TOWARDS A SEMANTIC WEB APPLICATION FOR NVD-CPE

Vaibhav Khadilkar
Jyothsna Rachapalli
Dr. Bhavani Thuraisingham
The University of Texas at Dallas

Agenda

- **Motivation** to opt for semantic web technology
- **Architecture** of a semantic web application
- **Semantic web technologies** overview
- **Strategy** for creation of semantic web application
- **Performance metrics**

Motivation

- National Vulnerability Database (NVD)
  - Contains product and vulnerability management data
  - Based on a relational model
- Goal is to enable automation of
  - Vulnerability management
  - Security measurement and compliance
- Relational model imposes limitations
  - Product composition difficult to achieve.
    - Find all products containing a TCP/IP device?
    - Find all products within common codebase?
- Advantage of semantic model - Reasoning!

Project Objectives

- Creation of products ontology for NVD-CPE
- Creation of a corresponding view in relational DB
- Migrate data from relational to semantic model
- Create a web application using the new model
- This application should enable user to
  - Navigate
  - Search
  - Query the data

Semantic Technology

- Converter
  - Converts data form various sources(e.g.,tables, spreadsheets, webpages) into RDF
- RDF Parser and Serializer
  - Facilitates reading and writing RDF in one of several file formats (e.g., N3, N-TRIPLE, RDF/XML)
- RDF Store (or triple store)
  - Is a database that is optimized for the storage and retrieval of many short statements called triples

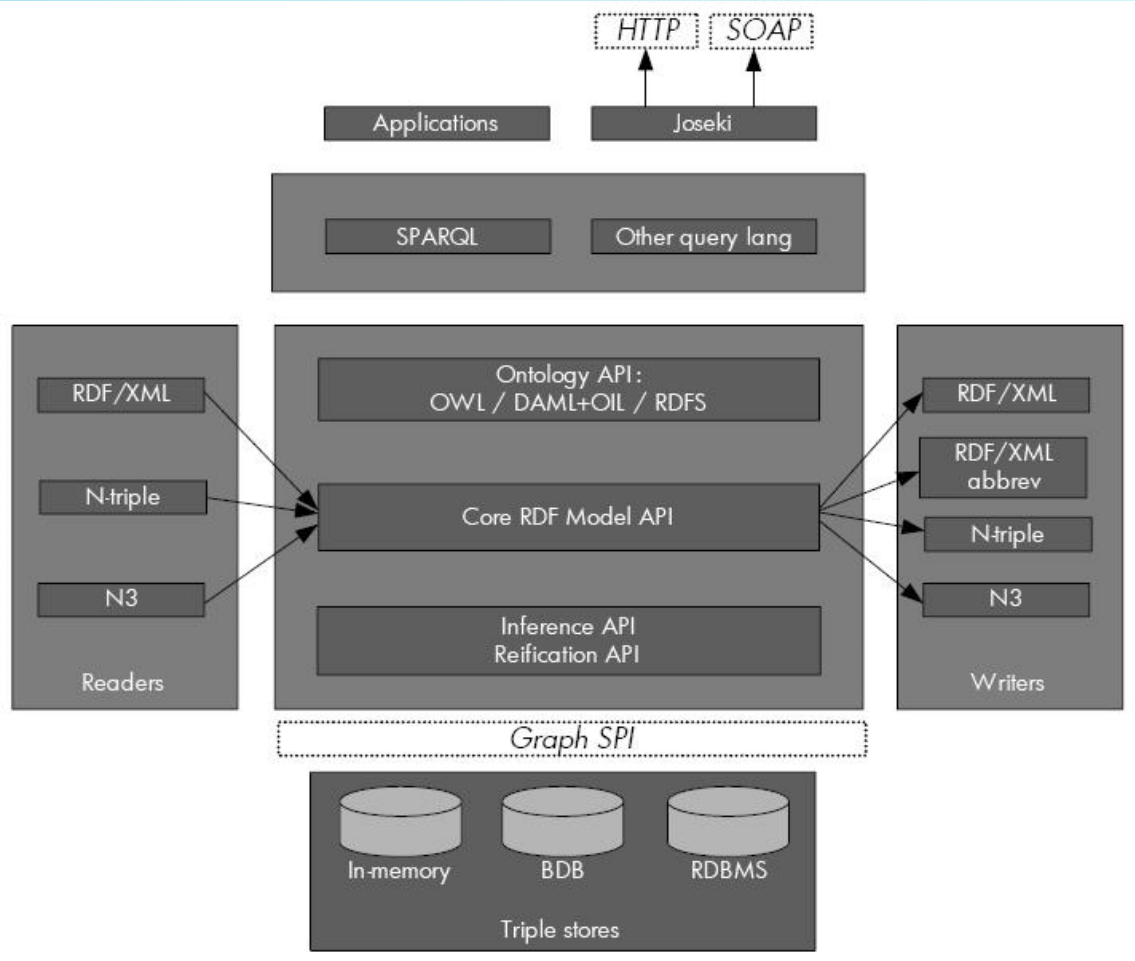
Semantic Technology

- Reasoner
  - A program that performs inferences according to specified inference rules
- SPARQL
  - The W3C standard query language for RDF
- Application interface
  - Uses the content of an RDF store in an interaction with some user

Semantic Technology-Examples

- Converters
  - D2RQ used during first approach
  - Jena API to read relational data into a Jena model
- Parser/Serializer
  - Jena API to read and write the triples into any serialization format
- RDF Store
  - RDB, SDB and Allegrograph
- Inferencing
  - Pellet Reasoner
- SPARQL
  - ARQ is a query engine for Jena that supports SPARQL

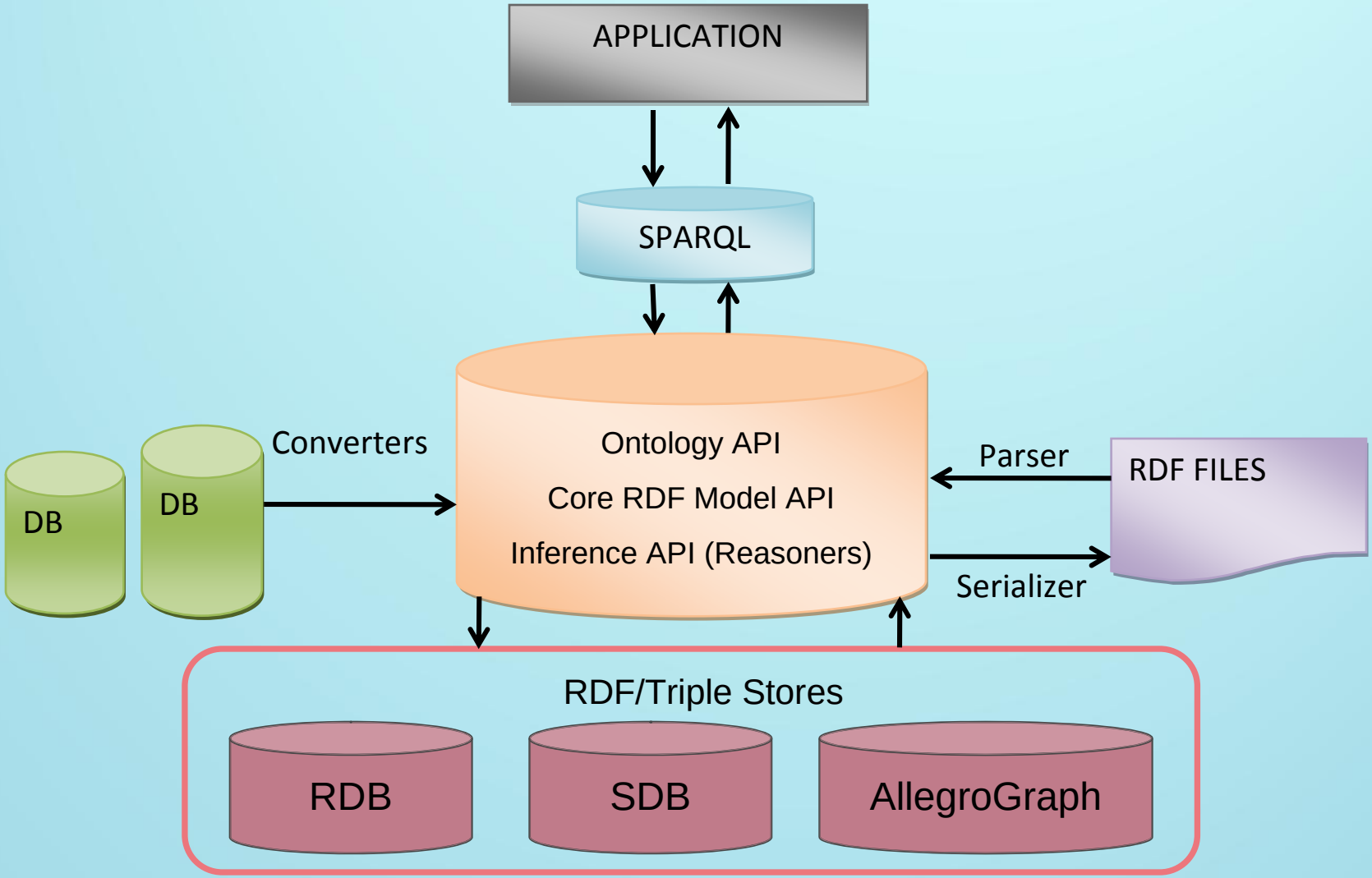
Semantic Technology-Jena



The Jena Framework provides

- A RDF API
- Reading and writing RDF in RDF/XML, N3 and N-Triples
- An OWL API
- In-memory and persistent storage
- SPARQL query engine
- Built in Reasoners
- Plug-in for external reasoners

Application Architecture



Strategy

- Step 1 - Use Cases
  - Describe initial, most difficult requirements in conversational, informal English
  - Work with domain experts to create use cases required by a given domain
  - Use case examples
    - Searching – “What are all the products that have a Vendor of Microsoft and a product name of windows_nt?”
    - Equality – “Determine if two instances are equal”

Strategy

- Step 2 - Ontology creation and validation
  - Use an ontology editor to create an ontology/schema based on the use cases created in Step 1
  - Ontology editor used: Protégé 4.0
  - External reasoner plug-in: Pellet
  - Creation of
    - Classes and corresponding subclasses
    - Properties: Object properties as well as data properties
    - Individuals of a class
  - Run the reasoner to validate the correctness of model

Strategy

- Step 3 - Ontology migration to Jena
  - Create Java classes using Ontology generated in Step 2
  - Java classes are created using Schemangen
    - Input to Schemagen: Ontology.owl
    - Output from Schemagen: Ontology.java
- Step 4 - Data migration
  - Perform Data Migration – Two approaches
  - First approach
    - Mapping relational data to RDF with a mapping tool
  - Second approach
    - Mapping relational data to RDF using database view

Data migration utility–First approach

- Database to Relational Query (D2RQ) allows us to view the relational database as an RDF triples
- D2RQ mapping file
  - Maps database columns to predicates in the ontology
- Use the mapping file to convert the relational database into triples
- A triple is created as follows
  - primary key of table ---> subject
  - column name ---> predicate
  - value of the cell ---> object

Data Migration Utility

- First approach limitations
  - D2RQ is not required when a combined view of different tables is used as is the case with the NVD-CPE database
  - D2RQ does not allow us to update database tables
- Second approach
  - Involves creating a new relational schema that is closely related to the ontology
  - This schema will serve as a stepping stone for the data along the path to the semantic store

Data migration utility–Second approach

- Create a view that combines required columns from various tables
- Read tuples from this view (table) to convert the product information into triples
- The triple is now created as
  - primary key (cpe name) ---> subject
  - predicate based on the ontology ---> predicate
  - value of the cell ---> object

Strategy -Continued

- Step 5 - Reasoning
  - The process by which new triples are systematically added to a graph based on patterns in existing triples.
  - Inference rules
    - Systematic patterns defining which of the triples should be inferred.
  - Steps involved
    - Choose a reasoner - Pellet (External reasoner)
    - Create inference rules as part of the ontology using OWL
    - Run the reasoner
    - Verify the correctness of the inference rules using inferred triples

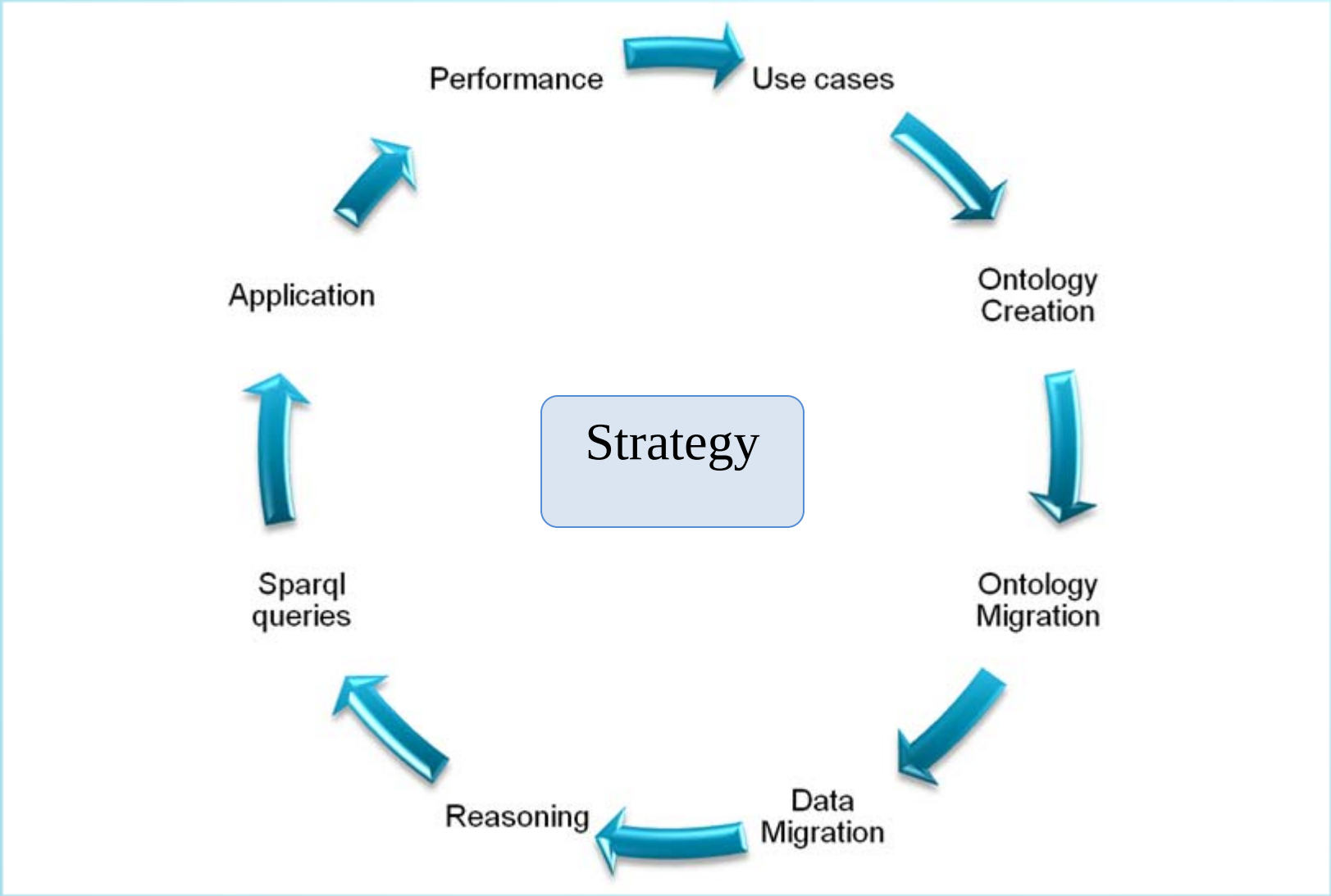
Strategy

- Step 6 - SPARQL queries
  - SPARQL queries are very similar to SQL queries.
  - Write SPARQL queries for each of the use cases from Step 1
- Step 7 - Application
  - Integrate the newly implemented functionality with the web application.
  - Create user interface that enables
    - Navigation
    - Search
    - Querying

Strategy

- Step 8 - Performance with triple stores
  - Performance metrics to test for
    - Load time - Load triples in to triple store
    - Query times - Running time of the sparql queries for various use cases
  - Perform testing on triple stores like RDB, SDB and AllegroGraph and document corresponding performance metrics
- Step 9 - Cyclic process
  - Write additional use case scenarios and repeat the process until all use cases have been modeled
  - Refine model until correct inferences are being drawn.

Strategy - Cyclic Process



Performance Metrics

- RDB,SDB and Allegrograph triple stores are optimized and indexed
- Metrics measure performance on
  - 94216 products without reasoning
  - 5961 products with reasoning
- Example Queries
  - List all the vendors
  - List all the products
  - List products created in given range of time period
  - List all products for a given vendor or given creation date
- Example Queries with reasoning
  - Products containing TCP/IP devices
  - Products containing a given shared library

Performance Metrics: Load Statistics

Metric	Relational View	RDB	SDB	AllegroGraph
Version	SQL Server 05	Jena-2.5.6	SDB-1.1	AllegroGraph-3.2
Size(Rows/Triples)	96485 (R)	982403 (T)	982403 (T)	982403 (T)
Total Space (MB)	13.08	1044.00	302.63	387.00
Index Space (MB)	0.008	674.22	75.55	316.06
Log Space (MB)	-	285.06	82.44	-
Load time	-	231.6 s	284.6 s	164.8 s

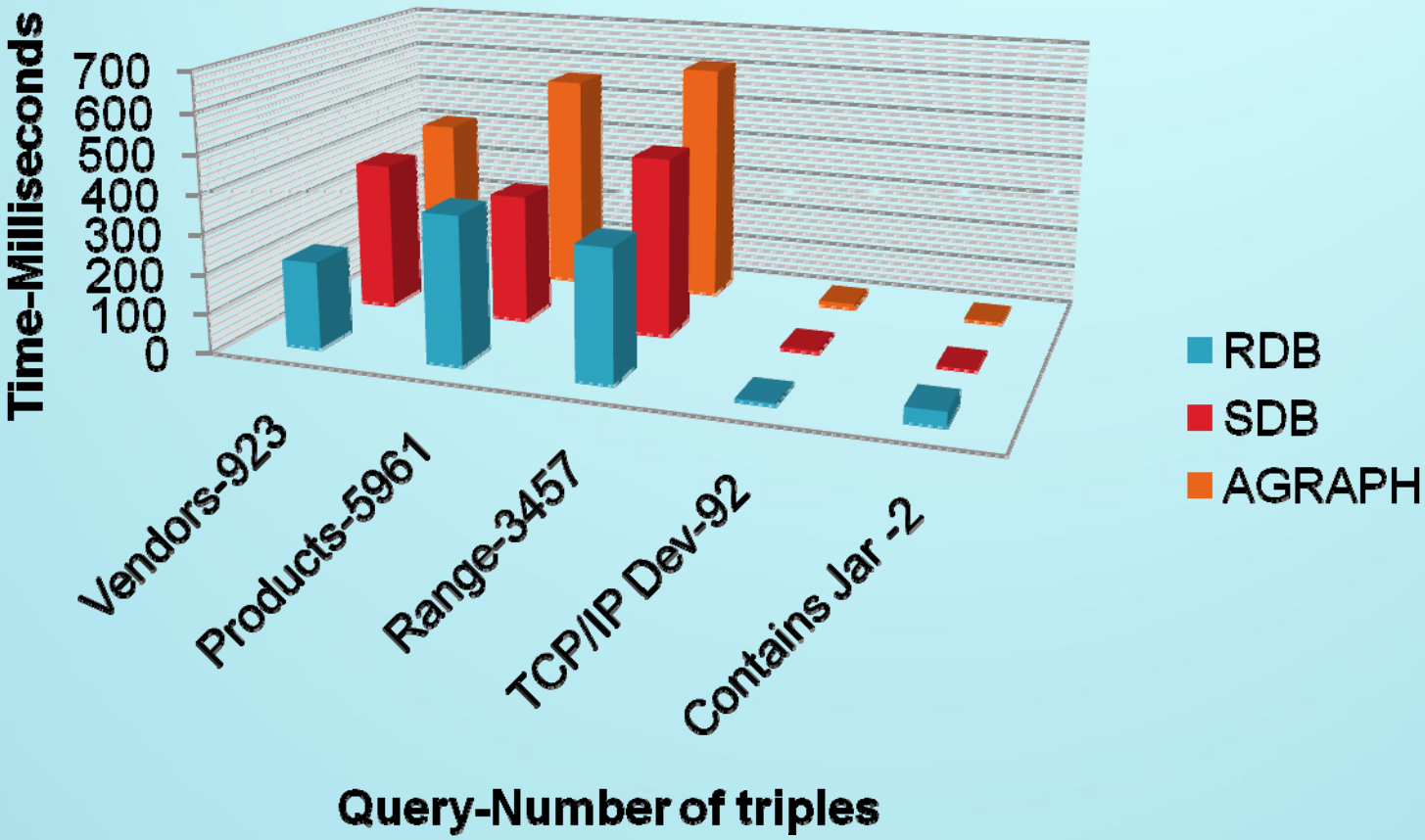
Load time with reasoning

Metric	RDB	SDB	AllegroGraph
Version	Jena-2.5.6	SDB-1.1	AllegroGraph-3.2
Size(Rows/Triples)	97814 (T)	97814 (T)	97814 (T)
Total Space (MB)	118.31	61.38	38
Index Space (MB)	66.98	9.65	31.46
Log Space (MB)	13.31	38.38	-
Load time	18.58 hrs	17.62 hrs	19.06 hrs

Performance Metrics: Query time

Query (Triples)	RDBMS(ms)	RDB(ms)	SDB(ms)	AllegroGraph(ms)
Vendors (9898)	53.2	737.4	711.2	945.6
Products (96216)	10.6	1013.2	723.4	5572.8
MS Products (2616)	12	26.4	30.0	141.4
‘win ce’ Agent (1)	27	74.8	8.4	11.0
All CPE names(96216)	11	1235.0	1274.6	7321.2
Given CPE name(1)	1	838.6	472.2	5425
All creation dates (96216)	8.2	1183.8	1499.4	5464.4
Given creation date (56811)	70.6	937.4	1427.4	5519
Type ‘a’ (82981)	34	749.6	1120.6	5325
Group by Type h=4941, o=8294, a=82981	92.6	768.4	1243.8	5406.2

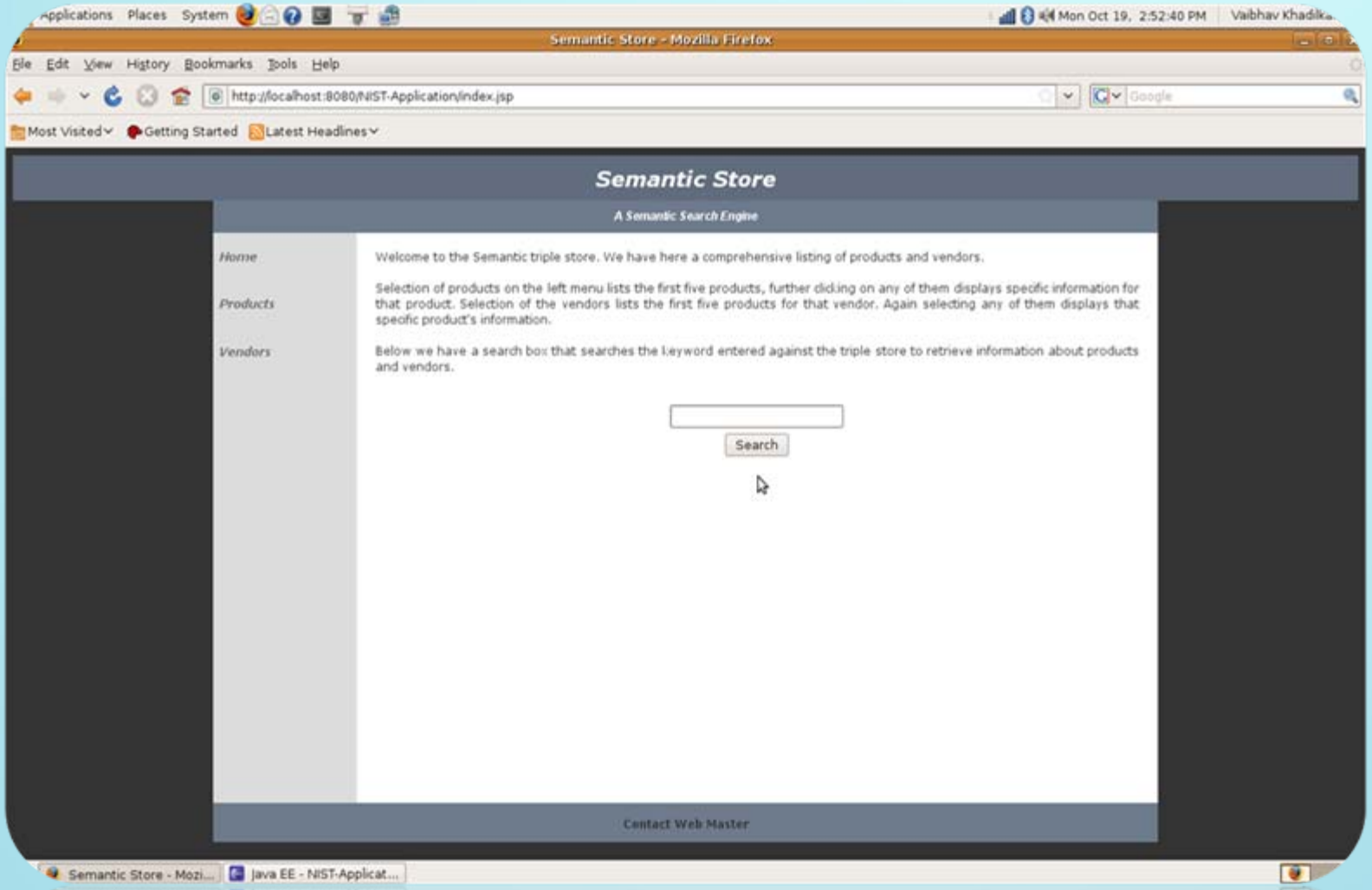
Query times with reasoning



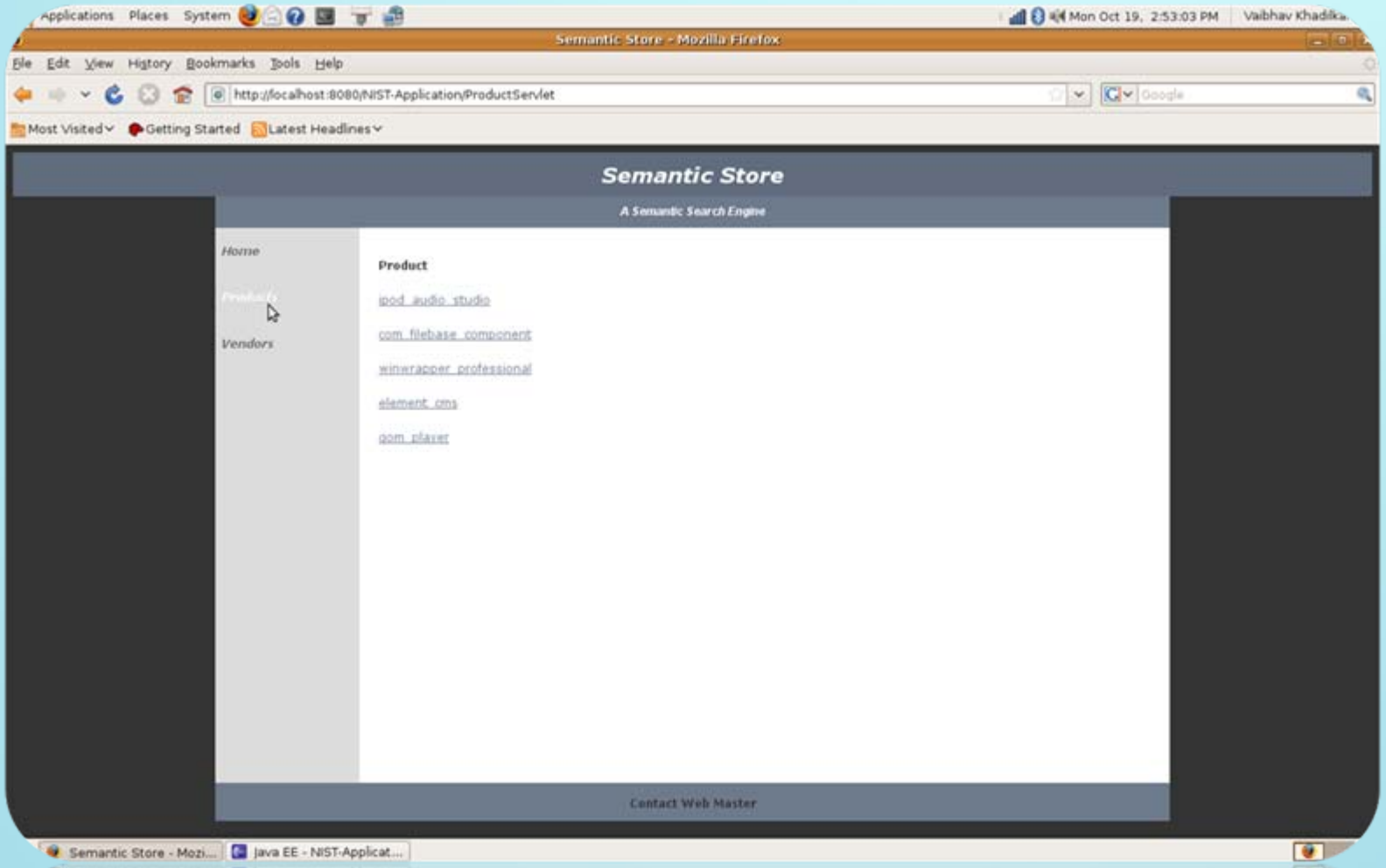
Reasoning Performed on 5961 products

Total Number of products - 96216

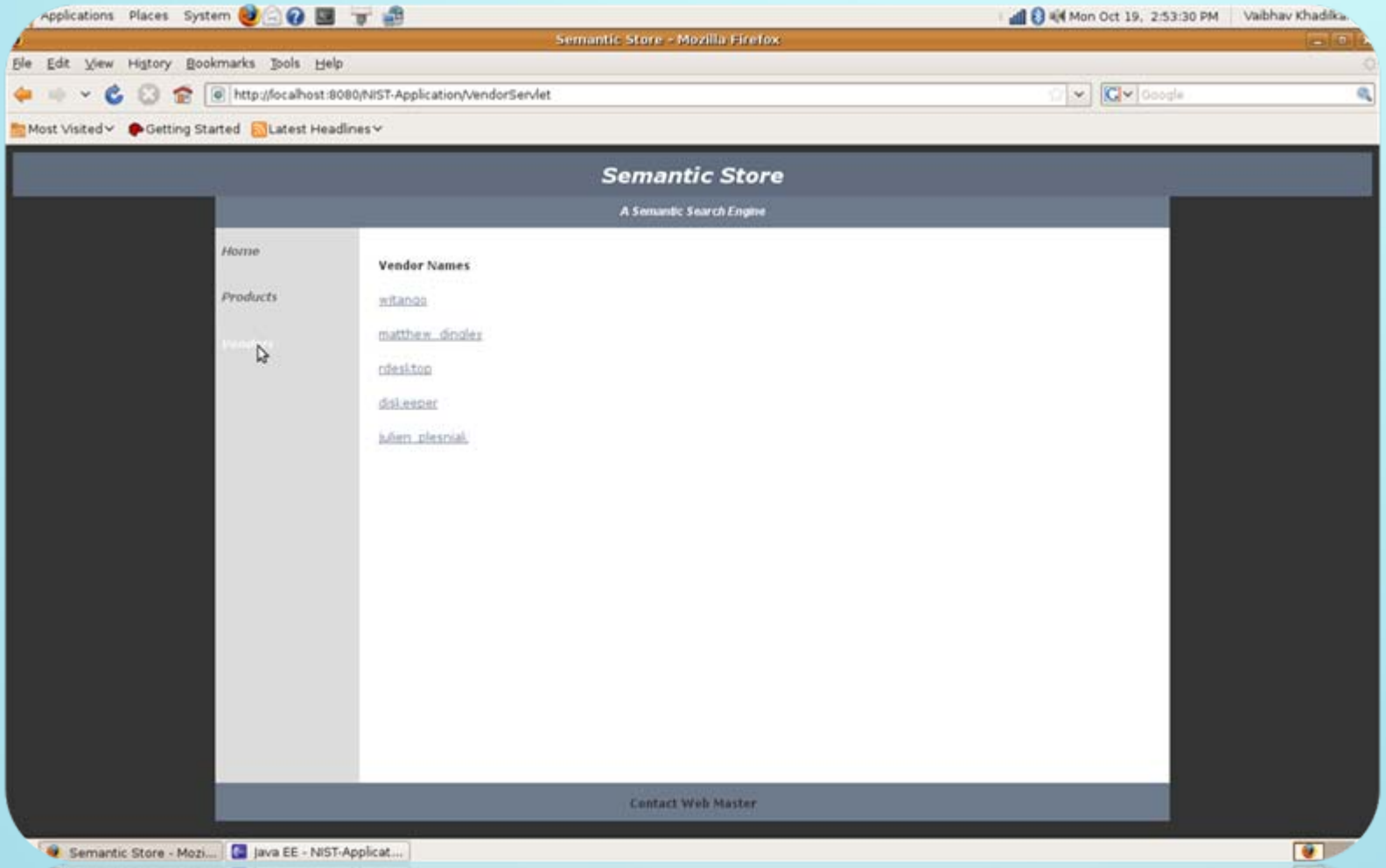
Application



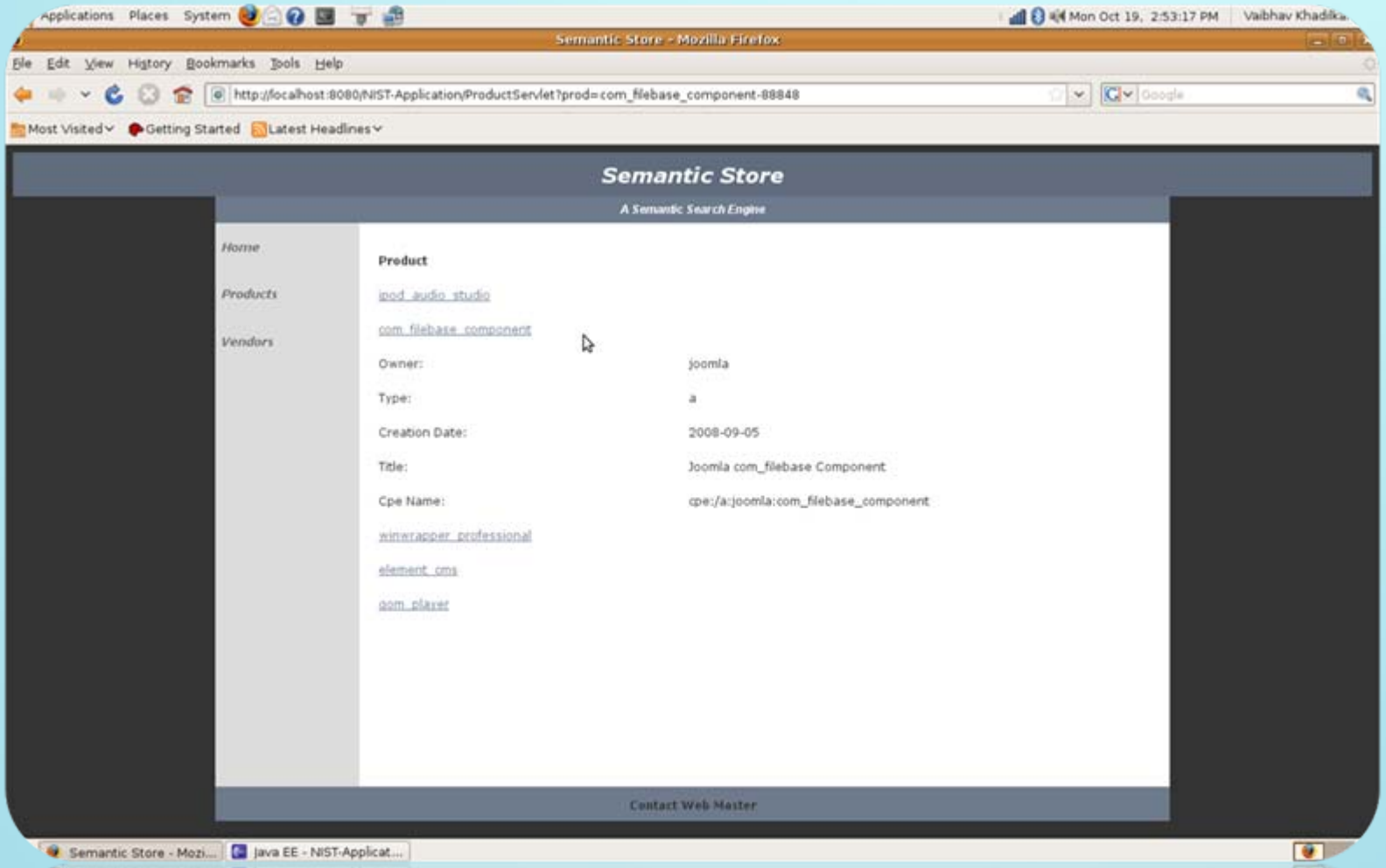
Application



Application



Application



Conclusion

- Choice of semantic model instead of relational model enhances automation of Vulnerability management
- Creating a comprehensive list of use cases at once is challenging.
  - Cyclical process makes incorporation of new use cases flexible
- Efforts must be taken to optimize triple store performance
- Implementation of a system must carefully choose a triple store/reasoner for their implementation
  - Trade-off between speed and power

References

- http://jena.sourceforge.net/
- http://nvd.nist.gov/
- http://www.semanticsupport.org/
- http://www.w3.org/2007/03/RdfRDB/papers/d2rq-positionpaper/
- http://www4.wiwiss.fu-berlin.de/bizer/D2RQ/spec/
- Dean Allemang, James Hendler: Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL
- John Hebeler , Matthew Fisher , Ryan Blace , Andrew Perez-Lopez:Semantic Web Programming

Acknowledgements

- NIST
  - Paul Cichonski
  - Harold Booth
  - Christopher S Johnson
- UTD
  - Dr. Bhavani Thuraisingham
  - Scott Streit
  - Aniruddh Bajirao

QUESTIONS?