SingerLinks
Connecting The Dots...

The Property Graph – A New Model For Metadata

Data Architecture Summit 2017 - Chicago IL

Presenter: John Singer

SingerLinks
Connecting The Dots...

The Property Graph – A New Model For Metadata

John Singer – Founder SingerLinks Consulting LLC

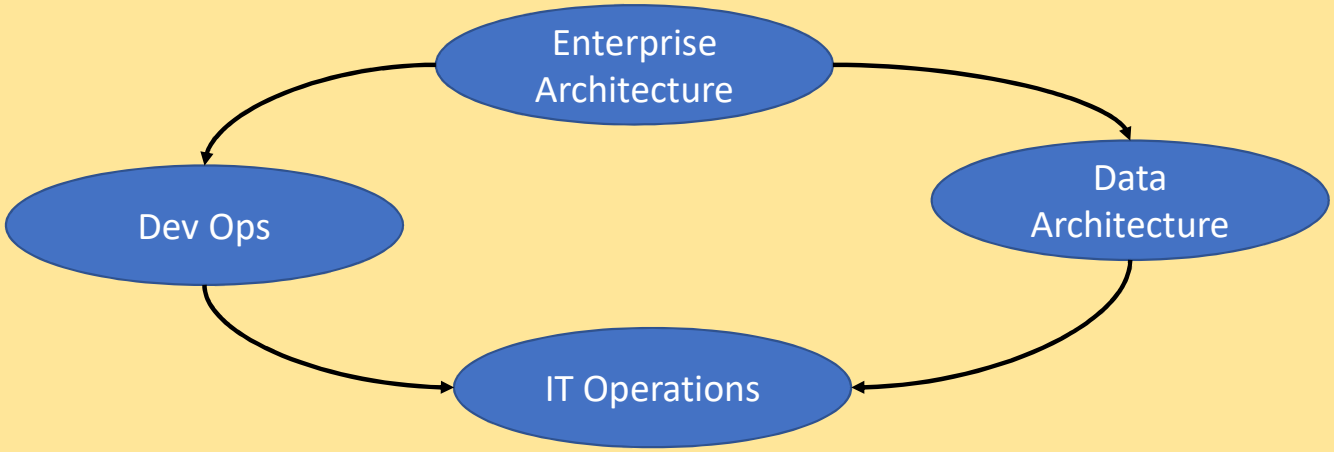
- 36 years IT experience in various Data Architecture roles
- Expansive ITIL CMDB implementation (Business, Application Technical metadata) motivates this presentation
- Currently working on Database to Java impact analysis Knowledge Map
- DATAVERSITY blogger - http://www.dataversity.net/author/john-singer/

SingerLinks
Connecting The Dots...

An Expanding Scope For Metadata

Communities of Metadata

- Enterprise Architecture (business process, service models, architectural blueprints)
- Data Architecture (data assets, data movement)
- Dev Ops (application components)
- ITIL CMDB's (IT system components).
- See my DAMA 2008 presentation: American ITIL – Winning The Metadata Contest



Disjointed processes and tools don't provide a holistic view of IT – Can Metadata management help?

SingerLinks
Connecting The Dots...

A History Of Mankind - The "IT Tool Age"

Stone Age

Bronze Age

Iron Age

IT Tool Age

- Metadata Repositories – essentially tools to manage tools
- Dev-Ops – "tool chains"

Metadata repositories and ITIL CMDB's provide a partial solution but hit limitations of relational databases.

SingerLinks
Connecting The Dots...

Traditional Approach – Domain Model

Traditional Metadata Management and CMDB tools use relational databases to store their data.

- **Domain Specific Model:** The domain being managed is modeled using traditional entity relationship or object oriented modeling techniques resulting in a table for every entity type needed. For a good example look at the DMTF website:

  The DMTF's Common Information Model (CIM) provides a common definition of management information for systems, networks, applications and services, and allows for vendor extensions.

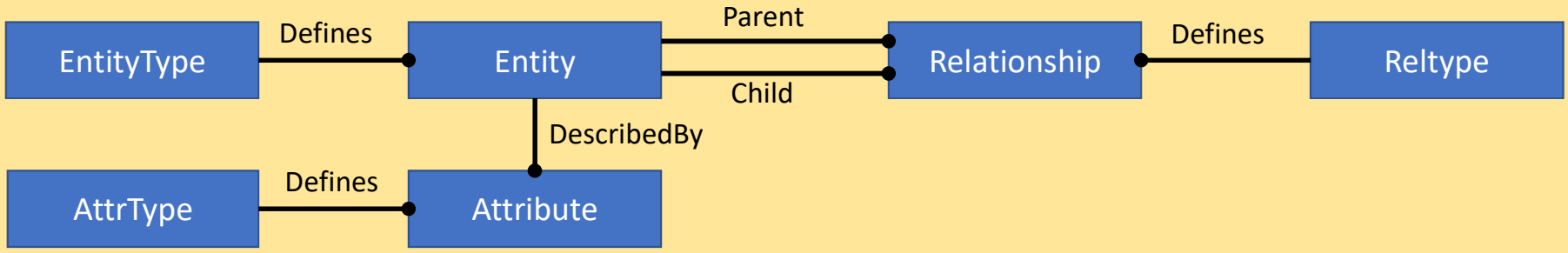
Pro's – Represents exactly what you want.

Con's – Expanding the model requires changing the database structure.

SingerLinks
Connecting The Dots...

Traditional Approach – Generic Abstract Model

Generic Abstract Model: The abstract model provides a physical schema that will store information about any entity type and their relationships.

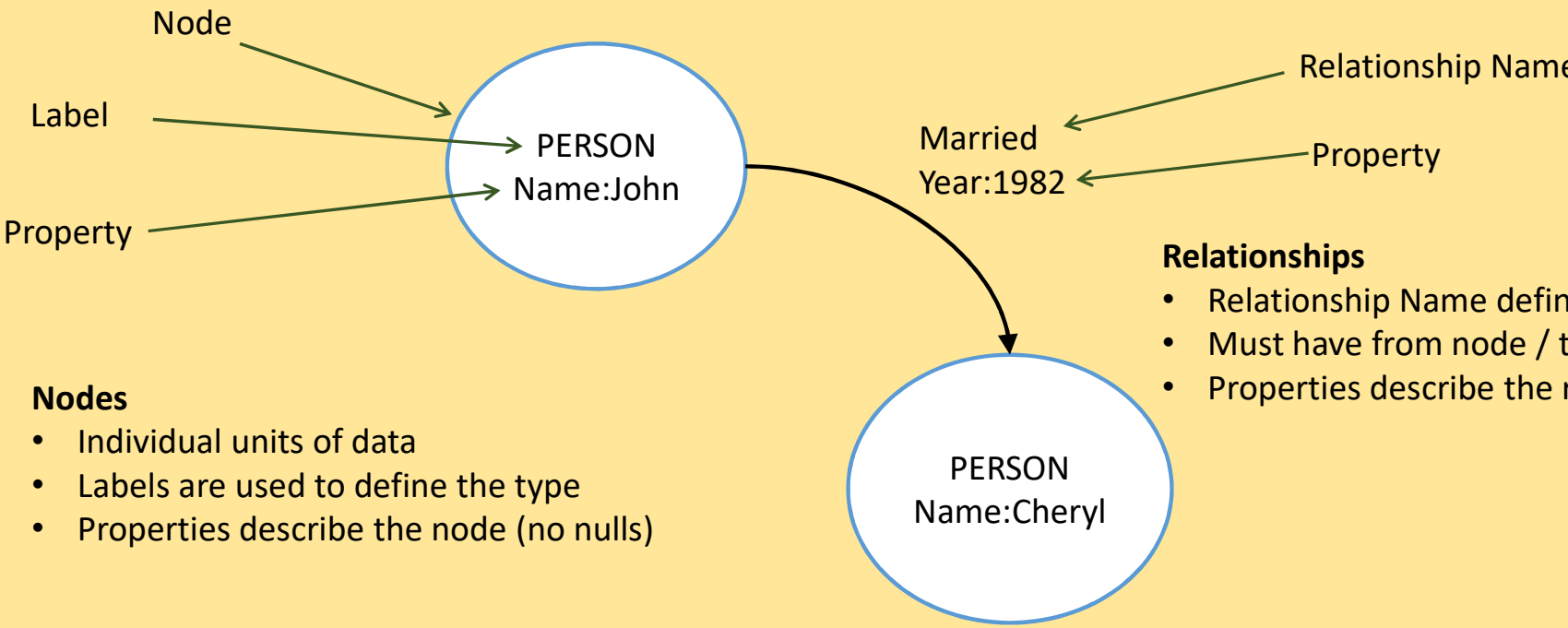


Pro's – Easily expands to support new entity and relationship types.
Con's – Complex queries require too many self joins

The generic parent entity – relationship – child entity is a Graph!

SingerLinks
Connecting The Dots...

A New Approach – The Property Graph



Nodes
- Individual units of data
- Labels are used to define the type
- Properties describe the node (no nulls)

Relationships
- Relationship Name defines the type
- Must have from node / to node (directed)
- Properties describe the relationship (no nulls)

Next generation metadata repositories and CMDB's will run on graph DB's

SingerLinks
Connecting The Dots...

Property Graph Advantages – Schema-less (?)

- You hear lots of terms: Schema-less, Schema on write, Schema on read – what does this mean?
- My view:
  - Physical data model (structure) is pre-defined – Nodes, Labels, Properties, Relationships
    - In a relational database, you first do a “create table”, then you do an “insert” to add a row
    - In a graph database, you simply do a “create” to add a node
  - Logical data model (meaning) is created at run-time
    - New Labels, Properties, or Relationship Types are added when you create nodes and relationships – no need to pre-define them

The logical model can gracefully evolve over time with no need to restructure the database.

SingerLinks
Connecting The Dots...

Property Graph Advantages - Relationships

- Relationships are now first class citizens. Each relationship is stored directly in the database which means:
  - Relationships can be queried directly by name (how many friends are in the system?)
  - Relationships can be counted (how many friends does XYZ have?)
  - Nodes can be filtered by the types of relationships they have or don't have (find people with no friends)
  - "Distance" can be calculated (how many friend of a friend relationships are between person A and person B?)
  - Relationships can have their own properties (when did A become a friend of B?)
  - Node to Node queries can work even if you don't know the number of "hops" between them.

The importance of relationships as objects in their own right cannot be over-stated

SingerLinks
Connecting The Dots...

Property Graph Advantages - Expandable

The combination of two nodes and a relationship can represent a "proposition" in the simple form of subject-predicate-object.

Subject	Predicate	Object
IP 10.100.1.256	BoundTo	xyz123abc

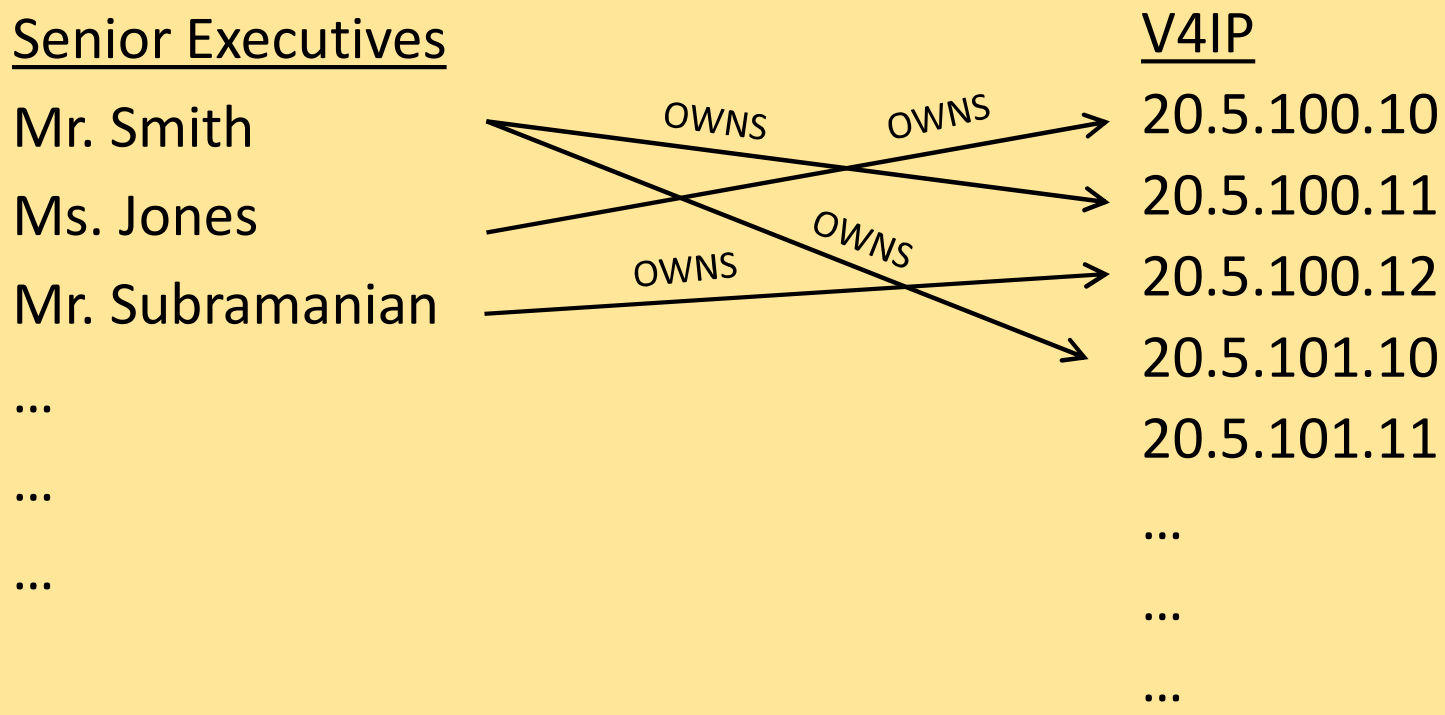
Subject	Predicate	Object
xyz123abc	Runs	DEVDB1

The object of one proposition can be the subject of another.

This ability to chain fact statements together indefinitely creates a powerful ability to merge data from many systems into one query-able graph – the Knowledge Map.

SingerLinks
Connecting The Dots...

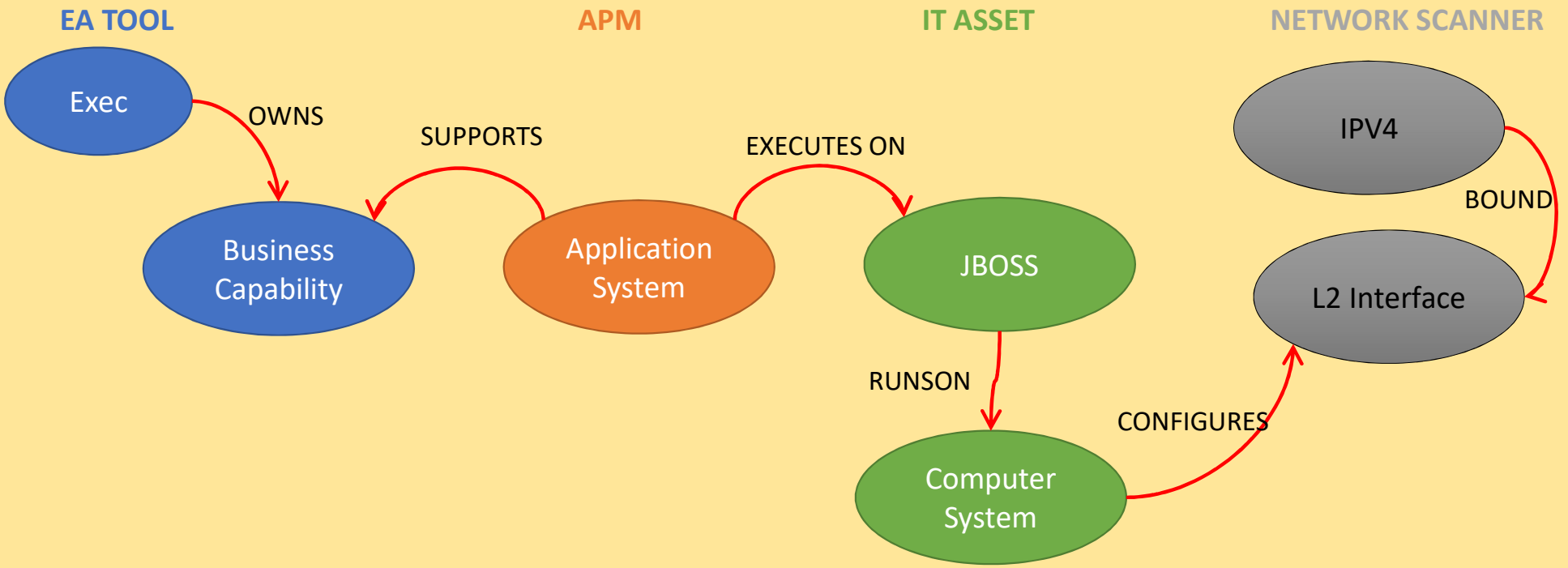
Interesting Or Boring?



It's the relationships that are interesting.

SingerLinks
Connecting The Dots...

The Path from Boring To Interesting



Answering difficult questions typically requires many hops through multiple domains of data.

SingerLinks
Connecting The Dots...

Conclusion – Why Graph Database for Metadata

- Typical Metadata Repository use cases are all graph problems:
  - Where used
  - Data Lineage
  - Impact Analysis
- Graph Database is a better representation
  - Data exists as a mesh of inter-related objects
  - Subject-Predicate-Object closer to natural language
- Graph Database provides better support for:
  - Visualization
  - Exploration
  - Network Analysis – algorithms

SingerLinks
Connecting The Dots...

And Finally…

- You can start NOW!
  - There are many high-impact use cases at your company waiting to be solved
  - Community edition Neo4j plus free/low cost front end tools provide a starting point.

- Shameless Self Promotion
  - Hit me up on LinkedIn
  - Read my blog and visit my webpage