Making BioPAX SPARQL

hands on ...

- start a terminal
- create a directory jena_workspace, move into that directory
- download jena.jar (http://tinyurl.com/**3vlp7rw**)
- download biopax data (http://www.biopax.org/Junk/Homosapiens.nt or a smaller file (http://www.biopax.org/Junk/ Escherichia_coli.nt)

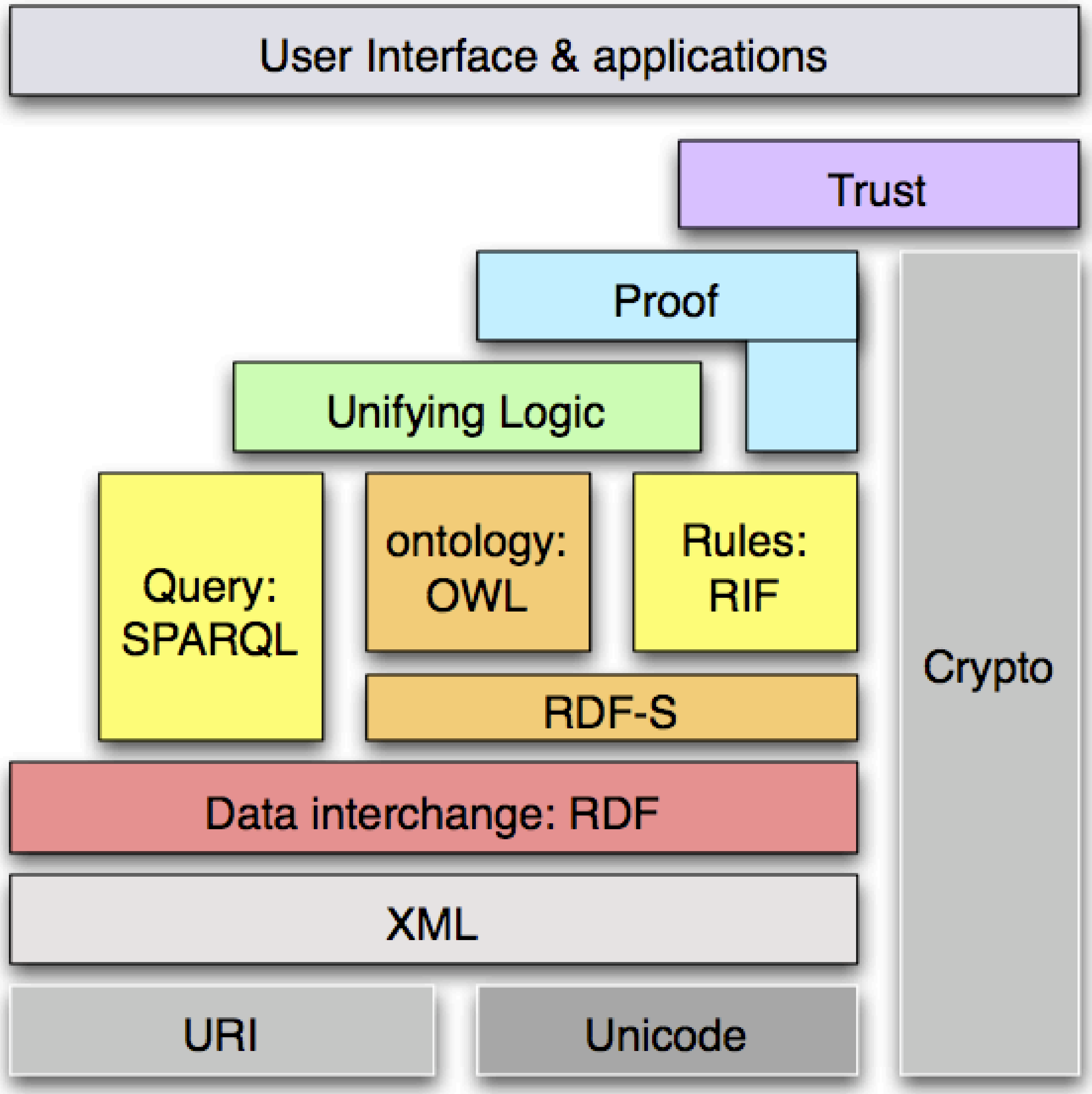
- Andrea on hand to help....

  //sourceforge.net/apps/mediawiki/biopax/index.php?title=BioPAX/OWL_and_SPARQL#Tutorial_Material
  or
  //sourceforge.net/apps/mediawiki/biopax/index.php?title=Harmony2011

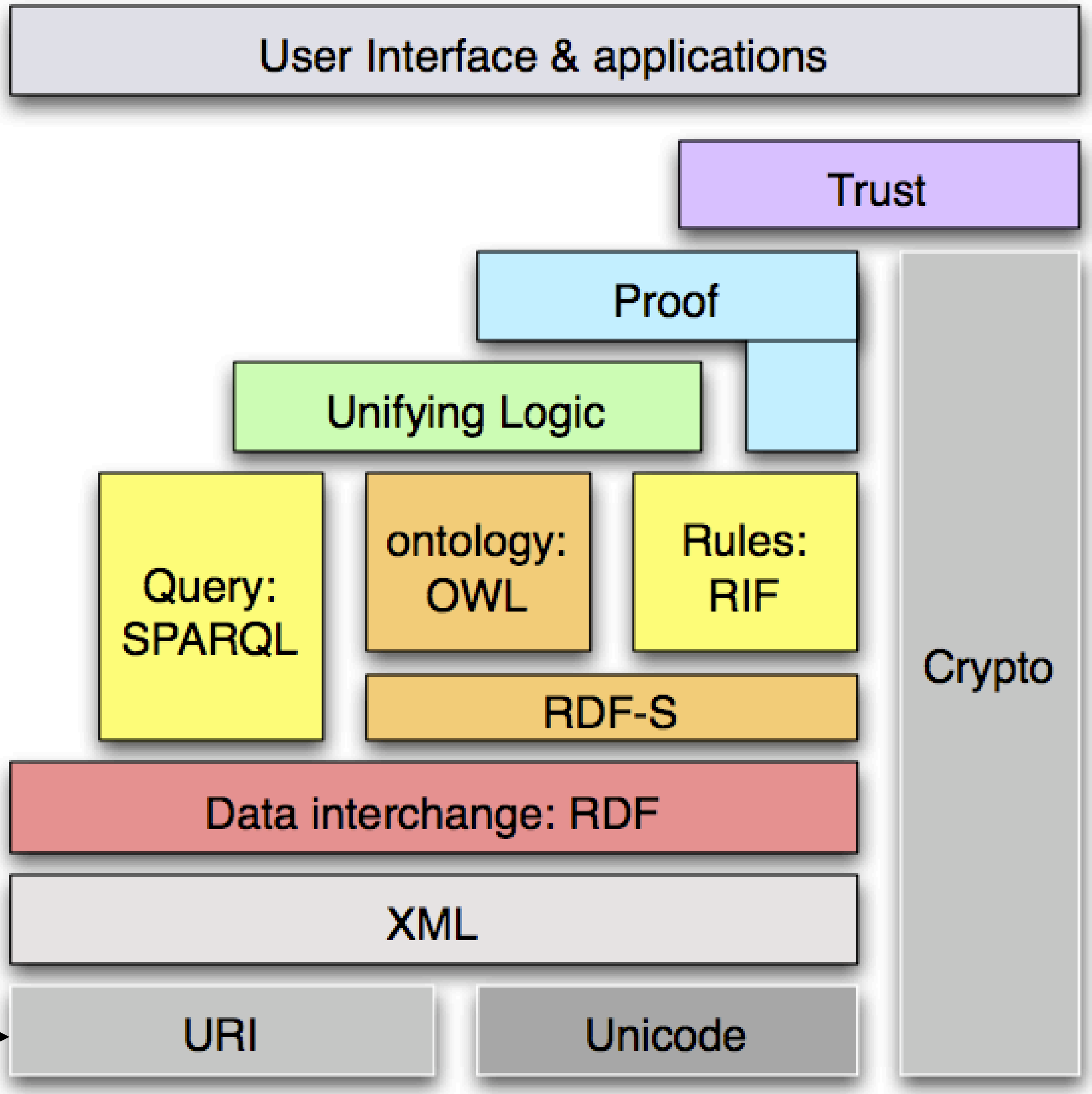
Coming up...

- Semantic Web Stack
- RDF URI based integration
- RDFS semantic integration
- OWL semantic integration
- BioPAX
- SPARQL
- Hands on
- Questions/Follow up

Can BioPAX SPARQL?

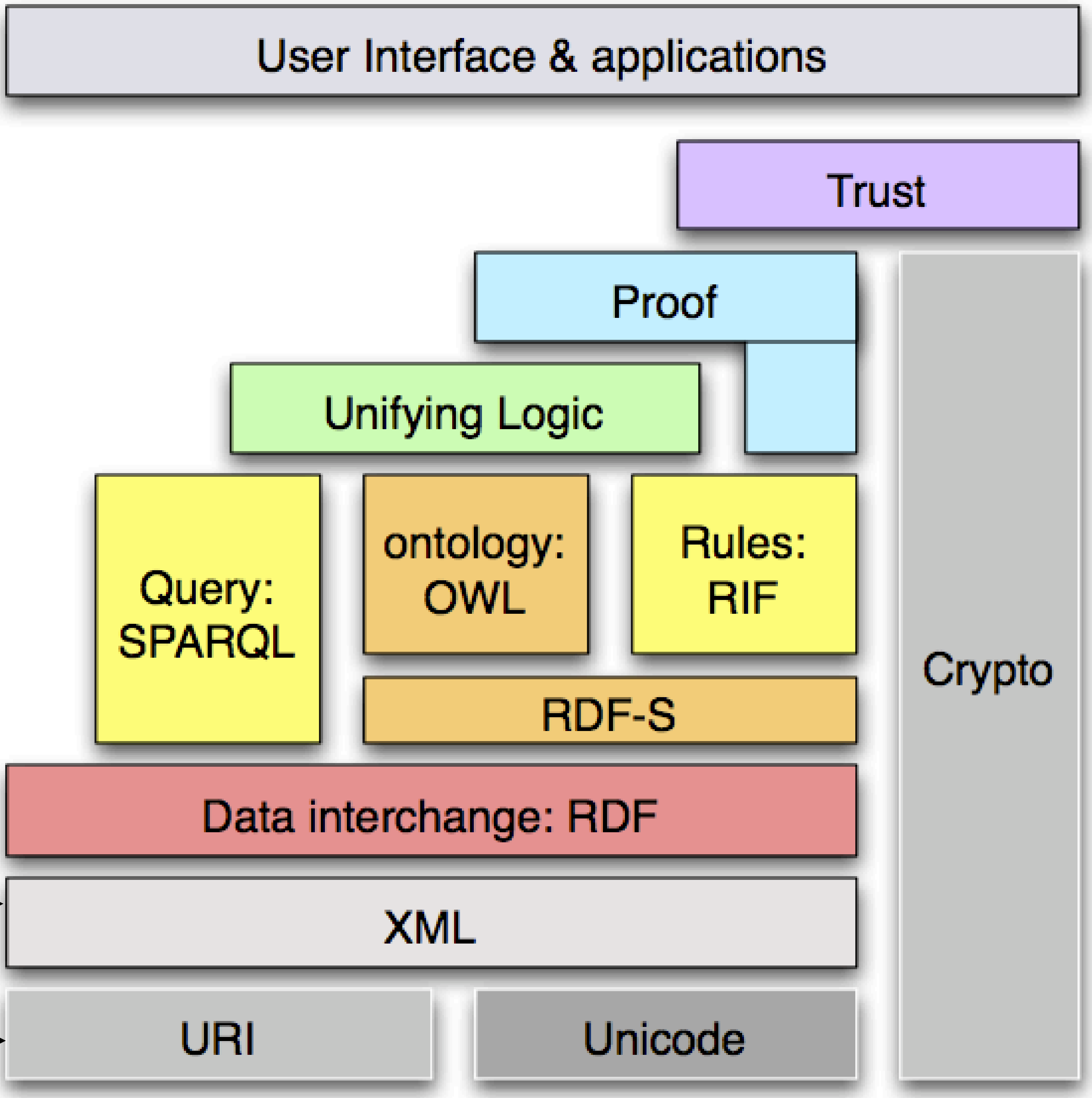


Can BioPAX SPARQL?



A global naming scheme

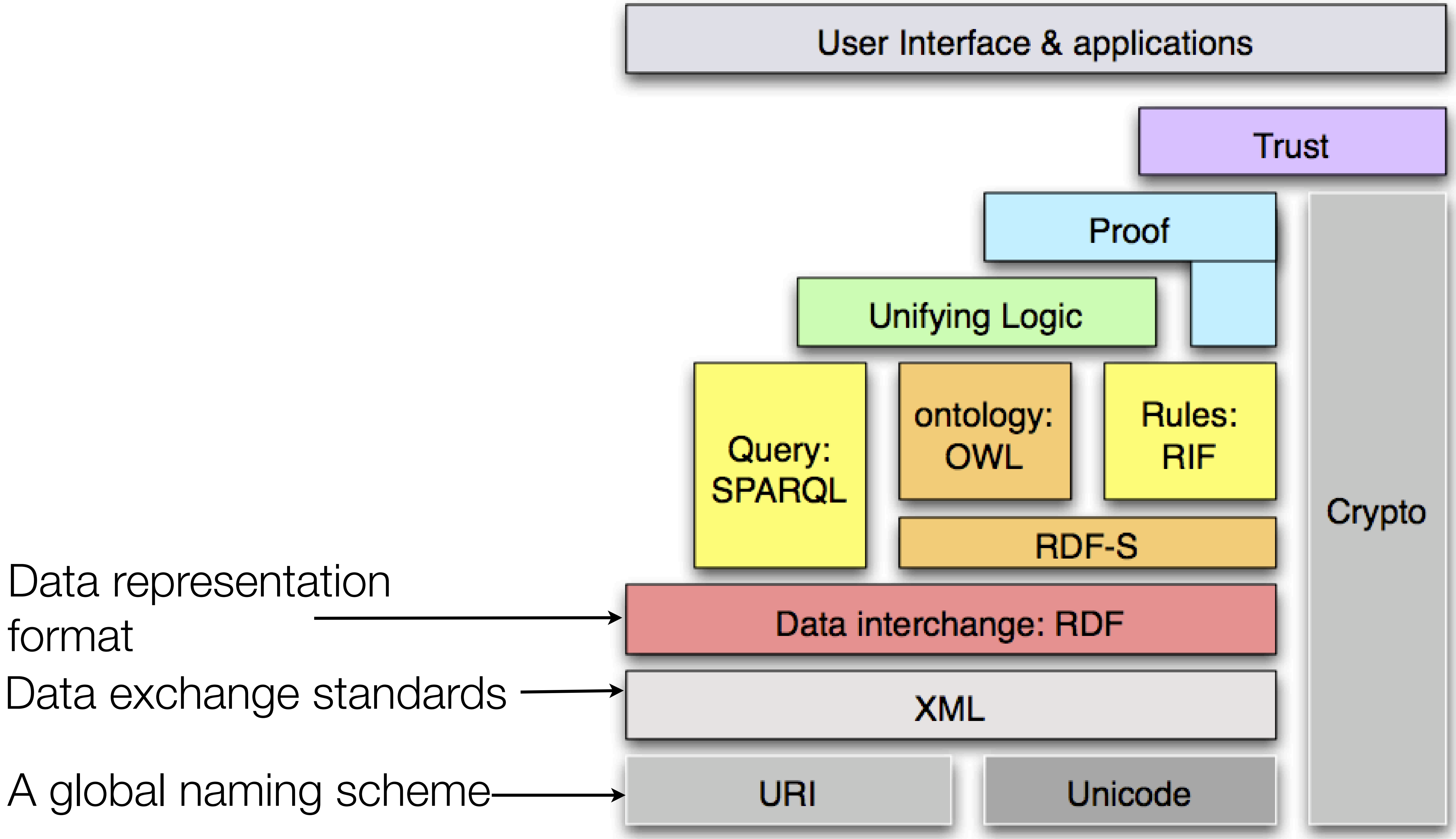
Can BioPAX SPARQL?



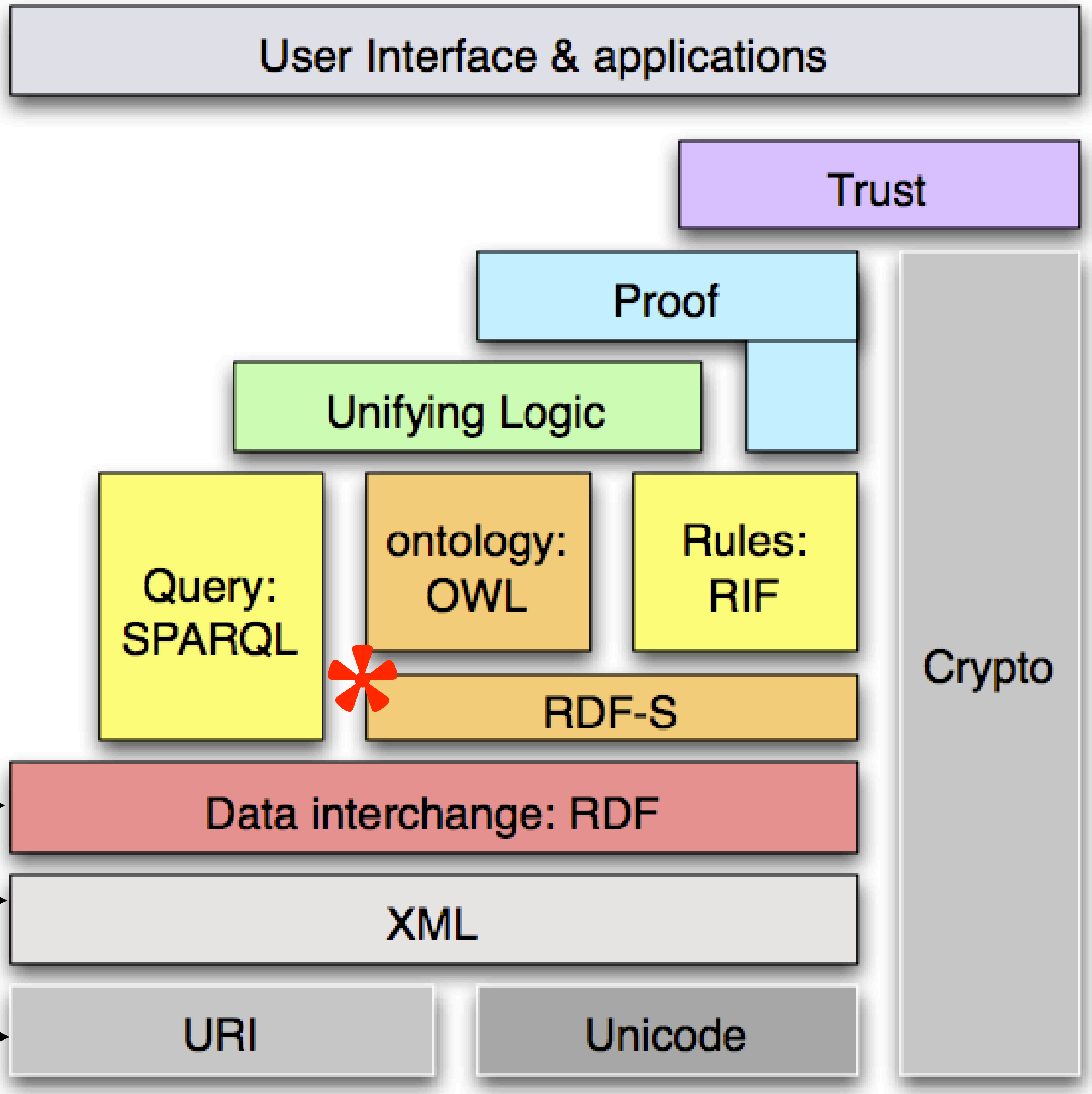
Data exchange standards →

A global naming scheme →

Can BioPAX SPARQL?



Can BioPAX SPARQL?

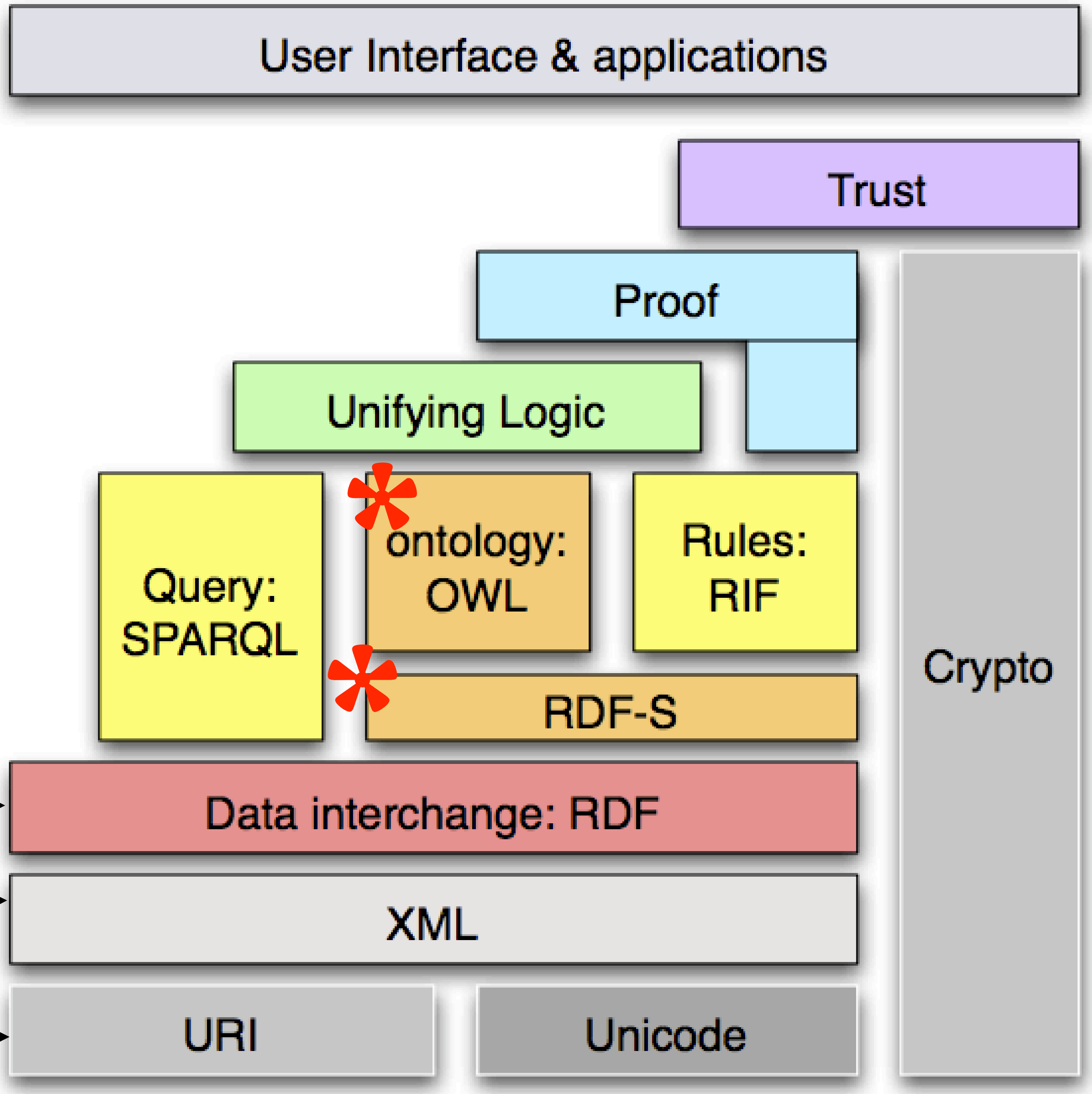


Data representation format →

Data exchange standards →

A global naming scheme →

Can BioPAX SPARQL?

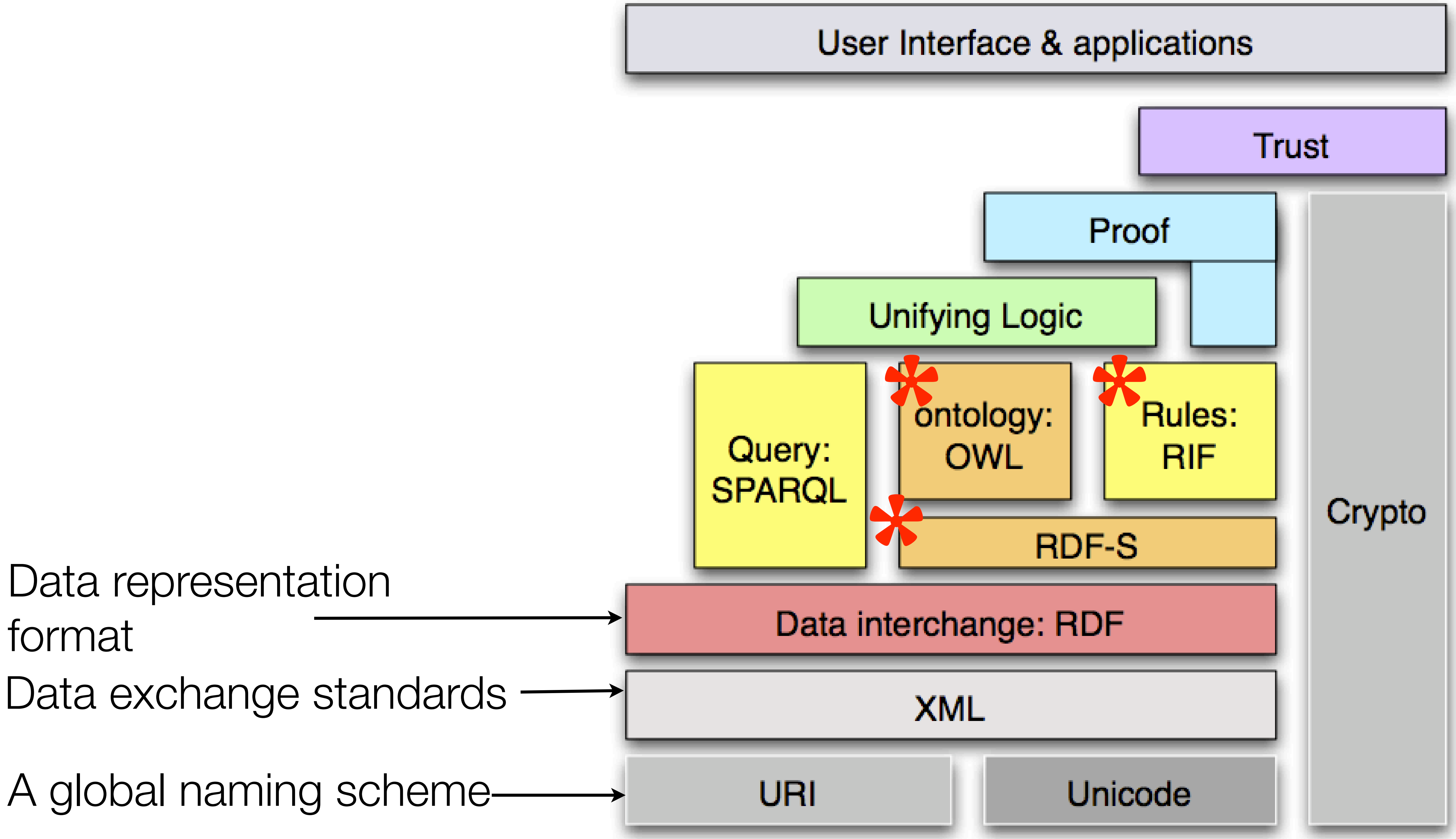


Data representation format →

Data exchange standards →

A global naming scheme →

Can BioPAX SPARQL?



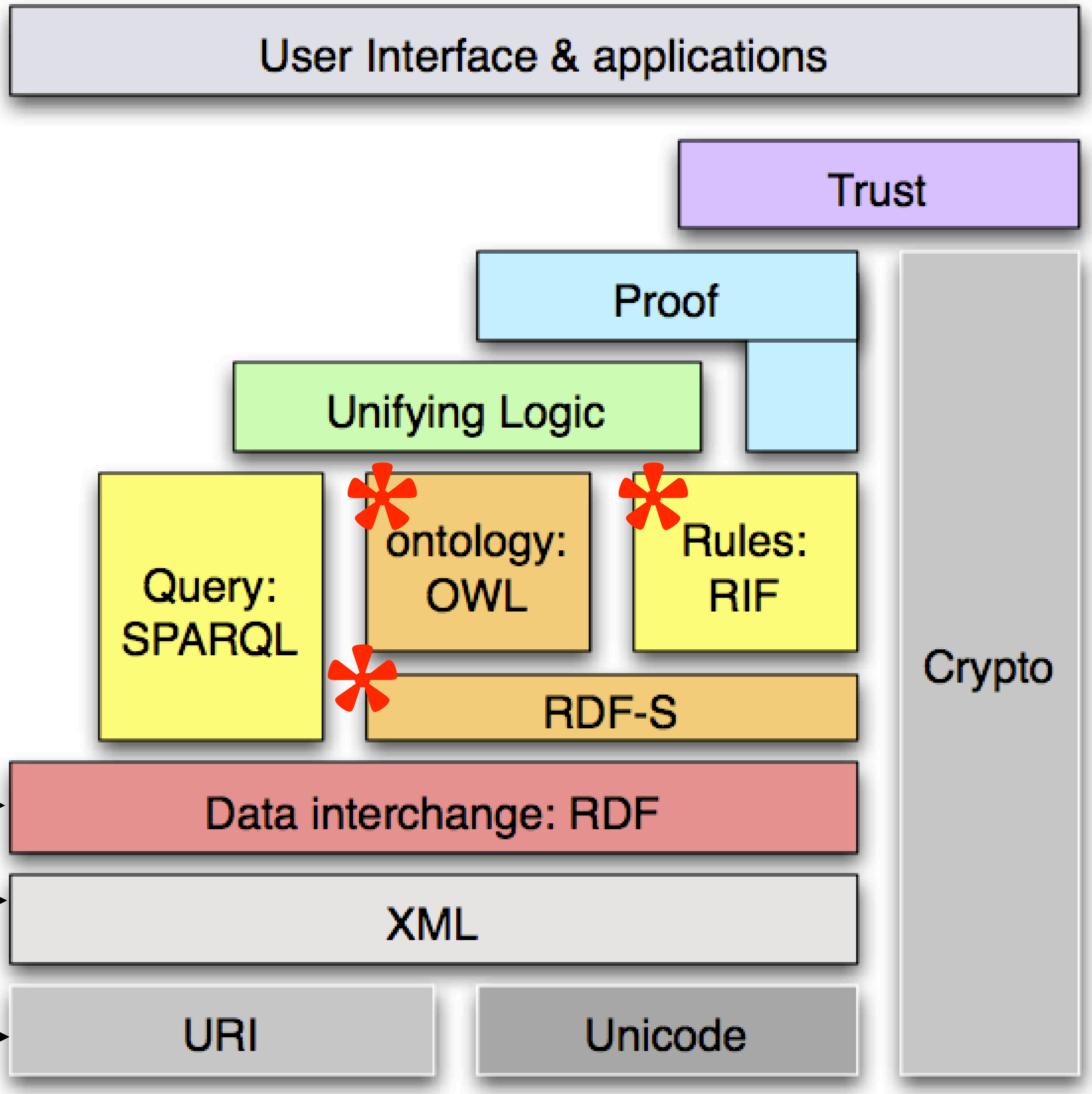
Can BioPAX SPARQL?

* Queries

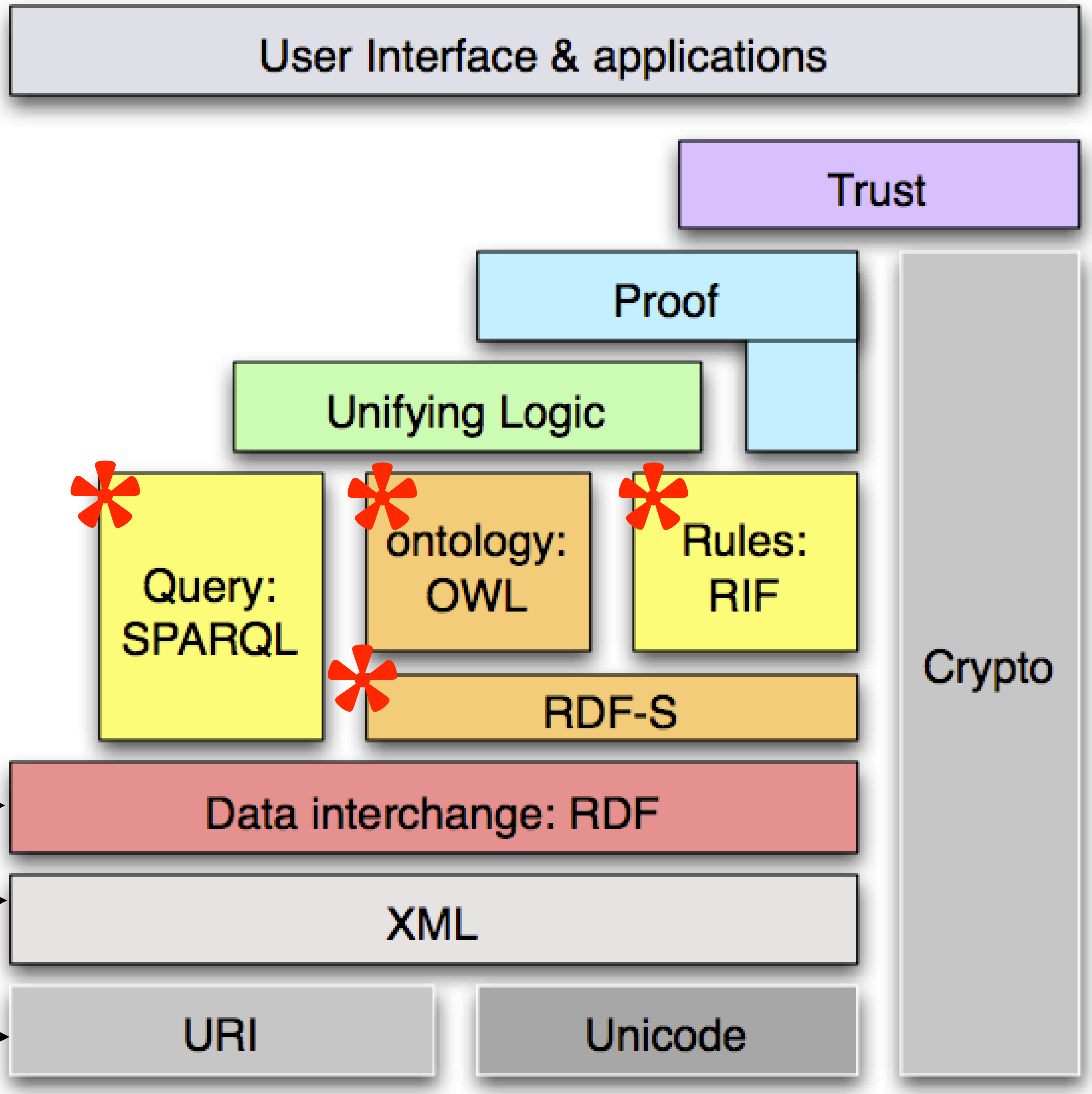
Data representation format

Data exchange standards

A global naming scheme



Can BioPAX SPARQL?



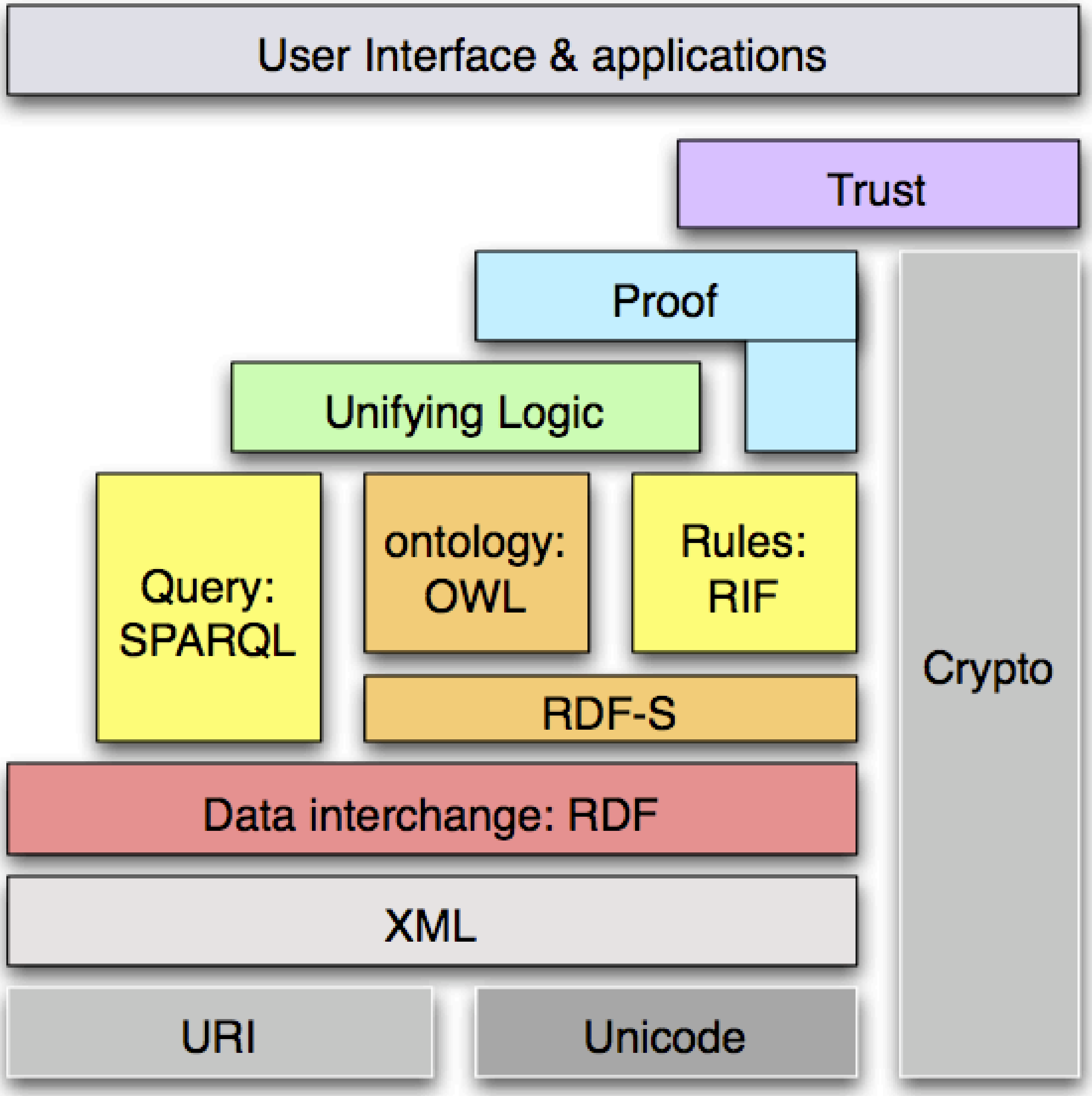
* Queries

Data representation format

Data exchange standards

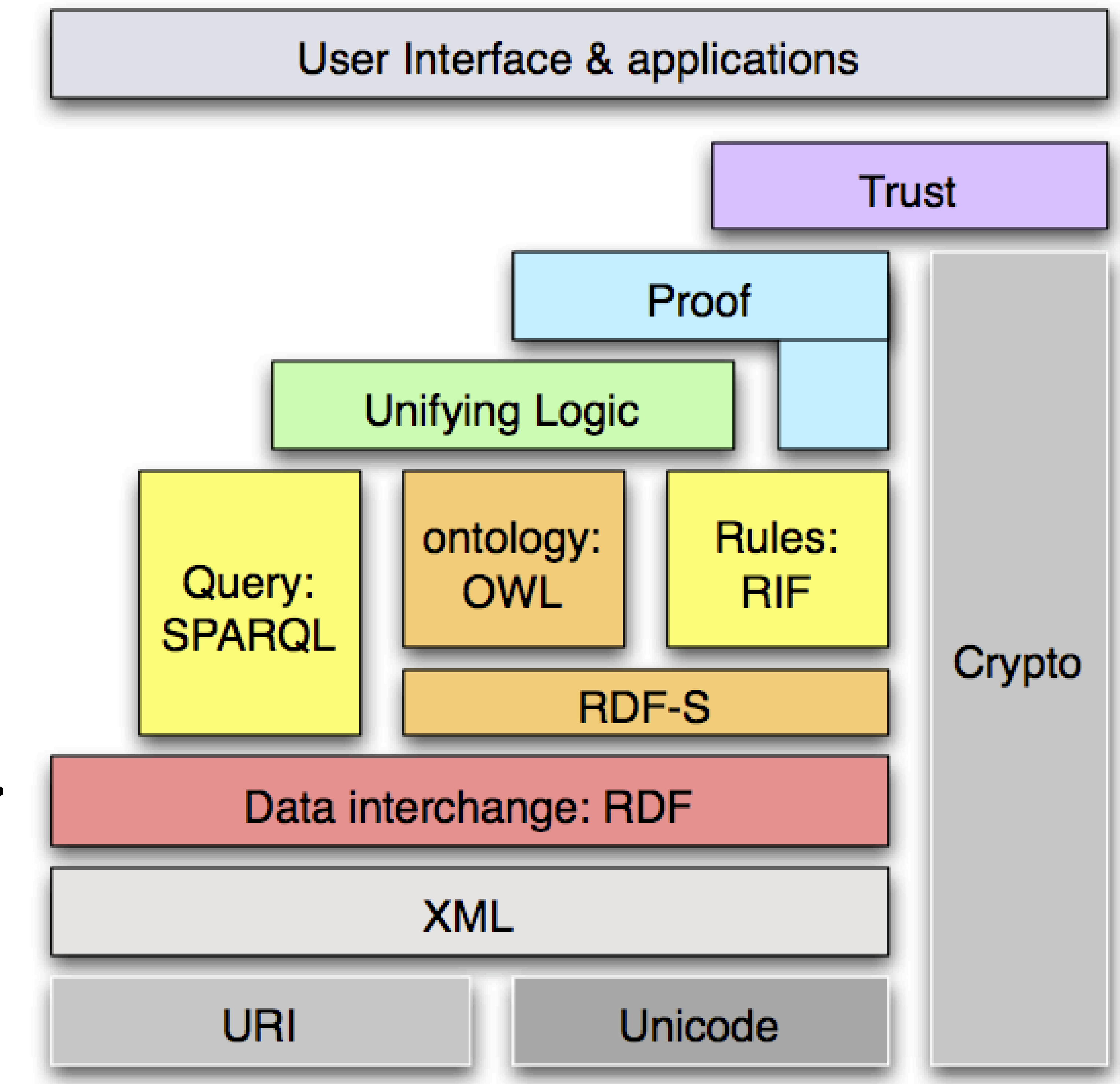
A global naming scheme

Semantic Web

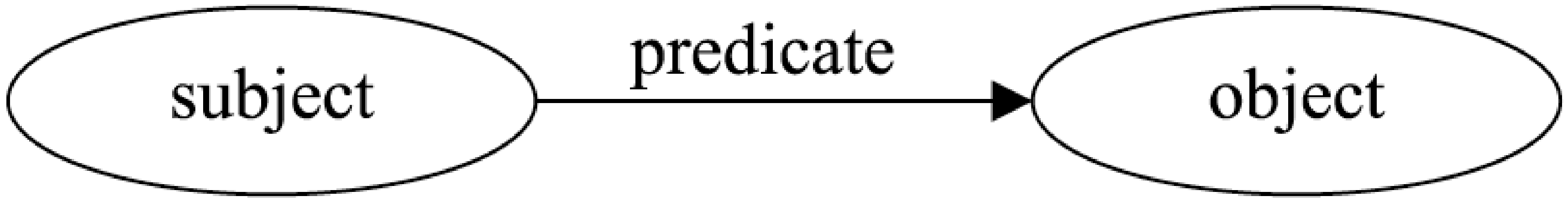


Semantic Web

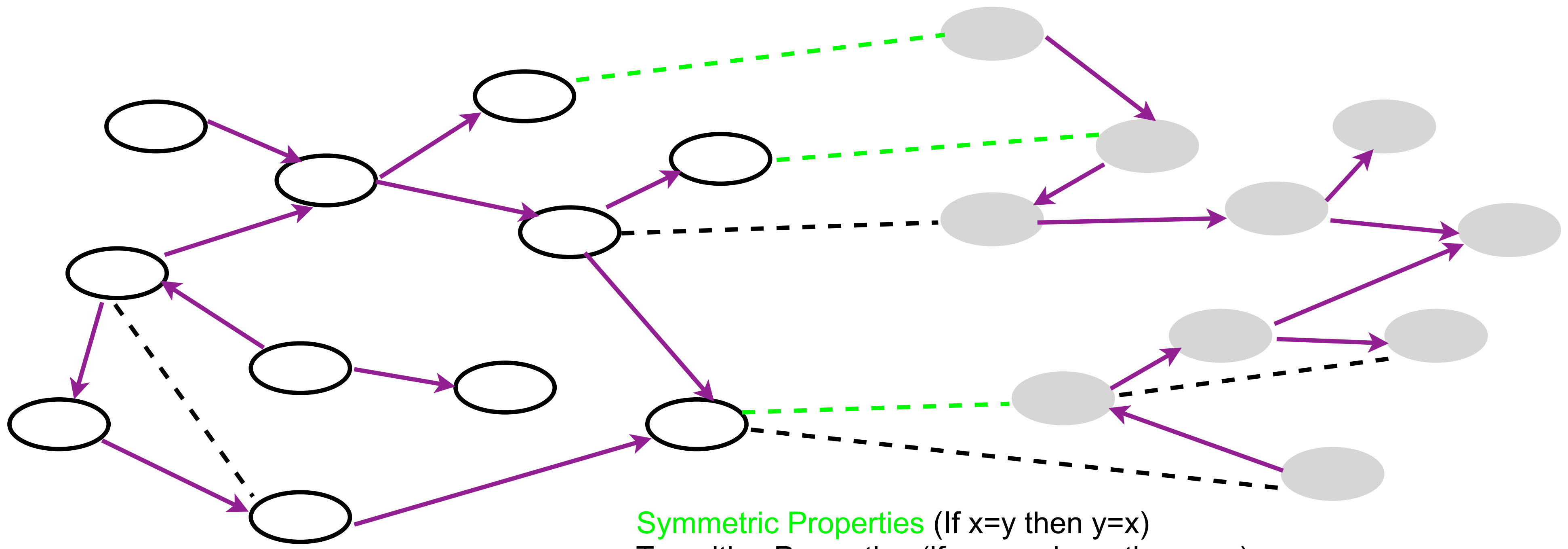
Data representation format



RDF - integration

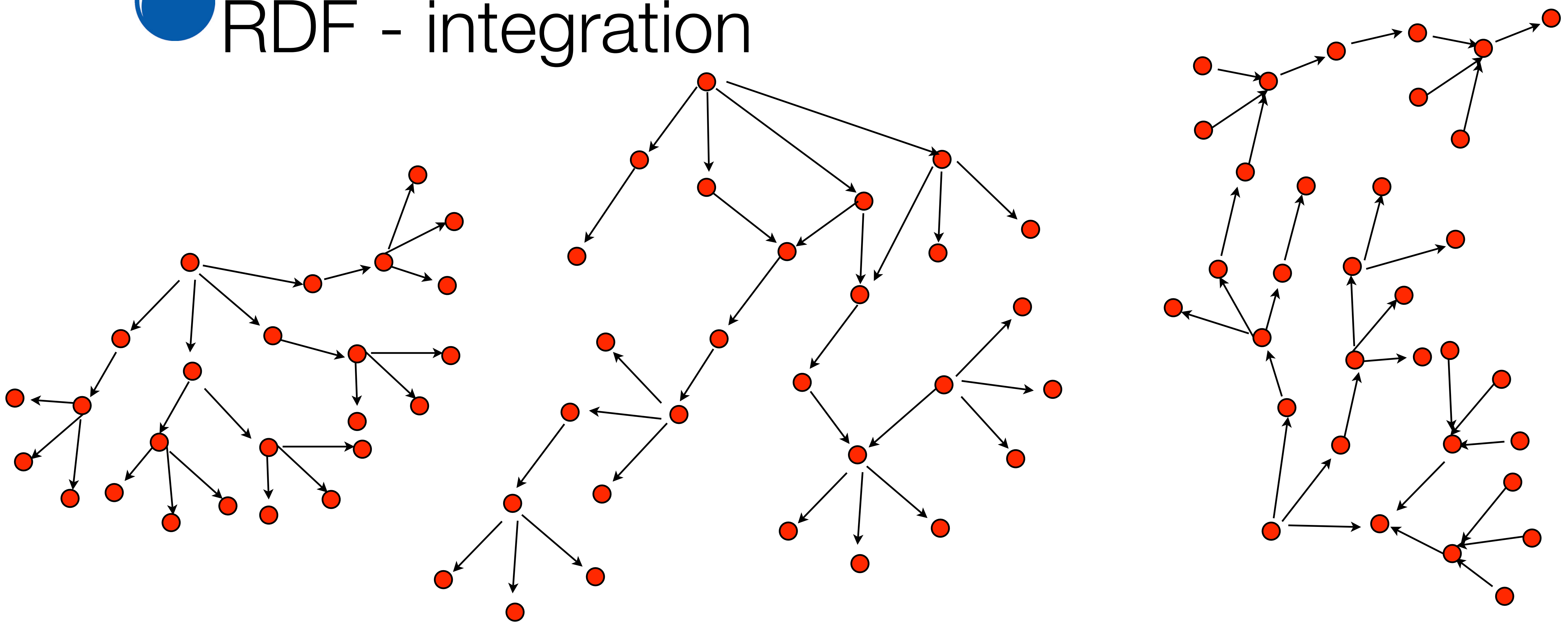


Source: W3C (2004)

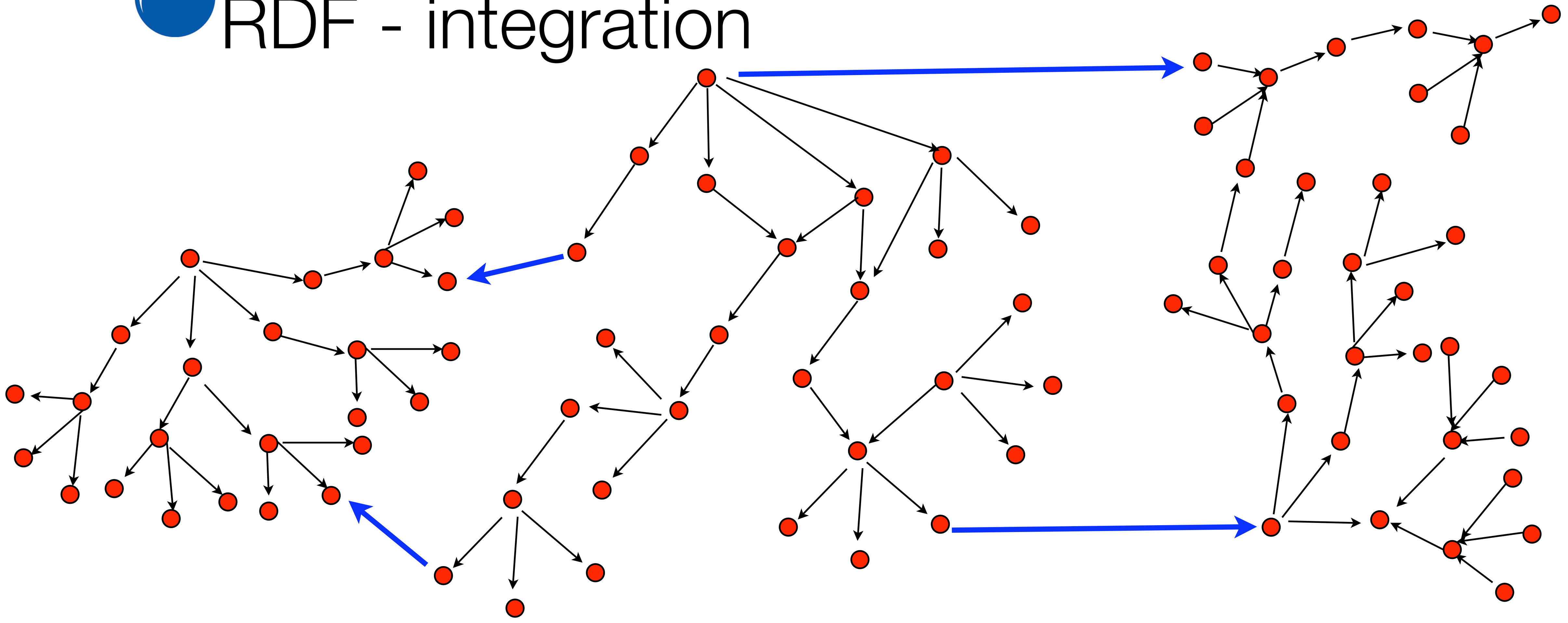


Symmetric Properties (If x=y then y=x)
Transitive Properties (if x=y and y=z then x=z)

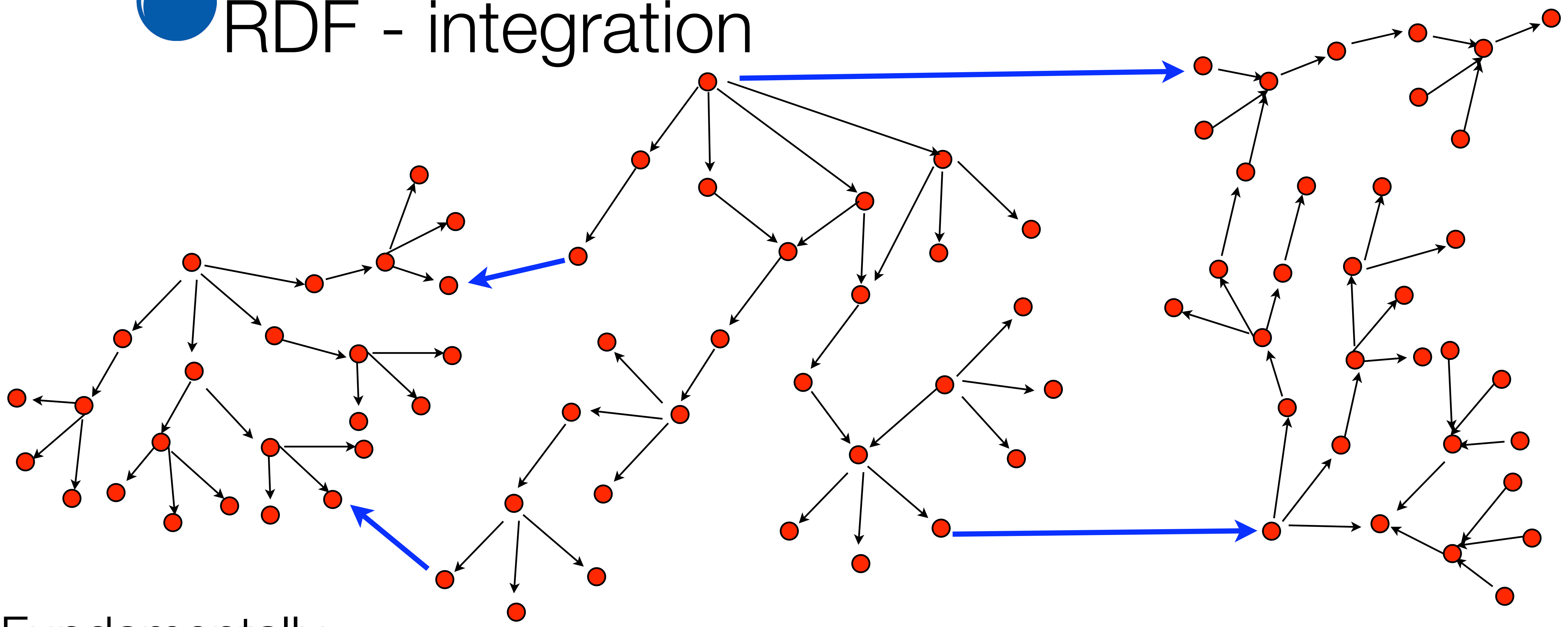
RDF - integration



RDF - integration



RDF - integration

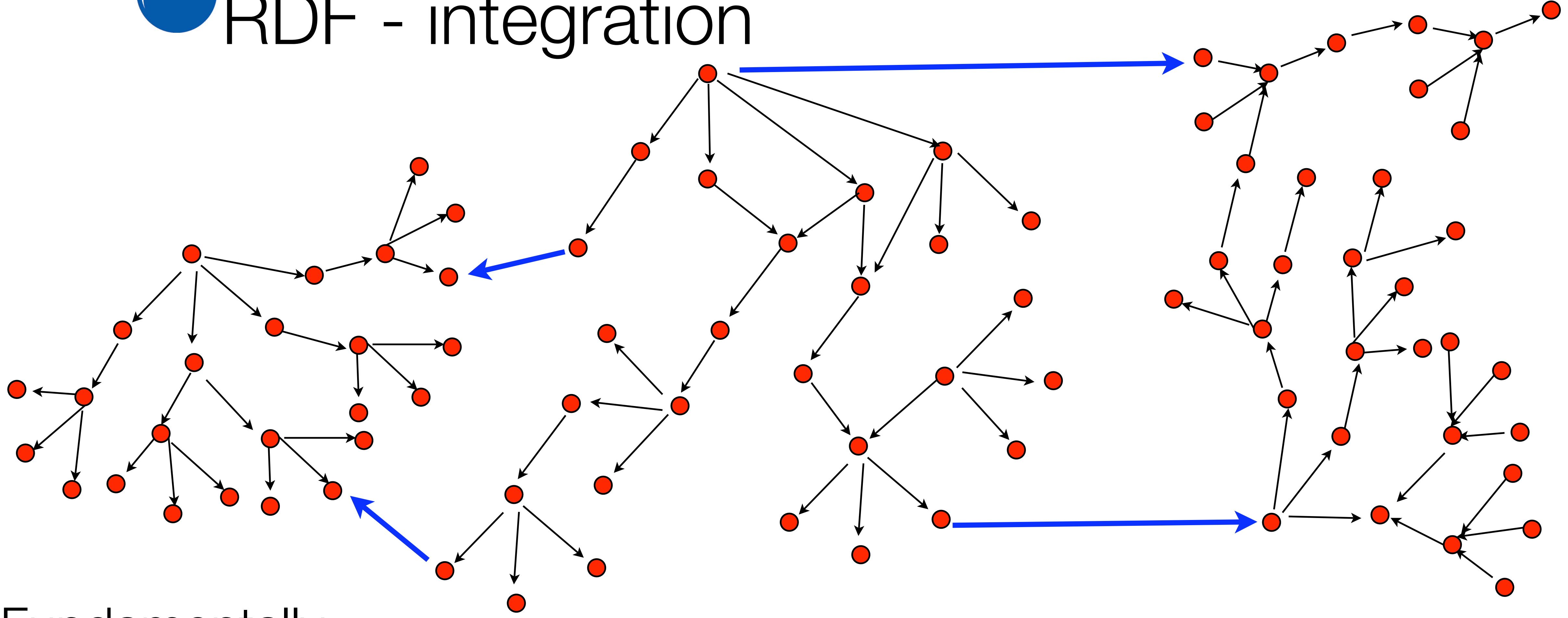


Fundamentally,

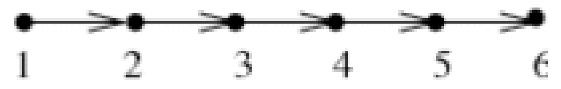
new data can be inferred from existing data by following **logical** rules.

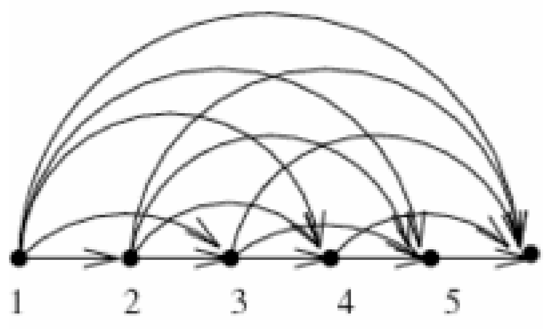
e.g. transitive closure

RDF - integration



Fundamentally,
new data can be inferred from existing data by following **logical** rules.
e.g. transitive closure

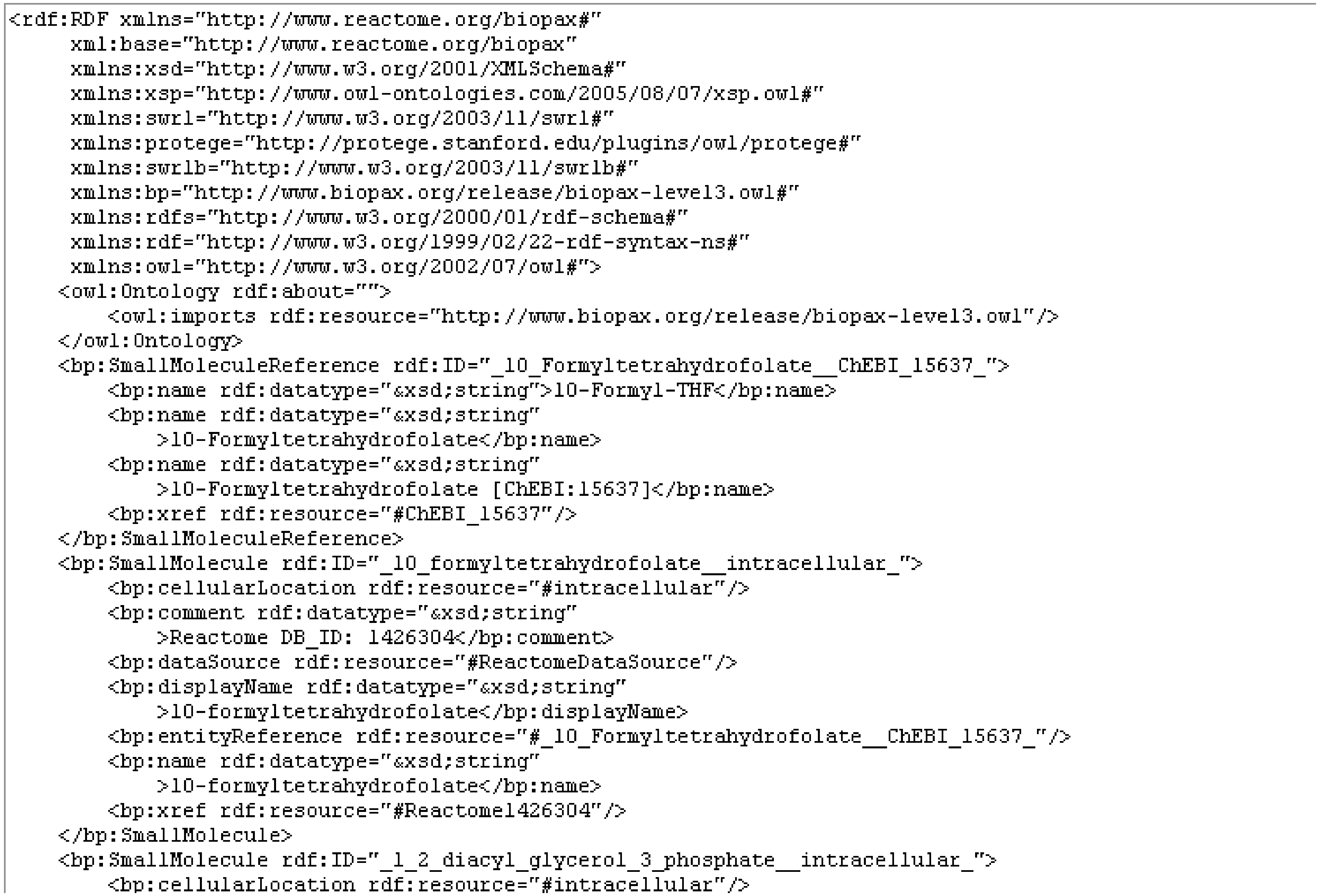




RDF - integration

- Biopax has an XML syntax and ...
    - N-triples
    - How do you query these triples (quick demo)

RDF - XML



```xml
<rdf:RDF xmlns="http://www.reactome.org/biopax#"
     xml:base="http://www.reactome.org/biopax"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
     xmlns:swrl="http://www.w3.org/2003/11/swrl#"
     xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
     xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
     xmlns:bp="http://www.biopax.org/release/biopax-level3.owl#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:owl="http://www.w3.org/2002/07/owl#">
    <owl:Ontology rdf:about="">
        <owl:imports rdf:resource="http://www.biopax.org/release/biopax-level3.owl"/>
    </owl:Ontology>
    <bp:SmallMoleculeReference rdf:ID="_10_Formyltetrahydrofolate__ChEBI_15637_">
        <bp:name rdf:datatype="&xsd;string">10-Formyl-THF</bp:name>
        <bp:name rdf:datatype="&xsd;string"
            >10-Formyltetrahydrofolate</bp:name>
        <bp:name rdf:datatype="&xsd;string"
            >10-Formyltetrahydrofolate [ChEBI:15637]</bp:name>
        <bp:xref rdf:resource="#ChEBI_15637"/>
    </bp:SmallMoleculeReference>
    <bp:SmallMolecule rdf:ID="_10_formyltetrahydrofolate__intracellular_">
        <bp:cellularLocation rdf:resource="#intracellular"/>
        <bp:comment rdf:datatype="&xsd;string"
            >Reactome DB_ID: 1426304</bp:comment>
        <bp:dataSource rdf:resource="#ReactomeDataSource"/>
        <bp:displayName rdf:datatype="&xsd;string"
            >10-formyltetrahydrofolate</bp:displayName>
        <bp:entityReference rdf:resource="#_10_Formyltetrahydrofolate__ChEBI_15637_"/>
        <bp:name rdf:datatype="&xsd;string"
            >10-formyltetrahydrofolate</bp:name>
        <bp:xref rdf:resource="#Reactome1426304"/>
    </bp:SmallMolecule>
    <bp:SmallMolecule rdf:ID="_1_2_diacyl_glycerol_3_phosphate__intracellular_">
        <bp:cellularLocation rdf:resource="#intracellular"/>
```

RDF - Ntriples

```
<http://www.reactome.org/biopax#MATE_substrates__intracellular_>
<http://www.biopax.org/release/biopax-level3.owl#memberPhysicalEntity
<http://www.reactome.org/biopax#Cimetidine__intracellular_> .

<http://www.reactome.org/biopax#MATE_substrates__intracellular_>
<http://www.biopax.org/release/biopax-level3.owl#memberPhysicalEntity
<http://www.reactome.org/biopax#Creatinine__intracellular_> .

<http://www.reactome.org/biopax#MATE_substrates__intracellular_>
<http://www.biopax.org/release/biopax-level3.owl#memberPhysicalEntity
<http://www.reactome.org/biopax#Guanidine__intracellular_> .

<http://www.reactome.org/biopax#MATE_substrates__intracellular_>
<http://www.biopax.org/release/biopax-level3.owl#memberPhysicalEntity
<http://www.reactome.org/biopax#Metformin__intracellular_> .

<http://www.reactome.org/biopax#MATE_substrates__intracellular_>
<http://www.biopax.org/release/biopax-level3.owl#memberPhysicalEntity
<http://www.reactome.org/biopax#Procainamide__intracellular_> .

<http://www.reactome.org/biopax#MATE_substrates__intracellular_>
<http://www.biopax.org/release/biopax-level3.owl#memberPhysicalEntity
<http://www.reactome.org/biopax#Tetramethylammonium__intracellular_>
```

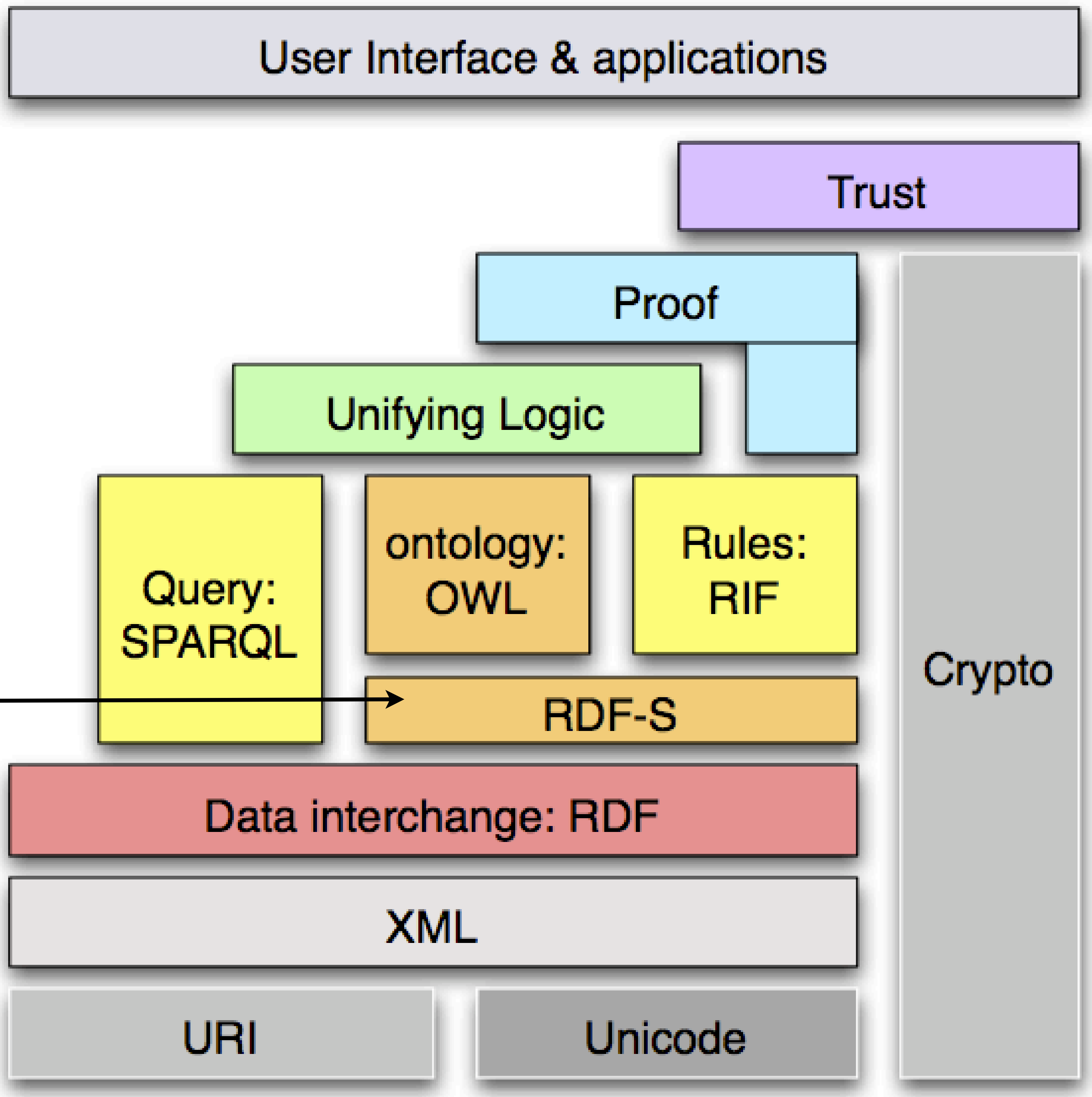
SPARQL - Graph Pattern 1

select all interactions in a pathway that are BiochemicalReactions:

```
SELECT ?name
WHERE
    {
    ?pathway rdf:type bp:Pathway .
    ?pathway bp:pathwayComponent ?c .
    ?c bp:name ?name.
    ?c rdf:type bp:BiochemicalReaction
    }
```

BioPAX uses RDFS

1st layer of Semantics

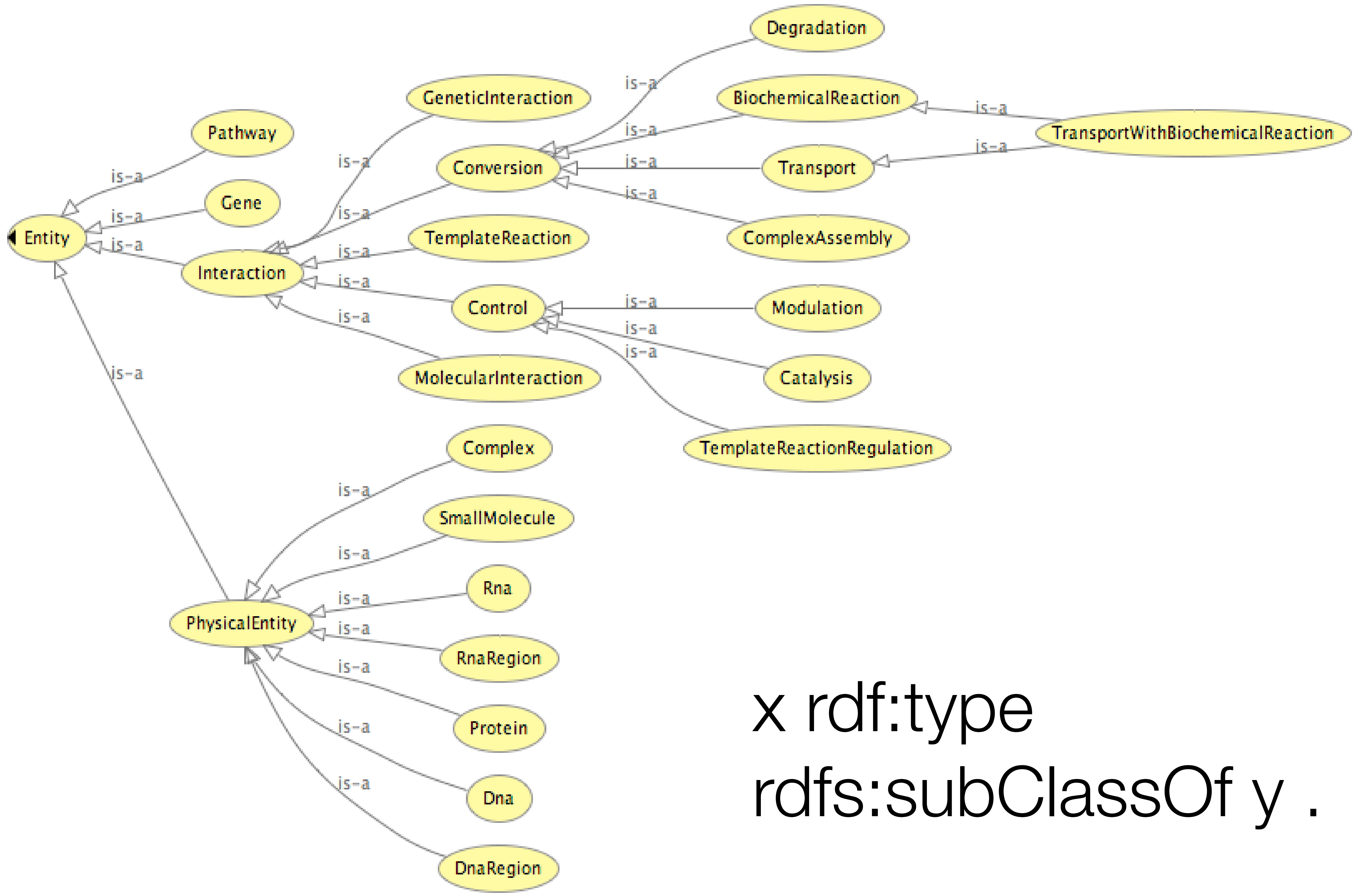


RDFS is about sets

- RDF creates a graph structure to represent data (there is no meaning)
- This is what SPARQL will Query.
- BioPAX uses RDFS contructs
- RDFS is the **vocabulary** that is used in the RDF
  - individuals that are related to one another and how....
- Introduces the concept of a "distinguished resource"
  - Specific triples have a special meaning (specified inferences)

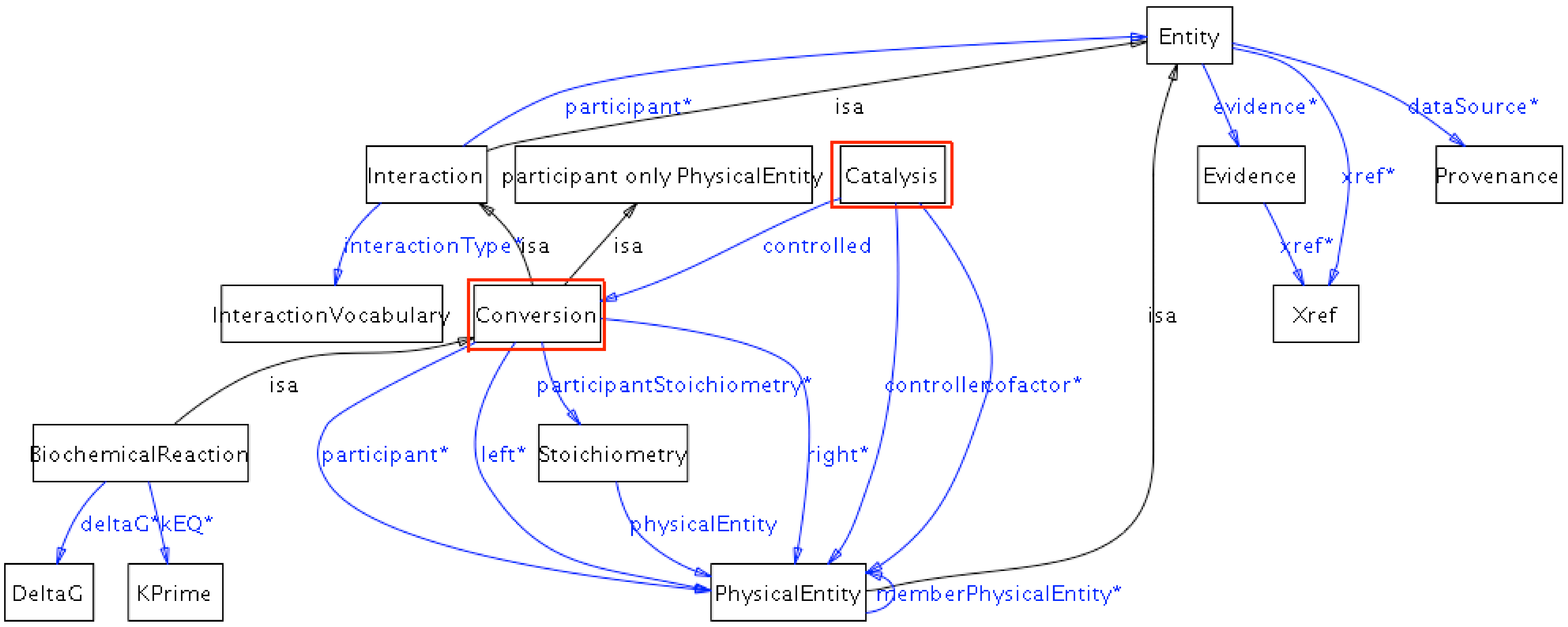
- Type Propogation
  - rdfs:subClassOf (isA subsumption hierarchy)
- Relationship Propogation
  - rdfs:subPropertyOf
- Usage
  - rdfs:domain
  - rdfs:range

BioPAX Type Propogation



x rdf:type
rdfs:subClassOf y .

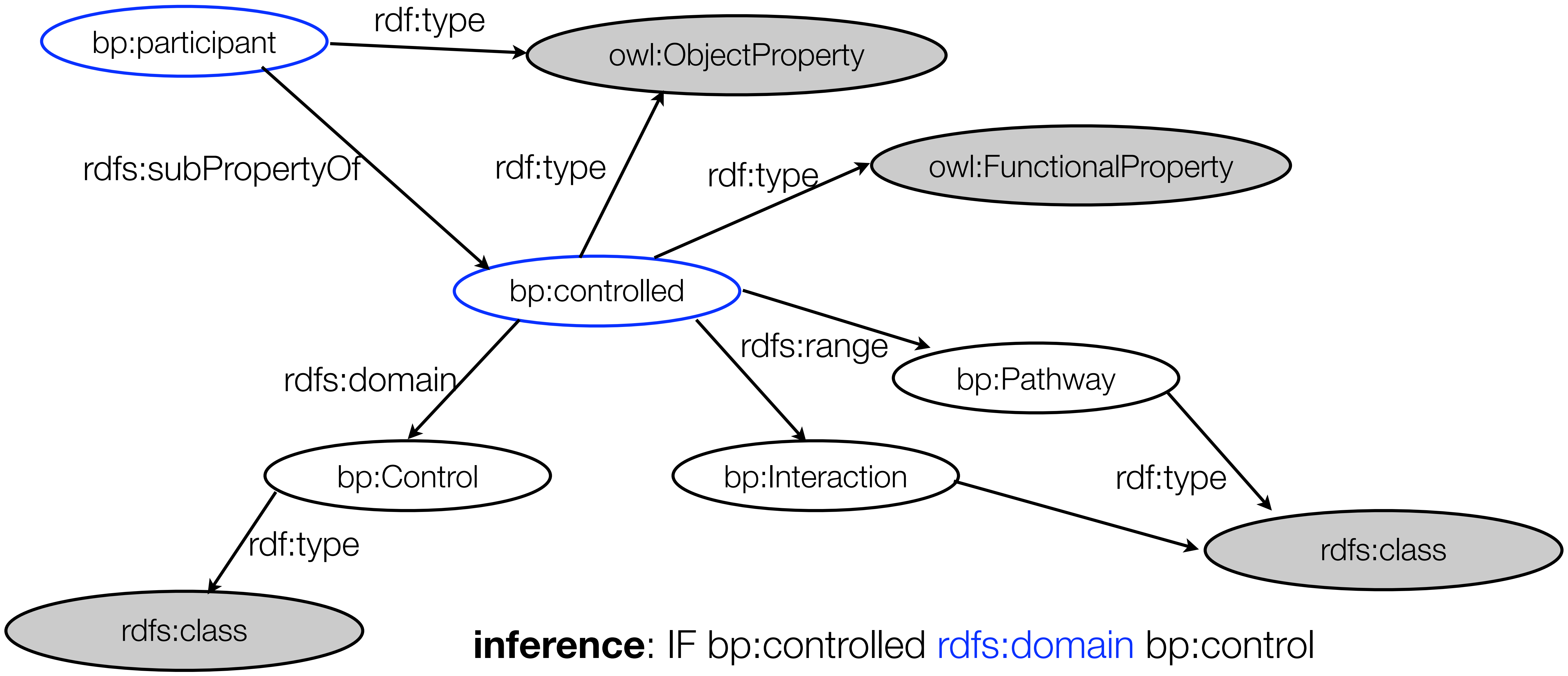
BioPAX - Relationship Propogation



$$\text{domain} \xrightarrow{\text{object property}} \text{range}$$

$$\text{catalysis} \xrightarrow{\text{controlled}} \text{conversion}$$

BioPAX - Relationship Propogation

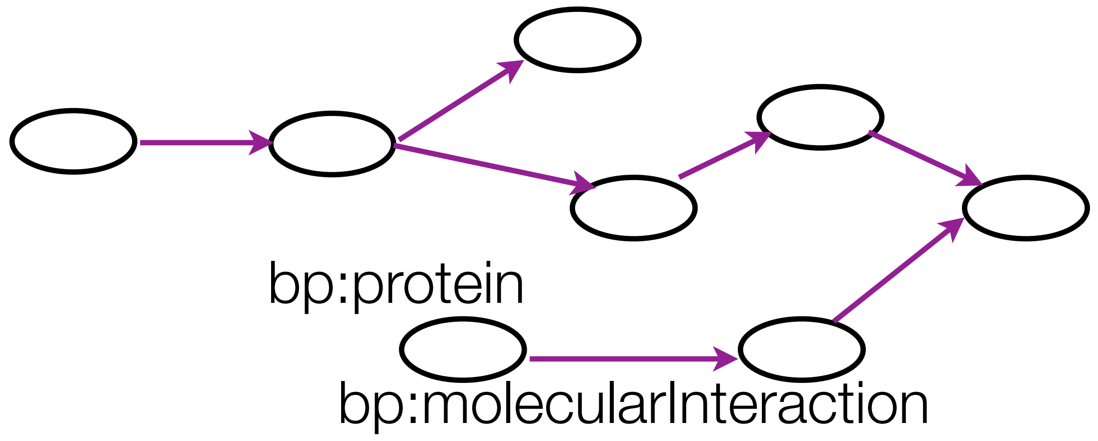


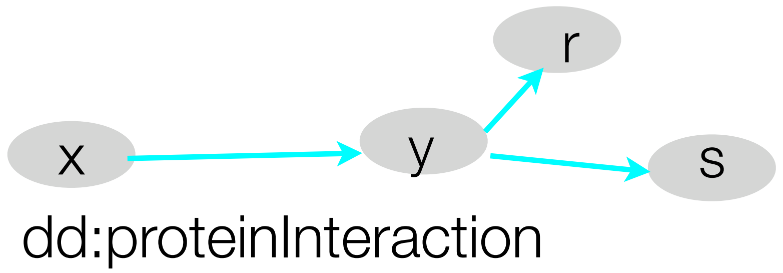
inference: IF bp:controlled rdfs:domain bp:control
and x bp:controlled y
THEN
x rdf:type bp:control

NB: inference not inheritance

RDFS - extended integration

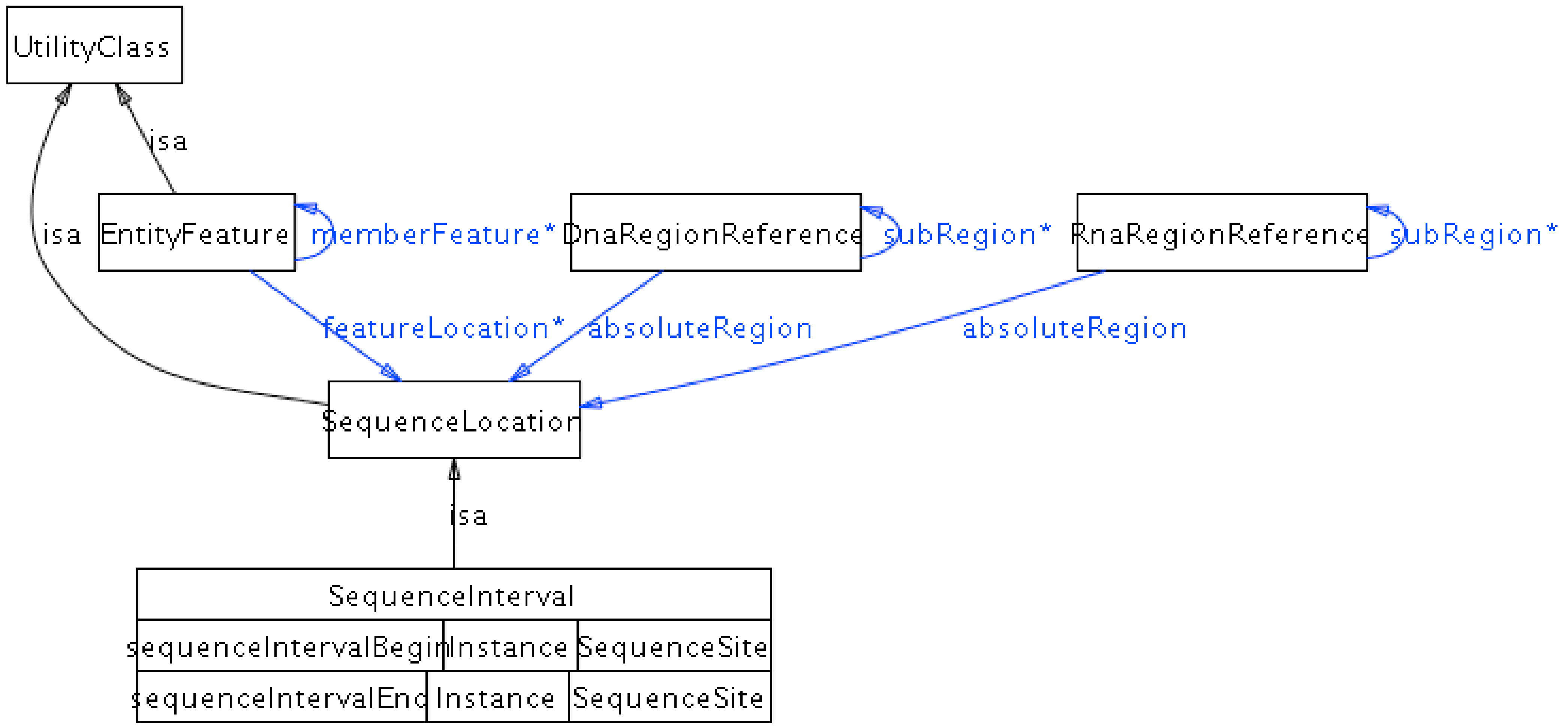
- disclaimer - set operations not **directly** supported
- Set Intersection C ⊆ A ∩ B
  - but it can be aproximated
    C rdfs:subClassOf A, C rdfs:subClassOf B
    given x rdf:type C then inferernce: x rdf:type A and x rdf:type B
- Set Union - integration! A ∪ B ⊆ C can be approximated in a similar way
- Property intersection and Property Union can work in the same way using rdfs:subPropertyOf





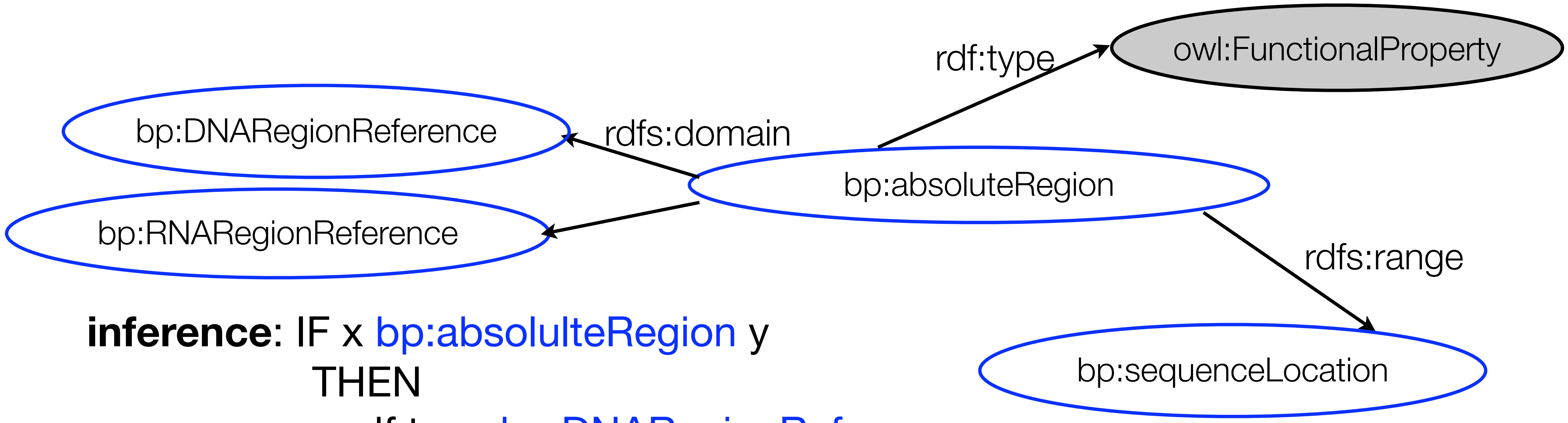
dp:proteinInteraction **rdfs:subPropertyOf** bp:MolecularInteraction

bp:absoluteRegion



RDFS - The semantics of domain and range

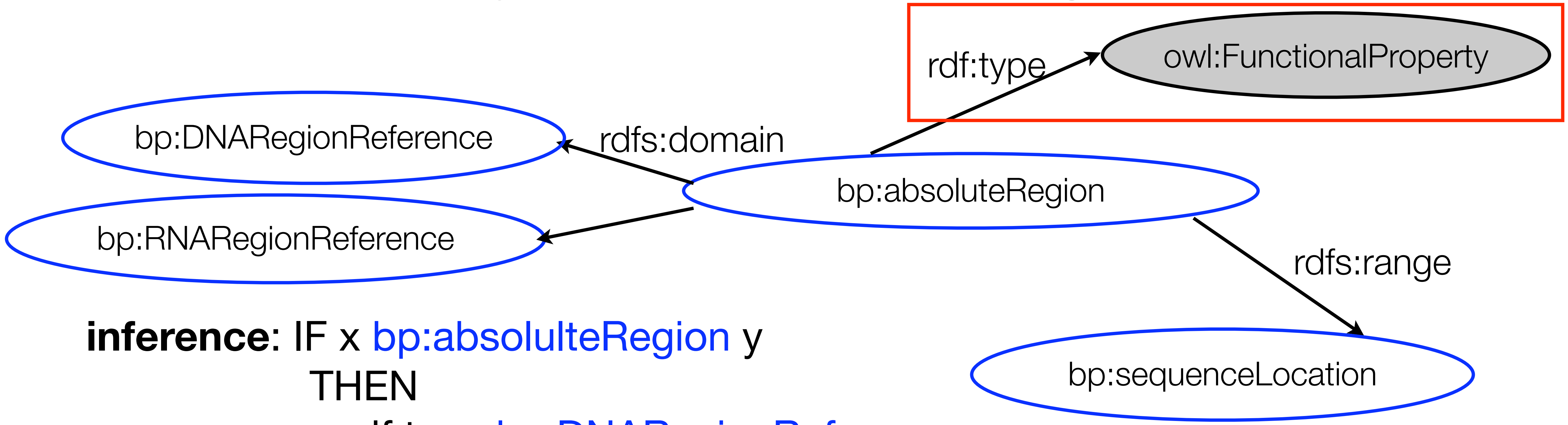
- N.B. inference (not restrictions)
- domain and range functions return sets which are to be interpreted as intersections
- Conjunctive:
- Properties can have any number of domains and ranges - **They all apply**



```
inference: IF x bp:absolulteRegion y
              THEN
              x rdf:type bp:DNARegionReference
              AND
              x rdf:type bp:RNARegionReference
```

RDFS - The semantics of domain and range

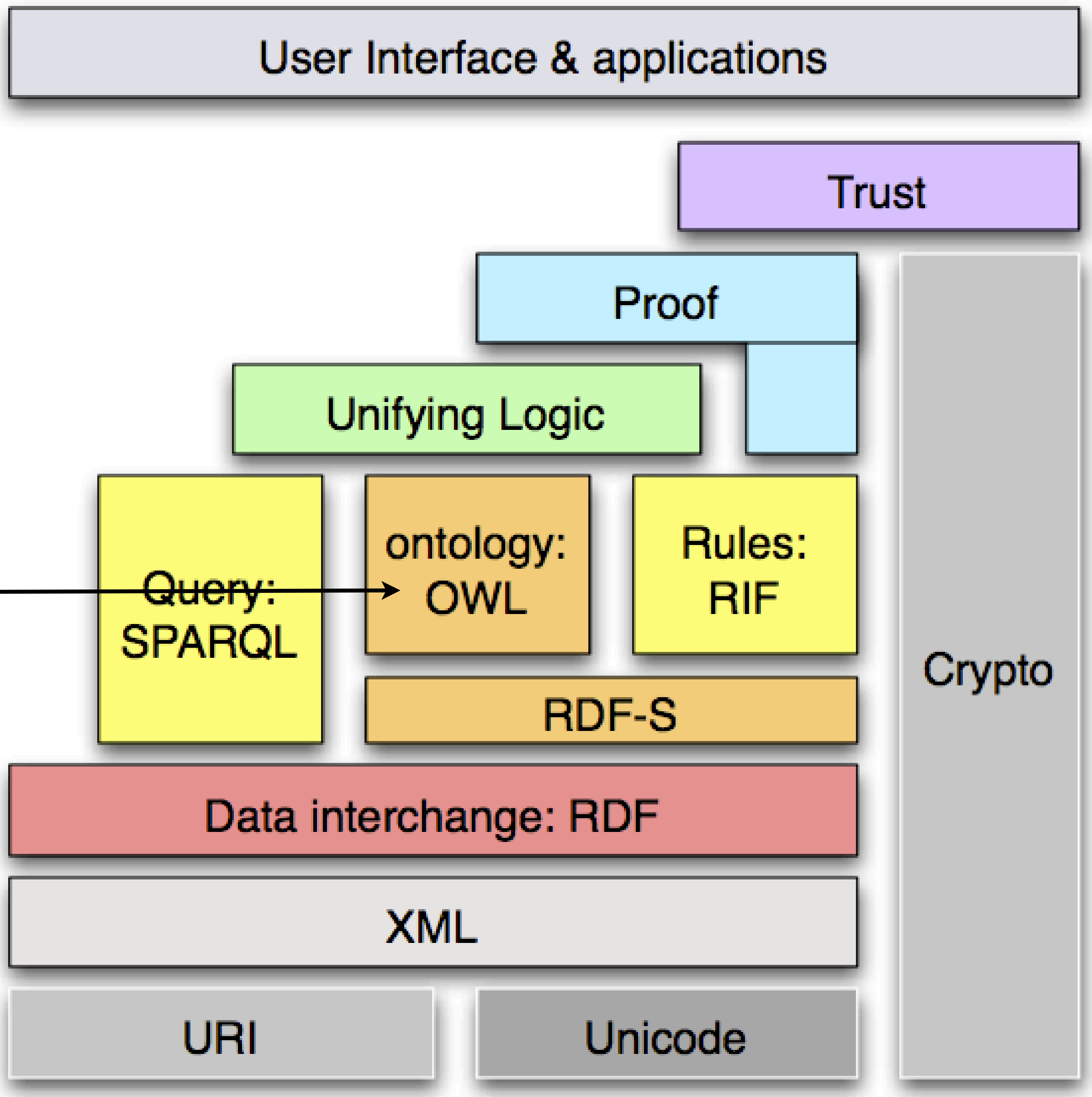
- N.B. inference (not restrictions)
- domain and range functions return sets which are to be interpreted as intersections
- Conjunctive:
- Properties can have any number of domains and ranges - **They all apply**



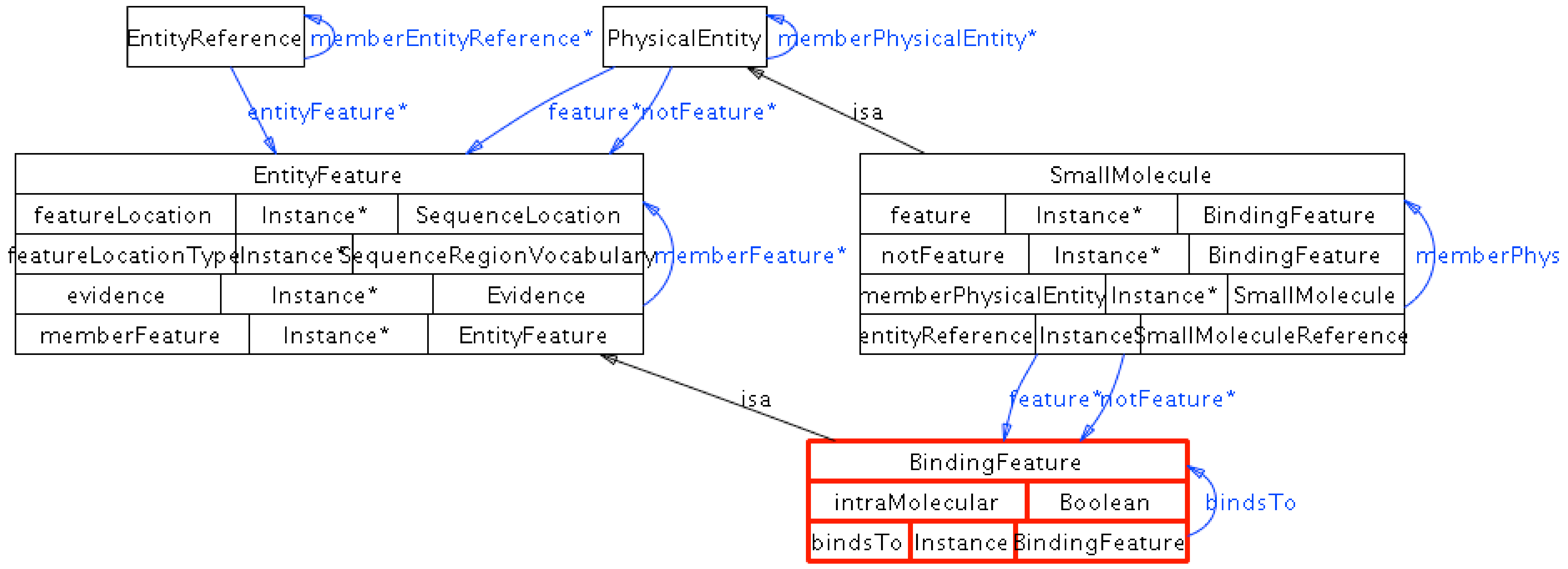
inference: IF x bp:absolulteRegion y
THEN
x rdf:type bp:DNARegionReference
AND
x rdf:type bp:RNARegionReference

BioPAX uses OWL

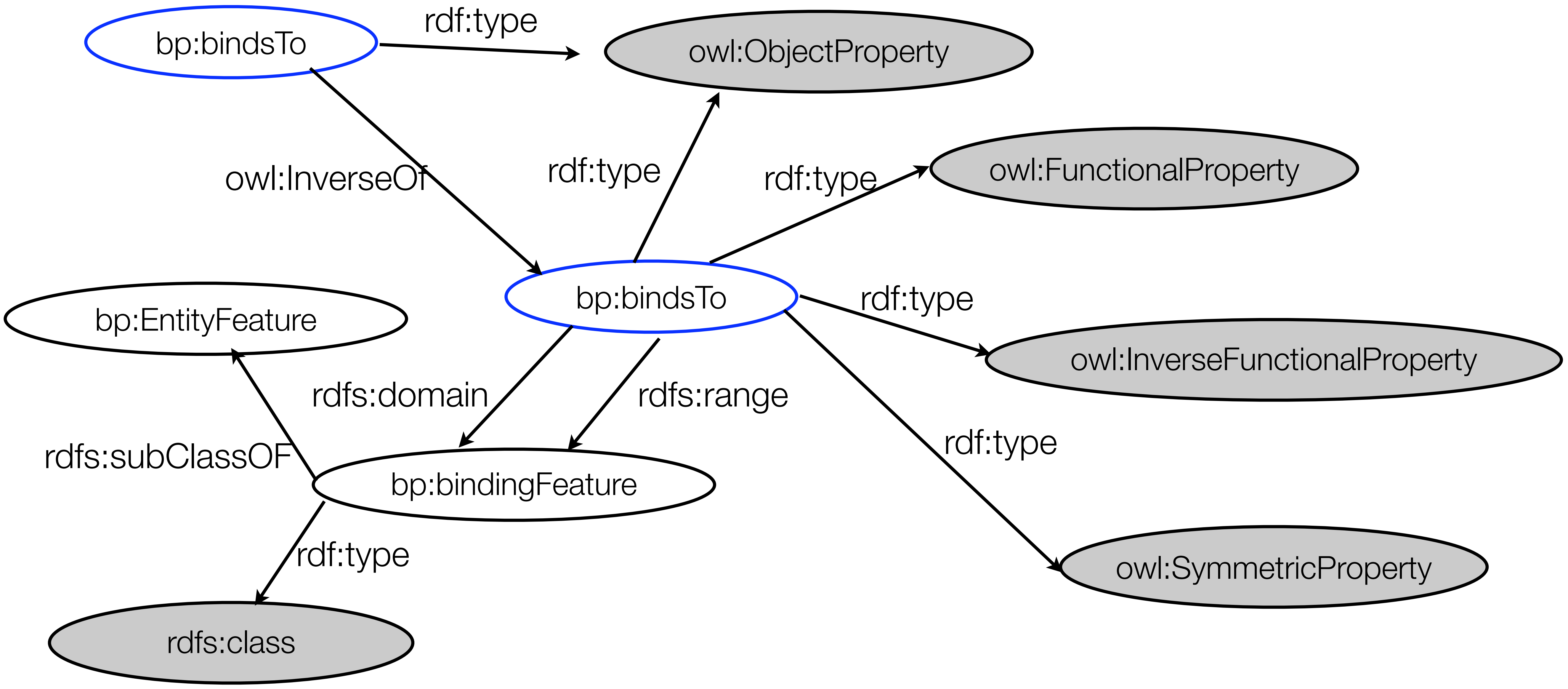
2nd layer of Semantics



bp:bindingFeature


EntityReference
memberEntityReference*
PhysicalEntity
memberPhysicalEntity*
entityFeature*
feature*
notFeature*
isa
EntityFeature
featureLocation | Instance* | SequenceLocation
featureLocationType | Instance* | SequenceRegionVocabulary
evidence | Instance* | Evidence
memberFeature | Instance* | EntityFeature
memberFeature*
SmallMolecule
feature | Instance* | BindingFeature
notFeature | Instance* | BindingFeature
memberPhysicalEntity | Instance* | SmallMolecule
entityReference | Instance | SmallMoleculeReference
memberPhys
isa
feature*
notFeature*
BindingFeature
intraMolecular | Boolean
bindsTo | Instance | BindingFeature
bindsTo

OWL



bp:bindsTo
rdf:type
owl:ObjectProperty
owl:InverseOf
rdf:type
rdf:type
owl:FunctionalProperty
bp:bindsTo
bp:EntityFeature
rdf:type
owl:InverseFunctionalProperty
rdfs:domain
rdfs:range
rdfs:subClassOF
rdf:type
bp:bindingFeature
owl:SymmetricProperty
rdf:type
rdfs:class

OWL

Symmetric Properties

- owl:inverseOf owl:inverseOf owl:inverseOf .
- owl:inverseOf rdf:type owl:SymmetricProperty .

Transitive Properties

- if (X partOf Y) Π (Y partOf Z) : (X partOf Z)

Functional Properties

- only one value (for any value of x there can be only one value of x^2)
- (x hasSquared A) Π (x hasSquared B) Π (hasSquared rdf:type owl:FunctionalProperty) : A sameAs B

InverseFunctional Properties

- think of keys in relational databases...
- (A componentOf x) Π (B componentOf x) Π (componentOf rdf:type owl:InverseFunctionalProperty) : A same as B (or using x we can get both A and B)

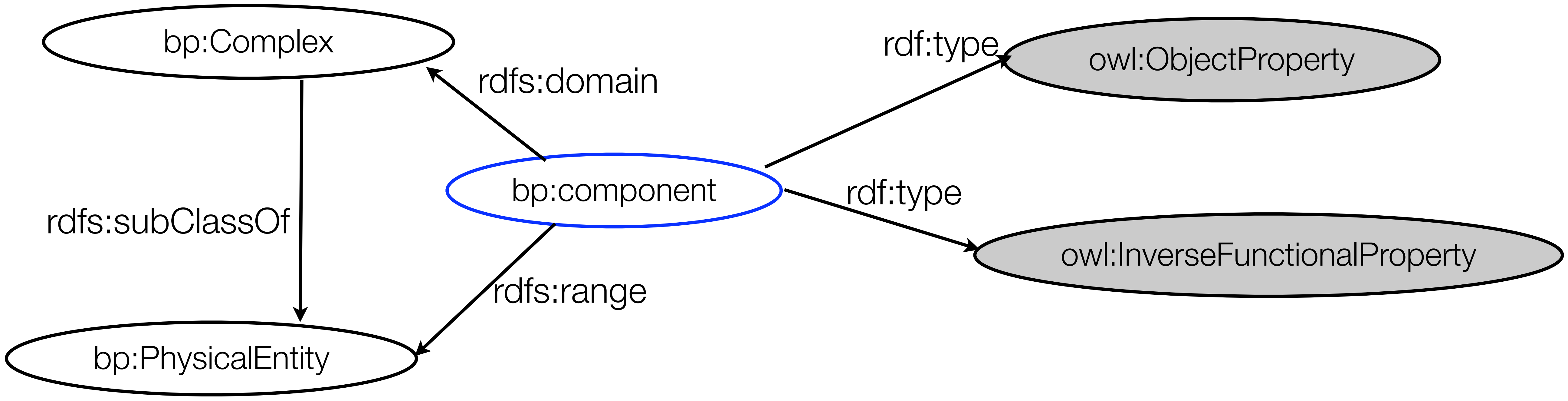
Caveats - Gotchas

BioPAX assumes a closed world
Integrity constraints (CWA)

- prevent “incorrect” values from being asserted in a model
- used for validation/parsing/data input
- single model that contains only the facts asserted
- in databases you are aiming for one version of the truth

BioPAX uses OWL constructs in strange ways, strange things can happen....
SPARQL does not understand OWL

OWL InverseFunctionalProperty

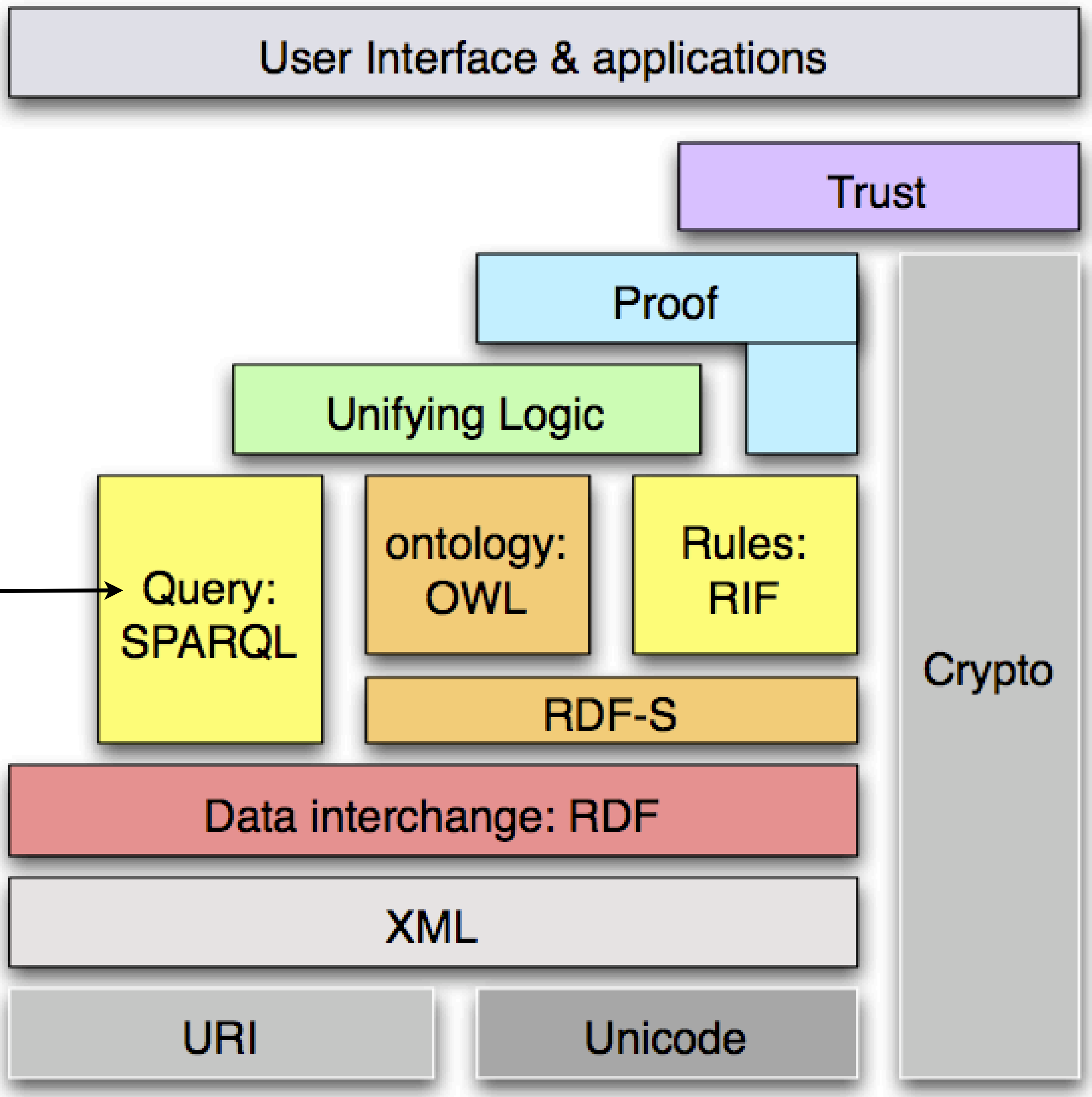


```
inference: IF A bp:component x
               AND
               B bp:component x
               THEN
               B rdf:type bp:complex, A rdf:type bp:complex
               AND
               A owl:isA B (i.e. the same complex)
```

strangeness!

SPARQL, how does this effect queries?

A Box Queries
RDF Graph



SPARQL - the basics

- Triple pattern, like an RDF triple but with a variable
  <subject> <predicate> ?obj .
- Variables
  ?subj rdf:type ?obj .
  Each triple that matches the pattern will bind an actual value from the RDF data set
  All possible bindings are considered
- Graph Patterns
  multiple triple patterns

SPARQL - Query Structure

BASE - all relative URIs resolved against

PREFIX - SPARQL Equivalent of XML namespace

```
BASE <http://www.biopax.org/release/>
PREFIX rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX bp:   <biopax-level3.owl#>
SELECT ?x
FROM <>
WHERE {
{?x  rdf:type  bp:Protein . }
```

SELECT - data items to be returned by the query

FROM - data against which query will run

WHERE - graph pattern

SPARQL - Graph Pattern

- Graph Patterns

  within a graph pattern a variable must have the same value no matter where it is used
  e.g. ?protein is always bound to the same resource
  therefore a resource that does not contain all the triples won’t satisfy the graph pattern

  you can not select a variable if it is not listed in the graph pattern

  you can use shortcuts e.g. SELECT *

SPARQL - Graph Pattern 1

select all interactions in a pathway that are BiochemicalReactions:

```
SELECT ?name
WHERE
    {
    ?pathway rdf:type bp:Pathway .
    ?pathway bp:pathwayComponent ?c .
    ?c bp:name ?name.
    ?c rdf:type bp:BiochemicalReaction
    }
```

SPARQL - Graph Pattern 2

select all proteins that take part in a molecular interaction and are part of a complex:

```
SELECT  ?participant
WHERE
{
?protein rdf:type bp:Protein.
?interaction rdf:type bp:ComplexAssembly.
?interaction bp:participant ?protein.
}
UNION
{
?protein rdf:type bp:Protein.
?complex rdf:type bp:Complex.
?complex bp:component ?protein.
}
```

SPARQL - Graph Pattern 3

select all complexes that have components that are both complexes and proteins (a left outer join in SQL)

```
SELECT *
WHERE
{
?x bp:component ?y.
?x rdf:type bp:Complex .
?y rdf:type bp:Complex .
OPTIONAL {?y rdf:type bp:Protein}
}
```

SPARQL - Query

SELECT

ASK

DESCRIBE

CONSTRUCT

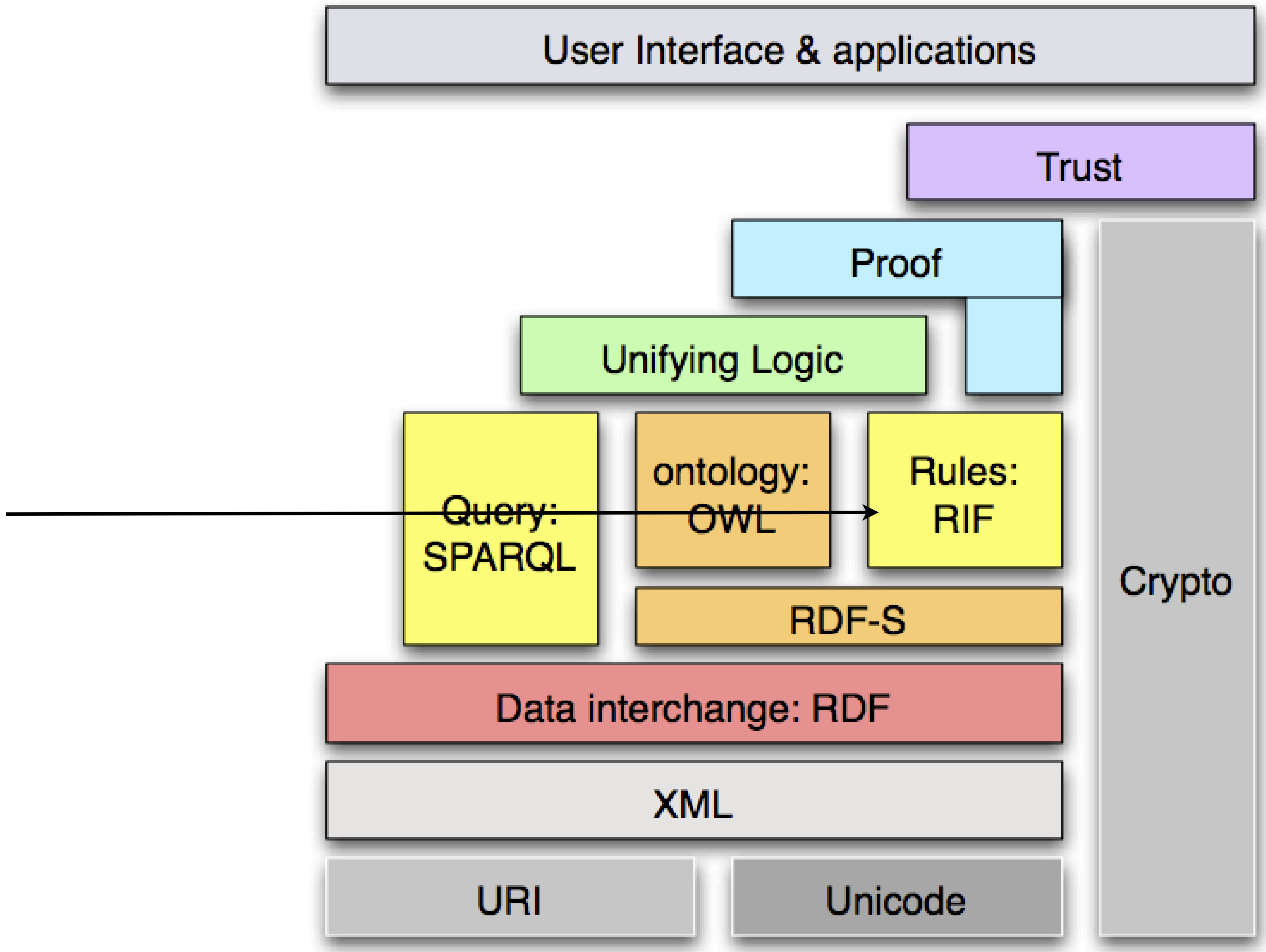
hands on ...

- start a terminal
- create a directory jena_workspace, move into that directory
- download jena.jar (http://tinyurl.com/**3vlp7rw**)
- download biopax data (http://www.biopax.org/Junk/Homosapiens.nt or a smaller file (http://www.biopax.org/Junk/ Escherichia_coli.nt)

- Andrea on hand to help....

Semantics

T Box Queries



but....not T box

- What does this mean?
  - If you are combining a BioPAX graph with another BioPAX graph - nothing - still an A box query
    BUT
  - if you are combining a BioPAX graph with another RDF graph with OWL property characteristics - then you need to compute the T box, materilize the inferred triples, and then you can SPARQL!!

- Can we compute a T Box?
- BioPAX++? A BioPAX extension with closure axioms?
- and now for the substance.

BioPAX is very SPARQLy

- For all that is wrong with BioPAX, there is alot that you can do with
  - a little RDFS, a little OWL and alot of SPARQL.
  - Using Protege BioPAX/SPARQL/SQWRL.....
  - Using Jena BioPAX/SPARQL.....