initiator followed by a ResponseConfirmation from the responder.

The RequestConfirmation message specifies the action that is requested.

The ResponseConfirmation message contains the enveloped RequestConfirmation message signed by the initiator and the disposition of the responder, Accept = true if the request is accepted and Accept = false otherwise.

The service sends out the following request:

```

{
  "RequestConfirmation":{
    "MessageId":"NCHB-PAFY-23JU-UWSK-Q4NK-LEKY-MFZX",
    "Sender":"console@example.com",
    "Recipient":"alice@example.com",
    "Text":"start"}}

```

Alice accepts the request and returns the following response:

```

{
  "ResponseConfirmation":{
    "MessageId":"MBX5-PSU2-MPUA-IRVR-WIJL-CS4Q-5YUS",
    "Sender":"alice@example.com",
    "Recipient":"console@example.com",
    "Request":[{
      "EnvelopeId":"MCDL-YDFD-GEC4-MJZO-FONM-AC2D-HNM7",
      "ContentMetaData":"ewogICJYbmlxdWVJZCI6ICJOQ0hCLVBBRlktMj
NKVS1VV1NLLVE0TkstTEVLWS1NRlpYIiwKICAiTWVzc2FnZVR5cGUiOiAiUmVxdWV
zdENvbmZpcm1hdGlvbiIsCiAgImN0eSI6ICJhcHBsaWNhdGlvbi9tbW0vb2JqZWNO
IiwKICAiQ3JlYXRlZCI6ICIyMDIxLTAxLTEzVDE2OjM4OjI3WiJ9",
      "SequenceInfo":{
        "Index":4,
        "TreePosition":6222},
      "Received":"2021-01-13T16:38:27Z"},
      "ewogICJSZXF1ZXN0Q29uZmlybWFW0aW9uIjogewogICAgIk1lc3NhZ2VJZC
I6ICJOQ0hCLVBBRlktMjNKVS1VV1NLLVE0TkstTEVLWS1NRlpYIiwKICAgICJTZW5
kZXII0iAiY29uc29sZUBleGFtcGxlLmNvbSIsCiAgICAiUmVjaXBpZW50IjogImFs
aWNlQGV4YW1wbGUuY29tIiwKICAgICJUZSh0IjogInN0YXJ0In19",
      {}
    ],
    "Accept":true}}

```

8. Device Connection Interactions

Connection of a device to a Mesh Account combines synchronous and asynchronous elements and therefore uses a combination of Mesh Service Protocol and Mesh Messaging interactions.

Three connection interactions are currently defined support connection of devices with difference affordances:

Witness Authenticated For connecting devices that provide data entry and display affordances and are connected to a network. The account the device is to be connected to is entered into the device which displays a witness code. This code is then compared with a code displayed on the administration device to authenticate the request, after which both devices can complete the interaction.

Dynamic QR Code (PIN) Authenticated For connecting devices that provide a camera affordance. The user sets the administration device into 'add device' mode, causing a QR code to be displayed.

The QR code is scanned by the device being connected after which both devices can complete the interaction.

Static QR Code Authenticated For connecting devices that have been preconfigured with a device profile identified by means of a QR Code containing an EARL. The QR code is scanned by the administration device after which both devices can complete the interaction.

Each of these interactions provide strong mutual authentication with minimal user effort.

The witness authenticated connection interaction is intended for use in cases in which the device is already connected to a network. The QR code interactions are intended to provide support for acquisition of networking capabilities as part of the connection process. These functions are not currently specified. The Static QR Code Authenticated interaction is intended to support Internet of Things (IoT) devices which provide minimal interaction affordances.

In each case, the objectives of the device connection interaction are the same:

- *Mutually authenticate the onboarding device and the Mesh such that the connection interaction only completes if both sides acquire the authentic profile of the other.
- *To provision the onboarding device with the Mesh ProfileAccount, and an ActivationDevice and ConnectionDevice record allowing the device to interact as a member of the Mesh with the set of rights specified by the user.
- *To create a CataloguedDevice record and append it to the Device catalog of the account to allow the device to be managed within that account.
- *(optional) to acquire networking capabilities to allow the above to be completed.

The connection of the device to the Mesh Account is achieved through the creation of the ActivationDevice, ConnectionDevice and CataloguedDevice records described earlier. These are created by the administration device in the third phase of each of the connection interactions described below and acquired by the onboarding device in the fourth phase.

8.1. Dynamic QR Code (PIN) Authenticated

The Dynamic QR Code (PIN) Authenticated interaction comprises four phases as follows:

Phase 1: Issue of PIN credential

A PIN code is created and registered with the PIN Registration interaction described earlier and transmitted to the user by an out of band communication (e.g. a QR code that is scanned by the device).

Phase 2: Onboarding Device Request to Service The onboarding device performs a Connect transaction on a host servicing the account. This results in the exchange of the account and device profiles and the computation of a witness value from the two profile fingerprints and two nonce values specified by the onboarding device and the service.

An AcknowledgeConnection message is posted to the Inbound spool of the account.

Phase 3: Administration Device Acceptance The account holder accepts or rejects the connection request from a device that has been granted administration capability.

Phase 4: Onboarding Device Completion The onboarding device periodically polls the service for acceptance of the request by the administration device using the Complete transaction.

The use of the PIN code to authenticate the request message is shown in \$\$\$\$.

The PIN code **MAY** be presented to the onboarding device in any format accepted by the device. Administration **MAY** support presentation of the account address PIN code as a URI code. Administration devices **SHOULD** support presentation of the account address PIN code as a QR code containing the corresponding URI.

8.1.1. Phase 1:

```
Alice> account pin
PIN=AB23-ZB0I-CEIZ-MTD4-VQ
(Expires=2021-01-14T16:38:30Z)
```

The registration of this PIN value was shown earlier in section \$\$\$

The URI containing the account address and PIN is:

```
mcu://alice@example.com/AB23-ZB0I-CEIZ-MTD4-VQ
```

8.1.2. Phase 2:

The onboarding device scans the QR code to obtain the account address and PIN code. The PIN code is used to authenticate a connection request:

```
Alice3> device request alice@example.com /pin AB23-ZBOI-CEIZ-MTD4-VQ
Device UDF = MCNR-5XZX-CCZC-XTMJ-RYP2-PACV-HCBM
Witness value = B4AE-RNXK-Y20C-ZS7R-ZQZK-VUBI-ZFPF
```

The device issues a connection request to the service corresponding to the account. This specifies the device profile, the account to which the device is to be connected and the client nonce value:

```
{
  "ConnectRequest": {
    "EnvelopedRequestConnection": [ {
      "EnvelopeId": "MAD4-DQE5-PU4Y-QCTJ-D5U2-ZZVS-WBYC",
      "ContentMetaData": "ewogICJVbmIxdWVJZCI6ICJOQlBTLVRFMkstNUJRWi0zSFdBLVBKTLUtV1lPWC1EQU1XIiwKICAiTWvc2FnZVR5cGUoI0AiUmVxdWVzdENvbm5lY3Rpb24iLAogICJjdHkiOiAiYXBwbGljYXRpb24vbW1tLT29iamVjdCIsCiAgIkNyZWFOZWQiOiAiMjAyMS0wMS0xM1QXNj0zODozMFoifQ"}],
      "ewogICJSZXF1ZXN0Q29ubmVjdGlvbiI6IHskICAgICJNZXNzYWdlSWQiOiAiTkJKUyY1URTJLLTVCUVotM0hXS1QSks5VLvdZT1gtREFNVyIsCiAgICAIqXV0aGVudGllYXRlZERhdGEiOiBBewogICAgICAgICJFbnZlbG9wZUlkJjogIk1DTlItNVhaWC1DQ1pDLVhUTUotU1lQMi1QQUNWLUhDQk0iLAogICAgICAgICJkaWciOiAiAiuUXMiIsCiAgICAgICAgIkNvb3RlbnRNZXRhRGF0YSI6ICJld29nSUNKVmJtbHhkV1ZKkwNjNkLDSk5RMDVTFRWVdsZ3RRME5hUXkxCiAgWVZFMTVMkpaVURJdFVFRkRWaTFJUTBKTKlpd0tJQ0FPvFdwemMyRm5aV1I1Y0dVaU9pQWLVSEp2Wm1sc1oKICBVUmxbWxqWlNJc0NpQWdJbU4wZVNJNkLDSmhjSEJzYVdOAGRhbHZiaTl0YlclwdmIySnFaV04wSwl3S0lDQogIGlRM0pswVhSbFpDSTZJQ0l5TURJeExUQXhmVEV6VkRFmk9qTTRPak13V2lkOSJ9LAogICAgICAIzXdvZ0lDSlFjbTltYVd4bFJHVjJhV05sSWpvZ2V3b2dJQ0FnSwxCeWIyWgogIHBiR1ZUYVdkdVlYUjFjbVVPt21CN0NpQWdJQ0FnSUNKVlphHWw1PaUFpVFVOT1VpMDFXRnBZTFVORFdrTXRXCIAGRlJOU2kxU1dwQXlMVkJKU0TFZdFNFTkNUU0lzQ2lBZ0lDQWdJQ0pRZFdkc2FXTlFZWEPoYldWMFPySnpJam8KICBnZXdvZ0lDQWdJQ0FnSUNKUWRXSnnNhV05MWlhSRlEWUklJam9nZXdvZ0lDQWdJQ0FnSUNBZ0ltTnlkaUk2SQogIENKRlpEUTBPQ0lzQ2lBZ0lDQWdJQ0FnSUNBaVVIVmlIR2xqSWpvZ0luUnBkv2h6VTNZM2RuUmFhR1pCU0dSCiAgTlVqbFV0bmNTXpnm2JHOUNARjlsTUZocGVsOXhhRE5OUUVsd1oyUXlwVzlgZHPrS0lDQnFZekZ1YVhKaFKKICAzTkVSVUpqUZB4WVJHSnBPWFZ4ZVVFawZYMTlMQW9nSUNBZ0lrSmhjMlZGYm10eWVYQjBhVzllSwpvZ2V3bwogIGdJQ0FnSUNBaVZXUm1Jam9nSwsxRVdVXXRMDlDVXkxS1VEVkrMVEkzVUVZdFNRXPwUzFETms1V0xVZEtcIAgRU1pTEFvZ0lDQWdJQ0FPvUhwWaWHbGpVR0Z5WVcxGRHVnljeUk2SUhzS0lDQWdJQ0FnSUNBaVVIVmlIR2wKICBqUZJWNVJVtkVTQ0k2SUhzS0lDQWdJQ0FnSUNBZ0lDSmpjbllpT2lBaVdEUTBPQ0lzQ2lBZ0lDQWdJQ0FnSQogIENBaVVIVmlIR2xqSWpvZ0ltMURREtAwZUhwTFZIRkJTR2h1ZEY5cFoycEVSAkjqZW1nM1JGbZfKVU5WY2pFCiAgNFowTnRaMGhDUzBkcU5EVmFURWhvZDBrS0lDQkthRUUpDVkcxFfftdDZim2RyTly5M2RFbzNNVTR3Y1VFawYKICYMTlMQW9nSUNBZ0lrSmhjMlZCFhShb1pXNTBhV05oZEdsdmJPSTZJSHNLSUNBZ0lDQWdJbFZRwm1JNkLDSgogIE5RMDVMTFRNe1NFNHRSRVZKVUMxYVRWUTBMVEPCV0RjdFEXZFVVAtaXVFZrMElpd0tJQ0FnSUNBZ0lsQjFZCiAgbXhwTFCaGNtRnRawFJsY25NaU9pQjdDaUFnSUNBZ0lDQWdJbEIXww14CFkdGxlVVZEUKVnaU9pQjdDaUEKICBnSUNBZ0lDQWdJQ0FPWTNMkklqb2dJbGcwTkrNaUXBb2dJQ0FnSUNBZ0lDQWdJbEIXww14CF15STZJQ0pCVQogIEhWdu0wMMW5OMHhvTUVwaWEzaGt0Qzf0ZVD0S1kzTmZaV3BNUTJGMFlYa3lSM05wU0ZKR1MwOXVSBk14ZDFkCiAgWE5twnpDaUFnVW05SWIZY3Rja3M0U2pwa2RtbERAakZuwWxGVVFvBEJJbjE5ZlN3S0lDQWdJQ0pDWvhObFUKICAYbg5ibUYwZFhkbElqb2dlld29nSUNBZ0lDQWlWV1JtSwpvZ0lrMUVOa1F0TWxKU1VpMNVTVTFXFTFV3M1J6TQogIHRORFprRU3kwM1ExbzBMVGRDVkuUwaUXBb2dJQ0FnSUNBaVVIVmlIR2xqVUDGeVlXMWxkrR1Z5Y3lJNklic0tJCiAgQ0FnSUNBZ0lDQWlVSFZPykdsalMyvjVSU5FU0NjNklic0tJQ0FnSUNBZ0lDQWdJQ0ppqY25Zau9pQWlSV1EKICAwTkRnaUXBb2dJQ0FnSUNBZ0lDQWdJbEIXww14CF15STZJQ0pDU0hSdu4zZDNXazF4WkdGbFZGYzJNME0wYgogIGxKV2REaEtTSFY1ZDJanWFGZHPkMHn4WVvwMVdUSk1XRUV6Ym5weGRGqlldaUFnTUZrd1dFcHpiek5oVfhGCiAgCWfQvkpSSGRRVKcxQ1p6SkJJbjE5Zlqx0SiScIAgICAgIHsKICAgICAgICAcic2lnbmF0dXJlcYI6IFT7CiAgIC
```

133

The service generates the service nonce value and uses it to create the AcknowledgeConnection message.

[illegible]


```
{
  "ConnectResponse":{
    "Status":201,
    "StatusDescription":"Operation completed successfully",
    "EnvelopedAcknowledgeConnection":[{
      "EnvelopeId":"MC3V-VXL7-NE6V-7YMB-FEOM-SJWU-SFQ0",
      "ContentMetaData":"ewogICJYbm1xdWVJZCI6ICJCNEFFLVJOWEstWT
JPQy1aUzdSLVpRWkstVlVCSS1aRlBGiIiwKICAiTWVzc2FnZVR5cGUiOiAiQWNRbm9
3bGVkZ2VDb25uZWNoaW9uIiwKICAiY3R5IjogImFwcGxpY2F0aW9uL21tbS9vYmpl
Y3QiLAogICJDCmVhdGVkIjogIjIwMjEtMDEtMTNUMTY6Mzg6MzBaIn0",
      "SequenceInfo":{
        "Index":6,
        "TreePosition":11843},
      "Received":"2021-01-13T16:38:30Z"},
    "ewogICJBY2tub3dsZWRnZUNvbm5lY3Rpb24iOiB7CiAgICAiTWVzc2FnZU
lkIjogIkI0QUUtUk5YSy1ZMk9DLVpTN1ItwlFaSy1WVUJJLVpGUEYiLAogICAgIkV
udmVsb3BlZFJlcXVlc3RDb25uZWNoaW9uIjogW3sKICAgICAgICAiRW52ZWxvcGVJ
ZCI6ICJNQUQ0LURRRtUtUFU0WS1RQ1RKLQU1VTItwlplWUy1XQ1lDIiwKICAgICAgI
CAiQ29udGVudE1ldGFZEYXRhIjogImV3b2dJQ0pWYm1seGRXVkpQ0k2SUNKT1FsQl
RMVlJGTwtzdE5VS1JXaTAKICB6U0ZkQkxwQktUbFV0VjFsUFDdMUVRVTFYSWl3S0l
DQWlUV1Z6YzJGblpWUjVjR1VpT2lBaVvtVnhkV1Z6ZAogIEV0dmJtNwZM1JWYjI0
aUxBb2dJQ0pqZEhraU9pQWlZWEl3YkdsallYUnBiMjR2YlxdEwyOWlhbVZqZENJC
iAgc0NpQWdJa055WldGMFpXUWlPaUFpTWpBeU1TMhdNUzB4TTFReE5qb3pPRG96TU
ZvawZRIIn0sCiAgICAgICJld29nSUNKU1pYRjFawe4wUTI5dWJtVmpkR2x2Ym1JNk1
Ic0tJQ0FnSUNKCiAgTlpYtNpZV2RsU1dRaU9pQWlUa0pRVXkxVVJUSkxMVfZDVVZv
dE0waFhRUzFRU2s1VkwZfPUMWd0UkVGTlYKICB5SXNDaUFnSUNBaVFYVjBhR1Z1Z
EdsallYUmxarVJoZEdFaU9pQmJld29nSUNBZ0lDQWdJQ0pGYm5abGJHOqogIHdaVW
xrSwpvZ0lrMURUbEl0TlZoYVdDMURRMXBETfZovVRVb3RVbGxRTWkxUVFVtldmVWh
EUWswaUxBb2dJCiAgQ0FnSUNBZ0lDSmthV2NpT2lBaVV6VXhNaUlzQ2lBZ0lDQWdJ
Q0FnSwT0dmJuUmxiBlJOWlhSaFJHRjBZU0kKICA2SUNKbGQyOW5TVU5LVm1KdGJlA
GtWMVpLV2t0Sk5rbERTazVSTURWVFRGUldXVmRzWjNSuk1FNWhVWgt4QwogIGlBZ1
dWwkZNVXRNVmtwYVZVUkpKlZGUMtSV2FURkpVVEJLVGtscGQWdEpRMEZwVkZkV2V
tTXlSbTVhVmxCiAgMVkwZfZhtVtWUvdsVlNfcDJXbTFfZyZfVs0lDQ1ZVbXhrYld4
cVdsTkpjME5wUVdkSmJVNHdaVh5KTmtSRFMKICBtaGpTRUp6WVZkT2FHUKhiSFppY
VRsMf1sY3dkbU15U25GYVYwNHdTV2wzUzBsRFFRb2dJR2xSTTBwc1dWaogIFNiRn
BEU1RaSlEwbDVUVVJKZUV4VVFYaE1WRVY2VmtSRk1rOXFUVfJQYwsxM1YybEtPU0o
5TEFvZ0lDQWdJCiAgQ0FpwlhkdlowbERTbEZqYlRsdFlwZDRiRkpIVmpKaFYwNXNT
V3B2WjJWM2IyZEprMEZuU1d4Q2VXSXlXZ28KICBnSuhCaVixWlVZVmRrZfZswVvqR
mpiVlZwVDJsQ04wTnBRV2RKUTBGblNVTKtWbHBIV1dsUGFVRnBWRlZPVAogIDFwcE
1ERlhSbkJaVEZWT1JGZHJUWfJYQ2lBZ1JsSk9VMmt4VTFkv1FYbE1Wa0pDVVRGwMR
GtKZUa05VVTBsCiAge1EyBEJaMGxEUVdkSlEwcFJaRmRLYzJGWFRsRlpXRXBvWwXk
V01GcFlTbnBKYW04S0lDQm5aWGR2WjBsRFEKICBxZEprMEZuU1VOS1VXUlhTbk5oV
jA1TVdsahNSbEV3VWtsSmFtOW5aWGR2WjBsRFFXZEprMEZuU1V0QlowbAogIHRUbm
xrYVVRm1NRb2dJRu5LUmxwRVVUQ1BRMGx6UTJsQlowbERRV2RKUTBGblNVTKJhVlZ
JVm1saVIyeHFTCiAgV3B2WjBsdVVuQmtwMmg2VlROWk0yUnVVbUZoUjFwQ1UwZfND
aUFnVGxwcWJGVk9ibU41VFhwbk0ySkhPVU4KICBhUmpsc1RVWm9jR1ZzT1hoaFJfN
U9VVLZzZDFveVYbFdWemxHWkhWU1MwbERRbkZaZWtaMVlwaEThRmtLSQogIENBel
RrVlNWVBxBxVXpCNfDwSkhTbkJQV0ZaNFpWVkhZv1pZTVRsTVFXOW5TVU5CWjBsc1N
taGpNbFpHWW0xCiAgT2VXVllRakJoVnpsMVNXcHZaM1YzYndvZ0lHZEprMEZuU1V0
```

QmFWw1hVbTFKYW05b1NXc3hSVmRWV1hSUK0KICBEbERWWGt4UzFWRVZrUk1WRWt6V
lVWwmRGtNJSWHBXVXpGRVRtczFWMHhWwKv0VENpQWdSVTFwVEVGd1owbAogIERRV2
RKUTBGcFZVaFdhv0pIYkdWVlIwWjVXVMn4YkdSSFZubGp1VWsyU1Voe1MwbERRV2R
KUTBGblNVTKJhCiAgVlZJVm1saVIyd0tJQ0JxVXpKV05WSlZUa1ZUUTBrM1NvaHpT
MGxEUVdkSlEwRm5TVU5CWjBsRFntcGpibGwKICBwVDJsQmFWZEvvVEJQUTBse1EyB
EJaMGxEUVdkSlEwRm5TUW9nSUV0QmFWVklWbWxpUjJ4cVNXCZaMGx0TQogIFVSum
VUQXdaVWhXVEZaSVJrSlRSMmgxwkVZNWNGb3ljRVZTYwtKUVpXMM5NMUpHYnpGa1Z
VNVdZMnBGQ2lBCiAgZ05Gb3dUblJhTUdoRFV6QmtjVTVFVm1GVVJXaHZaREJyUzBs
RFFrdGhSVXBEVmtjeFRGRnRkRFppTTJSeVQKICBSwTVNMlJGYnpOTlZUujNZMVZGY
VdZS0lDQl1NVGxNUVc5b1NVTKJaMGxyU21oak1sWkNaRmhTYjFwWE5UQgogIGhWMD
VvWkVkc2RtSnBTVFpKU0h0TFNVTKJaMGxEUVdkSmJGwnJXbWxKTmtsRFNnb2dJRTV
STURWTVRGUK5lCiAgbE5GTkhSU1JWwktWVU14WVZSV1VUQk1WRXBDVjBSamRGRXha
RlZWYVRBeFZGwnJNRWxwZDB0SlEwRm5TVU4KICBCWjBsc1FqRlpDaUFnYlhod1dUR
kNhR050Um5SYVdGsnNZMjv0YVU5cFFqZERhVUZuU1V0QlowbERRV2RKYgogIEVJeF
dXMTRjRmt3ZEd4bFZWwKvVa1ZuYVU5cFFqZERhVUULSUNCb1NVTKJaMGxEUVdkSlE
wRnBXVE5LTwtScIAGcWIyZEpIR2N3VGtSbmFVeEJiMmRKUTBGblNVTKJaMGxEUVdk
SmJFSXhXVzE0Y0ZsNVNUWkpRMHBDVlFvZ0kKICBFaFdkVTB3TVc1T01IaHZUVVZ3Y
VdFemFHD9RekYwWlZkT1Mxa3pUbVphVjNCTlVUSkdNRmxZYTnsU00wNQogIHdVMF
pLUjFNd09YVlNiazE0WkRGa0NpQWdXRTV0V25wRGFVRm5WVzA1U1dJe1kzUmpM00
wVTJwV2EyUnRiCiAgRVJhYwtadVdXeEdWVkJWYkVKSmJqRTVabE4zUzBsRFFXZEpR
MHBEV1ZoT2JGVUtJQ0F5Ykc1aWJVWXdarMgKICBLyKvscWIyZGxkMjluU1V0Qlowb
ERRV2xXVjFKdFNXChZaMGxyTVVWT2ExRjBUV3hLVWxWcE1WTlRWVEZYVAogIEZWM0
0xSjZUUW9nSUHST1JGcFJVM2t3TTFfEGJ6Qk1WR1JEVmtVd2FVeEJiMmRKUTBGblN
VTkjhVlZJVm1ScIAGaVIyeHFVWRHZVZsWE1XEGtSMVo1WTNsSk5rbEljMHRKQ2lB
Z1EwRm5TVU5CWjBsRFFXbFZTRlpwWtKc2EKICBSXlWalZTVlU1RlUwTkP0a2xJY
zB0SlEwRm5TVU5CWjBsRFFXZEpRMHBxWTI1WmFVOXBRV2xTVjFFS0lDQogIHdUa1
JuYVv4QmIyZEpRMEZuU1V0QlowbERRV2RKYkVJeFdXMTRjRmw1U1RaSlEwcERVMGh
TZFU0elpETlhhCiAgekY0V2tkR2JGwkdZekp0TUUwd1lnb2dJR3hLVjJSRWFfDfRT
RlKxWkRKYU5XRkdaSHBrTUh0NFdWvndNVmQKICBVU2sxWFJVVjZZbTV3ZUdSR1Fsb
ERhVUZuVfVacmQxZEZjSHBpZws1b1ZGaEdDaUFnY1dGcVZrcFNTR1JSVgogIGtjeF
ExcDZTa0pKYmpFNVpsZ3hPU0lZQ2lBZ0lDQWdJSHNLSUNBZ0lDQWdJQ0FpYzJsbmJ
tRjBkWepsY3lJCiAgNklGdDdDaUFnSUNBZ0lDQWdJQ0FnSUNKaGJHY2lPaUFpVXpV
eE1pSXNDaUFnSUNBZ0lDQWdJQ0FnSUNKcmEKICBXUWlPaUFpVFVOT1VpMDFXRnBZT
FVORFdrTXRXRlJOU2kxU1dWQXlMVkJCUTFZdFNFTkNUU0lZQ2lBZ0lDQogIGdJQ0
FnSUNBZ0lDSnphV2R1WvhSMWntVWlPaUFpVG5wdVRStlNUbFpHVjNjMVoxVnRXWgh
4VjAxdFvRSmFSCiAgA0pIVFMxQmFHUm1jbGhuZEVwdlVFNXZVRTV0T0MwMuxRb2dJ
SEpUYkR0aU1WSjZRMU41WTBFmFwaEpiMlYKICBQum10WwFXSkRRVXRIYnp0UE1XN
VdlbFl4VjNrmWNVWjFNe1J3VmxoVlNEUnZlSFZJTTJGak9UVTNDaUFnWAogIDFwbl
ZIUKhhbmX3WwXKaE1UQjRjVkJGYm14R1YySm1jazFXUTJsc2RvAGhORXA1WkRGU1J
VRWlMvJbZQ2lBCiAgZ0lDQWdJQ0FnSwxCaGVXeHZZV1JFYVdkbGMzUWlPaUFpYwXW
V05WSkh1akpRUKhsM1ltZHLUMjvRu214a1EKICBrVndVakZaVmpVe1NUVm1jRVZLU
zB0dWVyb3RlRU14Q2lBZ1FrBHRjbXhqU1R0NE4yMHdWVU5GUm5oZl1XNQogIERMWF
JxYzNkC00wTnhZbmhMYwt0T09HZFFZMUVpZ1Ywc0NpQWdJQ0FpUTJ4cFpXNTBUBt1
1WTJVaU9pQWlJciAgWFZKY0ZCamVHeGZTMmRyVjNjdFdFMUJVV1ZCVVNJc0NpQWdJ
Q0FpVUdsdVNxUWlPaUFpUVVGQlZTMDJTRloKICB0TFRkQlFUWXRWRTVGUnkxUVJsV
lRMVwxZV2pNdFFrMVpuQ0lZQ2lBZ0lDQWlVR2x1VjJsMGJtVnpjeUk2SQogIENJMW
VrWkNiMVJiYkdWwWQxTm5TbU0yT0dkU1RrSTFla3QxYjBwMFYzSnNZVKE1WmxwWlZ
GbFZjRzB5TmtGCiAgSFVVUUtJQ0JaWjB4QlUyTnZWbFp4Tms5d1RuTXRPVmxPVVd4
RFZXUjFiMkU1ZVVSFMFlsWnFMV1JMwnlJc0MKICBpQWdJQ0FpUVdOamIzVnVkRUZrW

```
khKbGMzTWlPaUFpWd4cFkyVkFawGhoYlhCc1pTNWpiMjBpZlgwIl0sCiAgICAIu2
VydmVyTm9uY2U0iAiTFaZV0RsUWRDdEJyTlVGyVRhVXZYQSIscCiAgICAIv2l0bmV
zcyI6ICJCNEFFLVJOWEstWTJPQy1aUzdSLVpRWkstVlVCSS1aRlBGIn19",
  {}
],
"EnvelopedProfileAccount": [{
  "EnvelopeId": "MAMU-5QXP-TWCD-7PKI-S4FC-IB76-XASH",
  "dig": "S512",
  "ContentMetaData": "ewogICJvbm1xdWVJZCI6ICJNQU1VLTVRWFatVF
dDRC03UETJLVM0RkMtSUI3Ni1YQVNIIiwKICAIvWzC2FnZVR5cGUiOiAiUHJvZm1
sZVVzZXIiLAogICJjdHkiOiAiYXBwbGljYXRpb24vbW1tL29iamVjdCIsCiAgIkNy
ZWFOZWQiOiAiMjAyMS0wMS0xM1QxNjozODox0VoifQ"},
  "ewogICJQcm9maWxlVXNlciI6IHsKICAgICJQcm9maWxlU2lnbmF0dXJlIj
ogewogICAgICAIvWRmIjogIk1BTUVUetNVFYUC1UV0NELTdQS0ktUzRGQy1JQjc2LVh
BU0giLAogICAgICAIUHvibGljUGFyYW1ldGVycyI6IHsKICAgICAgICAIUHvibGlj
S2V5RUNESCi6IHsKICAgICAgICAgICJjcnYiOiAiRWQ0NDgiLAogICAgICAgICAgI
lB1YmXpYyI6ICJjM2w5NmhtL2ZiWHpRYTdKb2hncF9YOVlJSXphUjRVMGRQQ2Z5b2
NxdUZXblp1aUzkdTl2CiAgbDlVSwd0WXYtdGpGVnBtazZxUkRqN21BIn19fSwKICA
gICJBY2NvdW50QWRkcmVzcyI6ICJhbGljZUBleGFtcGxlLmNvbSIsCiAgICAIu2Vy
dmljZVVkZiI6ICJNQTM2LVRVSkwtUVJaSi0zTTNMLVNSQlEtQlJZUS1XMl1NIiwKI
CAGICJBY2NvdW50RW5jcnlwdGlvbiI6IHsKICAgICAgICAgIClVkiI6ICJNRExPLUpKNE
ItUkJZNS1wWUQ3LUxKWlktUzNSSy1EQk0yIiwKICAgICAgICAgIClB1YmXpY1BhcmFtZXR
lcnMiOiB7CiAgICAgICAgICAgIClB1YmXpY0tleUVDREgiOiB7CiAgICAgICAgICAIY3J2
IjogIlg0NDgiLAogICAgICAgICAgIClB1YmXpYyI6ICl0TWk5WDhtWEk5bUdoXzda
kdaUDbHrLBSWEpOU0ZleFBCbk1BSzFCTl9fU1hSeHRXUVRzCiAgWHNnejFmbDVKYz
M4Wll4N01WZTJYOXdBIn19fSwKICAgICJBZG1pbmlzdHJhdG9yU2lnbmF0dXJlIjo
gewogICAgICAIvWRmIjogIk1DQ0stRjJXWi1RQUFDLUMzTkEtRVZBVy1TQkw3LUlI
RVEiLAogICAgICAIUHvibGljUGFyYW1ldGVycyI6IHsKICAgICAgICAIUHvibGljS
2V5RUNESCi6IHsKICAgICAgICAgICJjcnYiOiAiRWQ0NDgiLAogICAgICAgICAgI
lB1YmXpYyI6ICl0TWk5WDhtWEk5bUdoXzdaZkd0bWVudFNPZ25hdHVyZSI6IHsKIC
AgICAgIClVkiI6ICJNRExPLUpKNEItUkJZNS1wWUQ3LUxKWlktUzNSSy1EQk0yIiwKIC
AgICAgICAgICAgIClB1YmXpY1BhcmFtZXRlcnMiOiB7CiAgICAgICAgICAgIClB1YmXpY0tleUVD
REgiOiB7CiAgICAgICAgICAIY3J2IjogIkVkdDQ4IiwKICAgICAgICAgICJQdWJsa
WMI0iAiQUhTdFc4S0dwYnJvWnQ1ZXotd3ZiQ19GTXI5QWpxSThnTGlnQzVWm3doSG
NFUTRqUDlkVwogIDF4RHdaMzRqNzdxWE5JZkZFdk9FV0pNQSJ9fX19fQ",
  {
    "signatures": [{
      "alg": "S512",
      "kid": "MAMU-5QXP-TWCD-7PKI-S4FC-IB76-XASH",
      "signature": "_9tIDk5KvjeIuasHaXDawBB1VTw2YIzxBUxpLn78
a0qf09Cjuwh7auyUMHrCGvpuQRjjQrDR_OeATnhDzrIG5xcQbFwvfge_rfqvUjqQc
-CZqvT81LDQ2c1W6THP1Z0GcIZmxNEpYVkyYR-9AACDQAcA"}
    ]
  }
]
```

```

    ],
    "PayloadDigest": "fb_iksIe0dM4IWIWZjmKlYQSF-XttjIAg8Bww4tJ
jpOE0P9bxX42pkNorLfHQ8XyD8x9IHT-FKh-_lhLJNAUzA"}
  ]}}

```

The device generates the witness value and presents it to the user as shown above.

8.1.3. Phase 3:

The user synchronizes their pending messages:

```

Alice> message pending
MessageID: B4AE-RNXK-Y20C-ZS7R-ZQZK-VUBI-ZFPF
  Connection Request::
  MessageID: B4AE-RNXK-Y20C-ZS7R-ZQZK-VUBI-ZFPF
  To: From:
  Device: MCNR-5XZX-CCZC-XTMJ-RYP2-PACV-HCBM
  Witness: B4AE-RNXK-Y20C-ZS7R-ZQZK-VUBI-ZFPF
MessageID: NAL6-QEBL-5JF7-TEX3-AIM7-HV74-MPYQ
  Group invitation::
  MessageID: NAL6-QEBL-5JF7-TEX3-AIM7-HV74-MPYQ
  To: alice@example.com From: alice@example.com
MessageID: NCHB-PAFY-23JU-UWSK-Q4NK-LEKY-MFZX
  Confirmation Request::
  MessageID: NCHB-PAFY-23JU-UWSK-Q4NK-LEKY-MFZX
  To: alice@example.com From: console@example.com
  Text: start
MessageID: NBPI-ULGE-MSPK-FUE2-CBDB-WTNY-NJDZ
  Contact Request::
  MessageID: NBPI-ULGE-MSPK-FUE2-CBDB-WTNY-NJDZ
  To: alice@example.com From: bob@example.com
  PIN: ADHF-R2A7-SV2S-XUBQ-C7CU-HA6Y-JBVA
Alice> account sync /auto
ERROR - An attempt was made to create an object with an existing object identifier

```

The administration device determines that the device connection request is authenticated by a PIN code. The PIN code is retrieved and the message authenticated. This is shown in the PIN registration interaction example in section \$\$\$ above.

The Cataloged device record is created from the public key values corresponding to the combination of the public keys in the device profile and those defined by the activation:

The ActivationDevice and CatalogedDevice records are ???

8.1.4. Phase 4

The device periodically polls for completion of the connection request using the Complete transaction.

To provide a final check on the process, the command line tool presents the UDF of the account profile to which the device has connected if successful:

```
Alice3> device complete
Device UDF = MCNR-5XZX-CCZC-XTMJ-RYP2-PACV-HCBM
Account = alice@example.com
Account UDF = MAMU-5QXP-TWCD-7PKI-S4FC-IB76-XASH
Alice3> account sync
```

The completion request specified the device requesting completion:

```
{
  "CompleteRequest":{
    "AccountAddress":"alice@example.com",
    "ResponseID":"MADI-J7SC-HSPL-URE0-Z5NC-CWK6-KZIB"}}
```

The response contains the information the device requires to complete the connection to the mesh:

```
{
  "CompleteRequest":{
    "AccountAddress":"alice@example.com",
    "ResponseID":"MADI-J7SC-HSPL-URE0-Z5NC-CWK6-KZIB"}}
```

8.2. Witness Authenticated

The witness authenticated interaction is essentially the same as the Dynamic QR Code (PIN) Authenticated interaction except that mutual authentication is achieved by comparing the witness value computed by the device in the second phase to that computed by the administration device in the third phase.

8.2.1. Phase 1:

There are no first phase actions.

8.2.2. Phase 2:

Alice enters the connection request on the device to be connected. This specifies the address of the account to which she wishes to connect:

```
Alice2> device request alice@example.com
Device UDF = MCDZ-IZFY-PWT3-36UP-4L6Q-5QMJ-YWWY
Witness value = WGAZ-BX2A-7G72-J5U3-SGGT-UB3W-TV4L
```

8.2.3. Phase 3:

The user reviews their pending messages:

```
Alice> device pending
MessageID: WGAZ-BX2A-7G72-J5U3-SGGT-UB3W-TV4L
Connection Request::
MessageID: WGAZ-BX2A-7G72-J5U3-SGGT-UB3W-TV4L
To: From:
Device: MCDZ-IZFY-PWT3-36UP-4L6Q-5QMJ-YWWY
Witness: WGAZ-BX2A-7G72-J5U3-SGGT-UB3W-TV4L
```

The administration device receives the AcknowledgeConnection message from the service and verifies that the signature is valid and the witness value correctly computed.

The user verifies that the witness value presented in the AcknowledgeConnection message matches the one presented on the device. Since they match, the request is accepted:

```
Alice> device accept WGAZ-BX2A-7G72-J5U3-SGGT-UB3W-TV4L /message /web
```

8.2.4. Phase 4

The device completes the connection as before:

```
Alice2> device complete
Device UDF = MCDZ-IZFY-PWT3-36UP-4L6Q-5QMJ-YWWY
Account = alice@example.com
Account UDF = MAMU-5QXP-TWCD-7PKI-S4FC-IB76-XASH
```

8.3. Static QR Code Authenticated

The Static QR Code Authenticated mode is used to connect devices that have been preconfigured for use with the Mesh with a device configuration identified by a URI typically presented as machine readable QR code on the device itself.

The interaction has four phases:

Phase 1: Preconfiguration The device to be onboarded is preconfigured with a ProfileDevice and private key information and a DeviceDescription posted to a publication service. This process is typically performed during manufacture. An EARL providing the ability to locate and decrypt the description is printed on the device itself as a QR code.

Phase 2: Device description acquisition The administration device acquiring the onboarding device scans the QR code on the device and uses this information to obtain the device description by means of the PollClaim interaction described above.

Phase 3: Administration Device Acceptance This phase is performed in the same manner as the Dynamic QR Code (PIN) Authenticated interaction except that the administration device **MAY** advise the device that a connection request is being made by additional means described in the device description (e.g. WiFi, Bluetooth).

Phase 4: Onboarding Device Completion This phase is performed in the same manner as the Dynamic QR Code (PIN) Authenticated interaction except that the administration device requires notice that of the pending connection request.

The use of the PollClaim interaction to discover a pending claim for a document was shown in section \$\$\$ above. The claim is authenticated using the key specified in the EARL.

8.3.1. Phase 1

The device is prepopulated with a Device description, this is shown in section \$\$\$ of Schema.

8.3.2. Phase 2

The administration device scans the QR code and obtains the Device Description using the Claim transaction as shown in section \$\$\$.

8.3.3. Phase 3

The administration device creates the ActivationDevice and CatalogedDevice records and populates the service as before.

8.3.4. Phase 4

The device polls the publication service until a claim message is returned. This interaction is shown in section \$\$\$\$ above.

9. Protocol Schema

HTTP Well Known Service Prefix: `/.well-known/mmm`

Every Mesh Portal

Service transaction consists of exactly one request followed by exactly one response. Mesh Service transactions MAY cause modification of the data stored in the Mesh Service or the Mesh itself but do not cause changes to the connection state. The protocol itself is thus idempotent. There is no set sequence in which operations are required to be performed. It is not necessary to perform a Hello transaction prior to any other transaction.

9.1. Request Messages

A Mesh Portal Service request consists of a payload object that inherits from the MeshRequest class. When using the HTTP binding, the request MUST specify the portal DNS address in the HTTP Host field.

9.1.1. Message: MeshRequest

Base class for all request messages.

[No fields]

9.1.2. Message: MeshRequestUser

Base class for all request messages made by a user.

Inherits: **MeshRequest** The fully qualified account name (including **Account: String (Optional)** DNS address) to which the request is directed.

EnvelopedProfileDevice: Enveloped (Optional) Device profile of the device making the request.

9.2. Response Messages

A Mesh Portal Service response consists of a payload object that inherits from the MeshResponse class. When using the HTTP binding, the response SHOULD report the Status response code in the HTTP response message. However the response code returned in the payload object MUST always be considered authoritative.

9.2.1. Message: MeshResponse

Base class for all response messages. Contains only the status code and status description fields.

[No fields]

9.3. Imported Objects

The Mesh Service protocol makes use of JSON objects defined in the JOSE Signature and Encryption specifications and in the DARE Data At Rest Encryption extensions to JOSE.

9.4. Common Structures

The following common structures are used in the protocol messages:

9.4.1. Structure: KeyValue

Describes a Key/Value structure used to make queries for records matching one or more selection criteria.

Key: String (Optional) The data retrieval key.

Value: String (Optional) The data value to match.

9.4.2. Structure: ConstraintsSelect

Specifies constraints to be applied to a search result. These allow a client to limit the number of records returned, the quantity of data returned, the earliest and latest data returned, etc.

Container: String (Optional) The container to be searched.

IndexMin: Integer (Optional) Only return objects with an index value that is equal to or higher than the value specified.

IndexMax: Integer (Optional) Only return objects with an index value that is equal to or lower than the value specified.

NotBefore: DateTime (Optional) Only data published on or after the specified time instant is requested.

Before: DateTime (Optional) Only data published before the specified time instant is requested. This excludes data published at the specified time instant.

PageKey: String (Optional) Specifies a page key returned in a previous search operation in which the number of responses exceeded the specified bounds.

When a page key is specified, all the other search parameters except for MaxEntries and MaxBytes are ignored and the service returns the next set of data responding to the earlier query.

9.4.3. Structure: ConstraintsData

Specifies constraints on the data to be sent.

MaxEntries: Integer (Optional) Maximum number of entries to send.

BytesOffset: Integer (Optional) Specifies an offset to be applied to the payload data before it is sent. This allows large payloads to be transferred incrementally.

BytesMax: Integer (Optional) Maximum number of payload bytes to send.

Header: Boolean (Optional) Return the entry header

Payload: Boolean (Optional) Return the entry payload

Trailer: Boolean (Optional) Return the entry trailer

9.4.4. Structure: PolicyAccount

Describes the account creation policy including constraints on account names, whether there is an open account creation policy, etc.

Minimum: Integer (Optional) Specifies the minimum length of an account name.

Maximum: Integer (Optional) Specifies the maximum length of an account name.

InvalidCharacters: String (Optional) A list of characters that the service does not accept in account names. The list of characters MAY not be exhaustive but SHOULD include any illegal characters in the proposed account name.

9.4.5. Structure: ContainerStatus

Container: String (Optional)

Index: Integer (Optional) 9.4.6. Structure: ContainerUpdate

Digest: Binary (Optional)

Inherits: ContainerStatus The entries to be uploaded.

Envelopes: DareEnvelope [0..Many]

9.5. Transaction: Hello

Request: `HelloRequest`

Response: `MeshHelloResponse`

Report service and version information.

The Hello transaction provides a means of determining which protocol versions, message encodings and transport protocols are supported by the service.

The `PostConstraints` field MAY be used to advise senders of a maximum size of payload that MAY be sent in an initial Post request.

9.5.1. Message: `MeshHelloResponse`

ConstraintsUpdate: `ConstraintsData (Optional)` Specifies the default data constraints for updates.

ConstraintsPost: `ConstraintsData (Optional)` Specifies the default data constraints for message senders.

PolicyAccount: `PolicyAccount (Optional)` Specifies the account creation policy

EnvelopedProfileService: `Enveloped (Optional)` The enveloped master profile of the service.

EnvelopedProfileHost: `Enveloped (Optional)` The enveloped profile of the host.

9.6. Transaction: `BindAccount`

Request: `BindRequest`

Response: `BindResponse` Request creation of a new service account or group.

Attempt

9.6.1. Message: `BindRequest`

Request binding of an account to a service address.

Inherits: `MeshRequest` The service account to bind to.

AccountAddress: `String (Optional)`

EnvelopedProfileAccount: `Enveloped (Optional)` The signed assertion describing the account.

9.6.2. Message: `BindResponse`

Inherits: `MeshResponse`

Reports the success or failure of a Create transaction.

Reason: String (Optional)

Text explaining the status of the creation request.

URL: String (Optional) A URL to which the user is directed to complete the account creation request.

9.7. Transaction: UnbindAccount

Request: UnbindRequest

Response: UnbindResponse Request deletion of a service account.

9.7.1. Message: UnbindRequest

Request creation of a new portal account. The request specifies the requested account identifier and the Mesh profile to be associated with the account.

Inherits: MeshRequestUser

[No fields]

9.7.2. Message: UnbindResponse

Inherits: MeshResponse

Reports the success or failure of a Delete transaction.

[No fields]

9.8. Transaction: Connect

Request: ConnectRequest

Response: ConnectResponse Request information necessary to begin making a connection request.

9.8.1. Message: ConnectRequest

Inherits: MeshRequest The connection request generated by the **EnvelopedRequestConnection: Enveloped (Optional)** client

Rights: String [0..Many] List of named access rights.

9.8.2. Message: ConnectResponse

Inherits: MeshResponse The connection request generated by the **EnvelopedAcknowledgeConnection: Enveloped (Optional)** client

EnvelopedProfileAccount: Enveloped (Optional) The user profile that provides the root of trust for this Mesh

9.9. Transaction: Complete

Request: CompleteRequest
Response: CompleteResponse

9.9.1. Message: CompleteRequest

Inherits: StatusRequest
AccountAddress: String (Optional) **9.9.2. Message: CompleteResponse**
ResponseID: String (Optional)

Inherits: MeshResponse The signed assertion describing the result
EnvelopedRespondConnection: Enveloped (Optional) of the connect
request

9.10. Transaction: Status

Request: StatusRequest
Response: StatusResponse **9.10.1. Message: StatusRequest**

Inherits: MeshRequestUser
DeviceUDF: String (Optional) **9.10.2. Message: StatusResponse**
ProfileMasterDigest: Binary (Optional)
Catalogs: String [0..Many]
Spools: String [0..Many]

Inherits: MeshResponse The account profile providing the root of
EnvelopedProfileAccount: Enveloped (Optional) trust for this
account.

EnvelopedCatalogedDevice: Enveloped (Optional) The catalog device
entry

ContainerStatus: ContainerStatus [0..Many] **9.11. Transaction:**
Download

Request: DownloadRequest
Response: DownloadResponse Request objects from the specified
container with the specified search
criteria.

9.11.1. Message: DownloadRequest

Inherits: MeshRequestUser
Request objects from the specified
container(s).

A client MAY request only objects matching specified search criteria
be returned and MAY request that only specific fields or parts of
the payload be returned.

Select: ConstraintsSelect [0..Many] Specifies constraints to be
applied to a search result. These allow a client to limit the

number of records returned, the quantity of data returned, the earliest and latest data returned, etc.

ConstraintsPost: ConstraintsData (Optional) Specifies the data constraints to be applied to the responses.

9.11.2. Message: DownloadResponse

Inherits: MeshResponse

Return the set of objects requested.

Services SHOULD NOT return a response that is disproportionately large relative to the speed of the network connection without a clear indication from the client that it is relevant. A service MAY limit the number of objects returned. A service MAY limit the scope of each response.

Updates: ContainerUpdate [0..Many] The updated data

9.12. Transaction: Transact

Request: TransactRequest

Response: TransactResponse Attempt an atomic transaction on the containers and spools associated with an account.

9.12.1. Message: TransactRequest

Inherits: MeshRequestUser

Upload entries to a container. This request is only valid if it is issued by the owner of the account

Updates: ContainerUpdate [0..Many] The data to be updated

Accounts: String [0..Many] The account(s) to which the request is directed.

Outbound: Enveloped [0..Many] The messages to be sent to other accounts

Inbound: Enveloped [0..Many] Messages to be appended to the user's inbound spool. this is typically used to post notifications to the user to mark messages as having been read or responded to.

Local: Enveloped [0..Many] Messages to be appended to the user's local spool. This is used to allow connecting devices to collect activation messages before they have connected to the mesh.

9.12.2. Message: TransactResponse

Inherits: MeshResponse

Response to an upload request.

Entries: EntryResponse [0..Many] The responses to the entries.

ConstraintsData: ConstraintsData (Optional) If the upload request contains redacted entries, specifies constraints that apply to the redacted entries as a group. Thus the total payloads of all the messages must not exceed the specified value.

9.12.3. Structure: EntryResponse

IndexRequest: Integer (Optional) The index value of the entry in the request.

IndexContainer: Integer (Optional) The index value assigned to the entry in the container.

Result: String (Optional) Specifies the result of attempting to add the entry to a catalog or spool. Valid values for a message are 'Accept', 'Reject'. Valid values for an entry are 'Accept', 'Reject' and 'Conflict'.

ConstraintsData: ConstraintsData (Optional) If the entry was redacted, specifies constraints that apply to the redacted entries as a group. Thus the total payloads of all the messages must not exceed the specified value.

9.13. Transaction: Post

Request: PostRequest

Response: PostResponse Request to post to a spool from an external party. The request and response messages are extensions of the corresponding messages for the Upload transaction. It is expected that additional fields will be added as the need arises.

9.13.1. Message: PostRequest

Inherits: MeshRequest The account(s) to which the request is directed.

Accounts: String [0..Many]

Messages: Enveloped [0..Many] The messages to be sent to the addresses specified in Accounts.

9.13.2. Message: PostResponse

Inherits: TransactResponse

[No fields]

9.14. Transaction: Claim

Request: ClaimRequest

Response: ClaimResponse Claim a publication

9.14.1. Message: ClaimRequest

Inherits: MeshRequest The claim message

EnvelopedMessageClaim: Enveloped (Optional)

9.14.2. Message:

ClaimResponse

Inherits: MeshResponse The encrypted device profile

CatalogedPublication: CatalogedPublication (Optional)

9.15. Transaction: PollClaim

Request: PollClaimRequest

Response: PollClaimResponse Check party making claim

9.15.1. Message: PollClaimRequest

Inherits: MeshRequest The envelope identifier formed from the

PublicationId: String (Optional) PublicationId.

TargetAccountAddress: String (Optional) Account to which the claim is directed

9.15.2. Message: PollClaimResponse

Inherits: MeshResponse The claim message

EnvelopedMessage: Enveloped (Optional)

9.15.3. Structure:

CryptographicOperation

KeyId: String (Optional) The key identifier

KeyCoefficient: Binary (Optional) Lagrange coefficient multiplier to be applied to the private key

9.15.4. Structure: CryptographicOperationSign

Inherits: CryptographicOperation The data to sign

Data: Binary (Optional)

PartialR: Binary (Optional) Contribution to the R offset.

9.15.5. Structure: CryptographicOperationKeyAgreement

Inherits: CryptographicOperation

[No fields]

9.15.6. Structure: CryptographicOperationGenerate

Inherits: `CryptographicOperation`

[No fields]

9.15.7. Structure: `CryptographicOperationShare`

Inherits: `CryptographicOperation`

Threshold: `Integer (Optional)` **9.15.8. Structure:**

Shares: `Integer (Optional)` `CryptographicResult`

Error: `String (Optional)`

9.15.9. Structure:

`CryptographicResultKeyAgreement`

Inherits: `CryptographicResult`

[No fields]

9.15.10. Structure: `CryptographicResultShare`

Inherits: `CryptographicResult`

[No fields]

9.16. Transaction: `Operate`

Request: `OperateRequest`

Response: `OperateResponse` Perform a set of cryptographic operations

9.16.1. Message: `OperateRequest`

Inherits: `MeshRequest` The service account the capability is bound

AccountAddress: `String (Optional)` to

9.16.2. Message: `OperateResponse`

Inherits: `MeshResponse`

[No fields]

10. Security Considerations

The security considerations for use and implementation of Mesh services and applications are described in the Mesh Security Considerations guide [[draft-hallambaker-mesh-security](#)].

11. IANA Considerations

All the IANA considerations for the Mesh documents are specified in this document

12. Acknowledgements

A list of people who have contributed to the design of the Mesh is presented in [[draft-hallambaker-mesh-architecture](#)].

13. Normative References

[draft-hallambaker-mesh-architecture]

Hallam-Baker, P., "Mathematical Mesh 3.0 Part I: Architecture Guide", Work in Progress, Internet-Draft, draft-hallambaker-mesh-architecture-15, 2 November 2020, <<https://tools.ietf.org/html/draft-hallambaker-mesh-architecture-15>>.

[draft-hallambaker-mesh-schema]

Hallam-Baker, P., "Mathematical Mesh 3.0 Part IV: Schema Reference", Work in Progress, Internet-Draft, draft-hallambaker-mesh-schema-06, 2 November 2020, <<https://tools.ietf.org/html/draft-hallambaker-mesh-schema-06>>.

[draft-hallambaker-mesh-security]

Hallam-Baker, P., "Mathematical Mesh 3.0 Part VII: Security Considerations", Work in Progress, Internet-Draft, draft-hallambaker-mesh-security-06, 2 November 2020, <<https://tools.ietf.org/html/draft-hallambaker-mesh-security-06>>.

[draft-hallambaker-mesh-udf]

Hallam-Baker, P., "Mathematical Mesh 3.0 Part II: Uniform Data Fingerprint.", Work in Progress, Internet-Draft, draft-hallambaker-mesh-udf-11, 2 November 2020, <<https://tools.ietf.org/html/draft-hallambaker-mesh-udf-11>>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/rfc/rfc2119>>.

[RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002, <<https://www.rfc-editor.org/rfc/rfc3339>>.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <<https://www.rfc-editor.org/rfc/rfc4648>>.

[RFC7230] Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <<https://www.rfc-editor.org/rfc/rfc7230>>.

[RFC8446]

Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/rfc/rfc8446>>.

14. Informative References

[draft-hallambaker-mesh-developer]

Hallam-Baker, P., "Mathematical Mesh: Reference Implementation", Work in Progress, Internet-Draft, draft-hallambaker-mesh-developer-10, 27 July 2020, <<https://tools.ietf.org/html/draft-hallambaker-mesh-developer-10>>.

[ECMA-262] Ecma International, "ECMAScript(R) 2017 Language Specification", June 2017.