SDF & NIPC hackathon report

ASDF working group meeting

The ASDF hackathon team

Dublin 07/09/2024

The ASDF hackathon team

- Ari Keränen
- Bart Brinckman
- Braeden Sandford (remote)
- Carsten Bormann
- Eliot Lear
- Michael Richardson
- Niklas Meyer (remote)
- Rohit Mohan (remote)
- Sriram Sekar (remote)

Thank you for participating!!!

Hackathon Plan

Work on 2 classes of problems

- Converting models between eco-systems: SDF & Matter
- Using SDF in networking environments (NIPC & SCIM):
    - ASDF describes interaction of Things
    - SCIM for devices provides device onboarding
    - NIPC is an Application Gateway for Non-IP Protocols

    -> How can they interoperate?

IETF DRAFTS:

- draft-ietf-asdf-sdf
- draft-ietf-asdf-nipc
- draft-ietf-scim-device-model

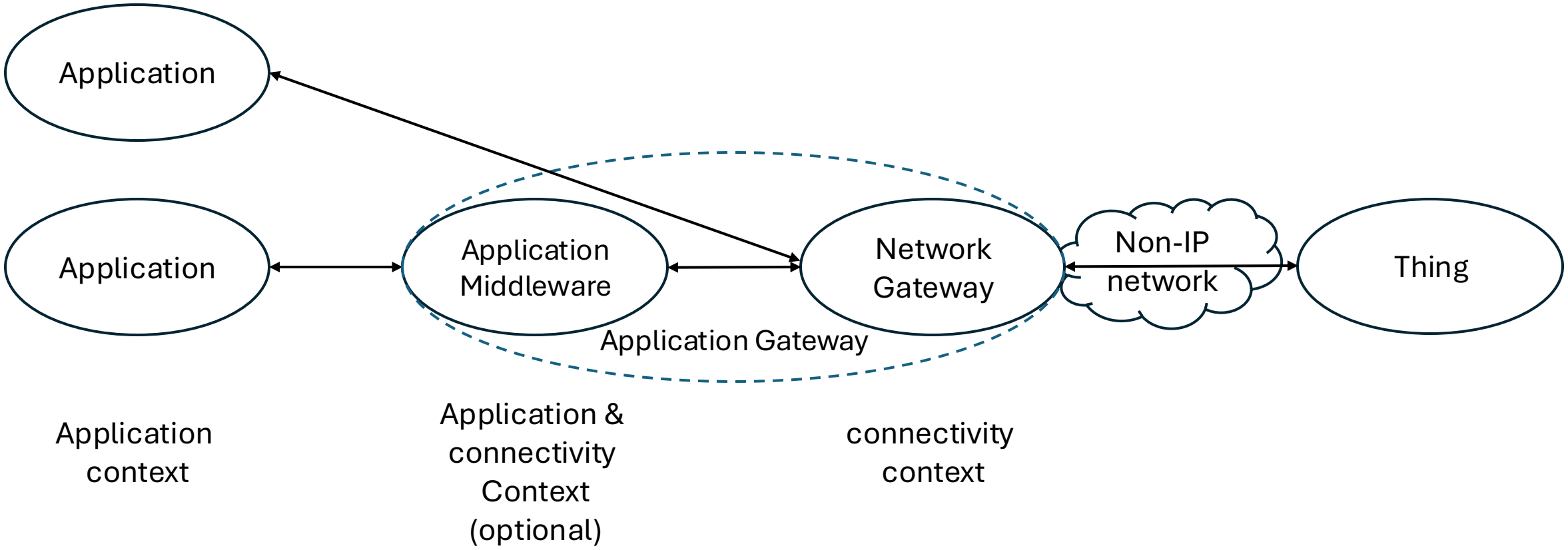
Matter ↔ SDF

Matter ↔ SDF

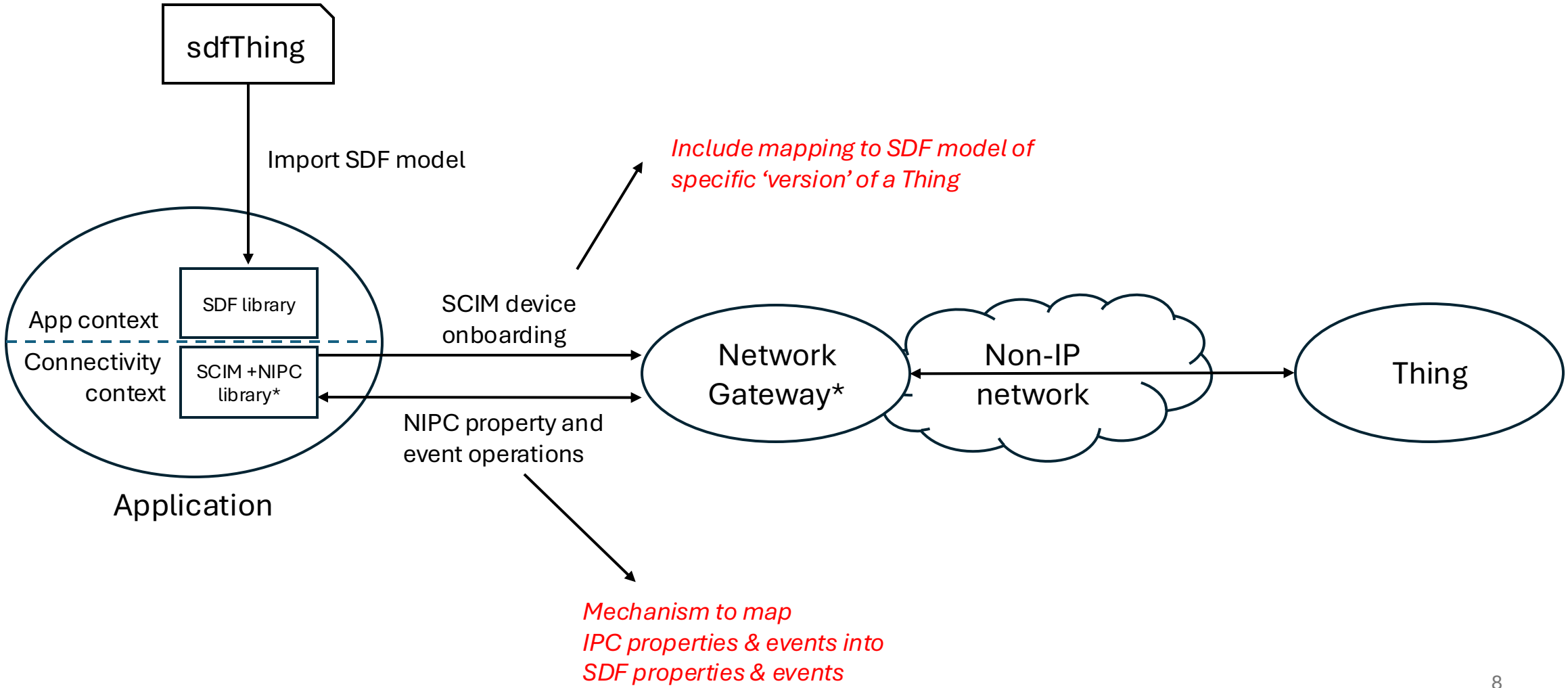
- Converter written by Niklas Meyer
- Achievements:
  - Verified: converter can be used by other ASDF people
  - Docker container for the conversion tool
- Work in Progress:
  - Ongoing: Document demo setup with physical Matter and non-Matter devices
    (Validating interoperation between LwM2M/IPSO and Matter via SDF)

NIPC in SDF

NIPC in SDF: Roles



Deployment model 1: Network Gateway



* SCIM Server + NIPC Gateway

Mapping NIPC properties and events to SDF properties & events

NIPC currently supports ‘registration’ APIs that map a NIPC property or event to a protocol-specific property

- **Example: POST /registration/property:**

```
{  "property": "temperature",
   "ble":
        {  "serviceID": "12345678-1234-5678-1234-56789abcdef4",
           "characteristicID": "12345678-1234-5678-1234-56789abcdef4",
        }
}
```

- **Example: POST /registration/event:**

```
{ "event": "enterprise/hospital/pulse_oximeter",
  "property": ”heartrate",
  "id": "12345678-1234-5678-1234-56789abcdef4",
  "ble":
        { "type": "gatt",
          "serviceID": "12345678-1234-5678-1234-56789abcdef0",
          "characteristicID": "12345678-1234-5678-1234-56789abcdef1"
        }
}
```

Mapping NIPC properties and events to SDF properties & events

Replacing individual registration APIs with single SDF model registration maps directly to SDF model and simplifies registration.

- a new NIPC quality with a NIPC protocol mapping for sdfProperties and sdfEvents in SDF
- NIPC gateway only needs to be able to interpret NIPC quality
- NIPC registration API registers an SDF model on a NIPC Gateway
- Application can send a subset of the model, e.g. only the required properties/events to perform NIPC operations
- In case the properties carry confidential information (e.g. patient heartrate), application can modify SDF model to obfuscate property or event names.

NIPC quality in sdfProperty

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

NIPC quality in sdfEvent

```
"sdfObject": {
    "healthsensor": {
      "sdfEvent": {
        "fallDetected": {
          "sdfOutputData": {
            "type": "boolean",
            "nipc": {
              "ble": {
                "type": "gatt",
                "serviceID": "12345678-1234-5678-1234-56789abcdef1",
                "characteristicID": "12345678-1234-5678-1234-56789abcdef1"
              }
            }
          }
        },
```

Mapping SDF model to a version of a thing?

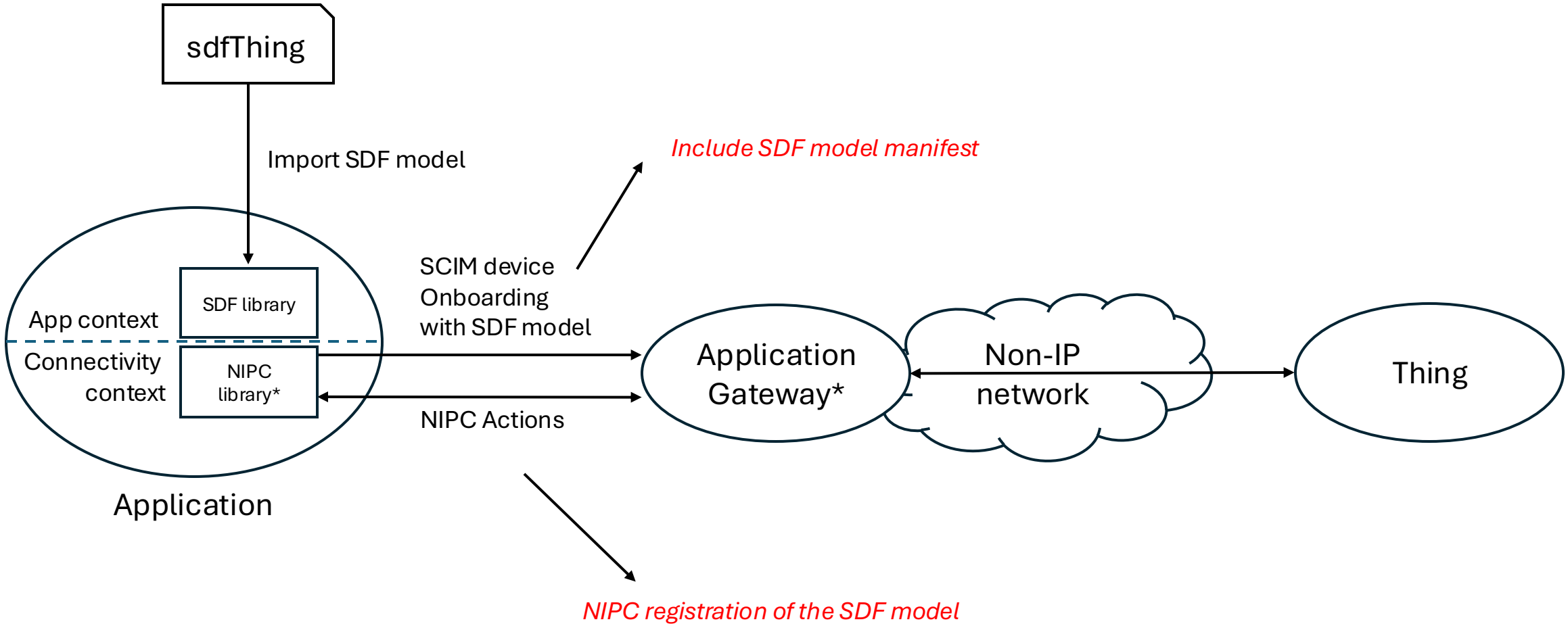
- Mapping device firmware/software version to an SDF model
- Retrieve appropriate SDF model based on that version
- Break it down into architectural building blocks
- There might be additional keys: firmware version, hardware version, app version,…

- Inclusion in SCIM
- Reference via MUD
- Direct import

Manifest

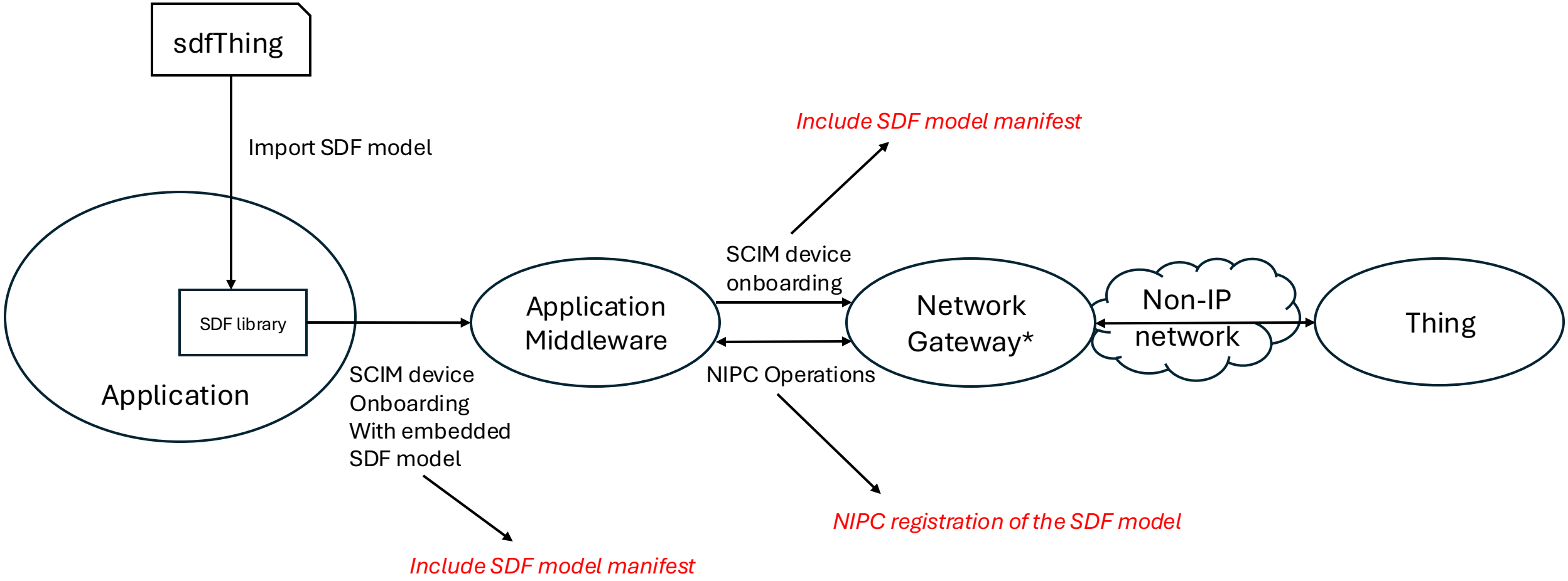
```
{
   manifest-version: 1,
   { firmware-version : 1.2,
     sdf-file : url1} ,
   {firmware-version: 1.3,
    sdf-file: url2}
}
```

Deployment model 2: Application Gateway



* SCIM Server + NIPC Gateway

Deployment model 3: Leveraging middleware



* SCIM Server + NIPC Gateway

Possible future projects

- Dummy NIPC Gateway Server, allowing developers to test against a few dummy devices
- Updating TieDie OpenSource
  - python/java libraries and Sample app
  - NIPC Gateway with BLE/Zigbee support
- SDF model (with NIPC quality) generation tool

Discussion