Making Sense of and Taking Control of Enterprise Content Silos

SEMANTiCS KARLSRUHE 2019 KEYNOTE

Michael J. Sullivan
Principal Cloud Solutions Architect
Oracle A-Team, Boston MA USA
michael.j.sullivan@oracle.com

ORACLE

Safe Harbor Statement

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 decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

My Journey: **1983**

- Started designing & implementing large-scale (print) documentation projects using declarative GML/ISIL as the markup language
- Worked with Linotype to generate page-composition directly from markup (predating PageMaker and other desktop-publishing by several years)
- Inspired by Edward Tufte, our focus was on creating a general-purpose pattern language for technical documentation to reduce obfuscation

Common refrain back then:
"if everything could be tagged properly, then all our problems would go away"

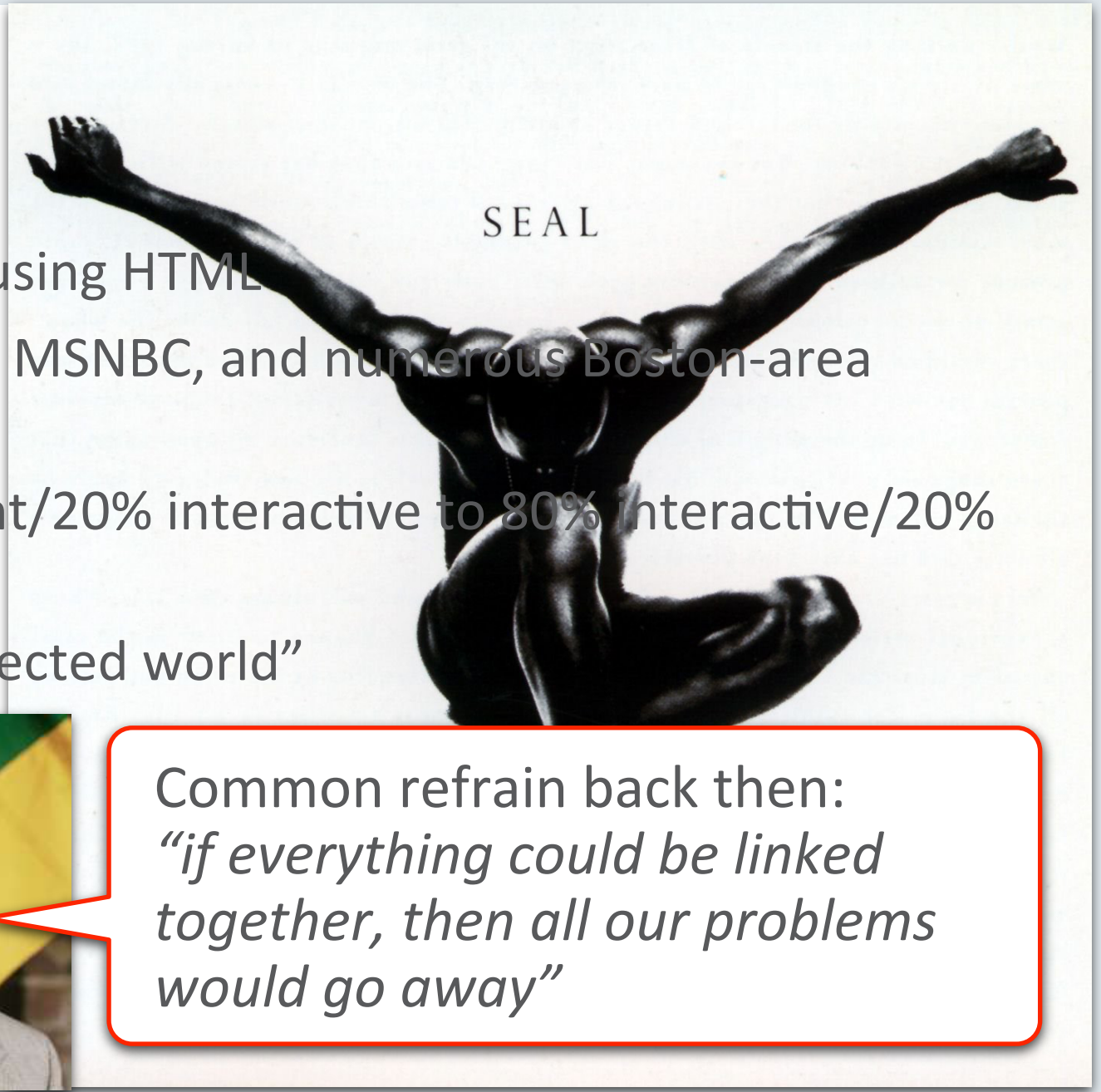
My Journey: **1988**

- Numerous projects for clients using Apple HyperCard/HyperTalk
- Uses the idea of separate cards and stacks to build interactive applications
- Would have been the first “web” browser if links worked between cards on different machines (i.e. needed the concept of URI)
- Instead, each deck was effectively walled in

Common refrain back then:
“if everything could be interactive, then all our problems would go away”

My Journey: **1994**

- First web-based projects for clients using HTML
- Clients included Stanford University, MSNBC, and numerous Boston-area software startups
- Our consultancy went from 80% print/20% interactive to 80% interactive/20% print in just two years
- The vision then was “one inter-connected world”



Common refrain back then:
“if everything could be linked together, then all our problems would go away”

My Journey: **1997**

- Founding partner of startup WayPoint Software — web electronic catalog software using C++ and an underlying Object Database
- Acquired by OpenMarket in 1998
- Pulled from the market in 2000 after acquiring FutureTense (later FatWire)
- But like much of the Dotcom bubble, it started out with much fanfare and promise

Common refrain back then:
"if everything could be described as an object, then all our problems would go away"

My Journey: **2007**

- Oracle acquires FatWire, and consequently meetes
- Biggest persistent problem faced by my clients: Taxonomy & Classification
- It is obvious to our team that tagging as implemented is nothing more than a “folksonomy” and is unsustainable
- I propose hybrid, graph-like, modeling to the PMs, but the idea is rebuffed
- Much of product focus becomes search-oriented instead

Common refrain back then:
“if we could just index everything properly, then all our problems would go away”

My Journey: **2017**

- Began investigating what it would take to implement “Taxonomy as a Service” hosted on Oracle OCI with uptake by Oracle CX applications
- Discovered that there were numerous experts in the Knowledge Graph industry with many years of experience (e.g. PoolParty)
- Discovered that Oracle DB was not on anyone’s radar from this Knowledge Graph industry, in spite of being one of the first to implement RDF
- Sought to bring th

New refrain:
“Keep the mess, but extract the knowledge, then all our problems will go away” :P


ORACLE
Live to Code
NAUTICA

Typical Enterprise data: a polyglot mess

- “Cross-industry studies show that on average, less than half of an organization’s structured data is actively used in making decisions, and less than 1% of its unstructured data is analyzed or used at all." — Harvard Business Review, 2017
- “The rising role of content and context for delivering insights with AI technologies, as well as recent knowledge graph offerings for AI applications have pulled knowledge graphs to the surface.” — Gartner, 2018
- Notwithstanding, chronic data integration problems remain entrenched as ever
- And traditional Data Warehouse technologies have their own problems

Typical Enterprise data: a polyglot mess (cont.)

- Graphs to the rescue — but especially RDF/Semantic graphs
  - **Reason 1**: RDF requires URIs not strings for resources, and this makes integration easier (e.g. no duplicates)
  - **Reason 2**: SPARQL/SHACL have built-in "reasoners" that can make semantic sense out of disparate data (e.g. sameAs, differentFrom, inverseOf, instanceOf, narrower, broader, etc.)
- Additionally, RDF "middleware" can hide the complexity of RDF
- Oracle's implementation of RDF/Semantics piggybacks on top of the power of Oracle's robust database features (e.g. Real-time materialized views, RMAN, RAC, DBlinks, DataGuard, Autonomous, etc.)

Data Warehouse Challenges

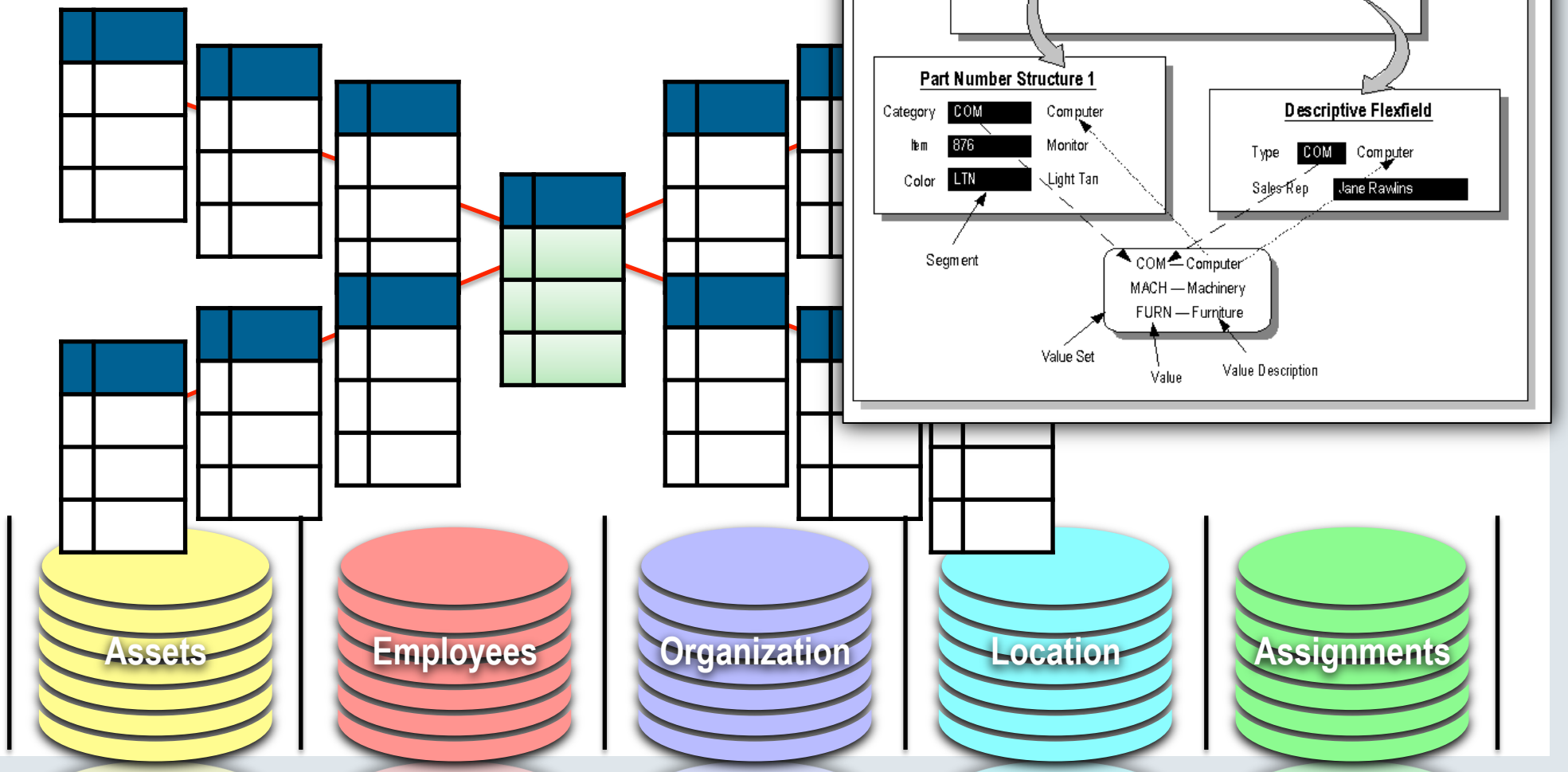
- **E-Business Suite Flexfields (Contexts)**
  - Need to build a table for each context — and there are hundreds of them
- **Conformed Dimensions**
  - Before moving forward the source system product owners must agree on a mutually agreeable resolution to resolve data anomalies
- **Slowly Changing Dimensions**
  - Typically requires changes to business logic in your app
- **Time Series Queries**
  - Not scalable because every change/addition requires dropping the tables and rebuilding from scratch

How RDF/OWL can help solve Data Warehouse challenges

- **E-Business Suite Flexfields (Contexts)**
  - Schema on Read
- **Conformed Dimensions**
  - sameAs Inference
- **Slowly Changing Dimensions**
  - Forward Chaining
- **Time Series Queries**
  - Events
  - dateWeb
  - Class/Subclass Inference
  - Multiple Inheritance
  - Forward Chaining

Typical Relational Data Warehouse (fragile, cumbersome)

Need to build a table for each context — and there are hundreds of them

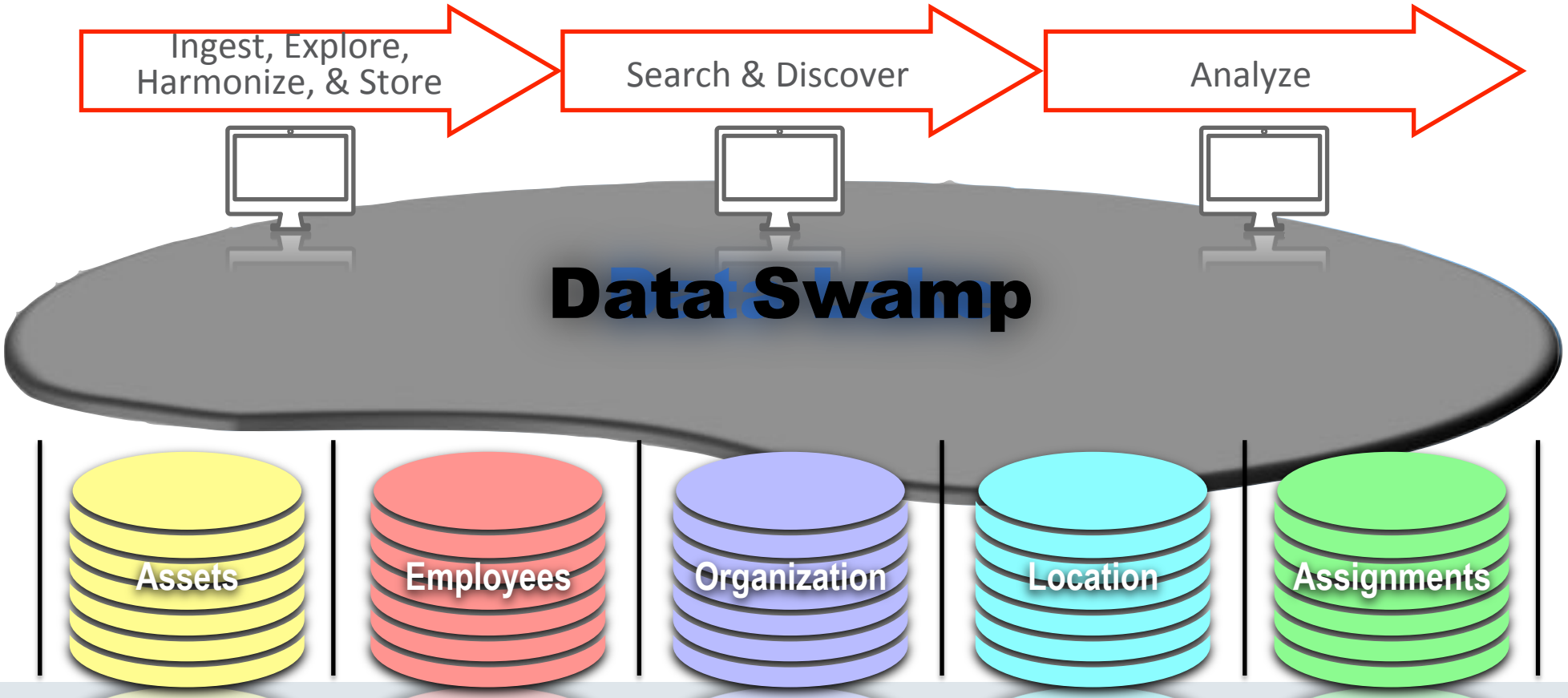


Typical Big Data Warehouse (Complex Methodology)

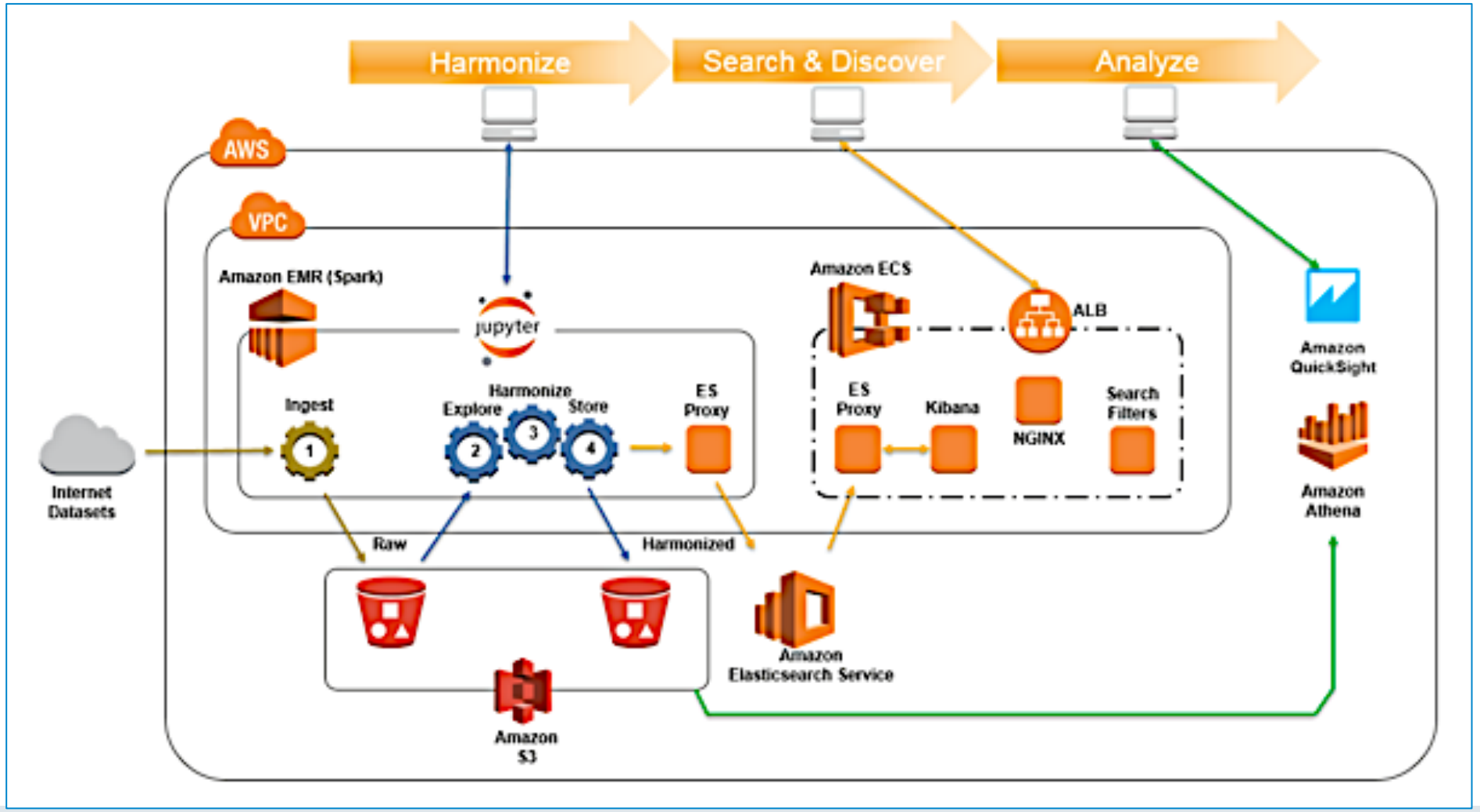
see: https://aws.amazon.com/blogs/big-data/harmonize-search-and-analyze-loosely-coupled-datasets-on-aws/

Tools needed:

AWS Glue, Catalog, Athena, S3, EMR, ECS, Redshift, QuickSight, etc.

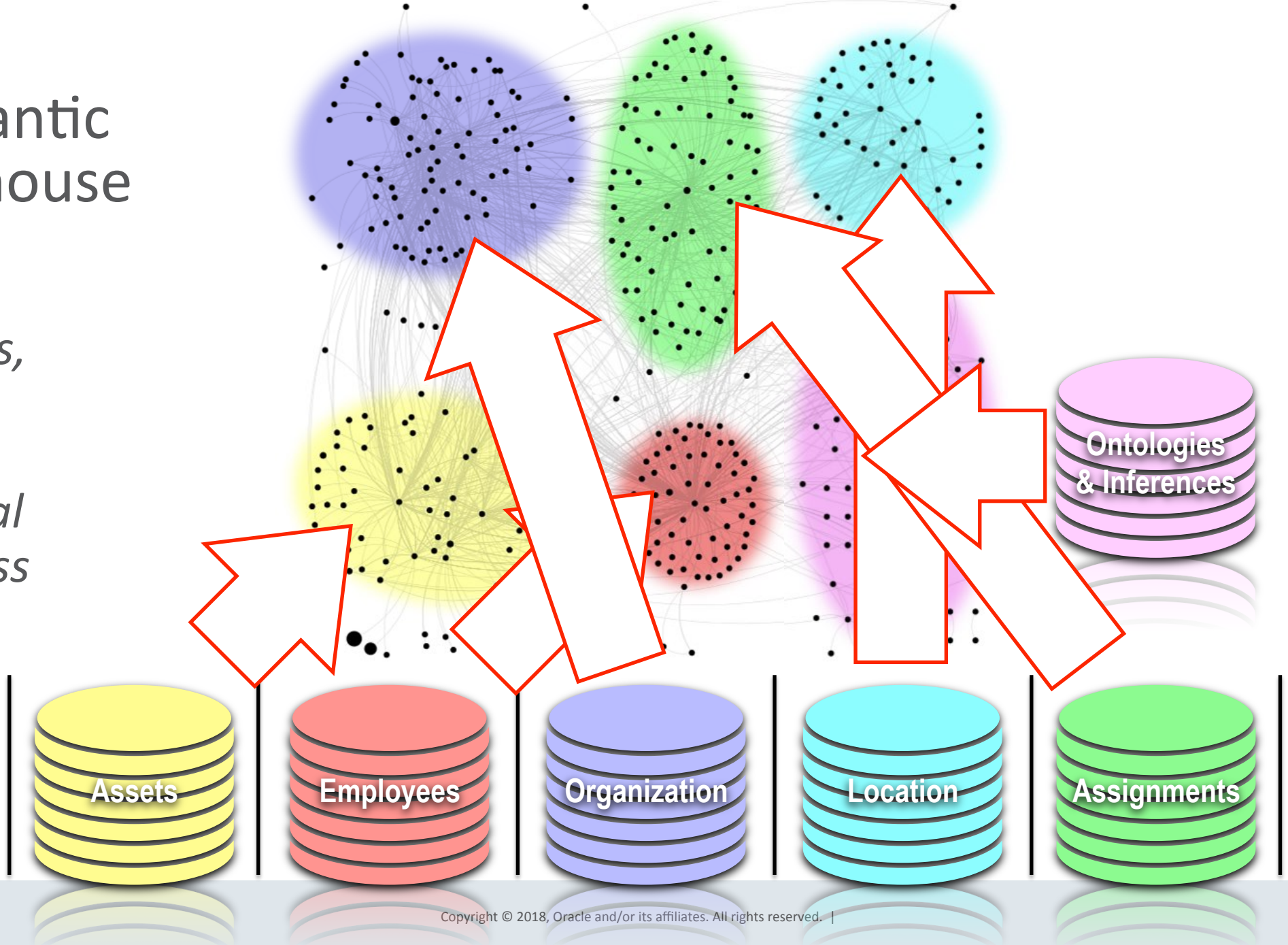


https://aws.amazon.com/blogs/big-data/harmonize-search-and-analyze-loosely-coupled-datasets-on-aws/



Knowledge Graph Semantic Data Warehouse

Because of URIs, RDF can intrinsically provide a virtual 360° view across all models



All wonderful in theory, but…

- Reconciling common URIs between siloed applications is problematic
  - Solving semantic heterogeneity issues is non-trivial
  - Will require domain expertise and/or statistical ML algorithms
  - e.g. michael.j.sullivan@oracle vs. Michael Sullivan vs. msulliv1234 etc.
  - Ontology/Schema Mapping & Reference Reconciliation are active areas of research
- Generally, ownership of the metadata must be maintained by the silo, so orchestration is also a big issue
- Because of the above, only the silo-owners should generally have write access
- Copying/migrating data from silo to silo is untenable
- ETL must follow from the bottom up, not top down (otherwise it will take years)

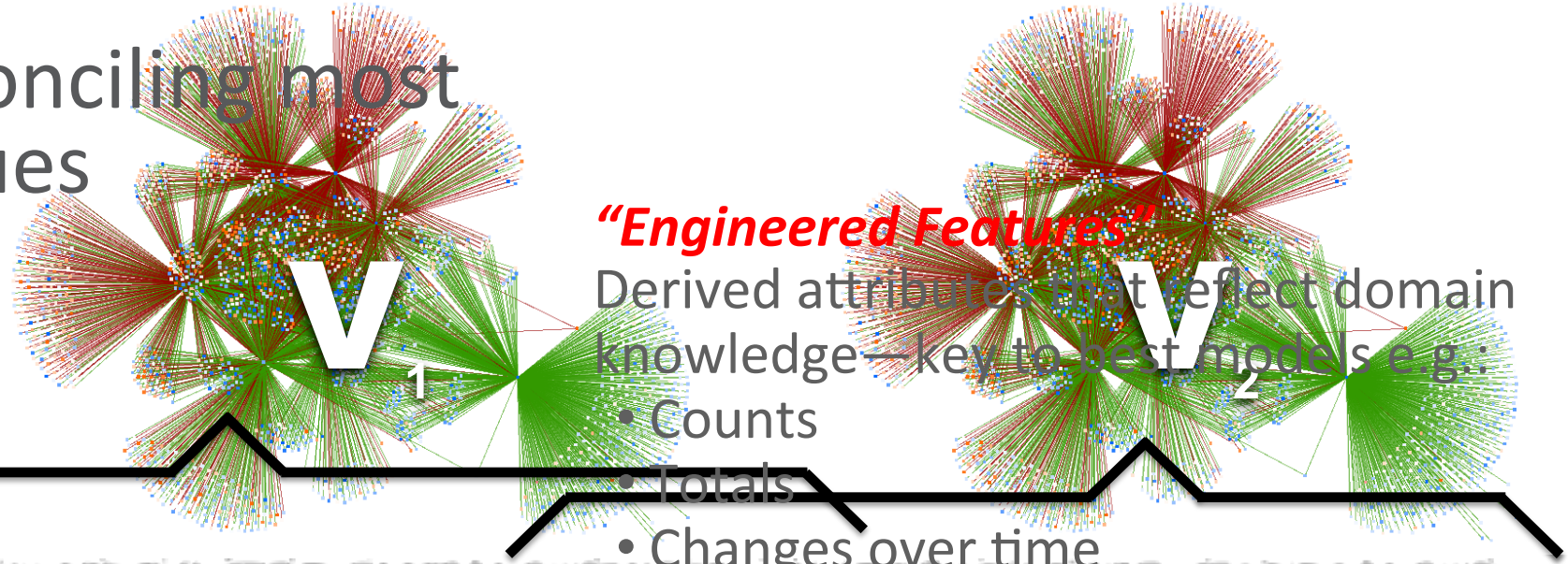
Solving Semantic Heterogeneity

A practical, workable methodology:

1. Collect a set of use-cases/queries you want to answer across the set of silos
2. Create a top-level schema **T** that contains just enough information to answer your use-cases
3. Map each silo schema to **T** using a set of OWL/SCHACL axioms **A**
4. Create an entailment **E** using **A** over **T** + silos
5. Create a virtual model **V** for **E** + **T** + silos
6. Query **V** to answer your use cases
7. Repeat steps 2-6 as new use cases come in

N.B. It is a mistake to try to completely map all the silos as step #1. It needs to be driven by use-cases and done iteratively.

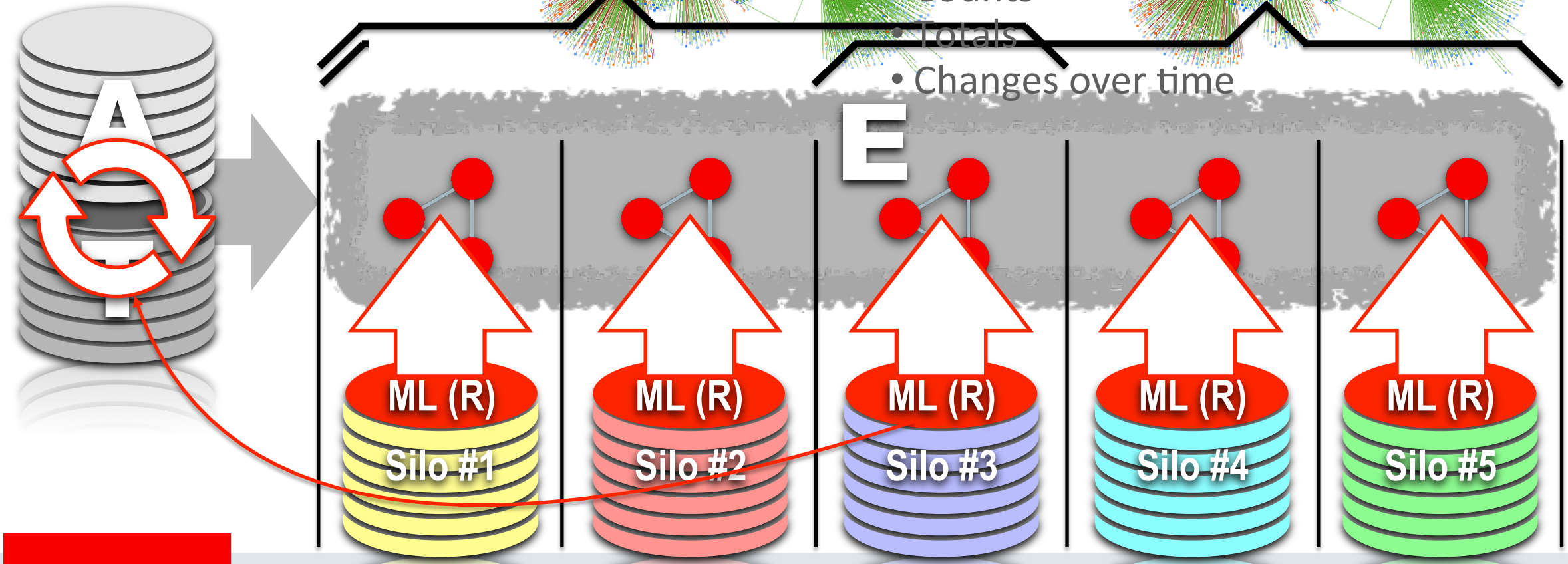
A pattern for reconciling most of the known issues



"Engineered Features"

Derived attributes that reflect domain knowledge—key to best models e.g.:

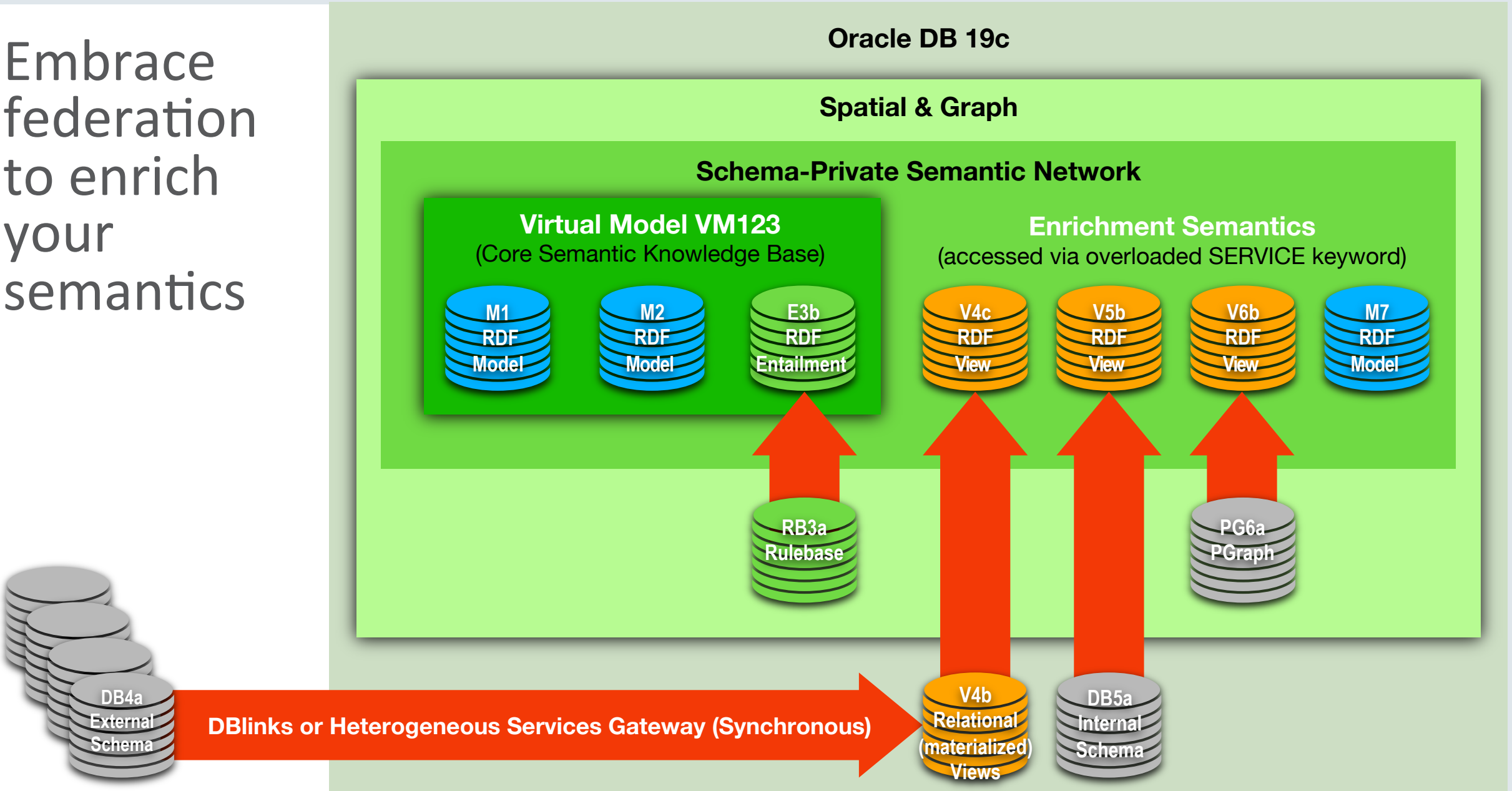
- Counts
- Totals
- Changes over time



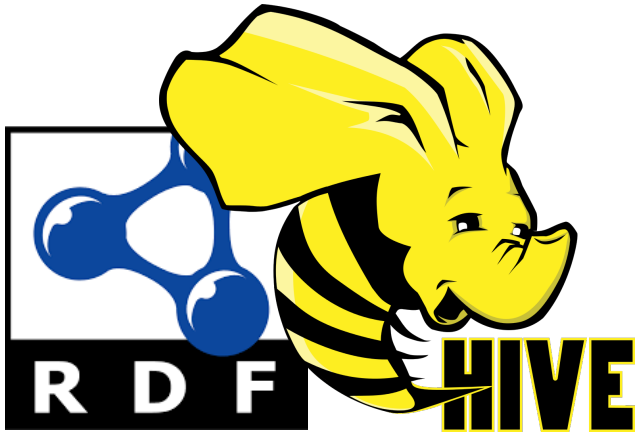
Multiple-Models versus Named-Graphs

- For named-graphs hosted within the same repository, independence of updates/deletes may not be guaranteed as all quads are in the same model (typically, there is no way for triple-level security for example)
- Scalability (one instance with multiple named graphs vs. multiple independent instances)
- Security/governance is easier with multiple independent instances

Embrace federation to enrich your semantics



Reading Triple Stores in HDFS?



- Possible? Yes.
- Here is how you would do it:
  - Create a Hive table on an n-triple format file, and then query this Hive table via an external table in Oracle Database using Big Data SQL
  - Then create RDF views on the external table and use RDF features in the database
  - The data will continue to reside in HDFS

ORACLE

What about multi/mixed cloud architectures?

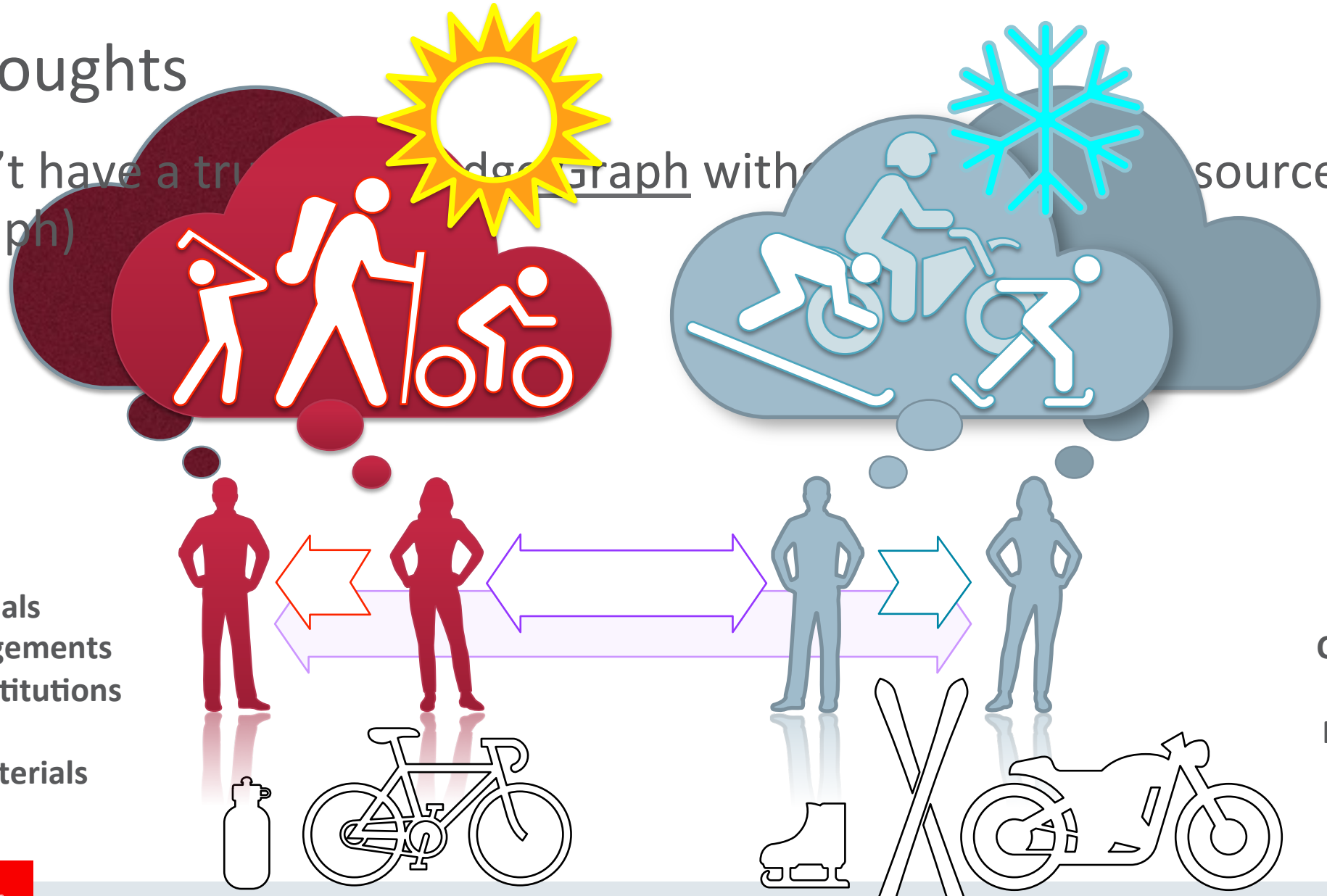
- 85 percent of enterprises currently have a multi-cloud strategy
- One obvious pattern: One LOB needs to share highly sensitive data while another LOB needs powerful processing for app development or big data projects — these teams might be best served by different types of cloud solutions
- e.g. Microsoft Azure + Oracle OCI. Currently in nascent stages, with Cross-cloud interconnect and Unified identity & Access Management coming
- A less-desirable pattern: ad-hoc multi-cloud usage

What about hybrid cloud architectures?

- Primary pattern today: on-prem legacy apps with storage and SaaS apps in the cloud
- The Vision: combine private and public clouds as needed to achieve optimal performance, efficiency, and economy across the enterprise
- The Goal: Flexibility/agility will dramatically lower costs and give businesses a competitive advantage
- Automated Kubernetes workflows, Ansible, Chef, etc. will enable the ability to stand-up sandboxes as needed -- and throw them away as needed

Final thoughts

- You can't have a tru dge Graph with sources in your graph)



Blogs
Articles
Whitepapers
Degrees
Skills
Tools
Awards
Training materials
Speaking engagements
Educational Institutions
Certifications
Conference materials

Authored
Approved
Published
Attended
Completed
Developed
Managed
Invented
Co-authored
Supported
Participated
Presented


ORACLE®