ETSI GS CIM 054 V1.1.1 (2025-07)

GROUP SPECIFICATION

Context Information Management (CIM); NGSI-LD Interoperability tests specification

Disclaimer

The present document has been produced and approved by the cross-cutting Context Information Management (CIM) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference
DGS/CIM-0054

Keywords
API, IoT, NGSI-LD, testing

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI IPR online database.

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™**, **LTE™** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) cross-cutting Context Information Management (CIM).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Introduction

Several NGSI-LD Context Brokers and related software modules, such as data ingestion, event processing, and visualization, are being developed globally, reflecting the widespread adoption of NGSI-LD standards. With multiple open-source implementations available, establishing comprehensive interoperability test cases is essential to ensure that different brokers, even from various vendors, can operate together seamlessly. While these tests do not directly validate the NGSI-LD specification, these tests may also be used to demonstrate whether the implementations can interoperate effectively and whether the NGSI-LD specification should be revised to avoid ambiguity (if any). The results can be applied in specific scenarios to highlight system compatibility, showcase cross-vendor collaboration, and provide valuable feedback for improving overall implementation quality.

1 Scope

The present document extends the draft interoperability test descriptions agreed with ISG CIM.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found in the ETSI docbox.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI GS CIM 009 (V1.8.1): "Context Information Management (CIM); NGSI-LD API".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents may be useful in implementing an ETSI deliverable or add to the reader's understanding, but are not required for conformance to the present document.

Not applicable.

3 Definition of terms, symbols and abbreviations

3.1 Terms

Void.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

GS	Group Specification
SUT	System Under Test

4 Prerequisites and Test Configurations

4.1 Request body resources

4.1.1 Entities

4.1.1.0 Foreword

The following clauses provide a detailed description of the specific Entities that shall be employed to perform the interoperability tests outlined in clause 6 of the present document. Each clause contains the Entities of the given "type".

4.1.1.1 OffStreetParking

```
{
  "id": "urn:ngsi-ld:OffStreetParking:1",
  "type": "OffStreetParking",
  "name": {
    "type": "Property",
    "value": "Downtown One"
  },
  "availableSpotsNumber": {
    "type": "Property",
    "value": 121,
    "observedAt": "2017-07-29T12:05:02Z",
    "reliability": {
      "type": "Property",
      "value": 0.7
    }
  },
  "totalSpotsNumber": {
    "type": "Property",
    "value": 200
  },
  "location": {
    "type": "GeoProperty",
    "value": {
      "type": "Point",
      "coordinates": [
        -8.5,
        41.2
      ],
    "observedAt": "2021-04-04T11:41:00Z"
    }
  }
}
```

Figure 4.1.1.1-1: Full version of OffstreetParking:1

```
{
  "id": "urn:ngsi-ld:OffStreetParking:1",
  "type": "OffStreetParking",
  "name": {
    "type": "Property",
    "value": "Downtown One"
  },
  "availableSpotsNumber": {
    "type": "Property",
    "value": 169,
    "observedAt": "2017-07-29T12:10:02Z",
    "reliability": {
      "type": "Property",
      "value": 0.3
    }
  },
  "totalSpotsNumber": {
    "type": "Property",
    "value": 200
  }
}
```

Figure 4.1.1.1-2: OffstreetParking:1 without location

```
{
  "id": "urn:ngsi-ld:OffStreetParking:1",
  "type": "OffStreetParking",
  "name": {
    "type": "Property",
    "value": "Downtown One"
  },
  "location": {
    "type": "GeoProperty",
    "value": {
      "type": "Point",
      "coordinates": [
        -8.9,
        42.2
      ] ,
    "observedAt": "2021-04-04T11:45:00Z"
    }
  }
}
```

Figure 4.1.1.1-3: OffstreetParking:1 with location and name only

```
{
  "id": "urn:ngsi-ld:OffStreetParking:2",
  "type": "OffStreetParking",
  "name": {
    "type": "Property",
    "value": "Downtown Two"
  },
  "availableSpotsNumber": {
    "type": "Property",
    "value": 112,
    "observedAt": "2017-07-29T12:05:02Z",
    "reliability": {
      "type": "Property",
      "value": 0.4
    }
  },
  "totalSpotsNumber": {
    "type": "Property",
    "value": 150
  },
  "location": {
    "type": "GeoProperty",
    "value": {
      "type": "Point",
      "coordinates": [
        -8.45,
        41.2
      ]
    }
  }
}
```


Figure 4.1.1.1-4: Full version of OffstreetParking:2


```
{
  "id": "urn:ngsi-ld:OffStreetParking:2",
  "type": "OffStreetParking",
  "name": {
    "type": "Property",
    "value": "Downtown Two"
  },
  "availableSpotsNumber": {
    "type": "Property",
    "value": 87,
    "observedAt": "2017-07-29T12:10:02Z",
    "reliability": {
      "type": "Property",
      "value": 0.55
    }
  },
  "totalSpotsNumber": {
    "type": "Property",
    "value": 150
  }
}
```


Figure 4.1.1.1-5: OffstreetParking:2 without location

```
{
  "id": "urn:ngsi-ld:OffStreetParking:2",
  "type": "OffStreetParking",
  "name": {
    "type": "Property",
    "value": "Downtown Two"
  },
  "availableSpotsNumber": {
    "type": "Property",
    "value": 92,
    "observedAt": "2017-07-30T12:10:02Z",
    "reliability": {
      "type": "Property",
      "value": 0.56
    }
  }
}
```


Figure 4.1.1.1-6: OffstreetParking:2 without location and totalSpotsNumber


```
{
  "id": "urn:ngsi-ld:OffStreetParking:2",
  "type": "OffStreetParking",
  "name": {
    "type": "Property",
    "value": "Downtown Two"
  },
  "location": {
    "type": "GeoProperty",
    "value": {
      "type": "Point",
      "coordinates": [
        -8.46,
        41.25
      ] ,
    "observedAt": "2021-04-04T11:46:00Z"
    }
  }
}
```


Figure 4.1.1.1-7: OffstreetParking:2 with location and name only

4.1.1.2 Vehicle


```
{
    "id": "urn:ngsi-ld:Vehicle:1",
    "type": "Vehicle",
    "brandName": {
        "type": "Property",
        "value": "Brand1"
    },
    "speed": {
        "type": "Property",
        "value": 0,
        "source": {
            "type": "Property",
            "value": "Speedometer"
        }
    },
    "isParked": {
        "type": "Relationship",
        "object": "urn:ngsi-ld:OffStreetParking:1",
        "observedAt": "2017-07-29T12:00:04Z"
    },
    "location": {
      "type": "GeoProperty",
      "value": {
        "type": "Point",
        "coordinates": [
          -8.45,
          41.2
        ]
      }
    }
}
```


Figure 4.1.1.2-1: Full version of Vehicle:1


```
{
    "id": "urn:ngsi-ld:Vehicle:1",
    "type": "Vehicle",
    "brandName": {
        "type": "Property",
        "value": "Brand1"
    },
    "speed": {
        "type": "Property",
        "value": 0,
        "source": {
            "type": "Property",
            "value": "Speedometer"
        }
    },
    "isParked": {
        "type": "Relationship",
        "object": "urn:ngsi-ld:OffStreetParking:1",
        "observedAt": "2017-07-29T12:00:04Z"
    },
}
```


Figure 4.1.1.2-2: Vehicle:1 without location

```
{
    "id": "urn:ngsi-ld:Vehicle:2",
    "type": "Vehicle",
    "brandName": {
        "type": "Property",
        "value": "Brand2"
    },
    "speed": {
        "type": "Property",
        "value": 5,
        "source": {
            "type": "Property",
            "value": "Speedometer"
        }
    },
    "isParked": {
        "type": "Relationship",
        "object": "urn:ngsi-ld:OffStreetParking:2",
        "observedAt": "2017-07-30T12:00:04Z"
    },
    "location": {
      "type": "GeoProperty",
      "value": {
        "type": "Point",
        "coordinates": [
          -9.45,
          40.2
        ]
      }
    }
}
```


Figure 4.1.1.2-3: Full version of Vehicle:2


```
{
    "id": "urn:ngsi-ld:Vehicle:2",
    "type": "Vehicle",
    "brandName": {
        "type": "Property",
        "value": "Brand2"
    },
    "speed": {
        "type": "Property",
        "value": 10,
        "source": {
            "type": "Property",
            "value": "Speedometer"
        }
    },
    "isParked": {
        "type": "Relationship",
        "object": "urn:ngsi-ld:OffStreetParking:2",
        "observedAt": "2017-07-28T12:00:04Z"
    },
}
```


Figure 4.1.1.2-4: Vehicle:2 without location

4.1.2 Attribute Fragments

None.

4.1.3 CSourceRegistrations

4.1.3.0 Foreword

The following clauses provide a detailed description of the specific CSourceRegistrations that shall be employed to perform the interoperability tests outlined in clause 6 of the present document. Each clause contains the CSourceRegistrations of the given "mode".

NOTE: In all subsequential clauses, the "endpoint" member is represented containing the placeholder text "xxx". The text should be replaced with a valid endpoint URL during the set-up of a test configuration.

4.1.3.1 Inclusive

```
{
  "id": "urn:ngsi-ld:ContextSourceRegistration:Inclusive:1",
  "type": "ContextSourceRegistration",
  "information": [
    {
      "entities": [
        {
          "type": "OffStreetParking"
        }
      ],
    }
  ],
  "mode": "inclusive",
  "operations": ["redirectionOps"],
  "endpoint": "xxx"
}
```

Figure 4.1.3.1-1: Inclusive:1 on OffStreetParking entities, only type

```
{
  "id": "urn:ngsi-ld:ContextSourceRegistration:Inclusive:2",
  "type": "ContextSourceRegistration",
  "information": [
    {
      "entities": [
        {
          "type": "OffStreetParking"
        }
      ],
      "propertyNames": [
        "availableSpotsNumber",
        "totalSpotsNumber"
      ]
    }
  ],
  "mode": "inclusive",
  "operations": ["redirectionOps"],
  "endpoint": "xxx"
}
```

Figure 4.1.3.1-2: Inclusive:2 on OffStreetParking entities, type and attributes

```
{
  "id": "urn:ngsi-ld:ContextSourceRegistration:Inclusive:3",
  "type": "ContextSourceRegistration",
  "information": [
    {
      "entities": [
        {
          "type": "Vehicle"
        }
      ]
    }
  ],
  "mode": "inclusive",
  "operations": ["redirectionOps"],
  "endpoint": "xxx"
}
```

Figure 4.1.3.1-3: Inclusive:3 on Vehicle entities, only type

4.1.3.2 Auxiliary

```
{
  "id": "urn:ngsi-ld:ContextSourceRegistration:Auxiliary:1",
  "type": "ContextSourceRegistration",
  "information": [
    {
      "entities": [
        {
          "type": "OffStreetParking"
        }
      ]
    }
  ],
  "mode": "auxiliary",
  "endpoint": "xxx"
}
```

Figure 4.1.3.2-1: Auxiliary:1 on OffStreetParking entities, only type

```
{
  "id": "urn:ngsi-ld:ContextSourceRegistration:Auxiliary:2",
  "type": "ContextSourceRegistration",
  "information": [
    {
      "entities": [
        {
          "type": "OffStreetParking"
        }
      ],
      "propertyNames": [
        "availableSpotsNumber",
        "totalSpotsNumber"
      ]
    }
  ],
  "mode": "auxiliary",
  "endpoint": "xxx"
}
```

Figure 4.1.3.2-2: Auxiliary:2 on OffStreetParking entities, only type

```
{
  "id": "urn:ngsi-ld:ContextSourceRegistration:Auxiliary:3",
  "type": "ContextSourceRegistration",
  "information": [
    {
      "entities": [
        {
          "type": "Vehicle"
        }
      ]
    }
  ],
  "mode": "auxiliary",
  "endpoint": "xxx"
}
```

Figure 4.1.3.2-3: Auxiliary:3 on Vehicle entities, only type

4.1.3.3 Exclusive

```
{
  "id": "urn:ngsi-ld:ContextSourceRegistration:Exclusive:1",
  "type": "ContextSourceRegistration",
  "information": [
    {
      "entities": [
        {
          "id": "urn:ngsi-ld:OffStreetParking:2",
          "type": "OffStreetParking"
        }
      ],
      "propertyNames": [
        "totalSpotsNumber"
      ]
    }
  ],
  "mode": "exclusive",
  "operations": ["redirectionOps"],
  "endpoint": "xxx"
}
```

Figure 4.1.3.3-1: Exclusive:1 on OffStreetParking:2 and totalSpotsNumber

```
{
  "id": "urn:ngsi-ld:ContextSourceRegistration:Exclusive:2",
  "type": "ContextSourceRegistration",
  "information": [
    {
      "entities": [
        {
          "id": "urn:ngsi-ld:OffStreetParking:2",
          "type": "OffStreetParking"
        }
      ],
      "propertyNames": [
        "location"
      ]
    }
  ],
  "mode": "exclusive",
  "operations": ["redirectionOps"],
  "endpoint": "xxx"
}
```

Figure 4.1.3.3-2: Exclusive:2 on OffStreetParking:2 and location

```
{
  "id": "urn:ngsi-ld:ContextSourceRegistration:Exclusive:3",
  "type": "ContextSourceRegistration",
  "information": [
    {
      "entities": [
        {
          "id": "urn:ngsi-ld:OffStreetParking:1",
          "type": "OffStreetParking"
        }
      ],
      "propertyNames": [
        "location"
      ]
    }
  ],
  "mode": "exclusive",
  "operations": ["redirectionOps"],
  "endpoint": "xxx"
}
```

Figure 4.1.3.3-3: Exclusive:3 on OffStreetParking:1 and location

4.1.3.4 Redirect

```
{
  "id": "urn:ngsi-ld:ContextSourceRegistration:Redirect:1",
  "type": "ContextSourceRegistration",
  "information": [
    {
      "entities": [
        {
          "type": "OffStreetParking"
        }
      ]
    }
  ],
  "mode": "redirect",
  "operations": ["redirectionOps"],
  "endpoint": "xxx"
}
```

Figure 4.1.3.4-1: Redirect:1 on OffStreetParking entities, only type

```
{
  "id": "urn:ngsi-ld:ContextSourceRegistration:Redirect:2",
  "type": "ContextSourceRegistration",
  "information": [
    {
      "entities": [
        {
          "type": "OffStreetParking"
        }
      ],
      "propertyNames": [
        "location"
      ]
    }
  ],
  "mode": "redirect",
  "operations": ["redirectionOps"],
  "endpoint": "xxx"
}
```

Figure 4.1.3.4-2: Redirect:2 on OffStreetParking entities and location

```
{
  "id": "urn:ngsi-ld:ContextSourceRegistration:Redirect:3",
  "type": "ContextSourceRegistration",
  "information": [
    {
      "entities": [
        {
          "type": "OffStreetParking",
          "id": "urn:ngsi-ld:OffStreetParking:2"
        }
      ],
      "propertyNames": [
        "location"
      ]
    }
  ],
  "mode": "redirect",
  "operations": ["redirectionOps"],
  "endpoint": "xxx"
}
```

Figure 4.1.3.4-3: Redirect:3 on OffStreetParking:2 and location

```
{
  "id": "urn:ngsi-ld:ContextSourceRegistration:Redirect:4",
  "type": "ContextSourceRegistration",
  "information": [
    {
      "entities": [
        {
          "type": "Vehicle",
          "id": "urn:ngsi-ld:Vehicle:2"
        }
      ],
      "propertyNames": [
        "location"
      ]
    }
  ],
  "mode": "redirect",
  "operations": ["redirectionOps"],
  "endpoint": "xxx"
}
```

Figure 4.1.3.4-4: Redirect:4 on Vehicle:2 and location

4.2 Test Configurations

4.2.0 Foreword

The following clauses provide a detailed description of the specific test configurations that shall be employed to perform the interoperability tests outlined in clause 6 of the present document. Each clause includes a diagram illustrating the topology of the configuration, a formal description that explains the rationale for selecting the given configuration, and a comprehensive list of the Context Source Registrations required when setting up the configuration.

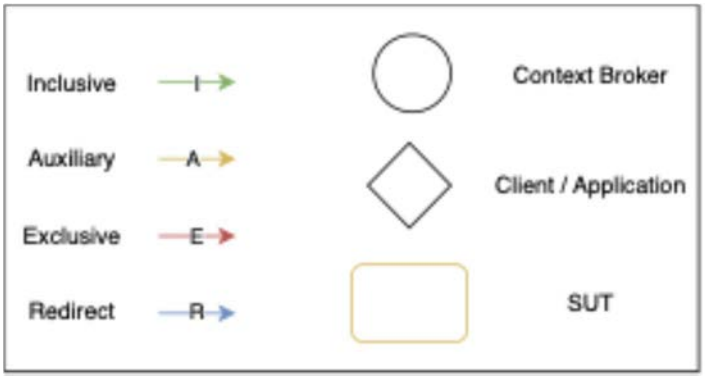

Figure 4.2.0-1: Legend

Figure 4.2.0-1 illustrates the legend for each element utilized in the diagrams of the configurations described in the present document. As depicted, registrations are represented as arcs, characterized by a color and a label that denote a specific Context Source Registration mode. Nodes are depicted as circles. Additionally, a dashed line is employed to indicate interactions with a specific node originating from an element external to the System Under Test (SUT), typically a client.

4.2.1 IOP_CNF_01: Three brokers with Inclusive and Exclusive

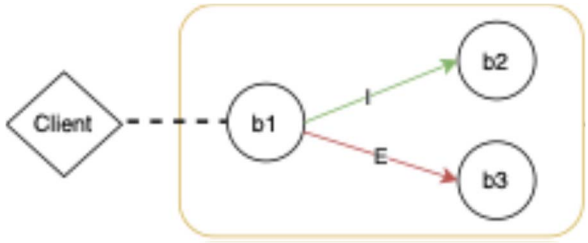

Figure 4.2.1-1: IOP_CNF_01 topology with registrations

Figure 4.2.1-1 shows the topology for IOP_CNF_01 which is one of the simplest possible setups, consisting of only three brokers. It is suitable for both Provision and Consumption operations, testing a scenario where broker b2 mirrors most of the information contained in b1 and b3 contains some exclusive data that cannot be stored in b1.

When this configuration is utilized, the following **registrations** shall be established:

- From b1 to b2, Inclusive
- From b1 to b3, Exclusive

Considering the above registrations, the following **restrictions** are to be expected:

- b1 cannot store any input data that matches against the Exclusive registration

The following **behaviours** are to be expected when a **Consumption** operation is triggered on b1:

- b1 searches locally when the input data **does not** match against the Exclusive registration established with b3
- b2 is contacted by b1when the input data matches against the Inclusive registration
- b3 is the **only** broker contacted by b1 when the input data matches against the Exclusive registration

The following **behaviours** are to be expected when a **Provision** operation is triggered on b1:

- b1 writes locally when the input data **does not** match against the Exclusive registration established with b3
- b2 is contacted by b1 when the input data matches against the Inclusive registration
- b3 is the **only** broker contacted by b1 when the input data matches against the Exclusive registration

4.2.2 IOP_CNF_02: Four brokers with Inclusive and Redirect

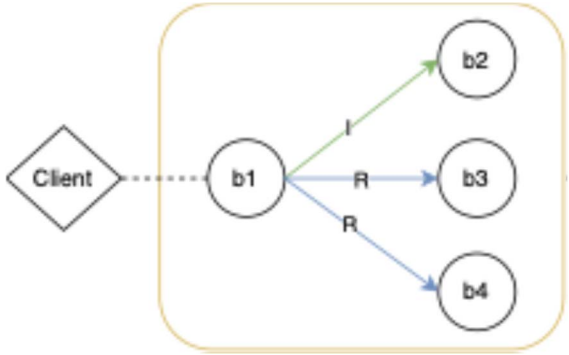

Figure 4.2.2-1: IOP_CNF_02 topology with registrations

Figure 4.2.2-1 shows the topology for IOP_CNF_02 which is a variant of IOP_CNF_01 with four brokers and Redirect registrations. It leverages the same approach, but it allows for more flexible tests. This is justified by the fact that Redirect registrations do not require to specify an Entity id, but can work with Entity types.

When this configuration is utilized, the following **registrations** shall be established:

- From b1 to b2, Inclusive
- From b1 to b3, Redirect
- From b1 to b4, Redirect

Considering the above registrations, the following **restrictions** are to be expected:

- b1 cannot store any input data that matches against the Redirect registration established with b3
- b1 cannot store any input data that matches against the Redirect registration established with b4

The following **behaviours** are to be expected when a **Consumption** operation is triggered on b1:

- b1 searches locally when the input data **does not** match against:
  - the Redirect registration established with b3
  - the Redirect registration established with b4
- b2 is contacted by b1when the input data matches against the Inclusive registration
- b3 is contacted by b1 when the input data matches against the Redirect registration
- b4 is contacted by b1 when the input data matches against the Redirect registration

The following **behaviours** are to be expected when a **Provision** operation is triggered on b1:

- b1 writes locally when the input data **does not** match against:
  - the Redirect registration established with b3
  - the Redirect registration established with b4
- b2 is contacted by b1when the input data matches against the Inclusive registration
- b3 is contacted by b1 when the input data matches against the Redirect registration
- b4 is contacted by b1 when the input data matches against the Redirect registration

4.2.3 IOP_CNF_03: Four brokers with Auxiliary and Inclusive

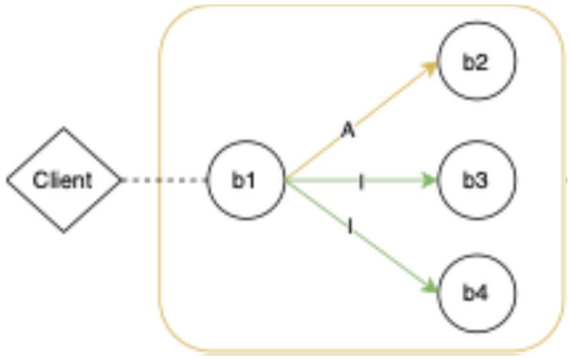

Figure 4.2.3-1: IOP_CNF_03 topology with registrations

Figure 4.2.3-1 shows the topology for IOP_CNF_03 which is a variant of IOP_CNF_02 with Auxiliary and Inclusive registrations. Considering the presence of an Auxiliary registration, this configuration is very suitable for testing Consumption operations. Provision operations can be tested as well, but the focus shifts towards a more "negative" approach where the objective is checking that an operation was **not** performed when specific requirements are met.

When this configuration is utilized, the following **registrations** shall be established:

- From b1 to b2, Auxiliary
- From b1 to b3, Inclusive
- From b1 to b4, Inclusive

The following **behaviours** are to be expected when a **Consumption** operation is triggered on b1:

- b1 searches locally
- b2 is contacted by b1 when the input data matches against the Auxiliary registration, but its data is retained **only** when all following points are met:
  - Data is not found locally within b1
  - Data is not returned by b3
  - Data is not returned by b4
- b3 is contacted by b1 when the input data matches against the Inclusive registration
- b4 is contacted by b1 when the input data matches against the Inclusive registration

The following **behaviours** are to be expected when a **Provision** operation is triggered on b1:

- b1 writes locally
- b2 is **not** contacted by b1
- b3 is contacted by b1 when the input data matches against the Inclusive registration
- b4 is contacted by b1 when the input data matches against the Inclusive registration

4.2.4 IOP_CNF_04: Five brokers with all registration modes

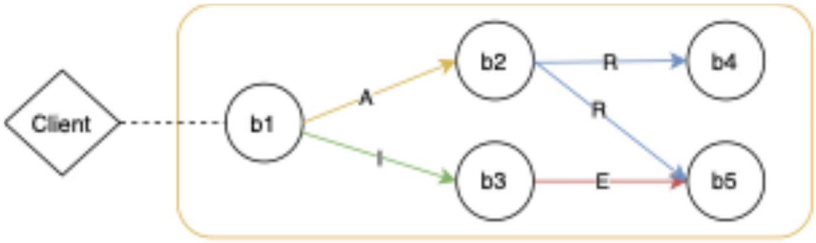

Figure 4.2.4-1: IOP_CNF_04 topology with registrations

Figure 4.2.4-1 shows the topology for IOP_CNF_04. This configuration contains all registration modes, enabling the execution of complex and comprehensive tests, particularly for evaluating Consumption operations. However, it is less suitable for Provision operations due to the presence of the Auxiliary registration from b1 to b2. In the case of a Provision request, b1 will not contact b2, which implies that nodes b2, b4, and potentially b5 (depending on the specific request) will remain unaffected.

When this configuration is utilized, the following **registrations** shall be established:

- From b1 to b2, Auxiliary
- From b2 to b4, Redirect
- From b2 to b5, Redirect
- From b1 to b3, Inclusive
- From b3 to b5, Exclusive

Considering the above registrations, the following **restrictions** are to be expected:

- b2 cannot store any input data that matches against the Redirect registration established with b4
- b2 cannot store any input data that matches against the Redirect registration established with b5
- b3 cannot store any input data that matches against the Exclusive registration established with b5

The following **behaviours** are to be expected when a **Consumption** operation is triggered on b1:

- b1 searches locally
- b2 is contacted by b1 when the input data matches against the Auxiliary registration, but its data is retained **only** when all following points are met:
  - Data is not found locally within b1
  - Data is not returned by b3
- When b2 is contacted by b1, the following behaviours are expected:
  - b2 searches locally when the input data **does not** match against:
    - the Redirect registration established with b4
    - the Redirect registration established with b5
  - b4 is contacted by b2 when the input data matches against the Redirect registration
  - b5 is contacted by b2 when the input data matches against the Redirect registration
- b3 is contacted by b1 when the input data matches against the Inclusive registration
- When b3 is contacted by b1:
  - b3 searches locally when the input data **does not** match against the Exclusive registration established with b5
  - b5 is contacted by b3 when the input data matches against the Exclusive registration

The following **behaviours** are to be expected when a **Provision** operation is triggered on b1:

- b1 writes locally
- b2 is **not** contacted by b1. Thus:
  - b4 is **not** contacted by b2
  - b5 is **not** contacted by b2
- b3 is contacted by b1 when the input data matches against the Inclusive registration. Thus:
  - b3 writes locally when the input data **does not** match against the Exclusive registration established with b5
  - b5 is contacted by b3 **when** b1 contacts b2 **and** the input data matches against the Exclusive registration

4.3 Initial Conditions

4.3.1 Initial data

4.3.1.1 IOP_INIT_COND_DATA_01: No data in any brokers

All NGSI-LD brokers shall start in a clean state with no pre-existing context data. This setup is strongly related to tests involving Provision operations (e.g. Create Entity).

4.3.1.2 IOP_INIT_COND_DATA_02: Data in the local broker

Only the local broker (the one directly contacted by the client) shall start with pre-existing context data. This setup can support both Provision and Consumption operations, but it is particularly suggested for testing the merge mechanism when provisioning context data.

4.3.1.3 IOP_INIT_COND_DATA_03: Data in brokers that are a leaf

Only the leaf brokers in the architecture shall start with pre-existing context data. This setup can support both Provision and Consumption operations, but it is particularly suggested for testing the merge mechanism when consuming context data.

4.3.1.4 IOP_INIT_COND_DATA_04: Data in all brokers

All NGSI-LD brokers shall start with pre-existing context data. This setup is strongly related to tests involving Consumption operations (e.g. Query Entities).

4.3.2 @context file

4.3.2.1 IOP_INIT_COND_CONTEXT_01: Core context only

When this configuration is applied, user contexts shall not be included in the HTTP requests in any way. In other words, embedding a user context in either the body of the NGSI-LD payload or in the LINK header is not allowed.

4.3.2.2 IOP_INIT_COND_CONTEXT_02: User context

When this configuration is applied, user contexts shall be included in the HTTP requests. In other words, it is required to embed a user context either in the body of the NGSI-LD payload or in the Link header.

The following user context shall be used when this initial condition is included:

```
"@context": [
   {
        "OffStreetParking": "https://ngsi-ld-test-suite/context#OffStreetParking",
        "Vehicle": "https://ngsi-ld-test-suite/context#Vehicle",
        "availableSpotsNumber": "https://ngsi-ld-test-suite/context#availableSpotsNumber",
        "brandName": "https://ngsi-ld-test-suite/context#brandName",
        "isParked": "https://ngsi-ld-test-suite/context#isParked",
        "name": "https://ngsi-ld-test-suite/context#name",
        "source": "https://ngsi-ld-test-suite/context#source",
        "speed": "https://ngsi-ld-test-suite/context#speed",
        "totalSpotsNumber": "https://ngsi-ld-test-suite/context#totalSpotsNumber",
   },
    "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context-v1.6.jsonld"
]
```

5 Interoperability Test Structure (ITS)

5.1 Test Case Structure

This clause contains the structure that shall be used for drafting the test cases included in clause 6 of the present document.

It follows the list of the Distributed Operations defined in ETSI GS CIM 009 [1] that can be chosen as an <Operation>:

- Create Entity
- Retrieve Entity
- Query Entities
- Merge Entity
- Replace Entity
- Delete Entity
- Update Entity Attribute
- Append Entity Attribute
- Partial Attribute Update
- Replace Attribute
- Delete Entity Attribute

Each <Test Case> consists of:

- A list of human-readable pre-conditions, describing the initial conditions that one shall comply to before executing the steps of a test case. Pre-conditions shall be taken from clause 4.3.
- A list of human-readable steps, describing each step that one shall follow when executing the test case. A test case is considered passed when all steps pass successfully.

Within the list of human-readable steps, the word "Agent" is used to refer either:

- A human performing and evaluating the test manually
- A software performing and evaluating the test programmatically

6 Test Descriptions

6.0 Foreword

The following clauses provide a detailed description of each test case.

Each subclause is structured as it follows:

- <Operation>
  - <Test Configuration>
    - <Test Case>

Where:

- <Operation> is the name of the specific Distributed Operation to test
- <Test Configuration> is the code of a test configuration taken from clause 4.2
- <Test Case> is a brief description of the end-goal of the test case

6.1 Create Entity

6.1.1 IOP_CNF_01

6.1.1.1 Create OffStreetParking:1

Pre-conditions:

- No user context
- No data in any broker
- Registrations that shall be established:
  - From b1 to b2, Inclusive:2 (Figure 4.1.3.1-2)
  - From b1 to b3, Exclusive:2 (Figure 4.1.3.3-2)

Steps:

1) Client sends an HTTP POST request to b1 to create the Full version of OffstreetParking:1 (Figure 4.1.1.1-1)
2) Agent checks that a success response has been returned
3) Agent checks (with local=true) that the full Entity was created in b1
4) Agent checks (with local=true) that the Entity was created in b2 and that the Entity only contains "availableSpotsNumber" and "totalSpotsNumber"
5) Agent checks (with local=true) that the Entity was NOT created in b3

6.1.1.2 Create OffStreetParking:2

Pre-conditions:

- No user context
- No data in any broker
- Registrations that shall be established:
  - From b1 to b2, Inclusive:2 (Figure 4.1.3.1-2)
  - From b1 to b3, Exclusive:2 (Figure 4.1.3.3-2)

Steps:

1) Client sends an HTTP POST request to b1 to create the Full version of OffstreetParking:2 (Figure 4.1.1.1-4)
2) Agent checks that a success response has been returned
3) Agent checks (with local=true) that the Entity was created in b1 and that the Entity does not contain the "location" Property
4) Agent checks (with local=true) that the Entity was created in b2 and that the Entity does not contain the "location" Property

5) Agent checks (with local=true) that the Entity was created in b3 and that the Entity contains only the "location" Property

6.1.2 IOP_CNF_02

6.1.2.1 Create OffStreetParking:1

Pre-conditions:

- No user context
- No data in any broker
- Registrations that shall be established:
  - From b1 to b2, Inclusive:2 (Figure 4.1.3.1-2)
  - From b1 to b3, Redirect:2 (Figure 4.1.3.4-2)
  - From b1 to b4, Redirect:2 (Figure 4.1.3.4-2)

Steps:

1) Client sends an HTTP POST request to b1 to create the Full version of OffstreetParking:1 (Figure 4.1.1.1-1)
2) Agent checks that a success response has been returned
3) Agent checks (with local=true) that the Entity was created in b1 and that the Entity does not contain the "location" Property
4) Agent checks (with local=true) that the Entity was created in b2 and that the Entity does not contain the "location" and "name" Properties
5) Agent checks (with local=true) that the Entity was created in b3 and that the Entity contains only the "location" Property
6) Agent checks (with local=true) that the Entity was created in b4 and that the Entity contains only the "location" Property

6.1.2.2 Create OffStreetParking:2

Pre-conditions:

- No user context
- No data in any broker
- Registrations that shall be established:
  - From b1 to b2, Inclusive:2 (Figure 4.1.3.1-2)
  - From b1 to b3, Redirect:2 (Figure 4.1.3.4-2)
  - From b1 to b4, Redirect:3 (Figure 4.1.3.4-2)

Steps:

1) Client sends an HTTP POST request to b1 to create the Full version of OffstreetParking:2 (Figure 4.1.1.1-4)
2) Agent checks that a success response has been returned
3) Agent checks (with local=true) that the Entity was created in b1 and that the Entity does not contain the "location" Property

4) Agent checks (with local=true) that the Entity was created in b2 and that the Entity does not contain the "location" and "name" Properties

5) Agent checks (with local=true) that the Entity was created in b3 and that the Entity contains only the "location" Property

6) Agent checks (with local=true) that the Entity was created in b4 and that the Entity contains only the "location" Property

6.1.3 IOP_CNF_03

6.1.3.1 Create OffStreetParking:1

Pre-conditions:

- No user context
- No data in any broker
- Registration that shall be established:
  - From b1 to b2, Auxiliary:2 (Figure 4.1.3.2-2)
  - From b1 to b3, Inclusive:1 (Figure 4.1.3.1-1)
  - From b1 to b4, Inclusive:2 (Figure 4.1.3.1-2)

Steps:

1) Client sends an HTTP POST request to b1 to create the Full version of OffstreetParking:1 (Figure 4.1.1.1-3)
2) Agent checks that a success response has been returned
3) Agent checks (with local=true) that the Entity was created in b1
4) Agent checks (with local=true) that the Entity was NOT created in b2
5) Agent checks (with local=true) that the Entity was created in b3
6) Agent checks (with local=true) that the Entity was created in b4 and that it only contains "availableSpotsNumber" and "totalSpotsNumber"

6.1.3.2 Create Partial OffStreetParking:1

Pre-conditions:

- No user context
- No data in any broker
- Registration that shall be established:
  - From b1 to b2, Auxiliary:2 (Figure 4.1.3.2-2)
  - From b1 to b3, Inclusive:1 (Figure 4.1.3.1-1)
  - From b1 to b4, Inclusive:2 (Figure 4.1.3.1-2)

Steps:

1) Client sends an HTTP POST request to b1 to create OffstreetParking:1 with location and name only (Figure 4.1.1.1-3)
2) Agent checks that a success response has been returned

3) Agent checks (with local=true) that the Entity was created in b1
4) Agent checks (with local=true) that the Entity was NOT created in b2
5) Agent checks (with local=true) that the Entity was created in b3
6) Agent checks (with local=true) that the Entity was NOT created in b4

6.1.4 IOP_CNF_04

6.1.4.1 Create OffStreetParking:1

Pre-conditions:

- No user context
- No data in any broker
- Registrations that shall be established:
  - From b1 to b2, Auxiliary:2 (Figure 4.1.3.2-2)
  - From b2 to b4, Redirect:1 (Figure 4.1.3.4-1)
  - From b2 to b5, Redirect:2 (Figure 4.1.3.4-2)
  - From b1 to b3, Inclusive:1 (Figure 4.1.3.1-1)
  - From b3 to b5, Exclusive:1 (Figure 4.1.3.3-1)

Steps:

1) Client sends an HTTP POST request to b1 to create the Full version of OffstreetParking:1 (Figure 4.1.1.1-1)
2) Agent checks that a success response has been returned
3) Agent checks (with local=true) that the Entity was created in b1
4) Agent checks (with local=true) that the Entity was NOT created in b2
5) Agent checks (with local=true) that the Entity was created in b3
6) Agent checks (with local=true) that the Entity was NOT created in b4
7) Agent checks (with local=true) that the Entity was NOT created in b5

6.1.4.2 Create OffStreetParking:2

Pre-conditions:

- No user context
- No data in any broker
- Registrations that shall be established:
  - From b1 to b2, Auxiliary:2 (Figure 4.1.3.2-2)
  - From b2 to b4, Redirect:1 (Figure 4.1.3.4-1)
  - From b2 to b5, Redirect:2 (Figure 4.1.3.4-2)
  - From b1 to b3, Inclusive:1 (Figure 4.1.3.1-1)
  - From b3 to b5, Exclusive:1 (Figure 4.1.3.3-1)

Steps:

1) Client sends an HTTP POST request to b1 to create the Full version of OffstreetParking:2 (Figure 4.1.1.1-4)
2) Agent checks that a success response has been returned
3) Agent checks (with local=true) that the Entity was created in b1
4) Agent checks (with local=true) that the Entity was NOT created in b2
5) Agent checks (with local=true) that the Entity was created in b3
6) Agent checks (with local=true) that the Entity was NOT created in b4
7) Agent checks (with local=true) that the Entity was created in b5 and that the Entity only contains the "totalSpotsNumber" Property

6.2 Retrieve Entity

6.2.1 IOP_CNF_01

6.2.1.1 Retrieve OffStreetParking:1

Pre-conditions:

- No user context
- Data only on leaves
  - b2 contains:
    - Full version of OffstreetParking:1 (Figure 4.1.1.1-1)
    - Full version of OffstreetParking:2 (Figure 4.1.1.1-4)
  - b3 contains:
    - Full version of OffstreetParking:2 (Figure 4.1.1.1-4)
- Registrations that shall be established:
  - From b1 to b2, Inclusive:2 (Figure 4.1.3.1-2)
  - From b1 to b3, Exclusive:2 (Figure 4.1.3.3-2)

Steps:

1) Client sends an HTTP GET request to b1 to retrieve the OffStreetParking:1
2) Agent checks that a success response has been returned and that only availableSpotsNumber and totalSpotsNumber of OffStreetParking:1 were returned
3) Client sends an HTTP GET request to b2 to retrieve the OffStreetParking:1
4) Agent checks that the Entity returned in step (3) is the full Entity

6.2.1.2 Retrieve OffStreetParking:2

Pre-conditions:

- No user context
- Data only on leaves
  - b2 contains:
    - OffstreetParking:2 without location (Figure 4.1.1.1-5)
  - b3 contains:
    - Full version of OffstreetParking:2 (Figure 4.1.1.1-4)
- Registrations that shall be established:
  - From b1 to b2, Inclusive:2 (Figure 4.1.3.1-2)
  - From b1 to b3, Exclusive:2 (Figure 4.1.3.3-2)

Steps:

1) Client sends an HTTP GET request to b1 to retrieve the Entity OffStreetParking:2
2) Agent checks that a success response has been returned and that the Entity OffStreetParking:2 was returned
3) Client sends an HTTP GET request to b2 to retrieve the Entity OffStreetParking:2
4) Client sends an HTTP GET request to b3 to retrieve the Entity OffStreetParking:2
5) Agent checks that the Entity returned in step (2) is structured as it follows:
   a) The attributes "availableSpotsNumber" and "totalSpotsNumber" match the ones from the Entity returned in step (3)
   b) The attribute "location" matches the one from the Entity returned in step (4)

6.2.2 IOP_CNF_02

6.2.2.1 Retrieve OffStreetParking:1

Pre-conditions:

- No user context
- Data only on leaves
  - b2 contains:
    - OffstreetParking:1 without location (Figure 4.1.1.1-2)
  - b3 contains:
    - Full version of OffstreetParking:1 (Figure 4.1.1.1-1)
  - b4 contains:
    - Full version of OffstreetParking:2 (Figure 4.1.1.1-4)

- Registrations that shall be established:
  - From b1 to b2, Inclusive:2 (Figure 4.1.3.1-2)
  - From b1 to b3, Redirect:2 (Figure 4.1.3.4-2)
  - From b1 to b4, Redirect:2 (Figure 4.1.3.4-2)

Steps:

1) Client sends an HTTP GET request to b1 to retrieve the Entity OffStreetParking:1
2) Agent checks that a partial success response has been returned and that the Full version of OffStreetParking:1 was returned
3) Client sends an HTTP GET request to b2 to retrieve the Entity OffStreetParking:1
4) Client sends an HTTP GET request to b3 to retrieve the Entity OffStreetParking:1
5) Agent checks that the Entity returned in step (2) is structured as it follows:
   a) The attributes "availableSpotsNumber" and "totalSpotsNumber" match the ones from the Entity returned in step (3)
   b) The attribute "location" matches the one from the Entity returned in step (4)

6.2.2.2 Retrieve location from OffStreetParking:1

Pre-conditions:

- No user context
- Data only on leaves
  - b2 contains:
    - Full version of OffstreetParking:1 (Figure 4.1.1.1-1)
  - b3 contains:
    - OffstreetParking:1 with location and name only (Figure 4.1.1.1-3)
  - b4 contains:
    - Full version of OffstreetParking:2 (Figure 4.1.1.1-4)
- Registrations that shall be established:
  - From b1 to b2, Inclusive:2 (Figure 4.1.3.1-2)
  - From b1 to b3, Redirect:2 (Figure 4.1.3.4-2)
  - From b1 to b4, Redirect:2 (Figure 4.1.3.4-2)

Steps:

1) Client sends an HTTP GET request to b1 to retrieve the "location" Property from the Entity OffStreetParking:1
2) Agent checks that a partial success response has been returned and that OffStreetParking:1 with the "location" Property was returned
3) Client sends an HTTP GET request (with local=true) to b3 to retrieve the Entity OffStreetParking:1
4) Agent checks that the Entity returned in step (2) contains the attribute "location" and that it matches the one from the Entity returned in step (3)

6.2.3 IOP_CNF_03

6.2.3.1 Retrieve OffStreetParking:1

Pre-conditions:

- No user context
- Data on every broker
  - b1 contains:
    - OffstreetParking:1 without location (Figure 4.1.1.1-2)
  - b2 contains:
    - Full version of OffstreetParking:1 (Figure 4.1.1.1-1)
  - b3 contains:
    - OffstreetParking:1 without location (Figure 4.1.1.1-2)
  - b4 contains:
    - Full version of OffstreetParking:2 (Figure 4.1.1.1-4)
- Registration that shall be established:
  - From b1 to b2, Auxiliary:2 (Figure 4.1.3.2-2)
  - From b1 to b3, Inclusive:1 (Figure 4.1.3.1-1)
  - From b1 to b4, Inclusive:2 (Figure 4.1.3.1-2)

Steps:

1) Client sends an HTTP GET request to b1 to retrieve the Entity OffStreetParking:1
2) Agent checks that a partial success response has been returned and that the Entity OffstreetParking:1 without the "location" Property was returned
3) Client sends an HTTP GET request to b1 (with local=true) to retrieve the Entity OffStreetParking:1
4) Client sends an HTTP GET request to b2 (with local=true) to retrieve the Entity OffStreetParking:1
5) Client sends an HTTP GET request to b3 (with local=true) to retrieve the Entity OffStreetParking:1
6) Agent checks that the Entity returned in step (2) matches the one returned in step (3) and that it does not contain the "location" Property returned in step (4)

6.2.3.2 Retrieve location from OffStreetParking:1

Pre-conditions:

- No user context
- Data on every broker
  - b1 contains:
    - OffstreetParking:1 without location (Figure 4.1.1.1-2)
  - b2 contains:
    - Full version of OffstreetParking:1 (Figure 4.1.1.1-1)

- b3 contains:
  - OffstreetParking:1 with location and name only (Figure 4.1.1.1-3)
- b4 contains:
  - OffstreetParking:1 without location (Figure 4.1.1.1-2)

- Registration that shall be established:
  - From b1 to b2, Auxiliary:1 (Figure 4.1.3.2-1)
  - From b1 to b3, Inclusive:1 (Figure 4.1.3.1-1)
  - From b1 to b4, Inclusive:2 (Figure 4.1.3.1-2)

Steps:

1) Client sends an HTTP GET request to b1 to retrieve the Entity OffStreetParking:1
2) Agent checks that a partial success response has been returned and that the Entity OffstreetParking:1 containing only the "location" Property was returned
3) Client sends an HTTP GET request to b3 (with local=true) to retrieve the Entity OffStreetParking:1
4) Agent checks that the Entity returned in step (2) contains the attribute "location" and that it matches the one from the Entity returned in step (3)

6.2.4 IOP_CNF_04

6.2.4.1 Retrieve OffStreetParking:1

Pre-conditions:

- No user context
- Data only on leaves
  - b4 contains:
    - OffstreetParking:1 without location (Figure 4.1.1.1-2)
  - b5 contains:
    - Full version of OffstreetParking:1 (Figure 4.1.1.1-1)
- Registrations that shall be established:
  - From b1 to b2, Auxiliary:2 (Figure 4.1.3.2-2)
  - From b2 to b4, Redirect:1 (Figure 4.1.3.4-1)
  - From b2 to b5, Redirect:2 (Figure 4.1.3.4-2)
  - From b1 to b3, Inclusive:1 (Figure 4.1.3.1-1)
  - From b3 to b5, Exclusive:1 (Figure 4.1.3.3-1)

Steps:

1) Client sends an HTTP GET request to b1 to retrieve the Entity OffStreetParking:1
2) Agent checks that a success response has been returned and that the Entity OffStreetParking:1 was returned
3) Client sends an HTTP GET request to b4 to retrieve the Entity OffStreetParking:1
4) Client sends an HTTP GET request to b5 to retrieve the Entity OffStreetParking:1

5) Agent checks that the Entity returned in step (2) is structured as it follows:

   a) The attributes "availableSpotsNumber" and "totalSpotsNumber" match the ones from the Entity returned in step (3)

   b) The attribute "location" matches the one from the Entity returned in step (4)

6.2.4.2 Retrieve location from OffStreetParking:1

Pre-conditions:

- No user context
- Data in all brokers
  - b1 contains:
    - OffstreetParking:1 without location (Figure 4.1.1.1-2)
  - b2 contains:
    - OffstreetParking:1 without location (Figure 4.1.1.1-2)
  - b3 contains:
    - OffstreetParking:1 without location (Figure 4.1.1.1-2)
  - b4 contains:
    - Full version of OffstreetParking:1 (Figure 4.1.1.1-1)
  - b5 contains:
    - OffstreetParking:1 with location and name only (Figure 4.1.1.1-3)
- Registrations that shall be established:
  - From b1 to b2, Auxiliary:1 (Figure 4.1.3.2-2)
  - From b2 to b4, Redirect:2 (Figure 4.1.3.4-1)
  - From b2 to b5, Redirect:3 (Figure 4.1.3.4-2)
  - From b1 to b3, Inclusive:1 (Figure 4.1.3.1-1)
  - From b3 to b5, Exclusive:3 (Figure 4.1.3.3-3)

Steps:

1) Client sends an HTTP GET request to b1 to retrieve the Entity OffStreetParking:1

2) Agent checks that a success response has been returned and that the Entity OffStreetParking:1 with the "location" Property was returned

3) Client sends an HTTP GET request to b2 (with local=true) to retrieve the Entity OffStreetParking:1

4) Client sends an HTTP GET request to b3 (with local=true) to retrieve the Entity OffStreetParking:1

5) Client sends an HTTP GET request to b5 (with local=true) to retrieve the Entity OffStreetParking:1

6) Agent checks that the Entity returned in step (2) is structured as it follows:

   a) The attributes "availableSpotsNumber" and "totalSpotsNumber" match the ones from the Entity returned in step (3)

   b) The attribute "name" matches the one from the Entity returned in step (4)

   c) The attribute "location" matches the one from the Entity returned in step (5)

6.3 Query Entities

6.3.1 IOP_CNF_01

6.3.1.1 Query Entities with type OffstreetParking (GET)

Pre-conditions:

- No user context
- Data only on leaves
  - b2 contains:
    - Full version of OffstreetParking:1 (Figure 4.1.1.1-1)
    - OffstreetParking:2 without location and totalSpotsNumber (Figure 4.1.1.1-6)
  - b3 contains:
    - Full version of OffstreetParking:2 (Figure 4.1.1.1-4)
- Registrations that shall be established:
  - From b1 to b2, Inclusive:2 (Figure 4.1.3.1-2)
  - From b1 to b3, Exclusive:2 (Figure 4.1.3.3-2)

Steps:

1) Client sends an HTTP GET request to b1 to retrieve all Entities with type OffstreetParking
2) Agent checks that a success response has been returned and that the following Entities were returned:
   a) OffstreetParking:1 with the following Attributes:
      i) availableSpotsNumber
      ii) totalSpotsNumber
   b) OffstreetParking:2 with the following Attributes:
      i) availableSpotsNumber
      ii) location
3) Client sends an HTTP GET request to b2 to retrieve all Entities with type OffstreetParking
4) Client sends an HTTP GET request to b3 to retrieve all Entities with type OffstreetParking
5) Agent checks that:
   a) b2 returns OffstreetParking:1 and OffstreetParking:2
   b) b3 returns OffstreetParking:2
6) Agent checks that:
   a) OffstreetParking:1 returned in step (2) is the same as the one returned in step (3)
   b) OffstreetParking:2 returned in step (2) is structured as it follows:
      i) The attribute "totalSpotsNumber" matches the one from the Entity returned in step (5.a)
      ii) The attribute "location" matches the one from the Entity returned in step (5.b)

6.3.1.2 Query Entities with type OffstreetParking (POST)

Pre-conditions:

- No user context
- Data only on leaves
  - b2 contains:
    - Full version of OffstreetParking:1 (Figure 4.1.1.1-1)
    - OffstreetParking:2 without location and totalSpotsNumber (Figure 4.1.1.1-6)
  - b3 contains:
    - Full version of OffstreetParking:2 (Figure 4.1.1.1-4)
- Registrations that shall be established:
  - From b1 to b2, Inclusive:2 (Figure 4.1.3.1-2)
  - From b1 to b3, Exclusive:2 (Figure 4.1.3.3-2)

Steps:

1) Client sends an HTTP POST request to b1 to retrieve all Entities with type OffstreetParking
2) Agent checks that a success response has been returned and that the following Entities were returned:
   a) OffstreetParking:1 with the following Attributes:
      i) availableSpotsNumber
      ii) totalSpotsNumber
   b) OffstreetParking:2 with the following Attributes:
      i) availableSpotsNumber
      ii) location
3) Client sends an HTTP POST request to b2 to retrieve all Entities with type OffstreetParking
4) Client sends an HTTP POST request to b3 to retrieve all Entities with type OffstreetParking
5) Agent checks that:
   a) b2 returns OffstreetParking:1 and OffstreetParking:2
   b) b3 returns OffstreetParking:2
6) Agent checks that:
   a) OffstreetParking:1 returned in step (2) is the same as the one returned in step (3)
   b) OffstreetParking:2 returned in step (2) is structured as it follows:
      i) The attribute "totalSpotsNumber" matches the one from the Entity returned in step (5.a)
      ii) The attribute "location" matches the one from the Entity returned in step (5.b)

6.3.2 IOP_CNF_02

6.3.2.1 Query Entities with type OffstreetParking (GET)

Pre-conditions:

- No user context
- Data only on leaves
  - b2 contains:
    - OffstreetParking:1 without location (Figure 4.1.1.1-2)
    - OffstreetParking:2 without location (Figure 4.1.1.1-5)
  - b3 contains:
    - Full version of OffstreetParking:1 (Figure 4.1.1.1-1)
    - Full version of OffstreetParking:2 (Figure 4.1.1.1-4)
  - b4 contains:
    - OffstreetParking:2 with location and name only (Figure 4.1.1.1-7)
- Registrations that shall be established:
  - From b1 to b2, Inclusive:2 (Figure 4.1.3.1-2)
  - From b1 to b3, Redirect:2 (Figure 4.1.3.3-2)
  - From b1 to b4, Redirect:3 (Figure 4.1.3.4-3)

Steps:

1) Client sends an HTTP GET request to b1 to retrieve Entities with type OffstreetParking
2) Agent checks that a success response has been returned and that the following Entities were returned:
   a) OffstreetParking:1 with all properties but "name"
   b) OffstreetParking:2 with all properties but "name"
3) Client sends an HTTP GET request to b2 to retrieve Entities with type OffstreetParking
4) Client sends an HTTP GET request to b3 to retrieve Entities with type OffstreetParking
5) Client sends an HTTP GET request to b4 retrieve Entities with type OffstreetParking
6) Agent checks that:
   a) OffstreetParking:1 returned in step (2) is structured as it follows:
      i) The attributes "availableSportsNumber" and "totalSpotsNumbers" are the same as the ones returned in step (3)
      ii) The Property "location" is the same as the one returned in step (4)
   b) OffstreetParking:2 returned in step (2) is structured as it follows:
      i) The attributes "availableSportsNumber" and "totalSpotsNumbers" are the same as the ones returned in step (3)
      ii) The Property "location" is the same as the one returned in step (5)

6.3.2.2 Query the "location" Property from Entities having type OffstreetParking or Vehicle (GET)

Pre-conditions:

- No user context
- Data only on leaves
  - b2 contains:
    - Full version of Vehicle:1 (Figure 4.1.1.2-1)
    - Full version of OffstreetParking:1 (Figure 4.1.1.1-1)
  - b3 contains:
    - OffstreetParking:1 with location and name only (Figure 4.1.1.1-3)
    - Full version of OffstreetParking:2 (Figure 4.1.1.1-4)
  - b4 contains:
    - OffstreetParking:2 with location and name only (Figure 4.1.1.1-6)
    - Full version of Vehicle:2 (Figure 4.1.1.2-2.1)
- Registrations that shall be established:
  - From b1 to b2, Inclusive:2 (Figure 4.1.3.1-2)
  - From b1 to b2, Inclusive:3 (Figure 4.1.3.1-3)
  - From b1 to b3, Redirect:2 (Figure 4.1.3.3-2)
  - From b1 to b4, Redirect:3 (Figure 4.1.3.4-3)
  - From b1 to b4, Redirect:4 (Figure 4.1.3.4-4)

Steps:

1) Client sends an HTTP GET request to b1 to retrieve the "location" Property from Entities having type OffstreetParking or Vehicle
2) Agent checks that a success response has been returned and that the following Entities were returned:
   a) OffstreetParking:1 with the "location" Property only
   b) OffstreetParking:2 with the "location" Property only
   c) Vehicle:1 with the "location" Property only
   d) Vehicle:2 with the "location" Property only
3) Client sends an HTTP GET request to b2 to retrieve the "location" Property from Entities having type OffstreetParking or Vehicle
4) Client sends an HTTP GET request to b3 to retrieve the "location" Property from Entities having type OffstreetParking or Vehicle
5) Client sends an HTTP GET request to b4 to retrieve the "location" Property from Entities having type OffstreetParking or Vehicle
6) Agent checks that:
   a) Vehicle:1 returned in step (2) is the same as the one returned in step (3)
   b) OffstreetParking:1 returned in step (2) is the same as the one returned in step (4)

c) Vehicle:2 and OffstreetParking:2 returned in step (2) are the same as the ones returned in step (5)

6.3.3 IOP_CNF_03

6.3.3.1 Query Entities with type OffstreetParking (GET)

Pre-conditions:

- No user context
- Data only on leaves
  - b2 contains:
    - OffstreetParking:2 without location (Figure 4.1.1.1-5)
  - b3 contains:
    - Full version of OffstreetParking:1 (Figure 4.1.1.1-1)
  - b4 contains:
    - OffstreetParking:1 without location (Figure 4.1.1.1-2)
- Registrations that shall be established:
  - From b1 to b2, Auxiliary:1 (Figure 4.1.3.2-1)
  - From b1 to b3, Inclusive:1 (Figure 4.1.3.1-1)
  - From b1 to b4, Inclusive:2 (Figure 4.1.3.1-2)

Steps:

1) Client sends an HTTP GET request to b1 to retrieve Entities with type OffstreetParking
2) Agent checks that a success response has been returned and that the following Entities were returned:
   a) Full version of OffstreetParking:1
   b) OffstreetParking:2 without location
3) Client sends an HTTP GET request to b2 to retrieve Entities with type OffstreetParking
4) Client sends an HTTP GET request to b3 to retrieve Entities with type OffstreetParking
5) Client sends an HTTP GET request to b4 retrieve Entities with type OffstreetParking
6) Agent checks that:
   a) OffstreetParking:1 returned in step (2) is structured as it follows:
      i) The attributes "name" and "location" are the same as the ones returned in step (4)
      ii) The Property "availableSpotsNumber" and "totalSpotsNumbers" are the same as the ones returned in step (5)
   b) OffstreetParking:2 returned in step (2) is the same as the one returned in step (3)

6.3.3.2 Query the "location" Property from Entities having type OffstreetParking or Vehicle (GET)

Pre-conditions:

- No user context
- Data only on leaves
  - b2 contains:
    - Full version of Vehicle:1 (Figure 4.1.1.2-1)
  - b3 contains:
    - OffstreetParking:1 with location and name only (Figure 4.1.1.1-3)
  - b4 contains:
    - Full version of OffstreetParking:1 (Figure 4.1.1.1-1)
    - Full version of OffstreetParking:2 (Figure 4.1.1.1-4)
    - Full version of Vehicle:2 (Figure 4.1.1.2-2.1)
- Registrations that shall be established:
  - From b1 to b2, Auxiliary:3 (Figure 4.1.3.2-3)
  - From b1 to b2, Inclusive:1 (Figure 4.1.3.1-1)
  - From b1 to b4, Inclusive:1 (Figure 4.1.3.1-2)

Steps:

1) Client sends an HTTP GET request to b1 to retrieve the "location" Property from all Entities having type OffstreetParking or Vehicle
2) Agent checks that a success response has been returned and that the following Entities were returned:
   a) OffstreetParking:1 with the "location" Property only
   b) OffstreetParking:2 with the "location" Property only
   c) Full version of Vehicle:1
3) Client sends an HTTP GET request to b2 to retrieve the "location" Property from Entities having type OffstreetParking or Vehicle
4) Client sends an HTTP GET request to b3 to retrieve the "location" Property from Entities having type OffstreetParking or Vehicle
5) Client sends an HTTP GET request to b4 to retrieve the "location" Property from Entities having type OffstreetParking or Vehicle
6) Agent checks that:
   a) OffstreetParking:1 returned in step (2) is structured as it follows:
      i) The attributes "name" and "location" are the same as the ones returned in step (4)
      ii) The attributes "totalSpotsNumber" and "availableSpotsNumber" are the same as the ones returned in step (5)
   b) OffstreetParking:2 returned in step (2) is the same as the one returned in step (5)
   c) Vehicle:1 returned in step (2) is the same as the one returned in step (3)

6.3.4 IOP_CNF_04

6.3.4.1 Query Entities of type OffStreetParking (GET)

Pre-conditions:

- No user context
- Data only on leaves
  - b4 contains:
    - OffstreetParking:1 with location and name only (Figure 4.1.1.1-3)
    - Full version of OffstreetParking:2 (Figure 4.1.1.1-4)
  - b5 contains:
    - Full version of OffstreetParking:1 (Figure 4.1.1.1-1)
    - OffstreetParking:2 with location and name only (Figure 4.1.1.1-7)
- Registrations that shall be established:
  - From b1 to b2, Auxiliary:2 (Figure 4.1.3.2-2)
  - From b2 to b4, Redirect:1 (Figure 4.1.3.4-1)
  - From b2 to b5, Redirect:2 (Figure 4.1.3.4-2)
  - From b1 to b3, Inclusive:1 (Figure 4.1.3.1-1)
  - From b3 to b5, Exclusive:2 (Figure 4.1.3.3-2)
  - From b3 to b5, Exclusive:3 (Figure 4.1.3.3-3)

Steps:

1) Client sends an HTTP GET request to b1 to retrieve Entities with type OffstreetParking
2) Agent checks that a success response has been returned and that the following Entities were returned:
   a) OffstreetParking:1 with the "location" Property
   b) OffstreetParking:2 with the "availableSpotsNumber", "totalSpotsNumber" and "location" Properties
3) Client sends an HTTP GET request to b4 to retrieve Entities with type OffstreetParking
4) Client sends an HTTP GET request to b5 retrieve Entities with type OffstreetParking
5) Agent checks that:
   a) OffstreetParking:1 returned in step (2) is structured as it follows:
      i) The Property "location" is the same as the one returned in step (4)
   b) OffstreetParking:2 returned in step (2) is structured as it follows:
      i) The Properties "availableSpotsNumber" and "totalSpotsNumber" are the same as the ones returned in step (3)
      ii) The Property "location" is the same as the one returned in step (4)

6.3.4.2 Query Entities of type OffS treetParking (POST)

Pre-conditions:

- No user context
- Data only on leaves
  - b4 contains:
    - OffstreetParking:1 with location and name only (Figure 4.1.1.1-3)
    - Full version of OffstreetParking:2 (Figure 4.1.1.1-4)
  - b5 contains:
    - Full version of OffstreetParking:1 (Figure 4.1.1.1-1)
    - OffstreetParking:2 with location and name only (Figure 4.1.1.1-7)
- Registrations that shall be established:
  - From b1 to b2, Auxiliary:2 (Figure 4.1.3.2-2)
  - From b2 to b4, Redirect:1 (Figure 4.1.3.4-1)
  - From b2 to b5, Redirect:2 (Figure 4.1.3.4-2)
  - From b1 to b3, Inclusive:1 (Figure 4.1.3.1-1)
  - From b3 to b5, Exclusive:2 (Figure 4.1.3.3-2)
  - From b3 to b5, Exclusive:3 (Figure 4.1.3.3-3)

Steps:

1) Client sends an HTTP POST request to b1 to retrieve Entities with type OffstreetParking
2) Agent checks that a success response has been returned and that the following Entities were returned:
   a) OffstreetParking:1 with the "location" Property
   b) OffstreetParking:2 with the "availableSpotsNumber", "totalSpotsNumber" and "location" Properties
3) Client sends an HTTP POST request to b4 to retrieve Entities with type OffstreetParking
4) Client sends an HTTP POST request to b5 retrieve Entities with type OffstreetParking
5) Agent checks that:
   a) OffstreetParking:1 returned in step (2) is structured as it follows:
      i) The Property "location" is the same as the one returned in step (4)
   b) OffstreetParking:2 returned in step (2) is structured as it follows:
      i) The Properties "availableSpotsNumber" and "totalSpotsNumber" are the same as the ones returned in step (3)
      ii) The Property "location" is the same as the one returned in step (4)

Annex A (informative): Change history

Date	Version	Information about changes
12/2024	0.0.1	Early draft
01/2025	0.0.2	Fixed bugs in the existing test cases
02/2025	0.0.3	Added up to two tests cases for each operation/configuration
07/2025	1.1.1	First published version

History

Version	Date	Status
V1.1.1	July 2025	Publication