SZTPD Administrator's Guide

Watsen Networks

February 19, 2023

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

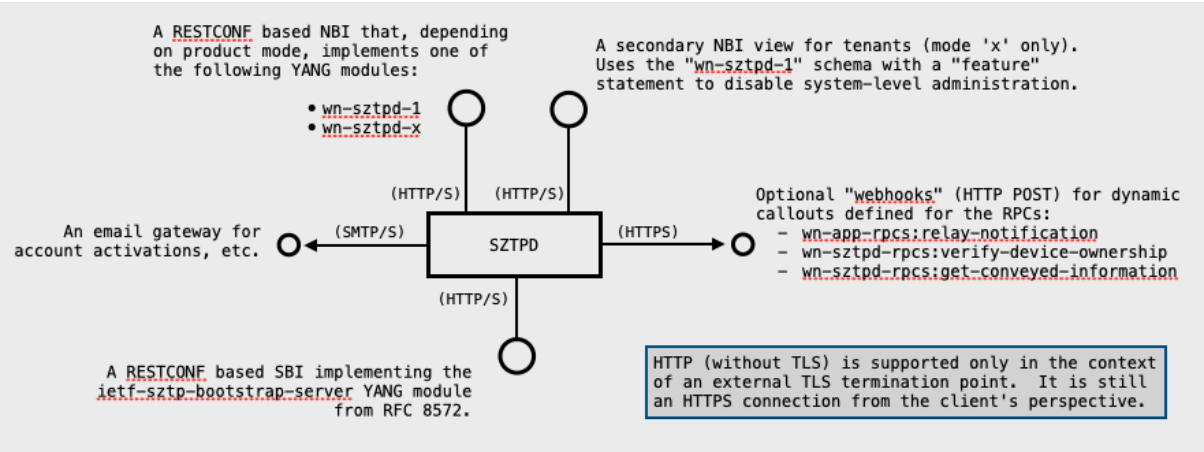
SZTPD is an implementation of a "bootstrap server", as defined in the Terminology section of RFC 8572, also known as an "SZTP server", as defined in the Terminology section of draft-kwatsen-netconf-sztp-csr".

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 and the viewing of operational state.

The NBI is a RESTCONF RFC 8040 based API that implements one of the two YANG modules based on the "product mode" selected during installation:

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

where '1' and 'x' refer to the SZTPD product mode ('1' for single-tenant, 'x' for multi-tenant). The API presented by the two product modes is nearly identical, as the 'wn-sztpd-x' YANG module uses definitions defined in the 'wn-sztpd-1' YANG module.

The maximum message size that a client can send is set 32MB.

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's 'get-bootstrapping-data' RPC-reply currently only returns plain (not signed nor encrypted) CMS structures. Returning signed and/or encrypted CMS structures is planned for a future release of SZTPD. That said, it is strongly suggested that application-level CMS-processing code switches on the top-most content-type's OID as follows:

- if id_ct_sztpConveyedInfoJSON or id_ct_sztpConveyedInfoXML
  - extract the JSON or XMl document from the CMS structure.
  - this is supported in SZTPD 1.0.
- if id-envelopedData (then first decrypt with private key)
  - first decrypt the data using the device's private key, and then examine the inner content-type's OID for the next step.
  - this is NOT supported in SZTPD 1.0.
- if id-signedData
  - first verify the signature using owner certificate, and then examine the inner content-type's OID for the next step.
  - this is NOT supported in SZTPD 1.0.

SZTPD's SBI is unaffected by SZTPD product modes.

See the Southbound Interface section for more information about the SBI.

2.3 Eastbound API

The Eastbound Interface API is for product integration into backend business systems, both internally (e.g., RDBMS, etc.) and externally (e.g., monitoring systems, orchestrtation systems, etc.)

The Eastbound API is also unaffected by the product mode ('1' or 'x') SZTPD is using.

See the Eastbound Interface section for more information about the eastbound API.

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_INIT_MODE=1; sztpd sqlite:///:memory:
```


3.1 Fetching Host-meta

RFC 8040 defines a discovery protocol for determining a RESTCONF server's root resource location. Note that SZTPD uses /restconf as the default root resource location. No "Content-Type" is required for this resource.

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 Determining the Encodings Supported

The NBI only supports JSON, while the SBI supports both XML and JSON.

This can be detemined by sending an HTTP "GET" operation to the RESTCONF Root location, discovered in the previous step, with no "Accept" type specified:

Request:

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

Response:

```
HTTP/1.1 400 Bad Request
Accept: application/yang-data+json
Accept-Charset: utf-8
Content-Type: text/plain; charset=utf-8
Server: <redacted>
Content-Length: 191
Date: Tue, 15 Dec 2020 20:01:03 GMT

Unable to determine response encoding.  An "Accept" value must be specified for
this resource.  The "Accept" value must be "application/yang-data+json".
```

3.3 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 -H "Accept:application/yang-data+json" http://127.0.0.1:8080/restconf/
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/yang-data+json; charset=utf-8
Server: <redacted>
Content-Length: 137
Date: Tue, 15 Dec 2020 20:00:11 GMT

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


3.4 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 in 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 -H "Accept:application/yang-data+json" \
       http://127.0.0.1:8080/restconf/ds/ietf-datastores:operational/ietf-yang-library:yang-library
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/yang-data+json; charset=utf-8
Content-Length: 6105
Date: Thu, 14 Jan 2021 22:54:28 GMT
Server: <redacted>

{
  "ietf-yang-library:modules-state": {
    "module-set-id": "TBD",
    "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": "implement"
      },
      {
        "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",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-sztp-conveyed-info",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-crypto-types",
      "revision": "2021-01-14",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-crypto-types",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-truststore",
      "revision": "2021-01-14",
      "feature": [
        "certificates"
      ],
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-truststore",
      "conformance-type": "import"
    },
    {
      "name": "ietf-keystore",
      "revision": "2021-01-14",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-keystore",
      "conformance-type": "import"
    },
    {
      "name": "ietf-tcp-common",
      "revision": "2021-01-14",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-tcp-common",
      "conformance-type": "import"
    },
    {
      "name": "ietf-tcp-client",
      "revision": "2021-01-14",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-tcp-client",
      "conformance-type": "import"
    },
    {
      "name": "ietf-tcp-server",
      "revision": "2021-01-14",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-tcp-server",
      "conformance-type": "import"
    },
    {
      "name": "ietf-tls-common",
      "revision": "2021-01-14",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-tls-common",
      "conformance-type": "import"
    },
    {
      "name": "ietf-tls-client",
      "revision": "2021-01-14",
      "feature": [
        "x509-certificate-auth",
        "client-auth-config-supported"
      ],
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-tls-client",
      "conformance-type": "import"
    },
    {
      "name": "ietf-tls-server",
      "revision": "2021-01-14",
      "feature": [
        "x509-certificate-auth",
        "client-auth-config-supported"
      ],
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-tls-server",
      "conformance-type": "import"
    },
    {
      "name": "ietf-http-client",
      "revision": "2021-01-14",
      "feature": [
        "basic-auth"
      ],
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-http-client",
      "conformance-type": "import"
    },
    {
      "name": "ietf-http-server",
      "revision": "2021-01-14",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-http-server",
```



```
        "conformance-type": "import"
      },
      {
        "name": "ietf-restconf-client",
        "revision": "2021-01-14",
        "feature": [
          "https-initiate"
        ],
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-restconf-client",
        "conformance-type": "import"
      },
      {
        "name": "ietf-restconf-server",
        "revision": "2021-01-14",
        "feature": [
          "http-listen",
          "https-listen"
        ],
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-restconf-server",
        "conformance-type": "import"
      },
      {
        "name": "wn-x509-c2n",
        "revision": "2021-01-14",
        "namespace": "https://watsen.net/wnc2n",
        "conformance-type": "implement"
      },
      {
        "name": "wn-app-rpcs",
        "revision": "2021-01-14",
        "namespace": "https://watsen.net/app-rpcs",
        "conformance-type": "implement"
      },
      {
        "name": "wn-app",
        "revision": "2021-01-14",
        "namespace": "https://watsen.net/wnapp",
        "feature": [
        ],
        "conformance-type": "import"
      },
      {
        "name": "wn-sztpd-rpcs",
        "revision": "2021-01-14",
        "namespace": "https://watsen.net/sztpd-rpcs",
        "conformance-type": "implement"
      },
      {
        "name": "wn-sztpd-0",
        "revision": "2021-01-14",
        "feature": [
          "onboarding-supported"
        ],
        "namespace": "https://watsen.net/sztpd-0",
        "conformance-type": "import"
      },
      {
        "name": "wn-sztpd-1",
        "revision": "2021-01-14",
        "feature": [
          "device-ownership-verification"
        ],
        "namespace": "https://watsen.net/sztpd-1",
        "conformance-type": "implement"
      }
    ]
  }
}
```


3.5 Get the Current (Default) Configuration

Request:

```
$ curl -i -H "Accept:application/yang-data+json" \
       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"
                        }
                    }
                }
            ]
        }
    }
}
```

3.6 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
```


3.7 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 wrapping per RFC 8792]

$ 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
```

3.8 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 wrapping per RFC 8792]

$ 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
```


3.9 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
```


3.10 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:

```
================================== NOTE: '\' 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": "127.0.0.1",
                            "local-port": $NBI_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": "127.0.0.1",
                            "local-port": $SBI_PORT
                        },
                        "tls-server-parameters": {
                            "server-identity": {
                                "certificate" : {
                                    "reference": {
                                        "asymmetric-key": "sbi-server-end-entity-key",
                                        "certificate": "sbi-server-end-entity-cert"
                                    }
                                }
                            },
                            "client-authentication": {
                              "ca-certs": {
                                "local-truststore-reference": "my-device-identity-ca-certs"
                              }
                            }
                        }
                    }
                }
            ]
        }
    },
    "wn-sztpd-1:admin-accounts": {
        "admin-account": [
            {
                "email-address": "my-admin@example.com",
                "password": "\$0\$my-secret",
                "access": "unrestricted"
            }
        ]
    },
    "wn-sztpd-1:keystore": {
        "asymmetric-keys": {
            "asymmetric-key": [
                {
                    "name": "nbi-server-end-entity-key",
                    "public-key-format": "ietf-crypto-types:subject-public-key-info-format",
                    "public-key": "$NBI_PUB_KEY_B64",
                    "private-key-format": "ietf-crypto-types:ec-private-key-format",
                    "cleartext-private-key": "$NBI_PRI_KEY_B64",
                    "certificates": {
                        "certificate": [
                            {
                                "name": "nbi-server-end-entity-cert",
                                "cert-data": "$NBI_EE_CERT_B64"
                            }
                        ]
                    }
                },
                {
                    "name": "sbi-server-end-entity-key",
                    "public-key-format": "ietf-crypto-types:subject-public-key-info-format",
                    "public-key": "$SBI_PUB_KEY_B64",
                    "private-key-format": "ietf-crypto-types:ec-private-key-format",
                    "cleartext-private-key": "$SBI_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/1.1 204 No Content
Date: Mon, 20 Feb 2023 02:34:29 GMT
Server: <redacted>
```


3.11 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.

3.12 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.

```
================================ NOTE: '\\' 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 -H "Accept:application/yang-data+json" http://127.0.0.\
\1:8080/restconf/ds/ietf-datastores:operational/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 -H "Accept:application/yang-data+json" https://127.0.0\
\.1:8080/restconf/ds/ietf-datastores:operational/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 -H "Accept:application/ya\
\ng-data+json" https://127.0.0.1:8080/restconf/ds/ietf-datastores:operational/ietf-yang-library:yang-librar\
\y 1>> output/get_yang_library.out 2> /dev/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 -H "Accept:application/yang-data+json" http://127.0.0.1:90\
\90/restconf/ds/ietf-datastores:operational/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 -H "Accept:application/yang-data+json" https://127.0.0.1:9\
\090/restconf/ds/ietf-datastores:operational/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 -H "Accept:application/yang-d\
\ata+json" 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

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 -H "Accept:application/yang-data+json" https://127\
\.0.0.1:9090/restconf/ds/ietf-datastores:operational/ietf-yang-library:yang-library 1>> output/get_yang_lib\
\rary.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
ore details here: https://curl.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
Content-Length: 5977
Date: Mon, 20 Feb 2023 02:34:37 GMT
Server: <redacted>



=== 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
ore details here: https://curl.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: application/yang-data+json; charset=utf-8
Content-Length: 143
Date: Mon, 20 Feb 2023 02:34:37 GMT
Server: <redacted>


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

HTTP/1.1 200 OK
Content-Type: application/yang-data+json; charset=utf-8
Content-Length: 1967
Date: Mon, 20 Feb 2023 02:34:37 GMT
Server: <redacted>
```


3.13 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.

```
================================ NOTE: '\' 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" : "BASE64VALUE="
  }
}
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.pe\
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 \
1> 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:

```
HTTP/1.1 404 Not Found
Content-Type: application/yang-data+json; charset=utf-8
Content-Length: 198
Date: Mon, 20 Feb 2023 02:34:38 GMT
Server: <redacted>

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


3.14 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/1.1 201 Created
Content-Length: 0
Content-Type: application/octet-stream
Date: Mon, 20 Feb 2023 02:34:38 GMT
Server: <redacted>
```


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.

```
================================ NOTE: '\' 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-da\
tastores:running 1> output/post_conveyed_information_responses.out 2>/dev/null

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

Output:

```
HTTP/1.1 201 Created
Content-Length: 0
Content-Type: application/octet-stream
Date: Mon, 20 Feb 2023 02:34:38 GMT
Server: <redacted>
```


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) tells STZPD to return the just configured 'my-redirect-information'.

```
================================== NOTE: '\' 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
Content-Length: 0
Content-Type: application/octet-stream
Date: Mon, 20 Feb 2023 02:34:38 GMT
Server: <redacted>
```


3.15 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)

```
================================== NOTE: '\' 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" : "BASE64VALUE="
  }
}
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.pe\
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 \
1> 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:

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

HTTP/1.1 200 OK
Content-Type: application/yang-data+json; charset=utf-8
Content-Length: 2675
Date: Mon, 20 Feb 2023 02:34:39 GMT
Server: <redacted>

{
  "ietf-sztp-bootstrap-server:output": {
    "conveyed-information": "MIIHlQYLKoZIhvcNAQkQASuggge EBIIHgHsKICAiaWV0Zi1zenRwLWNvbnZleWVkLWluZm86cmVkaX\
JlY3QtaW5mb3JtYXRpb24iOiB7CiAgICAiYm9vdHN0cmFwLXNlcnZlci6IFsKICAgICAgewogICAgICAgICJhZGRyZXNzIjogIjEyNy4wL\
jAuMSIsCiAgICAgICAgInBvcnQiOiA5NDQzLAogICAgICAgICJ0cnVzdC1hbmNob3IiOiAiTUlJRkRnWUpLb1pJaHZjTkFRY0NvSUlFL3pD\
Q0JQc0NBUUV4QURBTEJna3Foa2lHOXcwQkJ3R2dnZ1RqTUlJQ1dUQ0NBZitnQXdJQkFnSUJBVEFLQmdncWhrak9QUVFEQWpCMU1Rc3dDUVl\
EVlFRR0V3SllXREVkTUJzR0ExVUVDQXdVVFhrZ1UzUmhkR1VnYjNJZ1VISnZkbWx1WTJVeEdEQVdCZ05WQkFvTUQwMTVJRTI5WjJGdWFYcG\
hkR2x2YmpFUU1BNEdBMVVFQ3d3SFRYa2dWVzVwZERFYk1Ca0dBMVVFQXd3U2MySnBMM05sY25abGNpOXliMjkwTFdOaE1DQVhVEVl4TVRJe\
U9URTJNelF3TmxvWUR6azVPVGt4TWpNeE1qTTFPVFU1V2pCMU1Rc3dDUVlEVlFRR0V3SllXREVkTUJzR0ExVUVDQXdVVFhrZ1UzUmhkR1Vn\
YjNJZ1VISnZkbWx1WTJVeEdEQVdCZ05WQkFvTUQwMTVJRTI5WjJGdWFYcGhkR2x2YmpFUU1BNEdBMVVFQXd3U2MySnBMM05sY25abGNpOXliMjkwTFdOaE1Ga3dFd1lIS29aSXpqMENBUVlJS29aSXpqMERBUWNEUWdBRWlvelFoYnorelA2aj\
NQd29idzF0L1ZtQlhWUWxYdDVaNlV6b3RScHJkb010TzlyMWQ3VTlNRER1bENIRVNpblRRbC96V0w2OEhEdmpHK2doa01ZSldhTitNSHd3S\
FFZRFZSME9CQllFRk5Jc3VtK0RVYndRN0tleEF1ODlwR2c5WG1NT01Bd0dBMVVkRXdRRk1BTUJBZjh3RGdZRFZSMFBBUUgvQkFRREFnRUdN\
RDBHQTFVZEh3UTJNRFF3TXFBd29DNkdMR2gwZEhBNkx5OWpjbXd1WlhoaGJYQnNaUzVqYjIwL1kyRTljMkpwT25ObGNuWmxjanB5YjI5MEx\
XTmhNQW9HQ0NxR1NNNDlCQU1DQTBnQU1FVUNJRllCWkxneTBtN1czcTFmd0xNaG5jUndEUVgwV29RQzE5VDVCc2ZPSnpYMkFpRUFxQlYxS0\
4wbGNqITVHZ2lXaHJQNFk2ZUhzQnBhSmppckNSMFFFcnFFRmpNd2dnS0NNSUlDS0tBREFnRUNBZ0VDTUFvR0NDcUdTTTQ5QkFNQ01IVXhDe\
kFKQmdOVkJBWVRBbGhZTVIwd0d3WURWUVFJREJSTmVTQlRkR0YwWlNCdmNpQlFjbTkyYVc1alpURVlNQllHQTFVRUNnd1BUWGtnVDNKblIX\
NXBlbUYwYVc5dU1SQXdEZ1lEVlFRTERBZE5lU0JWYm1sME1Sc3dHUVlEVlFRRERCSnpZbWt2YzJWeWRtVnlMM0p2YjNRdFkyRXdJQmNOTWp\
FeE1qSTVNVFl6TkRBM1doZ1BPVGs1T1RFeU16RXlNelU1TlRsYU1Ic3hDekFKQmdOVkJBWVRBbGhZTVIwd0d3WURWUVFJREJSTmVTQlRkR0\
YwWlNCdmNpQlFjbTkyYVc1alpURVlNQllHQTFVRUNnd1BUWGtnVDNKbllXNXBlbUYwYVc5dU1SQXdEZ1lEVlFRTERBZE5lU0JWYm1sME1TR\
XdId1lEVlFRRERCaHpZbWt2YzJWeWRtVnlMMmx1ZEdWeWJXVmthV0YwWlRFd1dUQVRCZ2NxaGtqT1BRSUJCZ2dxaGtqT1BRTUJCd05DQUFT\
WmpJMFNidnJOTU0xUy9PMGtnQlhzMWhpeWYxOVl3TFphRGtzMHITaEduMlra05NaTd0OWpzdDVxWXFJTHFJNXYzdHprMlF3K0l5R1NDQnV\
LbE9vV280R2dNSUdkTUIwR0ExVWREZ1FXQkJRZTRrMDRDdUxScnNrSm9lbTJMNHITM0FqNVZqQWZCZ05WSFNNRUdEQVdnQlRTTExwdmcxRz\
hFT3luc1FMdlBhUm9QVjVqRGpBTUJnTlZIUk1FQlRBREFRSC9NQTRHQTFVZER3RUIvd1FFQXdJQkJqQTlCZ05WSFI4RU5qQTBNREtnTUtBd\
WhpeG9kSFJ3T2k4dllkzSnNMbVY0WVcxd2JHVXVZMjl0UDJOaFBYTmlhVHB6WlhKMlpYSTZjbTl2ZEMxallUQUtCZ2dxaGtqT1BRUURBZ05J\
QURCRkFpRUFnaE9QNUtDWWZ6dzFPSEY0amJBeVhHUnRqT2JXbVVFM0VJSEIvT1hXYVV3Q0lCdnpOWjgwOEZJbVpJSUhaaHBLYlNYVTBHeXJ\
rbjlIOHJ1amZESnJoeWxsTVFBPSIKICAgICAgfQogICAgXQogIH0KfQ==="
  }
}
```


3.16 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
Date: Mon, 20 Feb 2023 02:34:39 GMT
Server: <redacted>

HTTP/1.1 204 No Content
Date: Mon, 20 Feb 2023 02:34:39 GMT
Server: <redacted>

HTTP/1.1 204 No Content
Date: Mon, 20 Feb 2023 02:34:40 GMT
Server: <redacted>
```


3.17 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 $2}'`
cat << EOM > boot-images.json
{
    "wn-sztpd-1:boot-images": {
        "boot-image": [
            {
                "name": "my-boot-image.img",
                "download-uri": [ "https://example.com/my-boot-image.img" ],
                "image-verification": [
                    {
                        "hash-algorithm": "sha-256",
                        "hash-value": "$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:

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

HTTP/1.1 400 Bad Request
Content-Type: text/plain; charset=utf-8
Content-Length: 424
Date: Mon, 20 Feb 2023 02:34:40 GMT
Server: <redacted>

{
  "ietf-restconf:errors": {
    "error": [
      {
        "error-type": "application",
        "error-tag": "invalid-value",
        "error-message": "Validation failed: {/wn-sztpd-1:boot-images/boot-image[name=\"my-boot-image.img\"\
]/image-verification[hash-algorithm=\"wn-sztpd-1:sha-256\"]/hash-algorithm} invalid-type: 'wn-sztpd-1:sha-2\
56' not derived from 'ietf-sztp-conveyed-info:hash-algorithm'"
      }
    ]
  }
}
```

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
Content-Length: 0
Content-Type: application/octet-stream
Date: Mon, 20 Feb 2023 02:34:40 GMT
Server: <redacted>
```

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:runnin\
\g 1> output/post_configurations.out 2>/dev/null

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

Output:


```
HTTP/1.1 201 Created
Content-Length: 0
Content-Type: application/octet-stream
Date: Mon, 20 Feb 2023 02:34:40 GMT
Server: <redacted>
```


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-da\
tastores:running 1> output/post_conveyed_information_responses2.out 2>/dev/null

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

Output:

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

HTTP/1.1 400 Bad Request
Content-Type: text/plain; charset=utf-8
Content-Length: 358
Date: Mon, 20 Feb 2023 02:34:40 GMT
Server: <redacted>

{
  "ietf-restconf:errors": {
    "error": [
      {
        "error-type": "application",
        "error-tag": "invalid-value",
        "error-message": "Validation failed: {/wn-sztpd-1:conveyed-information-responses/onboarding-informa\
tion-response[name=\"my-onboarding-information\"]/onboarding-information/boot-image} instance-required"
      }
    ]
  }
}
```

Configure the 'response-manager' node inside the device object to return "my-onboarding-information". A "catch-all" rule tells STZPD to return the just configured 'my-onboarding-information'.

```
================================== NOTE: '\' 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:

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

HTTP/1.1 400 Bad Request
Content-Type: text/plain; charset=utf-8
Content-Length: 402
Date: Mon, 20 Feb 2023 02:34:41 GMT
Server: <redacted>

{
  "ietf-restconf:errors": {
    "error": [
      {
        "error-type": "application",
        "error-tag": "invalid-value",
        "error-message": "Validation failed: {/wn-sztpd-1:devices/device[serial-number=\"my-serial-number\"\
]/response-manager/matched-response[name=\"catch-all-response\"]/response/conveyed-information/onboarding-i\
nformation/reference} instance-required"
      }
    ]
  }
}
```


3.18 Simulate Device Getting Onboarding Information

Request: (Same as in the Simulate Device Trying to Get Bootstrapping Data section)

```
================================== NOTE: '\' 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" : "BASE64VALUE="
  }
}
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.pe\
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 \
1> 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:

```
HTTP/1.1 404 Not Found
Content-Type: application/yang-data+json; charset=utf-8
Content-Length: 198
Date: Mon, 20 Feb 2023 02:34:41 GMT
Server: <redacted>

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


3.19 Simulate Device Posting a Progress Report

In addition to RFC 8572 enabling a device to obtain bootstrapping data, it also enables a device to post progress reports that enable a controller / NMS application to:

1) learn of any issues preventing bootstrapping from succeeding.
2) learn of any dynamically-generated SSH host keys and/or TLS certificates, such as may be needed to authenticate the device via other protocols (e.g., NETCONF, RESTCONF, etc.)

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

#!/bin/sh

# initialize the 'input' node for the RPC
  cat << EOM > bootstrap-complete.json
{
  "ietf-sztp-bootstrap-server:input" : {
    "progress-type" : "bootstrap-complete",
    "message" : "Dynamically generated SSH host key included.",
    "ssh-host-keys" : {
      "ssh-host-key" : [
        {
          "algorithm" : "ssh-rsa",
          "key-data" : "BASE64VALUE="
        }
      ]
    }
  }
}
EOM

# POST it to the "report-progress" RPC resource
curl -i -X POST --data @bootstrap-complete.json -H "Content-Type:application/yang-data+json" --cacert sbi_t\
rust_chain.pem --key pki/client/end-entity/private_key.pem --cert pki/client/end-entity/my_cert.pem --user \
my-serial-number:my-secret https://127.0.0.1:9090/restconf/operations/ietf-sztp-bootstrap-server:report-pro\
gress 1> output/post_progress_report.out 2>/dev/null

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

# cleanup
rm -f bootstrap-complete.json
```

Output:

```
HTTP/1.1 204 No Content
Date: Mon, 20 Feb 2023 02:34:42 GMT
Server: <redacted>
```


3.20 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/nu\
ll

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:

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

HTTP/1.1 404 Not Found
Content-Type: text/plain; charset=utf-8
Content-Length: 262
Date: Mon, 20 Feb 2023 02:34:42 GMT
Server: <redacted>

{
  "ietf-restconf:errors": {
    "error": [
      {
        "error-type": "protocol",
        "error-tag": "unknown-element",
        "error-message": "Data node (/wn-sztpd-1:devices/device=my-serial-number/response-manager) does not\
\ exist."
      }
    ]
  }
}HTTP/1.1 404 Not Found
Content-Type: text/plain; charset=utf-8
Content-Length: 244
Date: Mon, 20 Feb 2023 02:34:42 GMT
Server: <redacted>

{
  "ietf-restconf:errors": {
    "error": [
      {
        "error-type": "protocol",
        "error-tag": "unknown-element",
        "error-message": "Data node (/wn-sztpd-1:conveyed-information-responses) does not exist."
      }
    ]
  }
}HTTP/1.1 404 Not Found
Content-Type: text/plain; charset=utf-8
Content-Length: 225
Date: Mon, 20 Feb 2023 02:34:42 GMT
Server: <redacted>

{
  "ietf-restconf:errors": {
    "error": [
      {
        "error-type": "protocol",
        "error-tag": "unknown-element",
        "error-message": "Data node (/wn-sztpd-1:boot-images) does not exist."
      }
    ]
  }
```



```
}HTTP/1.1 204 No Content
Date: Mon, 20 Feb 2023 02:34:42 GMT
Server: <redacted>

HTTP/1.1 204 No Content
Date: Mon, 20 Feb 2023 02:34:42 GMT
Server: <redacted>
```


3.21 View Audit Log

Notice how all the commands run above 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:audit-log 1> output/get_audit_log.out 2>/dev/null
```

Output:

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

HTTP/1.1 200 OK
Content-Type: application/yang-data+json; charset=utf-8
Content-Length: 7719
Date: Mon, 20 Feb 2023 02:34:43 GMT
Server: <redacted>

{
  "wn-sztpd-1:audit-log": {
    "log-entry": [
      {
        "timestamp": "2023-02-20T02:34:28Z",
        "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": "2023-02-20T02:34:37Z",
        "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": "2023-02-20T02:34:37Z",
        "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": "2023-02-20T02:34:38Z",
        "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": "2023-02-20T02:34:38Z",
        "source-ip": "127.0.0.1",
        "host": "127.0.0.1:8080",
        "method": "POST",
        "path": "/restconf/ds/ietf-datastores:running",
        "outcome": "success"
      },
      {
        "timestamp": "2023-02-20T02:34:38Z",
        "source-ip": "127.0.0.1",
        "host": "127.0.0.1:8080",
        "method": "POST",
        "path": "/restconf/ds/ietf-datastores:running",
        "outcome": "success"
      },
      {
        "timestamp": "2023-02-20T02:34:38Z",
        "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": "2023-02-20T02:34:39Z",
        "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": "2023-02-20T02:34:39Z",
      "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/response-m\
anager",
      "outcome": "success"
    },
    {
      "timestamp": "2023-02-20T02:34:39Z",
      "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": "2023-02-20T02:34:40Z",
      "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": "2023-02-20T02:34:40Z",
      "source-ip": "127.0.0.1",
      "host": "127.0.0.1:8080",
      "method": "POST",
      "path": "/restconf/ds/ietf-datastores:running",
      "outcome": "success"
    },
    {
      "timestamp": "2023-02-20T02:34:40Z",
      "source-ip": "127.0.0.1",
      "host": "127.0.0.1:8080",
      "method": "POST",
      "path": "/restconf/ds/ietf-datastores:running",
      "outcome": "success"
    },
    {
      "timestamp": "2023-02-20T02:34:40Z",
      "source-ip": "127.0.0.1",
      "host": "127.0.0.1:8080",
      "method": "POST",
      "path": "/restconf/ds/ietf-datastores:running",
      "outcome": "success"
    },
    {
      "timestamp": "2023-02-20T02:34:40Z",
      "source-ip": "127.0.0.1",
      "host": "127.0.0.1:8080",
      "method": "POST",
      "path": "/restconf/ds/ietf-datastores:running",
      "outcome": "success"
    },
    {
      "timestamp": "2023-02-20T02:34:40Z",
      "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": "2023-02-20T02:34:41Z",
      "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": "2023-02-20T02:34:42Z",
      "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/response-m\
anager",
      "outcome": "success"
    },
```



```
      {
        "timestamp": "2023-02-20T02:34:42Z",
        "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": "2023-02-20T02:34:42Z",
        "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": "2023-02-20T02:34:42Z",
        "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": "2023-02-20T02:34:42Z",
        "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"
      },
      {
        "timestamp": "2023-02-20T02:34:42Z",
        "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"
      }
    ]
  }
}
```

3.22 View Device's Bootstrapping Log

Notice how the "GET" on yang-library resource, and all of the get-bootstrapping-data RPC requests, and the report-progress RPC request, 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> outpu\
t/get_bootstrapping_log.out 2>/dev/null
```

Output:

```
HTTP/1.1 200 OK
Content-Type: application/yang-data+json; charset=utf-8
Content-Length: 7907
Date: Mon, 20 Feb 2023 02:34:44 GMT
Server: <redacted>

{
  "wn-sztpd-1:bootstrapping-log": {
    "log-entry": [
      {
        "timestamp": "2023-02-20T02:34:37Z",
        "source-ip": "127.0.0.1",
        "method": "HEAD",
        "path": "/restconf/ds/ietf-datastores:operational/ietf-yang-library:yang-library",
        "return-code": 200
      },
      {
        "timestamp": "2023-02-20T02:34:38Z",
        "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": {
            "hw-model": "model-x",
            "os-name": "vendor-os",
            "os-version": "17.3R2.1",
            "nonce": "BASE64VALUE="
          },
          "selected-response": "no-responses-configured"
        }
      }
    },
    {
      "timestamp": "2023-02-20T02:34:39Z",
      "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": {
            "hw-model": "model-x",
            "os-name": "vendor-os",
            "os-version": "17.3R2.1",
            "nonce": "BASE64VALUE="
          },
          "selected-response": "catch-all-response",
          "response-details": {
            "managed-response": {
              "conveyed-information": {
                "redirect-information": {
                  "referenced-definition": "my-redirect-information"
                }
              }
            }
          }
        }
      }
    },
    {
      "timestamp": "2023-02-20T02:34:41Z",
      "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": {
            "hw-model": "model-x",
            "os-name": "vendor-os",
            "os-version": "17.3R2.1",
            "nonce": "BASE64VALUE="
          },
          "selected-response": "no-responses-configured"
        }
      }
    },
    {
      "timestamp": "2023-02-20T02:34:42Z",
      "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": "BASE64VALUE="
                }
              ]
            }
          },
          "dynamic-callout": {
            "no-callout-configured": [
              null
            ]
          }
        }
      }
    ]
  }
}
```


4 Special Topics

This section goes over standards-based behavior that may not be immediately obvious to some readers.

4.1 HTTP Headers

SZTPD attempts to not require HTTP headers (e.g., "Accept", "Content-Type", etc.), but it strictly enforces that the correct headers are present when needed.

When an "Accept" header is not passed, an implied value will be derived from the "Content-Type" header, if it is present. Errors are returned using the implied value, or "text/plain" otherwise.

SZTPD does not have a default encoding, and so wildcard "Accept" header values, such as "/" or similar, are not supported. When the request demands a response (e.g., GET), an "Accept" header must be provided.

SZTPD is not able to auto-detect the encoding of request bodies, and thus a "Content-Type" header must be provided when a request body is included in a request.

4.2 Ordered Lists

An ordered list is one in which the order matters. In YANG, the 'ordered-by user' statement is used to indicate when the order matters for a 'list' or 'leaf-list'. Please see Section 7.7.1 in the YANG specification for details.

There are three ordered lists in SZTPD, one 'list' and two 'leaf-lists' as follows:

- leaf-lists:
  1) boot-images/boot-image/download-uri
  2) conveyed-information-responses/redirect-information-response/redirect-information/bootstrap-server
- lists:
  3) devices/device/response-manager/matched-response

The ordering of a user-ordered list is initiailly specified when the list is created, assuming more than one entry to defined at that time, and maintained thereafter whenever a new entry is added to the list (e.g., using the POST method) and/or replaced (i.e., using the PUT method) using the 'insert' and 'point' query parameters, as described in the RESTCONF specification by Section 4.8.5 and Section 4.8.6 respectively.

4.2.1 Example: POST-ing an entry after another entry

Assuming a device's "response-manager" contains the ordered list of "matched-response" entries: "resp1", "resp2", and "resp3", then the command:

```
$ cat matched-response.json
{
  "wn-sztpd-1:matched-response": [
    {
      'name': 'new',
      'response': {
        'none': [None]
      }
    }
  ]
}

$ curl -i -X POST --data @matched-response.json -H "Content-Type:application/yang-data+json" \
http://127.0.0.1:8080/restconf/ds/ietf-datastores:running/wn-sztpd-1:devices/device=my-serial\
-number/response-manager?insert=after&point=/wn-sztpd-1:devices/device=my-serial-number/respo\
nse-manager/matched-response=resp1
```

would result in the list of entries: "resp1", "new", "resp2", and "resp3".

4.2.2 Example: PUT-ing an entry before another entry

Assuming a redirect-information-response contains the ordered list of "bootstrap-server" entries: "bs1", "bs2", and "bs3", then the command:


```
$ cat bootstrap-server.json
{
  "wn-sztpd-1:bootstrap-server": [
    "bs-3"
  ]
}

$ curl -i -X PUT --data @bootstrap-server.json -H "Content-Type:application/yang-data+json" \
http://127.0.0.1:8080/restconf/ds/ietf-datastores:running/wn-sztpd-1:conveyed-information-re\
sponses/redirect-information-response=my-redirect-information/redirect-information/bootstrap\
-server=bs3?insert=before&point=/wn-sztpd-1:conveyed-information-responses/redirect-informat\
ion-response=my-redirect-information/redirect-information/bootstrap-server=bs2
```

would result in the list of entries: "bs1", "bs3", and "bs2".

4.3 Special Characters

Special characters are herein defined as those that must be escaped in order to be passed into or out of the server.

According to Section 3.5.3 of RFC 8040, any reserved characters in a resource identifier (i.e., key values) MUST be percent-encoded.

This applies to any URL that encodes a 'list' or 'leaf-list' instance, the the 'key' value contains reserved characters, they must be escaped when placed into the URL.

That said, a worst-case scenario arises when trying the insert or move an entry into the "boot-images/boot-image/download-uri" ordered list (see Ordered Lists) as, in this case, a URL must be encoded into a URL.

4.3.1 Example: POST-ing an entry after another

Assuming the configured "download-uri" leaf-list contains values as follows:

```
{
  "wn-sztpd-1:download-uri" : [
    https://cdn1.example.com/path/to/image/file ,
    https://cdn2.example.com/path/to/image/file ,
    https://cdn3.example.com/path/to/image/file
  ]
}
```

then the command:

```
$ cat download-uri.json
{
  "wn-sztpd-1:download-uri": [
    https://new4.example.com/path/to/image/file
  ]
}

$ curl -i -X POST --data @download-uri.json -H "Content-Type:application/yang-data+json" \
http://127.0.0.1:8080/restconf/ds/ietf-datastores:running/wn-sztpd-1:boot-images/boot-ima\
ge=vendoros-19.2r1b6.img?insert=after&point=/wn-sztpd-1:boot-images/boot-image=vendoros-1\
9.2r1b6.img/download-uri=https%3A%2F%2Fcdn1.example.com%2Fpath%2Fto%2Fimage%2Ffile
```

would result in the list of entries:

```
{
  "wn-sztpd-1:download-uri" : [
    https://cdn1.example.com/path/to/image/file ,
    https://new4.example.com/path/to/image/file ,
    https://cdn2.example.com/path/to/image/file ,
    https://cdn3.example.com/path/to/image/file
  ]
}
```


4.3.2 Example: PUT-ing an entry after another entry

Assuming the configured "download-uri" leaf-list contains values as follows:

```
{
  "wn-sztpd-1:download-uri" : [
    https://cdn1.example.com/path/to/image/file,
    https://new4.example.com/path/to/image/file,
    https://cdn2.example.com/path/to/image/file,
    https://cdn3.example.com/path/to/image/file
  ]
}
```

then the command:

```
$ cat download-uri.json
{
  "wn-sztpd-1:download-uri": [
    https://new4.example.com/path/to/image/file
  ]
}

$ curl -i -X POST --data @download-uri.json -H "Content-Type:application/yang-data+json" \
http://127.0.0.1:8080/restconf/ds/ietf-datastores:running/wn-sztpd-1:boot-images/boot-ima\
ge=vendoros-19.2r1b6.img/download-uri=https%3A%2F%2Fnew4.example.com%2Fpath%2Fto%2Fimage%\
2Ffile?insert=before&point=/wn-sztpd-1:boot-images/boot-image=vendoros-19.2r1b6.img/downl\
oad-uri=https%3A%2F%2Fcdn3.example.com%2Fpath%2Fto%2Fimage%2Ffile
```

would result in the list of entries:

```
{
  "wn-sztpd-1:download-uri" : [
    https://cdn1.example.com/path/to/image/file,
    https://cdn2.example.com/path/to/image/file,
    https://new4.example.com/path/to/image/file,
    https://cdn3.example.com/path/to/image/file
  ]
}
```

4.4 List Pagination

Some lists, especially lists representing time-series data (i.e., logs), can grow to be large in size. Client applications wishing to browse thru the list entries may only wish to do so in chunks or pages.

There is an effort, spearheaded by Watsen Networks, to defined a standard for list pagination. While it is too early to post even a link to the budding standard, SZTPD has initial support for three new query parameters:

Name	Methods	Description
limit	GET, HEAD	Limits the number of entries returned. If not specified, the number of entries that may be returned in unbounded.
offset	GET, HEAD	Indicates the number of entries in the result set that should the skipped over when preparing the response. If not specified, then no entries in the result set are skipped.
direction	GET, HEAD	Indicates the direction that the result set is to be traversed. If not specified, then the result set is traversed in the "forward" direction.

4.4.1 The "limit" Query Parameter

- The "limit" query parameter limits the number of entries from the working result set (i.e., after the "offset" parameter has been applied) that are returned.
- The allowed values are positive integers.
- This parameter is only allowed for the GET and HEAD methods on "list" and "leaf-list" data resources. A "400 Bad Request" status-line is returned if used for other methods or on other resource types.
- If this query parameter is not present, the number of entries that may be returned is unbounded.
- This query parameter MUST be supported for all lists and leaf-lists.

4.4.2 The "offset" Query Parameter

- The "offset" query parameter indicates the number of entries in the working result set (i.e., after the "direction" parameter has been applied) that should be skipped.
- The allowed values are positive integers. If the skip value exceeds the number of entries in the working result set, then "416 Range Not Satisfiable" status-line is returned.
- This parameter is only allowed for the GET and HEAD methods on
- This parameter is only allowed for the GET and HEAD methods on "list" and "leaf-list" data resources. A "400 Bad Request" status-line is returned if used for other methods or on other resource types.
- If this query parameter is not present, the default value is to not skip over any values from the working result set.
- This query parameter MUST be supported for all lists and leaf-lists.

4.4.3 The "direction" Query Parameter

- The "direction" query parameter indicates how the entries in the working result set should be traversed.
- The allowed values are:
  - forwards
    - Return entries in the "forward" direction. Also known as the "default" or "ascending" direction.
  - backwards
    - Return entries in the "backward" direction. Also known as the "reversed" or "descending" direction.
- This parameter is only allowed for the GET and HEAD methods on "list" and "leaf-list" data resources. A "400 Bad Request" status-line is returned if used for other methods or on other resource types.
- If this query parameter is not present, the default value is "forwards".
- This query parameter MUST be supported for all lists and leaf-lists.

4.5 Triggers

SZTPD reacts to some triggers automatically.

4.5.1 Configuration-based

SZTPD reacts to some configurations changes. 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.

4.5.2 Clock-based

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

4.6 Tracking

Much of SZTPD's reactive functionality depends on triggers.

4.6.1 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.

4.6.2 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.

4.6.3 Device Lifecycle Statistics

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

4.7 Plugins

SZTPD uses plugins to enable custom logic for Dynamic Callouts.

4.7.1 Installation

Configuring SZTPD to use a plugin is a two-step process:

1. Place the Python module (i.e. file) in the "plugins" directory on the SZTPD server[1]. The location of this directory can be determined via the following command[2]:

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

2. Configure a "plugin" under "/preferences/system/plugins"[3].

[1] This step entails file-system level access to the SZTP server.

[2] Ensure that you run the command using the same version of Python and/or Python virtual environment that SZTPD uses.

[3] In a multi-tenant deployment, only host-level administrators have access to this configuration; tenants can only use plugins that have been configured by host-level administrators.

4.7.2 Example

Assuming a plugin a installed as follows:

```
PLUGIN_DIR=`python -c 'import pkg_resources; print(pkg_resources.resource_filename("sztpd", "plugins"))'`
echo $PLUGIN_DIR
cat << EOM > $PLUGIN_DIR/my-plugin.py
import sys
def relay_progress_report_function(input, opaque):
    print("INSIDE relay_progress_report_function()...")
    print("input = " + str(input))
    print("opaque = " + str(opaque))
    print("sys.version = " + sys.version)
EOM
```

And that the configuration contains the following, in addition to other configurations:

```
"wn-sztpd-x:preferences": {
    "system": {
        "plugins": {
            "plugin": [
                {
                    "name": "my-plugin",
                    "functions": {
                        "function": [
                            {
                                "name": "relay_progress_report_function"
                            }
                        ]
                    }
                }
            ]
        }
    },
    "outbound-interactions": {
        "relay-progress-report-callout": "my-relay-progress-report-callout"
    }
},
"wn-sztpd-x:dynamic-callouts": {
    "dynamic-callout": [
        {
            "name": "my-relay-progress-report-callout",
            "rpc-supported": "wn-sztpd-rpcs:relay-progress-report",
            "callback": {
                "plugin": "my-plugin",
                "function": "relay_progress_report_function"
            },
            "opaque": {
                "empty": [None],
                "boolean": True,
                "string": "foobar",
                "list": [ 1, 2, 3 ],
                "dict": {
                    "a": 1,
                    "b": 2
                }
            }
        }
    ]
}
```

Then, when a device sends a progress report, the following may appear in the console where SZTPD is running[4]:

```
============================== Note: '\' line wrapping per RFC 8792] ================================

INSIDE relay_progress_report_function()...

input = {'serial-number': 'my-serial-number', 'source-ip-address': '127.0.0.1', 'from-device': {'iet\
f-sztp-bootstrap-server:input': {'progress-type': 'bootstrap-complete', 'message': 'message sent via\
 XML', 'ssh-host-keys': {'ssh-host-key': [{'algorithm': 'ssh-rsa', 'key-data': 'BASE64VALUE='}, {'al\
gorithm': 'rsa-sha2-256', 'key-data': 'BASE64VALUE='}]}, 'trust-anchor-certs': {'trust-anchor-cert':\
 ['BASE64VALUE=']}}}}

opaque = {'empty': [None], 'boolean': True, 'string': 'foobar', 'list': [1, 2, 3], 'dict': {'a': 1, \
'b': 2}}

sys.version = 3.8.2 (default, Nov 11 2020, 16:34:19)
[Clang 12.0.0 (clang-1200.0.32.21)]
```

[4]The "BASE64VALUE=" values are not real; they are used in this example only for readability.

5 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 unsually complex, due to the modules needing to present different product modes.

5.1 NBI Overview

The following sections are presented in sorted order.

5.1.1 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.

5.1.2 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
```


5.1.3 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.

5.1.4 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.

5.1.5 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.

5.1.6 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.

5.1.7 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.

Each device-type can optionally indicate if devices of that type authenticate themselves with an identity certificate (e.g., an IDevID) and, if so, further indicate the trust-anchor certificate in the Truststore that can authenticate the device's idenitity certificate. This enables application specific verification that the device's client certificate (i.e., its IDevID) exactly matches the expected certificate chain.

Each device-type can also optionally specify a callout SZTPD should call in order to determine if the specific tenant is the rightful owner of the devices of that device type. The callout is specified per device-type as backend logic may vary by device-type.

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.

```
================================ NOTE: '\' 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
```


5.1.8 Devices

The following tree diagram illustrates the NBI used for configuring device records.

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

  +--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/dynamic-c\
allout/name
        |                    +--:(redirect-information)
        |                    |  +--rw redirect-information
        |                    |     +--rw (local-or-reference)
        |                    |        +--:(reference)
        |                    |           +--rw reference?   -> ../../../../../../../../conveyed-information\
-responses/redirect-information-response/name
        |                    +--:(onboarding-information) {wn-sztpd-0:onboarding-supported}?
        |                       +--rw onboarding-information
        |                          +--rw (local-or-reference)
        |                             +--:(reference)
        |                                +--rw reference?   -> ../../../../../../../../conveyed-information\
-responses/onboarding-information-response/name
        +--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              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
|          |                        +--:(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       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
+--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
```


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.

5.1.9 Dynamic Callouts

The following tree diagram illustrates the NBI used for configuring dynamic callouts that, at this time, must be implemented by a plugin-based callback but, in a future release, may be implemented as via a webhook.

```
============================== NOTE: '\' 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]/functi\
ons/function/name
              +--rw opaque?      anydata
```

5.1.10 Keystore

The following tree diagram illustrates the NBI used for configuring keys held in the keystore.

```
  +--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
     |     +--rw certificates
     |     |  +--rw certificate* [name]
     |     |     +--rw name                          string
     |     |     +--rw cert-data                     ietf-crypto-types:end-entity-cert-cms
     |     |     +--ro reference-statistics
     |     |        +--ro reference-count    uint32
     |     |        +--ro last-referenced    union
     |     +--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
```

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).

5.1.11 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
   |  |  +--rw cc?        string
   |  |  +--rw body?      string
   |  +--rw passwords
   |     +--rw strength-testing!
   |        +--rw min-length?   uint16 <16>
   +--rw outbound-interactions
   |  +--rw relay-notification-callout?      -> ../../../dynamic-callouts/dynamic-callout/name
   |  +--rw relay-progress-report-callout?   -> ../../../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
```

5.1.12 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.

5.1.13 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.

5.1.14 Transport

The following tree diagram illustrates the NBI used for configuring transport records.

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

  +--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
           |        |  |                    +--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
```

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.

5.1.15 Truststore

The following tree diagram illustrates the NBI used for configuring trust-anchors held in the truststore.

```
  +--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
           |  +--ro reference-statistics
           |     +--ro reference-count     uint32
           |     +--ro last-referenced     union
           +--ro reference-statistics
              +--ro reference-count     uint32
              +--ro last-referenced     union
```

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

5.2 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/"))'
```

5.2.1 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
  |  |  |  +--rw cc?        string
  |  |  |  +--rw body?      string
  |  |  +--rw passwords
  |  |     +--rw strength-testing!
  |  |        +--rw min-length?   uint16 <16>
  |  +--rw outbound-interactions
  |  |  +--rw relay-notification-callout?      -> ../../../dynamic-callouts/dynamic-callout/name
  |  |  +--rw relay-progress-report-callout?   -> ../../../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
   |        |  +--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
   |  |     +--rw certificates
   |  |     |  +--rw certificate* [name]
   |  |     |     +--rw name                          string
   |  |     |     +--rw cert-data                     ietf-crypto-types:end-entity-cert-cms
   |  |     |     +--ro reference-statistics
   |  |     |        +--ro reference-count    uint32
   |  |     |        +--ro last-referenced    union
   |  |     +--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
   +--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
   |              +--rw opaque?     anydata
   +--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)
         |     |        +--:(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/dynamic-c\
allout/name
         |                    +--:(redirect-information)
         |                    |  +--rw redirect-information
         |                    |     +--rw (local-or-reference)
         |                    |        +--:(reference)
         |                    |           +--rw reference?   -> ../../../../../../../../conveyed-information\
-responses/redirect-information-response/name
         |                    +--:(onboarding-information) {wn-sztpd-0:onboarding-supported}?
         |                       +--rw onboarding-information
         |                          +--rw (local-or-reference)
         |                             +--:(reference)
         |                                +--rw reference?   -> ../../../../../../../../conveyed-information\
-responses/onboarding-information-response/name
         +--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               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
|          |                        +--:(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       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
|          |
+--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
```


5.2.2 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
  |  |  |  +--rw cc?        string
  |  |  |  +--rw body?      string
  |  |  +--rw passwords
  |  |     +--rw strength-testing!
  |  |        +--rw min-length?   uint16 <16>
  |  +--rw outbound-interactions
  |  |  +--rw relay-notification-callout?      -> ../../../dynamic-callouts/dynamic-callout/name
  |  |  +--rw relay-progress-report-callout?   -> ../../../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
  |        |  +--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
   |  |     +--rw certificates
   |  |     |  +--rw certificate* [name]
   |  |     |     +--rw name                          string
   |  |     |     +--rw cert-data                     ietf-crypto-types:end-entity-cert-cms
   |  |     |     +--ro reference-statistics
   |  |     |        +--ro reference-count    uint32
   |  |     |        +--ro last-referenced    union
   |  |     +--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
   +--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
   |           +--rw opaque?     anydata
   +--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
   |        |        |  |                    +--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 tenants
   +--rw tenant* [name]
      +--rw name                              string
      +--rw preferences
      |  +--rw admin-accounts
      |  |  +--rw new-account-verification
      |  |  |  +--rw subject?   string
      |  |  |  +--rw cc?        string
      |  |  |  +--rw body?      string
      |  |  +--rw passwords
      |  |     +--rw strength-testing!
      |  |        +--rw min-length?   uint16 <16>
      |  +--rw outbound-interactions
      |     +--rw relay-notification-callout?      -> ../../../dynamic-callouts/dynamic-callout/name
      |     +--rw relay-progress-report-callout?   -> ../../../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
      |        |  +--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
      |  |     +--rw certificates
      |  |     |  +--rw certificate* [name]
      |  |     |     +--rw name                          string
      |  |     |     +--rw cert-data                     ietf-crypto-types:end-entity-cert-cms
      |  |     |     +--ro reference-statistics
      |  |     |        +--ro reference-count    uint32
      |  |     |        +--ro last-referenced    union
      |  |     +--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
        +--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
        |              +--rw opaque?     anydata
        +--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
+--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              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?
stri\
ng
|           |                       |                       +--:(exited-normally)
|           |                       |                          +--ro exited-normally?
string
|           |                       +--:(redirect-information)
|           |                       |  +--ro redirect-information
|           |                       |     +--ro (local-or-reference)
|           |                       |        +--:(reference)
|           |                       |           +--ro referenced-definition?   string
|           |                       +--:(onboarding-information) {wn-sztpd-0:onboarding-suppo\
rted}?
|           |                          +--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       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
|
+--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
```


6 Southbound Interface

The southbound interface implements the "bootstrap server" defined in RFC 8572.

This section is used primarily to illustrate usage for developers.

6.1 Determining the SBI's Root Prefix

Just as with the NBI, described by Fetching Host-meta, a command can be sent to the sent to the SBI to detemine the RESTCONF server's root prefix. For the purpose of this tutorial, assume that the response is the same as shown for the NBI.

6.2 Determining the Encodings Supported by the SBI

Just as with the NBI, described by Determining the Encodings Supported, a command can be sent to the SBI to demine what encodings the server supports, shown below in XML. Note that no "Content-Type" header was included though required for a GET command, and an "Accept" header is returned, indicating that both JSON and XML are supported.

Request:

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

Response:

```
HTTP/1.1 400 Bad Request
Accept: application/yang-data+json, application/yang-data+xml
Accept-Charset: utf-8
Content-Type: text/plain; charset=utf-8
Server: <redacted>
Content-Length: 191
Date: Tue, 15 Dec 2020 20:01:03 GMT

Unable to determine response encoding. An "Accept" value must be specified for
this resource. The "Accept" value must be either "application/yang-data+json"
or "application/yang-data+xml".
```

6.3 Determining YANG-library Version Used by the SBI

Just as in Fetching the RESTCONF Root Resource, a command can be sent to the SBI to demine what encodings the server supports, shown below in XML

Request:

```
$ curl -i -H "Accept:application/yang-data+xml" http://127.0.0.1:8080/restconf/
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/yang-data+xml; charset=utf-8
Server: <redacted>
Content-Length: 163
Date: Tue, 15 Dec 2020 20:00:19 GMT

<restconf xmlns="urn:ietf:params:xml:ns:yang:ietf-restconf">
    <data/>
    <operations/>
    <yang-library-version>2016-06-21</yang-library-version>
</restconf>
```

6.4 The 'get-bootstrapping-data' RPC returning "redirect information":

6.4.1 XML-based Example

Request:

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

#!/bin/sh

# initialize the 'input' node for the RPC
```



```
cat << EOM > input.xml
<input xmlns="urn:ietf:params:xml:ns:yang:ietf-sztp-bootstrap-server">
    <hw-model>model-x</hw-model>
    <os-name>vendor-os</os-name>
    <os-version>17.3R2.1</os-version>
    <nonce>BASE64VALUE=</nonce>
</input>
EOM

# POST 'input' for the "get-bootstrapping-data" RPC resource
curl -i -X POST --data @input.xml -H "Content-Type:application/yang-data+xml" --cacert sbi_trust_chain.pem \
\--key pki/client/end-entity/private_key.pem --cert pki/client/end-entity/my_cert.pem --user my-serial-numb\
\er:my-secret https://127.0.0.1:9090/restconf/operations/ietf-sztp-bootstrap-server:get-bootstrapping-data \
\1> output/post_rpc_input_xml.out 2>/dev/null

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

# cleanup
rm -f input.xml
```

Response:

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

HTTP/1.1 200 OK
Content-Type: application/yang-data+xml; charset=utf-8
Content-Length: 2845
Date: Mon, 20 Feb 2023 02:34:39 GMT
Server: <redacted>

<?xml version="1.0" ?>
<output xmlns="urn:ietf:params:xml:ns:yang:ietf-sztp-bootstrap-server">
  <conveyed-information>MIIH3wYLKoZIhvcNAQkQASqgggfOBIIHyjw/eG1sIHZlcnNpb249IjEuMCIgPz4KPHJlZGlyZWN0LWluZm9\
ybWF0aW9uIHhtbG5zPSJ1cm46aWV0ZjpwYXJhbXM6eG1sOm5zOnlhbmc6aWV0Zi1zenRwLWNvbnZleWVkLWluZm8iPgogIDxib290c3RyYX\
Atc2VydmVyPgogICAgPGFkZHJlc3M+MTI3LjAuMC4xPC9hZGRyZXNzPgogICAgPHBvcnQ+OTQ0MzwvcG9ydD4KICAgIDx0cnVzdC1hbmNob\
3I+TUlJRkRnWUpLb1pJaHZjTkFRY0NvSUlFL3pDQ0JQc0NBUUV4QURBTEJna3Foa2lHOXcwQkJ3R2dnZ1RqTUlJQ1dUQ0NBZitnQXdJQkFn\
SUJBVEFLQmdncWhrak9QUVFEQWpCMU1Rc3dDUVlEVlFRR0V3SllXREVkTUJzR0ExVUVDQXdVVFhrZ1UzUmhkR1VnYjNJZ1VISnZkbWx1WTJJ\
VeEdEQVdCZ05WQkFvTUQwMTVJRTI5WjJGdWFYcGhkR2x2YmpFUU1BNEdBMVVFQ3d3SFRYa2dWVzVwZERFYk1Ca0dBMVVFQXd3U2MySnBMM0\
5sY25abGNpOXliMjkwTFdOaE1DQVhEVEl4TVRJeU9URTJNelF3TmxvWUR6azVPVGt4TWpNeE1qTTFPVFU1V2pCMU1Rc3dDUVlEVlFRR0V3S\
lIXREVkTUJzR0ExVUVDQXdVVFhrZ1UzUmhkR1VnYjNJZ1VISnZkbWx1WTJVeEdEQVdCZ05WQkFvTUQwMTVJRTI5WjJGdWFYcGhkR2x2YmpF\
UU1BNEdBMVVFQ3d3SFRYa2dWVzVwZERFYk1Ca0dBMVVFQXd3U2MySnBMM05sY25abGNpOXliMjkwTFdOaE1Ga3dFd1lIS29aSXpqMENBUVI\
JS29aSXpqMERBUWNEUWdBRWlvelFoYnorelA2ajNQd29idzF0L1ZtQlhWUWxYdDVaNlV6b3RScHJkb010TzlyMWQ3VTlNRER1bENIRVNpbl\
RRbC96V0w2OEhEdmpHK2doa01ZSldhTitNSHd3SFFZRFZSME9CQllFRk5Jc3VtK0RVYndRN0tleEF1ODlwR2c5WG1NT01Bd0dBMVVkRXdRR\
k1BTUJBZjh3RGdZRFZSMFBBUUgvQkFRREFnRUdNRDBHQTFVZEh3UTJNRFF3TXFBd29DNkdMR2gwZEhBNkx5OWpjbXd1WlhoaGJYQnNaUzVq\
YjIwL1kyRTljMkpwT25ObGNuWmxjanB5Yjl5MExXTmhNQW9HQ0NxR1NNNDlCQU1DQTBnQU1FVUNJRllCWkxneTBtN1czcTFmd0xNaG5jUnd\
EUVgwV29RQzE5VDVCc2ZPSnpYMkFpRUFxQlYxS04wbGNqTTVHZ2lXaHJQNFk2ZUhzQnBhSmppckNSMFFFcnFFRmpNd2dnS0NNSUlDS0tBRE\
FnRUNBZ0VDTUFvR0NDcUdTTTQ5QkFNQ0IIVXhDekFKQmdOVkJBWVRBbGhZTVIwd0d3WURWUVFJREJSTmVTQlRkR0YwWlNCdmNpQlFjbTkyY\
Vc1alpURVlNQllHQTFVRUNnd1BUWGtnVDNKblXNXBlbUYwYVc5dU1SQXdEZ1lEVlFRTERBZE5lU0JWYm1sME1Sc3dHUVlEVlFRRERCSnpZ\
bWt2YzJWeWRtVnlMM0p2YjNRdFkyRXdJQmNOTWpFeE1qSTVNVF16TkRBM1doZ1BPVGs1T1RFeU16RXlNelU1TlRsYU1Ic3hDekFKQmdOVkJ\
BWVRBbGhZTVIwd0d3WURWUVFJREJSTmVTQlRkR0YwWlNCdmNpQlFjbTkyYVc1alpURVlNQllHQTFVRUNnd1BUWGtnVDNKblXNXBlbUYwYV\
c5dU1SQXdEZ1lEVlFRTERBZE5lU0JWYm1sME1TRXdId1lEVlFRRERCaHpZbWt2YzJWeWRtVnlMMmx1ZEdWeWJXVmthV0YwWlRFd1dUQVRCZ\
2NxaGtqT1BRSUJCZ2dxaGtqT1BRTUJCd05DQUFTWmpJMFNidnJOTU0xUy9PMGtnQlhzMWhpeWYxOVl3TFphRGtzMHITaEduMlIra05NaTd0\
OWpzdDVxWXFJTHFJNXYzdHprMlF3K0l5R1NDQnVLbE9vV280R2dNSUdkTUIwR0ExVWREZ1FXQkJRZTRrMDRDdUxScnNrSm9lbTJMNHITM0F\
qNVZqQWZCZ05WSFNNRUdEQVdnQlRTTExwdmcxRzhFT3luc1FMdlBhUm9QVjVqRGpBTUJnTlZIUk1FQlRBREFRSC9NQTRHQTFVZER3RUIvd1\
FFQXdJQkJqQTlCZ05WSFI4RU5qQTBNREEtnTUtBdWhpeG9kSFJ3T2k4dlkzSnNMbVY0WVcxd2JHVXVZMjl0UDJOaFBYTmlhVHB6WlhKMlpYS\
TZjbTI2ZEMxallUQUtCZ2dxaGtqT1BRUURBZ05JQURCRkFpRUFnaE9QNUtDWWZ6dzFPSEY0amJBeVhHUnRqT2JXbVVFM0VJSEIvT1hXYVV3\
Q0lCdnpOWjgwOEZJbVpJSUhaaHBLYlNYVTBHeXJrbjlIOHJ1amZESnJoeWxsTVFBPTwvdHJ1c3QtYW5jaG9yPgogIDwvYm9vdHN0cmFwLXN\
lcnZlcj4KPC9yZWRpcmVjdC1pbmZvcm1hdGlvbj4K</conveyed-information>
</output>
```


6.4.2 JSON-based Example

Request:

```
================================ NOTE: '\' 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" : "BASE64VALUE="
  }
}
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.pe\
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 \
1> 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
```

Response:

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

HTTP/1.1 200 OK
Content-Type: application/yang-data+json; charset=utf-8
Content-Length: 2675
Date: Mon, 20 Feb 2023 02:34:39 GMT
Server: <redacted>

{
  "ietf-sztp-bootstrap-server:output": {
    "conveyed-information": "MIIHlQYLKoZIhvcNAQkQASugggeEBIIHgHsKICAiaWV0Zi1zenRwLWNvbnZleWVkLWluZm86cmVkaX\
JlY3QtaW5mb3JtYXRpb24iOiB7CiAgICAiYm9vdHN0cmFwLXNlcnZlci6IFsKICAgICAgewogICAgICAgICJhZGRyZXNzIjogIjEyNy4wL\
jAuMSIsCiAgICAgICAgInBvcnQiOiA5NDQzLAogICAgICAgICJ0cnVzdC1hbmNob3IiOiAiTUlJRkRnWUpLb1pJaHZjTkFRY0NvSUlFL3pD\
Q0JQc0NBUUV4QURBTEJna3Foa2lHOXcwQkJ3R2dnZ1RqTUlJQ1dUQ0NBZitnQXdJQkFnSUJBVEFLQmdncWhrak9QUVFEQWpCMU1Rc3dDUVl\
EVlFRR0V3SlXREVkTUJzR0ExVUVDQXdVVFhrZ1UzUmhkR1VnYjNJZ1VISnZkbWx1WTJVeEdEQVdCZ05WQkFvTUQwMTVJRTI5WjJGdWFYcG\
hkR2x2YmpFUU1BNEdBMVVFQ3d3SFRYa2dWVzVwZERFYk1Ca0dBMVVFQXd3U2MySnBMM05sY25abGNpOXliMjkwTFdOaE1DQVhEVEl4TVRJe\
U9URTJNelF3TmxvWUR6azVPVGt4TWpNeE1qTTFPVFU1V2pCMU1Rc3dDUVlEVlFRR0V3SlXREVkTUJzR0ExVUVDQXdVVFhrZ1UzUmhkR1Vn\
YjNJZ1VISnZkbWx1WTJVeEdEQVdCZ05WQkFvTUQwMTVJRTI5WjJGdWFYcGhkR2x2YmpFUU1BNEdBMVVFQ3d3SFRYa2dWVzVwZERFYk1Ca0d\
BMVVFQXd3U2MySnBMM05sY25abGNpOXliMjkwTFdOaE1Ga3dFd1lIS29aSXpqMENBUVlJS29aSXpqMERBUWNEUWdBRWlvelFoYnorelA2aj\
NQd29idzF0L1ZtQlhWUWxYdDVaNlV6b3RScHJkb010TzlyMWQ3VTlNRER1bENIRVNpblRRbC96V0w2OEhEdmpHK2doa01ZSldhTitNSHd3S\
FFZRFZSME9CQllFRk5Jc3VtK0RVYndRN0tleEF1ODlwR2c5WG1NT01Bd0dBMVVkRXdRRk1BTUJBZjh3RGdZRFZSMFBBUUgvQkFRREFnRUdN\
RDBHQTFVZEh3UTJNRFF3TXFBd29DNkdMR2gwZEhBNkx5OWpjbXd1WlhoaGJYQnNaUzVqYjIwL1kyRTljMkpwT25ObGNuWmxjanB5YjI5MEx\
XTmhNQW9HQ0NxR1NNNDlCQU1DQTBnQU1FVUNJRllCWkxneTBtN1czcTFmd0xNaG5jUndEUVgwV29RQzE5VDVCc2ZPSnpYMkFpRUFxQlYxS0\
4wbGNqTTVHZ2lXaHJQNFk2ZUhzQnBhSmppckNSMFFFcnFFRmpNd2dnS0NNSUlDS0tBREFnRUNBZ0VDTUFvR0NDcUdTTTQ5QkFNQ01IVXhDe\
kFKQmdOVkJBWVRBbGhZTVIwd0d3WURWUVFJREJSTmVTQlRkR0YwWlNCdmNpQlFjbTkyYVc1alpURVlNQllHQTFVRUNnd1BUWGtnVDNKblX\
NXBlbUYwYVc5dU1SQXdEZ1lEVlFRTERBZE5lU0JWYm1sME1Sc3dHUVlEVlFRRERCSnpZbWt2YzJWeWRtVnlMM0p2YjNRdFkyRXdJQmNOTWp\
FeE1qSTVNVFl6TkRBM1doZ1BPVGs1T1RFeU16RXlNelU1TlRsYU1Ic3hDekFKQmdOVkJBWVRBbGhZTVIwd0d3WURWUVFJREJSTmVTQlRkR0\
YwWlNCdmNpQlFjbTkyYVc1alpURVlNQllHQTFVRUNnd1BUWGtnVDNKblXNXBlbUYwYVc5dU1SQXdEZ1lEVlFRTERBZE5lU0JWYm1sME1TR\
XdId1lEVlFRRERCaHpZbWt2YzJWeWRtVnlMMmx1ZEdWeWJXVmthV0YwWlRFd1dUQVRCZ2NxaGtqT1BRSUJCZ2cbaGtqT1BRTUJCd05DQUFT\
WmpJMFNidnJOTU0xUy9PMGtnQlhzMWhpeWYxOVl3TFphRGtzMHlTaEduMlIra05NaTd0OWpzdDVxWXFJTHFJNXYzdHprMlF3K0l5R1NDQnV\
LbE9vV280R2dNSUdkTUIwR0ExVWREZ1FXQkJRZTRrMDRDdUxScnNrSm9lbTJMNHlTM0FqNVZqQWZCZ05WSFNNRUdEQVdnQlRTTExwdmcxRz\
hFT3luc1FMdlBhUm9QVjVqRGpBTUJnTlZlUk1FQlRBREFRSC9NQTRHQTFVZER3RUIvd1FFQXdJQkJqQTlCZ05WSFI4RU5qQTBNREEtnTUtBd\
WhpeG9kSFJ3T2k4dlkzSnNMbVY0WVcxd2JHVXVZMjl0UDJOaFBYTmlhVHB6WlhKMlpYSTZjbTl2ZEMxallUQUtCZ2dxaGtqT1BRUURBZ05J\
QURCRkFpRUFnaE9QNUtDWWZ6dzFPSEY0amJBeVhHUnRqT2JXbVVFM0VJSEIvT1hXYVV3Q0lCdnpOWjgwOEZJbVpJSUhaaHBLYlNYVTBHeXJ\
rbjlIOHJ1amZESnJoeWxsTVFBPSIKICAgICAgfQogICAgXQogIH0KfQ==="
  }
}
```


6.5 The 'get-bootstrapping-data' RPC returning "onboarding information":

6.5.1 XML-based Example

Request:

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

#!/bin/sh

# initialize the 'input' node for the RPC
cat << EOM > input.xml
<input xmlns="urn:ietf:params:xml:ns:yang:ietf-sztp-bootstrap-server">
    <hw-model>model-x</hw-model>
    <os-name>vendor-os</os-name>
    <os-version>17.3R2.1</os-version>
    <nonce>BASE64VALUE=</nonce>
</input>
EOM

# POST 'input' for the "get-bootstrapping-data" RPC resource
curl -i -X POST --data @input.xml -H "Content-Type:application/yang-data+xml" --cacert sbi_trust_chain.pem \
\--key pki/client/end-entity/private_key.pem --cert pki/client/end-entity/my_cert.pem --user my-serial-numb\
\er:my-secret https://127.0.0.1:9090/restconf/operations/ietf-sztp-bootstrap-server:get-bootstrapping-data \
\1> output/post_rpc_input_xml.out 2>/dev/null

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

# cleanup
rm -f input.xml
```

Response:

```
HTTP/1.1 404 Not Found
Content-Type: application/yang-data+xml; charset=utf-8
Content-Length: 359
Date: Mon, 20 Feb 2023 02:34:42 GMT
Server: <redacted>

<?xml version="1.0" ?>
<content-data xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-instance-data">
  <errors xmlns="urn:ietf:params:xml:ns:yang:ietf-restconf">
    <error>
      <error-type>application</error-type>
      <error-tag>data-missing</error-tag>
      <error-message>No responses configured.</error-message>
    </error>
  </errors>
</content-data>
```


6.5.2 JSON-based Example

Request:

```
================================ NOTE: '\' 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" : "BASE64VALUE="
  }
}
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.pe\
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 \
1> 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
```

Response:

```
HTTP/1.1 404 Not Found
Content-Type: application/yang-data+json; charset=utf-8
Content-Length: 198
Date: Mon, 20 Feb 2023 02:34:41 GMT
Server: <redacted>

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


6.6 The 'report-progress' RPC

6.6.1 XML-based Example

Request:

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

#!/bin/sh

# initialize the 'input' node for the RPC
cat << EOM > input.xml
<input xmlns="urn:ietf:params:xml:ns:yang:ietf-sztp-bootstrap-server">
  <progress-type>bootstrap-complete</progress-type>
  <message>example message</message>
  <ssh-host-keys>
    <ssh-host-key>
      <algorithm>ssh-rsa</algorithm>
      <key-data>BASE64VALUE=</key-data>
    </ssh-host-key>
    <ssh-host-key>
      <algorithm>rsa-sha2-256</algorithm>
      <key-data>BASE64VALUE=</key-data>
    </ssh-host-key>
  </ssh-host-keys>
  <trust-anchor-certs>
    <trust-anchor-cert>BASE64VALUE=</trust-anchor-cert>
  </trust-anchor-certs>
</input>
EOM

# POST it to the "report-progress" RPC resource
curl -i -X POST --data @input.xml -H "Content-Type:application/yang-data+xml" --cacert sbi_trust_chain.pem \
--key pki/client/end-entity/private_key.pem --cert pki/client/end-entity/my_cert.pem --user my-serial-numbe\
r:my-secret https://127.0.0.1:9090/restconf/operations/ietf-sztp-bootstrap-server:report-progress 1> output\
/post_progress_report_xml.out 2>/dev/null

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

# cleanup
rm -f input.xml
```

Response:

```
HTTP/1.1 204 No Content
Date: Mon, 20 Feb 2023 02:34:43 GMT
Server: <redacted>
```


6.6.2 JSON-based Example

Request:

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

#!/bin/sh

# initialize the 'input' node for the RPC
  cat << EOM > bootstrap-complete.json
{
  "ietf-sztp-bootstrap-server:input" : {
    "progress-type" : "bootstrap-complete",
    "message" : "Dynamically generated SSH host key included.",
    "ssh-host-keys" : {
      "ssh-host-key" : [
        {
          "algorithm" : "ssh-rsa",
          "key-data" : "BASE64VALUE="
        }
      ]
    }
  }
}
EOM

# POST it to the "report-progress" RPC resource
curl -i -X POST --data @bootstrap-complete.json -H "Content-Type:application/yang-data+json" --cacert sbi_t\
rust_chain.pem --key pki/client/end-entity/private_key.pem --cert pki/client/end-entity/my_cert.pem --user \
my-serial-number:my-secret https://127.0.0.1:9090/restconf/operations/ietf-sztp-bootstrap-server:report-pro\
gress 1> output/post_progress_report.out 2>/dev/null

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

# cleanup
rm -f bootstrap-complete.json
```

Response:

```
HTTP/1.1 204 No Content
Date: Mon, 20 Feb 2023 02:34:42 GMT
Server: <redacted>
```

6.7 Errors

Errors (e.g., HTTP status code 4XX) are also returned using the desired encoding.

6.7.1 XML-based Example

Request:

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

#!/bin/sh

# initialize the 'input' node for the RPC
cat << EOM > input.xml
<input xmlns="urn:ietf:params:xml:ns:yang:ietf-sztp-bootstrap-server">
  <foo-bar>bootstrap-initiated</foo-bar>
</input>
EOM

# POST it to the "report-progress" RPC resource
curl -i -X POST --data @input.xml -H "Content-Type:application/yang-data+xml" --cacert sbi_trust_chain.pem \
--key pki/client/end-entity/private_key.pem --cert pki/client/end-entity/my_cert.pem --user my-serial-numbe\
r:my-secret https://127.0.0.1:9090/restconf/operations/ietf-sztp-bootstrap-server:report-progress 1> output\
/post_progress_report_error_xml.out 2>/dev/null

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

# cleanup
rm -f input.xml
```


Response:

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

HTTP/1.1 400 Bad Request
Content-Type: application/yang-data+xml; charset=utf-8
Content-Length: 417
Date: Mon, 20 Feb 2023 02:34:43 GMT
Server: <redacted>

<?xml version="1.0" ?>
<content-data xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-instance-data">
  <errors xmlns="urn:ietf:params:xml:ns:yang:ietf-restconf">
    <error>
      <error-type>protocol</error-type>
      <error-tag>malformed-message</error-tag>
      <error-message>Unable to parse &amp;quot;input&amp;quot; document: Doesn't match schema: /input/foo-bar</erro\
r-message>
    </error>
  </errors>
</content-data>
```

6.7.2 JSON-based Example

Request:

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

#!/bin/sh

# initialize the 'input' node for the RPC
  cat << EOM > input.json
{
  "ietf-sztp-bootstrap-server:input" : {
    "foo-bar" : "bootstrap-initiated"
  }
}
EOM

# POST it to the "report-progress" RPC resource
curl -i -X POST --data @input.json -H "Content-Type:application/yang-data+json" --cacert sbi_trust_chain.pe\
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:report-progress 1> outp\
ut/post_progress_report_error.out 2>/dev/null

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

# cleanup
rm -f input.json
```

Response:

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

HTTP/1.1 400 Bad Request
Content-Type: application/yang-data+json; charset=utf-8
Content-Length: 275
Date: Mon, 20 Feb 2023 02:34:43 GMT
Server: <redacted>

{
  "ietf-restconf:errors": {
    "error": [
      {
        "error-type": "protocol",
        "error-tag": "malformed-message",
        "error-message": "Unable to parse \"input\" document: Doesn't match schema: /ietf-sztp-bootstrap-se\
rver:input/foo-bar"
      }
    ]
  }
}
```


7 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.

7.1 Inbound Facing

7.1.1 TLS Terminator

7.1.1.1 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.

7.2 Outbound Facing

7.2.1 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 shroudable by a TLS session (recommended).

7.2.2 Dynamic Callouts

Dynamic callouts may be configured to reach out to external systems to either push information to or fetch information from them.

Dynamic callouts are currently only implemented via plugins (see Plugins) loaded into SZTPD, enabling the plugin to implement a protocol of its choosing, but it is planned to also enable to SZTPD to also execute via RESTCONF (i.e., an HTTP "POST" request).

In either case, dyanmic callouts are modeled using YANG 'rpc' statements.

When executed via a plugin, the Python function takes two parameters. The first parameter is 'input', a Python object having a structure matching the 'rpc' statement's 'input' node. The second parameter is 'opaque', a Python object having an arbitrary ('anydata', in YANG parlance) structure configured for the specific 'dynamic-callout'.

The following "rpc" statement is defined in the "wn-app-rpcs" YANG module:

```
rpcs
  +-- rpc relay-notification
```

The following "rpc" statements are defined in the "wn-sztpd-rpcs" YANG module:

```
rpcs
  +-- rpc relay-progress-report
  +-- rpc verify-device-ownership
  +-- rpc get-conveyed-information
```

Please see Glossary of Dynamic Callouts for more information about the various dynamic callouts.

7.3 Glossary of Dynamic Callouts

Following lists all of the dynamic callouts implemented by SZTPD (see Dynamic Callouts).

7.3.1 The "relay-notification" Dynamic Callout

A dynamic callout implementing the "relay-notification" RPC is used to relay notifications, as defined by the "notification" statements in the various YANG modules implemented by SZTPD. All of the notifications are presented in the Glossary of Notifications section.

```
  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": "2019-01-22T10:32:12Z",
        "num-times-modified": 3,
        "last-modified": "2019-03-20T09:51:44Z"
      }
    }
  }
}
```

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
```

To configure SZTPD to relay notifications, the "relay-notification-callout" node under the host-level and/or tenant-level[5] "preferences" container must point to a "dynamic-callout" having the "rpc-supported" value "app-rpcs:relay-notification".

[5] In the multi-tenant mode, when the notification is tenant-specific, SZTPD will only use the tenant-level dynamic callout, if configured.

7.3.2 The "relay-progress-report" Dynamic Callout

A dynamic callout implementing the "relay-progress-report" RPC is used to send progress-reports, as sent by bootstrapping devices in their "report-progress" message.

The tree diagram for the "relay-progress-report" RPC follows:

```
rpcs:
  +---x relay-progress-report {relay-progress-report}?
     +---- input
        +---w serial-number            string
        +---w source-ip-address        ietf-inet-types:ip-address
        +---w identity-certificate?    ietf-crypto-types:cms
        +---w from-device?             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",
    "identity-certificate": "BASE64VALUE=",
    "from-device": {
      "ietf-sztp-bootstrap-server:input" : {
        "progress-type": "bootstrap-complete",
        "message": "example message",
        "ssh-host-keys": {
          "ssh-host-key": [
            {
              "algorithm": "ssh-rsa",
              "key-data": "BASE64VALUE="
            },
            {
              "algorithm": "rsa-sha2-256",
              "key-data": "BASE64VALUE="
            }
          ]
        },
        "trust-anchor-certs": {
          "trust-anchor-cert": [ "BASE64VALUE=" ]
        }
      }
    }
  }
}
```

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
```

To configure SZTPD to relay progress reports, the "relay-progress-report-callout" node under the host-level and/or tenant-level[6] "preferences" container must point to a "dynamic-callout" having the "rpc-supported" value "sztpd-rpcs:relay-progress-report".

[6] In a multi-tenant mode, when the progress report comes from a tenant's device, SZTPD will only use the tenant-level dynamic callout, if configured.

7.3.3 The "verify-device-ownership" Dynamic Callout

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:

The tree diagram for the "verify-device-ownership" RPC 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"
        }
      ]
    }
  }
}
```

To configure SZTPD to verify device ownership, the "ownership-authorization" node under the host-level "device-types" must point to a "dynamic-callout" having the "rpc-supported" value "sztpd-rpcs:verify-device-ownership".

7.3.4 The "get-conveyed-information" Dynamic Callout

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

The tree diagram for the "get-conveyed-information" RPC follows:

```
rpcs:
  +---x get-conveyed-information {get-conveyed-information}?
     +---- input
     |  +---w serial-number           string
     |  +---w source-ip-address       ietf-inet-types:ip-address
     |  +---w identity-certificate?   ietf-crypto-types:cms
     |  +---w from-device?            anydata
     +---- 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",
    "identity-certificate": "BASE64VALUE=",
    "from-device": {
      "ietf-sztp-bootstrap-server:input" : {
        "hw-model": "model-x",
        "os-name": "vendor-os",
        "os-version": "17.3R2.1",
        "nonce": "BASE64VALUE=",
        "ietf-sztp-csr:csr": {
          "p10": "BASE64VALUE="
        }
      }
    }
  }
}
```

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": "BASE64VALUE="
  }
}
```

To configure SZTPD to dynamically obtain conveyed-information from an external system, a device's "response-manager" configuration must have a "matched-response" of type "conveyed-information" with a "dynamic-callout" having the "rpc-supported" value "sztpd-rpcs:get-conveyed-information".

7.4 Glossary of Notifications

These are the notifications that may be sent via the "relay-notification" dynamic callout (see The "relay-notification" Dynamic Callout).

7.4.1 Overview

The following "notification" statements are defined in the "wn-app" 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
```

7.4.2 The "unreferenced-node-lingering" Notification

The tree-diagram for this notification is as follows:

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

7.4.3 The "unreferenced-node-purged" Notification

The tree-diagram for this notification is as follows:

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

7.4.4 The "admin-account-activation-expired" Notification

The tree-diagram for this notification is as follows:

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


7.4.5 The "admin-account-password-aging" Notification

The tree-diagram for this notification is as follows:

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

7.4.6 The "admin-account-disabled" Notification

The tree-diagram for this notification is as follows:

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

7.4.7 The "admin-account-purged" Notification

The tree-diagram for this notification is as follows:

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

7.4.8 The "voucher-expiration" Notification

The tree-diagram for this notification is as follows:

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

7.4.9 The "unused-device-record-lingering" Notification

The tree-diagram for this notification is as follows:

```
+---n unused-device-record-lingering {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
```

7.4.10 The "unused-device-record-purged" Notification

The tree-diagram for this notification is as follows:

```
+---n unused-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
```

7.4.11 The "used-device-record-lingering" Notification

The tree-diagram for this notification is as follows:

```
+---n used-device-record-lingering {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
```


7.4.12 The "used-device-record-purged" Notification

The tree-diagram for this notification is as follows:

```
+---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
```

7.4.13 The "certificate-expiration" Notification

The tree-diagram for this notification is as follows:

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

7.4.14 The "yang-library-change" Notification

The tree-diagram for this notification is as follows:

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

7.4.15 The "yang-library-update" Notification

The tree-diagram for this notification is as follows:

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

7.4.16 The "yang-library-change" Notification

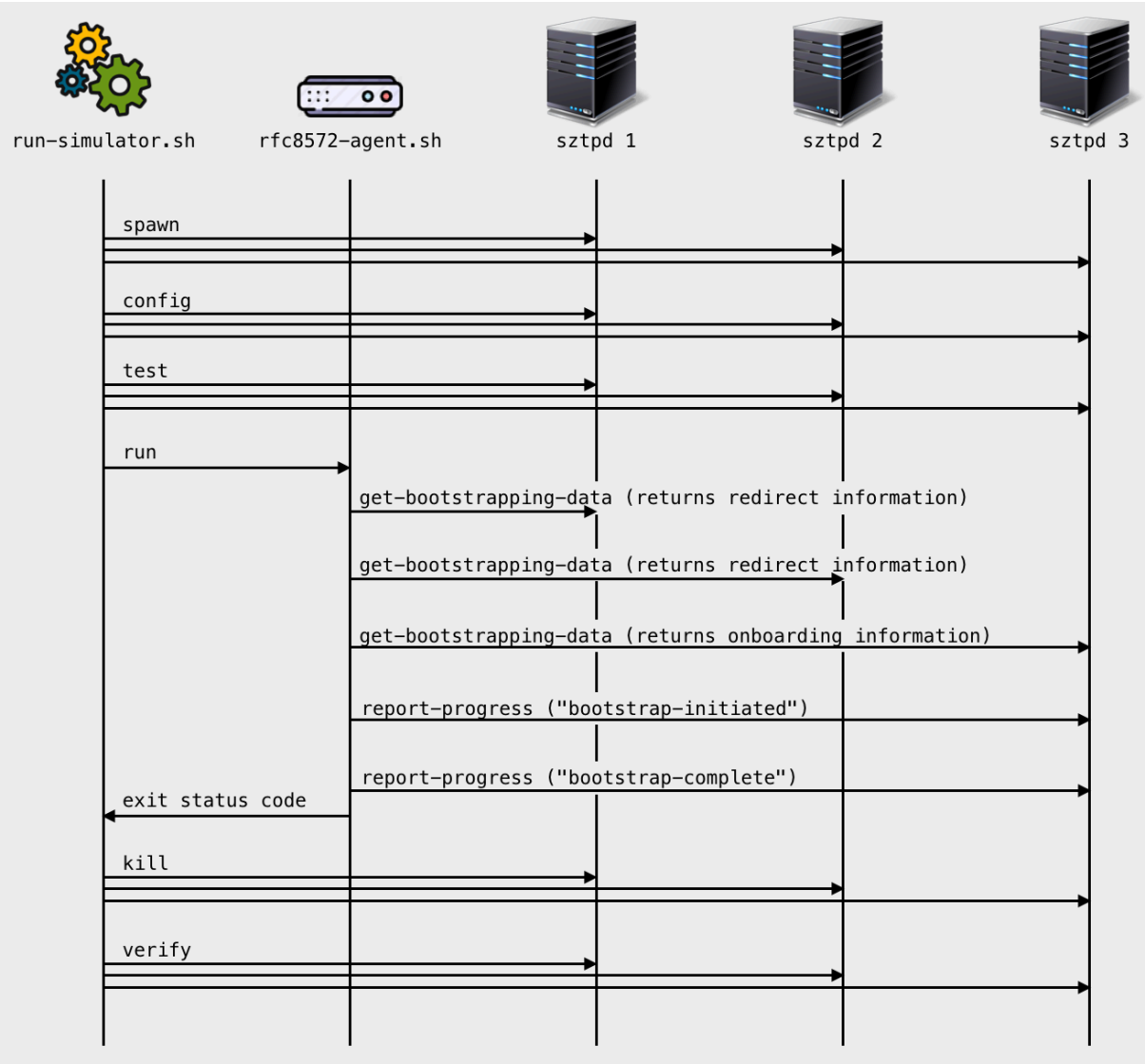
The tree-diagram for this notification is as follows:

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


8 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).

8.1 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 SZTP 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" database[7].

8.2 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

8.3 Downloading

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

8.4 Unarchiving

Unarchive the TGZ in a local directory, for example:

```
$ tar -xzvf ./sztpd-simulator-0.0.8.tgz

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

[7] For 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.

8.5 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.

8.6 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.

8.7 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!
```

8.8 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
```

