

Session 4 Notes – Feb 3, 2026

USP Integration and Management Path

Discussion covered the distinction between *what* connects Barton to USP and *how* that connection is implemented.

The *what* refers to exposing Barton functionality into USP so that ACS/USP systems can manage IoT devices. Direction is for USP to inherit the Matter data model, including device representations and supported interactions. This includes passthroughs from Barton where appropriate.

The *how* refers to mapping Barton into rbus first, then mapping rbus into USP. This allows Barton to remain modular while enabling existing RDK management frameworks to consume its data. Minimum requirements include build-time and version alignment so Barton's capabilities can be surfaced cleanly. Operators will also need an ACS that supports USP IoT extensions.

Cloud interface needs were raised. Any Barton API capable of cloud interaction could, in principle, be exposed as a method-style call through USP. USP and rbus already support this; the question is identifying which Barton interfaces should be published upward.

Two interface categories were identified:

1. Device data model exposure, expected to follow Matter's model so USP can represent devices and interactions consistently.
2. Control and management operations, including telemetry requests, commissioning, control-plane operations, and ongoing monitoring.

These are all considered suitable for USP's property-based data model and method mappings. Action item: Thomas to provide concrete examples, including asynchronous notifications.

SBMD Update

Discussion indicated holding off on expanding the PoC device set until SBMD is further along, as the current C++ driver and bridging model sets a higher bar than necessary. SBMD will simplify device mapping through dynamic loading, lowering that bar and reducing the likelihood of near-term work becoming obsolete.

Initial SBMD support is expected to arrive with a substantial set of new device types. Target coverage includes the new IKEA Matter device catalog, with a basic first pass expected to handle lights and similar simple devices.

Question was raised regarding Matter camera support. Cameras are in scope. Barton would commission them, initiate sessions, and handle aspects such as video streams that fall outside the Matter protocol itself.

Timeline is difficult to predict; early summer was suggested as a tentative goal.

Adding New Devices and Driver Models

Andrew (Liberty Global) asked whether adding a new device will always require creating a new description or library, and whether that could be limiting. He asked about the availability of reusable libraries (e.g., YAML-based definitions).

Rahul clarified that SBMD introduces a class-driver approach that avoids per-device custom drivers. For example, although IKEA lists 21 devices, 11 are lights, and a single YAML driver would likely cover all 11 and most other Matter-compliant lights. SBMD significantly reduces redundant effort.

Zigbee support will differ. Current Zigbee functionality uses native C drivers, not YAML. A Zigbee-specific SBMD path would require its own workstream. No timeline was proposed.