ORACLE®

Semantic Technologies in Oracle Database 11g Release 2: Capabilities, Interfaces, Performance

Xavier Lopez, Ph.D., Director
Souripriya Das, Ph.D., Architect

Oracle Database Semantic Technologies Sessions at the Semantic Technology Conference 2010

Date/Time	Title	Track
Tues, 22-Jun-2010		
1:15 - 4:45 p.m.	**Oracle Database Semantic Technologies: Understanding How to Install, Load, Query and Inference**	Semantics for Enterprise Data
Thurs, 24-Jun-2010		
11:45 – 12:45 p.m.	**Semantic Technologies in Oracle Database 11gR2: Capabilities, Interfaces, Performance**	Semantics for Enterprise Data
2:00 – 3:30 p.m.	**An Enterprise-Class Inference Engine Inside Oracle Database 11g Release 2**	Semantic Rules (incl. RuleML, RIF, SILK, RL Profile)

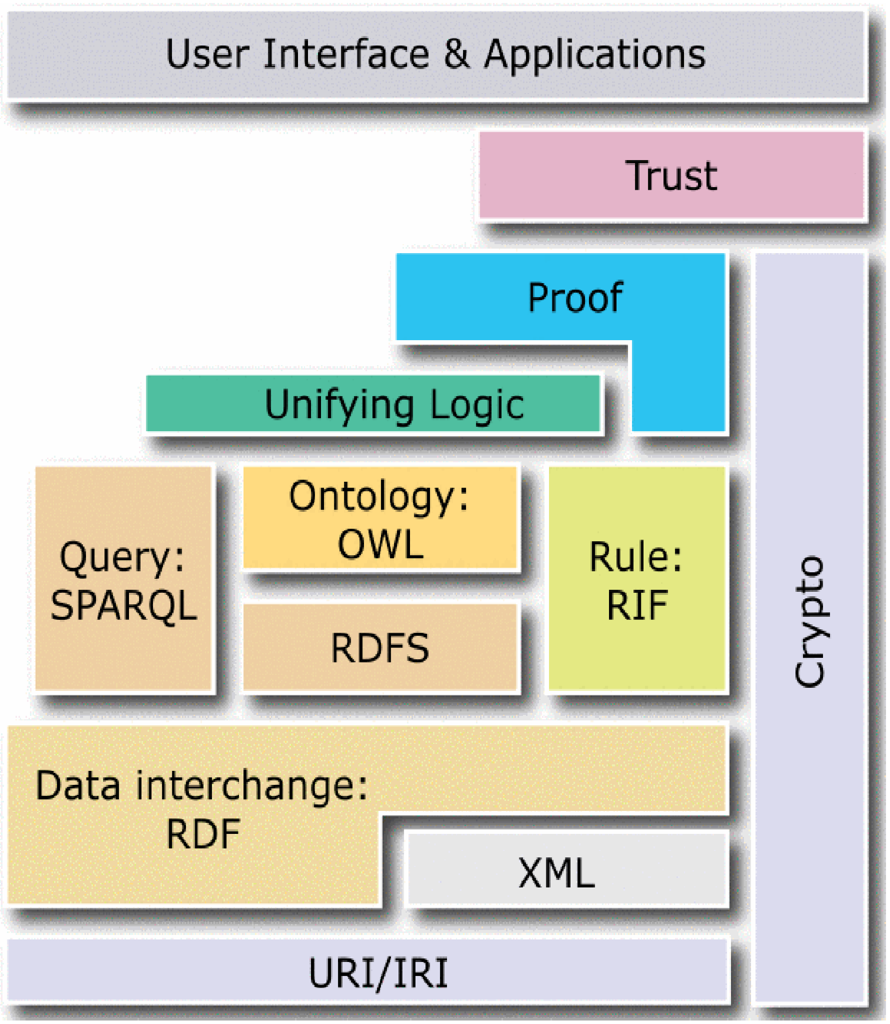
Agenda

- Intro to Semantic Web and Business Use Cases
- Architecture and Storage
- Capabilities
- Interfaces
- Performance

THE FOLLOWING IS INTENDED TO OUTLINE OUR GENERAL PRODUCT DIRECTION. IT IS INTENDED FOR INFORMATION PURPOSES ONLY, AND MAY NOT BE INCORPORATED INTO ANY CONTRACT. IT IS NOT A COMMITMENT TO DELIVER ANY MATERIAL, CODE, OR FUNCTIONALITY, AND SHOULD NOT BE RELIED UPON IN MAKING PURCHASING DECISION. THE DEVELOPMENT, RELEASE, AND TIMING OF ANY FEATURES OR FUNCTIONALITY DESCRIBED FOR ORACLE'S PRODUCTS REMAINS AT THE SOLE DISCRETION OF ORACLE.

Introduction to Semantic Web and Business Use Cases

Semantic Technology Stack



- **Basic Technologies**
  - *URI*
    - Uniform Resource Identifier
  - *RDF*
    - Resource Description Framework
  - *RDFS*
    - RDF Schema
  - *OWL*
    - Web ontology language
  - SPARQL
    - Protocol and Query Language

http://www.w3.org/2007/03/layerCake.svg

Extraction, Modeling, Reasoning & Discovery Workflow

Data Sources

- Transaction Systems
- Unstructured Content
- RSS, email
- Other Data Formats

Transform & Edit Tools

- Entity Extraction & Transform
- Ontology Engineering
- Categorization
- Custom Scripting

Partner Tools

Load, Query & Inference

- RDF/OWL Data Management
- SQL & SPARQL Query
- Inferencing
- Semantic Rules
- Security
- Semantic Indexing
- Versioning

ORACLE SPATIAL

Applications & Analysis Tools

- BI Analytics
- Graph Visualization
- Social Network Analysis
- Metadata Registry
- Faceted Search
- SPARQL Endpoint

Partner/Oracle Tools

Why Organizations use Oracle RDF Database

- Oracle database 11g is the leading commercial database with native RDF/OWL data management
- Scalable & secure platform for wide-range of semantic applications
- Readily scales to ultra-large repositories (10s billions of triples)
- Choice of SQL or SPARQL query
- Leverages Oracle Partitioning. RAC supported
- Growing ecosystem of 3rd party tools partners

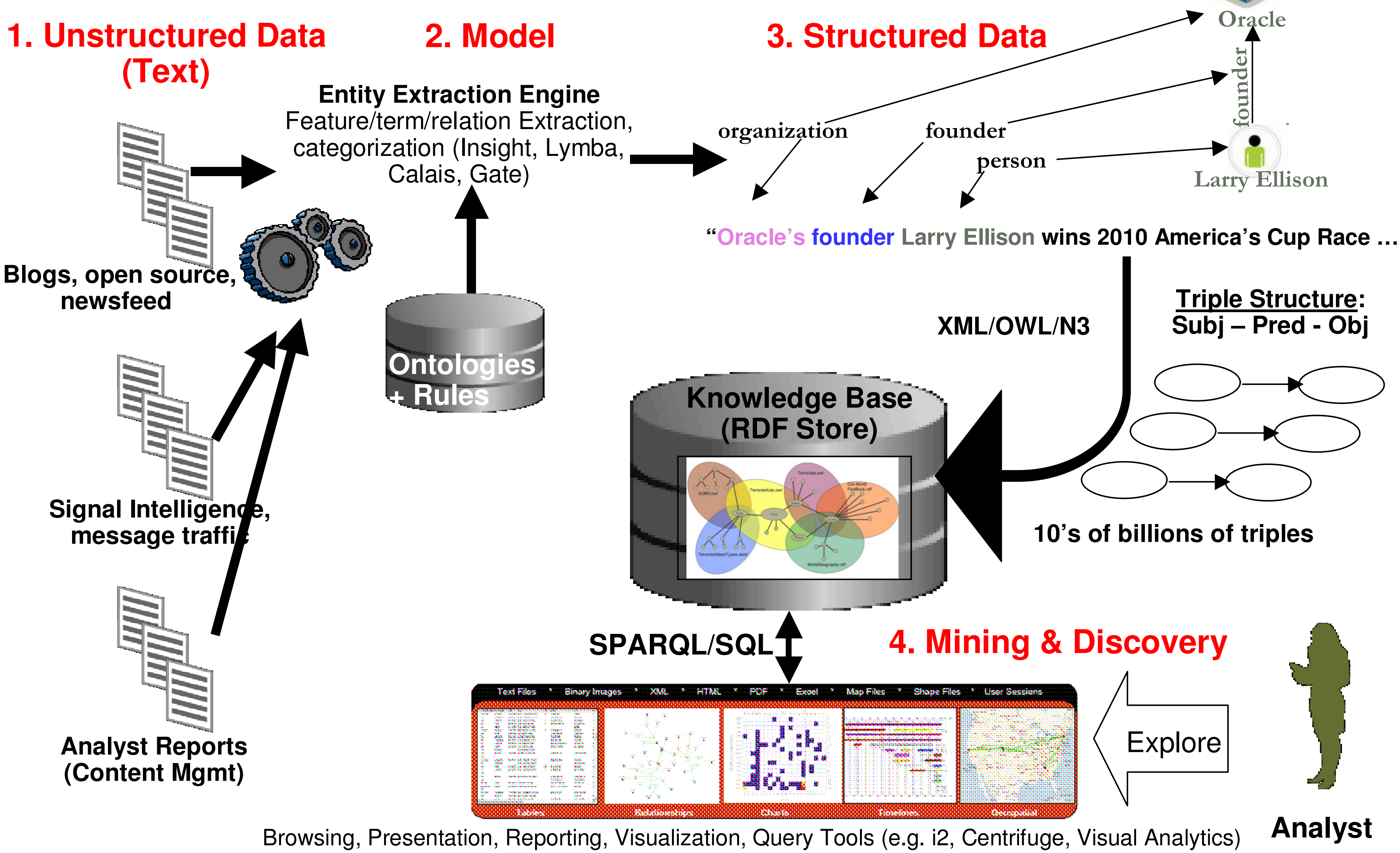
Key Capabilities:

Load / Storage
- Native RDF graph data store
- Manages billions of triples
- Fast batch, bulk and incremental load

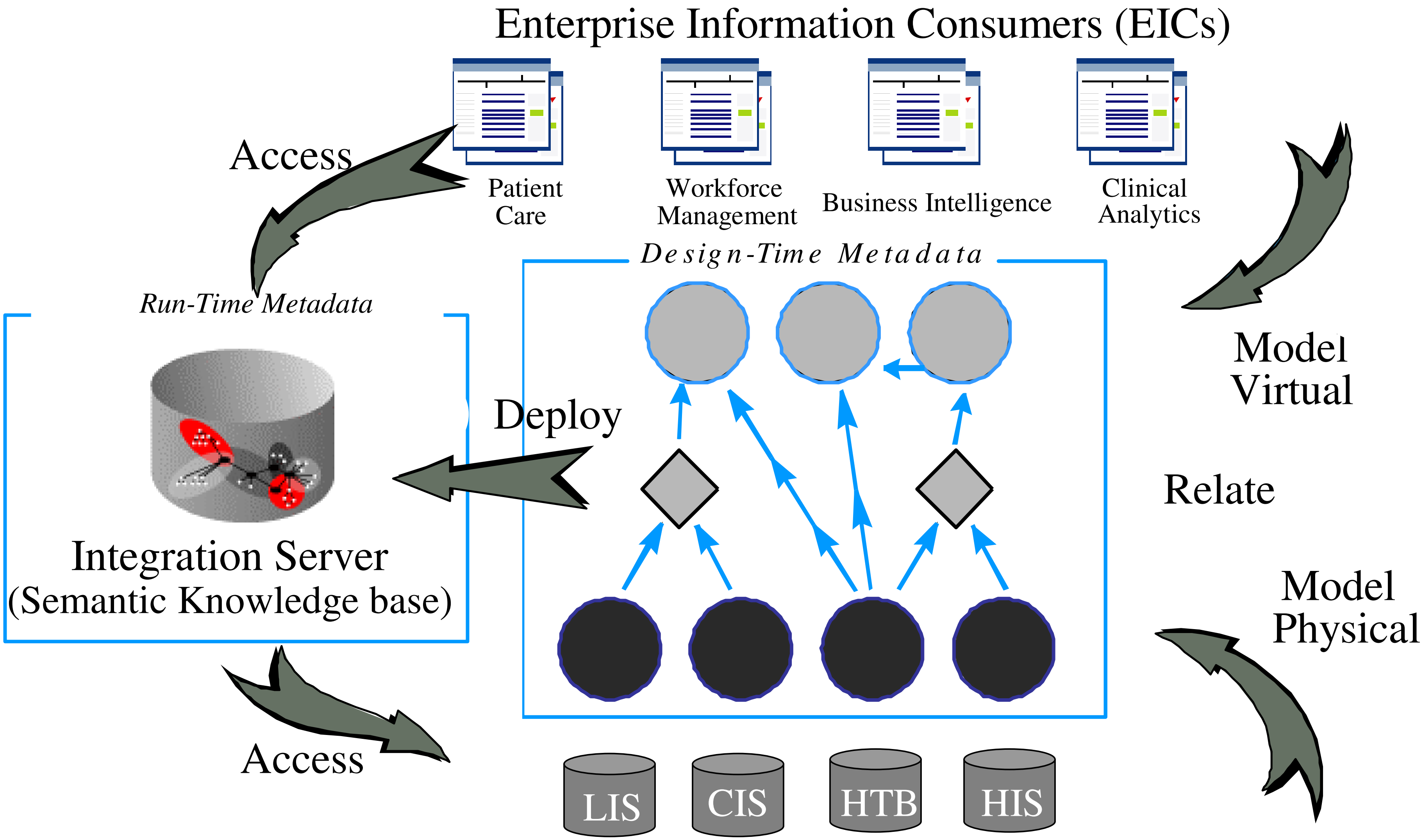
Query
- SQL: SEM_Match
- SPARQL: via Jena plug-in
- Ontology assisted query of RDBMS data

Reasoning
- Forward chaining model
- OWLprime, OWL 2 RL, RDFS
- User defined rule base

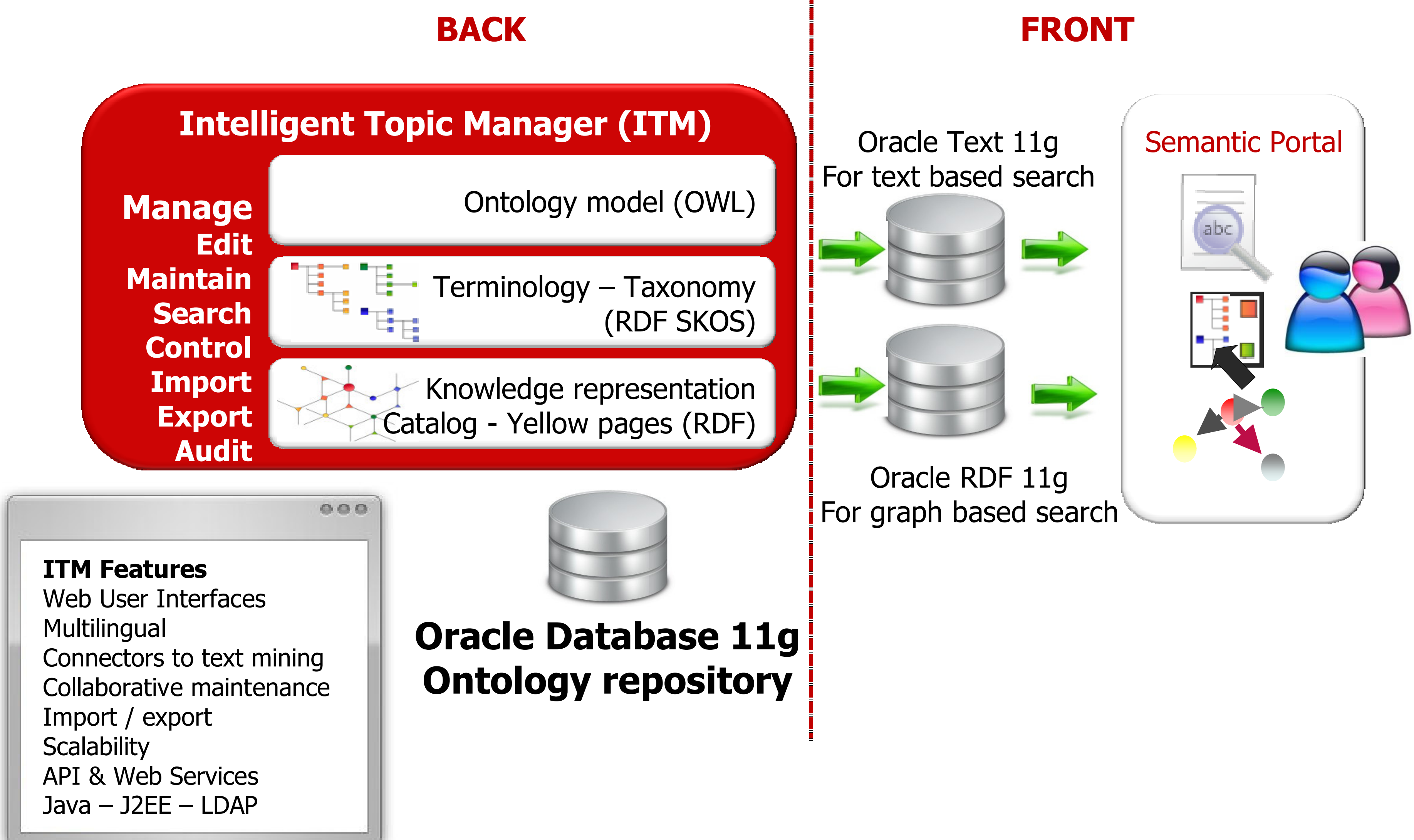
National Intelligence: Text Mining



Data Integration Platform in Health Informatics



Vocabulary Management



Oracle's Partners for Semantic Technologies

Integrated Tools and Solution Providers:

Ontology Engineering

TopQuadrant
ontoprise — know how to use Know-how
protégé

Reasoners

clark & parsia, llc — thinking clearly
ontoprise — know how to use Know-how

Applications

TERANODE
MONDECA
metatomix
MedTrust

Query Tool Interfaces

Jena semantic web framework
openRDF.org
Sesame
Joseki
cycorp

Standards

W3C World Wide Web Consortium
RDF

NLP Entity Extractors

CALAIS — Powered by Thomson Reuters
Linguamatics
GATE — general architecture for text engineering

SI / Consulting

NORTHROP GRUMMAN
Raytheon
orbis TECHNOLOGIES, INC.
BOEING
McDONALD BRADLEY
Cognia
accenture

Some Oracle Database Semantics Customers

Life Sciences

Lilly

Pfizer

SIB
Swiss Institute of Bioinformatics

Defense/ Intelligence

National Geospatial-Intelligence Agency

Education

The University of Michigan
1817

Clinical Medicine & Research

The University of Texas
Health Science Center
at Houston

Cleveland Clinic

Telecomm & Networking

Hutchinson 3G
Austria

CISCO

Publishing

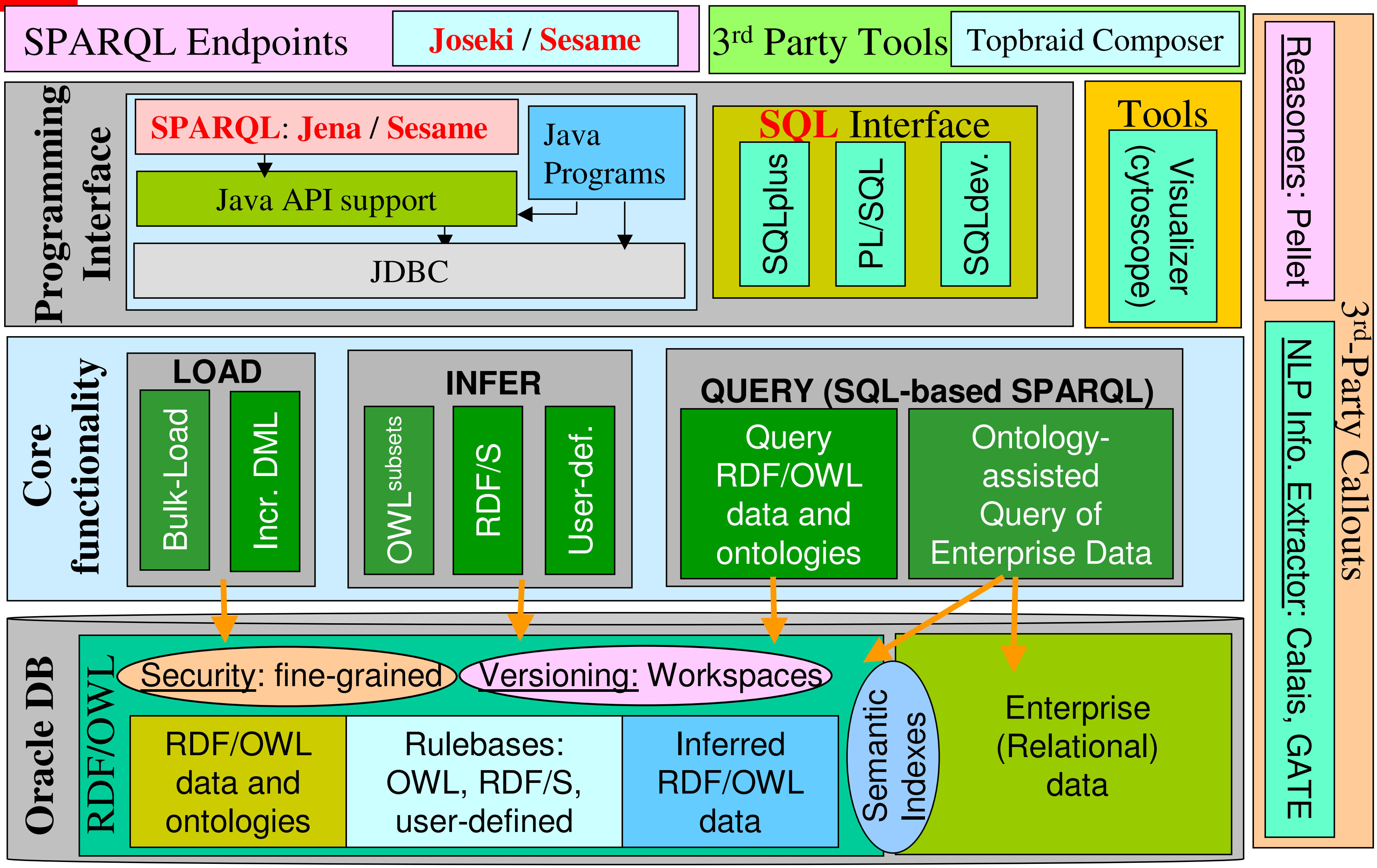
Westlaw
Thomson Reuters

Semantic Technologies in Oracle 11g Release 2

Agenda

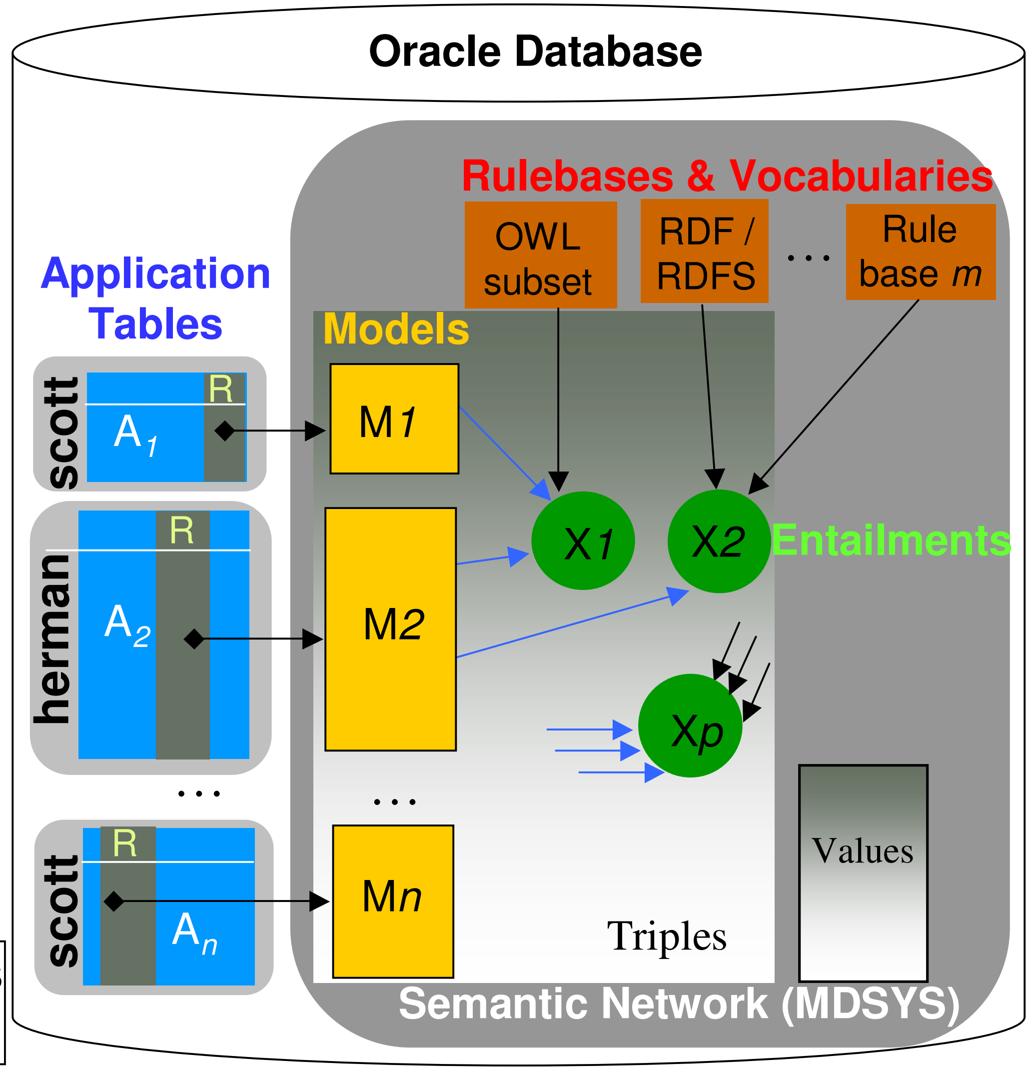
- Intro to Semantic Web and Business Use Cases
- Architecture and Storage
- Capabilities
  - Fundamentals
    - Load / Query / Inference
  - Enterprise
    - Semantic indexing of docs
    - fine-grained Security (at triple level)
    - Versioning
- Interfaces
  - SQL-based
  - Java-based: Jena Adapter and Sesame Adapter
  - SPARQL Endpoints
- Performance

Architectural Overview



Core Entities in Oracle Database Semantic Store

- **Sem. Network** ➔ Dictionary and data tables. New entities: models, rule bases, and rules indexes (entailments). OWL and RDFS rule bases are preloaded.
- **SDO_RDF_TRIPLE_S** ➔ A new object type for RDF.
- **Application Table** ➔ Contains col of object type sdo_rdf_triple_s to allow loading and accessing RDF triples, and storing ancillary values.
- **Model** ➔ A model holds an RDF graph (set of S-P-O triples) and is associated with an sdo_rdf_triple_s column in an application table.
- **Rulebase** ➔ A rulebase is a set of rules used for inferencing.
- **Entailments** ➔ An entailment stores triples derived via inferencing.

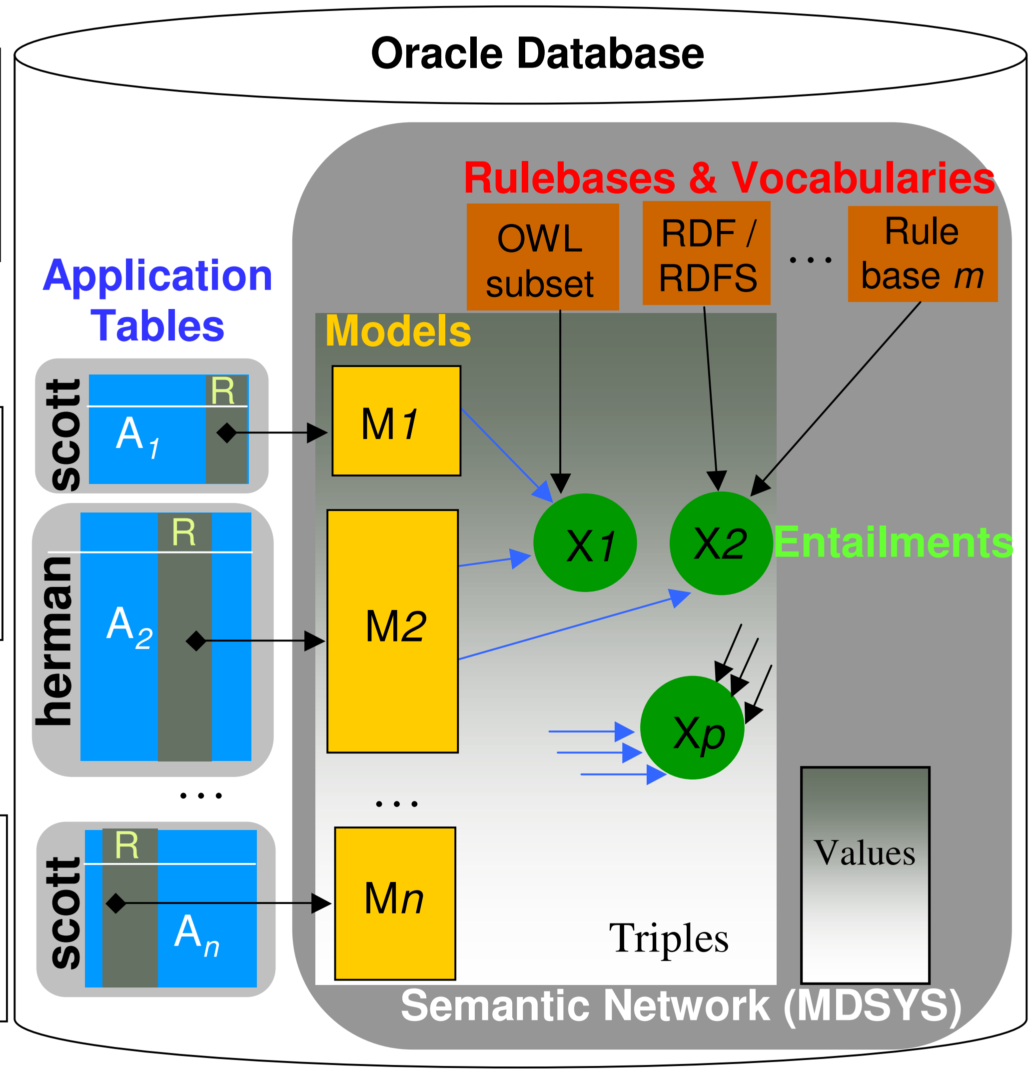


Load / Query / Inference

- **Load** ➔
  - Bulk load
  - Incremental load

- **Query** ➔
  - SPARQL (direct or SQL-based)
  - simple data access

- **Inference** ➔
  - using OWL 2 RL, RDFS, etc.
  - using User-defined rules



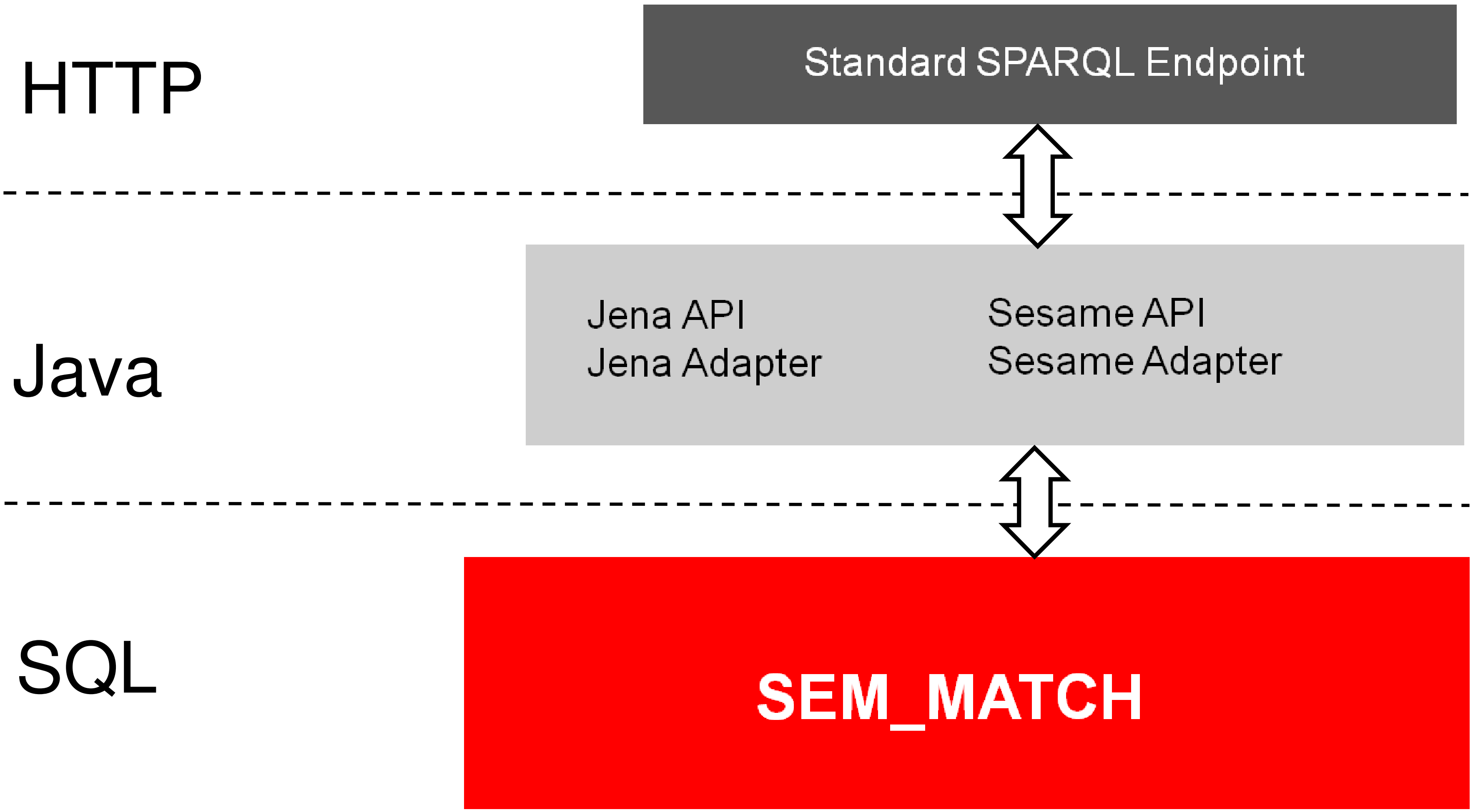
Interfaces

- SQL-based (SQL and PL/SQL)

- Java-based
  - Jena (using Jena Adapter from Oracle)
  - Sesame (using Sesame Adapter from Oracle)

- SPARQL Endpoints
  - Joseki
  - OpenRDF Workbench

SPARQL Query Architecture



SEM_MATCH: Adding SPARQL to SQL

SPARQL

```
PREFIX foaf: <http://...>
SELECT ?n1 ?n2
FROM <http://g1>
WHERE
 {?p foaf:name ?n1
  OPTIONAL {?p foaf:knows ?f .
            ?f foaf:name ?n2 }
  FILTER (REGEX(?n1, "^A"))    }
```

SQL

```
SELECT n1, n2
FROM TABLE(SEM_MATCH(
 '{?p foaf:name ?n1
   OPTIONAL {?p foaf:knows ?f .
             ?f foaf:name ?n2 }
   FILTER (REGEX(?n1, "^A"))   }',
SEM_MODELS('g1'),…,
SEM_ALIASES(
   SEM_ALIAS('foaf','http://…')), …))
```

projection

data selection

graph pattern

prefixes

SEM_MATCH: Adding SPARQL to SQL

Rewritable ~~SQL Table Function~~

```
SELECT n1, n2
FROM
   ( SELECT v1.value AS n1, v2.value AS n2
     FROM   VALUES v1, VALUES v2
            TRIPLES t1, TRIPLES t2, …
     WHERE  t1.obj_id = v1.value_id
       AND  t1.pred_id = 1234
       AND  …
   )
```

Get 1 declarative SQL query
- Query optimizer sees 1 query
- Get all the performance of Oracle SQL Engine
  - compression, indexes, parallelism, etc.

GeoSpatial Querying in an RDBMS

SQL-Based Approach

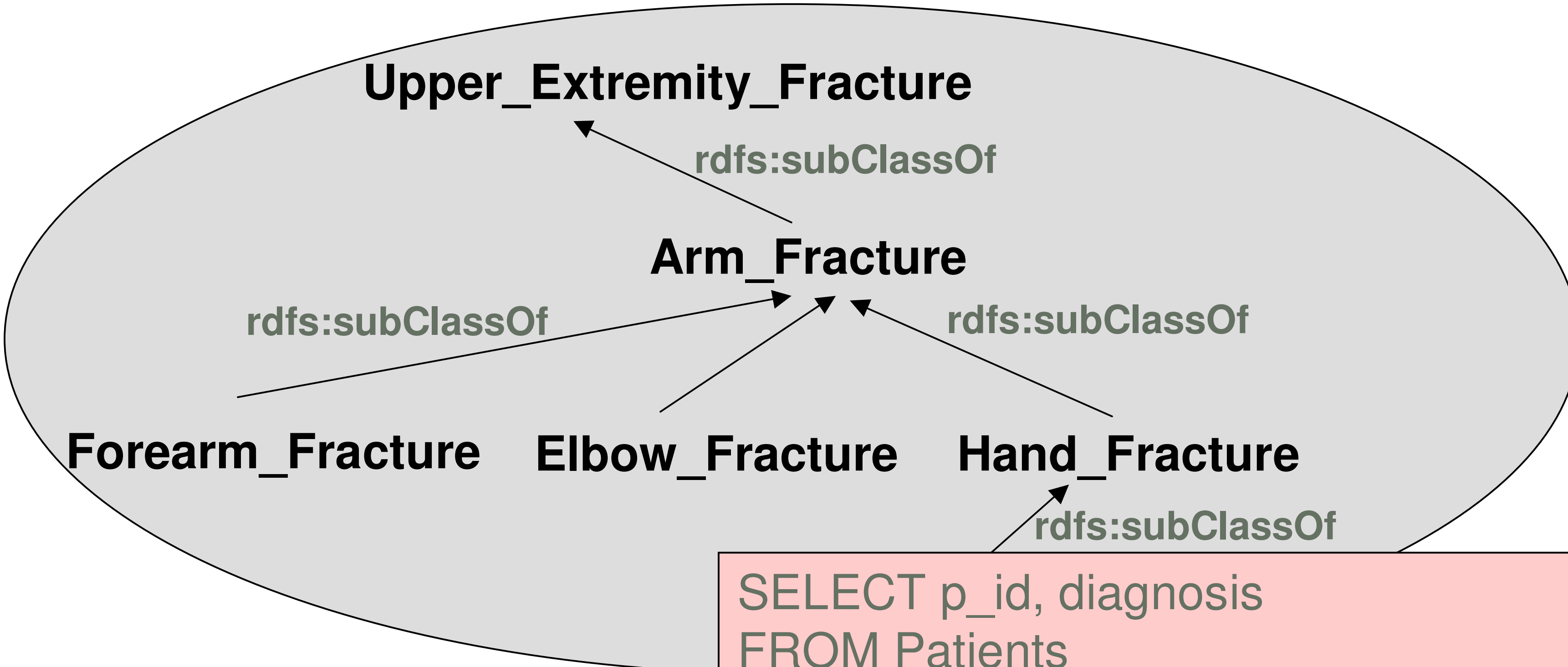
Oracle 11g

Object-Relational Data
(Spatial Objects)

Triple Store
(Semantic Data)

Geo-Semantic SQL-Based Querying:
SEM_MATCH + SDO_RELATE

Ontology-assisted Query using SQL Operators



Patients

ID	DIAGNOSIS
1	Hand_Fracture
2	Rheumatoid_Ar

```
SELECT p_id, diagnosis
FROM Patients
WHERE SEM_RELATED (
          diagnosis,
          'rdfs:subClassOf',
          'Upper_Extremity_Fracture',
           sem_models('Medical_ontology'),
           sem_rulebases('RDFS') …
           123) = 1
AND SEM_DISTANCE(123) <= 2;
```

Inference

- Inference done using forward chaining
  - Triples inferred and stored ahead of query time
  - Removes on-the-fly reasoning and results in fast query times
- Various native rulebases provided
  - E.g., RDFS, OWL 2 RL, SNOMED (EL+), SKOS
- Validation of inferred data
- User-defined rules
- Proof generation
  - Shows one deduction path

OWL Subsets Supported

- **OWL subsets for different applications**
  - RDFS++
    - RDFS plus owl:sameAs and owl:InverseFunctionalProperty
  - OWLSIF (OWL with IF semantics)
    - Based on Dr. Horst's pD* vocabulary[1]
  - OWLPrime
    - Includes RDFS++, OWLSIF with additional rules
    - Jointly determined with domain experts, customers and partners
  - OWL 2 RL
    - W3C Standard
    - Adds rules about keys, property chains, unions and intersections to OWLPrime
  - SNOMED

1 Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary

11g Release 2 Inference Features

- Richer semantics support
  - OWL 2 RL, SKOS, SNOMED (subset of OWL 2 EL)
- Performance enhancements
  - Large scale owl:sameAs handling
    - Compact materialization of owl:sameAs closure
  - Parallel inference
    - Leverage native Oracle parallel query and parallel DML
  - Incremental inference
    - Efficient updates of inferred graph through additions
  - Compact Data Structures

Semantic Indexing of Documents

Analytical Queries On Graph Data

Triples *table with rowid references*

Subject	Property	Object	rid
p:Marcus	rdf:type	rc::Person	r1
p:Marcus	:hasName	"Marcus"^^…	r1
p:Marcus	:hasAge	"38"^^xsd:..	r1
…	…	…	…

```
CREATE INDEX ArticleIndex
 ON NewsFeed (Article)
 INDEXTYPE IS SemContext
 PARAMETERS ('gate_nlp')
```

SemContext index on ***Article*** *column*

Newsfeed *table*

docId	Article	Source
1	Indiana authorities filed felony charges and a court issued an arrest warrant for a financial manager who apparently tried to fake his death by crashing his airplane in a Florida swamp. Marcus Schrenker, 38 …	CNN
2	Major dealers and investors …	NW
..		..

```
SELECT docId FROM Newsfeed
 WHERE SEM_CONTAINS (Article,
          '{?x rdf:type rc:Person .
           ?x :hasAge ?age .
           FILTER(?age >= 35)}')=1
      AND Source = 'CNN'
```

Enterprise Security for Semantic Data: VPD (Virtual Private Database)

[VPD] Access control policies on semantic data

Policy ➔ user can access value of projects s/he leads

Match pattern ➔ { ?x :hasValue ?v }

Apply pattern ➔ { ?x :hasLead "sys_context(…)" }

Query : Get the list of projects and their values

```
SELECT  ?proj  ?val
FROM    ProjectsGraph
WHERE   { ?proj   :hasValue  ?val  .
                ?proj   :hasLead "sys_context(…)" }
```

Enterprise Security for Semantic Data: OLS (Oracle Label Security)

[OLS] Data classification labels for semantic data

SELECT operation: Labels for triples are used to restrict access to the triples.

Triples table

Subject	Predicate	Object	Row Label
projectHLS	Organization	N.America	SE:HLS:US
projectHLS	ContractValue	1000000	SE:HLS,FIN:US

INSERT operation: Labels for resources to control their use as component of a triple during insertion. (Also, determines min. access label for inserted triple.)

Position & Resource	Row Label
Subject: projectHLS	**SE:HLS:US**
Predicate: hasContractValue	**TS:FIN:**

Label Security Modes

- Triple-level: Easier to use. Any one can insert, bulk-load, do inference. (Inserted/inferred triples get user's label. Query/Infer on visible triples.)
- Resource-level: Secures each resource independently.

Change Mgmt./Versioning for Sem. Data

- Manage public and private versions of semantic data in database workspaces (Workspace Manager)
- An RDF Model is version-enabled by version-enabling the corresponding application table.
  - exec DBMS_WM.enableVersioning (table_name => 'contracts_rdf_data');
- RDF data modified within a workspace is private to the workspace until it is merged.
- SEM_MATCH queries on version-enabled models are version aware and only return relevant data.
- New versions created only for changed data
- Versioning is provisioned for inference

Utility APIs

- SEM_APIS.merge_models
  - Can be used to clone model as well.
  - e.g. exec sem_apis.merge_models('model1','model2');
- SEM_APIS.alter_model
  - e.g. sem_apis.alter_model('m1', 'MOVE', 'TBS_SLOWER');
- SEM_APIS.swap_names
  - e.g. exec sem_apis.swap_names('production_model','prototype_model');
- SEM_APIS.rename_model/rename_entailment
- SEM_APIS.remove_duplicates
  - e.g. exec sem_apis.remove_duplicates('graph_model');

Agenda

- Intro to Semantic Web and Business Use Cases
- Architecture and Storage
- Capabilities
  - Fundamentals
    - Load / Query / Inference
  - Enterprise
    - Semantic indexing of docs
    - fine-grained Security (at triple level)
    - Versioning
- Interfaces
  - SQL-based
  - Java-based: Jena Adapter and Sesame Adapter
  - SPARQL Endpoints
- Performance

Query Performance tuning: Network Indexes

- Some B-tree indexes (including one for enforcing uniqueness constraint) are created automatically at sem. network creation time.
- The set of usable indexes can be manipulated using
  - sem_apis.add_sem_index
  - sem_apis.drop_sem_index
  - sem_apis.alter_index_on_model
  - sem_apis.alter_index_on_entailment
- *Tip*: when creating a network index, use the *model_id* column as part (usually, the last col) of the index key

Best Practice: Virtual Model

- Similar to simple database Views
- One may define a *Virtual Model* consisting of multiple sem. Models, and optionally the corresponding entailment.
- The relevant subprograms are
  - sem_apis.create_virtual_model
  - sem_apis.drop_virtual_model
- Benefits
  - Allows ease and flexibility in specifying target models plus (optionally) the corresponding entailment
  - Facilitates access control
  - Query performance (esp. for queries against <u>multi-model</u>, or <u>single/multi-model+entailment</u> RDF data source)

Bulk-Load performance tuning

- Various options for “flags” parameter
  - Parallel=<n>
  - Parallel_Create_Index
  - Join hints for sub-steps (e.g., MBV_JOIN_HINT=USE_HASH)
- ASM
- temp tablespace should be on a different disk than data and indexes
- For detailed event tracing, use RDF$ET_TAB table

Inference performance tuning

- SameAs optimization
- Incremental inference
- Parallel inference

Tuning Tips for Best Inference Performance

- Analyze models before running inference
  - `execute immediate sem_apis.analyze_model(…);`
- Need a **balanced** hardware setup to use parallel inference
  - E.g., a server with multi-core/multi-cpu processors and ample I/O throughput
  - Use Oracle Automatic Storage Management (ASM) to manage the disks
- Use RAW8=T option for compact data structures
  - Smaller data structures imply less I/O
- Dynamic incremental inference
  - Selectively applies semi-naïve rule evaluation while generating the entailment
  - Off by default, could be turned on with DYN_INC_INF=T option

Query Hint in SEM_MATCH

```
SELECT e, loc FROM
  TABLE(SEM_MATCH( '{?e rdf:type :SoftwareEngineer .                t0
                     ?e :worksFor ?company .                         t1
                     ?company :headQuarterLoc ?loc}',                t2
        SEM_Models('emp', 'gmap'), SEM_Rulebases('OWLPRIME'),
        SEM_ALIASES(SEM_ALIAS('', 'http://example.org/')),NULL,NULL,
        ' HINT0 = { LEADING(t0 t1 t2) USE_NL(t1)} '));
```

- HINTS in SEM_MATCH query
  - Similar to SQL hints
    - Join order: LEADING(…), ORDERED
    - Join type: USE_HASH(…), USE_NL(…), …
    - Access path: INDEX(…), …
  - Aliases are a bit different
    - Variable names: ?e, ?company, ?loc, …
    - Pattern ordinal based: t0, t1, t2, …

Query Performance Tuning: special hints

- Use ALLOW_DUP=T in the case of multi-model queries

Performance on Desktop PC

Bulk Loader Performance on Desktop PC (Core2Duo, 8GB memory, ASM with 3 disks)

Ontology size	Time on 11.2 Latest (internal) [1] (Core 2 Duo, 8GB, ASM)			
	Sql*loader time [3]	bulk-load API [2] Time (incl. parse)		
	Parallel=true	Parallel=1	Parallel=2	Parallel=4
LUBM50 6.9 million	0.4 min	3.9 min	2.7 min	2.7 min
LUBM500 69 million	4 min	49 min	33 min	33 min
LUBM1000 138 million	8 min	104 min	71 min	70 min
LUBM8000 1,106 million	65 min	919 min	598 min	555 min

[1] Assumes empty network to start with. [2] Uses flags=>' parse parallel=4 parallel_create_index ' plus a shadow option for value processing.
[3] Uses parallel=true option and 8 to 10 gzipped N-Triple files as data files and a no-parse control file.

Query Performance

Ontology LUBM50 6.8 million & 5.4 million inferred		LUBM Benchmark Queries						
OWLPrime & new inference components	**Query**	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**
	# answers	4	130	6	34	719	519842	67
	Complete?	Y	Y	Y	Y	Y	Y	Y
	Time (sec)	0.05	0.75	0.20	0.5	0.22	1.86	1.71
	Query	**Q8**	**Q9**	**Q10**	**Q11**	**Q12**	**Q13**	**Q14**
	# answers	7790	13639	4	224	15	228	393730
	Complete?	Y	Y	Y	Y	Y	Y	Y
	Time (sec)	1.07	1.65	0.01	0.02	0.03	0.01	1.47

- Setup: **Intel Q6600 quad-core, 3 7200RPM SATA disks, 8GB DDR2 PC6400 RAM, No RAID. 64-bit Linux 2.6.18.** Average of 3 warm runs

Query Performance on Desktop PC

Ontology LUBM50 6.8 million & 5.4 million inferred		LUBM Benchmark Queries						
OWLPrime & new inference components	**Query**	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**
	# answers	4	130	6	34	719	519842	67
	Complete?	Y	Y	Y	Y	Y	Y	Y
	Time (sec)	0.05	0.75	0.20	0.5	0.22	1.86	1.71
	Query	**Q8**	**Q9**	**Q10**	**Q11**	**Q12**	**Q13**	**Q14**
	# answers	7790	13639	4	224	15	228	393730
	Complete?	Y	Y	Y	Y	Y	Y	Y
	Time (sec)	1.07	1.65	0.01	0.02	0.03	0.01	1.47

- Setup: **Intel Q6600 quad-core, 3 7200RPM SATA disks, 8GB DDR2 PC6400 RAM, No RAID. 64-bit Linux 2.6.18.** Average of 3 warm runs

11.2 Inference Performance

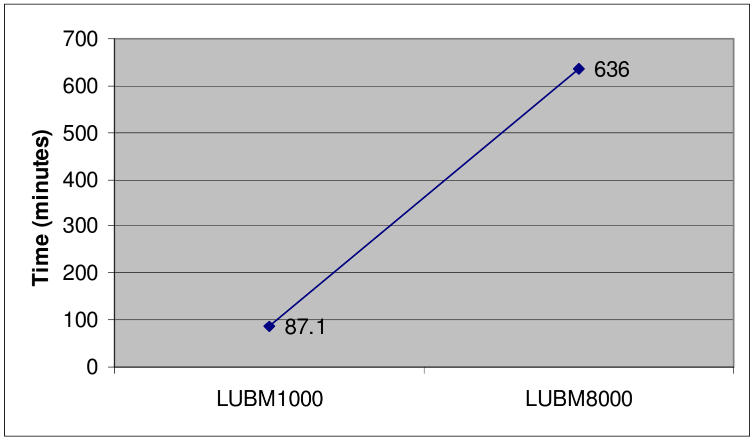
Parallel Inference (LUBM8000 1.06 billion triples + 860M inferred)	• Time to finish inference: 12 hrs. • **3.3x faster compared to serial inference in release 11.1**
Parallel Inference (LUBM25000 3.3 billion triples + 2.7 billion inferred)	• Time to finish inference: 40 hrs. • **30% faster than nearest competitor** • 1/5 cost of other hardware configurations
Incremental Inference (LUBM8000 1.06 billion triples + 860M inferred)	• Time to update inference: less than 30 seconds after adding 100 triples. • **At least 15x to 50x faster than a complete** inference done with release 11.1
Large scale owl:sameAs Inference (UniProt 1 Million sample)	• 60% less disk space required • 10x faster inference compared to release 11.1

- Setup: **Intel Q6600 quad-core, 3 7200RPM SATA disks, 8GB DDR2 PC6400 RAM, No RAID. 64-bit Linux 2.6.18.** **Assembly cost: less than USD 1,000**

Performance on Server

Load Performance on Server

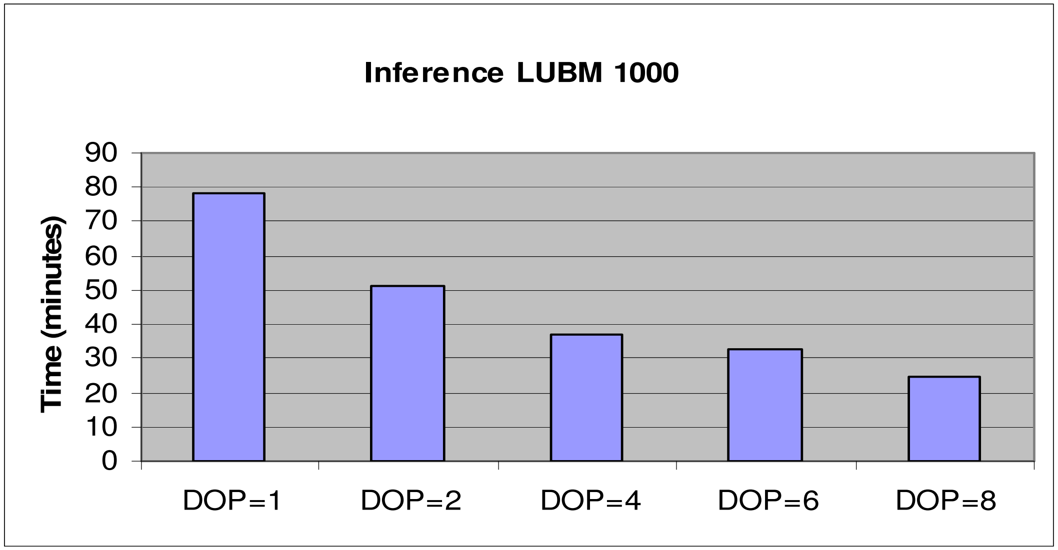
- **LUBM1000 (138M triples)**
  - 8.3 minutes to load data into staging table
  - 78.8 minutes to load data from staging table (DOP=8)



- **LUBM8000 (1B+)**
  - 25 minutes to load data into staging table
  - 10hr 36 minutes to load data from staging table (DOP=8)
- Setup: **Dual quad-core, Sun Storage F5100 Flash Array, 32 GB RAM**

Inference Performance on Server

- **Inference performance for LUBM1000 (138M)**
  - 24.6 minutes to infer 108M+ new triples (DOP=8)

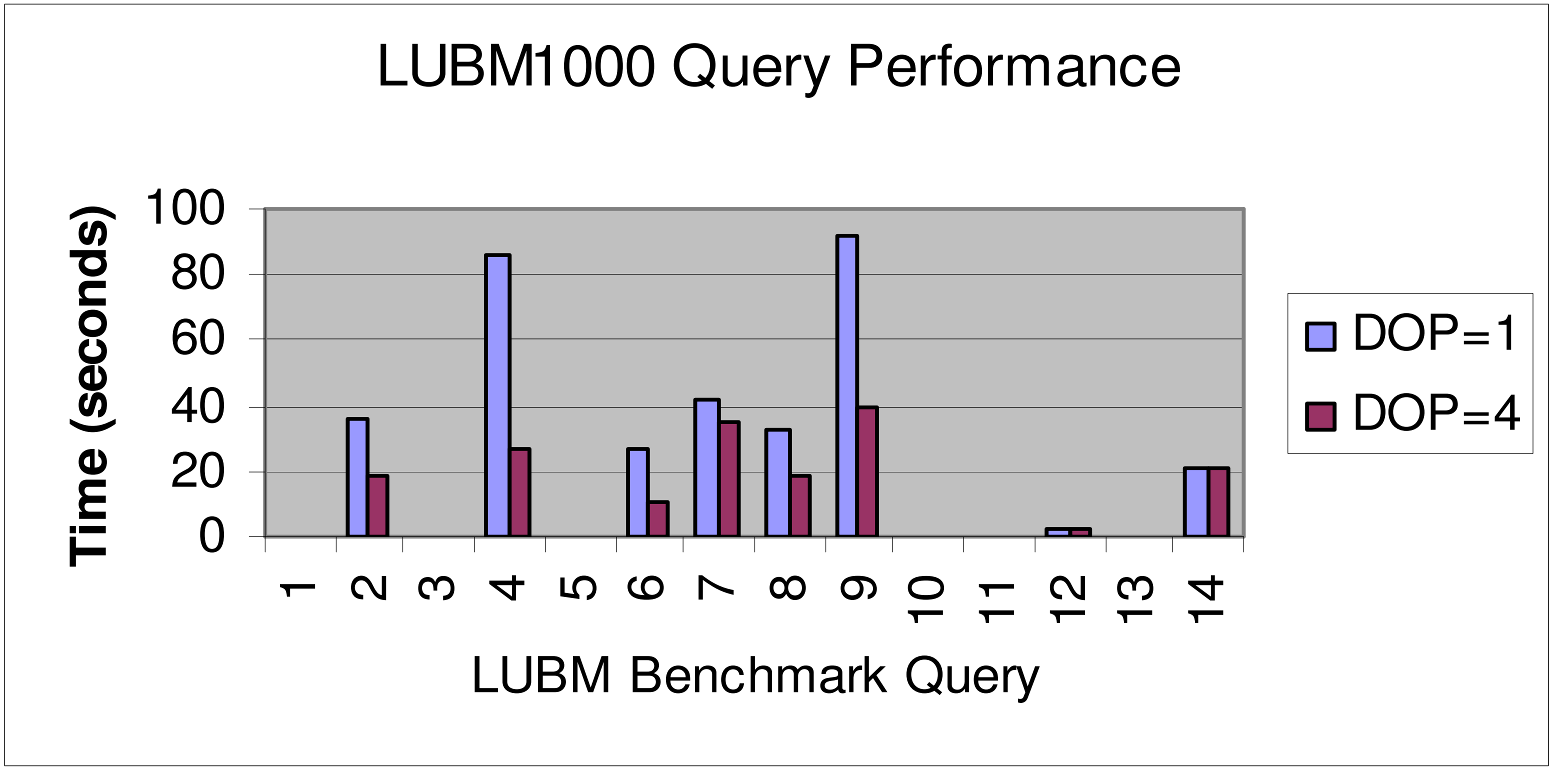


- **Inference performance for LUBM8000 (1B+)**
  - 226 minutes to infer 860M+ new triples (DOP=8)

- Setup: **Dual quad-core, Sun Storage F5100 Flash Array, 32 GB RAM**

Query Performance on Server

- **Parallel query execution**



- Setup: **Server class machine with 16 cores, NAND based flash storage, 32GB RAM,** Linux 64 bit, Average of 3 warm runs

Performance on Sun Oracle Database Machine (Exadata V2)

Load Performance on Exadata V2

- **LUBM 25K benchmark ontology (3.3 Billion triples)**
  - (*Note: These are preliminary numbers and will be updated.*)
  - 105 minutes to load the data into staging table
  - 730 minutes for the bulk-load API, but with values pre-loaded

- Setup: **Sun Oracle Data Machine and Exadata Storage Server (8 node cluster, Full Rack)**

http://www.oracle.com/technology/products/bi/db/exadata/pdf/exadata-technical-whitepaper.pdf

Inference Performance on Exadata V2

- **LUBM 25K benchmark ontology (3.3 Billion triples)**
  - OWLPrime inference with new inference components took 247 minutes (4 hours 7 minutes)
  - More than 2.7 billion new triples inferred
  - DOP = 32

- **Preliminary result on LUBM 100K benchmark ontology (13 Billion+ triples)**
  - One round of OWLPrime inference (limited to OWL Horst semantics) finished in 1.97 hours
  - 5 billion+ new triples inferred
  - DOP = 32

- Setup: **Full Rack Sun Oracle Data Machine and Exadata Storage Server (8 node cluster)**

Query Performance on Exadata V2

Ontology LUBM25K 3.3 billion & 2.7 billion inferred		LUBM Benchmark Queries						
OWLPrime & new inference components	**Query**	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**
	# answers	4	2528	6	34	719	260M	67
	Complete?	Y	Y	Y	Y	Y	Y	Y
	Time (sec)	0.01	20.65	0.01	0.01	0.02	23.07	4.99
	Query	**Q8**	**Q9**	**Q10**	**Q11**	**Q12**	**Q13**	**Q14**
	# answers	7790	6.8M	4	224	15	0.11M	197M
	Complete?	Y	Y	Y	Y	Y	Y	Y
	Time (sec)	0.48	203.06	0.01	0.02	0.02	2.40	19.45

- Setup: **Full Rack Sun Oracle Data Machine and Exadata Storage Server (8 node cluster)**
- **Auto DOP is used. Total # of answers 465,849,803 in less than 5 minutes**

Summary: Oracle Database 11g Release 2

Oracle database is the leading commercial database with native support for W3C standards compliant RDF/OWL data store w/ comprehensive capabilities for ….

Reasoning and Discovery supporting std. ontologies persistent, native & 3rd party inference, and user-defined rules

Scalability to evolve schemas dynamically and grow to 10's billions of triples, incremental & parallel inference

Data Integration to link structured & unstructured content, Loosely couple business silos

Security to protect data on a "need to know" basis

Integrated querying & manageability SPARQL & SQL for RDF/OWL, relational, XML, text, location, & multimedia data

For More Information

search.oracle.com

Semantic Technologies

or

oracle.com

ORACLE IS THE INFORMATION COMPANY