

YarcData™

A DIVISION OF CRAY INC.



uRIKA and Graph Analytics

uRIKA == universal RDF integration Knowledge Appliance

Agenda



A Quick Overview



Converting Data into RDF



Ingesting Data into uRIKA



Querying uRIKA



Visualizing Results



Summary

uRiKA – YarcData Hardware and Software

● Cray Hardware Engine

- Originally designed for deep analysis of large datasets
- Very large *scalable* shared memory
 - Architecture can support 512TB shared memory
 - Typical systems are 2 TB to 32 TB
- Multithreading
 - Unique highly multithreaded architecture
 - 128 hardware threads per processor
 - Extreme parallelism, hides memory latency

● Multithreaded Graph Database

- Highly parallel in-memory RDF quad store
- High performance inference engine
- High performance parallel I/O

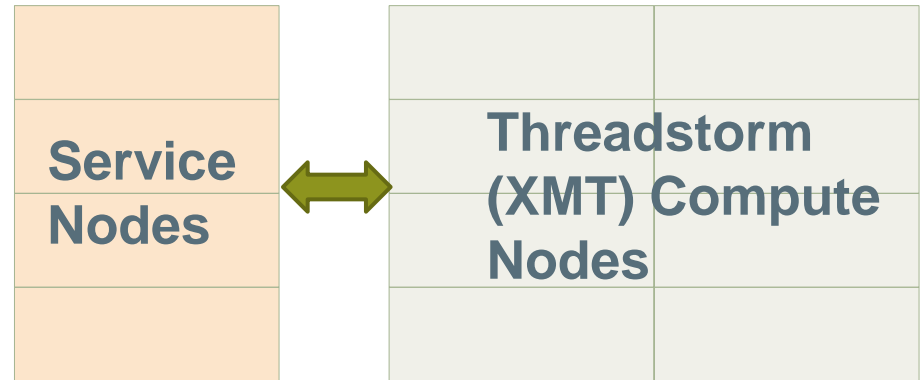
● Industry Standard Front End

- Based on Jena open source semantic DB
- WS02 application framework
- All standard SuSE Linux infrastructure and languages

Hybrid Graph Appliance: Ease of Use AND Performance!



- Proven Cray infrastructure
- Cray XT5 3D Torus High Speed Interconnect

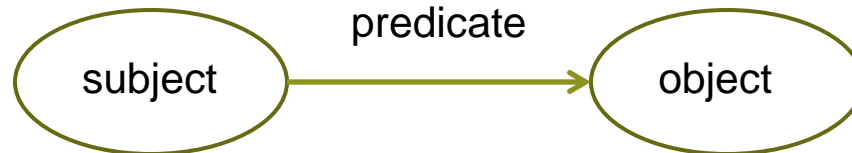


- **Service Nodes:**
 - AMD Opteron Processors, SuSe Linux
 - Open environment based on Jena
- **Threadstorm Processors:**
 - Multithreaded processors, MTK OS
 - Pre-programmed by YarcData

Emerging Web 3.0 Standards: **RDF** and SPARQL

● Resource Description Framework (RDF)

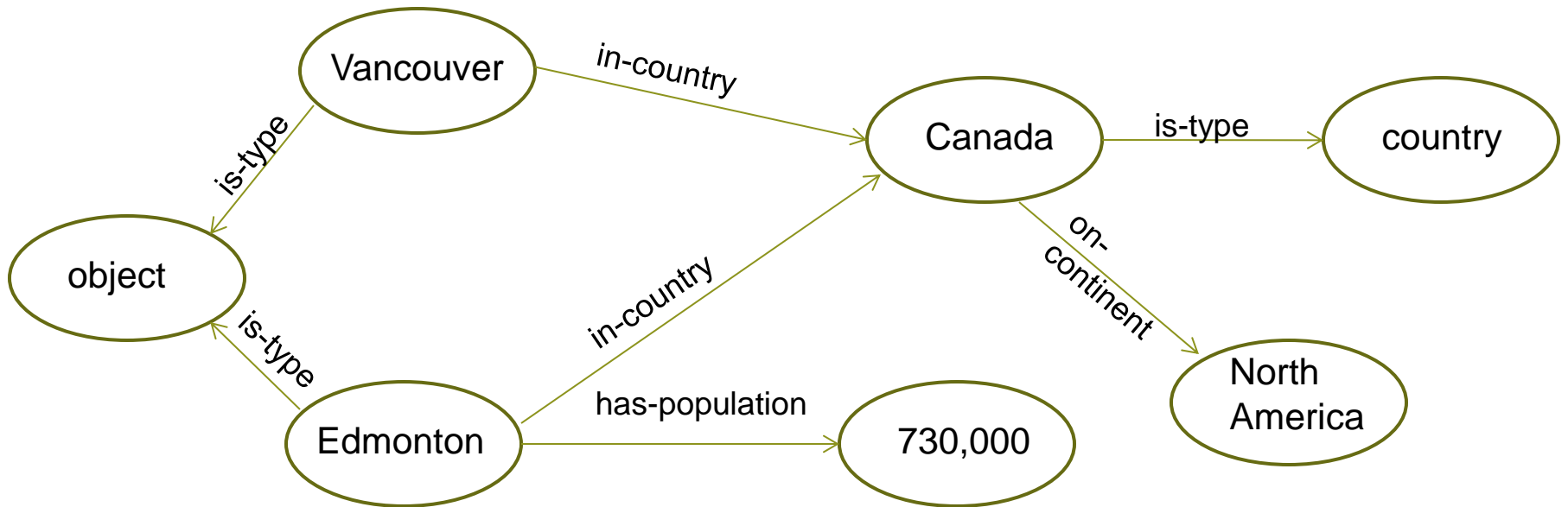
- Designed to enable semantic web searching and integration of disparate data sources
- W3C standard formats
- Every datum represented as subject/predicate/object
 - Ideally with each of those expressed with a URI
- Standard ontologies in some domains
 - *e.g.*, Open Biological and Biomedical Ontologies (OBO)
- Examples:



<ncbitax:NCBITaxon_840261>	rdf:type	owl:Class
<ncbitax:NCBITaxon_195644>	rdfs:subClassOf	<ncbitax:NCBITaxon_185881>
<ncbitax:NCBITaxon_816681>	rdfs:label	"Characiformes sp. BOLD:AAG5151"@en

Semantic Database of RDF Triples

- RDF triples databases are inherently graphical
- Some researchers call semantic databases “semantic graph databases”



Emerging Web 3.0 Standards: RDF and SPARQL

- **SPARQL Protocol and RDF Query Language (SPARQL)**
 - Enables matching of graph patterns in the semantic DB
 - Reminiscent of SQL

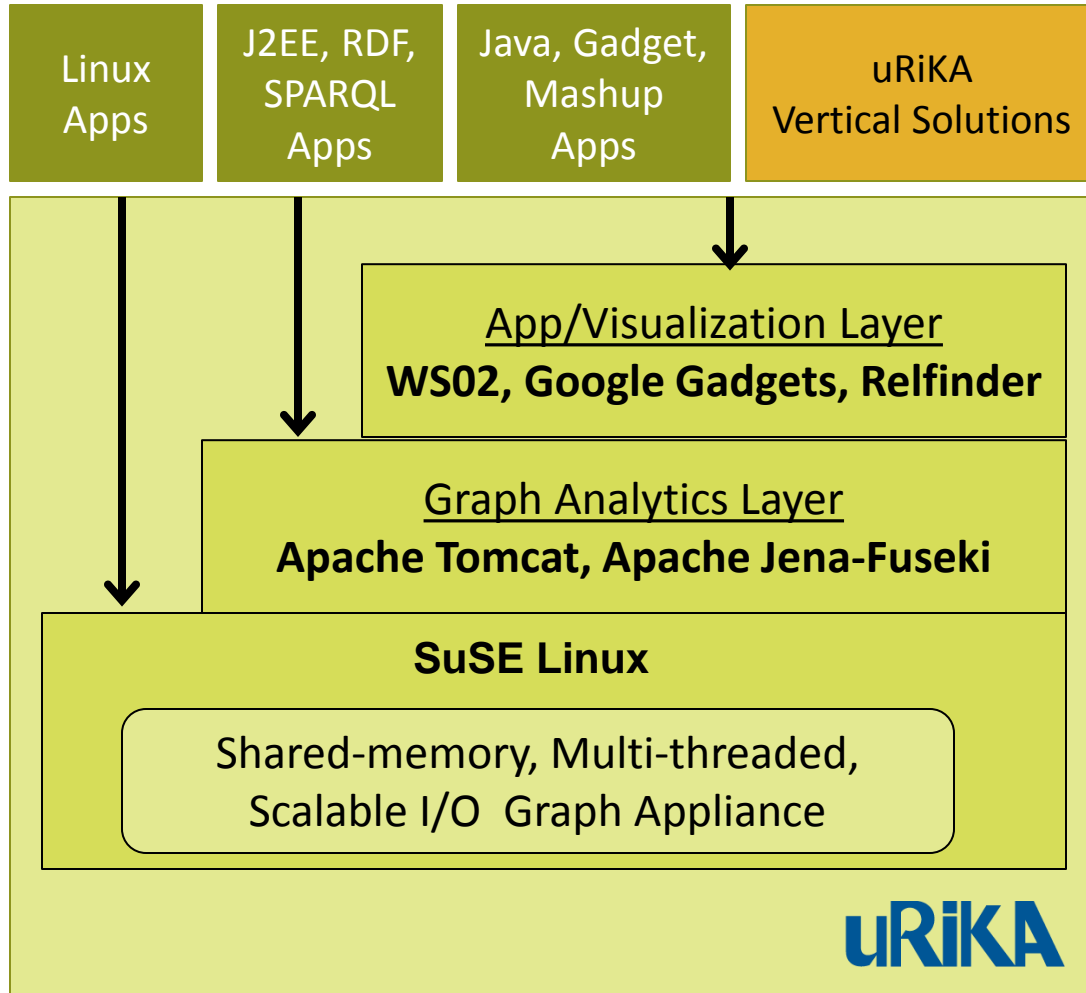
```
# Lehigh University BenchMark (LUBM) Query 9
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ub: <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#>
SELECT ?X, ?Y, ?Z
WHERE
{?X rdf:type ub:Student .
 ?Y rdf:type ub:Faculty .
 ?Z rdf:type ub:Course .
 ?X ub:advisor ?Y .
 ?Y ub:teacherOf ?Z .
 ?X ub:takesCourse ?Z}
```

PREFIX == shorthand for a URI

variables to be returned from the query

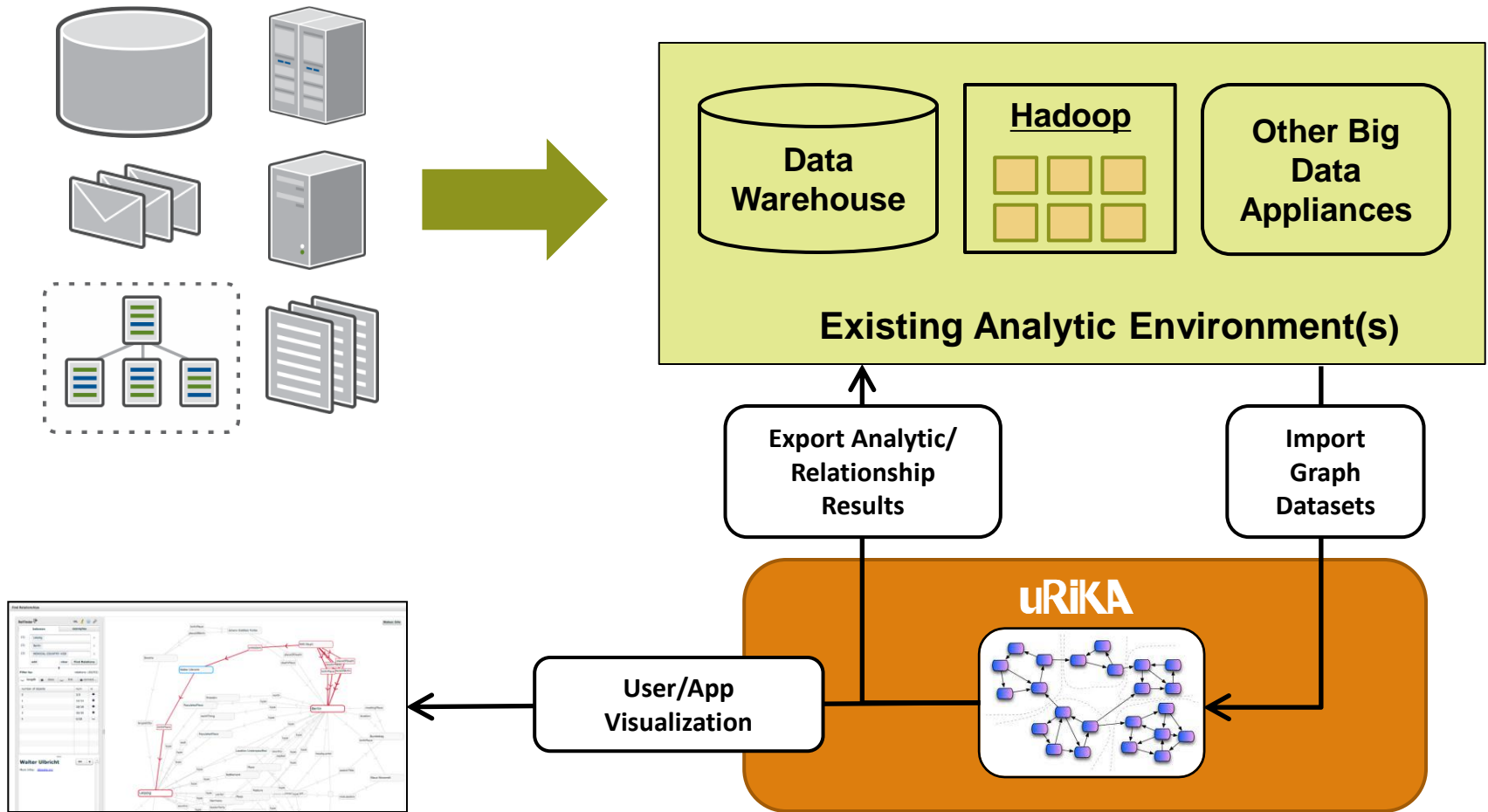
“find sets of (X, Y, Z) with a subject X of type Student, a subject Y of type Faculty, and a subject Z of type Course, where X is an advisee of Y, Y teaches course Z, and X takes course Z”

uRiKA Software Stack

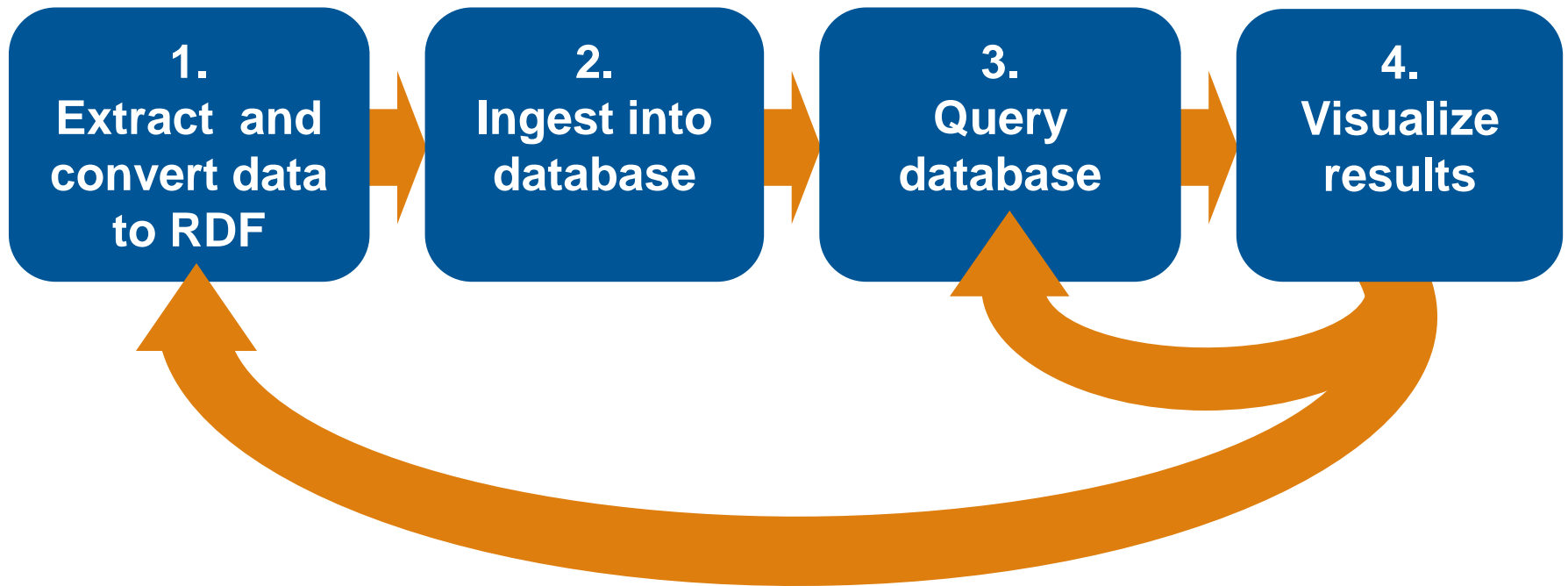


- **Industry-standard, Open-source Software Stack**
 - Linux, Java, Apache, WS02, Gadgets, Mashups...
- **Reusable Existing Skillsets**
 - OSGI, App Server, SOA, ESB, Web toolkit...
- **No Lock-in**
 - All applications and artifacts built on uRiKA can be run on other platforms

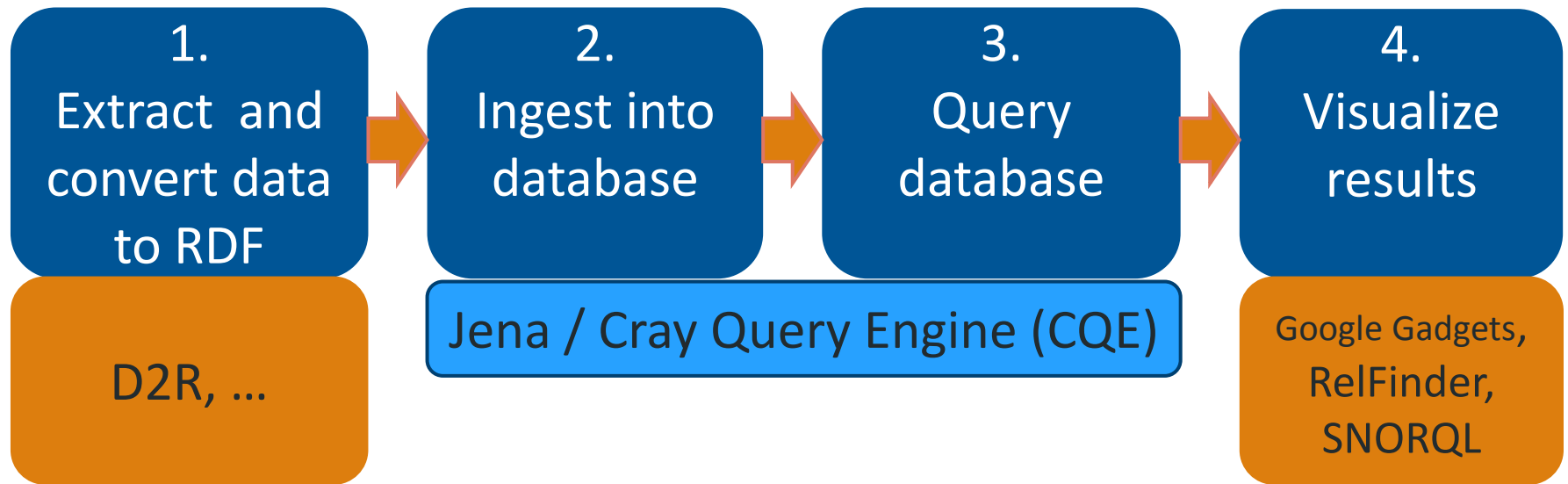
uRIKA complements existing Data Warehouse/Hadoop environment by offloading Graph Analytics



Workflow



Implementation



Agenda



A Quick Overview

Converting Data into RDF

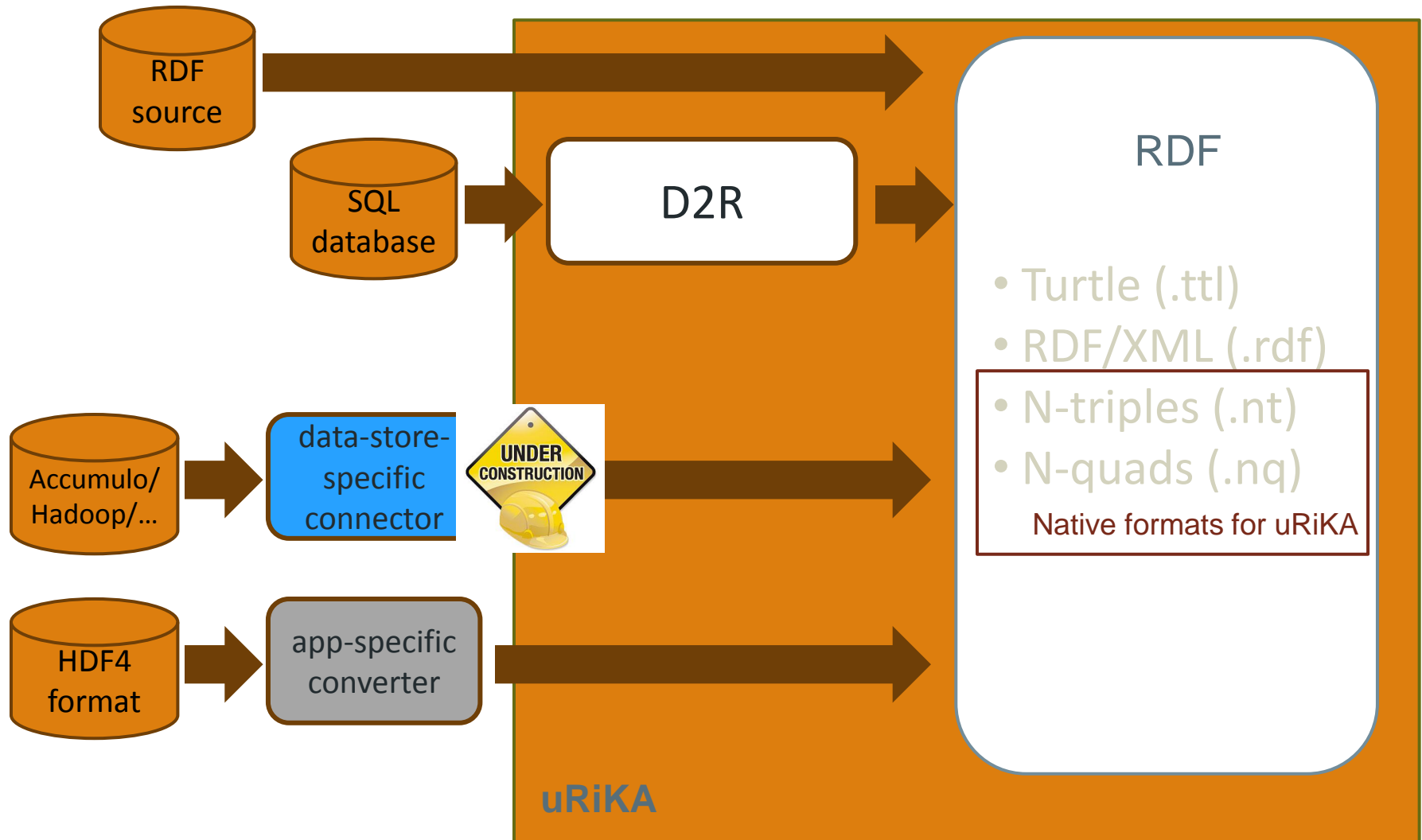
Ingesting Data into uRIKA

Querying uRIKA

Visualizing Results

Summary

Getting Data into RDF Format



... But Conversion is Not Enough

- Primary RDF goal is to be able to fuse data from different sources
- **Ontologies (definitions of entities and relationships) must either be the same or be mappable onto each other**
- **Same: Use common ontologies**
 - Generic: *e.g.*, RDF, OWL, Dublin Core
 - Social Networks: FOAF
 - Biology: *e.g.*, OBO, NCBI
- **Map: Use ontology-mapping tools (*e.g.*, Top Braid Composer)**



Converting Data among Different RDF Formats

- All files must be converted into .nt/.nq format before being Loaded into CQE
- In the fullness of time, the admin UI will convert from other formats as part of making the SDB
- Today, we use various conversion tools from the command line
 - rdf2rdf: Java tool, see <http://www.l3s.de/~minack/rdf2rdf/>
 - RIOT: runnable via bash scripts, see <http://incubator.apache.org/jena/documentation/io/riot.html>

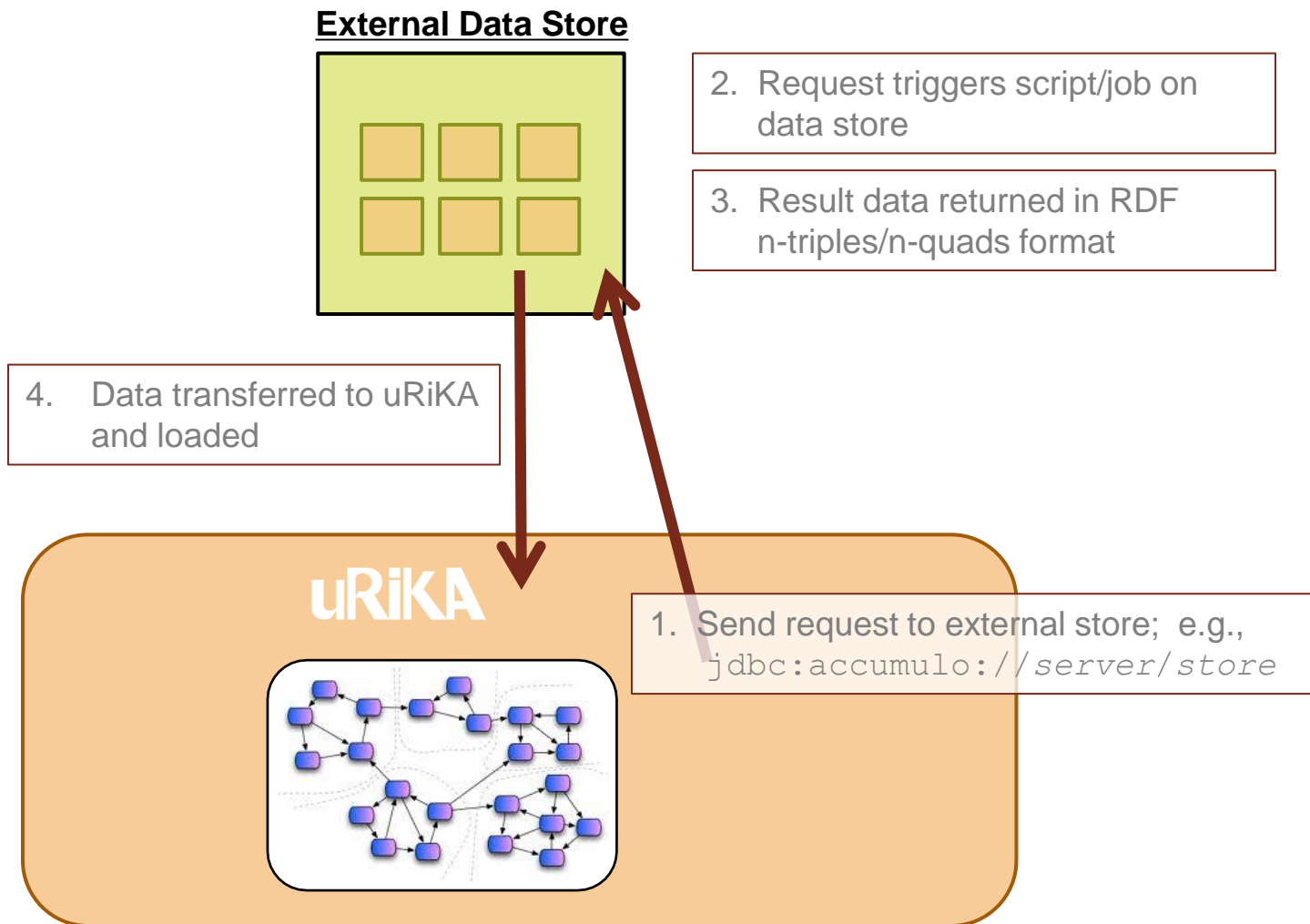
Extracting Data from an SQL Database

● Use the D2R Server*

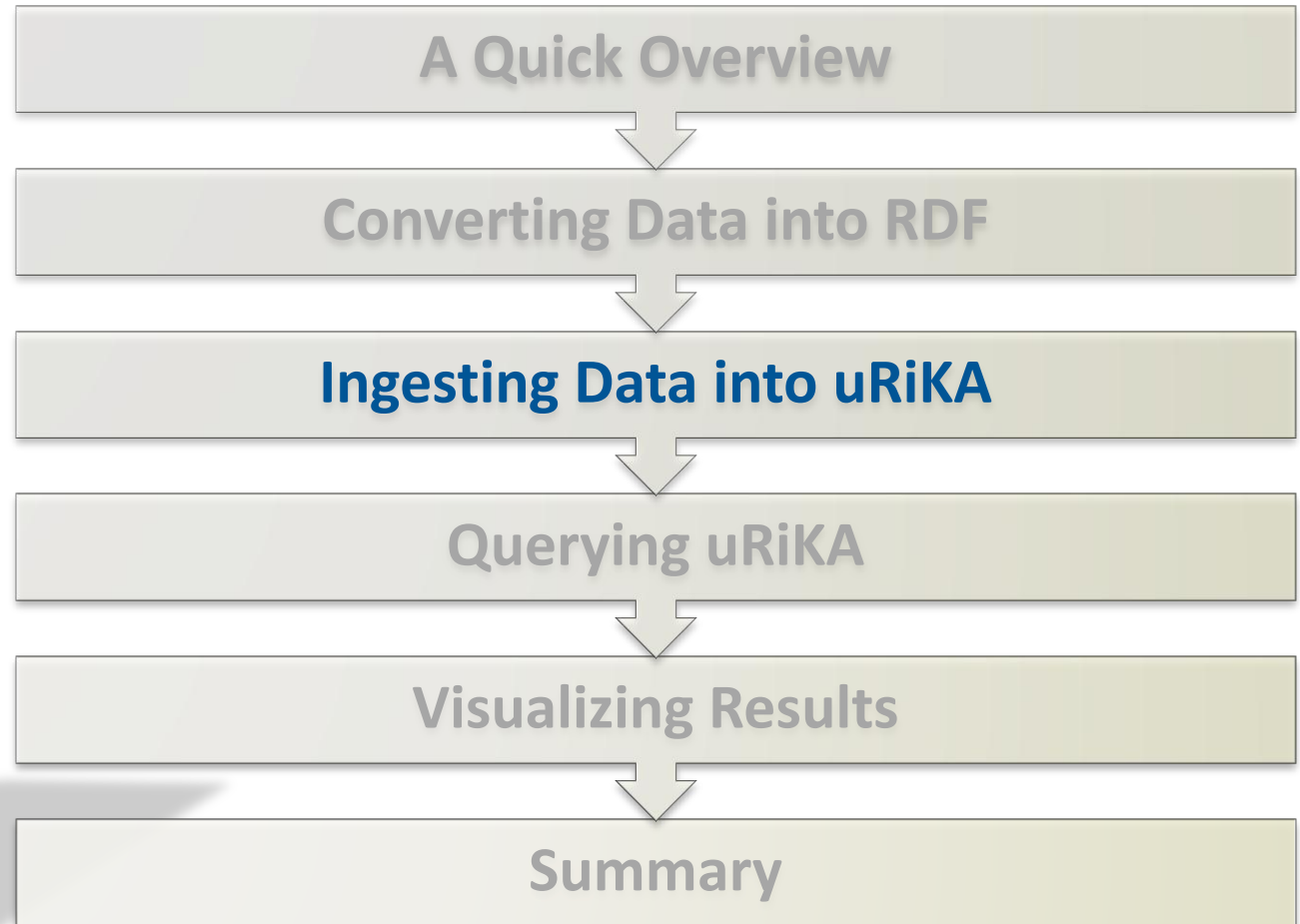
- D2R can be used for an RDB in place; we extract the data once for ingestion into uRiKA
- See [D2R Server Data Extraction Configuration Guide](#) for details
- Two-step process
 - Generate a mapping from the target database
 - e.g., `generate-mapping -d jdbc:mysql://server/database -o map.n3`
 - Many SQL DB types supported by D2R; MySQL, Oracle, and Postgres used so far with uRiKA
 - Edit the mapping file if (as typical) not all tables/fields are needed
 - Extract the data into an RDF file
 - e.g., `dump-rdf -m map.n3`
 - Move the result file into Lustre

* <http://www4.wiwiss.fu-berlin.de/bizer/d2rq/spec/#specification>

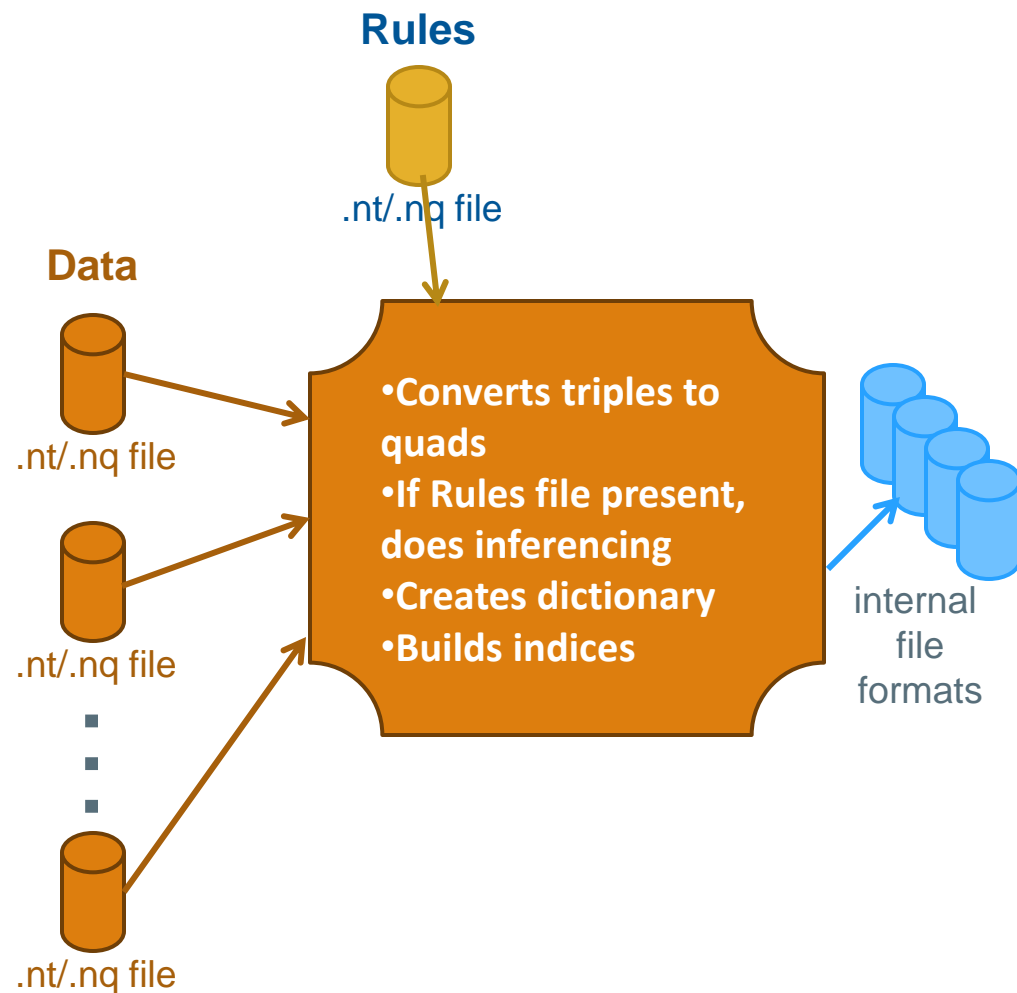
Importing Data from Data Stores e.g., Hadoop, Accumulo



Agenda

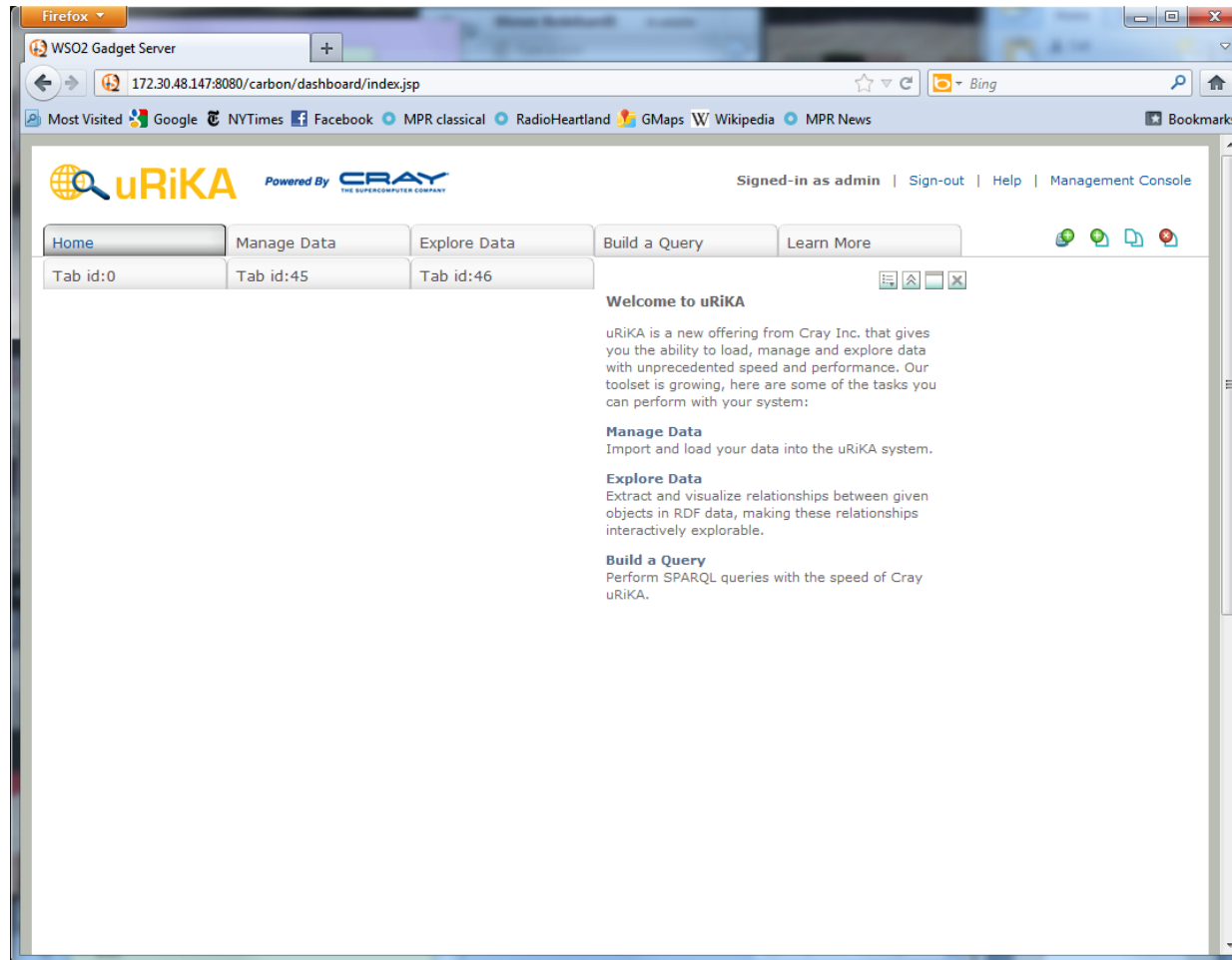


Ingestion Data Flow



- One or more .nt files (already resident on Lustre) can be combined into a single SDB
- A rules file may also be provided, which causes the ingestion step to run inferencing
- The output of the ingestion step is ~10 files in internal binary formats
- Those files are loaded directly into a uRiKA instance when it's initiated

UI – Home



Ingestion UI – Import Data

The screenshot shows a Firefox browser window displaying the uRiKA web interface. The browser's address bar shows the URL `172.30.48.147:8080/carbon/dashboard/index.jsp`. The page header includes the uRiKA logo, the text "Powered By CRAY THE SUPERCOMPUTER COMPANY", and the user status "Signed-in as admin | Sign-out | Help | Management Console".

The main navigation bar contains buttons for "Home", "Manage Data", "Explore Data", "Build a Query", and "Learn More". Below this, there are three tabs labeled "Tab id:0", "Tab id:45", and "Tab id:46".

The "Import" section is active, with sub-links for "Build", "Load", "Delete", and "Status". A warning message states: "Choose the type of data you want to import. ⚠ To import from a relational database (RDBMS) [click for instructions](#)." Below this, there are three tabs: "Structured" (selected), "Unstructured", and "Rules".

The "Structured Data File" section contains the following form elements:

- "Choose your data source:" with a dropdown menu set to "Local File".
- A "Local File" input field with a "Browse..." button.
- A warning icon and text: "⚠ Triple or Quad files (.nt or .nq) are required."
- A "Name Your Data File" input field with an "Import" button.

Ingestion UI – Import Rules

The screenshot shows a Firefox browser window displaying the uRiKA web application. The browser's address bar shows the URL `172.30.48.147:8080/carbon/dashboard/index.jsp`. The page header includes the uRiKA logo, the text "Powered By CRAY THE SUPERCOMPUTER COMPANY", and the user status "Signed-in as admin | Sign-out | Help | Management Console". A navigation bar contains buttons for "Home", "Manage Data", "Explore Data", "Build a Query", and "Learn More". Below the navigation bar, there are three tabs labeled "Tab id:0", "Tab id:45", and "Tab id:46".

The main content area is titled "Import | Build | Load | Delete | Status". It contains a warning message: "Choose the type of data you want to import. ⚠ To import from a relational database (RDBMS) [click for instructions.](#)". Below this message are three buttons: "Structured", "Unstructured", and "Rules". The "Rules" button is currently selected.

Under the "Rules" button, there is a section titled "Inferencing Rules". It contains a warning message: "Choose the inferencing rules you want to import. ⚠ For instructions on creating Rules [click here.](#)". Below this message are two input fields: "Local File" with a "Browse..." button, and "Name Your Rules Set" with an "Import" button.

Ingestion UI – Build

- Can select which (Lustre-resident) files to combine into a knowledge base

The screenshot shows the uRiKA Ingestion UI in a Firefox browser window. The page title is "uRiKA Powered By CRAY THE SUPERCOMPUTER COMPANY". The user is signed in as "admin". The main navigation bar includes "Home", "Manage Data", "Explore Data", "Build a Query", and "Learn More". Below the navigation bar, there are three tabs: "Tab id:0", "Tab id:45", and "Tab id:46". The "Build Knowledgebase" form is the central focus. It contains a warning message: "Knowledgebases cannot be built when a LOAD is in progress." Below this is a table with two columns: "Name" and "Date and Time". The table lists several datasets with checkboxes for selection. At the bottom of the form, there is a checkbox for "Enable Inferencing", a text input field for "Name Your Knowledgebase", and "Cancel" and "Build" buttons.

Name	Date and Time
<input type="checkbox"/> dataset	02/02/2012 2:39AM
<input type="checkbox"/> nasa_dataset	02/01/2012 9:46PM
<input type="checkbox"/> mondial-nala8443	01/05/2012 9:36PM
<input type="checkbox"/> dbpedia_duisburg_essen	01/05/2012 9:36PM
<input type="checkbox"/> dbpedia_einstein_stuttgart	01/05/2012 9:36PM
<input type="checkbox"/> dbpedia_leipzig_berlin	01/05/2012 9:36PM
<input type="checkbox"/> lubm0	01/05/2012 9:36PM
<input type="checkbox"/> inputData0142	01/05/2012 9:36PM

Ingestion UI – Load

The screenshot shows the uRIKA Ingestion UI in a Firefox browser window. The browser address bar shows the URL `172.30.48.147:8080/carbon/dashboard/index.jsp`. The page header includes the uRIKA logo, the text "Powered By CRAY THE SUPERCOMPUTER COMPANY", and the user is signed in as "admin". The main navigation bar has buttons for "Home", "Manage Data", "Explore Data", "Build a Query", and "Learn More". Below the navigation bar, there are three tabs labeled "Tab id:0", "Tab id:45", and "Tab id:46". The main content area has a menu with "Import", "Build", "Load", "Delete", and "Status". The "Load" option is selected, and a dialog box titled "Load Knowledgebase" is open. The dialog box contains the following text: "Select the Knowledgebase you want to load into your uRIKA system." and a warning: "Knowledgebases cannot be loaded when a BUILD is in progress." Below this is a table with two columns: "Name" and "Date and Time". The table lists three knowledgebases: "dbpedia+mondial+lubm0", "MAYO_POC", and "NASA_POC". The "MAYO_POC" knowledgebase is selected with a radio button. At the bottom of the dialog box are "Cancel" and "Load" buttons.

Name	Date and Time
<input type="radio"/> dbpedia+mondial+lubm0 Data Set Rule Set	02/02/2012 3:08PM
<input checked="" type="radio"/> MAYO_POC	02/02/2012 4:24PM
<input type="radio"/> NASA_POC	02/02/2012 1:53AM

- Load starts the SDB instance (after stopping a current instance)

UI – Learn More

WSO2 Gadget Server

172.30.48.147:8080/carbon/dashboard/index.jsp

Most Visited Google NYTimes Facebook MPR classical RadioHeartland GMaps Wikipedia MPR News

Powered By **CRAY** THE SUPERCOMPUTER COMPANY

Signed-in as admin | Sign-out | Help | Management Console

Home Manage Data Explore Data Build a Query **Learn More**

Tab id:0 Tab id:45 Tab id:46

Quick Start Installation and Configuration Guide

The following quick start guide will help get you up and running with uRiKA.

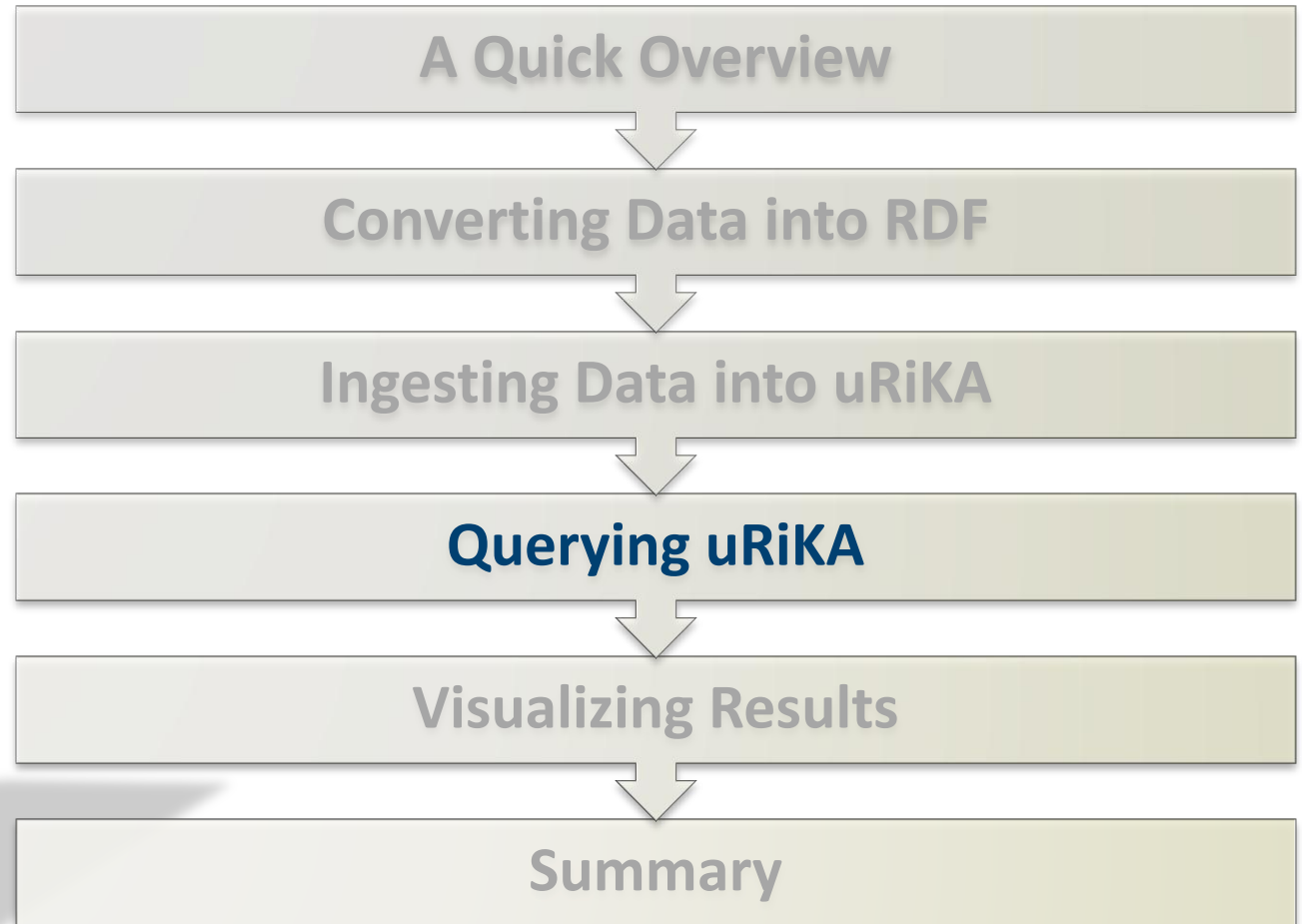
Description	Quick Start Guide
Front End (FE) environment assumptions, setup, and installation instructions	Deploying uRiKA to a service node environment

Quick Start User Guides

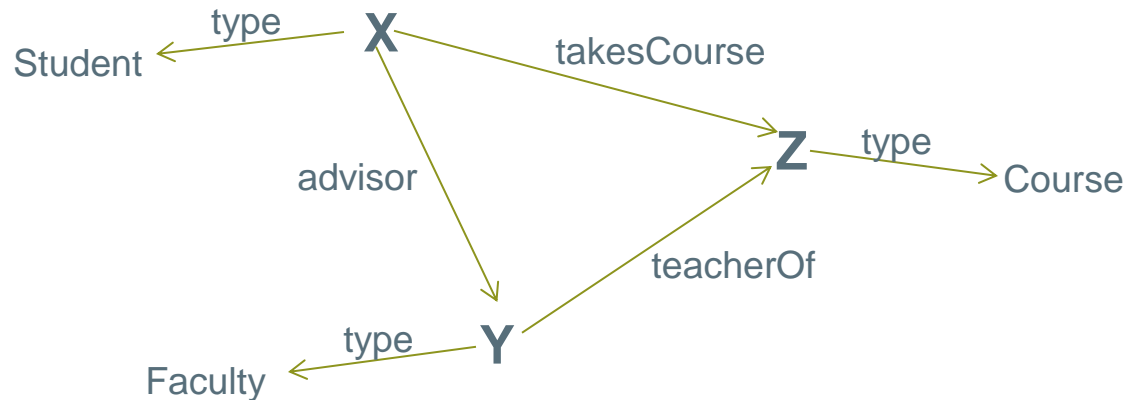
The quick start guides are intended to jump start the user's experience. Further details can be found by following the links to each tools' respective documentation website.

Description	Quick Start Guide	More Info
Extracting structured data from an RDBMS	D2R User Guide	D2R Documentation D2R Mapping Language
Visual relationship finder	RelFinder User Guide	Visual Data Web
Learning SPARQL		SPARQL Reference
Learning about WSO2	User and gadget permission management	WSO2 Gadget Site

Agenda



SPARQL Query



Lehigh University Benchmark (LUBM) Query 9

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ub: <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#>
SELECT ?X, ?Y, ?Z
WHERE
{?X rdf:type ub:Student .
  ?Y<del>rdf:type ub:Faculty .
  ?Z rdf:type ub:Course .
  ?X ub:advisor ?Y<del>.
  ?Y ub:teacherOf ?Z .
  ?X ub:takesCourse ?Z}
```

Use of Y as both a subject and an object requires a join

Inferencing

schema.ttl

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix : <http://example.org/vehicles/> .

:Vehicle a rdfs:Class .
:Car rdfs:subClassOf :Vehicle .
:SportsCar rdfs:subClassOf :Car .
```

data.ttl

```
@prefix ex: <http://example.org/vehicles/> .
@prefix : <http://myvehicledata.com/> .

:FordFiesta a ex:Car .
:AudiA8 a ex:Car .
:FerrariEnzo a ex:SportsCar .
```

query.rq

```
PREFIX ex: <http://example.org/vehicles/> .
PREFIX : <http://myvehicledata.com/> .

SELECT ?car
WHERE { ?car a ex:Car }
```

- Without inferencing, query will return

- FordFiesta, AudiA8

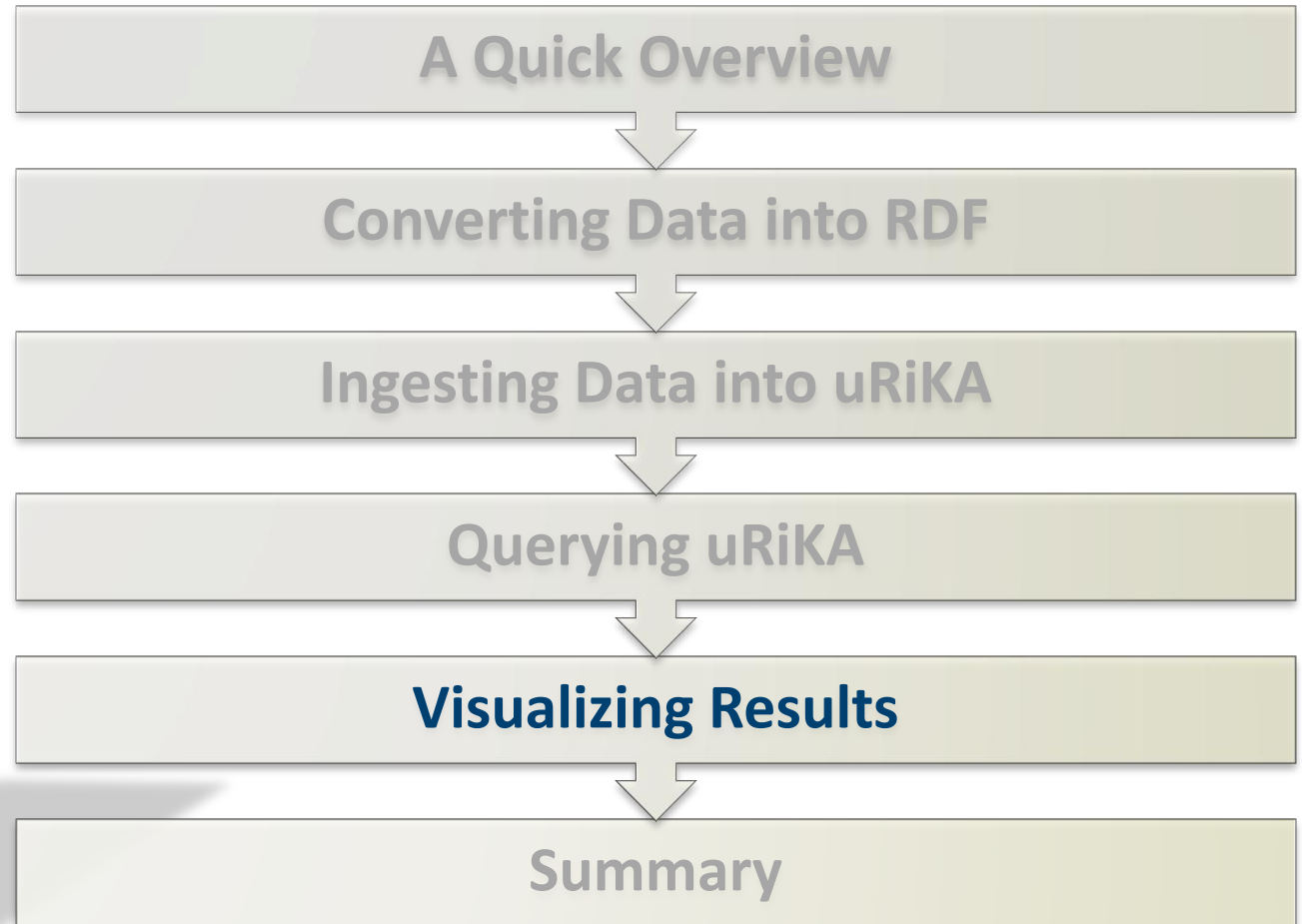
- With inferencing, query will return

- FordFiesta, AudiA8, FerrariEnzo

- We expect inferencing to be a uRiKA strength

Example from <http://www.dotnetrdf.org/content.asp?pageID=Inference%20and%20Reasoning>

Agenda



Visualizing Results

- RelFinder (built into the UI as Explore Data) does this with independently of the relationships in the data
- New visualization packages are being investigated for the next release
- Google Gadgets within the WSO2 framework are the primary interface for gadget development
 - These interfaces are not fully documented or exposed in 0.9



RelFinder

URL: []

between | examples

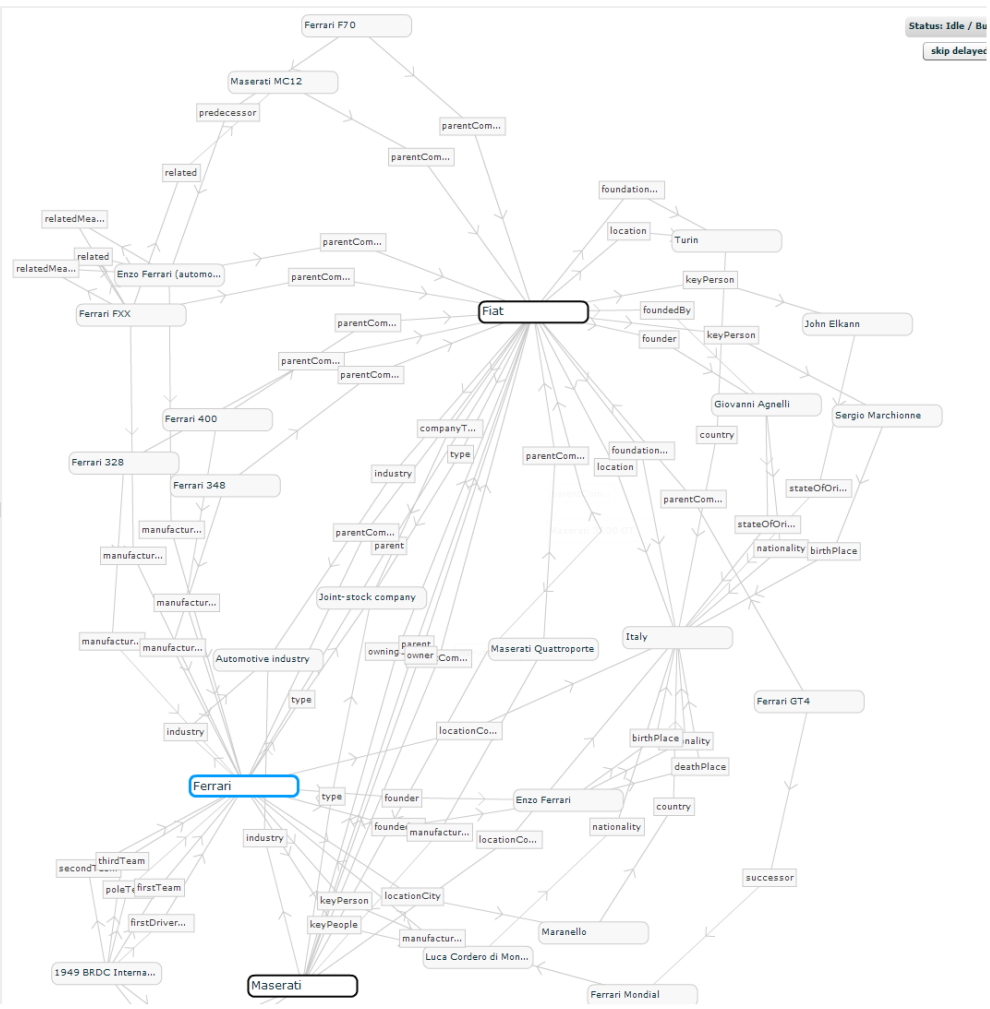
(1) Fiat
(2) Ferrari
(3) Maserati

add | clear | Find Relations

Filter by: relations: (118/136)

length | class | link | conne...

number of objects	num	vi
0	6/6	
1	38/42	
2	74/88	



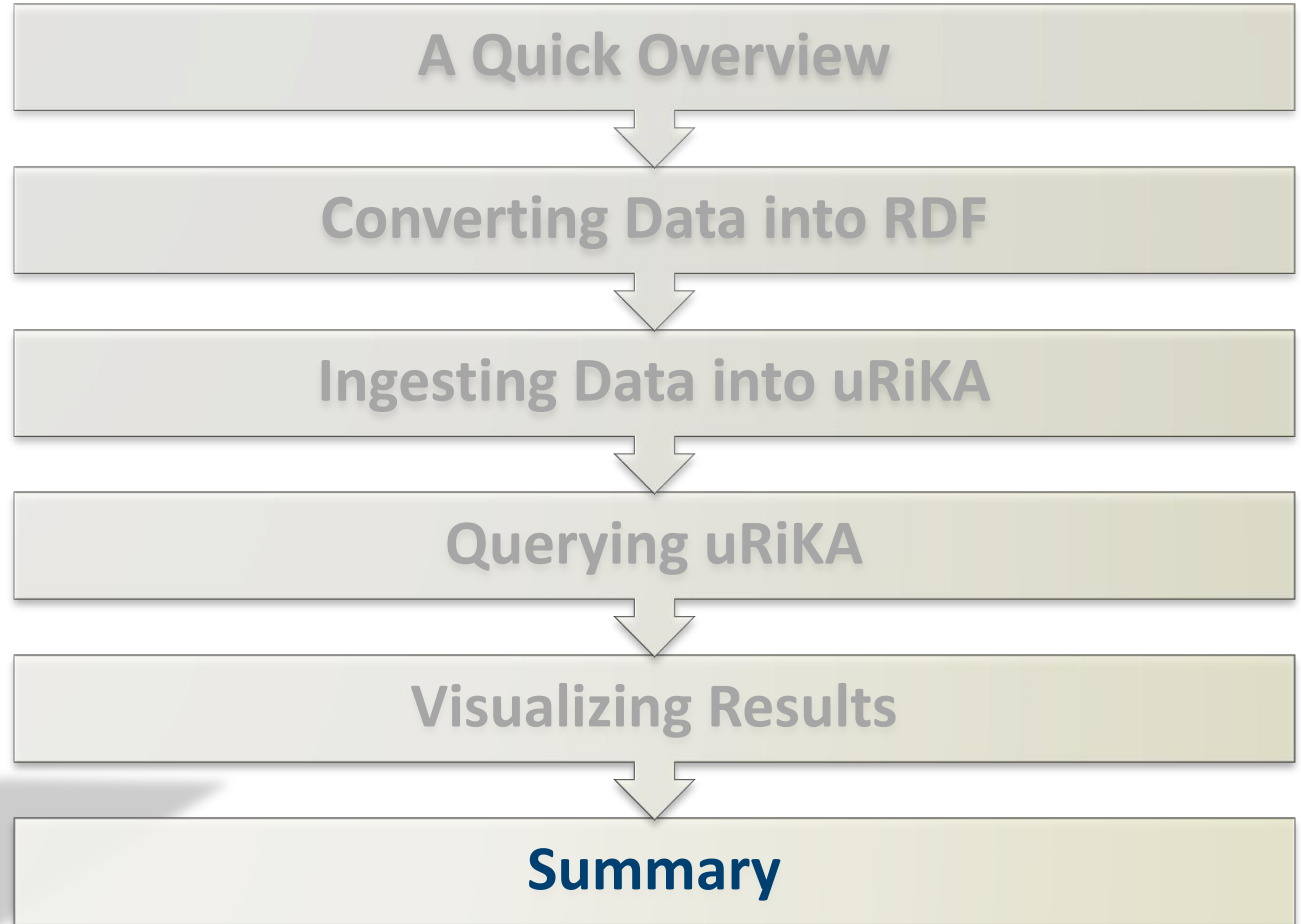
Ferrari

More Info: dbpedia.org

en



Agenda



Summary

- uRiKA is targeted at large-scale semantic-graph processing
- Some end-to-end ETL workflows need components beyond what uRiKA has been tested with today
- We expect to test and qualify third-party components
- Most customers have their own favorite ETL components- we will learn a lot by working with early customers
- ... and modify uRiKA based on those lessons

Learn More

- SPARQL by Example tutorial, by Lee Feigenbaum, <http://www.cambridgesemantics.com/2008/09/sparql-by-example/>
- Search RDF data with SPARQL, by Philip McCarthy <http://www.ibm.com/developerworks/xml/library/j-sparql/>
- Semantic Web for the Working Ontologist, by Dean Allemang and James Hendler, ISBN 978-0123859655
- Learning SPARQL, by Bob DuCharme, O'Reilly, ISBN 978-1-449-30659-5

- RDF: www.w3.org/RDF/
- SPARQL: www.w3.org/TR/rdf-sparql-query/
- D2R: www4.wiwiss.fu-berlin.de/bizer/d2r-server
- WSO2: wso2.com/products/application-server
- Google Gadgets: www.google.com/webmasters/gadgets/

Thank you!

James D. Maltby, Ph.D
jmaltby@yarcdata.com

