Device Lifecycle

In OCF, OneM2M, L2M2M, T2TRG vs WoT
References

OCF  https://openconnectivity.org/developer/specifications/
  - OCF Onboarding Tool Specification
    - 5.1, Table 1: definition of states from OBT point of view
  - OCF Security Specification
    - 5.3, Figure 8: Onboarding overview
    - Sections 7 and 8

OneM2M
  - Technical Report, Security 2016, Sections 11-12
  - Security Solutions 2014

LwM2M  Technical Specification: Core / Bootstrap

T2TRG Security (RFC 8576),  Thing Lifecycle
OCF

Stage 1. Onboarding:
- discover *unowned* devices and get supported ownership transfer methods (OTM)
- select and perform OTM (based on certificate / shared key / random pin / vendor-specific)
- provision *device identity*
- provision *owner credentials*

Stage 2. Security Provisioning (can happen even in operational state)
- provision *credentials* (other keys and certificates than ownership)
- provision *ACL's* to access other devices/services (like discovery/addressing/id's).

Stage 3. Security Configuration
- configure *apps* and access management (policies).

On an OCF device, WoT servient is an application (that acts as a bridge).

**Note:** OCF provides bridging specifications to BLE, OneM2M, AllJoyn, UPlus, ZigBee, Z-Wave.
OneM2M

GBA: Generic Bootstrapping Architecture
Trust Enabling Architecture:
  M2M Enrolment, Authentication, Authorization Functions
M2M Initial Provisioning:
  - M2M Node Enrolment and Service Provisioning
    - Pre-provisioning: out of band, or manufacturer certificate
    - Remote provisioning: relies on pre-existing credentials to access the Enrolment function
    - Service provisioning (credentials for using Authentication Functions)
  - Application Enrolment/provisioning (M2M Application key)
Lightweight M2M

4 bootstrap modes:

1. Factory: pre-provisioned for communicating with Bootstrap Server
2. Smartcard: carries initial secrets
3. Client initiated: needs a pre-provisioned account to access Bootstrap Server
4. Server initiated: if a (non-specified) secure connection can be established between client and server.

Bootstrap discovery is also supported (not the same as operational discovery).
T2TRG lifecycle

_Manufactured / _SW update / _Decommissioned

/ _Installed / _Application / _Removed &

/ _Commissioned / _Application reconfigured / replaced

/ _Application / _Application / _Reownership &

/ running / running / recommissioned

+##+##+###+#############+##+##+#############+##+##+##############>>>

/              /              /
/ ______________/
/ time //

/ Bootstrapping / Maintenance & / Maintenance &

/        / rebootstrapping / rebootstrapping

/ Operational / Operational

Operational
Possible Mapping to WoT device lifecycle model

- **Manufactured** | Factory defaults
  - Factory defaults, with or without initial secrets

- **Bootstrapping** | Provisioning, with sub-states:
  1. (Onboarding | Bootstrapping?): Provision with trust chain and device id in a solution (multiple types of onboarding: shared key, manufacturer certificate, etc).
  2. Service Provisioning: with security data to access basic services in a solution (discovery, credential/access management, etc) + solution/service configuration data (after this [+ reboot], MAY become Operational in some cases)
  3. App Provisioning: with security data to access other devices in a solution and app/user configuration data (after this [+ reboot], becomes Operational)

- **Data**: security + configuration (for solution [+ user])

- **Operational**, with sub-states:
  - Normal operation
  - (re)configuration by user or provider (can be done while operating)
  - Maintenance / SW updates (some can do it in background, while operating)
  - **Data**: solution [+user] configuration, solution [+user] data

- **Decommissioned**
  - All data and trust chain removed, a.k.a. Reset to factory defaults.
  - It can be still recommissioned with majority of protocols.
  - It could be merged with the Manufactured | Factory defaults state.

- **Destroyed**: device HW no longer usable