



UNIVERSIDAD
POLITÉCNICA
DE MADRID

POLITÉCNICA

This project has been partially funded by the Madrid Government (Comunidad de Madrid-Spain) under the Multiannual Agreement 2023-2026 with Universidad Politécnica de Madrid in the Line A, Emerging PhD researchers with the project GUIA (M230020126A-AJCA)



CONSEJERÍA DE EDUCACIÓN,
CIENCIA Y UNIVERSIDADES

Comunidad de Madrid

ODRL Policy Enforcement with ODRE

Andrea Cimmino

Universidad Politécnica de Madrid

andreajesus.cimmino@upm.es



POLITÉCNICA

Research

Semantic Interoperability

Distributed Discovery

Policies for Data Usage



**OPAL
WORKSHOP**

Standardization



Alliance for IoT
and Edge Computing
Innovation



Web of Things (WoT) Discovery

W3C Recommendation 05 December 2023



ODRL Formal Semantics

Draft Community Group Report 13 April 2026



ODRE

EU Projects



DELTA



COGITO



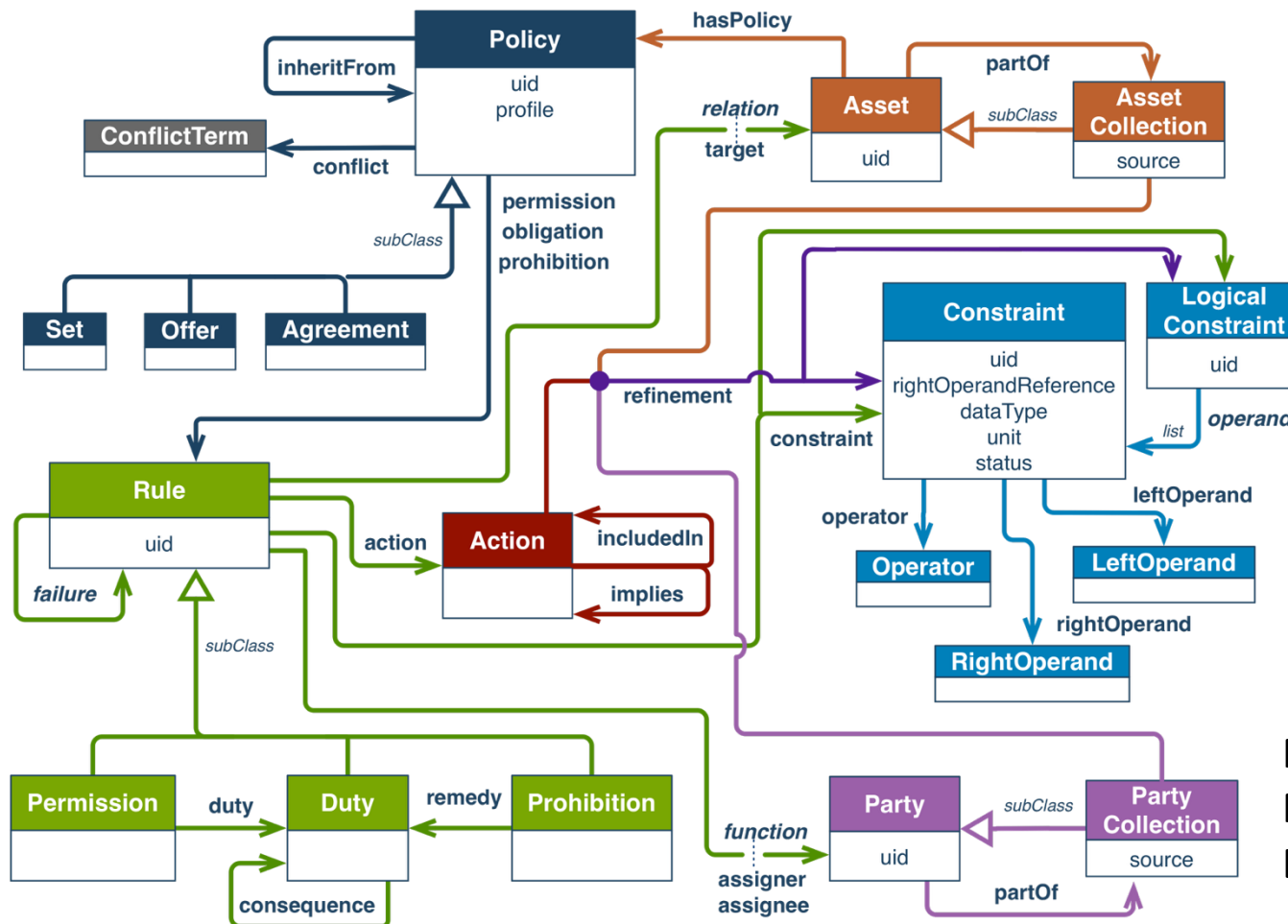
Smart Transformation & Innovation Consortium Slovakia





ODRL Information Model 2.2

W3C Recommendation 15 February 2018



Permission
Prohibition
Duty

ODRL Formal Semantics

Draft Community Group Report 13 April 2026

```
{
"@context": "http://www.w3.org/ns/odrl.jsonld",
"@type": "Set",
"uid": "http://example.com/policy/A1",
"permission": [{
  "@id": "http://example.com/rule/A1",
  "target": "http://example.com/document/1234",
  "assigner": "http://example.com/party/16",
  "action": "distribute",
  "constraint": [{
    "@id": "http://example.com/constraint/A1",
    "leftOperand": "dateTime",
    "operator": "lt",
    "rightOperand": { "@value": "2018-01-01", "@type": "xsd:date" }
  }]
}]
}
```

ODRL Formal Semantics

Draft Community Group Report 13 April 2026

```
{
"@context": "http://www.w3.org/ns/odrl.jsonld",
"@type": "Set",
"uid": "http://example.com/policy/A1",
"permission": [{
  "@id": "http://example.com/rule/A1",
  "target": "http://example.com/document/1234",
  "assigner": "http://example.com/party/16",
  "action": "distribute",
  "constraint": [{
    "@id": "http://example.com/constraint/A1",
    "leftOperand": "dateTime",
    "operator": "lt",
    "rightOperand": { "@value": "2018-01-01", "@type": "xsd:date" }
  }]
}]
}
```

Resource protected

Policy owner

Action allowed

Conditions

ODRL Formal Semantics

Draft Community Group Report 13 April 2026

```
{
"@context": "http://www.w3.org/ns/odrl.jsonld",
"@type": "Set",
"uid": "http://example.com/policy/A1",
"permission": [{
```

Case	Policy	State of the World	Evaluation Request	Satisfaction of class, target, assignee	Satisfaction state of constraint:A1	Activation state of the permission	Control state of the permission	Evaluation Report
A1-1	policy:A1	empty	party:1 distributes document:1234 on 2017-12-19T15:00:00	satisfied	satisfied	active	permit the action	report
A1-2	policy:A1	empty	party:1 distributes document:1234 on 2019-12-19T15:00:00	satisfied	not-satisfied	inactive	deny the action	report

```
}]
}]
}
```



Maven™

<https://github.com/orgs/ODRE-Framework>

- Native ODRL support
- ODRL extensions
- Code-base functionalities (interpolation)

ODRE in a nutshell: policy template and variables

1. ODRL TEMPLATE (JSON-LD)

policy-template.jsonld

```
{
  "@context": "http://www.w3.org/ns/odrl.jsonld",
  "@type": "Offer",
  "uid": "http://example.com/policy:6163",
  "profile": "http://example.com/odrl:profile:10",
  "permission": [{
    "target": "http://example.com/document:1234",
    "assigner": "http://example.com/org:616",
    "action": "distribute",
    "constraint": [{
      "leftOperand": "dateTime",
      "operator": "lt",
      "rightOperand": { "@value": "2025-01-01",
        "@type": "xsd:date" }
    }],
    "leftOperand": { "@value": "{{retrieve_token()}}",
      "@type": "xsd:string" },
    "operator": "eq",
    "rightOperand": { "@value": "{{token}}",
      "@type": "xsd:string" }
  }]
}
```

2. EVALUATION PAYLOAD (PYTHON)

evaluation_payload.py

```
# Evaluation payload as Python variables
evaluation_payload = {
  "token": token,
  "retrieve_token()": retrieve_token
}

# Data used for instantiating the template
token = "abc123xyz"

def retrieve_token() -> str:
    """
    Returns a token string.
    """
    ... # Implementation not shown
```

3. ODRL POLICY (JSON-LD)

odrl-policy.html

```
{
  "@context": "http://www.w3.org/ns/odrl.jsonld",
  "@type": "Offer",
  "uid": "http://example.com/policy:6163",
  "profile": "http://example.com/odrl:profile:10",
  "permission": [{
    "target": "http://example.com/document:1234",
    "assigner": "http://example.com/org:616",
    "action": "distribute",
    "constraint": [{
      "leftOperand": "dateTime",
      "operator": "lt",
      "rightOperand": { "@value": "2025-01-01",
        "@type": "xsd:date" }
    }],
    "leftOperand": { "@value": "abc123xyz",
      "@type": "xsd:string" },
    "operator": "eq",
    "rightOperand": { "@value": "abc123xyz",
      "@type": "xsd:string" }
  }]
}
```



This is the ODRL template written in JSON-LD. It contains placeholders (e.g., `{{retrieve_token()}}`, `{{token}}`) that will be replaced with values from the evaluation payload.



Result: a valid ODRL policy in JSON-LD (HTML serialization), with all placeholders replaced by the evaluation payload data.



ODRL Template (JSON-LD)
Structure with placeholders

+



Evaluation Payload (Python)
Values to instantiate the template

→



ODRL Policy (HTML)
Instantiated policy document

ODRE in a nutshell: interpolation



```
policy = ""
{
  "@context": "http://www.w3.org/ns/odrl.jsonld",
  "@type": "Offer",
  "uid": "http://example.com/policy:6163",
  "profile": "http://example.com/odrl:profile:10",
  "permission": [{
    "target": "http://example.com/document:1234",
    "assigner": "http://example.com/org:616",
    "action": "distribute",
    "constraint": [{
      "leftOperand": "dateTime",
      "operator": "lt",
      "rightOperand": { "@value": "2025-01-01", "@type": "xsd:date" }
    },{
      "leftOperand": { "@value": "{{retrieve_token()}}" , "@type": "xsd:string" },
      "operator": "eq",
      "rightOperand": { "@value": "{{token}}", "@type": "xsd:string" }
    }
  ]
}]
}
""
```

ODRE in a nutshell: interpolation

```
policy = ""  
  
def get_token():  
    return "AAA"
```

```
usage_decision = ODRE().set_debug(True).enforce(policy, interpolations={'token': 'AAA',  
'retrieve_token': get_token})  
print(usage_decision)
```

```
    "constraint": [{  
        "leftOperand": "dateTime",  
        "operator": "lt",  
        "rightOperand": { "@value": "2025-01-01", "@type": "xsd:date" }  
    }, {  
        "leftOperand": { "@value": "{{retrieve_token()}}" , "@type": "xsd:string"},  
        "operator": "eq",  
        "rightOperand": { "@value": "{{token}}", "@type": "xsd:string" }  
    }  
  ]  
}  
.....
```

ODRE in a nutshell: interpolation

```
def get_token():  
    return "AAA"
```

```
usage decision = ODRE().set debug(True).enforce(policy, interpolations={'token' : 'AAA',  
'retrieve_token' : get_token})  
print(usage_decision)
```

```
    "constraint": [{  
        "left0operand": "dateTime",  
        "operator": "lt",  
        "right0operand": { "@value": "2025-01-01", "@type": "xsd:date" }  
    }, {  
        "left0operand": { "@value": "{{retrieve_token()}}", "@type": "xsd:string" },  
        "operator": "eq",  
        "right0operand": { "@value": "{{token}}", "@type": "xsd:string" }  
    }  
  ]  
}  
.....
```

ODRE in a nutshell: interpolation

```
policy = ""  
  
def get_token():  
    return "AAA"
```

```
usage_decision = ODRE().set_debug(True).enforce(policy, interpolations={'token': 'AAA',  
'retrieve_token': get_token})  
print(usage_decision)
```

```
    "constraint": [{  
        "leftOperand": "dateTime",  
        "operator": "lt",  
        "rightOperand": { "@value": "2025-01-01", "@type": "xsd:date" }  
    }, {  
        "leftOperand": { "@value": "{{retrieve_token()}}" , "@type": "xsd:string"},  
        "operator": "eq",  
        "rightOperand": { "@value": "{{token}}", "@type": "xsd:string" }  
    }  
  ]  
}  
.....
```

ODRE in a nutshell: policy compilation

1. INPUT: ODRL POLICY (JSON-LD)

odrl-policy.jsonld

```
{
  "@context": "http://www.w3.org/ns/odrl.jsonld",
  "@type": "Offer",
  "uid": "http://example.com/policy:6163",
  "profile": "http://example.com/odrl:profile:10",
  "permission": [
    {
      "target": "http://example.com/document:1234",
      "assigner": "http://example.com/org:616",
      "action": "distribute",
      "constraint": [
        {
          "leftOperand": "dateTime",
          "operator": "lt",
          "rightOperand": {
            "@value": "2018-01-01",
            "@type": "xsd:date"
          }
        }
      ]
    }
  ]
}
```



This ODRL policy grants permission to distribute the target resource only if the current dateTime is before 2018-01-01.



ODRE COMPILER

The ODRE Compiler analyzes the ODRL policy, extracts the constraints, and generates a Python script that evaluates them and returns **True** or **False**.

2. OUTPUT: PYTHON CONSTRAINT SCRIPT

policy_check.py

```
1 from datetime import datetime
2
3 def odre_lt(left, right):
4     """Return True if left < right, else False."""
5     return left < right
6
7 def odre_datetime():
8     """Return the current date and time."""
9     return datetime.now().isoformat(timespec="seconds")
10
11 def evaluate():
12     """Evaluate the policy constraint:
13         current dateTime < 2018-01-01T00:40:30."""
14     return odre_lt(odre_datetime(), "2018-01-01T00:40:30")
15
16 if __name__ == "__main__":
17     print(evaluate()) # Prints True or False
```



The generated Python script evaluates the constraint from the policy and returns **True** if the current dateTime is before 2018-01-01T00:40:30, otherwise **False**.



ODRL Policy (JSON-LD)
Input policy with constraints



ODRE Compiler
Extracts constraints and generates Python script



Python Script
Evaluates constraints and returns **True** or **False**

- Policy templates:
 - Are not always correct RDF
 - Allow to “generate” on the fly a large number of policies (on demand or beforehand)
 - May contain complex logic
- Policy compilation:
 - Requires bounding ontological terms to code (e.g., Python or Java)



<https://colab.research.google.com/drive/10XHbTmSKtOJlYmg3f94vZDyIR8Txvvi5?usp=sharing>

