redbend
CATALYZING CHANGE

ALLJOYN UPDATE SERVICE

UPDATED CONTRIBUTION & DEMO PLAN

Aug 2015

STATUS UPDATE FROM MID 2014 ACTIVITY

- Alljoyn has matured significantly
- Device Update scenarios/use cases crystallized
- Alljoyn hardware requirements to implement device update client (as reference/contribution/demo) eased:
  - From QCA Boards to e.g. Raspberry Pi

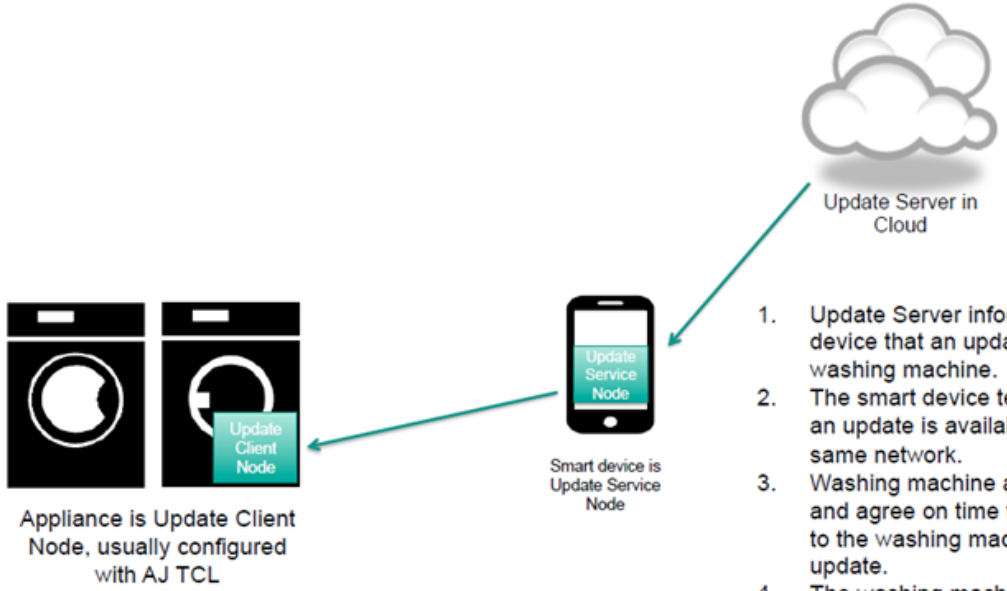
PLAN OF ACTION AUG 2015

- Redefine the update scenarios/use cases
  - Based on refinement of use cases
  - (including J.Spain input)
- Provide requirements for contribution and demo
  - For the simple scenario only
- Develop from 1st week of September
- Demo ready for AJ Summit

MAIN DEVICE UPDATE SCENARIOS

- Scenario 1: Alljoyn (mostly thin) devices are updated directly from OEM Cloud, no central gateway involved
- 1a variant
  - Update via AJ Smart Device implementing:
    - Update Service
    - Alljoyn Control App
  - Where the AJ Control App is configured to communicate with OEM update server
  - OEM update server notifies AJ Control App of update availability for the AJ thin device
  - AJ Control app downloads the update to the Smart Device
  - Update service notifies AJ Thin device (update client) of update availability
  - Update client downloads update package from Smart device
  - (update installation conditions rule engine implemented in client –out of scope)

SCENARIO 1A DEPICTION



1. Update Server informs homeowner's smart device that an update is available for their washing machine.
2. The smart device tells the washing machine that an update is available when they are both on the same network.
3. Washing machine and smart device negotiate and agree on time to transfer the update package to the washing machine and when to start the update.
4. The washing machine requests the update package and starts updating itself.
5. The smart device is notified upon completion.

MAIN DEVICE UPDATE SCENARIOS

- Scenario 1: Alljoyn (mostly thin) devices are updated directly from OEM Cloud, no central gateway involved
- 1b variant
  - AJ thin device (configured on WiFi network and able to communicate with OEM server) downloads update package directly from OEM update server
  - AJ Control app and Update service manage the update discovery, triggering and process control
  - AJ Smart Device implementing:
    - Update Service
    - Alljoyn Control App
  - Where the AJ Control App is configured to communicate with OEM update server
    - OEM update server notifies AJ Control App of update availability for the AJ thin device
    - Update service notifies device (AJ update client) of update availability
    - Update client triggers a direct download of update package from OEM server to AJ thin device
    - AJ Control App displays process to user, notified of status via Update service

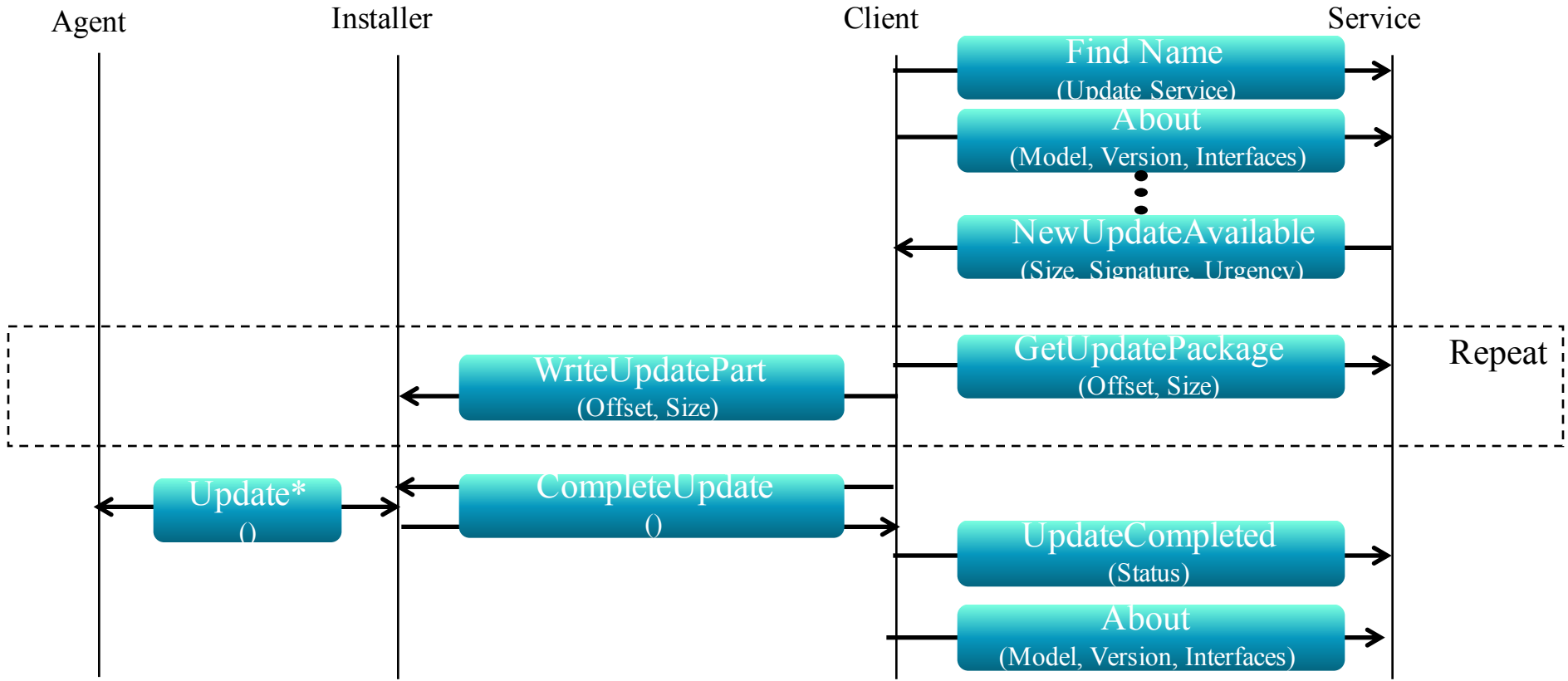
MAIN DEVICE UPDATE SCENARIOS

- Scenario 2: Alljoyn (mostly thin) devices are updated via Gateway Agent running on a dedicated gateway device

SW ELEMENTS REQUIRED FOR DEMO

- Alljoyn Thin Device
  - AllJoyn Core
  - AllJoyn Services
    - Control Panel
    - Notifications
  - Update Client
- Smart Device
  - AllJoyn Core
  - AllJoyn Services
    - Control Panel
    - Notifications
  - Update Service
  - AJ Dashboard app (QCOM)
  - (GW Agent)?
- Other AJ Device (able to announce notifications)
  - AllJoyn Core
  - AllJoyn Services
    - Control Panel
    - Notifications

ORIGINAL PROPOSED UPDATE FLOW

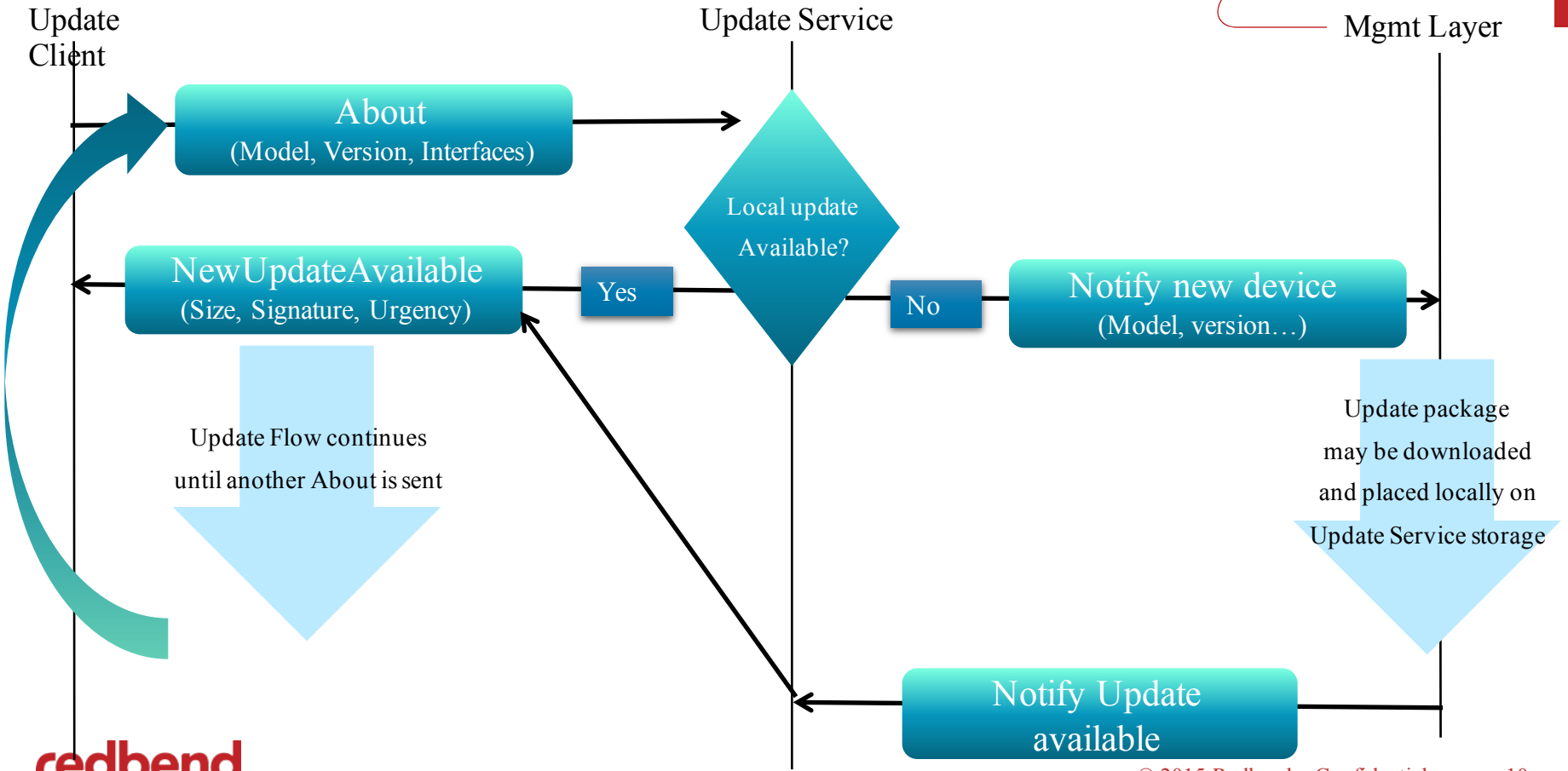


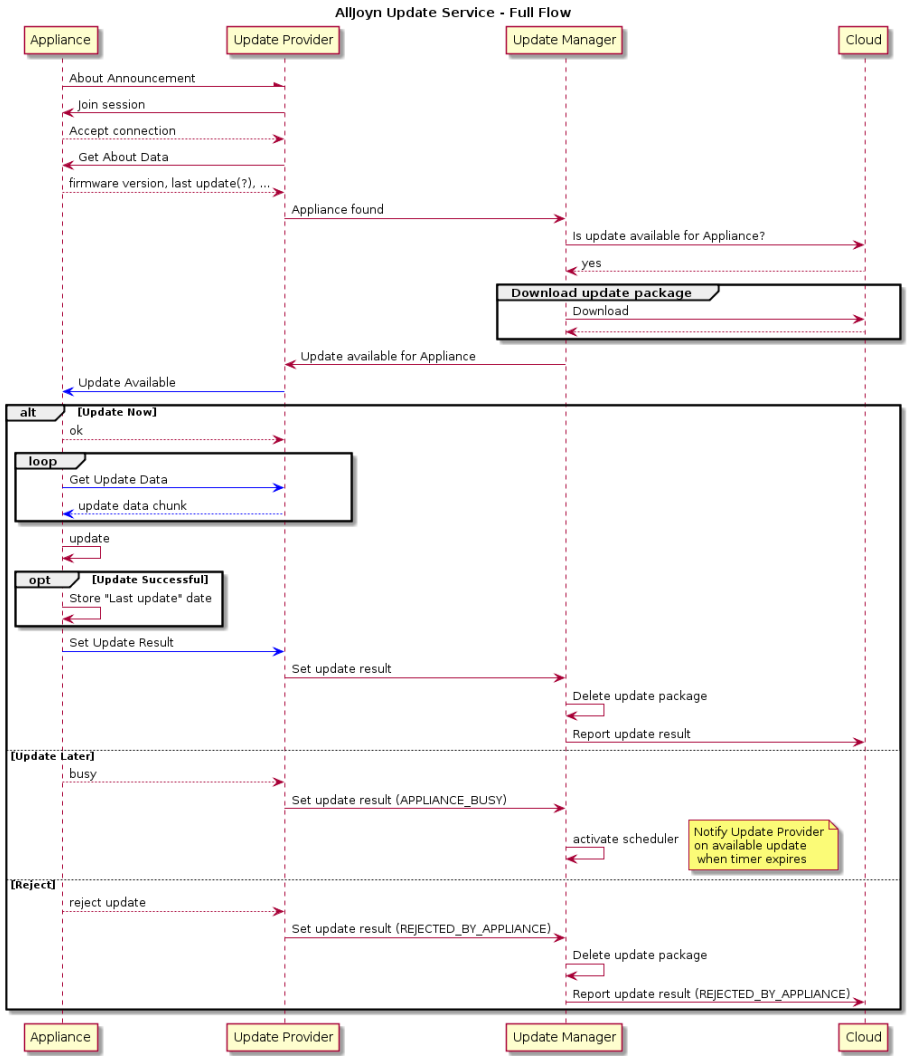
redbend

* Update Agent invocation depends on device flash layout (see later)

9

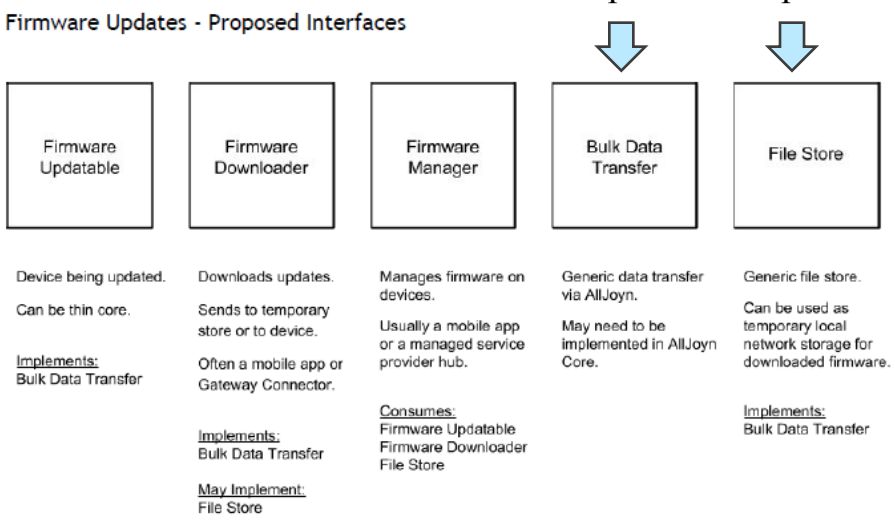
ADDITIONAL: SERVICE MGMT FLOW





JOSH SPAIN (AFFINEGY) SUGGESTION

We think these interfaces are generic and not specific to Update Services



FirmwareDownloader

Methods

- IsSupported(deviceid)
- CheckForUpdate(deviceid, firmwareversion)
- DownloadUpdate(deviceid, firmwareversion=latest) returns downloadid

Signals

- DownloadInProgress(deviceid, firmwareversion, progress)
- DownloadComplete(deviceid, firmwareversion)
- DownloadFailed(deviceid, firmwareversion, reason)

FirmwareUpdatable

Properties

- DeviceId (manufacturer, model, hardware version)
- FirmwareVersion
- UpdateStatus (idle, transfer in progress, update in progress)

Methods

- BeginFirmwareUpdate(token, port)

Signals

- FirmwareUpdateStatusChanged(status)
- FirmwareUpdateFailed(reason)

FirmwareManager

Aggregates firmware updates and downloads and provides the information as a friendly AllJoyn interface. This set of AllJoyn interfaces is not absolutely necessary since the work can be implemented manually through the FirmwareUpdatable and FirmwareDownloader interfaces; however, it provides a wrapper around the lower level interfaces to allow for controlling of

Properties

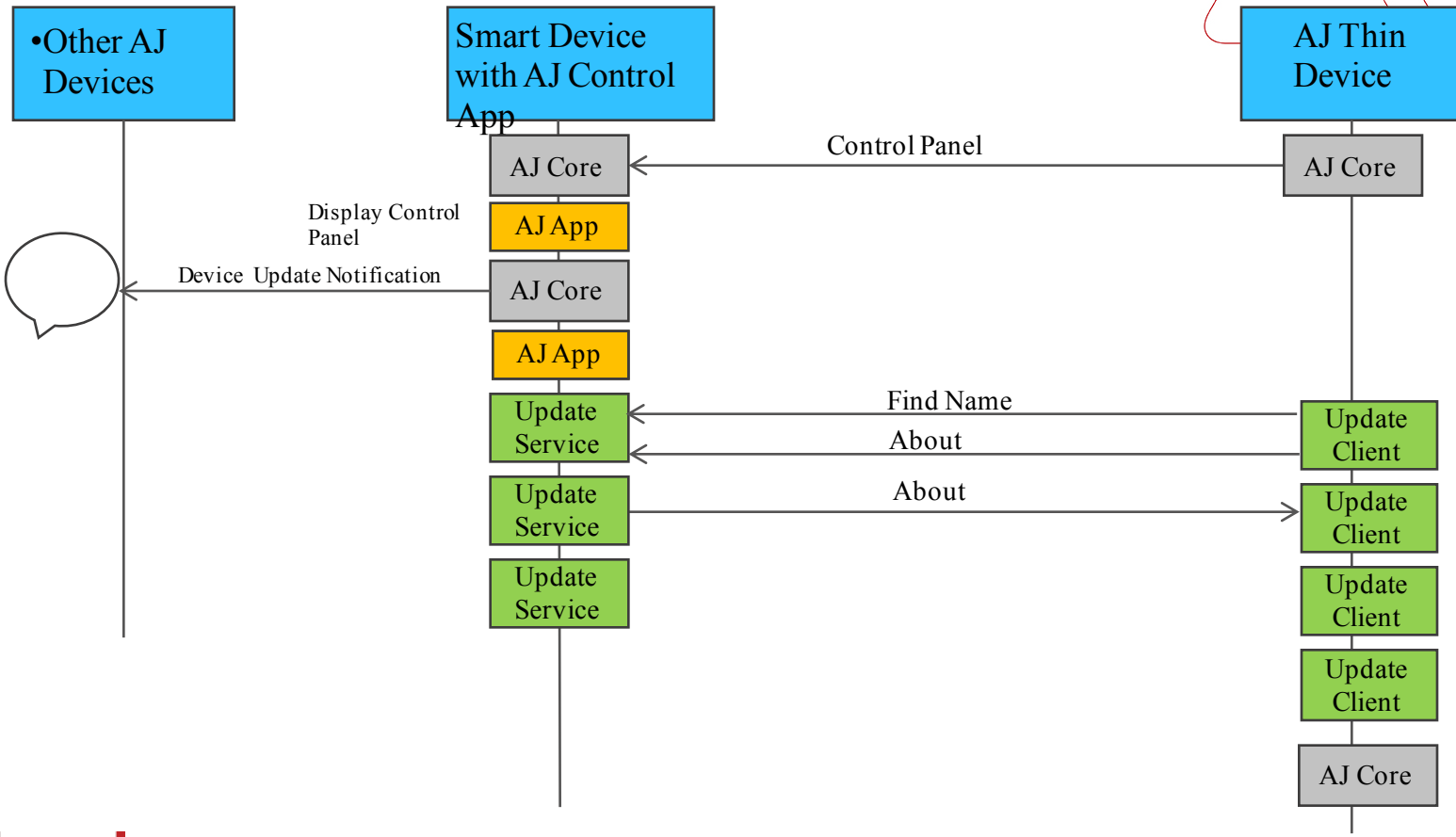
- AvailableUpdates({deviceid, oldfirmwareversion, newfirmwareversion }[])
- AutoUpdateMethod(ondemand, autodownload, autoupdate)

Methods

- UpdateAll
- UpdateFirmware(deviceid)
- HasFirmwareSource(deviceid)
- CheckForUpdate(deviceid)

Signals

- UpdateAvailable(deviceid, oldfirmwareversion, newfirmwareversion)
- UpdateReadyToInstall(deviceid, oldfirmwareversion, newfirmwareversion)
- UpdateInProgress(deviceid, updatestate)
- UpdateComplete(deviceid, oldfirmwareversion, newfirmwareversion)
- UpdateFailed(deviceid, oldfirmwareversion, newfirmwareversion, reason)



DEMO NOTES

- (on-boarding assumed already)
- Fridge sends ”Control Panel” to core AJ service
- Fridge sends “Find Name Update service”
- Fridge sends “About”
- Update availability notification (Update Service) to the Fridge
- Update availability general notification (using AllJoyn notification)
  - Text: “Fridge says: I have an important update. It was verified” …pls allow
- Update user acceptance
- Update progress (shown on other devices), LCD on Pi?
- Update complete notification (audio, visual)
- Control Panel/About- change viewed
  - [update of capability reflected in Control Panel]