NIPC update

ASDF working group meeting @IETF123

Bart Brinckman, Rohit Mohan, Braeden Sanford

24/07/2025

Agenda

- Status of the draft
- Hackathon @IETF123
- Issues that require WG consensus
  - Query parameters instead of path parameters
  - Protocol Mapping
  - NIPC as a GW for IP-connected devices: Name change?
  - Problem types
  - Content types
  - Explicit connection management
  - SCIM SDF extension
  - IANA Considerations
- Issues that are ready for closure
- Next steps

NIPC status

- Draft-asdf-nipc-10 released!

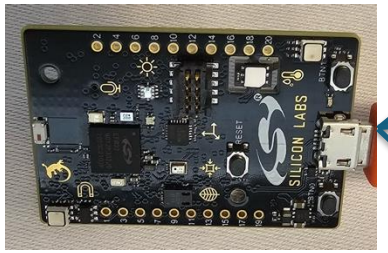
- https://www.ietf.org/archive/id/draft-ietf-asdf-nipc-10.html

- Changes since draft-09:
  - Protocol mapping naming
  - Registrations collection naming
  - Problem types
  - Property read/write single value
  - Content type application/nipc+json
  - Addressed mqttClient empty object
  - Various editorial updates

Hackathon Plan

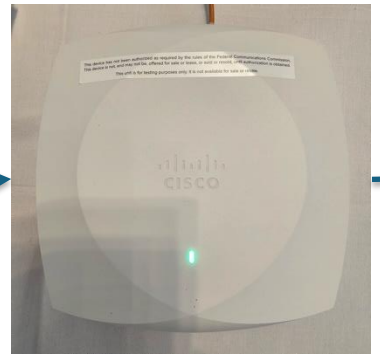
- Drafts involved:
  - draft-ietf-asdf-nipc
  - draft-ietf-asdf-sdf
  - draft-ietf-scim-device-model
- Review and solve issues in draft-ietf-asdf-nipc
- Interop with open-source and vendor implementations of an IoT Gateway

What got done - Interop Setup



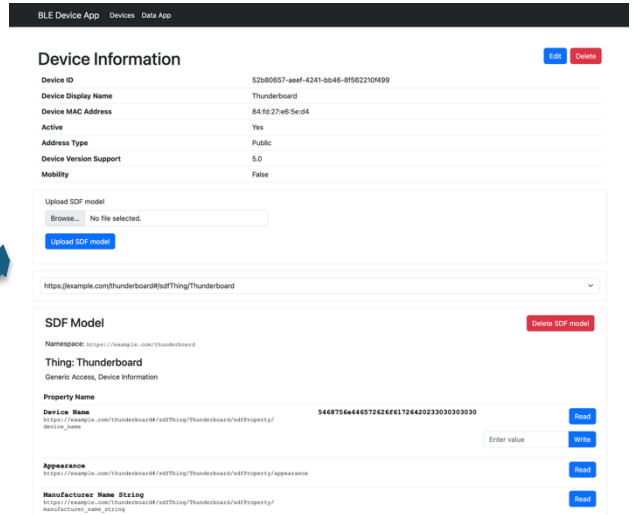
BLE device

Ericsson Research GW implementation



Cisco GW implementation

Open-source GW implementation



Control app

What else got done/What we learned

- Expand the scope of applicability of NIPC to IP devices
- Improvements on the SDF extensions and NIPC specification
  - NIPC protocol mapping in SDF
  - Defining NIPC problem types based on RFC 9457
  - 26 new issues on GitHub created and solved during the hackathon!
- Created tooling to create SDF models with NIPC protocol mapping
- Initial design for a SCIM extension for SDF information
- Hackathons are great for fast progress

Hackathon Wrap Up

Team members:

Ari Keränen

Bart Brinckman

Carsten Bormann

Jan Romann

Lorenzo Corneo

Michael Richardson

Rohit Mohan

First time @ hackathon

Lorenzo Corneo

Jan Romann

Learn more and contribute to the open-source:

https://github.com/iot-onboarding/tiedie

Read the drafts:

draft-ietf-asdf-nipc

draft-ietf-asdf-sdf (approved)

draft-ietf-scim-device-model

Issues that require WG consensus

Query parameters instead of path parameters

% encoding reserved characters should not be normalized per RFC 3986, so reserved characters can be used as path parameters in URIs.

Some server implementations take a cautious approach and do not allow percent encoded reserved characters, especially parameters used in URI paths such as '/' or '.' by default. They do this to avoid path traversal issues with improper normalization (not according to RFC 3986). This provides significant implementation friction and is an issue.

Example URL:

https://localhost/nipc/devices/12345678-1234-5678-1234-56789abcdef4/properties?propertyName=https%3A%2F%2Fexample.com%2Fheartrate%23%2FsdfObject%2Fthermostat%2FsdfProperty%2Ftemperature

Server implementation examples:

- Jetty: https://jakarta.ee/specifications/servlet/6.0/jakarta-servlet-spec-6.0#uri-path-canonicalization
- Tomcat: https://tomcat.apache.org/tomcat-9.0-doc/config/systemprops.html#Security(ALLOW_ENCODED_SLASH , defaults false)
- Apache: https://httpd.apache.org/docs/2.4/mod/core.html#allowencodedslashes (ALLOW_ENCODED_SLASH , defaults false)

Proposed solution: Leverage Query parameters instead of path parameters allows NIPC API to run on servers with default settings.

https://localhost/nipc/devices/12345678-1234-5678-1234-56789abcdef4/properties?propertyName=https://example.com/heartrate#/sdfObject/thermostat/sdfProperty/temperature

Issues that require WG consensus

Protocol mapping

As per SDF structure, ProtocolMap needs to be a reserved keyword, so instead of ‘protocolMap’, ‘sdfProtocolMap’ is proposed

```
"sdfProperty": {
  "heartrate": {
    "description": "The current measured heart rate",
    "type": "number",
    "unit": "beat/min",
    "observable": false,
    "writable": false,
    "protocolMap": {
      "ble": {
        "serviceID": "12345678-1234-5678-1234-56789abcdef4",
        "characteristicID": "12345678-1234-5678-1234-56789abcdef4"
      }
          }
     }
}
```

```
"sdfProperty": {
  "heartrate": {
    "description": "The current measured heart rate",
    "type": "number",
    "unit": "beat/min",
    "observable": false,
    "writable": false,
    "sdfProtocolMap": {
      "ble": {
        "serviceID": "12345678-1234-5678-1234-56789abcdef4",
        "characteristicID": "12345678-1234-5678-1234-56789abcdef4"
      }
          }
     }
}
```

Issues that require WG consensus
NIPC as a GW for IP-connected devices

SDF protocol map can be used to describe IP-based (e.g. HTTP) APIs as well

```
GET /heartrate/{id}/current
```

Could map to:

```
"sdfProperty": {
  "heartrate": {
    "sdfProtocolMap": {
      "openapi": {
        "operationRef": "https://example.com/openapi.json#/paths/~1heartrate~1{id}~1current/get",
                  "$ref": " https://example.com/openapi.json#/components/schema/HeartRate/properties/heartrate"
                }
          }
    }
}
```

- Should we change the name of the NIPC draft? “Non-Internet-Connected Physical Component”
- Should sdfProtocolMap be in the NIPC draft or a separate draft?

Issues that require WG consensus

NIPC problem types

RFC9457 defines the "HTTP Problem Types" registry as a registry for common, widely used problem type URIs, to promote reuse.

We leverage RFC9457 to define NIPC problem types, but they are not common widely used problem types, but specific to NIPC.

Therefore, we propose to create a new registry, the "NIPC Problem Type" registry for NIPC problem types, with following URL: https://www.iana.org/assignments/nipc-problem-types

Issues that require WG consensus
Content types

- Read and write properties
  - Use media types in the Content-type and Accept headers to represent the data to be read/written
  - PUT API: Property name is a query parameter, and the request body has the data
  - GET API: Property name is a query parameter, and the response body has the data
- This works for a single property
- To read/write multiple properties in the same API
  - Use the existing NIPC JSON format. The data is encoded as base64 with padding.

```
[
  {
    "property": "https://example.com/heartrate#/sdfObject/thermostat/sdfProperty/temperature",
    "value": "dGVzdA=="
  }
]
```

- This format uses a new media type that needs to be registered
  - application/nipc+json

Issues that require WG consensus
Explicit connection management

- Current API:

  /devices/{id}/manage/connection

- Is this inconsistent with other APIs which leverage collections? E.g:

  /devices/{id}/properties

- Suggestions?

Issues that require WG consensus

SCIM SDF extension

A NIPC server does not have to maintain a mapping between an instance and SDF models that apply to that instance, as every NIPC API call contains a globally unique reference to the SDF affordance it applies to.

Applications may however want to retrieve this information. As this is instance information, it needs to be retrieved from the SCIM server.

Proposed schema:

```
{
   "schemas": [
      "urn:ietf:params:scim:schemas:core:2.0:Device",
      "urn:ietf:params:scim:schemas:extension:ble:2.0:Device"
   ],
   "id": "e9e30dba-f08f-4109-8486-d5c6a3316111",
   "displayName": "BLE Heart Monitor",
   "active": true,
   "urn:ietf:params:scim:schemas:extension:ble:2.0:Device": {
      "deviceMacAddress": "2C:54:91:88:C9:E2",
      …
   },
   "urn:ietf:params:scim:schemas:extension:sdf:2.0:Device": {
      "sdf": [
         "https://example.com/thermometer#/sdfThing/thermometer",
         "https://example.com/heartrate#/sdfObject/healthsensor"
      ]
   }
```

In NIPC draft or separate SCIM draft?

Issues that require WG consensus
IANA considerations

- application/nipc+json media type registration
- New well-known URI registration: /.well-known/nipc
- New registries
  - NIPC protocol mapping registry: Specification required
  - NIPC API extensions registry: Specification required
  - NIPC Problem Type registry: Described in document + specification required

Issues that can be considered for closure

- sdfName instead of sdfRef when referencing a globalName #63: updated and published in draft-06
- Aasvg to figures & figure 2 consistency with SCIM draft #79: figures updated, figure 2 is consistent with SCIM but adds more detail
- .well_known #83 (+#67): updated in draft-09 & draft-10
- Async responses #87: The API should return a 202 Accepted with a Location header to the status endpoint which can be polled. Updated in draft-09
- path-based %-encoding #103: moved to query parameters in draft-09
- Discrepancy between APIs in the draft and OpenAPI model #111: fixed in draft-10
- figure 1 app #115: fixed in draft-10
- registrations path name #127: updated in draft-10
- supporting write and read single value #132: updated APIs and new media type in draft-10
- inconsistency in example in 4.2 #133: fixed in draft-10
- limitations on extension in 5.2 #136: fixed in draft-10
- problem types #138: new NIPC problem type in draft-10
- URI template for the API paths in the draft #141: updated in draft-10
- Mqttclient empty object #146: changed to Boolean in draft-10

Next steps

- Work on closing the rest of the existing issues
- More reviews from the WG (create issues on GitHub!)
- Interim meetings to discuss
- Propose experiment for IETF124 or later? Enable NIPC on the IETF network?