



***Semantics in the EPIM ReportingHub -  
Its Semantic Web-based Approach to  
ISO 15926***

**ISO 15926 and Semantic Technologies 2012**

**Sogndal 6 - 7. September 2012**

David Price, TopQuadrant  
London, UK  
dprice at topquadrant dot com



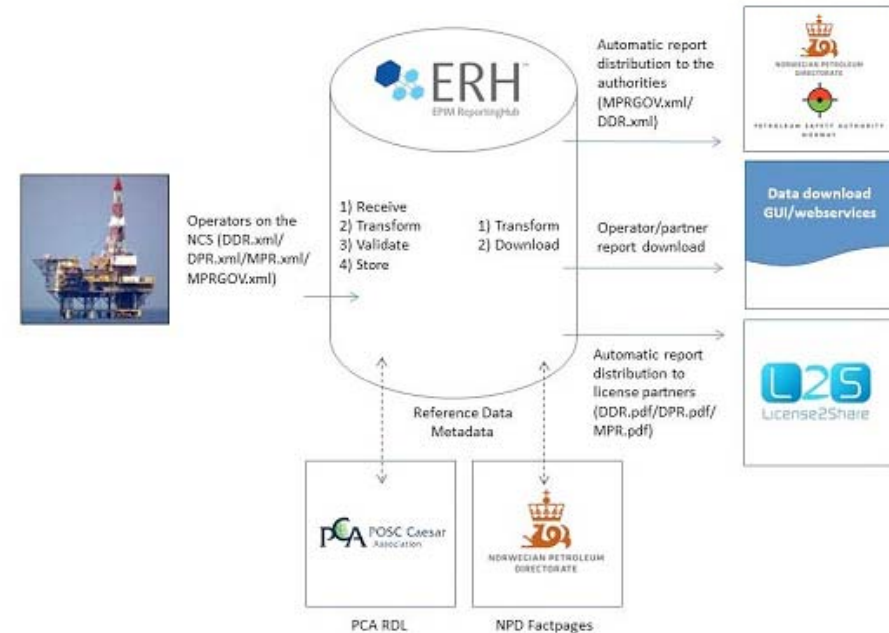
# Agenda

- ❑ The business problem and requirements
- ❑ The EPIM Reporting Hub solution architecture and semantics
- ❑ Examples of the ERH Approach to ISO 15926
- ❑ Conclusions



# EPIM ReportingHub (ERH)

- Shall receive drilling and production reports from operators on the NCS, validate, store and send them to authorities and partners
- Shall provide drilling and production data to the partners on request
- Shall be flexible for changes in the reports and easy to extend to new types of reports
- Shall be based on NPD Fact Pages and PCA E&P ontology





# EPIM ERH Core Requirements

- ❑ Phase 1 requirements limited, future phases can more capability
- ❑ Delivered to EPIM as **Software as a Service**
  - High availability, scalable service
  - Delivered on existing, secure network in Norway called SOIL
- ❑ Enterprise RDF database
- ❑ Interfaces
  - Simple browser-based user interface for report upload, download, query
  - HTTP service interface for report upload, download, query
- ❑ Semantics based in ISO 15926 standard and delivered to PCA as Reference Data (provides interoperability)



# EPIM ERH Core Requirements (2)

- ❑ Driven by Norwegian Petroleum Directorate (NPD) Facts
  - Validation (single source of truth about Norwegian Continental Shelf)
  - Access control based in licenses and partners
  - Query through Facts
- ❑ Input reports are XML files based on standard XML Schemas
  - Daily Drilling Report
  - Daily Production Report
  - Monthly Production Report (Government and Partner)
  - ... others in the future
- ❑ Outputs
  - submitted XML transmitted to Authorities via service interfaces and stored for later download by operators and partners
  - Generated PDF of report content driven from RDF database
  - Parameterized query over RDF database

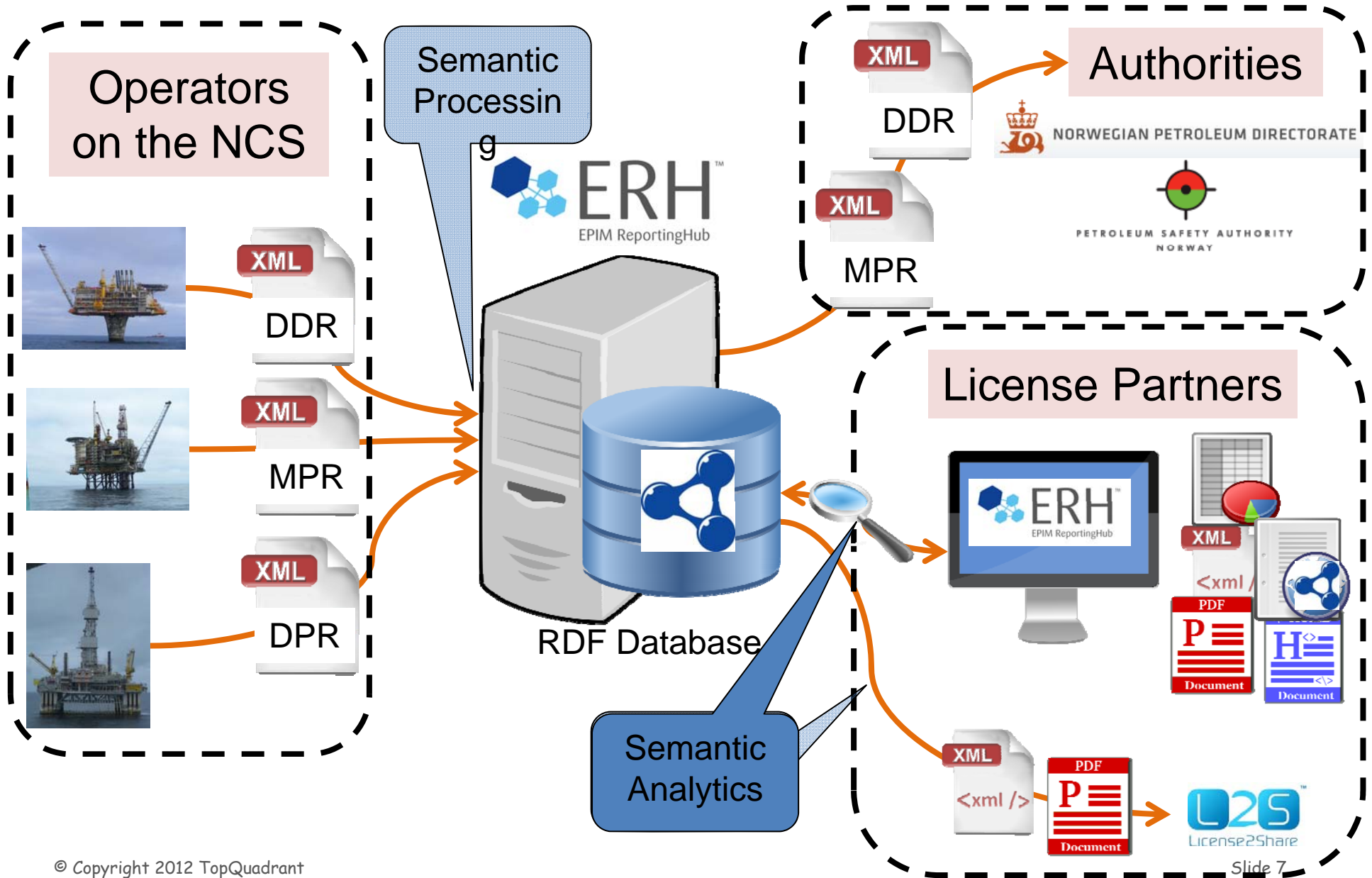


# Scale of ERH Solution

- ❑ 300 million triples in the next 4 years
  - Potential for many more if data sources added or historical data imported
- ❑ 40+ concurrent users
  - User interfaces and Service interfaces required
  - Potential for many more if data sources added
- ❑ Phase 1 XML Schemas have 2000-ish elements
  - resulting ontologies have 900-ish classes and 900-ish properties
- ❑ Delivered on an SaaS basis with high availability Service Level Agreement by TopQuadrant, Franz and Logica
  - Secondary app server running at all times
  - RDF database replication and warm fallover



# What does reporting mean?

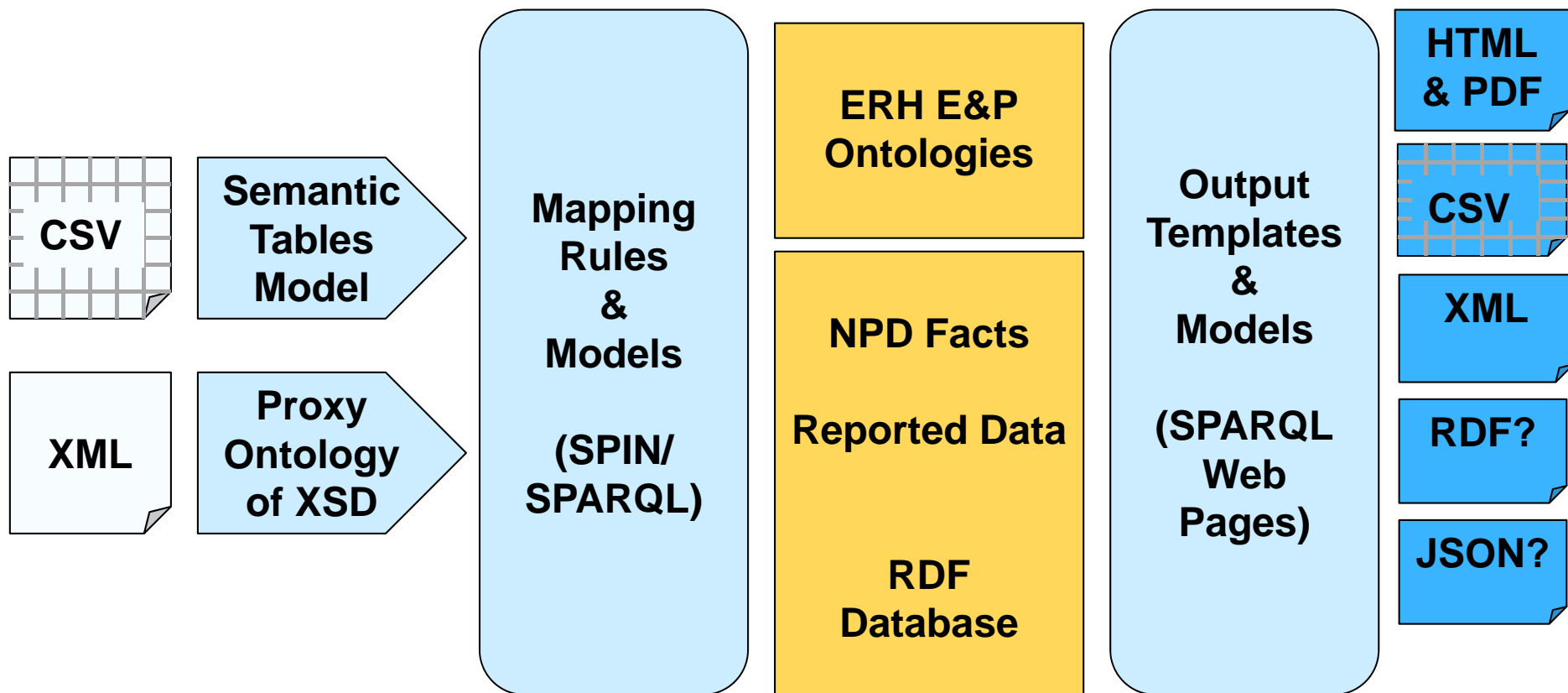


# Semantic Processing and Reporting

Upload

Processing

Download







## *It's Turtles all the way down*

- ❑ Core of approach to ERH development
  - Everything is RDF/OWL ontology-driven
  - Build framework so semantic processing and reporting can be done by modeling team, not Java development team
  - Avoid Oil & Gas/ISO 15926 -specific technical architecture components
- ❑ How?
  - Make all inputs look like ontologies and triples
  - Use SPIN/SPARQL for transformation from source to target, regardless of actual source
  - Build data flow pipes : XML->triples->SPIN->graph in RDF database->SWP->HTML->PDF



# Key technologies for ERH

- ❑ Semantic Tables makes spreadsheets and CSV data appear as triples
- ❑ XML Schema importer automatically builds *proxy ontology* of XML Schema
  - Imported of XML data then appears as triples based on the proxy ontology
- ❑ Now that all data is triples, we can use Semantic Web tools to process it
  - NPD Facts harvested daily as CSV download from NPD Web site
  - XML reports imported into data flow pipe as submitted



## Key technologies for ERH (2)

- ❑ SPIN – SPARQL Inference Notation
  - W3C Member Submission
  - SPIN/SPARQL rules for transforms
  - SPIN/SPARQL constraints for access control to named graphs
  - SPIN Templates, Constraints, Functions and functions enables modularity and reuse of parameterized SPARQL
  - SPINMap graphical mapping capability
- ❑ SPARQL Web Pages
  - Comparable to JSP, Flex, PHP for typical Web development
  - Template HTML pages with embedded SPARQL



# NPD Facts

## FACTPAGES

NORWEGIAN PETROLEUM DIRECTORATE



Wellbore Licence BAA **Field** Discovery Company Seismic Facility TUF Pipeline Stratigraphy

- Attributes
- Page view
  - All
  - PDO approved
  - Producing
  - Shut down
  - Ordered by..
- Table view
  - Overview**
  - Operators
  - Owners
  - Licensees
  - Production
  - Reserves
  - Investments
  - Description
  - Geography

Hide details Show details Map Feedback Download/Export ▶

Field name	Operator name	Current activity status	Discovery wellbore	Completion date, disc. wb.	Owner kind	Owner name	NPDID owner	NPDID field	NPDID wellbore
<a href="#">ALBUSKJELL</a>	ConocoPhillips Skandinavia AS	SHUT DOWN	1/6-1	26.11.1972	PRODUCTION LICENSE	018	20900	43437	239
<a href="#">ALVE</a>	Statoil Petroleum AS	PRODUCING	6507/3-1	26.10.1990	PRODUCTION LICENSE	159 B	2819945	4444332	1533
<a href="#">ALVHEIM</a>	Marathon Oil Norge AS	PRODUCING	24/6-2	08.07.1998	PRODUCTION LICENSE	203	22436	2845712	3397
<a href="#">ATLA</a>	Total E&P Norge AS	PDO APPROVED	25/5-7	24.10.2010	PRODUCTION LICENSE	102 C	5467035	21106284	6423
<a href="#">BALDER</a>	ExxonMobil Exploration & Production	PRODUCING	25/11-1	09.07.1967	PRODUCTION LICENSE	001	20756	43562	143



# Data transformed to triples using SPIN

- ❑ Implementing converters = writing SPARQL or using SPINMap, not Java development
- ❑ Same approach regardless of source being XML or CSV

```

spin:rule (incomplete)
# STEP 101 Create DailyDrillingReport, DailyDrillingActivity, TemporalPartOfAWellBore and WellBore
# Minimum data - createDate, dTimStart, nameWellBore
CONSTRUCT {
?dailyDrillingReport a ep-report:DailyDrillingReport .
?dailyDrillingActivity a ep-activity:DailyDrillingActivity .
?dailyDrillingReport ep-activity:reportOn ?dailyDrillingActivity .
?dailyDrillingReport ep-activity:createdAt ?createDate .
?dailyDrillingActivity ep-activity:startedAt ?dTimStart .
?temporalPartOfAWellBore a ep-well:TemporalPartOfAWellBore .
?wellBore a ep-well:WellBore .
?temporalPartOfAWellBore ep-core:temporalPartOf ?wellBore .
?wellBore ep-core:name ?nameWellBore .
?dailyDrillingActivity ep-activity:onWellBore ?temporalPartOfAWellBore .
}
WHERE {
?this ddr:nameWellbore ?nameWellBore .
?this ddr:dTimStart ?dTimStart .
?this ddr:dTimStart ?dTimEnd .
?this ddr:createDate ?createDate .
BIND (ep-spin-lib:normalizeString(?nameWellBore) AS ?normalizedWellBoreName) .
BIND (ep-spin-lib:buildDailyDrillingReportURI(?normalizedWellBoreName, ?createDate) AS ?dailyDrillingReportURI) .
BIND (ep-spin-lib:buildDailyDrillingActivityURI(?normalizedWellBoreName, ?dTimStart) AS ?dailyDrillingActivityURI) .
BIND (ep-spin-lib:buildTemporalPartOfAWellBoreURI(?normalizedWellBoreName, ?dTimStart, ?dTimEnd) AS ?temporalPartOfAWellBoreURI) .
BIND (rhspin:wellBoreByName(?nameWellBore) AS ?wellBore) .
}

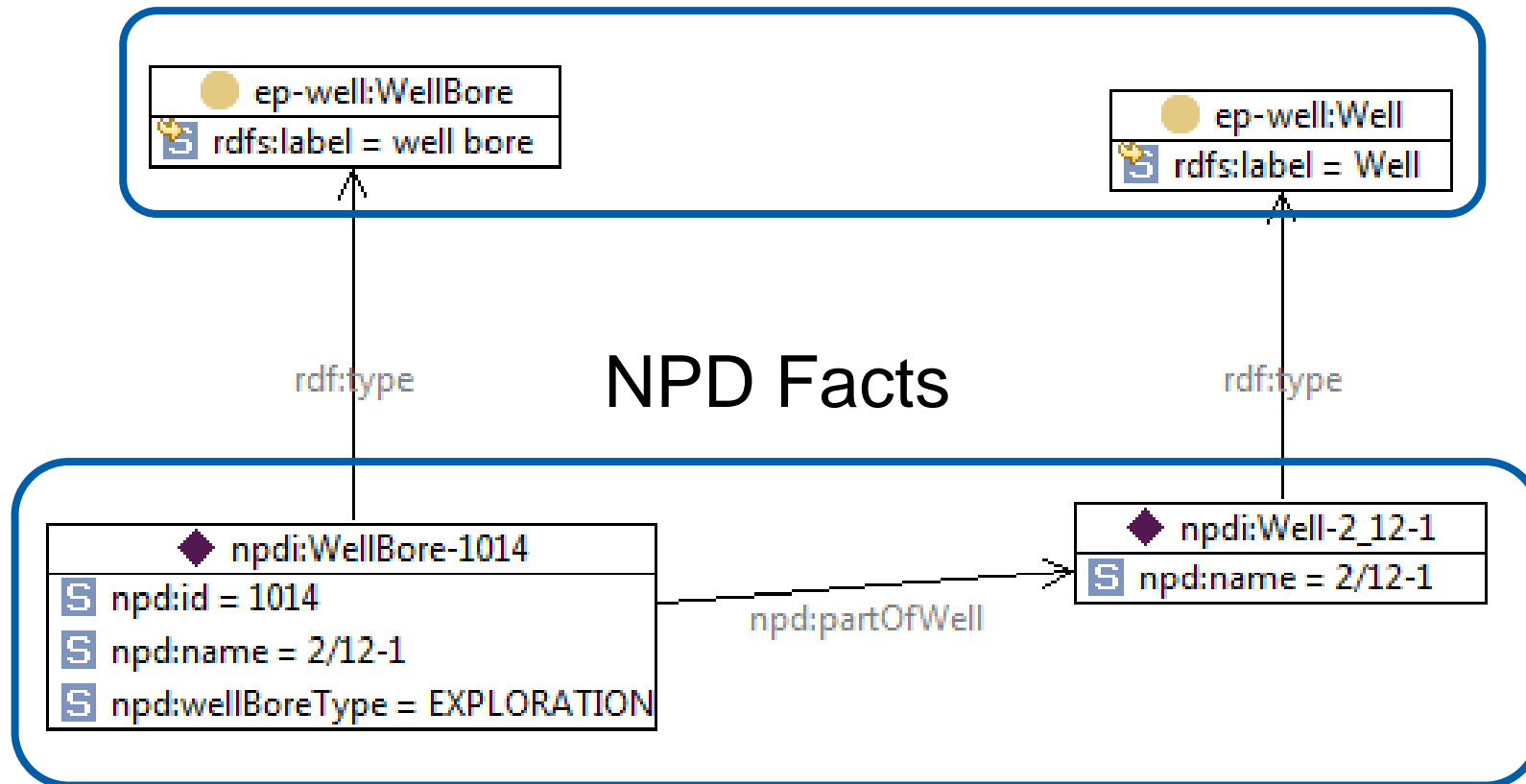
```

The diagram illustrates the transformation of data from a source resource 'a:Person' to target resources 'b:Address' and 'b:Customer'. The source resource 'a:Person' has properties: a:city, a:country, a:street, a:zipCode, a:dob, a:firstName, a:lastName, and a:state. The target resource 'b:Address' has properties: b:country, b:postalCode, b:streetAddress, b:suburb, and rdfs:label. The target resource 'b:Customer' has properties: b:address, b:birthDate, and b:fullName. The transformation is achieved through several SPINMap operations: 'build URI' to create the target URI, 'resource with value' to map source properties to target properties, 'xsd:integer' to handle data types, 'target resource' to identify the target resource, 'ddmmyyyy 2 date' to format dates, and 'concat with separator' to concatenate strings.



# NPD Facts as OWL

## ERH E&P and NPD Classes







# TopBraid creates proxy ontology automatically

TopBraid Resource

Navigator Classes

- Cs\_drillReportStatusInfo
- Cs\_drillReportStratInfo
- Cs\_drillReportSurveyStation
- Cs\_drillReportWellboreInfo
- Cs\_drillReportWellTestInfo
- Cs\_fluid
- Cs\_geodeticModel
- Cs\_localCRS
- Cs\_location
- Cs\_projectionx
- Cs\_refWellWellbore
- Cs\_refWellWellboreRig**
- Cs\_stimJobStage
- Cs\_stimTubular
- Cs\_wellAlias
- Cs\_wellCRS
- Cs\_wellDatum
- dtype:EnumeratedValue
- dtype:Enumeration
- dtype:ValueReference
- IndexedObject

SCHEMA\_EP-DDR-xsd-proxy-v1.0.ttl

## Class Form

URI: [http://www.witsml.org/schemas/1series#Cs\\_refWellWellboreRig](http://www.witsml.org/schemas/1series#Cs_refWellWellboreRig)

**Annotations**

- xmlns:element
- rdfs:comment
  - A reference to a rig within a wellbore. The wellbore may be defined represents a foreign key from one node to another.
- rdfs:label
  - Cs ref well wellbore rig

**Class Axioms**

- rdfs:subClassOf
  - owl:Thing
  - rigReferenceRef **exactly 1**
  - rigReferenceRef **only** RefNameString
  - wellParentRef **max 1**
  - wellParentRef **only** RefNameString
  - wellboreParentRef **max 1**
  - wellboreParentRef **only** RefNameString
- owl:equivalentClass





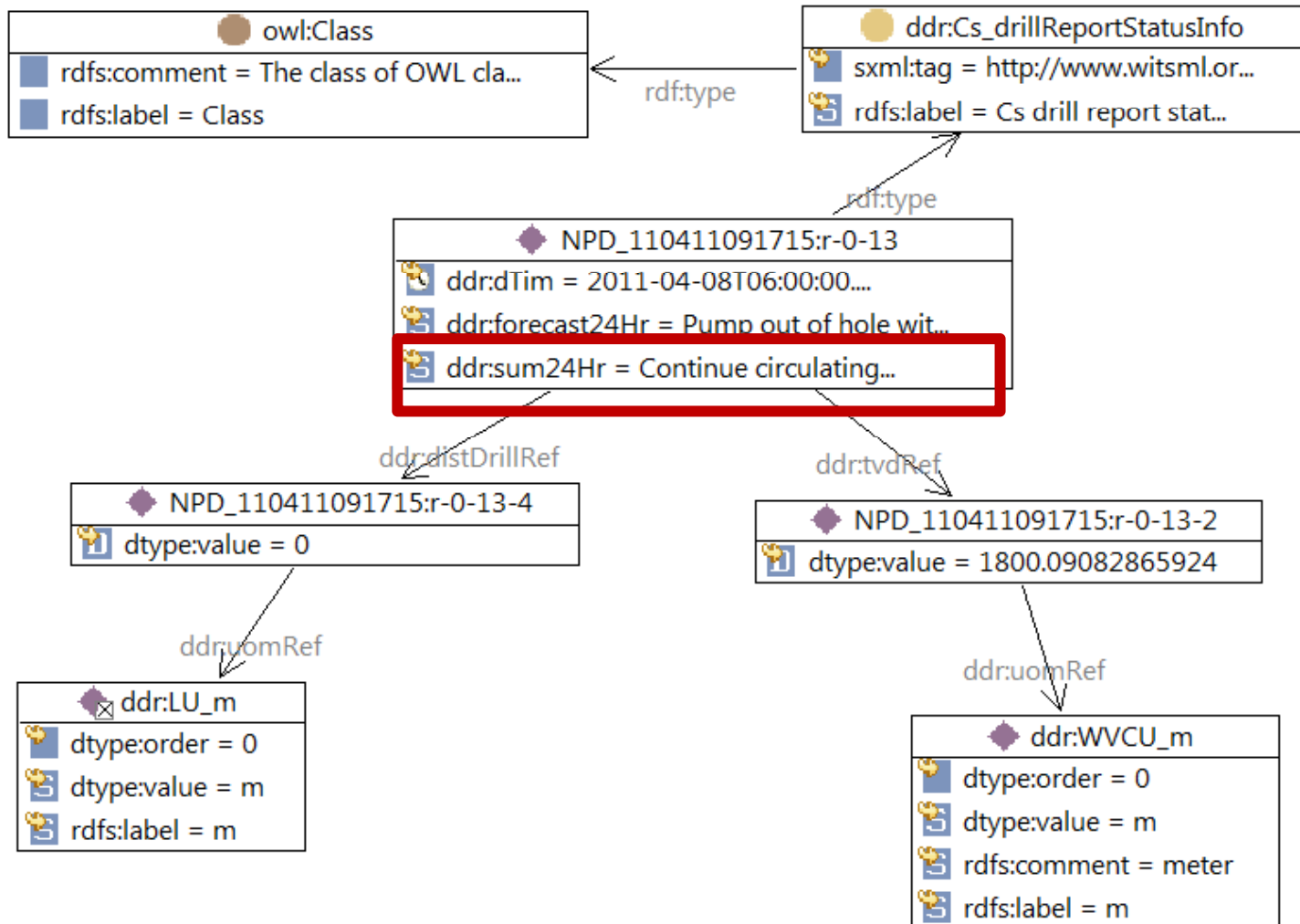
# XML input data example

```

- <witsml:statusInfo>
  <witsml:dTim>2011-04-08T06:00:00.001Z</witsml:dTim>
  <witsml:md uom="m">6414</witsml:md>
  <witsml:tvd uom="m">1800.09082865924</witsml:tvd>
  <witsml:presTestType>formation integrity test</witsml:presTestType>
  <witsml:distDrill uom="m">0</witsml:distDrill>
  <witsml:elevKelly uom="m">60.2999992370605</witsml:elevKelly>
  <witsml:sum24Hr>Continue circulating and cleaning hole section with 1.30 sg mud in hole. Increase pump
    rate in attempt to increase hole cleaning. Lost returns and hole attempting to pack off. Shut in well to
    maintain pressure on wellbore and establish circulation through choke. Circulate bottoms up through
    choke while maintaining pressure on wellbore. Open well and attempt to re-establish circulation. Hole
    attempting to pack off and losing returns. Continue to adjust parameters to maintain circulating and
    returns. </witsml:sum24Hr>
  <witsml:forecast24Hr>Pump out of hole with slow rotation to 9-5/8" casing shoe. Circulate and condtion
    mud. Increase MW back to 1.38. </witsml:forecast24Hr>
</witsml:statusInfo>

```

# Imported XML data appear as instances of proxy ontology





# Now can transform using SPIN/SPARQL

```

pin:rule (incomplete) ▾
# STEP 101 Create DailyDrillingReport, DailyDrillingActivity, TemporalPartOfAWellBore and WellBore
# Minimum data - createDate, dTimStart, nameWellBore
CONSTRUCT {
  ?dailyDrillingReport a ep-report:DailyDrillingReport .
  ?dailyDrillingActivity a ep-activity:DailyDrillingActivity .
  ?dailyDrillingReport ep-activity:reportOn ?dailyDrillingActivity .
  ?dailyDrillingReport ep-activity:createdAt ?createDate .
  ?dailyDrillingActivity ep-activity:startedAt ?dTimStart .
  ?temporalPartOfAWellBore a ep-well:TemporalPartOfAWellBore .
  ?wellBore a ep-well:WellBore .
  ?temporalPartOfAWellBore ep-core:temporalPartOf ?wellBore .
  ?wellBore ep-core:name ?nameWellBore .
  ?dailyDrillingActivity ep-activity:onWellBore ?temporalPartOfAWellBore .
}
WHERE {
  ?this ddr:nameWellbore ?nameWellBore .
  ?this ddr:dTimStart ?dTimStart .
  ?this ddr:dTimStart ?dTimEnd .
  ?this ddr:createDate ?createDate .
  BIND (ep-spin-lib:normalizeString(?nameWellBore) AS ?normalizedWellBoreName) .
  BIND (ep-spin-lib:buildDailyDrillingReportURI(?normalizedWellBoreName, ?createDate) AS ?dailyDrillingReport) .
  BIND (ep-spin-lib:buildDailyDrillingActivityURI(?normalizedWellBoreName, ?dTimStart) AS ?dailyDrillingActivity) .
  BIND (ep-spin-lib:buildTemporalPartOfAWellBoreURI(?normalizedWellBoreName, ?dTimStart, ?dTimEnd) AS
?temporalPartOfAWellBore) .
  BIND (rhspin:wellBoreByName(?nameWellBore) AS ?wellBore) .
}
# STEP 101 - Set name of DailyDrillingReport as label

```



# Processed by SPIN engine to make triples

[Subject]	Predicate	Object
<https://www.reportinghub.no/ep/data/WellBore/25_8-C-23_BS2011-04-10T19:0...>	ep-core:temporalPartOf	npdi:WellBore-6480
<https://www.reportinghub.no/ep/data/WellBore/25_8-C-23_BS2011-04-10T19:0...>	ep-well:finalBottomHoleMeasuredDepth	<@618d3d65:13313e45b6c:-5fcd>
<https://www.reportinghub.no/ep/data/WellBore/25_8-C-23_BS2011-04-10T19:0...>	ep-well:finalBottomHoleTrueVerticalDepth	<@618d3d65:13313e45b6c:-5fb3>
<https://www.reportinghub.no/ep/data/WellBore/25_8-C-23_BS2011-04-11T01:3...>	ep-core:temporalPartOf	npdi:WellBore-6480
<https://www.reportinghub.no/ep/data/WellBore/25_8-C-23_BS2011-04-11T01:3...>	rdf:type	ep-well:TemporalPartOfAWellBore
<https://www.reportinghub.no/ep/data/WellBore/25_8-C-23_BS2011-04-11T01:3...>	ep-well:finalBottomHoleMeasuredDepth	<@618d3d65:13313e45b6c:-5fce>
<https://www.reportinghub.no/ep/data/WellBore/25_8-C-23_BS2011-04-11T01:3...>	ep-well:finalBottomHoleTrueVerticalDepth	<@618d3d65:13313e45b6c:-5fb4>
<https://www.reportinghub.no/ep/data/WellBoreDrillBit/25_8-C-23_BSB6>	rdf:type	<@618d3d65:13313e45b6c:-5ef9>
<https://www.reportinghub.no/ep/data/WellBoreDrillBit/25_8-C-23_BSB6>	ep-eqt:manufacturedBy	<@618d3d65:13313e45b6c:-5ef7>
<https://www.reportinghub.no/ep/data/WellBoreDrillBit/25_8-C-23_BSB6>	ep-eqt:manufacturedBy	<@618d3d65:13313e45b6c:-5eff>
<https://www.reportinghub.no/ep/data/WellBoreDrillBit/25_8-C-23_BSB6>	rdf:type	<@618d3d65:13313e45b6c:-5f01>
<https://www.reportinghub.no/ep/data/WellBoreDrillBit/25_8-C-23_BSB6>	rdf:type	<@618d3d65:13313e45b6c:-5f09>
<https://www.reportinghub.no/ep/data/WellBoreDrillBit/25_8-C-23_BSB6>	ep-core:name	b6
<https://www.reportinghub.no/ep/data/WellBoreDrillBit/25_8-C-23_BSB6>	ep-eqt:manufacturedBy	<@618d3d65:13313e45b6c:-5f07>
<https://www.reportinghub.no/ep/data/WellBoreDrillBit/25_8-C-23_BSB6>	rdf:type	ep-eqt:DrillBitForWellBore



# ... as triples with report data extending them

## NPD Fact

◆	npdi:WellBore-6480
Ⓢ	ep-core:humanRecognizableName = 25/8-C-23 B
Ⓢ	ep-core:npdCode = 25/8-C-23 B
Ⓢ	ep-core:npdNumber = 6480
🕒	ep-well:wellBoreSpudAt = 2011-01-24T06:00:00....
🔑	npd:id = 6480
🔑	npd:name = 25/8-C-23 B
📅	npd:startDate = 2011-03-01
📅	npd:wellBoreCompletionDate = 2011-05-12
🔑	npd:wellBoreDrillPermit = 3294-P
🔑	npd:wellBoreType = DEVELOPMENT

●	ep-well:WellBore
Ⓢ	rdfs:label = wellbore (whole life...)

●	ep-well:TemporalPartOfAWellBore
Ⓢ	rdfs:label = (temporal part of a)...

Wellbore on a day is  
*a temporalPartOf*  
whole-life Wellbore

whole

part

ep-core:temporalPartOf

◆ <https://www.reportinghub.no/ep/data/WellBore/25\_8-C-23\_B\$2011-04-09T06:00:00.001Z\$2011-04-10T06:00:00.001Z>

ep-activity:onWellBore

◆	<https://www.reportinghub.no/ep/data/DailyDrillingActivity/25_8-C-23_B\$2011-04-09T06:00:00.001Z\$2011-04-10T06:00:00.001Z>
🕒	ep-activity:finishedAt = 2011-04-10T06:00:00....
🕒	ep-activity:startedAt = 2011-04-09T06:00:00....

part of wellbore

●	ep-activity:DailyDrillingActivity
Ⓢ	rdfs:label = daily drilling activ...

●	ep-report:NormalDailyDrillingReport
Ⓢ	rdfs:label = normal daily drillin...

activity

p-report:reportOn

rdfs:type

●	ep-report:DailyDrillingReport
---	-------------------------------

Daily activity happens on  
*TemporalPartOfAWellbore*



# Semantic Reporting : SPARQL Web Pages

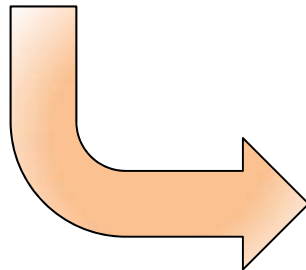
- ❑ For each report output type, create SWP HTML
  - Daily Drilling, Daily Production, Monthly Production Government, Monthly Production Partner
  - Normal CSS for style
- ❑ Embed SPIN/SPARQL directly, as functions or through tables capability
  - Query is over E&P ontology RDF data including NPD Fact data
- ❑ Generate PDF from HTML using off-the-shelf tool





# Embedded in HTML page

```
<table>
<tr>
<td class="ddrSummaryLeft">
  <rhwsp:NameValuePair arg:name="NPD Wellbore Id:"
    arg:value="{= rhspin:npdId(?wb) }" />
  <rhwsp:NameValuePair arg:name="Report Status:"
    arg:value="{= ep-spin-lib:selectDailyDrillingReportFlag(?repType) }" />
  <rhwsp:NameValuePair arg:name="Creation time:"
    arg:value="{= ep-spin-lib:normalizeDateTime(?createdAt) }" />
  <rhwsp:NameValuePair arg:name="Operator:"
    arg:value="{# SELECT ?operatorName WHERE
      {?activity ep-activity:drillingOperator ?operator .
        ?operator ep-core:name ?operatorName }}" />
```



## Summary report Wellbore: 2/4-K-8 B

Period: 2011-Sep-26, 00:00 - 2011-Sep-27, 00:00

Report Number:

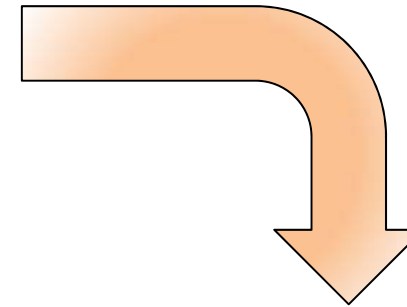
NPD Wellbore Id:	6609	Depth at Kick Off mMD:	1735.58
Report Status:	normal	Depth at Kick Off mTVD:	785.98
Creation time:	2011-Sep-27, 11:08	Depth mMD:	1250.948
Operator:	ConocoPhillips	Depth mTVD:	987.092
Drilling contractor:	ARCHER	Dist Drilled m:	45.89
Rig name:	EKOFISK K	Penetration Rate m/h:	6.22
NPD Rig Id:	271469	Hole Dia in:	43.33
Fixed rig:		Elevation RKB-MSL m:	56.693
		Plug Back Depth:	



# Query to Table

```
<ddrswp:Table arg:report="{= ?report }"
  arg:template="ddr-tables:LogInformation" />
```

```
spin:body
★ SELECT ?runNo ?depthTopmMD ?depthBottommMD ?tool ?bHCTdegC ?bHSTdegC ?serviceCompanyName
WHERE {
  ?report ep-report:reportOn ?dailyDrillingActivity .
  ?dailyDrillingActivity a ep-activity:DailyDrillingActivity .
  ?dailyDrillingActivity ep-core:hasPart ?logRun .
  ?logRun a ep-activity:WellBoreLogRun .
  ?logRun ep-activity:onWellBore ?wbState .
  OPTIONAL {
    ?logRun ep-activity:logRunNumber ?runNo .
  } .
  OPTIONAL {
    ?logRun ep-activity:loggedWellBoreInterval ?wbPartState .
    ?wbPartState ep-core:temporalPartOf ?wbPart .
    OPTIONAL {
      ?wbPart ep-well:measuredDepthToTheTopOfAPartOfAWellBore ?top .
      ?top ep-core:metre ?depthTopmMD .
    } .
    OPTIONAL {
      ?wbPart ep-well:measuredDepthToTheBottomOfAPartOfAWellBore ?bottom .
      ?bottom ep-core:metre ?depthBottommMD .
    } .
  } .
  OPTIONAL {
    ?circWb ep-core:temporalPartOf ?wbState .
    ?circWb a ep-well:WellBoreWithCirculation .
    ?circWb ep-well:bottomHoleTemperature ?circButtHoleTemp .
    ?circButtHoleTemp ep-core:celsius ?bHCTdegC .
  } .
  OPTIONAL {
    ?staticWb ep-core:temporalPartOf ?wbState .
    ?staticWb a ep-well:WellBoreWithoutCirculation .
    ?staticWb ep-well:bottomHoleTemperature ?staticButtHoleTemp .
    ?staticButtHoleTemp ep-core:celsius ?bHSTdegC .
  } .
  OPTIONAL {
    ?logRun ep-activity:wellBoreLoggingTool ?loggingToolState .
    ?loggingToolState ep-core:temporalPartOf ?loggingTool .
    ?loggingTool ep-core:description ?tool .
  } .
  OPTIONAL {
    ?logRun ep-activity:serviceCompany ?serviceCompany .
    ?serviceCompany ep-core:name ?serviceCompanyName .
  } .
}
```



## Log Information

Run no	Depth Top mMD	Depth Bottom mMD	Tool	BHCT degC	BHST degC
1	2427.67	2399.81	Gamma		107.49
4	2067.21	2412.13	Acoustic		111.35
1	2111.17	2093.71	Electrical		107.06





# Why SPIN for ERH?

- ❑ SPIN built on SPARQL
  - active W3C group and implementor community
  - on RDF, so not limited to OWL (e.g. can SPIN SKOS)
  - store transforms, queries, constraints, rules as RDF, but with human front end editor
- ❑ For data integrity constraints RDF+OWL+SPIN is powerful
  - Can close world and limit constraint scope (e.g. constraint on Class)
  - Can do inference and compute derived properties
  - Is a software engineer's tool (e.g. can add functions written using Javascript)



# Semantic Challenges

- ❑ ERH project is *The Semantic Web meets ISO 15926*
  - RDF/OWL, ISO 15926, SPARQL query, reasoning all mentioned in requirements
  - ISO 15926 upper ontology designed for data exchange with no domain content, that comes as ***reference data***
  - Blending OWL and bits of ISO 15926 that are a modeling language is difficult
  - Existing reference data incomplete for ontology building (e.g. no reference data connection between what in OWL are Classes and related Properties)



# Use of 15926

- ❑ For ERH requirements, an approach closer to the structures in 15926-2 data model made more sense, and the vocabulary required is actually relatively small
  - Drilling : Activity
  - Production : Activity, Flow (Stream) and Product (Material)
  - and properties are about temporal parts of these
- ❑ Provenance is handled by modeling the Report itself and linking all Activity/Flow instance to the Report
  - no need for reification of classification, composition
- ❑ 15926 Templates
  - Will define template for PCA from the indirect properties
  - Used as data definition for services extracting data from ERH



# Use of Semantic Web and RDF/OWL

- ❑ ISO 15926 complexity -> preference for expressiveness in modeling capability over decidability
  - But keep interest in reasoning in future in mind
  - Ontology architecture and partitioning named graphs should mean we can satisfy reasoning requirements
  - If not, then SPIN/SPARQL rules or transform then DL reasoner (or both)
- ❑ SPARQL is the phase 1 query language
- ❑ Under the covers, the triples from a report are maintained using named graphs
  - This is also how access control is managed
- ❑ URI strategy
  - `https:<reportinghub><type>-<identifier>`
  - Give URIs to whole-life objects and use blank nodes for cases like temporal parts

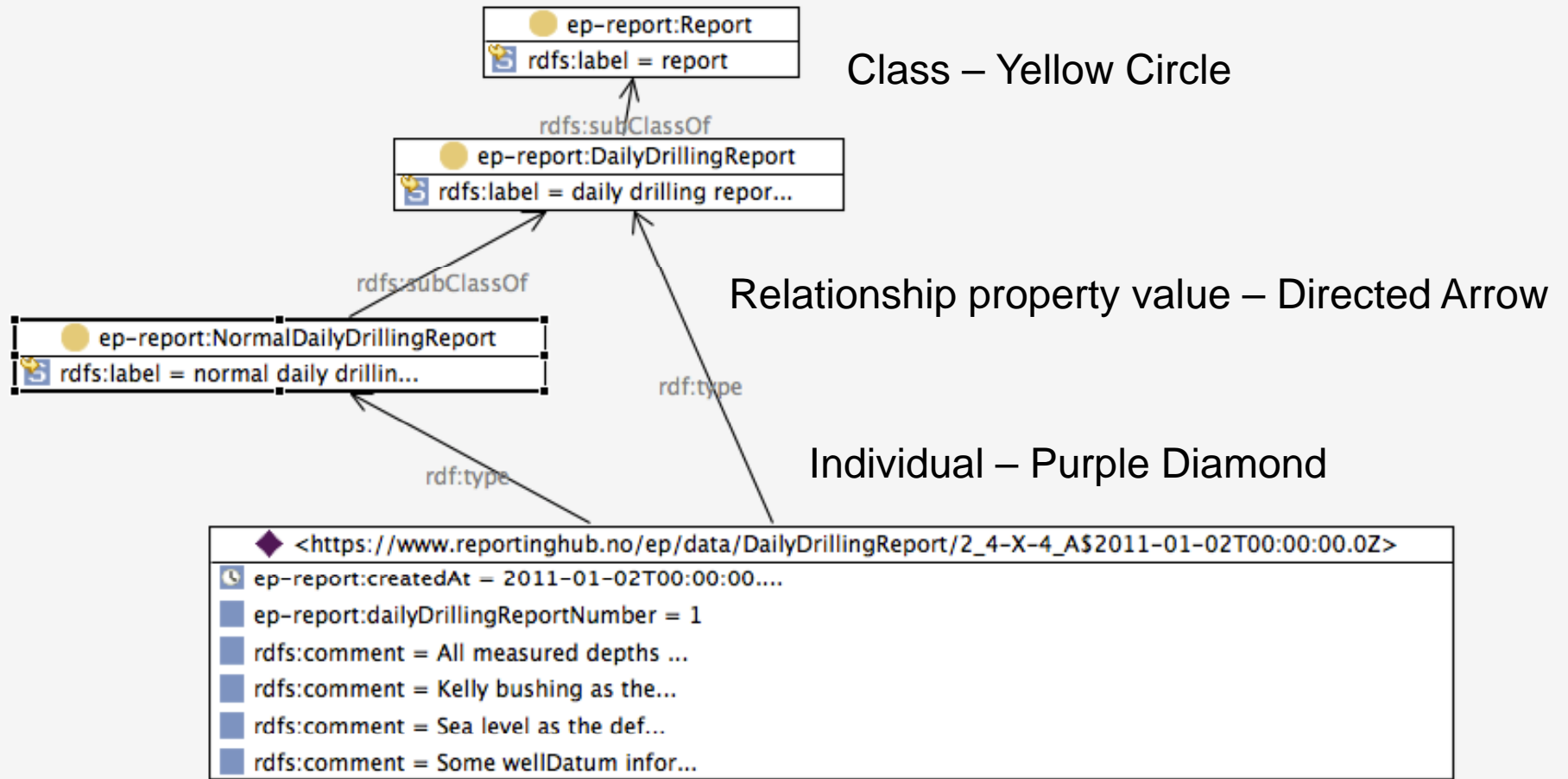


## Examples of the ERH Approach to ISO 15926

- ❑ Use a data-driven approach to show ERH approach to ISO 15926 using Daily Drilling Report example
- ❑ Start with Report and follow links through the data as graph diagrams
  - Next two pages are examples with legend

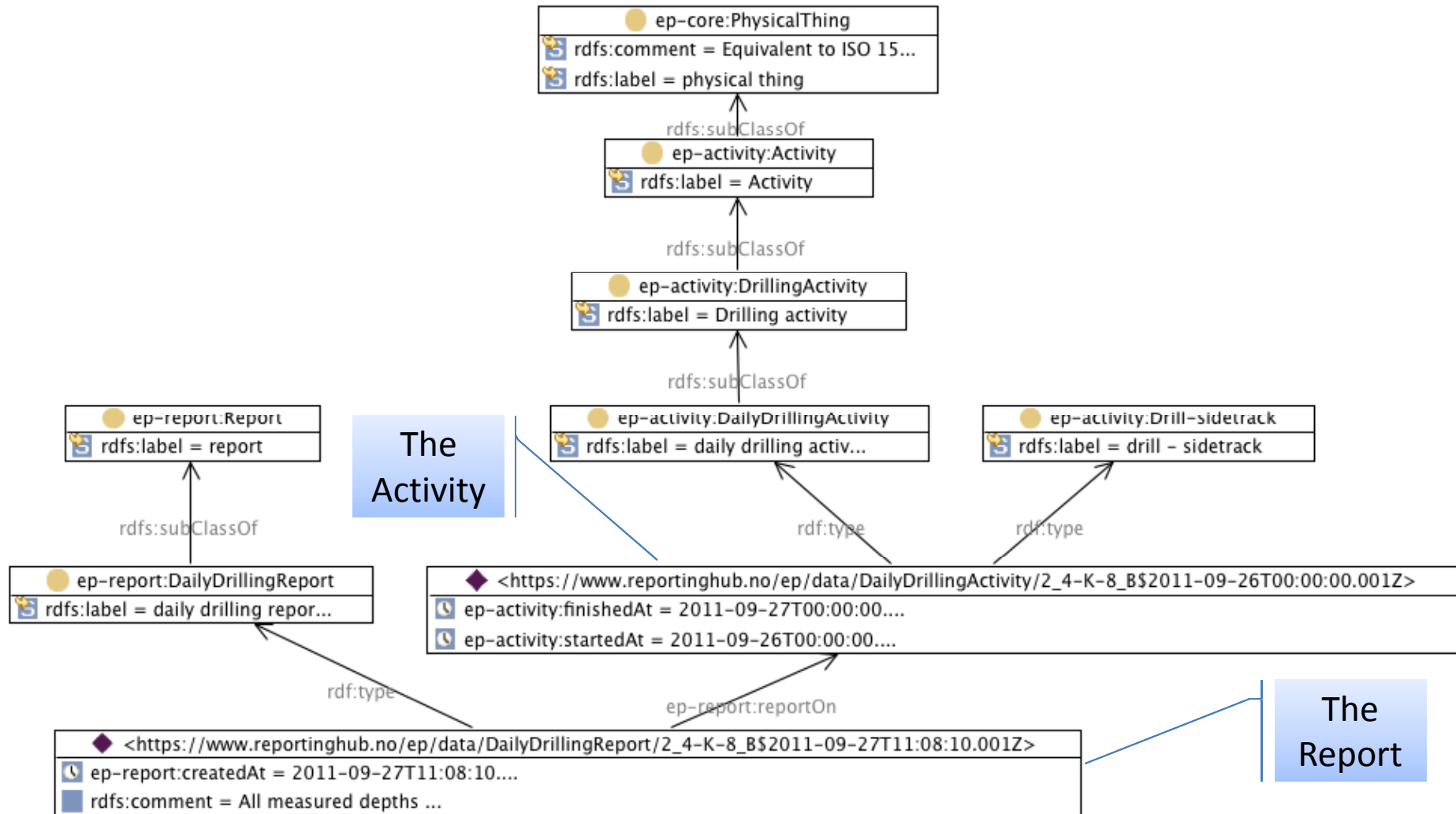


# Report – Graph View





# Report-On a DailyDrillingActivity





# Report - on Wellbore

The Whole Life Wellbore (NPD Fact)

◆ npdi:WellBore-6609
🕒 ep-core:name = 2/4-K-8 B
🕒 ep-well:wellBorePreSpudAt = 1982-06-06T13:15:00...
🕒 ep-well:wellBoreSpudAt = 1986-06-06T13:15:00...
🔖 npd:id = 6609
🔖 npd:name = 2/4-K-8 B
🕒 npd:startDate = 2011-09-23
🕒 npd:wellBoreCompletionDate = 2011-11-23
🔖 npd:wellBoreDrillPermit = 3363-P
🔖 npd:wellBoreType = DEVELOPMENT

● ep-well:WellBore
🔖 rdfs:label = wellbore (whole life...)

● ep-well:TemporalPartOfAWellBore
🔖 rdfs:label = (temporal part of a)...

● ep-core:PhysicalObject
🔖 rdfs:label = physical object

The Temporal Part of the Wellbore

◆ <https://www.reportinghub.no/ep/data/WellBore/2\_4-K-8\_B\$2011-09-26T00:00:00.001Z\$2011-09-27T00:00:00.001Z>

● ep-activity:DailyDrillingActivity
🔖 rdfs:label = daily drilling activ...

◆ <https://www.reportinghub.no/ep/data/DailyDrillingActivity/2_4-K-8_B\$2011-09-26T00:00:00.001Z>
🕒 ep-activity:finishedAt = 2011-09-27T00:00:00....
🕒 ep-activity:startedAt = 2011-09-26T00:00:00....

The Activity





# Wellbore – part of Well – part of Field

The Whole Life Field (NPD Fact)

● ep-fac:Field
🔗 rdfs:comment = facility that is a f...
🔗 rdfs:label = field

● ep-well:TemporalPartOfAWell
🔗 rdfs:label = (temporal part of a)...

◆ npdi:Field-43506
🔗 npd:currentActivityStatus = PRODUCING
🔗 npd:id = 43506
🔗 npd:name = EKOFISK

● ep-well:Well
🔗 rdfs:label = well (whole life)

● ep-well:WellBore
🔗 rdfs:label = wellbore (whole life...)

◆ npdi:Well-2_4-K-8
🔗 ep-core:name = 2/4-K-8
🔗 npd:name = 2/4-K-8

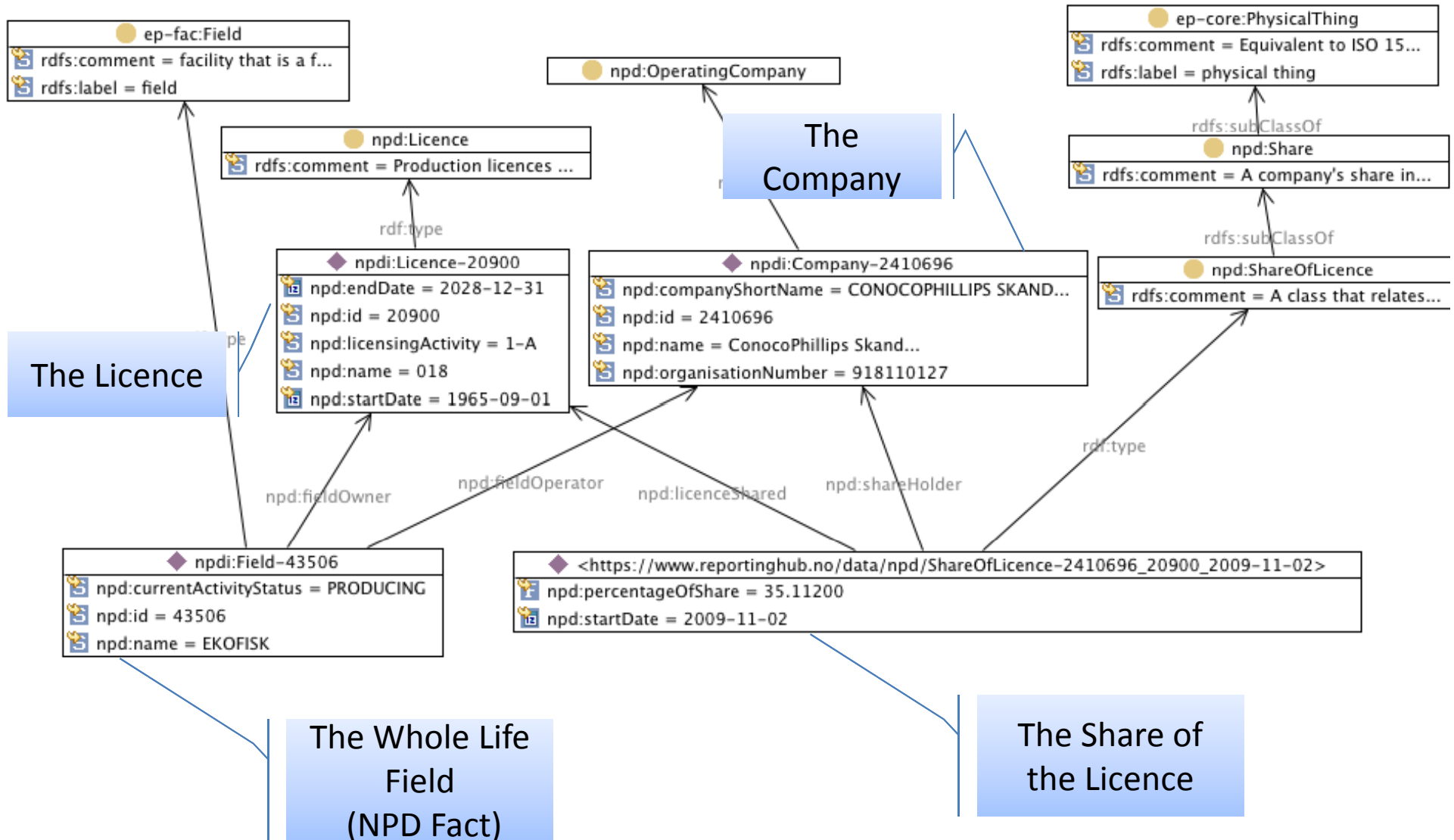
The Whole Life Well (NPD Fact)

The Whole Life Wellbore (NPD Fact)

◆ npdi:WellBore-6609
🔗 ep-core:name = 2/4-K-8 B
🕒 ep-well:wellBorePreSpudAt = 1982-06-06T13:15:00....
🕒 ep-well:wellBoreSpudAt = 1986-06-06T13:15:00....
🔗 npd:id = 6609
🔗 npd:name = 2/4-K-8 B
📅 npd:startDate = 2011-09-23
📅 npd:wellBoreCompletionDate = 2011-11-23
🔗 npd:wellBoreDrillPermit = 3363-P
🔗 npd:wellBoreType = DEVELOPMENT



# Field owner is License has Share Owners





# Partitioning Activity

- ❑ There are subclasses of OWL class used to partition the subclasses of Activity, for example
  - Activity Success Type
  - Drilling Activity Outcome Type
- ❑ See next slide for complete list including membership count
  
- ❑ This kind of thing is an example of where aspects of ISO 15926 are a modeling language in the same way OWL is a modeling language, and why blending the two is an interesting problem

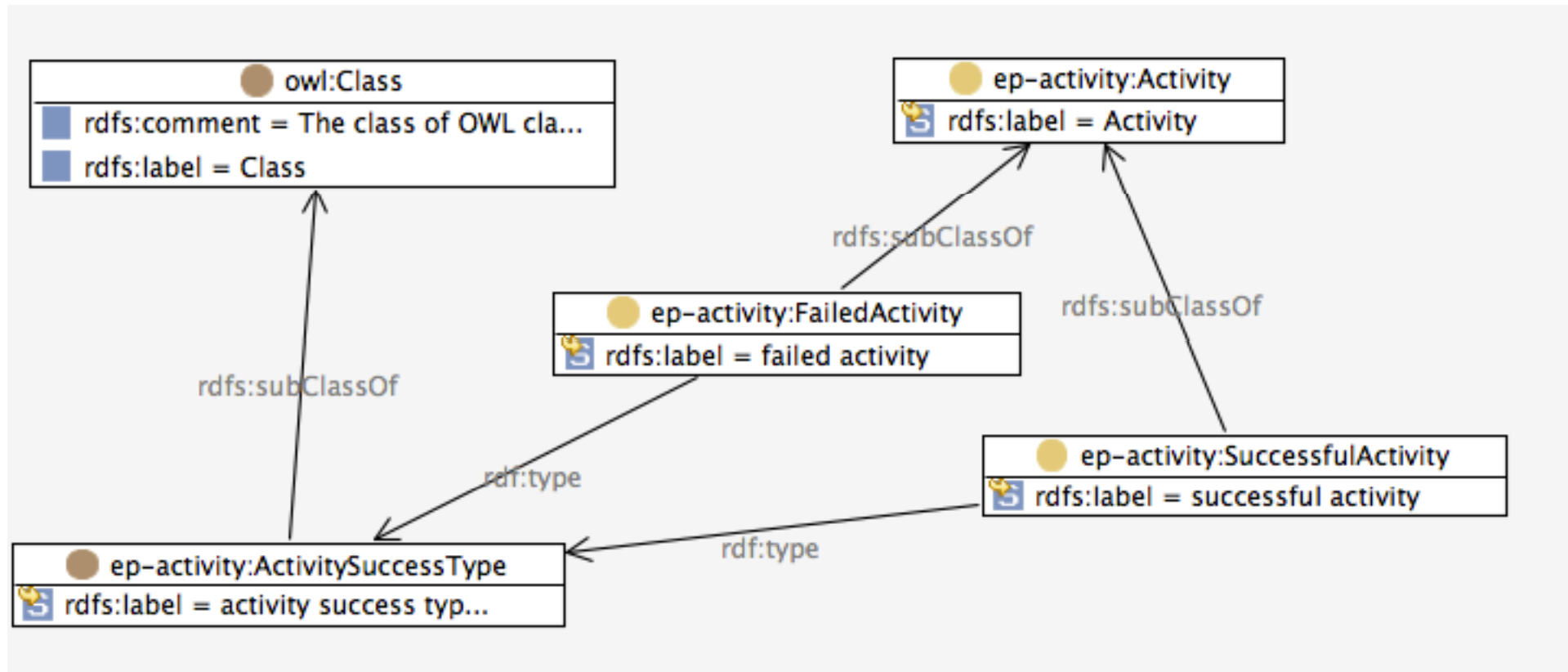


# Examples of the Partitions – aka Class Of Class

- owl:Class (1280)
- ep-activity:ActivitySuccessType (2)
- ➔ ep-activity:DrillingActivityOutcomeType (9)
- ep-activity:DrillingActivityTypeAccordingToEquipmentType (5)
- ep-activity:GasReadingType (9)
- ep-activity:PorePressureDeterminationType (2)
- ep-activity:PressureTestType (2)
- ep-activity:WellBoreDrillingType (6)
- ep-activity:WellCementJobBottomPlugType (2)
- ep-activity:WellCementJobCasingReciprocationType (2)
- ep-activity:WellCementJobCasingRotationType (2)
- ep-activity:WellCementJobFloatHoldingType (2)
- ep-activity:WellCementJobPlugBumpingType (2)
- ep-activity:WellCementJobPurposeType (4)
- ep-activity:WellCementJobTopPlugType (2)
- ep-activity:WellCementPumpingFluidReturnType (2)
- ep-activity:WellControlIncidentType (4)
- ep-activity:WellKillingType (6)
- ep-activity:WellTestType (2)
- ep-activity:WirelineFormationTestType (2)
- ep-actpurp:DailyDrillingActivityPurposeType (83)

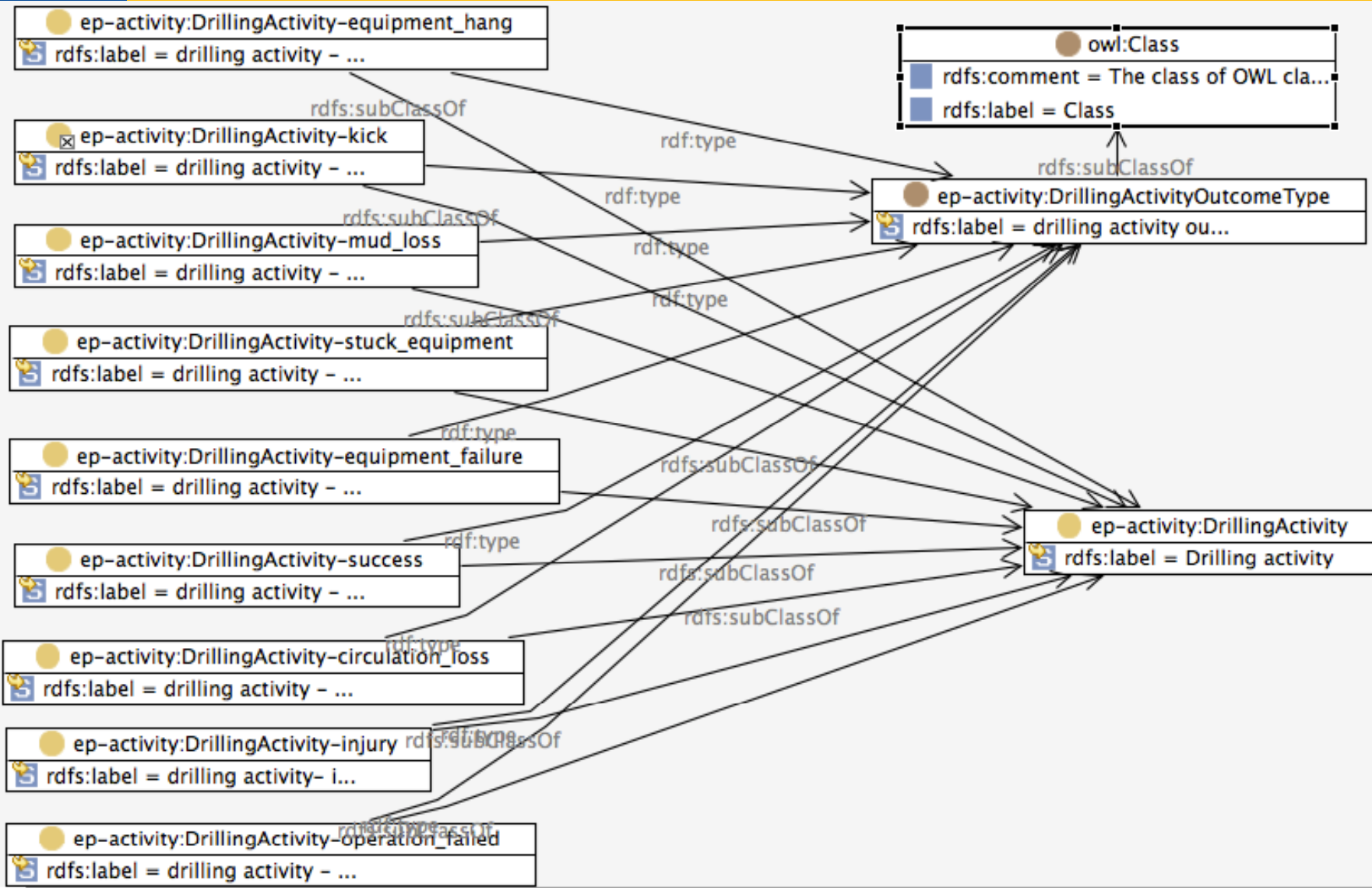


# Activity Success Type





# Drilling Activity Outcome Type



# Properties

TopQuadrant™

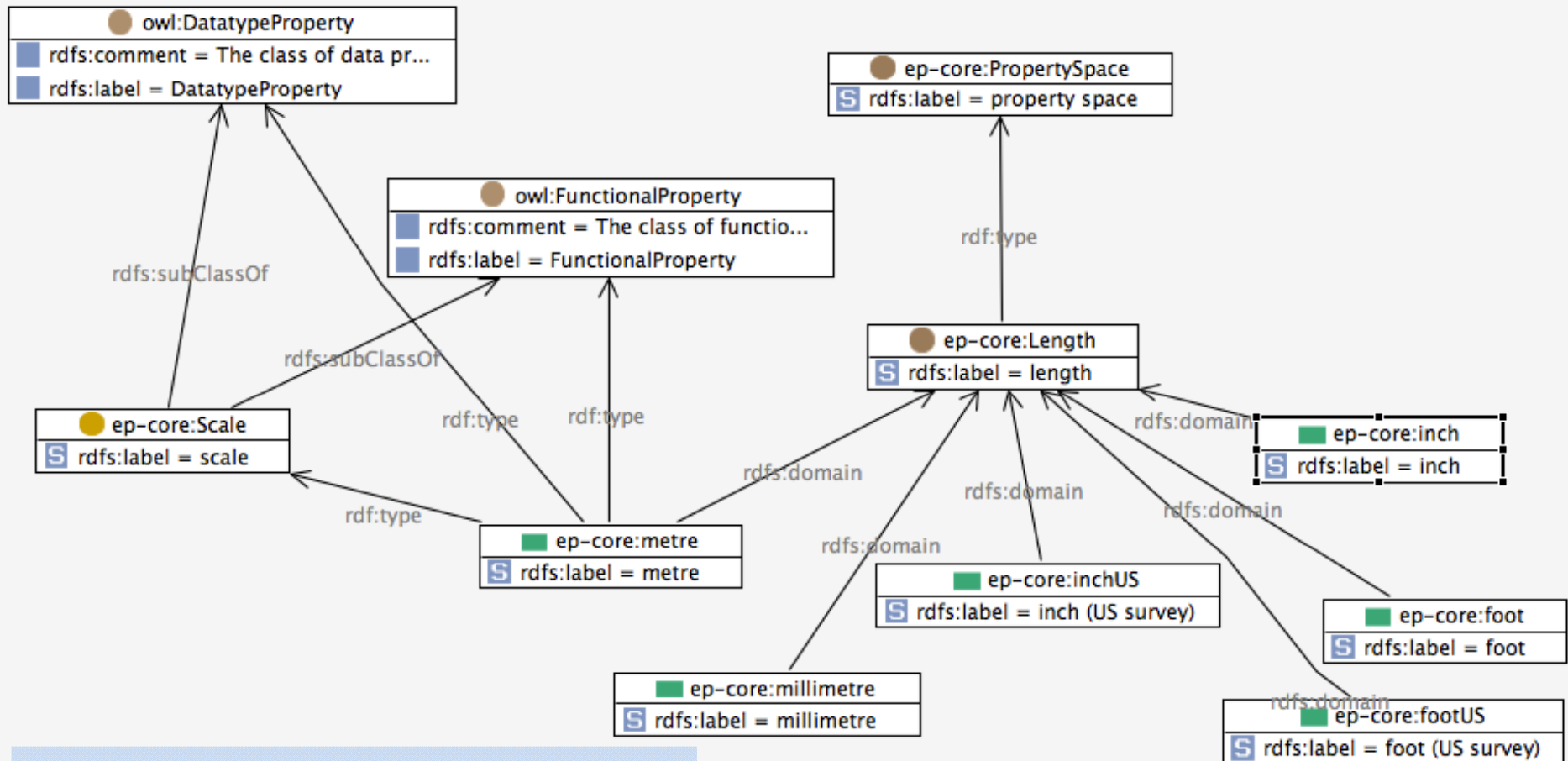


© Copyright 2012 TopQuadrant

Slide 39



# Length Example



Note Scale subClassOf  
OWL itself





# Members of Property Space

- ep-core:Density
- ep-core:Length
- ep-core:MassPerUnitLength
- ep-core:MassPerUnitMassConcentration
- ep-core:MeasureOfAmountOfMaterial
- ep-core:PlaneAngle
- ep-core:PlasticViscosity
- ep-core:Pressure
- ep-core:Ratio
- ep-core:SpecificVolume
- ep-core:Speed
- ep-core:StrengthOfRockFormation
- ep-core:Temperature
- ep-core:TimeDuration
- ep-core:Volume
- ep-core:VolumeAt15degC1atm
- ep-core:VolumeAt1degC1atm
- ep-core:VolumeAt60degF1atm
- ep-core:VolumeMassRatio
- ep-core:VolumePerUnitVolumeConcentration
- ep-core:VolumetricFlowRate
- ep-core:YieldPoint



# Members of Property Scale

- ep-core:bar
- ep-core:barrel
- ep-core:celsius
- ep-core:centiPoise
- ep-core:day
- ep-core:decimetre3
- ep-core:degreesOfAngle
- ep-core:dyne\_second\_metre-2
- ep-core:foot
- ep-core:foot3
- ep-core:foot3\_sack94-1
- ep-core:footUS
- ep-core:fractionByVolume
- ep-core:gram\_cm-3
- ep-core:hour
- ep-core:inch
- ep-core:inchUS
- ep-core:kilogramforce\_second\_metre-2
- ep-core:megaPa
- ep-core:megaPa\_second
- ep-core:metre
- ep-core:metre3
- ep-core:metre3\_day-1
- ep-core:metre3\_minute-1
- ep-core:metre\_hour-1
- ep-core:milliPascal\_second
- ep-core:millimetre
- ep-core:million\_barrel
- ep-core:million\_foot3
- ep-core:million\_metre3
- ep-core:million\_metre3\_day-1
- ep-core:minute
- ep-core:newton\_second\_metre-2
- ep-core:partPerMillionByMass
- ep-core:partPerMillionByVolume
- ep-core:pascal
- ep-core:pascal\_second
- ep-core:perCentByVolume
- ep-core:poise
- ep-core:pound\_foot-1
- ep-core:poundforce\_second\_foot-2
- ep-core:poundforce\_second\_inch-2
- ep-core:psi\_second
- ep-core:psig
- ep-core:second
- ep-core:specificGravity
- ep-core:undefinedScale
- ep-core:usGallon\_sack94-1
- ep-core:week

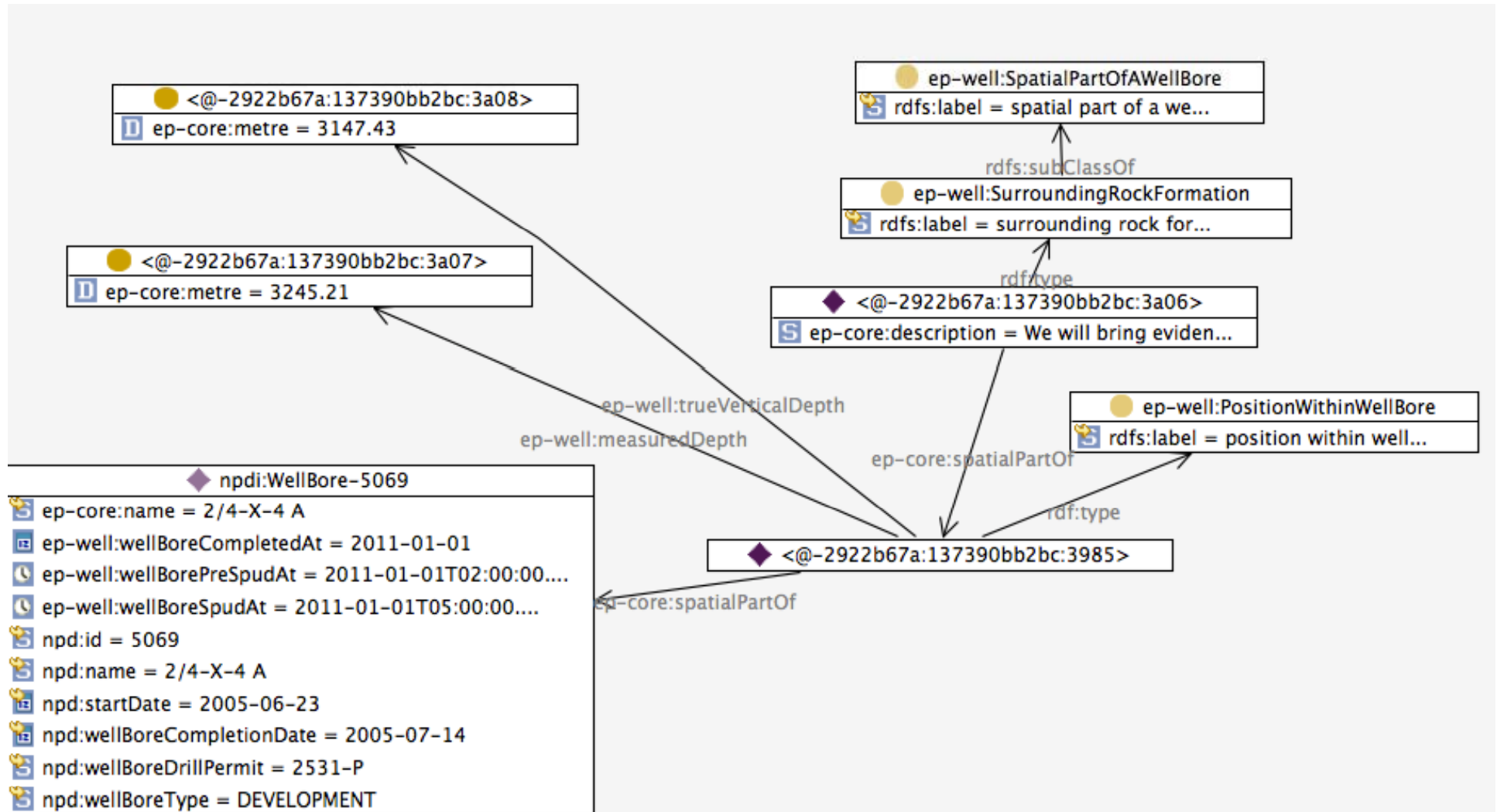


# Direct vs Indirect Properties

- ❑ When a physical object ‘has a property’ without any qualification at all, then it’s a Direct Property
  - <David> rdf:type <mass 39kg>
  - Property as class with members is basic to ISO 15926 4-D approach
- ❑ When a physical object has a property that requires qualification, then it’s an Indirect Property
  - <David> hasHeight <length 1.89m>
- ❑ In the following diagram we distinguish between trueVerticalDepth and measuredDepth



# Property – Description of Surrounding Rock at Depth





# Experiences

- ❑ The overall approach and architecture worked
  - Including isolating semantics into modeling team so Java developers don't have to learn 4 ontologies and related transformations
- ❑ We've built a very flexible, semantically solid asset management tool
  - This experience and knowledge can be applied elsewhere
- ❑ EPIM is as much a team member as a customer in working through some very difficult issues



# In Conclusion

- ❑ EPIM ReportingHub first stages are *live*
  - Final stages delivered over next couple of months
- ❑ Phase 1 has been building infrastructure and core capability ...
- ❑ ... now getting value from semantics can begin
  - Historical data?
  - Analytics and data visualizations?
  - Automated notification of potential problems based on data analysis?
  - Extractions/integrations in operators systems?