OCF “Gaborone-1” – Removal of wrong/obsolete text from Security Overview clause – Security WG CR 3230

Legal Disclaimer

5 Security Overview

5.1 Preamble

The goal of OCF's security architecture is to protect the data and device states represented by the OCF Resources. From the OCF perspective, a Device is a certifiable logical entity that participates in an OCF ecosystem. During interactions between Devices, the Device acting as the Server holds and controls the Resources and provides the Device acting as a Client access to those Resources, subject to a set of security mechanisms and conforming to the policies configured by the OCF Security Domain Owner. The Platform hosting the Device may provide security hardening to ensure robustness of the variety of operations described in this document. Multiple Devices may be hosted by the same Platform.

The security model is depicted in Figure 2 and described in the following steps:

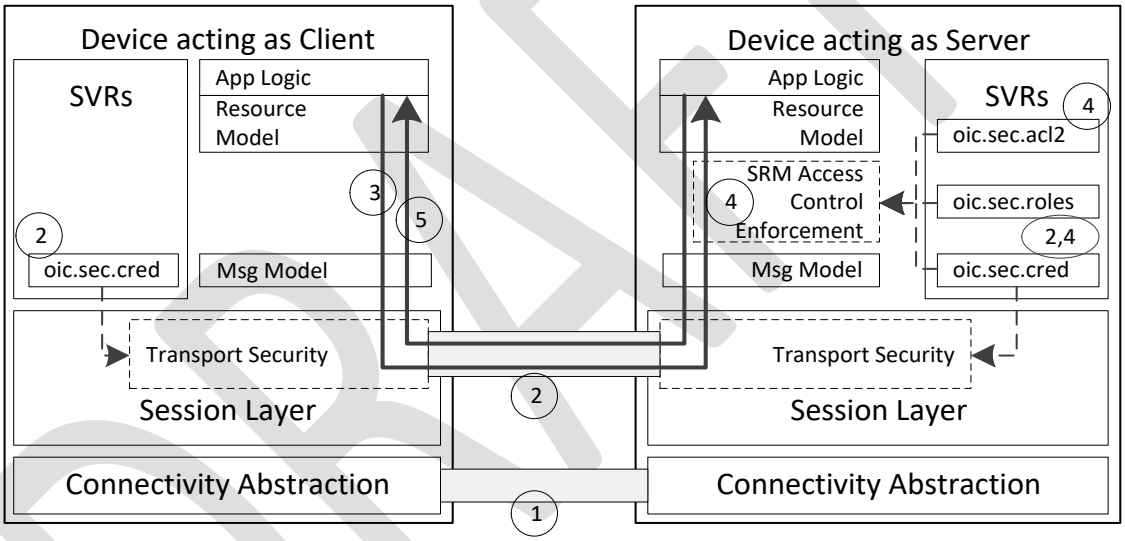

Figure 1 – OCF Layers

1) The Client establishes a network connection to the Server (Device holding the Resources).
2) The Devices (Server and Client) exchange messages either via a mutually-authenticated secure channel between the two Devices or via an unsecure connection.
   a) The "/oic/sec/cred" Resource on each Device holds the credentials used for mutual authentication and credentials used for role authorization.
   b) Messages received over a secured channel are associated with a "deviceUUID". In the case of a certificate credential, the "deviceUUID" is part of the certificate received from the other Device. In the case of a symmetric key credential, the "deviceUUID" is associated with the credential in the "/oic/sec/cred" Resource.
   c) The Client may present its role certificate to request association with a role identifier ("roleid"). The Server may associate the Client with any number of role identifiers.

d) Requests received by a Server over an unsecured channel are treated as anonymous and are not associated with any "deviceUUID" or "roleid".

3) The Client submits a request to the Server.

4) The Server receives the request.

a) If the request is received over an unsecured channel, the Server treats the request as anonymous and no "deviceUUID" or "roleid" are associated with the request.

b) If the request is received over a secured channel, then the Server associates the request with the "deviceUUID" of the Client and all valid "roleid" values of the Client by default.

c) The Server then consults the Access Control List (ACL), and looks for an Access Control Entry (ACE) matching the following criteria:

i) The requested Resource matches a Resource reference in the ACE

ii) The requested operation is permitted by the "permissions" of the ACE, and

iii) The "subjectUUID" contains either one of a special set of wildcard values or, if the Device is not anonymous, the subject matches the Client "deviceUUID" associated with the request or a valid "roleid" associated with the request. The special wildcard values authorize all Devices communicating over either authenticated and encrypted sessions or unsecured sessions to interact according to the ACE.

If there is a matching ACE, then access to the Resource is permitted; otherwise access is denied. Access is enforced by the Server's Secure Resource Manager (SRM).

5) The Server sends a response back to the Client.

Resource protection includes protection of data both while at rest and during transit. Aside from access control mechanisms, the OCF Security Specification does not include specification of secure storage of Resources. Secure storage may be accomplished through the use of hardware security or encryption of data at rest. The exact implementation of secure storage is subject to a set of hardening requirements that are specified in clause 14 and may be subject to certification guidelines.

Data in transit protection is specified fully as a normative part of this document. This document only supports in transit data protection at the transport layer through use of mechanisms such as DTLS.

NOTE: DTLS will provide packet by packet protection, rather than protection for the payload as whole. For instance, if the integrity of the entire payload as a whole is required, separate signature mechanisms must have already been in place before passing the packet down to the transport layer.

Figure 3 depicts OCF Security Enforcement Points.

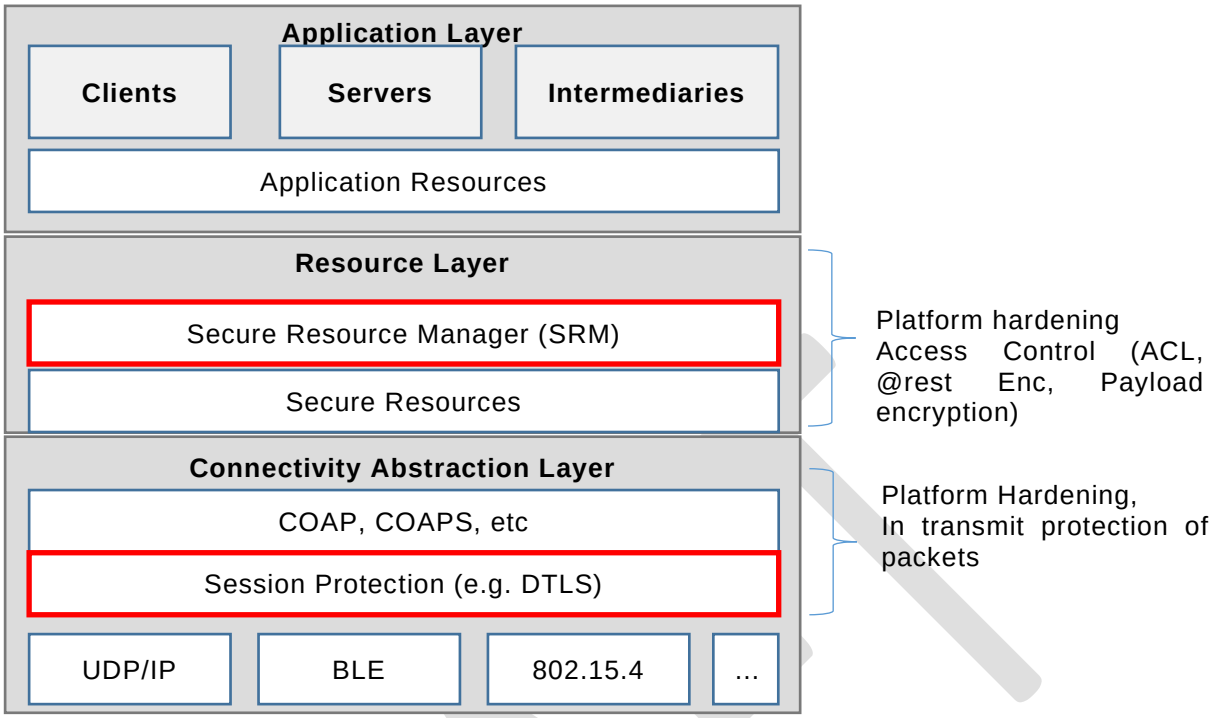

Figure 2 – OCF Security Enforcement Points

5.2 Access Control

5.2.1 Access Control General

The OCF framework assumes that Resources are hosted by a Server and are made available to Clients subject to access control and authorization mechanisms. The Resources at the Server are protected through implementation of access control, authentication and confidentiality protection. This clause provides an overview of access control through the use of Access Control Lists. However, access control in OCF is agnostic regarding transport and connectivity abstraction layers.

Implementation of access control relies on a-priori definition of a set of access policies for the Resource. The policies are stored locally in an ACL Resource provisioned by an Access Management Service (AMS) in the form of Access Control Entries (ACE). The lack of such an associated ACE results in the Resource being inaccessible. Multiple types of access control mechanisms may be applied:

– Subject-based access control (SBAC), where the ACE matches the identity of the Client against the subject included in the policy defined for the Resource. Asserting the identity of the Client requires an authentication process.
– Role-based Access Control (RBAC), where the ACE matches a role identifier included in the policy for the Resource to a role identifier associated with the Client.
– Wildcard-based Access Control, where the ACE matches a connection type, used to access the Resource (i.e. any mutually-authenticated connection)

.

The ACE only applies if the ACE matches both the subject (i.e. Client) and the requested Resource. There are multiple ways a subject could be matched, (1) Device UUID, (2) Role Identifier or (3) wildcard. The way in which the Client connects to the Server may be relevant for making access control decisions. Wildcard matching on authenticated vs. unauthenticated and encrypted vs. unencrypted connection allows an access policy to be broadly applied to subject classes.

Example Wildcard Matching Policy:

```
"aclist2": [
 {
  "subject": {"conntype" : "anon-clear" },
  "resources":[
    { "wc":"*" }
  ],
  "permission": 31
  },
  {
   "subject": {"conntype" : "auth-crypt" },
   "resources":[
    { "wc":"*" }
  ],
  "permission": 31
  },
]
```

Details of the format for ACL are defined in clause 12. The ACL is composed of one or more ACEs.

Some Resources, such as Collections, generate requests to linked Resources when appropriate Interfaces are used. In such cases, additional access control considerations are necessary. Additional access control considerations for Collections when using the batch OCF Interface are found in clause 12.2.7.3.ACL Resource requires the same security protection as other sensitive Resources when it comes to both storage and handling by the SRM

5.2.2 ACL Architecture

The Server examines the Resource(s) requested by the client before processing the request. The access control resource is searched to find one or more ACE entries that match the Client and the requested Resources. If a match is found, then permission and period constraints are applied. If more than one match is found, then each ACE entry is evaluated for a match independently.

The Server uses the connection context to determine whether the subject has authenticated or not and whether data confidentiality has been applied or not. If the user has authenticated, then subject matching may happen at increased granularity based on role or device identity.

Each ACE contains the permission set that will be applied for a given Client. Permissions consist of a combination of CREATE, RETREIVE, UPDATE, DELETE and NOTIFY (CRUDN) actions. Clients authenticate as a Device and optionally operating with one or more roles. Devices may acquire elevated access permissions when asserting a role. For example, an "oic.role.owner" role might expose additional Resources and OCF Interfaces not normally accessible.

Servers host ACL Resources locally. Local ACLs allow greater autonomy in access control processing.
The following use cases describe the operation of access control:

Use Case 1: As depicted in Figure 4, Server Device hosts 4 Resources (R1, R2, R3 and R4). Client Device D1 requests access to Resource R1 hosted at Server Device 5. ACL[0] corresponds to Resource R1 and includes D1 as an authorized subject. Thus, Device D1 receives access to Resource R1 because the local ACL "/oic/sec/acl2/0" matches the request.

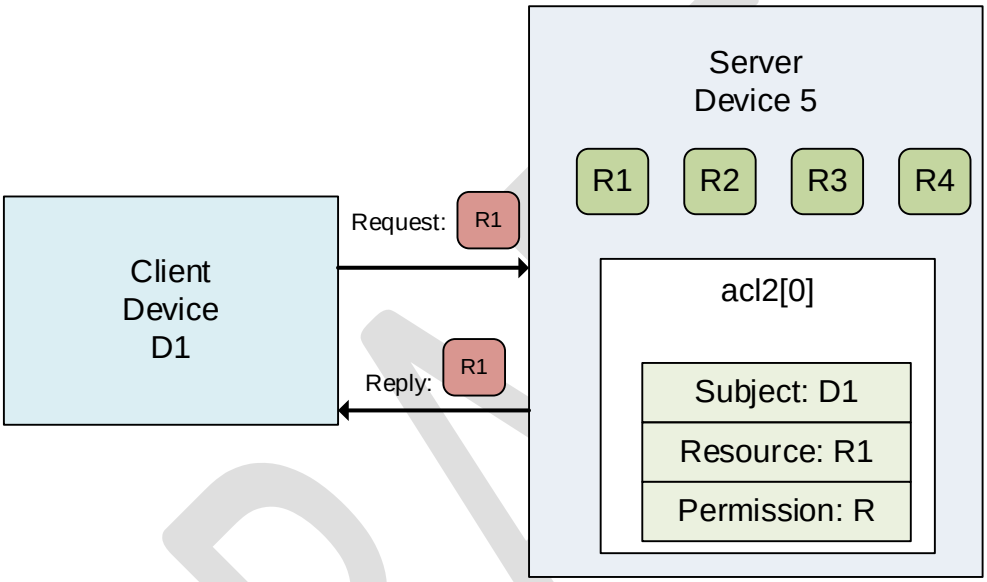

Figure 3 – Use case-1 showing simple ACL enforcement

5.3 Onboarding Overview

5.3.1 Onboarding General

Before a Device becomes operational in an OCF environment and is able to interact with other Devices, it needs to be appropriately onboarded. The first step in onboarding a Device is to configure the ownership where the legitimate user that owns/purchases the Device uses an Onboarding tool (OBT) and using the OBT uses one of the Owner Transfer Methods (OTMs) to establish ownership. Once ownership is established, the OBT provisions the Device, at the end of which the Device becomes operational and is able to interact with other Devices in an OCF environment.

Figure 8 depicts an overview of Onboarding.

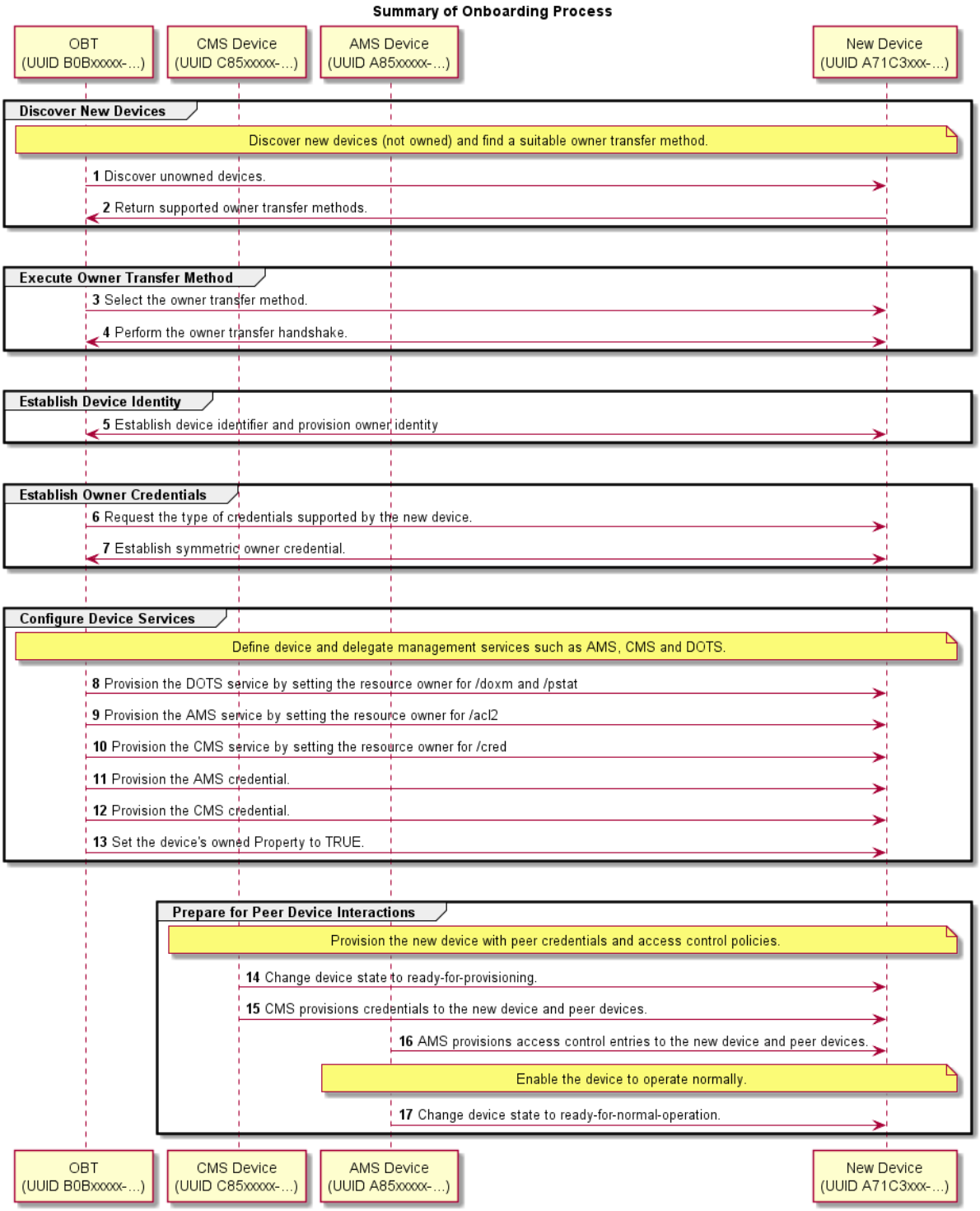

Figure 4 – Onboarding overview

This clause explains the onboarding and security provisioning process but leaves the provisioning of non-security aspects to other OCF documents. In the context of security, all Devices are required to be provisioned with minimal security configuration that allows the Device to securely interact/communicate with other Devices in an OCF environment. This

minimal security configuration is defined as the Onboarded Device "Ready for Normal Operation" and is specified in 7.5.

5.3.2 Onboarding Steps

The flowchart in Figure 9 shows the typical steps that are involved during onboarding. Although onboarding may include a variety of non-security related steps, the diagram focus is mainly on the security related configuration to allow a new Device to function within an OCF environment. Onboarding typically begins with the Device becoming an Owned Device followed by configuring the Device for the environment that it will operate in. This would include setting information such as who may access the Device and what actions may be performed as well as what permissions the Device has for interacting with other Devices.

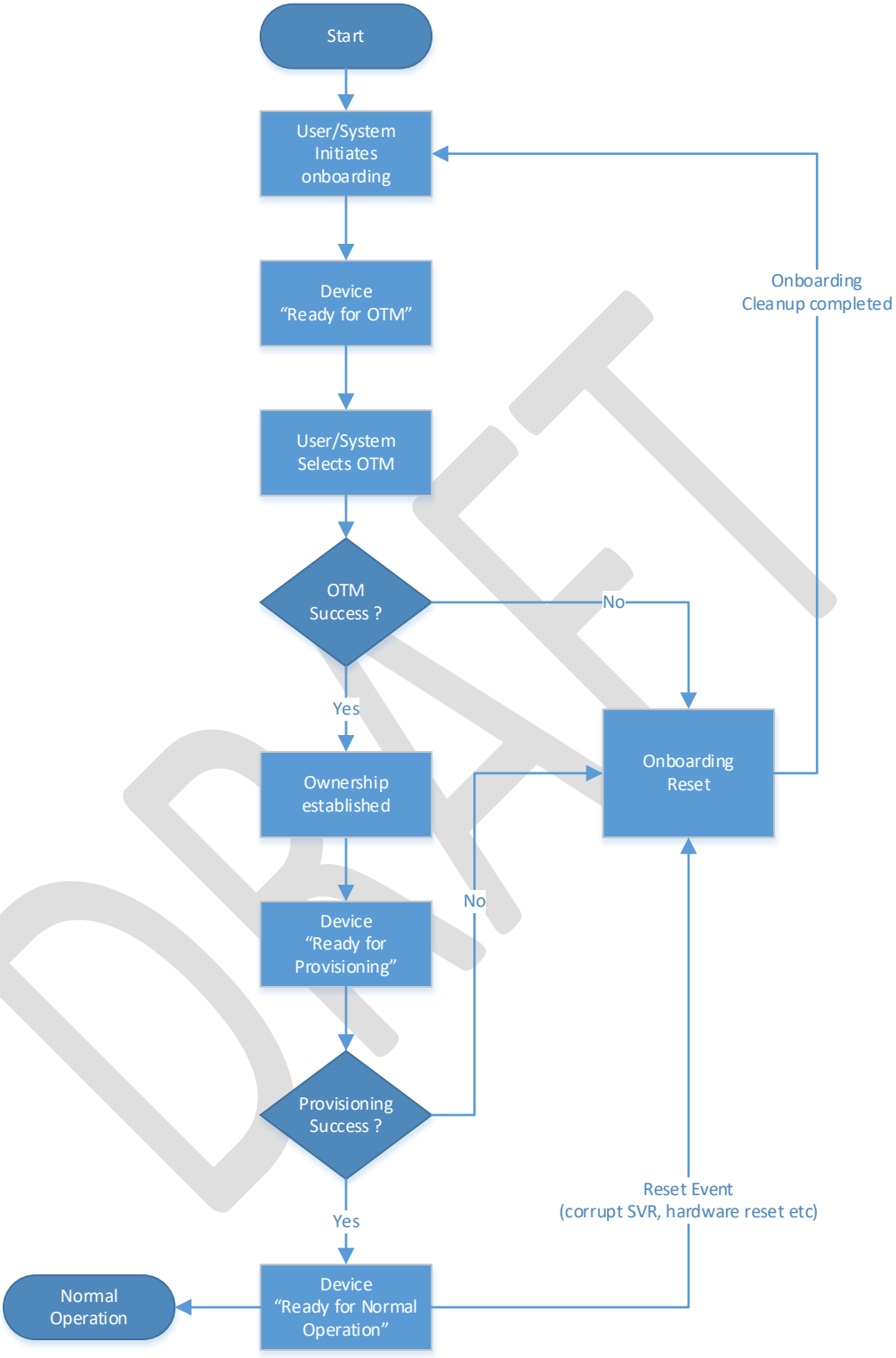

Figure 5 – OCF Onboarding Process

5.3.3 Establishing a Device Owner

The objective behind establishing Device ownership is to allow the OCF Security Domain Owner to assert itself as the owner and manager of the Device and introduce the Device into the OCF Security Domain. This is done through the use of a DOTS that includes the creation of an ownership context between the new Device and the DOTS and asserts operational control and management of the Device. The DOTS is hosted on an OBT.

The DOTS uses one of the OTMs specified in 7.3 to securely establish Device ownership.

An OTM establishes a new owner (the operator of DOTS) that is authorized to manage the Device. Ownership Transfer accomplishes the following:

– The DOTS provisions an Owner Credential (OC) to the "creds" Property in the "/oic/sec/cred" Resource of the Device. This OC allows the Device and DOTS to mutually authenticate during subsequent interactions. The OC associates the DOTS Device UUID with the "rowneruuid" Property of the "/oic/sec/doxm" Resource establishing it as the resource owner.

– The Device owner establishes trust in the Device through the OTM.

– Provisioning of appropriate credentials for the Device to be a member of the OCF Security Domain..

5.3.4 Provisioning for Normal Operation

Once the Device has the necessary information to initiate provisioning, the next step is to provision additional security configuration that allows the Device to become operational. This may include setting various parameters and may also involve multiple steps. Also provisioning of ACL's for the various Resources hosted by the Server on the Device is done at this time. The provisioning step is not limited to this stage only. Device provisioning may happen at multiple stages in the Device's operational lifecycle. However specific security related provisioning of Resource and Property state would likely happen at this stage at the end of which, each Device reaches the "Ready for Normal Operation" (RFNOP) State. The RFNOP State is consistent and well defined regardless of the specific OTM used or regardless of the variability in what gets provisioned. However individual OTM mechanisms and provisioning steps may specify additional configuration of Resources and Property states. The minimal mandatory configuration required for a Device to be in RFNOP state is specified in 8.

5.3.5 Device Provisioning for OCF Cloud and Device Registration Overview – moved to OCF Cloud Security document

This clause is intentionally left blank.

5.3.6 OCF Compliance Management System

The OCF Compliance Management System (OCMS) is a service maintained by the OCF that provides Certification status and information for OCF Devices.

The OCMS shall provide a JSON-formatted Certified Product List (CPL), hosted at the URI: https://www.openconnectivity.org/certification/ocms-cpl.json

The OBT shall possess the Root Certificate needed to enable https connection to the URI https://www.openconnectivity.org/certification/ocms-cpl.json.

The OBT should periodically refresh its copy of the CPL via the URI https://www.openconnectivity.org/certification/ocms-cpl.json, as appropriate to OCF Security Domain owner policy requirements.

5.4 Provisioning

5.4.1 Provisioning General

OCF security provisioning includes processes during and after the ownership transfer like configuration of credentials for interacting with provisioning services, configuration of any security related Resources and credentials for interacting with any services or Devices that the provisioned Device needs to contact later on.

The Device needs to engage with the CMS and AMS to be provisioned with:

– Security credentials through a CMS, which is currently assumed to be embedded in the same OBT as the DOTS.
– Access control policies and ACLs through an AMS, which is currently assumed to be embedded in the same OBT as the DOTS.

To be able to support the use of distinct device management services, some Device Secure Virtual Resources (SVRs) have an associated Resource owner identified in the Resource's rowneruuid Property.

The "rowneruuid" Property of the "/oic/sec/doxm" and "/oic/sec/pstat" resources identifies the DOTS.

The "rowneruuid" Property of the "/oic/sec/cred" resource identifies the CMS.

The "rowneruuid" Property of the "/oic/sec/acl2" resource identifies the AMS.

The DOTS provisions credentials that enable secure connections between OCF Services and the new Device. The DOTS initiates client-directed provisioning by signaling the OCF Service.

5.4.2 Access Control Provisioning

ACL provisioning is performed over a secure connection between the AMS and its Devices. The AMS provisions the ACL by updating the Device’s ACL Resource.

5.4.3 Credential Provisioning

The CMS securely provisions credentials for Device-to-Device interactions using the CMS credential provisioned by the DOTS during the onboarding procedure. The CMS is also expected to proactively monitor the credentials installed on the Device and update them when needed (e.g. close to the expiration date).

5.4.4 Role Provisioning

The Servers, receiving requests for Resources they host, need to verify the role identifier(s) asserted by the Client requesting the Resource and compare that role identifier(s) with the constraints described in the Server's ACLs Thus, a Client may need to be provisioned with one or more role credentials. Once provisioned, the Client can assert the role it is using as described in 10.4.2, if it has a certificate role credential.

Each Device holds the assertable role(s) information as a Property within the Credential Resource. Each Device holds the asserted role(s) information as Properties within the Roles Resource.

All asserted roles are used in ACL enforcement. When a server has multiple roles asserted for a Client, access to a Resource is granted if it would be granted under any of the roles.

5.5 Secure Resource Manager (SRM)

SRM plays a key role in the overall security operation. In short, SRM performs both management of SVR and access control for requests to access and manipulate Resources. SRM consists of 3 main functional elements:

– A Resource manager (RM): responsible for 1) Loading SVRs from persistent storage (using PSI) as needed. 2) Supplying the Policy Engine (PE) with Resources upon request. 3) Responding to requests for SVRs. While the SVRs are in SRM memory, the SVRs are in a format that is consistent with device-specific data store format. However, the RM will use JSON format to marshal SVR data structures before being passed to PSI for storage, or travel off-device.

– A Policy Engine (PE) that takes requests for access to SVRs and based on access control policies responds to the requests with either "ACCESS_GRANTED" or "ACCESS_DENIED". To make the access decisions, the PE consults the appropriate ACL and looks for best Access Control Entry (ACE) that can serve the request given the subject (Device or role) that was authenticated by DTLS.

– Persistent Storage Interface (PSI): PSI provides a set of APIs for the RM to manipulate files in its own memory and storage. The SRM design is modular such that it may be implemented in the Platform's secure execution environment; if available.

Figure 10 depicts OCF's SRM Architecture.

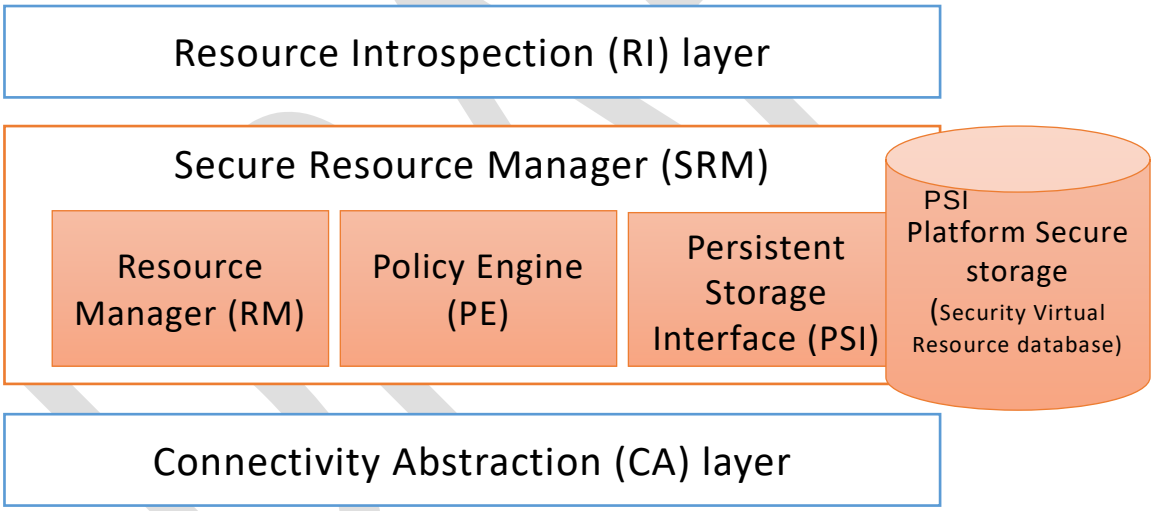

Figure 6 – OCF's SRM Architecture

5.6 Credential Overview

Devices may use credentials to prove the identity and role(s) of the parties in the Client to Server communication. Credentials may be symmetric or asymmetric. Each Device stores secret and public parts of its own credentials where applicable, as well as credentials for other Devices that have been provisioned by the DOTS or a CMS. These credentials may then be used in the establishment of secure communication sessions (e.g. using DTLS). Role certificates may be used after an authenticated session is established to assert one or more roles for a Device.

The credential types available within this document include:

– Pairwise symmetric keys
– Certificates
– Raw asymmetric keys

Devices may not support all of these credential types. The set of supported credential types for any Device is contained in its "sct" Property of the "/oic/sec/doxm" Resource.