OPEN CONNECTIVITY FOUNDATION®

OCF Security Primer for Device Vendors

June 21st, 2018

Nathan Heldt-Sheller

The Primary Roles of the OCF Security Layer

From a "normal" (non-security-expert) developer's perspective, the OCF Security Layer has two primary functions:

1. **Device Onboarding** – "pairing" a New Device (e.g. Out of Box) to the owner's domain. An example would be connecting a newly-installed lightbulb (a Server Device) to the home's OCF lighting control (a Client Device application).

2. **Access Control** – filtering incoming requests to a Device, and making a "Granted" or "Denied" decision for each request before it is passed to the Resource endpoint.
   Note: for the security-minded, this includes authentication in addition to authorization.

Both of these functions are achieved via a set of "Security Virtual Resources" (SVRs), which provide the interface through which Device Onboarding is accomplished, and Access Control is configured.

Introducing the Security Virtual Resources

Although there are quite a few SVRs defined in the OCF Security Specification, there are just **four that are essential to understanding basic security function**:

1. **/oic/sec/doxm (or "/doxm") – The Device Owner Transfer (Xfer) Method Resource**
   The /doxm Resource provides the interface for taking ownership of a Device. Usually, taking ownership is the first step in onboarding a new Device into the owner's domain.

2. **/oic/sec/pstat (or "/pstat") – The Provisioning Status Resource**
   The /pstat Resource is used to manage further provisioning of a Device, after ownership is established. Specifically, the /pstat Resource is used to put the Device in "Ready for Normal Operation" (RFNOP) state, which signals that the Device is fully configured and ready to start its normal steady-state functioning (e.g. a lightbulb in "RFNOP" is ready to handle "on/off/dim" requests from the Lighting Controller).

3. **/oic/sec/acl2 (or "/acl2") – the Access Control List Resource**
   The /acl2 Resource is used to configure the access control policy on the Device (i.e. which Clients are allowed to access which Resources, and what the access-modes – Retreive vs. Update, etc – are allowed).

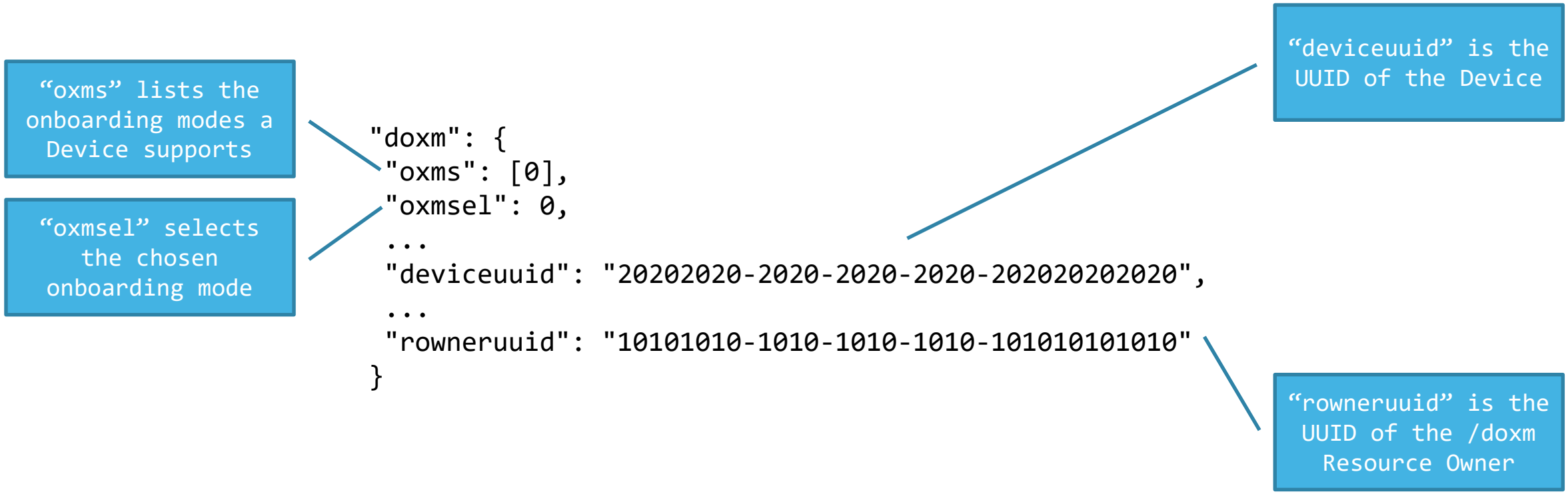
4. **/oic/sec/cred (or "/cred") – The Credentials Resource**
   The /cred Resource stores the credentials – cryptographic keys, certificates, etc. – that are required to establish secure connections, and verify Client identity, among other things.

Each of these Resources has a "Resource Owner", which is a Client Device that has automatically-Granted Retrieve/Update access to the Resource, identified by its "UUID" (Universally Unique Identifier).

Understanding /doxm

/oic/sec/doxm (or "/doxm") – The Device Owner Transfer (Xfer) Method Resource

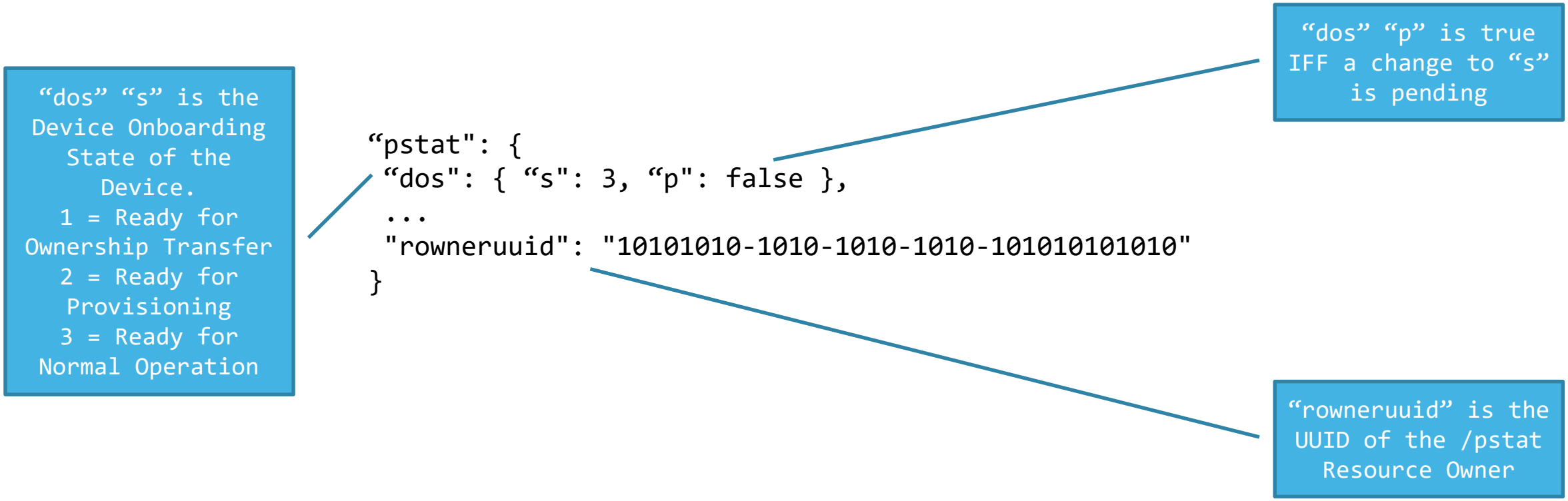
The /doxm Resource provides the interface for taking ownership of a Device. Usually, taking ownership is the first step in onboarding a new Device into the owner's domain.



Understanding /pstat

/oic/sec/pstat (or "/pstat") – The Provisioning Status Resource

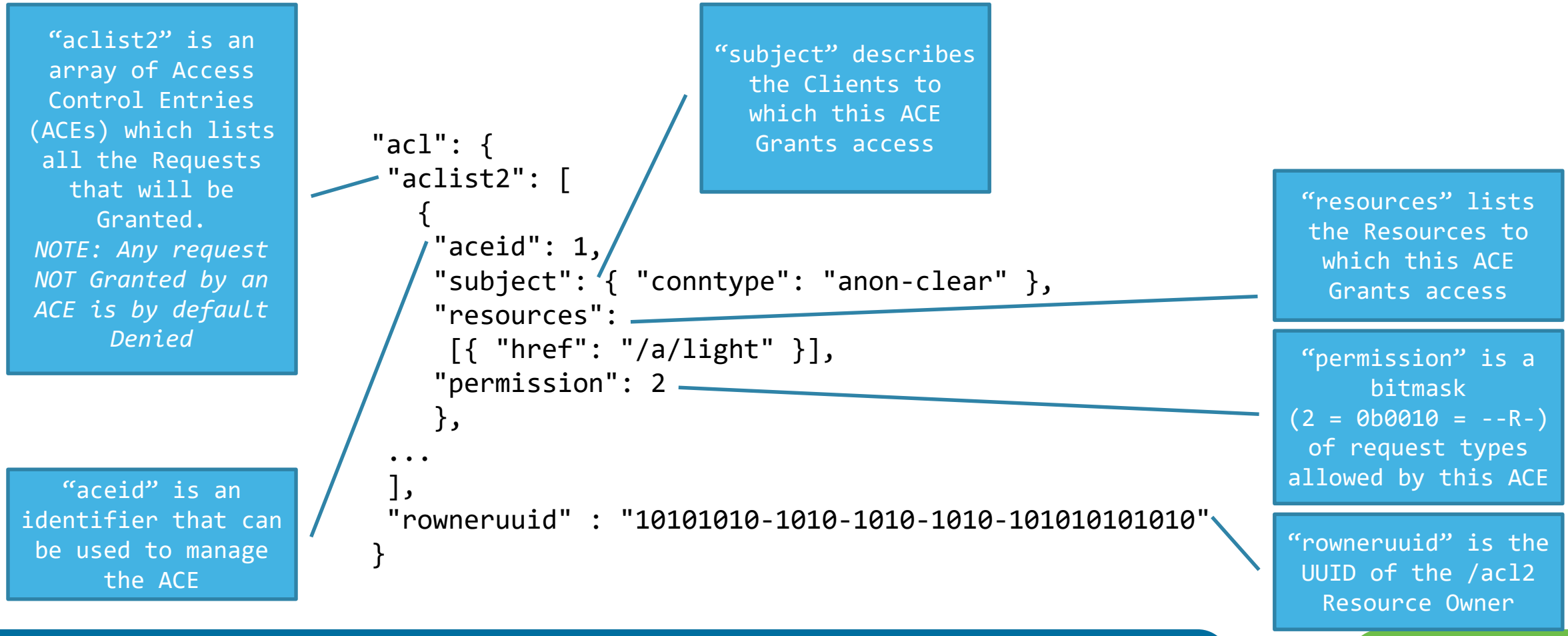
The /pstat Resource is used to manage further provisioning of a Device, after ownership is established.



Understanding /acl2

/oic/sec/acl2 (or "/acl2") – the Access Control List Resource

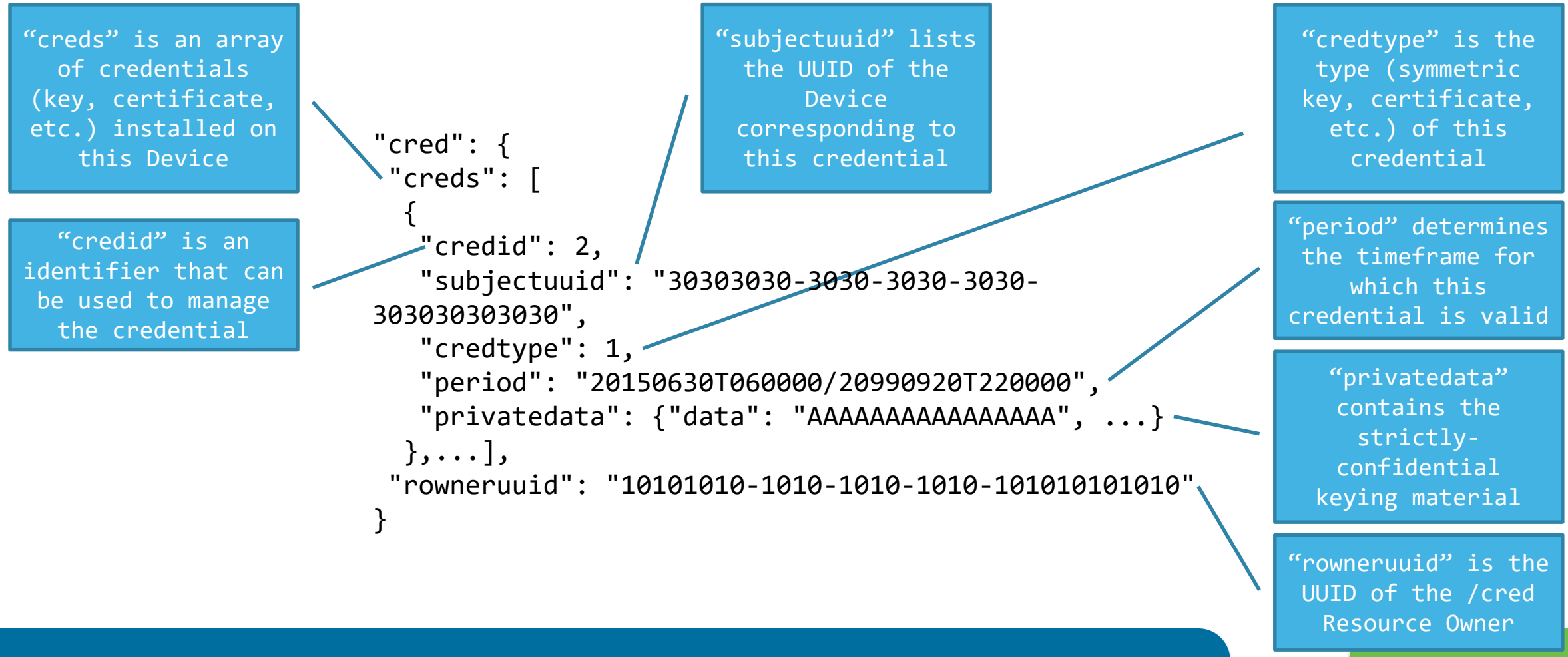
The /acl2 Resource is used to configure the access control policy on the Device (i.e. which Clients are allowed to access which Resources, and what the allowed access-modes – Retreive vs. Update, etc – are allowed.



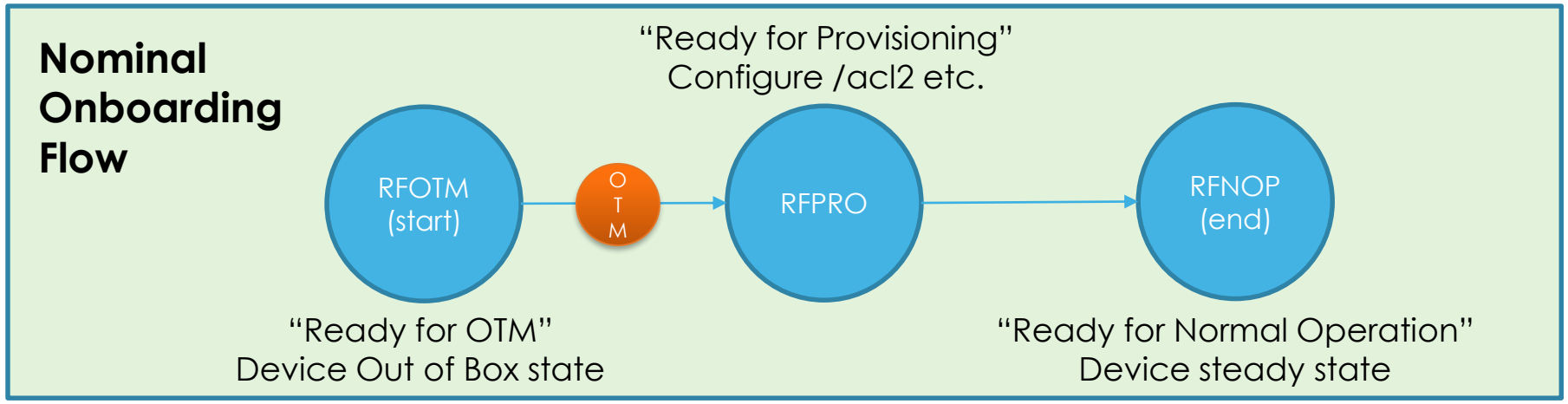
Understanding /cred

/oic/sec/cred (or "/cred") – The Credentials Resource

The /cred Resource stores the credentials – cryptographic keys, certificates, etc. – that are required to establish secure connections, and verify Client identity, among other things.



Onboarding Methods (OTMs) at a Glance

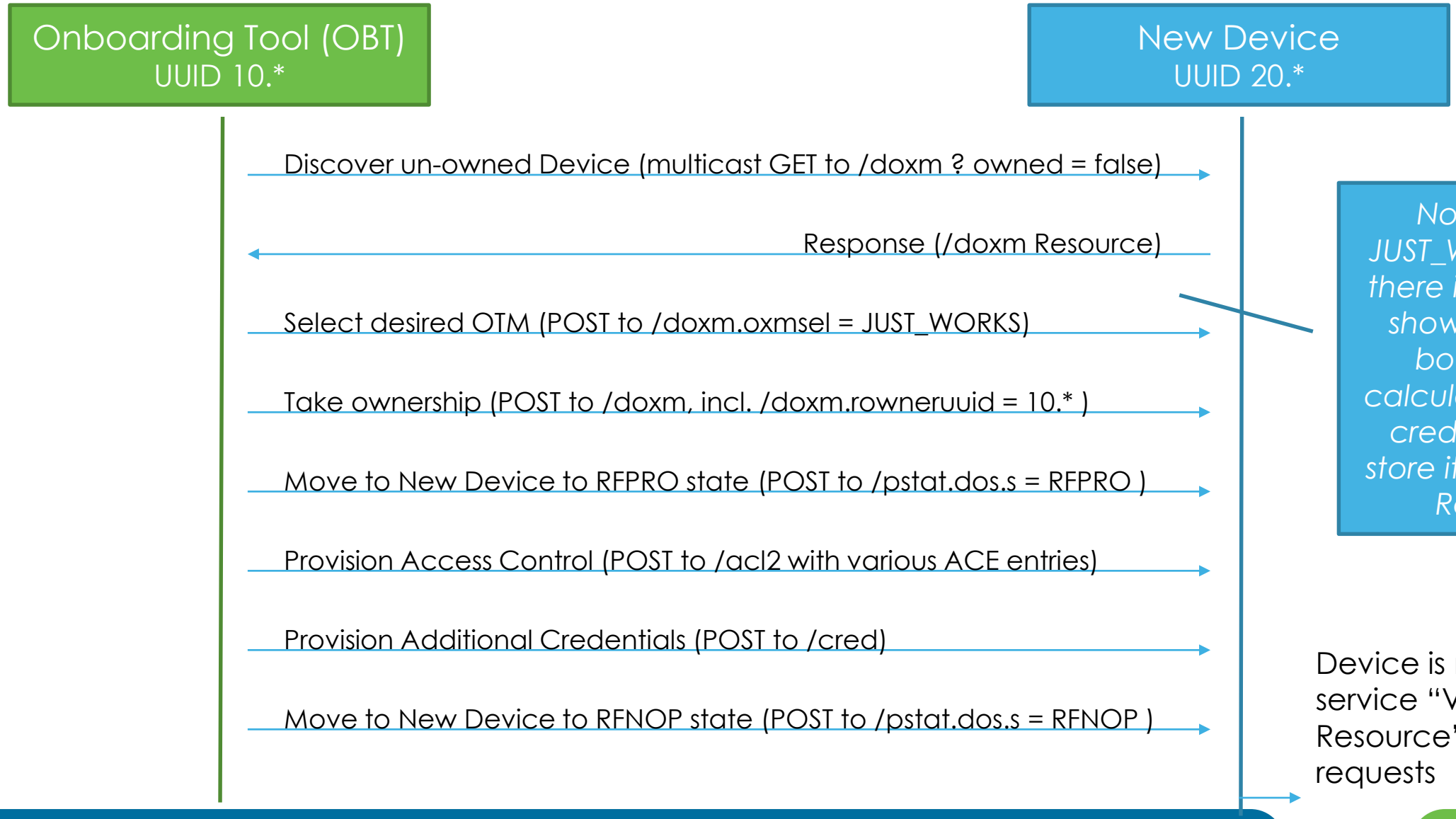


JustWorks OTM – the most basic onboarding method for getting Device into RFNOP
+ simple and functional
- vulnerable to MitM attacks on the onboarding network

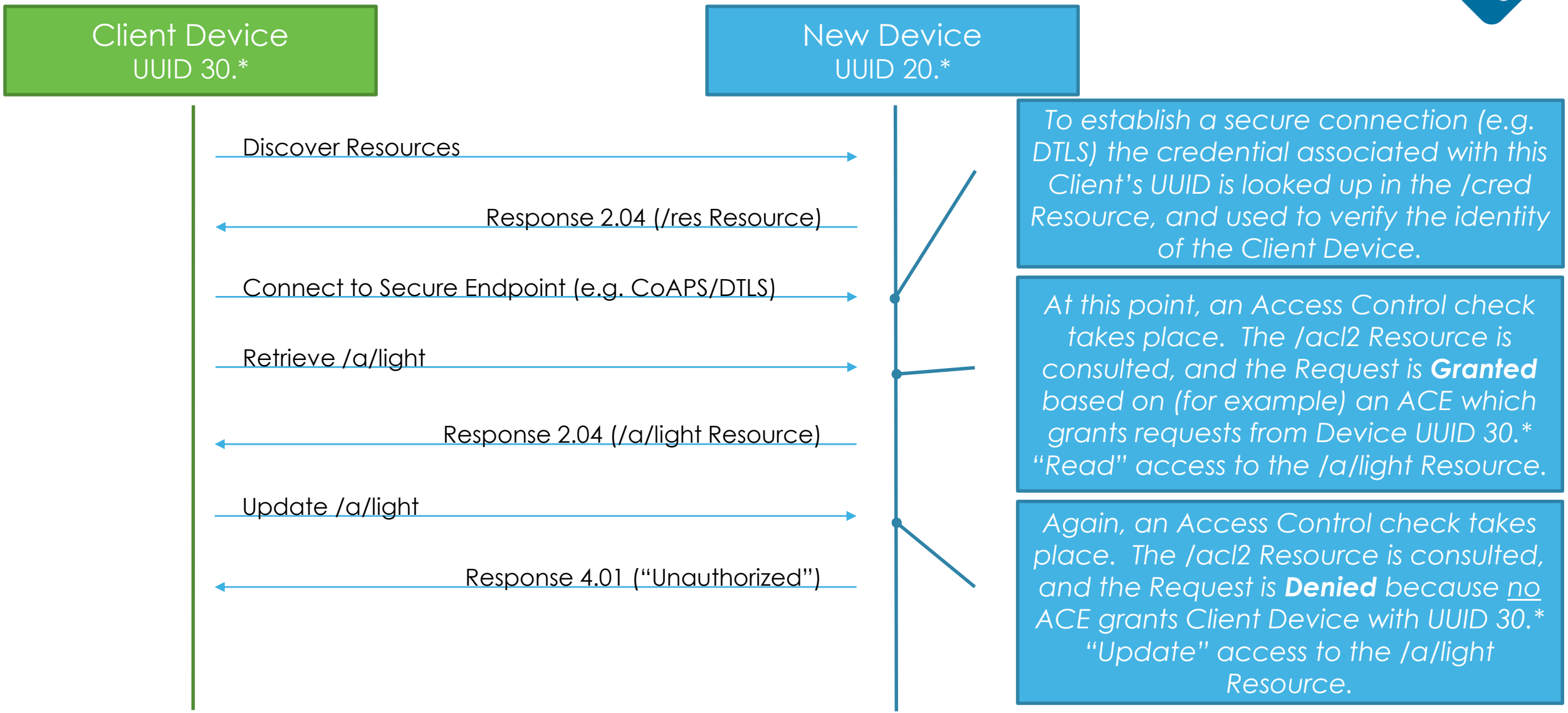
Random PIN OTM – require the User to enter a PIN to complete onboarding
+ resists MitM
- requires User Input (higher touch)
- requires PIN display and Input method to communicate PIN from Server->User->OBT

Certificate OTM – New Device supplies a Certificate to Onboarding Tool
+ best assurance and most informed onboarding decision via Cert meta-data
- requires Certificate capabilities (incl. Root Cert in OBT)

Onboarding a Device – an Incomplete Illustration



Processing a Request with Access Control applied



Implementation Example: Device SVR Configuration

- In IoTivity the SVRs are stored (in CBOR format) in a file with a ".dat" extension.

- IoTivity uses a custom tool "json2cbor" to encode a human-readable JSON-formatted SVR set into CBOR.

- Please refer to the file:

  `/iotivity/resource/examples/ocf_light/dl_server_security.json.in`

  for a complete SVR example, annotated with inline comments.

  Remember: if a request is not Granted by an ACE on the Device, then it will be Denied by default.

- For more details on IoTivity implementation specifics, please see the following IoTivity Wiki pages:
  - IoTivity Security implementation overview
    - Search for "json2cbor" for usage instructions for this format-conversion tool
  - IoTivity Provisioning Manager ("provisioningclient" – a sample OBT)

Symmetric Keys vs. Certs – Credential management and Access Control management implications

- There are security and performance related considerations for both, which should be understood by a Device Vendor choosing its credential model. **OCF's access control model has additional implications that should be taken into account.**

- **Implication to Credential (/cred Resource) Management: using Symmetric Keys requires higher-touch key management.**
  - In short, each time a New Client is added to the domain, the symmetric credential model requires each Server Device in the domain to be provisioned with a new symmetric credential, before the New Client can establish an authenticated connection with that Server.
  - By contrast, the certificate model allows the Server Device to mutually authenticate with Bob's Smartphone using the Root Cert (or CA Cert) already installed on the Server.

- **Implications to Access Control (/acl2 Resource) Management: using Certs enables more granular group-level access.**
  - With a symmetric credential installed on the Server, the New Client will match all "conntype":"auth-crypt" ACEs, and thus gain that level of access permission
    - Furthermore, a Client-specific ACE – naming the New Client by its UUID – can be installed to give the Client additional access permission
  - In a certificate model, the New Client will also match all "auth-crypt" ACEs, but can also be granted access to any "role" ACEs, on a per-role basis
    - The OBT just needs to the New Client a "role certificate" which is then supplied to the Server during connection establishment, and authorizes the New Client to effectively join the security group named in the "role"

Access Control using Groups and Wildcards

"Access Control" – filtering incoming requests to a Device, and making a "Granted" or "Denied" decision for each request before it is passed to the Resource endpoint.

- OCF defines the /acl2 Resource, which contains "Access Control Entry" objects (ACE2), to configure Access Control for a Device.

- Within an ACE2, there is a "subject" Property, which determines the Client(s) which may be Granted access by the ACE2, and this is where Group Access comes in
  - See subsequent slides

- Separate but related in the "resources" Property, which determines the Resource(s) to which the ACE2 applies
  - See subsequent slides

“subject” Groups

- **Two pre-defined groups, via the “conntype” Parameter of the ACE2 “subject” property:**
  - **“anon-clear”** – all Subjects which are connected via anonymous, clear-text channel
  - **“auth-crypt”** – all Subjects which are connected via authenticated, encrypted channel

```
{
    "aceid": 1,
    "subject": { "conntype": "anon-clear" },
    ...
}
```

- **Vendor-defined groups, via “roletype” Parameter of the ACE2 “subject” property:**
  - Symmetric keys: (1) “role” per Client, keyed on UUID
  - Certificates: n “roles” per Client, keyed on public key

```
{
    "aceid": 1,
    "subject": { "roletype": “<made_up_string_that_matches_string_in_cred>” },
    ...
}
```

“resources” Wildcards

- **OCF 1.3 defines (3) via the “wc” Parameter of the “resources” Property of ACE2:**
  - **“*”** – all Resources
  - **“+”** – all discoverable Resources
  - **“-”** – all non-discoverable Resources

  - ***These should be used with caution, because they apply also to Core and Security Resources e.g. /doxm or /acl2***

- For OCF 2.0, we have reduced the scope of these wildcards
  - **“*”** – all Non-Configuration Resources
    - NCRs exclude discovery and configuration Resources; see Security Specification Terms and Definitions
  - **“+”** – all NCRs exposing a Secure Endpoint
  - **“-”** – all NCRs exposing an Unsecure Endpoint

  - ***With these new definitions, a basic /acl2 configuration where “auth-crypt” has access to “+” and “anon-clear” has access to “-” may be a suitable baseline access control configuration for many Devices.***

Further Reading

- The **Vendor Security Considerations** whitepaper describes critical Security-related guidelines and best practices for OCF Device Vendors.
- The **OCF Security Specification** contains full descriptions of each SVR – and SVR Property – along with additional details on confidentially and integrity assurance in OCF Devices.
- The **IoTivity Wiki** gives more detail on IoTivity APIs, implementation detail, and developer usage guidelines
- The **IoTivity ocf_light App (/iotivity/resource/examples/ocf_light)** contains and annotated version of the SVR .json.
- The **IoTivity OCFSecure App (/iotivity/examples/OCFSecure)** shows an embedded-device Server example, with a thorough README.md that includes instructions for running the Certification Test Tool against the Server.

OPEN CONNECTIVITY FOUNDATION®