SIEMENS
Ingenuity for life

Security for IEC/IEEE 60802

NETCONF Security Deep-Dive

K. Fischer, A. Furch, L. Lindemann, O. Pfaff, T. Pössler, G. Steindl

Siemens AG 2021

Problem Statement

- Provide a **deep-dive** for **NETCONF security** (*as-is*) from the perspective of industrial automation esp. IA devices/controllers
- Report the **fitness** of NETCONF security for **industrial automation**
- Use **specification documents** for this analysis (implementations are not considered herein)
- See the accompanying overview slide-deck for the **abstractions/terms** etc. considered herein

- **Note**: deep-dives (according the same scheme) will be made for all short-listed candidates

Fitness of *As-Is* NETCONF Security for Industrial Automation

Security fulfilment disciplines*	Message exchange protection	Resource access authorization
Protect shared resources on IA devices/controllers	*Assessment*: covered (NETCONF-over-TLS or SSH) but has many options and is not yet profiled for industrial automation *Action item***: **profiling** for IEC/IEEE 60802	*Assessment*: addressed with respect to DAC (NACM) but not yet incarnated for industrial automation *Action item***: **profiling** for IEC/IEEE 60802
Establish security associations with endpoints on IA devices/controllers		
Manage initial credentials and overall security configuration at IA devices/controllers	*Assessment*: addressed (SZTP) but comes with many specifics and has white spots *Action item****: **profiling and/or specifying** for IEC/IEEE 60802	*Assessment*: NACM comes with a *chicken-and-egg* problem which is not elaborated in NACM RFCs *Action item****: **profiling and/or specifying** for IEC/IEEE 60802

*: see background slide for details

**: can be started without waiting for other deep-dive results

*** should wait for other deep-dive results

Profiling Action Items Include

SIEMENS
Ingenuity for life

- **Security for shared resources**:
  - Message exchange protection:
    - Select TLS and/or SSH
    - Profile scheme-specific details e.g. version of security protocols, handling of optional features…
  - Resource access authorization (NACM - if DAC is the preferred model):
    - Model authorization-controlled resources and actions
    - Assign NETCONF ‘users’ to groups
- **Shared security means**: compile a catalogue of cryptographic algorithms
- **Securing-the-security**:
  - Select SZTP with and/or without ‘call home’ feature (RFC 8071, RFC 8366)
  - Profile SZTP-specific sources and details of bootstrapping data e.g. sources of bootstrapping data, nonceless vouchers, revocation means
  - Select supported ‘user’ population: implicit (mapping from TLS/SSH), local and/or remote repositories

Action Items Possibly Beyond Profiling Include

- **Security for shared resources**:
  - Message exchange protection: n.a.
  - Resource access authorization: reconfirm authorization model DAC vs. MAC/ABAC/RBAC…
- **Shared security means**: n.a.
- **Securing-the-security**:
  - Supply of own (private keys and) EE certificates to NETCONF servers
  - SZTP bootstrapping/credentialing of network components without any initial credentials
  - Supply credentials/trust anchors to NETCONF clients
  - Push support for credential/trust anchor management
  - Elaborate the assignment/management/identification of the NACM root-of-authority
  - Cover equipment originality checks
  - Enforce overall security configuration, e.g. allow only protected access

NETCONF Security Mind-Map

SIEMENS

Ingenuity for life

```
Security for [NETCONF](https://tools.ietf.org/html/rfc6241)
# Security
- Communication pattern: client/server, over reliable transport (TCP)
- Components model: client, server
- Client incarnations: application-level component, usually part of a 'network manager' application, possibly operated by human user
- Server incarnations: application-level component, usually part of a 'network device'
- Note: client and server incarnations may be co-located in a system component
- Security: [mandatory i.e. there is no 'NETCONF-plain'](https://tools.ietf.org/html/rfc6241#section-2.2)

## Message Exchange Protection
- E2E span: NETCONF client-to-server
- Keying ownership: deployment-specific keys/credentials are needed
- Placement: transport-level security
- Packaging: encapsulating NETCONF APDUs
### [TLS](https://tools.ietf.org/html/rfc7589)
- Version: [TLS 1.2](https://tools.ietf.org/html/rfc5246)
- Hygiene: [Secure Use Recommendations](https://tools.ietf.org/html/rfc7525)
- Casting: NETCONF client=TLS client, NETCONF server=TLS server
- Cipher suites: must support [TLS_RSA_WITH_AES_128_CBC_SHA](https://tools.ietf.org/html/rfc5246#section-9)
- TLS extensions: not explicitly addressed, implicitly addressed via [Secure Use Recommendations](https://tools.ietf.org/html/rfc7525)
#### Entity Authentication
- Mode: mutual authentication
- Means: asymmetric (key pairs) schemes
- Key&entity binding: 3rd-party objects certifying the binding of a public key and entity identification
- Form factor: ASN.1, [X.509 public key certificates](https://tools.ietf.org/html/rfc6187)
- Certificate validation: path validation  or explicit acceptance of EE certificates
- Certification path validation:  [RFC 5280](https://tools.ietf.org/html/rfc5280#section-6)
- Server identification: DNS name or IP address in subjectAltName in EE certificate
- Server identity check: client-side TLS engine matches ["actual vs. expected"](https://tools.ietf.org/html/rfc7589#section-6)
- Client identification: arbitrary naming info in EE certificate, focusing but not mandating rfc822Name, dNSName, iPAddress as structural elements
- Client identity check: server-side application checks ["actual vs. expected"](https://tools.ietf.org/html/rfc7589#section-7) based on a list of known NETCONF 'usernames'
#### Integrity
- Means: symmetric, keyed message authentication codes
- Mode: [HMAC](https://tools.ietf.org/html/rfc2104)
#### Confidentiality
- Means: symmetric, stream or block ciphers
- Modes: [AEAD](https://tools.ietf.org/html/rfc5116) or classical
#### Replay Protection
- Replay, mis-ordering is aggressively defeated, no window of acceptance
#### Non-Repudiation
- Not elaborated
### [SSH](https://tools.ietf.org/html/rfc6242)
- Version: SSH-2
- Hygiene: n.a. (no IETF equivalent  to RFC 7525)
- Casting: NETCONF client=SSH client, NETCONF server=SSH server
- Level  of demand: [mandatory to implement](https://tools.ietf.org/html/rfc6241#section-2.3) but not mandatory to use
#### Entity Authentication
- Mode: mutual authentication
- Means: asymmetric (key pairs) or symmetric (passwords) schemes
- Key&entity binding: 3rd-party objects certifying the binding of a public key and entity identification or raw public key/password with verifier-local mapping tables to system entities
- Form factor: ASN.1, [X.509 public key certificates](https://tools.ietf.org/html/rfc5280#section-4) or raw public key or plaintext password
- Certificate validation: [path validation](https://tools.ietf.org/html/rfc6187#section-2.1) or dedicated checks against verifier-local mapping table
- Certification path validation:  [RFC 5280](https://tools.ietf.org/html/rfc5280#section-6)
- Server identification: [DNS name or IP address in subjectAltName](https://tools.ietf.org/html/rfc6187#section-4) in EE certificate or verifier-local mapping table
- Server identity check: client-side SSH engine matches ["actual vs. expected"](https://tools.ietf.org/html/rfc6187#section-4)
- Client identification: username
- Client identity check: server-side application checks ["actual vs. expected"](https://tools.ietf.org/html/rfc7589#section-7) based on a list of known NETCONF 'usernames'
#### Integrity
- Means: symmetric, keyed message authentication codes
- Mode: [HMAC](https://tools.ietf.org/html/rfc2104)
#### Confidentiality
- Means: symmetric, stream or block ciphers
- Modes: [AEAD](https://tools.ietf.org/html/rfc5116) or classical
#### Replay Protection
- Replay, mis-ordering is aggressively defeated, no window of acceptance
#### Non-Repudiation
- Not elaborated

## Resource Access Authorization
- Access to NETCONF server resources shall be limited to [dedicated 'users'](https://tools.ietf.org/html/rfc6241#section-9)
### [NACM](https://tools.ietf.org/html/rfc8341)
- Objective: resource access authorization for NETCONF and RESTCONF
- Pattern: NACM protects resources on NETCONF servers, NACM artifacts are managed through NETCONF
- Implication: this pattern introduces a 'chicken-and-egg' problem
#### Decision Enforcement
- Responsible entity: [NETCONF servers](https://tools.ietf.org/html/rfc8341#section-3.4)
#### Decision Making
- Responsible entity: NETCONF servers
- Information base: decision making instructions in local configuration data
- Expression: decision making instructions are expressed according [YANG/XML schema for NACM](https://tools.ietf.org/html/rfc8341#section-3.5.2)
#### Policy Making
- Responsible entity: NETCONF clients
- Management: supply of NACM artifacts happens by means of [NACM-via-NETCONF](https://tools.ietf.org/html/rfc8341#section-5.1))
- Manageable items: 'user' assignments to groups, resource/action assignments to groups
#### Authorization Strategy
- Model: Discretionary Access Control (DAC)
#### Access Control Matrix Axes/Fields
- Subjects: individual 'users', clustered into configurable groups
- Resources: incoming RPC messages, nodes in YANG data, outgoing notifications
- Actions: CRUDX - Create/Read/Update/Delete/eXecute
### [XACML](https://tools.ietf.org/html/draft-seitz-netconf-xacml-02)
- Private Internet Draft, abandoned
### [RBAC](https://tools.ietf.org/html/draft-cridlig-netconf-rbac-00)
- Private Internet Draft, abandoned

# Shared Security Means
- Not addressed

# Securing-the-Security
## Credential Management
- TLS: NETCONF clients and servers must be equipped with EE certificate/private key and CA certificate(s) aka trust anchors
- SSH: NETCONF clients and servers must be equipped with EE certificate/private key and CA certificate(s) aka trust anchors or raw public keys/passwords
### [SZTP](https://tools.ietf.org/html/rfc8572)
- Addresses the credentialing of 'network devices' i.e. NETCONF (or RESTCONF) servers
#### Functionality
- Prerequisite: a 'network device' has [initial credential/trust anchor info](https://tools.ietf.org/html/rfc8572#section-5.1)
- Event: the 'network device' is booting in factory-default state
- Task: provision information to this 'network device' during its boot from factory-default
- Forms of bootstrapping data: represented in YANG, expressed in JSON or XML
- Types of bootstrapping data: 'redirect' or 'onboarding' information, owner certificate, ownership voucher
- Redirect information: [host/port and trust anchor of another SZTP server](https://tools.ietf.org/html/rfc8572#section-2.1)
- Onboarding information: [boot image, configuration, scripting](https://tools.ietf.org/html/rfc8572#section-2.2)
- Owner certificate: 3rd party EE certification path, supersigned in a way that can be validated with the device's initial credential(s)
- Ownership voucher: owner info, local trust anchor, supersigned in a way that can be validated with the device's initial credential(s)
- Important: the own EE certificate (and private key) is not provisioned by SZTP
- Sources of bootstrapping data: SZTP bootstrapping server, DNS/DHCP server, removable  storage
- SZTP bootstrapping server protocol: HTTP-over-TLS according RESTCONF
- Supply model: pull i.e. a to-be-provisioned 'network device' (usually a NETCONF server) acts as client of a SZTP bootstrap or DNS/DHCP server
#### Security
- Strategy: deprotect_with_initial_or_current (plain vanilla) or deprotect_with_subsequent (an indirection trick, uses voucher objects)
##### Message Exchange Protection
- The 'bootstrapping data' sent to the device is to-be-protected (may also be plain in some case)
- Protection is supposed to happen on application object-level: 'bootstrapping data' is signed or first-signed-then-encrypted
- The protected 'bootstrapping data' is expressed in ASN.1, [CMS](https://tools.ietf.org/html/rfc5652). Note: this provokes media-breaks
- The protection must happen in a way so that the to-be-provisioned 'network device' can verify or decrypt-and-verify
- [To do this the 'network device' is assumed to have an initial equipment with credentials](https://tools.ietf.org/html/rfc8572#section-5.1)
- This may be an IDevID or LDevID credentials
- The use of [IDevID credentials](https://tools.ietf.org/html/rfc8572#section-9.2) is recommended
- The organization that equips network devices with initial credentials has to supply a 'call-home' infrastructure capable of supplying (device instance-specific) protected bootstrapping data upon request
- Alternative: coin such information in advance  and distribute it for supply via DNS/DHCP server or removable  storage
##### Resource Access Authorization
- No granularity (accept all - if verification with IDevID was ok)

## System Configuration
### Credential Metadata
- [YANG Data Model for a Truststore](https://tools.ietf.org/html/draft-ietf-netconf-trust-anchors-14)
- [YANG Data Model for a Keystore](https://tools.ietf.org/html/draft-ietf-netconf-keystore-20)
### ['Users'](https://tools.ietf.org/html/rfc7317)
- Representation: arbitrary strings e.g. "brmpflzipf" representing an authenticated entity (caller)
- Mapping options: i. from client identification in TLS/SSH, ii. against remote repo's, iii. against local info
#### Functionality
- Mapping from TLS/SSH: no explicit CRUD operations for 'user' objects (happens implicitly)
- Mapping against remote repos: CRUD is subject to remote systems e.g. RADIUS
- Mapping against local info: skipped for this deep-dive
#### Security
- Mapping from TLS/SSH: subject to client credential management practices
- Mapping against remote repos: subject to remote systems e.g. RADIUS
- Mapping against local info: skipped for this deep-dive
### Authorization Data
- Strategy: authorization management and authorization controlled operations use the same (!) channel
#### Functionality
- See NACM section above
#### Security
- Protecting NACM exchange: NETCONF-over-TLS or SSH (i.e. eat-the-own-dogfood)
- Authorizing NACM management operations: root-of-authority is allocated with ['recovery sessions'](https://tools.ietf.org/html/rfc8341#section-1.1)
- 'Recovery  session' privileges: allow-all  ("exempt from all access control enforcement")
- 'Recovery  session' identification: not elaborated ("are implementation specific and are outside the scope of this document")
- Delegation: root-of-authority may delegate the privilege  to manage NACM artifacts
### Others
#### IA Device/Controller-Global  Configuration e.g. 'PROTECTED-ONLY'
- Not elaborated
```

- Copy the markdown source from the grey text field on the left (don’t worry about the tiny font size)
- Paste this text into an interpreter e.g. https://markmap.js.org/repl
- Adjust the page zoom and browse the shown mind-map
- This map provides the NETCONF security essentials

Siemens AG 2021

Next Steps

1. Kicking-off - Done
2. Establish goals and constraints, agree on use cases (automation and security-specific)
3. Perform deep-dives for the security technology candidates
   i. NETCONF security – Largely done
   ii. SNMP security
   iii. DNS security
   iv. 802.1AE/X/AR
   v. 802.1AS security
   vi. NN, decide about items from the longlist
4. Identify cross-relation/common interests with middleware/application-specific security
   - Shortlist: security for IEC 61158 technologies, OPC-UA security, Web security…
5. Create the blueprint of an overarching security architecture (more details are tbd)

Abbreviations*

ABAC	Attribute-Based Access Control
DASA	Delegated Authorized Signing Authority
MAC	Mandatory Access Control
MASA	Manufacturer Authorized Signing Authority
NACM	NETCONF Access Control Model
RBAC	Role-Based Access Control
SZTP	Secure Zero Touch Provisioning
XACML	eXtensible Access Control Markup Language

*: see the accompanying overview slide-deck for further abbreviations

References, Chronologically Ordered

SIEMENS
Ingenuity for life

1. IETF RFC 4741: Network Configuration Protocol, 2006
2. IETF RFC 4742: Using the NETCONF Protocol over Secure Shell (SSH), 2006
3. IETF RFC 5539: NETCONF over Transport Layer Security (TLS), 2009
4. IETF RFC 6187: X.509v3 Certificates for Secure Shell Authentication, 2011
5. IETF RFC 6241: Network Configuration Protocol (NETCONF), 2011
6. IETF RFC 6242: Using the NETCONF Protocol over Secure Shell (SSH), 2011
7. IETF RFC 6536: Network Configuration Protocol (NETCONF) Access Control Model, 2012
8. IETF RFC 7589: Using the NETCONF Protocol over Transport Layer Security (TLS) with Mutual X.509 Authentication, 2015
9. IETF RFC 8071: NETCONF Call Home and RESTCONF Call Home, 2017
10. IETF RFC 8341: Network Configuration Access Control Model, 2018
11. IETF RFC 8366: A Voucher Artifact for Bootstrapping Protocols, 2018
12. IETF RFC 8572: Secure Zero Touch Provisioning (SZTP), 2019

Authors

SIEMENS
Ingenuity for life

Kai Fischer, Siemens AG, T RDA CST SES-DE, kai.fischer@siemens.com

Andreas Furch, Siemens AG, T RDA CST SES-DE, andreas.furch@siemens.com

Lars Lindemann, Siemens AG, DI FA CTR ICO ARC, lars.Lindemann@siemens.com

Oliver Pfaff, Siemens AG, T RDA CST, oliver.pfaff@siemens.com

Thomas Pössler, Siemens AG, RC-AT DI FA DH-GRAZ SAS, thomas.poessler@siemens.com

Günter Steindl, Siemens AG, DI FA TI ART EA, guenter.steindl@siemens.com

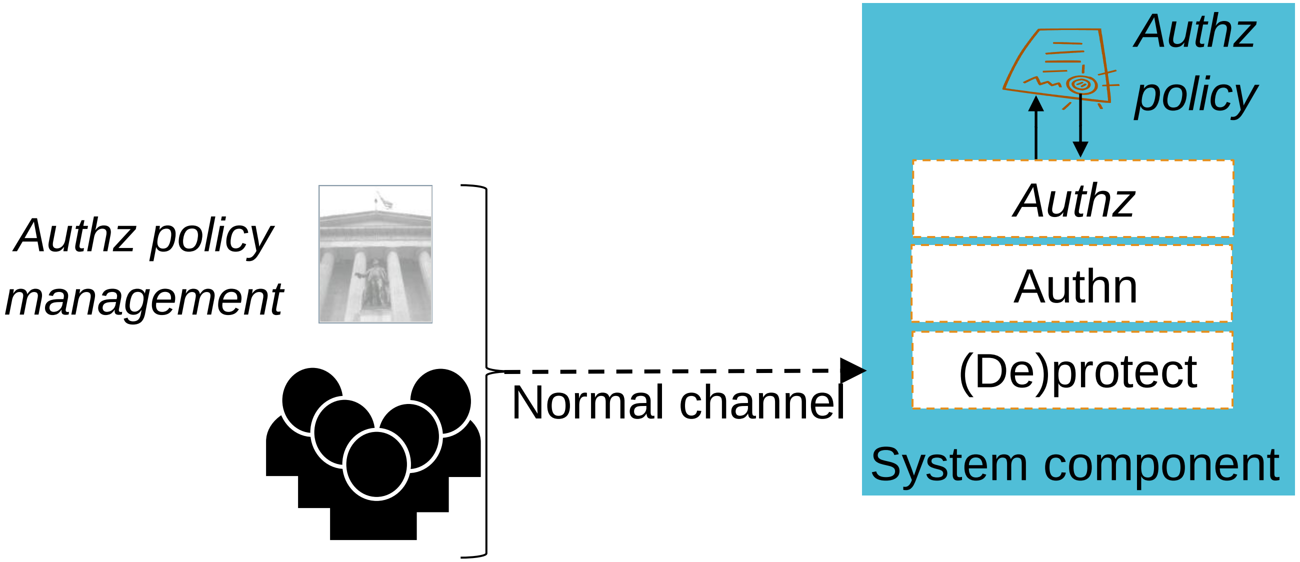
Security Fulfilment Disciplines Explained

SIEMENS
Ingenuity for life

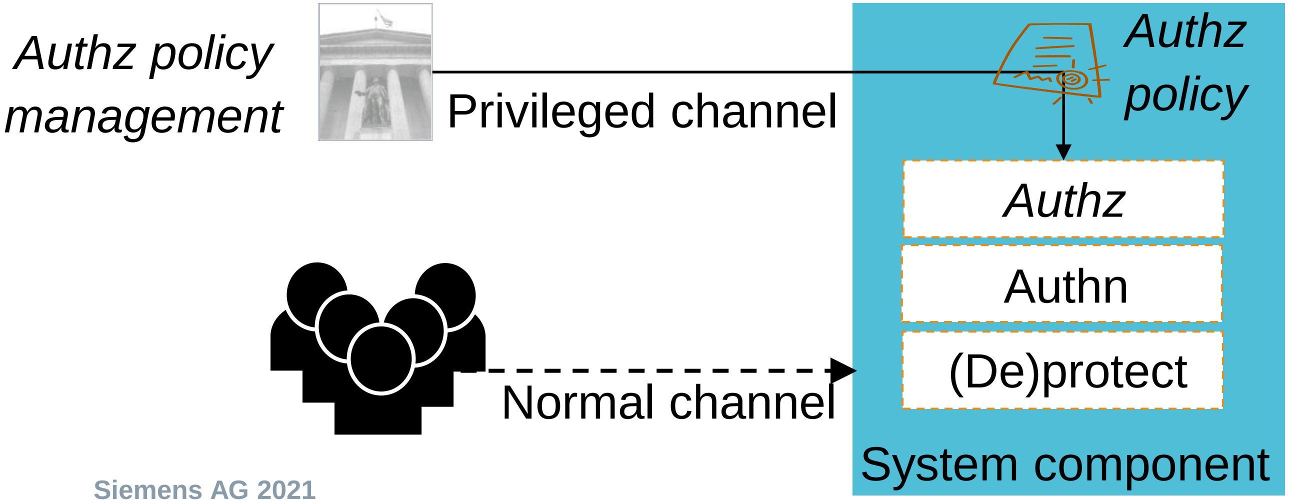
Security fulfilment disciplines	Meaning	Example for Web security*
Protect shared resources on IA devices/controllers	Exercise **message exchange protection** and **resource access authorization** for shared resources on IA devices/controllers	*Message exchange protection*: send HTTP requests/responses with TLS record layer protection *Resource access authorization*: enforce write/read access control to specific folders (paths) etc.
Establish security associations with endpoints on IA devices/controllers	Establish **(authenticated) keys** and further security settings **between communicating partners**	Prepare the TLS record layer(s) for operation by doing a TLS handshake
Manage initial credentials and overall security configuration at IA devices/controllers	Supply **(initial) credential/trust anchor(s)** to a **dedicate entity**	Prepare the TLS handshake layer(s) for operation by supplying credentials, trust anchors and other security configuration e.g. cipher suite preferences

*: not actually part of the shared resources but used for illustration - as Web security is familiar to all

Authorization Management Pattern: NACM



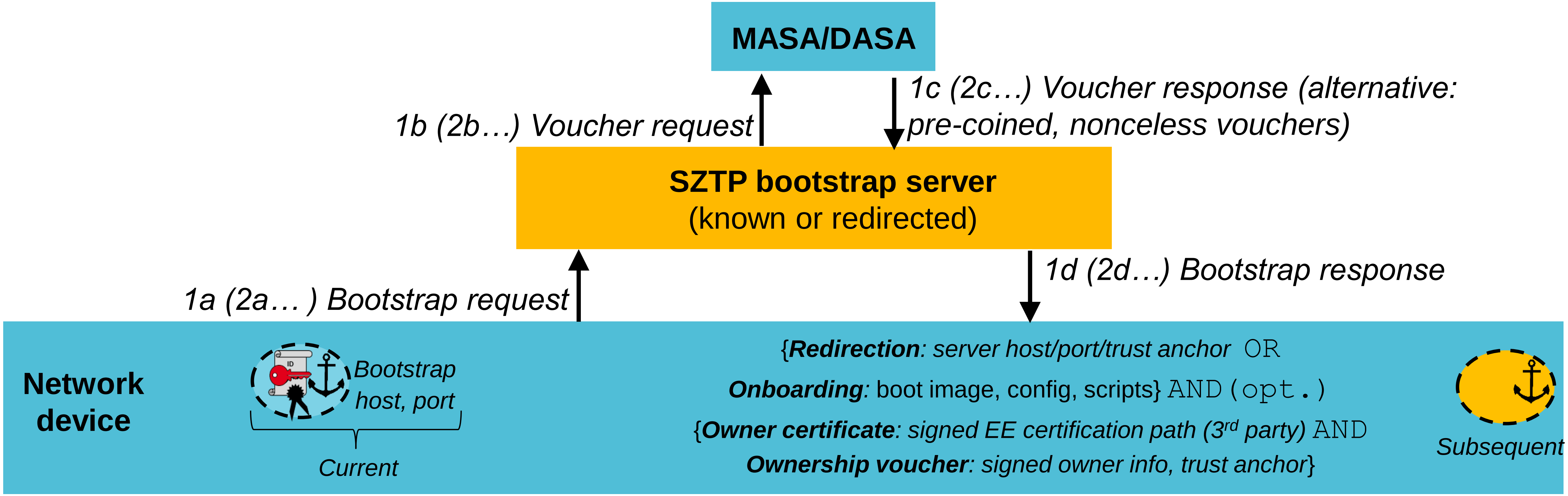
- *NACM pattern*: authorization management and authorization controlled operations **use** the **same channel**



- *Default pattern in IT*: authorization management and authorization controlled operations **use different channels**

Bootstrapping Pattern: SZTP

- *1 main event:* booting in **factory-default state**
- *2 main actors*: **network device, SZTP bootstrap server** (alternatives: removable storage, DNS/DHCP)
- *2 main security strategies*: **deprotect_with_current** or **_subsequent** (an indirection ➔ uses vouchers)
- *4 main supplies*: {**redirection** or **onboarding**} and opt. {**owner certificate** and **ownership voucher**}



MASA/DASA

1b (2b…) Voucher request

1c (2c…) Voucher response (alternative: pre-coined, nonceless vouchers)

SZTP bootstrap server
(known or redirected)

1a (2a…) Bootstrap request

1d (2d…) Bootstrap response

Network device

Bootstrap host, port

Current

{***Redirection***: *server host/port/trust anchor* OR
Onboarding: boot image, config, scripts} AND (opt.)
{***Owner certificate***: *signed EE certification path (3rd party)* AND
Ownership voucher: *signed owner info, trust anchor*}

Subsequent