Advanced Data Management (CSCI 490/680)

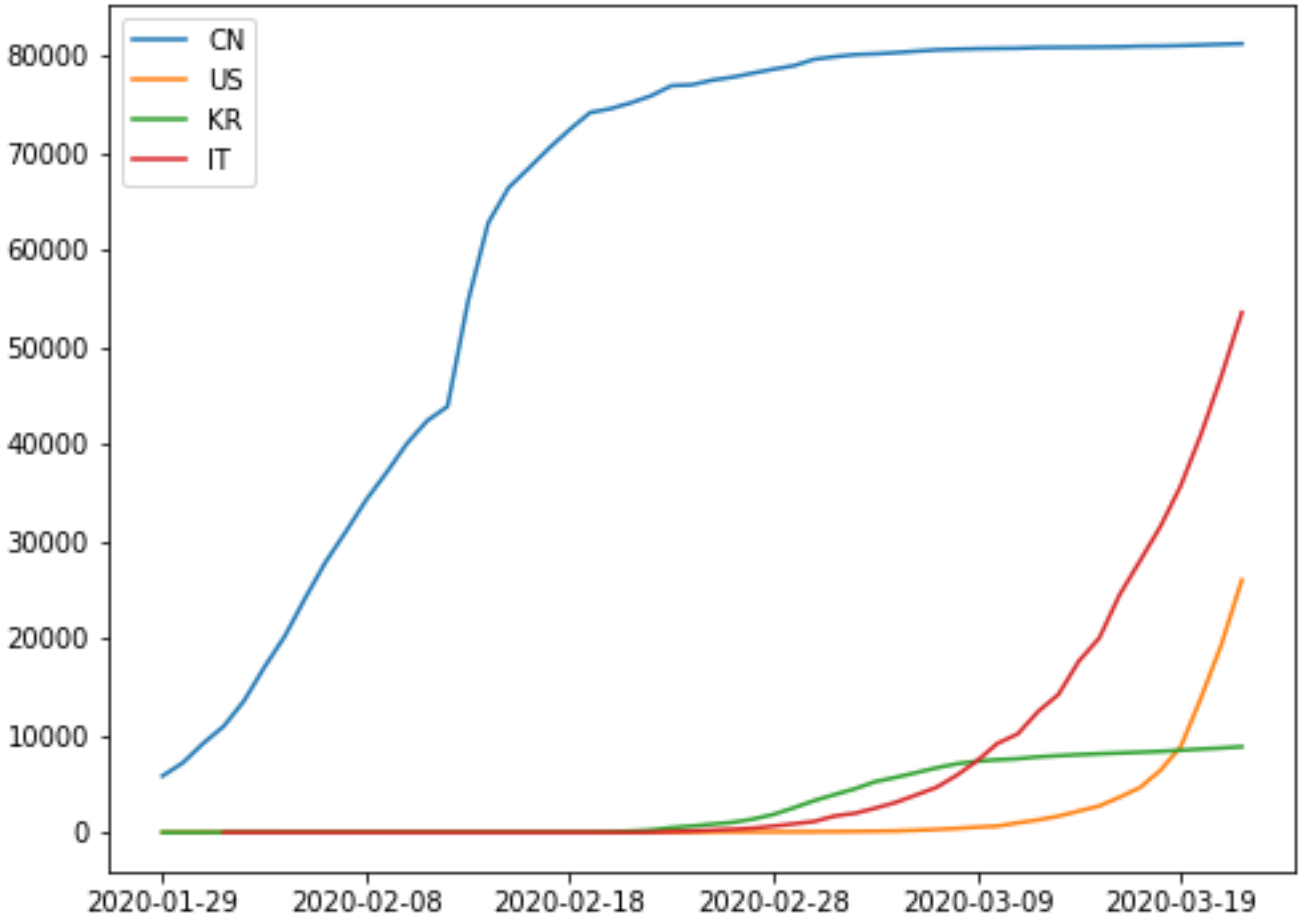
Graph Data

Dr. David Koop

Discussions

- Please post at least once on the discussion board in Blackboard with a question, answer, or discussion point about one of the lectures
- Feedback is useful—I hope the lectures are clear, but I am pretty sure there are still places where I can clarify things better
- You may also post questions about the assignment there if you believe they are relevant to all students

Assignment 4



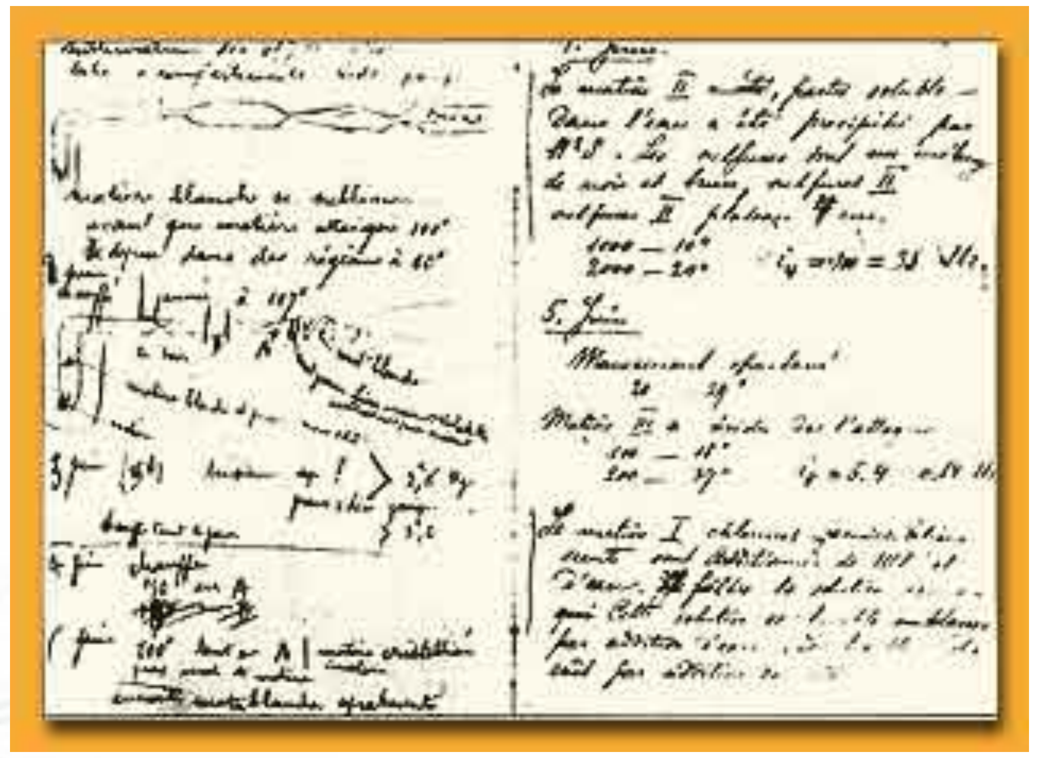
- COVID-19 data
- Data Integration
  - Population
  - Temperature
- Data Fusion:
  - Our World in Data
  - Johns Hopkins
  - Wikipedia
- Questions?

Test 2

- Online on Blackboard (webcourses.niu.edu)
- Thursday, April 9 from 3:30-4:45pm
- If you have conflicts, **let me know as soon as possible**
- Format:
  - Some multiple choice
  - More short answer/free response
- Focus on topics since the first test
- More details this week

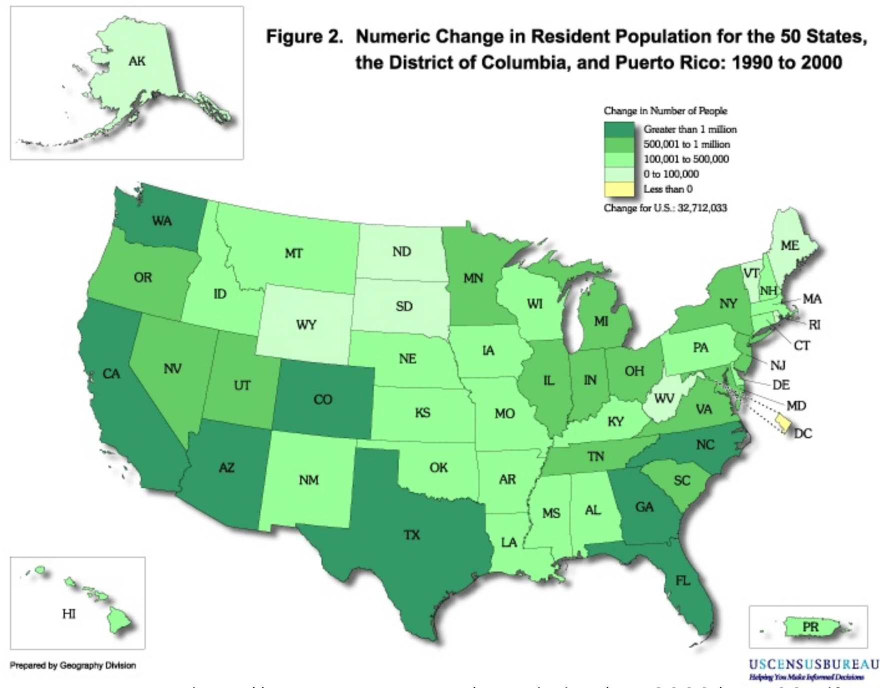
What is Data?

hudsonalpha.org

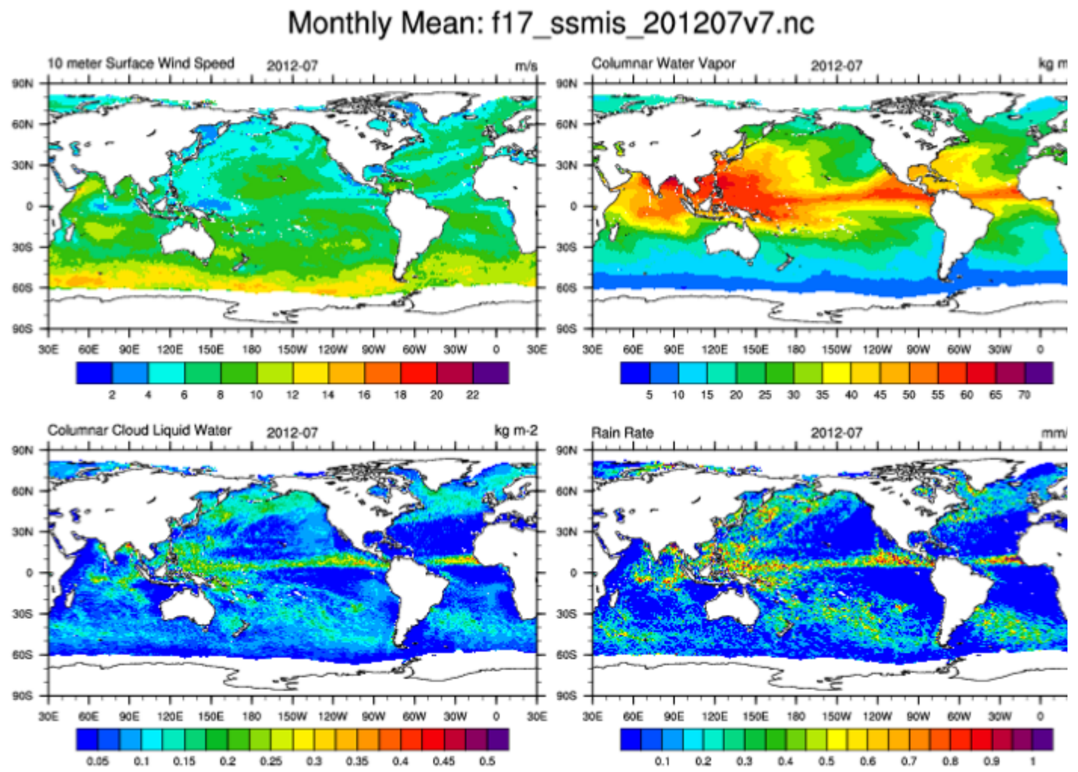


Marie Curie's notebook aip.org

Pisa Griffin



http://www.census.gov/population/cen2000/map02.gif



ncl.ucar.edu

Date:1/2.07.75 Place:Sakaltutan
Zafor
He will grow old in his present house; new house is for sons - 5 sons. Not sure they want to live in village. He will only build another if they want him to. eS came from Germany and did the plastering. He arranged the carpentry in Kayseri. Çok para gitti. (much money went) Has a tractor.

Date:July1980 Place:Sakaltutan
Zafor:
Household now Zafor and wife; Nazif Unal and wife and youngest son, still a boy. They run two dolmuß; one with a driver from Süleymanli. Goes in and out once a day. He gets 8,000 a month. Zafor then said, keskin de©il. (not sharp - i.e.? not profitable) I said he did very well on 8,000 TL with only two journeys a day. Nazif Unal has "bought" a Durak (dolmuß stop) from Belediye and works all day in Kayseri.

http://onlineqda.hud.ac.uk/Intro_QDA/Examples_of_Qualitative_Data.php

[C. Borgman]

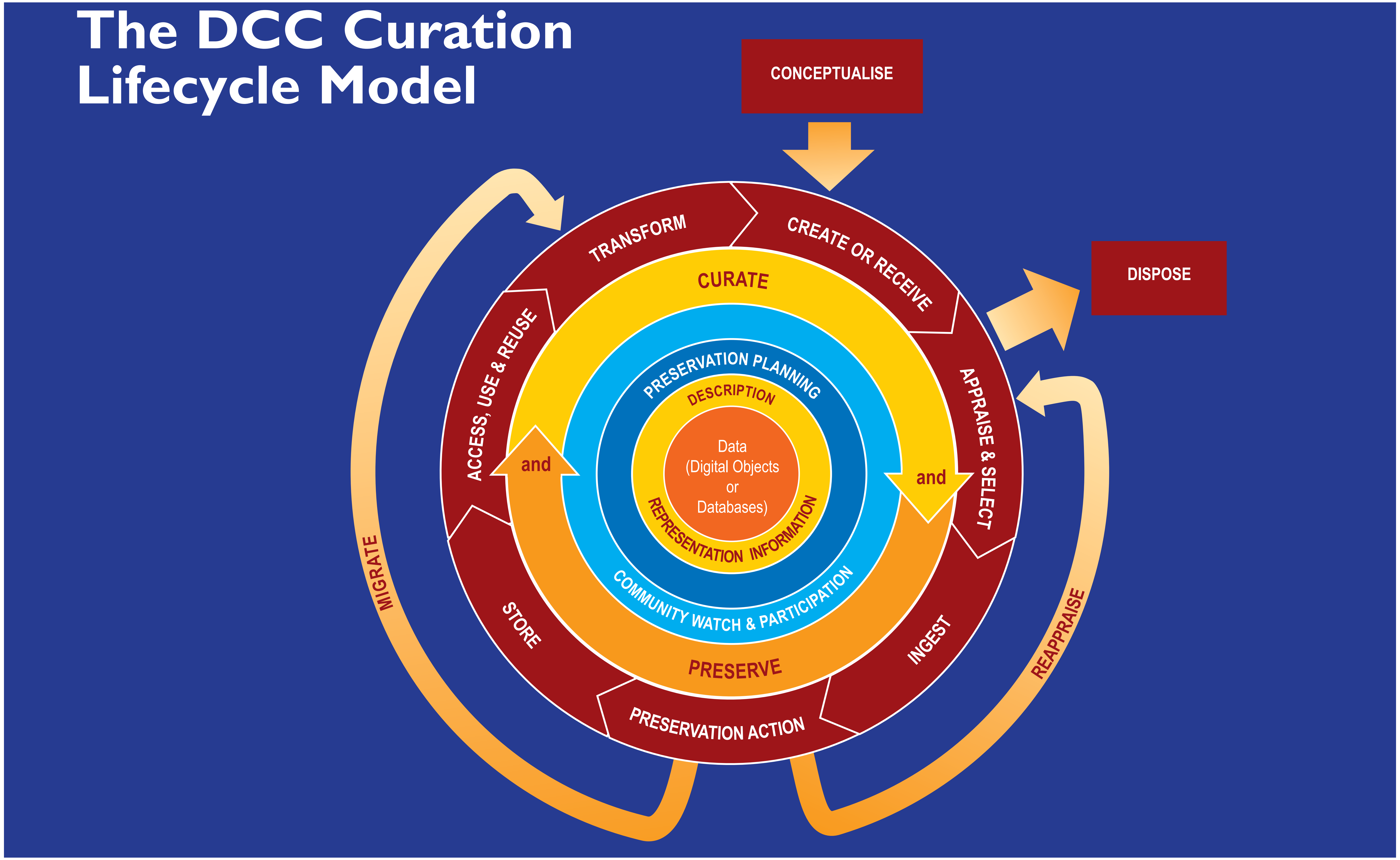
What is data?

- "Data are representations of observations, objects, or other entities used as evidence of phenomena for the purposes of research or scholarship."
  [C. L. Borgman]
- Data can be digital but can also be physical (e.g. sculptures)
- Semantics are important (e.g. temperature to engineer and biologist)
- Grey Data: surveys, student records—think about **privacy**

Sharing Data

- Required/encouraged by universities, funding agencies, publishers
- "Publications are arguments made by authors, and **data are the evidence** used to support the arguments." [C. L. Borgman]
- Questions:
  - How is data maintained? Who is responsible?
  - What is the process for curating data?
  - How long should data be kept?
  - How should data collection and curation be acknowledged?

Data Curation Lifecycle



[DCC]

Sequential Actions in Data Curation

- Conceptualize: Plan creation of data—capture method and storage options.
- Create or Receive: Create/receive data and make sure metadata exists
- Appraise and Select: Evaluate data and select for long-term curation and preservation
- Ingest: Transfer data to an archive, repository, data centre or other custodian
- Preservation Action: Data cleaning, validation (ensure that data remains authentic, reliable and usable)
- Store: Store the data in a secure manner adhering to relevant standards
  Access, Use and Reuse: Make sure is accessible to users and reusers
- Transform: Create new data from the original (migrate formats, subsets, etc.)

[DCC]

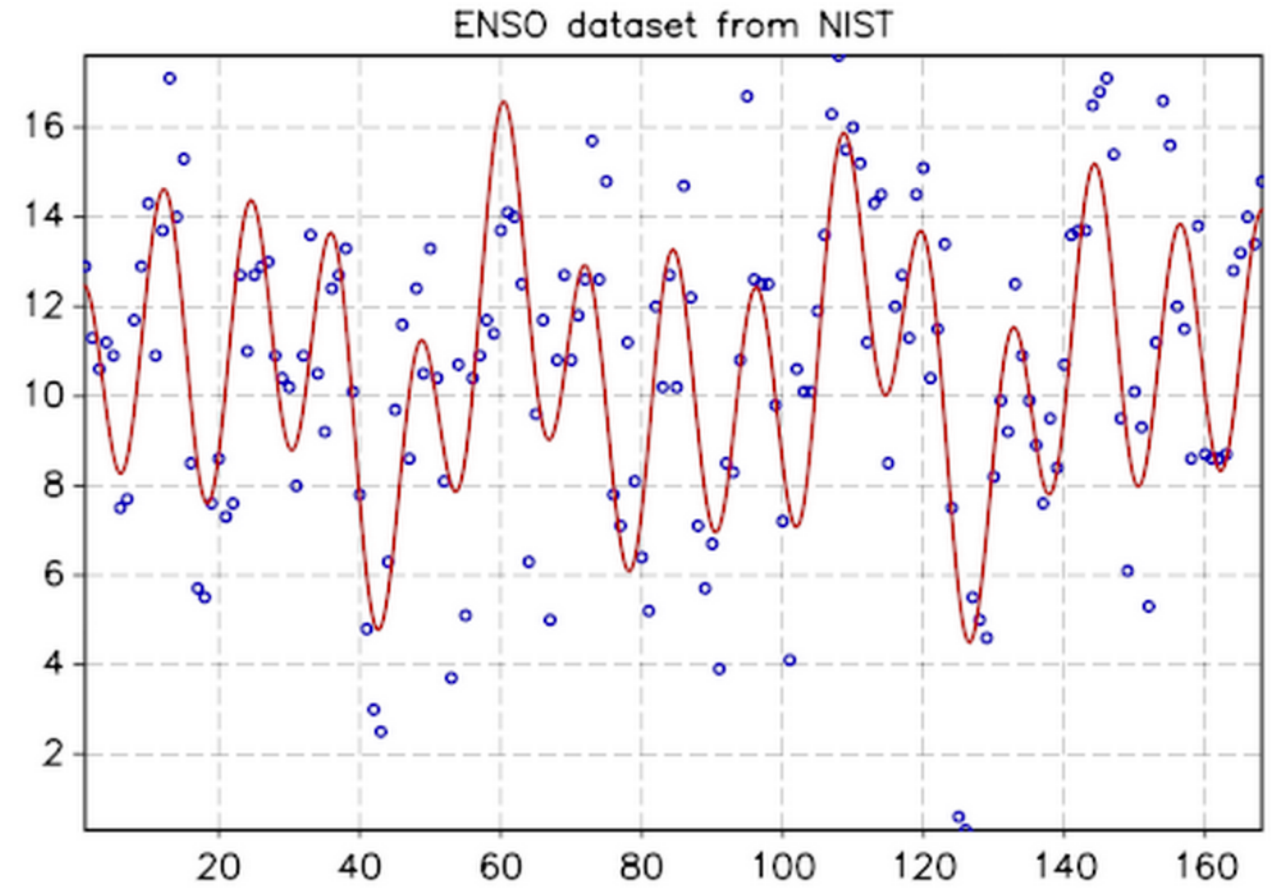
FAIR Principles

- **Findable**: Metadata and data should be easy to find for both humans and computers
- **Accessible**: Users need to know how data can be accessed, possibly including authentication and authorization
- **Interoperable**: Can be integrated with other data, and can interoperate with applications or workflows for analysis, storage, and processing
- **Reusable**: Optimize the reuse of data. Metadata and data should be well-described so they can be replicated and/or combined in different settings

[GO FAIR]

Findable: DataCite Workflow

1. Take a dataset



2. Describe it

- Title
- Authors
- Year
- Description
- And others…

3. Assign a DOI

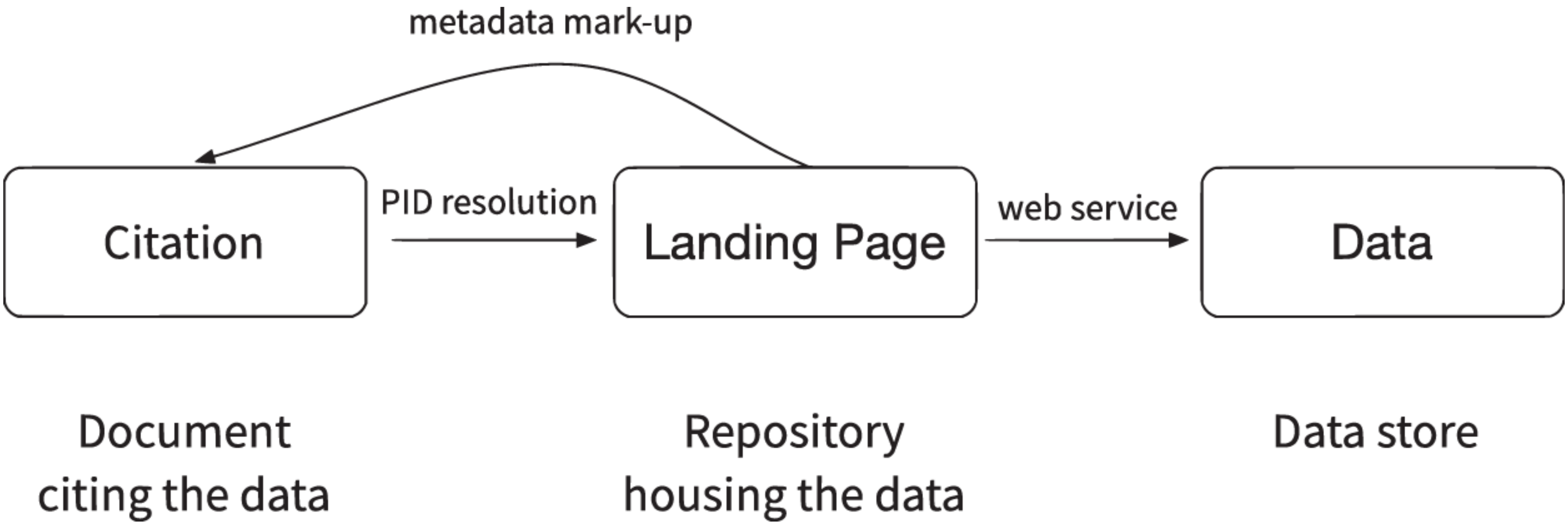
10.1234/exampledata

Proxy Prefix Suffix

https://doi.org/10.5438/n138-z3mk

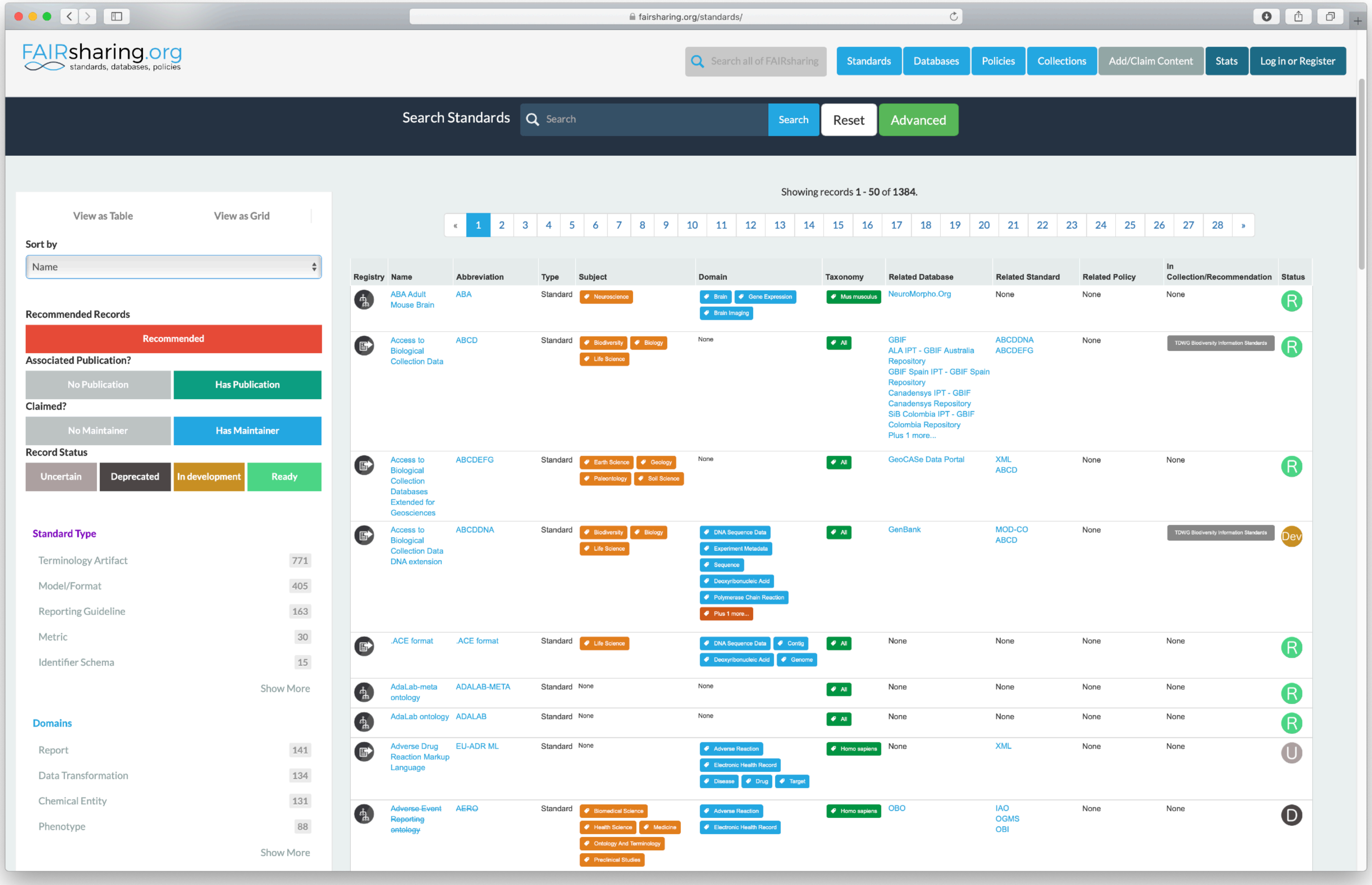
[DataCite]

Accessible: DOI to Landing Page with Metadata



[M. Fenner et al., 2019]

Interoperable: Standard vocabularies



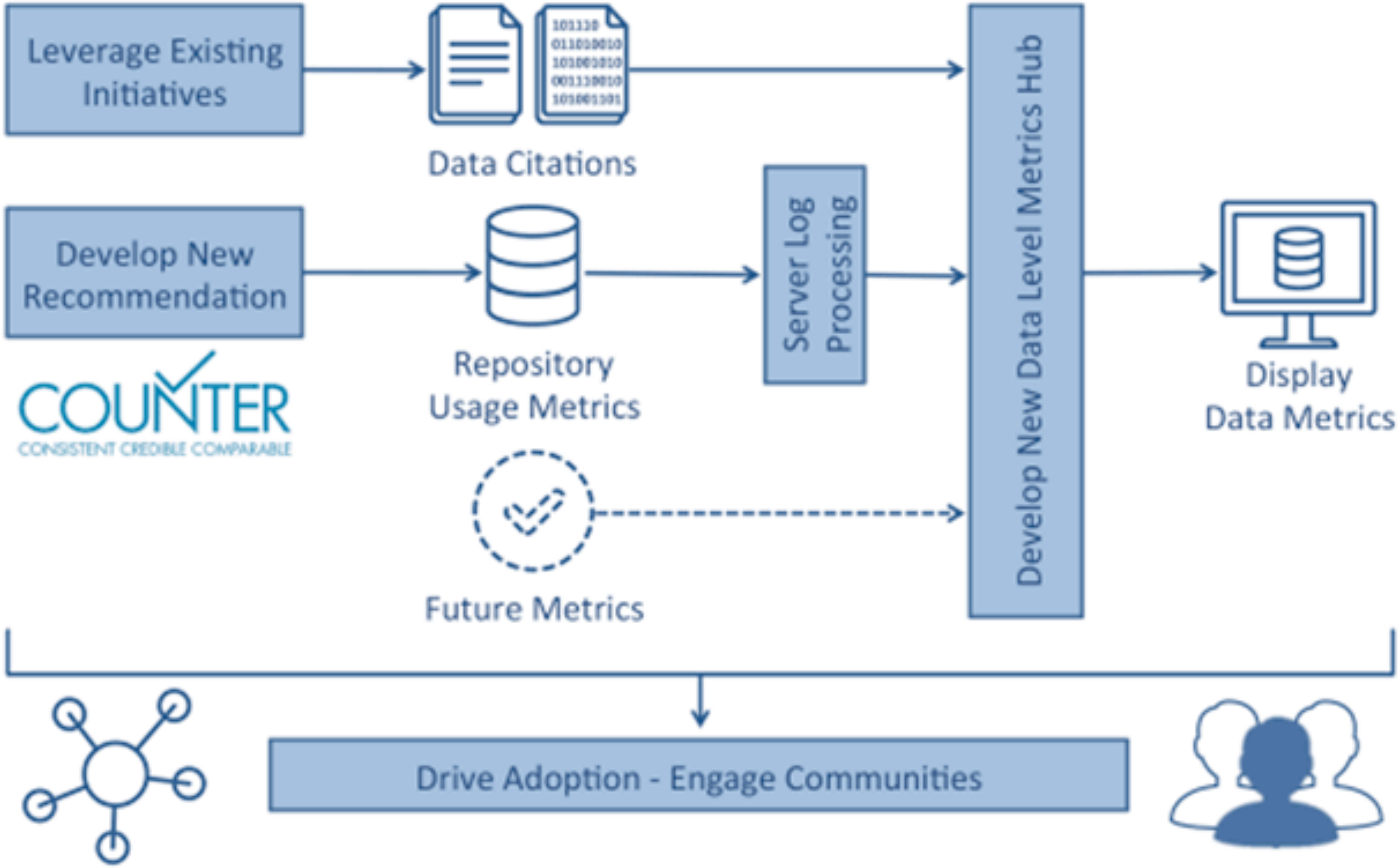
[fairsharing.org]

Reusable: Licensing

- Citation of a dataset is expected as a scholarly norm, not by law
- CC0:
  - "I hereby waive all copyright and related or neighboring rights together with all associated claims and causes of action with respect to this work to the extent possible under the law"
- CC BY: license, not a waiver as CC0
  - "You must give appropriate credit, provide a link to the license, and indicate if changes were made."
- Data Use Agreements (DUA): Used when data are restricted due to proprietary or privacy concerns.

[M. Crosas]

Reusable: Data Citation & Metrics



[H. Cousijn et al., 2019]

Specific Types of Data

Graphs: Social Networks

[P. Butler, 2010]

What is a Graph?

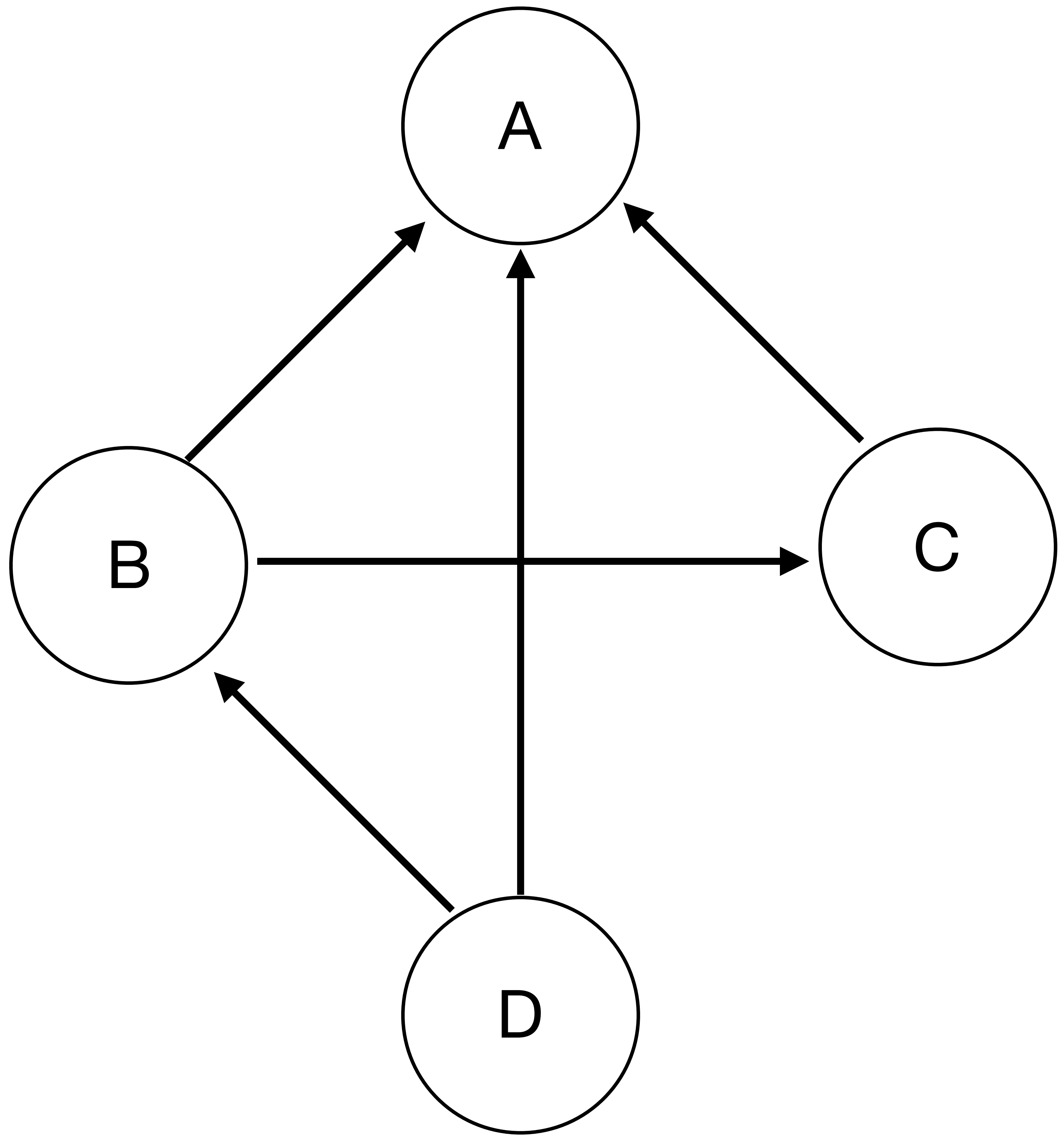
- An abstract representation of a set of objects where some pairs are connected by links.

Object (Vertex, Node)

Link (Edge, Arc, Relationship)

[M. De Marzi, 2012]

What is a Graph?

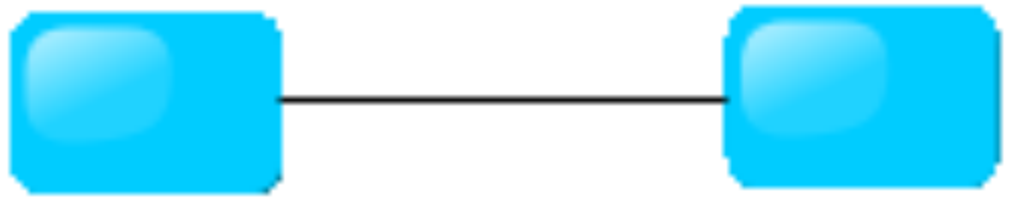


- In computing, a **graph** is an abstract **data structure** that represents set objects and their relationships as **vertices** and **edges/links**, and supports a number of graph-related **operations**
- Objects (nodes): `{A,B,C,D}`
- Relationships (edges): `{(D,B),(D,A),(B,C),(B,A),(C,A)}`
- Operation: shortest path from `D` to `A`

[K. Salama, 2016]

Different Kinds of Graphs

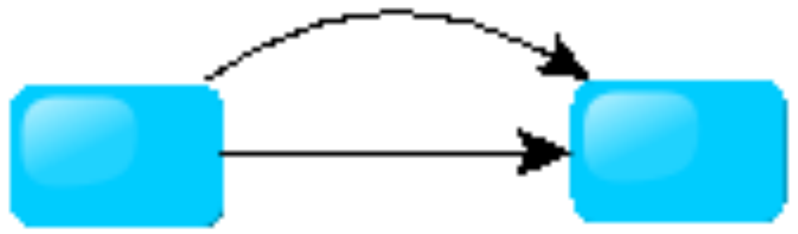
- Undirected Graph



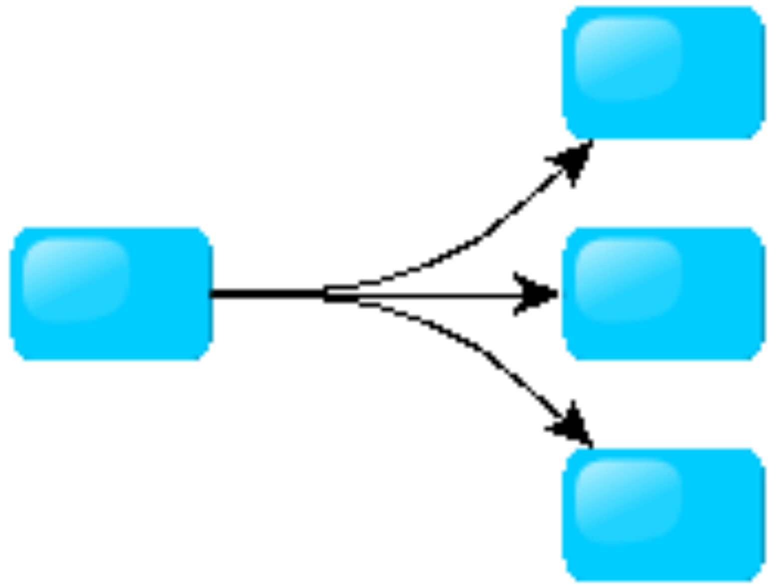
- Directed Graph

- Pseudo Graph

- Multi Graph



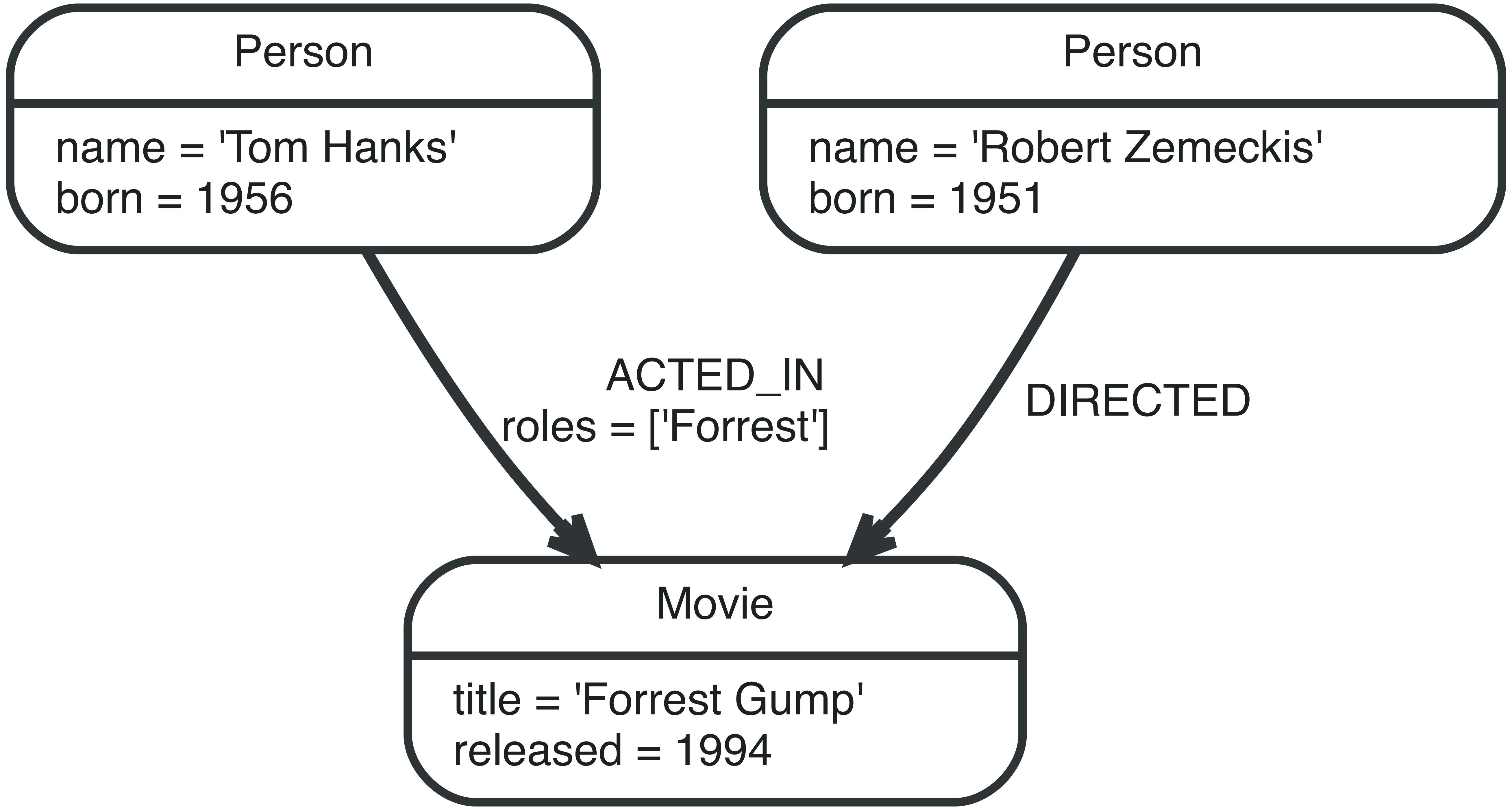
- Hyper Graph



[M. De Marzi, 2012]

Graphs with Properties

- Each vertex or edge may have properties associated with it
- May include identifiers or classes



[neo4j]

Types of Graph Operations

- Connectivity Operations:
  - number of vertices/edges, in- and out-degrees of vertices
  - histogram of degrees can be useful in comparing graphs
- Path Operations: cycles, reachability, shortest path, minimum spanning tree
- Community Operations: clusters (cohesion and separation)
- Centrality Operations: degree, vulnerability, PageRank
- Pattern Matching: subgraph isomorphism
  - can use properties
  - useful in fraud/threat detection, social network suggestions

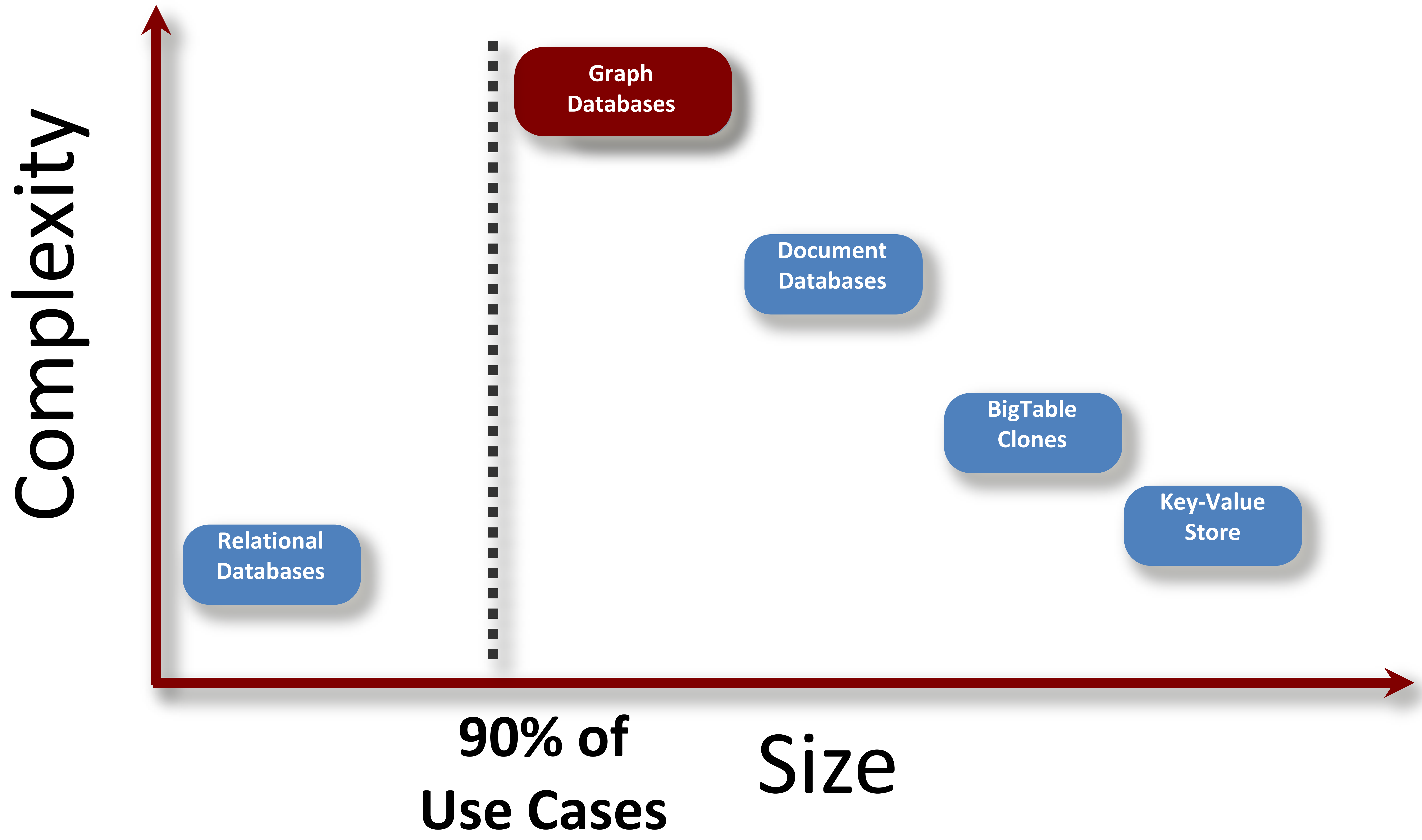
[K. Salama, 2016]

What is a Graph Database?

- A database with an explicit graph structure
- Each node knows its adjacent nodes
- As the number of nodes increases, the cost of a local step (or hop) remains the same
- Plus an Index for lookups

[M. De Marzi, 2012]

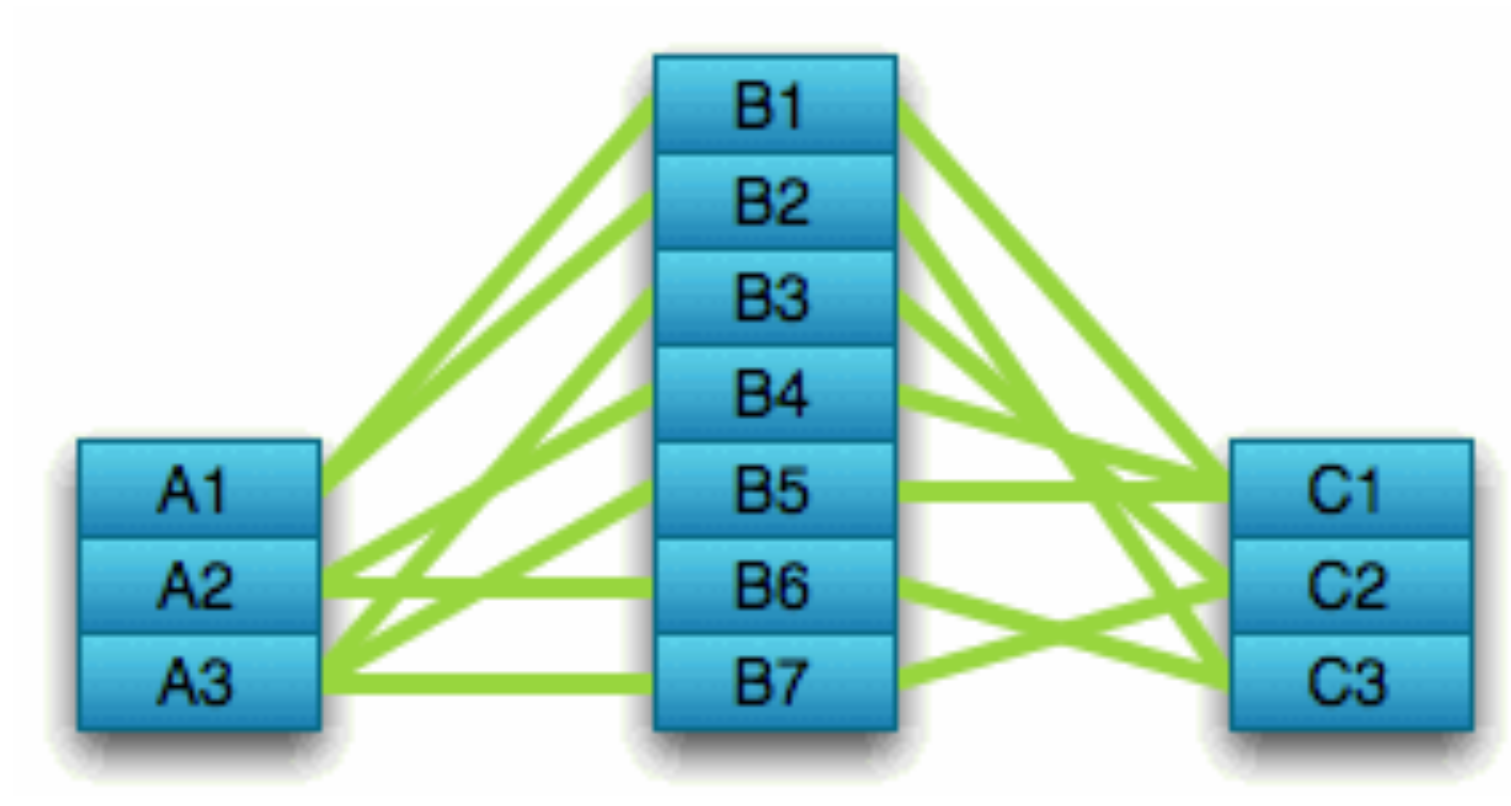
How do Graph Databases Compare?



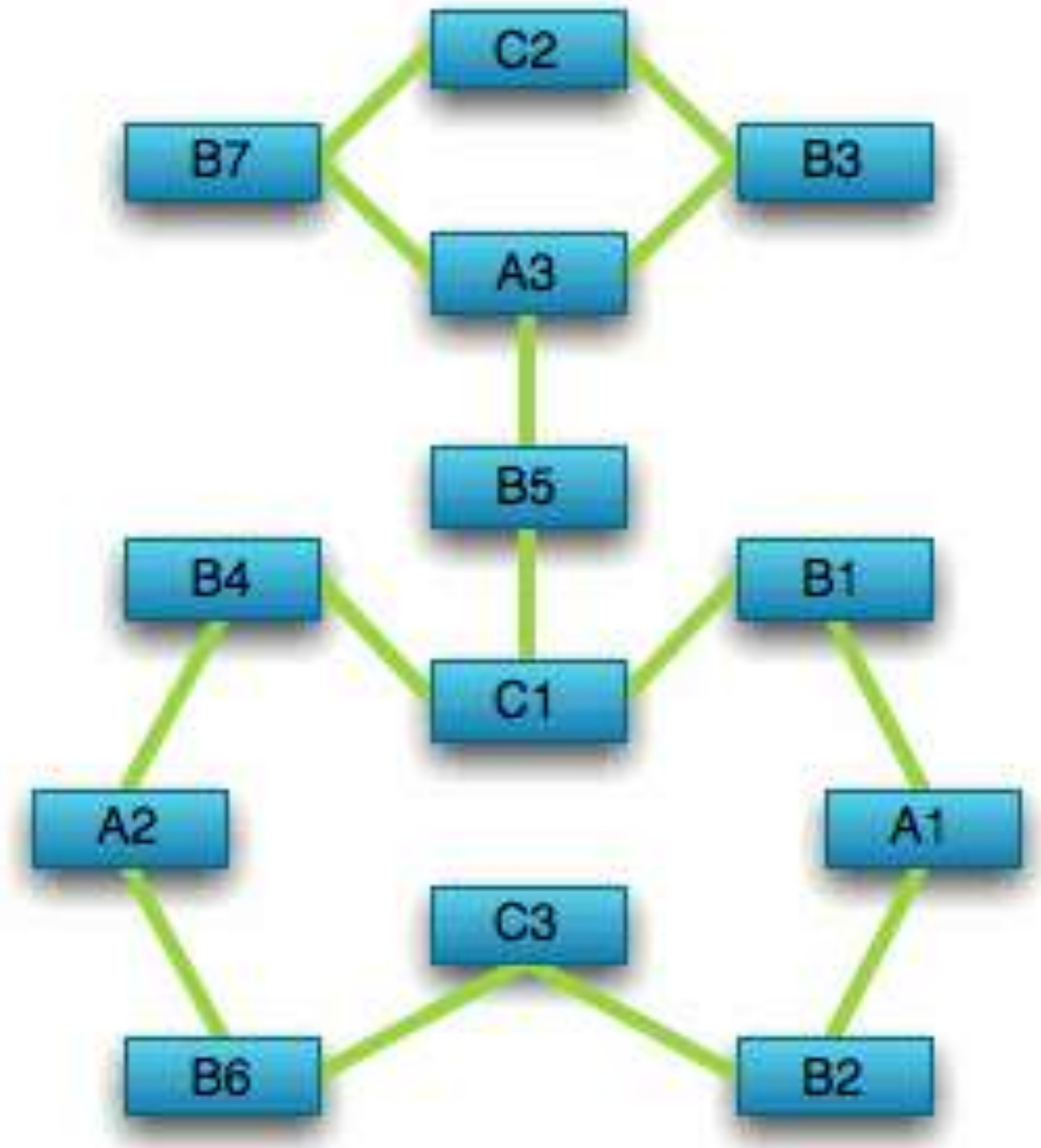
[M. De Marzi, 2012]

Graph Databases Compared to Relational Databases

Optimized for aggregation



Optimized for connections

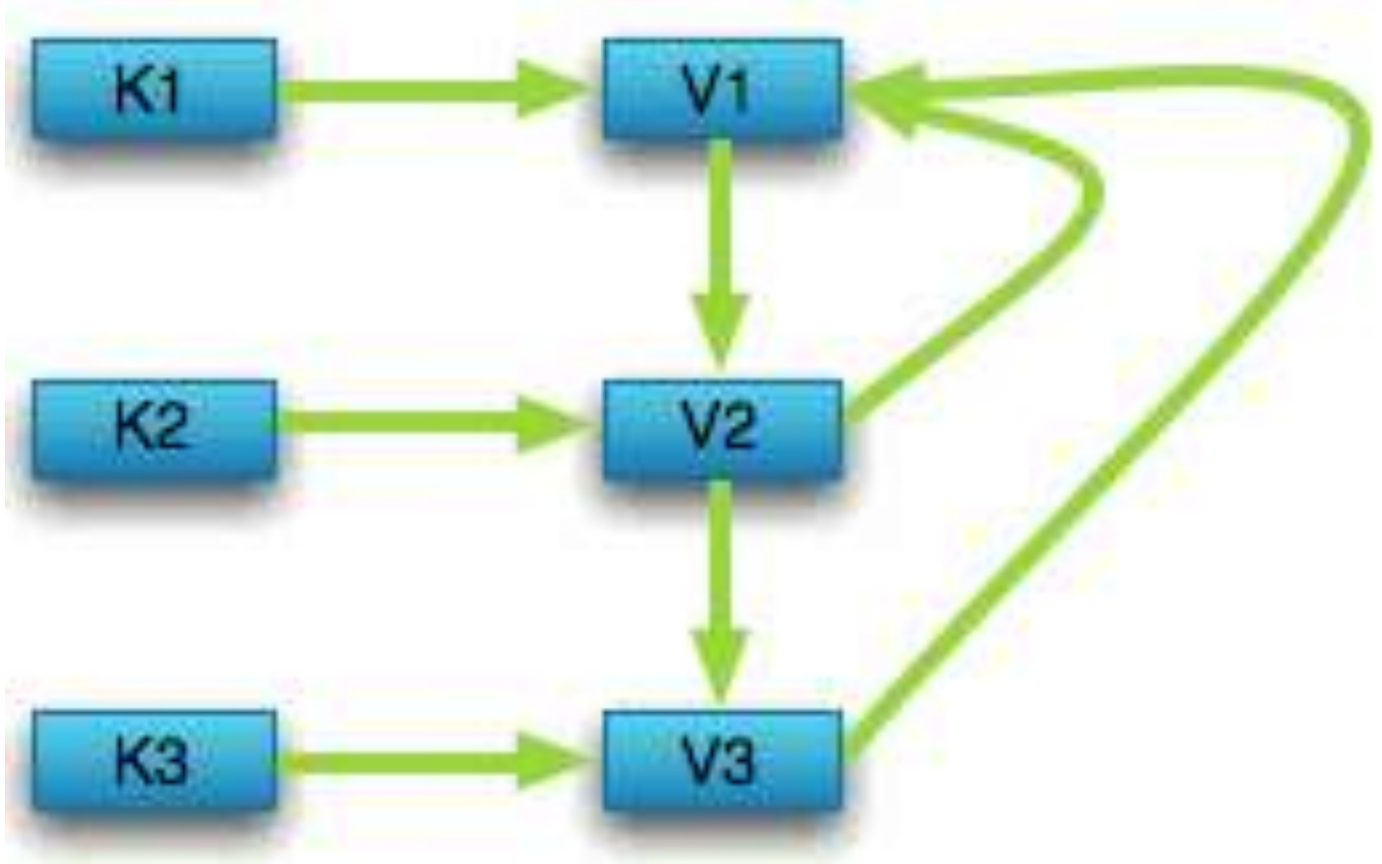


[M. De Marzi, 2012]

Graph Databases Compared to Key-Value Stores

Optimized for simple look-ups

Optimized for traversing connected data

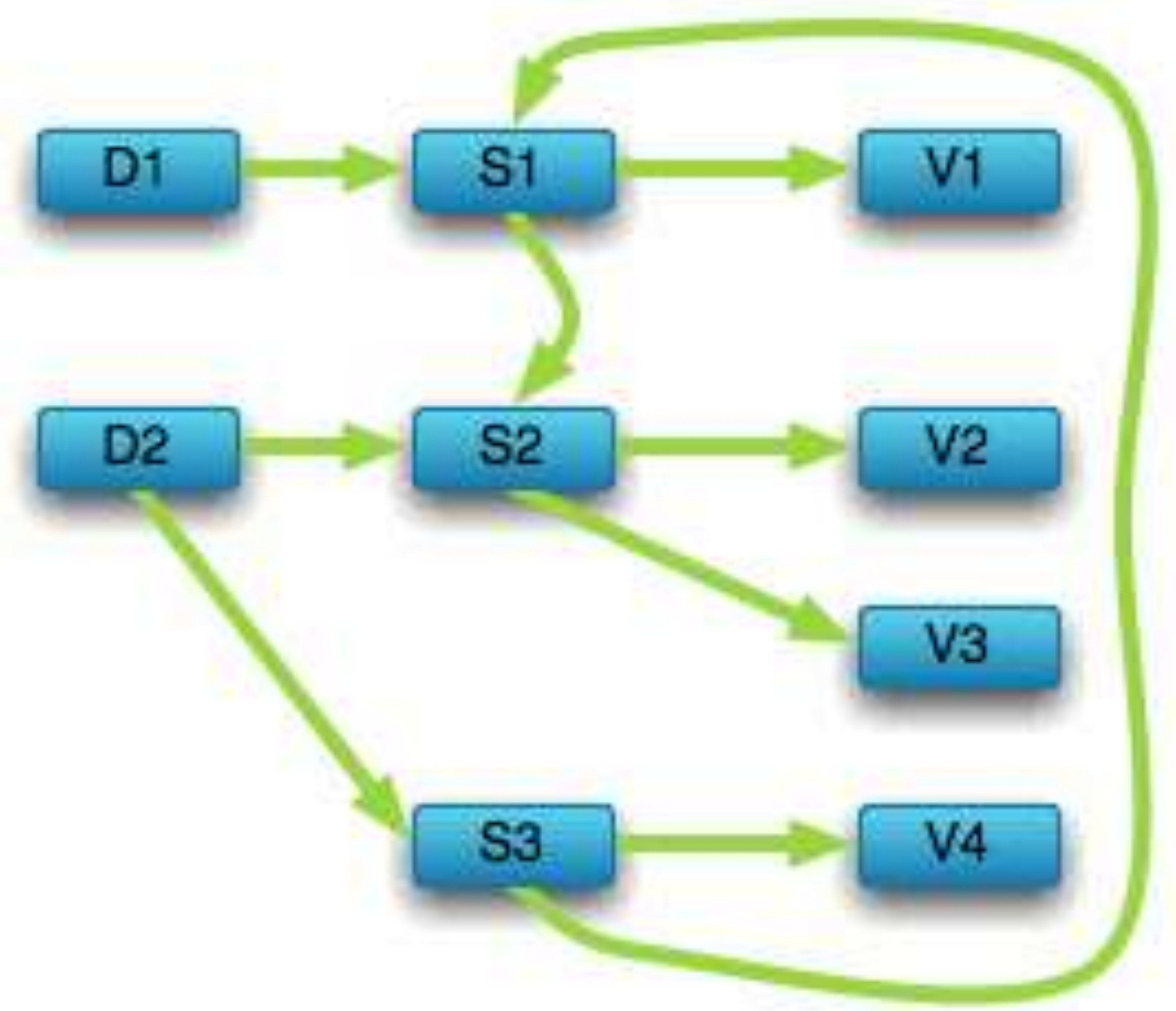


[M. De Marzi, 2012]

Graph Databases Compared to Document Stores

Optimized for "trees" of data

Optimized for seeing the forest and the trees, and the branches, and the trunks



[M. De Marzi, 2012]

Graph Databases

D. Lembo and R. Rosati

Why Graph Database Models?

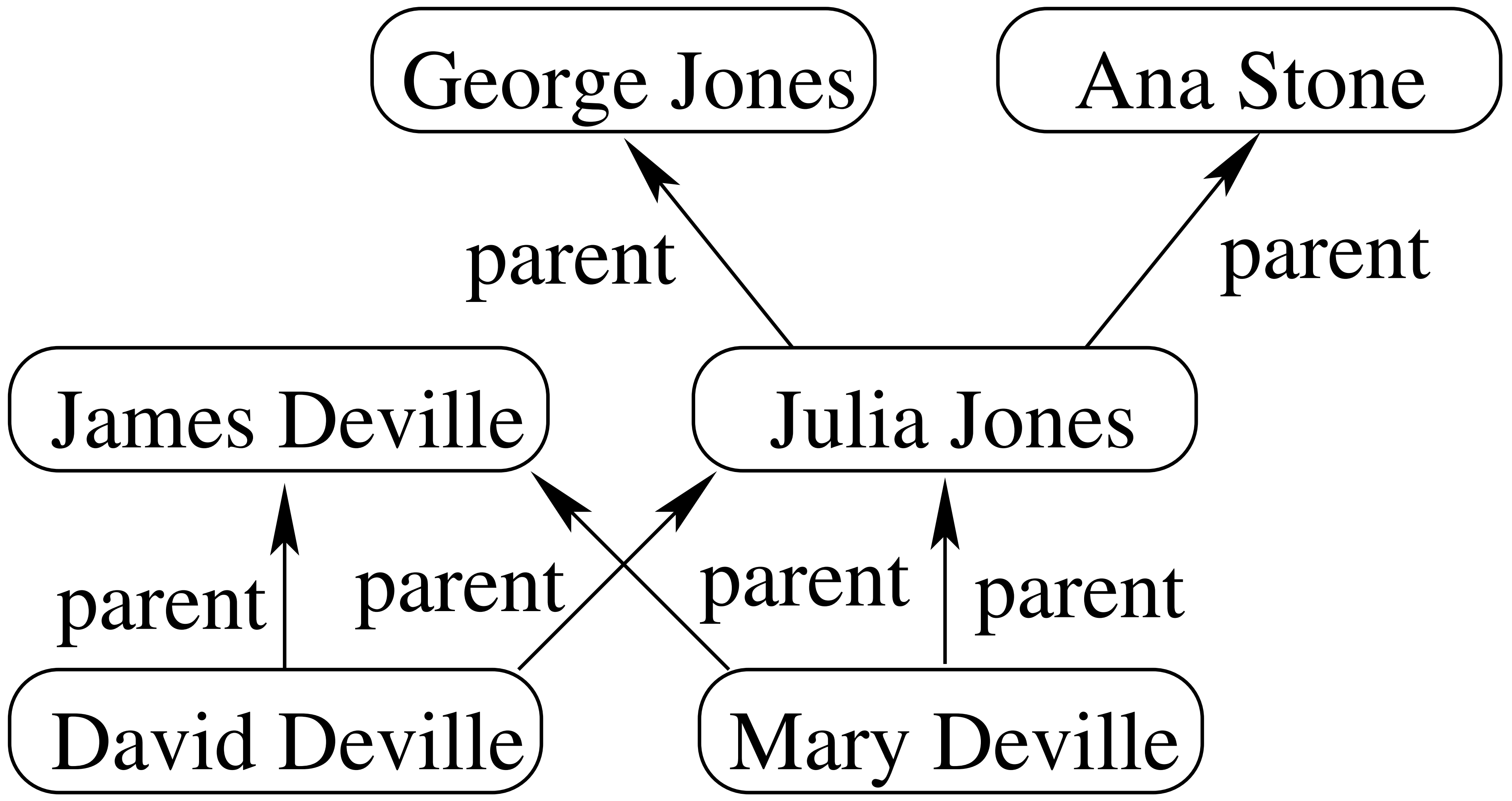
- Graphs has been long ago recognized as one of the most simple, natural and intuitive knowledge representation systems
- Graph data structures allow for a natural modeling when data has graph structure
- Queries can address direct and explicitly this graph structure
- Implementation-wise, graph databases may provide special graph storage structures, and take advantage of efficient graph algorithms available for implementing specific graph operations over the data

[R. Angles and C. Gutierrez, 2017]

Relational Model

NAME	LASTNAME
George	Jones
Ana	Stone
Julia	Jones
James	Deville
David	Deville
Mary	Deville

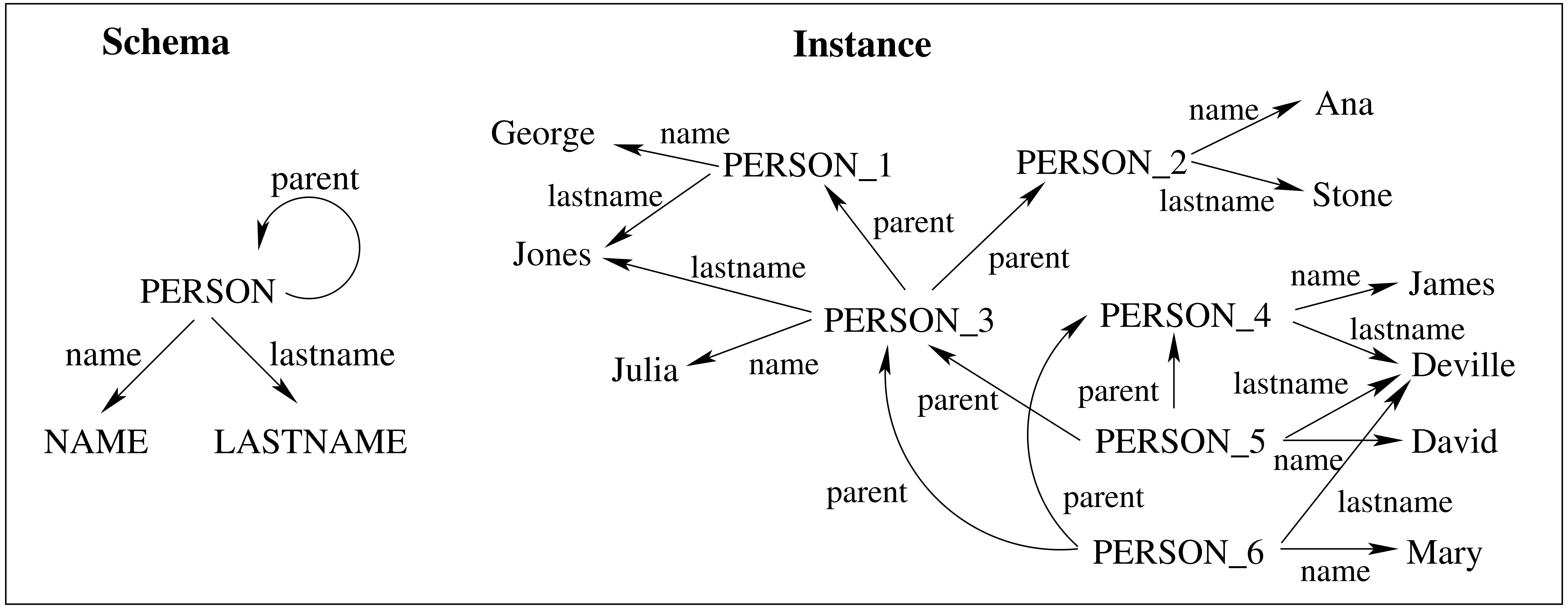
PERSON	PARENT
Julia	George
Julia	Ana
David	James
David	Julia
Mary	James
Mary	Julia



[R. Angles and C. Gutierrez, 2017]

Basic Labeled Model (Gram)

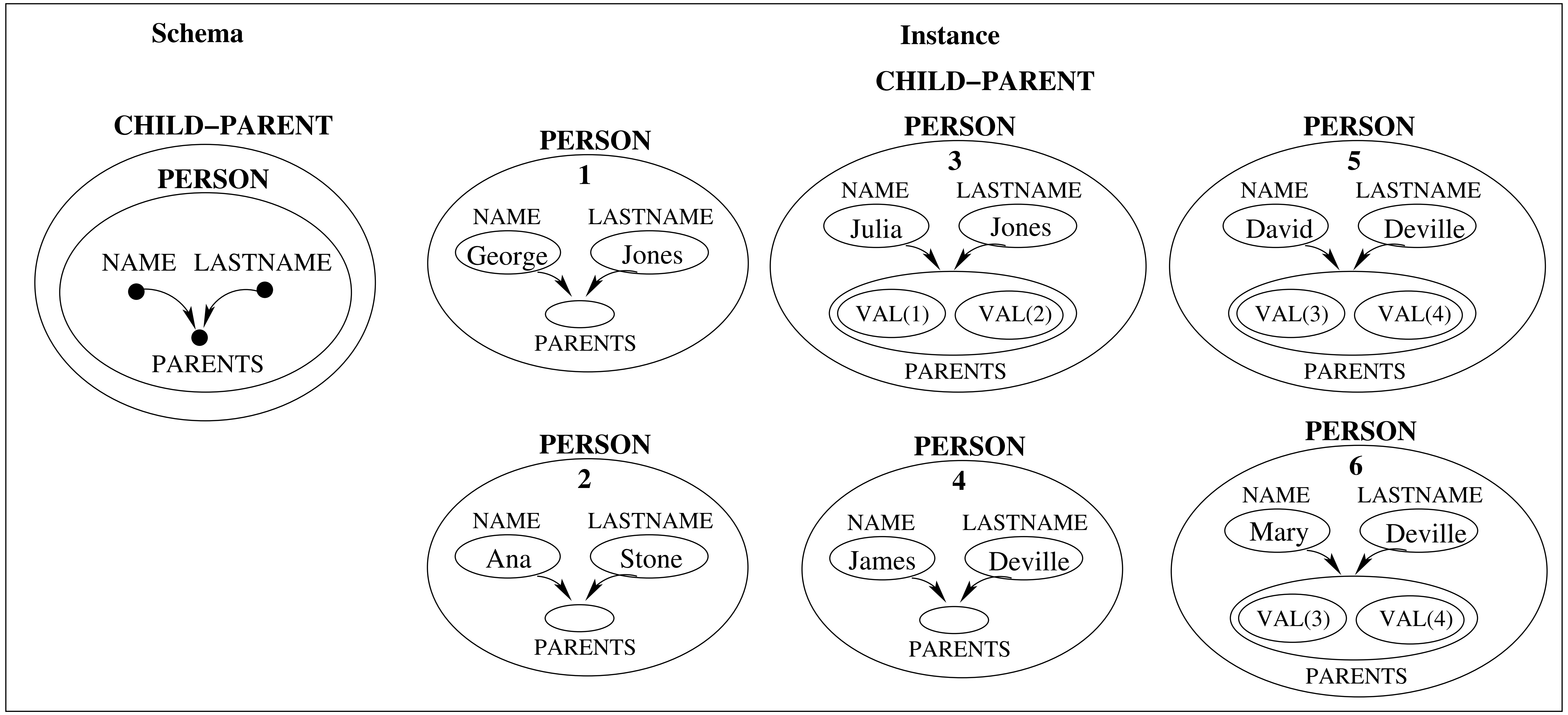
- Directed graph with nodes and edges labeled by some vocabulary
- Gram is a directed labeled multigraph
  - Each node is labeled with a symbol called a **type**
  - Each edge has assigned a label representing a **relation** between types



[R. Angles and C. Gutierrez, 2017]

Hypergraph Model (Groovy)

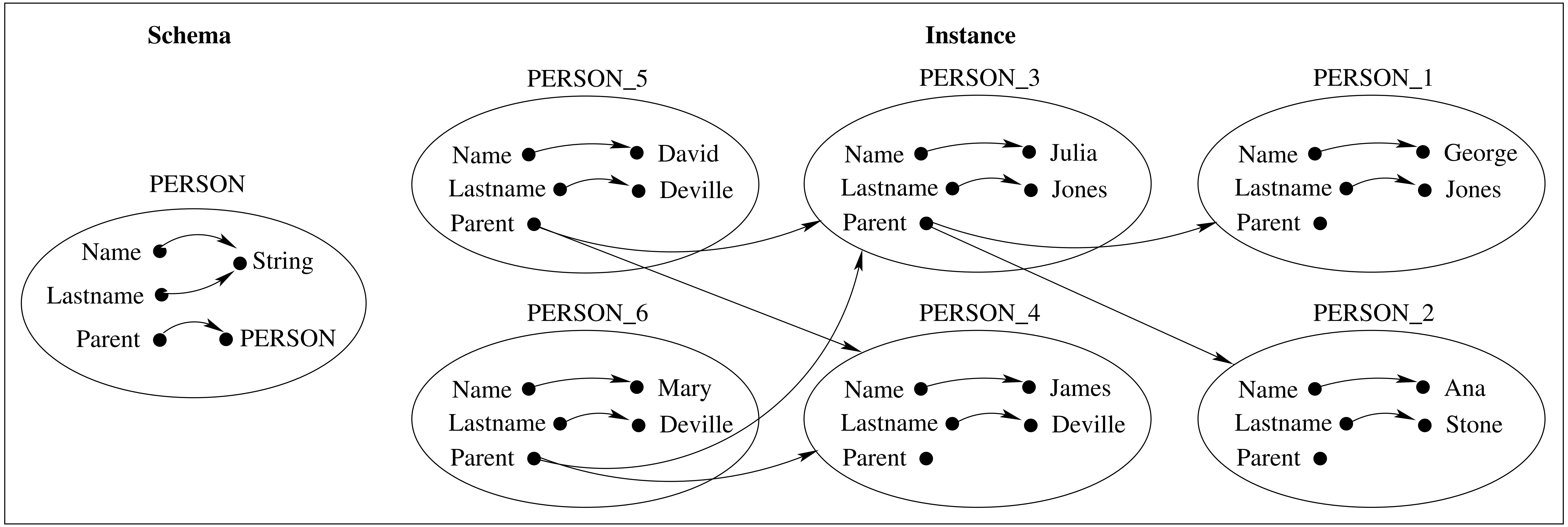
- Notion of edge is extended to **hyperedge**, which relates an arbitrary set of nodes
- Hypergraphs allow the definition of complex objects (undirected), functional dependencies (directed), object-ID and (multiple) structural inheritance



[R. Angles and C. Gutierrez, 2017]

Hypernode Model

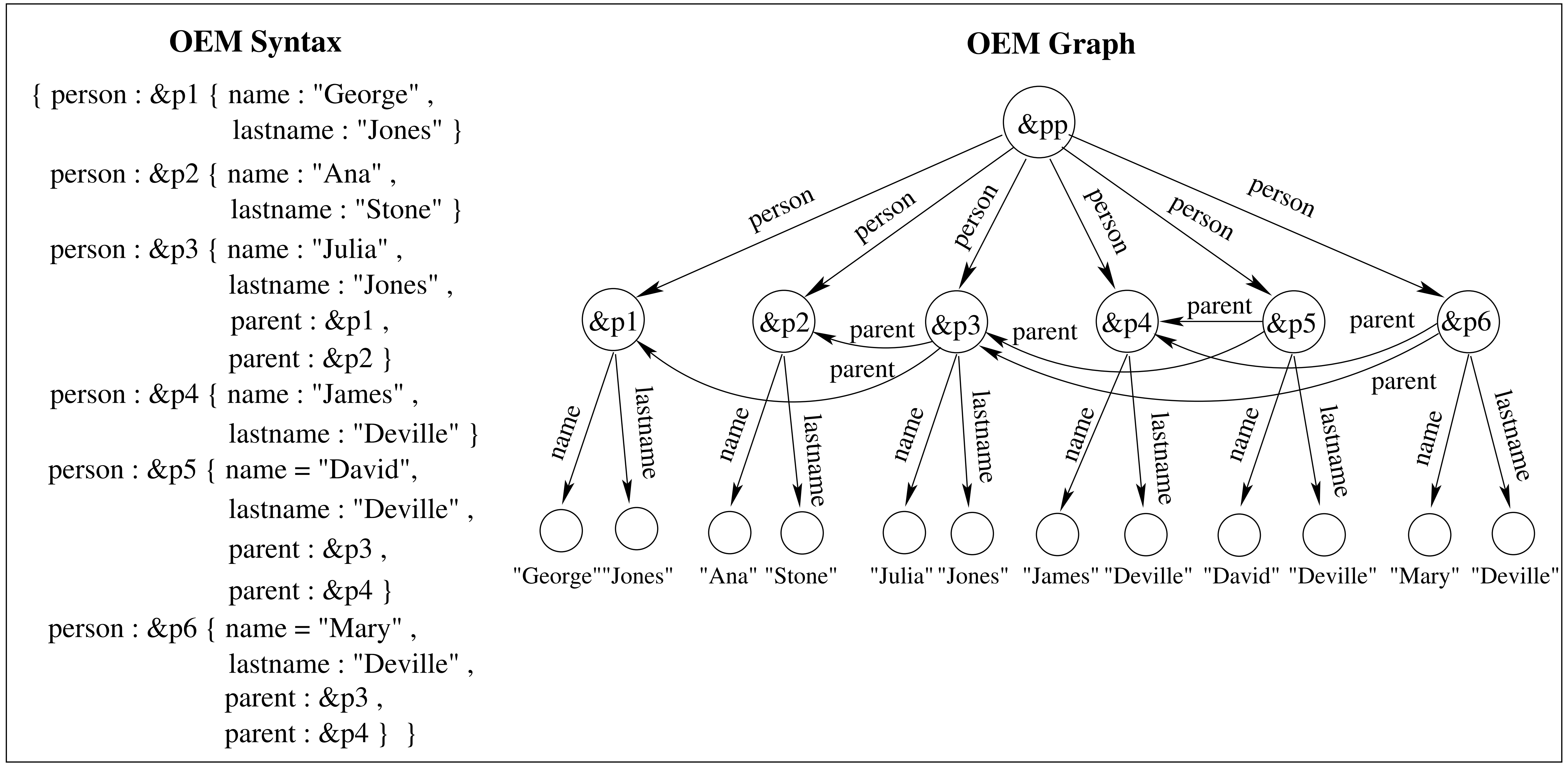
- Hypernode is a directed graph whose nodes can themselves be graphs (or hypernodes), allowing **nesting** of graphs
- **Encapsulates** information



[R. Angles and C. Gutierrez, 2017]

Semistructured (Tree) Model: (OEM Graph)

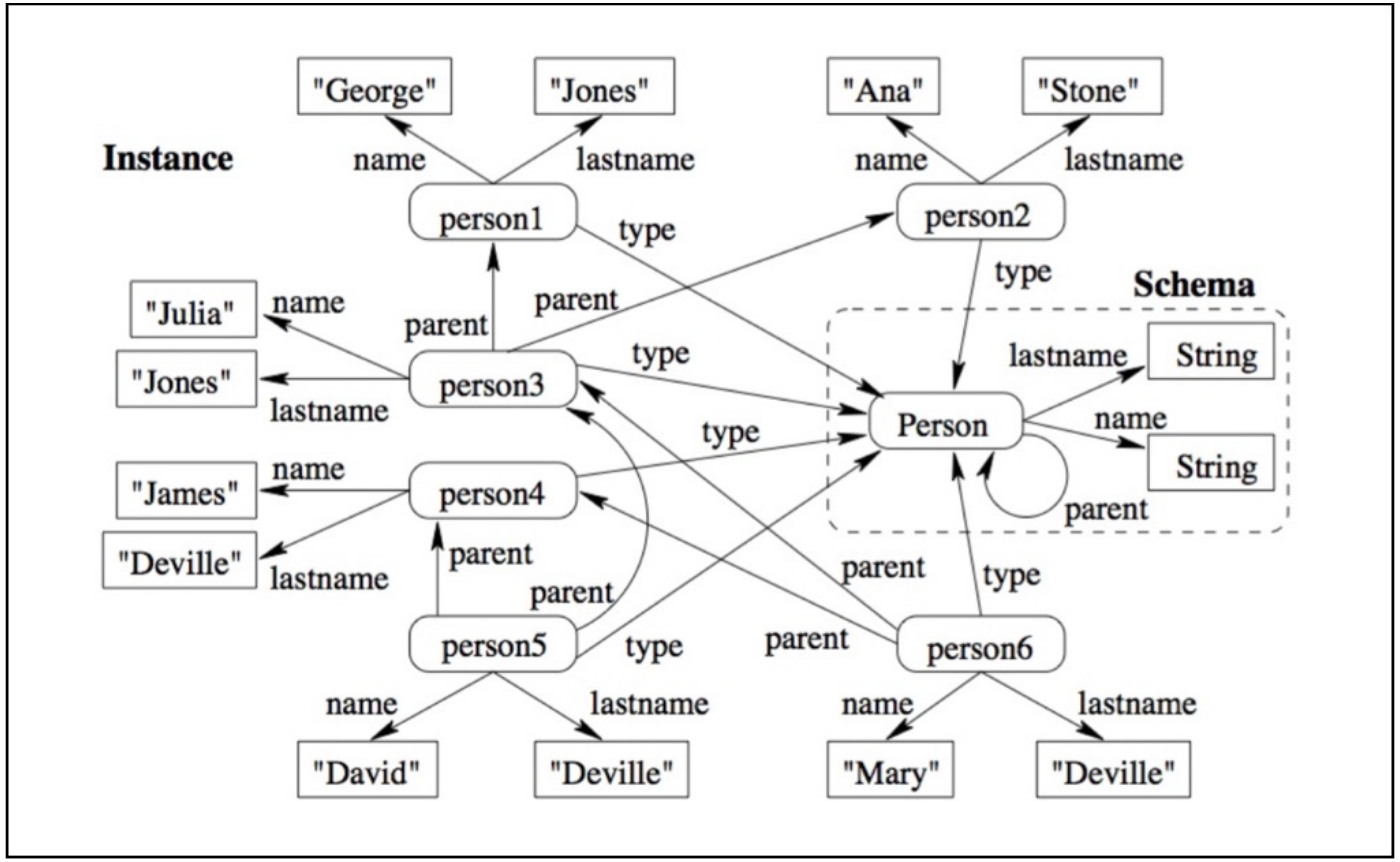
- "Self-describing" data like JSON and XML
- OEM uses pointers to data in the tree



[R. Angles and C. Gutierrez, 2017]

RDF (Triple) Model

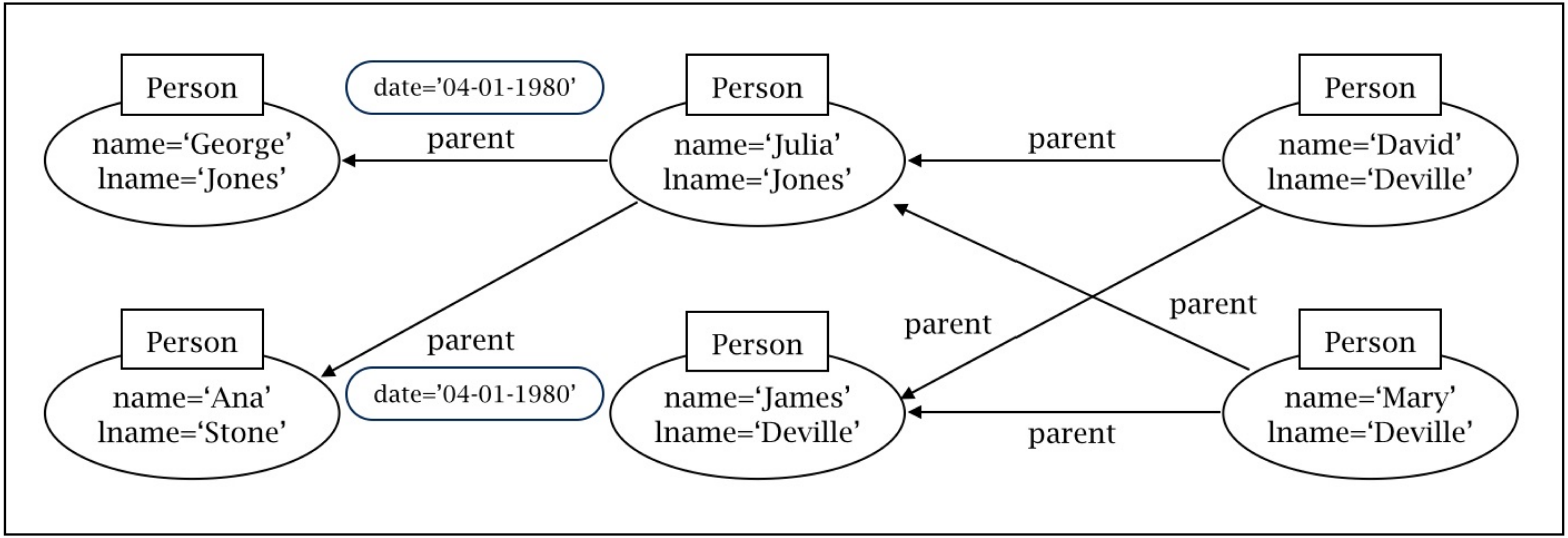
- Interconnect resources in an extensible way using graph-like structure for data
- Schema and instance are **mixed** together
- SPAQL to query
- Semantic web



[R. Angles and C. Gutierrez, 2017]

Property Graph Model (Cypher in neo4j)

- Directed, labelled, attributed multigraph
- Properties are **key/value pairs** that represent metadata for nodes and edges



[R. Angles and C. Gutierrez, 2017]

Types of Graph Queries

- Adjacency queries (neighbors or neighborhoods)
- Pattern matching queries (related to graph mining)
  - Graph patterns with structural extension or restrictions
  - Complex graph patterns
  - Semantic matching
  - Inexact matching
  - Approximate matching
- Reachability queries (connectivity)

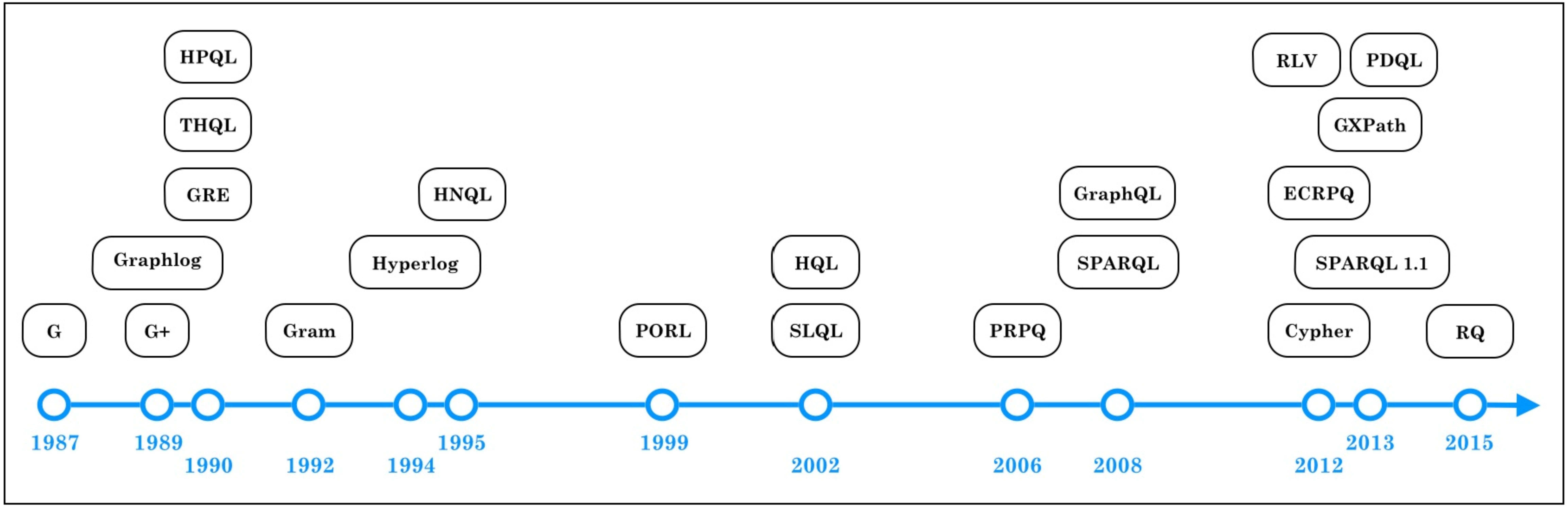
[R. Angles and C. Gutierrez, 2017]

Types of Graph Queries (continued)

- Analytical queries
  - Summarization queries
  - Complex analytical queries (PageRank, characteristic path length, connected components, community detection, clustering coefficient)

[R. Angles and C. Gutierrez, 2017]

Graph Query Languages



[R. Angles and C. Gutierrez, 2017]

Cypher

- Implemented by neo4j system
- Expresses reachability queries via path expressions
  - `p = (a)-[:knows*]->(b)`: nodes from `a` to `b` following `knows` edges
- ```
  START x=node:person(name="John")
  MATCH (x)-[:friend]->(y)
  RETURN y.name
  ```

[R. Angles and C. Gutierrez, 2017]
  ```

SPARQL (RDF)

- Uses SELECT-FROM-WHERE pattern like SQL
-
```
SELECT ?N
FROM <http://example.org/data.rdf>
WHERE { ?X rdf:type voc:Person . ?X voc:name ?N }
```

[R. Angles and C. Gutierrez, 2017]

Comparing Graph Database Systems: Features

Data Storage

Graph Database	Main memory	External memory	Backend Storage	Indexes
AllegroGraph	•	•		•
DEX	•	•		•
Filament	•		•	
G-Store		•		
HyperGraphDB	•	•	•	•
InfiniteGraph		•		•
Neo4j	•	•		•
Sones	•			•
vertexDB		•	•	

Operations/Manipulation

Graph Database	Data Definition Language	Data Manipulat. Language	Query Language	API	GUI
AllegroGraph	•	•	•	•	•
DEX				•	
Filament				•	
G-Store	•		•	•	
HyperGraphDB				•	
InfiniteGraph				•	
Neo4j				•	
Sones	•	•	•	•	•
vertexDB				•	

[R. Angles, 2012]

Comparing Graph Database Systems: Representation

Graph Data Structures

	Graphs				Nodes		Edges		
Graph Database	Simple graphs	Hypergraphs	Nested graphs	Attributed graphs	Node labeled	Node attribution	Directed	Edge labeled	Edge attribution
AllegroGraph	●				●		●	●	
DEX				●	●	●	●	●	●
Filament	●				●		●	●	
G-Store	●				●		●	●	
HyperGraphDB		●			●		●	●	
InfiniteGraph				●	●	●	●	●	●
Neo4j				●	●	●	●	●	●
Sones		●		●	●	●	●	●	●
vertexDB	●				●		●	●	

Entites & Relations

	Schema			Instance					
Graph Database	Node types	Property types	Relation types	Object nodes	Value nodes	Complex nodes	Object relations	Simple relations	Complex relations
AllegroGraph					●			●	
DEX	●		●	●	●		●	●	
Filament					●			●	
G-Store					●			●	
HyperGraphDB	●		●		●			●	●
InfiniteGraph	●		●	●	●		●	●	
Neo4j				●	●		●	●	
Sones					●			●	●
vertexDB					●			●	

[R. Angles, 2012]

Comparing Graph Database Systems: Queries

Query Support

	Type			Use		
Graph Database	Query Lang.	API	Graphical Q. L.	Retrieval	Reasoning	Analysis
AllegroGraph	○	●	●	●	●	●
DEX		●		●		●
Filament		●		●		
G-Store	●			●		
HyperGraphDB		●		●		
InfiniteGraph		●		●		
Neo4j	○	●		●		
Sones	●		●	●		●
vertexDB		●		●		

Types of Queries

	Adjacency		Reachability				
Graph Database	Node/edge adjacency	k-neighborhood	Fixed-length paths	Regular simple paths	Shortest path	Pattern matching	Summarization
Allegro	●		●			●	
DEX	●		●	●	●	●	
Filament	●		●			●	
G-Store	●		●	●	●	●	
HyperGraph	●					●	
Infinite	●		●	●	●	●	
Neo4j	●		●	●	●	●	
Sones	●					●	
vertexDB	●		●	●		●	

[R. Angles, 2012]

Reminder: Discussions

- Please post at least once this week on the discussion board in Blackboard with a question, answer, or discussion point about one of the lectures
- Feedback is useful—I hope the lectures are clear, but I am pretty sure there are still places where I can clarify things better
- You may also post questions about the assignment there if you believe they are relevant to all students