

# Going Live with Enterprise Solutions



## FIBO in SHACL: The Next Step in Data Validation and Interoperability

Irene Polikoff, TopQuadrant  
*EDW, San Diego, April 2018*



- CEO and co-founder at TopQuadrant
- W3C SHACL Working Group co-chair
  
- SHACL Working Group
  - Was chartered in October 2015
  - Successfully completed its charter in July 2017
  - Currently, in the “maintenance mode”



Irene Polikoff

- **SH**Apes **C**onstraint **L**anguage
  - W3C Recommendation to define the “shape” of data
  - Shapes are defined by constraints and validation rules
  - Shapes provide facilities for input checking and reporting
  - By design, a schema language for RDF & LinkedData (the W3C graph based, or network-of-nodes, data model)
  
- SHACL Shapes:
  - Shapes for node types reside in a “Shape Library”
  - Shape “inheritance” down the node tree
  - Shapes can be extended, reused, and shared

# SHACL's Role in the Sematic Web Stack



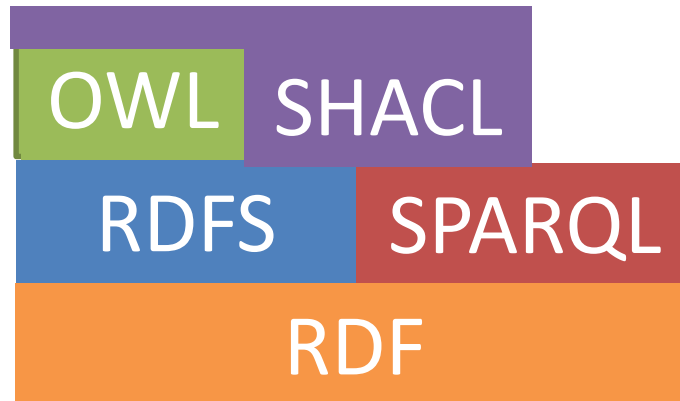
\*W3C = World Wide Web Consortium led by Tim Berners-Lee

## Classification:

*What is the type of this thing?*

## Rules:

*Is that term used correctly?  
You can't say that here!  
What can you infer?*



## Vocabulary:

*Shared terms can we use*

## Query:

*What did you say?*

## Statements:

*Saying things*

# Why talk about SHACL at EDW?

(Knowledge) graphs are powerful for enterprise data integration and reuse

SHACL ensures data quality and enhances the data

Strong adoption of SHACL is happening  
– including in Financial Services

# Motivation for SHACL at W3C

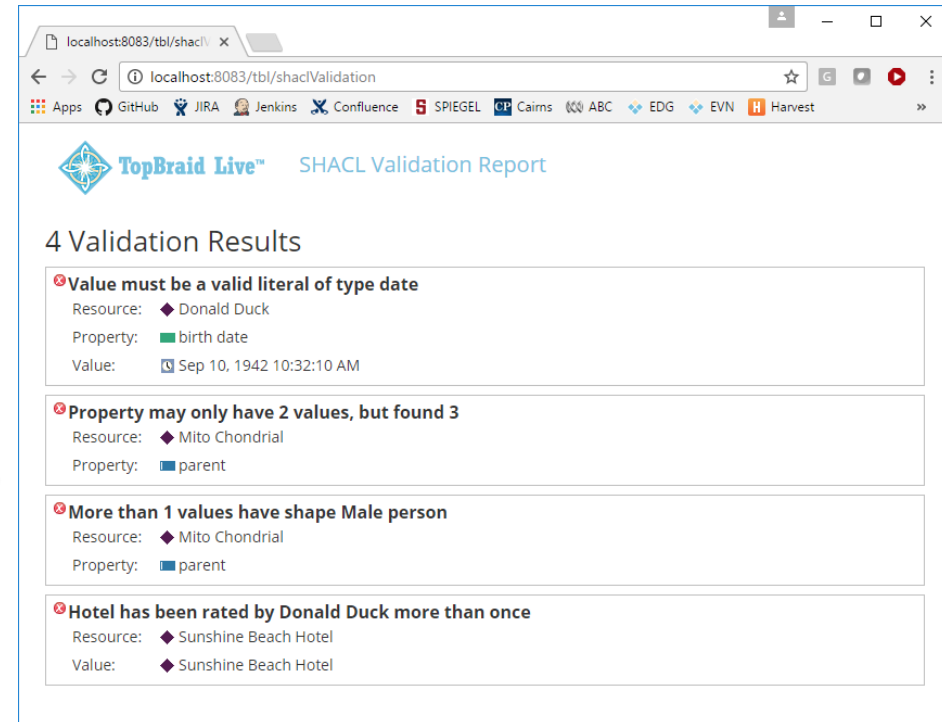
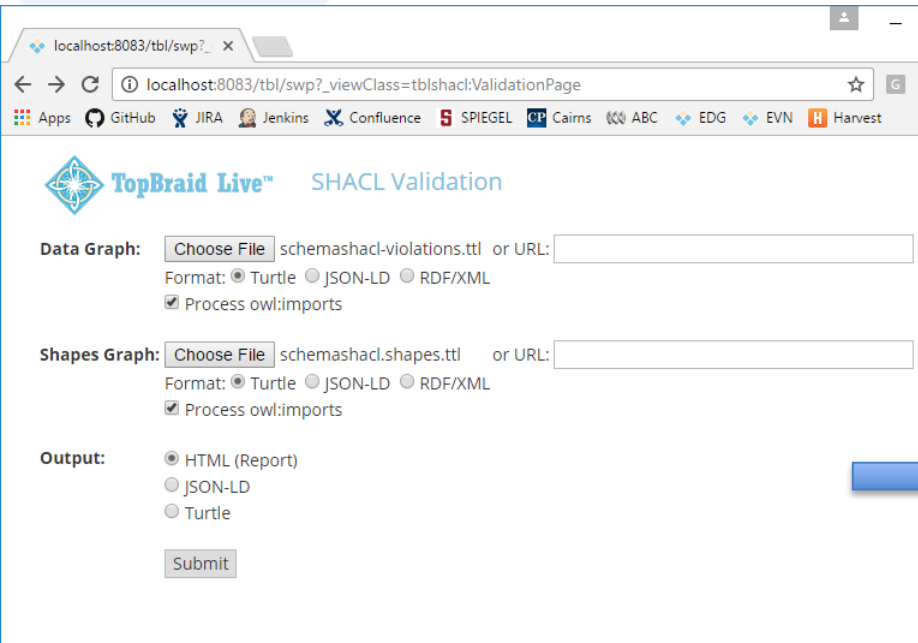
- SHACL is rooted in the world of the Semantic Web W3C standards
  - Until SHACL no official W3C recommendation for validating RDF
- RDF Schema / OWL are “schema languages”, however...
  - Descriptive, not prescriptive or restrictive
  - OWA, CWA, UNA ⇒ classification inferencing not validation
  - Even for inferencing/classification, OWA is often a problem - See using SHACL for classification rules:  
<https://henrietteharmse.com/2018/03/15/classification-with-shacl-rules/>
- W3C member submissions (semi-official)
  - SPIN, Resource Shapes, ShEx
  - SHACL unifies the best features of each to cover most use cases
- Other Standalone / ad-hoc solutions have also evolved
  - Most on top of SPARQL & RDFS/OWL
- SHACL gives us a vendor neutral solution and a governing body
  - A standard on par with XML, HTML, RDF, etc.

OWA: [Open World Assumption](#)










CWA: [Closed World Assumption](#)

UNA: [Unique Name Assumption](#)

# SHACL Ensures conformance of RDF data to a defined schema



4 Validation Results

- Value must be a valid literal of type date**  
Resource:  Donald Duck  
Property:  birth date  
Value:  Sep 10, 1942 10:32:10 AM
- Property may only have 2 values, but found 3**  
Resource:  Mito Chondrial  
Property:  parent
- More than 1 values have shape Male person**  
Resource:  Mito Chondrial  
Property:  parent
- Hotel has been rated by Donald Duck more than once**  
Resource:  Sunshine Beach Hotel  
Value:  Sunshine Beach Hotel

- Not conforming data is reported in the “Validation Report” – report structure is defined in the SHACL standard

# SHACL and Rules

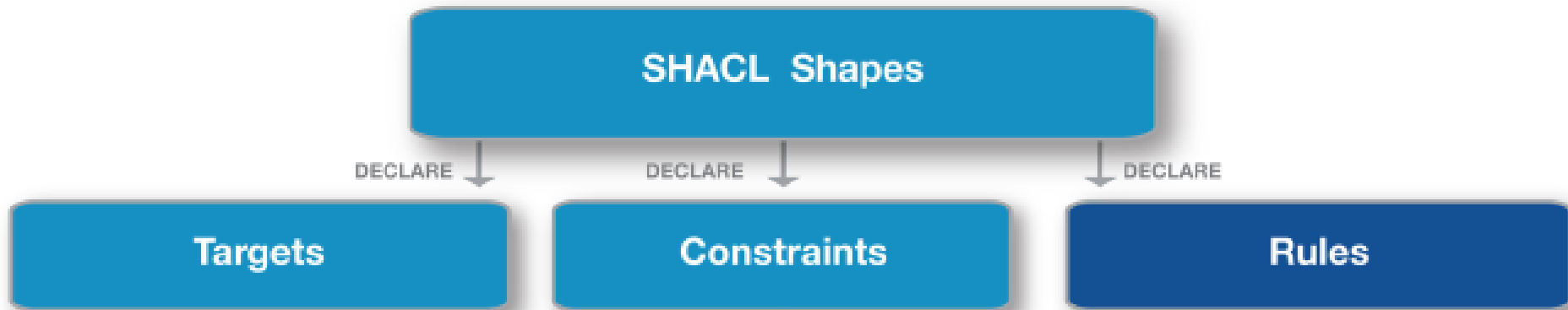
- SHACL let's you define a “shape” data should adhere to
  - This is effectively a “data quality rule” – if condition is not met, SHACL engine reports an exception
- SHACL also let's you ‘infer’ new data if condition is met
  - This is a more general data or business rule – if condition is met, SHACL engine can add new fact(s)



# What else SHACL can be used for


- Extended, but not limited to, goals:
  - interface building
  - data structure communication
  - code generation
  - data integration
  - rule-based inferencing
  - ...

# High Level View of SHACL



# FIBO is maintained in OWL published to various formats

- FIBO SKOS
- Spreadsheet data dictionary
- ...
  
- FIBO can be published in SHACL
- Caveat: SHACL is more expressive than OWL



≡ FIBO SHACL

Hello, **Administrator**

Assets

Dashboard Settings User Roles Import Export Reports Workflows Tasks Comments Manage

### Class Hierarchy

- agreement
  - contract
    - control related contract
    - credit agreement
    - credit support agreement
    - mutual contractual agreement
    - unilateral contract
    - verbal contract
    - written contract
  - defines terms for
  - has contract party
  - has contractual element
  - has counterparty
  - has effective date
  - has governing jurisdiction
  - has guarantor
  - has non-binding terms
  - has principal
  - has terms
  - has third party
  - is assignable

### contract (Class, Node shape)

🔍
⚠️ (0)
📄
Edit
Print
Start Workflow
(0) Tasks
(0) Comments

Show History

<https://spec.edmcouncil.org/fibo/FND/Agreements/Contracts/Contract>

#### Labels and Description

**label:** contract

**definition:** a voluntary, deliberate, and legally binding agreement between two or more competent parties

**definition origin:** <http://www.businessdictionary.com/definition/contract.html>

**explanatory note:** Contracts are usually written but may be spoken or implied, and generally have to do with employment, sale or lease, or tenancy.

---

#### Class Characteristics

**sub-class of:**

- `has contract party` **min 2** `[contract party](#)`
- `has contractual element` **some** `[contractual element](#)`
- `has effective date` **exactly 1** `[date](#)`
- `is assignable` **exactly 1** `[yes or no](#)`

[agreement](#) →

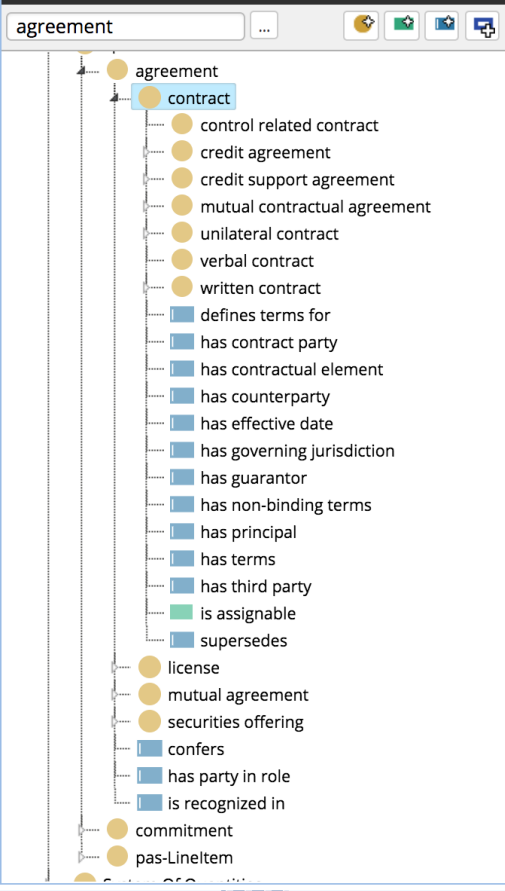
**equivalent class:**

- [contract](#) →
- [contract](#) →
- [contract](#) →
- [contract](#) →
- [contract](#) →
- [contract](#) →

### Instances of contract

|  |  |  |  |  |
|--|--|--|--|--|
|  |  |  |  |  |
|--|--|--|--|--|

## Class Hierarchy



## contract (Class, Node shape)

contract (Class, Node shape)

Assets (0) Edit Print Start Workflow (0) Tasks (0) Comments Show History

Included from **fibonacci-all.shapes**

**on property:** has contract party  
**class:** contract party  
**min count:** 2

Included from **fibonacci-all.shapes**

**on property:** has contractual element  
**has value with class:** contractual element  
**class:** contractual element

Included from **fibonacci-all.shapes**

**on property:** has counterparty  
**class:** contract counterparty

Included from **fibonacci-all.shapes**

**on property:** has effective date  
**class:** date  
**max count:** 1  
**min count:** 1

Included from **fibonacci-all.shapes**

**on property:** has governing jurisdiction  
**class:** jurisdiction

Included from **fibonacci-all.shapes**

**on property:** has guarantor  
**class:** guarantor

## Instances of contract

localhost:8083/tbl/fibo\_shacl.editor#

Fixes up unnecessary qualified cardinality restrictions: e.g. range of the property 'has contract party' is 'contract party' – qualified cardinality is not necessary and misleading

TopBraid EDG Enterprise Data Governance FIBO SHACL Hello, Administrator

Assets Dashboard Settings User Roles Import Export Reports Workflows Tasks Comments Manage

contract (Class, Node shape)

Details

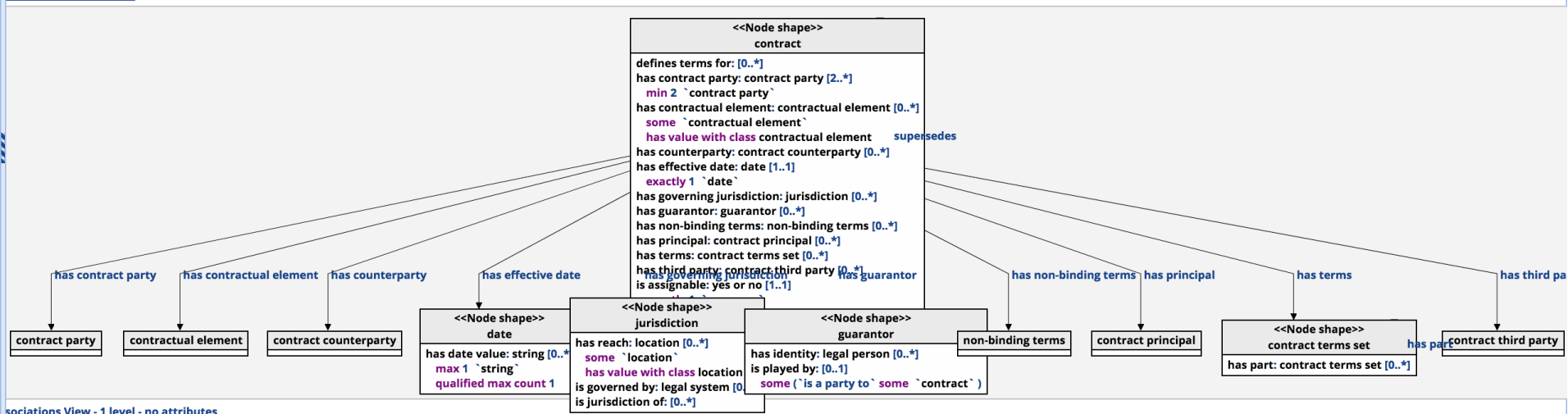
properties Table

frames

inheritance View

inheritance View - no attributes

associations View - 1 level



```

    graph TD
      contract["<<Node shape>>  
contract  
defines terms for: [0..*]  
has contract party: contract party [2..*]  
  min 2 `contract party`  
has contractual element: contractual element [0..*]  
  some `contractual element`  
has value with class contractual element  
has counterparty: contract counterparty [0..*]  
has effective date: date [1..1]  
  exactly 1 `date`  
has governing jurisdiction: jurisdiction [0..*]  
has guarantor: guarantor [0..*]  
has non-binding terms: non-binding terms [0..*]  
has principal: contract principal [0..*]  
has terms: contract terms set [0..*]  
has third party: contract third party [0..*]  
has governing jurisdiction  
is assignable: yes or no [1..1]"]
      contract -- "has contract party" --> contract_party["contract party"]
      contract -- "has contractual element" --> contractual_element["contractual element"]
      contract -- "has counterparty" --> contract_counterparty["contract counterparty"]
      contract -- "has effective date" --> date["<<Node shape>>  
date  
has date value: string [0..*]  
  max 1 `string`  
  qualified max count 1"]
      contract -- "has governing jurisdiction" --> jurisdiction["<<Node shape>>  
jurisdiction  
has reach: location [0..*]  
  some `location`  
has value with class location  
is governed by: legal system [0..*]  
is jurisdiction of: [0..*]"]
      contract -- "has guarantor" --> guarantor["<<Node shape>>  
guarantor  
has identity: legal person [0..*]  
is played by: [0..1]  
  some (`is a party to` some `contract`)"]
      contract -- "has non-binding terms" --> non_binding_terms["non-binding terms"]
      contract -- "has principal" --> contract_principal["contract principal"]
      contract -- "has terms" --> contract_terms_set["<<Node shape>>  
contract terms set  
has part: contract terms set [0..*]"]
      contract -- "has third party" --> contract_third_party["contract third party"]
      contract -- "supersedes" --> contract_terms_set
  
```

associations View - 1 level - no attributes

associations View - 2 levels

associations View - 2 levels - no attributes

# Automated Form Generation for a Contract

Enter log message

## Annotations

label:

## Properties

- confers:
- defines terms for:
- has contract party: \*
- has contractual element:
- has counterparty:
- has effective date: \*
- has governing jurisdiction:
- has guarantor:
- has non-binding terms:
- has party in role: \*
- has principal:
- has terms:
- has third party:
- is assignable: \*
- is recognized in:
- supersedes:
- type:

Hint: SHACL also lets you organize properties into sections (Property Groups) and specify order of the fields (properties) with a section

Use sh:group and sh:order

Sections can be ordered as well

# Example of ordered Sections using one of pre-built classes (Database) from EDG ontologies

TopBraid EDG Enterprise Data Governance | Data Assets Model | Hello, Administrator

Assets | Dashboard | Settings | User Roles | Import | Export | Reports | Workflows | Tasks | Comments | Manage

Class Hierarchy | Database (Asset Class) | Show History

Look up Class | [Icons] | [Alerts] | Edit | Print | Start Workflow | (0) Tasks | (0) Comments

Class Hierarchy:

- Data Asset
  - Data Element
  - Data Package
  - Data Structure
  - Database**
  - Database Extract
  - Database Synonym
  - Dataset
  - File Asset
  - Logical Data Asset
  - Physical Data Asset
  - Record
  - Schema
  - derived from

Property Layout Groups

**Class: Database**

| Order     | Property Group  |
|-----------|---|
| <b>10</b> | <b>Identifiers Metadata</b>                                       |
| 1         | type (from Aspect/Feature)  |
| 10        | label (from Identifiers, Codes and other Designators)             |
| 20        | alternative label (from Identifiers, Codes and other Designators) |
| 30        | asset name (from Identifiers, Codes and other Designators)        |
| 40        | acronym (from Identifiers, Codes and other Designators)           |
| 50        | identifier (from Identifiers, Codes and other Designators)        |
| 50        | tag (from Identifiers, Codes and other Designators)               |
| <b>12</b> | <b>Status Metadata</b>  |
| 10        | status (from Status Aspect)                                       |
| 30        | effective start date (from Status Aspect)                         |
| 40        | effective end date (from Status Aspect)                           |
| 50        | superseded by (from Status Aspect)                                |
| 70        | version (from Status Aspect)                                      |
| 80        | version notes (from Status Aspect)                                |
| 90        | created (from Status Aspect)                                      |
| 100       | last modified (from Status Aspect)                                |
| <b>20</b> | <b>Documentation</b>  |
| 10        | short description (from Narratable)                               |
| 20        | description (from Narratable)                                     |
| 40        | purpose (from Narratable)   |
| 50        | notes (from Narratable)   |
| 80        | document link (from Narratable)                                   |
| 90        | reference link (from Narratable)                                  |
| 100       | guidance link (from Narratable)                                   |
| 120       | see also (from Narratable)  |
| <b>30</b> | <b>Data Container Metadata</b>                                    |

http://edg.topbraid.solutions/model/Database  
Last edited by Administrator on Apr 24, 2018 6:06:44 PM

ent storage and retrieval structure comprising tables and data elements. The parent type 'database' has sub-types for databases, and also data warehouses.

Models →

other Designators →

Schema - Data Models

ality




Schema - Data Models

g scale - five values

base Metadata



**contract1 (contract)**

          Show History

<http://example.org/fibo/contract1>  
Last edited by Administrator on Apr 25, 2018 7:54:12 AM

### Other Properties

**has contract party:** [cparty1](#) →

**label:** contract 1

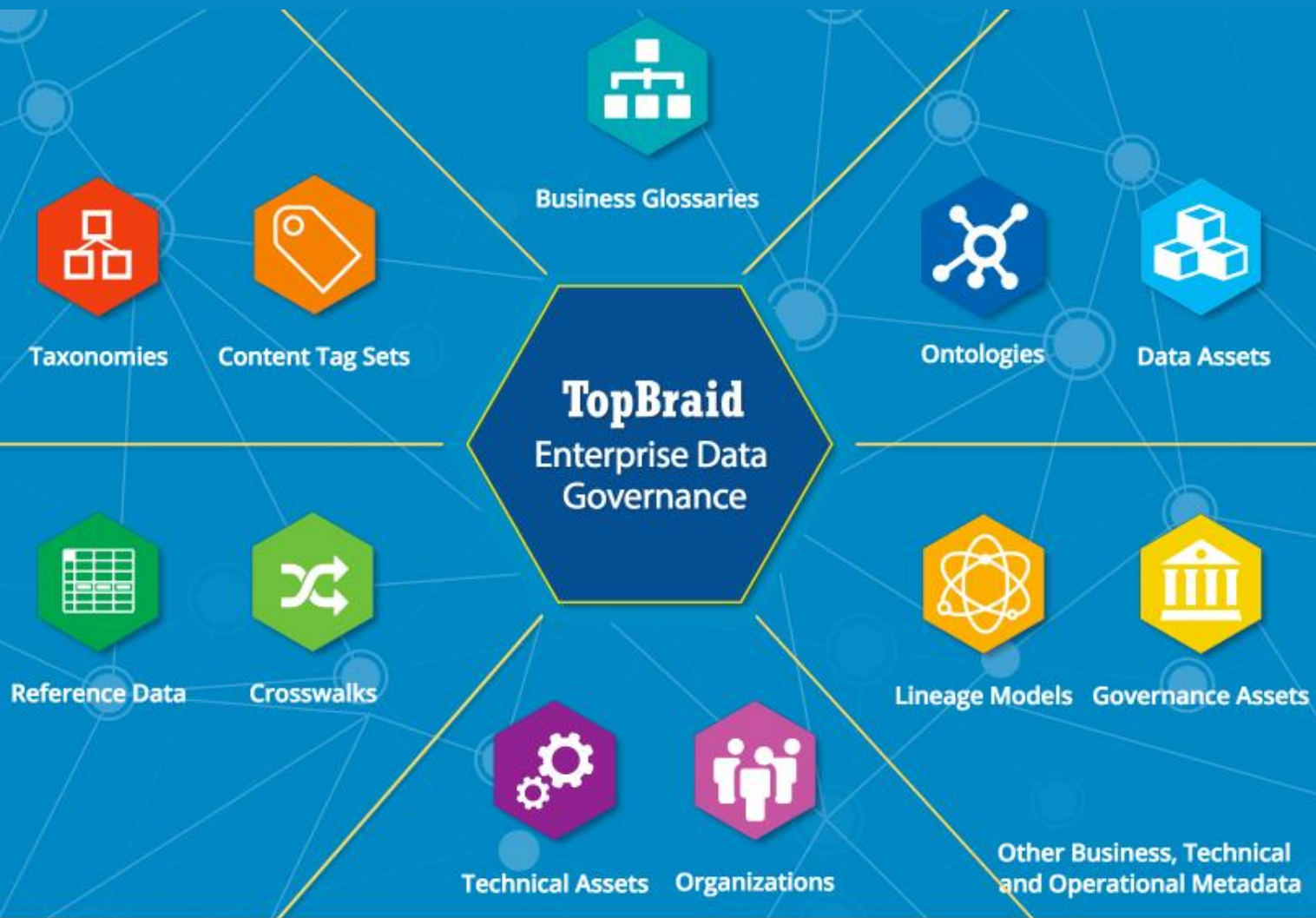
**type:** [contract](#) →

- ✘ At least one of the values must have class commitment [fibo-fnd-agr-agr:Commitment]**  
Property:  confers
- ✘ Property needs to have at least 2 values, but found 0**  
Property:  has party in role
- ✘ At least one of the values must have class contractual element**  
Property:  has contractual element
- ✘ Property needs to have at least 2 values, but found 1**  
Property:  has contract party
- ✘ Property needs to have at least 1 values, but found 0**  
Property:  has effective date
- ✘ Property needs to have at least 1 values, but found 0**  
Property:  is assignable

Other features and benefits e.g., automated APIs/GraphQL queries with SHACL

# SHACL and Enterprise Data Management

# SHACL for Data Governance





# TopQuadrant™ Asset Definition

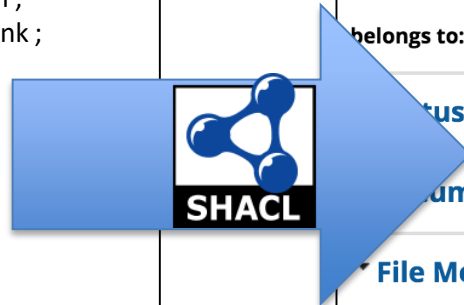
Models drive the User Experience (UX) i.e., what fields will appear on an (edit, search or browse) form, order of the fields, their division into sections and other relevant information.

```
edg:FileAsset
  rdf:type edg:AssetClass ;
  dash:abstract "false"^^xsd:boolean ;
  rdfs:label "File Asset" ;
  rdfs:subClassOf edg:AccessControllable ;
  rdfs:subClassOf edg:ComplianceAspect ;
  rdfs:subClassOf edg:Identifiable ;
  ;
  ;
  sh:property edg:File-formatType ;
  sh:property edg:File-name ;
  sh:property edg:File-residesIn ;
  sh:property edg:File-sourceLink ;
.
```

```
edg:File-formatType
  rdf:type sh:PropertyShape ;
  sh:path edg:format ;
  sh:class edg:FormatType ;
.
```

```
edg:File-name
  rdf:type sh:PropertyShape ;
  sh:path edg:name ;
  sh:datatype xsd:string ;
.
```

```
edg:File-residesIn
  rdf:type sh:PropertyShape ;
  sh:path edg:residesIn ;
  sh:class edg:FileSystem ;
.
```



**Identifiers Metadata**

type:  ▾ ✕

label:  Lang ▾ ✕

alternative label:  Lang ▾ ✕

name:  ✕

acronym:  Lang ▾ ✕

identifier:  Lang ▾ ✕

tag:  Lang ▾ ✕

belongs to:  ▾ ✕

---

**File Metadata**

name:  ✕

format:  ▾ ✕

source link:  ▾ ✕

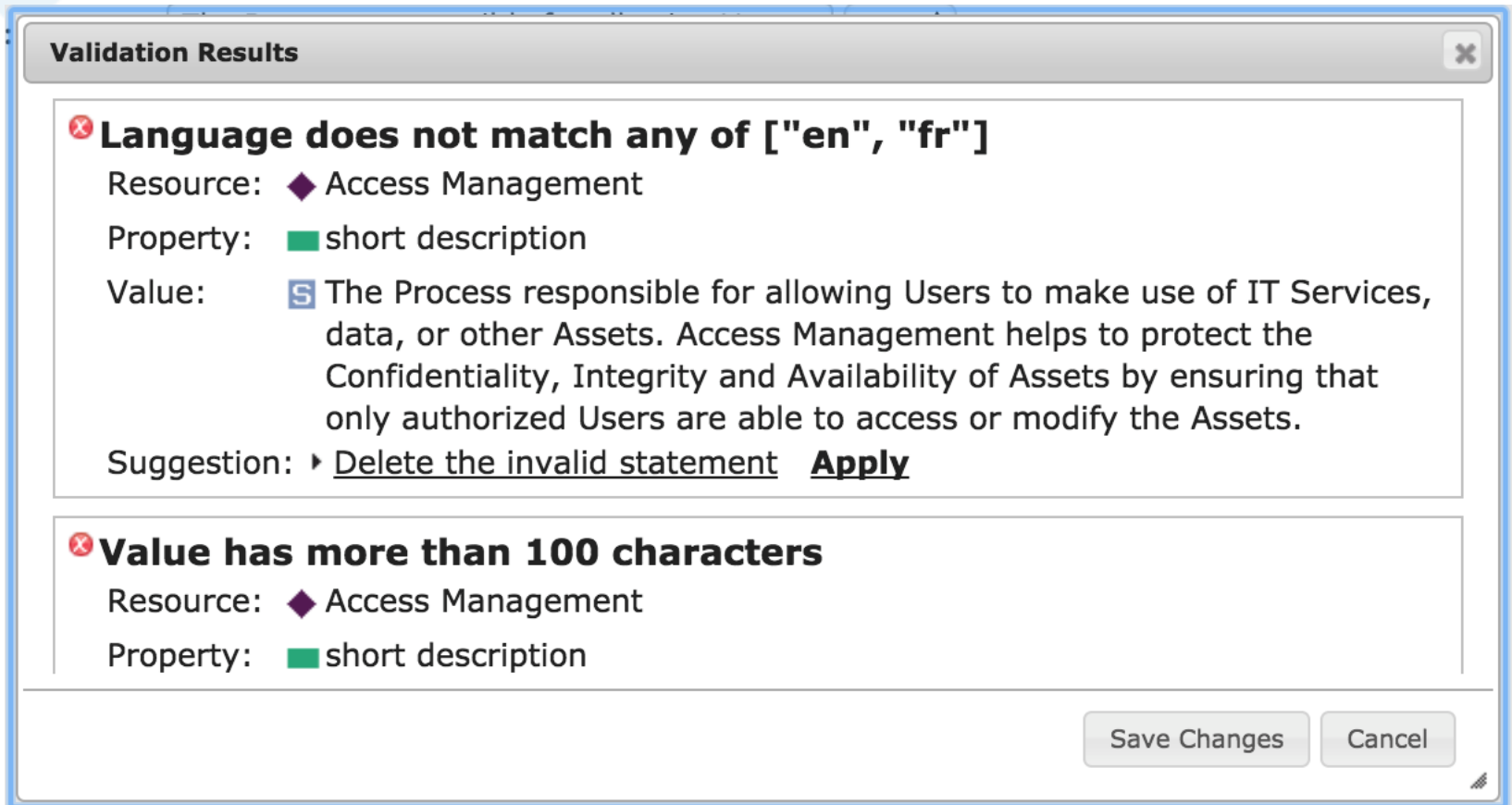
---

**File System Metadata**






resides in:  ▾ ✕

# Ability to Define Data Quality Rules

SHACL can be used to enforce organization's best practices. E.g.:  
*A business term must have a short description in both (and only) English and French with length under 100 characters.*

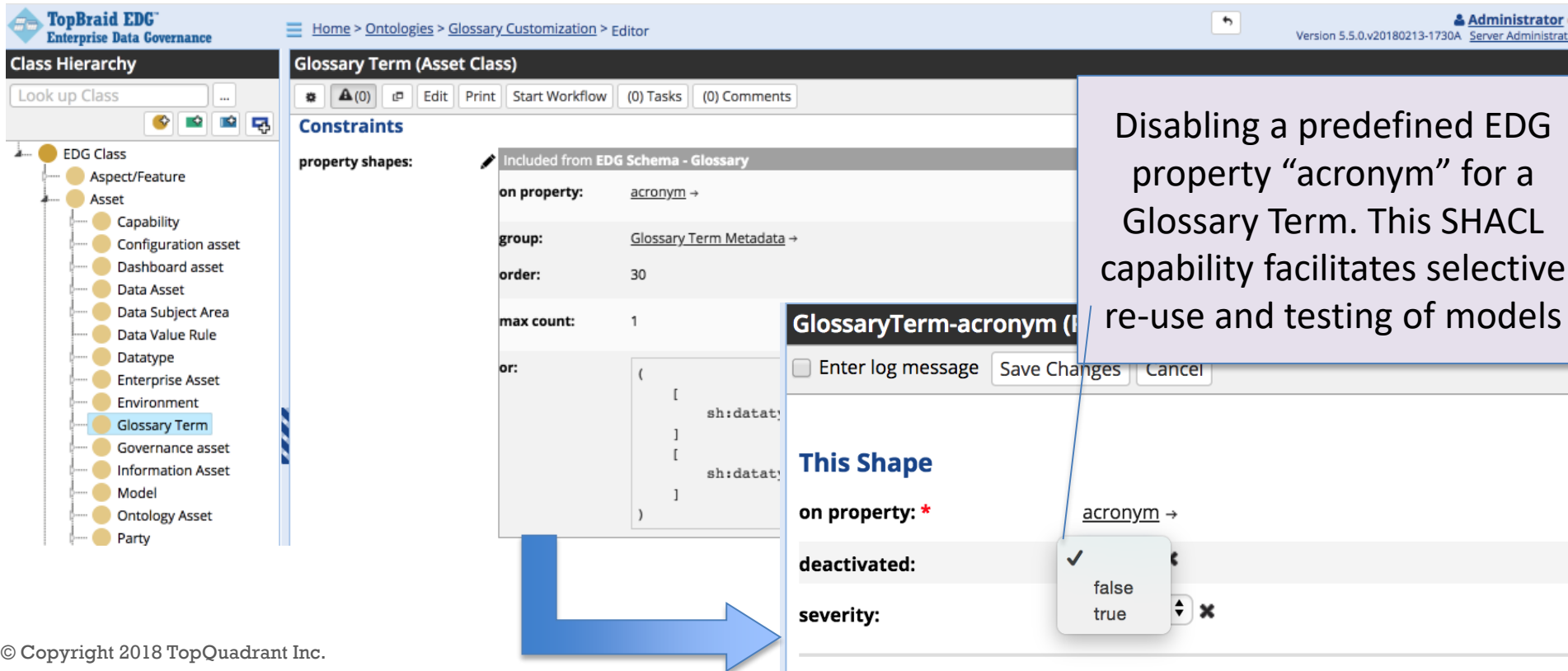


**Validation Results**

- ✘ Language does not match any of ["en", "fr"]**  
Resource:  Access Management  
Property:  short description  
Value:  The Process responsible for allowing Users to make use of IT Services, data, or other Assets. Access Management helps to protect the Confidentiality, Integrity and Availability of Assets by ensuring that only authorized Users are able to access or modify the Assets.  
Suggestion: ▶ Delete the invalid statement **Apply**
- ✘ Value has more than 100 characters**  
Resource:  Access Management  
Property:  short description

Save Changes Cancel

Organizations often need to configure the pre-defined models. SHACL not only offers flexibility to define new attributes and relationships for an asset type, but also lets users 'disable' predefined attributes and relationships if they decide they don't need to capture certain information.



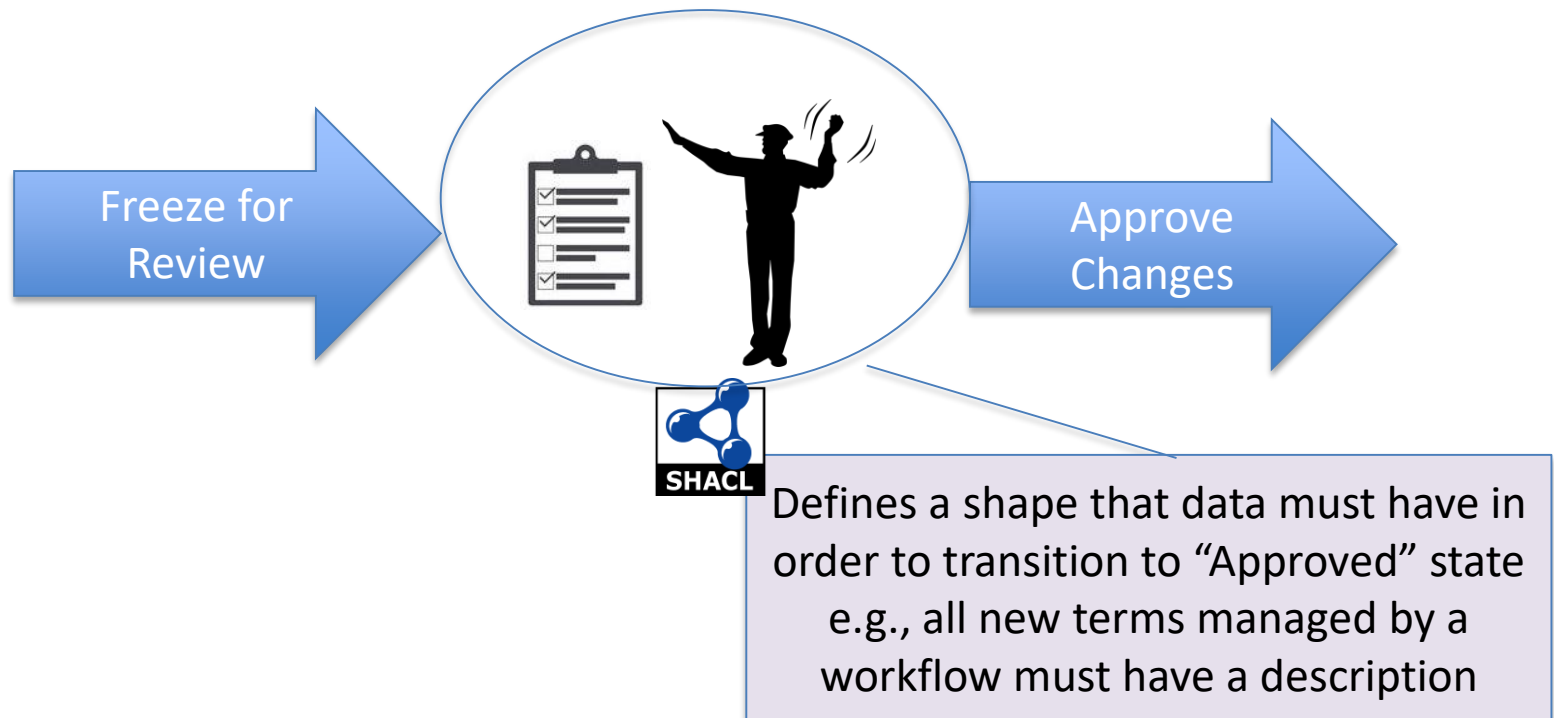
The screenshot shows the 'Glossary Term (Asset Class)' configuration page in TopBraid EDG. On the left is a 'Class Hierarchy' tree with 'Glossary Term' selected. The main area shows 'Constraints' for 'Included from EDG Schema - Glossary'. A property shape is defined for the 'acronym' property, with a group of 'Glossary Term Metadata' and a maximum count of 1. A modal dialog titled 'GlossaryTerm-acronym (I' is open, showing the 'This Shape' configuration. The 'deactivated' checkbox is checked, and a dropdown menu is open showing 'false' and 'true' options. A blue arrow points from the SHACL code area to the modal dialog.

Disabling a predefined EDG property "acronym" for a Glossary Term. This SHACL capability facilitates selective re-use and testing of models

# TopQuadrant™ Ensure Governance Workflows

TopBraid EDG supports configurable workflows.

Each transition of a workflow from one state to another can include SHACL statements specifying conditions that data must satisfy in order to pass on to the next state in a workflow.

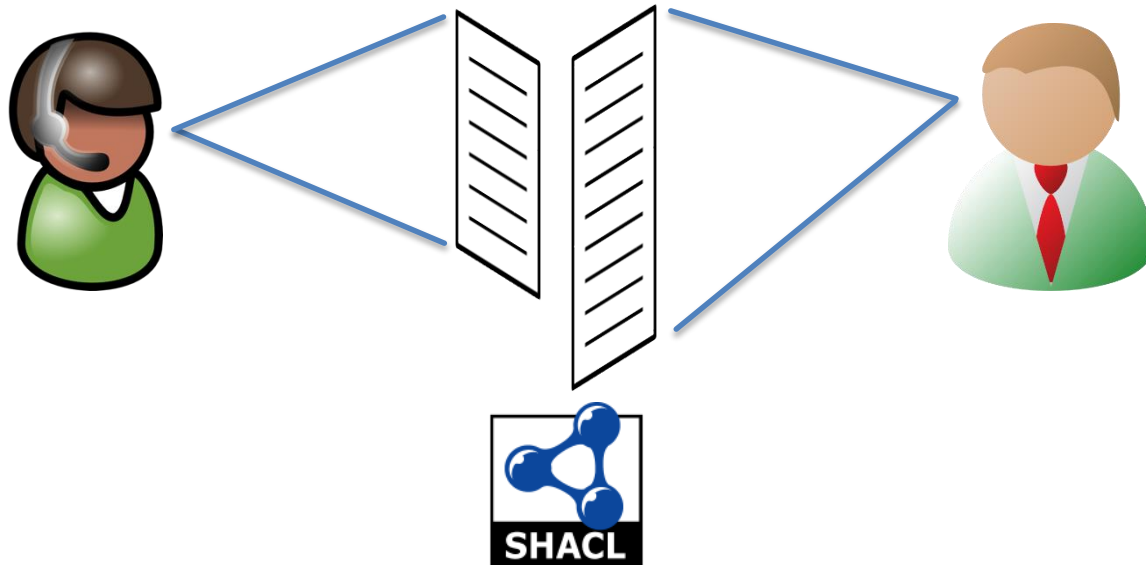




# Role Based Personalization

SHACL can be used to provide different views or perspectives for different users

SHACL let's us define multiple shapes for a given asset type. These shapes can be associated with different user roles. As a result, different users will get different views of the information. To support unique needs of each role, some fields can be specified as view-only. Other fields can be calculated based on available values.

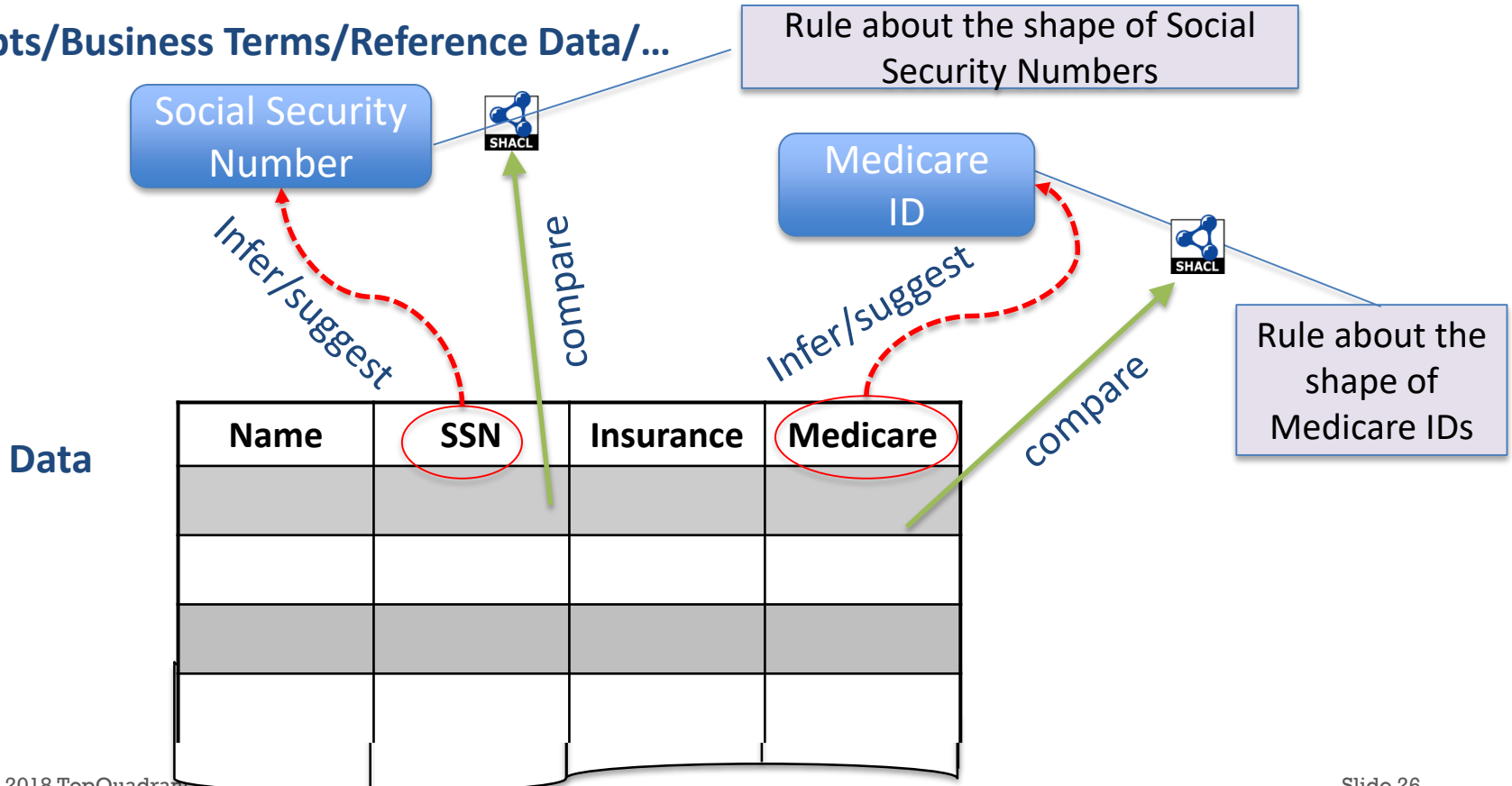


# Data Discovery and Mapping Automation

SHACL rules can be used to create new connections across collected information.

Rules defined in SHACL are directly executable and can be used to infer new information – for example, from existing technical metadata and results of data profiling. This SHACL capability is used extensively by EDG to cross-link, create new facts, add smartness and automation.

Concepts/Business Terms/Reference Data/...




## Problems and Suggestions (3)

[Download as TSV](#) [Download as JSON](#)

Minimum suggestion confidence:  0%

### Suggested mapping to term

Resource:  [CATEGORYNAME \(NORTHWIND.DBO.ALPHABETICAL LIST OF PRODUCTS\)](#)

Property:  [maps to term](#)

Suggestion: ▶ [Map column to Product Category, based on 5 out of 5 samples \(Confidence: 100%\)](#) **Apply**


### Suggested mapping to term


Resource:  [CATEGORYNAME \(NORTHWIND.DBO.PRODUCTS BY CATEGORY\)](#)

Property:  [maps to term](#)

Suggestion: ▶ [Map column to Product Category, based on 5 out of 5 samples \(Confidence: 100%\)](#) **Apply**

### Suggested mapping to term

Resource:  [CATEGORYNAME \(NORTHWIND.DBO.SALES BY CATEGORY\)](#)

Property:  [maps to term](#)

Suggestion: ▶ [Map column to Product Category, based on 5 out of 5 samples \(Confidence: 100%\)](#) **Apply**

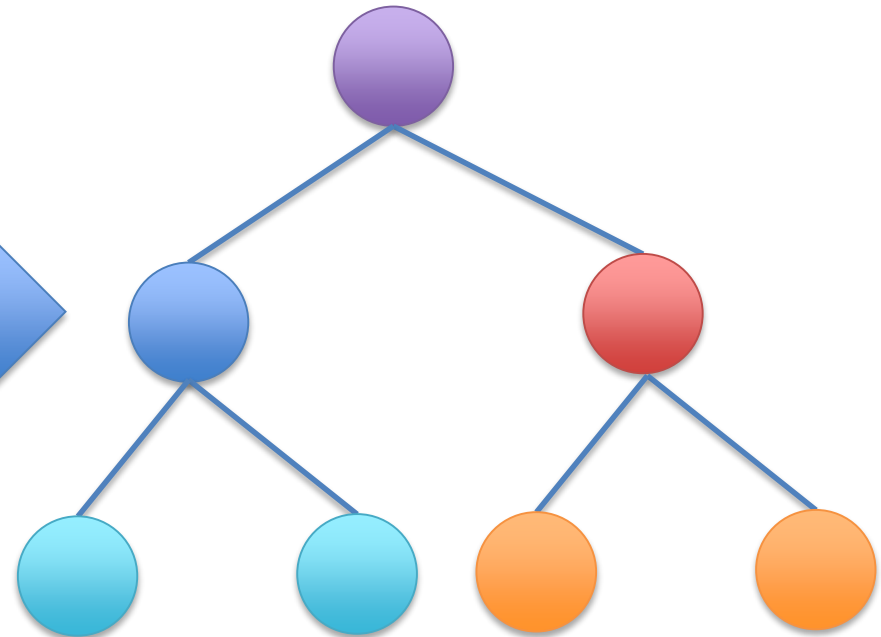
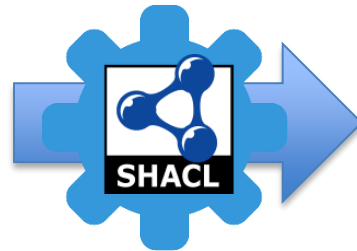
**[Apply all top suggestions](#)**

New facts auto-created by SHACL can have confidence attached to them

SHACL can be used to define data transformations.

When importing information from different sources (e.g., spreadsheets, XML, RDBMS), information needs to be mapped to the asset definitions in TopBraid EDG. SHACL let's us do this.

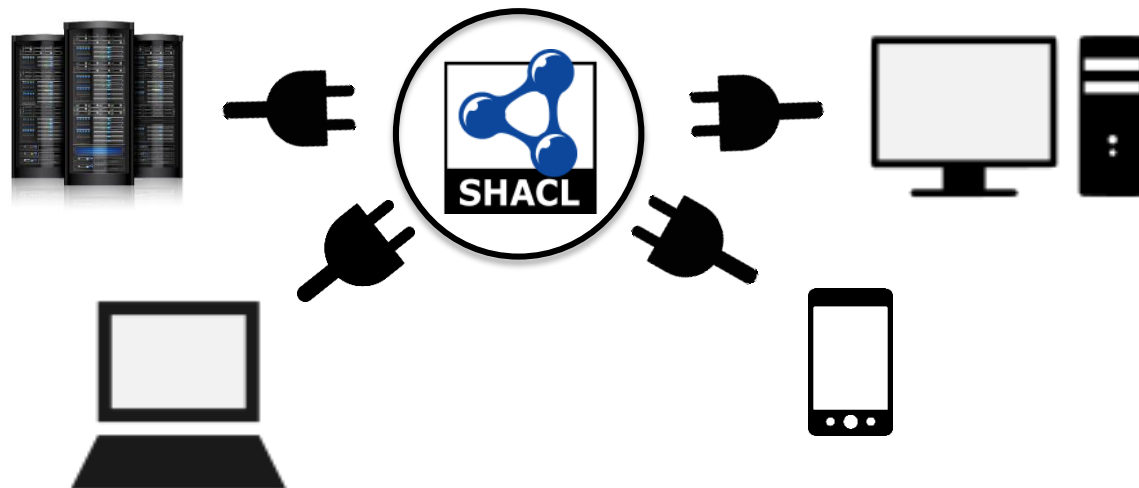
| N | N | N | N |
|---|---|---|---|
|   |   |   |   |
|   |   |   |   |
|   |   |   |   |

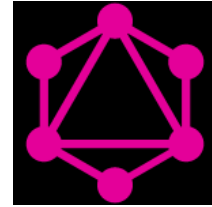


# Data Interface Configurability

SHACL can define data interfaces.

Just like SHACL can be used to specify different views of the information for different roles, it can also define different views or data structures to be delivered to systems that must interact with TopBraid EDG to access the information it stores. The definitions can include computations and transformations needed to satisfy the requirements of consuming systems.





- World's most popular graph “query language”
- Not really a query language, but rather a way to define data interfaces – JSON structures
- Very similar to SHACL – can serve as a compact syntax for SHACL

```
# A user account
type User {
  name: String!
  age: Int
  gender: Gender
  purchases: [Purchase]
}

type Purchase {
  # The internal ID of the product
  productId: String!
  date: String}

enum Gender {
  FEMALE
  MALE
}
```

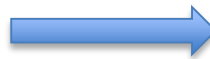
# GraphQL as a simple RDF Modelling Language

Converted to SHACL on the fly.  
GraphQL "directives" to express constraints, etc.

```
# A user account
type User {
  name: String!
  age: Int @shape(minInclusive: 18)
  gender: Gender
  purchases: [Purchase]
}

type Purchase {
  # The internal ID of the product
  productId: String! @shape(minLength: 8, pattern: "[0-9]+")
  date: String @shape(datatype: "xsd:date")
}

enum Gender {
  FEMALE
  MALE
}
```



⚙️ ⚠️(0) Edit Print Start Workflow (0) Tasks (0) Comments Show History

<http://example.org/ontologies/new#User>  
Last edited by Administrator on Mar 16, 2018 2:36:13 PM

### Labels and Description

**label:** User  
**comment:** A user account

### Constraints

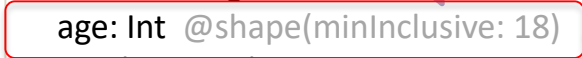
**property shapes:**

|                       |         |
|-----------------------|---------|
| <b>on property:</b>   | age     |
| <b>order:</b>         | 1       |
| <b>datatype:</b>      | integer |
| <b>min inclusive:</b> | 18      |

|                     |                      |
|---------------------|----------------------|
| <b>on property:</b> | gender               |
| <b>order:</b>       | 2                    |
| <b>in:</b>          | 1. MALE<br>2. FEMALE |

|                     |        |
|---------------------|--------|
| <b>on property:</b> | name   |
| <b>order:</b>       | 0      |
| <b>datatype:</b>    | string |

|                     |           |
|---------------------|-----------|
| <b>on property:</b> | purchases |
| <b>order:</b>       | 3         |
| <b>class:</b>       | Purchases |



# Opens Up Many Possibilities

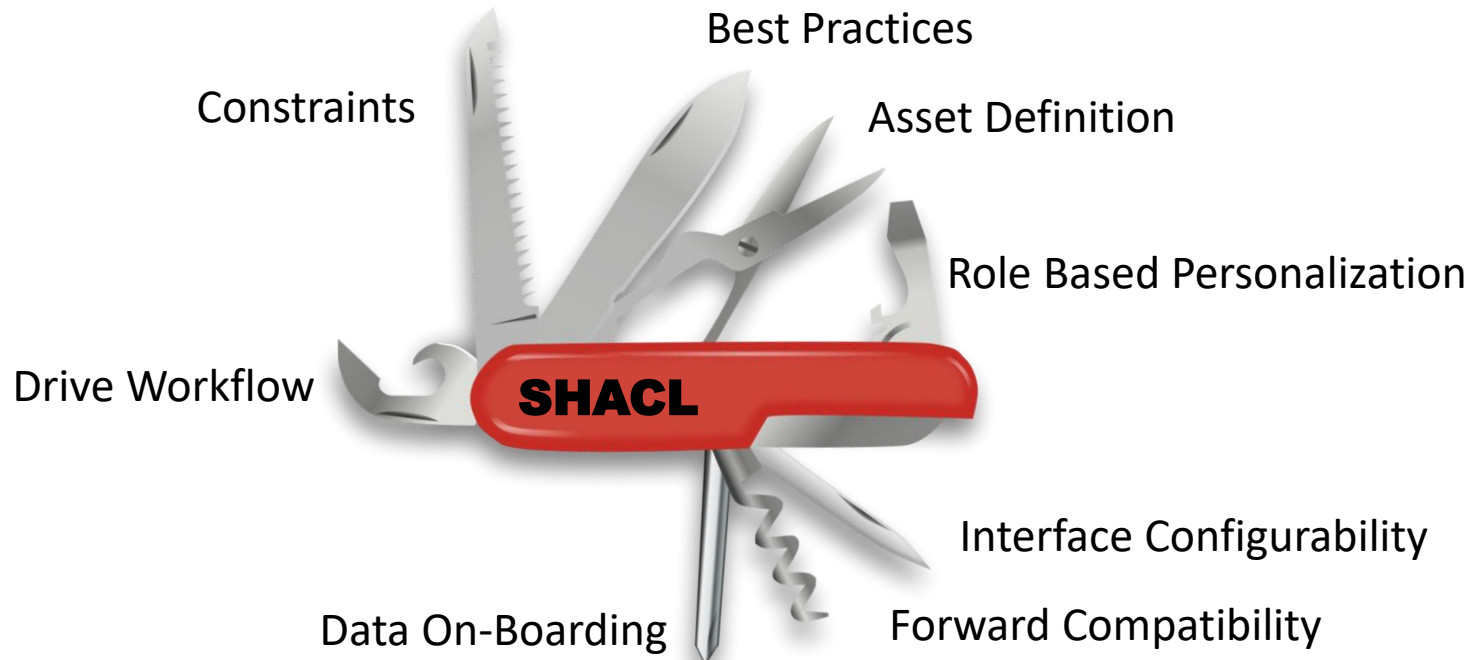
- Every GraphQL service treated as RDF
  - GraphQL schemas converted to SHACL shapes
  - JSON instances converted to RDF nodes
- Every RDF source treated as GraphQL
  - SHACL shapes converted to GraphQL schemas
  - RDF graphs exposed as GraphQL service
- GraphQL libraries and tools add to the RDF infrastructure
- RDF+SHACL brings IDs (URIs) to GraphQL objects, richer query, inferencing



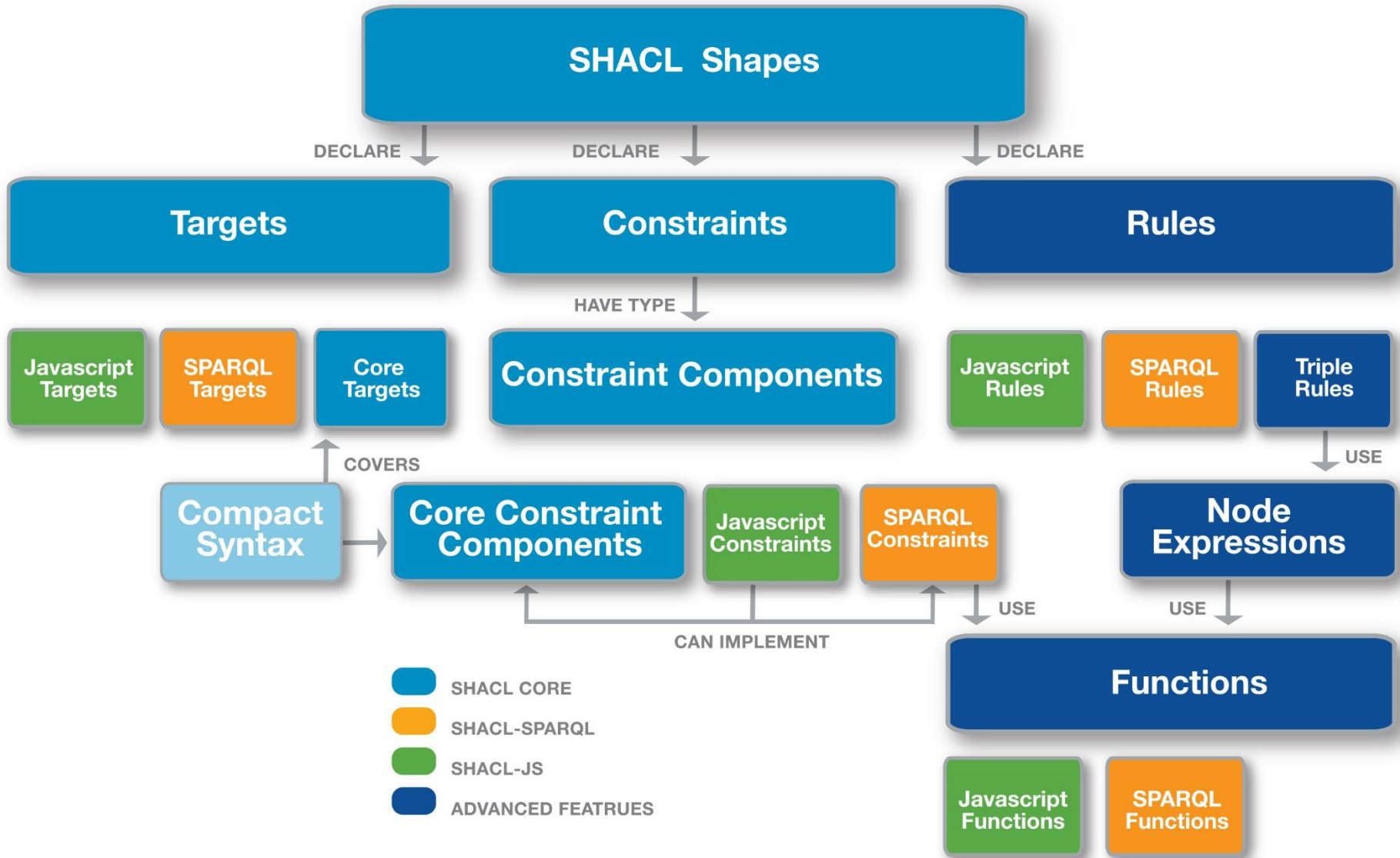
# Learn SHACL Once, Use for Many Needs

A single language for delivering a range of capabilities

All of the value propositions and features listed so far are important for a data governance tool. Each could be supported using its own technical approach requiring a separate investment in understanding and learning it. Supporting all of them with SHACL brings consistency. Users do not need to learn different approaches to take advantage of these capabilities.



# Detailed View of SHACL



## In Conclusion

- I have given a brief overview of SHACL
- Demoed FIBO in SHACL
- Discussed SHACL's features and use case – for data governance and beyond
  
- Come to the booth for more !



... Questions?

# Resources for learning SHACL

- SHACL Community Group <https://www.w3.org/community/shacl/>
- SHACL W3C Wiki [https://www.w3.org/2014/data-shapes/wiki/Main\\_Page](https://www.w3.org/2014/data-shapes/wiki/Main_Page)
  - Links to implementations, WG deliverables, Meeting minutes, ...
  - Some historic info – not so useful anymore
- TQ's SHACL page - <http://www.topquadrant.com/technology/shacl/>
  - Tutorials, articles, presentations – for example:
  - [AN OVERVIEW OF SHACL FEATURES AND SPECIFICATIONS](#)
  - [USING SHACL DATA CONSTRAINTS IN THE TOPBRAID WEB PRODUCTS EVN AND EDG](#)
  - [HOW TO DEFINE CONSTRAINTS ON RDF:LISTS USING SHACL](#)
  - [HOW TO USE TOPBRAID AS A DATA VALIDATION SERVER](#)
- SHACL OVERVIEW WEBINARS:
  - <https://www.topquadrant.com/knowledge-assets/topquadrant-webinars/#TQ-SHACL-overview-webinar>

To contact me, use: [irene@topquadrant.com](mailto:irene@topquadrant.com)

## To Learn More about TopBraid EDG, see:

### *EDG Product Info:*

- <http://www.topquadrant.com/products/topbraid-edg/>
- <http://www.topquadrant.com/products/topbraid-edg-gov-packs/>

### *EDG demos/webinar recordings:*

- <https://www.topquadrant.com/knowledge-assets/topquadrant-webinars/#TQ-EDG-appliedDG-part2-webinar>  
Webinar: Applied Data Governance – Part 2: A Day in the life of a Technical Data Steward
- <https://www.topquadrant.com/knowledge-assets/topquadrant-webinars/#TQ-EDG-realworldDG-part1-webinar>  
Webinar: Applied Data Governance – Part 1: A Day in the life of a Business Steward
- <http://www.topquadrant.com/knowledgeassets/videos/#edgoverviewdemo>  
Webinar: Data Governance for the Connected Enterprise: TopBraid EDG in Action
- <http://www.topquadrant.com/knowledge-assets/topquadrant-webinars/#TQ-EDG-metadata-mgt-webinar>  
Webinar: Metadata Management is Key to Data Governance Initiatives