

SZTPD Administrator's Guide

Watsen Networks

September 18, 2020

Abstract

This documentation is for the [SZTPD](#) product by [Watsen Networks](#).

This documentation is still a work-in-progress. Some sections clearly indicate when material is pending, but there are also some missing sections, and other sections may not have enough detail.

SZTPD is currently in its “alpha” stage and details captured in this document may change.

Contents

1	Introduction	4
2	API Introduction	4
2.1	Northbound API	4
2.2	Southbound API	5
2.3	Eastbound API	5
3	Getting Started	5
3.1	Fetching Host-meta	6
3.2	Fetching the RESTCONF Root Resource	6
3.3	Get the YANG Library for this RESTCONF server.	6
3.4	Get the Current (Default) Configuration	10
3.5	Configure an Administrator	10
3.6	Create a Device Type	11
3.7	Create a Device	11
3.8	Again, with a Single PUT	12
3.9	Initializing TLS	13
3.10	Give Server time to restart	15
3.11	Ensuring the Ports are up	16
3.12	Simulate Device Trying to Get Bootstrapping Data	17
3.13	Configuring a Redirect Response	18
3.14	Simulate Device Getting Redirect Information	21
3.15	Delete Redirect Information	22
3.16	Configuring an Onboarding Response	22
3.17	Simulate Device Getting Onboarding Information	26
3.18	Simulate Device Posting a Progress Report	26
3.19	Delete Onboarding Information	27
3.20	View Audit Log	28
3.21	View Device's Bootstrapping Log	30
4	Northbound Interface	33
4.1	NBI Overview	33
4.1.1	Administrator Accounts	33
4.1.2	Audit Log	33
4.1.3	Boot Images	34
4.1.4	Bootstrap Servers	34
4.1.5	Configurations	34
4.1.6	Conveyed Information Responses	35
4.1.7	Device-Types	35
4.1.8	Devices	36
4.1.9	Dynamic Callouts	38
4.1.10	Keystore	39
4.1.11	Preferences	40
4.1.12	Scripts	40
4.1.13	Tenants	40
4.1.14	Transport	41
4.1.15	Truststore	42
4.2	Advanced Features	42
4.2.1	Triggers	42
4.2.2	Tracking	42
4.2.3	Dyanmic Callouts	43
4.3	Complete Tree Diagrams	43
4.3.1	Mode-1's native view	43
4.3.2	Mode-X's native view	50
5	Eastbound Interface	59
5.1	Inbound Facing	59

5.1.1	TLS Terminator	59
5.2	Outbound Facing	59
5.2.1	RDBMS	59
5.2.2	Dynamic Callouts	59
5.2.3	Notifications	62
6	Simulator	65
6.1	Overview	65
6.2	Dependencies	66
6.3	Downloading	66
6.4	Unarchiving	66
6.5	Customizing Variables	67
6.6	Initializing the PKI	67
6.7	Running	67
6.8	Cleaning Up	68

1 Introduction

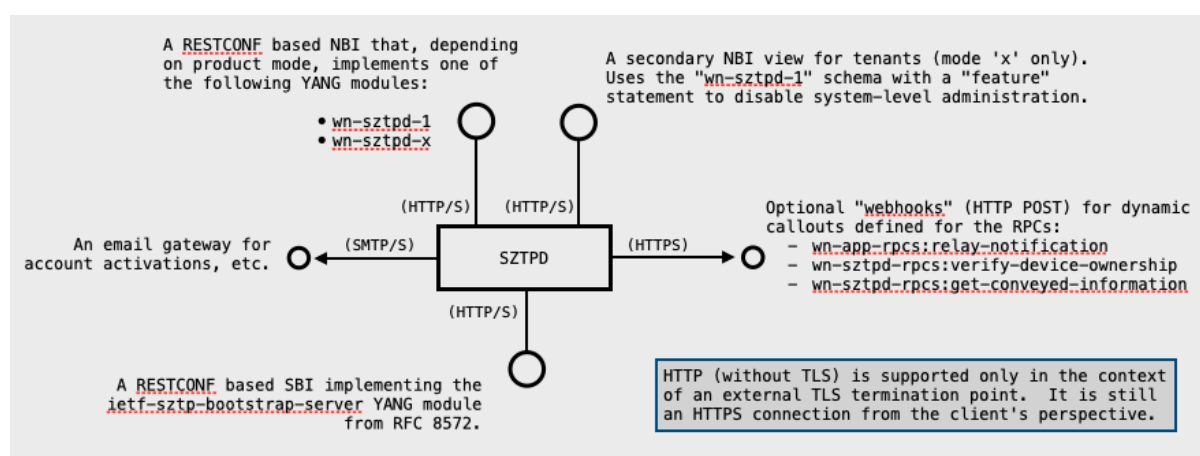
SZTPD is an implementation of a “bootstrap server”, as defined in the [Terminology section of RFC 8572](#), also known as an “SZTP server”, as defined in the [Terminology section of draft-kwatsen-netconf-sztp-csr](#)”.

SZTPD is provided as software, an asynchronous event-driven Python-based executable. SZTPD is an application (not a library) that includes a northbound API for configuration, a southbound API for device bootstrapping requests, and eastbound hooks for integration with other business systems.

This documentation is in preparation for a “1.0.0” release, using the common “major.minor.patch” [semantic versioning](#) convention.

2 API Introduction

The following picture illustrates the API interfaces implemented by SZTPD:



The various aspects of this picture are discussed in the following sections.

2.1 Northbound API

The [Northbound Interface](#) (NBI) API is the largest API presented by SZTPD. The NBI is the interface enabling the setting of configuration, the viewing of operational state, and the reception of notifications (both system events and bootstrapping progress reports).

The NBI is a RESTCONF [RFC 8040](#) based API that implements one of the three YANG modules:

- `wn-sztpd-1`
- `wn-sztpd-x`

where ‘1’ and ‘x’ refer to the SZTPD product mode. Each product mode has a distinct (though highly related) data model for the NBI.

The scope of the NBI is extended to also include the outbound interface SZTPD uses to deliver notifications to northbound controller/NMS applications (i.e., implementing a “receive-notification” HTTP POST request).

The data-model for the “receive-notification” request is formally defined in the YANG module “wn-sztp-callbacks” for specification purpose only. It does not imply that the northbound controller/NMS application must implement RESTCONF. Integrating applications only need to implement support for a single HTTP POST request (“bootstrap-complete”).

See the [Northbound Interface](#) section for more information about the NBI.

2.2 Southbound API

The southbound interface (SBI) implements the bootstrap server API defined in [Section 7 of RFC 8572](#). The SBI is a RESTCONF [RFC 8040](#) based API that implements the “ietf-sztp-bootstrap-server” YANG module.

The SBI is fully described in [Section 7 of RFC 8572](#).

Device-agent developers need to be aware that SZTPD currently only returns, in its `get—bootstrapping—data` RPC-reply, CMS structures that are neither signed nor encrypted, but these are planning to be supported in upcoming releases of SZTPD. It is strongly suggested that the CMS-processing code switches on the top-most content-type's OID:

- if `id_ct_sztpConveyedInfoJSON`, then
 - this IS supported in SZTPD 1.0
- if `id-envelopedData`, then first decrypt with private key
 - this is NOT supported in SZTPD 1.0 (but plan for it!)
- if `id-signedData`, then verify signature using owner certificate
 - this is NOT supported in SZTPD 1.0 (but plan for it!)

Device agent developers should also be aware that the SZTPD's RFC 8572 implementation currently only implements support for JSON-based API. An XML-based API could be introduced relatively easily, but it is a low-priority item pending market demand.

SZTPD's SBI is unaffected by SZTPD product modes.

2.3 Eastbound API

The so called “eastbound” API is for product integration into backend business systems. The Eastbound API is unaffected by the product mode ('1' or 'x') SZTPD is using.

Three API integration points are defined. These API points enable SZTPD to:

1. relay notifications.
2. verify device ownership.
3. get conveyed information.

Please see each section linked above for details.

3 Getting Started

This section presents a few examples illustrating some common interactions with SZTPD.

These examples use the ubiquitous curl command line utility program for illustration purposes only. It is expected that production code would use a programming language to achieve the same result.

These examples assume a freshly installed SZTPD, i.e., using the default values described in “Defaults” section in the [Installation Guide](#). To run a fresh instance of SZTPD, in one window, run the following command:

```
$ export SZTPD_MODE=0; sztpd sqlite:///memory:
```

3.1 Fetching Host-meta

RFC 8040 defines a discovery protocol for determining a RESTCONF server's root resource location. SZTPD uses `/restconf` for the root resource location.

Request:

```
$ curl -i http://127.0.0.1:8080/.well-known/host-meta
```

Response:

```
HTTP/1.1 200 OK
Server: <redacted>
Content-Type: application/xrd+xml; charset=utf-8
Content-Length: 104
Date: Thu, 23 Jan 2020 19:49:23 GMT

<XRD xmlns="http://docs.oasis-open.org/ns/xri/xrd-1.0">
  <Link rel="restconf" href="/restconf"/>
</XRD>
```

3.2 Fetching the RESTCONF Root Resource

RFC 8040 defines an ability to query the RESTCONF server's root resource to determine, for instance, what yang-library version it is using.

Request:

```
$ curl -i http://127.0.0.1:8080/restconf
```

Response:

```
HTTP/1.1 200 OK
Server: <redacted>
Content-Type: application/yang-data+json; charset=utf-8
Content-Length: 120
Date: Thu, 23 Jan 2020 19:50:07 GMT

{
  "ietf-restconf:restconf" : {
    "data" : {},
    "operations" : {},
    "yang-library-version" : "2019-01-04"
  }
}
```

3.3 Get the YANG Library for this RESTCONF server.

The YANG Library is described in [RFC 8525](#). RESTCONF clients may use it to understand what schema the connected server implements, which varies by the 'use-for' node in the [Transport](#) configuration.

Knowing the schema implemented is important to developer understanding SZTPD's various APIs. This is especially true for the 'native' and 'tenant' interface' types. These interface types present SZTPD's [Northbound API](#).

Note that, for understanding the 'rfc8572' interface type presented by SZTPD's [Southbound API](#), [section 7 of RFC 8572](#) is highly recommended.

The following request shows that the 'wn-sztpd-1' YANG module is "implemented". Had we chosen mode 'x' when the sztpd process was started in the [Getting Started](#) section, then 'wn-sztpd-1' would be "imported" while 'wn-sztpd-x' would be "implemented".

Note that the 'tenant' interface type is effectively the 'wn-sztpd-1' with the 'system-level-administration-disabled' feature set.

Request:

```
$ curl -i http://127.0.0.1:8080/restconf/ds/ietf-datastores:operational/ietf-yang-library:yang-library
```

Response:

```
HTTP/1.1 200 OK
Server: <redacted>
Content-Type: application/yang-data+json; charset=utf-8
Content-Length: 8422
Date: Thu, 23 Jan 2020 19:52:23 GMT

{
  "ietf-yang-library:modules-state": {
    "module-set-id": "unset",
    "module": [
      {
        "name": "ietf-yang-types",
        "revision": "2013-07-15",
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-yang-types",
        "conformance-type": "import"
      },
      {
        "name": "ietf-inet-types",
        "revision": "2013-07-15",
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-inet-types",
        "conformance-type": "import"
      },
      {
        "name": "ietf-datastores",
        "revision": "2018-02-14",
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-yang-datastores",
        "conformance-type": "import"
      },
      {
        "name": "ietf-yang-library",
        "revision": "2019-01-04",
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-yang-library",
        "conformance-type": "import"
      },
      {
        "name": "iana-crypt-hash",
        "revision": "2014-08-06",
        "namespace": "urn:ietf:params:xml:ns:yang:iana-crypt-hash",
        "conformance-type": "import"
      },
      {
        "name": "ietf-x509-cert-to-name",
        "revision": "2014-12-10",
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-x509-cert-to-name",
        "conformance-type": "implement"
      },
      {
        "name": "ietf-restconf",
        "revision": "2017-01-26",
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-restconf",
        "conformance-type": "import"
      },
      {
        "name": "ietf-netconf-acm",
        "revision": "2018-02-14",
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-netconf-acm",
        "conformance-type": "import"
      },
      {
        "name": "ietf-sztp-conveyed-info",
        "revision": "2019-04-30",
        "schema": "https://example.com/test.yang",
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-sztp-conveyed-info",
        "conformance-type": "implement"
      },
      {
        "name": "ietf-crypto-types",
        "revision": "2020-01-23",
        "schema": "https://example.com/test.yang",
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-crypto-types",
        "conformance-type": "implement"
      },
      {
        "name": "iana-hash-algs",
        "revision": "2020-01-23",
        "schema": "https://example.com/test.yang",
        "namespace": "urn:ietf:params:xml:ns:yang:iana-hash-algs",
        "conformance-type": "import"
      }
    ]
  }
}
```

```

    "name": "iana-symmetric-algs",
    "revision": "2020-01-23",
    "schema": "https://example.com/test.yang",
    "namespace": "urn:ietf:params:xml:ns:yang:iana-symmetric-algs",
    "conformance-type": "import"
  },
  {
    "name": "iana-asymmetric-algs",
    "revision": "2020-01-23",
    "schema": "https://example.com/test.yang",
    "namespace": "urn:ietf:params:xml:ns:yang:iana-asymmetric-algs",
    "conformance-type": "import"
  },
  {
    "name": "ietf-truststore",
    "revision": "2020-01-23",
    "feature": [
      "x509-certificates"
    ],
    "schema": "https://example.com/test.yang",
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-truststore",
    "conformance-type": "import"
  },
  {
    "name": "ietf-keystore",
    "revision": "2020-01-23",
    "schema": "https://example.com/test.yang",
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-keystore",
    "conformance-type": "import"
  },
  {
    "name": "ietf-tcp-common",
    "revision": "2020-01-23",
    "schema": "https://example.com/test.yang",
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-tcp-common",
    "conformance-type": "import"
  },
  {
    "name": "ietf-tcp-client",
    "revision": "2020-01-23",
    "schema": "https://example.com/test.yang",
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-tcp-client",
    "conformance-type": "import"
  },
  {
    "name": "ietf-tcp-server",
    "revision": "2020-01-23",
    "schema": "https://example.com/test.yang",
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-tcp-server",
    "conformance-type": "import"
  },
  {
    "name": "ietf-tls-common",
    "revision": "2020-01-23",
    "schema": "https://example.com/test.yang",
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-tls-common",
    "conformance-type": "import"
  },
  {
    "name": "ietf-tls-client",
    "revision": "2020-01-23",
    "feature": [
      "x509-certificate-auth",
      "client-auth-config-supported"
    ],
    "schema": "https://example.com/test.yang",
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-tls-client",
    "conformance-type": "import"
  },
  {
    "name": "ietf-tls-server",
    "revision": "2020-01-23",
    "feature": [
      "x509-certificate-auth",
      "client-auth-config-supported"
    ],
    "schema": "https://example.com/test.yang",
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-tls-server",
    "conformance-type": "import"
  },
  {
    "name": "ietf-http-client",
    "revision": "2020-01-23",
    "schema": "https://example.com/test.yang",
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-http-client",
    "conformance-type": "import"
  },
}

```



```
{
  "name": "ietf-http-server",
  "revision": "2020-01-23",
  "schema": "https://example.com/test.yang",
  "namespace": "urn:ietf:params:xml:ns:yang:ietf-http-server",
  "conformance-type": "import"
},
{
  "name": "ietf-restconf-server",
  "revision": "2020-01-23",
  "feature": [
    "http-listen",
    "https-listen"
  ],
  "schema": "https://example.com/test.yang",
  "namespace": "urn:ietf:params:xml:ns:yang:ietf-restconf-server",
  "conformance-type": "import"
},
{
  "name": "wn-x509-c2n",
  "revision": "2020-01-23",
  "namespace": "https://watsen.net/wnc2n",
  "conformance-type": "implement"
},
{
  "name": "wn-app",
  "revision": "2020-01-23",
  "schema": "https://example.com/test.yang",
  "namespace": "https://watsen.net/wnapp",
  "conformance-type": "import"
},
{
  "name": "wn-sztpd-0",
  "revision": "2020-01-23",
  "feature": [
    "onboarding-supported"
  ],
  "schema": "https://example.com/defs.yang",
  "namespace": "https://watsen.net/sztpd-0",
  "conformance-type": "imported"
},
{
  "name": "wn-sztpd-1",
  "revision": "2020-01-23",
  "feature": [
    "onboarding-supported"
  ],
  "schema": "https://example.com/defs.yang",
  "namespace": "https://watsen.net/sztpd-1",
  "conformance-type": "implement"
}
]
}
```

3.4 Get the Current (Default) Configuration

Request:

```
$ curl -i http://127.0.0.1:8080/restconf/ds/ietf-datastores:running
```

Response:

```
HTTP/1.1 200 OK
Server: <redacted>
Content-Type: application/yang-data+json; charset=utf-8
Content-Length: 486
Date: Thu, 23 Jan 2020 19:55:47 GMT

{
  "wn-sztpd-1:transport": {
    "listen": {
      "endpoint": [
        {
          "name": "default startup endpoint",
          "use-for": "native-interface",
          "http": {
            "tcp-server-parameters": {
              "local-address": "127.0.0.1"
            }
          }
        }
      ]
    }
  }
}
```

3.5 Configure an Administrator

As was mentioned [previously](#), a freshly installed SZTPD has no administrator account configured, and yet it requires one, and thus the first write operation to a freshly installed SZTPD instance must configure at least one administrator account. The following illustrates this using a POST request.

Request:

```
$ cat admin_accounts.json
{
  "wn-sztpd-1:admin-accounts": {
    "admin-account": [
      {
        "email-address": "my-admin@example.com",
        "password": "$0$my-secret",
        "access": "unrestricted"
      }
    ]
  }
}

$ curl -i -X POST --data @admin_accounts.json -H "Content-Type: application/yang-data+json" \
http://127.0.0.1:8080/restconf/ds/ietf-datastores:running
```

Response:

```
HTTP/1.1 201 Created
Server: <redacted>
Content-Length: 0
Content-Type: application/octet-stream
Date: Thu, 23 Jan 2020 20:19:27 GMT
```

3.6 Create a Device Type

Request:

```
cat device-types.json
{
  "wn-sztpd-1:device-types": {
    "device-type": [
      {
        "name": "my-device-type"
      }
    ]
  }
}

$ curl -i -X POST --data @device-types.json -H "Content-Type:application/yang-data+json" \
http://127.0.0.1:8080/restconf/ds/ietf-datastores:running
```

Response:

```
HTTP/1.1 401 Unauthorized
Server: <redacted>
Content-Length: 0
Content-Type: application/octet-stream
Date: Thu, 23 Jan 2020 20:24:21 GMT
```

Again, but this time with authentication!

Request:

```
[Note: '\' line endings per BCP XXX, RFC XXXX]

$ curl -i -X POST --user my-admin@example.com:my-secret --data @device-types.json \
-H "Content-Type:application/yang-data+json" http://127.0.0.1:8080/restconf/ds/ietf-datastores:running
```

Response:

```
HTTP/1.1 201 Created
Server: <redacted>
Content-Length: 0
Content-Type: application/octet-stream
Date: Thu, 23 Jan 2020 20:29:19 GMT
```

3.7 Create a Device

Alert: be mindful that this example uses the mode-1 schema, and each YANG schema has different ways to configured devices (i.e., if there is a list of devices under the “/tenants/tenant” tree).

Request:

```
[Note: '\' line endings per BCP XXX, RFC XXXX]

$ cat devices.json
{
  "wn-sztpd-1:devices": {
    "device": [
      {
        "serial-number": "my-serial-number",
        "device-type": "my-device-type",
        "activation-code": "$0$my-secret"
      }
    ]
  }
}

$ curl -i -X POST --user my-admin@example.com:my-secret --data @devices.json \
-H "Content-Type:application/yang-data+json" http://127.0.0.1:8080/restconf/ds/ietf-datastores:running
```

Response:

```
HTTP/1.1 201 Created
Server: <redacted>
Content-Length: 0
Content-Type: application/octet-stream
Date: Thu, 23 Jan 2020 20:32:23 GMT
```

3.8 Again, with a Single PUT

When initializing a fresh system, it is often easiest to replace the entire contents of the SZTPD running configuration with a single request to the server.

The following single PUT achieves the same result as all of the above commands combined.

Note that the system-provided default `/transport` node is included in the PUT as well, as otherwise the system will complain that it's being deleted.

Request:

```
$ cat running.json
{
  "wn-sztpd-1:transport": {
    "listen": {
      "endpoint": [
        {
          "name": "default startup endpoint",
          "use-for": "native-interface",
          "http": {
            "tcp-server-parameters": {
              "local-address": "127.0.0.1"
            }
          }
        }
      ]
    }
  },
  "wn-sztpd-1:admin-accounts": {
    "admin-account": [
      {
        "email-address": "my-admin@example.com",
        "password": "$0$my-secret",
        "access": "unrestricted"
      }
    ]
  },
  "wn-sztpd-1:device-types": {
    "device-type": [
      {
        "name": "my-device-type"
      }
    ]
  },
  "wn-sztpd-1:devices": {
    "device": [
      {
        "serial-number": "my-serial-number",
        "device-type": "my-device-type",
        "activation-code": "$0$my-secret"
      }
    ]
  }
}

$ curl -i -X PUT --data @running.json -H "Content-Type:application/yang-data+json" \
http://127.0.0.1:8080/restconf/ds/ietf-datastores:running
```

Response:

```
HTTP/1.1 204 No Content
Server: <redacted>
Content-Length: 0
Content-Type: application/octet-stream
Date: Thu, 23 Jan 2020 20:41:54 GMT
```

3.9 Initializing TLS

A production deployment must use TLS to protect the RESTCONF resources.

This section shows how to configure SZTPD to have:

- two listening ports, one for an NBI and the other for an SBI.
- two keys and their associated certificates in the Keystore.
- one trust anchor certificate in the Truststore for DevID-auth.
- one “device-type” identifying a specific Truststore cert to use.
- one “device” using the afore mentioned device-type.

For running code, please see the [Simulator](#) code.

This section uses scripts to generate its contents. These scripts use a PKI that has been instantiated as “pki” in the current directory. This ‘pki’ directory is the same as that in the [Simulator](#) code. These scripts could be run out of that directory, after creating an empty directory called “output”.

The code below uses a sed script to replace placeholder values in a template to produce the configuration sent by PUT. These values are defined as follows:

Variable	Description
NBI_PORT	The port that the ‘native’ interface listens on
SBI_PORT	The port that the ‘rfc8572’ interface listens on
NBI_PRI_KEY_B64	The base64 encoding of the server’s NBI private key.
NBI_PUB_KEY_B64	The base64 encoding of the server’s NBI public key.
NBI_EE_CERT_B64	The base64 encoding of the server’s NBI end-entity cert.
SBI_PRI_KEY_B64	The base64 encoding of the server’s SBI private key.
SBI_PUB_KEY_B64	The base64 encoding of the server’s SBI public key.
SBI_EE_CERT_B64	The base64 encoding of the server’s SBI end-entity cert.

These are the base64 of the DER (not the PEM), that is, the header and footer ===== lines are missing, and all the B64 characters are on one line.

The private and public key values are the native OpenSSL values for private and public keys.

The certificate values are the degenerate form of a CMS (PKCS #7) commonly used to communicate a chain of certificates.

Script:

```
=====NOIE: '\ ' line wrapping per RFC 8792=====
#!/bin/sh

OPENSSL="openssl"
TEMPDIR="mktemp -d"

# NBI Port
NBI_PORT="8080"
NBI_PRI_KEY_B64="$OPENSSL enc -base64 -A -in pki/sztpd1/nbi/end-entity/private_key.der"
NBI_PUB_KEY_B64="$OPENSSL enc -base64 -A -in pki/sztpd1/nbi/end-entity/public_key.der"
cat pki/sztpd1/nbi/end-entity/my_cert.pem pki/sztpd1/nbi/intermediate2/my_cert.pem > $TEMPDIR/cert_chain.pem
$OPENSSL crl2pkcs7 -nocrl -certfile $TEMPDIR/cert_chain.pem -outform DER -out $TEMPDIR/cert_chain.cms
NBI_EE_CERT_B64="$OPENSSL enc -base64 -A -in $TEMPDIR/cert_chain.cms"

# SBI Port
SBI_PORT="9090"
SBI_PRI_KEY_B64="$OPENSSL enc -base64 -A -in pki/sztpd1/sbi/end-entity/private_key.der"
SBI_PUB_KEY_B64="$OPENSSL enc -base64 -A -in pki/sztpd1/sbi/end-entity/public_key.der"
cat pki/sztpd1/sbi/end-entity/my_cert.pem pki/sztpd1/sbi/intermediate2/my_cert.pem > $TEMPDIR/cert_chain.pem
$OPENSSL crl2pkcs7 -nocrl -certfile $TEMPDIR/cert_chain.pem -outform DER -out $TEMPDIR/cert_chain.cms
SBI_EE_CERT_B64="$OPENSSL enc -base64 -A -in $TEMPDIR/cert_chain.cms"

# client cert (DevID) trust anchor
cat pki/client/root-ca/my_cert.pem pki/client/intermediate1/my_cert.pem pki/client/intermediate2/my_cert.pe\
m > $TEMPDIR/ta_cert_chain.pem
$OPENSSL crl2pkcs7 -nocrl -certfile $TEMPDIR/ta_cert_chain.pem -outform DER -out $TEMPDIR/ta_cert_chain.cms
CLIENT_CERT_TA_B64="$OPENSSL enc -base64 -A -in $TEMPDIR/ta_cert_chain.cms"

# initialize body for the PUT request
cat << EOM > $TEMPDIR/running.json
```

```

{
  "wn-sztpd-1:transport": {
    "listen": {
      "endpoint": [
        {
          "name": "native-interface",
          "use-for": "native-interface",
          "https": {
            "tcp-server-parameters": {
              "local-address": "0.0.0.0",
              "local-port": $NBL_PORT
            },
            "tls-server-parameters": {
              "server-identity": {
                "certificate": {
                  "reference": {
                    "asymmetric-key": "nbi-server-end-entity-key",
                    "certificate": "nbi-server-end-entity-cert"
                  }
                }
              }
            }
          }
        },
        {
          "name": "rfc8572-interface",
          "use-for": "rfc8572-interface",
          "https": {
            "tcp-server-parameters": {
              "local-address": "0.0.0.0",
              "local-port": $SBL_PORT
            },
            "tls-server-parameters": {
              "server-identity": {
                "certificate": {
                  "reference": {
                    "asymmetric-key": "sbi-server-end-entity-key",
                    "certificate": "sbi-server-end-entity-cert"
                  }
                }
              },
              "client-authentication": {
                "ca-certs": {
                  "local-truststore-reference": "my-device-identity-ca-certs"
                }
              }
            }
          }
        }
      ]
    },
    "wn-sztpd-1:admin-accounts": {
      "admin-account": [
        {
          "email-address": "my-admin@example.com",
          "password": "\$0\$my-secret",
          "access": "unrestricted"
        }
      ]
    },
    "wn-sztpd-1:keystore": {
      "asymmetric-keys": {
        "asymmetric-key": [
          {
            "name": "nbi-server-end-entity-key",
            "public-key-format": "ietf-crypto-types:subject-public-key-info-format",
            "public-key": "$NBL_PUB_KEY_B64",
            "private-key-format": "ietf-crypto-types:ec-private-key-format",
            "cleartext-private-key": "$NBL_PRI_KEY_B64",
            "certificates": {
              "certificate": [
                {
                  "name": "nbi-server-end-entity-cert",
                  "cert-data": "$NBL_EE_CERT_B64"
                }
              ]
            }
          }
        ]
      },
      {
        "name": "sbi-server-end-entity-key",
        "public-key-format": "ietf-crypto-types:subject-public-key-info-format",
        "public-key": "$SBL_PUB_KEY_B64",
        "private-key-format": "ietf-crypto-types:ec-private-key-format",
        "cleartext-private-key": "$SBL_PRI_KEY_B64",
        "certificates": {
          "certificate": [

```

```

        {
            "name": "sbi-server-end-entity-cert",
            "cert-data": "$SBI_EE_CERT_B64"
        }
    ]
}
},
"wn-sztpd-1:truststore": {
    "certificate-bags": {
        "certificate-bag": [
            {
                "name": "my-device-identity-ca-certs",
                "description": "A set of TA certs used to authenticate device client certs.",
                "certificate": [
                    {
                        "name": "my-device-identity-ca-cert-circa-2020",
                        "cert-data": "$CLIENT_CERT_TA_B64"
                    }
                ]
            }
        ]
    }
},
"wn-sztpd-1:device-types": {
    "device-type": [
        {
            "name": "my-device-type",
            "identity-certificates": {
                "verification": {
                    "local-truststore-reference": {
                        "certificate-bag": "my-device-identity-ca-certs",
                        "certificate": "my-device-identity-ca-cert-circa-2020"
                    }
                },
                "serial-number-extraction": "wn-x509-c2n:serial-number"
            }
        }
    ]
},
"wn-sztpd-1:devices": {
    "device": [
        {
            "serial-number": "my-serial-number",
            "device-type": "my-device-type",
            "activation-code": "\$0\$my-secret"
        }
    ]
}
}
EOM

# PUT running
curl -i -X PUT --data @${TEMPDIR}/running.json -H "Content-Type: application/yang-data+json" http://127.0.0.1:\
8080/restconf/ds/ietf-datastores:running 1> output/put_running.out 2> /dev/null

rm -rf "${TEMPDIR}"

```

Output:

```

HTTP/1.1 100 Continue

HTTP/1.1 204 No Content
Server: <redacted>
Content-Length: 0
Content-Type: application/octet-stream
Date: Fri, 18 Sep 2020 23:08:29 GMT

```

3.10 Give Server time to restart

Since this change modifies the “/transport” tree, the SZTPD instance sends a SIGHUP to itself, thus causing it to reload the configuration and re-open listening ports per configuration.

SZTPD takes about 2 seconds to reload with a relatively empty configuration. Once sufficient time has elapsed (a few seconds), the configured ports will be available.

3.11 Ensuring the Ports are up

Below we first use the wrong URL scheme “http”, and then correct it to “https”, and then, finally, add the missing “-cacert” parameter. All these commands use “HEAD” against yang-library.

```
===== NOIE: '\ ' line wrapping per RFC 8792 =====
#!/bin/sh

printf "==== Output from NBI Port ====\n" 1> output/get_yang_library.out

printf "\n1) Incorrect 'http' (should be 'https'):\n\n" 1>> output/get_yang_library.out
curl -i --head --user my-admin@example.com:my-secret http://127.0.0.1:8080/restconf/ds/ietf-datastores:oper\
ational/ietf-yang-library:yang-library &> temp.out
sed -n '/curl/, $p' temp.out 1>> output/get_yang_library.out

printf "\n2) Correct 'https' (but missing '--cacert'):\n\n" 1>> output/get_yang_library.out
curl -i --head --user my-admin@example.com:my-secret https://127.0.0.1:8080/restconf/ds/ietf-datastores:ope\
rational/ietf-yang-library:yang-library &> temp.out
sed -n '/curl/, $p' temp.out 1>> output/get_yang_library.out

printf "\n3) Correct 'https' and '--cacert':\n\n" 1>> output/get_yang_library.out
curl -i --head --cacert nbi_trust_chain.pem --user my-admin@example.com:my-secret https://127.0.0.1:8080/re\
stconf/ds/ietf-datastores:operational/ietf-yang-library:yang-library 1>> output/get_yang_library.out 2> /de\
v/null

printf "\n\n==== Output from SBI Port ====\n" 1>> output/get_yang_library.out

printf "\n1) Incorrect 'http' (should be 'https') [also note 'my-serial-number']:\n\n" 1>> output/get_yang\
_library.out
curl -i --head --user my-serial-number:my-secret http://127.0.0.1:9090/restconf/ds/ietf-datastores:operatio\
nal/ietf-yang-library:yang-library &> temp.out
sed -n '/curl/, $p' temp.out 1>> output/get_yang_library.out

printf "\n2) Correct 'https' (but missing '--cacert'):\n\n" 1>> output/get_yang_library.out
curl -i --head --user my-serial-number:my-secret https://127.0.0.1:9090/restconf/ds/ietf-datastores:operati\
onal/ietf-yang-library:yang-library &> temp.out
sed -n '/curl/, $p' temp.out 1>> output/get_yang_library.out

printf "\n3) Correct 'https' and '--cacert' (but missing '--key' and '--cert'):\n\n" 1>> output/get_yang_li\
brary.out
curl -i --head --cacert sbi_trust_chain.pem --user my-serial-number:my-secret https://127.0.0.1:9090/restco\
nf/ds/ietf-datastores:operational/ietf-yang-library:yang-library 1>> output/get_yang_library.out 2> /dev/nu\
ll

printf "\n4) Correct 'https', '--cacert', '--key', and '--cert':\n\n" 1>> output/get_yang_library.out
curl -i --head --cacert sbi_trust_chain.pem --key pki/client/end-entity/private_key.pem --cert pki/client/e\
nd-entity/my_cert.pem --user my-serial-number:my-secret https://127.0.0.1:9090/restconf/ds/ietf-datastores:\
operational/ietf-yang-library:yang-library 1>> output/get_yang_library.out 2> /dev/null

# cleanup
rm temp.out
```

Output:

```
==== Output from NBI Port ====

1) Incorrect 'http' (should be 'https'):

curl: (52) Empty reply from server

2) Correct 'https' (but missing '--cacert'):

curl: (60) SSL certificate problem: unable to get local issuer certificate
More details here: https://curl.haxx.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.

3) Correct 'https' and '--cacert':

HTTP/1.1 200 OK
Content-Type: application/yang-data+json; charset=utf-8
Server: <redacted>
Content-Length: 7627
Date: Fri, 18 Sep 2020 23:08:34 GMT

==== Output from SBI Port ====

1) Incorrect 'http' (should be 'https') [also note 'my-serial-number']:
```



```
curl: (52) Empty reply from server

2) Correct 'https' (but missing '--cacert'):

curl: (60) SSL certificate problem: unable to get local issuer certificate
More details here: https://curl.haxx.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.

3) Correct 'https' and '--cacert' (but missing '--key' and '--cert'):

HTTP/1.1 401 Unauthorized
Content-Type: text/plain; charset=utf-8
Server: <redacted>
Content-Length: 95
Date: Fri, 18 Sep 2020 23:08:34 GMT

4) Correct 'https', '--cacert', '--key', and '--cert':

HTTP/1.1 200 OK
Content-Type: application/yang-data+json; charset=utf-8
Server: <redacted>
Content-Length: 1935
Date: Fri, 18 Sep 2020 23:08:34 GMT
```

3.12 Simulate Device Trying to Get Bootstrapping Data

This command fails because no “responses” have been configured for the device on the server. Note that error code 404 is used to indicate that the device may try again.

```
===== NOIE: '\ ' line wrapping per RFC 8792 =====

#!/bin/sh

# initialize the 'input' node for the RPC
cat << EOM > input.json
{
  "ietf-sztp-bootstrap-server:input" : {
    "hw-model" : "model-x",
    "os-name" : "vendor-os",
    "os-version" : "17.3R2.1",
    "nonce" : "extralongbase64encodedvalue="
  }
}
EOM

# POST 'input' for the "get-bootstrapping-data" RPC resource
curl -i -X POST --data @input.json -H "Content-Type: application/yang-data+json" --cacert sbi_trust_chain.p\
m --key pki/client/end-entity/private_key.pem --cert pki/client/end-entity/my_cert.pem --user my-serial-num\
ber:my-secret https://127.0.0.1:9090/restconf/operations/ietf-sztp-bootstrap-server:get-bootstrapping-data \
l> output/post_rpc_input.out 2>/dev/null

# line-return needed for markdown
echo "" 1>> output/post_rpc_input.out

# cleanup
rm -f input.json
```

Output:

```
===== NOIE: '\ ' line wrapping per RFC 8792 =====

HTTP/1.1 404 Not Found
Content-Type: text/plain; charset=utf-8
Server: <redacted>
Content-Length: 142
Date: Fri, 18 Sep 2020 23:08:35 GMT

{"ietf-restconf:errors": {"error": [{"error-type": "application", "error-tag": "data-missing", "error-messa\
ge": "No responses configured."}]}}
```

3.13 Configuring a Redirect Response

Configure the `/bootstrapping-servers` node. This list contains all `'bootstrap-server'` definitions, an ordered subset of which may be referenced by subsequent steps. Note that the port and `trust-anchor` nodes are replaced by variables.

```
===== NOTE: '\ ' line wrapping per RFC 8792 =====
#!/bin/sh
OPENSSL="openssl"

# generate values for placeholders
cat pki/sztpd2/sbi/root-ca/my_cert.pem pki/sztpd2/sbi/intermediate1/my_cert.pem > cert_chain.pem
$OPENSSL crl2pkcs7 -nocrl -certfile cert_chain.pem -outform DER -out cert_chain.cms
BOOTSVR_TA_CERT_B64=`$OPENSSL enc -base64 -A -in cert_chain.cms`

# initialize body for the POST request
cat << EOM > bootstrap-servers.json
{
  "wn-sztpd-1:bootstrap-servers": {
    "bootstrap-server": [
      {
        "name": "my-bootstrap-server",
        "address": "127.0.0.1",
        "port": 9443,
        "trust-anchor": "$BOOTSVR_TA_CERT_B64"
      }
    ]
  }
}
EOM

# POST 'bootstrap-servers' to running
curl -i -X POST --data @bootstrap-servers.json -H "Content-Type:application/yang-data+json" --cacert nbi_tr\
ust_chain.pem --user my-admin@example.com:my-secret https://127.0.0.1:8080/restconf/ds/ietf-datastores:runn\
ing 1> output/post_bootstrap_servers.out 2>/dev/null

# cleanup
rm -f bootstrap-servers.json
rm -f cert_chain.pem
rm -f cert_chain.cms
```

Output:

```
HTTP/1.1 100 Continue
HTTP/1.1 201 Created
Server: <redacted>
Content-Length: 0
Content-Type: application/octet-stream
Date: Fri, 18 Sep 2020 23:08:35 GMT
```

Configure the '/conveyed-information-responses' node. Configure a 'my-redirect-information' to return 'my-bootstrap-server'. An ordered list of bootstrap servers may be configured.

```
=====NOIE: '\ ' line wrapping per RFC 8792=====
#!/bin/sh
OPENSSL="openssl"

# initialize body for the POST request
cat << EOM > conveyed-information-responses.json
{
  "wn-sztpd-1:conveyed-information-responses": {
    "redirect-information-response": [
      {
        "name": "my-redirect-information",
        "redirect-information": {
          "bootstrap-server": [
            "my-bootstrap-server"
          ]
        }
      }
    ]
  }
}
EOM

# POST 'conveyed-information-responses' to running
curl -i -X POST --data @conveyed-information-responses.json -H "Content-Type: application/yang-data+json" --cacert nbi_trust_chain.pem --user my-admin@example.com:my-secret https://127.0.0.1:8080/restconf/ds/ietf-datastores:running 1> output/post_conveyed_information_responses.out 2>/dev/null

# cleanup
rm -f conveyed-information-responses.json
```

Output:

```
HTTP/1.1 201 Created
Server: <redacted>
Content-Length: 0
Content-Type: application/octet-stream
Date: Fri, 18 Sep 2020 23:08:35 GMT
```

Configure the 'response-manager' node inside the device object to return "my-redirect-information". A "catch-all" rule (i.e., one without any 'match-criteria' defined) that tells STZPD to return the just configured 'my-redirect-information'.

```
=====NOIE: '\ ' line wrapping per RFC 8792=====
#!/bin/sh

# initialize body for the POST request
cat << EOM > response-manager.json
{
  "wn-sztpd-1:response-manager": {
    "matched-response": [
      {
        "name": "catch-all-response",
        "response": {
          "conveyed-information": {
            "redirect-information": {
              "reference": "my-redirect-information"
            }
          }
        }
      }
    ]
  }
}
EOM

# POST 'response-manager' to *device*
curl -i -X POST --data @response-manager.json -H "Content-Type: application/yang-data+json" --cacert nbi_tru\
st_chain.pem --user my-admin@example.com:my-secret https://127.0.0.1:8080/restconf/ds/ietf-datastores:runni\
ng/wn-sztpd-1:devices/device=my-serial-number 1> output/post_response_manager.out 2>/dev/null

# cleanup
rm -f response-manager.json
```

Output:

```
HTTP/1.1 201 Created
Server: <redacted>
Content-Length: 0
Content-Type: application/octet-stream
Date: Fri, 18 Sep 2020 23:08:36 GMT
```

3.14 Simulate Device Getting Redirect Information

Accessing the get–bootstrapping–data RPC on the SBI again, now it works.

Request:

- Same as in the [Simulate Device Trying to Get Bootstrapping Data](#) section.

Output:

```
===== NOTE: '\ ' line wrapping per RFC 8792 =====
HTTP/1.1 200 OK
Content-Type: application/yang-data+json; charset=utf-8
Server: <redacted>
Content-Length: 2751
Date: Fri, 18 Sep 2020 23:08:36 GMT

{
  "ietf-sztp-bootstrap-server:output": {
    "conveyed-information": "MIHhYQYLKozIhvcNAQkQASugges4BIIHtHsKICAgICJpZXRmLXN6dHAtY29udmV5ZWQtaW5mbz\
pyZWRpcmVjdC1pbmZvcmlhdGlvbiI6IHsKICAgICAgICAiYm9vdHN0cmFwLXNlcnZlciI6IFsKICAgICAgICAgewogICAgICAgICAgICAgI\
CAGlCAgImFkZHZhc3MiOiAiMTI3LjAuMC4xIiwKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIC\
dC1hbmNob3IiOiAiTiUJRkVRWUpLb1pJaHZjTkFRY0NvSUIGQWpDQ0JQENBUUUV4QURBTEJna3Foa2IHOXcwQkI3R2dnZ1RrTUJlQ1dqQ0N\
BZitmQXdkJkFhSUJBVEFLQmdncWhrak9QUVFEQWpCMU1Rc3dDUVIEVFRFR0V3SllXREVkTUJzR0ExVUVDQXdVVFhrZ1UzUmhkR1VnYjNjZl\
VlSnZkbWx1WTJVeEdEQVdCZ05WCkFvTUQwMTVJRTI5WjJGdWFFYcGhkR2k2YmpFUU1BNEdBMVVFQ3d3SFRYa2dWVzVwZERFYklCa0dBMVVFQ\
Xd3U2MySnBMM05sY25abGNpOXIiMjkwTFdOaE1DQVhEVEl3TURreE9ESXdVNGswTkZvWUR6azVPVGt4TWpNeE1c1TTFPVFU1V2pCMU1Rc3dD\
UVIEVFRFR0V3SllXREVkTUJzR0ExVUVDQXdVVFhrZ1UzUmhkR1VnYjNjZlVlSnZkbWx1WTJVeEdEQVdCZ05WCkFvTUQwMTVJRTI5WjJGdWFF\
YcGhkR2k2YmpFUU1BNEdBMVVFQ3d3SFRYa2dWVzVwZERFYklCa0dBMVVFQXd3U2MySnBMM05sY25abGNpOXIiMjkwTFdOaE1Ca3dFdlIIS2\
9aSXpqMENBUVJIS29aSXpqMERBUWNEUWdBRtB1REM5ZFMWkOSW5OK3hKdG9HWntG1mUsVU9YWkdLSmFVV0v4XFoM1VMWjA1dWdFRmRIV\
mc1WkzR3BGWVdGV2VFMU8rTDJxUGJ3WUZpTFdjVHPLTitNSHd3SFFZRFZSMER9CQIlFRktoYjkyRIZEWnpCSnBoR2N5RC9XU0FvVvHzWU1B\
d0dBMVVFkRXdRkR1BTUJBJzjh3RGdZRFZSMFBBUgYQkFRREFhRUdNRDBHQITFVZEh3UTJNRFF3TXFBd29DNkdMR2gwZEhBNk5OWpjbXdlWlh\
caGJYQnNaUzVqYjwLlkyRtTijMkpwT25ObGNuWmxjanB5YjI5MEExTmhnNQ0FYRFRJd01Ea3hPREI3TVRrME5Wb1IEems1T1RreE1c1TXhNak0xT1RVNvdqQj\
dNUNXN3Q1FZRFRZRUUdFd0pZV0RFZE1C0dBMVVFQ0F3VVRya2dVM1JoZEIVZ2ZSwdVSEpZ2ZG1sdVkyVXhHREFXQmdOVkIBb01EMDEISU5e\
VoyRnVhWHBoZE4sdmJqRVFNQTRHQITFVRUN3d0hUWGtmVklcGRERWhNqjhHQITFVRUF3d1ljMkpwTDNOBGNuWmxjaTlwYm5SbGNtMWsaR2xo\
ZEIVeE1Ga3dFdlIIS29aSXpqMENBUVJIS29aSXpqMERBUWNEUWdBRXdpMIVmTDkxWnpNa3lZLzFJbi8zZDFDMkdlnkwcZjFEU1BzZERNYWx\
sUkdJMINMVFUNOXFEZENuZldvTVRydmBuYm5ldmo5cVFOR1pTbm8wc0dMU2l2T0JvRENcblRBZEJnTIZIUTRFRmdRVWVG0dUUVJzZk00SW\
poNGREelZiZ0NOUmfPc3dlld1EVIWakJCZ3dGb0FVcUZ2M1lWVU5uTUVtbUVaekdQOVpJQ2hSZGhnd0RBWURWUjBUQkFVd0F3RUlvekFPQ\
mdOVkhhROEJBZjhFQkFNQ0FRWXdQUVIEVlWzKJEWXdOREF5b0RDZ0xvWXNhSFfwY0RvdkwyThiQzVsZUdGdGNHcGxMbU52YlQ5allUMXpZ\
bWs2YzJWeWRtVnlPbkp2YjNRdFkyRXkdZl1JS29aSXpqMEVBD0EU0FBd1JRSWhBTmXQVDFwEERAMHBlczJTaC9qMkdnNEFWQ01qQTVKRMx\
DYnJUQmo4MGtkSkfpQRDNWdFfZcWHFvRWJ6Sk0wMmxEVTdiSTJGZGtXT2pWnNfFamk4MTd1WDZfQU1RQTOiCiAgICAgICAgICAgICAgIC\
AgICAgICBdCiAgICB9Cn0="
  }
}
```

3.15 Delete Redirect Information

Now delete all of the above config to get back to baseline, needed for upcoming [Configuring an Onboarding Response](#) section.

Note that the configuration is removed in the opposite order to avoid SZTPD throwing a validation error due to dangling references.

```
=====NOTE: '\ ' line wrapping per RFC 8792=====
#!/bin/sh

# These are removed in opposite order to avoid dangling reference validation errors.
curl -i -X DELETE --cacert nbi_trust_chain.pem --user my-admin@example.com:my-secret https://127.0.0.1:8080\
/restconf/ds/ietf-datastores:running/wn-sztpd-1:devices/device=my-serial-number/response-manager 1> output/\
delete_redirect_info.out

curl -i -X DELETE --cacert nbi_trust_chain.pem --user my-admin@example.com:my-secret https://127.0.0.1:8080\
/restconf/ds/ietf-datastores:running/wn-sztpd-1:conveyed-information-responses 1>> output/delete_redirect_i\
nfo.out

curl -i -X DELETE --cacert nbi_trust_chain.pem --user my-admin@example.com:my-secret https://127.0.0.1:8080\
/restconf/ds/ietf-datastores:running/wn-sztpd-1:bootstrap-servers 1>> output/delete_redirect_info.out
```

Output:

```
HTTP/1.1 204 No Content
Server: <redacted>
Content-Length: 0
Content-Type: application/octet-stream
Date: Fri, 18 Sep 2020 23:08:36 GMT

HTTP/1.1 204 No Content
Server: <redacted>
Content-Length: 0
Content-Type: application/octet-stream
Date: Fri, 18 Sep 2020 23:08:36 GMT

HTTP/1.1 204 No Content
Server: <redacted>
Content-Length: 0
Content-Type: application/octet-stream
Date: Fri, 18 Sep 2020 23:08:36 GMT
```

3.16 Configuring an Onboarding Response

Configuring an onboarding response is done in much the same way, just with different nodes reflecting the difference between a “redirect” and an “onboarding” response.

First configure information about a hypothetical boot-image the device should be running:

```
=====NOTE: '\\ ' line wrapping per RFC 8792=====
#!/bin/sh

OPENSSL="openssl"

# initialize body for the POST request
dd if=/dev/urandom of=my-boot-image.img bs=64k count=1 >> /dev/null 2>&1
BOOT_IMG_HASH_VAL=$(OPENSSL dgst -sha256 -c my-boot-image.img | awk '{print $2}')
cat <<EOM > boot-images.json
{
  "wn-sztpd-1:boot-images": {
    "boot-image": [
      {
        "name": "my-boot-image.img",
        "download-uri": [ "https://example.com/my-boot-image.img" ],
        "image-verification": [
          {
            "hash-algorithm": "ietf-sztp-conveyed-info:sha-256",
            "hash-value": "$BOOT_IMG_HASH_VAL"
          }
        ]
      }
    ]
  }
}
EOM
```

```
# POST 'boot-images' to running
curl -i -X POST --data @boot-images.json -H "Content-Type:application/yang-data+json" --cacert nbi_trust_ch\
\ain.pem --user my-admin@example.com:my-secret https://127.0.0.1:8080/restconf/ds/ietf-datastores:running 1\
\> output/post_boot_images.out 2>/dev/null

rm -rf my-boot-image.img boot-images.json
```

Output:

```
HTTP/1.1 201 Created
Server: <redacted>
Content-Length: 0
Content-Type: application/octet-stream
Date: Fri, 18 Sep 2020 23:08:36 GMT
```

Configure information about the “pre” and “post” scripts the device should run:

```
=====NOTE: '\ ' line wrapping per RFC 8792=====

#!/bin/sh

OPENSSL="openssl"

# encode a pre-configuration script
cat << EOM > my-pre-configuration-script
#/bin/bash
echo "inside the pre-configuration-script..."
EOM
PRE_SCRIPT_B64=`$OPENSSL enc -base64 -A -in my-pre-configuration-script`

# encode a post-configuration script
cat << EOM > my-post-configuration-script
#/bin/bash
echo "inside the post-configuration-script..."
EOM
POST_SCRIPT_B64=`$OPENSSL enc -base64 -A -in my-post-configuration-script`

# initialize body for the POST request
cat << EOM > scripts.json
{
  "wn-sztpd-1:scripts": {
    "pre-configuration-script": [
      {
        "name": "my-pre-configuration-script",
        "script": "$PRE_SCRIPT_B64"
      }
    ],
    "post-configuration-script": [
      {
        "name": "my-post-configuration-script",
        "script": "$POST_SCRIPT_B64"
      }
    ]
  }
}
EOM

# POST 'scripts' to running
curl -i -X POST --data @scripts.json -H "Content-Type:application/yang-data+json" --cacert nbi_trust_chain.\
pem --user my-admin@example.com:my-secret https://127.0.0.1:8080/restconf/ds/ietf-datastores:running 1> out\
put/post_scripts.out 2>/dev/null

# cleanup
rm -f scripts.json
rm -f my-pre-configuration-script
rm -f my-post-configuration-script
```

Output:

```
HTTP/1.1 201 Created
Server: <redacted>
Content-Length: 0
Content-Type: application/octet-stream
Date: Fri, 18 Sep 2020 23:08:36 GMT
```

Set the device-specific configuration the device should run:

```
===== NOTE: '\\\ ' line wrapping per RFC 8792 =====
#!/bin/sh

OPENSSL="openssl"

# initialize body for the POST request
cat << EOM > my-configuration
<top xmlns="https://example.com/config">
  <any-xml-content-okay/>
</top>
EOM
CONFIG_B64=`$OPENSSL enc -base64 -A -in my-configuration`

cat << EOM > configurations.json
{
  "wn-sztpd-1:configurations": {
    "configuration": [
      {
        "name": "my-configuration",
        "configuration-handling": "merge",
        "config": "$CONFIG_B64"
      }
    ]
  }
}
EOM

# POST 'configurations.json' to running
curl -i -X POST --data @configurations.json -H "Content-Type:application/yang-data+json" --cacert nbi_trust\
\_chain.pem --user my-admin@example.com:my-secret https://127.0.0.1:8080/restconf/ds/ietf-datastores:running\
\g 1> output/post_configurations.out 2>/dev/null

# cleanup
rm -f my-configuration
rm -f configurations.json
```

Output:

```
HTTP/1.1 201 Created
Server: <redacted>
Content-Length: 0
Content-Type: application/octet-stream
Date: Fri, 18 Sep 2020 23:08:37 GMT
```

Configure “conveyed-information” referencing all of the above:

```
===== NOTE: '\\\ ' line wrapping per RFC 8792 =====
#!/bin/sh

# initialize body for the POST request
cat << EOM > conveyed-information-responses.json
{
  "wn-sztpd-1:conveyed-information-responses": {
    "onboarding-information-response": [
      {
        "name": "my-onboarding-information",
        "onboarding-information": {
          "boot-image": "my-boot-image.img",
          "pre-configuration-script": "my-pre-configuration-script",
          "configuration": "my-configuration",
          "post-configuration-script": "my-post-configuration-script"
        }
      }
    ]
  }
}
EOM

# POST 'conveyed-information-responses' to running
curl -i -X POST --data @conveyed-information-responses.json \
-H "Content-Type: application/yang-data+json" --cacert nbi_trust_chain.pem --user my-admin@example.com:my-\
secret \
https://127.0.0.1:8080/restconf/ds/ietf-datastores:running 1> output/post_conveyed_information_responses2\
.out 2>/dev/null

rm -f conveyed-information-responses.json
```


Output:

```
HTTP/1.1 201 Created
Server: <redacted>
Content-Length: 0
Content-Type: application/octet-stream
Date: Fri, 18 Sep 2020 23:08:37 GMT
```

Configure the 'response-manager' node inside the device object to return "my-onboarding-information". A "catch-all" rule (i.e., one without any 'match-criteria' defined) that tells STZPD to return the just configured 'my-redirect-information'.

```
===== NOIE: '\ ' line wrapping per RFC 8792 =====

#!/bin/sh

# initialize body for the POST request
cat << EOM > response-manager.json
{
  "wn-sztpd-1:response-manager": {
    "matched-response": [
      {
        "name": "catch-all-response",
        "response": {
          "conveyed-information": {
            "onboarding-information": {
              "reference": "my-onboarding-information"
            }
          }
        }
      }
    ]
  }
}
EOM

# POST 'response-manager' to *device*
curl -i -X POST --data @response-manager.json -H "Content-Type:application/yang-data+json" --cacert nbi_tru\
st_chain.pem --user my-admin@example.com:my-secret https://127.0.0.1:8080/restconf/ds/ietf-datastores:runni\
ng/wn-sztpd-1:devices/device=my-serial-number 1> output/post_response_manager2.out 2>/dev/null

# cleanup
rm -f response-manager.json
```

Output:

```
HTTP/1.1 201 Created
Server: <redacted>
Content-Length: 0
Content-Type: application/octet-stream
Date: Fri, 18 Sep 2020 23:08:37 GMT
```


Output:

```
HTTP/1.1 204 No Content
Server: <redacted>
Content-Length: 0
Content-Type: application/octet-stream
Date: Fri, 18 Sep 2020 23:08:38 GMT
```

3.19 Delete Onboarding Information

Now delete all of the onboarding config to get back to baseline (not that it matters).

Note that the configuration is removed in the opposite order to avoid SZTPD throwing a validation error due to dangling references.

```
=====NOTE: '\ ' line wrapping per RFC 8792=====

#!/bin/sh

# These are removed in opposite order to avoid dangling reference validation errors.

curl -i -X DELETE --cacert nbi_trust_chain.pem --user my-admin@example.com:my-secret https://127.0.0.1:8080\
/restconf/ds/ietf-datastores:running/wn-sztpd-1:devices/device=my-serial-number/response-manager 1> output/\
delete_onboarding_info.out 2>/dev/null

curl -i -X DELETE --cacert nbi_trust_chain.pem --user my-admin@example.com:my-secret https://127.0.0.1:8080\
/restconf/ds/ietf-datastores:running/wn-sztpd-1:conveyed-information-responses 1>> output/delete_onboarding\
_info.out 2>/dev/null

curl -i -X DELETE --cacert nbi_trust_chain.pem --user my-admin@example.com:my-secret https://127.0.0.1:8080\
/restconf/ds/ietf-datastores:running/wn-sztpd-1:boot-images 1>> output/delete_onboarding_info.out 2>/dev/\
null

curl -i -X DELETE --cacert nbi_trust_chain.pem --user my-admin@example.com:my-secret https://127.0.0.1:8080\
/restconf/ds/ietf-datastores:running/wn-sztpd-1:scripts 1>> output/delete_onboarding_info.out 2>/dev/null

curl -i -X DELETE --cacert nbi_trust_chain.pem --user my-admin@example.com:my-secret https://127.0.0.1:8080\
/restconf/ds/ietf-datastores:running/wn-sztpd-1:configurations 1>> output/delete_onboarding_info.out 2>/dev\
/null
```

Output:

```
HTTP/1.1 204 No Content
Server: <redacted>
Content-Length: 0
Content-Type: application/octet-stream
Date: Fri, 18 Sep 2020 23:08:38 GMT

HTTP/1.1 204 No Content
Server: <redacted>
Content-Length: 0
Content-Type: application/octet-stream
Date: Fri, 18 Sep 2020 23:08:38 GMT

HTTP/1.1 204 No Content
Server: <redacted>
Content-Length: 0
Content-Type: application/octet-stream
Date: Fri, 18 Sep 2020 23:08:38 GMT

HTTP/1.1 204 No Content
Server: <redacted>
Content-Length: 0
Content-Type: application/octet-stream
Date: Fri, 18 Sep 2020 23:08:38 GMT

HTTP/1.1 204 No Content
Server: <redacted>
Content-Length: 0
Content-Type: application/octet-stream
Date: Fri, 18 Sep 2020 23:08:39 GMT
```

3.20 View Audit Log

Notice how all the commands run above appear below.

```
#!/bin/sh
curl -i -X GET --cacert nbi_trust_chain.pem --user my-admin@example.com:my-secret \
  https://127.0.0.1:8080/restconf/ds/ietf-datastores:operational/wn-sztpd-1:audit-log \
  1> output/get_audit_log.out 2>/dev/null

# line-return needed for markdown
echo "" 1>> output/get_audit_log.out
```

Output:

```
=====NOTE: '\ ' line wrapping per RFC 8792=====
HTTP/1.1 200 OK
Content-Type: application/yang-data+json; charset=utf-8
Server: <redacted>
Content-Length: 7773
Date: Fri, 18 Sep 2020 23:08:39 GMT

{
  "wn-sztpd-1:audit-log": {
    "log-entry": [
      {
        "timestamp": "2020-09-18T23:08:29Z",
        "source-ip": "127.0.0.1",
        "host": "127.0.0.1:8080",
        "method": "PUT",
        "path": "/restconf/ds/ietf-datastores:running",
        "outcome": "success",
        "comment": "No authorization required for fresh installs."
      },
      {
        "timestamp": "2020-09-18T23:08:34Z",
        "source-ip": "127.0.0.1",
        "host": "127.0.0.1:9090",
        "method": "HEAD",
        "path": "/restconf/ds/ietf-datastores:operational/ietf-yang-library:yang-library",
        "outcome": "failure",
        "comment": "Client cert required but none passed for serial number my-serial-number"
      },
      {
        "timestamp": "2020-09-18T23:08:34Z",
        "source-ip": "127.0.0.1",
        "host": "127.0.0.1:9090",
        "method": "HEAD",
        "path": "/restconf/ds/ietf-datastores:operational/ietf-yang-library:yang-library",
        "outcome": "success"
      },
      {
        "timestamp": "2020-09-18T23:08:34Z",
        "source-ip": "127.0.0.1",
        "host": "127.0.0.1:9090",
        "method": "POST",
        "path": "/restconf/operations/ietf-sztp-bootstrap-server:get-bootstrapping-data",
        "outcome": "success"
      },
      {
        "timestamp": "2020-09-18T23:08:35Z",
        "source-ip": "127.0.0.1",
        "host": "127.0.0.1:8080",
        "method": "POST",
        "path": "/restconf/ds/ietf-datastores:running",
        "outcome": "success"
      },
      {
        "timestamp": "2020-09-18T23:08:35Z",
        "source-ip": "127.0.0.1",
        "host": "127.0.0.1:8080",
        "method": "POST",
        "path": "/restconf/ds/ietf-datastores:running",
        "outcome": "success"
      },
      {
        "timestamp": "2020-09-18T23:08:35Z",
        "source-ip": "127.0.0.1",
        "host": "127.0.0.1:8080",
        "method": "POST",
        "path": "/restconf/ds/ietf-datastores:running/wn-sztpd-1:devices/device=my-serial-number",
        "outcome": "success"
      }
    ]
  }
}
```

```

    "timestamp": "2020-09-18T23:08:36Z",
    "source-ip": "127.0.0.1",
    "host": "127.0.0.1:9090",
    "method": "POST",
    "path": "/restconf/operations/ietf-sztp-bootstrap-server:get-bootstrapping-data",
    "outcome": "success"
  },
  {
    "timestamp": "2020-09-18T23:08:36Z",
    "source-ip": "127.0.0.1",
    "host": "127.0.0.1:8080",
    "method": "DELETE",
    "path": "/restconf/ds/ietf-datastores:running/wn-sztpd-1:devices/device=my-serial-number/re\
sponse-manager",
    "outcome": "success"
  },
  {
    "timestamp": "2020-09-18T23:08:36Z",
    "source-ip": "127.0.0.1",
    "host": "127.0.0.1:8080",
    "method": "DELETE",
    "path": "/restconf/ds/ietf-datastores:running/wn-sztpd-1:conveyed-information-responses",
    "outcome": "success"
  },
  {
    "timestamp": "2020-09-18T23:08:36Z",
    "source-ip": "127.0.0.1",
    "host": "127.0.0.1:8080",
    "method": "DELETE",
    "path": "/restconf/ds/ietf-datastores:running/wn-sztpd-1:bootstrap-servers",
    "outcome": "success"
  },
  {
    "timestamp": "2020-09-18T23:08:36Z",
    "source-ip": "127.0.0.1",
    "host": "127.0.0.1:8080",
    "method": "POST",
    "path": "/restconf/ds/ietf-datastores:running",
    "outcome": "success"
  },
  {
    "timestamp": "2020-09-18T23:08:36Z",
    "source-ip": "127.0.0.1",
    "host": "127.0.0.1:8080",
    "method": "POST",
    "path": "/restconf/ds/ietf-datastores:running",
    "outcome": "success"
  },
  {
    "timestamp": "2020-09-18T23:08:37Z",
    "source-ip": "127.0.0.1",
    "host": "127.0.0.1:8080",
    "method": "POST",
    "path": "/restconf/ds/ietf-datastores:running",
    "outcome": "success"
  },
  {
    "timestamp": "2020-09-18T23:08:37Z",
    "source-ip": "127.0.0.1",
    "host": "127.0.0.1:8080",
    "method": "POST",
    "path": "/restconf/ds/ietf-datastores:running",
    "outcome": "success"
  },
  {
    "timestamp": "2020-09-18T23:08:37Z",
    "source-ip": "127.0.0.1",
    "host": "127.0.0.1:8080",
    "method": "POST",
    "path": "/restconf/ds/ietf-datastores:running/wn-sztpd-1:devices/device=my-serial-number",
    "outcome": "success"
  },
  {
    "timestamp": "2020-09-18T23:08:37Z",
    "source-ip": "127.0.0.1",
    "host": "127.0.0.1:9090",
    "method": "POST",
    "path": "/restconf/operations/ietf-sztp-bootstrap-server:get-bootstrapping-data",
    "outcome": "success"
  },
  {
    "timestamp": "2020-09-18T23:08:38Z",
    "source-ip": "127.0.0.1",
    "host": "127.0.0.1:9090",
    "method": "POST",
    "path": "/restconf/operations/ietf-sztp-bootstrap-server:report-progress",
    "outcome": "success"
  }

```

```

    },
    {
      "timestamp": "2020-09-18T23:08:38Z",
      "source-ip": "127.0.0.1",
      "host": "127.0.0.1:8080",
      "method": "DELETE",
      "path": "/restconf/ds/ietf-datastores:running/wn-sztpd-1:devices/device=my-serial-number/re\
sponse-manager",
      "outcome": "success"
    },
    {
      "timestamp": "2020-09-18T23:08:38Z",
      "source-ip": "127.0.0.1",
      "host": "127.0.0.1:8080",
      "method": "DELETE",
      "path": "/restconf/ds/ietf-datastores:running/wn-sztpd-1:conveyed-information-responses",
      "outcome": "success"
    },
    {
      "timestamp": "2020-09-18T23:08:38Z",
      "source-ip": "127.0.0.1",
      "host": "127.0.0.1:8080",
      "method": "DELETE",
      "path": "/restconf/ds/ietf-datastores:running/wn-sztpd-1:boot-images",
      "outcome": "success"
    },
    {
      "timestamp": "2020-09-18T23:08:38Z",
      "source-ip": "127.0.0.1",
      "host": "127.0.0.1:8080",
      "method": "DELETE",
      "path": "/restconf/ds/ietf-datastores:running/wn-sztpd-1:scripts",
      "outcome": "success"
    },
    {
      "timestamp": "2020-09-18T23:08:39Z",
      "source-ip": "127.0.0.1",
      "host": "127.0.0.1:8080",
      "method": "DELETE",
      "path": "/restconf/ds/ietf-datastores:running/wn-sztpd-1:configurations",
      "outcome": "success"
    }
  ]
}

```

3.21 View Device's Bootstrapping Log

Notice how the three get-bootstrapping-data RPC requests from above all appear below.

```

===== NOTE: '\ ' line wrapping per RFC 8792 =====

#!/bin/sh

curl -i -X GET --cacert nbi_trust_chain.pem --user my-admin@example.com:my-secret https://127.0.0.1:8080/re\
stconf/ds/ietf-datastores:operational/wn-sztpd-1:devices/device=my-serial-number/bootstrapping-log 1> output\
t/get_bootstrapping_log.out 2>/dev/null

# line-return needed for markdown
echo "" 1>> output/get_bootstrapping_log.out

```

Output:

```

HTTP/1.1 200 OK
Content-Type: application/yang-data+json; charset=utf-8
Server: <redacted>
Content-Length: 6987
Date: Fri, 18 Sep 2020 23:08:39 GMT

{
  "wn-sztpd-1:bootstrapping-log": {
    "log-entry": [
      {
        "timestamp": "2020-09-18T23:08:34Z",
        "source-ip": "127.0.0.1",
        "method": "HEAD",
        "path": "/restconf/ds/ietf-datastores:operational/ietf-yang-library:yang-library",
        "return-code": 200
      },
      {
        "timestamp": "2020-09-18T23:08:35Z",

```

```

"source-ip": "127.0.0.1",
"method": "POST",
"path": "/restconf/operations/ietf-sztp-bootstrap-server:get-bootstrapping-data",
"return-code": 404,
"error-returned": {
  "ietf-restconf:errors": {
    "error": [
      {
        "error-type": "application",
        "error-tag": "data-missing",
        "error-message": "No responses configured."
      }
    ]
  }
},
"event-details": {
  "get-bootstrapping-data-event": {
    "passed-input": {
      "input": [
        {
          "key": "hw-model",
          "value": "model-x"
        },
        {
          "key": "os-name",
          "value": "vendor-os"
        },
        {
          "key": "os-version",
          "value": "17.3R2.1"
        },
        {
          "key": "nonce",
          "value": "extralongbase64encodedvalue="
        }
      ]
    },
    "selected-response": "no-responses-configured"
  }
},
{
  "timestamp": "2020-09-18T23:08:36Z",
  "source-ip": "127.0.0.1",
  "method": "POST",
  "path": "/restconf/operations/ietf-sztp-bootstrap-server:get-bootstrapping-data",
  "return-code": 200,
  "event-details": {
    "get-bootstrapping-data-event": {
      "passed-input": {
        "input": [
          {
            "key": "hw-model",
            "value": "model-x"
          },
          {
            "key": "os-name",
            "value": "vendor-os"
          },
          {
            "key": "os-version",
            "value": "17.3R2.1"
          },
          {
            "key": "nonce",
            "value": "extralongbase64encodedvalue="
          }
        ]
      },
      "selected-response": "catch-all-response",
      "response-details": {
        "managed-response": {
          "conveyed-information": {
            "redirect-information": {
              "referenced-definition": "my-redirect-information"
            }
          }
        }
      }
    }
  }
},
{
  "timestamp": "2020-09-18T23:08:38Z",
  "source-ip": "127.0.0.1",
  "method": "POST",
  "path": "/restconf/operations/ietf-sztp-bootstrap-server:get-bootstrapping-data",

```

```
    "return-code": 200,
    "event-details": {
      "get-bootstrapping-data-event": {
        "passed-input": {
          "input": [
            {
              "key": "hw-model",
              "value": "model-x"
            },
            {
              "key": "os-name",
              "value": "vendor-os"
            },
            {
              "key": "os-version",
              "value": "17.3R2.1"
            },
            {
              "key": "nonce",
              "value": "extralongbase64encodedvalue="
            }
          ]
        },
        "selected-response": "catch-all-response",
        "response-details": {
          "managed-response": {
            "conveyed-information": {
              "onboarding-information": {
                "referenced-definition": "my-onboarding-information"
              }
            }
          }
        }
      }
    },
    "timestamp": "2020-09-18T23:08:38Z",
    "source-ip": "127.0.0.1",
    "method": "POST",
    "path": "/restconf/operations/ietf-sztp-bootstrap-server:report-progress",
    "return-code": 204,
    "event-details": {
      "report-progress-event": {
        "passed-input": {
          "progress-type": "bootstrap-complete",
          "message": "Dynamically generated SSH host key included.",
          "ssh-host-keys": {
            "ssh-host-key": [
              {
                "algorithm": "ssh-rsa",
                "key-data": "base64encodedvalue=="
              }
            ]
          }
        },
        "dynamic-callout-result": {
          "no-webhooks-configured": [
            null
          ]
        }
      }
    }
  ]
}
```


4 Northbound Interface

This section provides a high-level overview of the API using [YANG tree diagrams](#).

Tree diagrams enable the general structure of the data to be understood but, for full understanding, it is necessary to look at the YANG modules themselves.

The YANG modules used by SZTPD can be found in the “yang” directory in the sztpd installation directory (e.g., site-packages/sztpd/yang/*.yang). Each Python installation has its own location for where it installs Python modules. This may vary by the version of Python installed and if virtual environments are used. The following command can be used to find the directory the YANG files are located in:

```
python -c 'import pkg_resources; print(pkg_resources.resource_filename("sztpd", "yang/"))'
```

DISCLAIMER: the YANG models are unusually complex, due to the modules needing to present different product modes.

4.1 NBI Overview

The following sections are presented in sorted order.

4.1.1 Administrator Accounts

The following [tree diagram](#) illustrates the NBI used for configuring “admin-accounts”.

```

+--rw admin-accounts
  +--rw admin-account* [email-address]
    +--rw email-address          string
    +--rw fullname?             string
    +--rw password               iana-crypt-hash:crypt-hash
    +--ro password-last-modified ietf-yang-types:date-and-time
    +--rw access                 enumeration
    +--x resend-activation-email

```

Each account is keyed by an email address. Use of an email address enables mapping to subtenant mapping (for deployments running in product mode ‘x’, as email addresses are globally unique, and otherwise enables the email-based administrator account verification mechanism provided by SZTPD.

4.1.2 Audit Log

The following [tree diagram](#) illustrates the NBI used viewing audit logs (“ro” stands for “read-only”).

```

+--ro audit-log
  +--ro log-entry*
    +--ro timestamp          ietf-yang-types:date-and-time
    +--ro source-ip          ietf-inet-types:ip-address
    +--ro source-proxies*    string
    +--ro host                string
    +--ro method              enumeration
    +--ro path                string
    +--ro outcome             enumeration
    +--ro comment?           string

```

4.1.3 Boot Images

The following [tree diagram](#) illustrates the NBI used for configuring boot-image records.

```
+--rw boot-images {wn-sztpd-0:onboarding-supported}?
  +--rw boot-image* [name]
    +--rw name string
    +--rw os-name? string
    +--rw os-version? string
    +--rw download-uri* ietf-inet-types:uri
    +--rw image-verification* [hash-algorithm]
      | +--rw hash-algorithm identityref
      | +--rw hash-value ietf-yang-types:hex-string
    +--ro reference-statistics
      +--ro reference-count uint32
      +--ro last-referenced union
```

The boot-image records are used in constructing RFC 8572 based “onboarding information” responses.

4.1.4 Bootstrap Servers

The following [tree diagram](#) illustrates the NBI used for configuring “bootstrap-server” records.

```
+--rw bootstrap-servers
  +--rw bootstrap-server* [name]
    +--rw name string
    +--rw address ietf-inet-types:host
    +--rw port? ietf-inet-types:port-number <443>
    +--rw trust-anchor? ietf-crypto-types:trust-anchor-cert-cms
    +--ro reference-statistics
      +--ro reference-count uint32
      +--ro last-referenced union
```

The bootstrap-server records are used in constructing RFC 8572 based “redirect information” responses.

4.1.5 Configurations

The following [tree diagram](#) illustrates the NBI used for configuring configuration records.

```
+--rw configurations {wn-sztpd-0:onboarding-supported}?
  +--rw configuration* [name]
    +--rw name string
    +--rw configuration-handling? enumeration
    +--rw config? binary
    +--ro reference-statistics
      +--ro reference-count uint32
      +--ro last-referenced union
```

The configuration records are used in constructing RFC 8572 based “onboarding information” responses.

4.1.6 Conveyed Information Responses

The following [tree diagram](#) illustrates the NBI used for configuring conveyed-information-responses records.

```

+--rw conveyed-information-responses
  +--rw redirect-information-response* [name]
    | +--rw name string
    | +--rw redirect-information
    | | +--rw bootstrap-server* -> ../../../../bootstrap-servers/bootstrap-server/name
    | +--ro reference-statistics
    | | +--ro reference-count uint32
    | | +--ro last-referenced union
    +--rw onboarding-information-response* [name] {wn-sztpd-0:onboarding-supported}?
      +--rw name string
      +--rw onboarding-information
        | +--rw boot-image -> ../../../../boot-images/boot-image/name
        | +--rw pre-configuration-script? -> ../../../../scripts/pre-configuration-script/name
        | +--rw configuration? -> ../../../../configurations/configuration/name
        | +--rw post-configuration-script? -> ../../../../scripts/post-configuration-script/name
        +--ro reference-statistics
          +--ro reference-count uint32
          +--ro last-referenced union
  
```

The conveyed-information-responses records are used in constructing RFC 8572 based “redirect” and “onboarding” information responses.

4.1.7 Device-Types

The “/device-types” tree enables configuration of device types to be recognized by the system.

Not only is being able to associate a device with a type helpful on its own, but each “device-type” entry enables more than just that, as explained below.

The device-types list contains an optional reference to the specific trust-anchor certificate in the Keystore used by the [transport](#). This enables application specific verification that the device’s client certificate (i.e., its IDDevID) exactly matches the expected certificate chain.

Similarly, the device-types list contains an optional “ownership-authorization” webhook enabling SZTPD to ensure that the specific tenant is the rightful owner of configured serial-numbers.

Important: In the “tenant” view, this entire data-tree is read-only. That is, all the “rw” would be shown as “ro” if this were the tenant view’s tree diagram. The /device-types tree is read-only in the “tenant” view because device types can only be set at the system level via the ‘native’ view.

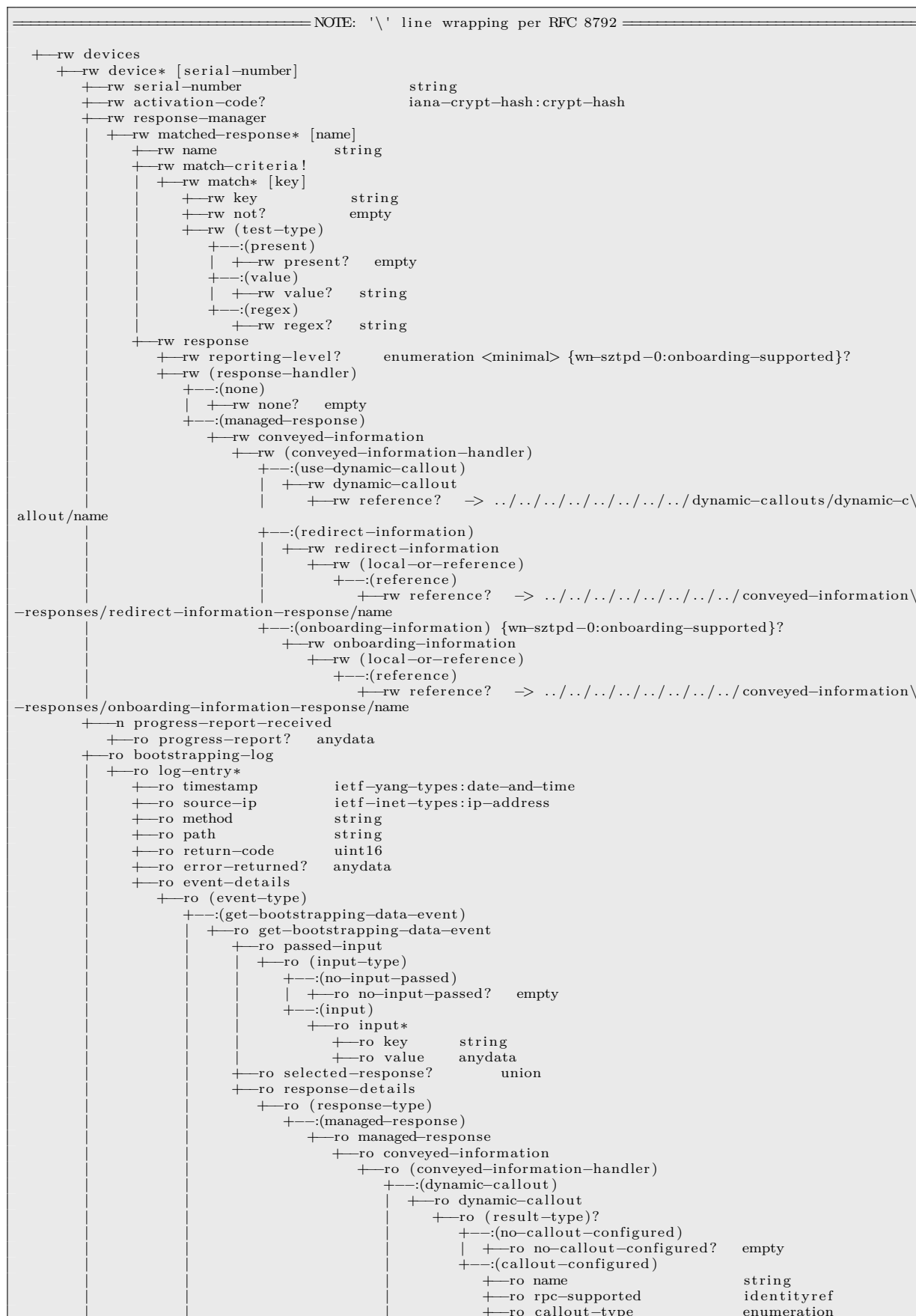
The following [tree diagram](#) illustrates the NBI used for configuring device records.

```

===== NOIE: '\ ' line wrapping per RFC 8792 =====
+--rw device-types
  +--rw device-type* [name]
    +--rw name string
    +--rw identity-certificates!
      | +--rw verification
      | | +--rw local-truststore-reference {ietf-truststore:certificates}?
      | | | +--rw certificate-bag -> /truststore/certificate-bags/certificate-bag/name
      | | | +--rw certificate -> /truststore/certificate-bags/certificate-bag[name = current()]/
      | | | .. /certificate-bag/certificate/name
      | | +--rw serial-number-extraction? identityref <wn-x509-c2n:serial-number>
      +--rw ownership-authorization!
        | +--rw dynamic-callout
        | | +--rw reference? -> /dynamic-callouts/dynamic-callout/name
      +--rw voucher-aquisition!
        +--rw dynamic-callout
        | +--rw reference? -> /dynamic-callouts/dynamic-callout/name
  
```

4.1.8 Devices

The following [tree diagram](#) illustrates the NBI used for configuring device records.



uri	<pre> +ro (callback-or-webhook)? +--:(callback) +ro callback-details +ro plugin string +ro function string +ro callback-results +ro (exit-status) +--:(exception-thrown) +ro exception-thrown? string +--:(exited-normally) +ro exited-normally? string +--:(webhook) {wn-app:webhook-based-callouts}? +ro webhook-details +ro num-webhooks-configured uint8 +ro webhook-results +ro webhook* +ro name string +ro uri ietf-inet-types:\ +ro (result-type)? +--:(connection-error) +ro connection-error? string +--:(http-status-code) +ro http-status-code string +--:(redirect-information) +ro redirect-information +ro (local-or-reference) +--:(reference) +ro referenced-definition? string +--:(onboarding-information) {wn-sztpd-0:onboarding-supported}? +ro onboarding-information +ro (local-or-reference) +--:(reference) +ro referenced-definition? string +--:(report-progress-event) {wn-sztpd-0:onboarding-supported}? +ro report-progress-event +ro passed-input +ro progress-type string +ro message? string +ro ssh-host-keys? anydata +ro trust-anchor-certs? anydata +ro dynamic-callout +ro (result-type)? +--:(no-callout-configured) +ro no-callout-configured? empty +--:(callout-configured) +ro name string +ro rpc-supported identityref +ro callout-type enumeration +ro (callback-or-webhook)? +--:(callback) +ro callback-details +ro plugin string +ro function string +ro callback-results +ro (exit-status) +--:(exception-thrown) +ro exception-thrown? string +--:(exited-normally) +ro exited-normally? string +--:(webhook) {wn-app:webhook-based-callouts}? +ro webhook-details +ro num-webhooks-configured uint8 +ro webhook-results +ro webhook* +ro name string +ro uri ietf-inet-types:uri +ro (result-type)? +--:(connection-error) +ro connection-error? string +--:(http-status-code) +ro http-status-code string +ro lifecycle-statistics +ro nbi-access-stats +ro created ietf-yang-types:date-and-time +ro num-times-modified uint16 +ro last-modified ietf-yang-types:date-and-time +ro sbi-access-stats +ro num-times-accessed uint16 +ro first-accessed ietf-yang-types:date-and-time +ro last-accessed ietf-yang-types:date-and-time +rw device-type -> /device-types/device-type/name </pre>
-----	--

Device records are keyed by serial number, enabling device to tenant mapping, as serial numbers are globally unique.

Each device record includes the device's lifecycle statistics (key north and south bound interactions) and bootstrapping log (a history of device-initiated interactions)

The device records are used in constructing RFC 8572 based "redirect" and "onboarding" information responses.

4.1.9 Dynamic Callouts

The following [tree diagram](#) illustrates the NBI used for configuring dynamic callouts, which may either be implemented via a call into a plugin or via a webhook¹.

```

===== NOIE: '\ ' line wrapping per RFC 8792 =====
+--rw dynamic-callouts
  +--rw dynamic-callout* [name]
    +--rw name string
    +--rw rpc-supported identityref
    +--rw (callout-type)
      +--:(use-callback)
        +--rw callback
          +--rw plugin -> /preferences/system/plugins/plugin/name
          +--rw function -> /preferences/system/plugins/plugin[name = current()]/plugin/function/name
      +--:(use-webhooks) {wn-app:webhook-based-callouts}?
        +--rw webhooks
          +--rw webhook* [name]
            +--rw name string
            +--rw encoding? enumeration <json>
            +--rw (transport)
              +--:(https) {ietf-restconf-client:https-initiate}?
                +--rw https
                  +--rw tcp-client-parameters
                    +--rw remote-address ietf-inet-types:host
                    +--rw remote-port? ietf-inet-types:port-number <443>
                  +--rw tls-client-parameters
                    +--rw client-identity!
                    +--rw (auth-type)
                      +--:(certificate) {ietf-tls-client:x509-certificate-auth}?
                        +--rw certificate
                          +--rw (local-or-keystore)
                            +--:(local-keystore)
                              +--rw local-keystore-reference
                                +--rw asymmetric-key -> ../keystore/asymmetric-keys/asymmetric-key/name
                              +--rw certificate -> ../keystore/asymmetric-keys/asymmetric-key[name = current()]/certificates/certificate/name
                            +--rw server-authentication
                              +--rw ca-certs! {ietf-tls-client:x509-certificate-auth}?
                                +--rw (local-or-truststore)
                                  +--:(local-truststore) {ietf-truststore:certificates}?
                                    +--rw local-truststore-reference? -> ../truststore/certificate-bags/certificate-bag/name
                                +--rw ee-certs! {ietf-tls-client:x509-certificate-auth}?
                                  +--rw (local-or-truststore)
                                    +--:(local-truststore) {ietf-truststore:certificates}?
                                      +--rw local-truststore-reference? -> ../truststore/certificate-bags/certificate-bag/name
                              +--rw http-client-parameters
                                +--rw client-identity!
                                +--rw (auth-type)
                                  +--:(basic)
                                    +--rw basic {ietf-http-client:basic-auth}?
                                      +--rw user-id string
                                      +--rw (password-type)
                                        +--:(cleartext-password)
                                          +--rw cleartext-password? string
                                        +--:(encrypted-password) {ietf-crypto-types:password-encrypt\
ion}?
                                          +--rw encrypted-password
                                            +--rw encrypted-by
                                              +--rw encrypted-value binary
                                  +--:(restconf-client-parameters)
                                    +--:(http)
                                      +--rw http
                                        +--rw tcp-client-parameters
                                          +--rw remote-address ietf-inet-types:host

```

¹A list of webhooks are configurable only for HA reasons.

```

| +-rw remote-port?      ietf-inet-types:port-number <80>
+-rw http-client-parameters
  +-rw client-identity!
    +-rw (auth-type)
      +--:(basic)
        +-rw basic {ietf-http-client:basic-auth}?
          +-rw user-id      string
          +-rw (password-type)
            +--:(cleartext-password)
              | +-rw cleartext-password?  string
              +--:(encrypted-password) {ietf-crypto-types:password-encrypt\
ion}?
                +-rw encrypted-password
                  +-rw encrypted-by
                    +-rw encrypted-value  binary
+-rw round-robin!
  +-rw amount      uint16
  +-rw units       enumeration

```

4.1.10 Keystore

The following [tree diagram](#) illustrates the NBI used for configuring keys held in the keystore.

```

===== NOIE: '\ ' line wrapping per RFC 8792 =====
+-rw keystore
  +-rw asymmetric-keys
    +-rw asymmetric-key* [name]
      +-rw name                string
      +-rw public-key-format    identityref
      +-rw public-key          binary
      +-rw private-key-format?  identityref
    +-rw (private-key-type)
      +--:(cleartext-private-key)
        | +-rw cleartext-private-key?  binary
        +--:(hidden-private-key)
          | +-rw hidden-private-key?  empty
          +--:(encrypted-private-key) {ietf-crypto-types:private-key-encryption}?
            +-rw encrypted-private-key
              +-rw encrypted-by
                +-rw encrypted-value  binary
  +-rw certificates
    +-rw certificate* [name]
      +-rw name                string
      +-rw cert-data           ietf-crypto-types:end-entity-cert-cms
      +-n certificate-expiration {ietf-crypto-types:certificate-expiration-notification}?
        +-ro expiration-date   ietf-yang-types:date-and-time
      +-ro reference-statistics
        +-ro reference-count    uint32
        +-ro last-referenced    union
    +-x generate-certificate-signing-request {ietf-crypto-types:certificate-signing-request-genera\
tion}?
      | +-rw input
      | | +-w csr-info      ietf-crypto-types:csr-info
      | +-rw output
      | | +-ro certificate-signing-request  ietf-crypto-types:csr
      +-ro reference-statistics
        +-ro reference-count    uint32
        +-ro last-referenced    union
  +-rw symmetric-keys
    +-rw symmetric-key* [name]
      +-rw name                string
      +-rw key-format?         identityref
    +-rw (key-type)
      +--:(cleartext-key)
        | +-rw cleartext-key?  binary
        +--:(hidden-key)
          | +-rw hidden-key?  empty
          +--:(encrypted-key) {ietf-crypto-types:symmetric-key-encryption}?
            +-rw encrypted-key
              +-rw encrypted-by
                +-rw encrypted-value  binary

```

The keys held in the keystore are used in the [transport](#) configuration, e.g., to hold SZTPD's private key used for each listening port (see the [Listening Ports and the APIs They Present](#) in the [Installation Guide](#)).

4.1.11 Preferences

The following [tree diagram](#) illustrates the NBI used for configuring system and tenant-level preferences.

```

+--rw preferences
  +--rw admin-accounts
    | +--rw new-account-verification
    | | +--rw subject? string <SZTPD new account verification required.>
    | | +--rw cc? string <$CREATOR-FULLNAME <CREATOR-EMAIL-ADDRESS>>
    | | +--rw body? string <$ACCOUNT-FULLNAME
  >
  You are receiving this email because I created a new account
  for you on the SZTPD ($SZTPD-SERVER-ADDRESS). You
  account's login is "$ACCOUNT-EMAIL-ADDRESS".
  >
  Please activate your account via the following URL:
  $VERIFICATION-URI (This URL expires in 48-hours).
  >
  Note that I CC-ed myself on this email and you may contact
  me to verify the veracity of this email.
  >
  Thanks,
  $CREATOR-FULLNAME
  >
  | +--rw passwords
  | | +--rw strength-testing!
  | | | +--rw min-length? uint16 <16>
  +--rw notification-delivery!
  | +--rw dynamic-callout
  | | +--rw reference -> ../../../../dynamic-callouts/dynamic-callout/name
  +--rw system
  +--rw hostname? ietf-inet-types:host
  +--rw email-from-address? string <SZTPD Administrator <root@$SZTPD-SERVER-ADDRESS>>
  +--rw features
  | +--rw onboarding-supported? boolean <true> {wn-sztpd-0:onboarding-supported}?
  +--rw plugins
  +--rw plugin* [name]
  | +--rw name string
  | +--rw functions
  | | +--rw function* [name]
  | | +--rw name string
  >

```

4.1.12 Scripts

The following [tree diagram](#) illustrates the NBI used for configuring script records.

```

+--rw scripts {wn-sztpd-0:onboarding-supported}?
  +--rw pre-configuration-script* [name]
  | +--rw name string
  | +--rw script? ietf-sztp-conveyed-info:script
  | +--ro reference-statistics
  | | +--ro reference-count uint32
  | | +--ro last-referenced union
  +--rw post-configuration-script* [name]
  | +--rw name string
  | +--rw script? ietf-sztp-conveyed-info:script
  | +--ro reference-statistics
  | | +--ro reference-count uint32
  | | +--ro last-referenced union
  >

```

The script records are used in constructing RFC 8572 based “onboarding information” responses.

4.1.13 Tenants

For deployments using product mode ‘x’, the following [tree diagram](#) illustrates a snippet of the NBI used for configuring tenants.

```

+--rw tenants
  +--rw tenant* [name]
  >

```

Unlike all the other [Northbound Interface](#) sections, this section illustrates only the place in the hierarchy where the tenant containers are defined. Not shown are that each tenant container contains an instance of every other [Northbound Interface](#) section in it as well (with exception to the [transport](#), which can only be configured by the host system).

4.1.14 Transport

The following [tree diagram](#) illustrates the NBI used for configuring transport records.

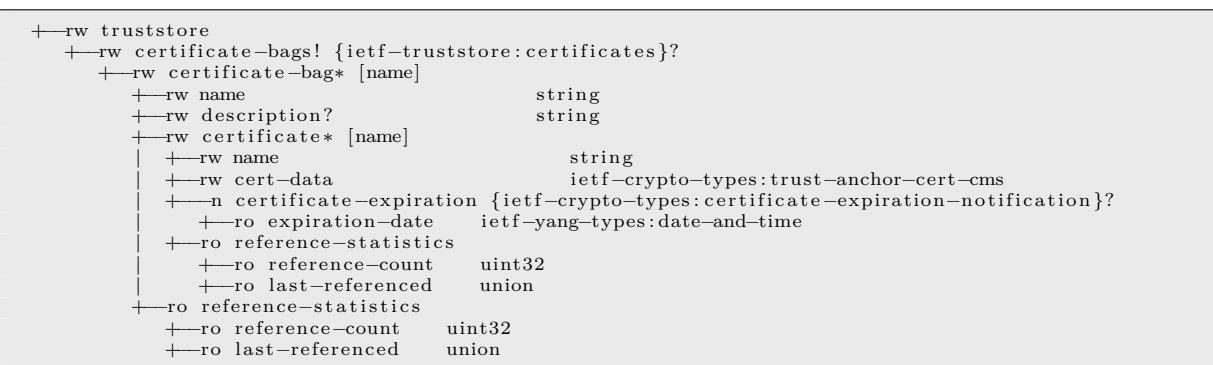


Each device records includes the device's bootstrapping log, a history of interactions with the device.

The transport configuration defines the listening ports SZTPD opens.

4.1.15 Truststore

The following [tree diagram](#) illustrates the NBI used for configuring trust-anchors held in the truststore.



The trust-anchors held in the truststore are use for many purposes (e.g., to authenticate clients, to authenticate remote systems, etc.).

4.2 Advanced Features

4.2.1 Triggers

4.2.1.1 Configuration-based

SZTPD reacts to some triggers automatically.

SZTPD reacts to some configurations chanfges. Simple examples include:

- automatically hash client passwords.
- automatically SIGHUP when transport is configured.
- incrementing/decrementing opstate counters.

Configuration-based triggers are discussed further in [Tracking](#).

4.2.1.2 Clock-based

SZTPD may set clock triggers. For instance, to note when an object should be purged.

4.2.2 Tracking

Much of SZTPD's reactive functionality depends on triggers.

4.2.2.1 Reference Statistics

SZTPD will be able to track where referenced objects are used so as to detect when the objects are no longer referenced. Timestamps track when the object was created, last modified, first referenced, and last referenced.

4.2.2.2 Expiration Tracking

Known when an object was last referenced enables SZTPD to determine when the object will be subject to expiration, unless updated or referenced in the interim.

When implemented, SZTPD sends notifications with expedited urgency when an object is getting close to expiration.

4.2.2.3 Device Lifecycle Statistics

Devices are special objects in that nothing references them and yet it is desired to purge them when appropriate and, furthermore, to track if/when a device performed bootstrapping events.

4.2.3 Dyanmic Callouts

SZTPD uses “dynamic callouts” to send data to or request data from another system. Each dynamic callout may be implemented as a plugin (in order to use a proprietary API) or as a “webhook”, which is an HTTP “POST” request.

4.2.3.1 Notification Logging

SZTPD uses a dynamic callout that implements the “[relay-notification](#)” RPC, defined in `wn-app-rpcs.yang`, to send notifications to remote systems.

Which dynamic callout to use is configured under to top-level (or per-tenant) “[preferences](#)” subtree.

4.2.3.2 Device Ownership Authorization

SZTPD uses a dynamic callout that implements the “[verify-device-ownership](#)” RPC, defined in `wn-sztpd-rpcs.yang`, to ascertain if a `tenant` is the rightful owner of a device, by way of the device’s serial number.

Which dynamic callout to use is configured under the top-level “[device-types](#)” subtree.

4.2.3.3 Dynamic Bootstrapping Response SZTPD uses a dynamic callout that implements the “[get-conveyed-information](#)” RPC, defined in `wn-sztpd-rpcs.yang`, to obtain a dynamic response to send to a bootstrapping device.

Which dynamic callout to use is configured as part of each device’s “[response-manager](#)” subtree (e.g., `response-manager/response/conveyed-information/dynamic-callout`). Please see the [Devices](#) for details.

4.3 Complete Tree Diagrams

This section presents the complete tree diagrams for Mode-1 and Mode-x (native). Modes ‘1’ and ‘x’ are discussed in the “First-time Initialization” section in the [Installation Guide](#).

The tree diagrams provide an overview of the API’s data model but, for complete understanding of the YANG model, it is necessary to look at the YANG modules directly. The YANG modules are installed as part of the SZTPD package. Run the following command to discover the location for the YANG modules on your system:

```
python -c 'import pkg_resources; print(pkg_resources.resource_filename("sztpd", "yang/"))'
```

4.3.1 Mode-1’s native view

The following [tree diagram](#) illustrates the entire NBI, from the perspective of the Mode ‘1’ native interface.

```
===== NOTE: '\ ' line wrapping per RFC 8792 =====
module: wn-sztpd-1
  +-rw preferences
  |   +-rw admin-accounts
  |   |   +-rw new-account-verification
  |   |   |   +-rw subject?      string <SZTPD new account verification required.>
  |   |   |   +-rw cc?          string <CREATOR-FULLNAME <CREATOR-EMAIL-ADDRESS>>
  |   |   |   +-rw body?        string <ACCOUNT-FULLNAME>
  |   |   |
  |   |   You are receiving this email because I created a new account
  |   |   for you on the SZTPD ($SZTPD-SERVER-ADDRESS). You
  |   |   account's login is "%ACCOUNT-EMAIL-ADDRESS".
```

Please activate your account via the following URL:
\$VERIFICATION-URI (This URL expires in 48-hours).

Note that I CC-ed myself on this email and you may contact me to verify the veracity of this email.

Thanks,

\$CREATOR-FULLNAME

>

```

+--rw passwords
+--rw strength-testing!
+--rw min-length? uint16 <16>
+--rw notification-delivery!
+--rw dynamic-callout
+--rw reference -> ../../../../dynamic-callouts/dynamic-callout/name
+--rw system
+--rw hostname? ietf-inet-types:host
+--rw email-from-address? string <SZTPD Administrator <root@$SZTP-SERVER-ADDRESS>>
+--rw features
+--rw onboarding-supported? boolean <true> {wn-sztpd-0:onboarding-supported}?
+--rw plugins
+--rw plugin* [name]
+--rw name string
+--rw functions
+--rw function* [name]
+--rw name string
+--rw admin-accounts
+--rw admin-account* [email-address]
+--rw email-address string
+--rw fullname? string
+--rw password iana-crypt-hash:crypt-hash
+--ro password-last-modified ietf-yang-types:date-and-time
+--rw access enumeration
+--x resend-activation-email
+--ro audit-log
+--ro log-entry*
+--ro timestamp ietf-yang-types:date-and-time
+--ro source-ip ietf-inet-types:ip-address
+--ro source-proxies* string
+--ro host string
+--ro method enumeration
+--ro path string
+--ro outcome enumeration
+--ro comment? string
+--rw truststore
+--rw certificate-bags! {ietf-truststore:certificates}?
+--rw certificate-bag* [name]
+--rw name string
+--rw description? string
+--rw certificate* [name]
+--rw name string
+--rw cert-data ietf-crypto-types:trust-anchor-cert-cms
+--n certificate-expiration {ietf-crypto-types:certificate-expiration-notification}?
+--ro expiration-date ietf-yang-types:date-and-time
+--ro reference-statistics
+--ro reference-count uint32
+--ro last-referenced union
+--ro reference-statistics
+--ro reference-count uint32
+--ro last-referenced union
+--rw keystore {not system-level-administration-disabled}?
+--rw asymmetric-keys
+--rw asymmetric-key* [name]
+--rw name string
+--rw public-key-format identityref
+--rw public-key binary
+--rw private-key-format? identityref
+--rw (private-key-type)
+--:(cleartext-private-key)
+--rw cleartext-private-key? binary
+--:(hidden-private-key)
+--rw hidden-private-key? empty
+--:(encrypted-private-key) {ietf-crypto-types:private-key-encryption}?
+--rw encrypted-private-key
+--rw encrypted-by
+--rw encrypted-value binary
+--rw certificates
+--rw certificate* [name]
+--rw name string
+--rw cert-data ietf-crypto-types:end-entity-cert-cms
+--n certificate-expiration {ietf-crypto-types:certificate-expiration-notification}?
+--ro expiration-date ietf-yang-types:date-and-time
+--ro reference-statistics
+--ro reference-count uint32
+--ro last-referenced union
+--x generate-certificate-signing-request {ietf-crypto-types:certificate-signing-request-genera\

```

```

tion}?
|
| +--rw input
| | +--w csr-info      ietf-crypto-types:csr-info
| | +--rw output
| |   +--ro certificate-signing-request      ietf-crypto-types:csr
+--rw reference-statistics
| +--ro reference-count      uint32
| +--ro last-referenced      union
+--rw symmetric-keys
| +--rw symmetric-key* [name]
| | +--rw name      string
| | +--rw key-format?      identityref
| | +--rw (key-type)
| |   +--:(cleartext-key)
| |   | +--rw cleartext-key?      binary
| |   +--:(hidden-key)
| |   | +--rw hidden-key?      empty
| |   +--:(encrypted-key) {ietf-crypto-types:symmetric-key-encryption}?
| |   | +--rw encrypted-key
| |   |   +--rw encrypted-by
| |   |   +--rw encrypted-value      binary
+--rw dynamic-callouts
| +--rw dynamic-callout* [name]
| | +--rw name      string
| | +--rw rpc-supported      identityref
| | +--rw (callout-type)
| |   +--:(use-callback)
| |   | +--rw callback
| |   |   +--rw plugin      -> /preferences/system/plugins/plugin/name
| |   |   +--rw function      -> /preferences/system/plugins/plugin[name = current()../plugin]/function/name
| |   +--:(use-webhooks) {wn-app:webhook-based-callouts}?
| |   +--rw webhooks
| |     +--rw webhook* [name]
| |     | +--rw name      string
| |     | +--rw encoding?      enumeration <json>
| |     | +--rw (transport)
| |     |   +--:(https) {ietf-restconf-client:https-initiate}?
| |     |   | +--rw https
| |     |   |   +--rw tcp-client-parameters
| |     |   |   | +--rw remote-address      ietf-inet-types:host
| |     |   |   | +--rw remote-port?      ietf-inet-types:port-number <443>
| |     |   |   +--rw tls-client-parameters
| |     |   |   | +--rw client-identity!
| |     |   |   |   +--rw (auth-type)
| |     |   |   |   | +--:(certificate) {ietf-tls-client:x509-certificate-auth}?
| |     |   |   |   |   +--rw certificate
| |     |   |   |   |   | +--rw (local-or-keystore)
| |     |   |   |   |   |   +--:(local-keystore)
| |     |   |   |   |   |   | +--rw local-keystore-reference
| |     |   |   |   |   |   |   +--rw asymmetric-key      -> ../../../../../../../../../../\
| |     |   |   |   |   |   |   | +--rw certificate      -> ../../../../../../../../../../\
| |     |   |   |   |   |   |   |   +--rw server-authentication
| |     |   |   |   |   |   |   |   | +--rw ca-certs! {ietf-tls-client:x509-certificate-auth}?
| |     |   |   |   |   |   |   |   | +--rw (local-or-truststore)
| |     |   |   |   |   |   |   |   |   +--:(local-truststore) {ietf-truststore:certificates}?
| |     |   |   |   |   |   |   |   |   | +--rw local-truststore-reference?      -> ../../../../../../../../../../\
| |     |   |   |   |   |   |   |   |   |   +--rw ee-certs! {ietf-tls-client:x509-certificate-auth}?
| |     |   |   |   |   |   |   |   |   |   | +--rw (local-or-truststore)
| |     |   |   |   |   |   |   |   |   |   |   +--:(local-truststore) {ietf-truststore:certificates}?
| |     |   |   |   |   |   |   |   |   |   |   | +--rw local-truststore-reference?      -> ../../../../../../../../../../\
| |     |   |   |   |   |   |   |   |   |   |   |   +--rw http-client-parameters
| |     |   |   |   |   |   |   |   |   |   |   |   | +--rw client-identity!
| |     |   |   |   |   |   |   |   |   |   |   |   |   | +--rw (auth-type)
| |     |   |   |   |   |   |   |   |   |   |   |   |   |   +--:(basic)
| |     |   |   |   |   |   |   |   |   |   |   |   |   |   | +--rw basic {ietf-http-client:basic-auth}?
| |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   +--rw user-id      string
| |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   | +--rw (password-type)
| |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | +--:(cleartext-password)
| |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | +--rw cleartext-password?      string
| |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   +--:(encrypted-password) {ietf-crypto-types:password-encrypt\
| |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | +--rw encrypted-password
| |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   +--rw encrypted-by
| |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | +--rw encrypted-value      binary
| |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   +--rw restconf-client-parameters
| |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   +--:(http)
| |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | +--rw http
| |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   +--rw tcp-client-parameters
| |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | +--rw remote-address      ietf-inet-types:host
| |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | +--rw remote-port?      ietf-inet-types:port-number <80>
| |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | +--rw http-client-parameters
| |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | +--rw client-identity!

```

```

    +-rw (auth-type)
    +--:(basic)
        +-rw basic {ietf-http-client:basic-auth}?
            +-rw user-id string
            +-rw (password-type)
                +--:(cleartext-password)
                    | +-rw cleartext-password? string
                    +--:(encrypted-password) {ietf-crypto-types:password-encrypt\
ion}?
                        +-rw encrypted-password
                            +-rw encrypted-by
                                +-rw encrypted-value binary
            +-rw round-robin!
                +-rw amount uint16
                +-rw units enumeration
+-rw transport {not system-level-administration-disabled}?
+-rw listen! {ietf-restconf-server:http-listen or ietf-restconf-server:https-listen}?
+-rw endpoint* [name]
    +-rw name string
    +-rw (transport)
        +--:(http) {ietf-restconf-server:http-listen}?
            +-rw http
                +-rw external-endpoint!
                    | +-rw address ietf-inet-types:ip-address
                    | +-rw port? ietf-inet-types:port-number <443>
                +-rw tcp-server-parameters
                    | +-rw local-address ietf-inet-types:ip-address
                    | +-rw local-port? ietf-inet-types:port-number <80>
                +-rw http-server-parameters
                    | +-rw server-name? string
                +-rw restconf-server-parameters
                    +-rw client-identity-mappings
                        +-rw cert-to-name* [id]
                            +-rw id uint32
                            +-rw fingerprint? ietf-x509-cert-to-name:tls-fingerprint
                            +-rw map-type identityref
                            +-rw name string
        +--:(https) {ietf-restconf-server:https-listen}?
            +-rw https
                +-rw tcp-server-parameters
                    | +-rw local-address ietf-inet-types:ip-address
                    | +-rw local-port? ietf-inet-types:port-number <443>
                +-rw tls-server-parameters
                    +-rw server-identity
                    +-rw (auth-type)
                        +--:(certificate) {ietf-tls-server:x509-certificate-auth}?
                            +-rw certificate
                                +-rw (local-or-keystore)
                                    +--:(local-keystore)
                                        +-rw reference
                                            +-rw asymmetric-key? -> /keystore/asymmetric-keys/asymmetric\
-key/name
                                                +-rw certificate? -> /keystore/asymmetric-keys/asymmetric\
-key[name = current()]/../asymmetric-key]/certificates/certificate/name
                            +-rw client-authentication! {ietf-tls-server:client-auth-config-supported}?
                                +-rw ca-certs! {ietf-tls-server:x509-certificate-auth}?
                                    +-rw (local-or-truststore)
                                        +--:(local-truststore) {ietf-truststore:certificates,ietf-tls-server:client\
-auth-config-supported}?
                                            +-rw local-truststore-reference? -> /truststore/certificate-bags/cert\
ificate-bag/name
                                +-rw ee-certs! {ietf-tls-server:x509-certificate-auth}?
                                    +-rw (local-or-truststore)
                                        +--:(local-truststore) {ietf-truststore:certificates,ietf-tls-server:client\
-auth-config-supported}?
                                            +-rw local-truststore-reference? -> /truststore/certificate-bags/cert\
ificate-bag/name
                    +-rw http-server-parameters
                        | +-rw server-name? string
                    +-rw restconf-server-parameters
                        +-rw client-identity-mappings
                            +-rw cert-to-name* [id]
                                +-rw id uint32
                                +-rw fingerprint? ietf-x509-cert-to-name:tls-fingerprint
                                +-rw map-type identityref
                                +-rw name string
                    +-rw use-for enumeration
+-rw device-types
+-rw device-type* [name]
    +-rw name string
+-rw identity-certificates!
    +-rw verification
        +-rw local-truststore-reference {ietf-truststore:certificates}?
            +-rw certificate-bag -> /truststore/certificate-bags/certificate-bag/name
            +-rw certificate -> /truststore/certificate-bags/certificate-bag[name = current()]\
../certificate-bag]/certificate/name
        +-rw serial-number-extraction? identityref <wn-x509-c2n:serial-number>

```

```

+rw ownership-authorization!
| +rw dynamic-callout
|   +rw reference? -> /dynamic-callouts/dynamic-callout/name
+rw voucher-aquisition!
| +rw dynamic-callout
|   +rw reference? -> /dynamic-callouts/dynamic-callout/name
+rw bootstrap-servers
+rw bootstrap-server* [name]
| +rw name string
| +rw address ietf-inet-types:host
| +rw port? ietf-inet-types:port-number <443>
| +rw trust-anchor? ietf-crypto-types:trust-anchor-cert-cms
+ro reference-statistics
| +ro reference-count uint32
| +ro last-referenced union
+rw boot-images {wn-sztpd-0:onboarding-supported}?
+rw boot-image* [name]
| +rw name string
| +rw os-name? string
| +rw os-version? string
| +rw download-uri* ietf-inet-types:uri
| +rw image-verification* [hash-algorithm]
| | +rw hash-algorithm identityref
| | +rw hash-value ietf-yang-types:hex-string
+ro reference-statistics
| +ro reference-count uint32
| +ro last-referenced union
+rw scripts {wn-sztpd-0:onboarding-supported}?
+rw pre-configuration-script* [name]
| +rw name string
| +rw script? ietf-sztp-conveyed-info:script
+ro reference-statistics
| +ro reference-count uint32
| +ro last-referenced union
+rw post-configuration-script* [name]
| +rw name string
| +rw script? ietf-sztp-conveyed-info:script
+ro reference-statistics
| +ro reference-count uint32
| +ro last-referenced union
+rw configurations {wn-sztpd-0:onboarding-supported}?
+rw configuration* [name]
| +rw name string
| +rw configuration-handling? enumeration
| +rw config? binary
+ro reference-statistics
| +ro reference-count uint32
| +ro last-referenced union
+rw conveyed-information-responses
+rw redirect-information-response* [name]
| +rw name string
| +rw redirect-information
| | +rw bootstrap-server* -> ../../../../bootstrap-servers/bootstrap-server/name
+ro reference-statistics
| +ro reference-count uint32
| +ro last-referenced union
+rw onboarding-information-response* [name] {wn-sztpd-0:onboarding-supported}?
| +rw name string
| +rw onboarding-information
| | +rw boot-image -> ../../../../boot-images/boot-image/name
| | +rw pre-configuration-script? -> ../../../../scripts/pre-configuration-script/name
| | +rw configuration? -> ../../../../configurations/configuration/name
| | +rw post-configuration-script? -> ../../../../scripts/post-configuration-script/name
+ro reference-statistics
| +ro reference-count uint32
| +ro last-referenced union
+rw devices
+rw device* [serial-number]
| +rw serial-number string
| +rw activation-code? iana-crypt-hash:crypt-hash
+rw response-manager
| +rw matched-response* [name]
| | +rw name string
| | +rw match-criteria!
| | | +rw match* [key]
| | | | +rw key string
| | | | +rw not? empty
| | | | +rw (test-type)
| | | | | +rw (present)
| | | | | | +rw present? empty
| | | | | | +rw (value)
| | | | | | | +rw value? string
| | | | | | +rw (regex)
| | | | | | | +rw regex? string
| | +rw response
| | | +rw reporting-level? enumeration <minimal> {wn-sztpd-0:onboarding-supported}?
| | +rw (response-handler)

```

	<pre> +---:(none) +--rw none? empty +---:(managed-response) +--rw conveyed-information +--rw (conveyed-information-handler) +---:(use-dynamic-callout) +--rw dynamic-callout +--rw reference? -> ../../../../dynamic-callouts/dynamic-c\ </pre>
allout/name	<pre> +---:(redirect-information) +--rw redirect-information +--rw (local-or-reference) +---:(reference) +--rw reference? -> ../../../../conveyed-information\ </pre>
-responses/redirect-information-response/name	<pre> +---:(onboarding-information) {wn-sztpd-0:onboarding-supported}? +--rw onboarding-information +--rw (local-or-reference) +---:(reference) +--rw reference? -> ../../../../conveyed-information\ </pre>
-responses/onboarding-information-response/name	<pre> +--n progress-report-received +--ro progress-report? anydata +--ro bootstrapping-log +--ro log-entry* +--ro timestamp ietf-yang-types:date-and-time +--ro source-ip ietf-inet-types:ip-address +--ro method string +--ro path string +--ro return-code uint16 +--ro error-returned? anydata +--ro event-details +--ro (event-type) +---:(get-bootstrapping-data-event) +--ro get-bootstrapping-data-event +--ro passed-input +--ro (input-type) +---:(no-input-passed) +--ro no-input-passed? empty +---:(input) +--ro input* +--ro key string +--ro value anydata +--ro selected-response? union +--ro response-details +--ro (response-type) +---:(managed-response) +--ro managed-response +--ro conveyed-information +--ro (conveyed-information-handler) +---:(dynamic-callout) +--ro dynamic-callout +--ro (result-type)? +---:(no-callout-configured) +--ro no-callout-configured? empty +---:(callout-configured) +--ro name string +--ro rpc-supported identityref +--ro callout-type enumeration +--ro (callback-or-webhook)? +---:(callback) +--ro callback-details +--ro plugin string +--ro function string +--ro callback-results +--ro (exit-status) +---:(exception-thrown) +--ro exception-thrown? string +---:(exited-normally) +--ro exited-normally? string +---:(webhook) {wn-app:webhook-based-callouts}? +--ro webhook-details +--ro num-webhooks-configured uint8 +--ro webhook-results +--ro webhook* +--ro name string +--ro uri ietf-inet-types:\ </pre>
uri	<pre> +--ro (result-type)? +---:(connection-error) +--ro connection-error? string +---:(http-status-code) +--ro http-status-code string +---:(redirect-information) +--ro redirect-information +--ro (local-or-reference) +---:(reference) </pre>


```

| | | | | +--ro referenced-definition? string
| | | | | +--:(onboarding-information) {wn-sztpd-0:onboarding-supported}?
| | | | | +--ro onboarding-information
| | | | | | +--ro (local-or-reference)
| | | | | | | +--:(reference)
| | | | | | +--ro referenced-definition? string
+--:(report-progress-event) {wn-sztpd-0:onboarding-supported}?
+--ro report-progress-event
| +--ro passed-input
| | +--ro progress-type string
| | +--ro message? string
| | +--ro ssh-host-keys? anydata
| | +--ro trust-anchor-certs? anydata
+--ro dynamic-callout
| +--ro (result-type)?
| | +--:(no-callout-configured)
| | | +--ro no-callout-configured? empty
| | +--:(callout-configured)
| | | +--ro name string
| | | +--ro rpc-supported identityref
| | | +--ro callout-type enumeration
| | +--ro (callback-or-webhook)?
| | | +--:(callback)
| | | | +--ro callback-details
| | | | | +--ro plugin string
| | | | | +--ro function string
| | | | +--ro callback-results
| | | | +--ro (exit-status)
| | | | | +--:(exception-thrown)
| | | | | | +--ro exception-thrown? string
| | | | | +--:(exited-normally)
| | | | | | +--ro exited-normally? string
| | | +--:(webhook) {wn-app:webhook-based-callouts}?
| | | | +--ro webhook-details
| | | | | +--ro num-webhooks-configured uint8
| | | | +--ro webhook-results
| | | | | +--ro webhook*
| | | | | | +--ro name string
| | | | | | +--ro uri ietf-inet-types:uri
| | | | | +--ro (result-type)?
| | | | | | +--:(connection-error)
| | | | | | | +--ro connection-error? string
| | | | | | +--:(http-status-code)
| | | | | | +--ro http-status-code string
+--ro lifecycle-statistics
+--ro nbi-access-stats
| | +--ro created ietf-yang-types:date-and-time
| | +--ro num-times-modified uint16
| | +--ro last-modified ietf-yang-types:date-and-time
+--ro sbi-access-stats
+--ro num-times-accessed uint16
+--ro first-accessed ietf-yang-types:date-and-time
+--ro last-accessed ietf-yang-types:date-and-time
+--rw device-type -> /device-types/device-type/name

```

4.3.2 Mode-X's native view

The following [tree diagram](#) illustrates the entire NBI, from the perspective of the Mode 'x' native interface (not the tenant interface).

```

===== NOTE: '\ ' line wrapping per RFC 8792 =====
module: wn-sztpd-x
+--rw device-types
|   +--rw device-type* [name]
|   |   +--rw name string
|   |   +--rw identity-certificates!
|   |   |   +--rw verification
|   |   |   |   +--rw local-truststore-reference {ietf-truststore:certificates}?
|   |   |   |   |   +--rw certificate-bag -> /truststore/certificate-bags/certificate-bag/name
|   |   |   |   |   +--rw certificate -> /truststore/certificate-bags/certificate-bag[name = current()]/\
|   |   |   |   |   ../certificate-bag/certificate/name
|   |   |   |   |   +--rw serial-number-extraction? identityref <wn-x509-c2n:serial-number>
|   |   |   |   +--rw ownership-authorization!
|   |   |   |   |   +--rw dynamic-callout
|   |   |   |   |   |   +--rw reference? -> /dynamic-callouts/dynamic-callout/name
|   |   |   |   +--rw voucher-aquisition!
|   |   |   |   |   +--rw dynamic-callout
|   |   |   |   |   |   +--rw reference? -> /dynamic-callouts/dynamic-callout/name
|   |   +--rw preferences
|   |   +--rw admin-accounts
|   |   |   +--rw new-account-verification
|   |   |   |   +--rw subject? string <SZTPD new account verification required.>
|   |   |   |   +--rw cc? string <$CREATOR-FULLNAME <CREATOR-EMAIL-ADDRESS>>
|   |   |   |   +--rw body? string <$ACCOUNT-FULLNAME
+--rw admin-accounts
|   +--rw admin-account* [email-address]
|   |   +--rw email-address string
|   |   +--rw fullname? string
|   |   +--rw password iana-crypt-hash:crypt-hash
|   |   +--ro password-last-modified ietf-yang-types:date-and-time
|   |   +--rw access enumeration
|   |   +--x resend-activation-email
+--ro audit-log
|   +--ro log-entry*
|   |   +--ro timestamp ietf-yang-types:date-and-time
|   |   +--ro source-ip ietf-inet-types:ip-address
|   |   +--ro source-proxies* string
|   |   +--ro host string
|   |   +--ro method enumeration
|   |   +--ro path string
|   |   +--ro outcome enumeration
|   |   +--ro comment? string
+--rw truststore
|   +--rw certificate-bags! {ietf-truststore:certificates}?
|   |   +--rw certificate-bag* [name]
|   |   |   +--rw name string
|   |   |   +--rw description? string
|   |   +--rw certificate* [name]

```

You are receiving this email because I created a new account for you on the SZTPD (\$SZTPD-SERVER-ADDRESS). You account's login is "\$ACCOUNT-EMAIL-ADDRESS".

Please activate your account via the following URL: \$VERIFICATION-URI (This URL expires in 48-hours).

Note that I CC-ed myself on this email and you may contact me to verify the veracity of this email.

Thanks,
\$CREATOR-FULLNAME

```

>
|   |   +--rw passwords
|   |   |   +--rw strength-testing!
|   |   |   |   +--rw min-length? uint16 <16>
+--rw notification-delivery!
|   +--rw dynamic-callout
|   |   +--rw reference -> ../../../../dynamic-callouts/dynamic-callout/name
+--rw system
|   +--rw hostname? ietf-inet-types:host
|   +--rw email-from-address? string <SZTPD Administrator <root@$SZTPD-SERVER-ADDRESS>>
+--rw features
|   +--rw onboarding-supported? boolean <true> {wn-sztpd-0:onboarding-supported}?
+--rw plugins
|   +--rw plugin* [name]
|   |   +--rw name string
|   |   +--rw functions
|   |   |   +--rw function* [name]
|   |   |   +--rw name string
+--rw admin-accounts
|   +--rw admin-account* [email-address]
|   |   +--rw email-address string
|   |   +--rw fullname? string
|   |   +--rw password iana-crypt-hash:crypt-hash
|   |   +--ro password-last-modified ietf-yang-types:date-and-time
|   |   +--rw access enumeration
|   |   +--x resend-activation-email
+--ro audit-log
|   +--ro log-entry*
|   |   +--ro timestamp ietf-yang-types:date-and-time
|   |   +--ro source-ip ietf-inet-types:ip-address
|   |   +--ro source-proxies* string
|   |   +--ro host string
|   |   +--ro method enumeration
|   |   +--ro path string
|   |   +--ro outcome enumeration
|   |   +--ro comment? string
+--rw truststore
|   +--rw certificate-bags! {ietf-truststore:certificates}?
|   |   +--rw certificate-bag* [name]
|   |   |   +--rw name string
|   |   |   +--rw description? string
|   |   +--rw certificate* [name]

```

```

    +-rw name string
    +-rw cert-data ietf-crypto-types:trust-anchor-cert-cms
    +-n certificate-expiration {ietf-crypto-types:certificate-expiration-notification}?
    +-ro expiration-date ietf-yang-types:date-and-time
    +-ro reference-statistics
    +-ro reference-count uint32
    +-ro last-referenced union
  +-ro reference-statistics
  +-ro reference-count uint32
  +-ro last-referenced union
+-rw keystore
  +-rw asymmetric-keys
  +-rw asymmetric-key* [name]
  +-rw name string
  +-rw public-key-format identityref
  +-rw public-key binary
  +-rw private-key-format? identityref
  +-rw (private-key-type)
  +--:(cleartext-private-key)
  | +-rw cleartext-private-key? binary
  +--:(hidden-private-key)
  | +-rw hidden-private-key? empty
  +--:(encrypted-private-key) {ietf-crypto-types:private-key-encryption}?
  +-rw encrypted-private-key
  +-rw encrypted-by
  +-rw encrypted-value binary
  +-rw certificates
  +-rw certificate* [name]
  +-rw name string
  +-rw cert-data ietf-crypto-types:end-entity-cert-cms
  +-n certificate-expiration {ietf-crypto-types:certificate-expiration-notification}?
  +-ro expiration-date ietf-yang-types:date-and-time
  +-ro reference-statistics
  +-ro reference-count uint32
  +-ro last-referenced union
  +-x generate-certificate-signing-request {ietf-crypto-types:certificate-signing-request-genera\
tion}?
  | +-rw input
  | | +-w csr-info ietf-crypto-types:csr-info
  | +-rw output
  | +-ro certificate-signing-request ietf-crypto-types:csr
  +-ro reference-statistics
  +-ro reference-count uint32
  +-ro last-referenced union
+-rw symmetric-keys
  +-rw symmetric-key* [name]
  +-rw name string
  +-rw key-format? identityref
  +-rw (key-type)
  +--:(cleartext-key)
  | +-rw cleartext-key? binary
  +--:(hidden-key)
  | +-rw hidden-key? empty
  +--:(encrypted-key) {ietf-crypto-types:symmetric-key-encryption}?
  +-rw encrypted-key
  +-rw encrypted-by
  +-rw encrypted-value binary
+-rw dynamic-callouts
  +-rw dynamic-callout* [name]
  +-rw name string
  +-rw rpc-supported identityref
  +-rw (callout-type)
  +--:(use-callback)
  | +-rw callback
  | +-rw plugin -> /preferences/system/plugins/plugin/name
  | +-rw function -> /preferences/system/plugins/plugin[name = current()/../plugin]/functi\
ons/function/name
  +--:(use-webhooks) {wn-app:webhook-based-callouts}?
  +-rw webhooks
  +-rw webhook* [name]
  +-rw name string
  +-rw encoding? enumeration <json>
  +-rw (transport)
  +--:(https) {ietf-restconf-client:https-initiate}?
  +-rw https
  +-rw tcp-client-parameters
  +-rw remote-address ietf-inet-types:host
  +-rw remote-port? ietf-inet-types:port-number <443>
  +-rw tls-client-parameters
  +-rw client-identity!
  +-rw (auth-type)
  +--:(certificate) {ietf-tls-client:x509-certificate-auth}?
  +-rw certificate
  +-rw (local-or-keystore)
  +--:(local-keystore)
  +-rw local-keystore-reference
  +-rw asymmetric-key -> ../../../../../../../../../../\

```

```

../keystore/asymmetric-keys/asymmetric-key/name
|                                     +-rw certificate          -> ../../../../../../../../../../\
../keystore/asymmetric-keys/asymmetric-key [name = current()../asymmetric-key]/certificates/certificate/name
|                                     +-rw server-authentication
|                                     +-rw ca-certs! {ietf-tls-client:x509-certificate-auth}?
|                                     +-rw (local-or-truststore)
|                                     +---:(local-truststore) {ietf-truststore:certificates}?
|                                     +-rw local-truststore-reference? -> ../../../../../../../../../../\
../truststore/certificate-bags/certificate-bag/name
|                                     +-rw ee-certs! {ietf-tls-client:x509-certificate-auth}?
|                                     +-rw (local-or-truststore)
|                                     +---:(local-truststore) {ietf-truststore:certificates}?
|                                     +-rw local-truststore-reference? -> ../../../../../../../../../../\
../truststore/certificate-bags/certificate-bag/name
|                                     +-rw http-client-parameters
|                                     +-rw client-identity!
|                                     +-rw (auth-type)
|                                     +---:(basic)
|                                     +-rw basic {ietf-http-client:basic-auth}?
|                                     +-rw user-id              string
|                                     +-rw (password-type)
|                                     +---:(cleartext-password)
|                                     | +-rw cleartext-password?  string
|                                     +---:(encrypted-password) {ietf-crypto-types:password-encrypt\
ion}?
|                                     +-rw encrypted-password
|                                     +-rw encrypted-by
|                                     +-rw encrypted-value       binary
|                                     +-rw restconf-client-parameters
|                                     +---:(http)
|                                     +-rw http
|                                     +-rw tcp-client-parameters
|                                     | +-rw remote-address       ietf-inet-types:host
|                                     | +-rw remote-port?        ietf-inet-types:port-number <80>
|                                     +-rw http-client-parameters
|                                     +-rw client-identity!
|                                     +-rw (auth-type)
|                                     +---:(basic)
|                                     +-rw basic {ietf-http-client:basic-auth}?
|                                     +-rw user-id              string
|                                     +-rw (password-type)
|                                     +---:(cleartext-password)
|                                     | +-rw cleartext-password?  string
|                                     +---:(encrypted-password) {ietf-crypto-types:password-encrypt\
ion}?
|                                     +-rw encrypted-password
|                                     +-rw encrypted-by
|                                     +-rw encrypted-value       binary
|                                     +-rw round-robin!
|                                     +-rw amount                uint16
|                                     +-rw units                 enumeration
+-rw transport
| +-rw listen! {ietf-restconf-server:http-listen or ietf-restconf-server:https-listen}?
| +-rw endpoint* [name]
| | +-rw name          string
| | +-rw (transport)
| | | +---:(http) {ietf-restconf-server:http-listen}?
| | | +-rw http
| | | | +-rw external-endpoint!
| | | | | +-rw address       ietf-inet-types:ip-address
| | | | | +-rw port?        ietf-inet-types:port-number <443>
| | | | +-rw tcp-server-parameters
| | | | | +-rw local-address   ietf-inet-types:ip-address
| | | | | +-rw local-port?    ietf-inet-types:port-number <80>
| | | | +-rw http-server-parameters
| | | | | +-rw server-name?    string
| | | | +-rw restconf-server-parameters
| | | | | +-rw client-identity-mappings
| | | | | | +-rw cert-to-name* [id]
| | | | | | | +-rw id          uint32
| | | | | | | +-rw fingerprint? ietf-x509-cert-to-name:tls-fingerprint
| | | | | | | +-rw map-type     identityref
| | | | | | | +-rw name         string
| | | | +---:(https) {ietf-restconf-server:https-listen}?
| | | | +-rw https
| | | | | +-rw tcp-server-parameters
| | | | | | +-rw local-address   ietf-inet-types:ip-address
| | | | | | +-rw local-port?    ietf-inet-types:port-number <443>
| | | | | +-rw tls-server-parameters
| | | | | | +-rw server-identity
| | | | | | | +-rw (auth-type)
| | | | | | | | +---:(certificate) {ietf-tls-server:x509-certificate-auth}?
| | | | | | | | +-rw certificate
| | | | | | | | | +-rw (local-or-keystore)
| | | | | | | | | | +---:(local-keystore)
| | | | | | | | | | +-rw reference
| | | | | | | | | | | +-rw asymmetric-key? -> /keystore/asymmetric-keys/asymmetric\

```

```

-key/name
-key[name = current()/.-/asymmetric-key]/certificates/certificate/name
    +-rw certificate? -> /keystore/asymmetric-keys/asymmetric\
    +-rw client-authentication! {ietf-tls-server:client-auth-config-supported}?
    +-rw ca-certs! {ietf-tls-server:x509-certificate-auth}?
    +-rw (local-or-truststore)
    +---:(local-truststore) {ietf-truststore:certificates,ietf-tls-server:client\
-auth-config-supported}?
    +-rw local-truststore-reference? -> /truststore/certificate-bags/cert\
ificate-bag/name
    +-rw ee-certs! {ietf-tls-server:x509-certificate-auth}?
    +-rw (local-or-truststore)
    +---:(local-truststore) {ietf-truststore:certificates,ietf-tls-server:client\
-auth-config-supported}?
    +-rw local-truststore-reference? -> /truststore/certificate-bags/cert\
ificate-bag/name
    +-rw http-server-parameters
    +-rw server-name? string
    +-rw restconf-server-parameters
    +-rw client-identity-mappings
    +-rw cert-to-name* [id]
        +-rw id uint32
        +-rw fingerprint? ietf-x509-cert-to-name:tls-fingerprint
        +-rw map-type identityref
        +-rw name string
    +-rw use-for enumeration
+-rw tenants
  +-rw tenant* [name]
    +-rw name string
    +-rw preferences
      +-rw admin-accounts
        +-rw new-account-verification
          +-rw subject? string <SZTPD new account verification required.>
          +-rw cc? string <$CREATOR-FULLNAME <CREATOR-EMAIL-ADDRESS>>
          +-rw body? string <$ACCOUNT-FULLNAME

```

You are receiving this email because I created a new account for you on the SZTPD (\$SZTPD-SERVER-ADDRESS). Your account's login is "\$ACCOUNT-EMAIL-ADDRESS".

Please activate your account via the following URL:
\$VERIFICATION-URI (This URL expires in 48-hours).

Note that I CC-ed myself on this email and you may contact me to verify the veracity of this email.

Thanks,
\$CREATOR-FULLNAME
>

```

    +-rw passwords
    +-rw strength-testing!
    +-rw min-length? uint16 <16>
    +-rw notification-delivery!
    +-rw dynamic-callout
    +-rw reference -> ../../../../dynamic-callouts/dynamic-callout/name
+-rw admin-accounts
  +-rw admin-account* [email-address]
    +-rw email-address string
    +-rw fullname? string
    +-rw password iana-crypt-hash:crypt-hash
    +-ro password-last-modified ietf-yang-types:date-and-time
    +-rw access enumeration
    +-x resend-activation-email
+-ro audit-log
  +-ro log-entry*
    +-ro timestamp ietf-yang-types:date-and-time
    +-ro source-ip ietf-inet-types:ip-address
    +-ro source-proxies* string
    +-ro host string
    +-ro method enumeration
    +-ro path string
    +-ro outcome enumeration
    +-ro comment? string
+-rw truststore
  +-rw certificate-bags! {ietf-truststore:certificates}?
    +-rw certificate-bag* [name]
      +-rw name string
      +-rw description? string
      +-rw certificate* [name]
        +-rw name string
        +-rw cert-data ietf-crypto-types:trust-anchor-cert-cms
        +-n certificate-expiration {ietf-crypto-types:certificate-expiration-notification}?
          +-ro expiration-date ietf-yang-types:date-and-time
        +-ro reference-statistics
          +-ro reference-count uint32
          +-ro last-referenced union
        +-ro reference-statistics

```

```

    +-ro reference-count    uint32
    +-ro last-referenced   union
+-rw keystore
  +-rw asymmetric-keys
    +-rw asymmetric-key* [name]
      +-rw name                string
      +-rw public-key-format   identityref
      +-rw public-key          binary
      +-rw private-key-format? identityref
      +-rw (private-key-type)
        +---:(cleartext-private-key)
          | +-rw cleartext-private-key?  binary
        +---:(hidden-private-key)
          | +-rw hidden-private-key?    empty
        +---:(encrypted-private-key) {ietf-crypto-types:private-key-encryption}?
          +-rw encrypted-private-key
            +-rw encrypted-by
            +-rw encrypted-value    binary
      +-rw certificates
        +-rw certificate* [name]
          +-rw name                string
          +-rw cert-data           ietf-crypto-types:end-entity-cert-cms
          +-n certificate-expiration {ietf-crypto-types:certificate-expiration-notification}?
            +-ro expiration-date    ietf-yang-types:date-and-time
          +-ro reference-statistics
            +-ro reference-count    uint32
            +-ro last-referenced   union
        +-x generate-certificate-signing-request {ietf-crypto-types:certificate-signing-request-\
generation}?
          | +-rw input
          | | +-w csr-info    ietf-crypto-types:csr-info
          | +-rw output
          | +-ro certificate-signing-request    ietf-crypto-types:csr
          +-ro reference-statistics
            +-ro reference-count    uint32
            +-ro last-referenced   union
      +-rw symmetric-keys
        +-rw symmetric-key* [name]
          +-rw name                string
          +-rw key-format?         identityref
          +-rw (key-type)
            +---:(cleartext-key)
              | +-rw cleartext-key?    binary
            +---:(hidden-key)
              | +-rw hidden-key?      empty
            +---:(encrypted-key) {ietf-crypto-types:symmetric-key-encryption}?
              +-rw encrypted-key
                +-rw encrypted-by
                +-rw encrypted-value    binary
      +-rw dynamic-callouts
        +-rw dynamic-callout* [name]
          +-rw name                string
          +-rw rpc-supported       identityref
          +-rw (callout-type)
            +---:(use-callback)
              +-rw callback
                +-rw plugin        -> /preferences/system/plugins/plugin/name
                +-rw function      -> /preferences/system/plugins/plugin[name = current()/../plugin]/\
functions/function/name
            +---:(use-webhooks) {wn-app:webhook-based-callouts}?
              +-rw webhooks
                +-rw webhook* [name]
                  +-rw name                string
                  +-rw encoding?          enumeration <json>
                  +-rw (transport)
                    +---:(https) {ietf-restconf-client:https-initiate}?
                      +-rw https
                        +-rw tcp-client-parameters
                          +-rw remote-address    ietf-inet-types:host
                          +-rw remote-port?     ietf-inet-types:port-number <443>
                        +-rw tls-client-parameters
                          +-rw client-identity!
                            +-rw (auth-type)
                              +---:(certificate) {ietf-tls-client:x509-certificate-auth}?
                                +-rw certificate
                                  +-rw (local-or-keystore)
                                    +---:(local-keystore)
                                      +-rw local-keystore-reference
                                      +-rw asymmetric-key    -> ../../../../../../../\
../../../../../keystore/asymmetric-keys/asymmetric-key/name
                                +-rw certificate          -> ../../../../../../../\
../../../../../keystore/asymmetric-keys/asymmetric-key[name = current()/../asymmetric-key]/certificates/certifica\
te/name
                              +---:(local-truststore) {ietf-truststore:certificates}?
                                +-rw server-authentication
                                  +-rw ca-certs! {ietf-tls-client:x509-certificate-auth}?
                                  +-rw (local-or-truststore)
                                  +---:(local-truststore) {ietf-truststore:certificates}?

```

```

+--rw local-truststore-reference?  -> ../../../../../../
../../../../truststore/certificate-bags/certificate-bag/name
+--rw ee-certs! {ietf-tls-client:x509-certificate-auth}?
+--rw (local-or-truststore)
+--:(local-truststore) {ietf-truststore:certificates}?
+--rw local-truststore-reference?  -> ../../../../../../
../../../../truststore/certificate-bags/certificate-bag/name
+--rw http-client-parameters
+--rw client-identity!
+--rw (auth-type)
+--:(basic)
+--rw basic {ietf-http-client:basic-auth}?
+--rw user-id          string
+--rw (password-type)
+--:(cleartext-password)
| +--rw cleartext-password?  string
+--:(encrypted-password) {ietf-crypto-types:password-e\
ncryption}?
+--rw encrypted-password
+--rw encrypted-by
+--rw encrypted-value      binary
+--rw restconf-client-parameters
+--:(http)
+--rw http
+--rw tcp-client-parameters
| +--rw remote-address      ietf-inet-types:host
| +--rw remote-port?       ietf-inet-types:port-number <80>
+--rw http-client-parameters
+--rw client-identity!
+--rw (auth-type)
+--:(basic)
+--rw basic {ietf-http-client:basic-auth}?
+--rw user-id          string
+--rw (password-type)
+--:(cleartext-password)
| +--rw cleartext-password?  string
+--:(encrypted-password) {ietf-crypto-types:password-e\
ncryption}?
+--rw encrypted-password
+--rw encrypted-by
+--rw encrypted-value      binary
+--rw round-robin!
+--rw amount      uint16
+--rw units       enumeration
+--rw bootstrap-servers
+--rw bootstrap-server* [name]
+--rw name              string
+--rw address           ietf-inet-types:host
+--rw port?             ietf-inet-types:port-number <443>
+--rw trust-anchor?    ietf-crypto-types:trust-anchor-cert-cms
+--ro reference-statistics
+--ro reference-count  uint32
+--ro last-referenced  union
+--rw boot-images {wn-sztpd-0:onboarding-supported}?
+--rw boot-image* [name]
+--rw name              string
+--rw os-name?          string
+--rw os-version?      string
+--rw download-uri*    ietf-inet-types:uri
+--rw image-verification* [hash-algorithm]
| +--rw hash-algorithm  identityref
| +--rw hash-value     ietf-yang-types:hex-string
+--ro reference-statistics
+--ro reference-count  uint32
+--ro last-referenced  union
+--rw scripts {wn-sztpd-0:onboarding-supported}?
+--rw pre-configuration-script* [name]
| +--rw name              string
| +--rw script?           ietf-sztp-conveyed-info:script
| +--ro reference-statistics
| +--ro reference-count  uint32
| +--ro last-referenced  union
+--rw post-configuration-script* [name]
+--rw name              string
+--rw script?           ietf-sztp-conveyed-info:script
+--ro reference-statistics
+--ro reference-count  uint32
+--ro last-referenced  union
+--rw configurations {wn-sztpd-0:onboarding-supported}?
+--rw configuration* [name]
+--rw name              string
+--rw configuration-handling?  enumeration
+--rw config?           binary
+--ro reference-statistics
+--ro reference-count  uint32
+--ro last-referenced  union
+--rw conveyed-information-responses

```

```

+rw redirect-information-response* [name]
  +rw name string
  +rw redirect-information
  | +rw bootstrap-server* -> ../../../../bootstrap-servers/bootstrap-server/name
  +ro reference-statistics
  +ro reference-count uint32
  +ro last-referenced union
+rw onboarding-information-response* [name] {wn-sztpd-0:onboarding-supported}?
  +rw name string
  +rw onboarding-information
  | +rw boot-image -> ../../../../boot-images/boot-image/name
  | +rw pre-configuration-script? -> ../../../../scripts/pre-configuration-script/name
  | +rw configuration? -> ../../../../configurations/configuration/name
  | +rw post-configuration-script? -> ../../../../scripts/post-configuration-script/name
  +ro reference-statistics
  +ro reference-count uint32
  +ro last-referenced union
+rw devices
  +rw device* [serial-number]
  +rw serial-number string
  +rw activation-code? iana-crypt-hash:crypt-hash
  +rw response-manager
  | +rw matched-response* [name]
  | | +rw name string
  | | +rw match-criteria!
  | | | +rw match* [key]
  | | | | +rw key string
  | | | | +rw not? empty
  | | | | +rw (test-type)
  | | | | | +--:(present)
  | | | | | | +rw present? empty
  | | | | | +--:(value)
  | | | | | | +rw value? string
  | | | | | +--:(regex)
  | | | | | +rw regex? string
  | | +rw response
  | | | +rw reporting-level? enumeration <minimal> {wn-sztpd-0:onboarding-supported}?
  | | | +rw (response-handler)
  | | | | +--:(none)
  | | | | | +rw none? empty
  | | | | +--:(managed-response)
  | | | | +rw conveyed-information
  | | | | | +rw (conveyed-information-handler)
  | | | | | | +--:(use-dynamic-callout)
  | | | | | | | +rw dynamic-callout
  | | | | | | | +rw reference? -> ../../../../dynamic-callouts/dyn\
amic-callout/name
  | | | | | | +--:(redirect-information)
  | | | | | | | +rw redirect-information
  | | | | | | | | +rw (local-or-reference)
  | | | | | | | | +--:(reference)
  | | | | | | | | +rw reference? -> ../../../../conveyed-infor\
mation-responses/redirect-information-response/name
  | | | | | | +--:(onboarding-information) {wn-sztpd-0:onboarding-supported}?
  | | | | | | | +rw onboarding-information
  | | | | | | | | +rw (local-or-reference)
  | | | | | | | | +--:(reference)
  | | | | | | | | +rw reference? -> ../../../../conveyed-infor\
mation-responses/onboarding-information-response/name
  | | +n progress-report-received
  | | +ro progress-report? anydata
  +ro bootstrapping-log
  +ro log-entry*
  +ro timestamp ietf-yang-types:date-and-time
  +ro source-ip ietf-inet-types:ip-address
  +ro method string
  +ro path string
  +ro return-code uint16
  +ro error-returned? anydata
  +ro event-details
  +ro (event-type)
  +--:(get-bootstrapping-data-event)
  | +ro get-bootstrapping-data-event
  | | +ro passed-input
  | | | +ro (input-type)
  | | | | +--:(no-input-passed)
  | | | | | +ro no-input-passed? empty
  | | | | +--:(input)
  | | | | +ro input*
  | | | | | +ro key string
  | | | | | +ro value anydata
  | | +ro selected-response? union
  | +ro response-details
  | | +ro (response-type)
  | | | +--:(managed-response)
  | | | +ro managed-response
  | | | +ro conveyed-information

```


			<pre> +ro (conveyed-information-handler) +--:(dynamic-callout) +ro dynamic-callout +ro (result-type)? +--:(no-callout-configured) +ro no-callout-configured? empty +--:(callout-configured) +ro name string +ro rpc-supported identityref +ro callout-type enumeration +ro (callback-or-webhook)? +--:(callback) +ro callback-details +ro plugin string +ro function string +ro callback-results +ro (exit-status) +--:(exception-thrown) +ro exception-thrown? </pre>
string			<pre> +--:(exited-normally) +ro exited-normally? </pre>
string			<pre> +--:(webhook) {wn-app:webhook-based-callou\ </pre>
uint8			<pre> +ro webhook-details +ro num-webhooks-configured </pre>
types:uri			<pre> +ro webhook-results +ro webhook* +ro name string +ro uri ietf-inet-\ </pre>
s\tring			<pre> +ro (result-type)? +--:(connection-error) +ro connection-error? </pre>
s\tring			<pre> +--:(http-status-code) +ro http-status-code </pre>
rted}?}			<pre> +--:(redirect-information) +ro redirect-information +ro (local-or-reference) +--:(reference) +ro referenced-definition? string +--:(onboarding-information) {wn-sztpd-0:onboarding-suppo\ </pre>
			<pre> +ro onboarding-information +ro (local-or-reference) +--:(reference) +ro referenced-definition? string +--:(report-progress-event) {wn-sztpd-0:onboarding-supported}? +ro report-progress-event +ro passed-input +ro progress-type string +ro message? string +ro ssh-host-keys? anydata +ro trust-anchor-certs? anydata +ro dynamic-callout +ro (result-type)? +--:(no-callout-configured) +ro no-callout-configured? empty +--:(callout-configured) +ro name string +ro rpc-supported identityref +ro callout-type enumeration +ro (callback-or-webhook)? +--:(callback) +ro callback-details +ro plugin string +ro function string +ro callback-results +ro (exit-status) +--:(exception-thrown) +ro exception-thrown? string +--:(exited-normally) +ro exited-normally? string +--:(webhook) {wn-app:webhook-based-callouts}? +ro webhook-details +ro num-webhooks-configured uint8 +ro webhook-results +ro webhook* +ro name string </pre>

	+-ro uri	ietf-inet-types:uri
	+-ro (result-type)?	
	+-:(connection-error)	
	+-ro connection-error?	string
	+-:(http-status-code)	
	+-ro http-status-code	string
+-ro lifecycle-statistics		
+-ro nbi-access-stats		
+-ro created	ietf-yang-types:date-and-time	
+-ro num-times-modified	uint16	
+-ro last-modified	ietf-yang-types:date-and-time	
+-ro sbi-access-stats		
+-ro num-times-accessed	uint16	
+-ro first-accessed	ietf-yang-types:date-and-time	
+-ro last-accessed	ietf-yang-types:date-and-time	
+-rw device-type	->	/device-types/device-type/name

5 Eastbound Interface

SZTPD uses an “eastbound” interface to interact with peering systems.

This information is mostly interesting to those wanting to standup a peering-system for SZTPD to interact with.

5.1 Inbound Facing

5.1.1 TLS Terminator

5.1.1.1 The “X-Client-Cert” HTTP Header Extension Field

In order to enable external TLS terminators to convey the client certificate passed during the TLS handshake, SZTPD looks for the “X-Client-Cert” header field. Also discussed in Section 2.3.1.2 (HTTP vs HTTP) of SZPTD Installation Guide.

5.2 Outbound Facing

5.2.1 RDBMS

SZTPD may be configured to reach out to an external RDBMS server for its database. The protocol used for this connection is database-specific, and may be shrouded by a TLS session (recommended).

5.2.2 Dynamic Callouts

Dynamic callouts may either be implemented as a plugin loaded into SZTPD, or as a webhook (an HTTP “POST” request) to an external peering system.

5.2.2.1 The “relay-notification” RPC

A dynamic callout implementing the “relay-notification” RPC is used to send notifications, as defined by the various YANG modules implemented by SZTPD. Notifications are those defined by the YANG “notification” statement. All of the notifications are presented in the [Notifications](#) section.

The RPC that SZTPD uses to send notifications is called “relay-notification”. This RPC is defined as follows:

```
rpcs:
  +---x relay-notification {relay-notification}?
    +--- input
      +---w notification? anydata
```

Following is what an HTTP request sent from SZTPD might look like:

REQUEST

```
POST /restconf/operations/wn-app-rpcs:relay-notification HTTP/1.1
HOST: example.com
Content-Type: application/yang.data+json

{
  "wn-app-rpcs:input" : {
    "notification": {
      "eventTime": "2019-03|22T12:35:00Z",
      "wn-sztpd-1:unused-device-record-lingering": {
        "device-record": "fault",
        "created": "20190122T10:32:12Z",
        "num-times-modified": 3,
        "last-modified": "20190320T09:51:44Z"
      }
    }
  }
}
```

Following is an example HTTP response sent from the remote endpoint:

RESPONSE

```
HTTP/1.1 204 OK
Date: Sat, 31 Oct 2015 17:02:40 GMT
Server: example-server
Content-Type: application/yang.data+json
```

5.2.2.2 The “verify-device-ownership” RPC

A dynamic callout implementing the “verify-device-ownership” RPC is used to verify that devices (serial numbers) to the configured authority (i.e., a specific tenant). The RPC’s tree diagram follows:

```
rpcs:
  +--x verify-device-ownership {verify-device-ownership}?
    +-- input
      | +--w tenant          string
      | +--w serial-number* string
    +-- output
      +--ro verification-results
        +--ro verification-result* [serial-number]
          +--ro serial-number  string
          +--ro result         enumeration
```

Following is what an HTTP request sent from SZTPD might look like:

REQUEST

```
POST /restconf/operations/wn-sztpd-rpcs:verify-device-ownership HTTP/1.1
HOST: example.com
Content-Type: application/yang.data+json

{
  "wn-sztpd-rpcs:input" : {
    "tenant": "not-applicable",
    "serial-number": [ "sn-111", "sn-222", "sn-333" ]
  }
}
```

Following is an example HTTP response sent from the remote endpoint:

RESPONSE

```
HTTP/1.1 200 OK
Date: Sat, 31 Oct 2015 17:02:40 GMT
Server: example-server
Content-Type: application/yang.data+json

{
  "ietf-sztpd-rpcs:output" : {
    "verification-results": {
      "verification-result": [
        {
          "serial-number": "sn-111",
          "result": "success"
        },
        {
          "serial-number": "sn-222",
          "result": "failure"
        },
        {
          "serial-number": "sn-333",
          "result": "success"
        }
      ]
    }
  }
}
```

5.2.2.3 The “get-conveyed-information” RPC

A dynamic callout implementing the “get-conveyed-information” RPC, used to fetch a just-in-time generated response to a bootstrapping device. The RPC’s tree diagram follows:

```
rpcs:
  +--x get-conveyed-information {get-conveyed-information}?
    +-- input
      | +--w serial-number          string
      | +--w source-ip-address     ietf-inet-types:ip-address
      | +--w from-device?          anydata
      | +--w identity-certificate? ietf-crypto-types:cms
      +-- output
        +--ro conveyed-information anydata
```

Following is what an HTTP request sent from SZTPD might look like:

REQUEST

```
POST /restconf/operations/wn-sztpd-rpcs:get-conveyed-information HTTP/1.1
Host: example.com
Content-Type: application/yang.data+json

{
  "wn-sztpd-rpcs:input" : {
    "serial-number": "my-serial-number",
    "source-ip-address": "10.20.30.40",
    "from-device": {
      "ietf-sztp-bootstrap-server:input" : {
        "hw-model": "model-x",
        "os-name": "vendor-os",
        "os-version": "17.3R2.1",
        "nonce": "extralongbase64encodedvalue=",
        "ietf-sztp-csr:csr": {
          "p10": "base64encodedvalue=="
        }
      }
    },
    "identity-certificate": "base64encodedvalue=="
  }
}
```

Following is an example HTTP response sent from the remote endpoint:

RESPONSE

```
HTTP/1.1 200 OK
Date: Sat, 31 Oct 2015 17:02:40 GMT
Server: example-server
Content-Type: application/yang.data+json

{
  "ietf-sztpd-rpcs:output" : {
    "conveyed-information": "base64encodedvalue=="
  }
}
```

5.2.3 Notifications

The following “notification” statements are defined in the “wn-app-rpcs” YANG module:

```
notifications
+-- unreferenced-node-lingering
+-- unreferenced-node-purged
+-- admin-account-activation-expired
+-- admin-account-password-aging
+-- admin-account-disabled
+-- admin-account-purged
+-- audit-log-entry-purged
```

The following “notification” statements are defined in the “wn-sztpd-0” YANG module, which is used as a basis for all the SZTPD YANG modules (mode ‘1’ and mode ‘x’ models):

```
notifications
+-- voucher-expiration
+-- unused-device-record-lingering
+-- unused-device-record-purged
+-- used-device-record-lingering
+-- used-device-record-purged
+-- progress-report-received
```

The following “notification” statement is defined in the “ietf-crypto-types” YANG module:

```
notifications
+-- certificate-expiration
```

The following “notification” statements are defined in the “ietf-yang-library” YANG module:

```
notifications
+-- yang-library-update
+-- yang-library-change
```

Similarly, per RFC 8572, SZTPD uses notifications to relay “progress-reports” to integrating applications.

5.2.3.1 The “unreferenced-node-lingering” Notification

```
+--n unreferenced-node-lingering {wn-app:storage-reclamation-implemented}?
+--ro last-referenced union
```

5.2.3.2 The “unreferenced-node-purged” Notification

```
+--n unreferenced-node-purged {wn-app:storage-reclamation-implemented}?
+--ro last-referenced union
```

5.2.3.3 The “admin-account-activation-expired” Notification

```
+---n admin-account-activation-expired {wn-app:account-activation-implementation-implementation}?
```

5.2.3.4 The “admin-account-password-aging” Notification

```
+---n admin-account-password-aging {wn-app:password-aging-implementation-implementation}?  
  +---ro expiration-date ietf-yang-types:date-and-time
```

5.2.3.5 The “admin-account-disabled” Notification

```
+---n admin-account-disabled {wn-app:storage-reclamation-implementation-implementation}?
```

5.2.3.6 The “admin-account-purged” Notification

```
+---n admin-account-purged {wn-app:storage-reclamation-implementation-implementation}?
```

5.2.3.7 The “voucher-expiration” Notification

```
+---n voucher-expiration  
  +---ro expiration-date ietf-yang-types:date-and-time
```

5.2.3.8 The “unused-device-record-lingering” Notification

```
+---n unused-device-record-lingering {wn-app:storage-reclamation-implementation-implementation}?  
  +---ro nbi-access-stats  
    +---ro created ietf-yang-types:date-and-time  
    +---ro num-times-modified uint16  
    +---ro last-modified ietf-yang-types:date-and-time
```

5.2.3.9 The “unused-device-record-purged” Notification

```
+---n unused-device-record-purged {wn-app:storage-reclamation-implementation-implementation}?  
  +---ro nbi-access-stats  
    +---ro created ietf-yang-types:date-and-time  
    +---ro num-times-modified uint16  
    +---ro last-modified ietf-yang-types:date-and-time
```

5.2.3.10 The “used-device-record-lingering” Notification

```
+---n used-device-record-lingering {wn-app:storage-reclamation-implementation-implementation}?  
  +---ro nbi-access-stats  
    | +---ro created ietf-yang-types:date-and-time  
    | +---ro num-times-modified uint16  
    | +---ro last-modified ietf-yang-types:date-and-time  
  +---ro sbi-access-stats  
    +---ro num-times-accessed uint16  
    +---ro first-accessed ietf-yang-types:date-and-time  
    +---ro last-accessed ietf-yang-types:date-and-time
```

5.2.3.11 The “used-device-record-purged” Notification

```
+---n used-device-record-purged {wn-app:storage-reclamation-implemented}?
  +---ro nbi-access-stats
  |   +---ro created          ietf-yang-types:date-and-time
  |   +---ro num-times-modified  uint16
  |   +---ro last-modified     ietf-yang-types:date-and-time
  +---ro sbi-access-stats
  |   +---ro num-times-accessed  uint16
  |   +---ro first-accessed     ietf-yang-types:date-and-time
  |   +---ro last-accessed      ietf-yang-types:date-and-time
```

5.2.3.12 The “certificate-expiration” Notification

```
+---n certificate-expiration {ietf-crypto-types:certificate-expiration-notification}?
  +---ro expiration-date      ietf-yang-types:date-and-time
```

5.2.3.13 The “yang-library-change” Notification

```
x---n yang-library-change
  x---ro module-set-id      -> /ietf-yang-library:modules-state/ietf-yang-library:module-set-id
```

5.2.3.14 The “yang-library-update” Notification

```
+---n yang-library-update
  +---ro content-id        -> /ietf-yang-library:yang-library/ietf-yang-library:content-id
```

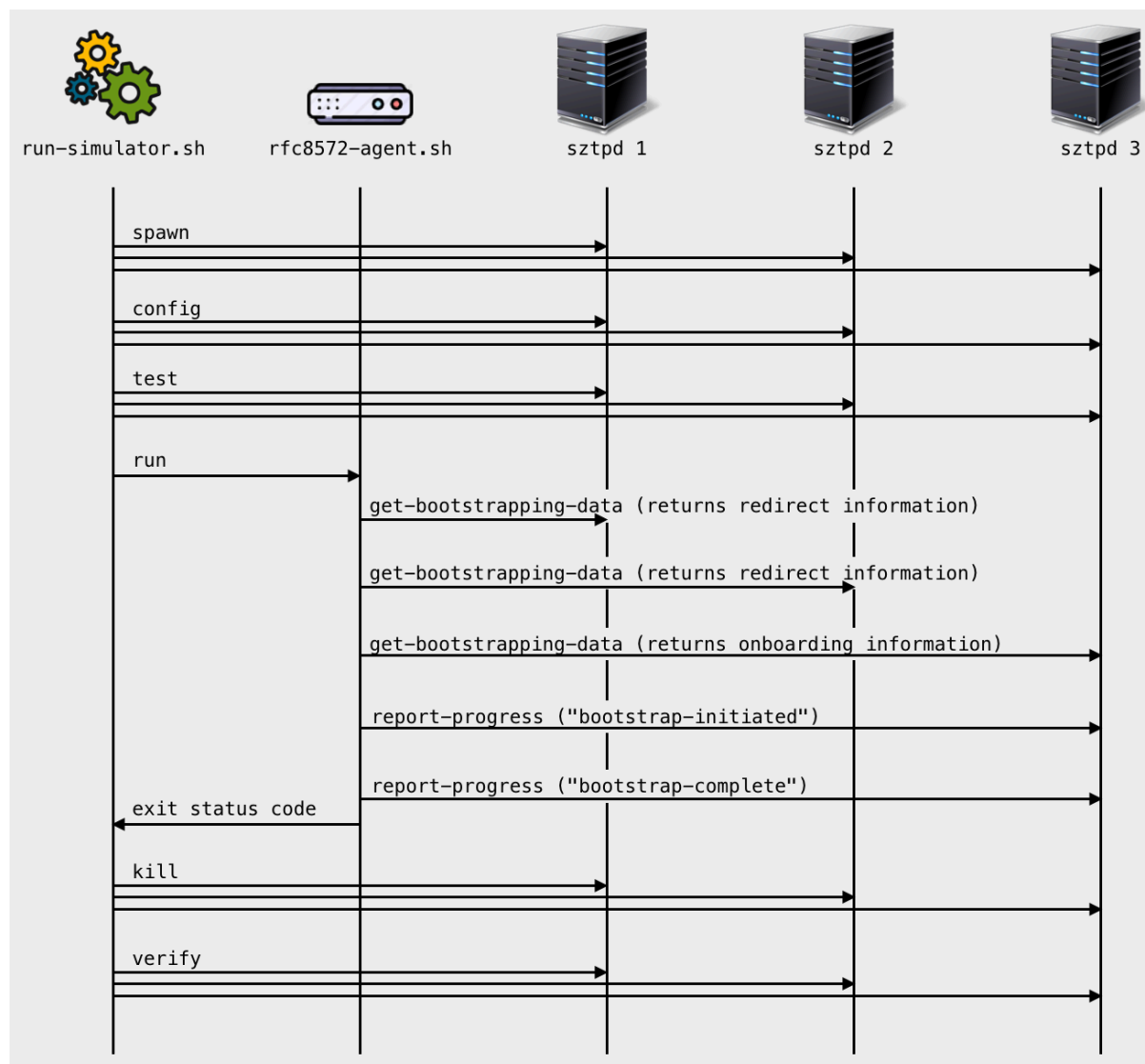
5.2.3.15 The “yang-library-change” Notification

```
x---n yang-library-change
  x---ro module-set-id      -> /ietf-yang-library:modules-state/ietf-yang-library:module-set-id
```


6 Simulator

A simulator has been developed, primarily to aid in the development of the RFC 8572 compliant bootstrapping agent, though inspection of its source code is helpful towards understand how to use SZTPD's [Northbound Interface](#) (e.g., its use of the curl command line utility).

6.1 Overview



The simulator (i.e., the script `run-sztpd-test.sh`) performs the following steps:

1. Start and initialize three SZTPD servers for the following roles:

Instance	Purpose
1	A trusted bootstrap server to redirect the device to instance #2.
2	An initially untrusted bootstrap server to redirect the device to instance #3.
3	An initially untrusted bootstrap server to give the device its onboarding data.

Each SZTPD server is an instance of the `sztpd` process.

2. Start an instance of a demo RFC 8572 agent (the `rfc8572-simulator.sh` script).
Parameters are passed into the agent providing necessary values (e.g., the “serial-number” value).
3. Wait for the agent to complete and then shutdown the three SZTPD instances.

No trace on the filesystem is retained as:

1. the `run-sztpd-test.sh` and `rfc8572-simulator.sh` scripts cleans up after themselves.
2. the `sztpd` processes are run using the “in-memory” database².

6.2 Dependencies

Successful execution of the requires a few packages to be installed.

- A UNIX-based system. (untested on Windows)
- Bash (the Bourne-Again SHell, any recent version)
- OpenSSL (both libraries and the command-line utility)
- Python (version 3.7 or greater)
- The following Python modules (all installed as dependencies to SZTPD)
- Curl (the `curl` utility, many times installed by OS)
- SZTPD

6.3 Downloading

When posted, the simulator can be downloaded here: <https://watsen.net/support/>.

6.4 Unarchiving

Unarchive the TGZ in a local directory, for example:

```
$ tar -xzf ./sztpd-simulator-0.0.2.tgz

// This command creates a directory called "sztpd-simulator".
```

²For information about the in-memory database, please See the “In-memory Database” section in the Installation Guide.

Inside the “sztpd-simulator” directory are a number of files and directories:

Resource	Purpose
run-simulator.sh	The script that starts 3 SZTPD servers and an RFC 8572 agent.
rfc8572-agent.sh	A simple curl based agent that simulates an RFC 8572 agent.
templates/	A directory containing config used to init the SZTP instances.
pki/	A directory containing code to initialize the PKI for the demo.
xrd/	A directory containing files to validates the “host-meta” XRD.

6.5 Customizing Variables

There is little need to customize the simulator scripts before running.

The most likely customization is to adjust the “PYTHON” and “PIP” variables, located at the top of the “run-simulator.sh” and “rfc8572-agent.sh” files. For instance, if Python is installed as “python3.8”, then the values should be set as follows:

```
PYTHON=python3.8
PIP=pip3.8
```

Another possible customization is to adjust the ports that the various sztpd instances will open, for instance, to deconflict from a port already in use. To adjust the ports used, edit the top of the “run-sztpd-test.sh” and modify the various “PORT” values listed at the top of that file.

6.6 Initializing the PKI

The simulator creates a distinct PKI for each endpoint. As each SZTPD instance (in this demo) has two endpoints (an NBI and and SBI) and there are three SZTPD instances, a total of six certificate chains are created. Each certificate chain contains four certificates (a root certificate, two intermediate certificates, and an end-entity certificate), thus a total of Twenty-four certificates are created in total.

Whilst the certificates could be dynamically generated for each execution of the simulator, for multiple executions, having the PKI created already saves time.

Assuming the current directory is the “sztpd-simulator” directory, the following commands:

```
$ cd pki
$ make pki
$ cd ..
```

If ever needed, make clean can be used to return the directory hierarchy to its original form.

6.7 Running

Assuming the current directory is the “sztpd-simulator” directory, the following command will run the simulator:

```
$ ./run-sztpd-test.sh
```

Each time the simulator runs, it first tests if the PKI has been initialized. If the PKI has not been initialized, the simulator exits with the message:

```
PKI needs to be initialized first.
- e.g., `cd pki; make pki; cd ..;`
```

Otherwise the simulator runs without any other prompts, producing output such as:

```
Creating instances ...
^-- Creating SZTPD instance 1...okay. (SZTPD instance 1 running with PID 22089)
^-- Creating SZTPD instance 2...okay. (SZTPD instance 2 running with PID 220155)
^-- Creating SZTPD instance 3...okay. (SZTPD instance 3 running with PID 22091)

Giving servers a couple seconds to startup...

Configuring instances...
^-- Configuring SZTPD instance 1...
^-- Configuring SZTPD instance 2...
^-- Configuring SZTPD instance 3...

Giving servers a couple seconds to open their ports...

Testing instances...
^-- Testing SZTPD instance 1
^-- Testing SZTPD instance 2
^-- Testing SZTPD instance 3

Running simulator...
^-- Getting bootstrapping data...
^-- Processing bootstrapping data...
^-- Processing redirect information...
^-- Getting bootstrapping data from next server...
^-- Processing bootstrapping data...
^-- Processing redirect information...
^-- Getting bootstrapping data from next server...
^-- Processing bootstrapping data...
^-- Processing onboarding information...
^-- Bootstrap complete.

Killing `sztpd` instances...
^-- Sending SIGTERM to instance 1 (PID 22089)
^-- Sending SIGTERM to instance 2 (PID 22090)
^-- Sending SIGTERM to instance 3 (PID 22091)

Giving servers a couple seconds to shutdown...

Verifying instances are killed...
^-- Verifying SZTPD instance 1 killed...okay.
^-- Verifying SZTPD instance 2 killed...okay.
^-- Verifying SZTPD instance 3 killed...okay.

All done!
```

6.8 Cleaning Up

The simulator makes effort to clean up after itself. For instance, all files are created in a temporary directory. However, currently, if the simulator experiences an unexpected error, it may exit without first shutting down the sztpd instances.

If the sztpd instances are not removed, they can be removed manually. For instance:

```
$ ps | grep sztpd
23816 ttys024    0:02.01 python sztpd sqlite:///memory:
23817 ttys024    0:02.01 python sztpd sqlite:///memory:
23818 ttys024    0:02.01 python sztpd sqlite:///memory:

$ kill 23816
$ kill 23817
$ kill 23818
```