

Umsetzung von IDS Policyklassen

- Policy Pattern
- Ergebnisse für das Informationsmodell v3.1.0
 - Policyklasse 1: Verwendung von Daten erlauben
 - Policyklasse 2: Verwendung von Daten verbieten
 - Policyklasse 3: Datennutzung für Benutzer oder Systeme beschränken
 - Policyklasse 4: Datennutzung für bestimmte Zwecke einschränken
 - Policyklasse 5: Datennutzung nach Ereignis einschränken
 - Policyklasse 6: Datennutzung in Zeitraum einschränken
 - Policyklasse 7: Datennutzung N mal erlauben
 - Policyklasse 8: Datennutzung mit anschließender Löschung erlauben
 - Policyklasse 9: Verwendung von modifizierten Daten (in Transit) erlauben
 - Policyklasse 10: Verwendung von modifizierten Daten (at Rest) erlauben
 - Policyklasse 11: Informationen über Datennutzung loggen
 - Policyklasse 12: Entitäten über Datennutzung informieren
 - Policyklasse 13: Daten unter gewissen Umständen Teilen
 - Policyklasse 14: Verwendung von Daten fein-granular einschränken
- Ergebnisse für das Informationsmodell v4.0.0
- Transformation von Policies
 - Verarbeitung von IDS-Policies
 - Abbildung auf Nukleus & Degree Policies
- Usage Control Objekt
 - Details zur Umsetzung in D°
 - Schnittstelle D° Nukleus

Versionen

Datum	Notiz	Version
30.09.2020	Erste Veröffentlichung <ul style="list-style-type: none">• Positionierung zu den 14 IDS Policy-Pattern• Transformation von ODRL Policies• Einführung des D° Usage Control Objekts	1.0

Autoren

Name	Institution	Email
Fabian Bruckner	Fraunhofer ISST	fabian.bruckner@isst.fraunhofer.de
Ralf Nagel	Fraunhofer ISST	ralf.nagel@isst.fraunhofer.de

Die Entwicklung von D° ist aus den International Data Spaces hervorgegangen und fokussierte sich bisher auf die Entwicklung eines Fundaments, welches für die Umsetzung der Ziele im Bereich Usage Control der IDS verwendet werden kann. Dieses Fundament umfasst die erweiterbare domänenspezifische Programmiersprache D°, welche das Programmierparadigma der policy-agnostischen Programmierung umsetzt. Dabei wird das Thema der Datennutzungskontrolle von D° als Querschnittsthema aufgefasst, welches vom Beginn der Entwicklung an berücksichtigt werden muss. Die mit D° entwickelten Applikationen enthalten maßgeschneiderten Programmcode zur Umsetzung von Policies, welcher untrennbar mit der eigentlichen Applikationslogik verbunden ist.

Aus der Entwicklung von D° ist das Typsystem Nukleus hervorgegangen. Nukleus ist ein wichtiger Bestandteil von D°, kann aber auch in anderen Einsatzgebieten bzw. vollkommen Eigenständig verwendet werden. Datentypen in Nukleus werden in JSON, YAML oder XML definiert. Anschließend können die Datentypen direkt verwendet werden. Alternativ können aus den Typdefinitionen Javaklassen generiert werden, welche im Anschluss mit Nukleus verwendet werden können. Die Datentypen, die in Nukleus geladen sind, stehen in D° als atomare Elemente zur Verfügung. Dabei erfolgt die Verwendung der Datentypen direkt und erfordert keinen Umweg über Nukleus. Der Anwender benötigt somit keinerlei Kenntnis über Nukleus und dessen Funktionsweise. Dabei ist das Ziel von Nukleus die Definition und Verwendung von domänenspezifischen Datentypen (bspw. Lieferschein) anstelle von primitiven (bspw. Integer), welche technisch möglichst Effizient gespeichert werden können. Somit verzichtet Nukleus bewusst auf maximale Speichereffizienz zu Gunsten einer stärkeren Semantik. Nukleus bietet diverse weitere Funktionen, welche entweder die Nutzung vereinfachen bzw. verbessern, oder zusätzliche Funktionale Aspekte zum Typsystem hinzufügen. Diese Funktionalitäten umfassen unter anderem die folgenden Punkte:

- Automatische Generierung einer Wiki-basierten Dokumentation für das gesamte Typsystem auf Basis der Datentypdefinition
- Abbildung komplexer Zusammenhänge zwischen Datentypen über Sub- und Supertypen
- Definition und automatische Überprüfung von umfangreichen Regeln zur Validation von Datentyp-Instanzen
- Native Unterstützung von Arrays und Listen für alle Datentypen
- Definition und Überprüfung von Policies auf Datentypenebene

Policy Pattern

Um eine gemeinsame Basis für Diskussionen und Vereinbarung zu bieten, werden in den IDS diverse Klassen von Policies definiert, welche beim Datenaustausch und in Workflows verwendet werden können. Dabei ist es nicht das Ziel, dass jede Usage Control Lösung alle dieser Muster umsetzen kann. Abhängig vom verwendeten Verfahren zur Umsetzung von Usage Control, sowie dem intendierten Einsatzgebiet der jeweiligen Lösung, besitzt jede Lösung für Usage Control in den IDS eine eigene Menge an Policies, welche umgesetzt werden können. Aus diesem Grund ist es notwendig, dass jede Lösung für Usage Control, welche in den IDS verwendet werden soll, eine klare Aussage dazu trifft, welche der IDS Policyklassen umgesetzt werden können und welche nicht.

Die IDS haben ursprünglich 14 unterschiedliche Policyklassen definiert, welche ausführlich im Dokument "Usage Control in the International Data Spaces", Version 2.0 (<https://www.internationaldataspaces.org/wp-content/uploads/2020/06/IDSA-Position-Paper-Usage-Control-in-IDS-2.0.pdf>) dargestellt sind. Mit der Veröffentlichung des Informationsmodells in Version 4.0.0 wurde diese Menge durch 20, teilweise im Usage Control Dokument auftauchende, unterschiedliche Policyklassen abgelöst.

Da die Arbeiten an diesem Projekt vor der Veröffentlichung des Informationsmodells Version 4.0.0 begonnen haben, existiert ein Zwischenergebnis für die Policyklassen, welche im Usage Control in the International Data Spaces Version 2.0 Dokument definiert wurden. Dieses Zwischenergebnis wird nachfolgend präsentiert, bevor auf die finalen Ergebnisse für das Informationsmodell Version 4.0.0 vorgestellt werden.

Ergebnisse für das Informationsmodell v3.1.0

Die nachfolgende Tabelle gibt eine Zusammenfassung der Positionierung von D° gegenüber den verschiedenen Policyklassen. In den folgenden Abschnitten werden die einzelnen Klassen detaillierter betrachtet und deren Umsetzung in D° genauer beschrieben bzw. die nicht erfolgte Umsetzung begründet. Die einzelnen Abschnitte enthalten unter Umständen an einigen Stellen technische Details über D° und Nukleus und sind ohne weitere Kenntnisse über die jeweiligen Lösungen nicht zu verstehen.

	Name	Beschreibung	In D° verwendbar	Anmerkung
1	Verwendung von Daten erlauben	Bestimmte Aktionen auf den Daten (bspw. schreiben, löschen, anzeigen) erlauben. Die Menge von möglichen Aktionen wird durch die IDS definiert.	✓	Umgesetzt in D° und Nukleus ¹
2	Verwendung von Daten verbieten	Bestimmte Aktionen auf den Daten (bspw. schreiben, löschen, anzeigen) verbieten. Die Menge von möglichen Aktionen wird durch die IDS definiert.	✓	Umgesetzt in D° und Nukleus ¹
3	Datennutzung für Benutzer oder Systeme beschränken	Die Datennutzung für eine Gruppe von Benutzern/Systemen erlauben oder verbieten. Die notwendigen Informationen über Benutzer und Systeme gehen aus dem Informationsmodell hervor.	✓	Umgesetzt in D° ¹
4	Datennutzung für bestimmte Zwecke einschränken	Die Nutzung der Daten für bestimmte Zwecke (bspw. Risikomanagement) erlauben oder verbieten. Der Zweck ist in diesem Kontext nicht genauer spezifiziert.	✓	Umgesetzt in D° ¹
5	Datennutzung nach Ereignis einschränken	Unter bestimmtem Umständen bzw. nachdem ein Ereignis eingetreten ist, soll die Datennutzung verboten oder erlaubt werden. Eine Menge von möglichen Ereignissen soll in den IDS definiert werden.	✓	Umgesetzt in Nukleus ¹
6	Datennutzung in Zeitraum einschränken	Diese Policy verwendet ein definiertes Zeitintervall. Die Daten dürfen entweder ausschließlich in diesem Intervall verwendet werden oder die Nutzung ist in diesem Intervall verboten.	✓	Umgesetzt in Nukleus ¹
7	Datennutzung <i>N</i> mal erlauben	Einschränkung der Datennutzung/Ausführung von Aktionen auf eine bestimmte Anzahl	✓	Umgesetzt in Nukleus ¹
8	Datennutzung mit anschließender Löschung erlauben	Die Nutzung von Daten wird für einen gewissen Zeitraum erlaubt. Diese Policy erfordert die Löschung der Daten, nachdem der erlaubte Nutzungszeitraum abgelaufen ist.	✗	Keine Unterstützung
9	Verwendung von modifizierten Daten (in Transit) erlauben	Die Ausführung von gewissen Aktionen ist auf den Rohdaten nicht erlaubt, sondern nur nach einer vorhergehenden Modifikation (bspw. Anonymisierung) der Daten. Dabei muss diese Modifikation auf dem Transportweg erfolgen.	✗	Keine Unterstützung
10	Verwendung von modifizierten Daten (at Rest) erlauben	Die Daten dürfen ohne vorherige Modifikation (bspw. Anonymisierung) nicht in einen inaktiven Zustand überführt werden (z.B. in einer Datenbank gespeichert).	✓	Umgesetzt in D° ¹
11	Informationen über Datennutzung loggen	Bestimmte Aspekte der Datennutzung müssen geloggt werden (bspw. die Anonymisierung von Daten).	✓	Umgesetzt in Nukleus ¹
12	Entitäten über Datennutzung informieren	Entitäten müssen über bestimmte Ereignisse proaktiv informiert werden. Beispielsweise wenn die Daten beim Konsumenten ankommen. Formate und Möglichkeiten dieser Benachrichtigungen müssen in den IDS noch weiter ausgearbeitet werden.	✓	Umgesetzt in Nukleus ¹
13	Daten unter gewissen Umständen Teilen	Im Normalfall darf der Datenkonsument die empfangenen Daten nicht an weitere Parteien in oder außerhalb der IDS weiterleiten. Diese Policy erlaubt es hierfür Ausnahmen zu definieren, welche die Weitergabe von Daten dennoch erlaubt.	✗	Keine Unterstützung
14	Verwendung von Daten fein-granular einschränken	Die Aktionen welche in den Policyklassen 1 und 2 verwendet werden sind für manche Anwendungsfälle unter Umständen noch zu generisch. In diesen Fällen können die erlaubten bzw. verbotenen Aktionen in dieser Policy näher spezifiziert werden.	✗	Keine Unterstützung

¹ Da D° nicht ohne Nukleus verwendet werden kann, sind sämtliche Policyklassen, die ausschließlich in Nukleus umgesetzt sind, auch in D° verwendbar. Dies gilt umgekehrt nicht. Die Umsetzung der einzelnen Policyklassen erfolgt dort, wo es jeweils am sinnvollsten und besten realisierbar ist.

Bei der nachfolgenden genaueren Betrachtung der einzelnen Policyklassen werden die notwendigen Eingaben beschrieben, sowie jeweils Details über die (nicht erfolgte) Umsetzung in D° und Nukleus aufgezeigt. Dabei werden bei den Eingaben nur Daten aufgelistet, die charakteristisch für die jeweilige Klasse sind. Daten die allgemein für alle Policyklassen zur Verfügung stehen werden nicht jedesmal aufgelistet. Diese ausgelassenen Informationen umfassen:

- Identifier des Data Providers
 - Identifier des Data Consumers
 - Identifier des Workflows, auf den die Policy angewendet werden soll
 - Die Aktion, welche durch die Policy betroffen ist
- Dokumentation der Action-Klasse: <https://international-data-spaces-association.github.io/InformationModel/docs/index.html#Action>
RDF-Definition der Action-Klasse: <https://github.com/International-Data-Spaces-Association/InformationModel/blob/v4.0.0/codes/Action.ttl>
Mögliche Ausprägungen: anonymize, attribute, compensate, delete, distribute, grant use, notify, modify, next policy, read, track provenance, use, write

Dabei werden Informationen über Aufbau und Struktur sofern möglich aus dem IDS Informationsmodell Version 4.0.0 entnommen, welches zur Abbildung von Policies verwendet wird. Entsprechende Informationen über das IDS Informationsmodell können der Dokumentation (<https://international-data-spaces-association.github.io/InformationModel/docs/index.html>) oder der RDF-Definition (<https://github.com/International-Data-Spaces-Association/InformationModel/tree/v4.0.0>) entnommen werden.

Unterschiedliche Granularität/Semantik der Policy Trigger:

System	Action (Infomodel) Event (Nukleus)
IDS Infomodel	anonymizeattribute, compensate, delete, distribute, grant use, notify, modify, next policy, read, track provenance, use, write
D°	Policies in D° verfügen über eine API, welche mehrere unterschiedliche Enforcementpoints bereitstellen. Diese können für unterschiedliche Enforcement-Zeitpunkte und -Szenarien verwendet werden. Die D°-Policy API und deren Methoden sind in den entsprechenden D°-Dokumentationen erläutert.
Nukleus	NEW_INSTANCE, READ, READ_ALL, WRITE, ADD, REMOVE, REMOVE_ALL, GET, SET, NULLIFY, LINK, LOOKUP, CLONE, CAST, MAP, SERIALIZE, DESERIALIZE, VALIDATION_FAILED

Policyklasse 1: Verwendung von Daten erlauben

	D°	Nukleus
Umgesetzt	✓ ²	✓ ³
² Mit Einschränkungen:		
³ Nur für ids:read, ids:modify und ids:use		

Eine der beiden simpelsten Policyklassen in den IDS. Erlaubt die Nutzung der Daten für eine bestimmte Aktion.

Eingaben

Keine weiteren Eingaben notwendig.

Details zur Umsetzung in D°

D° erlaubt es Eingabeparameter von Aktivitäten mit Tags zu versehen, welche die Aktionen beschreiben, die auf den übergebenen Daten ausgeführt werden. Bei Aktivitäten handelt es sich im Kontext von D° um die atomare funktionale Einheit der Programmiersprache. Die Vergabe der Tags für die Eingabeparameter von Aktivitäten findet während der Definition und Implementation der Aktivität statt und wird durch den Entwickler durchgeführt. Dabei ist eine sorgfältige und gewissenhafte Vergabe der Tags von hoher Wichtigkeit, da ansonsten die spätere Auswertung von Policies zu unerwarteten und falschen Ergebnissen kommen kann.

Das IDS Informationsmodell verwendet zur Definition von Policies das Konstrukt Action. Um eine Auswertung von diesem IDS spezifischem Konstrukt im Rahmen von D° Policies zu erlauben, werden diese Actions in Activity Tags überführt, welche semantisch äquivalent zu den Actions sind. Darüber hinaus verwendet der IDS URIs, um einzelne Datensätze zu identifizieren. Um eine solche Identifikation auch in D°-Applikationen zu ermöglichen, bietet D° die Möglichkeit beliebige Arten von externen IDs auf die internen IDs von D°, welche durch das Typsystem Nukleus bereitgestellt werden, abzubilden.

Hierdurch wird es möglich während der Policyauswertung diese Tags miteinander zu vergleichen und hierdurch Aussagen über die Verwendung von einzelnen Datensätzen zu machen.

Das nachfolgende Beispiel zeigt die entsprechende Policyklasse, ausgedrückt im IDS-Informationsmodell und die Definition einer entsprechenden D°-Policy.

```
IDS.AllowUsage:
  degree.Constraint@IDSAllowUsage:
    name:
      Identifier: "IDS.AllowUsage"
    attribute:
      degree.Parameter:
        - name:
            Identifier: "provider"
          type:
            Type: "core.URI"
        - name:
            Identifier: "consumer"
          type:
            Type: "core.URI"
        - name:
            Identifier: "targetArtifact"
          type:
            Type: "core.URI"
        - name:
            Identifier: "action"
          type:
            Type: "IDS.Action"
```

Dabei arbeitet die D° Implementierung von dieser Klasse mit einem Whitelisting-Ansatz. Während bei der Abwesenheit von Policies jede Verwendung der Daten erlaubt ist, sorgt die Verwendung von dieser Klasse das ausschließlich die Actions (bzw. die daraus resultierenden ActivityTags), welche in der Policy definiert sind, zugelassen werden.

Das nachfolgende Codebeispiel zeigt die Signatur einer D°-Applikation, welche zeigt, wie Policies an die Eingabeparameter einer Data App geknüpft werden können. Dabei wird die Policy `AllowModification`, eine Instanz der ersten Policy Klasse, an den Eingabeparameter `payload` vom Typ `Text` gebunden.

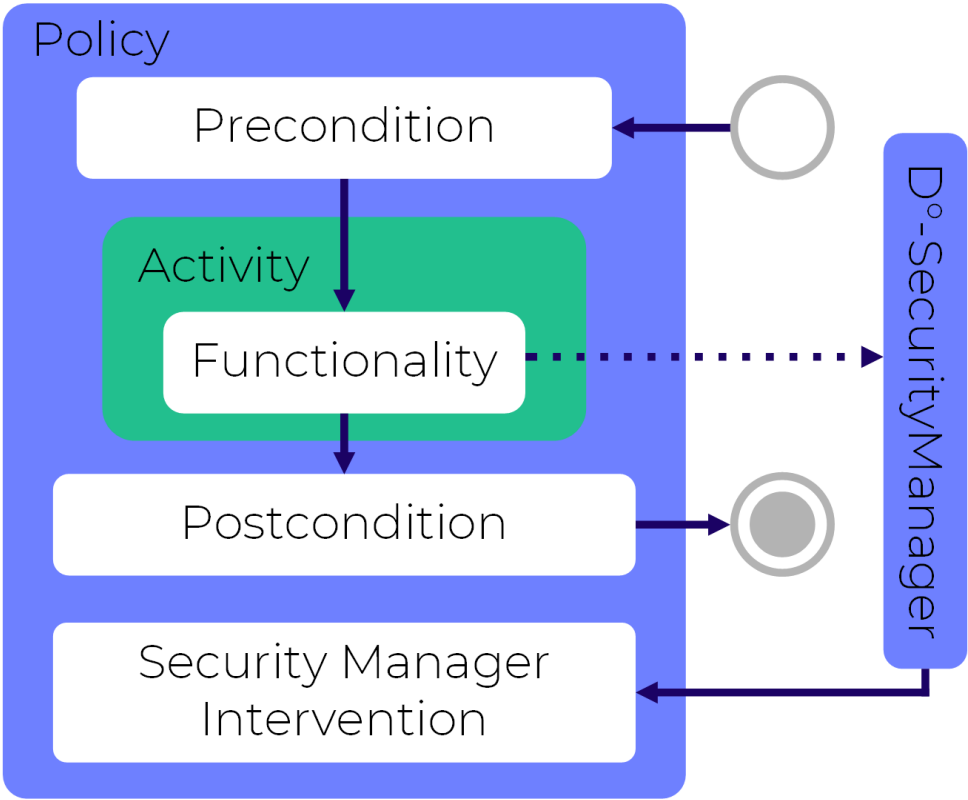
```
code
[payload = $Text(@policy["AllowModification"])] -> begin
```

Die dazugehörige Instanz `AllowModification` ist im nachfolgenden Codeblock zu sehen.

```
AllowModification:
  degree.ConstraintInstance@Pattern1TestB:
    name:
      Identifier: "Pattern1TestB"
    definition:
      degree.ConstraintReference: "degree.Constraint@IDSAllowUsage"
    mappedElements:
      degree.InstanceMap:
        - key:
            Text: "provider"
          value:
            Json: "{ \"core.URI\": \"https://provider\" }"
        - key:
            Text: "consumer"
          value:
            Json: "{ \"core.URI\": \"https://consumer\" }"
        - key:
            Text: "targetArtifact"
          value:
            Json: "{ \"core.URI\": \"https://artifact\" }"
        - key:
            Text: "action"
          value:
            Json: "{ \"IDS.Action\": \"MODIFY\" }"
```

Wird die Applikation nun aufgerufen, wird für jeden Aufruf eine Instanz vom Typ `Text` mit dem Namen `payload` im Typsystem Nukleus erzeugt. Diese Instanz hat darüber hinaus eine ID in Nukleus, welche eine eindeutige Identifikation der Instanz erlaubt. Dabei wird für jeden Aufruf auch gespeichert, zu welcher Nukleus ID die ID aus der IDS Policy (- in diesem Fall <https://artifact> -) gehört. Dies ist notwendig, da eine Policy in D° zunächst nicht weiß, an welches Konstrukt (bspw. Typ-Instanz oder Aktivität) sie gebunden ist und ein Auflösen ohne diese Abbildungsinformationen nicht mehr möglich wäre.

Jede Policy in D° hat eine Schnittstelle, welche drei verschiedene Punkte zur Ausführung von Überprüfungslogik bietet. Diese erlauben es Bedingungen vor und nach der Ausführung von Aktivitäten umzusetzen, oder APIs der verwendeten Hostsprache zu überwachen und eine Auswertung an Aufrufe dieser APIs aus einer Aktivität raus zu binden. Detailliertere Beschreibungen der einzelnen Punkte kann den entsprechenden D°-Dokumentationen entnommen werden. Die nachfolgende Abbildung zeigt eine schematische Darstellung des Aufbaus einer D°-Policy.



Die IDS Policyklassen verwenden zu ihrer Umsetzung die Precondition, welche vor der Ausführung von Aktivitäten ausgeführt wird. Dies liegt darin begründet, dass bei jedem Aktivitätsaufruf die notwendigen Policies überprüft werden und falls ein Datensatz, welcher mit Policies versehen ist, als Eingabe für eine Aktivität verwendet werden soll, findet vor der eigentlichen Ausführung der Aktivität die Überprüfung statt, ob der Aufruf die, an die Daten angehängten Policies erfüllt.

Details zur Umsetzung in Nukleus

Analog wäre es möglich einzelne Instanzen zu taggen. Da die Nukleus Policy per Default wenn keine Regel zutrifft den Zugriff erlauben muss eine DECLINE Regel an das Ende gesetzt werden (damit andere Zugriff verboten sind). Damit sind allerdings alle anderen Datentypen nicht mehr nutzbar. Daher müssen in realistischen Szenarien weitere (lokale) Datentypen freigegeben werden.

IDS-ODRL (JSON-LD)	Nukleus Policy (YAML)
--------------------	-----------------------

<pre>{ "@context": "http://www.w3id.org/ids /contract.jsonld", "@type": "ContractAgreement", "@id": "{?contractID}", "refersTo": "https://w3id.org/idsa/code /policytemplate/PermitUsage" , "targetArtifact": "{?content}", "provider": "{?ProviderParticipantURI}", "consumer": "{?ConsumerParticipantURI}", "permission": [{ "action": "{?action}" }] }</pre>	<pre>--- version: 2 init: tags: # maps 'targetArtifact': "{?content}" to Nukleus Data Type DataA: ["http://localhost/ids/targetArtifact"] rules: - event: ["&create", "&read"] # mapped from: "action": "ids:read" condition: "(context.getPolicyContext().isTagged (context.getRootContext().typeName(),\ \"http://localhost/ids/targetArtifact\"))" action: "APPROVE" - event: ["&update"] condition: "(context.getPolicyContext().isTagged (context.getRootContext().typeName(),\ \"http://localhost/ids/targetArtifact\"))" action: "DECLINE"</pre>
---	---

Nukleus nutzt Tags für das Mapping von IDS targetArtifact auf Nukleus Typnamen. Dazu wird der entsprechende Nukleus Type mit der URL der targetArtifact getagged. Es ist darauf zu achten, dass die genutzten Tag sich nicht mit anderen Regeln überschneiden.

Die in der IDS Policy definierte Action (permission/action) muss anhand der folgenden Tabelle übersetzt werden. Die verwendeten Eventgruppe (&create, &read und &update) sind in der [Nukleus Dokumentation](#) hinterlegt.

IDS Action	ACCEPT Nukleus events	DECLINE Nukleus events
idsc:READ	&create, &read	&update
idsc:MODIFY	&create, &read, &update	
idsc:USE	&create, &read, &update	

Policyklasse 2: Verwendung von Daten verbieten

	D°	Nukleus
Umgesetzt	✓	✓
Unter den Einschränkungen, welche für Policyklasse 1 definiert wurden.		

Das zweite der simpelsten Policyklassen in den IDS. Die Nutzung von Daten für bestimmte Zwecke wird verboten.

Eingaben

Keine weiteren Eingaben notwendig.

Details zur Umsetzung in D°

Die allgemeine Funktionsweise der Auswertung und Abbildung von IDS Komponenten auf D° Konzepte ist identisch zu der Umsetzung von Klasse 1.

Einen Unterschied findet man erst in der anschließenden Auswertung der Policy. Während die Umsetzung von Klasse 1 einen Whitelisting-Ansatz verfolgt, wird für Klasse 2 ein Blacklisting-Ansatz umgesetzt. Daraus folgt, dass alle Actions, welche nicht explizit in der Policy verboten sind erlaubt sind.

Das nachfolgende Beispiel zeigt die entsprechende Policyklasse, ausgedrückt im IDS-Informationsmodell und die Definition einer entsprechenden D°-Policy.

```
IDS.DenyUsage:
  degree.Constraint@IDSDenyUsage:
    name:
      Identifier: "IDS.DenyUsage"
    attribute:
      degree.Parameter:
        - name:
            Identifier: "provider"
            type:
              Type: "core.URI"
        - name:
            Identifier: "consumer"
            type:
              Type: "core.URI"
        - name:
            Identifier: "targetArtifact"
            type:
              Type: "core.URI"
        - name:
            Identifier: "action"
            type:
              Type: "IDS.Action"
```

Details zur Umsetzung in Nukleus

Die Invertierung des Pattern 1 geschieht in Nukleus einfach durch Invertierung der aus der Regeln resultierenden Aktionen:

- APPROVE DECLINE
- DECLINE APPROVE

Zur Laufzeit macht dieses Pattern nur wenig Sinn, da durch das Sperren der `&create` Events es nicht mehr möglich ist den Datentyp zu instanzieren (NEW_INSTANCE/CLONE). Mit den feingranulareren Nukleus Event ist es jedoch sehr wohl möglich sinnvolle Einschränkungen auf Datentypen vorzunehmen. Beispielsweise durch gezieltes Sperren des SERIALIZE Event kann der Datentyp nur lesend genutzt werden, aber die persistieren bzw. Übertragung in andere System wird verhindert bzw. erschwert. Leider ist dies in den entsprechenden IDS pattern so nicht vorgesehen.

Nukleus Policy (YAML):

```
---
version: 2
init:
  tags:
    DataA: ["http://localhost/ids/targetArtifact"]
rules:
  - event: ["DESERIALIZE"]
    condition: "(context.getPolicyContext().isTagged(context.getRootContext().typeName(),\
      \"http://localhost/ids/targetArtifact\"))"
    action: "DECLINE"
  - event: ["&create", "&read"]
    condition: "(context.getPolicyContext().isTagged(context.getRootContext().typeName(),\
      \"http://localhost/ids/targetArtifact\"))"
    action: "APPROVE"
```

Policyklasse 3: Datennutzung für Benutzer oder Systeme beschränken

	D°	Nukleus
Umgesetzt	✓	✗
Unter den Einschränkungen, welche für Policyklasse 1 definiert wurden.		

Diese Policyklasse wird verwendet, um die Durchführung von bestimmten Aktionen auf den Daten für bestimmte Nutzer oder Systeme zu verbieten, oder ausschließlich für diese zu erlauben.

Template für eine Policy, welche Nutzern mit bestimmten Rollen die Datennutzung erlaubt - im IDS Informationsmodell: <https://github.com/International-Data-Spaces-Association/InformationModel/blob/v3.1.0/codes/UsageAgreementByRoleTemplate.ttl>

Eingaben

- Rolle von Nutzern/Systemen, welche die Daten nutzen dürfen/nicht nutzen dürfen und/oder
- Namen von Nutzern/Systemen, welche die Daten nutzen dürfen/nicht nutzen dürfen

Details zur Umsetzung in D°

D° verfügt bereits über ein Subsystem zum Management von Kontextinformationen. Dieses Subsystem umfasst auch Daten über den Nutzer, welcher die Ausführung der Applikationslogik ausgelöst hat. Die notwendigen Information zur Validierung von erlaubten bzw. verbotenen Benutzern und Rollen sind somit bereits vorhanden. Daher verwendet die Umsetzung dieser Policyklasse das Subsystem für Kontextinformationen von D° und bezieht die Daten nicht aus einem dedizierten PIP. Generell ist der PIP ein Konzept aus dem XACML-Architekturvorschlag, welches D° in dieser Form nicht kennt.

Dabei werden die Daten, welche durch das entsprechende Subsystem bereitgestellt werden, automatisch an die jeweilige Data App angepasst. Bei einer D°-Applikation, welche über die Kommandozeile aufgerufen wird, werden die Daten Attribute über den Benutzer vom Betriebssystem verwendet. Bei einer Data App, die eine HTTP-Schnittstelle zur Verfügung stellt, kann beim Aufruf ein Json Web Token (JWT) verwendet werden, welches die entsprechenden Informationen enthält. Wird kein JWT übergeben, wird die Ausführung nicht automatisch abgebrochen, aber die Auswertung von bspw. Whitelisting Policies würde fehlschlagen.

Die Überprüfungen, welche diese Policy durchführt besteht aus zwei teilen. Zum einen wird die Überprüfung, welche in den Policyklassen 1 und 2 beschrieben wurde durchgeführt. Die konkrete Auswahl hängt davon ab ob eine verbotende (Blacklisting), oder eine erlaubende (Whitelisting) Policy definiert werden soll. Zusätzlich wird ein Abgleich zwischen den erlaubten/verbotenen Benutzernamen/Rollen und den durch den Aufrufer bereitgestellten durchgeführt.

Dabei findet keine 1:1 Übersetzung der IDS-Klasse in D°-Policies statt. Stattdessen gibt es je eigene Policies, um Benutzernamen und Rollen zu erlauben und zu verbieten. Darüber hinaus gibt es jeweils eigene Policies für Nutzernamen und Policies. Somit gibt es insgesamt vier verschiedene Policies, welche zur Umsetzung von Policyklasse 3 des IDS verwendet werden.

Das nachfolgende Beispiel zeigt die entsprechende Policyklasse, ausgedrückt im IDS-Informationsmodell und die Definition einer entsprechenden D°-Policy. Dabei zeigt das Beispiel nur zwei der vier verwendeten D°-Policies. Da die vier Policies, welche zur Umsetzung der Policyklasse verwendet werden, sehr ähnlich im Aufbau und ihrer Attribute sind, sind die beiden Beispiele ausreichend, um die anderen beiden Policies abzuleiten.

```
IDS.AllowUsageByUser:
degree.Constraint@IDSAllowUsageByUser:
  name:
    Identifier: "IDS.AllowUsageByUser"
  attribute:
    degree.Parameter:
      - name:
          Identifier: "provider"
          type:
            Type: "core.URI"
      - name:
          Identifier: "consumer"
          type:
            Type: "core.URI"
      - name:
          Identifier: "targetArtifact"
          type:
            Type: "core.URI"
      - name:
          Identifier: "action"
          type:
            Type: "IDS.Action"
      - name:
          Identifier: "user"
          type:
            Type: "core.Username"
```



```

IDS.DenyUsageByRole:
  degree.Constraint@IDSDenyUsageByRole:
    name:
      Identifier: "IDS.DenyUsageByRole"
    attribute:
      degree.Parameter:
        - name:
            Identifier: "provider"
            type:
              Type: "core.URI"
        - name:
            Identifier: "consumer"
            type:
              Type: "core.URI"
        - name:
            Identifier: "targetArtifact"
            type:
              Type: "core.URI"
        - name:
            Identifier: "action"
            type:
              Type: "IDS.Action"
        - name:
            Identifier: "role"
            type:
              Type: "core.Userrole"

```

Das Typsystem Nukleus hat keine Definition von unterschiedlichen Benutzern und deren Rollen. Ohne entsprechende Konzepte für diese Konstrukte, ist es nicht möglich diese Policyklasse umzusetzen.

Stattdessen wird diese Policyklasse ausschließlich in D° umgesetzt, was mehrere Gründe hat:

1. Das Management und die Auswertung von Nutzerdaten ist auf der Applikationsebene intuitiver als auf der Datenebene
2. D° verfügt bereits über entsprechende Konzepte, welche die richtigen Daten, abhängig von der Art der jeweiligen Applikation, bereitstellen

Policyklasse 4: Datennutzung für bestimmte Zwecke einschränken

	D°	Nukleus
Umgesetzt	✓	✗
Unter den Einschränkungen, welche für Policyklasse 1 definiert wurden.		

Um Datenverarbeitung für einen bestimmten Zweck zu erlauben/verbieten, wird diese Policyklasse verwendet.

Template für eine Policy, welche Datennutzung für einen bestimmten Zweck erlaubt - im IDS Informationsmodell: <https://github.com/International-Data-Spaces-Association/InformationModel/blob/v3.1.0/codes/UsageAgreementForPurposeTemplate.ttl>

Eingaben

- Der Zweck der durch die Policy eingeschränkt werden soll.

Details zur Umsetzung in D°

Bereits bei Policyklasse 4 wurde beschrieben, wie es möglich ist in D° die Eingabeparameter einer Aktivität mit Tags zu versehen, welche die Verwendung der übergebenen Daten beschreibt. Dies ist aber nicht ausreichend und nicht darauf ausgelegt, um den Einsatzzweck von Aktivitäten zu beschreiben. Aus diesem Grund ist es möglich auch Aktivitäten mit speziellen Tags zu versehen.

Dies ist sowohl für Definitionen von Aktivitäten, als auch die dazugehörigen Instanzen möglich. Die Entscheidung, wo welches Tag vermerkt wird (Definition vs. Instanz) ist nicht eingeschränkt, aber es gibt für die unterschiedlichen Tags unterschiedlich sinnvolle Einsatzgebiete. Zum einen gibt es Tags, welche das Verhalten bzw. die Struktur von Aktivitäten näher beschreiben. Beispiele hierfür wären die Tags `STATEFUL` und `STATELESS`. Da diese Beschreibungen für alle Instanzen, welche aus der Definition erzeugt werden gelten, ist es sinnvoll diese Art von Tags an die Definition der Aktivität anzubringen. Darüber hinaus gibt es Tags, welche den Verwendungszweck beschreiben. Beispiele hierfür sind `RISK_MANAGEMENT` und `MARKETING`. Da unterschiedliche Instanzen einer einzelnen Aktivität in unterschiedlichen Einsatzgebieten und zu unterschiedlichen Zwecken verwendet werden können, ist es sinnvoll diese Tags an die Instanz der Aktivität zu beschreiben. Trotz diesen Empfehlungen ist es aber möglich jedes dieser Tags sowohl an Definitionen als auch an Instanzen zu heften.

Dies hätte vermieden werden können, indem zwei unterschiedliche Arten von Tags verwendet würden. Davon wurde abgesehen, da sich dadurch in Kombination mit den Tags für Eingabeparametern von Aktivitäten drei unterschiedliche Arten von Tags existieren würden und hierdurch Verwirrungen bei den Anwendern entstehen könnten.

Da die Tags durch die Entwickler, welche die Definitionen und Instanzen von Aktivitäten anlegen vergeben werden, ist hier ein verantwortungsvolles Vorgehen bei der Vergabe von Tags zwingend notwendig, um eine korrekte Funktionalität des Policysystems gewährleisten zu können. Dieses Problem lässt sich nicht anders lösen, da es rein aus dem Programmcode einer Aktivität nicht ableiten lässt, für welchen Zweck er angewendet werden soll. Da an dieser Stelle keine automatische Schlussfolgerung möglich ist, ist es notwendig externe Informationen zu verwenden, die in diesem Fall vom Entwickler bereitgestellt werden müssen.

Das nachfolgende Beispiel zeigt die entsprechende Policyklasse, ausgedrückt im IDS-Informationsmodell und die Definitionen der entsprechenden D°-Policies.

```
IDS.AllowUsageForPurpose:
  degree.Constraint@IDSAllowUsageForPurpose:
    name:
      Identifier: "IDS.AllowUsageForPurpose"
    attribute:
      degree.Parameter:
        - name:
            Identifier: "provider"
            type:
              Type: "core.URI"
        - name:
            Identifier: "consumer"
            type:
              Type: "core.URI"
        - name:
            Identifier: "targetArtifact"
            type:
              Type: "core.URI"
        - name:
            Identifier: "action"
            type:
              Type: "IDS.Action"
        - name:
            Identifier: "purpose"
            type:
              Type: "degree.ActivityTag"
```

```
IDS.DenyUsageForPurpose:
  degree.Constraint@IDSDenyUsageForPurpose:
    name:
      Identifier: "IDS.DenyUsageForPurpose"
    attribute:
      degree.Parameter:
        - name:
            Identifier: "provider"
            type:
              Type: "core.URI"
        - name:
            Identifier: "consumer"
            type:
              Type: "core.URI"
        - name:
            Identifier: "targetArtifact"
            type:
              Type: "core.URI"
        - name:
            Identifier: "action"
            type:
              Type: "IDS.Action"
        - name:
            Identifier: "purpose"
            type:
              Type: "degree.ActivityTag"
```

Da das Typsystem einer Programmiersprache keine Kenntnis über den Zweck der Verwendung der Daten verfügt, wird diese Policyklasse ausschließlich in D° umgesetzt.

Policyklasse 5: Datennutzung nach Ereignis einschränken

	D°	Nukleus
Umgesetzt	✗	✓
Unter den Einschränkungen, welche für Policyklasse 1 definiert wurden.		

Datennutzung soll nach dem Auftreten eines bestimmten Ereignisses erlaubt/verboten werden. Die Information, ob das Ereignis eingetreten ist kommt aus einem PIP.

Template für eine Policy, welche Datennutzung nach dem Eintreten eines Ereignisses erlaubt - im IDS Informationsmodell: <https://github.com/International-Data-Spaces-Association/InformationModel/blob/v3.1.0/codes/UsageAgreementAfterEventTemplate.ttl>

Template für eine Policy, welche Datennutzung nach Bezahlung erlaubt - im IDS Informationsmodell: <https://github.com/International-Data-Spaces-Association/InformationModel/blob/v3.1.0/codes/UsageAgreementAfterPaymentTemplate.ttl>

Eingaben

- Das Ereignis auf das sich die Policy bezieht
- Adresse des PIPs, welcher für Abfragen verwendet wird

Details zur Umsetzung in Nukleus

Diese Pattern nutzt ein externes Flag, um die Datenutzung freizuschalten. Da Nukleus (und D°) keinen sogenannten PIP verwenden, wird ein beliebiger HTTP Service hierzu verwendet. Dieser ist als Service im PolicyContext definiert und die Service-Antwort wird in der Policy-Auswertung verwendet. Die Antwort des Service kann zwischengespeichert werden, um die Dienste nicht mit Anfragen zu fluten. Die Antwort-"Lifetime" kann als Wert in Millisekunden an die Service-URL angehängt werden (mit Leerzeichen getrennt).

IDS-ODRL (JSON-LD)	Nukleus Policy (YAML)
<pre>{ "@context": "http://www.w3id.org/ids/contract.jsonld", "@type": "ContractAgreement", "@id": "{?contractID}", "refersTo": "https://w3id.org/idsa/code/policytemplate/UsageAgreementAfterPaymentTemplate", "targetArtifact": "{?content}", "provider": "{?ProviderParticipantURI}", "consumer": "{?ConsumerParticipantURI}", "action": "{?action}", "constraint": { "leftOperand": "payAmount", "operator": "eq", "rightOperandReference": "{?PipURI}" } }</pre>	<pre>--- version: 2 init: tags: DataA: ["http://localhost/ids/targetArtifact"] services: EventA: "http://nukleus.ralf-nagel.de/dip/get.php?EventA" rules: - event: ["&create", "&read", "&update"] condition: "(context.getPolicyContext().isTagged(\ context.getRootContext().typeName(), \ \"http://localhost/ids/targetArtifact\") && \ (! \"true\".equals(context.getPolicyContext().readText(\"EventA\"))))" action: "DECLINE"</pre>

Da D° Entscheidungen auf der Basis von Ereignissen höchstens auf der Ebene von Applikationen und Aktivitäten durchführen kann, findet keine Umsetzung dieser Policyklasse in D° statt.

Policyklasse 6: Datennutzung in Zeitraum einschränken

	D°	Nukleus
Umgesetzt	✗	✓
Unter den Einschränkungen, welche für Policyklasse 1 definiert wurden.		

Die Nutzung von Daten soll für einen bestimmten Zeitraum erlaubt/verboten werden.

Template für eine Policy, welche Datennutzung in einem definiertem Zeitraum erlaubt - im IDS Informationsmodell: <https://github.com/International-Data-Spaces-Association/InformationModel/blob/v3.1.0/codes/TimeRestrictedUsageAgreementTemplate.ttl>

Eingaben

- Startzeit für das Zeitintervall
- Endzeit für das Zeitintervall

Details zur Umsetzung in Nukleus

Nukleus verwendet das aus Pattern 1/2 bekannte Verfahren um die Zuordnung zwischen Nukleus Type und IDS `targetArtifact` vorzunehmen. Auch die Zuordnung der `Action` ist wie gehabt. Für die Evaluierung von temporalen Bedingungen nutzt Nukleus den Java Datentypen `LocalDateTime` und stellt im `Policy Macro` die Funktion `intervall()` bereit. Die Berechnung in `LocalDateTime` erfolgt ohne die Berücksichtigung von Zeitzonen, d.h. die Konvertierung in die lokale Zeit des Servers muss vom ODRL Converter vorgenommen werden.

IDS-ODRL (JSON-LD)	Nukleus Policy (YAML)
<pre>{ "@context": "http://www.w3id.org/ids/contract. jsonld", "@type": "ContractAgreement", "@id": "{?contractID}", "refersTo": "https://w3id.org/idsa/code /policytemplate /TimeRestrictedUsageAgreementTemplate" , "targetArtifact": "{?content}", "provider": "{?ProviderParticipantURI}", "consumer": "{?ConsumerParticipantURI}", "permission": { "action": "{?action}" , "constraint" : [{ "leftOperand" : "DATE_TIME", "operator" : "gt", "rightOperand" : "\"yyyy-mm-ddThh:mm:ss. xxx+hh:mm\"^^xsd:dateTime" } , { "leftOperand" : "DATE_TIME", "operator" : "lt", "rightOperand" : "\"yyyy-mm-ddThh:mm:ss. xxx+hh:mm\"^^xsd:dateTime" }] } }</pre>	<pre>--- version: 2 init: tags: DataA: ["{?content}"] rules: - event: ["&create", "&read", "&update"] condition: "(context.getPolicyContext().isTagged (context.getRootContext().typeName(), \ \"{?content}\") && \ (! Policy.interval(context, \"1980-01-01T00: 00:00.000\", \"2000-01-01T00:00:00.000\")))" action: "DECLINE"</pre>

Diese Policyklasse ist nicht in D° umgesetzt worden. D° erlaubt es Policies zu definieren, welche die Nutzung von Applikationen und Aktivitäten auf/für einen Zeitraum einzuschränken, aber eine Umsetzung solcher Policies auf Ebene der Datentypen wird nur durch Nukleus geboten. Der Unterschied dabei ist die Betrachtungsrichtung. In Nukleus geht es darum einzuschränken, wann Daten verwendet bzw. nicht verwendet werden dürfen. D° erlaubt identische Einschränkungen, aber nicht für die verwendeten Daten, sondern für die erzeugen Applikationen.

Policyklasse 7: Datennutzung N mal erlauben

	D°	Nukleus
Umgesetzt	✗	✓
Unter den Einschränkungen, welche für Policyklasse 1 definiert wurden.		

Die Verwendung von Daten soll für eine Bestimmte Anzahl von Aufrufen erlaubt werden.

Template für eine Policy, welche Datennutzung N -mal erlaubt - im IDS Informationsmodell: <https://github.com/International-Data-Spaces-Association/InformationModel/blob/v3.1.0/codes/UsageAgreementNUsagesTemplate.ttl>

Eingaben

- Die maximal erlaubte Menge von Datennutzungen
- Die Anzahl bereits erfolgter Datennutzungen

Da Nukleus in seiner Rolle als Typsystem von D° besser dazu geeignet ist genau zu bestimmen wann Daten verwendet werden, findet keine Umsetzung der Policyklasse in D° statt.

Details zur Umsetzung in Nukleus

Der PolicyContext von Nukleus besitzt den Datentyp COUNTER (in Java: AtomicLong) um beliebige Operationen zu zählen und durch Regeln zu überwachen. Durch die Wahl der Keys können Zähler pro Instanz/pro Datentyp oder pro Attribut realisiert werden.

IDS-ODRL (JSON-LD)	Nukleus Policy (YAML)
<pre>{ "@context": "http://www.w3id.org/ids/contract. jsonld", "@type": "ContractAgreement", "@id": "{?contractID}", "refersTo": "https://w3id.org/idsa/code /policytemplate/UsageAgreementNUsagesTemplate" , "targetArtifact": "{?ArtifactURI}", "provider": "{?ProviderParticipantURI}", "consumer": "{?ConsumerParticipantURI}", "permission": { "action": "{?action}", "constraint" : { "leftOperand" : "quantity", "operator" : "lt", "rightOperand" : "{?n + 1}" } } }</pre>	<pre>--- version: 2 init: tags: DataA: ["http://localhost/ids/targetArtifact"] rules: - event: ["&read"] condition: "(context.getPolicyContext().isTagged (context.getRootContext().typeName(), \ \"http://localhost/ids/targetArtifact\") && \ (context.getPolicyContext().incrementCounter (context.identity()) > 3))" action: "DECLINE"</pre>

Policyklasse 8: Datennutzung mit anschließender Löschung erlauben

	D°	Nukleus
Umgesetzt	✗	✗

Die Verwendung von Daten wird bis zu einem Zeitpunkt in der Zukunft erlaubt. Sobald dieser Zeitpunkt erreicht ist, müssen die Daten und alle eventuell erzeugten Kopien gelöscht werden. Eine Unterlassung weiterer Nutzung nach Ablauf des erlaubten Nutzungszeitraums, reicht nicht aus, um die Policy zu erfüllen.

Template für eine Policy, welche Datennutzung bis zu einem definiertem Zeitpunkt erlaubt - im IDS Informationsmodell: <https://github.com/International-Data-Spaces-Association/InformationModel/blob/v3.1.0/codes/UsageAgreementUntilDeletionTemplate.ttl>

Eingaben

- Der Zeitpunkt zu dem die Datennutzung nicht mehr erlaubt ist

Details zur Umsetzung

Daten, welche persistiert worden sind, befinden sich nicht mehr unter der Hoheit von D° und Nukleus. Sie befinden sich in Dateien, Datenbanken oder anderen Systemen, welche nicht durch D° und Nukleus kontrolliert werden können. Darüber hinaus sind vielleicht Backups von den Daten angelegt worden oder rechtliche Vorschriften (bspw. Aufbewahrungspflichten) stehen der Policy entgegen. Auf Grund dieser Problematiken und Unklarheiten ist diese Policyklasse weder in D° noch in Nukleus implementiert.

Policyklasse 9: Verwendung von modifizierten Daten (in Transit) erlauben

	D°	Nukleus
--	----	---------

Umgesetzt	✗	✗
-----------	---	---

Bevor die Daten an verarbeitende Applikationen übergeben werden, ist es notwendig, dass die Daten gewisse Anforderungen erfüllen. Beispielsweise kann eine Policy definiert werden, welche eine Anonymisierung erfordert, bevor eine Data App Zugriff auf die Daten erhält.

Eingaben

Keine weiteren Eingaben notwendig. Aber die vorhandenen Eingaben werden anders interpretiert als in den vorherigen Policyklassen. Die Actions müssen zwingend auf den Daten durchgeführt werden.

Details zur Umsetzung

Da weder Nukleus noch D° Kontrolle über den eigentlichen Transport von Daten zwischen unterschiedlichen Komponenten (bspw. Data Apps) haben, ist diese Policyklasse weder für D°, noch für Nukleus umgesetzt.

Policyklasse 10: Verwendung von modifizierten Daten (at Rest) erlauben

	D°	Nukleus
Umgesetzt	✓	✗
Unter den Einschränkungen, welche für Policyklasse 1 definiert wurden.		

Bevor die Daten in ein persistierendes System überführt werden und damit in einen inaktiven Zustand übergehen, ist es notwendig Modifikationen an den Daten vorzunehmen (bspw. Anonymisierung).

Eingaben

Es sind keine zusätzlichen Eingaben notwendig, aber die gegebenen Actions werden anders als bei den anderen Policyklassen interpretiert (s.u.)

Details zur Umsetzung in D°

Die Umsetzung dieser Policyklasse besteht aus mehreren Stufen. In der ersten Phase der Überprüfung wird getestet, ob die aktuelle Aktivität (für die auch die Policies ausgewertet werden), über das Tag `PERSISTING` verfügt, denn nur dann müssen weitere Überprüfungen durchgeführt werden. Anschließend wird überprüft, ob der Parameter, der die Daten enthält, an denen diese Policy hängt, über das Tag `PERSIST` verfügt. Ist auch dies der Fall, muss die Evaluierung fortgesetzt werden. In diesem Fall muss der entsprechende Eingabeparameter der Aktivität auch über alle Tags verfügen, welche den Actions aus der Policy entsprechen. Ist dies der Fall, ist die Ausführung erlaubt, andernfalls nicht.

Daraus folgt, dass wenn eine Policy verwendet werden soll, welche die Persistierung der Daten nur nach vorhergehender Anonymisierung erlaubt, die Anonymisierung in der selben Aktivität wie die Persistierung durchgeführt werden muss, da ansonsten die Policy nicht erfüllbar ist. Dies kann in einer zukünftigen Version generalisiert und somit verbessert werden, indem auf die Verwendung Tags, welche an Nukleus-Instanzen hängen, gewechselt wird. In diesem Fall würde es ausreichen, wenn die Daten von einer vorherigen Aktivität anonymisiert worden wären. Die persistierende Aktivität kann dann den Zustand der Daten anhand der Tags überprüfen und somit sicherstellen, dass eine vorhergehende Anonymisierung stattgefunden hat.

Da Nukleus nicht bestimmen kann, ob Daten persistiert werden, wird diese Policyklasse nur in D° umgesetzt. Die Serialisierung einer Nukleus-Instanz, welche Daten enthält, kann aus verschiedenen Gründen erfolgen (bspw. Erzeugung eines JSON-Objekts als Rückgabewert), weswegen nicht automatisch auf eine Persistierung geschlossen werden kann.

Das nachfolgende Beispiel zeigt die entsprechende Policyklasse, ausgedrückt im IDS-Informationsmodell und die Definition der entsprechenden D°-Policy.

```

IDS.RequireModificationAtRest:
  degree.Constraint@IDSRequireModificationAtRest:
    name:
      Identifier: "IDS.RequireModificationAtRest"
    attribute:
      degree.Parameter:
        - name:
            Identifier: "provider"
            type:
              Type: "core.URI"
        - name:
            Identifier: "consumer"
            type:
              Type: "core.URI"
        - name:
            Identifier: "targetArtifact"
            type:
              Type: "core.URI"
        - name:
            Identifier: "action"
            type:
              Type: "IDS.Action"

```

Policyklasse 11: Informationen über Datennutzung loggen

	D°	Nukleus
Umgesetzt	✗	✓
Unter den Einschränkungen, welche für Policyklasse 1 definiert wurden.		

Die Verwendung von Daten ist erlaubt, muss aber durch einen (externen) Logging Service erfasst werden.

Template für eine Policy, welche Datennutzung erlaubt und erfordert, dass jeder Datenzugriff geloggt wird - im IDS Informationsmodell: <https://github.com/International-Data-Spaces-Association/InformationModel/blob/v3.1.0/codes/UsageAgreementLogAccessTemplate.ttl>

Eingaben

- Die Adresse des entfernten Logging Service, welcher verwendet werden soll

Details zur Umsetzung in Nukleus

Diese Pattern beinhaltet eigentlich keine Datennutzungsbedingung sondern besteht aus eine Verpflichtung (obligation). In Nukleus entspricht dies eher einem Event-Listener als ein Policy.

Das nachfolgende Beispiel nutzt einen Seiteneffekt von Java um ein identisches Verhalten nachzubilden (die rechte Seite der UND Verknüpfung wird nur ausgewertet, falls die linke Seite schon WAHR ist).

IDS-ODRL (JSON-LD)	Nukleus Policy (YAML)
--------------------	-----------------------

```
{
  "@context": "http://www.w3id.org/ids/contract.jsonld",
  "@type": "ContractAgreement",
  "@id": "{?contractID}",
  "refersTo": "https://w3id.org/idsa/code/policytemplate/UsageAgreementLogAccessTemplate",
  "targetArtifact": "{?ArtifactURI}",
  "provider": "{?ProviderParticipantURI}",
  "consumer": "{?ConsumerParticipantURI}",
  "permission": {
    "action": "{?action}"
  },
  "obligation": {
    "action": "https://w3id.org/idsa/code/action/LOG",
    "constraint": {
      "leftOperand": "systemDevice",
      "operator": "eq",
      "rightOperand": "{?LoggingEndpoint}"
    }
  }
}
```

```
---
version: 2
init:
  tags:
    DataA: ["http://localhost/ids/targetArtifact"]
rules:
  - event: ["&create"]
    condition: "(context.getPolicyContext().isTagged(context.typeName(), \"http://localhost/ids/targetArtifact\") && \"(nukleus.policy.Policy.wget(\"http://nukleus.ralf-nagel.de/log/log.php?targetArtifact=URI&identity=\" + context.getInstance().identity())))"
    action: "APPROVE"
```

Da Nukleus wesentlich besser dazu geeignet ist genauen Nutzungszeitpunkte von Daten zu erfassen als D°, findet eine Umsetzung dieser Policyklasse nur in Nukleus statt.

Policyklasse 12: Entitäten über Datennutzung informieren

	D°	Nukleus
Umgesetzt		
Unter den Einschränkungen, welche für Policyklasse 1 definiert wurden.		

Die Datennutzung ist erlaubt, aber bestimmte Entitäten (bspw. der Data Provider) sollen über die Datennutzung proaktiv informiert werden.

Eingaben



- Entitäten (Identifizier oder Adressen) die über Datennutzung informiert werden sollen

Details zur Umsetzung in Nukleus

Diese Pattern entspricht Nr. 11 mit einer geänderten URL. Wir verweisen daher hier auf die Dokumentation von Policyklasse 11.

Da Nukleus wesentlich besser dazu geeignet ist genauen Nutzungszeitpunkte von Daten zu erfassen als D°, findet eine Umsetzung dieser Policyklasse nur in Nukleus statt.

Policyklasse 13: Daten unter gewissen Umständen Teilen

	D°	Nukleus
Umgesetzt		

Eine sehr generische Policyklasse, welche es erlaubt Daten mit anderen Parteien zu teilen, sofern gewisse Umstände gegeben sind/Bedingungen erfüllt sind.

Eingaben

- Umstände welche erfüllt werden müssen
- Parteien, mit denen die Daten bei gegebenen Umständen geteilt werden dürfen

Details zur Umsetzung

Da diese Policyklasse extrem generisch ist, ist eine Umsetzung weder in D° noch Nukleus einfach möglich. Da der Einsatzgebiet dieser Policyklasse aktuell noch unbekannt ist, ist es nicht möglich sinnvolle Einschränkungen zu definieren, welche eine technische Umsetzung erlauben würden. Aus diesem Grund findet eine Umsetzung dieser Policyklasse weder in D° noch in Nukleus statt.

Policyklasse 14: Verwendung von Daten fein-granular einschränken

	D°	Nukleus
Umgesetzt	✗	✗

Diese Policyklasse ist eine Verfeinerung der Policyklassen 1 und 2. Neben der erlaubten/verbotenen Aktion werden weitere Einschränkungen definiert, welche erfüllt werden müssen. Diese Policyklasse ist das generischste aller Policyklassen.

Template für eine Policy, welche Datennutzung in einem bestimmten Geofence erlaubt - im IDS Informationsmodell: <https://github.com/International-Data-Spaces-Association/InformationModel/blob/v3.1.0/codes/SpatialRestrictedUsageAgreementTemplate.ttl>

Eingaben

- Zu erfüllende Einschränkungen










Details zur Umsetzung

Da diese Policyklasse noch generischer als Policyklasse 13 ist gibt es keine Umsetzung für D° und Nukleus.

Ergebnisse für das Informationsmodell v4.0.0

Während in den ursprünglichen 14 Policyklassen sowohl White- als auch Blacklisting unterstützt wurde, sind in den 20 Policyklassen des Informationsmodells 4.0.0 nur noch Whitelisting Policies vorhanden.

	Name	Beschreibung	Abbildung auf vorherige Policyklassen 1	In D° /Nukleus verwendbar	Anmerkung
1	Purpose Restricted Policy	Die Nutzung von Daten soll nur für bestimmte Zwecke erlaubt werden. Es ist vorgesehen, dass ein PIP diese Daten bereitstellt.	Entspricht Policyklasse 4	✓ ✗	D° erlaubt es den Nutzungszweck an Aktivitäten innerhalb von Data Apps zu annotieren. Eine Policy kann entsprechend prüfen, ob ein bestimmter Zweck bei Aktivitäten vorhanden ist, welche die Daten, die mit der Policy versehen sind, vorhanden ist. Da das PIP Konzept in D° nicht verwendet wird und es aktuell keine "zentralen" und standardisierten PIPs gibt, kann D° für die Auswertung keinen PIP verwenden. Die für die Auswertung notwendigen Daten sind alle in der Data App vorhanden, weswegen die Verwendung eine PIPs auch nicht notwendig ist. Es muss nur sichergestellt werden, dass der Zweck aus der IDS-Policy auf einen D°-Zweck (ActivityTag) abgebildet wird.
2	Connector Based Policy	Die Nutzung von Daten soll nur für einen bestimmten Connector erlaubt werden.	Teilmenge von Policyklasse 3	? ✗	Diese Policy ließe sich theoretisch in D° abbilden, sollte aber eigentlich bereits auf der Ebene des Connectors überprüft werden, da es nicht sinnvoll ist Daten an Applikationen weiterzuleiten, wenn bekannt ist, dass diese nicht verarbeitet werden dürfen. Die Auswertung einer Policy sollte immer frühestmöglich erfolgen.
3	Distribute Encrypted Policy	Erlaubt, dass die Daten für die Action DISTRIBUTE verwendet werden, sofern die preDuty ENCRYPT erfüllt ist. Darüber hinaus ist eine beliebige Menge von zusätzlichen constraints, preDutys und post Dutys für diese Policy möglich.	Verfeinerung von Policyklasse 13	⚠ ✗	D° unterstützt das Konzept von Duties nicht. Diese erfordern es, dass die Usage Control Lösung in den Ablauf der Applikation eingreift, um die definierten Duties zu erfüllen. Stattdessen ist es möglich in D° eine Policy zu definieren, welche sicherstellt, dass eine Aktivität, welche für den Parameter, der die Daten mit der Policy entgegennimmt, DISTRIBUTE nur ausführt, wenn die Daten bereits mit ENCRYPT markiert sind. Die Umsetzung dieser Policy basiert stark auf Interaktionen zwischen den Mechanismen von D° und Nukleus. Wird eine der erlaubten Ergänzungen der Policy um beliebige constraints, preDutys und postDutys verwendet, ist es nur auf Basis des Einzelfalls und manuell möglich diese Policy in D° zu übersetzen.
4	Event Restricted Policy	Erlaube die Nutzung von Daten bei einem bestimmten Ereignis (bspw. während Veranstaltungen oder nach Unfällen). Es ist vorgesehen, dass ein PIP diese Daten bereitstellt.	Entspricht Policyklasse 5	✗ ✓	Das Konzept des PIP wird weder in D° noch in Nukleus verwendet. Nukleus bietet aber die Möglichkeit Anfragen an Services, welche eine HTTP(S)-Schnittstelle bieten zu senden und die Rückgabewerte in die Policyauswertung mit einzubeziehen.

5	N Times Policy	<p>Erlaube die Nutzung für Daten <i>N</i> mal.</p> <p>Darüber hinaus ist eine beliebige Menge von zusätzlichen <code>constraints</code>, <code>preDutys</code> und <code>postDutys</code> für diese Policy möglich.</p> <p>Es ist vorgesehen, dass ein PIP die Zugriffe zählt.</p>	Erweiterung von Policyklasse 7		<p>Der <code>PolicyContext</code> von Nukleus besitzt den Datentyp <code>COUNTER</code> (in Java: <code>AtomicLong</code>) um beliebige Operationen zu zählen und durch Regeln zu überwachen. Durch die Wahl der Keys können Zähler pro Instanz/pro Datentyp oder pro Attribut realisiert werden.</p> <p>Auch wenn Nukleus nicht das PIP Konzept verwendet, besteht die Möglichkeit Services mit HTTP-Schnittstellen aufzurufen und somit den Zählerstand, den ein PIP bereitstellt, zu verwenden.</p> <p>Ist kein PIP vorhanden kann Nukleus auch lokal eigenständig die Nutzungen zählen.</p> <p>Wird eine der erlaubten Ergänzungen der Policy um beliebige <code>constraints</code>, <code>preDutys</code> und <code>postDutys</code> verwendet, ist es nur auf Basis des Einzelfalls und manuell möglich diese Policy in Nukleus zu übersetzen.</p>
6	Rental Policy	<p>Erlaube die Nutzung von Daten, sofern der zur Policy gehörende Contract nicht abgelaufen ist, eine <code>preDuty</code>, welche eine Zahlung überprüft, erfüllt ist und die Auswertung der Policy innerhalb eines definierten Zeitraums stattfindet.</p> <p>Kurz gesagt bildet diese Policy das Vermieten von Daten ab.</p> <p>Darüber hinaus ist eine beliebige Menge von zusätzlichen <code>constraints</code>, <code>preDutys</code> und <code>postDutys</code> für diese Policy möglich.</p> <p>Es ist möglich, dass ein PIP diese Daten bereitstellt.</p>	Keine Entsprechung in den ursprünglichen Policyklassen.		<p>Diese Policy sollte nicht in einem laufendem Workflow überprüft werden, sondern zu Beginn des Workflows. Somit ist eine Überprüfung auf Applikationsebene nicht sinnvoll und es müsste eine Überprüfung auf Connector Ebene durchgeführt werden.</p>
7	Role Based Policy	<p>Erlaube die Nutzung von Daten nur, wenn der Nutzer eine bestimmte Rolle innerhalb seines Unternehmens hat <u>und</u> eine bestimmte Rolle besitzt, welche über das Unternehmen hinaus geht.</p> <p>Es ist vorgesehen, dass ein PIP diese Daten bereitstellt.</p>	Teilmenge und Verfeinerung von Policyklasse 3		<p>Diese Policy kann theoretisch in D° umgesetzt werden, aber da in einem Workflow jeder einzelne Arbeitsschritt diese Policy überprüfen muss und das Ergebnis stets das selbe ist, ist eine Überprüfung dieser Policy auf Connector Ebene sinnvoller.</p>
8	Permission Policy	<p>Erlaube die Nutzung von Daten für bestimmte Aktion.</p> <p>Darüber hinaus ist eine beliebige Menge von zusätzlichen <code>constraints</code>, <code>preDutys</code> und <code>postDutys</code> für diese Policy möglich.</p>	Entsprechung von Policyklasse 14		<p>Diese Policyklasse ist ohne Verwendung der möglichen Erweiterungen sowohl in D° als auch in Nukleus für die unterstützten Actions möglich.</p> <p>Wird eine der erlaubten Ergänzungen der Policy um beliebige <code>constraints</code>, <code>preDutys</code> und <code>postDutys</code> verwendet, ist es nur auf Basis des Einzelfalls und manuell möglich diese Policy in Nukleus zu übersetzen.</p>
9	Sales Policy	<p>Erlaube die Nutzung von Daten für bestimmte Aktion, falls die <code>preDuty</code> erfüllt ist, welche Ausdrückt, dass eine bestimmte Zahlung erfolgt ist.</p> <p>Darüber hinaus ist eine beliebige Menge von zusätzlichen <code>constraints</code>, <code>preDutys</code> und <code>postDutys</code> für diese Policy möglich.</p>	Keine Entsprechung in den ursprünglichen Policyklassen		<p>Nukleus unterstützt das Konzept von Duties nicht. Diese erfordern es, dass die Usage Control Lösung in den Ablauf der Applikation eingreift, um die definierten Duties zu erfüllen.</p> <p>Stattdessen ist es möglich eine Policy zu definieren welche überprüft, ob die notwendige Zahlung bereits erfüllt ist. Dies wäre dann eine Ausprägung von Policyklasse 4.</p> <p>Wird eine der erlaubten Ergänzungen der Policy um beliebige <code>constraints</code>, <code>preDutys</code> und <code>postDutys</code> verwendet, ist es nur auf Basis des Einzelfalls und manuell möglich diese Policy in Nukleus zu übersetzen.</p>
10	Trust Level Policy	<p>Erlaube die Nutzung von Daten für bestimmte Security Level, wobei Trust+ immer erlaubt ist.</p>	Keine Entsprechung in den ursprünglichen Policyklassen		<p>Diese Policy ließe sich theoretisch in D° abbilden, sollte aber eigentlich bereits auf der Ebene des Connectors überprüft werden, da es nicht sinnvoll ist Daten an Applikationen weiterzuleiten, wenn bekannt ist, dass diese nicht verarbeitet werden dürfen.</p> <p>Die Auswertung einer Policy sollte immer frühestmöglich erfolgen.</p>
11	Spatial Policy	<p>Erlaube die Nutzung von Daten, wenn der ausführende Connector sich innerhalb definierter Grenzen befindet.</p> <p>Darüber hinaus ist eine beliebige Menge von zusätzlichen <code>constraints</code>, <code>preDutys</code> und <code>postDutys</code> für diese Policy möglich.</p>	Keine Entsprechung in den ursprünglichen Policyklassen		<p>Ohne existierende Hilfsmethoden, welche die Auflösung von Geolokationen erlaubt (bspw. als Teil des PIPs) wird diese Policy nicht in D° und Nukleus nutzbar sein, da erheblicher Overhead für jede einzelne Applikation entstehen würde.</p> <p>Darüber hinaus sollte diese Policy auf Connector Ebene überprüft werden, da die Position eines Connectors kein hochdynamisches Datum darstellt.</p> <p>Die Auswertung einer Policy sollte immer frühestmöglich erfolgen.</p>
12	Swap Policy	<p>Bildet einen Datenaustausch zwischen zwei Participants ab. Jeder der beiden Participants erhält die Erlaubnis ein Datenasset zu benutzen.</p> <p>Die beiden Erlaubnisse sind unabhängig voneinander und können unterschiedlich ausgestaltet sein.</p> <p>Darüber hinaus ist eine beliebige Menge von zusätzlichen <code>constraints</code>, <code>preDutys</code> und <code>postDutys</code> für jede Erlaubnis möglich.</p>	Keine Entsprechung in den ursprünglichen Policyklassen		<p>Eine solche Policy ließe sich besser mit anderen Technologien (bspw. Blockchain) lösen, als auf der Applikationsebene.</p>
13	Duration Policy	<p>Erlaubt die Nutzung von Daten für eine bestimmte Zeit.</p> <p>Darüber hinaus ist eine beliebige Menge von zusätzlichen <code>constraints</code>, <code>preDutys</code> und <code>postDutys</code> für diese Policy möglich.</p>	Keine Entsprechung in den ursprünglichen Policyklassen.		<p>Die Policy ist nicht ausreichend spezifiziert, um eine eindeutige Aussage zur Umsetzbarkeit in D° und Nukleus zu treffen.</p>

14	Interval Policy	Erlaubt die Nutzung innerhalb eines definierten Zeitfensters. Darüber hinaus ist eine beliebige Menge von zusätzlichen constraints, preDutys und post Dutys für diese Policy möglich.	Verfeinerung von Policyklasse 6		Für die Evaluierung von temporalen Bedingungen nutzt Nukleus den Java Datentypen LocalDateTime und stellt im Policy Macro die Funktion interval() bereit. Die Berechnung in LocalDateTime erfolgt ohne die Berücksichtigung von Zeitzonen, d.h. die Konvertierung in die lokale Zeit des Servers muss vom ODRL Converter vorgenommen werden. Wird eine der erlaubten Ergänzungen der Policy um beliebige constraints, preDutys und postDutys verwendet, ist es nur auf Basis des Einzelfalls und manuell möglich diese Policy in Nukleus zu übersetzen.
15	Interval Deletion Policy	Erlaubt die Nutzung der Daten innerhalb eines definierten Zeitfensters und erfordert die anschließende Löschung in Form einer postDuty. Darüber hinaus ist eine beliebige Menge von zusätzlichen constraints, preDutys und post Dutys für diese Policy möglich.	Entspricht Policyklasse 8		Eine anschließende Löschung würde es notwendig machen, dass die Policy eigenständig Aktivitäten ausführt. Dies wird weder von D° und Nukleus unterstützt.
16	Logging Policy	Erlaubt die Nutzung von Daten, sofern die postDuty, welche ein loggen des Datenzugriffs erfordert, erfüllt ist. Somit muss jede Datennutzung nach der Nutzung geloggt werden. Die Policy enthält keine Möglichkeit einen externen Loggingservice anzugeben, welcher für das Logging verwendet werden soll. Darüber hinaus ist eine beliebige Menge von zusätzlichen constraints, preDutys und post Dutys für diese Policy möglich.	Keine Entsprechung in den ursprünglichen Policyklassen.		Nukleus erlaubt es Zugriffe auf Daten im Log (bspw. Syslog/log4j) festzuhalten. Wird eine der erlaubten Ergänzungen der Policy um beliebige constraints, preDutys und postDutys verwendet, ist es nur auf Basis des Einzelfalls und manuell möglich diese Policy in Nukleus zu übersetzen.
17	Notification Policy	Erlaubt die Nutzung von Daten, sofern die postDuty erfüllt ist, welche es erfordert, dass die Aktion NOTIFY ausgeführt wird. Bei dieser Policy muss jeder Datenzugriff an einen externen Service gemeldet werden. Darüber hinaus ist eine beliebige Menge von zusätzlichen constraints für diese Policy möglich.	Erweiterung von Policyklasse 12		Diese Pattern beinhaltet eigentlich keine Datennutzungsbedingung sondern besteht aus einer Verpflichtung (obligation). In Nukleus entspricht dies eher einem Event-Listener als einer Policy. Das nachfolgende Beispiel nutzt einen Seiteneffekt von Java um ein identisches Verhalten nachzubilden (die rechte Seite der UND Verknüpfung wird nur ausgewertet, falls die linke Seite schon WAHR ist). Wird eine der erlaubten Ergänzungen der Policy um beliebige constraints, preDutys und postDutys verwendet, ist es nur auf Basis des Einzelfalls und manuell möglich diese Policy in Nukleus zu übersetzen.
18	Agreement Policy	Erlaubt die Nutzung von Daten für bestimmte Aktion(en).	Entspricht Policyklasse 1		Für die von D° und Nukleus unterstützten Aktionen ist diese Policy abbildbar.

¹ Die Referenzen von Policyklassen in dieser Spalte beziehen sich auf die Ergebnisse für das Informationsmodell v3.1.0

Die Verwendbarkeit der einzelnen Policyklassen wurde in der vorherigen Tabelle durch die Verwendung unterschiedlicher Symbole aufgezeigt. Dabei wurde das Symbol verwendet für Policies, die vollständig unterstützt werden. Dies umfasst 3 der Policyklassen. Diverse Policyklassen können nicht vollständig unterstützt werden. Dies liegt häufig daran, dass die Policyklassen die Möglichkeit für beliebige Erweiterungen bieten. Diese Policies werden nicht in allen Ausprägungen durch D° bzw Nukleus unterstützt und sind mit dem Symbol markiert. Dies betrifft 7 Policyklassen. Ebenfalls befinden sich unter den Policyklassen solche, die in Nukleus bzw. D° umgesetzt werden könnten, aber keine Umsetzung erfolgt, da eine Umsetzung an anderen Stellen sinnvoller ist. Dies betrifft vorallem solche Policies, die auf der Ebene des Connectors oder des Workflows umgesetzt werden sollten und nicht in den individuellen Applikationen, welche die einzelnen Arbeitsschritte eines Workflows bilden. Hiervon sind 3 Policyklassen betroffen, welche mit dem Symbol gekennzeichnet sind. Alle Policies die keinerlei Unterstützung in D° bzw. Nukleus haben sind mit markiert. Dies betrifft 5 Policyklassen.

Transformation von Policies

Da die unterschiedlichen Lösungen für Usage Control intern unterschiedliche Policysprachen verwenden, ist es notwendig eine Basis für die IDS zu schaffen, welche für den Policyaustausch verwendet wird. Dabei wird im Rahmen der IDS eine an Open Digital Rights Language (ODRL) angelehnte Policysprache verwendet. Dabei ist es notwendig diese IDS Policies in einem mehrstufigem Prozess in ein Format zu überführen, welches durch D° und Nukleus verarbeitet werden kann.

Dieser Prozess wird in den nachfolgenden Abschnitten beschrieben.

Verarbeitung von IDS-Policies

Zunächst ist es notwendig, dass die verwendete Policyklasse in der gegebenen IDS-Policy erkannt wird.

Dieser Schritt wurde durch die Veröffentlichung des Informationsmodells in Version 4.0.0 erheblich vereinfacht. Jede IDS-Policy verfügt in ihrer Representation in der IDS-Policysprache über ein Attribut @type, welches eindeutig die verwendete Policyklasse benennt. Dies war in Version 3.1.0 des Informationsmodells noch nicht gegeben, weswegen es notwendig gewesen wäre an dieser Stelle erheblich mehr Aufwand zu betreiben um eine zuverlässige Erkennung der Policyklassen zu erreichen.

Sobald bekannt ist, welche Policyklasse verwendet werden soll, ist es notwendig die Transformation an das entsprechende Modul für D° bzw. Nukleus zu delegieren.

Abbildung auf Nukleus & Degree Policies

Da im Informationsmodell Version 4.0.0 Templates für alle verwendeten Policyklassen vorliegen, ist der einfachste Weg zur Überführung in D°- und Nukleus-Policies die Erstellung eigener Templates für die unterstützten Policyklassen zu erzeugen. Im Rahmen einer M2M-Transformation werden unter Verwendung dieser Templates entsprechende Policies für D° und Nukleus erzeugt.

Die erzeugten Policies werden in Dateien abgelegt, welche später im Übersetzungsvorgang bzw. in der Applikation verwendet werden, um die erzeugten Policies in der Applikation einzubetten. Für Nukleus wird eine Datei mit Policydefinitionen erzeugt, die anschließend direkt vom Typssystem der Data App verwendet werden kann, um die nötigen Policies zu laden. Die Datei, die für D° erzeugt wird, enthält Instanzen für die Policy-Definitionen, welche verwendet werden um die Policyklassen umzusetzen. Diese wird während des Übersetzungsvorgangs eingelesen und dazu verwendet die Policies in die übersetzte Applikation einzubetten. Ein weiterer Schritt der zwischen der Erzeugung der Policydatei und dem Übersetzungsvorgangs stattfindet, ist eine Modifikation des Programmcodes, welche die Eingabeparameter der Applikation mit den entsprechenden Policies verknüpft. Effektiv wird während der Transformation von IDS-Policies in D°-Policies dynamisch eine Spracherweiterung für D° erzeugt.

Spracherweiterungen für D° sind ausführlich in den entsprechenden D°-Dokumentationen beschrieben und können dort eingesehen werden. Da direkt für D° und Nukleus passende Policies erzeugt werden, ist hier kein weiterer Austausch von Policies zwischen D° und Nukleus notwendig. Stattdessen wird durch das Verfahren sichergestellt, dass die korrekten Policies an den notwendigen Stellen ausgewertet werden.

Usage Control Objekt

Analog zu ODRL, welches für den Austausch von Policies zwischen den verschiedenen Teilnehmern und deren potentiell unterschiedlichen Lösungen für Usage Control verwendet wird, ist es notwendig ein IDS weites Verfahren für den Austausch von Policyrelevanten Daten zu haben. Beispielsweise erfordert Policyklasse 7 dass der aktuelle Zählerstand zwischen den unterschiedlichen Lösungen für Usage Control ausgetauscht werden kann, um die Policy erfolgreich umzusetzen. Um einen bilateralen Austausch und entsprechende Schnittstellen zwischen den einzelnen Lösungen unnötig zu machen, wird in den IDS das sogenannte "Usage Control Objekt" verwendet. Das Usage Control Objekt ist die vertrauenswürdige Quelle für Informationen und Daten rund um die Durchsetzung von Policies in den IDS.

Das Usage Control Objekt der IDS befindet sich aktuell noch in der Spezifikationsphase und ist nicht final fertiggestellt.

Es ist notwendig dieses IDS weite Usage Control Objekt in ein Format zu überführen, welches von D° und Nukleus verwendet werden kann.

Details zur Umsetzung in D°

D° verfügt bereits über ein sogenanntes Subsystem für Kontextinformationen. Dieses entstand vor den Arbeiten am Usage Control Objekt und enthält daher andere Informationen und ein eigenes Zugriffsschema. Das Subsystem erlaubt die Baumartige Organisation von Kontextinformationen, welche alle über eindeutige Namen erreichbar sind. Die im Subsystem enthaltenen Werte werden persistiert und stehen somit auch über mehrere Ausführungen einer Data App hinweg zur Verfügung. Jede Änderung an den Werten wird direkt persistiert und darüber hinaus in einem persistenten Protokoll erfasst.

Das Subsystem stellt die folgenden Arten von Daten zur Verfügung:

- ReadOnly
- ReadWrite
- Switch
- Counter
- DecrementCounter
- IncrementCounter

Die einzelnen Werte sind dabei in sog. `ContextModule` organisiert. Dabei soll ein Modul logisch/semantisch zusammengehörige Werte zusammenfassen und gemeinsam verfügbar machen. Jedes Modul verfügt über einen Namen und enthält eine beliebige Menge weiterer Module und Items für Kontextinformationen.

Um einen Wert, der im Subsystem abgelegt ist, wird eine Anfrage an das oberste `ContextModule` übergeben und anschließend aufgelöst. Dabei haben Anfragen die Form 'MODULE_NAME(.MODULE_NAME)+.ITEM_NAME'.

Das folgende Codebeispiel zeigt eine Beispielanfrage an das ContextInformation-Subsystem in der Programmiersprache Java.

Nutzung des ContextInformation-Subsystems

```
ExecutionContext.getInstance().resolve("UserInformation.username")
```

Die einzelnen Elemente in diesem Subsystem können für die Auswertung von Policies verwendet werden, sind aber auch in anderen Teilen von Data Apps (bspw. Aktivitäten) verfügbar.

Darüber hinaus bietet D° ein Konstrukt mit dem Namen "Usage Control Object". Das Usage Control Object ist eine zentrale Anlaufstelle, die verwendet werden kann, um Daten abzufragen welche für die Auswertung von Policies verwendet werden kann. Das besondere am Usage Control Object ist, dass die Implementierung, abhängig vom verwendeten Data App Typen, ausgetauscht werden kann. Die Standardimplementierung (welche aktuell die einzige ist) bezieht einen Großteil der verwendeten Informationen aus dem Subsystem für Kontextinformationen.

Sobald die Arbeiten am gleichnamigen Konzept in den IDS abgeschlossen sind und eine Spezifikation vorliegt, kann eine entsprechende IDS-Implementierung des D°-UsageControlObjects entwickelt werden und ohne weiteren Aufwand in entsprechenden D°-Applikationen verwendet werden.

Schnittstelle D° Nukleus

D° erzeugt die Data App zur Verfügung und verwendet intern Nukleus. Daher verfügt die D°-Applikation über Berührungspunkte mit dem Connector und damit dem Rest der IDS, wogegen Nukleus (in diesem Einsatzszenario) nur Berührungspunkte mit D° hat. Da auch Policies in Nukleus umgesetzt werden, ist es notwendig einen Datenkanal zu etablieren, welcher verwendet werden kann, um Daten aus dem Usage Control Objekt in Nukleus verfügbar zu machen und umgekehrt muss Nukleus auch Updates mit geänderten Daten an das Usage Control Objekt schicken können.