nCircle

W3C Semantic Technologies

Beyond XML

SCAP Developer Days 2009

Tim “TK” Keanini, CTO

Agenda

- Introduction & Poll
- Goals/Challenges of SCAP
- Brief introduction to W3C semantic technology
  - Address myths and absurdities
  - Intro to RDF/RDFS/OWL, SPARQL
- Useful Vocabularies for Ontological Engineering
  - FOAF, EARL
- Illustrative ideas for exploratory discussion
  - Windows OS Naming, Composite Vulnerability, Class Escalation
- References
  - Libraries, RDF Stores, IDE's, Webpages, blogs, etc

Introduction and Poll

- Tim “TK” Keanini
  - CTO for nCircle
  - Why I became interested in Interoperability and then the W3C Semantic Stack?
    - Supply-side problem
      - IT Security market is too fragmented
      - Companies will acquire or be acquired
        - » Same time to market problem
    - Demand-side problem
      - Customers all share a common requirement for multivendor interoperability (every vendor on the floor of RSA interoperating)
      - Syntax-level interoperability will not be sufficient
- Disclaimer and Objective
- Quick poll: Who has investigated:
  - RDF? RDFS? OWL? SPARQL? Description Logic?

SCAP Definitions and Goals

- NIST SP800-117 – "Guide to Adopting and Using Security Content Automation Protocol"
  - "Comprehensive & Standardized Approach"
    - …**organizing and expressing security-related information**…
  - Demonstrate compliance with security requirements
  - **Content Interoperability across automated tools**
- NIST SP800-126 – "Specification for the Security Content Automation Protocol"
  - "…describes the basics of the SCAP **components** specification and **interrelationships**."
  - …**characteristics of SCAP content** and **SCAP level requirements** not defined in individual component specifications
  - ".. to achieve **security automation**…"

We are off to a great start!

- We have common Names (syntax level)
- We have a common method of ranking vulnerabilities (members within that set)
- We have a call-to-action for software developers to provide benchmarks for their “platform”
- Today we have a common repository for content (NVD)

Interoperability challenges to address (IMHO)

- Syntax Interoperability versus Semantic Interoperability
  - Regex'able versus Inference
- Semantic Interoperability across SCAP
  - SP-800-126 does provide some semantic framework, it would benefit greatly from the machine-readability of RDF/RDFS/OWL
  - The horizontal nature of security and compliance demands support for heterogeneous view-points.
  - Ability to express "sentence" from the "words" of the enumerations (composability at the SCAP level)
- Knowledge Representation Problem/Opportunity
  - XML/XML-Schema/XSLT are useful and stable
  - RDF/RDFS/OWL/SPARQL
    - Leverages what we already know
    - Use only what you need
    - W3C technologies interoperability

nCircle

W3C Semantic Technologies

nCircle

Myths about Ontologies and Semantic Web

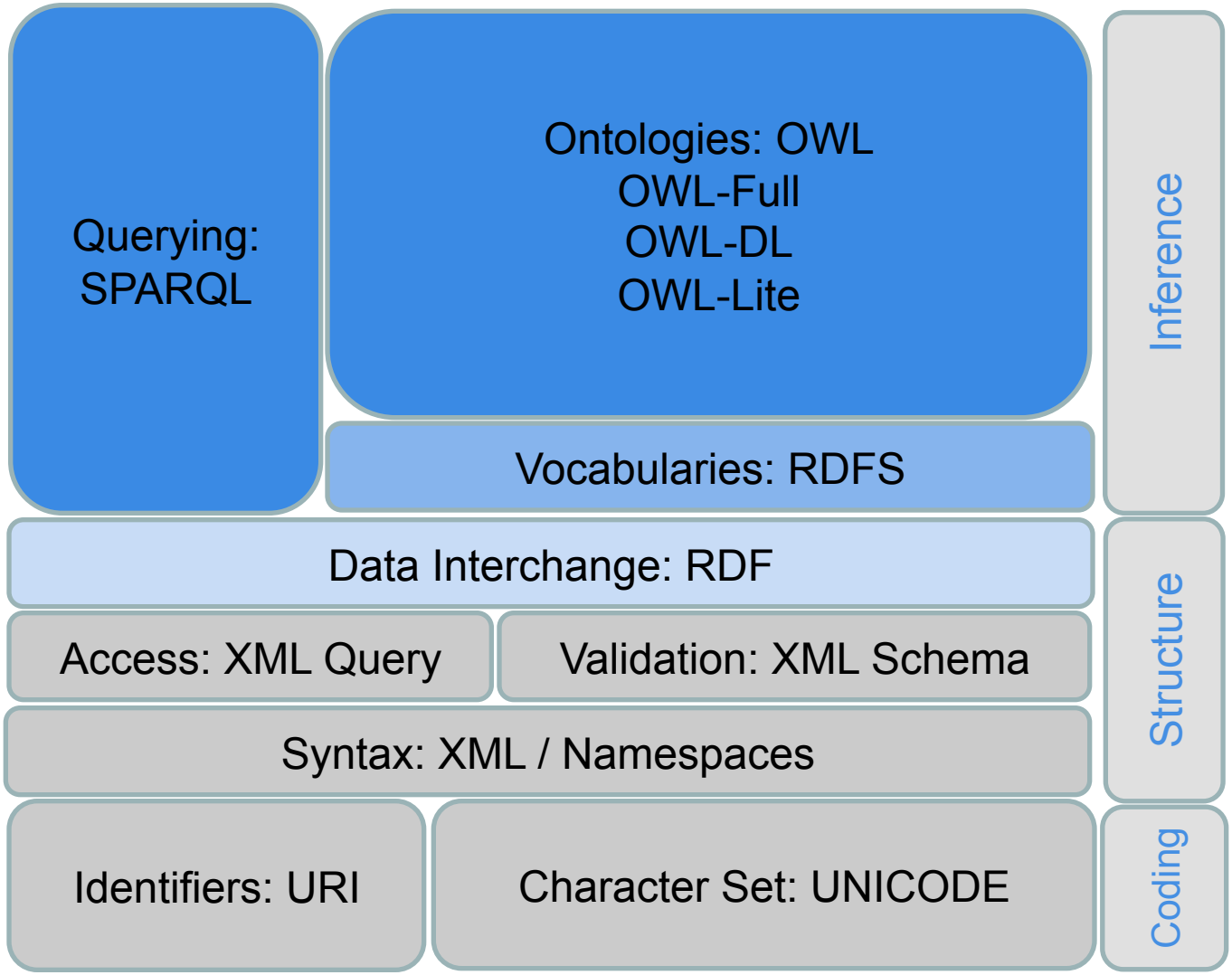
- Semantic Technologies are only about the Web
  - False: Semantic Modeling is about Knowledge Representation
- Semantic Technologies are unrelated to XML
  - False: It pick up where XML leaves off.
- Ontologies are too complex to understand or use
  - False: It can be only as complex as it needs to be. Use what you need.
- Ontologies and Taxonomies are the same
  - False: Taxonomies only allow for parent-child relationships
    - Ontologies are much more expressive and dynamic than Taxonomies
- Ontologies are difficult to create and change all the time
  - False: It is just another language to help model your domain
  - False: change happens! It offers a robust language for versioning
- W3C standards are the only way to perform semantic modeling
  - False: but the interoperability goals of the W3C show great potential

nCircle

Absurdities

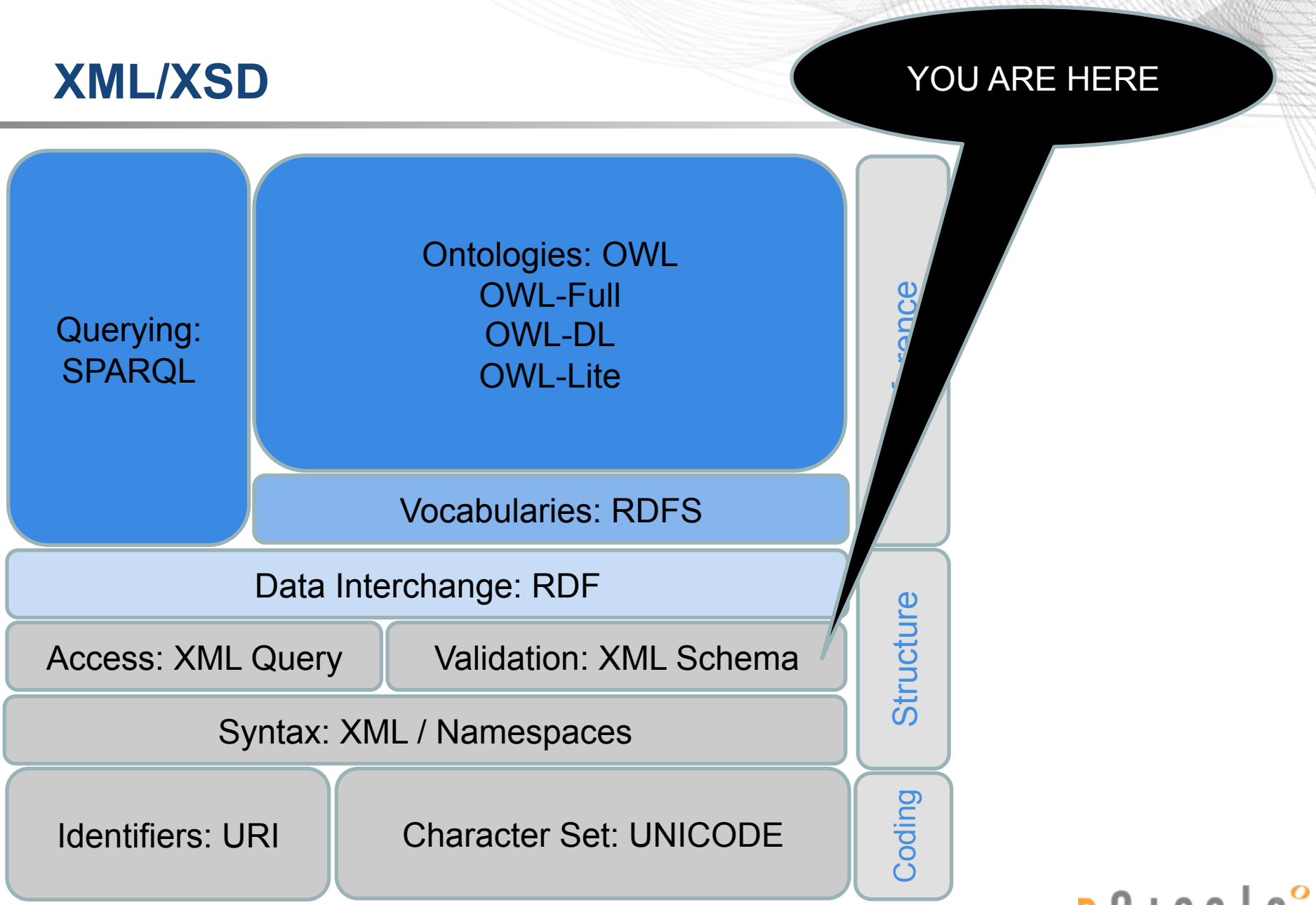
- Machine understanding on par with human understanding
- Describe all of the aspects of the observable world
- Create sentient machines
- It is the silver bullet for all of SCAP

W3C Semantic Technology Stack



nCircle

XML/XSD

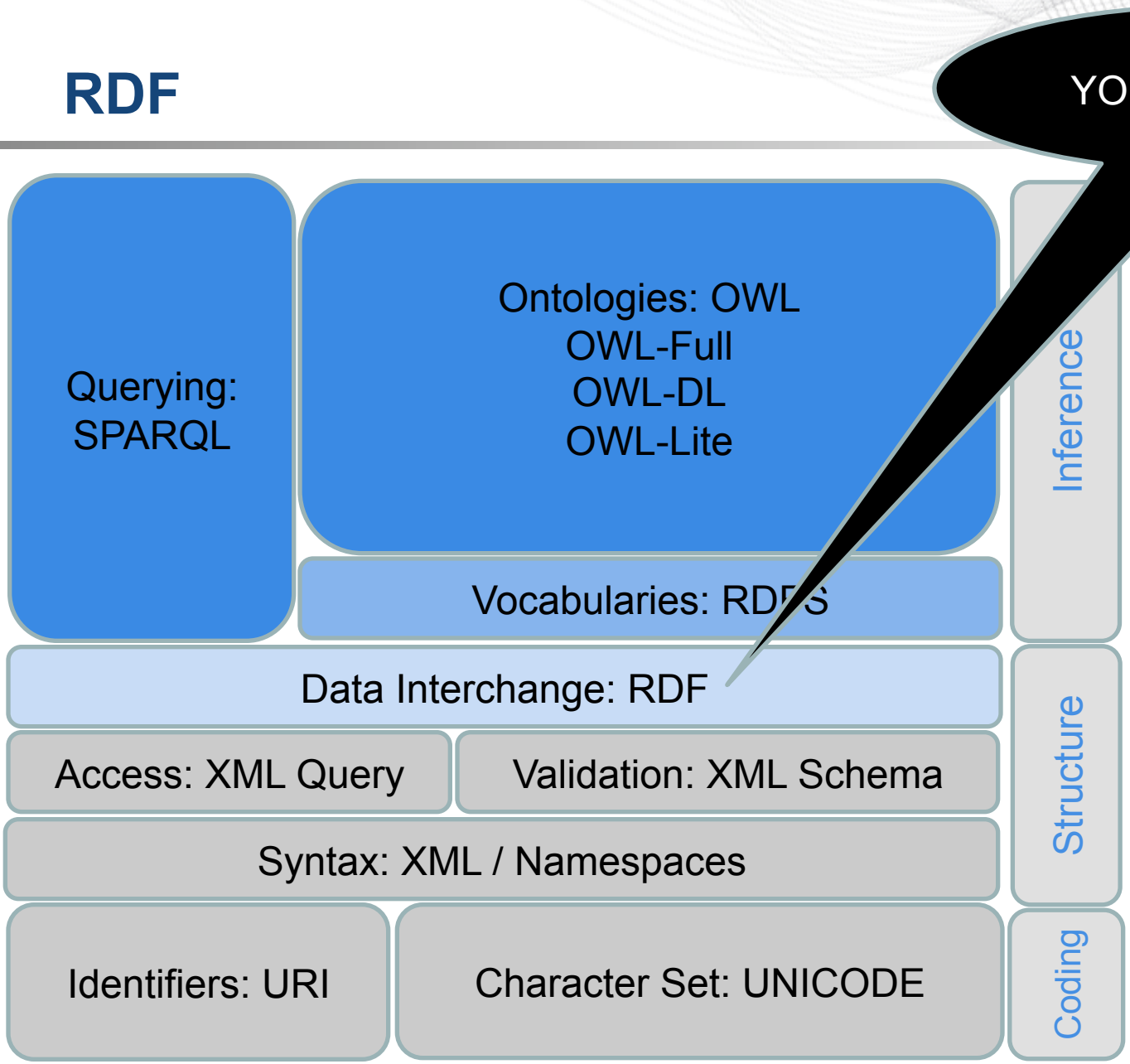


Where are we today? XML/XML Schema

- General purpose markup-language to describe structured documents
- Tree-like syntax for tree-structured data
- Like a taxonomy, terms are classiffied hierarchically
  - Limited to generalization, is-a, type-of, parent-child, etc
  - From general to more specific concepts
- XML Schemas support explicit *application-specific* structures, cardinality, and datatyping constraints. Example:
  - "title is mandatory"
  - "date must be after 1980"
  - "title must be a string"
  - "there can be no more than three titles“
- Infrastructure for serialization and data-level policy

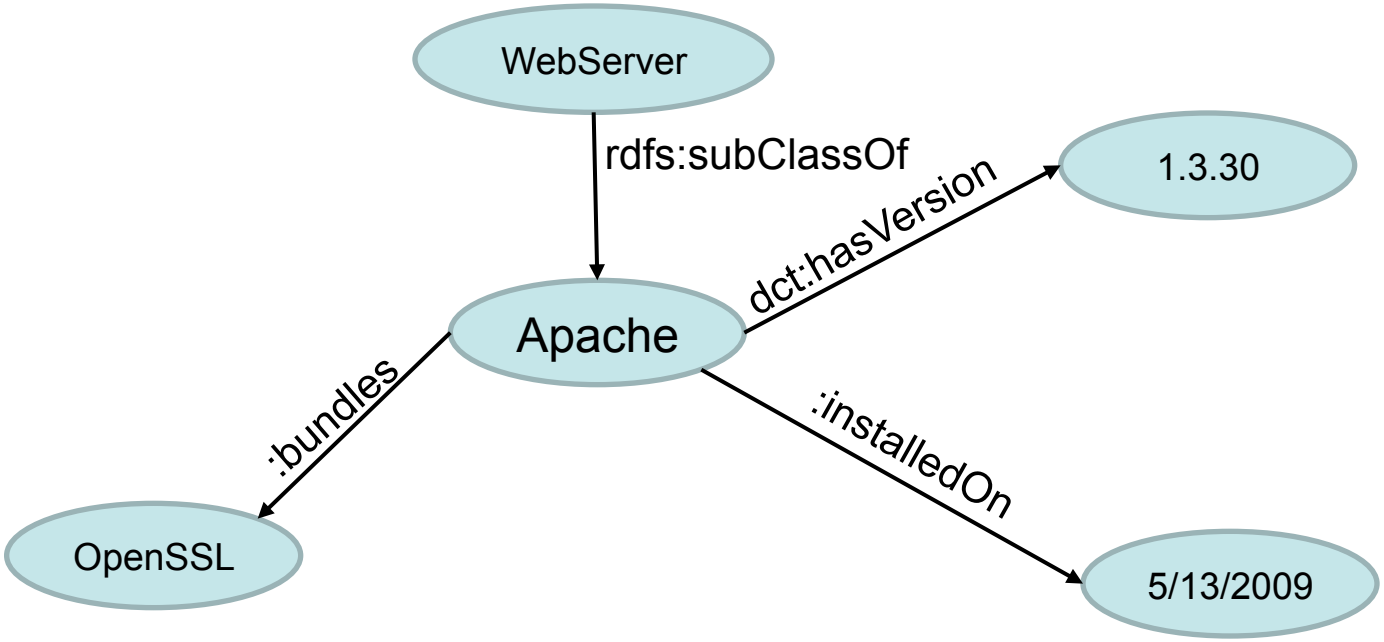
nCircle

RDF



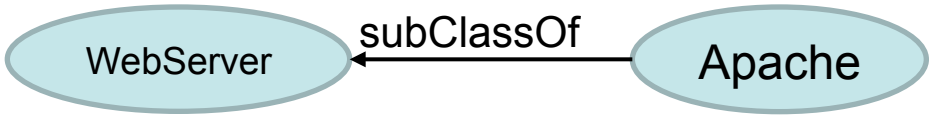
RDF – Resource Description Framework

- Data Model is a ‘labeled-directed graph’
  - All nodes and arcs have some type of label (identifier)
  - Arcs point only in one direction



RDF – Resource Description Framework

- All statements in the form of a triple
  - Subject-Predicate-Object (S,P,O)
  - Set of these triples begin to model a domain in the form of a graph



- Statement and Reification
  - `cpe:App123 cve:isVulnerableTo cve:CVE-1999-0067 .`

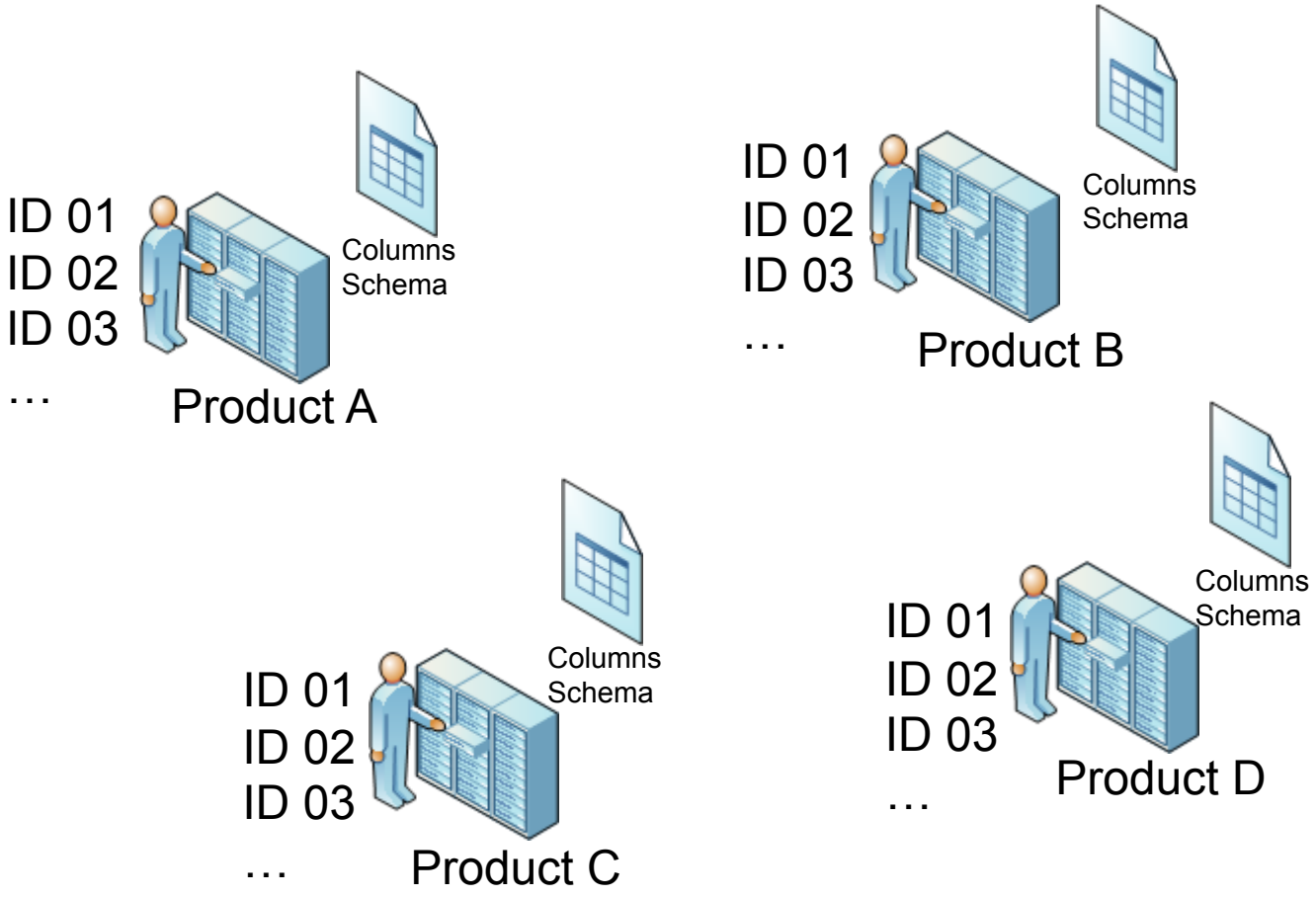
  Reification
    - `foo:triple456 rdf:type rdf:Statement .`
    - `foo:triple456 rdf:subject cpe:App123 .`
    - `foo:triple456 rdf:predicate cve:isVulnerableTo .`
    - `foo:triple456 rdf:object cve:CVE-1999-0067 .`
  - `foo:triple456 cve:discoveredBy vendor:scanner22 .`

nCircle

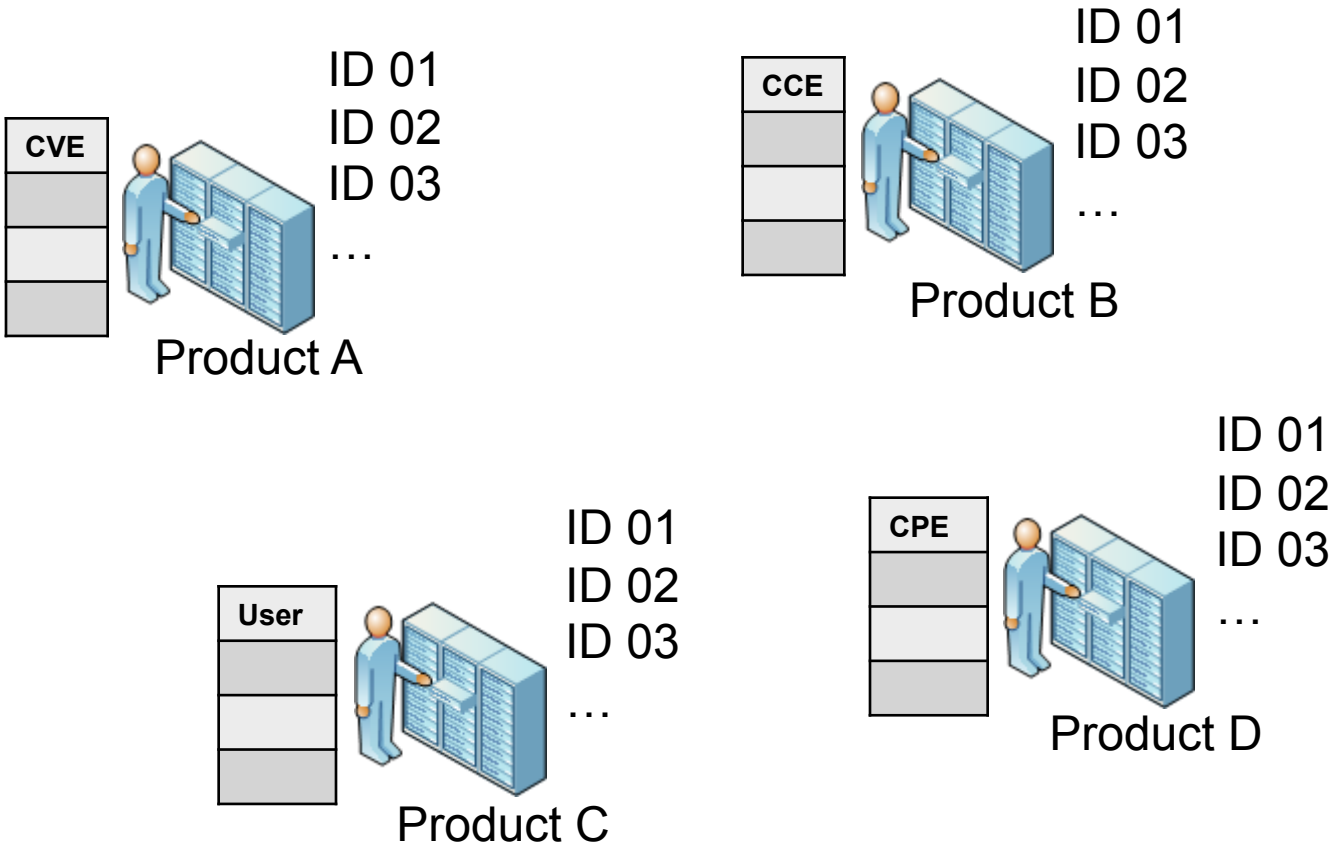
Table == Row, Column, Value == Set of Triples

ID	hasOnline Account	End-Point _Address	CVE	CCE	Business Function
01	Alice	10.20.10.11/32	CVE-1999-888	CCE-2002-787	eCommerse
02	Alice	10.20.10.10/32	CVE-2001-234	CCE-2005-345	Supply Chain
03	Bob	10.20.10.11/32	CVE-2002-444	CCE-2006-666	Supply Chain
04	Bob	10.20.10.12/32	CVE-2004-555	CCE-2002-222	Supply Chain
05	Bob	10.30.10.10/32	CVE-2006-111	CCE-2002-322	Back Office
06	Carol	10.40.10.10/32	CVE-2006-234	CCE-2007-999	Back Office
06	Carol	10.50.10.10/32	CVE-2007-777	CCE-2007-111	HR

Row Based Multi-Vendor Architecture



Column Based Multi-Vendor Architecture



RDF Based Architecture and SPARQL Query

Product A

	CVE
ID1	CVE-1999-888

	CCE
ID4	CCE-2002-222

Product B

	CVE
ID2	CVE-2001-234

	CCE
ID3	CCE-2006-666

Product D

	hasOnlineAccount
ID1	Alice

	End-Point-Address
ID1	10.20.10.11/32

	hasOnlineAccount
ID4	Bob

Product C

```
PREFIX cve: <http://nvd.nist.gov/cve/1.1/>
SELECT  ?who ?vulnerability ?cidr
FROM <ProductA>
FROM <ProductB>
FROM <ProductC>
FROM <ProductD>
WHERE {
 ?x :hasOnlineAccount      ?who .
 ?x cve:End-Point-Address ?cidr .
 ?x cve:hasCVE   ?vulnerability .
```

RDF – Resource Description Framework

Common Term	Synonyms
Resource	Subject, Object
Resource identifier	Name, URI, ID, identifier, URL, label
Statement	Triple, statement, assertion
Subject	Source, resource, “row”, node
Predicate	Property, “column”, arc
Object	Value, resource, literal, node
RDF Store	Triple Store, Graph Database

nCircle

RDF Syntax

- How one would express:
  - Apache is a member of the set Webserver
- RDF/XML

```
<rdf:Description rdf:about="#Apache">
   <rdf:type rdf:resource="#Webserver"/>
 </rdf:Description>
```

- N3

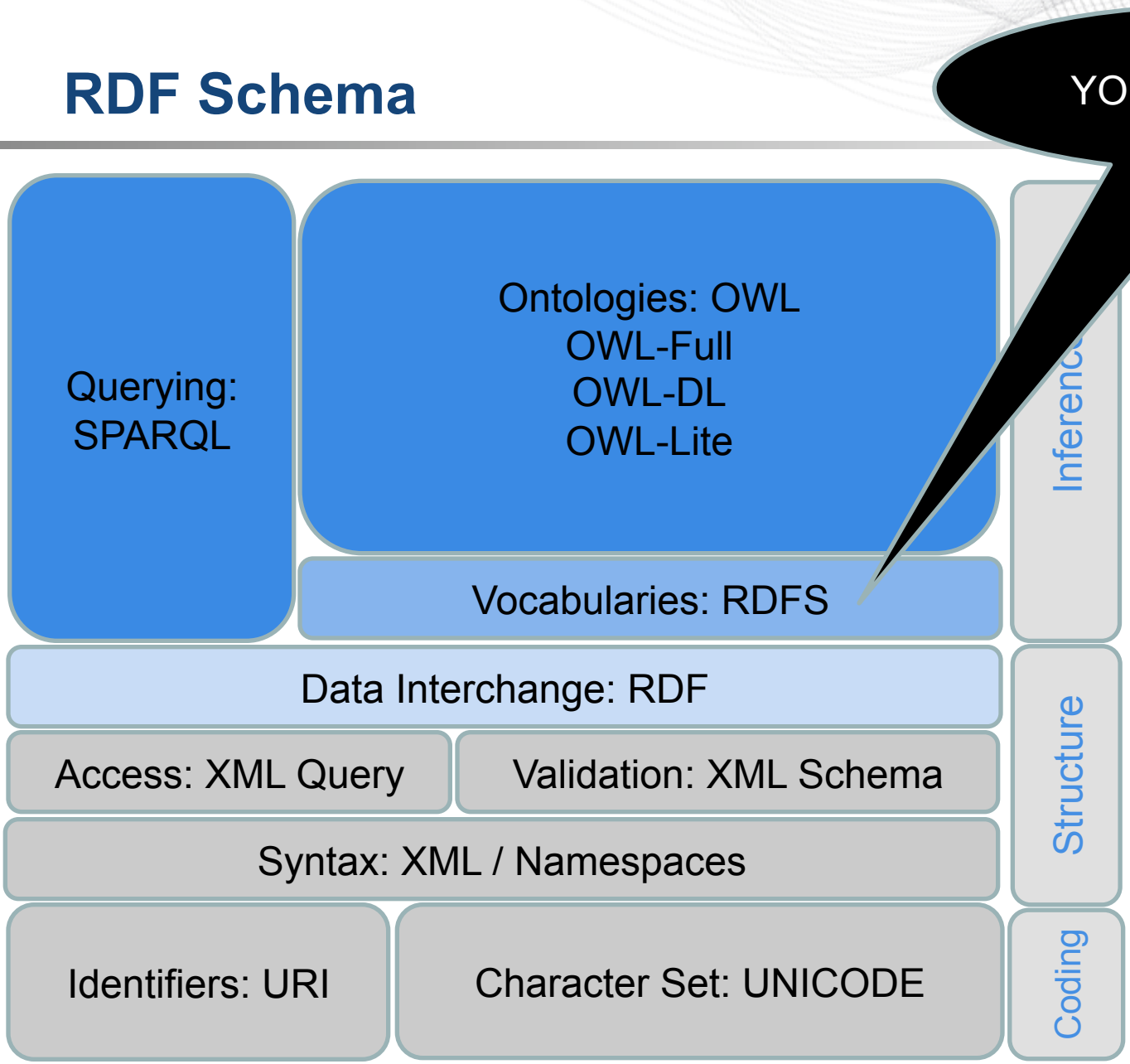
```
:Apache   rdf:type    :Webserver .
:Apache   a   :Webserver .
```

- RDF/XML-ABBREV

```
<Webserver rdf:ID="Apache"/>
```

- SeeAlso: TURTLE and N-TRIPLE

RDF Schema



nCircle

RDF Schema (RDF-S)

- RDF Vocabulary Description Language 1.0: RDF Schema
  - Vocabulary defined with RDF statements (triples)
- RDF-S is about
  - Relation between classes (Class , subClassOf)
  - Relation between properties (Property, subPropertyOf)
  - Class membership of individuals via properties (domain, range)
- Provides some sense of “meaning” to the RDF data
  - Meaning = what we can explicitly infer from the data
  - Axioms that express exactly what inference can be drawn
  - Semantics expressed through the mechanism of inference
  - Lets explore in the next slides how this works

nCircle

Type Propagation

- rdfs:Class

```
:Root_Kit rdf:type  rdfs:Class .
:Malware  rdf:type  rdfs:Class .
```

AXIOM
IF
A rdfs:subClassOf B .
r rdf:type A .
THEN
r rdf:type B .

- rdfs:subClassOf

```
:Root_Kit rdfs:subClassOf  :Malware  .
:foobar   rdf:type         :Root_Kit .
```

we can then infer the triple

```
:foobar   rdf:type         :Malware  .
```


nCircle

Relationship Propagation

- rdfs:Property

```
:hasBrother rdf:type rdfs:Property .
:hasSibling rdf:type rdfs:Property .
```

- rdfs:subPropertyOf

```
:hasBrother rdfs:subPropertyOf :hasSibling .
:alice          :hasBrother    :bob        .
```

we can infer the triple

```
:alice  :hasSibling  :bob .
```

AXIOM
IF
P rdfs:subPropertyOf R .
A P B .
THEN
A R B .

Class Membership through Relationships

- Similar to domain and range in math

```
:property_P rdfs:domain D .
:property_P rdfs:range  R .
```

- Example:

```
:usesSharedLib rdf:domain :Application .
:usesSharedLib rdf:range  :SharedLib   .
```

  - Assertion

```
:Apache :usesSharedLib :OpenSSL .
```

  - Inference

```
:Apache  rdf:type :Application .
:OpenSSL rdf:type :SharedLib   .
```

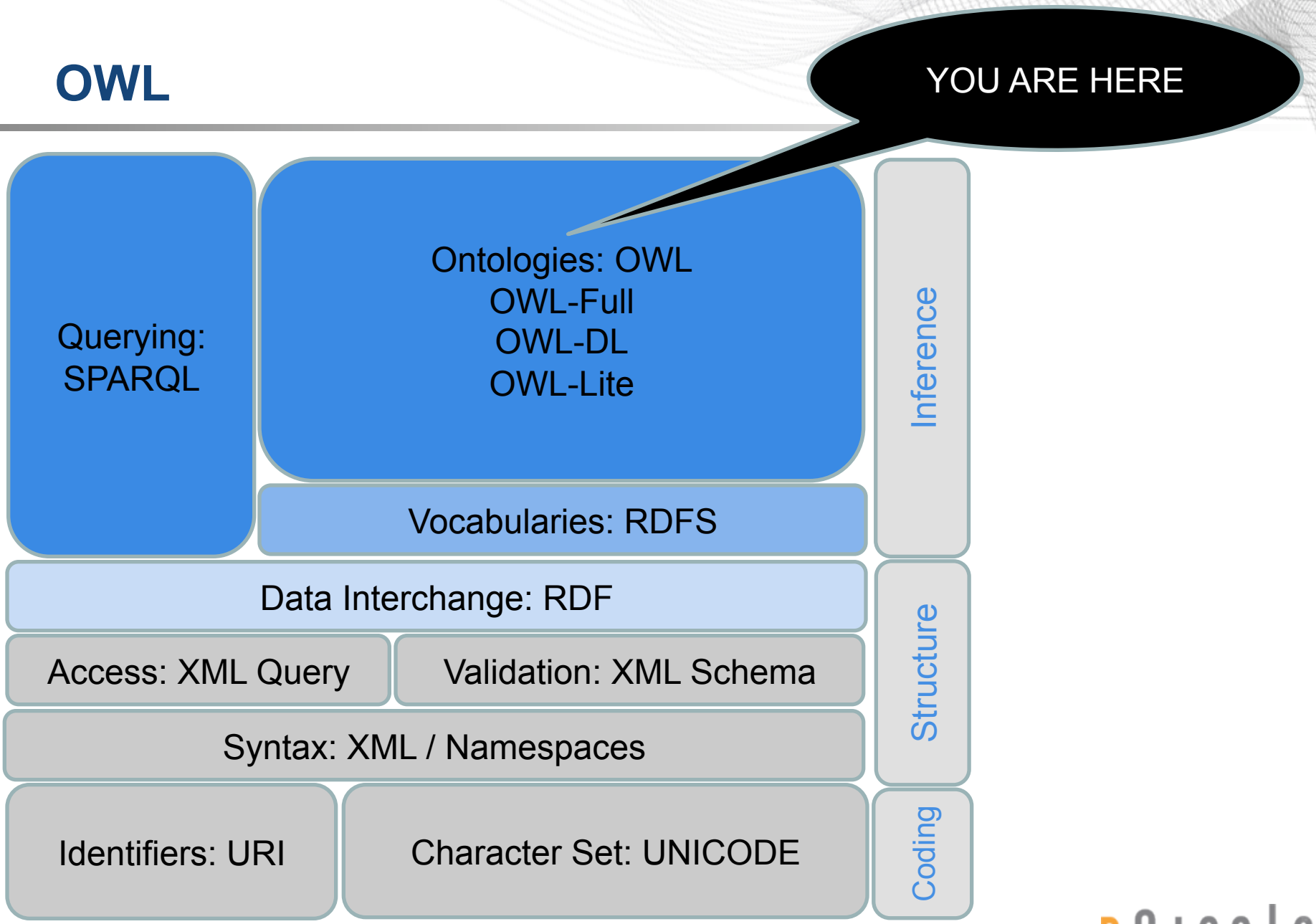
AXIOM (subject)
IF
P rdfs:domain D .
and
x P y .
THEN
X rdf:type D .

AXIOM (object)
IF
P rdfs:range R .
and
x P y .
THEN
y rdf:type R .

What are the limits to RDFS?

- RDFS may not have enough detail for your modeling
  - No localised range and domain constraints
    - Can't say that "the range of hasChild is person when applied to persons and elephant when applied to elephants"
  - No existence/cardinality constraints
    - Can't say that "all *instances* of person have a mother that is also a person", or that persons have exactly 2 parents
  - No transitive, inverse or symmetrical properties
    - Can't say that *isAncestorOf* is a transitive property
    - Can't say that *bundles* is the inverse of *isBundledBy*
    - Can't say that *isMarriedTo* or *isPeeredWith* is symmetrical

OWL

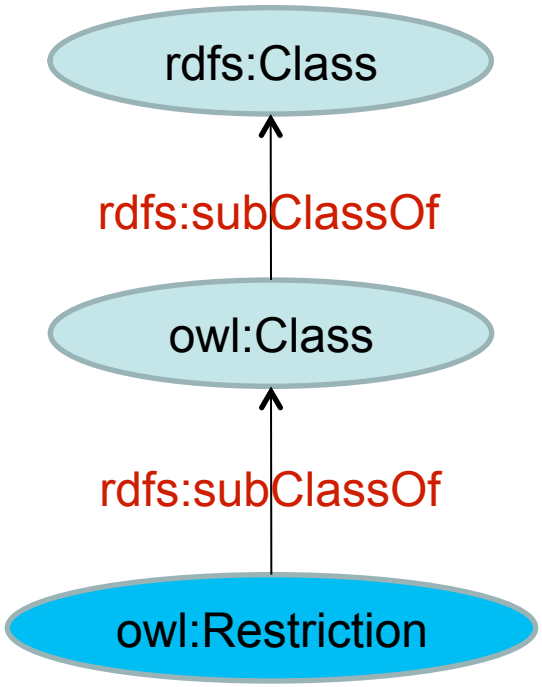


OWL

- OWL-Lite
  - Subset of OWL-DL
- OWL-DL & OWL-FULL
  - They use the same constructs
  - Objectives are different: Provability versus Executability
  - DL stands for Description Logics which is a First Order Logic
- What are _SOME_ things you do with OWL?
  - Anything that can be done with RDFS
  - Use inference for:
    - Classification – richer, more expressive than RDFS
      - Localized domain and range
    - Schema validation and constrains checking
      - Existence/Cardinality
    - Exploring network of relationships
      - Transitive/Inverse/Symmetrical

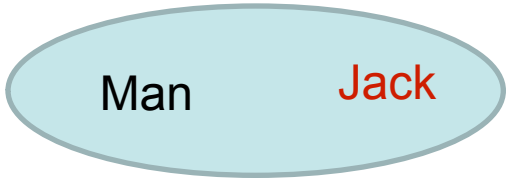
OWL

- OWL terms are defined in terms of old RDFS terms; RDFS terms are defined in terms of old RDF terms.

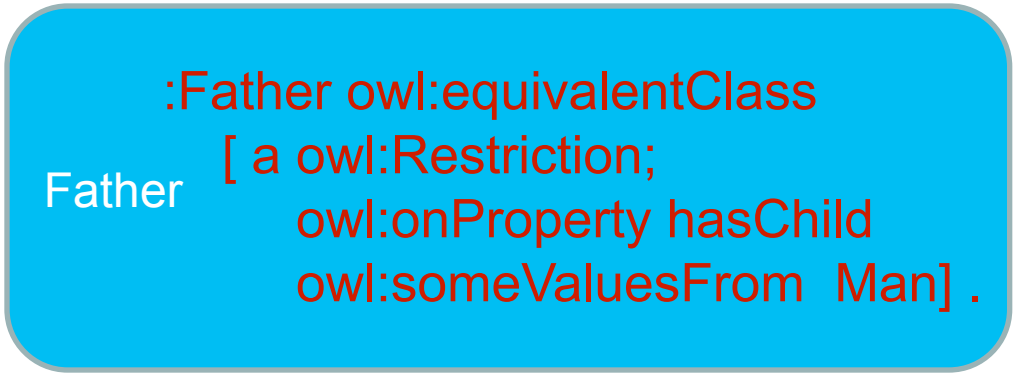


OWL Restriction

- owl:Restrictions allows you to describe a class in terms of other things we have already modeled
- Concept of Father: If a Man has a child, that man is a Father.



:jack rdf:type :Man .



Assert

:jack :hasChild :joe .

Infer

:jack rdf:type :Father .

A partial account of the OWL vocabulary

OWL	Example	Concept
someValuesFrom	hasChild someValuesFrom Man	Father
allValuesFrom	eats allValuesFrom VegetarianFood	Vegetarian
hasValue	hasCountryOfOrigin hasValue USA	American
minCardinality	hasChild min 3	--
cardinality	hasChild exactly 3	--
maxCardinality	hasChild max 3	--
intersectionOf	Doctor and Female	Female Doctor
unionOf	Man or Woman	Person
complementOf	not Client	Server
equivalentClass	WindowsXP equivalentClass WinXP	Equivalency
equivalentProperty	hasVuln equivalentProperty hasVulnerability	Equivalency
inverseFunctionalProperty	SSN rdf:type inverseFunctionalProperty	Identity

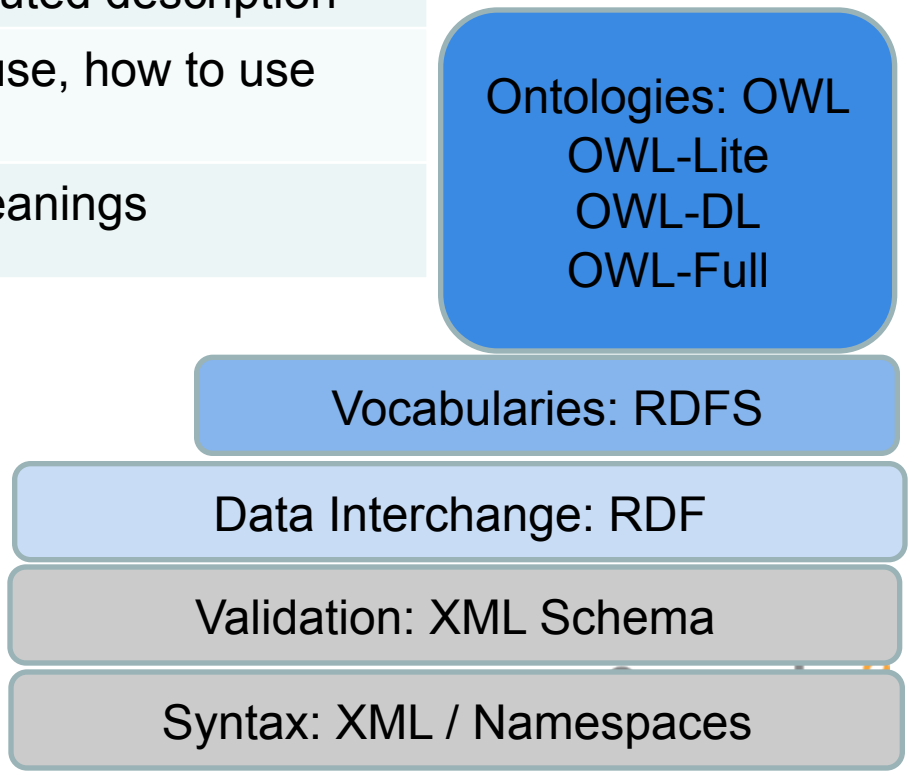
OWL: Managing Ontologies

OWL	Brief Description
DeprecatedClass	Specifies that the class is deprecated in a particular version (and should not be used)
DeprecatedProperty	Specifies that the property is deprecated in a particular version (and should not be used)
versionInfo	Annotation property for version info
priorVersion	Refer one ontology to another ontology that is a prior version
backwardCompatibleWith	Like *priorVersion* but further states the new ontology is backward compatible with the previous one
inCompatibleWith	Like *priorVersion* but further states the new ontology is incompatible with the previous one
Imports	Allows one ontology to refer explicitly to another

nCircle

Progressive Levels of Expressivity

Solution	Issue
OWL	Define logical constraints for entities and relationships
RDFS	Provide inference about types and inclusion
RDF	Identify items for distributed description
XML Schema	Describe what tags to use, how to use them (syntax)
XML Namespaces	Same word has two meanings



nCircle

Quick introduction to Useful Ontologies

Meaning and Inference

- External agreement on meaning of annotations
  - E.g., Original XML Dublin Core
    - Agree on the meaning of a set of annotation tags
  - Challenges
    - Inflexible
    - Limited number of things can be expressed
- Use Ontologies to specify meaning of annotations
  - Ontologies provide a vocabulary of terms
  - New terms can be formed by combining existing ones
  - Meaning (semantics) of such terms is formally specified
  - Can also specify relationships between terms in multiple Ontologies (important to SCAP)

A few examples

- FOAF (Friend of a Friend)
  - Describe yourself and the people you know
- EARL (Evaluation and Report Language)
  - Describes test results
- Knock yourself out!
  - http://www.schemaweb.info/default.aspx

Friend of a Friend (FOAF)

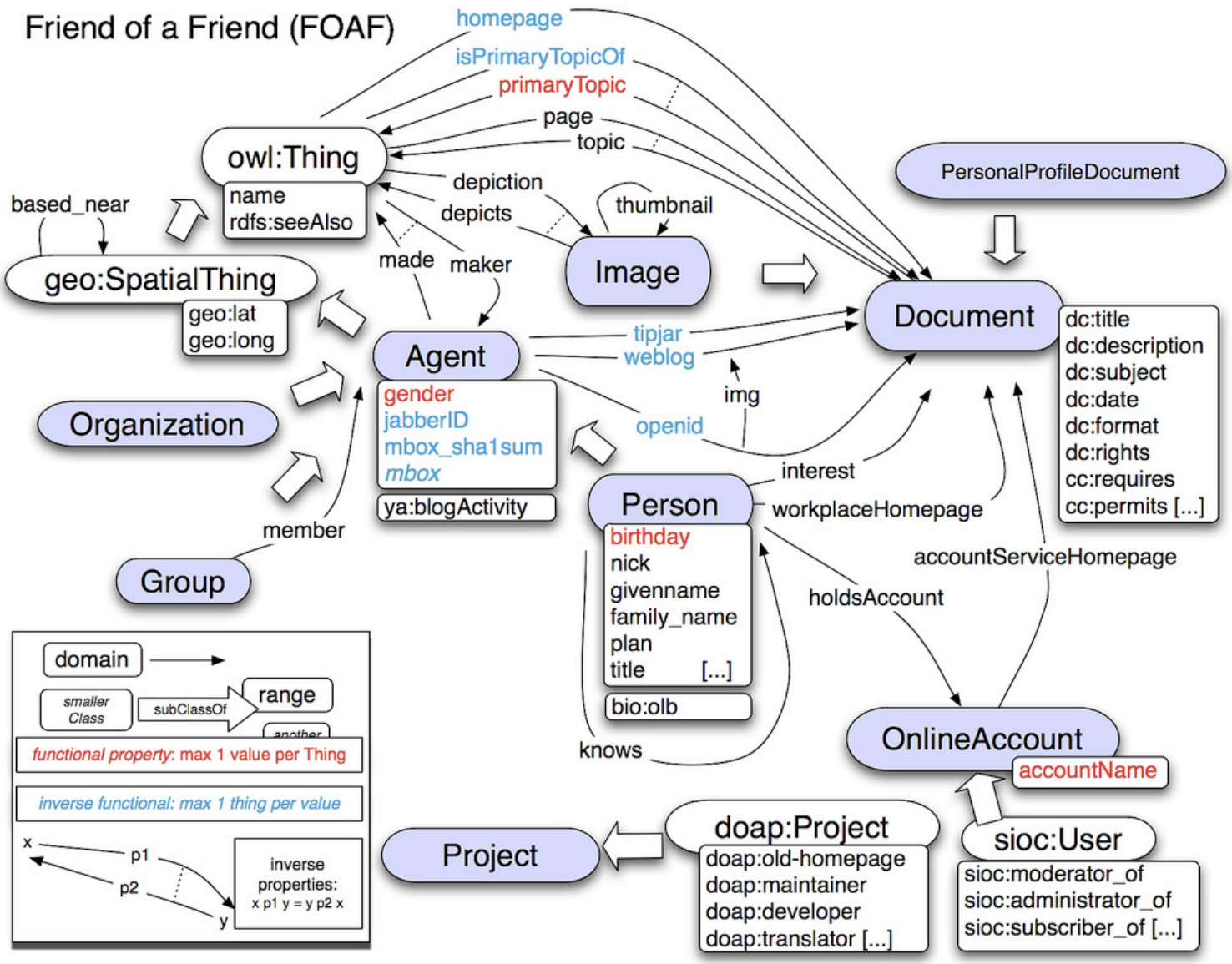
- Ontology describing persons, their activities, and their relationships to other people and objects
- Specification
  - http://xmlns.com/foaf/spec/
- Makes use of RDF and OWL

"FOAF is an application of the Resource Description Framework (RDF) because the subject area we're describing -- people -- **has so many competing requirements that a standalone format could not do them all justice**. By using RDF, FOAF gains a **powerful extensibility mechanism**, allowing FOAF-based descriptions can be **mixed with claims made in *any other RDF vocabulary*"**

Goals

- Goal is not to replace existing systems but to provide framework for distribution, interoperability, and extensibility
- FOAF provides a small number of classes and properties as a starting point
  - New statements can be made by relating them to statements that have already been made
- Standard maintained by a committee but
  - It does not prescribe how to represent things
  - It does provide a means to transfer one presentation to another

FOAF Overview



nCircle

FOAF Introduction

- Example in TopBraid Composer
  - Very stable concepts
    - Person, Name, mbox, img
  - Makes use of other vocabularies
    - Dublin Core Elements, Terms, & Abstraction Model
    - WGS84 geo
    - xsd,rdf,rdfs,owl
    - Web of Trust Ontology (wot)
    - Semantic Web Vocabulary Status Ontology

nCircle

EARL (Evaluation and Reporting Language)

- http://www.w3.org/WAI/ER/EARL10/WD-EARL10-Guide-20090422
- http://www.w3.org/TR/EARL10-Schema/
- http://www.w3.org/WAI/ER/EARL10/schema.rdf
- Description:
  - The Evaluation and Report Language (EARL) is a **machine-readable format for expressing test results**. The primary motivation for developing EARL is to facilitate the processing of test results, such as those generated by Web accessibility evaluation tools, using a vendor-neutral and platform-independent format.

What EARL is not

- EARL is not a comprehensive vocabulary for describing
  - test procedures
  - test criteria
  - test requirements
- EARL describing the outcomes from such testing.
- EARL can be supplemented by
  - test description vocabularies
  - other vocabularies for different aspects of the testing cycle.

nCircle

RDF Triples in EARL

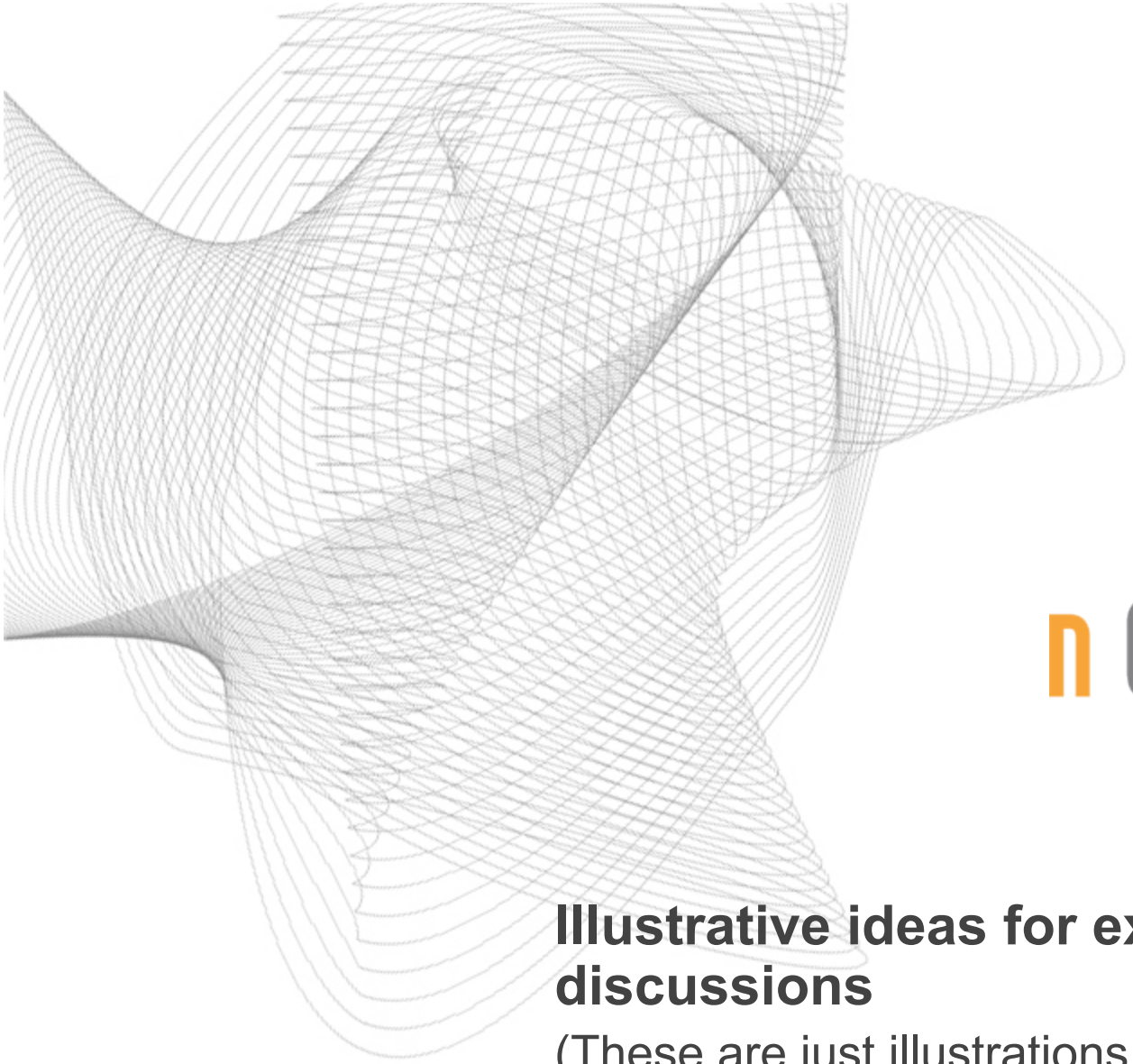
```
<subject>          <predicate>          <object> .
-------------------------------------------------------------
<#someone>         <#checks>            <#resource> .
<#resource>        <#fails>             <#test>     .
```

- **Who** (or which tool) runs a test: this is known in the EARL terminology as the **Assertor**.
- The **resource** tested: known as the **Test Subject**.
- The tested **criterion**: known as the **Test Criterion**.
- The **result** of the test: known as the **Test Result**.

nCircle

Imports other Ontologies

- DC - Dublin Core
- FOAF -- Friend of a Friend
- Content-RDF -- Content in RDF
  - Vocabulary for representing any type of content
- HTTP-RDF – HTTP Vocabulary in RDF
  - Vocabulary for HTTP
- Pointers-RDF – Pointer Methods in RDF
  - Ability to point to particular parts within a HTML or XML document



nCircle

Illustrative ideas for exploratory discussions

(These are just illustrations of what RDF/RDFS/ OWL can do for you)

#1: Windows OS Naming

- Requirements
  - Globally Unique Identifier for Classes and Properties
  - Remain compatible with the current CPE specification
  - Ability to articulate these properties (relationships)
    - codeBaseDerivedFrom
    - asSeenInAPIas
      - asSeenInWMIas
      - asSeenInSNMPas
    - Some unforeseen property that is meaningful to someone
  - Ability to categorize classes and individuals by:
    - Windows
    - OS
    - Server or Client
    - Compliant or Not-Compliant
    - Some unforeseen category that is meaningful to someone

nCircle

#1: Windows OS Naming Illustration

- A URI for the CPE Identity
  - http://nvd.nist.gov/cpe/3.0/cpe234141
  - This would operate much like a UPC, ISBN, or UDDI.
  - Using FOAF as an analogy, this is the inverse-functional-property of the person like their homepage.
- A URI for Ontologies (terminological metadata)
  - http://nvd.nist.gov/cpe/marketing/
  - http://nvd.nist.gov/cpe/technical/
  - http://nvd.nist.gov/cpe/compatible/
  - This would offer up the NameSpace URI to be mapped to a local prefix
  - Much like FOAF itself defining the core Classes and Properties
- Vendors could then self-publish their own Ontology to NIST extended from a core
- Non-SCAP communities could extend the CPE-core ontology to their own vocabulary and inferences

nCircle

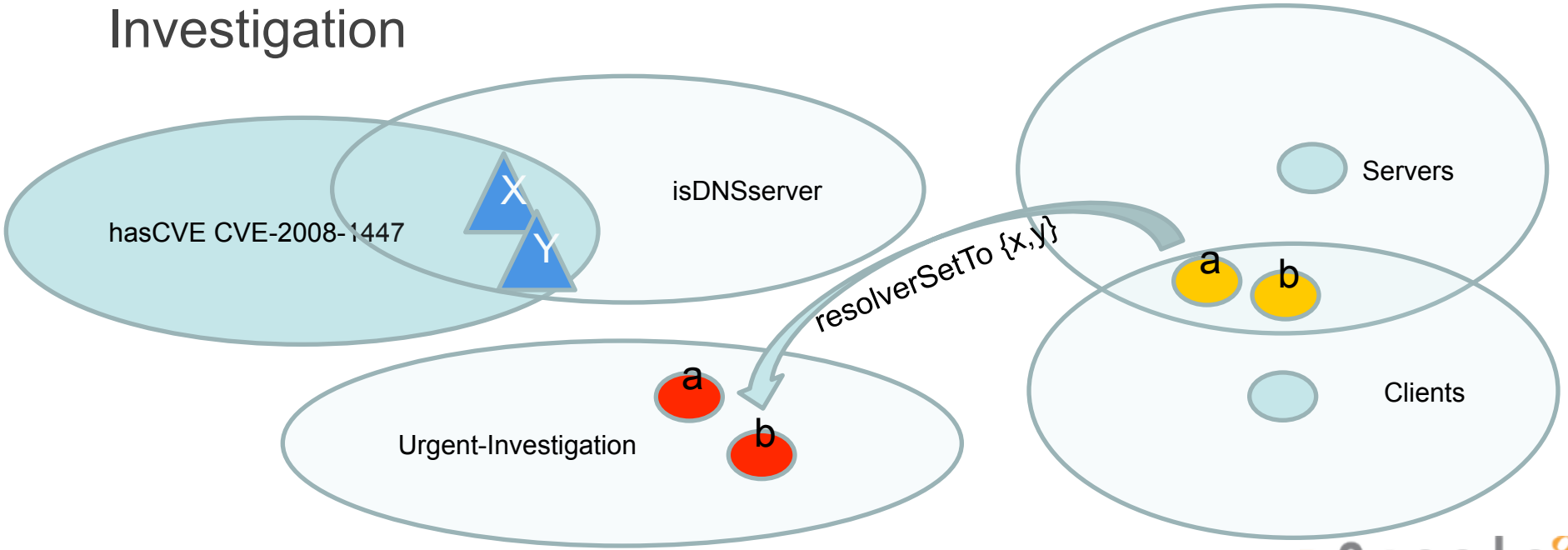
#2: Complex Vulnerability Representation

- Playbook *(thanks to jgraver and treguly)*
  - Attacker PushExploitTo Windows Web Server in DMZ
  - WindowsWebServer isExploitedWith MS08-067
  - WindowsWebServer hasPrivateConnectionto Int-SQLServer
  - Int-SQLServer isExploitedWith MS09-004
  - Int-SQLServer floodsNetworkWith Web-Proxy-Auto-Detect (WPAD)
    - updates for a MaliciousProxy
  - WebClients PullExploitsFrom MaliciousProxy
  - Attacker OwnsEveryClientOn theNetwork
- This model could be done in owl such that an assertion in any part of the graph could check the other relationships and classes for evidence or flag the risk.

nCircle

#3: Change in feasibility for an entire class of attacks

- DNS Cache Poisoning
  - CVE-2008-1447
- If a Server is a DNS server, and has CVE-2008-1447; assign client or server who has resolvers pointing at members X/Y (in this case a/b) to a class called Urgent-Investigation



nCircle

Resources

IDEs

- TopBraid Composer™
  - http://www.topquadrant.com/products/TB_download.html
  - 30-DAY TRIAL
- Protégé 4.0
  - http://protege.stanford.edu/
  - Open-Source Ontology Editor and knowledge-base framework
- Both of these IDE's are base on the Eclipse Foundation

Tools available

- Parsers
  - Jena (Java)
    - http://jena.sourceforge.net/
    - http://jena.sourceforge.net/tutorial/RDF_API/
  - Redland/Raptor (C)
  - cwm (coom) general purpose data processor
    - http://www.w3.org/2000/10/swap/doc/cwm.html
- Google for a parser in your favorite language

RDF Stores

- RDF Stores (triple-stores)
  - Oracle
    - http://www.oracle.com/technology/tech/semantic_technologies/index.html
  - Franz Inc AllegroGraph
    - http://www.franz.com/
  - Sesame
    - http://www.openrdf.org/
- D2R Server
  - Publishing RDBS via RDF and SPARQL Clients
    - http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/

Websites

- http://planetrdf.com/guide/
- http://semanticuniverse.com/
- http://composing-the-semantic-web.blogspot.com/
- http://dallemang.typepad.com/

- Why RDF is different from XML (Sept 1998)
  - http://www.w3.org/DesignIssues/RDF-XML

Tutorials

- 2009 Semantic Technology Conference
  - June 14-18 San Jose, CA
  - http://www.semantic-conference.com/
- Franz Inc Web Seminars
  - http://www.franz.com/agraph/services/conferences_seminars/index.lhtml#recorded-web-seminars
- Semantic Universe Webinars
  - http://www.semanticuniverse.com/learning.html
- W3C Website
- Description Logics – Courses and Tutorials
  - http://dl.kr.org/courses.html
- Random tutorial on RDF
  - http://realtech.burningbird.net/semantic-web/rdf-and-rdfa/bottoms-rdf-tutorial

nCircle

Books

- Semantic Web for the Working Ontologist
  - http://workingontologist.org/
- Semantic Web – Concepts, Technologies and Applications
  - Breitman, Casanova, Truszkowski
- Ontological Engineering
  - Gomez-Perez, Fernadez-Lopez, Corcho
- Model Driven Architecture and Ontology Development
  - Gasevic, Djuric, Devedzic
- Enabling Semantic Web Services
  - Fensel, Lausen, Polleres, etc

Thank You

tk@ncircle.com

Thanks to these people for my education on this domain:
Dean Allemang, Jim Hendler, Holger Knublauch, Jans Aasman, Irene Polikoff, Ralph Hodgson, Scott Henninger, Jeremy Carroll, Peter F. Patel-Scheider, Dan Brickley and everyone else…