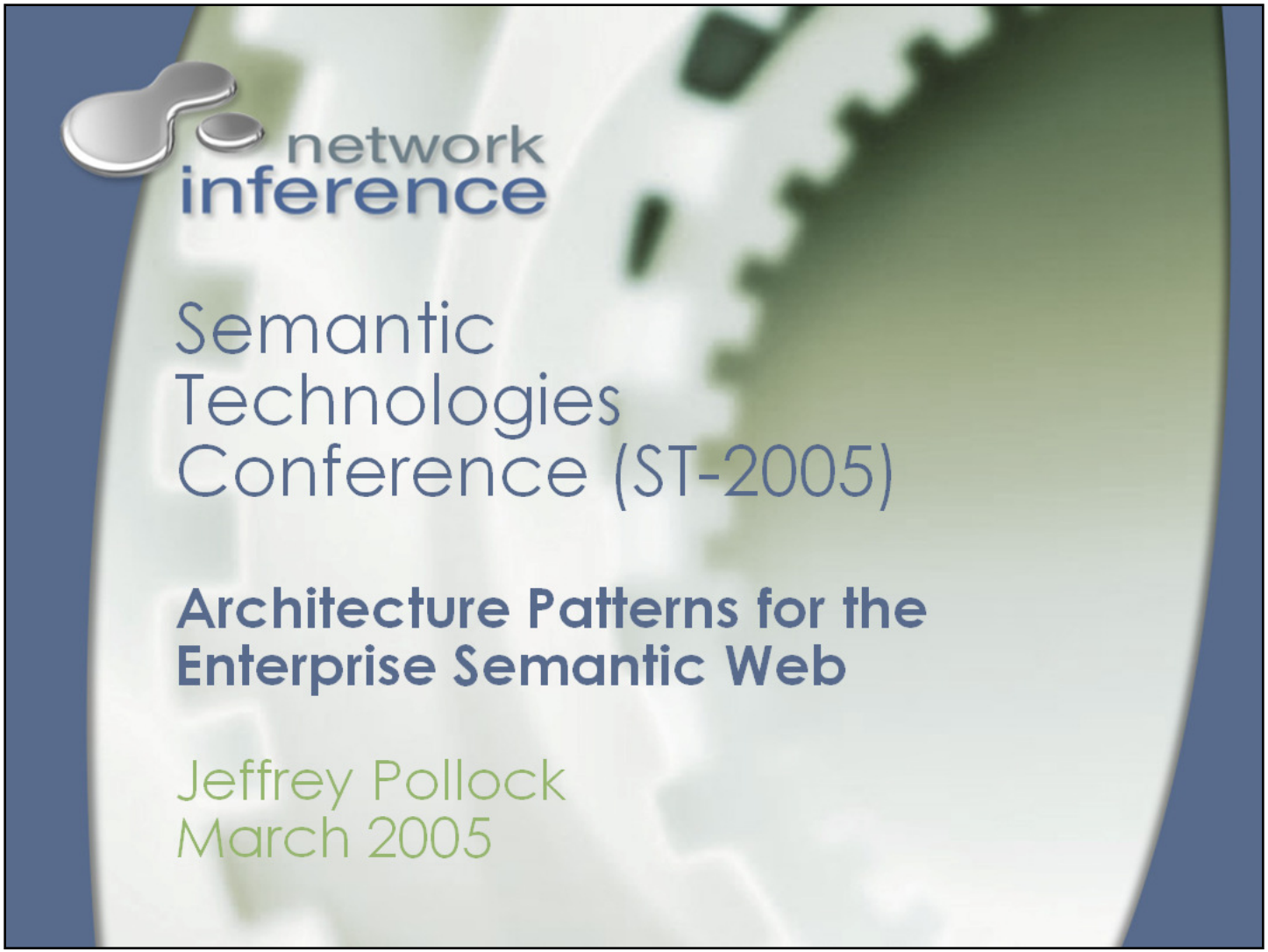
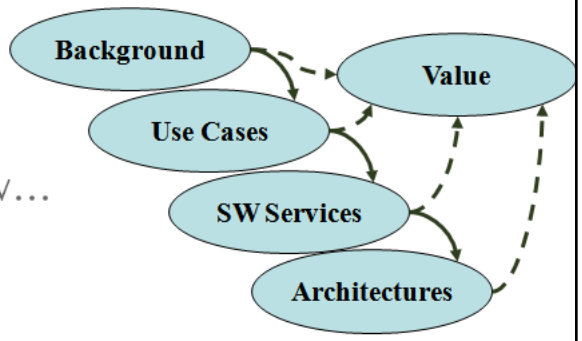


NOTES

Welcome to this presentation about the “Enterprise Semantic Web” – its uses, software, architectures and the way it will change how enterprise information management is accomplished.

Prelude

- Presentation Goals
  - Straight Talk; **No Hype**!
  - **Education** – what, why, how…
  - Explain **Value-drivers** & ROI



- Presentation Takeaways
  - If you're serious about data, you need to be serious about '**semantic technology**'
  - The core enabling technologies and services are available to you now – **RDF, OWL & Core Services**
  - Semantic Web technology is not perfect, it may never be, but it is showing real value for enabling an **Enterprise Semantic Web**

network inference

2

NOTES

The key takeaway from this presentation is that this technology base is (a) here today, (b) standards-based, and (c) a powerful new way of getting control over large quantities of enterprise information. Take the time to learn how these technologies can improve your job.

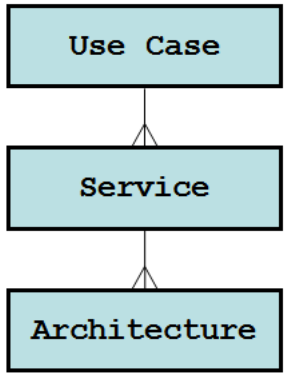


NOTES

There are big things going on here. Historically, information has always been power – both for nation-states and commercial businesses. But our ways of managing information are stone-age in comparison to the rate at which we acquire new data. Something significant and fundamental must change. XML and Java based systems are not the end-state of technology. While W3C standards are not perfect, they offer the most viable path toward a fundamentally better way of managing information.

Agenda

- Semantics Backgrounder
  - Why is any of this important? Should you be skeptical?
- Enterprise Use Cases
  - Technology capabilities for the business
- Service Utilities
  - New software components for enterprise infrastructures
- Architecture Patterns
  - Under the hood look at how they get the job done
- Conclusions
  - Words of promise, words of caution – semantics

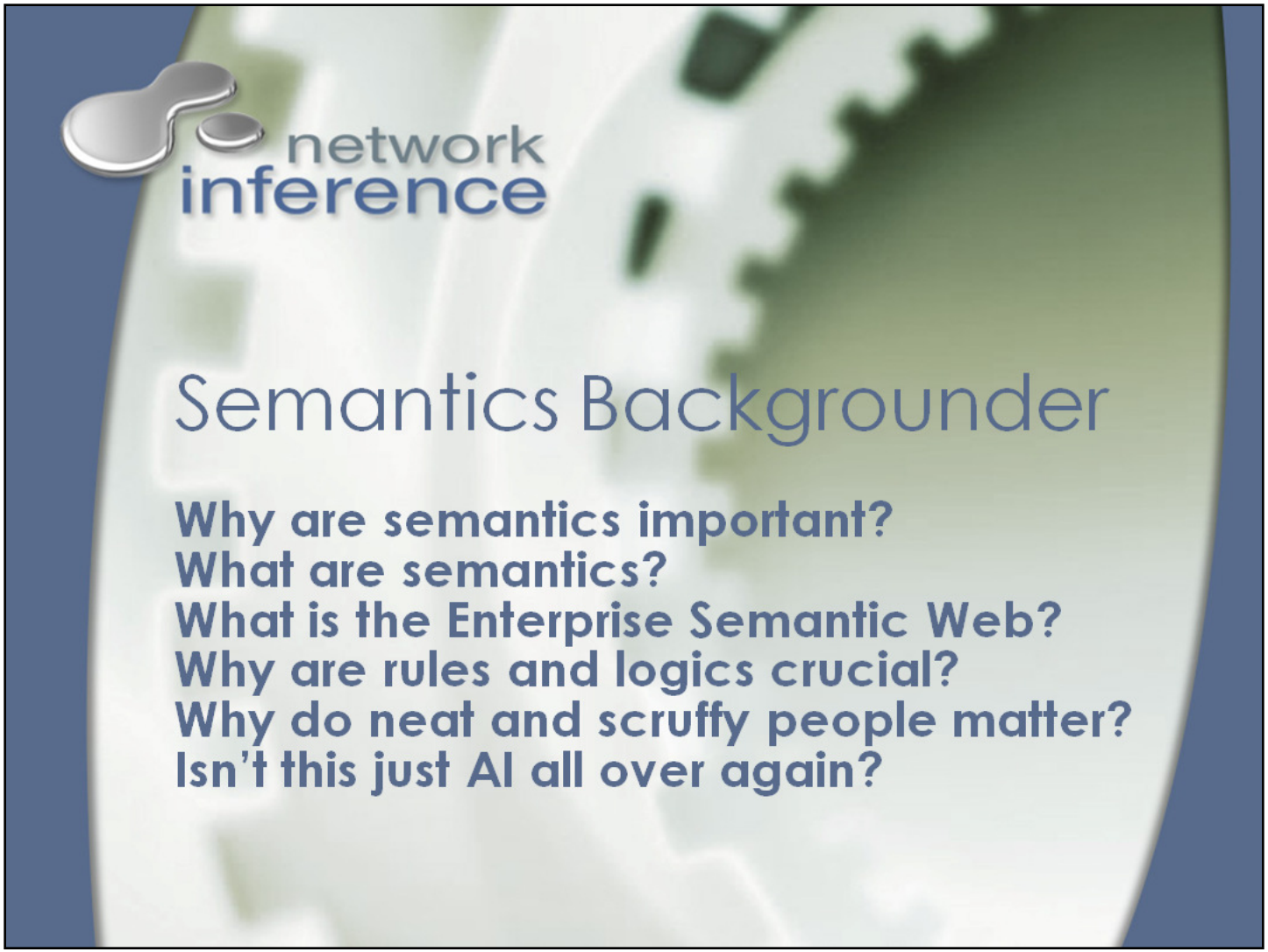


network inference

4

NOTES

For the remainder of this presentation we will be breaking down the problem into discrete areas: a backgrounder to frame a common starting point, business problems to be addressed by use cases, solution architecture problems to be addressed by a services architecture, and semantic web design problems to be addressed by component level architectures. Taken together, these solution areas should provide you with the patterns required to start solving problems in your daily job.



NOTES

The first section addresses several background questions to frame the discussion in a common context.

Information Overload

Q: In General, why are semantics important?

- Maintenance and Support of "Data Meaning" is the largest, most expensive, and time-consuming portion of IT budgets
  - Mars Climate Observer – Feet vs. Meters
  - Y2K – "01" equals 1901 or 2001; What will happen on 1/1/00?
  - Product Hierarchies – Is my PDA a phone or a camera or both?
- Over $200B is spent each year on integration issues involving the meaning of data (semantics)
  - Classic "n-squared" problem is more severe at the data layer
  - Data-to-data interfaces outnumber "pipes"
  - Tightly-coupled = Brittle
- Information growth is accelerating – FAST!
  - 2002-2005 – more new data than all of history
  - 5 exabytes of new digital data created in 2002 – enough for .5 million new Libraries of Congress

network inference

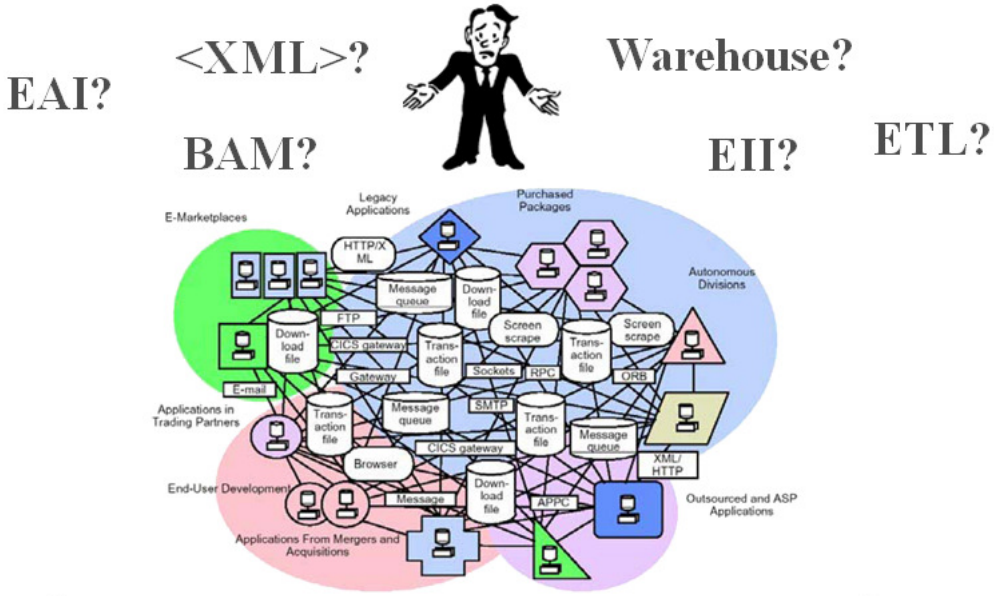
6

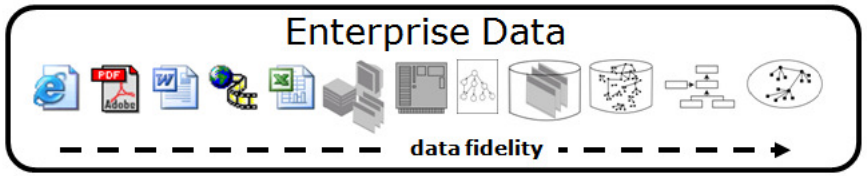
NOTES

Semantics are important, in the most general sense, because they can help tame the chaos created by ever increasing volumes and complexity of data – across industry, society, and governments.

Infrastructure Complexity

Q: For business, why are semantics important?



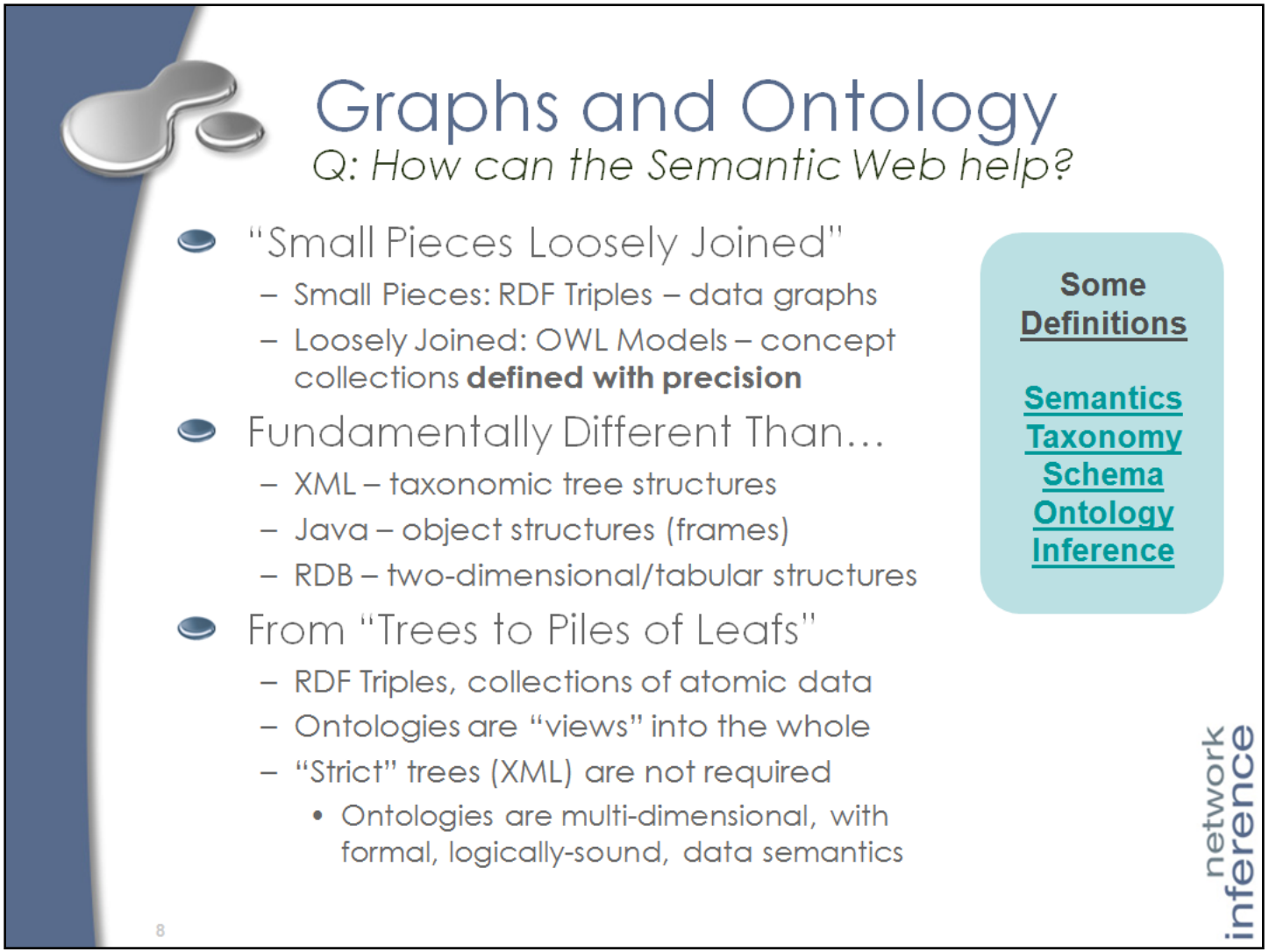


network inference

7

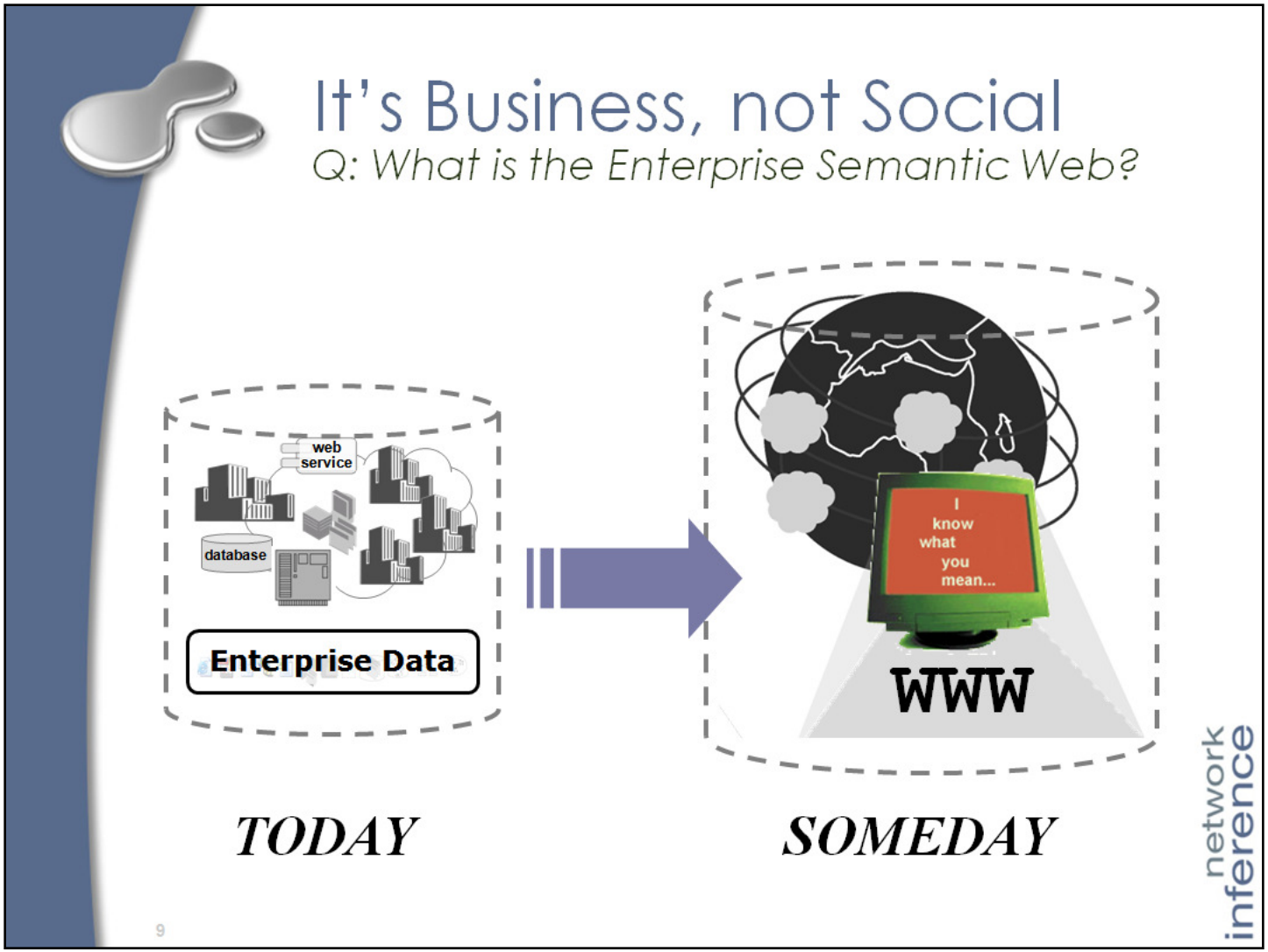
NOTES

More particularly, for businesses, semantics are important because they can help impose a manageable structure on top of an out-of-control infrastructure. Today's popular approaches to the same problem (EAI, XML, EII, ETL, DW…) are not sufficient for long-term manageability. All these approaches do is create more silos of difficult to understand data.



NOTES

So what's different about semantic approaches? The W3C standards RDF and OWL have two foundational characteristics that differentiate themselves from other technical approaches: (1) they are based on graph data systems and (2) they can leverage "true" ontology formats with formal, logically sound, semantics. Taken together, these languages fundamentally improve the way meta/data relate to and evolve with IT systems.

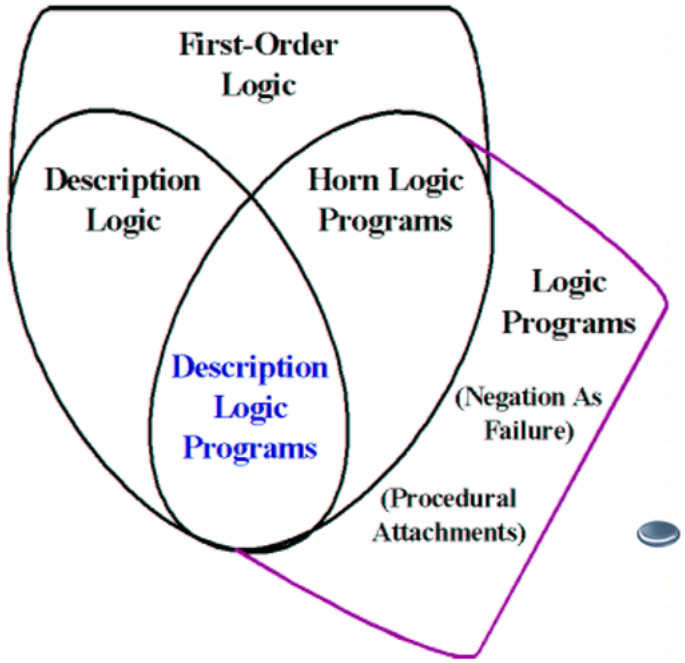


NOTES

In the context of this presentation I am speaking about the "Enterprise Semantic Web" (ESW) – which is a more constrained version of the uber-Semantic Web vision. The ESW is concerned with how OWL/RDF can reshape the enterprise information infrastructure, enable more effective master reference data, and embody business rules and processes alongside data models. The ESW is focused on linking graphs and ontology to traditional legacy formats like XML, RDB, ERP systems and EAI platforms.

Math Matters

Q: Why are rules and logics crucial?



- What if your relational database gave unreliable query responses?
  - Can pigs fly?
  - How many units did you sell last quarter?
  - Is this annual audit report reliable?
- If the data or rule is there, the query ought to know of it…

network inference

10

NOTES

Sometimes people criticize the logical foundations of the Semantic Web. We believe that the mathematical foundations of the languages are crucial to reliability, predictability, and performance. In the same way that relational algebra enables quality RDBMS characteristics, the graph logics and calculus that underlie the Semantic Web are crucial to enterprise adoption. Over time, tool support and education will make Semantics as easy to deal with as databases are today.

Two Sides of a Coin

Q: Why do scruffy/neat people matter?

- Artificial Intelligence (AI) factions and levels
  - "**Neats**" – theoreticians, symbologists, proving truth
  - "**Scruffies**" – engineers, connectionists, getting a job done
- Theory and practice meeting in the middle…
  - Modern RDBMS
  - Nuclear Power
  - Stealth Technology
  - …more great things





Web Ontology Language

Resource Description Framework

network inference

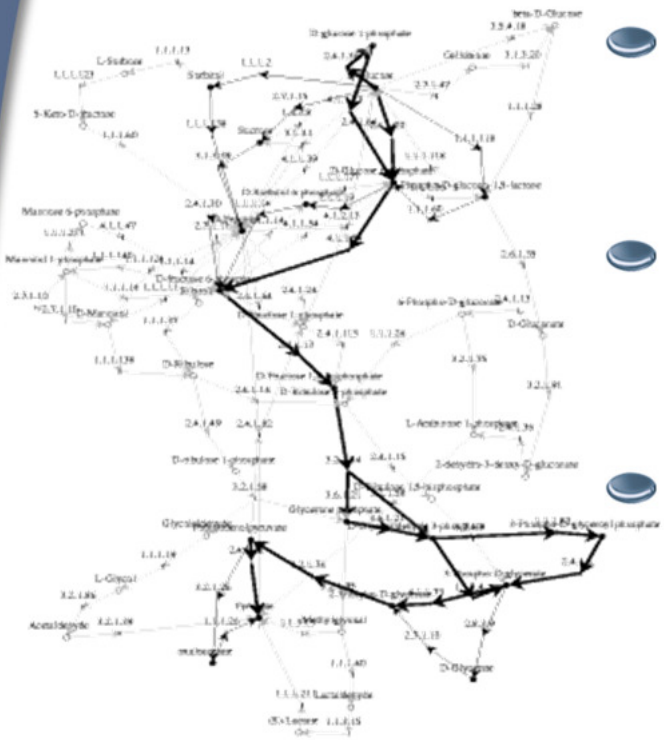
11

NOTES

Historically, the evolution of the Semantic Web can be characterized as the interplay between two kinds of people – the neats and the scruffies. One is more concerned with the provability of the logics of a language, while the other is concerned with how useful the languages are in practice. Both are crucial to the success of a foundational technology. Like other great engineering achievements, the balance of the two perspectives produces breakthrough technology.

"Web-Effect" + Standards

Q: Isn't this just AI all over again?



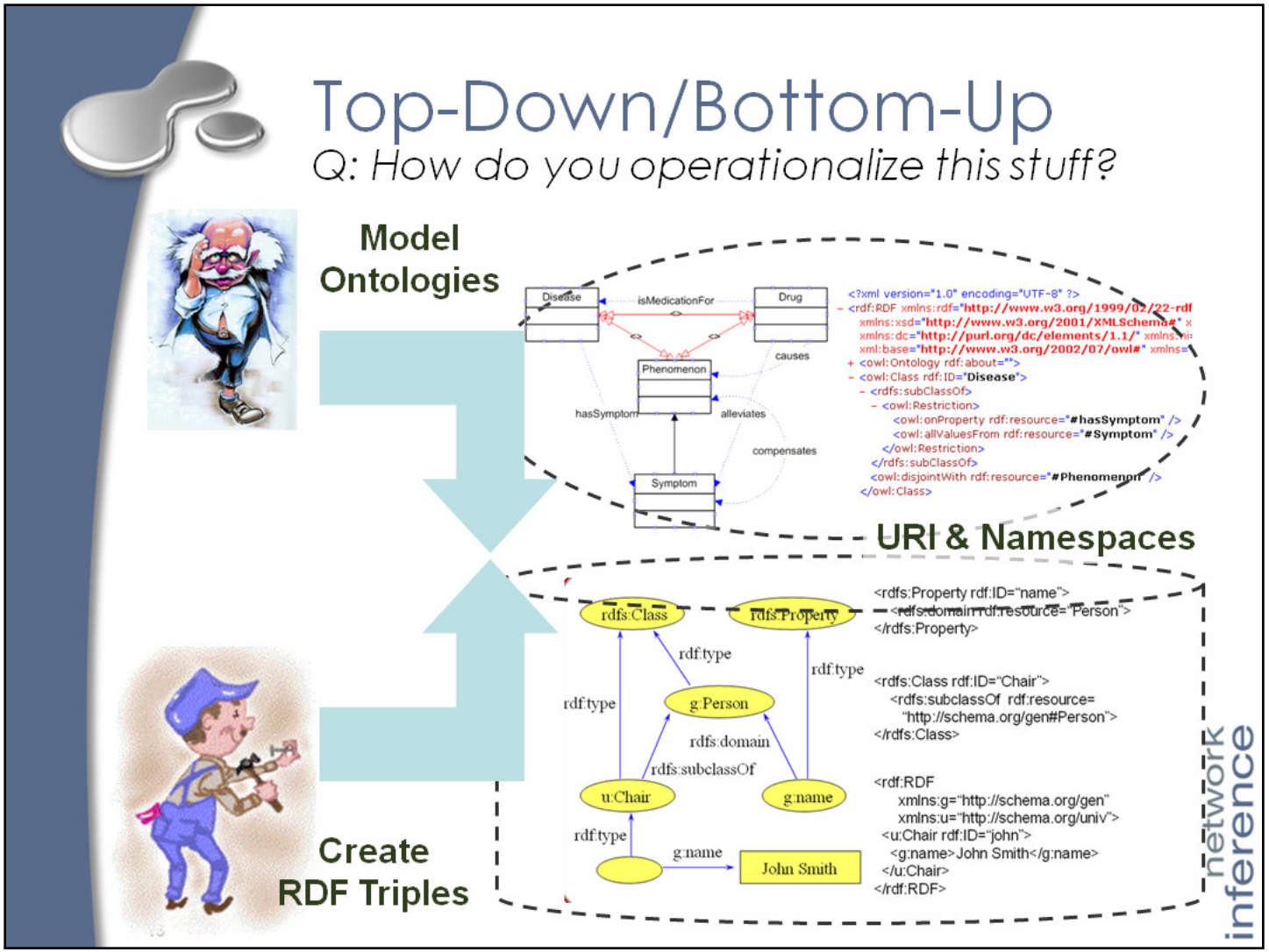
- **Ssshhhhhh!**
  - OWL/RDF are a type of AI "Knowledge Representation"
- **Traditional AI**
  - "Silos" of knowledge
  - Incompatible algorithms
  - Inconsistent "expertise"
- **Semantic Web**
  - Everything has a URI
    - Models, classes, data
  - Predictable semantics
    - Std. Triples & Axioms

network inference

12

NOTES

So, isn't all this just artificial intelligence with a new face? Not really. Two significant changes make these technologies truly disruptive. First, the introduction of the URI as the primary linking mechanism for the model, classes, relationships, and instances creates a "web effect" among your information – which is dramatically unlike any previous AI technology. Second, the standardization of formal semantics (the allowable relationships among data) means that common baseline can be put in place.



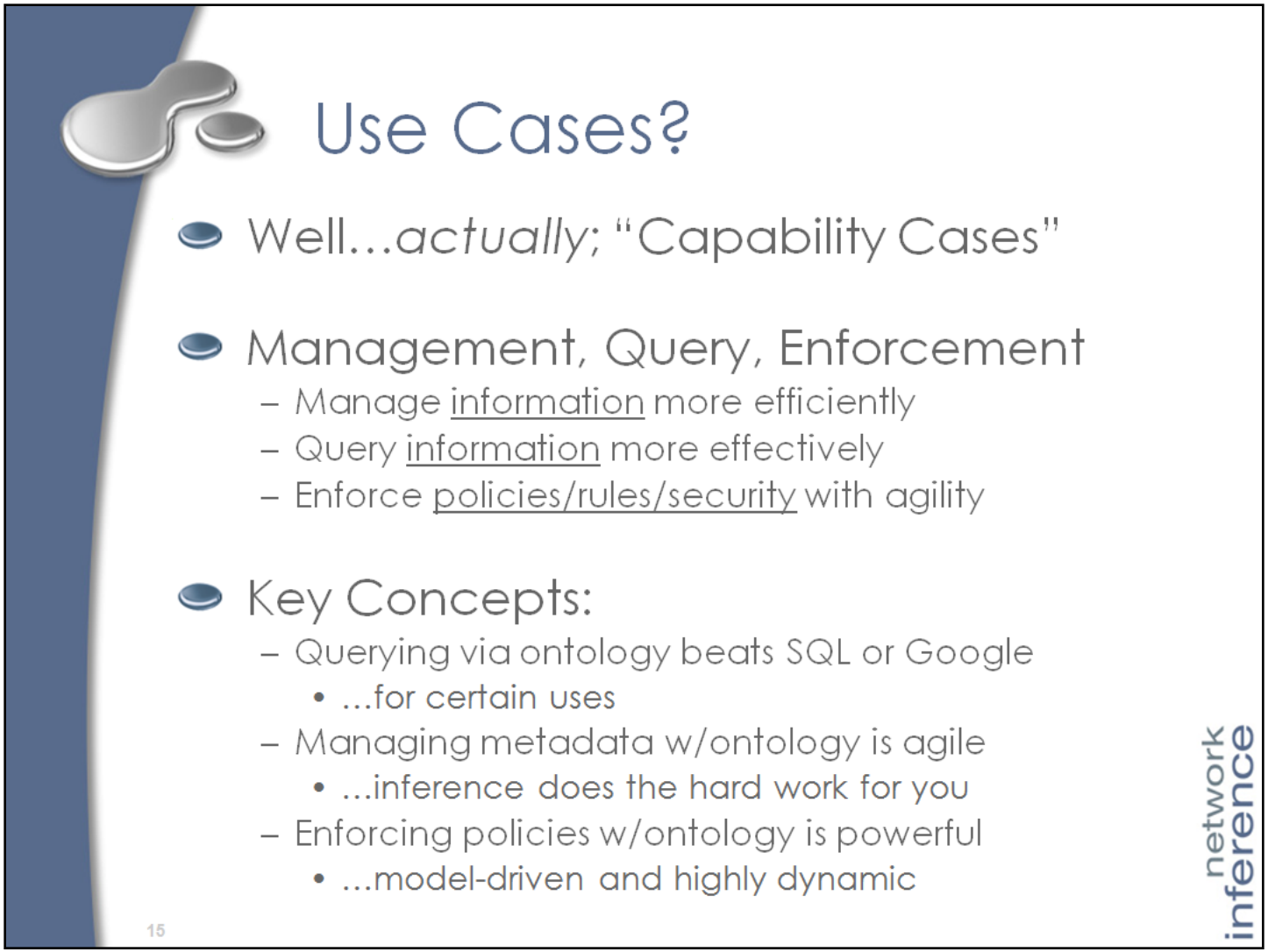
NOTES

Two general approaches are being taken towards adoption. Some people are starting "top-down" by creating ontologies to be connected to IT systems at a later point. Other people are working hard to convert legacy data of all types to a "triples" format; understanding that this gives them incredible flexibility down the road. It is where the two meet in the middle that 1+1 can = 3. Significant new information capabilities are at hand.



NOTES

The following use cases are by no means comprehensive, with respect to the total number of ways semantic web technology can impact business, but these are representative of the sorts of business areas that our customers are wanting realize tangible value.

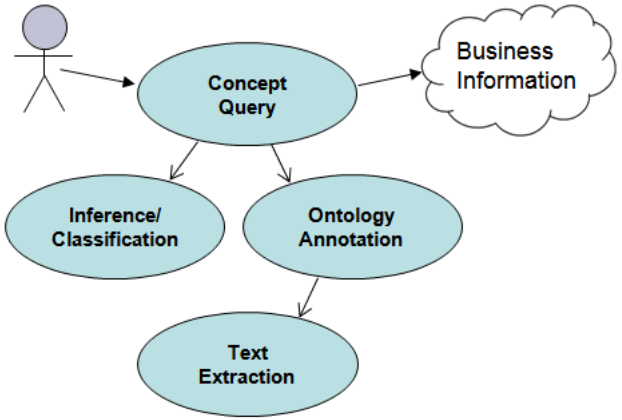


NOTES

People are putting the technology to practice in a variety of ways, which can be articulated by a series of use cases. More appropriately, the following enumeration describes a set of capabilities that are important to business. They can be summarized as improving the way people or machines can manage, query, or enforce policies on top of enterprise data.

Concept-based Query

Use Case	Concept-based Query
Actors	Human Agent, Software Agent, Unstructured Information, Semi-Structured Information, Structured Information
Description	The ability to issue queries about concepts, rather than string matching.
Business Value	Finding the right information faster.
Motivation	Traditional, "bottom-up' approaches for search rely upon symbol matching algorithms, and in advanced cases, latent semantic indexing techniques. These approaches cannot, by themselves, solve problems that require ontology or thesaurus to link words to synonyms and/or the business concepts they belong to.
Why OWL/RDF?	RDF – schemaless data RDF – federated/webbed data OWL – correct, predictable reasoning OWL – "real" ontology (v. XML/RDB) OWL – vendor independence
Key Uses	Search and Retrieval, Guided Navigation, Portals, Federated Query, Data Integration, Business Intelligence



Q: Vehicles made in UK?

A:

Triumph	Motorcycle	Great Britain
Austin Healy	Coupe	Warwick
Centurion	Tank	United Kingdom
Greeves	Cycle	England
Bedford	Truck	London

network inference

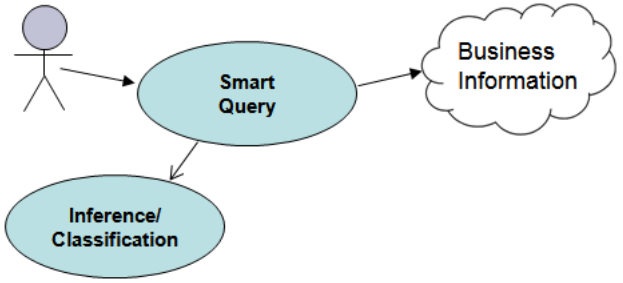
16

NOTES

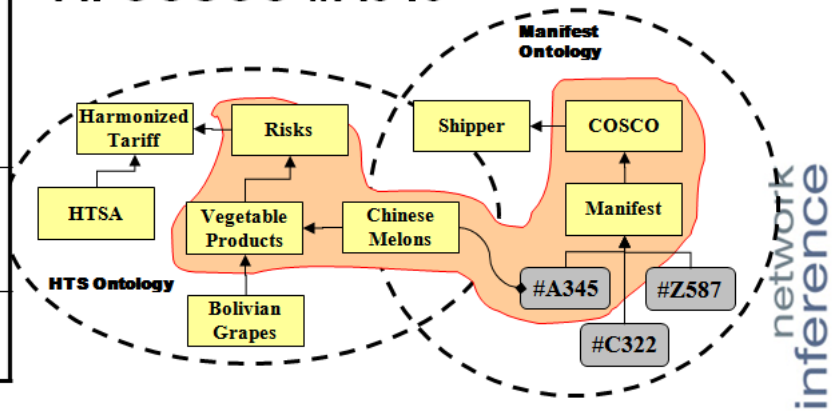
In traditional IT systems people have to query against a physical schema. The physical schema can be confusing, mislabeled, and misused. A separate ontology layer can allow information architects to create more intuitive information models, expose them to end users or machine interfaces, and enable queries to be driven from the concepts, not the physical schemas.

Smart Query

Use Case	Smart Query
Actors	Human Agent, Software Agent, Unstructured Information, Semi-Structured Information, Structured Information
Description	The ability to derive implicit knowledge from stated models in a consistent and predictable way.
Business Value	Finding "hidden" information and rules. Doing so across federated schemas w/minimal code.
Motivation	Traditional search approaches can only join disparate information using lexical equivalencies. The most sophisticated approaches (Google) use proprietary lexical pattern matchers to determine a subset of possible associations. RDB technology relies on manually implemented joins in the schema and in functions/procedures.
Why OWL/RDF?	RDF – atomic, portable instances OWL – correct, predictable reasoning OWL – solid theoretic underpinnings OWL – vendor "plugability"
Key Uses	Search and Retrieval, Federated Query, Data Integration, Business Intelligence, Risk Management



Q: What shipments are at risk?
A: COSCO #A345



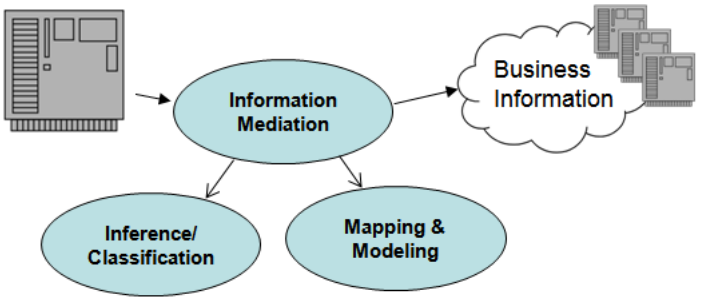
17

NOTES

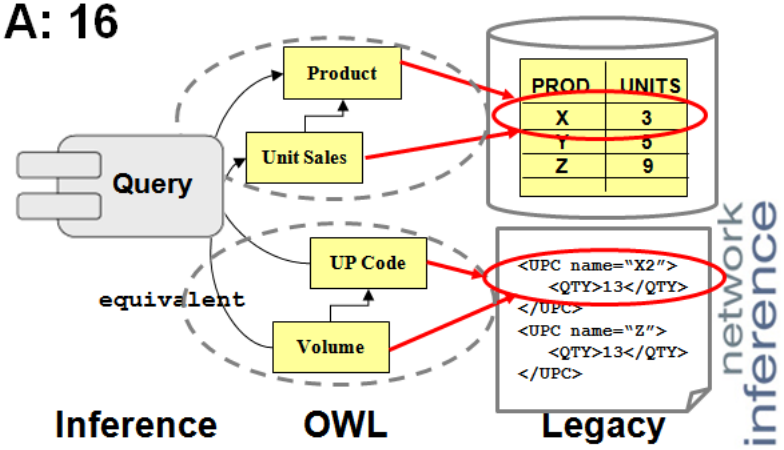
The notion of a smart query is one where inferencing is done at query time to find the logical joins that were not made explicit in the query. Unlike SQL, all joins needn't be expressed on the query in order to ensure that the appropriate matching results are found and returned in the query response.

Information Mediation

Use Case	Information Mediation
Actors	Software Agent, Unstructured Information, Semi-Structured Information, Structured Information
Description	The ability to mediate information from different sources, formats, and contexts into relevant views.
Business Value	Getting what information you need in the right format for consumption by legacy systems.
Motivation	Traditional EAI, EII, B2B approaches rely on outdated data formats (XML, RDB, OO), which result in brittle, high-effort master reference data. Older formats are merely strucure, not ontology. Additionally, older metadata formats are unsound, incomplete and require ad hoc customizations that don't age well over a sustainment lifecycle.
Why OWL/RDF?	RDF – schemaless instance cache RDF/OWL – easily federated OWL – powerful domain modeling OWL – vendor "plugability"
Key Uses	Integration, Metadata Management, Portals, Federated Query, Data Warehousing, Business Intelligence



Q: How many X product sold?
A: 16

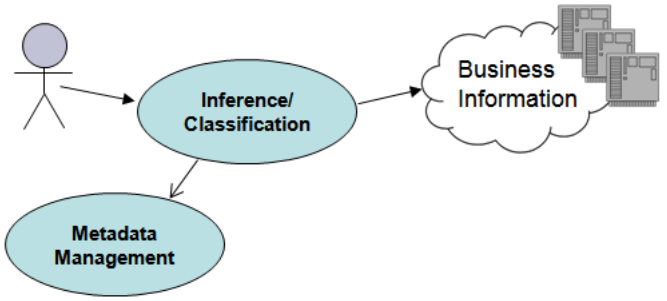


NOTES

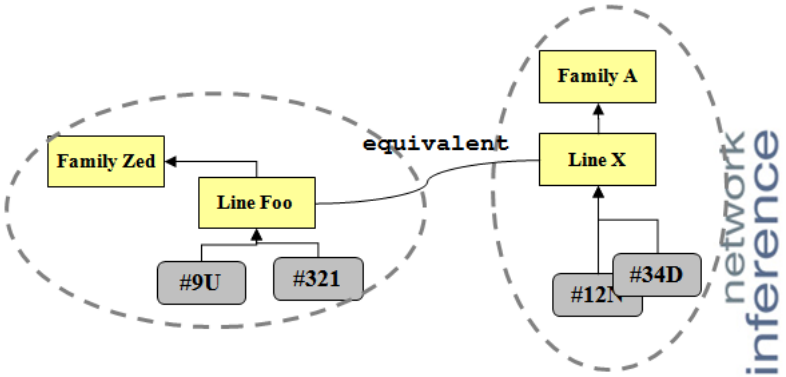
Information mediation is the use of ontologies to overlay disparate IT systems, thereby allowing for a centralized place to retrieve heterogeneous data. Ontologies are a powerful approach because they can capture higher-level business domains and still link them to the more confusing physical data schemas underneath.

Taxonomy/Rule Alignment

Use Case	Taxonomy/Name/Rule Alignment
Actors	Data architect, Metadata analyst, Software Agent, Legacy data schemas
Description	Knowing what you know. Building master reference data for IT alignment across programs, organizations and software product families.
Business Value	Reduced taxonomy and business rules maintenance for complex IT environments.
Motivation	Approaches that do not use ontology – XML repositories, Thesaurus approaches, and RDB data dictionaries – are extraordinarily difficult to monitor, evolve, and perform "what-if" scenarios with. Even small changes require schema rebuilding, query reformulation, and application re-coding.
Why OWL/RDF?	RDF – schemaless repository RDF – RDB style transactions OWL – powerful domain modeling OWL – "evolvable" taxonomy OWL – classification business rules
Key Uses	ERP term alignment, Financial Reporting, Regulatory Compliance, Identity Management, Portals, Business Intelligence & Analytic Reporting



Q: What products are type X?
A: #9U, #321, #12N, #34D

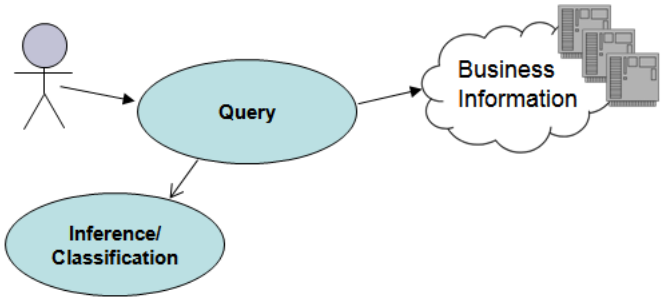


NOTES

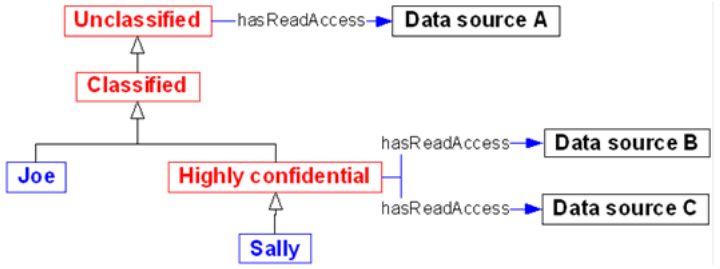
Many organizations have significant problems managing vocabularies and names. These problems can surface in companies with many different product lines, or companies that need to share master reference data across different software platforms in a highly consistent manner. Ontologies provide for a highly adaptive way of managing these vocabularies and rules because inference engines can dynamically reclassify data and rules depending on the context of who is issuing the query.

Multi-level Security/Views

Use Case	Information View Management
Actors	Data architect, Metadata analyst, Security controller, Software Agent, Legacy data schemas
Description	Use a distinct model-based policy layer for identifying profiles and access privileges to federated data.
Business Value	Reduced reliance on hard-coded security permissions layers; enabling more agile, yet fully secure, information dissemination – especially for federated data.
Motivation	Authorization techniques – used to create secure data views – typically require a large degree of coding and integration because of the need to secure data at the smallest atomic level. In homogenous environments this is usually not a problem, but the challenge becomes exponentially difficult in heterogeneous systems.
Why OWL/RDF?	OWL – multidimensional permissions OWL – "evolvable" security OWL – closed loop guarantee
Key Uses	Portals, Business Intelligence & Analytic Reporting, Service-Oriented Architecture fine-grained data security



Q: Who can see comp data?
A: Sally



network inference

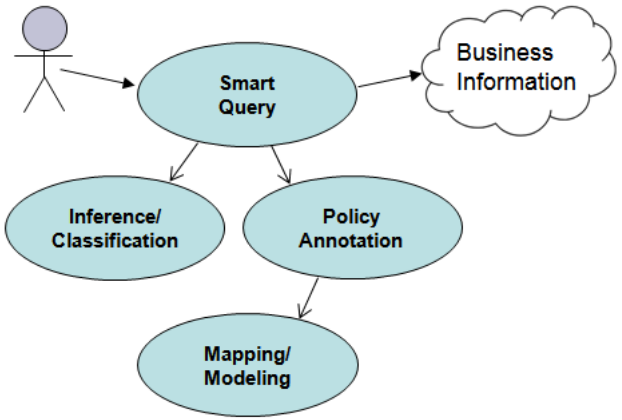
20

NOTES

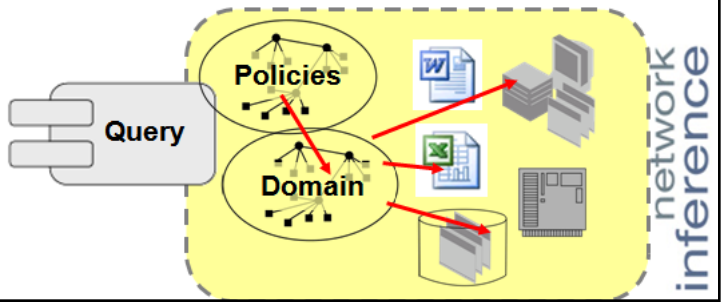
In complicated security environments, with multiple realms, groups, privileges and fine-grained restrictions on individual pieces of data – ontologies are a perfect solution. The adaptive reclassification of small parts of the model, along with the ability to apply classification rules based on knowledge of groups or user identification, allows an inference engine to present filtered views of information in response to queries. Perhaps most importantly, the DL type of inference can guarantee correctness – a closed loop of security with no back-doors into the information (for a given ID).

Secure Audit/Compliance

Use Case	Audit and Compliance
Actors	Regulatory controller, Auditor, Data architect, Metadata analyst, Software Agent, Legacy data schemas
Description	Implement mathematically correct and provable compliance policies (HIPPA, SOX, etc.) on legacy data.
Business Value	More reliable and consistent audit procedures.
Motivation	Although most modern compliance regulations are primarily procedural in nature, most have a distinct IT information audit requirement. Older integration technologies cannot enforce business policy in a global manner because integrations are inherently fragmented.
Why OWL/RDF?	OWL – strict semantics for policies OWL – federated, yet global OWL – accurate, repeatable inference OWL – policies linked to strategies
Key Uses	Financial Reporting, Regulatory Compliance, Identity Management, Portals, Business Intelligence & Analytic Reporting, Sarbanes Oxley, HIPPA



Q: Where are my SOX policies?
A: Reporting ontologies



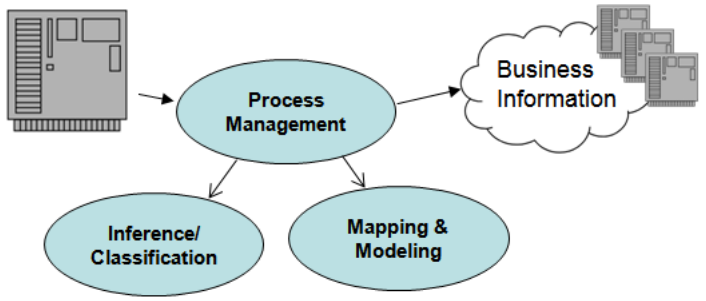
21

NOTES

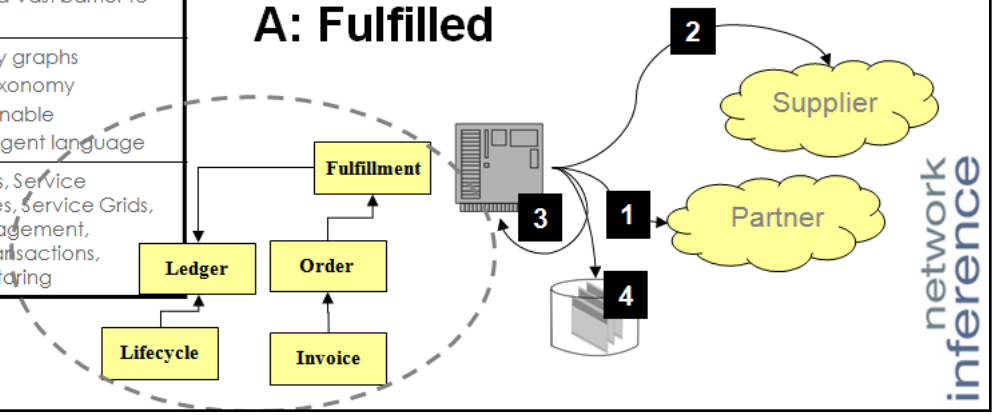
With the increasing attention these days to audit, compliance, and other financial and health care requirements, it is worth pointing out that the combination of the previous two use cases "Taxonomy/Rule Alignment" and "Multi-Level Security" can be used effectively as a baseline for reporting capabilities. In other words, the combination of adaptive information views plus guaranteed correctness of queries (according to the ontology) provide an ideal foundation for audit and compliance reporting.

Process Orchestration

Use Case	Process Orchestration
Actors	Integration architect, Business Process Analyst, Software Agent, Web Services
Description	Allow software and services to coordinate long-lived and complex business transactions by using the semantics of an interface description.
Business Value	Less integration overhead for complex system-to-system interactions.
Motivation	Integration with classic EAI, EII and Web Services still require hard-coded scripts to manage multi-step transactions. For environments, particularly B2B, where services may be transient, this creates a vast barrier to efficient connectivity.
Why OWL/RDF?	RDF – federated query graphs OWL – "evolvable" taxonomy OWL – machine actionable OWL – purpose built agent language
Key Uses	Semantic We Services, Service Oriented Architectures, Service Grids, Business Process Management, Business to Business transactions, Business Activity Monitoring

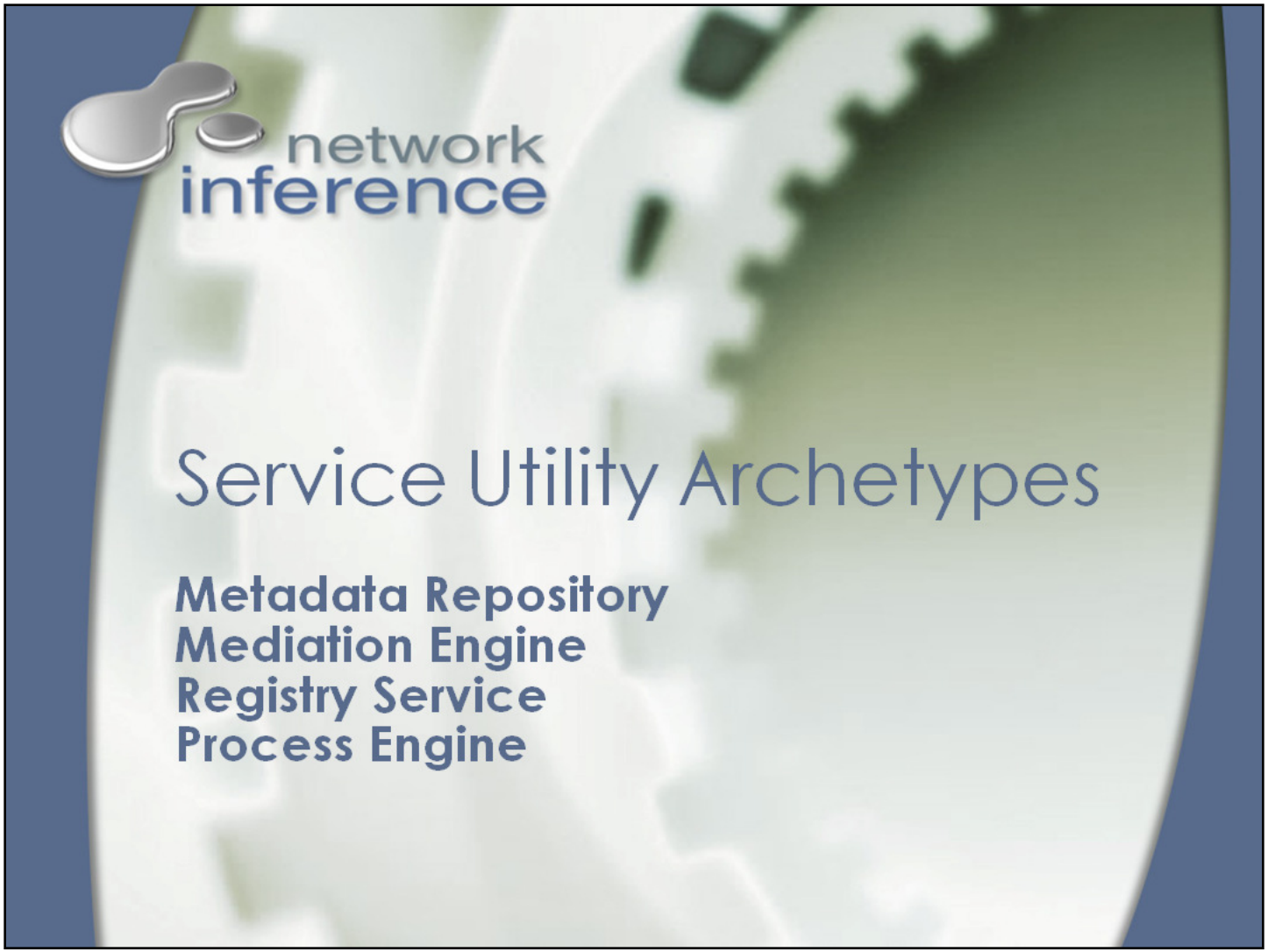


Q: Resolve invoice status?
A: Fulfilled



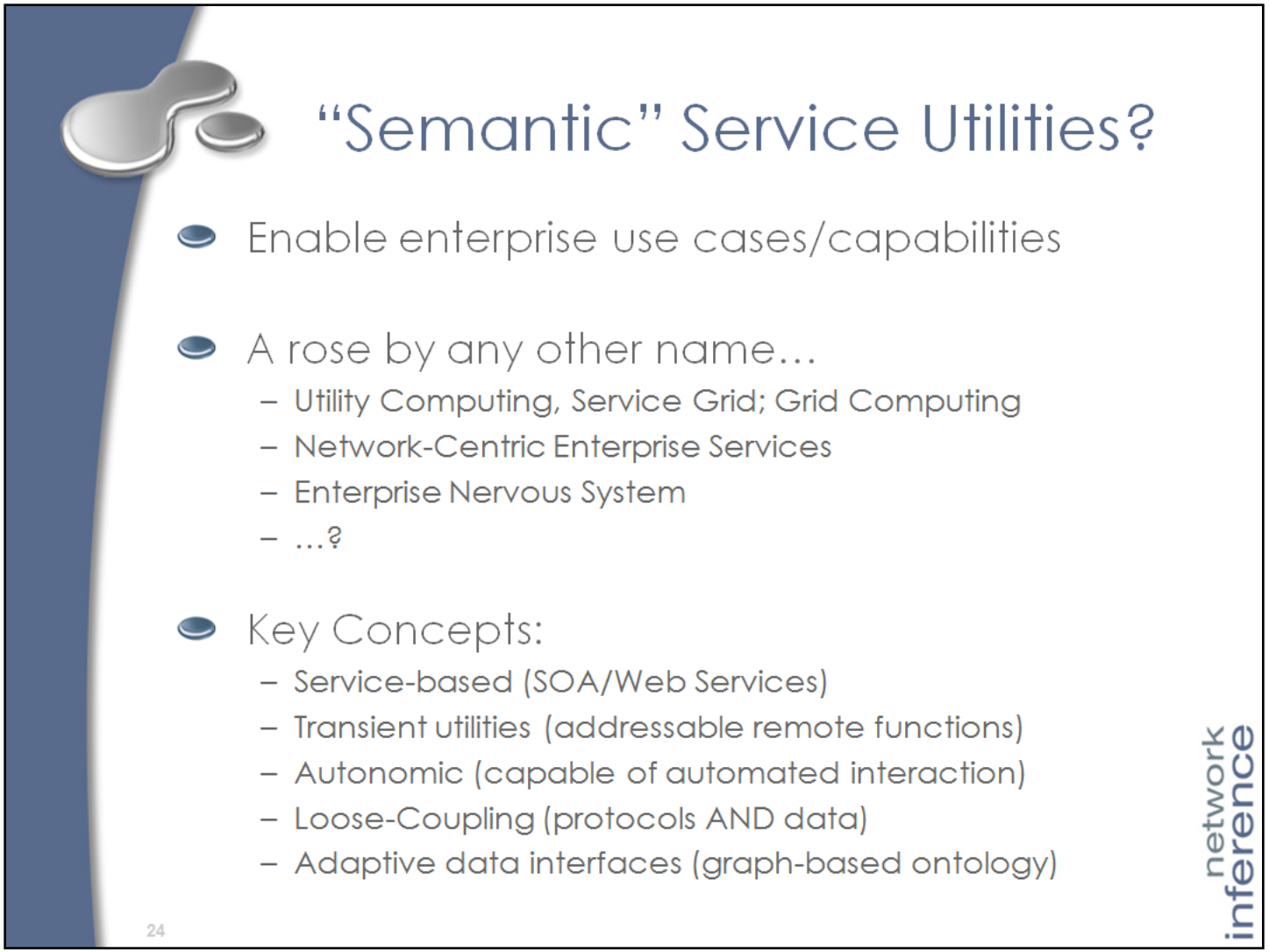
NOTES

Under the banner of "Semantic Web Services," much work is being done to realize the vision of automatic business process coordination. While not fully a reality today in any commercial tools, many testbeds have proven important capabilities that show how autonomous agent can automatically discover and collaborate with previously unknown services. This is the first step in an important network capability for tomorrow.



NOTES

The following section begins to define some key service utilities that can be deployed within the context of an SOA, to deliver the use case values discussed in the previous section.



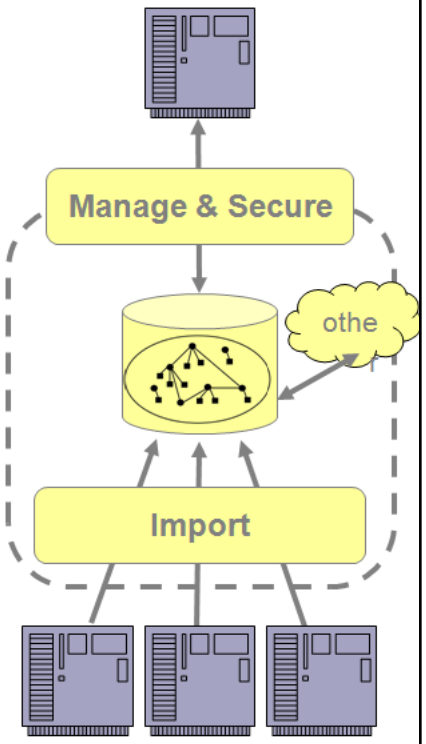
NOTES

Now that we've explored some of the use cases, you might be wondering what kind of software can actually make these come to life. Semantic software tools have been evolving lock-step with Web Services. The combination of the two forces is exponential. Today, services are emerging that are SOA capable software utilities that can enable the use cases discussed in the last section. The vision has been around for awhile under different names, but the capabilities and goals have remained consistent.

Metadata Repository

Metadata repositories are transaction safe, workflow-capable locations for storing enterprise vocabularies, taxonomy, names, content blobs in a highly interconnected manner.

- Metadata repositories are services
  - They may be used in applications or middleware
- Metadata repositories are "master references"
  - Master reference data is metadata, rules, and operational data that can be referenced from all enterprise systems reliably – knowing that they contain up-to-date and accurate data from other sources.
- Metadata repositories are semantics aware
  - RDF triples and OWL axioms improve meaning
- Metadata repositories unify organizations
  - Data from different sources, times, and contexts can be centrally managed, linked to legacy systems, and provide a common local for rules and vocabularies.



25

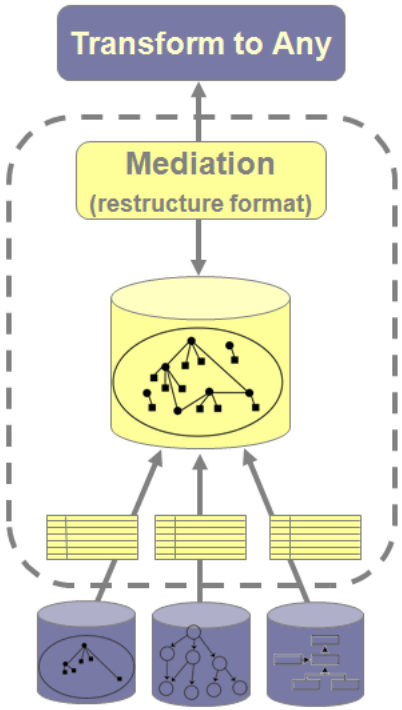
NOTES

Unlike any of the other services described in this section, the metadata repository's core function is to persist available data and documents. Envisioned as an RDF data store, the graph-based nature of RDF provides unparalleled flexibility in the creation and maintenance of service-based information. In contrast to XML and RDB persistence approaches, the core data model need not be rigidly defined or require complicated XPath style navigation methods.

Mediation Engine

Mediation engines are tools that can dynamically output data in different formats and syntax that comply to disparate schema types. They enable automated data transformation.

- Mediation engines are services
  - They may be used in applications or middleware
- Mediation engines work with structured data
  - Unstructured and semi-structured data must first be bound to a schema prior to creating the mediation maps: text extraction and annotation
- Mediation engines let business agree to disagree about schemas and data
  - Use your own XML, relational or OO schemas
- Mediation engines enable interoperability of data from many format types
  - Automated transformation, any-to-any, is possible



26

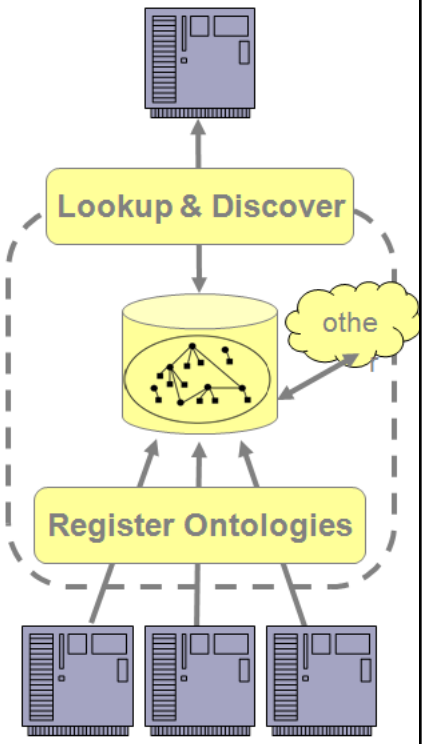
NOTES

The mediation engine's responsibility is to get information from line of business Web services and reclassify – using inference – the data into an appropriate format and context for the requesting/receiving application. Core to this service are XML to OWL mapping/annotation conversion facilities, using this component and a pre-process text to XML indexing service, unstructured text may be handled via mediation as well.

Registry Service

Semantic registries provide a federated registry for the semantics of schemas, ontologies, and applications – they will become the DNS for application semantics.

- Registries are services
  - They may be used in applications or middleware
- Semantic registries are smart directories
  - Registries are a neutral place to register information about services, applications, schemas etc.
- Semantic content registries use OWL and/or RDF for internal metadata and inference
  - Better, standard, data and knowledge representation
- Semantic content registries enable discovery
  - With inference, discovery of new content and services will be as easy as issuing an ad-hoc query



27

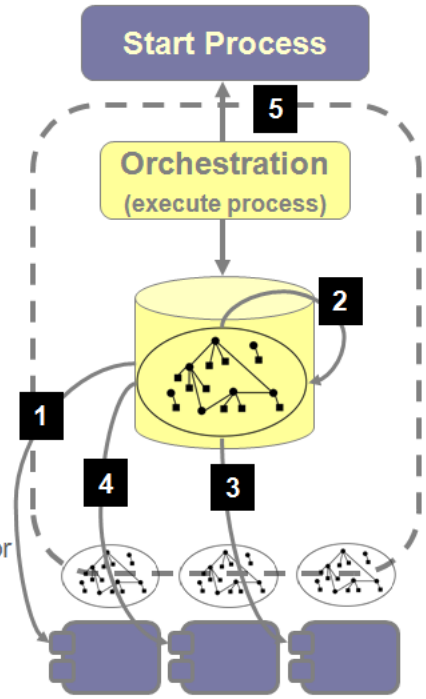
NOTES

The registry service may act as a façade to the repository, or as a standalone document registry – providing pointers to information resources that have been searched for via a SOAP interface.

Process Engine

Process engines orchestrate multi-step and long-lived transactions by intelligently determining which services to use and how to collaborate with them in order to fulfill a business request

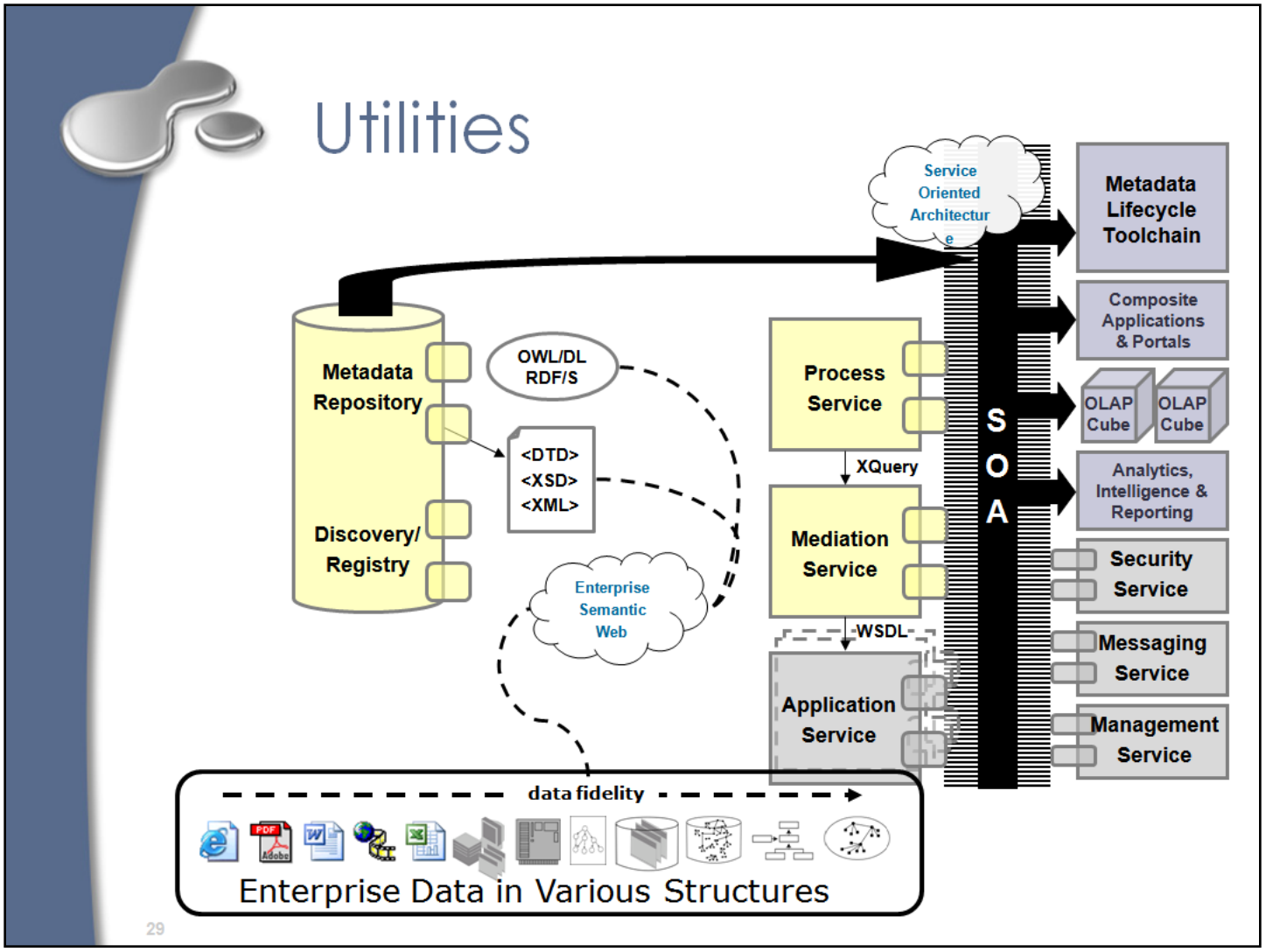
- Process engines are services
  - They may be used in applications or middleware
- Process engines act on workflow/procedures
  - A process engine can interpret a process model and interoperate with annotated web services
- Process engines use OWL ontologies
  - Better, standard, data and knowledge representation for process models as well as interface descriptions
- Process uses Mediation, Repository, Registry
  - Process engines use registry for finding services, mediation for transforming data, and repository for storing process models



28

NOTES

Still in early phases of research, the process engine would provide a multi-step execution environment for ontology-driven processes and long-lived transactions. Although there is much shared vision for the kinds of capabilities provided for in a process engine, the industry has yet to consolidate around a single approach, instead, multiple approaches – including OWL-S, WSMO, and BPEL – have wide followings.



NOTES

The service utilities act in unison, as federated SOA utilities, that can be called independently of one another, or in multiple steps to accomplish a business goal or use case. They all share an XML based exchange doctrine of WSDL interfaces and SOAP transport protocols.

Utility Specifications

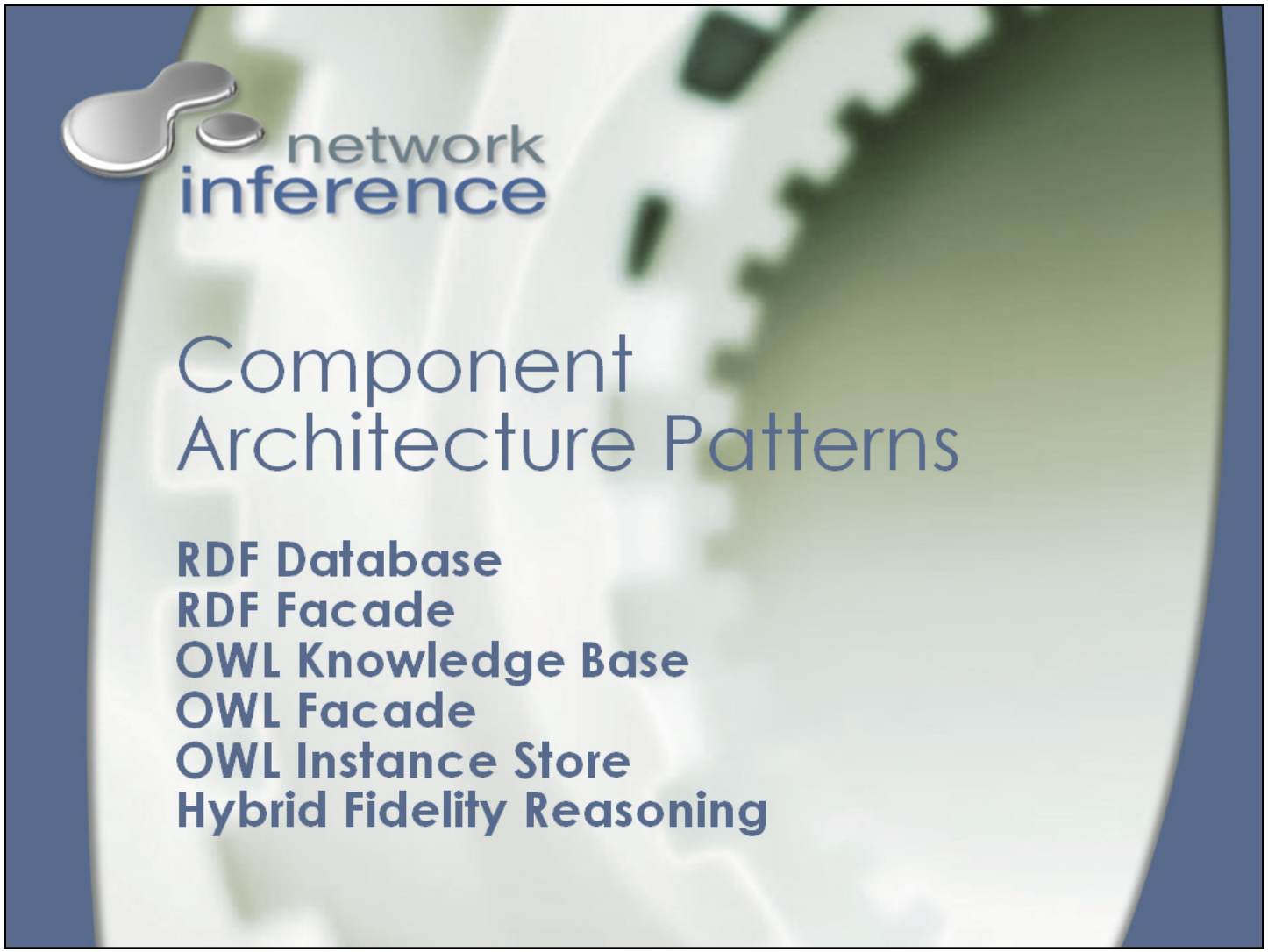
Separation of "semantic" concerns

	Metadata Service	Mediation Service	Registry Service	Process Service
Key APIs	ebXML, JAX-R, OWL-API, proprietary	Query (XQuery, proprietary), XML results	UDDI, ebXML	OWL-S, WSML, BPEL
Transactions	Request/Reply ACID Object-based	Request/Reply Publish/Subscribe Query-based	Request/Reply Query-based	Publish/Subscribe Request/Reply ACID
Input	Management APIs	Query	Query	Management API/invocation
Output	Objects and Documents (XML, OWL, RDF)	XML Documents	Pointers/URIs	Multi-step actions and confirmations
Uses (other services)	Mediation, Registry	Metadata, Registry	Metadata	Registry, Mediation, Metadata
Maturity	Medium (COTS available)	Medium (COTS available)	Low (COTS w/out OWL)	Low (COTS w/out OWL/WSML)

network inference

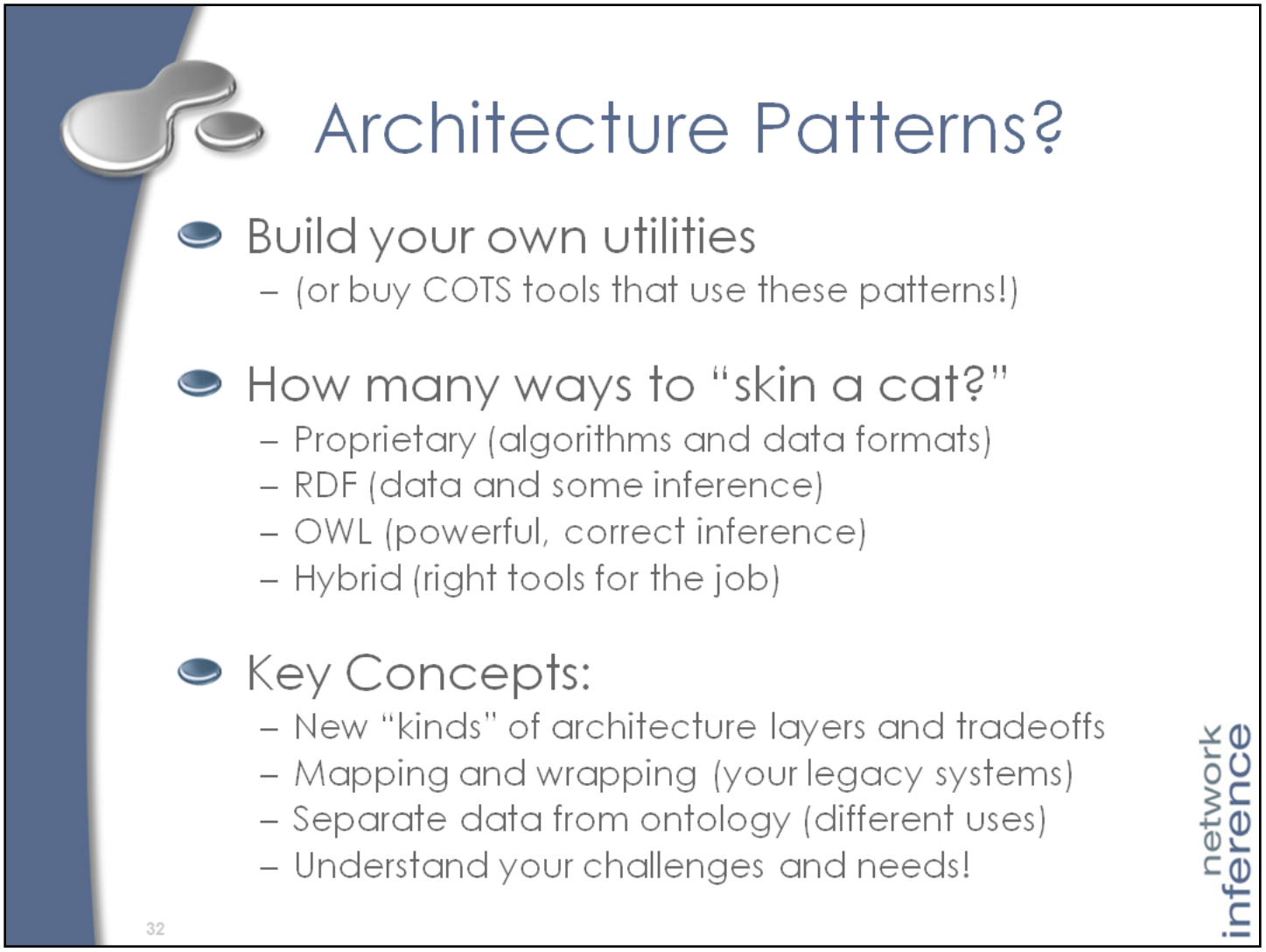
NOTES

In summary, the different utilities accomplish different goals by exploiting different interfaces. The interface defines their function. Not all of these services have the same maturity level, some are quite advanced, while others are still needing foundational research and development.



NOTES

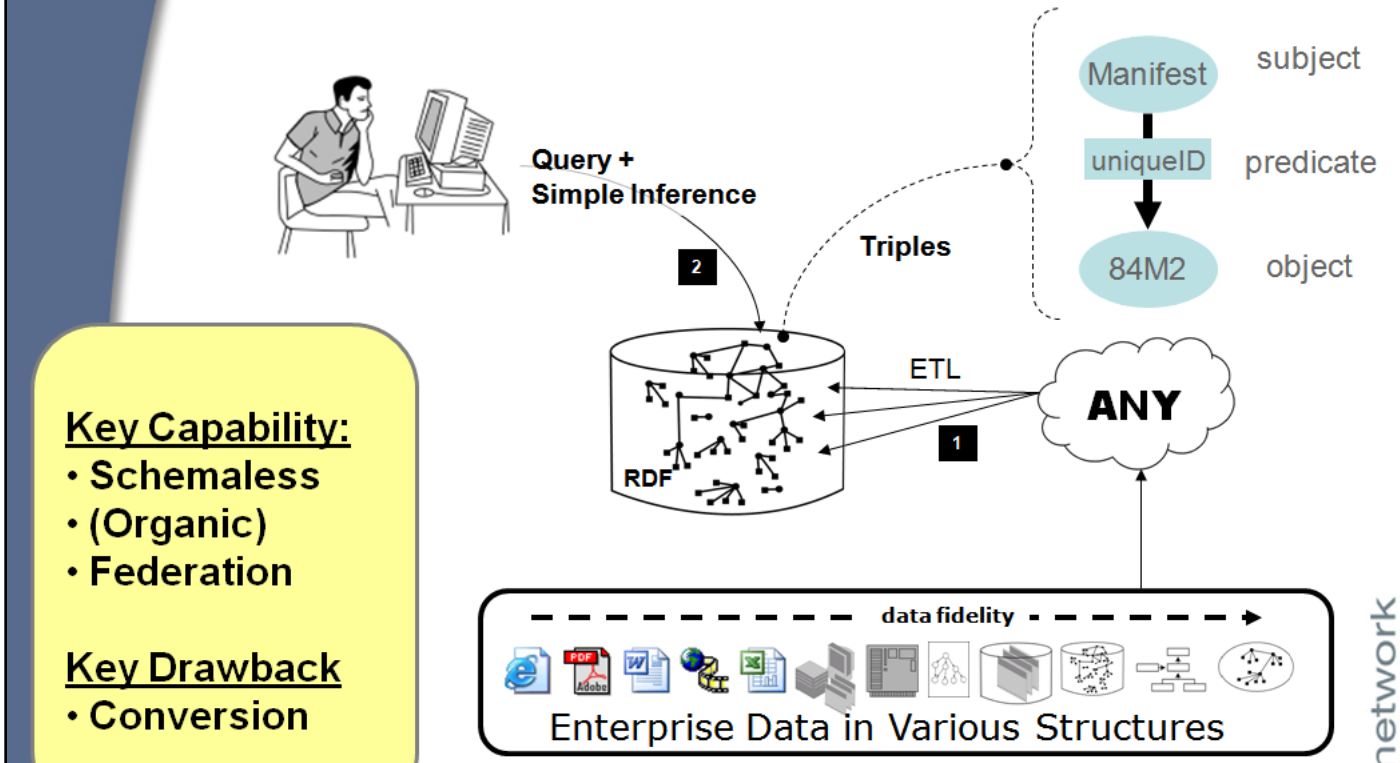
Now that we've gone through some core use cases, and seen how SOA service utilities can bring the use cases to life, it is time to peek under the covers at the different architectures that can be employed to build any one of the given service utilities.



NOTES

So far we've discussed use cases and service utilities, you might be wondering how all of this stuff works operationally. Behind the scenes these utilities can be implemented in different ways, using RDF and OWL to play to different strengths. There is no "one best way" to build RDF and OWL solutions today, but there are patterns emerging, the following section addresses these patterns.

RDF Database (1)



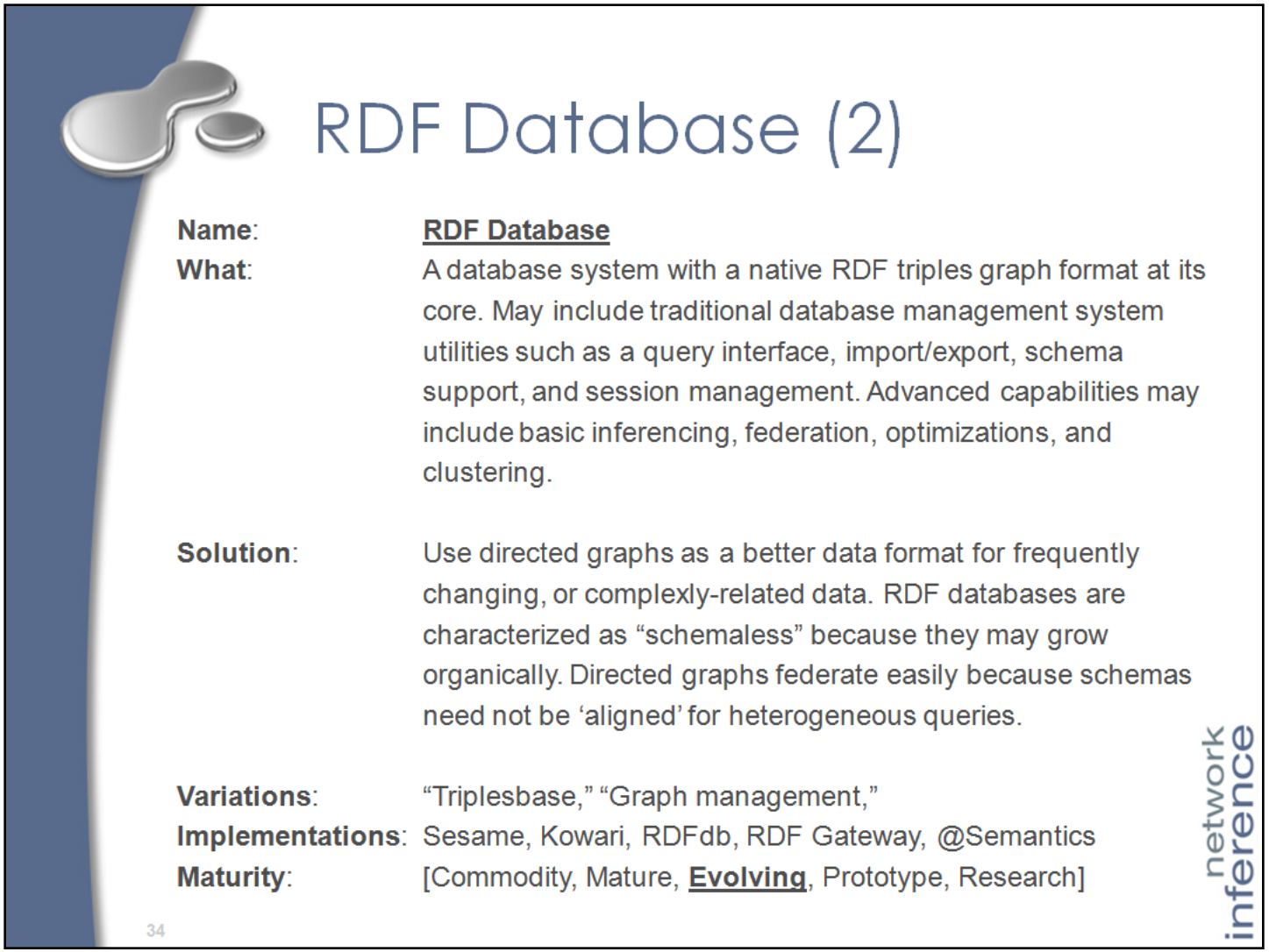
Key Capability:
- Schemaless
- (Organic)
- Federation

Key Drawback
- Conversion

33

NOTES

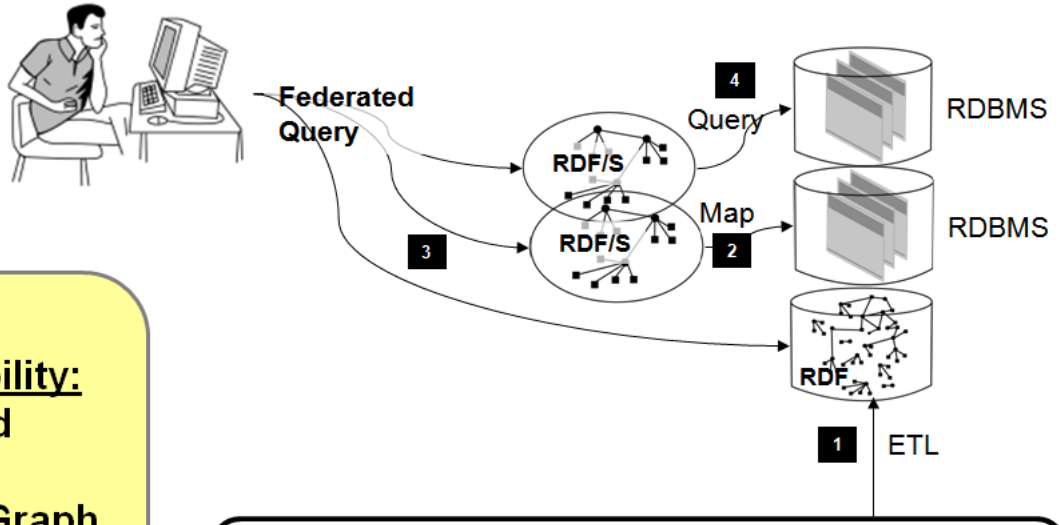
The RDF database is one of the most mature patterns in the software space, having been around in early implementations since the 2000 timeframe. The core pattern is to mimic the capabilities of a traditional database management system, with a different underlying data representation – triples. Today, mature RDF databases will come with complete ACID support, federated query, and an array of management facilities that let a DBA work with RDF data.



NOTES

An RDF database can be an ideal way to persist rapidly changing data that may have unforeseen formatting requirements.

RDF Façade (1)



Key Capability:
- Federated
- (Query)
- Organic/Graph

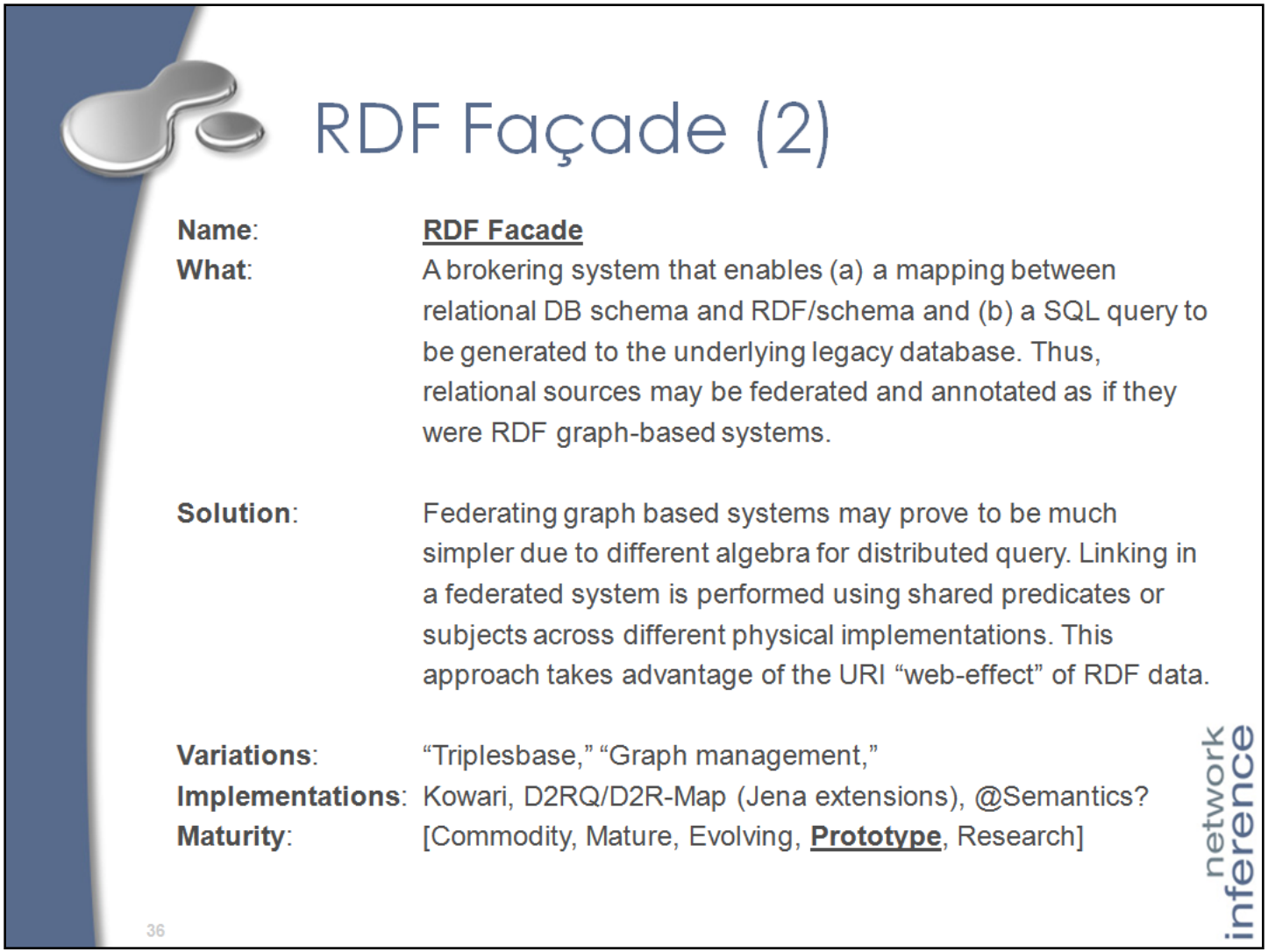
Key Drawback
- Simplistic

network inference

35

NOTES

The RDF façade is similar to the RDF database, but leverages traditional RDB technology for the actual persistence of RDF data. It can be used in two primary ways (1) to provide an RDF interface layer over a greenfield relational database system such that the Façade layer maps closely to the actual physical table space, or (2) to provide an RDF interface layer over a legacy database with more complicated mappings between the RDF graph layer and the legacy relational table structure.



NOTES

An RDF façade can be a simple way to normalize legacy systems into a graph-based data environment and provide a simpler way to federate information across multiple database implementations.

OWL Knowledge Base (1)



Query +
Term Inference (TBox) and
Data Inference (ABox)

3

OWL

DL

In-Memory

2

Query

RDBMS

SOA/
WSDL/
XML

1

Mapping &
Transform

data fidelity

Enterprise Data in Various Structures

network inference

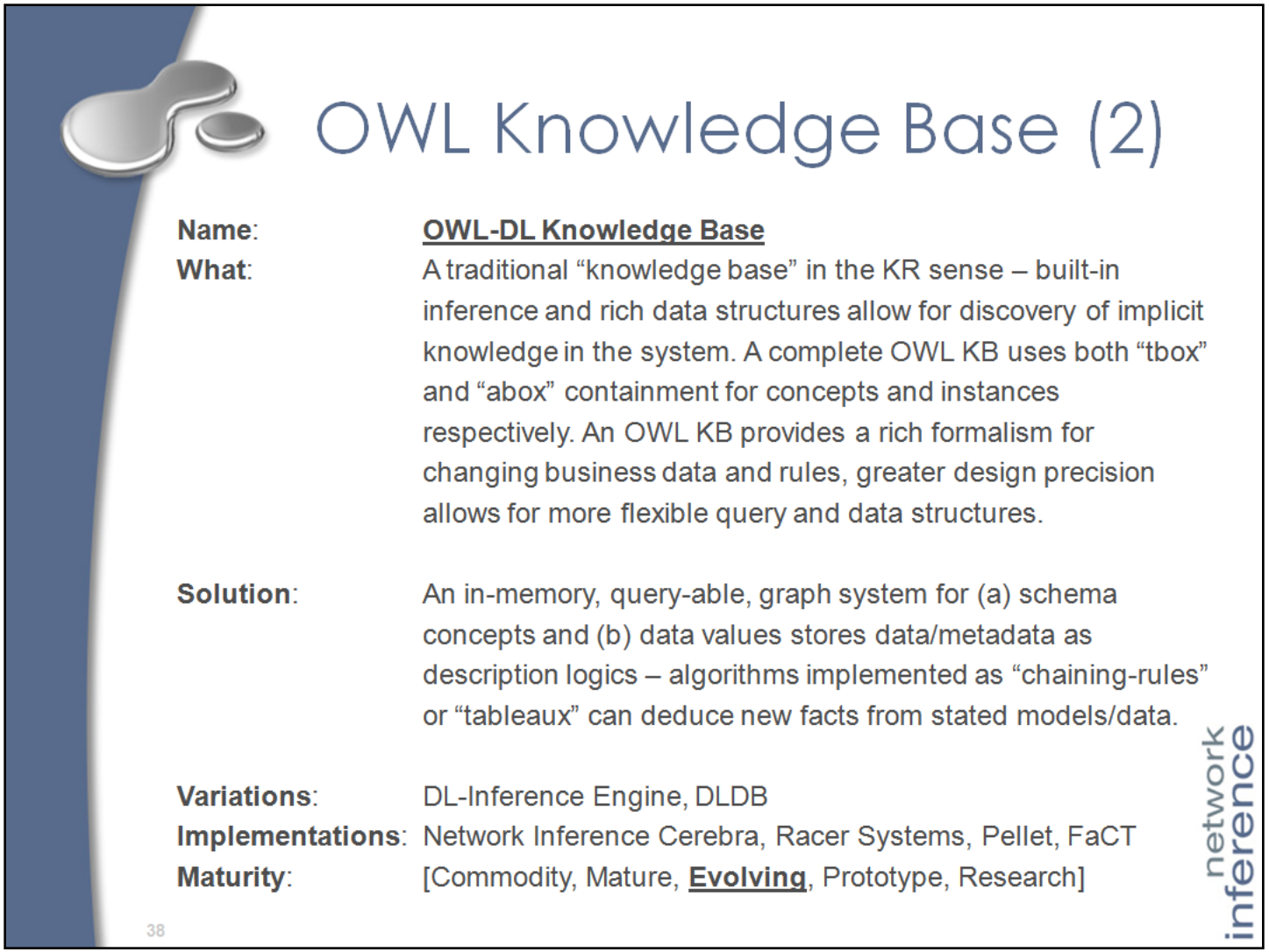
Key Capability:
- Data Values
- (Inference)
- Powerful

Key Drawback
- Scalability

37

NOTES

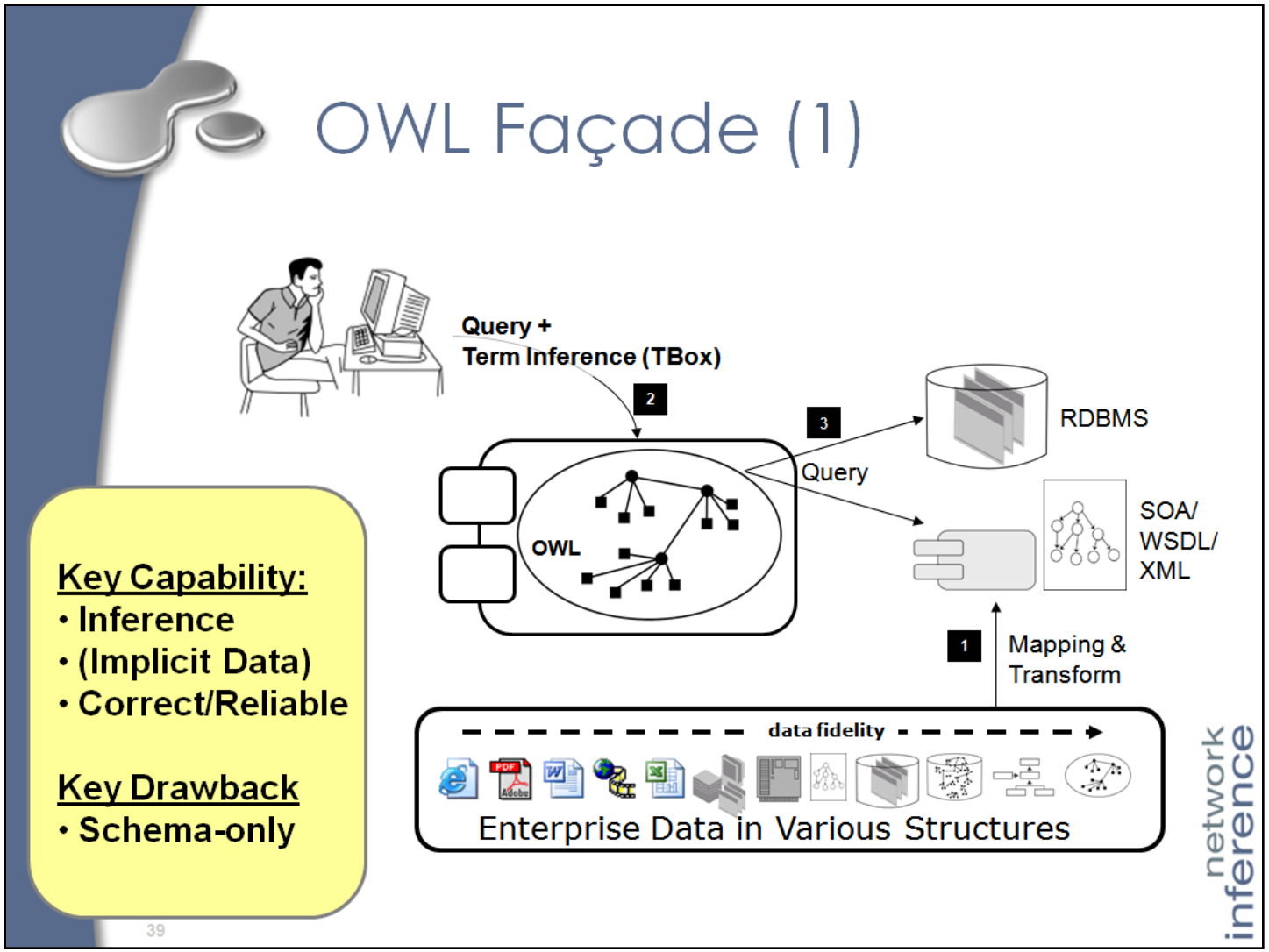
Like the RDF database, the OWL knowledge base pattern is quite mature for a semantic web standard. The knowledge base concept is similar to a database concept, but operates with a far richer set of data relationships, and can thus provide built-in inference during query time to find information easier and allow users to find links in information without having to be deeply intimate with the data models.



NOTES

The OWL knowledge base uses "description logics" (DLs) as the core data formatting approach. DLs provide the foundation upon which a correct and reliable form of inferencing can be done on both schema concept relations and instance data relations. The interface to a knowledge base typically has two core styles: a query interface, and an object access interface. Taken together, a declarative query access layer and an object-oriented layer provide both applications and services the ability to interact with richly formatted data.

OWL Façade (1)



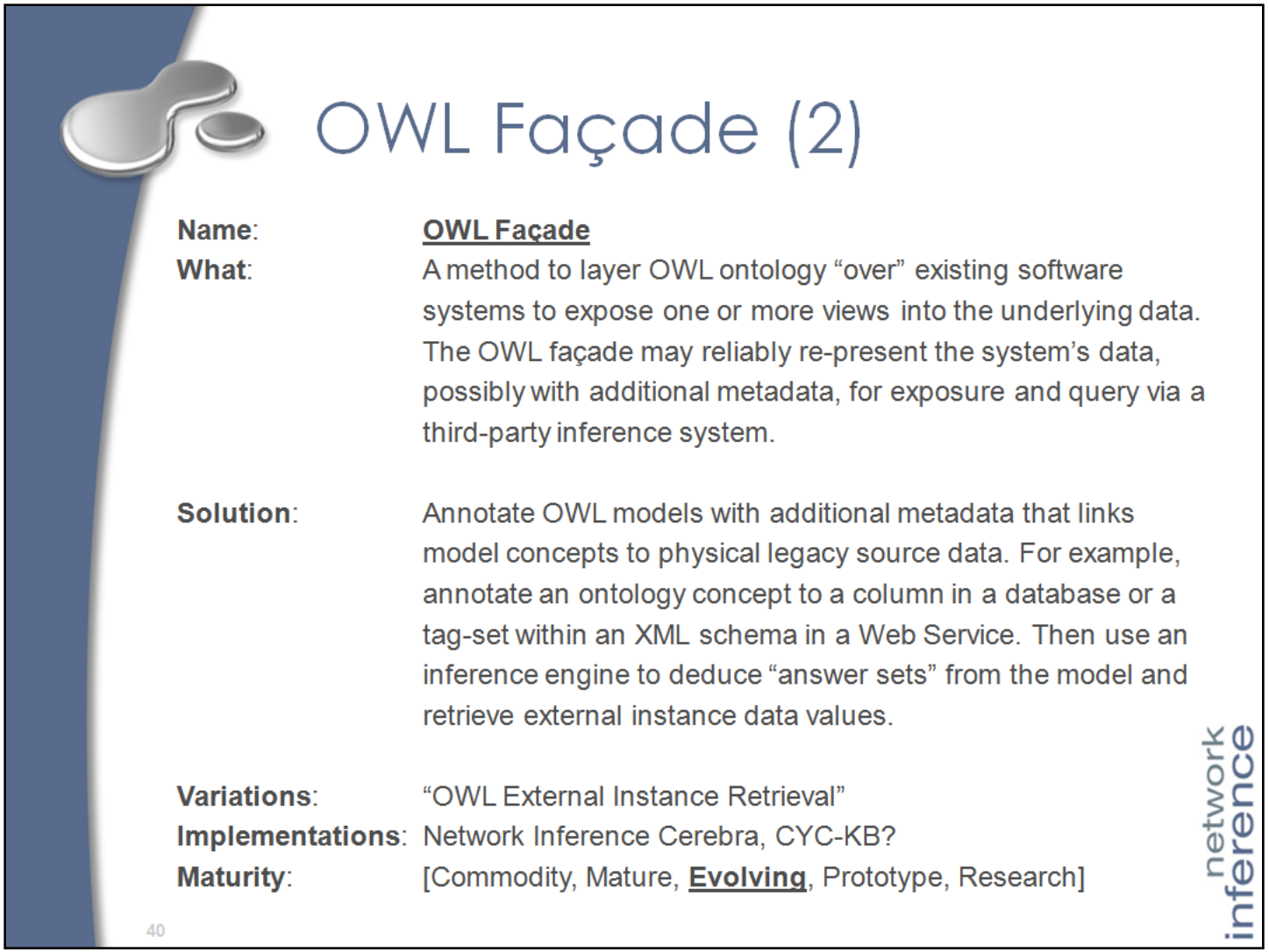
Key Capability:
- Inference
- (Implicit Data)
- Correct/Reliable

Key Drawback
- Schema-only

39

NOTES

The OWL façade is an optimized implementation of the OWL knowledge base that leverages taxonomic, or schema, level inferencing only. This optimization improves overall performance and provides a clean separation of instance and schema level processing so that external instances (in XML or RDB formats) can be easily retrieved by generating query plans from the results of inferencing over the ontology.



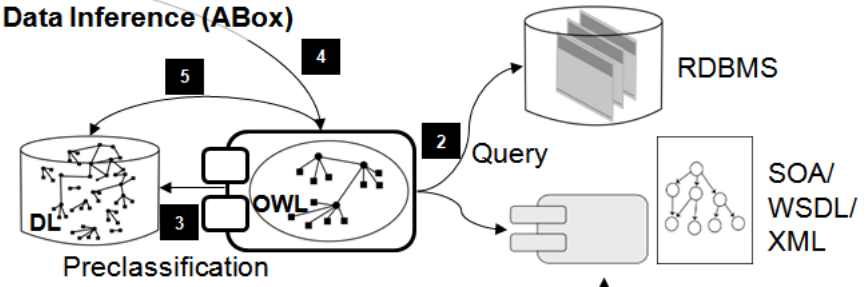
NOTES

Quite mature compared to other Semantic Web software approaches, the OWL façade is supported with COTS tools and can be plugged into an XML based SOA messaging layer to provide mediation services.

OWL Instance Store (1)



Query +
Term Inference (TBox) and
Data Inference (ABox)



Key Capability:
- Data Values
- (Inference)
- Scalability

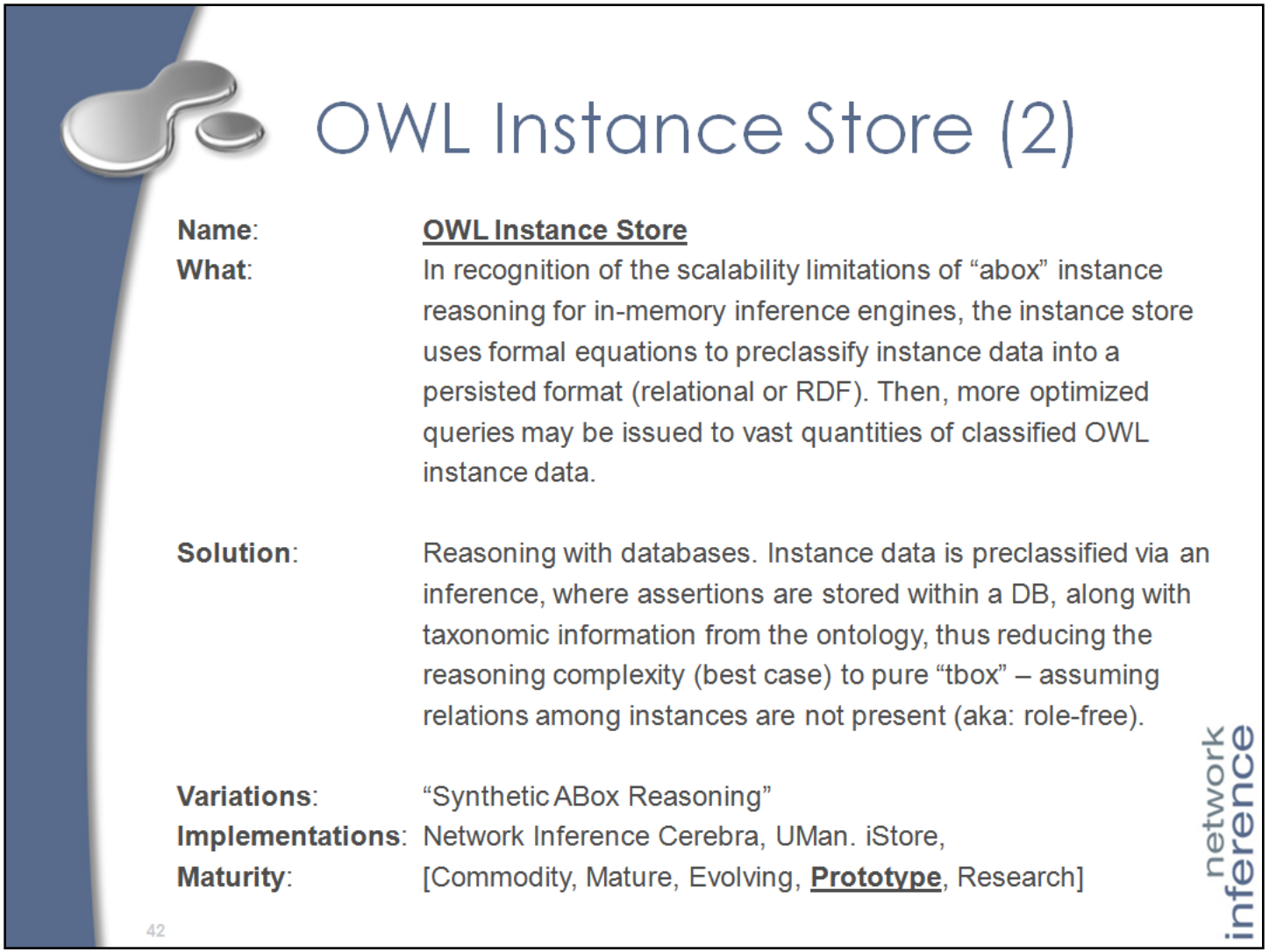
Key Drawback
- DL-subset

network inference

41

NOTES

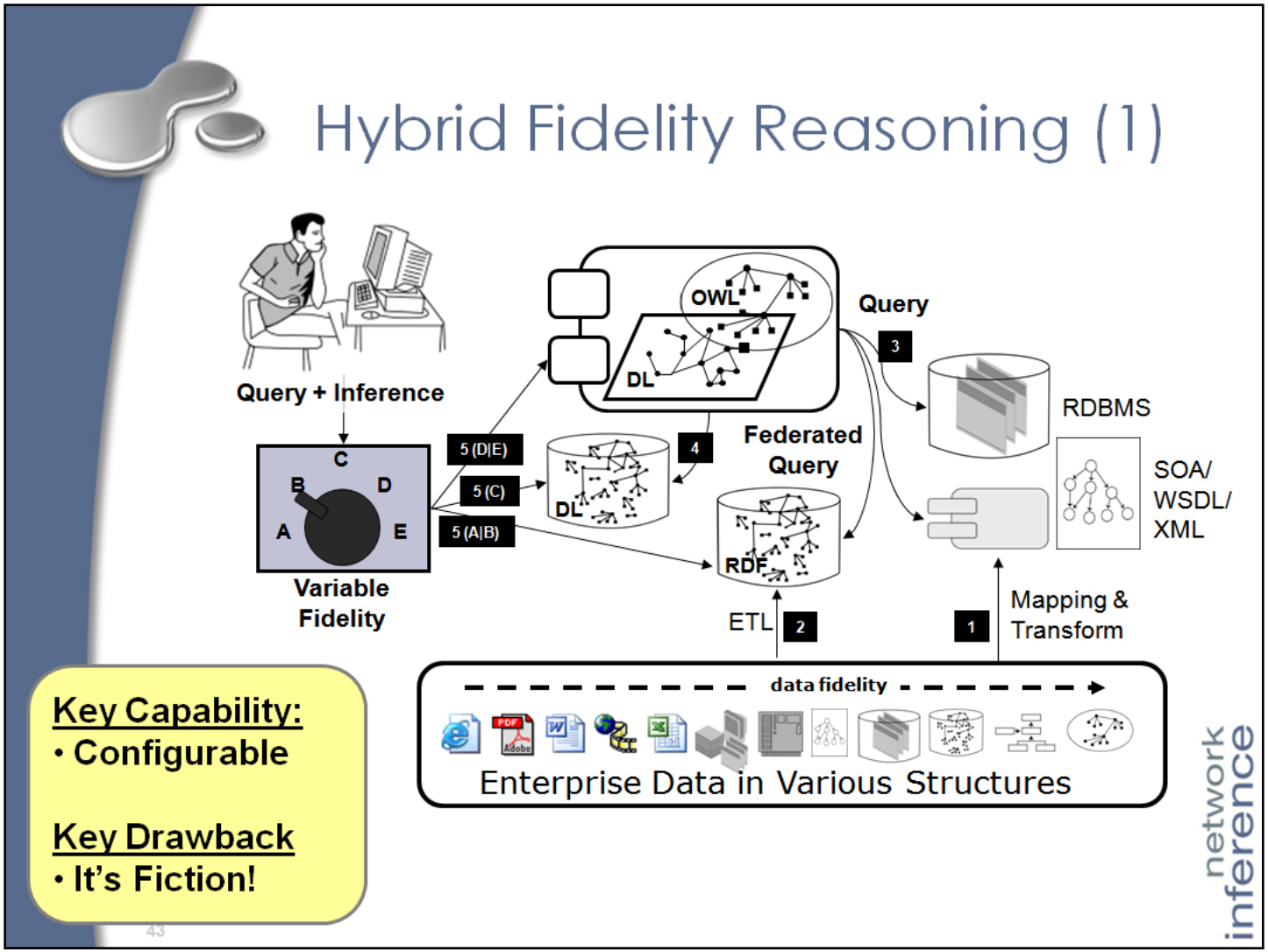
The OWL instance store is a middle ground between the complete OWL knowledge base and the OWL façade. The key advantage with the instance store is that it can operate on large numbers of instances with minimal performance impact, but it sacrifices some of the reasoning power on those instances.



NOTES

In the instance store approach, data is "preclassified" into a persistence store, then retrieved as needed based upon specialized algorithms to draw inferences based upon the preclassified stored relationships.

Hybrid Fidelity Reasoning (1)



NOTES

Still more of a research topic than an off-the-shelf solution, a hybrid reasoning system will make use of different entailment levels – reasoning power – at runtime to optimize given transactions for an appropriate balance between speed and reasoning power.

Hybrid Fidelity Reasoning (2)

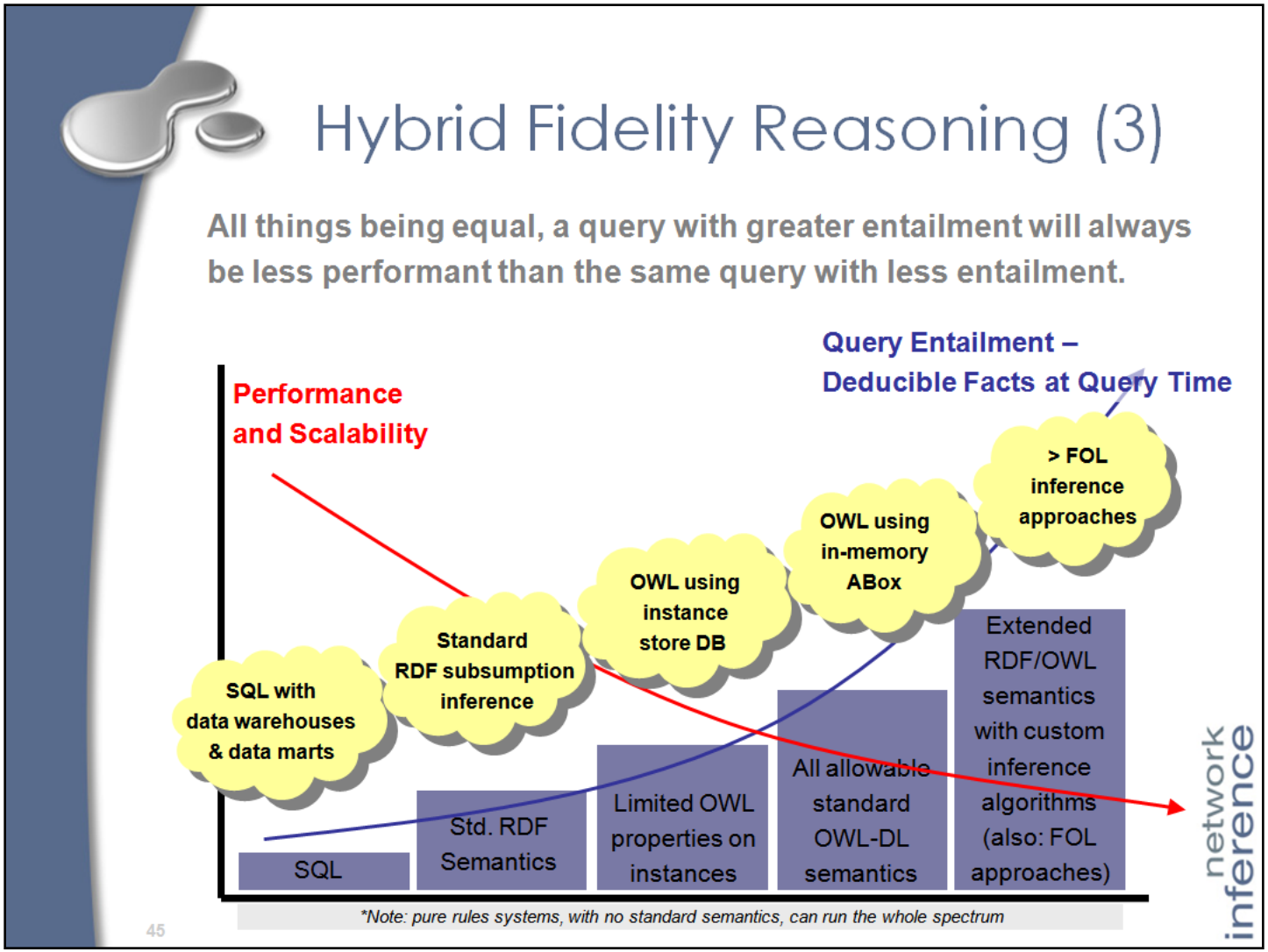
Name:	**Hybrid Fidelity Reasoning**
What:	A hybrid approach to reasoning on semantic web data; in recognition that different queries will necessitate different levels of entailment, or reasoning. Generally speaking, the capability should be in place to sacrifice performance for power, or deductive power for performance within a federated knowledge system.
Solution:	The collection of previous patterns, in a shared architecture, with a common query façade layered in front. The query façade should allow developers the capability to "select" an inference power/performance or execute an "anytime algorithm" that gives answers progressively, as more inference completes.
Variations:	Anytime Reasoning, (related: Approximate Entailment)
Implementations:	Various research, university testbeds
Maturity:	[Commodity, Mature, Evolving, Prototype, **Research**]

network inference

44

NOTES

The envisioned approach can take two forms, (1) the requesting application or service can specify a reasoning level at query time to suit its needs or, (2) special algorithms may be employed to self-optimize based on available knowledge about data complexity, results size, and query entailment requirements.



NOTES

The overall pattern to be aware of with reasoning systems is that the greater power of entailment (deduction/induction of pertinent relationships among data) the less optimal the performance. In some ways the industry is still in debate about the most appropriate levels of power that is optimal, but ongoing facts about performance requirements are generating balance and realism in the solutions which are being deployed today.

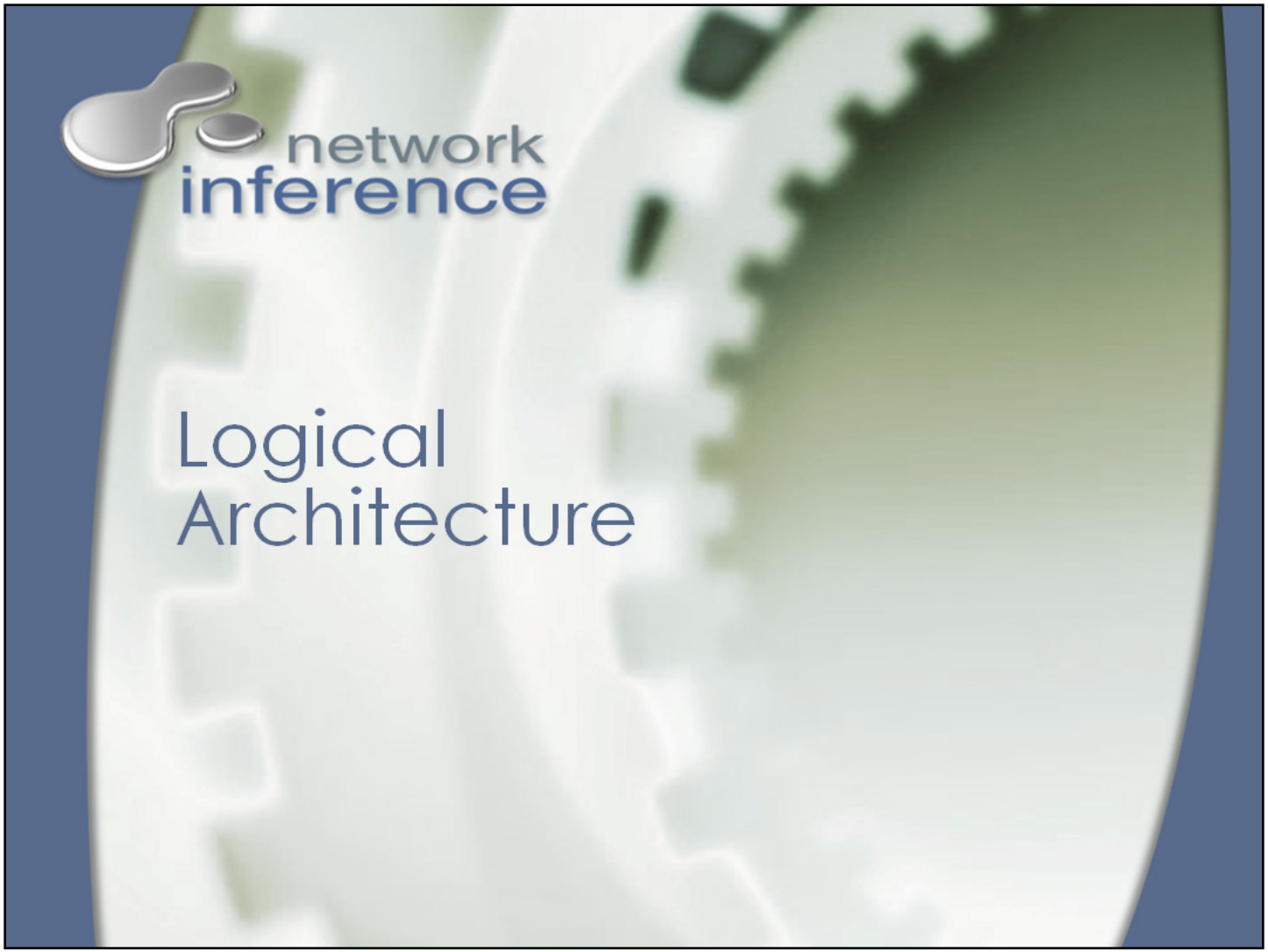
Pattern Summary

	RDF Database	RDF Façade	OWL Façade	OWL Instance Store	OWL Knowledge Base	Hybrid Fidelity
Core Data Format	RDF Graphs	RDF Graphs	OWL Concepts	OWL Concepts & DB Data	OWL Concepts & Instances	RDF & OWL (all)
Inference Capability	Simple Subsumption/ custom RDF	Simple Subsumption/ custom RDF	OWL "TBox"	OWL "TBox" & some "ABox"	OWL "TBox & "ABox"	RDF & OWL
Performance	High – (billions & fast)	High – DB quality + RDF inference	High – DB quality + OWL inference	Medium – OWL "TBox" for concepts and data (100's millions)	Low – OWL in-memory "TBox" and "Abox" (millions)	Variable
Schema Federation	RDF/S	RDF/S	OWL-DL	OWL-DL	OWL-DL	RDF/S & OWL-DL
Data Federation	RDF Triples	RDF Triples via RDF/S mappings	RDF Triples via OWL-DL mappings	RDF Triples via OWL-DL mappings	RDF Triples via OWL-DL mappings	RDF Triples via OWL-DL or RDF/S
Maturity	High (evolving)	Medium (prototype)	High (evolving)	Medium (prototype)	High (evolving)	Low (research)

network inference

NOTES

There are many ways to understand the various semantic web architectures evolving in the industry today. This chart attempts to highlight some of the key differences, implications, and maturity among the various patterns. There is no single best approach, but for your specific needs, there are very likely better and worse choices to consider.

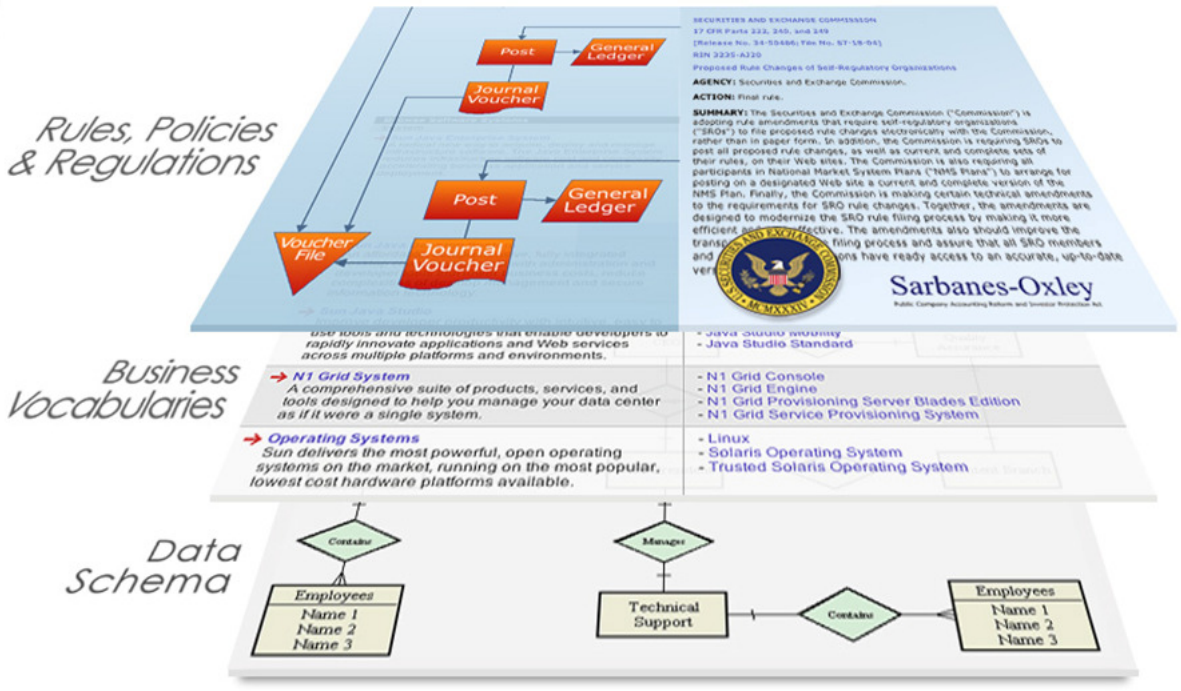


NOTES

Thus far we have examined different physical architectures, but without a solid logical information architecture no business solution would come to fruition.

Logical Architecture (1)

Decouple critical business IP from the software that executes the logic

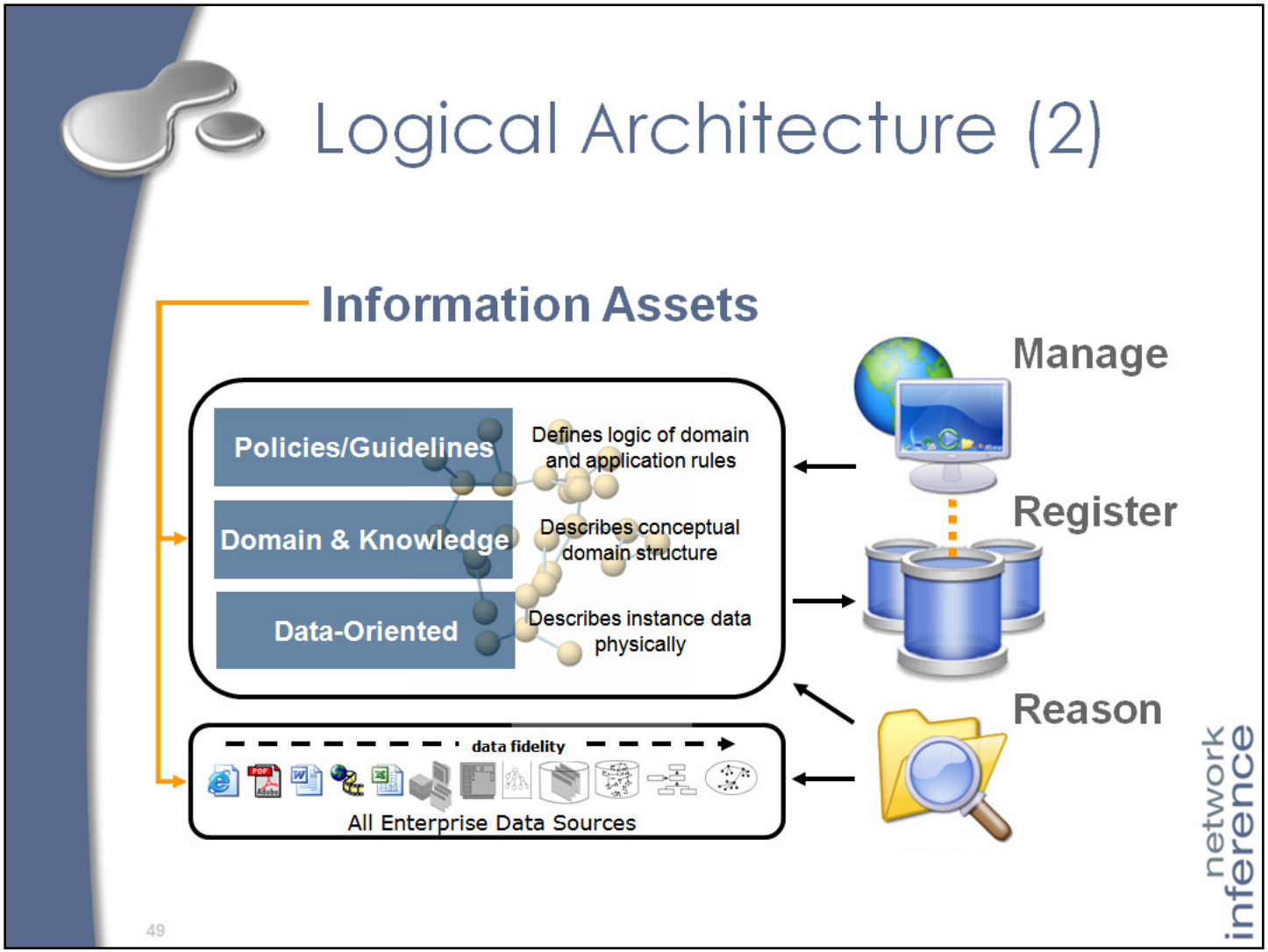


network inference

48

NOTES

Ontologies are not magic. They need to be deployed in sound engineering discipline with distinct separation of concerns. The core pattern emerging for a logical architecture is to separate the nuts-and-bolts data schema (for describing legacy sources) from the business vocabulary (for describing the domain independent of legacy IT) from the rules and policies that constrain the domain (for capturing business rules). Taken together, these layers enable a data architect to create a master set of enterprise information – and link it to "real" IT data sources.

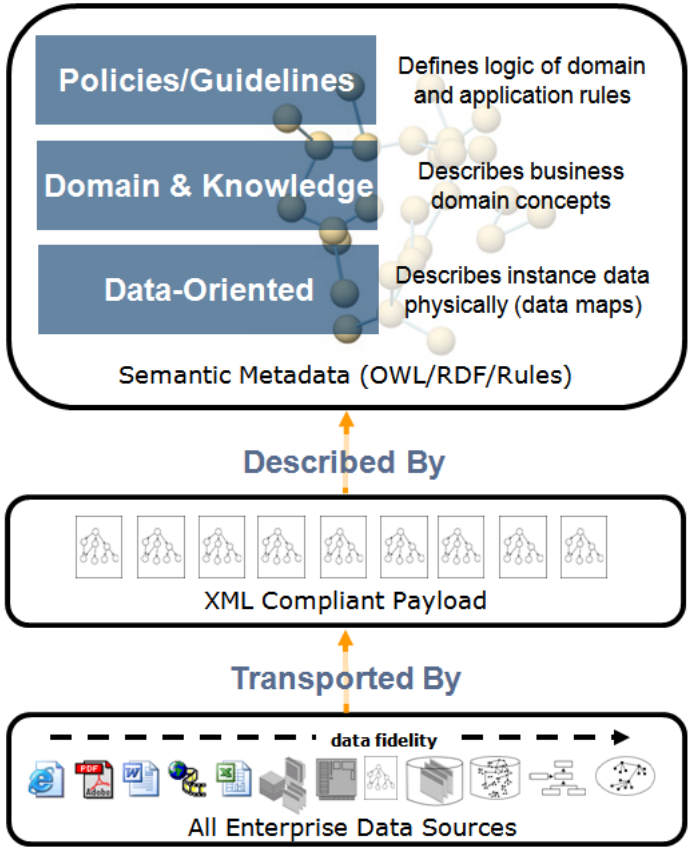


NOTES

Once these three layers of ontology concerns are deployed, they can be reasoned with, managed, and queried to fulfill different business objectives.

Logical Architecture (3)

- SOA
  - In the context of an SOA, XML should provide the basis for transport – not semantics.
- XML
  - XML may be mapped to OWL for runtime conversion.



network inference

50

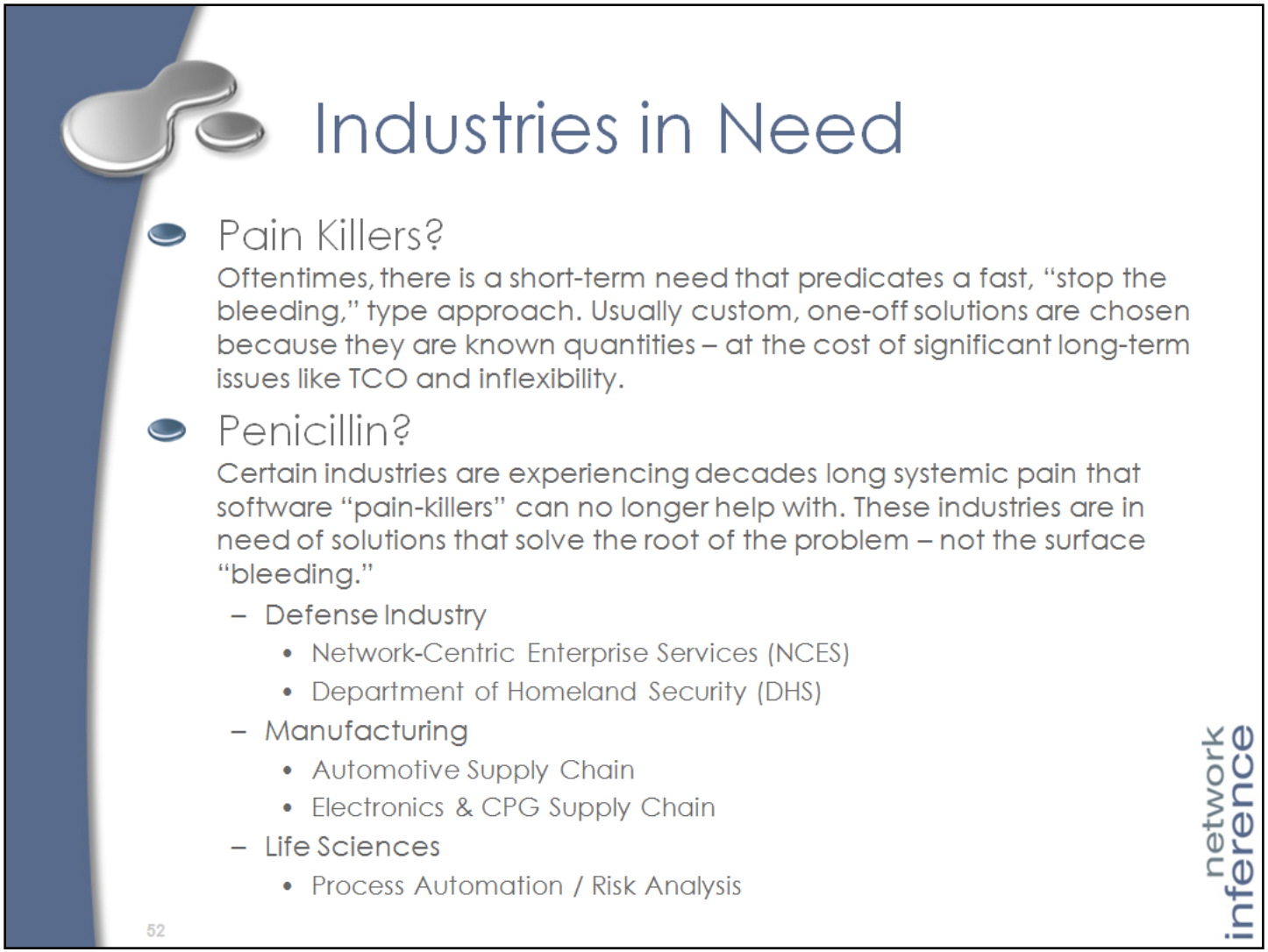
NOTES

The previous two slides look at the logical architecture independent of implementation, this slide attempts to show how concerns may be separated within an SOA. XML should be the message transport structure, while OWL/RDF are the metadata which encapsulates the semantics of given XML messages – thus enabling a global view into heterogeneous XML data.



NOTES

In this last section, we will step back and look at the overall trends and provide some words of caution about adopting these technologies.



NOTES

Not all problems require semantic technology solutions. For some people, there are close-proximity pain-points that just need quick-fix solutions. In these cases, the trade off may be made between long-term viability and short term gains.

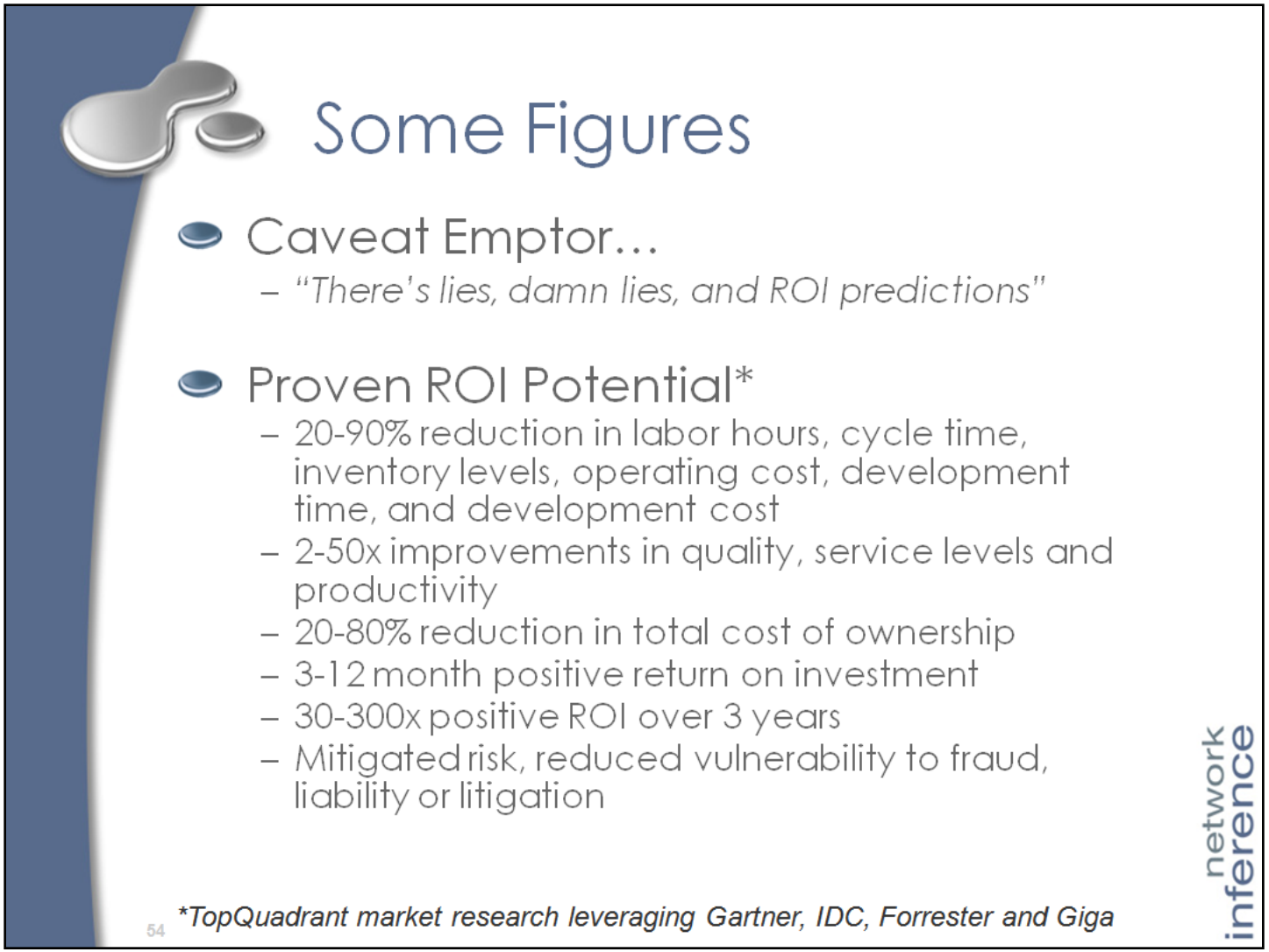
ROI Value Drivers

- New Information Capabilities
  - **What**: Doing something that you haven't done before to provide the business new value.
    Such As:
    - Linking annual risk strategies to live data inspection
    - Assembling SOA service integrations faster
    - Testing realtime "what-if" financial scenarios
    - Securing XML SOA payload with no code
- Total Cost of Ownership (reduction)
  - **What**: Doing what you are already doing, but doing it more efficiently at less operational cost.
  - **Such As**:
    - More productive integration maintenance staff
    - More productive naming/metadata staff

network inference

NOTES

Generally speaking, the value drivers for this technology base come in two forms (1) new capabilities and (2) TCO reduction. Today, as the technology is still early, many are finding value in developing new information capabilities. Others, committed to reducing long term maintenance problems, are driving programs intended to reduce long term TCO.



NOTES

Analysts are beginning to take note of the importance and value of these technologies. Not all the data available is 100% reliable, but it is clear that there is tremendous potential here.

Uptake Momentum

- Commercial
  - Audi, General Motors, Cisco Systems, Fed-Ex, France Telecom, Nokia, Fujitsu, Hitachi, NEC, Unisys, Pfizer, General Electric…
- Federal
  - Federal CIO, SICoP, Navy, Army, DLA, EPA, NCI, DHS, CBP, NGA, MoD, NATO + other foreign federal
- Vendors/Systems Integrators
  - Oracle, IBM, Microsoft, Adobe, Hewlett Packard, WebMethods, CGEY, BearingPoint, SAIC, Lockheed Martin, SRI, MITRE, Booz Allen Hamilton
- Standards Bodies
  - W3C, OMG (MDA), OASIS (DCML, ebXML), ISO

network inference

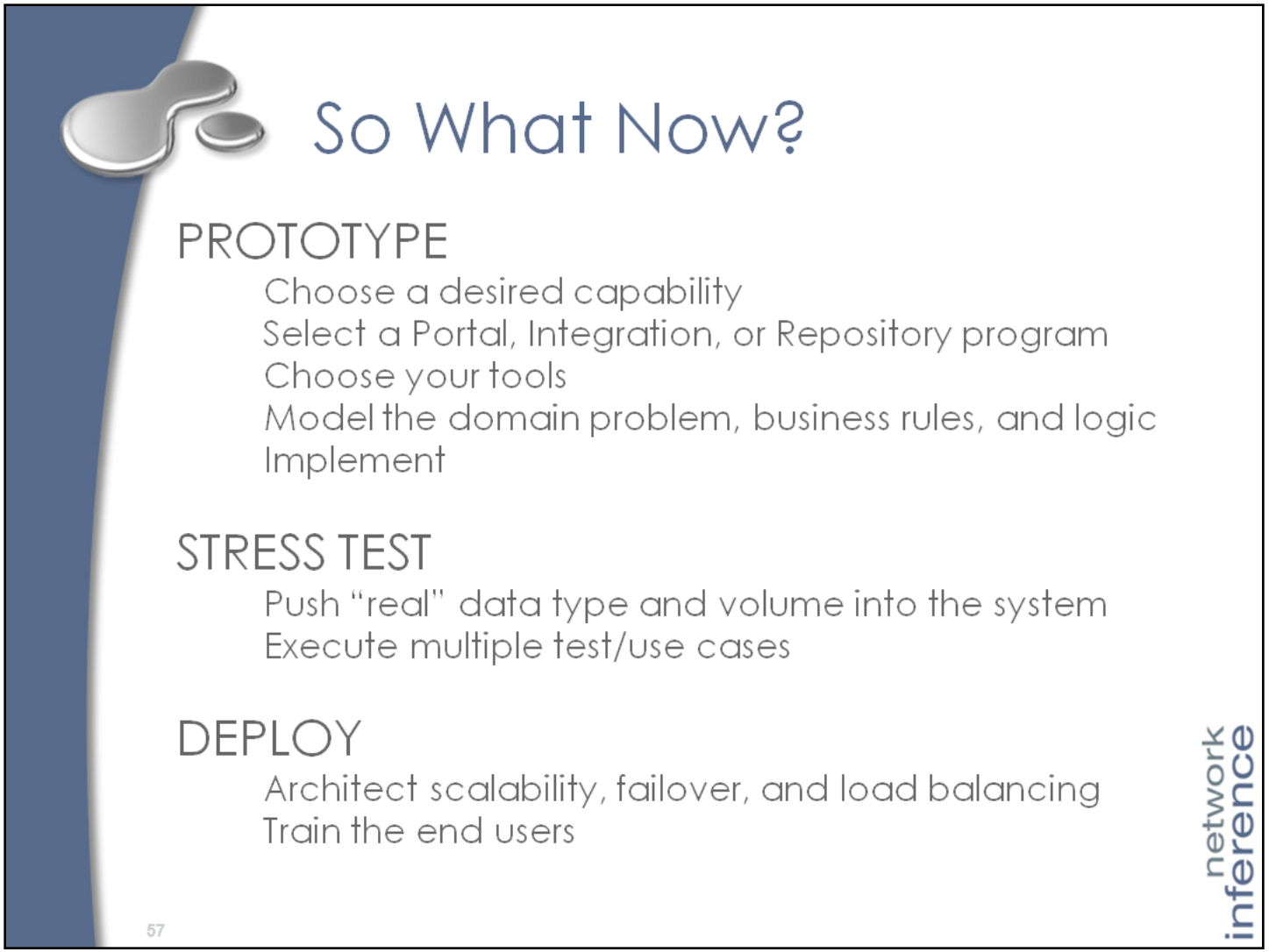
NOTES

Further evidence of the value of these technologies can be found in organizations who are working with them. This partial list of adopters has been culled from public resources and demonstrates that many thought leaders are seeing the potential of RDF, OWL and semantic technologies.



NOTES

It is important to understand that new technology does not come without cost. It will take training, tools and in-the-trenches experiences to fully leverage the benefits of semantic technology.

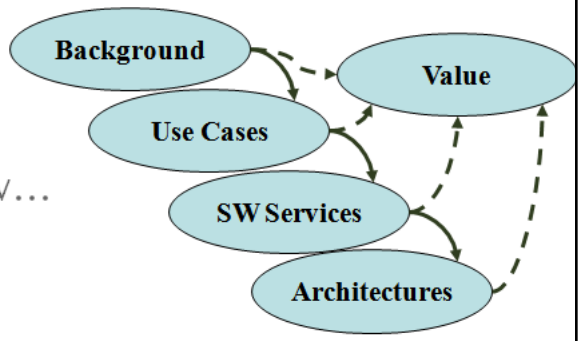


NOTES

Like any new capability or emerging technology, a key prototype program should be identified to test the objectives of the technology. Depending on the desired capabilities – to improve a portal environment, create a change resilient integration, or a more flexible metadata service – choose a focal point and test the technology. Typical experiences with COTS supported semantic technologies are that a 6-9 week prototype can prove out important capabilities and provide tangible business value.

Postscript

- Presentation Goals
  - Straight Talk; **No Hype**!
  - **Education** – what, why, how…
  - Explain **Value-drivers** & ROI



- Presentation Takeaways
  - If you're serious about data, you need to be serious about '**semantic technology**'
  - The core enabling technologies and services are available to you now – **RDF, OWL & Core Services**
  - Semantic Web technology is not perfect, it may never be, but it is showing real value within the enterprise – an **Enterprise Semantic Web**

network inference

58

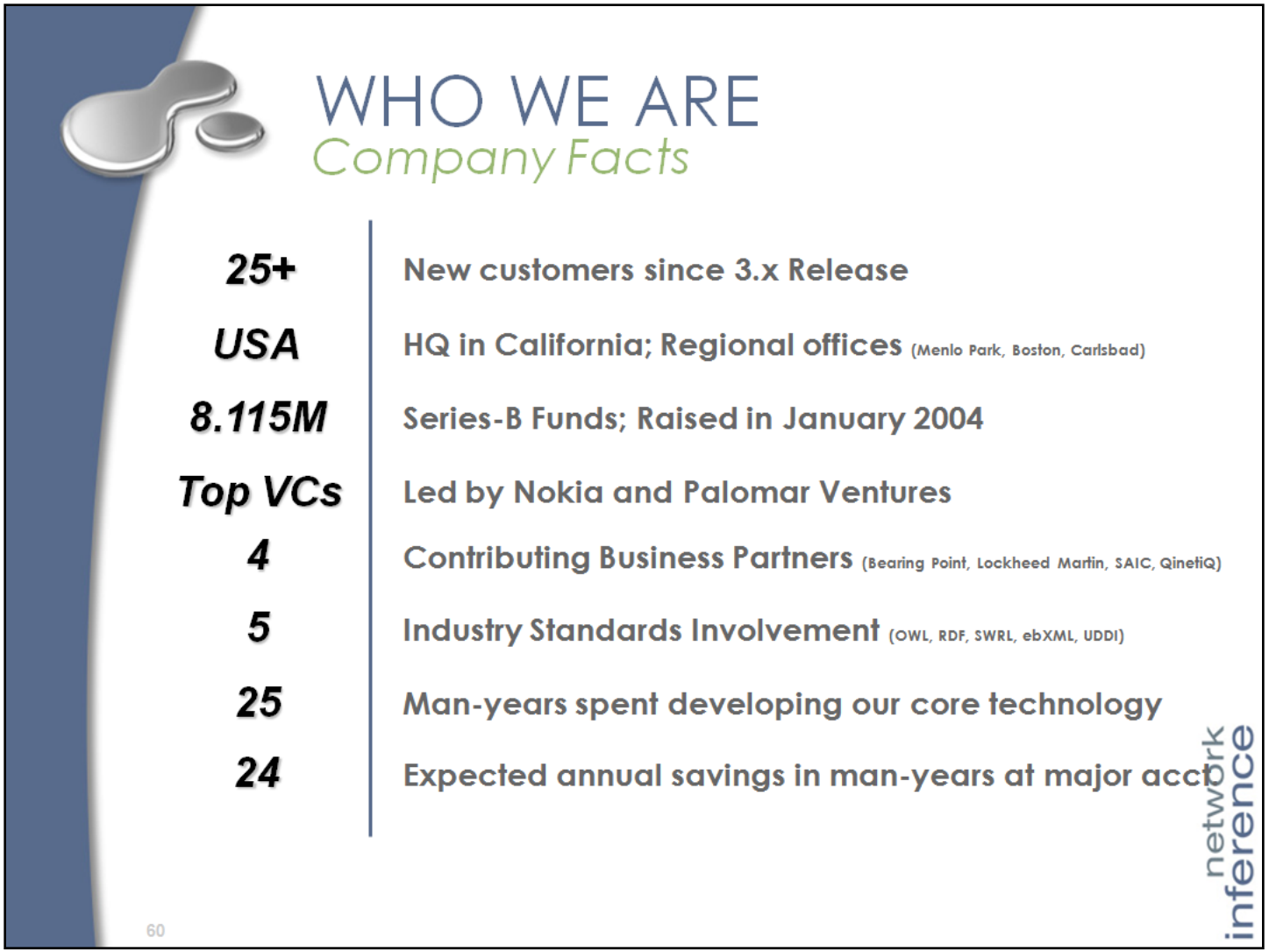
NOTES

In conclusion, the time is now to start examining semantic technologies to solve your pressing IT information problems in more cost-effective and long-term manner.



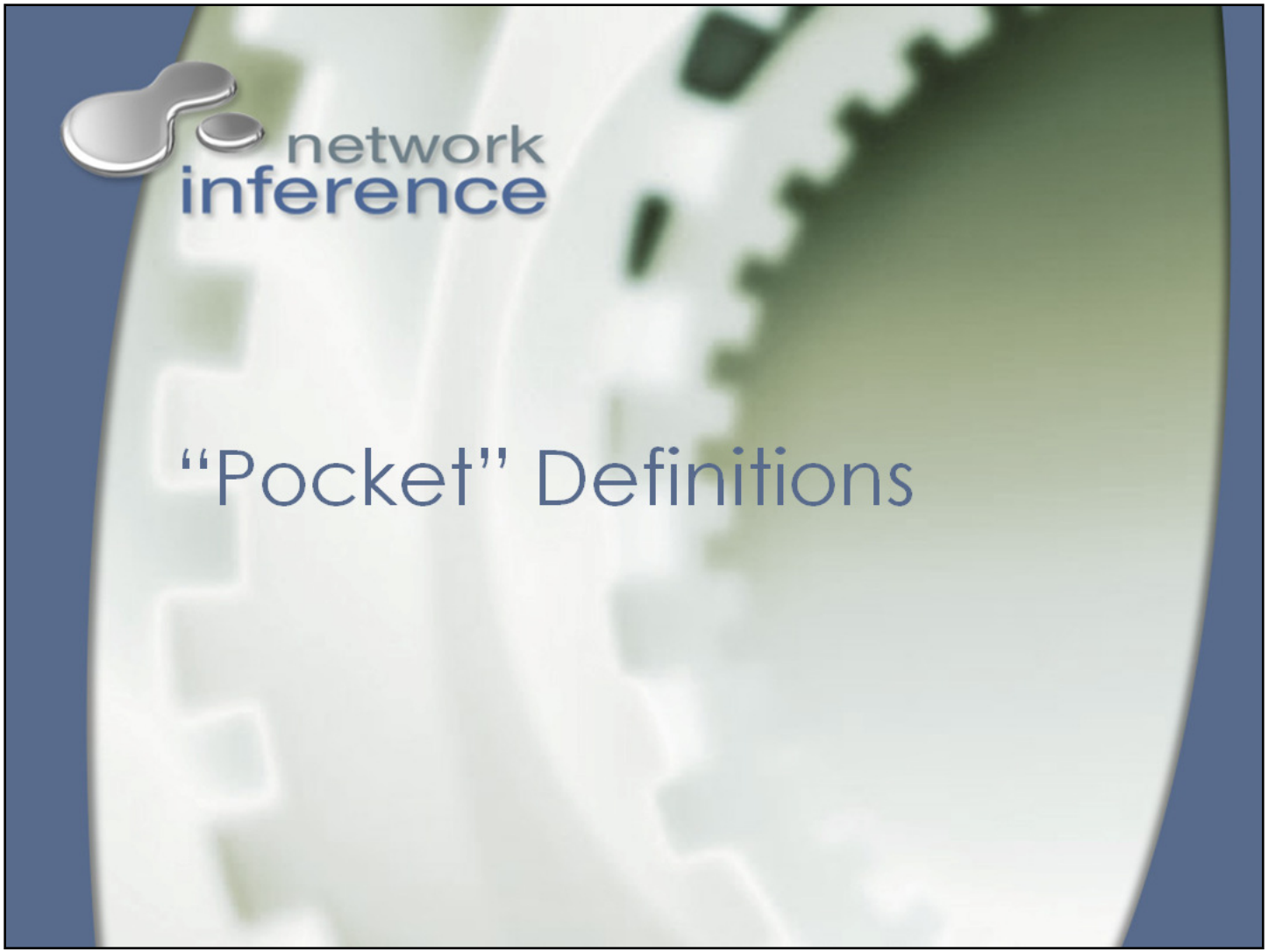
NOTES

Please feel free to contact me with any follow-up questions or comments regarding this presentation.



NOTES

Network Inference is the market-leading provider of innovative, standards-based, and semantics-aware enterprise software.



NOTES

Inference?

Logic With Less Code

Deduce **implicit facts** on explicit data at runtime. Get more application knowledge with **less coding.** Keep your business knowledge in **models, not algorithms.** Maybe learn a few new things too…

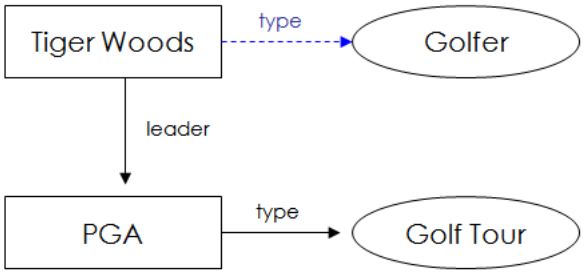
```
<owl:Property rdf:ID="leader">
  <rdfs:subPropertyOf
        rdf:resource="member" />
</owl:Property>

<owl:Class rdf:ID="Golfer">
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty
            rdf:resource="member" />
      <owl:someValuesFrom
            rdf:resource="Golf Tour" />
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
```

The **leader** of a **tour** is also a **member** of it

A **member** of a **golf tour** is a **golfer**

Therefore, the **leader** of a **golf tour** is a **golfer**

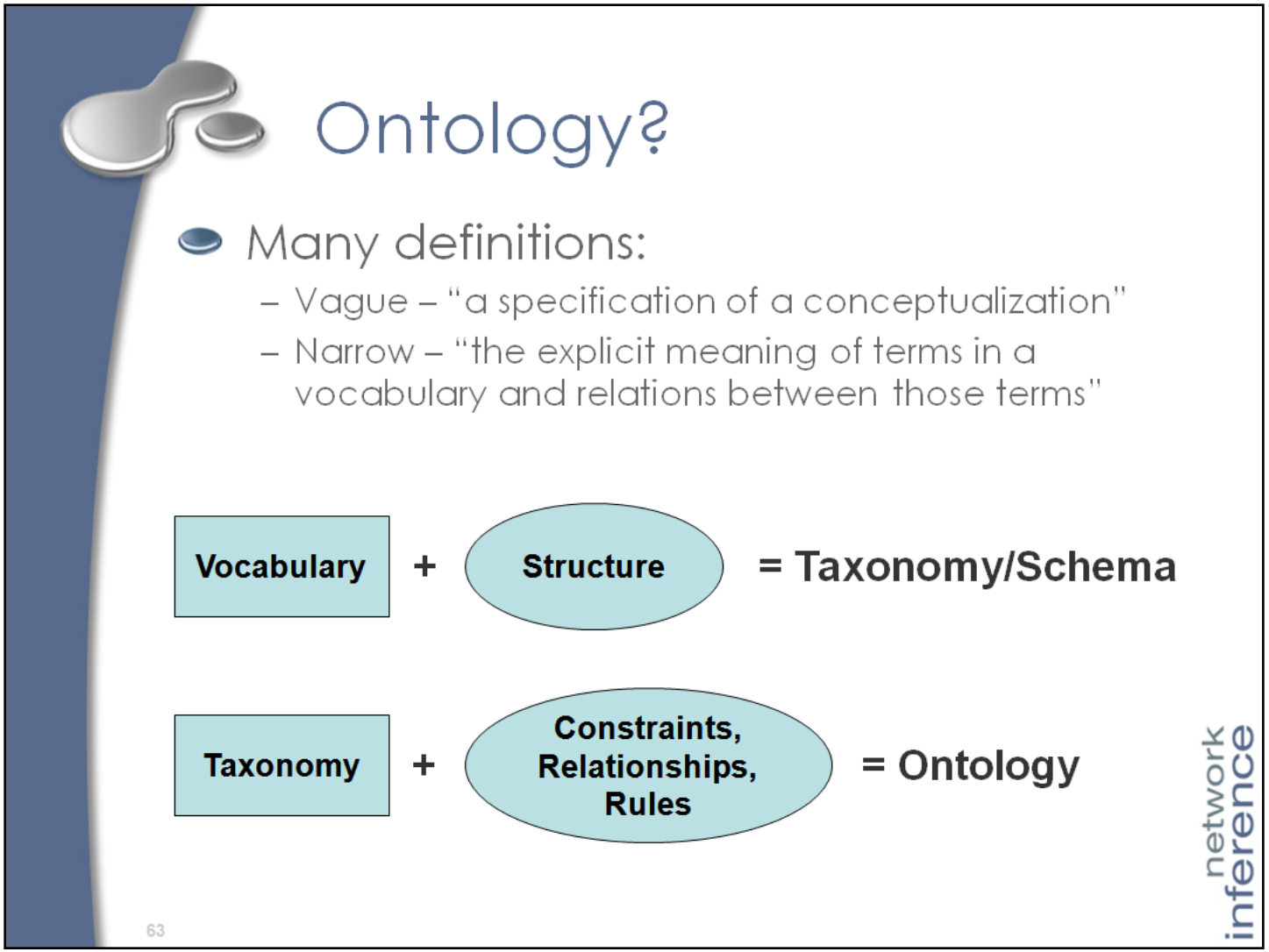


network inference

62

NOTES

Inference is about drawing conclusions based on implied relationships.

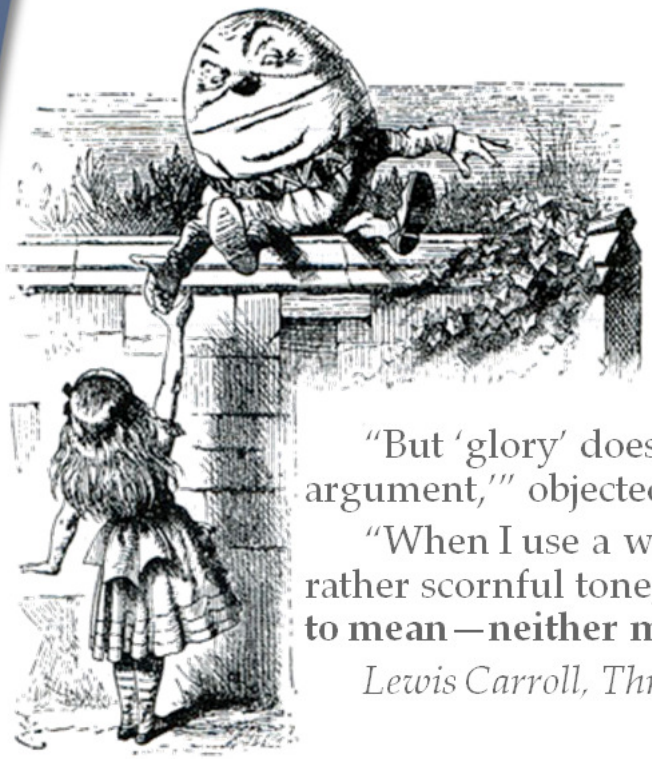


NOTES

Ontology – in the context of this presentation – is just data structure plus constraints, relationships, and rules.

A Little Ironic…

Q: What are semantics?



"I don't know what you mean by 'glory,'" Alice said.

Humpty Dumpty smiled contemptuously. "Of course you don't—till I tell you. I meant 'there's a nice knock-down argument for you!'"

"But 'glory' doesn't mean 'a nice knock-down argument,'" objected Alice.

"When I use a word," Humpty Dumpty said, in a rather scornful tone, "**it means just what I chose it to mean—neither more nor less.**"

Lewis Carroll, Through The Looking-Glass

network inference

64

NOTES

In an industry obsessed with meaning, there is some small irony in the multi-faceted ways that people use the term "semantics." At this conference alone, you will probably hear dozens of definitions.

Semantics? (OWL Formal)

Constructor	DL Syntax	Example
intersectionOf	$C_1 \sqcap \ldots \sqcap C_n$	Human ⊓ Male
unionOf	$C_1 \sqcup \ldots \sqcup C_n$	Doctor ⊔ Lawyer
complementOf	$\neg C$	¬Male
oneOf	$\{x_1 \ldots x_n\}$	{john, mary}
allValuesFrom	$\forall P.C$	∀hasChild.Doctor
someValuesFrom	$\exists P.C$	∃hasChild.Lawyer
maxCardinality	$\leq nP$	≤1hasChild
minCardinali		

CLASS CONSTRUCTORS

AXIOMS

Axiom	DL Syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human ⊑ Animal ⊓ Biped
equivalentClass	$C_1 \equiv C_2$	Man ≡ Human ⊓ Male
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male ⊑ ¬Female
sameIndividualAs	$\{x_1\} \equiv \{x_2\}$	{President_Bush} ≡ {G_W_Bush}
differentFrom	$\{x_1\} \sqsubseteq \neg\{x_2\}$	{john} ⊑ ¬{peter}
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter ⊑ hasChild
equivalentProperty	$P_1 \equiv P_2$	cost ≡ price
inverseOf	$P_1 \equiv P_2^-$	hasChild ≡ hasParent$^-$
transitiveProperty	$P^+ \sqsubseteq P$	ancestor$^+$ ⊑ ancestor
functionalProperty	$\top \sqsubseteq \leqslant 1P$	⊤ ⊑ ⩽1hasMother
inverseFunctionalProperty	$\top \sqsubseteq \leqslant 1P^-$	⊤ ⊑ ⩽1hasSSN$^-$

network inference

65

NOTES

In the context of OWL, the term “semantics” actually has a formal meaning – semantics are the allowable, and formal, relationships you can build between concepts, data, and relationships. This chart shows the foundational class constructors and axioms that comprise OWL semantics.



NOTES

Please feel free to contact me with any follow-up questions or comments regarding this presentation.