NIPC

non-IP control

Bart Brinckman
Rohit Mohan
Braeden Sanford
Standardized API’s allow the network to act as a platform for all IoT applications.

Apps and devices can integrate with devices regardless of wireless technology of use case.

Open system, scales to many apps and devices.

- Proprietary stovepipes
  - Deep integration, heavy lift for every use case
  - Closed system, does not scale beyond a couple apps.

- Standards-based approach
  - Onboarding: app-based real-time/non-real-time
  - Control: Policy-based (defined in the network)
  - Telemetry/data: Structured API, data + meta-data

Standardized API’s allow the network to act as a platform for all IoT applications.

APPS can integrate with devices regardless of wireless technology of use case.

Open system, scales to many apps and devices.
Common APIs across radio technologies enables streamlined and accelerated integration.
Non IP Control (NIPC)

Perform operations on SCIM-provisioned devices or groups objects

- SCIM Object
  - User
  - Group
  - Device
  - Connectivity extensions
    - BLE
    - Zigbee
    - Thread
    - Wi-Fi
  - Application extensions
    - Control
    - Data
  - Security extension
    - MUD

- Connection
  - Binding
  - Connection

- Data
  - Attribute
  - Subscription

- Registration
  - Topic
  - File

NIPC operations
Example: BLE Advertisement (broadcast) from device A

**BLE device A**

**Access Point & Controller**

**Application**

- SCIM onboarding request {SCIM object Device A}
- SCIM onboarding response {SCIM object + Device ID}
- POST /registration/topic {topic=adv, type=advertisement, id(s)}
- /registration/topic response {SUCCESS}

- Subscribe to topic=adv

- Device A advertisement on topic=adv

- Device A advertisement

- Device A advertisement

- Device A advertisement on topic=adv

- Device A advertisement
Example: Attribute read/write from BLE device A

BLE device A

- BLE Connect [device MAC]
- BLE Connection established [device MAC, connection handle]
- BLE Service discovery [connection handle]
- BLE Service discovery [connection handle, services]
- BLE read [connection handle, characteristic]
- BLE read response [connection handle, characteristic, value]
- BLE write [connection handle, characteristic, value]
- BLE write response [connection handle, characteristic, value]
- BLE Disconnect [connection handle]
- BLE Disconnect response [connection handle]

Access Point & Controller

- POST /connectivity/connection [id]
- /connectivity/connection response [SUCCESS, id, services]
- GET /data/attribute [id, attribute]
- /data/attribute response [SUCCESS, id, attribute, value]
- PUT /data/attribute [id, attribute, value]
- /data/attribute response [SUCCESS, id, attribute, value]

Application

- DELETE /connectivity/connection [id]
- /connectivity/connection response [SUCCESS, id]
Example: Attribute read/write from Zigbee device B

Zigbee device B

Access Point & Controller

Application

Zigbee network binding {MAC address}
Zigbee network binding {MAC address, node-id, pan-id}
Zigbee read {MAC address, attribute}
Zigbee read response {MAC address, attribute, value}
Zigbee write {MAC address, attribute, value}
Zigbee write response {MAC address, attribute, value}

POST /connectivity/binding [id]
GET /data/attribute [id, attribute]
PUT /data/attribute [id, attribute, value]

/data/attribute response {SUCCESS, id, attribute, value}
/connectivity/binding response {SUCCESS, id, node-id, pan-id}
/data/attribute response {SUCCESS, id, attribute, value}

SCIM onboarding request {SCIM object Device B}
SCIM onboarding response {SCIM object + Device ID}