

Workgroup: Network Working Group

Internet-Draft:

draft-hallambaker-mesh-protocol

Published: 25 October 2021

Intended Status: Informational

Expires: 28 April 2022

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 28 April 2022.

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](#)
- [6. Mesh Service Operations](#)
 - [6.1. Service Description](#)
 - [6.2. Account Management](#)
 - [6.2.1. Bind Account](#)
 - [6.2.2. Unbind Account](#)
 - [6.2.3. Account Recovery and Transfer.](#)
 - [6.3. Persistence Store Management](#)
 - [6.3.1. Status](#)
 - [6.3.2. Download](#)
 - [6.3.3. Transact](#)
 - [6.4. Device Connection](#)
 - [6.4.1. Connect](#)
 - [6.4.2. Complete](#)
 - [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. Threshold Sign](#)
 - [6.7. Messaging](#)
 - [6.7.1. Sender.](#)
 - [6.7.2. Outbound Service](#)
 - [6.7.3. Inbound Service](#)
 - [6.7.4. Recipient](#)
- [7. Access Control](#)
 - [7.1. Direct authorization](#)
 - [7.2. Access Catalog authentication](#)

- 8. [Message Interactions](#)
 - 8.1. [Message PIN Interaction](#)
 - 8.1.1. [Registration](#)
 - 8.1.2. [Authentication](#)
 - 8.1.3. [Validation](#)
 - 8.1.4. [Example](#)
 - 8.2. [Completion Interaction](#)
 - 8.3. [Contact Exchange Interaction](#)
 - 8.3.1. [Remote](#)
 - 8.3.2. [PIN](#)
 - 8.3.3. [EARL](#)
 - 8.4. [Group Invitation](#)
 - 8.5. [Confirmation Interaction](#)
- 9. [Device Connection Interactions](#)
 - 9.1. [Witness/PIN Authenticated](#)
 - 9.1.1. [Phase 1:](#)
 - 9.1.2. [Phase 2:](#)
 - 9.1.3. [Phase 3:](#)
 - 9.1.4. [Phase 4](#)
 - 9.2. [Preconfigured \(Static QR Code\)](#)
 - 9.2.1. [Phase 1](#)
 - 9.2.2. [Phase 2 & 3](#)
 - 9.2.3. [Phase 4](#)
 - 9.2.4. [Phase 5](#)
- 10. [Protocol Schema](#)
 - 10.1. [Request Messages](#)
 - 10.1.1. [Message: MeshRequest](#)
 - 10.1.2. [Message: MeshRequestUser](#)
 - 10.2. [Response Messages](#)
 - 10.2.1. [Message: MeshResponse](#)
 - 10.3. [Imported Objects](#)
 - 10.4. [Common Structures](#)
 - 10.4.1. [Structure: KeyValue](#)
 - 10.4.2. [Structure: ConstraintsSelect](#)
 - 10.4.3. [Structure: ConstraintsData](#)
 - 10.4.4. [Structure: PolicyAccount](#)
 - 10.4.5. [Structure: ContainerStatus](#)
 - 10.4.6. [Structure: ContainerUpdate](#)
 - 10.5. [Transaction: Hello](#)
 - 10.5.1. [Message: MeshHelloResponse](#)
 - 10.6. [Transaction: BindAccount](#)
 - 10.6.1. [Message: BindRequest](#)
 - 10.6.2. [Message: BindResponse](#)
 - 10.7. [Transaction: UnbindAccount](#)
 - 10.7.1. [Message: UnbindRequest](#)
 - 10.7.2. [Message: UnbindResponse](#)
 - 10.8. [Transaction: Connect](#)
 - 10.8.1. [Message: ConnectRequest](#)
 - 10.8.2. [Message: ConnectResponse](#)

10.9.	Transaction: Complete
10.9.1.	Message: CompleteRequest
10.9.2.	Message: CompleteResponse
10.10.	Transaction: Status
10.10.1.	Message: StatusRequest
10.10.2.	Message: StatusResponse
10.11.	Transaction: Download
10.11.1.	Message: DownloadRequest
10.11.2.	Message: DownloadResponse
10.12.	Transaction: Transact
10.12.1.	Message: TransactRequest
10.12.2.	Message: TransactResponse
10.12.3.	Structure: EntryResponse
10.13.	Transaction: Post
10.13.1.	Message: PostRequest
10.13.2.	Message: PostResponse
10.14.	Transaction: Claim
10.14.1.	Message: ClaimRequest
10.14.2.	Message: ClaimResponse
10.15.	Transaction: PollClaim
10.15.1.	Message: PollClaimRequest
10.15.2.	Message: PollClaimResponse
10.15.3.	Structure: CryptographicOperation
10.15.4.	Structure: CryptographicOperationSign
10.15.5.	Structure: CryptographicOperationKeyAgreement
10.15.6.	Structure: CryptographicOperationGenerate
10.15.7.	Structure: CryptographicOperationShare
10.15.8.	Structure: CryptographicResult
10.15.9.	Structure: CryptographicResultKeyAgreement
10.15.10.	Structure: CryptographicResultShare
10.16.	Transaction: Operate
10.16.1.	Message: OperateRequest
10.16.2.	Message: OperateResponse
11.	Security Considerations
12.	IANA Considerations
13.	Acknowledgements
14.	Normative References
15.	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 makes use of Reliable UDP Datagram (RUD) [[draft-hallambaker-mesh-rud](#)] for framing and authentication of individual requests and responses. These RUS packets are in turn exchanged over either HTTPS (i.e. a Web Service) or directly 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 UDP 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

The protocol binding maps the abstract protocol definition specified in this document to the network protocol format.

- *Discovery of network services.

- *Construction of the payload data by serializing request and response messages.

- *Authentication of the payload data.

- *Confidentiality controls to protect against traffic analysis

Currently only one protocol binding is specified: JSON-BCD Application Binding [[draft-hallambaker-jsonbcd](#)] over Reliable User Datagram (RUD) [[draft-hallambaker-mesh-rud](#)].

JSON-BCD Application Binding specifies the means by which data types such as 'integer' and 'datetime' etc. given in this document are serialized using JSON/JSON-B encoding.

Reliable User Datagram offers a presentation layer over a choice of HTTP or UDP transport.

6. Mesh Service Operations

The Mesh Service operations are divided into the following functional groups:

Service Description Describes the service.

Account Management Operations used to create, reclaim, and delete accounts.

Persistence Store Management Operations used to synchronize persistence store data across connected devices. [May be replaced in a future revision]

Device Connection Operations used by devices requesting connection to the account.

Publication Operations allowing a watched document to be posted to the service and claims made on the document returned to a device.

Cryptographic Cryptographic operations, including threshold operations performed by the service.

Messaging Exchange of messages between Mesh Services.

6.1. Service Description

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

The request payload only specifies that is is a request for the service description:

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

The response payload describes the service and the host providing that service:

[illegible]

```
KG9XC0gTVE0_YXG1DZWSSzhG6qXfEUU5QV8WiQXqFsEU9Q"}  
  ]}}
```

The current revision of the specification is designed for small scale deployments in which the service is provided by a single host. The approach will require revision in future versions to fully support a service being provided by multiple hosts with accounts being transferred between the hosts to allow balancing of load.

6.2. Account Management

There are three account management operations:

BindAccount Create an account bound to a service address.

UnbindAccount Delete an account bound to a service address

RecoverAccount [TBS] Reclaim an account using a recovered primary secret.

The BindAccount operation is used to create User and Group accounts. Currently, these account types are distinct. This may change in future releases.

6.2.1. Bind Account

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

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 contains all the information needed to create the account:

- *The account address

- *The account profile

Since there is no Access Catalog until the account is created, the Bind Account request and subsequent requests used to initialize the access catalog for the account **MUST** be authenticated by the Account Authentication key.

Alice requests creation of the account `alice@example.com`. The request payload is:

[illegible]

6.2.1.2. Account Recovery

Should all the administration devices be lost, an account **MAY** be recovered by the process of recovering the profile master secret and using it to access the account through the account authentication key.

6.2.2. Unbind Account

An account registration is deleted using the UnbindAccount transaction.

>>>> Unfinished ProtocolAccountDelete

The request payload:

The response payload:

6.2.2.1. Account Transfer

Should a user wish to transfer their account to a new service provider, they first use the Bind Account operation to bind the account to the new service provider, then populate the account entry at the new account using the account authentication key.

Only after the new account binding has been completed and is ready for use, is the unbind operation used to delete the account entry at the old service provider.

Future versions of the protocol will elaborate on this mechanism so that the change of address can be signaled to connected devices and parties sending messages to the account.

6.2.3. Account Recovery and Transfer.

Account recovery is necessary in the case that user has lost control of every administration device connected to the account and must re-create the account profile and bind a new set of administrative devices. Account transfer is the process of unbinding an account from one service and rebinding it to a new one.

These capabilities are both critical to the long term success of the Mesh but have been deleted from the current revision of the specification as their implementation is interdependent on the architecture of the callsign registry.

>>>> Unfinished ProtocolAccountRecover

[TBS]

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 32 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.

Alice adds an entry to her bookmark catalog. Before the bookmark can be added, the device synchronizes to the service. The synchronization process begins with a request for the status of all the stores associated with the account that it has access rights for:

```
{
  "StatusRequest":{
    "CatalogedDeviceDigest":"MDNG-A3QX-657G-UH45-KEKO-DIV2-4Q"}}
```

If the account has a very large number of stores, the device might only ask for the status of specific stores of interest.

The response specifies the status of each store specifying the index and Merkle tree apex digest values for each:

```

{
  "StatusResponse":{
    "Status":201,
    "StatusDescription":"Operation completed successfully",
    "ContainerStatus":[{
      "Container":"MMM_Inbound",
      "Index":3},
      {
        "Container":"MMM_Outbound",
        "Index":1,
        "Digest":"FEHy24Y6cLModDXWH31kVc2a3TdhjXPooKHpLAb2Jbs01YQ
nJolmowXAYHhkOGY0kg3jrKNTjds0myf4Dw1sdg"},
      {
        "Container":"MMM_Local",
        "Index":2},
      {
        "Container":"MMM_Access",
        "Index":3},
      {
        "Container":"MMM_Credential",
        "Index":4},
      {
        "Container":"MMM_Device",
        "Index":3},
      {
        "Container":"MMM_Contact",
        "Index":2},
      {
        "Container":"MMM_Application",
        "Index":1},
      {
        "Container":"MMM_Publication",
        "Index":1},
      {
        "Container":"MMM_Bookmark",
        "Index":1},
      {
        "Container":"MMM_Task",
        "Index":1}
    ]}]
}

```

Bug: The current version of the reference code is only returning the digest values for the outbound store.

6.3.2. Download

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

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

The previous status operation has reported that a new envelope has been added to the credential store. The device requests this data from the service:

```
{
  "DownloadRequest":{
    "Select":[{
      "Container":"MMM_Credential",
      "IndexMin":3,
      "IndexMax":4}
    ]}}

```

The response contains the requested envelope:

```

{
  "DownloadResponse":{
    "Status":201,
    "StatusDescription":"Operation completed successfully",
    "Updates":[{
      "Container":"MMM_Credential",
      "Envelopes":[{
        "PayloadDigest":"scKJJY0e21lHKRImyYAHL98MSo62-eVSTz
8JkFFaicCDM1Nskxm5JW1WIUy4XhKdhYTYagTRFxNTsbABRAOT7w",
        "enc":"A256CBC",
        "dig":"S512",
        "Salt":"28sn1l7vR0Y1rqAjTNaxzg",
        "recipients":[{
          "kid":"MC7F-DLCK-JI67-VFL7-B0HX-T62Q-KCVG",
          "epk":{
            "PublicKeyECDH":{
              "crv":"X448",
              "Public":"lGsU2MtoCW3h7kBLBfm4eN9xXqVSVbR_9
Es_47TEqVo2HYkeS0lkFE1hPNCz98yD-xFx_9omFj4A"}},
          "wmk":"tyvbkb9eXzVAFqYyTn12v0cC18vtSIIIfmPR6hpS
LPoA0RyeVaD2rg"}
        ],
        "ContentMetaData":"ewogICJVbm1xdWVJZCI6ICI6ZnRwLmV4
YW1wbGUuY29tIiwKICAiRXZlbnQiOiAiVXBkYXRlIiwKICAiRmlyc3QiOiAxLAogI
CJQcmV2aW91cyI6IDF9",
        "SequenceInfo":{
          "Index":3,
          "TreePosition":716},
        "Received":"2021-10-25T15:48:48Z"},
        "GnC2lneENSTxMbQ6W91xcsDRs1Ap9P5PRn7MHvCQ1hEMWiGww91t
r5llzPtdeZz1-Fx5Cc49FrdanP8dtWZVNMTg4yQMf5bbaRzte4CzUrKihFYdJ3SK
GAm2EC317mujVLb29kqnkJkmdLUJu41yYZ40LRe1rM_xR1t0VlkaE",
        {}
      ]
    ]}
  ]}
}

```

Future: The current implementation of the download operation is limited by the capabilities of the HTTP binding of the RUD transport. A future binding allowing operations that consist of a single request followed by a sequence of responses will allow much greater flexibility.

Future versions of the protocol may support optional filtering criteria so that the service only returns objects matching specific criteria and/or only return certain parts of the selected messages.

6.3.3. 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 allows Mesh stores to provide ACID (Atomicity, Consistency, Isolation, Durability) properties to the applications they serve.

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.

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.

The request payload specifies the data to be appended to the stores.

```
{
  "TransactRequest":{
    "Updates":[{
      "Container":"MMM_Bookmark",
      "Envelopes":[{
        "dig":"S512",
        "ContentMetaData":"ewogICJVbmlxdWVJZCI6ICJTaXRlcy4y
IiwKICAiRXZlbnQiOiAiTmV3In0",
        "SequenceInfo":{
          "Index":1,
          "TreePosition":0}},
        "ewogICJDYXRhbG9nZWRCb29rbWFyayI6IHsKICAgICJVcmkiOiAi
aHR0cDovL3d3dy5leGFtcGxlLm5ldCIsCiAgICAiVG10bGUiOiAic2l0ZTIiLAogI
CAgIlBhdGgiOiAiU2l0ZXMuMiJ9fQ",
        {
          "PayloadDigest":"gtpamSravs9YkD3Wi6-rIFqFOINwLFj8Q2
eGpMjmbyp-_TRCgRs9Hqpo3bJPhoRSgUmFIUsQTDNeiT414W56eA",
          "TreeDigest":"TpXg14cDEx_-1Qe-h1qiryihsl00MrUCLW0L7
wvq-YLCEWZfAIrp9FmBwNE0se8UN1nFY4h1aqXbN3yBuKfg9w"}
        ]
      }
    ]}
  ]}}
```

The response reports successful completion:

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

6.4. Device Connection

In order to support the wide range of affordances supported by devices, four device connection interactions are currently specified. The use of these mechanisms is described in [[draft-hallambaker-mesh-architecture](#)] and the interactions themselves are described in section ??? following.

Device connection operations are always issued by a device requesting connection to a Mesh account and must therefore be authenticated under the device profile rather than the account profile. Two device connection operations are currently defined:

Connect Requests connection to the account.

Complete Polls for completion of a connection request.

Since the second operation is merely polling for completion of the transaction requested by the first, it is likely that these will be combined in a future revision of the specification.

6.4.1. Connect

If the connection request is initiated by the device being connected, the device constructs a RequestConnection message which is posted to the Mesh Service using the Connect operation.

If the Connect operation is accepted (i.e. the service determines it is not abuse), the service constructs an AcknowledgeConnection message which is forwarded to the inbound spool of the account to which connection is requested. The requesting device receives a copy of the AcknowledgeConnection message and the profile of the account it is requesting connection to.

As described in the following section, the AcknowledgeConnection message contains the request details presented by the device and a nonce value generated by the service. This nonce value is used to compute the witness value that will be used for mutual authentication of the device and account.

The connect request is made to the service, not the account. The payload contains the enveloped connection request:

[illegible]


```
{
  "ConnectResponse": {
    "Status": 201,
    "StatusDescription": "Operation completed successfully",
    "EnvelopedAcknowledgeConnection": {
      "EnvelopeId": "MBQD-SA00-FLPI-PKWI-WYR6-PNVY-VTC4",
      "ContentMetaData": "ewogICJYbmlxdWVJZCI6IClYV1pQLUt0WkYtSk
1LTy1SUUVNLVdZVEgtVVUzNS10V1hWIiwKICAiTWVzc2FnZVR5cGUiOiAiQWNRbm9
3bGVkZ2VDb25uZWNoaW9uIiwKICAiY3R5IjogImFwcGxpY2F0aW9uL21tbS9vYmpl
Y3QiLAogICJDCmVhdGVkIjogIjIwMjE0MTA0MjU0MDk6MDJhIn0",
      "ewogICJBY2tub3dsZWRnZUNvbm5lY3Rpb24iOiB7CiAgICAiTWVzc2FnZU
lkIjogIjJXWlAtS05aRi1KTUtPLVJR1U1V1lUSC1VVTM1LU5XWFYiLAogICAgIkv
udmVsb3BlZFJlcXVlc3RDb25uZWNoaW9uIjogW3sKICAgICAgICAgICAgICAgICAg
ZCI6ICJNRETXLNTLT0QtWlRXNi1NUKlCLUFBUkstVUFDTs1QRE9aIiwKICAgICAgI
CAiQ29udGVudE1ldGFEYXRhIjogImV3b2dJQ0pWYm1seGRXVkpkaQ0k2SUNKT1FUUT
JMVWVhUmtjdFRqVk9WUzEKICBGV0V0YUxU1lOMGN0UjFOR055MUVWVmRUSWl3S0l
DQWlUV1Z6YzJGblpWUjVjR1VpT2lBaVvtVnhkV1Z6ZAogIEV0dmJtNwXZM1JwYjI0
aUxBb2dJQ0pqZEHraU9pQWlZWEJ3YkdsallYUnBiMjR2YlY3dEYyOWlhbVZqZENJC
iAgc0NpQWdJa055WldGMFpXUWlPaUFpTWpBeU1TMHhNQzB5TlZReE5UbzBPVG93TW
xvawZRIIn0sCiAgICAgICJld29nSUNKU1pYRjFawE4wUTI5dWJtVmpkR2x2Ym1JNk1
Ic0tJQ0FnSUNKCiAgTlpYTnpZV2RsU1dRaU9pQWlUa0UwTmkxSVUxWkhMVTQxVGxV
dFJwaExaTAwV0RKSExVZFRSamN0UkZWWFUKICB5SXNDaUFnSUNBaVFYVjBhR1Z1Z
EdsallYUmxarVJoZEdFaU9pQmJld29nSUNBZ0lDQWdJQ0pGYm5abGJHOQogIHdaVw
xrSwpvZ0lrMURWazR0V0V4TVZDMU1URTvYTFZVMFNGSXRRazl0UnkxU1FUWmFMVlZ
YVWxJaUxBb2dJCiAgQ0FnSUNBZ0lDSmthV2NpT2lBaVv6VXhNaUlzQ2lBZ0lDQWdJ
Q0FnSWtOdmJuUmxiY0lJbWlSaFJHRjBZU0kKICA2SUNKbGQyOW5TVU5LVm1KdGJlIa
GtWMVpLV2t0Sk5rbERTazVSTVZwUFRGwm9UUVlJHVVhSVVJYaFBWbmt4QWogIGlBZ1
ZrNUZhRk5NVlVwUVZGVmpkRlZyUlRkWGfURldWakZLVtBscGQWdEpRMEZwVkJkV2V
tTXlSbTVhVmxJCiAgMVkwZGFhV1lUUVdsVlNfcDlJbWlZbWlZbWlZbWlZbWlZbWlZ
cVdsTkpjME5wUVdkSmJVNHdaVh5KTmtSRFMKICBtaGpTRUp6WVZkT2FHUKhiSFppY
VRsMFlsY3dkbU15U25GYVYwNHdTV2wzUzBsRFRB2dJR2xSTTBwc1dWaogIFNiRn
BEU1RaSlEwbDVUVVJKZUV4VWJYZE1WRWt4VmtSRk1VOXFVVFZQYwtGNVYyYbEtPU0o
5TEFvZ0lDQWdJCiAgQ0FpWlhdlowbERTbEZqYlRsdFlwZDRiRkpIVmpKaFYwNXNT
V3B2WjJWM2IyZEpRMEZuU1d4Q2VXSXlXZ28KICBnSuhCaVixWlVZVmRrZFZsWVvQr
mpiVlZwVdJsQ04wTnBRV2RKUTBGblNVTktwBHBIV1dsUGFVRnBWRlZPVgogIDFScE
1wbFVSWGhWVEZWNFRWUnNZM1JXQ2lBZ1ZGSkpWV2t4UTFRd01VaE1Wa3BDVG14dmR
GWldaRk5WYVVsCiAge1EybEJaMGxEUVdkSlEwcFJaRmRLYzJGWFRRlRlpXRXBvWwXk
V01GcFlTbnBKYW04S0lDQm5aWGR2WjBsRFEKICBxZEprMEZuU1VOS1VXUlhTbk5oV
jA1TVdsaHNSbEV3VWtsSmFt0W5aWGR2WjBsRFFXZEprMEZuU1VOQlowbAogIHRUbm
xrYVVRm1NRb2dJR1U5LUmxwRVVUQlBRMGx6UTJlQlowbERRV2RKUTBGblNVTkjhVlZ
JVM1saViYeHFTCiAgV3B2WjBsdFVYcE5iR2N4WwpOT1NFMVZkRk1JTVkZaM1lsZGFD
aUFnVjFVSlduRlNSWE13VFVZNE5HVkhTaLYKICBTVkU1eVdteFdNMVF6WkZWwMjFS
1lUVMhhu21WWE9IcFJNRTVQWkd0dlMwbERRVZ0V0VkelPwUkdRazBLSQogIENCc1
p6UlVhbFp0Wldwb1dGTllVbFJYlYld3e1V6QkZhV1pZTVRsTVFXOW5TVU5CwJbSc1Z
uVlpNMG8xWTBoCiAgU2NHSXl0R2xQYVVMjEybEJaMGxEUVFvZ0lHZEprMHXBV2tk
WmFVOXBRV2xvVlZKT1Zua3hXRkV3YkZ0TVYKICBwcExWRlU0ZEU0eGNFbE9hVEZFV
kdWt1MweFZkRlRlpYTNkMFZXdDRRBGREU1h0RENpQWdhVUZuU1VOQlowbAogIERTbE
ZrVjBwellWZE9VVMxZU21oaVYxWXdXbGhLZwtscWiYzGxkMjluU1VOQlowbERRV2R
KUTBwUlpGZEtjCiAgMkZYVGt4YVdHd0tJQ0JHVVRU1NVbHFiMmRsZDI5b1NVTKJa
```

MGxEUVdkSlEwRm5TVzFPZVdScFNUWkpRMHAKICBaVGtSUK5FbHBkMHRKUTBGblNVT
kJaMGxEUVdkSlEwCFJaQW9nSUZkS2MyRlhUV2xQYVVGcFltNVNabU5JVgogIFRCWF
ZrcHBaVmHkZDFkVmFNVpNVXBZWkcxc1RreFlTbFZYkdwVdteENNVlZXyUZkaE1
XZ3dWRmRrQ2lBCiAgZ2RXUXlhSGRsVlZaWVpHcENTRlZ0Y0hOaFFXOW5TVVJzVm10
dVFsQmpNakZxVmxSSk0weFhlSFJsYm1oS1YKICBETmtWRmRIY0VKU1UwbzVabGd3Y
zBNS0lDQnBRV2RKUTBGcFZUSnNibUp0UmpCa1dFcHNTV3B2WjJWM2IyZAogIEpRME
ZuU1VOQmFWWlhVbTFKYW05b1NXc3hRMDfzVlhST2EyUk9Ubm4VVZORmF3b2dJSHB
NVkU1V1ZsVTBkCiAgRlJGYkdGT2FURldwbFZrUzB4VmJGaE9Wa1ZwVEVGdlowbERR
V2RKUTBGcFZVaFdhV0pIYkdwVlIwWjVXVMmKICB4YkdSSFZubGpDaUFnZVVRm1NVa
HpTMGxEUVdkSlEwRm5TVU5CYVZWsvZtbG1SMnhxVXpKV05WSlZUa1ZUUQogIDBrM1
NVaHpTMGxEUVdkSlEwRm5TVU5CWjBsRFNtcGpibGtLSUNCcFQybEJhVkpYVVRCT1J
HZHBURUZ2WjBsCiAgRFFXZEprMEZuU1VOQlowbHNRakZaYlhod1dYbEp0a2xEU2xa
aWJXc3lUbFZXVlZrd1VsbFpiVlpZWVfVz0kKICBGY3hVbUpHYXpWUfDHaG9VMGMxV
TFkdGNHbFRSbkJDVXpKc1ZVNXRvbXhhUkZJd1RWZHDNbFJIY0VaV2JXaAogIE5Zak
5zV1V0cFFXZFdiRvp5WVZkU0NpQWdiMw94Ykh0V01qRm1Va2h0Tws5RVpGQmtWWEJ
2V0RGV1FrhHVNCiAgVGxtVTNkTFNVTkJaMGxEU2tKa1dGsnZXbGMxTudGWFRtaGtS
MngyWW1sSk5ra0tJQ0JJYzB0SlEwRm5TVU4KICBCWjBsc1ZtdGFhVWsyU1VOS1RsR
nJiRVJNVlVsNVV6QlJkRkZyU2x0WFF6RklUa1pDUlV4VVVrcE5hemgwVgogIFVVeF
JWUnBNuW9nSUZovU1XUkNTV2wzUzBsRFFXZEprMEZuU1d4Q01wbHRlSEJaTVVKb1k
yMUdkRnBZVW14CiAgamJrMXBUMmxDTjB0cFFXZEprMEZuU1VOQlowbHNRakZaQ2lB
Z2JYaHdXVEIwYkdWlZrU1NSV2RwVDJsQ04KICAwTnBRV2RKUTBGblNVTkJaMGxEU
Vdswk0wb3lTV3B2WjBsc1p6Qk9SR2RwVEVGdlowbERRV2RKUTBFS0lDQgogIG5TVU
5CWjBsc1FqRlpiWgh3V1hsSk5rbERTalZqUjBwVvZqRmFibE5JUlhkaU1qVnZWREo
wVlZReFJqUk5WCiAgVTvyV2p0a1NWSldVbEZVUld4VFZBb2dJRLpSTVdGWE1WTlRN
SEJtVFVkdMVsWjZRa3RhYm10MFZtczFkVU4KICBwUVdkUFZFcHRWSHBhY2x0R2JEQ
k5NbVJVWpCb1dGTnRNRfJVV0ZwWFRraGtDaUFnUWtsdU1UbG1XREU1SQogIGl3S0
lDQwdJQ0FnZXdvZ0lDQwdJQ0FnSUNKemFXZHvZWFIXY21weklqb2dXM3NLSUNBZ0l
DQwdJQ0FnSUNBCiAgZ0ltRnNaeUk2SUNKVE5URXlJaXdLSUNBZ0lDQwdJQ0FnSUNB
Z0ltDhBaQ0k2SUNKTlExwk9MVmhNVEZRdFQKICBFeE9WeTFWTKVoU0xVS1BUVWNOV
WtFMldpMVZWMUpTSWl3S0lDQwdJQ0FnSUNBZ0lDQwdJbk5WwJi1aGRIVgogIHlaU0
k2SUNKR1VHcGpRM3B5TjNnd1JtSlZTSEphVDA5b1ZVZDFaWE5WVGtKS1QwNVlPVVp
sTFV0ZlH6ZzNaCiAgWGxyU0ZjMVZVOTVdaUFnYkV4b2JtWjRabXRWVEZGV1VrbFpN
MmRrUm1kbVRGTktOVzFCVEZsUk0zWTVVa3gKICBLZEdoa1VHaE5jSGhFWm5WNVNXb
EVNMVowTFdOb2J6SlJSMkVLSUNCVGNUZHBiVTh0V214TFdreFFYMnAzVAogIHpRNF
FXNXhZMDVhYmtwM1kyUkxUVVpvyTnWYVJHcHJRu0o5WfN3S0lDQwdJQ0FnSUNBaV
HRjViRzlowkVSCiAgcFoyVnPkQ0k2SUNKSVowdEpOVlpqWTNwc1Vta3dYMGc1Y1VW
bFlubGZXV3hyT0hwRFZHeGxURzFvZW1WwFYKICBtOW1hMWRqWTJZS0lDQm5jRU5zU
1RGb1VrZ3pabTVmU2xwQlNscHhZWfUzTm04eVFXRlZWRkIXTXkxRVpYVQogIDVWRm
hoZHlKOVhTd0tJQ0FnSUNKRGJHbGxib1JPYjI1alpTSTZJQ0l0Y0hsMWEwRTRTMHB
4WkhaV1gyaGxVCiAgRWxMUmxwUklpd0tJQ0FnSUNKUWFXNUpaQ0k2SUNKQlFWQlBM
VkJWUTBzdFFVbFpXaTFHVTA5WuxV0UNTfVUKICB0V1ZwYVfPMVNwbFF5SWl3S0lDQ
wdJQ0pRYVc1WGFYUnVaWE56SWpvZ0luZFdNRfJqY2tKaFpqZG9MVfZtWQogIDJ4V1
FVZE5TWE41TlZwVldtNUxZM0ZpVUZoRVZXSjFxbVpZV1c5NmRVazVXZ29nSUVJdFp
XMWxkmjV4TW1GCiAgM2RuQjN0bWszYjBaMloxa3RiMWN3YwxGeVZWbHhTbE5VVjBS
bk1pd0tJQ0FnSUNKQlkyTnZkVzUwUVdSa2MKICBtVnpjeUk2SUNKaGJHbGpaVUJsZ
UdGdGNHeGxMbU52YlNKOWZRIl0sCiAgICaiU2VydmVyTm9uY2Ui0iAicU85UjNvVD
I0RURPNudDWxZQ0JzZyIsCiAgICaiV2l0bmVzcyI6ICiYv1pQLUt0WkytSk1LTy1
SUVNVlVdZVEgtVVUzNS10V1hwIn19"

],

[illegible]

```

Z9e60BCW7kFIqbADT4jF0fB0VI7bf15uh3coVtpXAtHehAA"}
    ],
    "PayloadDigest": "0_av1I9T_vQ-6biLixf0vQ-_JLiUttOyYnb5fPbq
u5l3agCn0lgRF18uGdSgmzVqzUSIxQl36g-SDrhWApbyEw"}
  ]}]

```

6.4.2. Complete

The complete operation is used to complete the binding of a device to the account regardless of whether the operation is initiated by the administration device or the connecting device.

The complete request is made to the service, not the account. The payload specifies the account the device is requesting completion for and the identifier of the completion message.

```

{
  "CompleteRequest": {
    "AccountAddress": "alice@example.com",
    "ResponseID": "MDT3-TM62-G3X0-ESY0-WQZX-IR2B-YNHW"}}

```

The response payload:

```
{
  "CompleteResponse":{
    "Status":201,
    "StatusDescription":"Operation completed successfully",
    "EnvelopedRespondConnection":[{
      "EnvelopeId":"MA5F-7LDW-G2AQ-N4PQ-N7XJ-243G-CH4B",
      "enc":"A256CBC",
      "Salt":"5N4XeR89WnRUPeNN9eehLw",
      "recipients":[{
        "kid":"MDMW-WCIR-FJMO-7ZH6-CN3J-KUZL-RLAX",
        "epk":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"x4NzHBx1XxAiMAvIgZh2htXH9is-DGf71wwwqJh
j1WZcZds2vBOGHhXCRI85oGRbSWr-rXRNSuYA"}},
          "wmk":"DRB_GkoKIfvQZ7RTrJHrYxj5e82Npx6MPiXee-tIrWhmP
rA025oPw"}
        ],
        "ContentMetaData":"ewogICJVbmlxdWVJZCI6ICJNRFQzLVRNNjItRz
NYTy1FU1lPLVdRWlgtSVIyQi1ZTkXIIiWKAiTWVzc2FnZVR5cGUiOiAiUmVzcG9
uZENvbml5Y3Rpb24iLAogICJjdHkiOiAiYXBwbGljYXRpb24vbW1tL29iamVjdCI
sCiAgIkNyZWFOZWQwI0iAiMjAyMS0xMC0yNVQxNT00TowM1oifQ",
        "SequenceInfo":{
          "Index":3,
          "TreePosition":426},
        "Received":"2021-10-25T15:49:03Z"},
        "T1FrG5eIYXEffmfCQyy6gUW1bLLzDyAbog1Np006M8qoFT2MPW8xwZoqc
sE0014nHsPbx0eFfs9VC0oS78oZBDxayLqSImKNQc6xs6thMCvRbnGyZ30TSj0DI9
-aKyr52FcC47d4ZUPS2u1-egzr3LUUHX623rBjbtz9eIu3jWaqBT2G2Fwa6AE7ekX
R06xPsRK7exnHVTJZV2P-KMa4fSA11i5SZTgxnX7uIMbTmn4fA-a1K7z-EalQ63wR
L28xpqw5ajQp-1P8F4gXc1o_MLZHRfEvZC-dkhr_5iaB53UJH9tjxPRzgdikY6nH
saifrQReQcTrgKGdGUKcN2USIN33wUwIEnmr7dwiJIIdFWxqFWMbE-8ni0LL4PEkyp
SNieWNxW1G4ED2sjcFL8wHjD1k892F7Qh6w2D9Y1UCNNuoa6-f6o6i1f7h-2IgoVG
B56yOYHYP7omRCaYk2-l8w-hqITzdsoj1xU1uqYBFF7rapyGAGS1FMHq4_wDI3YSF
pRckLGEV_MzHPM6WKgXRY1Ri-lPPooBDmr6PRerM7ki_EKx68r6XV0q43_d0FoMa6
VgEvu-kIXd1JLAN5ky-dNrdELfwgF8HFvshNLvsMfXx9y93DzA8rRNqeJ9BCnsE9
pSAVzJ5S3eL6AKYsdQ6gJDtFD_Tfcxf0SEBq0w8gc02Q6uIkRq2Y0SVIR_LuqZSEC
wTpWr8_VcK3iHXcCaoLrFf3_x92QSR5HcrRImBMY9CUCTGLJ45ry-G0Je0ZM8C7WQ
MvufGTIKE-4z0YnrZKYwfgWmIMdKXv-G54QRHSM_sAxxEd7NF8r6bzuu_kwTJnjKZ
zR5tDy6Bgsc6wf60s-Q1kJ52VpdBMxKzBDCcH_2JNv2k4rV7F5aQzYnLBP4jWsxh
pWMXtYTTsvEeg3UTatSxeh04XuvhFeFtiqH9o6JJonJUq0KLw0TiULFn1GXqXizxv
PX5PzXfjbtBEFVlaadHX9K7k6ebzRmVSpwWn4h4io78e1FkdyLjJR23_ID53IbEtB
V14KqevNyuXLKy3WVZW9jWR1DggON-xG6h1WwyTq9JFBCPj6qkzjI6wz6wTUTiNxt
PZoh_Rj8PbouzWD62Zm0CqFr9c9-9EHJZcMSMPQAi1H0FR0ao9NiWU0Ya_je2he1B
CFgf8zTBT1L_kZXbpkvm1Qf0YaJUddkSkJRIASLBtreLhJF9QT-9ln42oE1_r61Y6
mv6hSG6nMBDKI7jlua_HbTq7c0dPfbC3wJki-oj0J_hH07IuqtgrMDeHih8dsonhk
mYKahV8k0-J_t02ki92h1iDYbNzAftM5FbfbwzMsu7E1xtMxKdhHemHUZZilhi--G
oipVdXUfouTK4TQ31Q6X5afpnvZ_wBr90UpW0rKUabQ1DRocie-pf6g_M337csJqN
lLDpwXeItNuD9beZvjpt71K9HAI2uqDbZwwUXFquDbK_ZsU3N7NYFYRi1_CxgS5IO
```


XyJNRI05H6_no9GGUy8r0poP2gpEDK01VKJlU90ztjHlde_tEG1iRuFrr8bfAvow1
HLrijGIEXQDC7J8Dax_zJj1uRcH3PBhuNY0qQ-Ipa60PIQY-hwC5ylfBuB3bSMcua
FYWKCMmQt0BX6Q4-JxvD-uYmyslXq3iQBA2VFAR8wCfZrahuUwdH1mmQSBikyK2i2
KD6Aiuas7RaOm0eHEX9n3dMzOGsL10pf0UWHqGvLp7lwmpSHD0aYe6fpjVF3Rb4dn
naGq6nzNH8AGqLgLIB2Gff-GPqCaQp2y6fyjeIKFMWEZYpxiJ2fgkGMInN_1Q8WeI
vUCEzJc-xqLEaVsbylYNKv8YXk8dd-Jw1qoia16YP0W7_aMMm-_wHDYMKU8W6q0GT
ZVZZHXtZKVNESRC5RsI5wJBrUo-e2-0R0_5AUDKBdpJ1LJBAYF1nhkXA0beha07fa
fEGLUS8b46pGxDiZs17mwF1PRmnR8bFyXPbs8lGBMYM0vR5NiDZE00kRbEMTB_e_nM
_HzyYAmzClyP4dPtpH3nDXT0hjkf81bxKJ6hMOHP-fX1jXciYqKJg-fJrWkCDLXRU
k_Yh8G6Vzoh2lEKHhLdw_uHV8ivo4Ur8D1UY1DYsWhab3K2U6a1lFC6P5PSIHy00M
3GR3egXb56woJR0syi0x7zA1185Bk-2BQkMvLZcTtZ0nmuk64yJBchHc0rFR4780p
4BXdhAfcuNy0q9dtY7Yw2s2NlJ9sbRgnqMfRRkOWChOZ8eDCgq-u5PoTdHAwek43S
RV7jHn6LlUVKl76_EaivyWwgWIKd67lqlvzw8o8yxeDiq3KScT1vDtC1-oYlMQ3Wc
pqgs_PMbXTebTgGjKcengZ2ZxYPPkJLqU_05tvFP-kI3SY00zG73YeICbhAWfTEZ7
AwBZwxE2100WkJvNMW0sWv3bPfCaaYs63F0TqLz9tb0ASj3gl0sWlJALJy-N033Vh
FvzaUqETHNAAR12o5Zr3kPNTaQ2GtT9zDA2UGlFK0oDs5Z5GDzCRlM0mJfeAq7ITm
dGBvn1vSjF3R9kX78vELK6JtUXVYTrDzdJbKRKoQFaxXKYFwxKnwaaYLEVafVThW_
Tis2G_9CvEVyX3oizzK3nn_jJZiCW7z7ajEcLTIJDEC3mPZZKIhb51yWA-aKQ_fbK
5iJVZbolQGZNMjvYDkpaFNMcg49HQ8Nz-b3cYVzoMIEEIj1BDVwc_Xx60N6wj1YUA
JA-Nnz1YrX2Av0yh5qeaYG6__9qdNT00h1HXX2AtVJJ1wzvE2jQwFuU1ii__gYE5I
6RVcNM0sTM-JNNtQDhw2Z_UTCWj-7xbCRuqzLqQ-lkKVzMo063N7L7uPB3j0HCpN5
N3__1A12d-yeQR47b5JEWHYadeU00kPSdmIwaEDoGZ7nHhNQhV4q23EOPQ4xYr315
7hXwtc0sJR7EcuNi-dXt6Mi1-08QxEiedGDORZbMzqhNIsiVL74gfK7YMuPQbwcTo
MMvzydfSDwa0MZZjHvr-7_uP6fkGpGijB2sYnKzMIKYSneExdRuABNXEgEsc0jKsS
KGDpevK8iy0AqYCyHXXWjzWrrEUhLx80ZXsPTJlHu_IYTo5Ui_2Txgg-9Dp7YPuYi
yXyl4GXrmmAFrUB0MMBIlt4o32MTugKlfQs6xnCUQLnbPj_vDLJKD4Sb21Jf40H4p
uF3z8EvPQE0X32zUaL7ztDK4r8qs-BaeD6NaiMG5quE1kktKYPfYTrLWkv5pUTtI8
YJrZ2lqsTHQB9iaEaPSNaDExpfiFpM3SdlAaUXoUXMUWm8BrTEf1QWAEWaP3WrDh
3PhEUKjuE52zemrctLB5nQupLQzPK8nv0DafjMZAikypj53MORFh3Us8sHxRoauk4
9QwEg7Uhw-gDTirMIpW-xlUYzoG72AY01k7GgJ0pr3F8ZxeBAzKN-Yv0DipnINAcc
0QnDtorSnsScQsrUwmn_KRyPg-zit6k1DQPNQc7-28nRkWqczMxqyK9040Rde-3cpF
Wbc4N7Kd2iuQawPRRDmcCLd_JzmfowdL3UaLGzUTaxdKbvTirtPtgaGyi95V5lsq1
0yvI1B8d4VR0uvFvWAl6TEYVeFhvtJyWQyWSLMwOG_8aVCyxI35rMSB2f8bjwxAiF
L050tSuen9waUV_wTIuax80V0jy4w5inXSF6B0SdXAJso4KJB9j42eW__ATXfQivf
Bdz5bgvUXucQPS2GJTF-qkLUQ9nDnr7uSzB2h18D9DCYtid5ahTDGUESaZjMK_KK
qVw4fp3GmoTxxoXs1p708VF16tjnqPhH6-EC1X3pmglW95raBZAEhc2TZTgjrd_Qj
pPcWAsEqLouVbU3Q8dusGPGIx-WJfpGi0ZHznFoTqDoMvFfSu-UBeha7gHzudU_hQ
Fo0yvFBuij94F72hx8vVKbc-iLGXmw_tSicy3kHfCXB0TsoUaxKYowKM1-SsGHDbG
jPFYU7XWD9DIXrYXwFgdqQ_bWh2lXLFi-Vvn0JxUEN6Psb6hYAA17vAg_P47NiAo5
0zkAPaoSh2BZLqQlYnAsFmy9zM0W2SceEIPhiqwn9e1H5cSYnr_a3KSgaLxV-M6A
Smr73KUzb-kdeHf4Yx-Hxm8inKhwkFNBIyudnyS-V7Wa80mvlgy8th-psFBqtGgpk
Mpkj7pOM1-FoX5-E121Z95Zdw581VQ1iYg6V1MQXPk17qBbX7xUYPWgQcq5XxyTq3
zGKo_0IHNXaorhj3eDoUwso7YQeJhs0NcxmrU5jiFM2JKVsLwXlsWaAjsJt07Coan
kQCXZ2h8XS5jB-pKcPx-072s4chxwFEY48cEA4zadPJeQZ9DcUr50QhfgVhdQrEt9
u7N4PbP0xuh7xINQ4hkgjfH-oRzPM96uRbEiDAFl1L62dV95JdKPr4NP9k2JN7Q0q
n1AoFLZB2Apz_nU1oT8EiJb1DzocQZ6e-xHQgdbUx5rz4fIREZSpQo5l_swknavfk
wF52a2GZKk4090jBnrNKMA43iDA5gdmG_sMzKB9-0e1clj7hwaVOTamV06KNDzvnp
fG6h_1nCoGubSxNurRmaNYfD5peV1D8icP0cD0tTJ7RpzmSqp80AFWIZ7NU3cNsNd
9E_AowNSmL5kCfHz47vSgRGSnILJAXdLc9A-0cDl3q9vroUXggVU2nJztC8ngtL71

Yy3VIDhMjDm0V9aeCpLhKuzS9qjKLCrF_SexHWX6AfpjjoKvf9A88doKWapx6tgsi
i4y9ms0YGgphjNgzDW-mXR5wW9QRH3iMYWw-lu6yUKTeg5pkZGEHMu12Ppq12ebej
vgFYTwdGhT_2lwZbIF00BhH5Gnn-bmqR30w7K0KhXb9XJtQU0gSkmw500SQT6ZNT
010xGtLuR0dMjj-DopqD2gJVKmWCPTjdsuVyv_HZKtgrce8b0Cxs70nsWDL05JdVu
uel0w0sya3HwQScwaZRt70QnEw6r1gYJ11hx1L1AKK2ZzyRSSVBL6WPfXjQn139j0
mPooM5Q5NfdQaFkjIqj0lu0cMJCAMMeY7B80R3evBHGTv7oYvarbhtoDvDYfB9g-b
O_5eICs9EDtD-vs7-VGW4TQNQ0xuRmSzvYgIGZlXH8R1HWT7x6I62fBM0Kb0L2jMo
4G2WLGuhEy_TK3yNb1G9ozxrMm-xZVru120vkKmJu5AJZLGSUir4vm1aueDqC4pDh
IbmVRPLk0NHDFib6ffUcgEpNhKpfa04yr5Tzkajg6pIMZ6wrwZBaySRvNM8qhhHZK
la1s2peaQ-ym20IW4M_TDGiqj0E0rwcTsr3mLcMjRSUajHsNfRhqmjsR6eQxdIJr
Kjdk6rrKCzNYD_hg_oBb1Cn-NyLR-zU0fyqBNr_OET1tr2Hk08rCqrItZdEheQSe
q9of4XwcJE-VgW-VeGw1GQIHj4FVVGYaSr50AJSJ6a84Tf4iizbBPzEfdqFpyvm0L
N710E5_wdzwQZgXgSEnR3PXvecjIyg_gS47QBZc9k06rCnW6dgXK1lnYqsQqRt5bK
vCzDVkXFjkcdieHv0D3z2XPYw2CM4wCaotLTD5puJzLwp2uweCq7HtfPulksyjnFD
VoU0fM0FvH8rJaAoOMV-rnX4-MHcDwZmMK0RrmVZfFrJ0TJcR8Ar35PRAwh1F1Ur5
2XTtF74Fi2vhJ0uc3_p09AGrmobfd5zfkCtVXr0wM4dTvI87xVmFW1tgzMtG0Qzp
BEZ6P20oJ7zr9ZxfYknis5bV7k-cF4ZEw06fTtF178heHTZY13uAk2vnKdMGmpDs9
cc8u1CDc0A0g0wQyyIxRL2WVPGJCpSNlib8e9eAtXLffYAiQZ4bb1ZLIUoH-G2QBs
NL9tL8Gz85EB1fZU05TRF6X6pI8vgGg0cEQl0-9Lnwin0YLhr23iCwN0vQnh2WSnw
vgcMsUup-FlX00VrX1atcTBffJDCYNc3rZsTA2Qs3Qkl-JthYvZgAEIQXzxE_EMoS
rIRpRbckDFQnq998yCh3dbJfQA3rayL6ZSq0Nhuwf_SqL17N0ZCVPzPvGE3G0nSAA
yCd0Uor3NZyXOdULkv3Tpi440CD6n0atAV97Hq40nnpI-K4vIwt6xIfB0-Tidshxi
JgXwZQlwcKvzShL3WAUInFCPM4AF1UKqTuKdTQs1DXQuIoZZxFf4MkCt037zfdSR7
C_UQFbVdsGkEcRgpL2JE3WE8XTqN_LfBX4X0VspT-rJUXNKQDoFZChLFizNasAGyx
n0DzzR7m-LLwHZY3-rnpwpiUKKRNT3a7VQtCAsIQazmh2sqM1QyCuaDisN19_JIp1
32rMBYwvs6iMGVvPwyA6joKdUNBzMkLrIoK0e8MAEeCkk8DV-TNyYgNIEm8MRwA0N
YHJ17fu9xlCjGiyMt9_g27mCgY3oBimsdRb1KEjGSGT3WHDrcKJNGpM77cp6KqyDm
63oKN5qLlt1vCDS8Ni9jtRIZbUn0PIDmTtgj-VKdLvUarNFnisyIZhxoNy8emjpsz
ct3zCpLpQ8UU3rskSPBuJu02yQZimLiM9wGnhrlvADdxKY6k1VZIFeONv6U3676BZ
eMzVUADZsIceBAZIpPf-uqCxFnpjWf8QyN5tg5n99NIvsVflwQUYp5hnVqdUBFg2_
8jKwv3xJ9hLwnNigEvzrW4s3FuludlZkNdjs6i9sqeYvwSRyGZyjC-XtWhdALLsMU
O_C4UhrS0vmLCpzGEvIUx3aUYQU00vH_-4t72mQIeNpZFEo-SMUxWxc4AUpte2fV
Dpn30NVRNCIa5m7SyH08A92Yv5KusVeYYxUd98U5F-3YRh-9G_LFFbiyvc5-R3EIY
sEIqJnT9op7o2ClRo12Hn06D13Q1tv1Qsu6AMDiaLU4BboteV1dzLpyaxwk_cy7Am
lCuE2Jx6BQGH6n8oz-iGxXUNGY1-Q_kA0DtwvSL6yTEoL_KfmKNL2lSS1KEGeq-fQ
-L9_FW8UjXvizNWDBTByGEMtxM6hE-3jJBGMvbgkp3oBF_jr1GB2Jyb0QpsFRTGdk
2x1mx1G7WGD4c5-G1J4L_KpgazJ3D9FNJkKbCdKRxeHUTVgusv1LiXw1EN2pohyx
U7Cnn8cYymTUWa2iF4t1d4HBfiRdWwzjTQjxYgYJh14xUPcPBfttqyNlFW-_32m50
Zo6N8LauuQjmAk-YMc-EEbL8rNrKck5_aRV1icV2bpkPSR1EYAkF00ATbdVgGcnZE
bJQaJHUFQYkfzSimhSs83rcNHHIz6_3gMJfZADdRMFuq50fjrgttFehf00Cbo8glS
EAtpAnaLKEUL9xzFiJE55Bkunn35LvfhZc0VhXZhhJXaNASwMxMIC1JG-mT2nLFmy
lVBwHncEdiMa6m8MXdGPCFAD9k_ndE78vBzdcy3RvQB1kqrkDWupPmEyr4Y36jidn
rGmKeQTEQXB3h1m2N3X7nWW5hHNDSE22qZ_TK-_5zYnwtfRiDK6oQP0HX9jovPszG
35neiY40tiaK4Fac88te4haHretD-4lwnu1NWK20LiGKKVLiogoy8rW4mtxu1gp9b
zDV6ETbJ_QlswnIGpksy5lmyA9H80HrxXgzsv_Buf5vtE0y_hVDn5vmrL7wzV0jqU
L4Co4M_E-jMvGDB1cgfVu91yY3VVC-MFp1ashLuFUdrzrwiHPwnCKB6wC4BmNgrRM
UBvrV1gtevVXX1iKoZFHB0F4yk-LvRWejr7QNw6nsdjJiv9hN3vp0vAUsz4I_1M1T
LYKfB9q6C825XytaXQ33_K66ds--4RNNGGz5Ua7EgCtTJhJ03xr1V3fmSLH2IDPzh
2aMnWpJb7AHZeeH56lJQAbrf8S1KdHwp0VM0i-6IpLrS08g-5LJ201M1ybZv6izem

V70_785ejdrEby2CH_aYp_2JXHG5REbTTdi2-ckSMswwsiR8FZfhYoRjpQjbauxGI
4SXU-phA94wiUqRnS6VryquwLpj0na53E88ohyG6Dr_K9GeiXbd8vaL_zobZt1m2m
00V5J1pdx0HmUwOZOvVHPvDQSDedr4s831LDpsFUHJqkMZ2YTNGf3EsB8hFvqwaiB
Aqu-G5o19dod1ajRfcUkTemyqYpTE0q8hzhejYNWrt54f60vrK5GVqyoJIdnfXxcp
0cItIAuWh0dIyTtKJvd02zm9Xg0lgvCkd7GhMeeMUxPe6_0TdpHT2c89WC5LnV3V6
xhPS1eJjY660ZGu-9XCxNLntyWXWqESHNOTg7ZMDA3qWr09pk2gBka1Jozi3UsH8i
mnt4YKrLJ_jjnDtF1W7No2SuWpSYXnw2-7Yf1NtLluf-H195Ts82HY7QuCvDbCemW
GBFppe1hsKQg8Xii-506AiiY0-XDsPxijzRACm2P_A_SBsYanrgQwz2-E3jGnfnMm
DxTq0mWHWEc2k7Jm8fy3P8F_UWSL7rPr07VrEK6tS7UQ0UiSnbd_B_ZSjtV3a0jeS
T-t3v7irfzfVRkh1QiUIX-aDiBpNM1I99aT7iVfW1nmx_uQd0XGHqFPRGjTHxRqtd
ogXXhbK2fqoVBk73eaDZAR7J-g-B15CnTxnIAoZEB3azd_Dp3At2PMpBuFiGmAqHj
2DgoJx1RWrpPPpX-ndBx1m7YVL0RI7FfNPSdXtd3pr33gw2v_tciULLjaE8A8gpD
oGKtG2mvXnM2yDzwNQeZvtBmW0qEMnkXu3XbQjWwdptVFr13K78KYvnaY8o4viJk
D0eRQxGXq0JPD1u3g3uTjVkbkcv0mZXpBPhcNq91b5Cuvu7Sj0x5ANY8Wh41Y9nBF
JAl1GKdewyJkLZhWfBhGxEIimHFj32nJHYds2WJDmSAXAQ_F32YF6UT9ZAegeGQs
bLbX2ccyZLo-3RVDZ2_1UGZYbx3wEH4M9Xq0Lf6RtGFfgw1_3PLF3v1N17-wJ6jb
qt4CGrvIcEdXWsfw9PKcauH36081jFEQnh9khDoZ-1BWQnjDBFBEHQjwR1ivgwryS
Y8YCTSWUhiSBreJ6fAgal5J2k-PrjIa913qIu0InZ-irBfG2zMtpPuN5p647VpRT4
lM1-NM-SV31gzE3va4vE17T1nZX4joqG55ASER1gtZKN2oS40qiWvpr8dJ3b7IUPL
CYyArZEGXdAw38n68t7PyWvzFdWv8yUXQxiYHJRcF24R0GJeDBxjwkKTDJEzdapmI
YKhXaElgzD0isLRK805cQVwv1yvE4MY_cbnLMvcs4lVcQEiQqT6VuIrCiq3rrXvMA
C3woX-bhn1-ct3BZIBKjlx0_wat7hm3ZGyNVE_jcuLoihgXBZ1PMNQFJGQED3PVhd
xjQixPy8b5wy6IEX37uL1PwY7jk0qawfizmn-sH6Nou-6QY-tqQCTjFG5BXZlkzaK
hWKIaYd7CC965r_f1786rzLrLkY46Fz5wLS6qVbiKvcAGTbdkNhwAf8eVYwqbW004
2pgfQ8YYuTEJRZtULsVIZTaQTonW3dG_AwH_bUsRz3mrNPXImv220PZsoqcrnETea
2hx7gVSWN3f0SJNHjvi4Tls2QLm0_nuCvgIlz7aK--PFT8u6DqYgtAF-dnHF0J_22
9_oSBXPasJlno4ajosMs3nE6lpJidLH-r0EaLlVmQJU8rGYaH4oqu6etg5dAkoHdT
nLy_gNHGKArhPMQUQ8_eTg1GFw0SGZDwmMI-X50xyIJ00MVVO_0hr4ogphvZWunCG
2VB4FM_mn6VsmtiMrcy9o8swKi0kbS9hJqAC0dmIiDSCNougEReLmtxf-yRZnqxtl
u9qwURK4EKHq8m1M2JiN-H9-HzQjEj8g7jZziv75l8gdwDrMM3uBSqtarni0VFs7F
MRbg5YYofnZHY0VQH_zyH6wrRrjfpqgrU4yy7pAy3zFXraSHff1eHiBsikTzm9sBT
HZrjJ8iitwtfIFiqUIEIUm_HCCcGG_wwMSSl42RpAQd40y0bFNlPUquQWf96Hb93ld
MUJZHvSR0BvlVmdVdgcmbQpK1kYoxGf4MdupEwYXrNQN67uu6ry3WHu66bhZ6W4F
bLutQXKzVpyjCiyOP94LmeKLSe5r6zHUT4wRqEEVNY2o3wA4R1cKIAPcIknCRjDXK
sw6Wdko0-FWgWKKMEGBwW8PizFKujnwPN61_FGDgAS3pXiiGhd9NR3LQCKDYc1yGi
m6quL-DXu3BGRq9f1BqIL0w2Ibm_9XG_PaGspQPFCY6Y-5WjK3jODDzW2q-xnUw6Z
7MIphoX-GGICwQmybh47MqJlFVTsxnax0xi4q-ONBxKtmk2VRPE99KYZq3NkcHA2E
3F2Cfar2LZ7Hp648b8yBDG-GTuczVzSpMhPNnfYhYzCbjtXrni6DkVd-jQgy1rXW-
-Ti5S-SxrMWLGo1S9zf0G-T8o1lEjCx1s7miY2Asf3R8GpU98DfIibJBj57B9xLFy
latdfdR_oYilHCQlaKVowF5qm0Leua2vCH4YqQp08NeV4D1XdJsDmY1nq3XFSbyGe
wLQnJ4UFEvvRiMcCCp4qPwWnFqwrNmJwKo5X2fy0eQaMXQf5SY6gxNCZG5reHjAzM
uUrgs5czDZ54YAHUZDJ-1FWJAKY6g09X5byDGoISNKMzWqVOxCKUQJeIBrevK5fmp
ZwkZPpYOC-uL8hGw-fvyzGBQ_sksLgrB6wkPpbwReDN2xldx4WnzwA3CDY8IF6EzS
YzDrVc0LwLfx3iTksy6zh31cbvPZZEb_yPSMyUkC0bBLckrOTIizdRa6khtqTJmqI
AM7YmeQF81MsquZfxD68o-op0Z77GQxapVAD0jy_h6ZURpilVx-rUjW7ipeAjGplk
VRJvuSkSf2KNywiwbcS62lJi6LKLrB_3-_wvI3VreBaZ_P3sNySMYi1Sqm7mirBe
lAzIctYI4H9ajp0U17xLF4fLQFFj4YVlIrH90bUFCqIG80Mbw0L-C_89kjaT_Mj7J
AgrYhL_LHrVgE9LEW86H4f5n03v0lYw5eeunM-FGIjr46Kj0MdP9v2i6VlZFXm0jN
fPmep0VPXBEqK1YHmXZCRXFm_ufvsvMXuuJs3Yx790Vd0IN-vQ16Y0Ju3F2v_p8BV

UD-UQLhc1rCanwPZhtpkJp-Xdw0LRNRicPPkbnxFL7nkatsYigWoQZGcaw8T2Nv_
NQ3UAzc_HsUvtYFW8C-xkIk8fdwDzBcLD1yAKnv2dQ-tw-9UtH45tri9r9t-Q1Wfg
ugiHDEo69KNOLdNMdmjTSq3rNl00IDCiwHYvMQb6Ew9px7ALvBBT36sk8dJU1hby
f0-_IdXYDMD_GdSiaEWlylBpasMpe0bFtUIpPDn-MKX00LlBVkflUCF55y0fNC4zY
a29oey0BND3bCmvXXz7e1FZ_vy3s5wwYtIyYi460oU1QgMwaxVxRWUwQuH0VAhS4J
F2FDYYAZJWxbAfSphYlYt4NuRgA-wadDq0DI8tSh0ICdIy4vB-_qgtSGjUzQbHYdh
DzN-tMC50z0lQCoCj013f_045qAMnzE5Mw0BehatGLyaPEmq4mDXZl3zjitJoMeph
L7P-WX-ttlH0DdQEq9hhVtazd7Gd74oEZGTHsqJ6dBalpvIbdzzkWyb1D6zEvrY0-
S4Pf9f6yK959m8MzLj6seBcME1Wwsu5Z9P_fncjs_EBqLs7ndJ6nzTqXC69I153ED
L44kPXOT2k4FhU8T5IbWwiIzeE0WY1Z131ZrnWxHrxpn2pTlmeGSD-37hH3uJFFFQ
Z2yvo2oFdln6Pr7MNw40xNRNh08DwnToh6PewUMk1_SsHyGjePhIIyUm7eoPtacwG
_2isR8Kd_EMh22wJuPHmq0K6j-nh9FSt197E3Kmy7j3kB0Tctm0RJVAKznNy5ymp1
jai901f7UrYgumRU-GjR2lovJqP5ooF8FAE_oeT9oc9z58o2qzLPBQdqGoZRubbSL
Q5KAlKDXUWLZVBUPGiFSV4qTpH9LIYtjao3egQHTpfcPZVYEsFn52y82zFs3s2t_5
IquRVJxYXHAYR0JYz145WGOS0TQRUI4SE3UsoEdRZHLsW02WoRGDitx7u4GjvGgvz
R0ISFKGQHZDSw1m-ZzTawNDLhD9qqns0kswNG7cQA3MUh1hgo_uHKhdKhkiR9pvr1
fuF-Ik8kSvQyYcqBMP4MFXQkadhYbFXRpMMxC09ShqwLyQgkFGbiVc2RwKoz2XA7F
hhNBK5PI7F4_z0sTcGYRFMYsuvL_uv9yCdweXblPpf5JkyTwRaPClYR_jJNeIRJ9a
iYEvXbvQDigTPQMvC3rqu8JIGaB1G008wg0IHK-5--GOUKi1aeawbZ0mgFpqmFJp0
2W2L7kleUKBNT1LlHFVHOHZYNNavKfgRf63u01lI_hs0Npm4QED_Og6xhq6jHqp7a1
0WdCSbzDpBpqzSpj5BNx09nubaqVT8L81JUQRpxvMcVoXKwf8nFlI20zzad7yGnav
Yye9eFJH-72gBCSZaXa-cJAx2Q6JTMbMeuFoCB_a9L0ENTzJG-aklTd3yiKILQpKn
PCmCQm52LzKKdGsbT6eqdnAFYiDbWkoHV5f8KqfrdSwB3RmvuvQiW3W2j0Ycdbctz
GxImGXCRNrteBbLCpr0k5pVxo00dF_AFPhtL7t_Wg-xhjfg-u_Wy3uY5CnUGc1u7X
nNEg00hhnc_8L2icwlIb_enrluwV2MgCEGP5IHxmdkILlLFatBWuZ7L05Gp4F5Nxy
sBPf_ESk5zyVMT9R01sXdjj8mfl5HKcyM3_orH5VPi25MBHAX9m9HW6lmKP6oz13E
rhJEFW0TKLRwg4rAfKypXMATR04ve9BcvIESoXws3FI8pCy0zaaiEaH4kBU8at4fy
uV4Ph7Guydy6brlZS0j0GGhNQ2LL1Z1SfN7Ye-q_MC9svxsXwcXQXswQQsWPkBYa
0TTCX0QYU3Jk30i8xAdCmTtssqEKYSe9Sm_axauAQlISZDPsT53h1iikbPRE7SCT
ULWA5S2M9X_IAoZwj3NVTR-nD-A6nt0-h0dhhaZRth80X_Z09bXW8hp_-B0aF3-dM
ArCboXMa2e0x3R11sQgnvRsQNeet6mJutjdeE17An0j-cAUze5QEJc8EEHYWQyVnD
ZSA4kGkC4hxdd10CgKsg4hKEV8y8AyNYSHrKhrubPKLqeYg4-ye2niK57PH25uqir
p6nnhLusFwnCZKNAWKgqWq5Mg05YUCbvTrJi9u5dy2Bv7mcGUdCIJyX8P9Fzcdj20
-XcdeoMMOygg0hWb29wqBnQtMzRHSFiIv9yMwxiNsJowuoqMTWskJosDy-vKul3Et
mNjPskEeyyo_MdMq17dJFE3kRlRluWypBFsa9FjzDwlAn7ztGkwyZpYACrwPkry7e
dAmUQgyfAL7G68j8faRrVGHKsboal5ipZVX5pD0Amis7hUa5BEKvPEYqakm3T3860
a_X3UKd9zPNgKEqghqM4SMm8GhTur3rP3UaEgoIVj98JBz7Np2n3coAnUxertKFcv
JZqadvj7_BTsvdKK4eEY4im5h1KJapIks0bFHzK9Pgt8cAzJOEYy2UCa4xvLY3VGk
WYioBrNpdIWWCdMTGJrm4Q_bMWNbn_bcQL6osNxjLCs4Ra21yR6opbwZj4hiZQ0m4
ZjJQs92prkkNUR_dyjttHPCzaqM0hS2oBYZFa0vI3xhkgrYI8EIHxAiUaPvSy2g7i
wfeF8fcUrTvA7nRLhJwsokj5GnDyZ4rK7a91zW4oaCvN1yLcFXvoJiXHP0AS5vB3H
1q5nqQkMMEP2dnx0laNGkbNxcwfsU_D1jpJu92FL5N5wufAPHrD8gKjBOBRJAKj9-
3D_CWIExFq8CKM8bzWAUketQEBuML_6hLKLNECo8CD_0a6yoxlF6E_0hG8oI3PT2Z
vh3T3PznblJzquKV_h_nkDFHNJctljLcttctKtj6KAlYrrI-nZHiXj3siK0N07t0uJ
CorPLhy8XTMxawSzQ00HLxjvK0hdg7XyATii39RNocazHl8kt9dtwWhkZrWFAYFGW
Hb6_pwfWoziHPOQg_HLT-5l_xbM_4xVTFvfT4IRs8sQReM56Qn-jjb6xgFZu-6Wus
2sVX5Y0xkBma_eQj-D5iR2KMyft5XX_JETv-9tiGKamy5-I2RNK9DICNzYMysj99p
WS4dduKlR9bd4c0zmHhODDo7iGxa94nliWSuNJoQhM6FghG1hYKuzRbv0JIF0WNgE
ZLBJob8zoKh5K8zMuW_8ThFTjBerIEGCR46t0aFmSau4cdzo5y0ojci8zNN_XYvTD

CMoB5jUXJQTi2zHCgv7X0V7Zk2ZK8vzVCq705GiHJY0VsV2EEg7e7MYHEczt23Ek0
ar7uJV9Lo5XpeAT4tRjKETupu-EVePJEdrzDP5rZehRlfJ3GhmPiJFtEzkrXKhzcC
c7PvU5BaQgoEtmMs36Kfrpxc4qj6x1UiJc_4yuA-8AQFmynmyN1D9LCFbY08Ds0J6
zxQa1hGZdRsDbhaqQqMozUov0d8UtI1IJAM1SsIHXseZpaDJ90ke80g9Mxosn1cCC
eb3Jn66Tuwe4zeMBJhPloIYL0Cx-90LjIReXNtEc2RxTaQnUmIrRu6-RWJcH5_xjM
7zK-rkqCY1rPpZrJdQzWpaCKjYpdzZ00qhUBAzs2M63GjS19rwuyNSlaBYD6K0gRB
KOCv3g5-SsLlmp-YKr4o1IVU0QPIntD0NcYnvCgVj1Bv4V6noAxI0rdzMKy5YJlgN
38HyhpUHCg-RNM4nMshBLUGS9I3Eq7EbucH3Y17T0pGPryfCALnyL31DAzGc92ApI
l-ByI1M2ZgIivCUIdDxwaSJSfgdUweELFx6ZnIdz3ZfctxPs5Enh-BBva2KrI59Sn
jjafNHosP-XFMM6lh3RmvtfuPT0Ijxnsab7p9ybEiCDuF1ysG98vfk4uTgyCN4AvL
sXlSFLUIkqwl2INBUqWJEP2KyZ0Xrp1MQXjkJ4E1PVI1jZ9Es0LL00_0uCuPkhx9c
f90Y6rpZRMvtTTuA5S14w2Ukp6QBkVI8q33BqWmhBYw4rtf5fWGDWRCy4_A1CVAPA
W0x5bgecf1HFQujVtzczrIJC0MqbJI2Psm0K8EyX7hi6sILpAcWqHE3n_LXekkdE-
OHgkSahJLrz9LvPkwgujTb4QFF9RSjClKvHVhtdQPRIlUmpPZ-mAbS99v_ddKY6ED
c_ZC9VFuVtNsbyv1EXEzdUVVmXx0MDw4zHRDrXTaGUN7SbbxYYxRS_nrjk49ZpJ3D
CWQ4ciyFFRXjkHWNbuWXQiVHM5pDBd7y93A6QsUQuc6ZhogLpGAXy_SQU49JtLFSg
REVS5Ln_J1RLA2AAzIUaKNRKgcIVgFiBhtmj330UN0qxXb7tF43Czo5WhoFTjRfPX
VySnHyClLnnaWkruZbtFzPDs3j8iti55s0xaYpS-DTuH_7vwn2N7AX8SuuLL7DSEs
Nkux3o862Sk3xWSanWuI02V0_YnkW0mvLwwlqUg9bD3oVxd8KWEciVCI-MRDZ0u3L
WSKXC2tptETRe-0evg0txQ7AN8pr-6Pc5ASY-kBQViCcszIdS358_pEUgzAntiEs4
jzmsYUwUrfQsGP6Wg7Y7K0pdK56UXxQQ0vKulzGD0EK39T0GogXu3rY_5lNwi2tpu
6Yv7NHninziVhxBo-gAyQHqBuZgxNpV_7fH_JjW3aC6cxtigBMpW9orcG5PnuDUyy
mFI6KQUYUqQbC4cSFU8KxC98x1F8aPt0xEPhasDE2jmmKlL7NPQkN1dRG026h9ZUY
vaVexhl4zTca9b5mI6S3_AtdLz12H_5-tm_3m0Kca_LxE4gzQjNkb-A4inLR4RWpA
0Dek6-X32cTP5_8RcsH1NCfjv5R0c-17a2TaBAczfuIWIJZurNmuglD01a1ZwYVT6
0rr6bU1N__gCkpaduRppzkQD5ED4TB7_0PcFYCu7l-qZVERsA91ZD2IYyKrKmjGpX
n-ktC1S-FY0bif7i0ixm_aKdX0AD1kL10-nyn-bcb-pFuyj9u4iemuQHm13T4k_TX
fSVh2wGp6Bwi4CUzlo_Cy_mA2nERQfPh5FST0nfQUH0A8L28bv87XXU7rPelh0ZIn
W0XRS1Bh-y8SvvdKohTH5f4EFCg575K73iDTSa-Z_T-Nz1D0zPK5mQPGULfnJafm1
AWZiaXdn8bOpK02iNO_iWUi6Q0h1q2VN0nMgs2NYW3ieKk6gw9X-7E5CNxa40GbBk
snsWjdNLCELYUA7K-L4rNpGQHvrrMcmYeAe2dRyS3NykupS9fLNZuCr8QsiwnmYPM
yivq50W3jTYpIjbx2iJHOA10yIMQsUrvQrTN7jEnhkaQMumsZnm53Jc51Udt976Nv
RdArZKSd3elJ5ceVAIDI1Nxs8Nsm9n4k3VtwvnaDBphcJk37djWWA4MyFpx0Dv5ee
D-8uAPs3k3187x3LCP-gpeuXXLaCinb2BNvueJAvFkYG44arX42Xi4a7Adtq9zu26
KVt4jfYz0dTz_PbIEJM-zdyaBLkVcAfU7g_UqVo120BVblAKRoA8ZC6NE14823Xp6
-ch-B0B50L-qRo5Ngi51qoXT5SKjwuykt3PYPu51ocpB08UI8w_-lrUfMsRi7tjTG
byUz3u08dJhihSMAVQlhoPmP6i-_8gkl4jL4d0EA0ImFxU-W6iq-a5EN2t7mifuIH
D_2pX234jCmANNYqIlzKo_wzxJItaULf0006Y1eC_5a_vU2DfY4d8E-Uo7eYDCLQw
et-oxcGsUJEuVyjrU0VSeeTS0TdbCx6-7e9u_-n4P01xIwnZmKlF5kAo4jrg9lX1T
Sbfyq-vwW0uE35eTS2XEPAg-k6_4_CL-n11Pq8crBcDnda0fokp9qRI17DePhh41G
hK9-f-qA9jg2oixToVxKzwcexxcC0vQYBd4fJhD0laLD_1uFZuxckvjUy9EiffmCm
45K_WtSJZ1P0qgx-Wn2pPHPLLU_CY_NDBaHHdyYZwwi13yntEN8-p0J_1lvjyM4j
nb_hDNFJHeZaCigzwxBG0WLjBkqwdt9XALil87XrXff3CwGBLq_yteYIe4oEIXUrx
Rm_58GEuZJ-qSC4QPZyKZ0aAZsw5e3BkvQpzdrP3yP3wc-bWDhpwjfwTMz0Ck3k5o
ARWH_8iSEV6oYi_pT6lyOMyABg4Rp_IzYyK0kATghyXmjWr19E56b132wJg207Aa6
IaMFggaagQ6LiD64nfmouYQPOC2CQtIZQI180Yfd9zZL9_zvAc9L1V6BTsd6IDpx1
bs67SzUznVv9981fgu0-uW55gK1m0vZFz_C4yd_IAKFWP_9VF90Tn7dJTzYVX-pkB
rPyli1lg6iyM236GAFGoa0cvg1u6rRLZ_W0_I2T4idyztDZp_QmCFsQ5kMdEsm1Xxw
ugU09uh8v2NPXy8jUbXwCZ1RGIan8d8QS91dXpHIcrD2mGcB1CggX-ujrj1Rw04Pq

A8ssdnbBpIkW60Q9D320jB5M1ytqGvHdRkKwAv8qgwV6EpyVsd90Bj24H6wAoH0Z8V
795lPw5bIzgsKuKmkbzEjtS5H0ltxqhaIvLRvaIYyDDdcCQHhKS0IwjwRw5QUnpat
M6z - AhpWGeT3VDvqmDcXD0dVummwB_sc4k5UddrgTKGQ3I1jar2D0H5foB4fk - aBT
VpCkRw2fbdUvE2Stu8mdcgq7k4w9f0g3VWTOdglG_0auhijOpw7EkKX5l8l1N - 9lE
Bq0siez6TR9gWhquG8xvLZHCPKN0hVhvBw9jW_YX2jzaUgEsmoSJJcF0onM7Z9Dce
MWLRfDgmYk1h5TctOu7mqoITK5FSgP8FkUb040 - 49W99HakCgRxtKj - R3hT2Exyxx
aGFBbc2bD6RqhF6R4tWjxMwM3kSE8rV3saJUR6DKY19kXTmD1AHThs3a - 1sE1JveD
RzkzOWDhOop1TwSeTxoiqlU5VYGYSPPvgpP - l5MUMjkDLh9_4EGs8B - Wm7keroEuVS
NzAmaRL6BY2HaJIPeH5HiDFG - iui5 - zfBI2Drq - l1Sfytp_EqMKY24uVsqx6bAstH
YYfjvAhmLf7SEbGgAFMl2ezIIN7 - N58kybi - Z4IoUtCHrtVhm - SXancjjxYsYmys5
VXPfdj_KtHgildtUBdcnuXzrgsvfSXNx2KhZABn22yqBwefW0BBNmbi0s0_1qK - OE
mZAJGeq09Kzxq7WpeuAIWVv597yUGEXqPjN80w0mXg_3jT2Wp3K8scZiiv3fWzTuc
aw5v76paDewDPTVNjDk9IHDYEjBgwExZXhdHqQyvufGbwdKEWxzE0pfK2thHfDmSP
7apEwqZrJ - 0szRR7P0MHDZAJj3qqS2hNQu_Owz_9oc - 6pENXkQUg77HIj1TN13jPa
z_E4fc0fC8Gbcxycf0hhG - mZa2p33ld - Birz8hqF2iyspysRw_L_Kgx7POTYwT4o2
OmM - bHwPeiXayBm - H4Tv_iJpAhRVTIQ1Cgt8QbxA9kApyPg4LWRxcCe5vCBYD_2PT
nooHNDIbc9wlKpqxNFx6i7SczlCawCk5gBFakXiBiA5ZtwVX0bPG3a82zKWHe - nxc
F7BxfbDdiY0D0S9YBztYyfNTsR8cFASiVYNb - 5f6vzjQn7CV_M - GDC6snSRJsUvKB
OswpJ3JZMJNaXvXZtw3Mzix - pr0Wfw1T3wqBiAJcSwm57hgySotR1e4KDNJdmoQw0
JNZ7KBLvV - lJF9acwvk4We0BxXez8TnCOt1eIyptQqG7sJpFrlox6pS9nWFIARwk
zA47udjY82050vtupxBvQt - qkzS0u - Vm2X - ETG3tjPke5cuYi4IHmpCNmqVHvNdwW
j2Yx3Zi6imLoxYmI69hm1bw6D61nRA6 - TnzTfH9BsDx_ - 7p2 - 06Q_yxwCJbEvwBEO
JvU - XGu1b9C0XJQLV4hgT - ld160zUN0cYMAUWAvYoNKzqt0 - NMzVk2j3GR_VKA23x
A_iE9P9fnhzF_JazqeuCC_mX - 03Zt8qRDYUdi3ht0WFrC3qxloD7Hc - _9iQqU00w_
FvtmYls8qreZwzke6kA9 - 6EycwdTaAHFMVRcL9UnX3KPPP4IGXPuEpaWGxEPxdkdo
_Mdr9j3XLg0e5cj016H0KPr9s50DbAyzzuYK4JyS1pUEurYp3cGf3sHLkdN - pwoig
fTkqgCIRNtv3ivusjQ2AJtGSLeMdooadQZJUJZtSxmZXZ7hY - pHwQCvvH6KzJQWsi
czhy0zEREkt03qz7ea8V9UjW9wUisQGTczNSr10NNgI2BQf34KtGJmh8cuZ3XQe3a
2gWH8u3pk3cgqYk1fCn1Q2kZM5GpMFqyZwv0hak071fza - dzQtSupBoItL0YuDlRb
Lid03Y2kFkB10t3448hdArKJDEL8D7Alb2ljh6ZDzg051GaCdoPPMBmyF4WJCaP7g
22dNqMowYDrjHdF2hA_adeTswg4tZimUnTRL4j8BwdA3lAiHFpAut - gCk2i6e_RKQ
p - 6Umi3pwi6hLoU2V911jopD59AuKQtjm9309RpR90adrp0mfYsiGm1kn6BBLojDv
RxouNaaEZZXIDkGyk97ryHo2Dhh3ENHG6HSTtMIOV6gTTS6FgAzzC_rixtpkRyj1
oPtkdkurpfRwW40r1l3xwAwZ80p1 - csUnUJnCR5b00hgg9_stkU10QzsLzYgYS7uH
iRg5mDjHDS3X - tJ3FndPMT7W5_SrcsYMYJHYRymodF - cVn6E - 94B5ynUUpaj8HCRQ
U3uvQg2xMnKmZ9ExvWwNucbTwU80P2uJEOdCtCc6c8FWifwBYcik90bmYEGVs2YT
9zAI5PqiMvbdAT3506zvXneKIOfeAfaorexf30BxITSqNGrFn39ZdZSCbuqZAZmnq
B02weQADlxfflsuJ3hycP8umq52IS9756t32l2baMA49mJC3wriCb4byFlpaVq5cb
tCZRPV1Uss0k8QJdUbg5Myita30Mg2JvpRrpi2J7NPEjWwSkDUEQMXwMcKhlfkyb6
qW7WcK0c4lr0g3yHhuAGG5xYIq_Ej2xh947CwCTdG8X86MiL4Yx4 - isn9y2TNhLpv
Nd - B8tZ - imYRwsOsXBSzFVvcTtnUMSF1sNP6 - h2RWn76Z - iBNNuofiH5Rgu_OnjXM
PwLZD3Hwsdidp0V75u_stzw6WE1KQ1drhQpj1JBqp59iRe5kN9KtIj9UFHEcVltRD
dvXov0sn47la1fWIK - 53Sd - LWZB1DV8T28lxw7LEYmb20B9j2nSKTY6bJax7ezdm2
vsUmQw07qVBlawZpOneYjvz0NLPCqUc0PKWlwpBlXs8RBYdbVmLxZ3 - 0J_1VUtHs
wCzVouz_jI9ldpR65jDqK3v32DDRZzM - XIgF01Vb4IsYkcKijEarjk1hMPjmHSmGm
NxxePVwvtQBoeUzfA8bfotSYIZPhVQmUTGAGFppyCf6gXILPA0lxbVRe06TTnVwre
kBQ5who1YyNJAYRgDzt7gYDhzFR2EjAlZon3rD_B7b4lT - 5W - WpqQsP8h - Nq_Y_x2
EwyV1JzjEda04RZQVJgmGsQdF7gIuVc9pQx5_RvXVhGta2Dqn1bf9DcqHm1r6jdJ2
Eroga - e3rj1MNfxbQs5q6pJ8b0EfHPKtjSP020QXY9Jh4INahe31pLa5LDKD0CkD6

ABN0LQFGhB4RWJU9-vqY2CgYaihF1V-7wn_g_r3TXM4qSQ3x6ra3K_rNTiDx__sp-
Fq4KitNqqe3wEQ9rf57_sbkG65bKBuHbjpIMemAzRGybyFVcrd4iVHuEktC0sWC00
73Ye0jyCbf1D0l8mCbgNjP0331Lqj-h-vP6Bnk81xEv1DS7-rxBcx1CS0dyn9g3yH
qpJVb00DoN2r8Ha99Zoqm1RX-ghuXKoCk5WLN64rOPddT0Phw88yihCfwrCxQDLrF
MWwxZoGDIu76DQK0IQSi5I-P6cT0_a1QkAaPYEs5sJWjy1dg4lCjzqZm4XwPjbuGG
hv2Q4gRuaZEjC2K7qIYEy1P9lN5hBn0ygb-z5ugdmbwXmBRbySxUhCgAETahk-DD
jWBHgzz81dueke12tvEnvc338fHqzd7N4lJdrLq94WlKF12y-mQkOX3vG1yiTQ4qS
tNdF4Q5fztHkqBKH7XQp9SHjJl_XUymA70trmm7GXJXMIeOPU2z4rPqKirqGGuCkd
XN1-ONv6ccb5eP2mrC81nld9MFiQkKwz_pGGrpALYr10tIpqmqb6yJiayZux0Jvigl
YWe7cYbWV4DhLQrJoM5HrORSBiybLxoG5d0gT59luFhkMR1gzs1eZXvTTx6xW0YF8
0xP5QTC47uqP0fEEQVNElABEeqNBk1GtZ77gUAQdLX0xHc60lKQ7aY-V4v_iQ8e7v
oN4FI_02gnUbMLqYLndAZqhR8De_PXk2XWOCQMqtx4r9c1-LvTCWsonN_5jvbk-uL
s-hcl90BtMIOWeikJCU0VCJfC58mH5RSgpVP0DTJE1TpFzrq23bA8q721ub1TequL
oJK0dFb0k6XEL30JelKXPGk4kuyhelSghGdgaa_knZ69E5ysKRurw2NDN0PCJs0lI
HRN5k-R7t32ykbi5bBfsJuXtWbDdrImgbL-J6Y0Zk2yHzjAbfFqrbqE1Phly2FBjy
U6UGrkr_xzE0FKbmCSvb16N6NUTf_tJ_M8hDx3w40I8njHo07uyzq0sRf-j4REEw9
gXawrKzPBYyq-RnritGh1lPUFh18i9cE321Tl9lTl3PYrWYcdAwrbTH8wieypVoX5
LzPQ4c_sp_6-U-lhDAEEBV7C6yMLKeLVtX9sBvWE17PDXwbEh66eTLzg5Mv_Hr6CJ
K0e90wTyp3bl5vIQsxedT2objjAM6d9H2Ny0TVPSeCVpDtDPZa9a2NZnfKHhuisLC
EhYxn2LQudzBlEzV1vTS-ajSeQclhna0_KeHugQl8C0tsL7A7AtBm3uw5sQjw2X8E
XhhVz0jxFwj2WRnYF8qqiTv2i6BFY1U_6hQCi4-711j_K8h84Fgxv6UWQKtqh1FWa
5oe-2Az6eJb4iNWR-S5RuRlSFaxSxLamT0v317CFMZ2E1AGXITfeifDL8I-jPYPoy
bED4QpuPF12kTA5MyvocDrRpXVhIC9z6-v7N72knS8s0W7UTM6Zj4Ii6dHpJ4NjSm
N66MGp81uyPP1RU4R4CjkPPoaF5j8JpB3jA_iyI6Vru6NJKNPB70yqn1V8sbrM19k
y10Vn-Ro_ODcLCeIRgzkSQ3kjTevbHiHc5RrufK36ZxU2QpmTX8-PWCcHpx_3cIdl
-SWtjfQ5RIQndpz36Ho0S1XgQSmjsa2TDkTz4ICivGJN0DAwnYghWdTXzbTkG7gRz
e2x8MlcGNvkk1t-r8TaWmeA0s7gmwYrQZYEVhuRRMxQty-w9_wTchoagKZt15u7gm
IqaL1086xYTRfJ2pZipTNHbA2ZtNEuuDnM5Wp8t4NPTSnhlLBw8bVwcZmLQSAapwa
VkhbQC9V-EMmIucJgKEpG_T6cFogE3mCiwfHM2NZtiEsIaKkUnCgZMurS-gN3jlZY
uxTAcIaT8gpxIZK-j0fZ9lqsmTC9EsM4FR6w-8hsCLueBW-acpa9TC104AIUjYFzz
NdhGrh00630uTR6207JzglQTF67pZy-1z-AYxA7xPdNCX297dXqoo5HYFD2t0T-xA
wkCm-xCk5JLf30E4502TONfmqRX0t7iJ9h7oY-ejzq7byB8Mw2R56Y4esasg7k-h0
CHcrbLT8IN9JvljSti0oKfrnImJwlj8iv_wAdyGrMS6-H_xGh1rJmqI00Xl0pDuVd
Mv2lHxmIrvTf0qSbDPb1002S0knx8uljjPu2ZqaEi2oy2NtQogxnTKImSWZS-v_B7
U_r0nioTdMgOf9vgF0ZdSaPp1TjyVBqW97oHeAno-2EUuRTBf5SFVtyQdJF0DvLdG
2eN7DktRwzI4Dm1GFwtXrIcLmh8EBu3CjbE6_KLhC2w0Fdc110minR-Kx6pV5aCzq
pWj5NCrZm2V9MY9Cstw0cDNGkMp7QhrQD_q0MI9l8nqAYx6a2Q-ALJWxfo068ae5-
YxIe9S0kmj_q5_XaBlWxlIXy9g8lYuyd_0X0AeyZEyhUub-2ePkpY3ES3RiEh9DCM
bGAE98deGm4Z2Y2xQRN8swGxgLIhpSDwlMDSztFU0XHeYrRiu0FpnSsBz9fus3E6K
Y0HjLLZesKAdt54MK01pfMUDQXmq6uxHG03L3fb_E9mPTpae0l4zFNFaEpBktZm0S
23lSGbgA0W0BEewBUKkPUoZ5X76GnWu17S0j9jpgIw01iNFM52BPxQ_okg2aFx3jY
UG4AnehcHPu3mk6IwnUY0l-0mfEvZoLNUAHJF996Z5IKn8waBhAqV8UoF1FhVL_iA
ll94xp75wqaP-pffhKMLJl0Gj9FYZro75neIk6mEZvew0YSphrfEUZI0ZcdPwkQHq
yfeya0yRxEXy36hGr2M6dFYWUaVJWXdKu26yDXVEViZyMEzT_scplDeGytX4i44S
z8uR2CiC18_MjnedM5F1LC7B-F6mE9yf77wdWMLcTnvDbboX2ZDgPZi5GC2WbNAoc
IPIp8IWvcWcf4BhFRuRqHJn5osJ-XSVG0fChle5wZH-Zn56aRroe8S4z1Bkv5tjmr
-IXpCvdW05VyXCudjc6qUbVMDSha0WogDlPrTSFbc5exmr0V-_oEu6dZMMmIn0mPq
SQtkFzc-yHsk059WPQWJadWHK1GBC4yjqBMAyVfhTWntv-qAvMA_143jld_ybrXyo
NEb4tY_RnOLyNsc4BptXwLbKlhlSppgiXlQb52N4AcaoHT_3QVyi6jwrd5HyBhkQC

YnF7YGoy3TqA3RyAbItFXhK6LvZx6WIl1eKc6WAnwVMLJIP35jAo1ECr95K0sRIbt
DStpQhJMxn7HZrEc3CyzejBx2gkifsrA96NJ7eeW1gDDCg_QcvQozcKhSValB16z5
jtHeTsVMbsPk73Z_u2Rxyg05mc3vD1hTNWaJCYNGK1jY4U4oIZK0RpmU0jBuqS01
RDwR_Pa0Nwhv9-OCymYZ4kYr6_6z4FDRiss-g1DeMvKJir-ljHhU3MRY1czXGMMM6
FRJjxiIkRwLRhyf1cUYN2mZyaMq1A38GkKANhXRqE_FHbMzoq8DE-xiQBmSJUqb5-
3Uux_0eiMA_bz0HvROMTfEUqwWbnZer5NvDhmd1VVv0dH_1xFONcJHkvYU1-XC8sk
Kmk8E_fs2y9QQVcnNNL4t6U0CbfiaUpX5SDKSM794YhfRy1Sdr6QUHr5PZX0GKtJS
CKPYrP0LhRT9rghe0hVVHwn0XB2ydYJa6L7xHeSy-yx3vS-8Sk2FfNgdakS8vukQK
y9TV8XhN1ggwA144dGueprz-eqY89G-wIh6oa_-1tGF1s4eoh9EMDqhK46B12Q3Cw
58cdvBsiH_P2I3Lj2dsD719st0duskd0FDpfEcK3NX8sIG95ZopptHZhQCCyxHVXa
AQt0oLpA8T7NP6aNoPMGcQYw2_WGwGpGyMan4rqF-13zBnWpo3hYY547WXnRoaJuo
rYmoLeg25ZYhcTCN1C4U6rZu_W6lM3Rg_p3DwI4bSUzDRMLRJHW3YrLE4CFM8BwQA
3tdv4CZpIZQjeb51EGHg9Whwcf0_nBurdN8KZCJSI6-F1iL6j8GMr7kD_FN0TKqc8
mBKWBklfwY4iAv8xxVcp0iEhyx-v9Iq2CumiXclG0GSRbtzVf2pXZDUomkaMSv4n8
SlrCZyvntuXJwi9jGEn8PpJtV84rZ_2hIt5g1xxtynwe_4v5T6DYhuChMg9S1Fo1T
TTdnG1Wa78Irh7v5LsqE5CaLpGSeuopb001VjHPmtLxnuE5QFX5mTVp3YYyAwQ_W-
gan8AC9pmanTmPrxIHEu28fk2rSkWA53hRyvrgPENaJeioS0ec04iFcIlJFeU2S6Q
pZqze2em0w2yruTSM1dEdGViUezcPQ5Bc4x7XuPJLIywXVLW3mOWlpdjkM7c0lGIv
Pp_uPU2Byv1wnSxgd8xhqqT-ZMccGF1G4jAqDH5wJ0NW_wdEIjHMmEpQ1Kd7ULTN1
oybhZ1Xx-JJgk0pbgF1ebs_LYUjq_nfgZEUzoEaNqRs3I6qTa6_KSAE0YyPMKyq1u
rjKLYwxkHdcFMDXKYCWqjrccaKHaT3MrYd1t6hTy6JPoAB_b08-_4KGg5yRf5opPU
xzZ_mHeL2Fv8f0LSYOnYRWy40xw8Ep6eoNtnFMu7ECnJ_MtBoXTv4UEON_5ZORGKB
4QEqtMh2LwKN474u84ohHArgRLxFDCah_DESxG4xiNIgpT4x4408dE0-gdE6wp_GS
kzrRCXgVi4yr1GNZLDo5roCEj38Mh5QxhgVPi3x13tDYjCtjUPpcgE8N0nFKr5fi7
BuPManZrg7qaGT8KQAb5vSC27J0mHMePXvKkbqNx18tyqi0GBVnffzkLqJcS6IQwy
mefi0v3rg9bz35bV60hIB63Cwba2VDBceIrAgpul1G8Bz0AL0dGD_Hj3RIednYhe
DNxlzUrLJ-Diz8aBsuszyt7J6No1LAsswuXZuDpyKNchhrLljgpDmDogm-qn7_qXz
6vKZVvW8HitLDLTVmF3_l0gFiN00C9vF2P8DMvlwdbW0tlq090poZw1XFoA-lGjY7
gp_WAztAxykHwbKwmoWbpfrFu8Ii4kuLhAzzI0tk_CrELcU8qjAB_UXgdEJAkeZxi
nvCqQ0E1ri-7Hcuui0VKidKQXBaEpg00DiEKZwmVe4YrG7uzdVPX0BxIabNnx2goJ
3wT9_DybwAjpg0AtaYtRfQlJrfsNVUy45LVn_WJzRw-CvSJPdx0deIrwANNRM93L_0
1kWPDrp_Ym1ZMpNU8RvGVWm_f_rJLxNfpYFJ4ZE5Ydmf9QQ8jXwvsA6IlK058jZwf
SPNQjXnrZ0zcV8wLZTCqh7RD_BQeZK9yE1e0xup_3fTp1u9hwsQFSTH4dQ19oH6Ps
NLfdI91BnRdpa0Zbv4No0WAMTQdrF_sRxRwBJ3xod2dMCCnBtWdgopkRtLTnH4ndP
k7tzjnE384ZN2iQVBgFlApweK56Hzo-bjHFU-YjA2wdMX-uAIX63MM8k0y8j8H9IA
UhZAL4wog10X_cRb50nCL_7vKEjFKbUr-RL1kiANwGyA40Zk5zy_J5nmc6_nnacFg
aFN2StnzIt13IAGM84tct0bu43go1_rpmWCUSnMgLcvd0sLAroYUwUKWEkC3UhopS
M9YpFtxWgcWjxYW2rYsoV7A0gDAwCF01b-ZTjD-K00d56a-PC0Gu2s-2nKl-jY5q6
hNW2AHDTP1ewydGUQfNn9LKD0ihVwMlmiBDkwBwlitMKPalsfUAg9lCYd5pMYo55
58lFdJMWJ4TScr2pUvTiYgVBT05sPz4ILOsCYn0eEIMPFRfiTSnlar01cRKv204EW
qY3BY8DjftDp9zuAH-eBdoQWy1tx1ilZwbe1ysLaR4Nx1DuSA8cCSA85xDpAknHhZ
4Dop4Pt_h0LoYN6WxzJo9NcGxVjgrkfnZVfEQTjPlip2kcsv8fi14V0fkuSBBEJP
ClPrEfAwZRLcadElMGvZFTrfRc7ncDXRfGYcq801Xsuqo10Jn0fklxC_GXW1eudy1
2scazCBUD_DmSm2eVbMw7fbWojBhd1rmjtLDNZftTjXBdjHblUBbrh3cNahb9W1oP
QPU-zT1LOYw5-cf1PVaV6_RLFGd-ZYlcvu82S7WZgTTgMJi24xuaHUBTWRFrME5Xq
R5ISpsAc0VmKRHlM0BgxL4BZR0dRGbcTOD-bZ1fgpj1Q7ctTnt4KVh4Nep7Jit3pe
k-0xKzSKS5BZoVuz6IekbWC_89d_N71jkY1JqnaFMQpL7DU9tkRlHzKhx2RNozrbB
SDtmtqKGEsNIubTG3xxqQPwz--Po4HoIn4hMR4YQY71Ps-wU3MRbptyLzL2-C67BI
pjACGMPrvhUhDm8H6YH1hTxp5ZqEtpj-0Yn1mhVfIjE6pmRHRdcdqX6bICX-z6c_4

```
XSunM-46Mc1kt6M1ebP7Rpwj7chpnn2BMteyq-gFiyockWbciHlYaRpdhvuD_2M0
fNUYiJ8JPw--0YBj3eqIxQd980S1-U6tACkIQXLqlbIMW6tnHqP_Fdf82pyjS19B
QLOf-65Y345n-beHTmIeJVImheboaVmyt-mkG1QIElHN10h1Tmusg-NKaYU1L0kAz
5b5Izf5gdp-7YmeHXbNGpxKVDBs1UL0vadY3TboXPbbJIpnX2UNn1C8Bj9hAmG900
bRo5qYBUaGvita111DZ5tyifgj4aPGWoQpRHqXhgoa6T10SmV02t4hgHHHcKC3u0-
Lrvp0yJLz0aIrCxtroiEsEyc8F8UaMgq0H0XzilWlUFA5ShCGmSKFL8NPY207Ci5
kJMeqyJT0ndVT6XQTkHNwvrbV3sTFu75ik7VUcS9cWShzYsgIVjfGBzHTqJI60QRz
wS08bSIgi7oOuZ_ZQf9xTE0C6bbpTUMZGeZeZz1FC-3WqNIDIwdyVj4T4lRNzYIp-
b49laGDtBplWZsw4mvi8n76eFLKSms5Y6AFkwoUSWQRikExkt80yHbPt8FGYyy4T
ZSZd6euWeMl7tkT1egnovstgASeN3JO",
    {}
  ]}}
```

6.5. Publication

[Future: Consider eliminating this mechanism entirely and instead using messaging flows. The means of achieving this should become better apparent when the problem of publishing large messages via a pull mechanism is considered.]

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. There are two operations:

Claim Post a claim to a published document

PollClaim Check to see if a claim has been posted.

Content is published by appending an entry to an account's Publication catalog by means of a Transact operation. 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 catalog 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 ClaimRespose 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":"NCWK-70N4-VB2S-3J0X-6QYI-EE5V-QIHM",
    "Sender":"alice@example.com",
    "Recipient":"maker@example.com",
    "PublicationId":"EBQK-LU3P-VJLT-ZPG7-B667-L53L-MBEN",
    "ServiceAuthenticate":"ADQX-SBRA-6ACX-ZGGB-IU3L-2TZS-TIKZ",
    "DeviceAuthenticate":"ADNB-SNE2-GEL5-GQQS-JBUB-TY32-JGCC"}}}
```

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


```
{
  "ClaimResponse":{
    "Status":201,
    "StatusDescription":"Operation completed successfully",
    "CatalogedPublication":{
      "Id":"EBQK-LU3P-VJLT-ZPG7-B667-L53L-MBEN",
      "Authenticator":"EAYS-MGDB-YH2G-DWVI-TTWV-4Z6P-UVLM-FENY-Z7MP
-7ZYV-35EA-6GQK-CHBN-K",
      "EnvelopedData":[{
        "enc":"A256CBC",
        "kid":"EBQA-ZKEC-3A4N-DSSH-TQMT-6GON-M6G3",
        "Salt":"9PKNerq6mIRxK2feTIUJ9g",
        "recipients":[{
          "kid":"EBQK-LU3P-VJLT-ZPG7-B667-L53L-MBEN",
          "wmk":"W-1DFznpNx2ZcN6_eWCG0ja0VHFV25k5TVdBMMYbAPc
ZGPZvLj1ww"}
        ]},
        "I01c378amieTGSky6lqXoT8infXsu_EwVamZ6Lp3Q_Ent97vfxd0Y8m1
g_eUVW7ui1Ede6raopIeC7ZWHdZ0lfHDV7agFF6-X_5CmpIDCT110ouL-PIVxXJik
2FB2K0QGwVqMHPu6HpeK4mIro00pJmGJksPiBezVxEq0NGzDw8nbANSxr9tSds0
4sJuApk7IJUZGZ1mSa0dqUF8byioqACRFXDie0-ox23XItr7IOwZf--HUIHnPEtnv
_cxEwCKdAiYus1x3n3WdAxZc9_rqAY-FgXic0qZetde6uH0apnUeCTdnHS3-x84DG
1XaBqthhHMhlwFz0CqftYsB85EjorNKIjw_mwqMmX2QnVSVkhLKW9NdA3w5_mzfd_
nJk0SisY97xeuIsl6-imXTM8LEdyQpuilK7R5qr7ZaYFya9SCje6Ui0mbL8dVqjF
zvDTMQRQ2qnYJKAGDI_Z7P8ehqHkprB2ZDF-5Wqh3JnvYyW5GTJwNgg1ig2iW2pS
QqIXd-Oe_DCLITFgq8ncfM7mEhaXcJUQe01Fj_c4a3o0lhfmv2ts3iK9dieba65Kd
LOWaHau2P0TivQQImMg-3WhCWfU4MbpQlQ_3yEZ51QPGVeUIV0bj3kc1QbaBA1JaB
ch0_E54A9TRWz2xmakyjrEuVSi5aPvd_s-U8DrSDR_YqXSedhMwSBdGqcW7nybFMK
gOxomDZyHnEbNYvwN43vHFDu0yvQWs_dRRe97ylx27VTcAw7AhAt2s3rMZeBwJYfS
rJptABHFEeXPE52g6oSuzrHrG3P_ryAa1Sh9YuoKEYV4UXG8_9B4d8s0D8s503j04
-7Ix1LhqIrSDVYBmn8l6l4nJzSs9_9sbBwtAUIozPNL8LqhbDd1-q0GNjNyt9sQqV
QX76qD8HB6EtdrrZsoBewmwohlLiathJH_909ANUD7tFJONkcyQAXaU64B-j9Z6rF
4vh-9UxYlbbHr0hi9C0kpbnsAsAUzltg8_IvnH8JSTZI4J13WYwkYTx4BUjggX8cV
ekob4ZZupdcfS0UnNjKNqGDjQ8IJqrMwfqL1l1nnBQWpSxsRoRp-ernxQ_Ax8pnWtG
oMxA8yc0WwAr4_S_YQQ0TVxA5KRDQZqF-NC7NX8tacAS8CdPfJwPhjebhDgKwJv0P
ccyjJq8HfTtV9YshXfQQjyxYyA4QT8bvgI4JtJsa_YQ5Zw92tyhb-SwKHNIwnhVrz3
_z6vQPfPdQJUfMtz6GC13eihC19_0vM3IPATyJcenZnQz3rL0m-cFseviCxoLgN5a
LZGCrvlWrwtfn5-evHZF66guRy2ff3BSM4eUi71r2ehU0kYJ8vIjx6NVWzjL-Q2zX
1WHL_bJOaLp0yHV6_2j0TBnaj6QT69Dp0ikXwEEtVzpxVUKLHbnpje2EXXIzNV1Fn
MM65KPT40zxyK0NSq32-xxIi5sxS-woyXxtDv8atRbbE0KPLtLf gianqok9rx2WuN
JpjJpGcIwKk80_1MqFotKI1hVPwCPza688bi41ESyrus1T917HfyVyqfKrDyTjAMK
mYAGkVaa3ASVudNyhJP8rQWwwc_qxruMMQrCe51JvKh7FwsIL14dT6HFNaNcLXSDE
LU7u0_jcViwpu-EysogSeXJ0Z31yG34ve0G_h3K5dN_hk_6PgFK-Soe1dhReVr0b1
JWM3oUmKUv0Kx-XYvsWB_KiuJWRqKePmIhWo9_uZaybh21PoJ75Ct5b3Qk9u49RKf
_x17YE11VQBMdgsV7TS2Qpw0olk4yNg_J9ZjTu1d9UBjabTG0FqdYkeEscDPyIoIq
7DreogM3Y650nqLSpdvJRnueJI5r1a_J9UAVzmk2TA r33fA1VlgmpHptd_t15i2KV
TvMwKl1wmSncGcX-krfCQAIH9ZBmjyVwgurFR9Q6UwuDJYKiRgLLE14ZJmz9hn_j8
ki253lZaYCEYR5F7-nqQpj9lVrYzb476McYISLHzl2SDQKc7vkVo00ZWcsWBjrcwF
MK9LPJMLu8E06TCf_7rRwZ6SmoD0upGVCWikGuAczW7lvCtKty-r_oC1YNGeOxyqG
```

```
9lTij6EzXDdGS7CHRRoIdtYnB6LDLdY8hYQ1cncW3A5g6RHrafChrEihem_0NLplw
2yPofZGIzhj_gMchX7Sg_Exp_0cS32hU1LYX-Hcc0dK8bwQl14a7xsU8dHt8TNbLA
5VuP_bPAv95fTFQvW_Lj6-GGyh7gH2b05QffpusKAddsisIdGvj-r-hNVF99EjsJF
FetvlCP1CFipmCuxo8WgXiayK-_b32Rvav_BxZ4YIcZGRsrH5oI6JywKyG309TAwz
f7853NMhqsVnmLyhGC6Ezs7tQ__WgYqNcfJfo0YG8y78nMvDN2np_pDLw7NRGFhJj
FrVjtdsj7E2gCorPC52JdKJ9jUrQCYpSglZU06CGTRQHnQvMk-H2ftQ4AFVEBJrj0
6527KYwSylkZGMV2WBSrDsYu70Rs1SqI6e-u8pL_VdYPALWvD7SRVPCb90YoPX8lh
gZR0BeCp_kNN6C30B9P9yWRR4JdTsg-LcWtK6iCs_igIOcjfelbsUBXTs7nxI0m1P
mzlvPzkN3Acwrxa6HUEA_t0Lulovb7i04IeDb8nKZgi02Dyohi4aWoyFY2ExsQmgF
qQLExsWPbYJiZEc5BM3idhRDzvjn0N7aPkvqQ568y-e3d30sRXV9uKikFJghh89j5
HHZ3HhTeIHAQAfDKVx7vPULs_5-mleI2v_ZkSag3vXRb23XLfI3x33l_Zxw9MC7YX
_kdJgEqQiJu64AvYWKiA-Fd8uEQMfaUpHamNDl04GFQ8uLWJ7cKkuDbbbi0fiT5z-
77DQAMUiIjJ3mQ_kmileeTNb_qu2jEdJGzx-JQagB7ZIZxn0wZpBSCthhG0uXsERA
XMFamXWLYZtWdWyUSL6AUzcgRf0RRmcvk7yvH-T3dGiIJn4TX0nDp0DBW-MByTeHv
Wrk5k24lpPq07QI6WCdj4b2qSsY0L_ew6zbI9f7Aq8868WMxQ"
    ]}}}
```

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":"EBQK-LU3P-VJLT-ZPG7-B667-L53L-MBEN",
    "TargetAccountAddress":"maker@example.com"}}
```

The response returns the latest claim made as signed message:

```

{
  "PollClaimResponse":{
    "Status":201,
    "StatusDescription":"Operation completed successfully",
    "EnvelopedMessage":[{
      "PayloadDigest":"wMSgIvLpoj69Rw_P_YJ7yXYvo18eCvgU3Hd8DgJ_
07Jv0nqIkC4sZMGFGW6Ntl_3PVwk7bAj51GVrPwqzZ12sg",
      "EnvelopeId":"MBQG-HNR6-TNS7-5N2M-BN4R-ECNA-6ET0",
      "dig":"S512",
      "signatures":[{
        "alg":"S512",
        "kid":"MBUX-YI5W-NTAH-UJN2-4FFC-4PAY-NI73",
        "signature":"fxh1PHK56aMoB9Qkxb3Kcv4UrPQkfGRCd9LwW4Un
3EcR_EqxpWaxZjcXFqdX6d4j9lEstBR2QxKAE_GCYpaXV0LOAVTbb4pwaV9fLDo8r
FtlKfnFFoBZMs10fqcKsJrAnc4AQsT2H5nu6xZ0dq927jYA"}
      ],
      "ContentMetaData":"ewogICJvbm1xdWVJZCI6ICJ0Q1dLLTdPTjQtVk
IyUy0zSk9YLTZRWUktRUU1Vi1RSUhNIiwKICAiTWVzc2FnZVR5cGUiOiAiTWVzc2F
nZUNsYWltIiwKICAiY3R5IjogImFwcGxpY2F0aW9uL21tbS9vYmplY3QiLAogICJD
cmVhdGVkIjogIjIwMjEtMTAtMjVUMTU6NDk6MDhaIn0",
      "SequenceInfo":{
        "Index":1,
        "TreePosition":0},
      "Received":"2021-10-25T15:49:08Z"},
      "ewogICJNZXNzYWdlQ2xhaw0iOiB7CiAgICAiTWVzc2FnZUlkIjogIk5DV0
stN090NC1WQjJTLNKT1gtNlFZSS1FRTVWLVFJSE0iLAogICAgI1NlbnRlciI6ICJ
hbGljZUBleGftcGx1LmNvbSIsCiAgICAiUmVjaXBpZW50IjogIm1ha2VyQGV4YW1w
bGUuY29tIiwKICAgICJQdWJsaWNhdGlvbklkIjogIkVVCUustTFUzUC1WSkxULVpQR
zctQjY2Ny1MNNTNMLU1CRU4iLAogICAgI1NlcnZpY2VBdXR0ZW50aWNhdGUiOiAiQU
RRWC1TQ1JBLTBQ1gtWkdHQi1JVTNMLTJlUW1tVellWiIsCiAgICAiRGV2aWNLQXV
0aGVudGljYXRlIjogIkFETkItU05FMi1HRUw1LUDRUVMtSkJVQi1UWTMyLUpHQ0Mi
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 Access catalog, the request **MUST** meet the authentication criteria specified by the catalog entry. These typically include the request being authenticated by a specific key.Key Agreement

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

Alice added Bob to groupw@example.com as a member. This resulted in Bob receiving the invitation described in section ??? and the following access entry being added to the Access catalog of the group account:

```
{
  "CatalogedAccess":{
    "Capability":{
      "CapabilityDecryptServiced":{
        "Id":"MD6W-KDFX-PSF7-5NBQ-WFJE-34PL-7JWQ",
        "Active":true,
        "GranteeUdf":"bob@example.com",
        "EnvelopedKeyShare":[{
          "enc":"A256CBC",
          "kid":"EBQL-CG0H-LPTR-WNYL-RXDU-7LK4-G4GZ",
          "Salt":"xUeBS0V_Z4GJV9s2N30APw",
          "recipients":[{
            "kid":"MBBR-KLL4-YRFX-K63E-2DCT-6UGQ-Z5JC",
            "epk":{
              "PublicKeyECDH":{
                "crv":"X448",
                "Public":"2hYs3byJFTbFtPgYaBplyVgtBJ0uYjMSvi8
BcCEFoOy8JRa0Wg37yggj8m4hUjoZhPlC6bZ-MQEoA"}},
              "wmk":"9iJlp63lB90g5mog6030f1NJtrVsTFCC2GzUKDLv-P
Hb3Dn9tNTepw"}
          ],
          "ContentMetaData":"ewogICJNZXNzYWdlVHlwZSI6ICJLZX1EYX
RhIiwKICAiY3R5IjogImFwcGxpY2F0aW9uL21tbS9vYmplY3QiLAogICJDcmVhdGV
kIjogIjIwMjEtMTAtMjVUMTU6NDk6MDBaIn0"},
          "hY6vVvnbC6y-a5ULJNUh0aVv_YEiu1BRiTDS-gBY6nKX7tqs_oIMFR
Qipj8bUHdtE46dYu2Ud0ue88CM0X3qsLTDP_5d7UgqKUK5yfr3n1SJz1KU7u3W7YF
b9t0W-svw3sE4UncLN7vJ5V90bF9DUS_xek8s9SxnyeZxWY5z5GvCTJr0TxCO5hFh
FR6Bwv2cwqG9eqyFhWCu4GrrgLUk-fk5rOdF9KTZW6g0wqg0xTUFduY4tfTTwzp4N
iffzB2rj88nOkvvW-6SwVVrExLY5E4l-7C1fnlBH-20Zz-z8faYy85gDl6zVGdyYd
JelTP1Rmlbm_tsw2NRYKNz4WCareq_QZx3XBBqoKNi1CA4Gd02qiM00E6NigTKB9C
Rlc4EzUtwCU8Zdw6yUWxxCEtHoXVh80pvWixTdmznouCLyDUVsvfx1dM-PIrSfbEl
_K2v9IHZrtcgh5vahoawY60ELJYLnARsvnWchyfZCgZMNbknDYiAyNAwxI_Wgm3xo
7vyTkF0Are0-RwofxYlyzDcFk2yA6-edTAq0PHi0FSl3j90hVmcaWC903uI_keNGU
4egZ370UCWrFuz-05woKJD1lJu-GubgpJ5YTc628m-_6ASaEw9G4uGiIx0oCcSqR
SENXe9tnD90HkaZCCzhz2Cscfm8uRAVG9wasFrDjaKxUwf6N3nVjOmFVJwBl_G14i
go4PmNHI2dgJBLqxlof76g"
        ]}}}}
}
```

The private key (in this case a key share) is encrypted under the service key.

To make use of the access entry, a request is made that specifies the key share to be operated on and the public key parameters to perform the agreement with.

The request payload:

```
{
  "OperateRequest":{
    "AccountAddress":"groupw@example.com",
    "Operations":[{
      "CryptographicOperationKeyAgreement":{
        "KeyId":"MD6W-KDFX-PSF7-5NBQ-WFJE-34PL-7JWQ",
        "PublicKey":{
          "PublicKeyECDH":{
            "crv":"X448",
            "Public":"mNrpSHZmFqcMBHYAwEyp0tUshHkBafWjCe3mDMcoV
PIuqBhrbj5ZIpQdgfcS1BWgb5cwGXmIEPcA"}}}}
    ]}]}
```

The service checks to see if the request is authorized and if so, performs the operation and returns the result:

```
{
  "OperateResponse":{
    "Status":201,
    "StatusDescription":"Operation completed successfully",
    "Results":[{
      "CryptographicResultKeyAgreement":{
        "KeyAgreement":{
          "KeyAgreementECDH":{
            "Curve":"X448",
            "Result":"JmEGQN2398IzXNI4j1CG4ZWtLVk1u8P8SnBw1_cEQ
0s5INanZUEewbDxEQp5oc1_QP0EnBoSdUMA"}}}}
    ]}]}
```

Future: Currently, the access catalog is encrypted under the service encryption key. It would be better to encrypt the catalog under an encryption key specified by the service during the process of account binding. This would allow a service to assign a unique encryption key to each account and limit access to that key to the hosts servicing that specific account.

6.6.1. Generate Key Shares

Generation of threshold key shares is planned but not currently supported.

6.6.2. Threshold Sign

Threshold signature is planned but not currently supported.

6.7. 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.

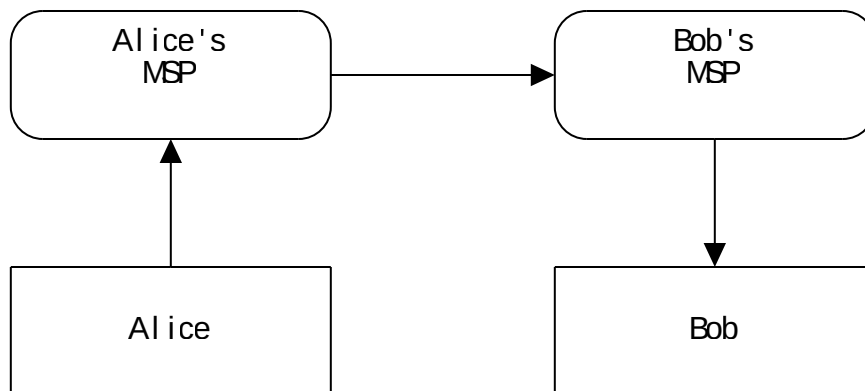


Figure 2: The Mesh Four Corner Messaging Model

The Post transaction is only used to exchange messages between services. The client sends and receives messages through interactions with the outbound and inbound spools of the account.

6.7.1. Sender.

To send a message, the client creates the Mesh Message structure, encapsulates it in a DARE Message and appends the message to the Outbound spool of the account using the Transact operation..

The DARE Message **MUST** be signed under the account signature key.

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.

6.7.2. Outbound Service

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 my means of either profile or ticket authentication.

>>>> Unfinished ProtocolPostServiceService

[Not Yet Implemented]

After the message has been sent, the service updates the message status on the outbound spool.

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 completed in less than a second, the number of threads allocated to sending outbound messages might be increased.

6.7.3. Inbound Service

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.

6.7.4. Recipient

Messages are received by synchronizing the outbound spool.

7. Access Control

[This section to be expanded in future drafts]

Access control is effected through the usual division of authentication and authorization.

Authentication of operation requests is performed by the RUD layer [[draft-hallambaker-mesh-rud](#)] .

7.1. Direct authorization

Any request authenticated under the profile authentication key is authorized to perform any account operation without restriction.

7.2. Access Catalog authentication

If the authentication key presented has a matching Access Catalog entry, the device is authorized to perform operations as specified in that entry.

8. 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.

8.1. 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](#)].

8.1.1. 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.

8.1.2. 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.

8.1.3. 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.

8.1.4. 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. The message specifies the salted pin value used to verify attempts to use the PIN, the action for which it is authorized. Since this PIN has been issued to authorize a device connection, the roles for which the device are authorized as well. This allows the connection request to be accepted without asking for further input from the user.

```
{
  "MessagePin":{
    "MessageId":"AAP0-PUCK-AIYZ-FS0X-0BI5-YZZB-RVT2",
    "Account":"alice@example.com",
    "Expires":"2021-10-26T15:49:02Z",
    "Automatic":true,
    "SaltedPin":"ADL6-MGFR-DK2V-XMCH-Y4VK-FG4R-AIDL",
    "Action":"Device",
    "Roles":["threshold"]
  }}
}
```

8.2. 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":"NCGB-6PXA-YG6T-GSC3-37HF-5QG2-SC43",
    "References":[{
      "MessageId":"AAP0-PUCK-AIYZ-FS0X-0BI5-YZZB-RVT2",
      "ResponseId":"MDT3-TM62-G3X0-ESY0-WQZX-IR2B-YNHW",
      "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.

8.3. 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.

8.3.1. 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]

b2d1d29nSUNBZ01DQWlWV1JtSwpvZ01rMUJUe1F0V1U5CiAgRk5DMU5TalpGTFZsV
FVFTXRMVpFVVMxTlJqUTNMVXBHuzBzaUxBb2dJQ0FnSUNBaVVIVmliR2xqvUdGeV
kKICBXMWxkR1Z5Y3lJNklIc0tJQ0FnSUNBZ01DQWlVSFZpYkdsalMyVjVSVU5FU0N
JNklIc0tJQ0FnSUNBZ01DQQogIGdJQ0pqY25ZaU9pQWlSV1EwTkRnaUxBb2dJQ0Fn
SUNBZ01DQWdJbEIXWW14cFl5STZJQ0kxVDJwMVRqazNTCiAgVlpwY0VGTlYyNVVlR
2gyU21VNwVIZFNiSF10VEV0SVlUSXdWVlpCTmxKU1ZWwXpXbXAZwVZsbFpXNwxDaU
EKICBnTTBweVdERKJNMTlLYzFwTU16SldRVUp3WwtSTlRWZEJJbjE5ZlN3S01DQWd
JQ0pCWTJ0dmRXNTBRWFYwYQogIEdWdWRHbGpZWfJwYjI0aU9pQjdDaUFnSUNBZ01D
SlZaR1lpT2lBaVRVRkt0UzFLVGtWTeXUVkXJXRFV0V0VGCiAgVlJ5MVhUVFJPTFZVM
VRGWXRxa0pKU2lJc0NpQWdJQ0FnSUNKUWRXSnNhV05RWVhKaGJXVjBawEp6SwpvZ2
UKICB3b2dJQ0FnSUNBZ01DSlFkv0pzYVd0TFpYbEZRMFJJSwpvZ2V3b2dJQ0FnSUN
BZ01DQWdJbU55ZGlJNklDSgogIFl0RFE0SWl3S01DQWdJQ0FnSUNBZ01DSlFkv0pz
YVdNaU9pQWlTbEIZvVZzdFJVNWpSR05UTXpCNWFXNXRUCiAgV0ZpVUVKQlFuZ3hUV
Gx4V0dvMFJVChBOekJav1VJMlJsUjBZMU42YTBadvl3b2dJRVp6Vmp0eWVHRlVaMl
YKICBNV2xwTWVEUm9SRE5DUjFwbFFTSjlmwDBzQ2lBZ01DQWlRV05qYjNwdWRGTnB
aMjVoZEhWeVpTSTZJSHNLSQogIENBZ01DQWdJbFZrWmlJNklDSk5RbFZhTFRSTVUx
Z3RRVWRTUmkwMFZVWldMVUZSVlRjdFZGRkJSerFXV0V4CiAgRklpd0tJQ0FnSUNBZ
0lsQjFZbXhwWTFCaGntRnRawFJsY25NaU9pQjdDaUFnSUNBZ01DQWdJbEIXWW14cF
kKICAwdGxlVVZEUKvNaU9pQjdDaUFnSUNBZ01DQWdJQ0FpWTNKMklqb2dJa1ZrTkR
RNElpd0tJQ0FnSUNBZ01DQQogIGdJQ0pRZFdkc2FXTWlPaUFpYjFCQlNXOVZNWfZU
TFhvME9GRmFPVjkWV2xCMk4yeHlSR3RvUTB0d1NYQk9kCiAgV1JLYjFOS2JXbDRkV
1JTZURCeVwNVdXQW9nSUC1WVMzUmZUR2xHVG5WbVUySnBjMDF3WwXRNfowTTRRU0
oKICA5ZlgxOWZRIiwKICAgICAgICAgIHsKICAgICAgICAgICAgInNpZ25hdHVyZXM
iOiBbewogICAgICAgICAgICAgICAgImFsZyI6ICJTNTEyIiwKICAgICAgICAgICAg
ICAgICJrawQiOiAiTUJYVy10UFdWLVY3SlEtUVI3Ri1GUDVLLVA1U0EtSkI0SCIsC
iAgICAgICAgICAgICAgICAic2lnbmF0dXJlIjogIjNlSnVall2MWQyVXlIMG1IUE
9rTEdXWHZ6UFZBRlFwak5FZEkyck2k3Tl9nTXZCeTgKICBLVFJWLV80ZnptZ2tvWkw
0Nn1OdGhCRHkyU0F2ZFVLdFVWUTc0c0NsQlY1aDFlSFpkcm41Wl84ekk3d0lGbaog
IHM3MTZVUnFOa0tvd1V0OWhkQnN0emw2WURfZ1lWwJhSUEXkMDlQqjhBIn1dLAogI
CAgICAgICAgICAiUGF5bG9hZERpZ2VzdCI6ICJDUdMxd1hrdUdFQS0tNUdaWWJ1TV
VMZHv4dFdmZmYyZEs3RUFCLVRZX0RCSzEKICB0ZnB2Z2FiMm5zRjh4Y3JYRGZDRUp
KQl1Famd4TzFXbHJGMWl6amV2QSJ9XSwKICAgICAgICAiUHJvdG9jb2xzIjogW3sK
ICAgICAgICAgICAgIlByb3RvY29sIjogIm1tbSJ9XX1dfX0",

```
{
  "signatures": [{
    "alg": "S512",
    "kid": "MBUZ-4LSX-AGRF-4UFV-AQU2-TQAG-VXLE",
    "signature": "DbnBy9SoZHxD9EUbFvBpQW7KEKnNm-EUKMdvNE_b
J6QU-8gVevRj1BQbnnnTt8EtA6WiTl-vMQ2ATfDuFXt8r9FtataQKWOSBq_zWqXBY
KQiQtP0R0gYAr7b8VavWhXpJbScrFg_RDrdVo8PIPdsmDkA"}
  ],
  "PayloadDigest": "ZFUZc0M0__xinLcbyzTF33GyMZ3pqFe1WQhimoDJ
YCGrwEvyBnsMTV4LDG3oYbwYJQQzyEF2LRC3pD76R4AcrQ"}
},
"Reply": true,
"Subject": "alice@example.com",
"PIN": "AAIE-IVI5-54X0-5PHG-VE62-FFS7-62GQ"]}]}
```

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.

8.3.2. 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.

8.3.3. 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.

8.4. 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 CapabilityDecryptServiced 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.

>>>> Unfinished ProtocolGroupInvite

Missing example 3

8.5. 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":"NDAD-KLJY-C5J0-JGXL-VUWG-Y6PP-PSFJ",
    "Sender":"console@example.com",
    "Recipient":"alice@example.com",
    "Text":"start"}}
```

Alice accepts the request and returns the following response:

```
{
  "ResponseConfirmation":{
    "MessageId":"MCT4-SVZ2-BL5Y-DR5B-TF4S-WIGH-CJTM",
    "Sender":"alice@example.com",
    "Recipient":"console@example.com",
    "Request":[{
      "EnvelopeId":"MDVA-HSIH-UJBT-PEVO-GZNQ-JF30-YHTM",
      "ContentMetaData":"ewogICJVbmlxdWVJZCI6ICJOREFELUtMSlktQz
VKTy1KR1hMLVZVV0ctWTZQUC1QU0ZKIiwKICAiTWVzc2FnZVR5cGUiOiAiUmVxdWV
zdENvbmZpcm1hdGlvbiIsCiAgImN0eSI6ICJhcHBsaWNhdGlvbi9tbW0vb2JqZWNO
IiwKICAiQ3JlYXRlZCI6IClYMDIxLTEwLTI1VDE1OjQ0jU3WiJ9",
      "SequenceInfo":{
        "Index":4,
        "TreePosition":6201},
      "Received":"2021-10-25T15:48:57Z"},
      "ewogICJSZXN0Z29uZmlybWFWaW9uIjogewogICAgIk1lc3NhZ2VJZC
I6ICJOREFELUtMSlktQzVKTy1KR1hMLVZVV0ctWTZQUC1QU0ZKIiwKICAgICJTW5
kZXIiOiAiY29uc29sZUBleGFtcGxlLmNvbSI6ICAgICAgICAgICAgICAgICAgICAg
aWNlQGV4YW1wbGUuY29tIiwKICAgICJUZSh0IjogInN0YXJ0In19",
      {}
    ],
    "Accept":true}}
```

9. 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.

Four connection interactions are currently defined support connection of devices with different 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.

PIN Authenticated A variation of the Witness Authenticated interaction in which the connection process is initiated by creating a PIN value which is communicated to the device by some out of band means and used to authenticate the connection request.

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. Implementation of this mechanism is identical to the PIN authenticated scheme except that the PIN code is presented to the connecting device by means of a QR code.

Preconfigured (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 in [[draft-hallambaker-mesh-schema](#)]. 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.

9.1. Witness/PIN Authenticated

The witness authenticated, PIN authenticated, and Dynamic QR code interactions all follow a common interaction pattern.

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

Phase 1: Issue of PIN credential (PIN and Dynamic QR code only) 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. In the case of the Dynamic QR code interaction, this is a QR code that is scanned by the connecting device.

Phase 2: Onboarding Device Request to Service The onboarding device creates a RequestConnect message. In the PIN authenticated and Dynamic QR Code interactions, the RequestConnect is authenticated by the Device Authentication key and the PIN issued earlier. In the Witness Authenticated interaction, it is authenticated by the Device Authentication key alone.

The onboarding device presents the RequestConnect message to the service by means of a Connect operation to the service 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 and returned to the connecting device.

Phase 3: Administration Device Acceptance

The account holder authenticates RequestConnect message and uses an administrative device to accept or reject the connection request.

If the RequestConnect message has been authenticated by a PIN code, the connection request can be accepted automatically without additional user interaction.

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.

9.1.1. Phase 1:

```
Alice> account pin /threshold
PIN=ABQR-G05I-FPIE-TK50-M4VU-DALE-WM
(Expires=2021-10-26T15:49:02Z)
```

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

The URI containing the account address and PIN is:

```
mcu://alice@example.com/ABQR-G05I-FPIE-TK50-M4VU-DALE-WM
```

9.1.2. 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 ^
ABQR-G05I-FPIE-TK50-M4VU-DALE-WM
Device UDF = MCVN-XLLT-LLNW-U4HR-BOMG-RA6Z-UWRR
Witness value = 2WZP-KNZF-JMKO-RQSU-WYTH-UU35-NWXV
```

The device generates a RequestConnect message as follows:


```
{
  "RequestConnection": {
    "MessageId": "NA46-HSVG-N5NU-EXKZ-4X7G-GSF7-DUWS",
    "AuthenticatedData": [ {
      "EnvelopeId": "MCVN-XLLT-LLNW-U4HR-BOMG-RA6Z-UWRR",
      "dig": "S512",
      "ContentMetaData": "ewogICJVbmlxdWVJZCI6ICJNQ1ZOLVhMTFQtTE
xOVy1VNEhSLUJPTUctUkE2Wi1VV1JSIiwKICAiTWVzc2FnZVR5cGUiOiAiUHJvZm1
sZURldm1jZSIsCiAgImN0eSI6ICJhcHBsawNhdGlvbi9tbW0vb2JqZWNOIiwKICAi
Q3JlYXRlZCI6ICIyMDIxLTEwLTI1VDE1OjQ5Q5jAYWiJ9"}],
      "ewogICJQcm9maWxlRGV2aWNL1IjogewogICAgIClByb2ZpbGVTaWduYXR1cm
UiOiB7CiAgICAgICJVGyOiAiAiTUNWTi1YTEXULUxMTlctvTRIUI1CT01HLVJBNlO
tVvdSUiIsCiAgICAgICJQdWJsawNQYXJhbWV0ZXJzIjogewogICAgICAgICJQdWJs
awNLZXlFQ0RIIjogewogICAgICAgICAgImNydiI6ICJFZDQ0Q0CIsCiAgICAgICAgI
CAiUHVibGljIjogImQzMlg1b3NHMUTPRTVvbWZWTUZqRES0MF84eGJ5RTNrZlV3T3
dUY1BXMxZJew8zQ0N0dkoKICA5aXBseTFBmlg4TjVMejhXSXRTWml3S0EifX19LAo
gICAgIkVuY3J5cHRpb24iOiB7CiAgICAgICJVGyOiAiAiTURNVY1XQ01SLUZKTU8t
N1pInI1DTjNKLUTVwkwUkxBWCIsCiAgICAgICJQdWJsawNQYXJhbWV0ZXJzIjoge
wogICAgICAgICJQdWJsawNLZXlFQ0RIIjogewogICAgICAgICAgImNydiI6ICJYND
Q4IiwKICAgICAgICAgICJQdWJsawMiOiAiAibnRfcHU0WVJieXIwWUxNY1JXdm1NLXJ
UWlhXZlB1UUhwa1h0TWdud2hweUVXdjBHUmpsaAogIDlvcnBPC21jVTI3LWxtenhJ
T3dTWGPBQSJ9fX0sCiAgICAiU2lnbmF0dXJlIjogewogICAgICAiVWRmIjogImk1CM
lUtNkdNNy1QSEkzLTNVVU4tTElaNi1VVudKLULXNVEiLAogICAgICAiUHVibGljUG
FyYW1ldGVycyI6IHsKICAgICAgICAiUHVibGljS2V5RUNESCi6IHsKICAgICAgICAg
gICJjcncyOiAiAiRWQ0NDgiLAogICAgICAgICAgIClB1YmXpYyI6ICJVbmk2NUVYY0RY
YmVXaw1RbFk5OXhhSG5SWmpiSFpBS2lUNmRlZDR0MwP2TGpFVmhMb3lYCiAgVlFra
WRoZ1lsV21fRHM2ODdpdUpoX1VBI19fSwKICAgICJBdXRoZW50aWNhdGlvbiI6IH
sKICAgICAgIClB1YmXpYyI6ICJNQ1ZOLVhMTFQtTEwvZlV3T3dUY1BXMxZJew8zQ0
N0dkoKICA5aXBseTFBmlg4TjVMejhXSXRTWml3S0EifX19LAogICAgICAgICAgIClB
1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCi
AgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYy
I6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAg
ICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJF
ZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgIC
AgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0
CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1Y
mXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAg
ICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6
ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgIC
AgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZD
Q0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAg
IClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0C
IsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1Ym
XpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAg
ICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6
ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgIC
AgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZD
Q0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAg
IClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0C
IsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1Ym
XpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAg
ICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6
ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgIC
AgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZD
Q0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAg
IClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0C
IsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1Ym
XpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAg
ICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6
ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgIC
AgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZD
Q0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAg
IClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0C
IsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1Ym
XpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAg
ICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6
ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgIC
AgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZD
Q0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAg
IClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0C
IsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1Ym
XpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAg
ICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6
ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgIC
AgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZD
Q0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAg
IClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0C
IsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1Ym
XpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAg
ICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6
ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgIC
AgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZD
Q0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAg
IClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0C
IsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1Ym
XpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAg
ICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6
ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgIC
AgICAgIClB1YmXpYyI6ICJFZDQ0Q0CIsCiAgICAgICAgICAgIClB1YmXpYyI6ICJFZD
Q0
```

The service receives the connect request and authenticates the message under the device key. The service cannot authenticate the message under the PIN code because that is not known to the service as the service cannot decrypt the local spool.

Having authenticated the connect request, the service generates a random nonce value. The random nonce together with the device and account profiles are used to calculate the witness value.

The AcknowledgeConnection message is created by the service:

```
{
"AcknowledgeConnection":{
"MessageId":"2WZP-KNZF-JMKO-RQSU-WYTH-UU35-NWXV",
"EnvelopedRequestConnection":[{
"EnvelopeId":"MDKW-3KOD-ZTW6-MRIB-AARK-UACM-PDOZ",
"ContentMetaData":{"ewogICJvbm1xdWVjZCj0QQTQ2LUhTVkctTjVOVS1FWETalTRYN0ctR1NGNy1EVVDtIiwKICAiTWVzc2FnZVR5cGUiOiAiUmVxdWVzdENvbml5Y3Rpb24iLAogICJjdHkiOiAiYXBwbGljYXRpb24vbW1tLT29iamVjdCIsCiAgIkNyZWFi0ZWQiOiAiMjAyMS0xMC0yNVQxNT00TowMloifQ"}},
"ewogICJSZXNF1ZXN0Q29ubmVjdGlvbiEIHsKICAgICJNZXNzYWdlSWQiOiAiTkE0Ni1IU1ZHLU41TlUtRVhlLWi00WddHlUdTRjctRFVXUyIsCiAgICAIQXV0aGVudGljYXRlZERhdGEiOiBbewogICAgICAgICJfbnZlbG9wZULkIjogIk1DVk4tWEExMVCI0TE5XLVU0SFItQk9NRy1SQTZaLVVXUlIiLAogICAgICAgICJkaWciOiAiUzUxMiIsCiAgICAgICAgIkNvbmlnbnRNZXRhRGF0YSI6ICJld29nSUNKVmJtbHhkV1ZKwkNjNkLDSk5RMVPPTFZoTVRGUXRURXhpVnkxCiAgVk5FaFNMVUpQVFVjdFVRRTJXaTFWvjFKU0lpd0tJQ0FPvFdwemMyRm5aVlI1Y0dVaU9pQWlVSEp2Wm1sc1oKICBVUmxbWxqwlNJc0NpQwdJbU4wZVNjNkLDSmhjSEJzYVd0AGRhbHZiaTl0YlcwdmIySnFaV04wSwl3S0lDQogIGlRM0pswVhSbFpdSTZJQ0l5TURJeEXURXdMVEkxVkrFMU9qUTVPakF5V2lKOSJ9LAogICAgICAIzXdvZ0lDSlFjbTltYVd4bFJHVjJhV05sSWpvZ2V3b2dJQ0FnSWxCeWIyWgogIHBiR1ZUYVdkdVlYUjFjbVVPt2lCN0NpQwdJQ0FnSUNKVlpHWWlPaUFpVFV0V1RpMVlURXhVTFV4TVR5Y3RWCAgVFJJVWkxQ1QwMUhMVkpCTmxvdFZWZFNVaUlzQ2lBZ0lDQwdJQ0pRZFdKc2FXTlFZWEPoYldWMFPySnpJam8KICBnZXdvZ0lDQwdJQ0FnSUNKUWRXSnnNhV05MWlhSRlEWUklJam9nZXdvZ0lDQwdJQ0FnSUNBZ0ltTnlkaUk2SQogIENKRlpEUTBPQ0lzQ2lBZ0lDQwdJQ0FnSUNBaVVIVmliR2xqSWpvZ0ltUXpNbGcxYjNOSE1vdFBVSFZZYldacIAgV1RVWnFSRXMwTUy4NGVHSjVSVE5yWmxWM1QzZFFVZbEJYTvhSaSmVXOHpRME5PZGtvS0lDQTVhWEJzZVRGQk0KICBsZzRUa1ZNZwpofFNYUlRXbwWzUzBFawZYMTlMQW9nSUNBZ0lrVnVZM0o1Y0hScGIyNgIPauI3Q2lBZ0lDQogIGdJQ0pWkdZaU9pQWlUVVJOVnkvWFEBFNMVVPVLFVU4dE4xcEl0aTFEVEGpOS0xvdfZXA3d0Vwt4QldSXNDCAgaUFnSUNBZ0lDSlFkV0pzYVdOUVlYSmhiV1YwWlhKeklbq2dld29nSUNBZ0lDQwdJQ0pRZFdKc2FXTkxaWGwKICBGUTBSSUlqb2dld29nSUNBZ0lDQwdJQ0FnSW10eWRpSTZJQ0pZTKRRNElpd0tJQ0FnSUNBZ0lDQwdJQ0pRZAogIFdKc2FXTWlPaUFpYm5SZmNIVTBXVkpZVhJd1dVeE5ZMUPyZG1sTkxySlVxBghYWmxCMVWwaFdhMWgwVfdkCiAgdWQyaHdlVVZYZGpCSFVtcHNhQW9nSURSVmNuQlBjMjFqVlRJMOxxeHRlbnhmKVdNKVfDHCEJRu0o5Zlgwc0MKICBpQwdJQ0FPVTJsbmJtRjBkWepsSWpvZ2V3b2dJQ0FnSUNBavZXUM1Jam9nSWsxQ01sVXR0a2R0TnxUVNFawogIHpMVE5WVlU0dFRfbGFoaTFwVlVks0xVbFhOVkVpTEFvZ0lDQwdJQ0FPVUhWaWJHbGpVR0Z5WvcxbGRHVnljCiAgeUK2SUhzS0lDQwdJQ0FnSUNBavVIVmliR2xqUZJWNJVJTKVTQ0k2SUhzS0lDQwdJQ0FnSUNBZ0lDSmpjblkkICBpT2lBaVJXUTBORgdptEFvZ0lDQwdJQ0FnSUNBZ0lsQjFZbXhwWlJNkLDSlZibwsyTlVwWkwUl1ZbVZYYQogIFcxUmJGazVPWGHou0c1U1dtcGlTRnBCUzJsVU5tUmxaRFIwTVdwMlRHCEZwbWhNYjNswUNpQwdWbEZyYVdSciAgb1oxbHNWmjFmUkhNMk9EZFBkvVBvWDFWQklumTlmU3dLSUNBZ0lDSkJkWFJvWlclMGFXTmhkR2x2Ym1JNkkKICBIc0tJQ0FnSUNBZ0lsVmtaaUk2SUNKTlFrBERMVU15UzBRdFFrSlNXQzFITkZCRUXUUKpNazh0VUUxRVRPMQogIFhUMWRCSWl3S0lDQwdJQ0FnSWxCMVlTeHBZMUJoY21GdFPyUmxbk1pT2lCN0NpQwdJQ0FnSUNBZ0lsQjFZCiAgbXhwWTB0bGVVVVkrSRWdpT2lCN0NpQwdJQ0FnSUNBZ0lDQWLZM0oySWpvZ0lsZzBORGDptEFvZ0lDQwdJQ0EKICBnSUNBZ0lsQjFZbXhwWlJNkLDSjVjR0PUVjFablNIRXdimjVvVDJ0VvQxrRjRNVU5rwjNkSVJWUlFURWxtVAogIFZRMWFxMvNTMHBMtUdvelZ6Qktabmt0Vms1dUNpQwdPVEptVHPaclngbDBNMMruYjBoWFntMDRUWFpXTkhkCiAgOkluMTlmWDE5IiwKICAgI
```

```
CAGewogICAgICAgICJzaWduYXR1cmVzIjogW3sKICAgICAgICAgICAgImFsZyI6IC
JTNTeyIiwKICAgICAgICAgICAgICAgImtpZCI6ICJNQ1Z0LVhMTFQtTEw0Vj1VNEhSLUJ
PTUctUkE2Wi1V1JSIiwKICAgICAgICAgICAgICAgInNpZ25hdHVyZSI6ICJGUgPjQ3py
N3MwRmJVSHJaT09oVUd1ZXNVTKJKT05YOUZlLUNfXzg3ZXlrSFc1VU95CiAgbExob
mZ4ZmtVTFFFWUklZM2dkRmdmTFNKNW1BTFlRM3Y5UkxKdGhkUGhNcHhEZnV5SWlEM1
Z0LWNobzJRR2EKICBTcTdpbU8tWmxLWkxQX2p3TzQ4QW5xY05abkp3Y2RLTUZoa3p
aRGprQSJ9XSwKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIC
X0g5bUVlYnlfWwXrOHpDVGx1TG1oemVXVm9ma1djY2YKICBncENSSTFoUkgzM5fS
lVBSlpxYXU3Nm8yQWVVFVB1My1EZXU5VFhhdYJ9XSwKICAgICAgICAgICAgICAgICAg
I6ICItcHl1a0E4S0pxZHZX2hlUElLRlpRIiwKICAgICAgICAgICAgICAgICAgICAgIC
VQ0stQUlZWl1GU09YLU9CSTUtWVpaQi1SVlQyIiwKICAgICAgICAgICAgICAgICAgIC
IndWMDRjckJhZjdoLTVmY2xwQUdNSXN5NVpVWm5LY3FiUFhEVWJ1WmZYWW96dUk5W
gogIEItZWl1d25xMmF3dnB3NmK3b0Z2Z1ktb1cwalFyVVlxSlNUV0RnIiwKICAgIC
JBY2NvdW50QWRkcmVzcyI6ICJhbGljZUBleGFtcGxlLmNvbSJ9fQ"
```

```
  ],
  "ServerNonce": "q09R3oT24ED05GcYlYCBsg",
  "Witness": "2WZP-KNZF-JMKO-RQSU-WYTH-UU35-NWXV"}}}
```

The AcknowledgeConnection message is appended to the Inbound spool of the account to which connection was requested so that the user can approve the request. The ConnectResponse message is returned to the device containing the AcknowledgeConnection message and the profile of the account.

The device generates the witness value, verifies it against the value provided by the server and presents it to the user as seen in the console example above.

9.1.3. Phase 3:

The user synchronizes their pending messages:

```

Alice> message pending
MessageID: 2WZP-KNZF-JMKO-RQSU-WYTH-UU35-NWXV
    Connection Request::
    MessageID: 2WZP-KNZF-JMKO-RQSU-WYTH-UU35-NWXV
    To: From:
    Device: MCVN-XLLT-LLNW-U4HR-BOMG-RA6Z-UWRR
    Witness: 2WZP-KNZF-JMKO-RQSU-WYTH-UU35-NWXV
MessageID: NCVP-LUEL-F30I-QOAM-HND2-WG5Z-346D
    Group invitation::
    MessageID: NCVP-LUEL-F30I-QOAM-HND2-WG5Z-346D
    To: alice@example.com From: alice@example.com
MessageID: NDAD-KLJY-C5JO-JGXL-VUWG-Y6PP-PSFJ
    Confirmation Request::
    MessageID: NDAD-KLJY-C5JO-JGXL-VUWG-Y6PP-PSFJ
    To: alice@example.com From: console@example.com
    Text: start
MessageID: NAP7-MRKY-LGHV-W6C2-IDAX-ELNG-V5NT
    Contact Request::
    MessageID: NAP7-MRKY-LGHV-W6C2-IDAX-ELNG-V5NT
    To: alice@example.com From: bob@example.com
    PIN: AAIE-IVI5-54X0-5PHG-VE62-FFS7-62GQ
Alice> account sync /auto

```

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.

Bug: This command is currently showing superfluous pending messages due to the failure to clear messages processed in earlier examples.

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.

This is returned to the onboarding device by wrapping it in a RespondConnection message posted to the local spool of the account.

[illegible]

[illegible]

```
ofkWccfgpCII1hRH3fn_JUAJZqau76o2AaUTPu3-Deu9TXaw"}
],
"EnvelopedConnectionAddress": [{
  "dig": "S512"},
  "e7QRQ29ubmVjdGlvbkFkZHJlc3N7tA5BdXRozW50aWNhdGlvbnu0EFB1
YmxpY1BhcmFtZXRLcnN7tA1QdWJsaWNlZXlFQ0RIe7QDY3J2gARYNDQ4tAZQdWJsa
W0IOdulr0WkJsqoELzV6ZGITA3QJhpT6D22IPFeUSgiSp-K8l1msYOAPUExAkdsvR
WgGhs_o0v7o4kEgH19fbQHQWNjb3VudIARYWxpY2VAZXhhbXBsZS5jb219fQ",
{
  "signatures": [{
    "alg": "S512",
    "kid": "MCB0-ZK4F-QFYM-63TK-TA2C-LHQY-7QW5",
    "signature": "03oFK4MzMkSRYzgImz6WJw3JSZ4fmU7djU63aS
oQWhCmzNDCf4XCKfx-bKoJesukT_VTGq5bw-AAWq0_2Zf03pqjr5CwSH9yOKBzH0t
pPFDAeKi7oBM43kk5rqljTN0f4EtcAiEqYFohIVoPbn75NAwA"}
  ]}
],
"EnvelopedConnectionService": [{
  "dig": "S512",
  "ContentMetaData": "ewogICJNZXNzYWdlVHlwZSI6ICJDb25uZWNO
aw9uU2VydmVjZSI6ICAgImN0eSI6ICJhcHBsaWNhdGlvbi9tbW0vb2JqZWNOIiwKI
CAiQ3JlYXRlZCI6ICIyMDIxLTEwLTI1VDE10jQ50jAzWiJ9"},
  "e7QRQ29ubmVjdGlvb1NlcnZpY2V7tA5BdXRozW50aWNhdGlvbnu0A1Vk
ZoAiTURCVy03TjVTLUpJWUUtT0JPNi1HWlR0LURLUTYtRDNTSLQQUHVibGljUGFyY
W1ldGVyc3u0DVB1YmxpY0tleUVDREh7tAnjcnaABFg0NDi0B1B1YmxpY4g526WvRa
QmyqqQvNXpkYhNrdAmG1PoPbYg8V5RKCKJn4ryXWaxg4A9QTEAp2y9FaAaGz-g6_u
jiQSAfX19fX0",
{
  "signatures": [{
    "alg": "S512",
    "kid": "MCB0-ZK4F-QFYM-63TK-TA2C-LHQY-7QW5",
    "signature": "9m0vcnmYQFNYfzGd0dEH605dJzn72QGwibh4j9
Lvr2skx_QmOz52TCT9P884t5KzVfAv0SdRFsUAfqm-0lo3vDDRBAAGbm9uBQ-1YR7
r3B430lHR1KnUD5Iwqsbn2lpFHNpZLt0fkmliATLNRd6UiYA"}
  ]},
  "PayloadDigest": "vPLHz2roZch2iYj7GhGYup4R1v4b1WDC0rAI03
R-hq5AVRT8FxVmVwhFK5TF8Zh_KFsti0qU9gP6-QliFCPnCg"}
],
"EnvelopedConnectionDevice": [{
  "dig": "S512",
  "ContentMetaData": "ewogICJNZXNzYWdlVHlwZSI6ICJDb25uZWNO
aw9uRGV2aWNlIiwKICAiY3R5IjogImFwcGxpY2F0aw9uL21tbS9vYmplY3QiLAogI
CJDCmVhdGVkIjogIjIwMjEtMTAtMjVUMTU6NDk6MDNaIn0"},
  "e7QQQ29ubmVjdGlvbkRldmVjZlZlbnRlbnRyY2F0aW9ue7QDVWRm
gCJNREJXLtdONVMTskLZRS1PQk82LUdaVE4tREtRni1EM1NIitBBQdWJsaWNQYXJhb
WV0ZXJze7QNUHVibGljS2V5RUNESHU0A2NydoAEWDQ00LQGUHVibGljIjDnbp9FpC
bKqBC81emRiE2t0CYaU-g9tiDxXlEoIkqfivJdZrGDgD1BMQCnbL0VoBobP6Dr-60
JBIB9fX20BVJvbGVzW4AJdGhyZXNob2xkXbQJU2lnbmF0dXJle7QDVWRmgCJNQjRQ
LU9DVjctV0RUSS1LRlk2LUE3VzUtWEEdNTi1JTEtMjBBQdWJsaWNQYXJhbWV0ZXJze
```



```
7QNUHVibGljS2V5RUNESHu0A2NydoAFRWQ0NDi0B1B1YmXpY4g5ju0AZ2RcHBqm9o
YYcRag8h4hPDUzGu0aYyTVrk0XlpyXUMblnSbyJ_Xj5KouRYnm3aOS4AtWZakAfX1
9tApFbmNyeXB0aW9ue7QDVWRmgCJNQlZYLvVYTEQtUEg2Ry1XS1E0LTROTUItnKlH
Sy0zNVFatBBQdWJsaWNQYXJhbWV0ZXJze7QNUHVibGljS2V5RUNESHu0A2NydoAEW
DQ00LQGUHVibGljiDnXexA8QK1De4Ivy5Yaz7n085iB8QW0GgntwgdARK70i90afE
hsyVyXISV887BxPL5rCFpmXsP2CAB9fX19fQ",
{
  "signatures": [{
    "alg": "S512",
    "kid": "MCB0-ZK4F-QFYM-63TK-TA2C-LHQY-7QW5",
    "signature": "7QKcgJS4Ub14gYYjiBP301RC5U-tDs5meh0V_Y
Mc7Rq4PV_I1WzTTZ5wnPPWdXxsS4-gjz1xra4AhMKYT4tN0z3Hc954sJVIAoZDHED
bSxxeSWsp6Fd06hCa5Bt43tYt3TyEmnIHFgUmSFNIwIp0fw0A"}
  ],
  "PayloadDigest": "0stbMttGc4Jpw5qapxrMx_wAdIyJ1ozebqUCmW
SrE0G_bbYLYcdNVYq35C6hesgUUGwAItS3939DgezPSm91xg"}
],
"EnvelopedActivationDevice": [{
  "enc": "A256CBC",
  "dig": "S512",
  "kid": "EBQH-KLCY-C4F3-VF7J-YSU3-EQFU-FZHQ",
  "Salt": "6K4bKrm5-_rjgI4_I08wTw",
  "recipients": [{
    "kid": "MDMW-WCIR-FJMO-7ZH6-CN3J-KUZL-RLAX",
    "epk": {
      "PublicKeyECDH": {
        "crv": "X448",
        "Public": "nvVIGPvOYs0J8IIm-19RbyLkAWivgWgZ5a-0A
Rp9ftWFsoSSAQsXaxWNMJr-X4HRTb65eFiuWSA"}},
    "wmk": "Dxo1wkjSF6tx00kZ3YtNDWzK-e_95sEbZay2cEJnm3aa
Pt-oMy9PbA"}
  ],
  "ContentMetaData": "ewogICJNZXNzYWdlVHlwZSI6ICJBjY3RpdmF0
aW9uRGV2aWwIiwKICAiY3R5IjogImFwcGxpY2F0aW9uL21tbS9vYmplY3QiLAogI
CJDcmVhdGVkIjogIjIwMjEtMTAtMjVUMTU6NDk6MDNaIn0"}},
  "ki85W0Qnh960wNqp5BCXx-pDm5ZyRU7w93rIb9kZQVc00mvkvuiAq28q
zje1vLZ4b-XwiqY8MOCLqRkscz_PL8ALVscz5rBQz4JsCHtD1xCRMqkG0U0YeY205
8lGpQ5MLZgNEeQCx0RALAdNrKRLvdUG6q2G0M9G0PzG9ocyhrI3D4P4T62_KJq9-G
YmyHA7UEIXE04mCDn7dFWeP8lzsnpBTwbIFC4SNY1t0JeudyPR01PQbkvFIn4whoF
lUDgL",
{
  "signatures": [{
    "alg": "S512",
    "kid": "MCB0-ZK4F-QFYM-63TK-TA2C-LHQY-7QW5",
    "signature": "KHnfiH_mZ14FUavWwY5vdhUFqu7WiBjh1hvHy0
HTCQbdw35_6UmLcqTT2bzSxaEAXs18yDTSvB6AK51Q3DvtVXp23AH4Try8p66EvNr
azKWjh4CAr1QJnUdIJnu-Y3ja2WQJaseKm04g2svNAXadhAYA",
    "witness": "TPx0v-bF80f0cYXGh0ZB-yJ5JRgwXYqmwcITxUQb
BEE"}
  ]
}
```

```
    ],
    "PayloadDigest": "Mqa4f-6bQ07E2IYl2hQUqfXTighNhW9KF-WSEy
mbPQMTqfXsXMGrExxUg4KCgPX1EYVt9k-UfdyZBwtFUxFNiw"}
  ],
  "EnvelopedActivationAccount": [{
    "enc": "A256CBC",
    "dig": "S512",
    "kid": "EBQM-73UM-YBTS-SOUZ-PADZ-SVFD-AV7U",
    "Salt": "mX9D0llVVNyFxJ9ocDg4vg",
    "recipients": [{
      "kid": "MBVX-UXLD-PH6G-WKQ4-4NMB-6IGK-35QZ",
      "epk": {
        "PublicKeyECDH": {
          "crv": "X448",
          "Public": "zxVZRX_9E5bQaBCq_T3de_j-6WWQvNH84u03M
G7UfjtP6mDcbDopF3KuzMIebcuch8Lb7UtLz1AA"}},
        "wmk": "rg3puHIMXDoR7ATz0jfxsJUeIwk4Sb2-L5FzwvFVfK-V
Hol4H35yhg"}
    ]
  ],
  "ContentMetaData": "ewogICJNZXNzYWdlVHlwZSI6ICJBY3RpdmF0
aw9uQWNjb3VudCIsCiAgImN0eSI6ICJhcHBsaWNhdGlvi9tbW0vb2JqZWNOIiwKI
CAiQ3JlYXRlZCI6ICIyMDIxLTEwLTI1VDE10jQ50jAzWiJ9"}},
  "gBlhcdXySmUZ3MHsR9XY00pwp4zrqtnfvxnEI3S1Zjq7dsfAPhHoVFOI
MT7ArKsCojQ-Tikc65a4t0my5aDr-s_CQ5QQA1Y6CKhsFW01hjtITxZnL63w8XH59
XrniVqdI5ehtVHR036eiuDDHe97tBrCFUky2pZdbuL09BDm1r96GsboXBT1vuCbXE
Y8jJreAeroNhvJKCDFqq3m0hBC1Cndj1g15_MOPPd3KQ7rry0HoGpwN2DP2kicx8F
S3jGLtuGdIDxXawAR255D-QesHwr5zZyHY3F_FkJ-1ga50xKH0QtU-oRIdivpnYSA
Lt2cBTTZJ0K2qzWkYgMLPRHxtQRIVvzGt80mIncUuoGSmyJpr0-BSR0ow8N573ubC
GUQzrfwHggM36L2UrPBzm50u8bX5IKu4iLjgE9Wq9J1bYuerqqHYJ6csZFtUG1pf
IGQTuNoFJwi49_NzI-vHpo3RBU_aXsClJo3Kgi8Wn4KJsT2rtQuCTbVLmt0Vja_YJ
lqlJJwjmBSDNqPsitwZtvW2uVZF0whFSxXI0L9ZZiyzW0dHccOtJnZkzJiKy4RaGA
1ArpJ2Nm4MLbKvjDb4gJOvcL_a19qF1khPwt56SySzGN95n5FQA8nRq2EvjQdJmGC
OCn1MGFB1NooUWi3dX0eV2acHy8nTavVNEwVQeqWid5D2QxFSmV8cWghYn5nuFsy2
OGNbIH-kNGxjt9W5IbQKowUvhAfpqQ3P-RlNMCEKzxZZioA6YeNG6Hh9kRak2gYAm
Z9LlISS0mu0pvavRb8rXW-mNttPmF1C8g232avdLyZyR9XjckzHBNaz-cVqKraQf
vR_3M-8mIqGKCS_1oRYG00pDIxw9VU0JCE06DFBHEv-2a3JDg8XvyRuZMIlzJFz0
9usilNns0DaygIabfqA_Gs2qe_t4iA0q1a_yBRr0PW2Tba6BuuorLYm0xL9k7EzFs
BULkAHjJIdeixgwerVnmAnFXfXqKsc2tQ7LRRUkMfOQ3UCypn3niYn4B7M7eJ5efq
-xXu4hWSzrdBpxwOUiLjb9ofvClQjkFWvi-VwRtKGIqdk8V-I2Wvc2nS8zzGaK0zU
-8GnCP9eIe-1Jka2s4qPLD6ASA9QBz6IbtYlgi8HmEh-JfVowwVdsSM0zJ-1V2ILx
CN2YJIWV-q49xurbCDog4CFeHLvDPLsJMGYmofzcBHkWl-0H3qee-r6A7m4UuyDUw
9LqyWyGzo0pjDkXCEGp8v31jxHI4P7Nz1T4K2rN1CzYEnVrqYuqxLR3S-7ySnEQWu
Fk4HrX07bQG5j1iNGghQVV5rQ-uk-7-1uFY8GIIUpiBsI75IwbCeaIv0iCI4v4U3Q
vTR8ssV0gIDDuZdvQFRCDgnTzbQda3isGT-fdrGkp5tX94a6ASuIWQ4jC2yqfndW
y5NVkglyPEyVCPYVyaBI-SxSwzPtmc3X1a4GVo7gMmVOALyx-7nMzxerc8HsB-hr
S9bAgyY7V-cCmbQ6VEteR5GVYUx1Nu2w4aXWwU3-yrIx8NXI22Pe9DPcNtj44wySa
BijioVduLwnlg-b5aKxfjfvCJwfqL7kEZE0ioCSLE0y-cWPM3MsM3I0gwfEuy7-8
b-nSMPGYjCtBxOHBxmLt_-srSDjg89fSC4pLN9He2NoamM3B12ZZnItE-8vmQbH8
Wy304nJNPGHJ_W2juLh758V9XqumJHrADHrjU4n-S0jRngrFjLwYVufxtrtGehcC_
```

W2BzNnFhwtBRCA0A7Jjm8tjF0fHH7hxuaFDJKpIXYbJP2LqB8USizq_KRR6i6jsKM
pxc7wG_81b_unXx1cPs9xMFdQwnn_2Izc7jXKJYC6cVN0GH3dQDZfIgL402KoFXzm
7goJ8hE1HqX3trdsEjKC7c1UM-cdxFzNKMrt064I7kjJdVHpswFleKSHwxgQ93sLh
J38mq2FKYQP_rMp40GgWABCTY78P-XZoc6Mev5bBwxN7NL3WlpnrZKTgJGyz920ri
bPQPz4Ma3aZ_3Wqip015r9QDZWjSfZzNGSXpDnXh0uHBtqcqlu87y4LdmHkKgQqZ
PCioP5KtpgusscKBbIhkQThe4oe7zUEiRFLd8ff8fp0LImOnmo0WDrMgS3p96driz
9d9FaUW0FP0rK20nWVZ9PhvqEzLERr99WVTJ68ZITE5NoaF10wLGdgWFYs610VJoQ
jsXZsT0dmR0Zj-vl6yei8sVZjFLosjst9uh0Wv7k_URAUee95p-Ty2ZLjmwHweWNS
Siw4Qk_LwlkxRxSuuad9RfGum0yduRsKh3I8qerv0PmHz0WD4ZHAqa8c9ZLitvX51
WC0BeLD2iaExGdjThfn03sJh1CC0R_SqbCV6xSmkcfN_iCEKfE_d-3Twwp2cVKJ2l
gEvTwsbYX4PkbI_P1SzChkr1C6sn1EBsk7-tnBngnU_OVhUIq2Dxjv9KrrR3pa3DJ
3s471xfSZpNywnmjxFFv8uPaczmBPf_5BSC5ikub2nT5UTfMUN1AERh5XoDmDZJsE
seikGCN0yFh0ZLdiLYgL6cUpFw171QZILt7v_1a6e2wKw6gMc_9k74qTEISlLyuVP
tNz7PryNMyVG4aEwKQMB0jHqUN5R8zG3X-Kx0TD7PCV9ZtgKiP9SvdzBLzN9ww5Y1
3zGvi3f256i5v0Ie073nE-mEeTDtBdH2WTjHUPtTmTwvH1TruHJlMhV4W5Zs2XjjE
_MsrFvX5en13koNMwAFu0TgPpLZI9xKu60wXwwcc9XHPQVUPUJK2qAWD1R6Dms2kd
hmXoLC6Hia5w9R9jSnei4SNyvZafkKH1peZHHamxeHBEZ0bARFnxFXq77zp9H9VqF
pALocgAYXbuUBU6FHPIWFX1-xR4IGjswPgrKq6lA2QKsf9D7d2hheMrB0KZR5k8fV
K5T5kRhICgu9NxErQkvf2Rjfgk707lMmDdE-g8L27AxNqdku0wxRGFIUxcaVutXAB
5cHH3pKrIJfZfEfAggDooU8RMUI20J6IVZxyZ8nsTdT9ECZuoolAoXh1C4_w-zSpr
5A6WTLzYTLpnyChxueFewD-qQWjzXjeN4b-aqJUc089xEZ4AfrGE8xMeIwp72GzKE
vMhXfL5v-zHOFJUJ-yenAkR0t33qTZH20lXMFvqIrtW_wTW7-y-B7wD9RHI67tooH
3pm6QYFTkUdJu0nL6FV9WN28zBVPRuS0mJEm_MJXzUr5fLY7SCDbGOAbxy-a8B_yQ
40xDftA7FNU4Ylugg2Si5NNBwoXCawfMf0wy_0IMZhXudDTgI30LwNqpLH1hF0thv
DKGJn4qPF-wOu5QaeRdwJgpi0E_XwQI2Xsx80Nyb7xGysaY04xtEiZALwCH057454
jnbJpAdoOAMLZHRi0khYNoyjp_SRxZqr3pB27SFzCz1P5s7EnaSd79hIfKY7Ii5p5
JjTZRKcVMF0Ger9CuuRPccR0hS6y3yCBL5VUQ_0hKmWYPJ9vE3mRz7h037cBjKvBB
eiB0xT-EpNko9JWhtvd0jDnUPK2W2Q75R8qpvvLSX2jWibqsdrvBm2TkTXv4jx0JZ
I-YBZZ8AmA-AQlu8-oEafCHvL65sgu9qpmTgt0NI6Q0_7Fbu2WKeUuSTugbL7q2EN
u1jQAzz2itiF9GZ9p4ol93FjsK0tirjyoKlqSsEAA_fBETnTzmlhblMTRwNEwoVQ1
_kygDezz0fuklKICAwLgHL6wLc_9oI3gGyS2W62dRvsIDcz4mmMsMkJizqdDNxKBr
TJrkpWJXu1Kozk81xjbECG8Rbic_QGwbC7yyaiYxU9lazzb7VMnlNskWitx4GxgXN0
nKISYPH00jnZLMYtvhog3mbIqqNrcJGWRmEl7sRJa2_k84FI-MOEESYxaCndq0CwS
atguRgg0LY8wVFNkdAUDCT-T1j43eTVcKpQGrTB6rDxhAClHkFfwW8bIrRenLmxMW
HplKju7xbjCaj-3Usz6v5p_-0TudlFsoPcEU3Cb_b4LgvyvZA_stq1TbQoyctrVQA
OcSxFX2-aImyJYuyKkyMNTKmQNgv62f6h111Q034vPZqrLgArTgpnQhu5eYEgymfU
empy7SRLkwQdywm14jeImqtZ-AfzM3Test1FsU2xGaMs1xrQz80h2J8FHZ0kxTAP
CUs68MNOXRUCyZUvRespSXCPzisFFmZWJkFAFYayWVHPRxcWADHTJWKlc17NS1j
b5pprmqCynRE5AbPIeDMEGx96r57nIb63l6TadwQMFiQHDiaH0dP1EAREPT9tYURx
JelgcpNdqWELXkGGkUaBpZoF0Wh3aHJMd3beMG1dZaMkY3hMCagpeoArs4w96393w
gQik5AFeh22kjdF2PaRAvakmMsZ9PvvyHzTRVKj0kUyegloQH5Ep0heYfmXcQL2Hd
p3_LMAIfZtT9EJ7avVQCftCaDVhAuJtVA0TUzZUTw9y5D0FXoCD8fQ0NzeAywhP-G
XK3Bm3duH-T9BLXN3NC3hbKKAJzVkgJuVDihWuMB5aXKj_R65vLixgx8Z4V1qYjQf
p9vIcENaDKT0j6qRjlnwwKdPLeJQfJBqBvCU7RBXZr7k5mCLsBmij3GZWYvpb6FZ2
ekxJ0902Xov1-EcOPATjqFvo2ZqhLqKzGzvAa8l-blua10eF5000Yd32uk5WpY7X1
kcSjE8XKfCPmh-sVkrdxTV0bB-1mwbRkcofJh2SFBTV12D9MgP13dybD3hh2Ab5fE
02rc1kJx4kke4p8LykCpRP8KP9T1ZRmy03GxpCjW-Q7_165ZciX6JCs-_0QoTgn9R
9DBX2KcvoBEH0Cu5GJczfd_j6dS5KgLz_-sI3xLFjaKEyRdpJHlfnMv-ca9m-yCZr
17Rk_kHb-IxPa6DA9t-EwNvE4PKGbt9WUqYPDBUV2hJyQrIevSEqH84sSmY0ycvjN

-ORwv1JwE0okT7uJ7TRgxf2eSGMkXJQx_FVnsj5pz_EPxMj72zAX1r5mUvFBEz1ce
Q1vwcc1H08_JPASKK_e1hiq7nABMyfdfAR48HKdzDxFCVd4e87oeqsQrYeaLm4nM-
ojNYBQJmbwwhewiRai-CK_K3Q-PY1bhPvmzIbp2yE79EawtCNBTPn3myWo79ejxdY
CekhJRdevRCPsCKTg3Yuj8m59IPCKG7-p_GFFmPnF2zu7QAy11NID7wbr16L5xqJC
r3YYu-xHEbN8ovNXbusNCTKEBss2Evn-XS6zq4Spci8g0H7V5Pv9MK8G0MuPkbssc
jkqc5IqsG-YVAi0tI0bn6RCqqwoQV16QfLpY0NNP7KMBjRxsspERxBd05o46xAuzZ
jYs-J0SbR3vy8z6-1IAJEPVw2po8DnuXccwrMns2p_unIiwWqHgeUbau6MsE06aA
BaboDrE-aqQh1b0YcTgZmcznF0zghm1Z9X_Hoke058KT9uVR39h6o-ZPzctnN-e8c
U07EwjWUkyGgUoYNI6MWW9J_z_pUSBYdydj2m2jQchV7CcdN1UBBlid91Zb0QZpwwJ
_RcUfTgV3LvUnUT7SxNZL2QAZf1TwXQGrVmEUzhPUxeu7m4WgQ5kTr0poF2T2Xt8p
Qv1CnzM1z9y0akQtF8bkQfdwj_JJTMr9EGRrnkq99xmHudyxPwhSx9RzhPJPqEK2
qiQxAs8Uj60m61nDoyqmCRpjz5u5WG-1B6EMXATOVnj6Mkky1smwb-ZiB0nPS3n80
m2Ua58ULY1_e5UpPsdPZDRqN2ypTgy3IFoFoXf6RR1L32us3-SoYLoqdZR05tDzcn
ARz1rRVomW7CoVEShhP_I0cpy4QvK66_K3dVLQnrGoFsJHAD2AZ41f4yyQ0qy7NVT
C8qTVyC4tJ23rQ14E9UZ0FgnK6REMFwq-UhEurVTZWwbLaGq10H_cAFk9DXRwbVM
Z_ty20Gd0e_JKFQ8zSZJxtqWFPzb2hVINzLhgeB0YfH-Exmew4vy5THHhvFKAIL9h
5CKGCqztEyUIpsZLOkN0EYDHZrN2oshiq-BACnoMcE9q0z18JeNxxKnw-_u3pUEFP
JNyvWpROA3if7-1Eu7b06Rh0kmDn1U0f1kBWwsWDQkA6tdD6hwiYg1U36XuTLutD_
FfR7IBnQqBx_QWJJzLnMyz_nkQHki1u1lWHHab0zuVbcSAH4P5M7P4Z6XKE0oLgG0
wGFgK5MJYOC2wSGBgXIIdbr-GJGp4wCZ2mGZ9zE8DW84cZWys-Nv5rmrgrcPUeSK9W
MXDj-T5LiQo3y8lgHu0yNjGakYM4M9ysT-XJBHNxYC1a5SxJtAM8IXax-ba-u0tby
nkV7X5ANub1K03bAU1mDgndmtA3kq77ykGWL1TjT5q805bTe3mxWwoT724EV_rUD1
Nba99vzfRVxtyNDxBg0ZsBJUP4IghMZJm--bQ9jq28MGduAhtgicra_hz0Hs1kuWw
1qh6vsmP6RmR7R4_yDNoQ16ccHz3_DLzd2oAlWCIJ9Mi1xuP041zJ6d42Pv7w9tdb
vDw1avTq_1n2JF2pnC-nC6frxXepuIG0eDdq1bVBLrrq0LG_HiBEZqVZFfFOMY4et
SNcUujrtzBGqVCJbLbrkairiozJi0fw324VKW2ESFgiKjmw_gmMckQAqwsXsAKdib
VP_pwQ9C19ZJGqVchgFxmXbfcZsmPSkajWHg2Y1Jc0EE2vXZsN5e6dkjm1Xck2Ng
xjRxMg54-v0_eA9UJkX8EIGeFHGv1qs43Xoi4_3JqUy2qwhFI-aPuoioC0Rf-nXS4
Ahudwcqh3uCN-JwrI4dRG4XmFhniEFS8P04-ZPB0J-xQNGtg8qZxxSueoMIMSITKi
GN86Dd1tPnWBU1f5EWz0x1EOZuhteeaIkRwB1l-pBSWA-2jiHadqetS7LqMsDYeBk
uwTehlCqWXTXuGEENiRLTwtjF6MTV84kkUuN8WsnKJ87f0Rr9oUwbDwjCJhFfV5T0
3xWZbvL97I-ydU6h-45QoTW2myGD_OdKHZT9jEgGNJXmmdMYECAjUUDVu01-wecfT
VqytzQ_aYb2YbtrHwMY5D_gSj6WCqCxmnl3coPxrMcbyaRUHwzxouN9u6A2fEq-Xg
pXD1oJz1amm15NRxaYGTvAgz9wqxMyQkqsLrMri3qeGVAX1ZGm40JREo5gbw7eSgK
Dkz1a604Da6aN1GvcTtpP4oM0Cu2IsFNII2nmqawJ5iaG_U1UtD_3uwX6bjly6YAy
m5q4YNprmkYFFwyJIPFmtd_B4QEoZ65MTcp_EUDtzFq12TLv8GKpQmvySkammIVFA
A3eHDDusfQEi6CyaCs_DG_wmtTQkSncCmJBvWAKgkZgPkWtXt0Smj0S0xH-L250mH
3DJhG14PvsV-zMvcsOphb8oENH_BZVeolpbHCL_YBFibp4N3PlbJHZdhMLB3wg04G
JLs8Dz0Jl21DL7m3F997TASwyDdDymk_LzjND-T8V7YYQVVT0mxRnK9jefTF_C4YR
m1YhN-6HA5SKYtMLRSMNnyWyYgeUMCK9m2jXW-AMc111Ikrshhb_JR8sjGR5A97_R
qVc1AYBmaCmHiCh1b39YzhEWovNQDvTfqLwlgM6WXXN7BMrDmv898t028AEDQaRJ
dWSCwNXq2Vs7P43gr-Vo0zf6ZErqs6S1grgPaqgpZqoKS1GbH84d_RcmDQqzhsXA0
cR1g6qgj3w6WJd-Ft8H-JxJjE0nuNpwgVYj2Uht1-3kXC0kUTHGCZL6Wx9p4kqxYa
GQI5MVzYg7Ao1cPjGagR210UdD5nTJdkputAIDZZ73PAFJ9H1JuZPhH7kD8eJQJyy
Broh_pnBXpgFi_6SZVJCh5NJWErajfFibPBePkpDrXVYehFzTJw1Jzsq-rPbL_oIo
2N08_i7GpDjSqliDI2ENkEvdMrX6zrIWoP-FbFT9U-xij4FQWiNzg4Ww7c1d7hU81I
4gJQSY5SvZ6eSdx1SYc_6C165E3F5eN2FBSUi3kPy-jIwmQVJ59UniV8eCCBetPdp
nT40gCQaytamAOIATcP8IK5HRQQIksmgcZxtYITyiE0_ZK_30sSyo_Bft-USwa8MF
0w5cEQTBsKfyGYxOr_8Wo_4N_rrEvbvJTDMMow5BTcvE8vq4r2FnXNFC4xUdy5Lk5

```

HG7XKNUKeZQSTkwP0Ry95Yzw8Be4qovtYmm7UkGiWdNB_abRQGnb3bQcrZv5GX0H7
TlDUE2m0n9BcG3W_oqCESdXpNL-w83u-puP_vF19I_ma0kUtpyUWJXXj0kRd8c0WF
osuYsGg6sIy_nC2ErgEQPDvuAL0cq0Qkc8gX9SckW0W3v0PxAkQdYN3eMt-k13Afs
y7IePBNP_hXordX3WdkAK-55A7JuPBN7zW5MqL96jvLCJk_10ms66e2Ae5MK73Wp1
HXQl3IQ7JwCAqXdb-u0MfdViIs6a47y7EXJikZNacrKthzunoyD1F8cZ6wL4ecRTn
Jsl_2jstlwKcZEuEcvZGAvPqzI19qhYqEw708MOWQMS8cDAUrAb0etfbSmDNwwQR3
RgaMM6BuevUgFBdVAvf_wvI9994rQRDroeipduhis8xvFDVD_pH_ZpJEl6eQpijxA
tIdtdap3kwib1gylmAxJ6-zfIm0BJo5_aBszj0JWiMmB3Cd0",
    {
        "signatures": [{
            "alg": "S512",
            "kid": "MCB0-ZK4F-QFYM-63TK-TA2C-LHQY-7QW5",
            "signature": "V5h7pFoycR-2WNj7rS08cyfzbrMHW8GeyME3Wp
wnFW9a0X1f1FbKuRAQzgo6CAQR9Cm0ffm92lqACM-DA_Rylu0zKWbq3PK3u2iRrJt
YZ4RyqDam8DoaDEejYJTz7CtK0h88q62L1iF7QirYhqLcWicA",
            "witness": "KqVr0TlFoo8dHQd3Slg-eowe2e_90aoPHfDB4Duy
Fo8"}
        ],
        "PayloadDigest": "3-0c-MhvBsyqxImZbVh0iLXXtd8Av4Rj8C-z0F
B0utKwzhvOdQ_x5Y1DV7tEM1HqeIZnzhdb6onFEVlSdSvA"}
    ]}]

```

9.1.4. 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 = MCVN-XLLT-LLNW-U4HR-BOMG-RA6Z-UWRR
Account = alice@example.com
Account UDF = MB5I-R24M-QXJT-KDBF-XFOA-DGC3-U3AA
Alice3> account sync

```

The completion request specifies the witness value for the transaction whose completion is being queried:

```

{
  "CompleteRequest": {
    "AccountAddress": "alice@example.com",
    "ResponseID": "MDT3-TM62-G3X0-ESY0-WQZX-IR2B-YNHW"}
}

```


The Service responds to the complete request by checking to see if an entry has been added to the local spool. If so, this contains the RespondConnection message created by the administration device.

9.2. Preconfigured (Static QR Code)

The preconfigured device connection interaction is used to connect devices that lack affordances such as a display or a keyboard. It is also known as the static QR code interaction because a static QR code printed on the device itself is used to connect it to a user's account.

Future: Note that this interaction is likely to be changed substantially in future revisions of the specification and the Claim/PollClaim mechanism removed and replaced with a messaging based approach.

The interaction has five 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 a Claim operation described above as described in the Device Description.

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: Poll Claim Notification When connected to a network, the preconfigured device periodically attempts to poll the connection sources specified to find out if there is a pending request. If a connection request is posted, the device decrypts it to allow it to complete the connection process.

Phase 5: 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 main differences between this connection interaction and the witness/PIN connection interactions are that the device is preconfigured with the device profile at the time of manufacture and the onboarding device **MAY** be acquiring network configuration information during the connection process.

9.2.1. Phase 1

The manufacturer preconfigures the device

```
Maker> device preconfig  
Device UDF: MDDT-KTDT-AZ62-55HV-FFVY-JYNU-Y3YE  
File: EC6P-K0IX-T3B4-YIKE-OLX3-BUUD-64.medk
```

This results in the creation of a primary secret which is used to compute a ProfileDevice and corresponding connection records signed by the manufacturer's administrator key.

The data is combined to create a DevicePreconfiguration record that is provisioned to the firmware of the device being preconfigured.


```
{
  "DevicePreconfigurationPrivate":{
    "EnvelopedProfileDevice":[{
      "EnvelopeId":"MDDT-KTDT-AZ62-55HV-FFVY-JYNU-Y3YE",
      "dig":"S512",
      "ContentMetaData":"ewogICJVbmlxdWVJZCI6ICJNRERULUtURFQtQV
o2Mi01NUhWLUZGVlktSl10VS1ZM1lFIiwKICAiTWVzc2FnZVR5cGUiOiAiUHJvZm1
sZURldm1jZSIsCiAgImN0eSI6ICJhcHBsaWNhdGlvbi9tbW0vb2JqZWNOIiwKICAi
Q3JlYXRlZCI6ICImYMDIxLTEwLTI1VDE10jQ50jA3WiJ9"},
      "ewogICJQcm9maWxlRGV2aWNlIjogewogICAgIClByb2ZpbGVTaWduYXR1cm
UiOiB7CiAgICAgICJvZGYiOiAiTUREVC1LVERULUFaJiItNTVlVi1GRlZZLUpZTlU
tWTNZRSIsCiAgICAgICJQdWJsawNQYXJhbWV0ZXJzIjogewogICAgICAgICJQdWJs
aWNlZXlFQ0RIIjogewogICAgICAgICAgImNydiI6ICJFZDQ0OCIsCiAgICAgICAgI
CAiUHVibGljIjogInF0TDhCYVN3UUptNk12bE1BUXY0MkpsSk9MWFZMY0gxTWnweU
p1SwxJazhXbVpvYTLHd2MKICB4WjFIMmI5VE5MZGFZUGp1VlVawHRkb0EifX19LAo
gICAgIkVuY3J5cHRpb24iOiB7CiAgICAgICJvZGYiOiAiTUFUCQy1MR1k1LUJVMk8t
U0FaTi1ESjJFLVMzQ0ItQkc2NSIsCiAgICAgICJQdWJsawNQYXJhbWV0ZXJzIjoge
wogICAgICAgICJQdWJsawNLZXlFQ0RIIjogewogICAgICAgICAgImNydiI6ICJYND
Q4IiwKICAgICAgICAgICJQdWJsawMiOiAiWDZUaF9IOEJZOC1zRHpydWNVV3F4S0c
1YVloenhTVC12dDE5STlKOU83TmlnRGYxZmhcQogIGZCT1pWwk9uUDhYNVdTMkJJ
WGQ3SjltQSJ9fX0sCiAgICAiU2lnbmF0dXJlIjogewogICAgICAiVWRmIjogIk1EW
lQtREFFNC02TkJQUJSQ08tUzVUTC01Q1E2LVNDWTMiLAogICAgICAiUHVibGljUG
FyYW1ldGVycyI6IHsKICAgICAgICAiUHVibGljS2V5RUNESCi6IHsKICAgICAgICA
gICJjcnYiOiAiRWQ0NDgiLAogICAgICAgICAgIClB1YmXpYyI6ICJpM1hia3lpT201
Wn1XawxBeU9DZnFua1BMaUtVLUGyNTJZVUdqRVd3MWgtZ2haR3Nkb09aCiAgcXRkQ
0k4Q0hRYWtZS3JHTWZdDdMxbjRBI19fSwKICAgICJBdXRoZW50aWNhdGlvbiI6IH
sKICAgICAgIClVbVkiI6ICJNc0k1DLVE3SFctNUL0Sy1RU1pPLVBLRFETs01aNS1BT01
GIiwKICAgICAgIClB1YmXpY1BhcmFtZXRlcnMiOiB7CiAgICAgICAgIClB1YmXpY0t1
eUVDREgiOiB7CiAgICAgICAgICAgICAiY3J2IjogIlg0NDgiLAogICAgICAgICAgIClB1Y
mXpYyI6ICJSX08tZnplUnp4aExsdHh1Nko5VG05MVNHSWFCY2g0LXFfNnFwNTZ4WU
YtVTZqa0hSall2CiAgT2hjNm12OUdLOVhNUjZtVFNOUEstV0tBIn19fX19",
      {
        "signatures":[{
          "alg":"S512",
          "kid":"MDDT-KTDT-AZ62-55HV-FFVY-JYNU-Y3YE",
          "signature":"VFD-9f8AXHdm38HR7y7JKsPStGNRu7wW5SXsJgc1
lbRyzQ0XVyDyNtqR5e19TCEuJKC0vU4lq4QAQfzJlUaa-viM7xhTcvJhVZ_YGiYEW
wq3Nb1-sortDNudi7FGmG9C5Nh-ErWxy2oKkH8Nht19LDQA"}
        ],
        "PayloadDigest":"PRkvfQ8djpN_Z3tY_p8qPRR4rTy_ZFEFW_WAqBcQ
2WpffnNZf_dPVKtW1XW9IpGjxYg2h0zB-hSVnCWViSUiEQ"}
      ],
      "EnvelopedConnectionDevice":[{
        "dig":"S512",
        "ContentMetaData":"ewogICJNZXNzYWdlVHlwZSI6ICJDb25uZWNOaw
9uRGV2aWNlIiwKICAiY3R5IjogImFwcGxpy2F0aw9uL21tbS9vYmplY3QiLAogICJ
DcmVhdGVkIjogIjIwMjEtMTAtMjVUMTU6NDk6MDdaIn0"},
        "ewogICJDb25uZWNOaw9uRGV2aWNlIjogewogICAgIClF1dGh1bnRpy2F0aw
9uIjogewogICAgICAiVWRmIjogIk1BQkMtTlZNS1CVTJPLVNBWk4tREoyRS1TM0N
```

[illegible]

```
"PrivateKey":{
  "PrivateKeyUDF":{
    "PrivateValue":"ZAAQ-APQL-QS4L-SY3L-RER2-TYEA-V4EF-Q30B-6N2
F-DKDP-UJQ6-KXUN-LI2H-7RXH",
    "KeyType":"MeshProfileDevice"}}},
  "ConnectUri":"mcu://maker@example.com/EC6P-K0IX-T3B4-YIKE-OLX3-
BUUD-64"}}
```

An EARL is created specifying the means by which an administration device can acquire the information required to complete a connection to the device:

```
QR = {Connect.ConnectEARL}
```

The preconfigured ProfileDevice is encrypted under the encryption key and published to the location key derived from the EARL.

9.2.2. Phase 2 & 3

The administration device scans the QR code and obtains the Device Description using the Claim operation as shown in section \$\$\$\$\$. The administration device creates the ActivationDevice and CatalogedDevice records and populates the service as before.

```
Alice> account connect ^
mcu://maker@example.com/EC6P-K0IX-T3B4-YIKE-OLX3-BUUD-64 /web
```

9.2.3. Phase 4

The device polls the publication service until a claim message is returned.

```
Alice4> device complete
Device UDF = MDDT-KTDT-AZ62-55HV-FFVY-JYNU-Y3YE
Account = alice@example.com
Account UDF = MB5I-R24M-QXJT-KDBF-XFOA-DGC3-U3AA
```

9.2.4. Phase 5

Having been advised that an account has published a claim to bind to it, the device posts a connection Complete request to the specified account and completes the connection process as before.

10. 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.

10.1. 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.

10.1.1. Message: `MeshRequest`

Base class for all request messages.

[No fields]

10.1.2. 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.

10.2. 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.

10.2.1. Message: `MeshResponse`

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

[No fields]

10.3. 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.

10.4. Common Structures

The following common structures are used in the protocol messages:

10.4.1. 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.

10.4.2. 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.

10.4.3. 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

10.4.4. 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.

10.4.5. Structure: ContainerStatus

Container: String (Optional)

Index: Integer (Optional) 10.4.6. Structure: ContainerUpdate

Digest: Binary (Optional)

Inherits: ContainerStatus The entries to be uploaded.

Envelopes: DareEnvelope [0..Many]

10.5. 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.

10.5.1. 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.

10.6. Transaction: BindAccount

Request: BindRequest

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

Attempt

10.6.1. 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.

10.6.2. 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.

10.7. Transaction: UnbindAccount

Request: UnbindRequest

Response: UnbindResponse Request deletion of a service account.

10.7.1. 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]

10.7.2. Message: UnbindResponse

Inherits: MeshResponse

Reports the success or failure of a Delete transaction.

[No fields]

10.8. Transaction: Connect

Request: ConnectRequest

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

10.8.1. Message: ConnectRequest

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

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

10.8.2. 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

10.9. Transaction: Complete

Request: CompleteRequest

Response: CompleteResponse **10.9.1. Message: CompleteRequest**

Inherits: StatusRequest
AccountAddress: String (Optional)
ResponseID: String (Optional) 10.9.2. Message:
CompleteResponse

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

10.10. Transaction: Status

Request: StatusRequest
Response: StatusResponse 10.10.1. Message: StatusRequest

Inherits: MeshRequestUser
DeviceUDF: String (Optional) 10.10.2. 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] 10.11. Transaction: Download

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

10.11.1. 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.

10.11.2. 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

10.12. Transaction: Transact

Request: TransactRequest

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

10.12.1. 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.

10.12.2. 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.

10.12.3. 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.

10.13. 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.

10.13.1. Message: PostRequest

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

Accounts: String [0..Many] directed.

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

10.13.2. Message: PostResponse

Inherits: TransactResponse

[No fields]

10.14. Transaction: Claim

Request: ClaimRequest
Response: ClaimResponse Claim a publication

10.14.1. Message: ClaimRequest

Inherits: MeshRequest The claim message
EnvelopedMessageClaim: Enveloped (Optional)

10.14.2. Message:

ClaimResponse

Inherits: MeshResponse The encrypted device profile
CatalogedPublication: CatalogedPublication (Optional)

10.15. Transaction: PollClaim

Request: PollClaimRequest
Response: PollClaimResponse Check party making claim

10.15.1. Message: PollClaimRequest

Inherits: MeshRequest The envelope identifier formed from the
PublicationId: String (Optional) PublicationId.

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

10.15.2. Message: PollClaimResponse

Inherits: MeshResponse The claim message
EnvelopedMessage: Enveloped (Optional)

10.15.3. Structure:

CryptographicOperation

KeyId: String (Optional) The key identifier

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

10.15.4. Structure: CryptographicOperationSign

Inherits: CryptographicOperation The data to sign
Data: Binary (Optional)
PartialR: Binary (Optional) Contribution to the R offset.

10.15.5. Structure: CryptographicOperationKeyAgreement

Inherits: CryptographicOperation
[No fields]

10.15.6. Structure: CryptographicOperationGenerate

Inherits: `CryptographicOperation`

[No fields]

10.15.7. Structure: `CryptographicOperationShare`

Inherits: `CryptographicOperation`

Threshold: `Integer (Optional)` **10.15.8. Structure:**

Shares: `Integer (Optional)` `CryptographicResult`

Error: `String (Optional)`

10.15.9. Structure:

`CryptographicResultKeyAgreement`

Inherits: `CryptographicResult`

[No fields]

10.15.10. Structure: `CryptographicResultShare`

Inherits: `CryptographicResult`

[No fields]

10.16. Transaction: `Operate`

Request: `OperateRequest`

Response: `OperateResponse` Perform a set of cryptographic operations

10.16.1. Message: `OperateRequest`

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

AccountAddress: `String (Optional)` to

10.16.2. Message: `OperateResponse`

Inherits: `MeshResponse`

[No fields]

11. 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](#)].

12. IANA Considerations

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

13. Acknowledgements

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

14. Normative References

[draft-hallambaker-jsonbcd]

Hallam-Baker, P., "Binary Encodings for JavaScript Object Notation: JSON-B, JSON-C, JSON-D", Work in Progress, Internet-Draft, draft-hallambaker-jsonbcd-21, 5 August 2021, <<https://datatracker.ietf.org/doc/html/draft-hallambaker-jsonbcd-21>>.

[draft-hallambaker-mesh-architecture]

Hallam-Baker, P., "Mathematical Mesh 3.0 Part I: Architecture Guide", Work in Progress, Internet-Draft, draft-hallambaker-mesh-architecture-18, 20 September 2021, <<https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-architecture-18>>.

[draft-hallambaker-mesh-rud]

Hallam-Baker, P., "Mathematical Mesh 3.0 Part VI: Reliable User Datagram", Work in Progress, Internet-Draft, draft-hallambaker-mesh-rud-00, 5 August 2021, <<https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-rud-00>>.

[draft-hallambaker-mesh-schema]

Hallam-Baker, P., "Mathematical Mesh 3.0 Part IV: Schema Reference", Work in Progress, Internet-Draft, draft-hallambaker-mesh-schema-08, 5 August 2021, <<https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-schema-08>>.

[draft-hallambaker-mesh-security]

Hallam-Baker, P., "Mathematical Mesh 3.0 Part IX Security Considerations", Work in Progress, Internet-Draft, draft-hallambaker-mesh-security-08, 20 September 2021, <<https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-security-08>>.

[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-14, 20 September 2021,

<<https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-udf-14>>.

[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>>.

15. 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://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-developer-10>>.