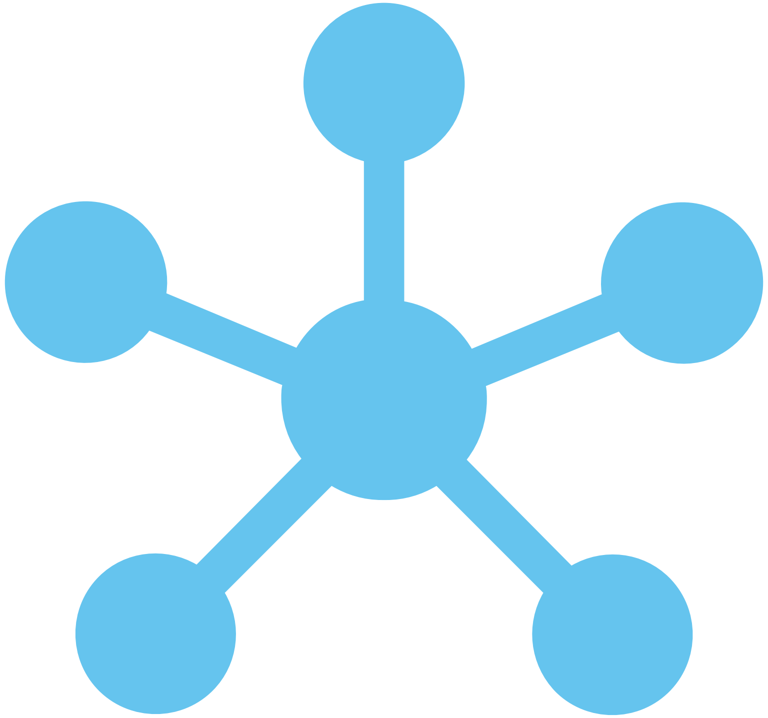
What is Semantic Technology and Why it Owns the Future

Mark Ouska
Ontologist

Agenda

- Semantic Knowledge Graphs
- Globally unique identifiers, data self-assembly
- Design – vs – run-time logic
- No nulls required – Finally!
- Copernican future starts now!



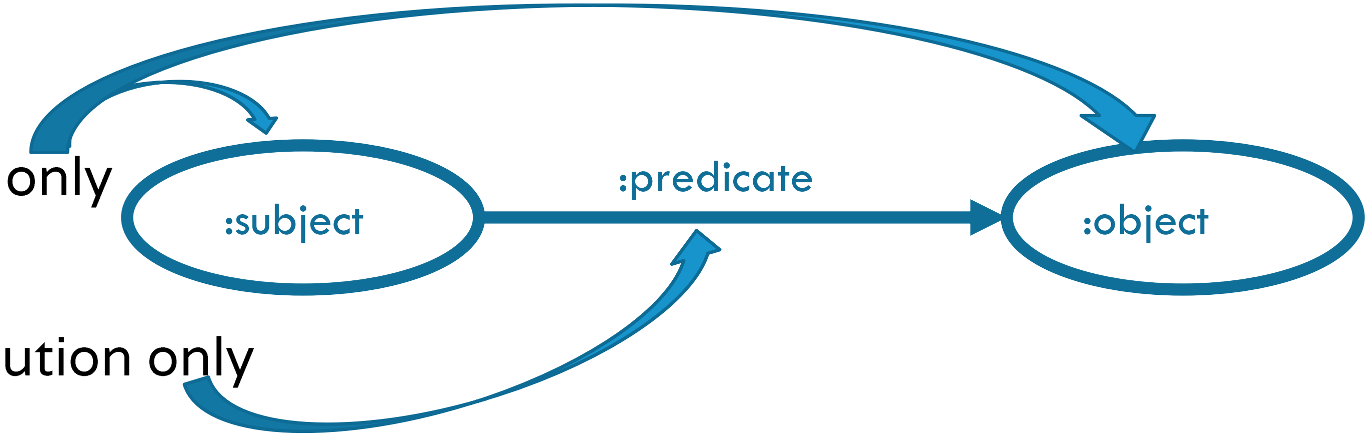
Note on Graph Database Types

Semantic Knowledge Graphs ≠ Property Graphs

Our focus is **semantic Knowledge Graphs.**

Knowledge Graphs have

- Ontology, URIs, node attribution only



Property Graphs

- No ontology, no URIs, edge attribution only

Key point: you can go from a semantic Knowledge Graph to a Property Graph, but not the other way

- Better off starting with a semantic knowledge graph

Knowledge Graphs by Market Cap.

	Company name	Location	Industry	Change in market cap 2009-2018 ($bn)	Market cap 2018 ($bn)
1	Apple	United States	Technology	757	851
2	Amazon.Com	United States	Consumer Services	670	701
3	Alphabet	United States	Technology	609	719
4	Microsoft Corp	United States	Technology	540	703
5	Tencent Holdings	China	Technology	483	496
6	Facebook	United States	Technology	383(1)	464
7	Berkshire Hathaway	United States	Financial	358	492
8	Alibaba	China	Consumer Services	302(1)	470
9	JPMorgan Chase	United States	Financials	275	375
10	Bank of America	United States	Financials	263	307

Known knowledge graph builders (rows 1–6)

Operator of Taobao and AliBot KG builder (row 8)

Known KG builders (rows 9–10)

- IBM and Citi are also working on cross-enterprise knowledge graphs
- Many have cross-enterprise knowledge graph ambitions, but most are focused on a single use case
- S&P does cross-enterprise data management using relational tech

(1) Change in market cap from IPO date
(2) Market cap at IPO date
Source: Bloomberg and PwC analysis

Agenda

- Semantic Knowledge Graphs
- **Globally unique identifiers, data self-assembly**
- Design – vs – run-time logic
- No nulls required – Finally!
- Copernican future starts now!

midwest architecture community collaboration

Global Reach

Open Standards

- Tim Berners-Lee & W3C

Breakthrough

- Search
- Linked Open Data

Uniform Resource Identifiers

- Globally unique identifiers
- Self assembly into meaningful sets

Open Standards

RDF *(Resource Description Framework)*

- Framework for describing resources on the web

OWL *(Web Ontology Language)*

- Description logics for ontology building

SPARQL *(Sparql Protocol And Rdf Query Language)*

- Query language for data interaction

PROV-O *(Provenance)*

- Ontology for provenance in RDF

SKOS *(Thesauri, Taxonomies, etc)*

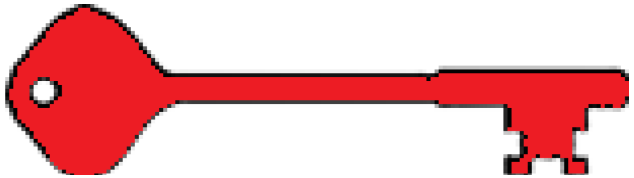
- Simple Knowledge Organizing System

How does Semantic Technology Work?

Triples (RDF)

- Keys to flexible data structures

URIs (Globally unique identifiers)



- Keys to self-assembling data

Description Logics (OWL)

- Remove meaning of information from code and specifications
- Make it understandable and reusable.

Modular Ontologies (Built in OWL)

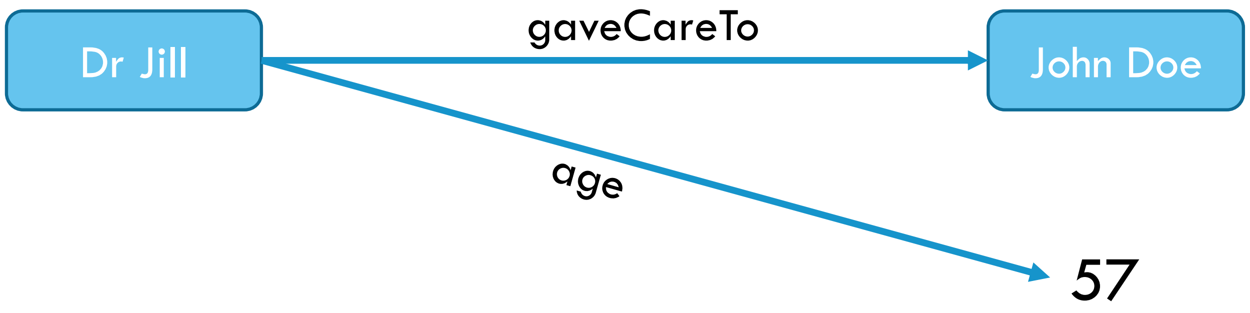
- Different groups can agree, as needed

Triples

Very little pre-existing structure in a semantic database

There are just *triples*

- Facts asserted as small sentences
- With subject—predicate—object
  - *(Node – Edge – Node)*



semantic arts

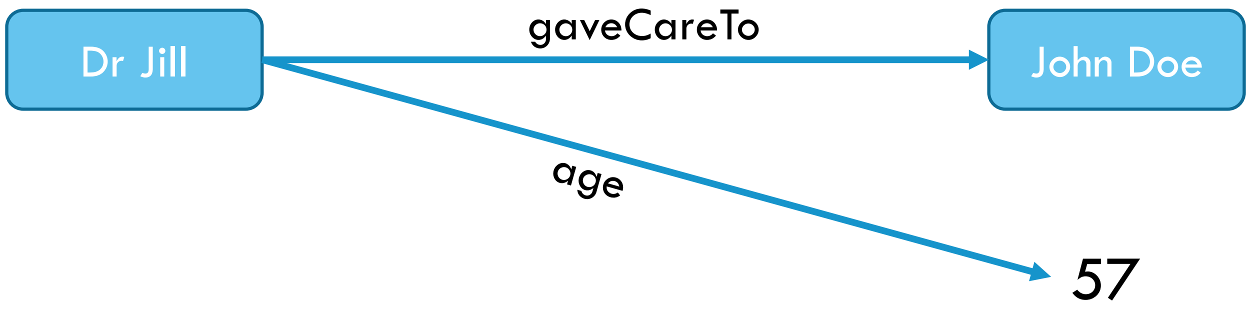
Graphs (but not Graphics)

Sets of triples make a directed graph (aka 'digraph')

A graph has nodes and edges that connect them

Nodes are either the subject or object of the triple.

Edges are the predicates



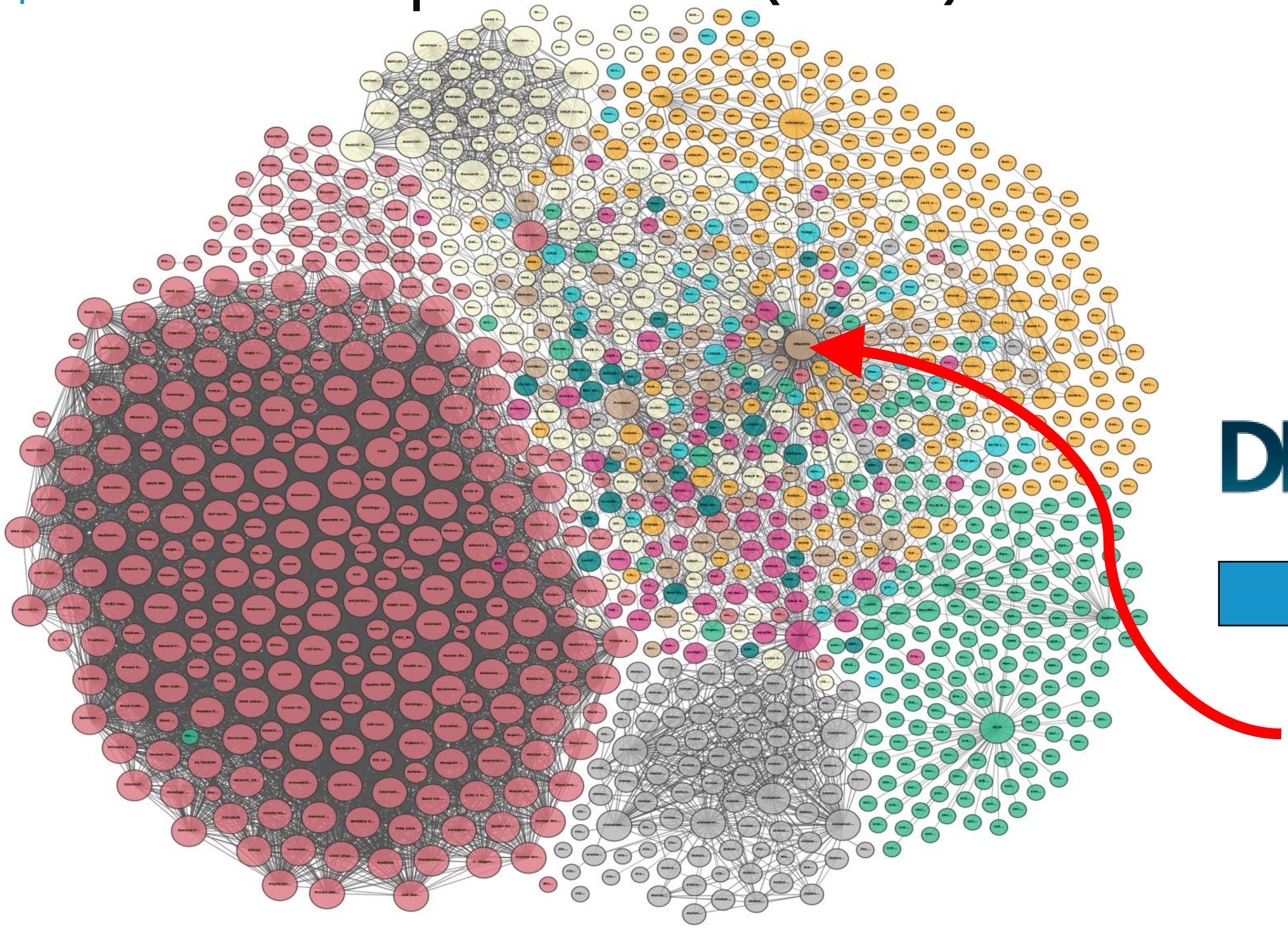
semantic arts

Semantic Tinker Toys

Connection:

Pointing:

Linked Open Data (LOD)



DBpedia

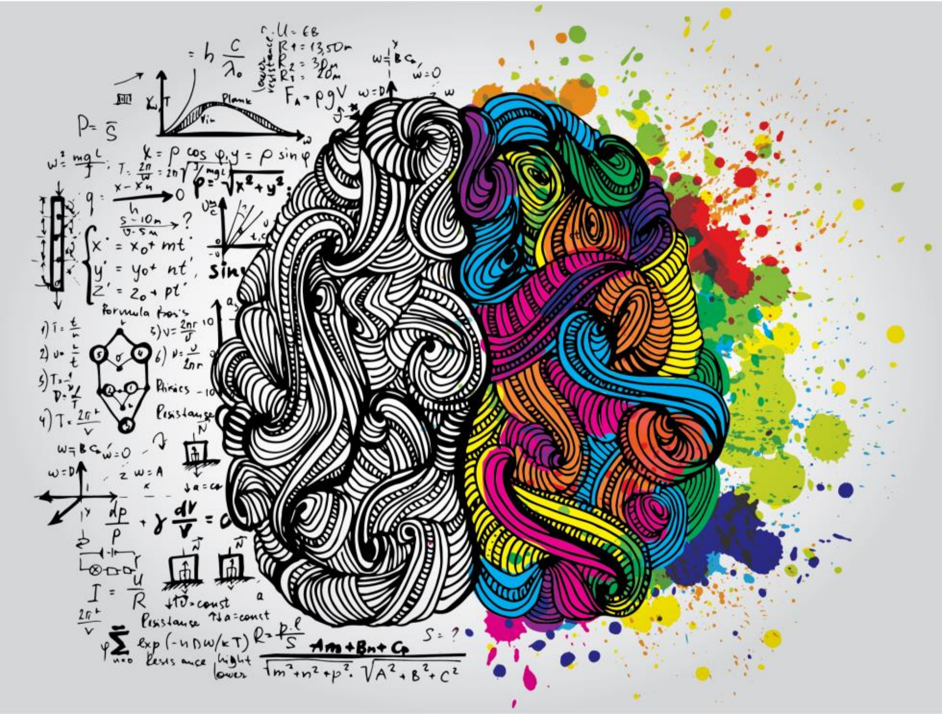
SPARQL

Wikipedia
The Free Encyclopedia

3 billion triples*
596 million entities

http://linkeddatacatalog.dws.informatik.uni-mannheim.de

Semantic Mind Shift



Strategic differences

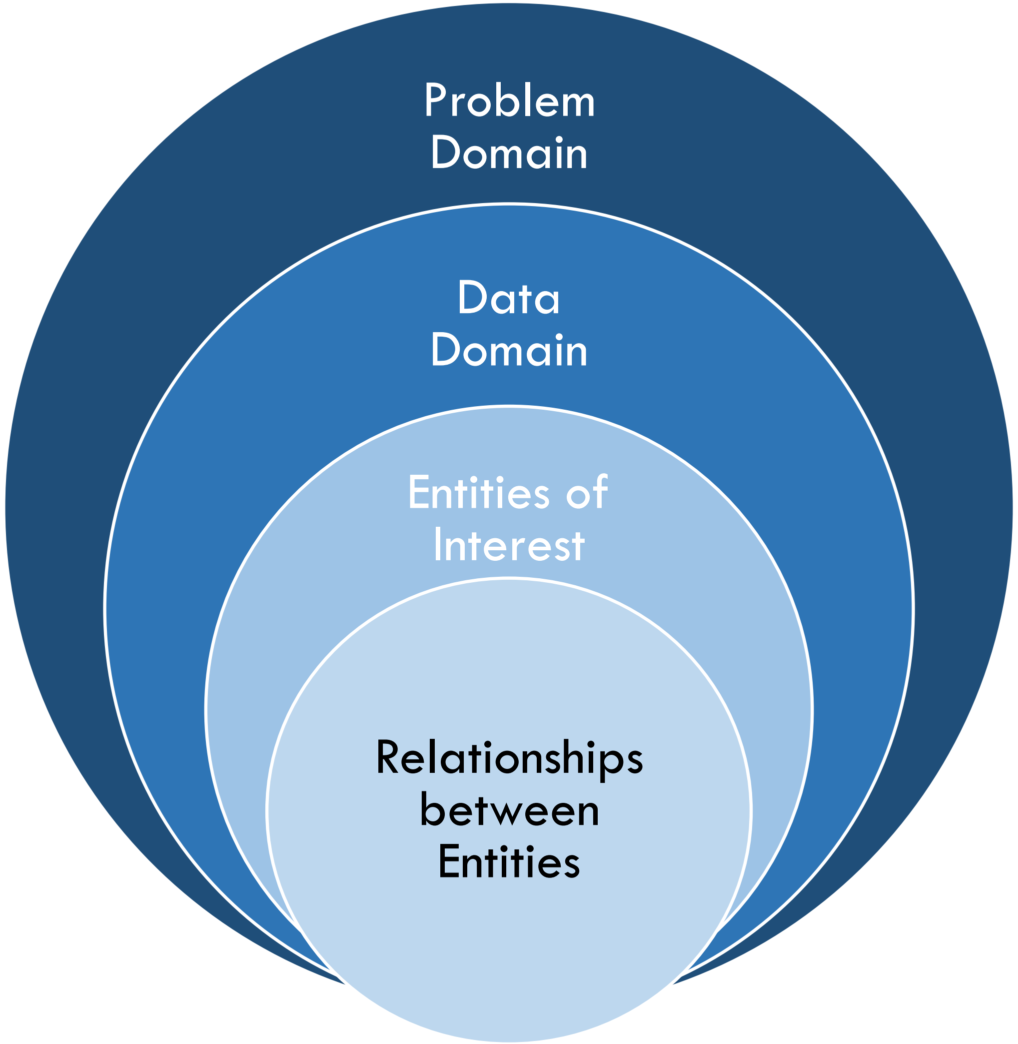
- Semantic meaning drives everything
- Meaning encoded with set theory logic
- Set membership assembled by machine

Tactical differences

- Structure (nearly) irrelevant
  - RDF Triples for everything
- Schema reuse
  - A given schema (ontology) used in disparate solutions
  - Import existing subject area schemas unchanged (modularity)
- Easy data (and schema) portability
  - Build in one vendor database (triple store)
  - Deploy in different vendor triple store unchanged

Everything Doesn't Change

- Problem domain
  - What problem needs solving?
  - What opportunity needs exploiting?
- Data domain
  - What data is relevant to the problem?
  - Customer, Order, Inventory?
- Entities of interest
  - What are the things, the nouns?
- Relationships between entities
  - What verbs connect the nouns?



Semantic Search Breakthrough

Google

- Leaped ahead of the competition by dropping keyword search and leveraging knowledge graph technology.

Apple

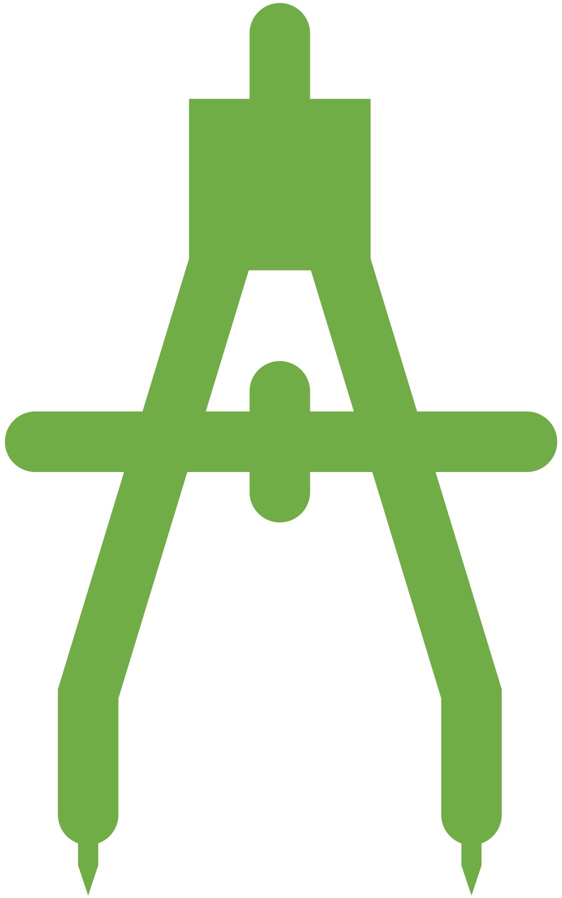
- Added voice recognition; Siri is powered by a knowledge graph.

Morgan Stanley

- Leverages NLP, based on a knowledge graph, with AskResearch AI chatbot, that harvests answers from Wall Street-related terminology.
  - https://gritdaily.com/ai-chatbot-morgan-stanley/

Agenda

- Semantic Knowledge Graphs
- Globally unique identifiers, data self-assembly
- **Design – vs – run-time logic**
- No nulls required – Finally!
- Copernican future starts now!



Runtime Logic

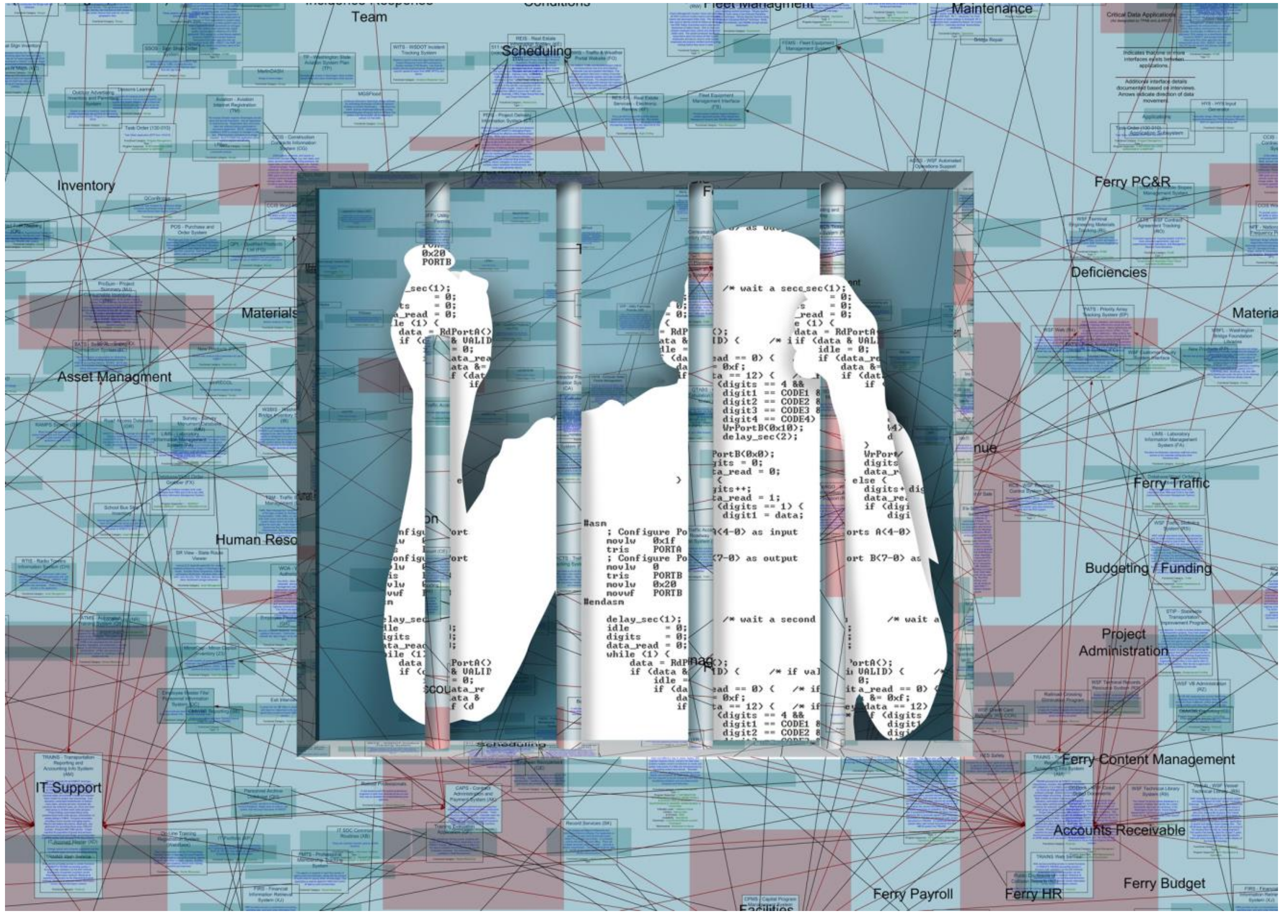
Structure-First Approach

- Grounded in physical thinking
- Things require managed, physical space
- Stuff into small containers (cells)
- Point to point connectors (join tables)
- Custom solutions for each physical thing

Structure-First Tactical Solutions

Proprietary solutions

- Vendor specific
- Lack portability
- Require vendor-specific skills
- Exorbitant cost of change



semantic arts

Structure-First Result (Application Centric)

Data requires physically restricted structures

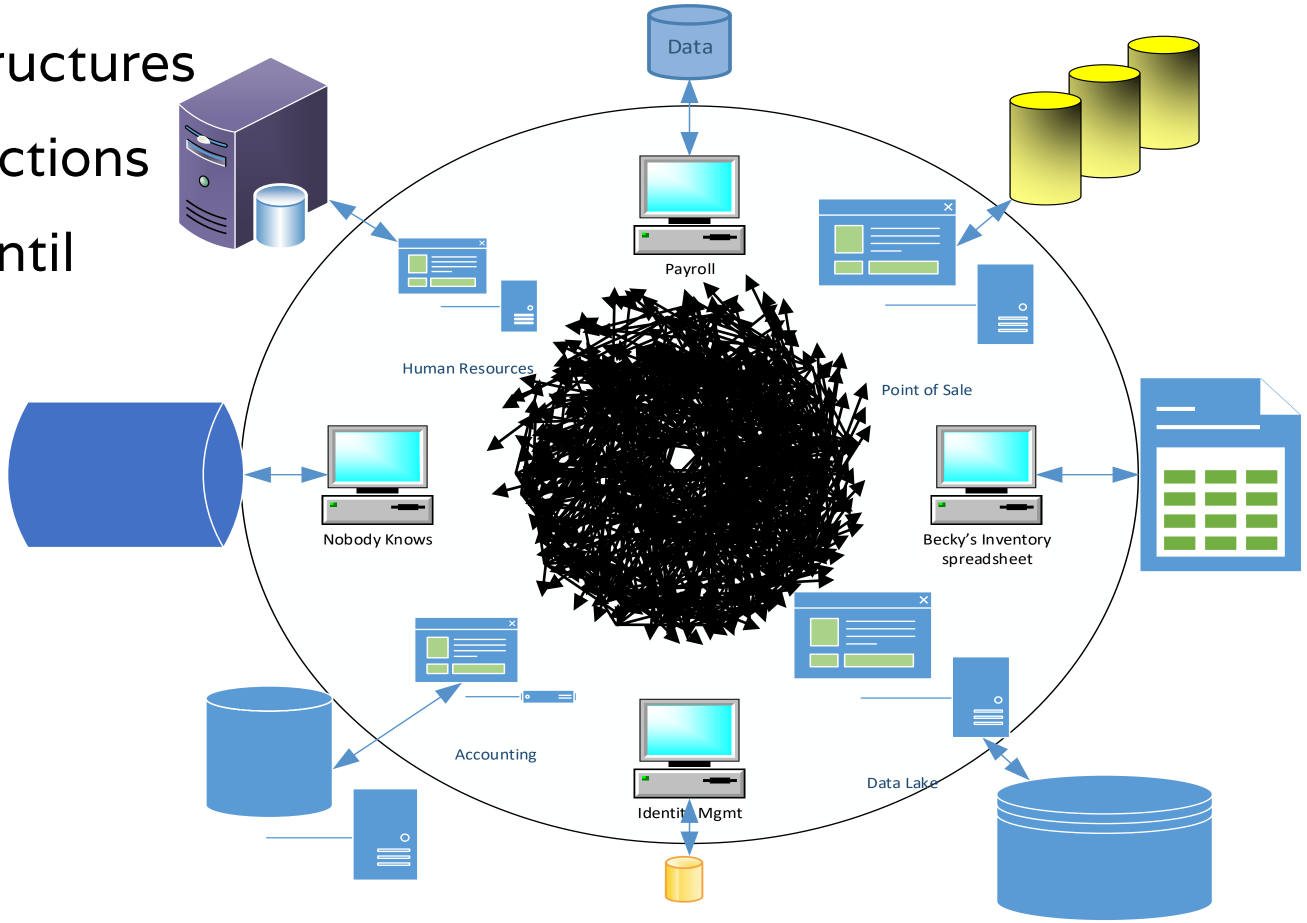
Join structures required for all connections

New information can't be captured until

- Tables have been created or modified
- Join structures created or modified

Data is held hostage by

- Each application
- Each database
- Each API
- …



Tooling Requirements – What Doesn't Change

Required work products

- Information models
  - Data representation
  - Concept development
- Data content storage
  - Container
  - Serialization
- Data management & query
  - Create
  - Read
  - Update
  - Delete

semantic arts

Information Models

Traditional

Mostly proprietary tools

- Major vendor support
  - De facto standard ERD
  - Graphically rich
  - Very mature product offerings
- Complex landscape
  - Plethora of DBMS targets
  - Loosely standardized
  - Virtually every implementation varies

Semantic

Mostly open source tools

- W3C published standards support
  - Standard Web Ontology Language (OWL)
  - Graphically challenged
  - Maturing product offerings
- Simple landscape
  - Common triple structure
  - Very standardized
  - Very little implementation variances

Data Content Storage

Traditional

Structure based

- Wildly varying table structures
  - Unpredictable column count
  - Obfuscated column names
- Locally unique identity management
  - Arbitrary (made up) key values
  - Keys unique only within each table
- More scaffolding than content
  - Complex key structures
  - Materialized joins required

Semantic

Meaning based

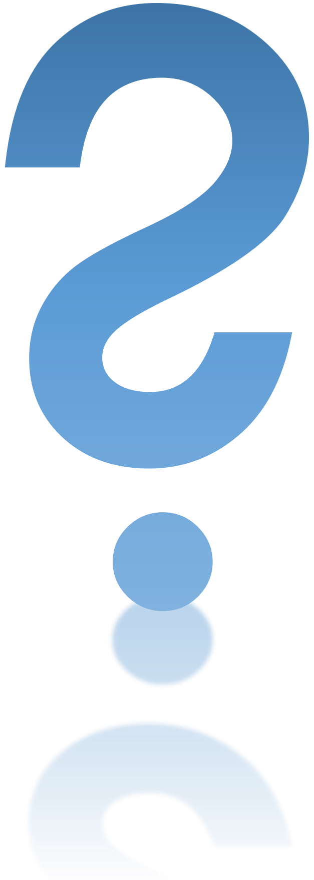
- Single structure: RDF triples
  - Subject, Predicate, Object
  - Each with independent existence
- Globally unique identity management
  - Universal Resource Identifiers (URI)
  - Can be directly resolvable URLs
- Self assembling without scaffolding
  - URI in common – triples snap together
  - No common URI – partitioned out of set

Data Management & Query – Deeper Dive

Getting hands-on with semantic data
- Same output format, rows and columns
- Different query options available

Structure-first
- Must know structure details to begin
- No option to query for schema details

Meaning-first
- No prerequisite repository knowledge needed
- Can query repository for data set definition



Data Query – SPARQL & SQL Overview

SPARQL

SPARQL has one language:

- RDF for creating schema and data

Semantics doesn’t make distinctions between the schema and the data.

Since schema and data are both comprised of RDF triples, any query can interconnect data and metadata and often does.

SQL

SQL has two languages:

- DDL for setting up schema
- DML for querying databases.

It’s hard to query schema in relational, and almost impossible to combine a query against the schema with one against the database.

Data Query – Required Clauses

SPARQL

SELECT

- Unbound variables that will be in the result set and can be filled with anything

WHERE

- Unbound variables are assigned value in each line of this clause
- Membership in the result set is determined by binding values to the variables
- Any number of variables can be introduced and used in this clause
- The result set only binds values declared in the SELECT clause

SQL

SELECT

- Bound variables that are tied to specific **table.column** structures or literals coded into this clause

WHERE

- **Table.column** structures are joined by values in other **table.column** structures or literals coded into this clause

FROM

- **Database.table** structure locations, generally in a single database, are identified as the exclusive data source that will produce the result set

Data Query – Query Type Overview

Type	SPARQL	SQL DML
Select	SELECT query (returns table-like results)	SELECT query
Insert	INSERT / INSERT DATA (inserts triples)	INSERT INTO
Delete	DELETE / DELETE DATA (deletes triples)	DELETE (rows or entire tables)
Update	no update, delete a triple and insert a new one	UPDATE (cells)
Validation	ASK query (true/false response)	N/A
Returns triples	CONSTRUCT query	N/A
Federated	(with SERVICE clause can query many databases or endpoints in one query)	Proprietary, rare, very complex

semantic arts

Knowledge Graph Enabled Solutions?

Sample output from a Knowledge Graph DB

empName	task	location	startDateTime	hours
Mark Ouska	ProjectOne	Minnesota	2019-08-09T18:30:00Z	4.00
Mark Ouska	PTO	Earth	2019-08-02T13:00:00Z	8.00
Mark Ouska	ProjectOne	New York	2019-08-05T13:00:00Z	8.00
Mark Ouska	ProjectOne	New York	2019-08-08T13:00:00Z	8.00
Mark Ouska	ProjectOne	New York	2019-08-07T13:00:00Z	8.00
Mark Ouska	PTO	Earth	2019-07-31T13:00:00Z	8.00
Mark Ouska	PTO	Earth	2019-08-01T13:00:00Z	8.00
Mark Ouska	ProjectTwo	Minnesota	2019-07-04T09:00:00Z	8.00
Mark Ouska	ProjectTwo	Colorado	2019-05-27T13:00:00Z	8.00
Mark Ouska	ProjectOne	California	2019-04-29T13:00:00Z	8.00
Mark Ouska	ProjectOne	Minnesota	2019-03-14T13:00:00Z	8.00

Sample output from an RDBMS

empName	task	location	startDateTime	hours
Mark Ouska	ProjectOne	Minnesota	2019-08-09T18:30:00Z	4.00
Mark Ouska	PTO	Earth	2019-08-02T13:00:00Z	8.00
Mark Ouska	ProjectOne	New York	2019-08-05T13:00:00Z	8.00
Mark Ouska	ProjectOne	New York	2019-08-08T13:00:00Z	8.00
Mark Ouska	ProjectOne	New York	2019-08-07T13:00:00Z	8.00
Mark Ouska	PTO	Earth	2019-07-31T13:00:00Z	8.00
Mark Ouska	PTO	Earth	2019-08-01T13:00:00Z	8.00
Mark Ouska	ProjectTwo	Minnesota	2019-07-04T09:00:00Z	8.00
Mark Ouska	ProjectTwo	Colorado	2019-05-27T13:00:00Z	8.00
Mark Ouska	ProjectOne	California	2019-04-29T13:00:00Z	8.00
Mark Ouska	ProjectOne	Minnesota	2019-03-14T13:00:00Z	8.00

At the point of extraction, they are …

the same!

Design-Time Logic

Meaning-first approach

- Grounded in conceptual thinking
- Concepts consume no physical space
- Toss into triple stores (bag of triples)
- Globally unique connectors (URI driven)
- Reusable solutions for each concept

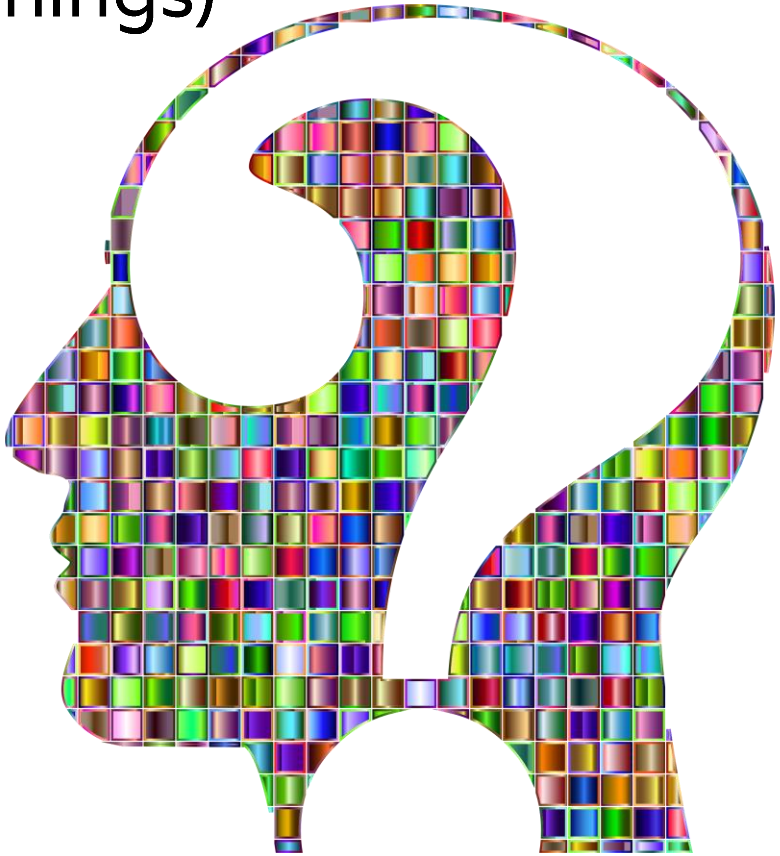
Meaning-First in Semantic Graphs

Terminology and Concepts (How we say things)

- Description logic for our domain, Ontology
- Partitioning strategies, namespaces
- In other words, the metadata

Actual data content (What we say)

- General subject matter information, Data
- Domain specific subject information
- In other words, our data content



Modeling Meaning

Conceptual

Logical

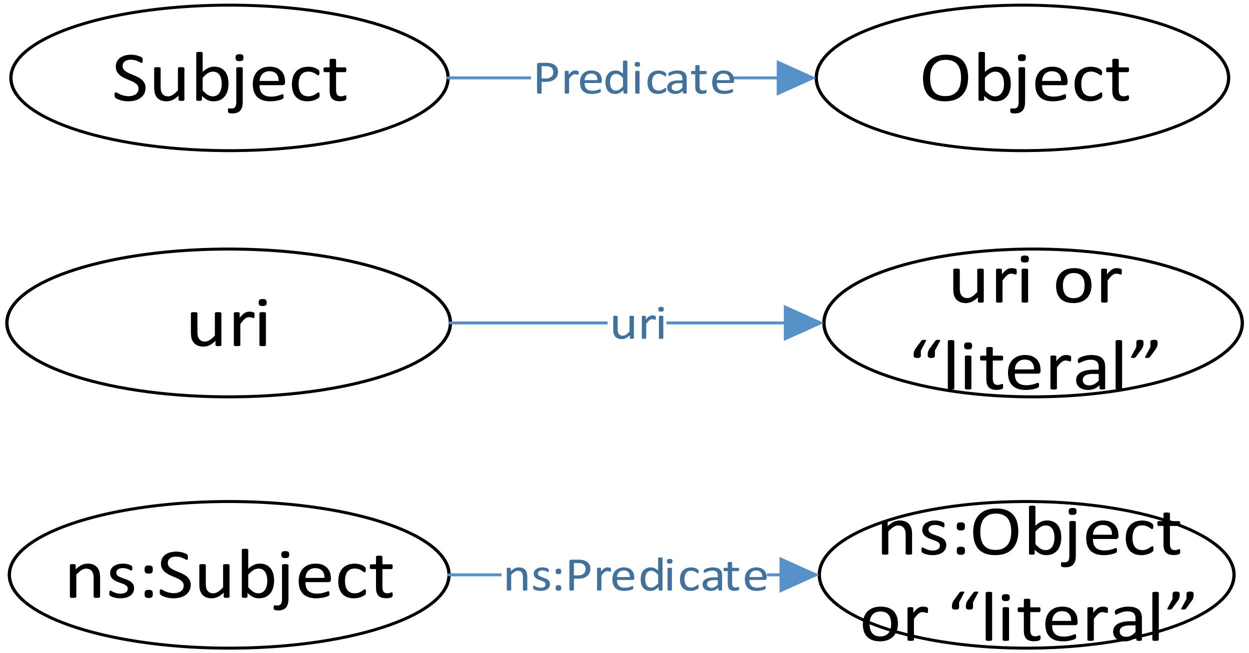
Physical

Relational - vs Semantics

Semantics

Data management is an open standard

- Defined in classes, contained in triples
- Bound to URIs
- Partitioned with namespaces
- Application refactoring rare
  - New table/column names simply mapped to ontology
  - db connections, added in queries mapped to ontology
  - Really cheap and low risk



Meaning-First Result (Data Centric)

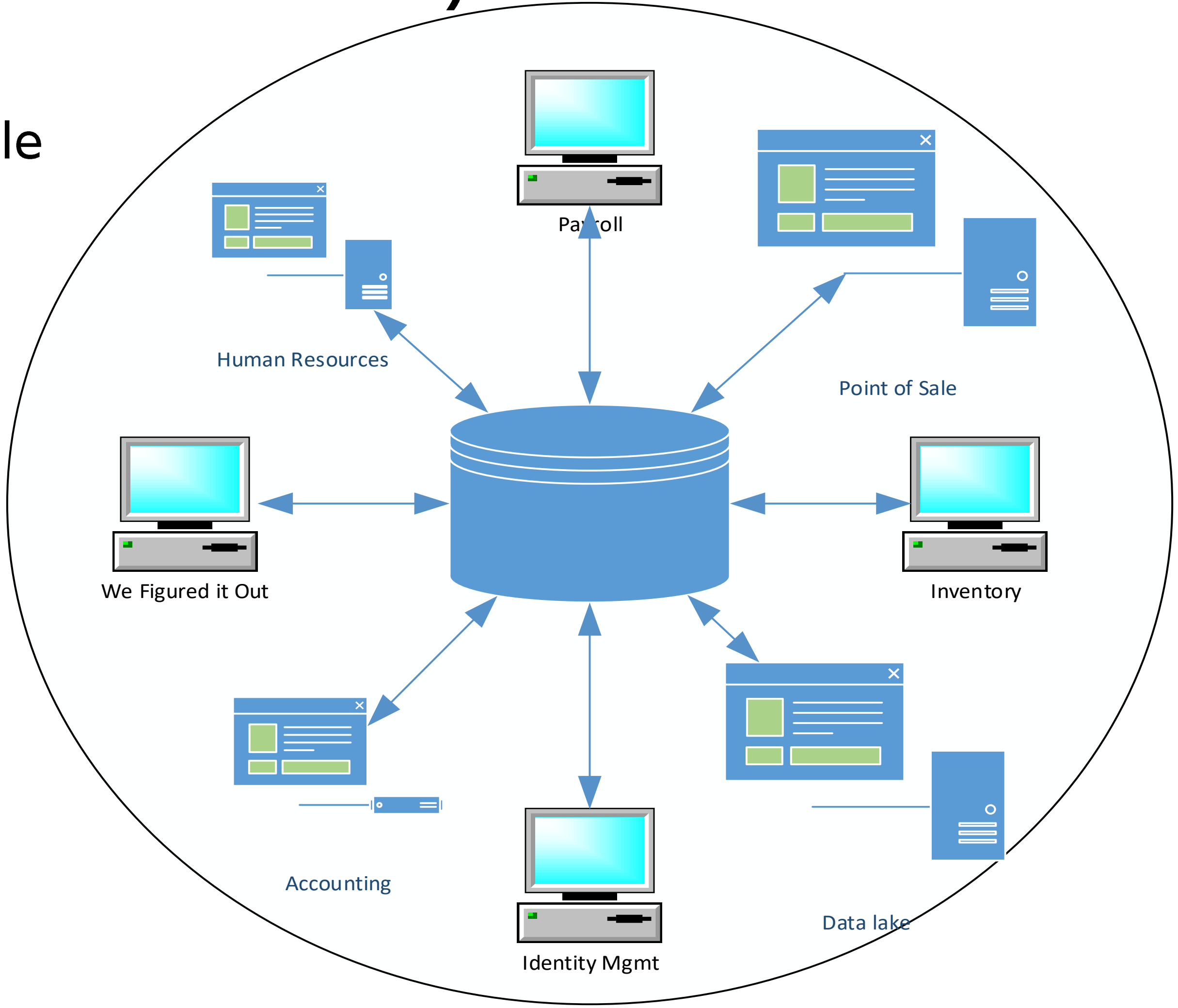
Data (*and its schema!*) freely co-mingle

- In single structure – triples
- In the same format - RDF

Data self-assembles into information

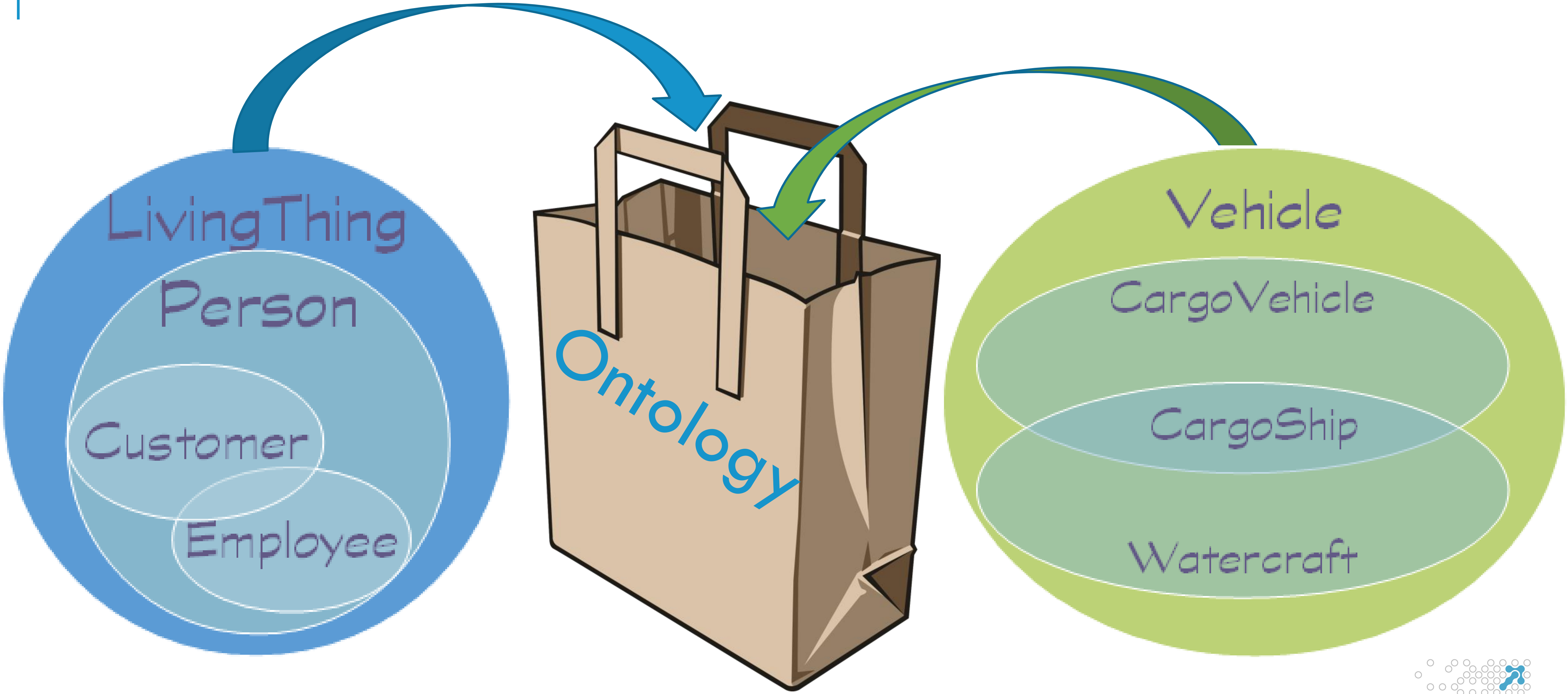
New schema information can be

- Discovered in data
- Applied to new data sets

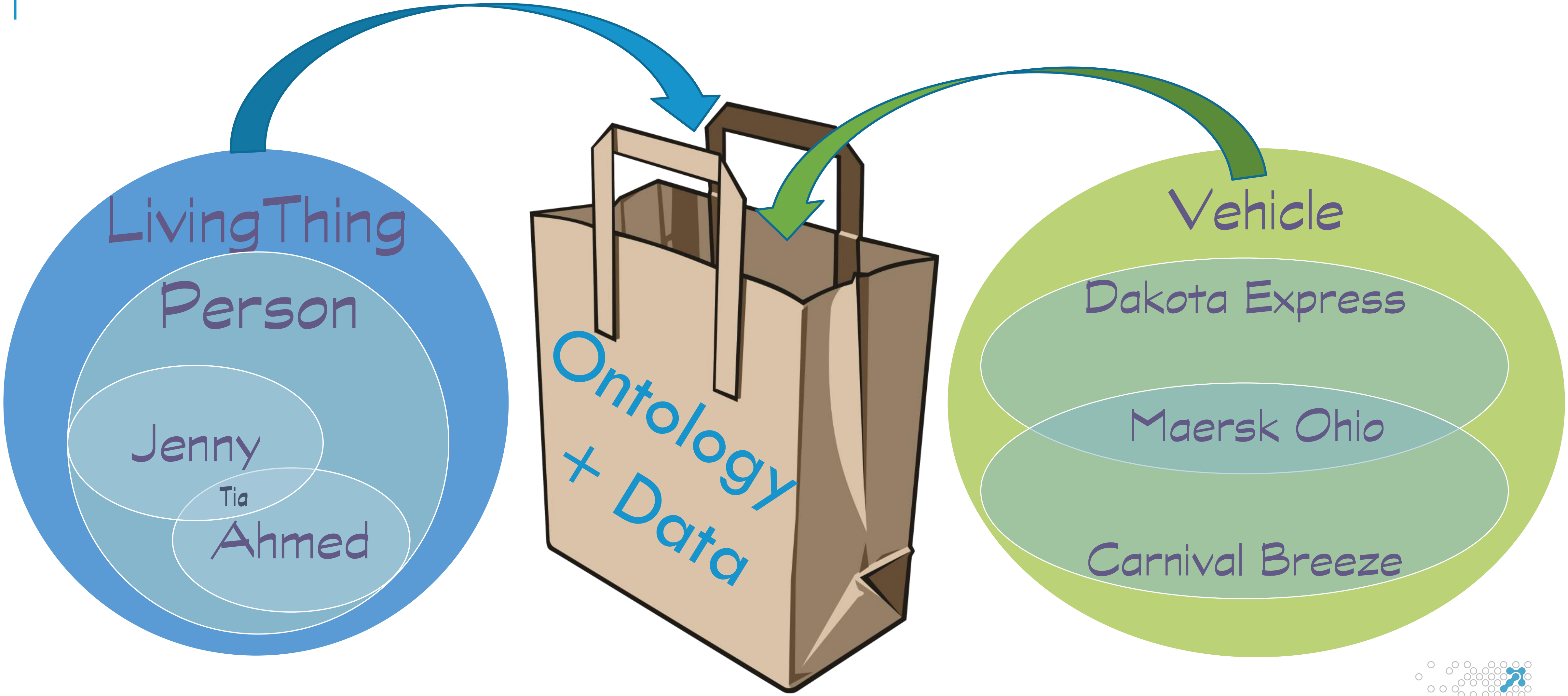


semantic arts

GraphDB = Defined Sets = Bag of Triples



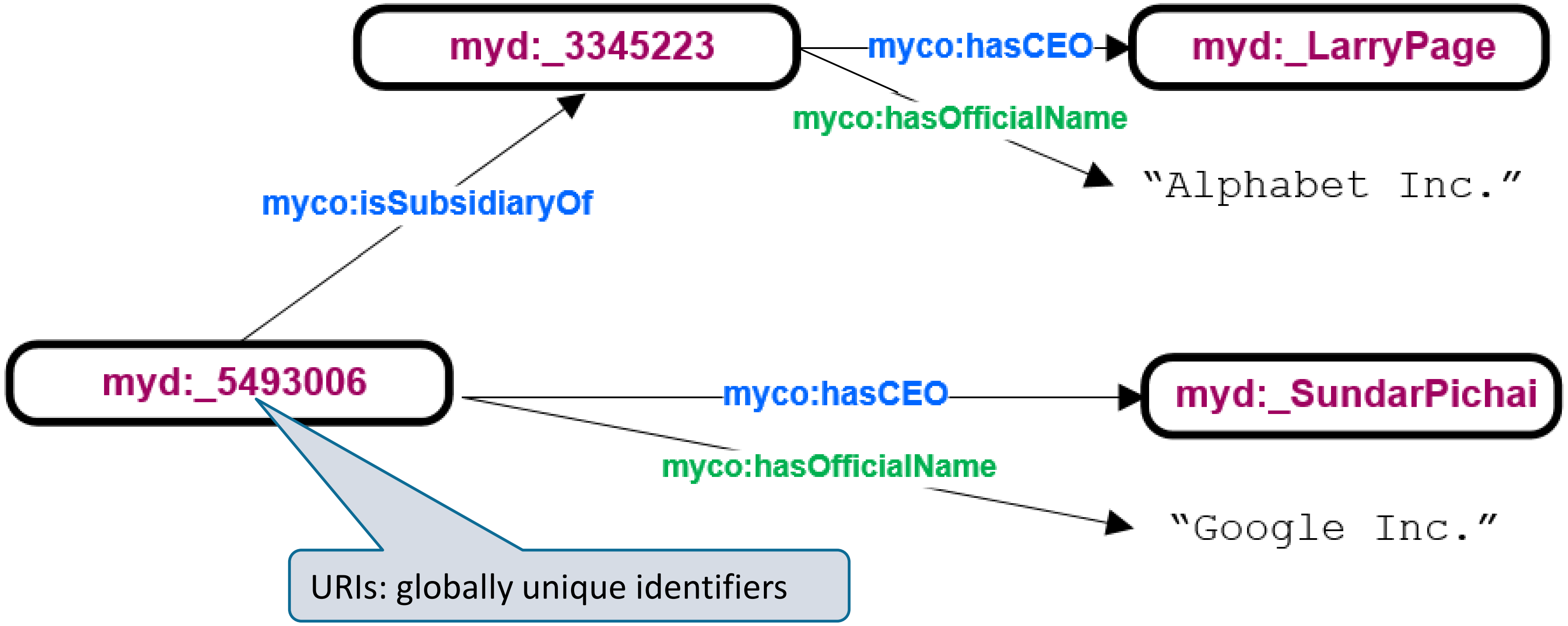
GraphDB = Defined Sets = Bag of Triples



Knowledge Graph

A set of nodes linked to other nodes.

A link is a triple representing an assertion. Subject, Predicate, Object. Here are 5 triples:



Ontology as Schema

Corporation

CoorporationID	Subsidiary Of	Official Name	CEO
3345223	NULL	Alphabet Inc.	Larry Page
5493006	3345223	Google Inc.	Sundar Pichai

Metadata:

- RDBMS schema:
  - Tables: Corporation, Person
  - Columns: CorporationID, Subsidiary Of, Official Name, CEO.
- Ontology
  - Classes: LegalEntity, Corporation, Person
  - Properties: isSubsidiaryOf, hasCEO, hasOfficialName

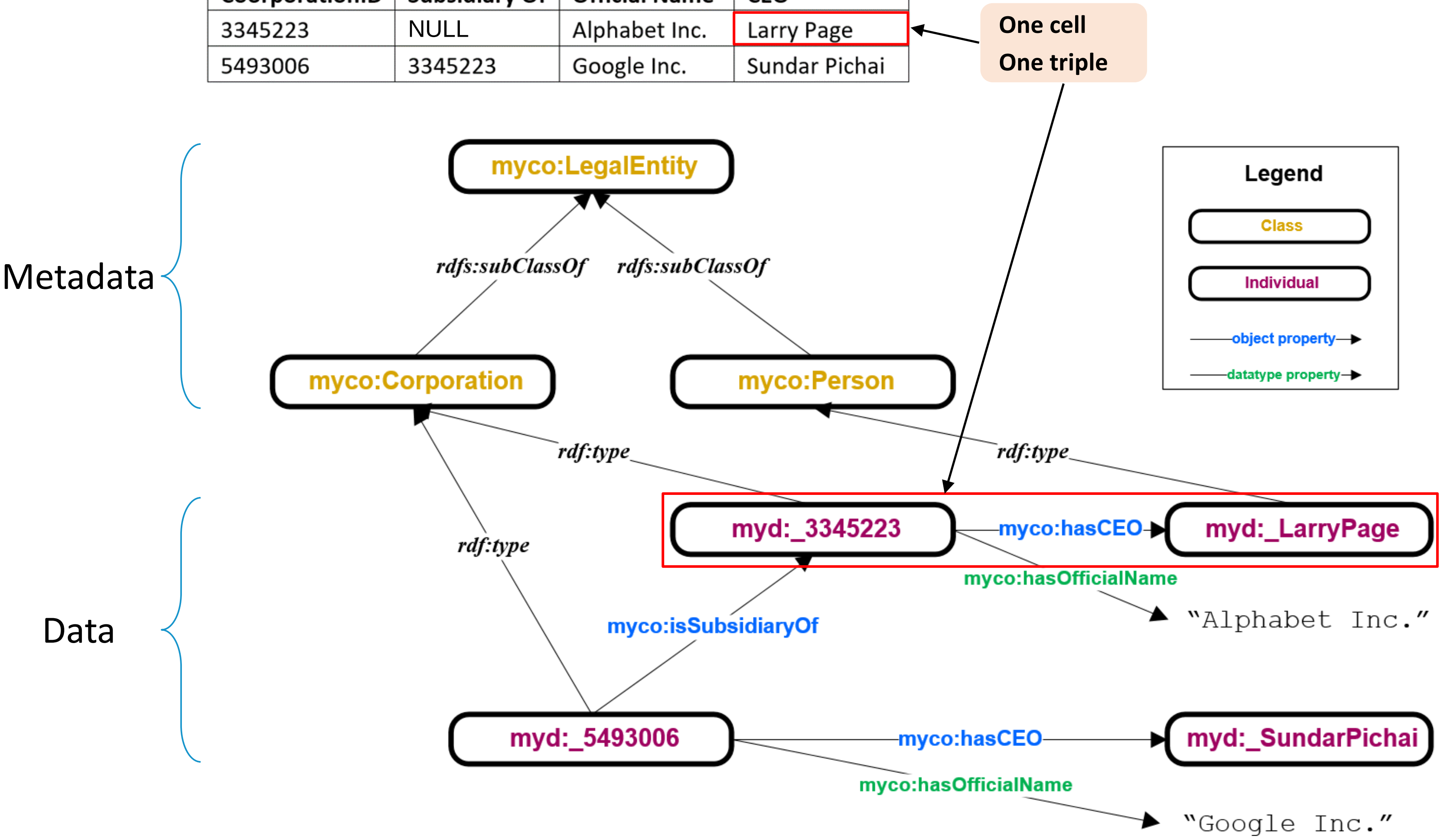
Data:

- RDBMS: Rows in Tables with columns with cell values
- Ontology: triples

OWL ontology language is used for ***both***.

Data and Metadata in One Graph

CoorporationID	Subsidiary Of	Official Name	CEO
3345223	NULL	Alphabet Inc.	Larry Page
5493006	3345223	Google Inc.	Sundar Pichai



Agenda

- Semantic Knowledge Graphs
- Globally unique identifiers, data self-assembly
- Design – vs – run-time logic
- No nulls required – Finally!
- Copernican future starts now!

No Place for Nothing

Meaningless Nulls

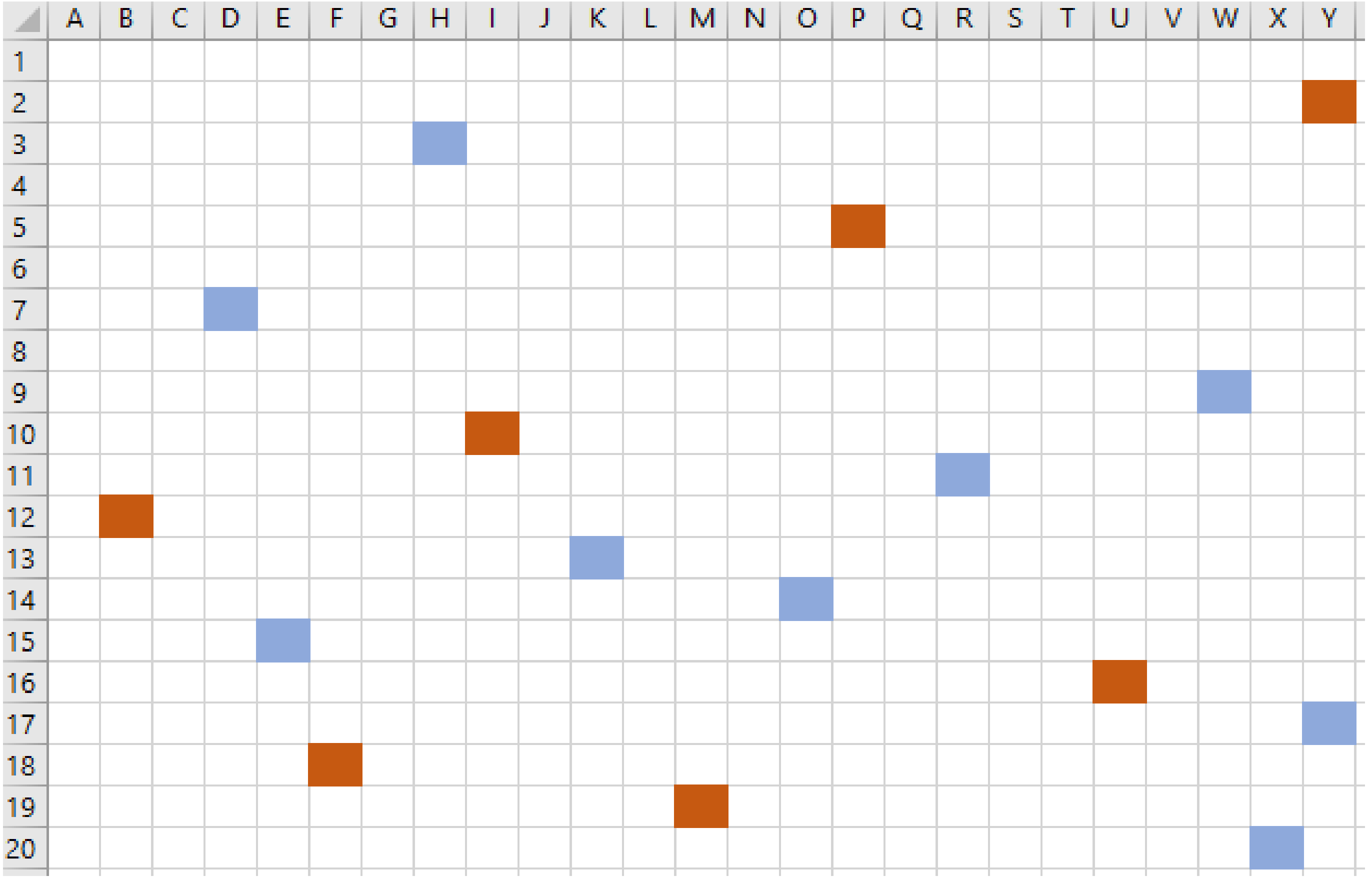
- Nulls
  - Ambiguous at best
  - Require code to interpret
- Flags
  - Generally codes
  - Require code/look ups to understand
- Booleans
  - Only one concept
  - Why two values?

Nulls – Meaningless and Costly

Require processing steps

- 1a. Find/eliminate nulls
  - or
- 1b. Find blue values
  - or
- 1c. Find red values

- 2. Determine meaning
  - Does null have meaning?
  - What does blue mean?
  - What does red mean?



Nulls – Meaningless and Costly

Require processing steps

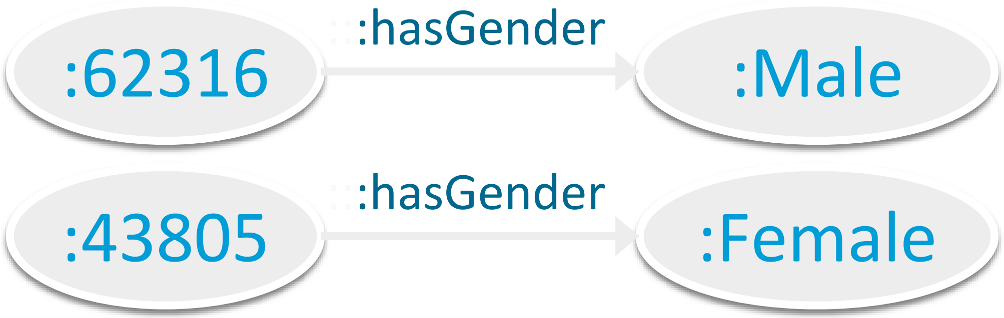
- 1a. Find/eliminate nulls
  - or
- 1b. Find blue values
  - or
- 1c. Find red values

- 2. Determine meaning
  - Does null have meaning?
  - What does blue mean?
  - What does red mean?

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1																									
2																									2
3								1																	
4																									
5																2									
6																									
7				1																					
8																									
9																							1		
10									2																
11																		1							
12		2																							
13											1														
14															1										
15					1																				
16																					2				
17																									1
18						2																			
19													2												
20																								1	

Flags, RDBMS Dance

1. Select data
   - SELECT Person FROM DB.Prsn
2. Determine category
   - ORDER BY Gender
3. Evaluate meaning
   - Join w/lookup table…
   - *(better yet, use a graph!)*

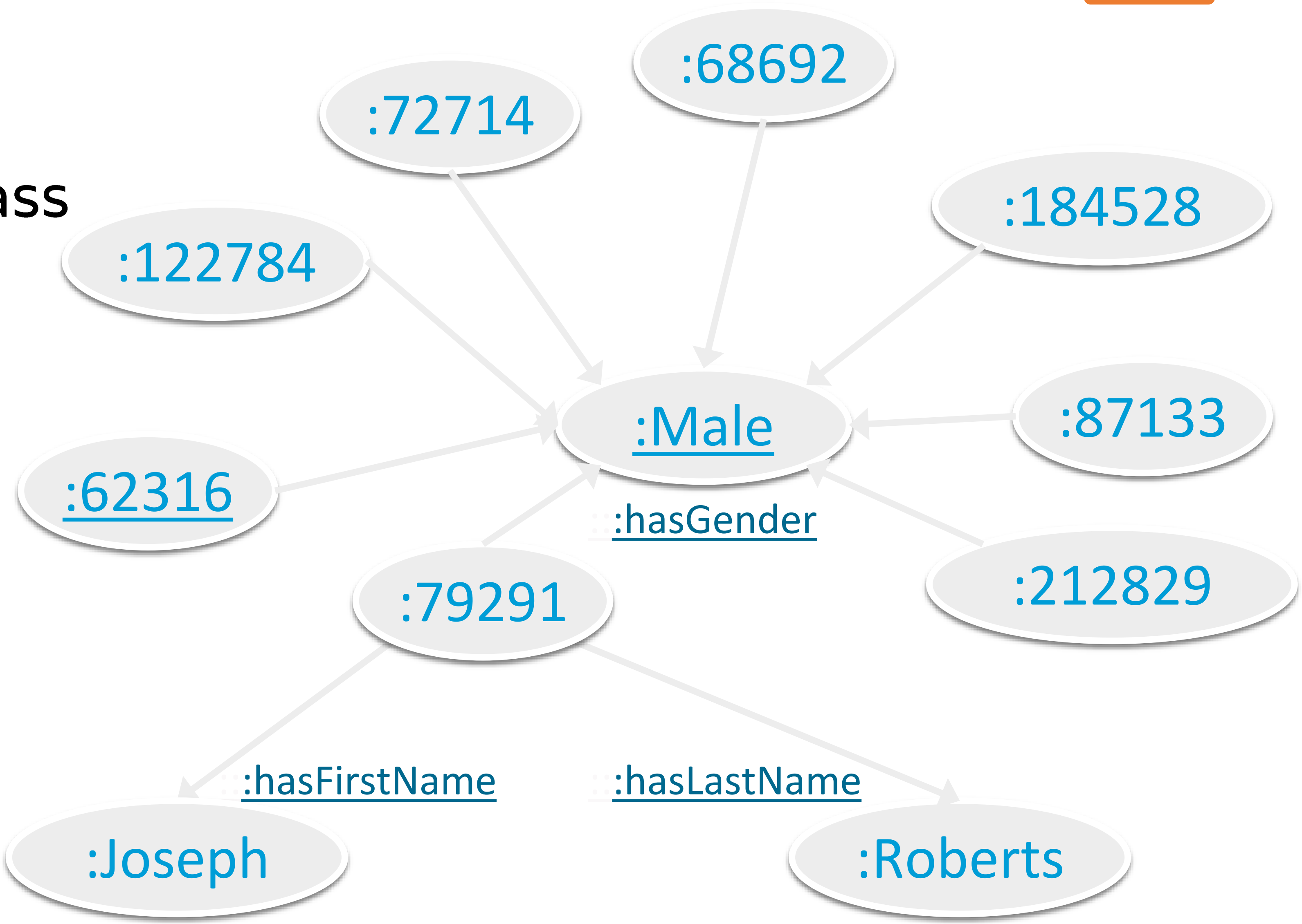


ID	Name1	Name2	Gender
43805	Jessica	Dube	2
48046	James	Wood	2
48341	Susan	Hall	4
50915	John	Wong	2
57088	Robert	Van	2
60911	William	Russell	2
62143	Michael	Stewart	4
62316	Richard	Ross	1
64211	Barbara	Jackson	5
68692	Elizabeth	Johnson	1
71588	David	Roy	2
72714	Linda	Kennedy	1
77512	Jennifer	King	3
79291	Joseph	Roberts	1
84410	Mark	Macdonald	2
87133	Patricia	Leblanc	1
90832	Donald	Michaud	6
100395	Anthony	Ouellet	2
108710	Thomas	Richard	3
122784	Matthew	Parsons	1
157211	Daniel	Pelletier	5
184528	Christopher	Poirier	1
196117	Mary	Lee	2
212829	Charles	Reid	1

semantic arts

Semantic Sets

1. Select desired class
  - Male
    - Follow your nose
  - Female
  - Transgender
  - Transsexual
  - Two-spirited
  - Intersex



Booleans

Meaningless binary values

- Y/N, Yes/No, T/F, True/False
- Yes what? False what?
- Only the application knows

Semantically meaningful

- ServiceContractSigned
- SafetyReviewCompleted
- No need for receipts for things you *didn't* buy!

semantic arts

Agenda

- Semantic Knowledge Graphs
- Globally unique identifiers, data self-assembly
- Design – vs – run-time logic
- No nulls required – Finally!
- **Copernican future starts now!**

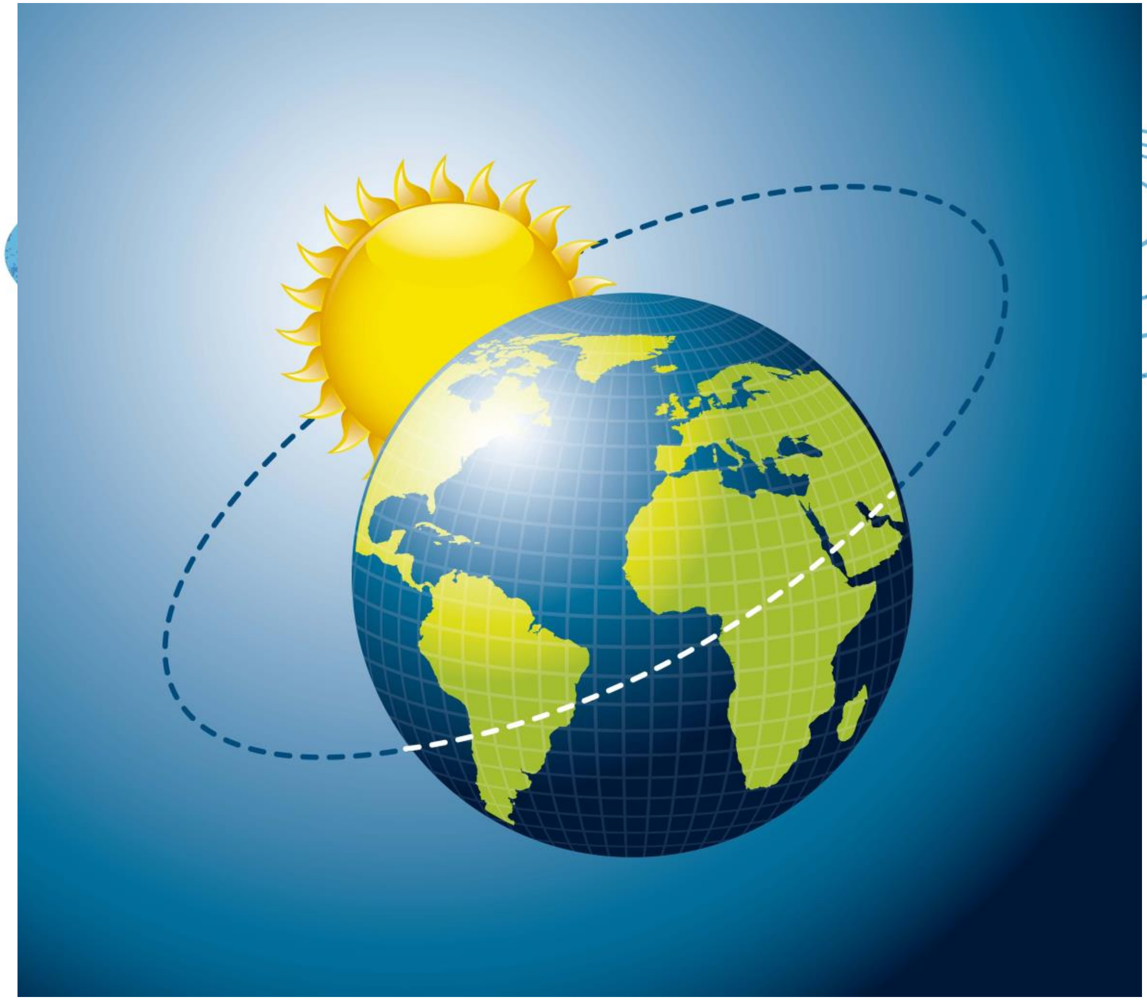
Copernican (Data Centric) Revolution

The problem:

- Application centric
  - *Sun revolves around Earth*
  - Obscured data ownership
  - Application defended data

The solution:

- Data Centric
  - *Earth revolves around Sun*
  - Shared data ownership
  - Self-defending data



Meaning-First Result (Data Centric)

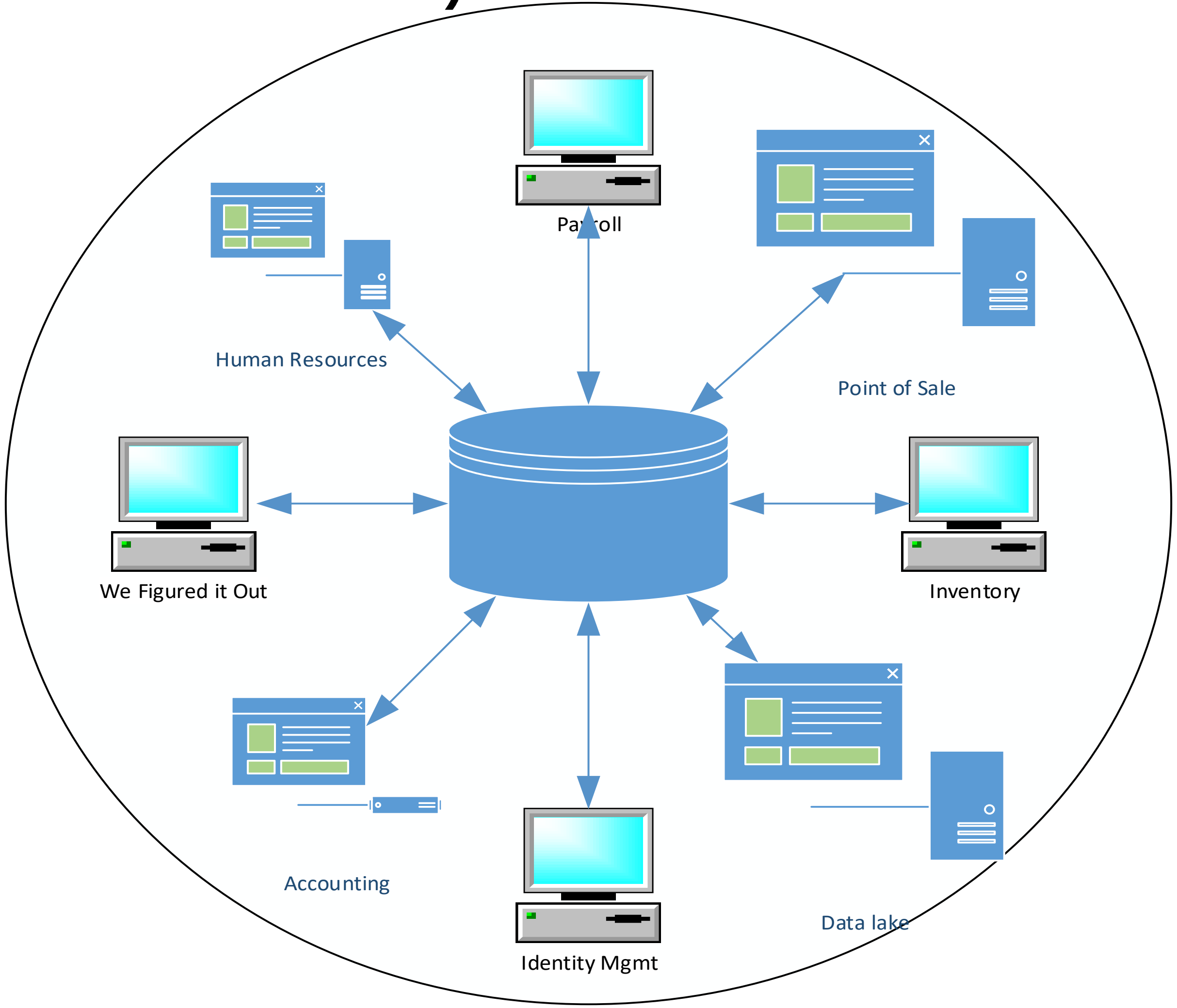
Data (*and its schema!*) freely co-mingle

- In single structure – triples
- In the same format - RDF

Data self-assembles into information

New schema information can be

- Discovered in data
- Applied to new data sets



Why Semantics Owns the Future

- **Structure-first**
  - **Late binding design**
  - **Structure building**
- Meaning-first
  - Early binding
  - Set definition
- Leverage
  - Schema transportability
  - Schema modularity

Structure-first

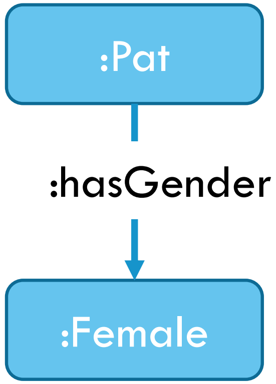
- Structure has no meaning
- Code is commonly developed to branch based on value
- Zero might mean false, but could also mean contractor
- All we really know is where in the structure to start looking

Why Semantics Owns the Future

- Structure-first
  - Late binding design
  - Structure building
- Meaning-first
  - Early binding
  - Set definition
- Leverage
  - Schema transportability
  - Schema modularity

Meaning-first

- Description logics have precise meaning
- Retrieve desired set without value branching code
- :Pat :hasGender :Female isn't ambiguous
- The data contains exact meaning

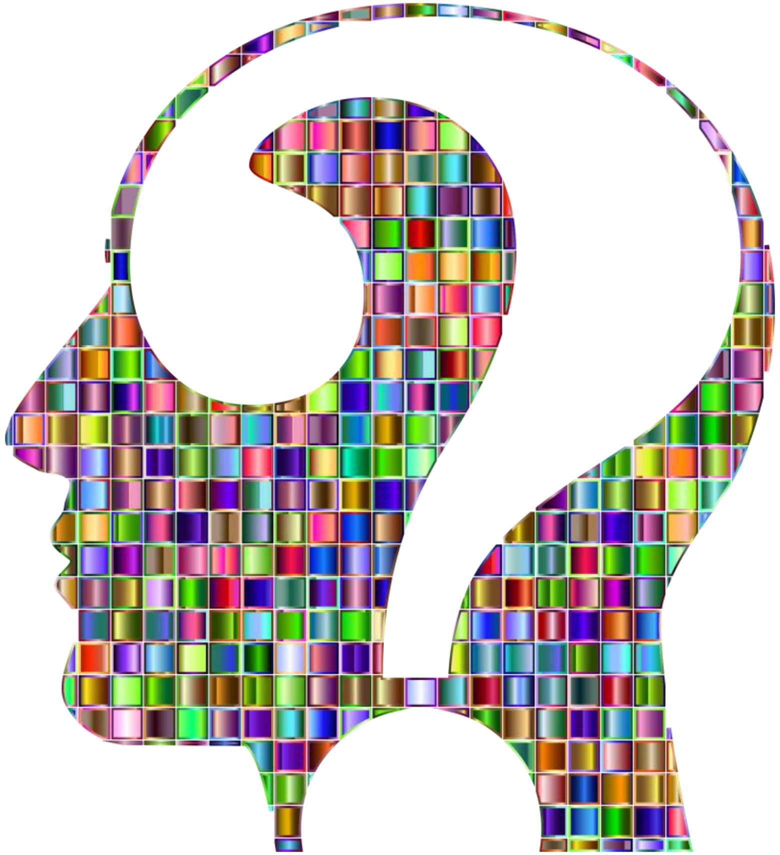


Why Semantics Owns the Future

- Structure-first
  - Late binding design
  - Structure building
- Meaning-first
  - Early binding
  - Set definition
- Leverage
  - Schema transportability
  - Schema modularity

Leverage

- Schema/ontology can be applied to new data sets
- Data sets contain meaning, ontologies expose it
- Modeled meanings can be imported as modules
- Independent modules separated by governance need



midwest architecture community collaboration

Thank You!

semantic arts

Mark Ouska
Mark.Ouska@SemanticArts.com

What is Semantic Technology and Why it Owns the Future