

SDF Extension for Non-Affordance Information

draft-hong-asdf-sdf-nonaffordance-01

– IETF ASDF WG –

Jungha Hong and Hyunjeong Lee
ETRI

Proposed Extension: sdfNonAffordance

- Introduce new class: sdfNonAffordance
- Attributes:
 - name: Identifier
 - description: Human-readable text
 - type: Data type (string, number, object, etc.)
 - unit: Optional unit if applicable

```
{
  "sdfObject": {
    "device": {
      "sdfNonAffordance": {
        "attribute-name": {
          "description": "Attribute description",
          "type": "data type",
          "unit": "unit if applicable"
        }
      }
    }
  }
}
```

Distinction from sdfProperty

Category	sdfProperty	sdfNonAffordance
Purpose	Direct user access / control of live data	Static or descriptive context (device metadata)
Semantics	Observable, measurable value (an affordance)	Non-interactive state / metadata
Typical Content	Temperature, humidity, switch state	Installation location, model name, certification data
Mutability	Dynamic; may change frequently or periodically	Mostly static; updates only when context changes
Access Pattern	Exposed via GET / PUT / observe interfaces	Read-only reference or UI meta-rendering only

SDF Extension for Non-Affordance Information

- Non-affordance information must be handled in two different life-cycle phases:
 - Design-time: when the model author decides which contextual fields a Thing exposes and how they are validated.
 - Run-time: when an actual device (or its digital twin) needs to convey current values for those fields, or report that they have changed.
- The extension is organized into two complementary parts:
 - 4.1 Static model Definition – shows how contextual metadata is embedded directly in the SDF document under the new sdfNonAffordance class. These definitions are authored once, validated like any other SDF schema fragment, and travel with the model wherever it is stored or published.
 - 4.2 Run-Time Context Messaging – introduces three JSON envelopes (contextSnapshot, identityManifest, contextPatch) that let a deployed device or its digital twin report changes to that metadata over time. Keeping these messages outside the affordance model preserves the core principle that non-affordance data is descriptive, not interactive, while still allowing systems to keep the context up-to-date.

Three envelopes for run-time context exchange

Envelope	Purpose	Notes
contextSnapshot	Convey the entire current non-affordance context.	Sent on boot, on demand, or scheduled audit.
identityManifest	contextPatch Declare immutable identity: model, manufacturer, capability tags, certificates.	Usually sent once at commissioning or upon firmware update.
contextPatch	Send only the fields that changed since the last snapshot.	Minimizes bandwidth when a subset of context is updated.

Table of Contents

- [1. Introduction](#)
- [2. Terminology and Conventions](#)
- [3. Motivation and Use Cases](#)
 - [3.1. Motivation](#)
 - [3.2. Use Cases](#)
- [4. SDF Extension for Non-Affordance Information](#)
 - [4.1. Concept](#)
 - [4.2. Syntax and Semantics](#)
 - [4.3. Examples](#)
- [5. Security Considerations](#)
- [6. IANA Considerations](#)
- [7. Normative References](#)
- [Authors' Addresses](#)

- 1. Introduction
- 2. Terminology and Conventions
- 3. Motivation and Use Cases
 - 3.1. Motivation
 - 3.2. Use Cases
- 4. SDF Extension for Non-Affordance Information
 - 4.1. Static Model Definition (sdfNonAffordance)
 - 4.2. Run-Time Context Messages
- 5. Security Considerations
- 6. IANA Considerations
- 7. Normative References
- Authors' Addresses

Example of static model definition (sdfNonAffordance)

```
{
  "sdfObject": {
    "device": {
      "description": "Environment sensor cluster",
      "sdfNonAffordance": {
        "installationInfo": {
          "description": "Fixed deployment context (non-
interactive).",
          "type": "object",
          "properties": {
            "floor": {
              "description": "Floor number where the sensor is
mounted.",
              "type": "integer",
              "minimum": 0
            },
            "mountType": {
              "description": "Mounting method (wall, ceiling,
pole...).",
              "type": "string",
              "enum": ["wall", "ceiling", "pole", "desktop"]
            }
          },
          "required": ["floor", "mountType"]
        }
      }
    }
  }
}
```

Example of contextSnapshot

```
{
  "contextSnapshot": {
    "thingId": "sensor-001",
    "timestamp": "2025-06-18T04:00:00Z",
    "context": {
      "installationInfo": {
        "floor": 3,
        "mountType": "wall"
      },
      "location": { "lat": 37.5665, "lon": 126.9780 }
    }
  }
}
```

Example of identityManifest

```
{
  "identityManifest": {
    "thingId": "sensor-001",
    "declaredAt": "2025-06-18T00:00:00Z",
    "manifest": {
      "deviceClass": "environment-sensor",
      "model": "KSX-200",
      "manufacturer": "KOSMOS Tech",
      "capabilityTags": ["temperature", "humidity"],
      "certifications": [
        { "scheme": "CE", "id": "CE-2024-12345" },
        { "scheme": "FCC", "id": "FCC-B-987654" }
      ]
    }
  }
}
```

Example of contextPatch

```
{
  "contextPatch": {
    "thingId": "sensor-001",
    "patchTime": "2025-06-20T11:30:00Z",
    "changes": {
      "installationInfo": {
        "mountType": "ceiling"
      },
      "location": {
        "lat": 37.5670,
        "lon": 126.9790
      }
    }
  }
}
```

Q&A

Thank you