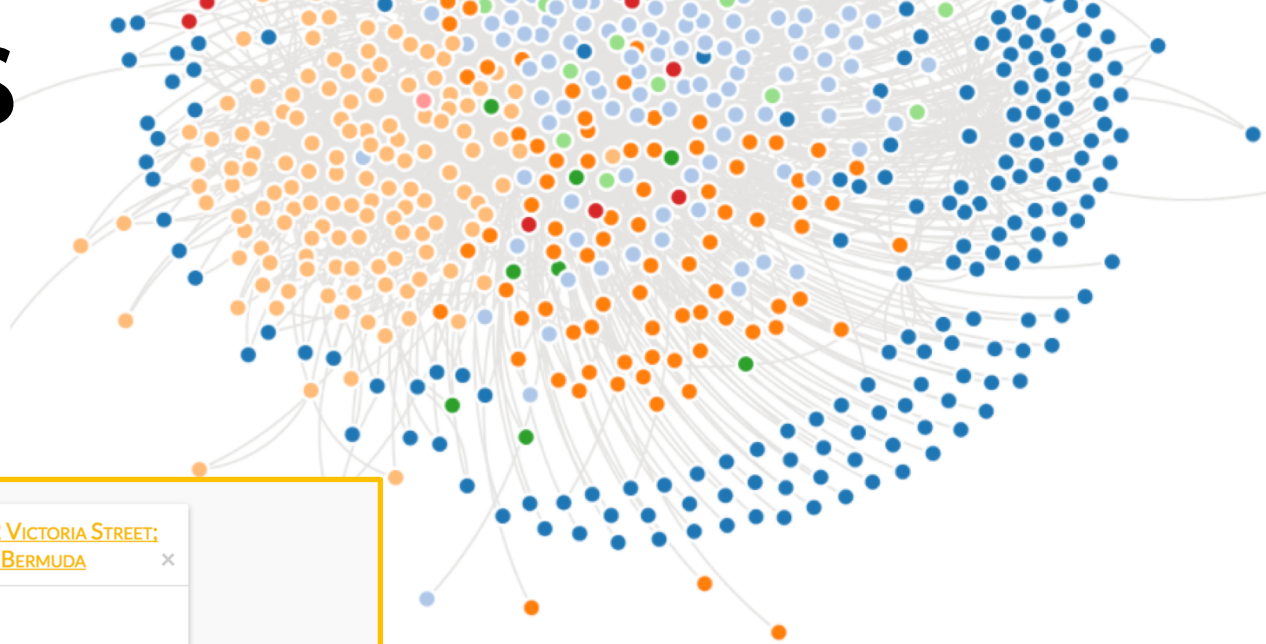
One Graph Query Language

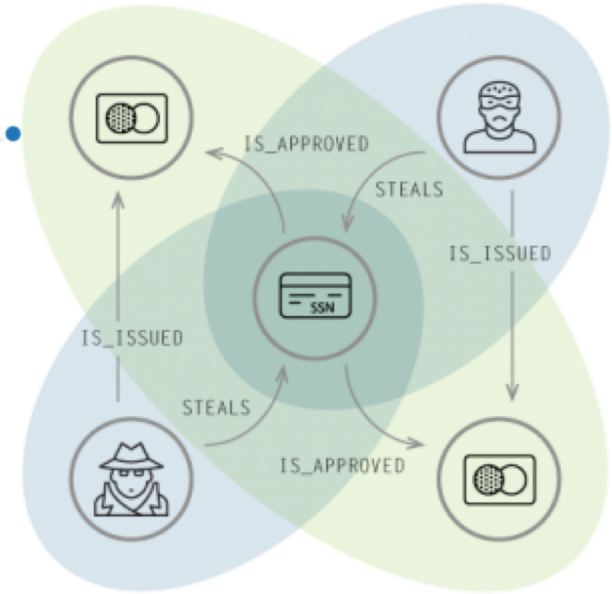
4th openCypher Implementers Meeting
22 – 24 May, 2018
Report

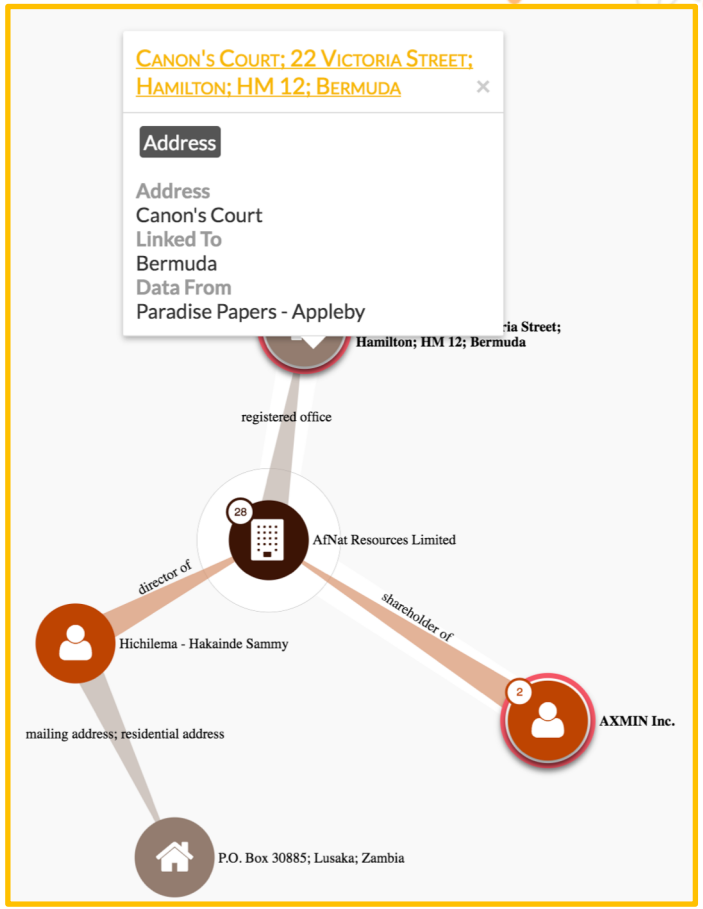
OVERVIEW

- Introduction to graph technologies
- Conference overview
- **GQL Manifesto and design**
- Other talks

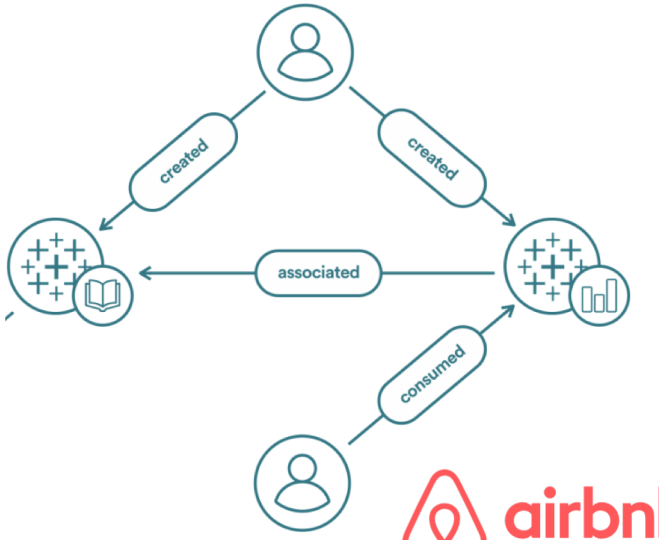
GRAPHS

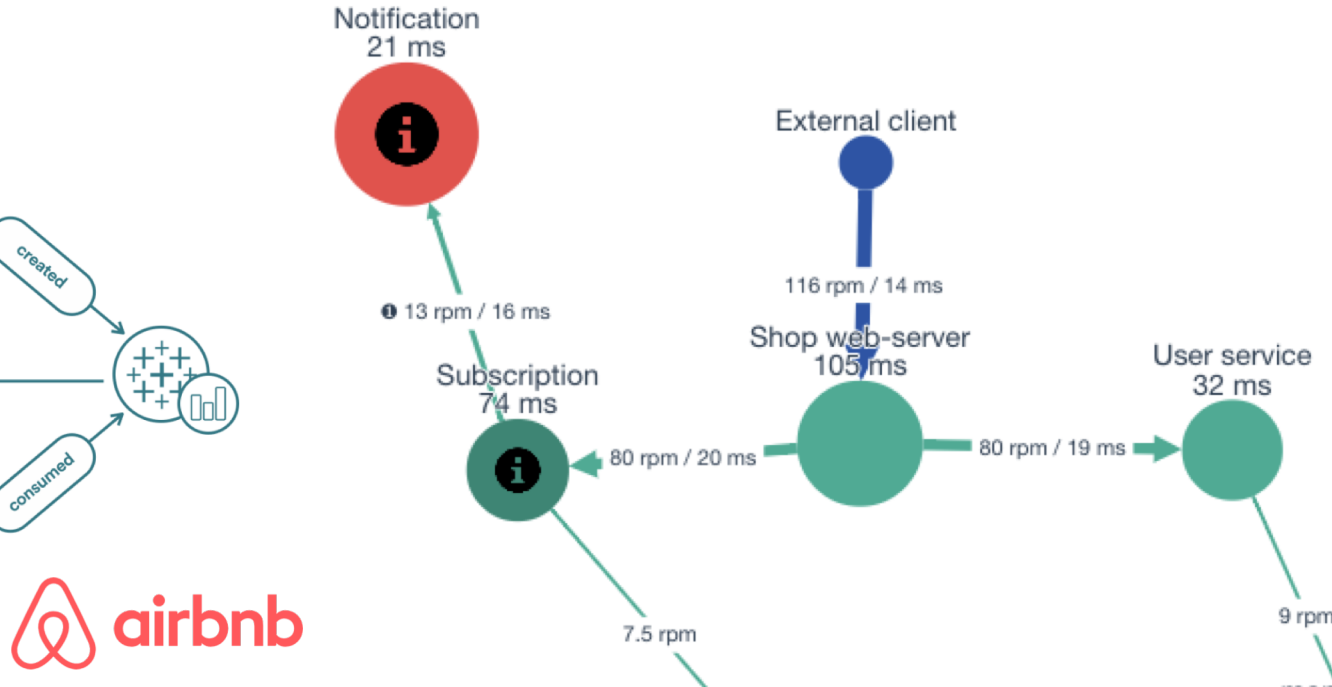




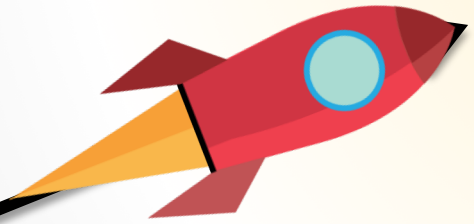


$$G = (V, E) \quad E \subseteq V \times V$$





GRAPH DATABASES - HISTORY



Spark

Cypher

Gremlin

$G = (V, E)$

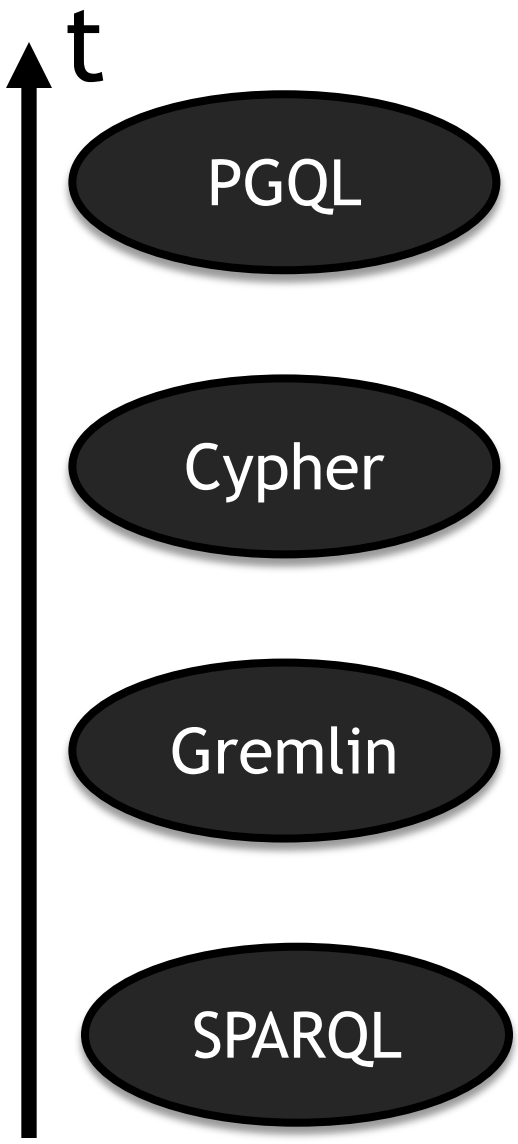
DATASTAX

neo4j

GraphLab

APACHE GIRAPH

GRAPH QUERY LANGUAGES - HISTORY



Property graph QL

Graph traversal API

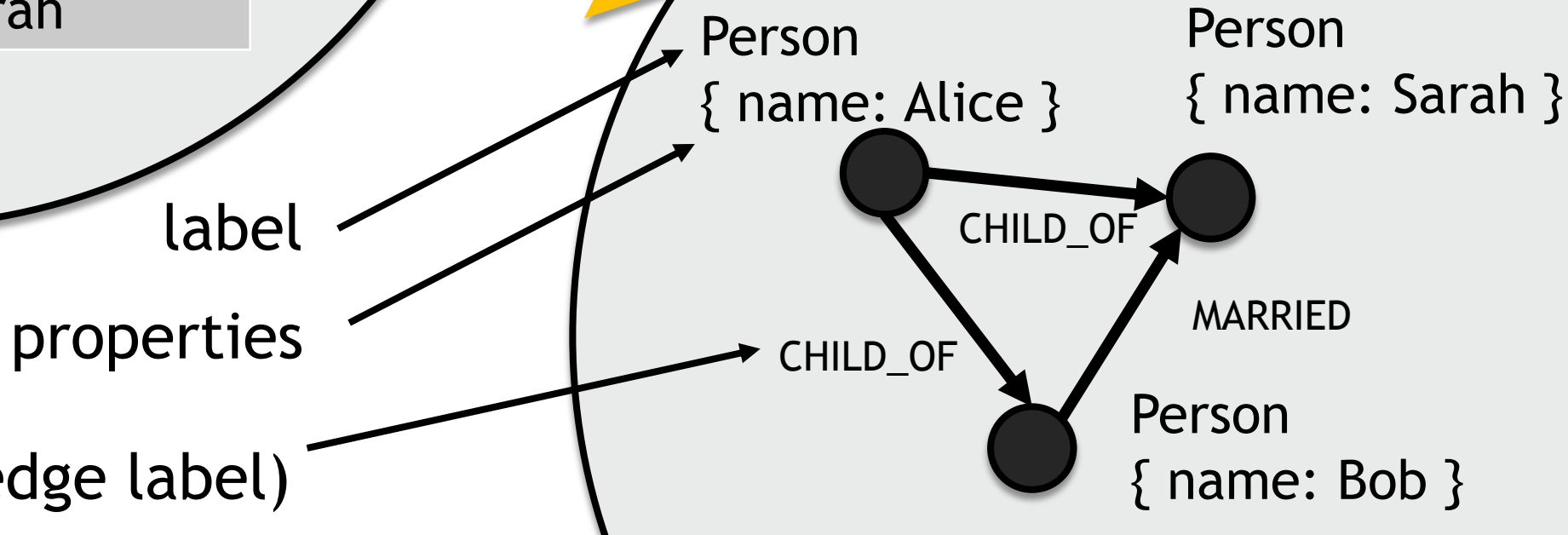
A query language for the semantic web

RELATIONAL

Parent	
person	**parent**
Alice	Sarah
Alice	Bob
Married	
a	**b**
Bob	Sarah

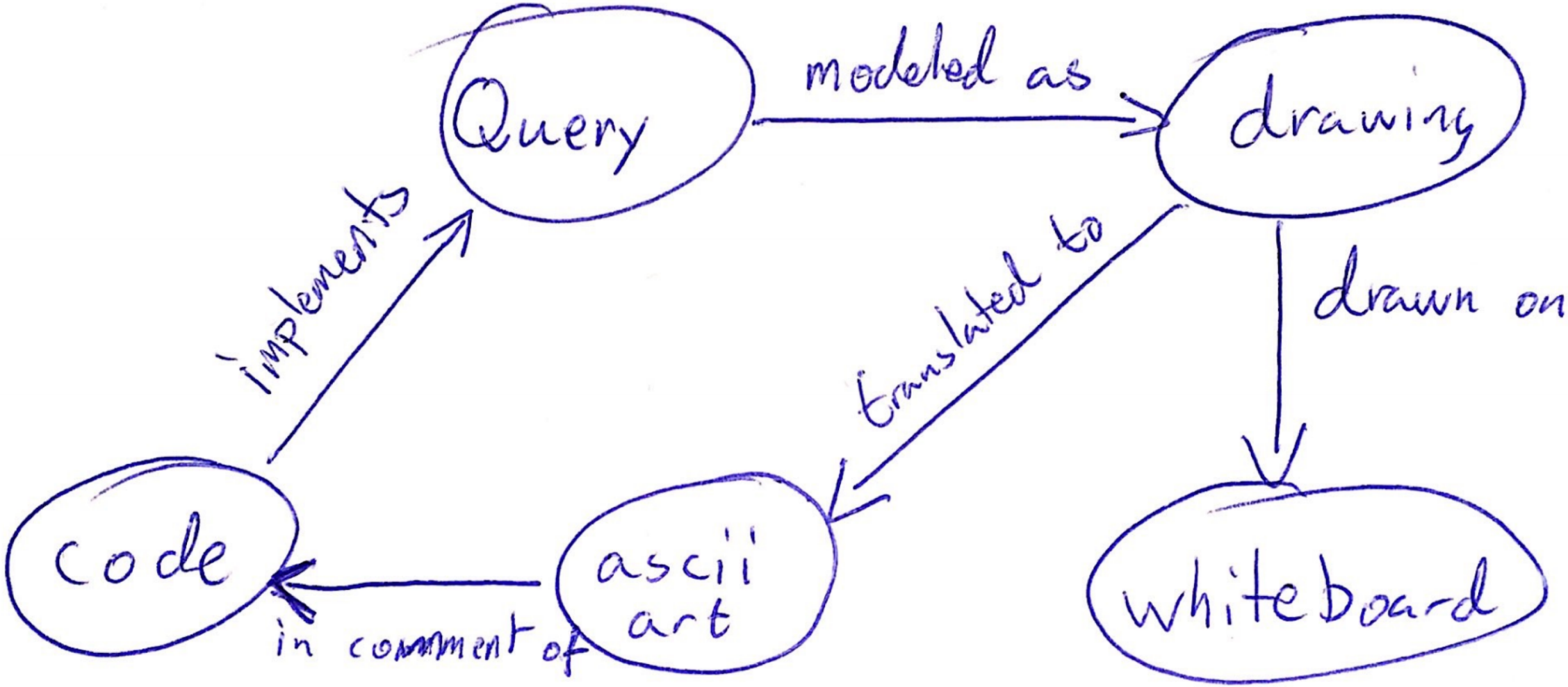
- rich structure
- unityped
- directed edges

PROPERTY GRAPH



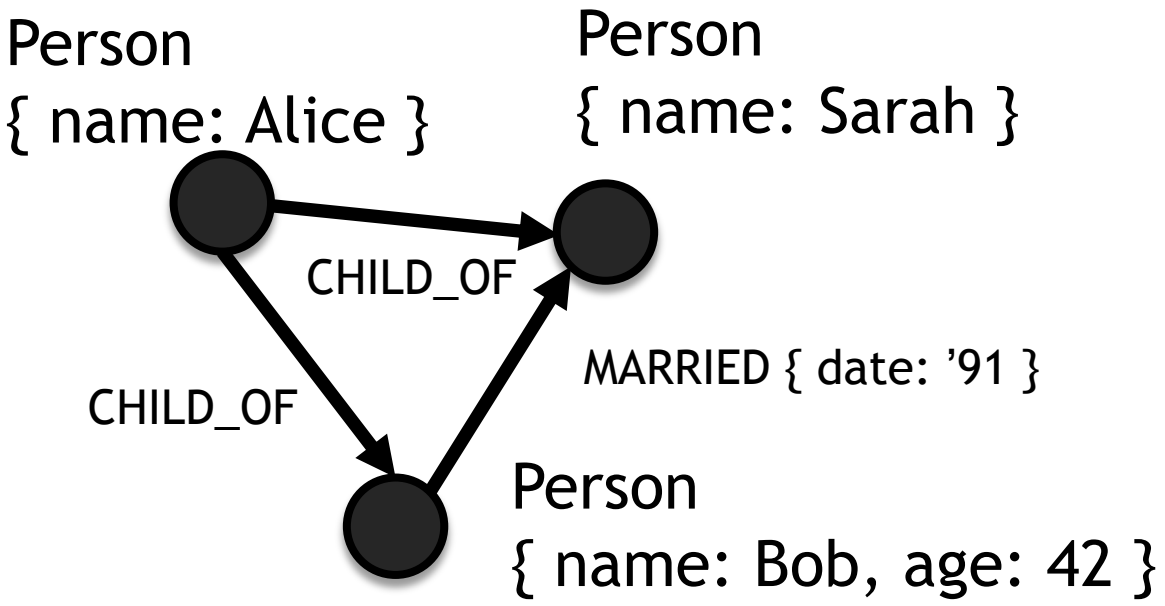
CYPHER

- http://www.opencypher.org/
- human readable
- expressive, intuitive, “immediately familiar”
- openCypher: 2016
- **SAP HANA Graph**, **Redis Graph**, **AgensGraph**, **Neo4j**


Query
modeled as
drawing
implements
translated to
drawn on
code
ascii art
whiteboard
in comment of

CYPHER

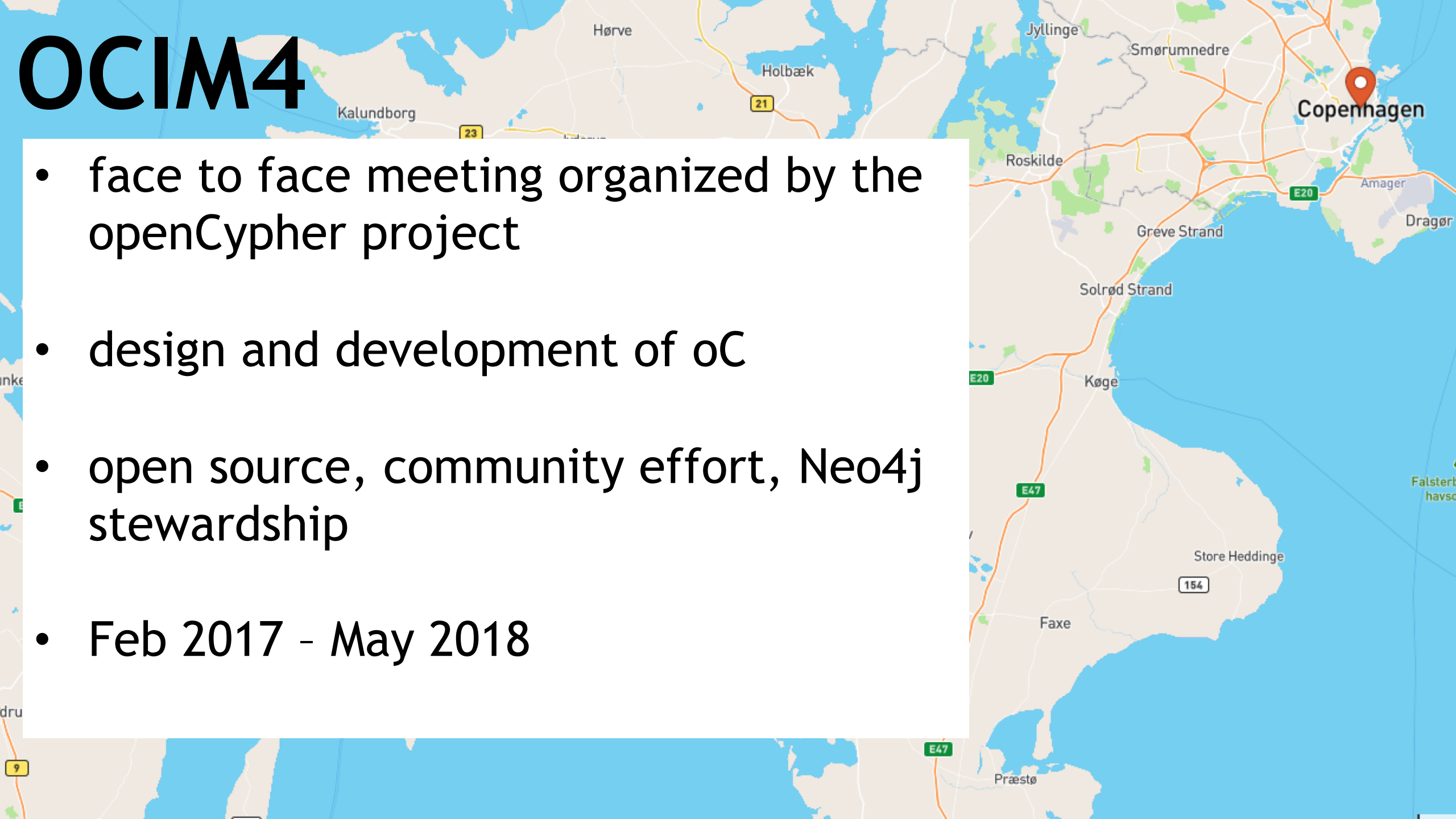
```
MATCH (p:Person {name: $name})-[:CHILD_OF]->
      (f:Person {age: $f_age})-[:MARRIED]->()
RETURN name, f_age
ORDER BY m.date DESC
LIMIT 10
```



name	f_age
Alice	42

OCIM4

- face to face meeting organized by the openCypher project
- design and development of oC
- open source, community effort, Neo4j stewardship
- Feb 2017 – May 2018



AGENDA

TUE 22nd

- openCypher and The GQL Manifesto
- SQL and Property Graphs
- Graph Schema
- HyperGraphQL
- Demo of Cypher for Gremlin (CfoG) and Gremlin Cypher Differences
- Comparing Cypher, PGQL, and G-CORE

WED 23rd

- Multiple graphs and graph projection
- Updatable views and syntax options
- Incremental View Maintenance for openCypher Queries
- An overview of the recent history of Property Graph Query Languages
- Support for the GQL Manifesto from Cypher Implementers
- Learning Timed Automata with Cypher
- Formal Semantics for Cypher Queries and Updates
- Graph Algebra - Graph operations in the language of linear algebra

THU 24th

- Cypher.PL: Specifying Cypher in Prolog
- Temporal support in Cypher
- Graph abstraction

AGENDA

TUE 22nd

- openCypher and The GQL Manifesto
- SQL and Property Graphs
- Graph Schema
- HyperGraphQL
- Demo of Cypher for Gremlin (CfoG) and Gremlin Cypher Differences
- Comparing Cypher, PGQL, and G-CORE

WED 23rd

- Multiple graphs and graph projection
- Updatable views and syntax options
- Incremental View Maintenance for openCypher Queries
- An overview of the recent history of Property Graph Query Languages
- Support for the GQL Manifesto from Cypher Implementers
- Learning Timed Automata with Cypher
- Formal Semantics for Cypher Queries and Updates
- Graph Algebra - Graph operations in the language of linear algebra

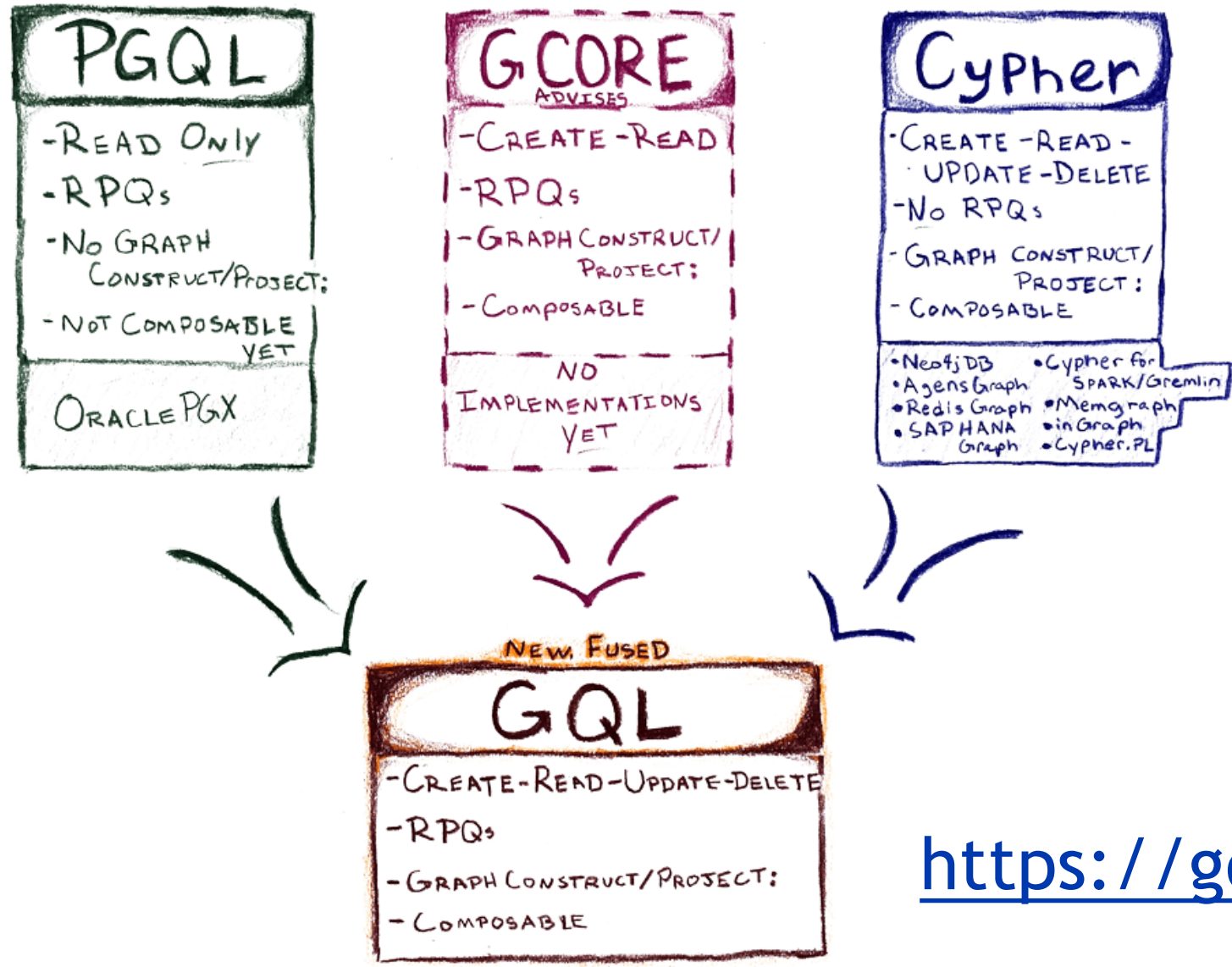
THU 24th

- Cypher.PL: Specifying Cypher in Prolog
- Temporal support in Cypher
- Graph abstraction

GQL MANIFESTO

CYPHER - PROBLEMS

- “The initial reaction to openCypher was not as warm as the reaction that we’ve seen to GQL”
- “You could triangulate the actual Cypher language to get the semantics”
- lack of composability



https://gql.today/

GQL MANIFESTO

- [dʒiːkəʊl]
- goal is to influence the future of standard query languages
- totally open process, APL v2
- June 2019
- smooth adoption process:
  - supersedes Cypher
  - one- or bi-directional mapping tools
- OpenCypher initiatives will be aligned to GQL
- top prio: read-only core

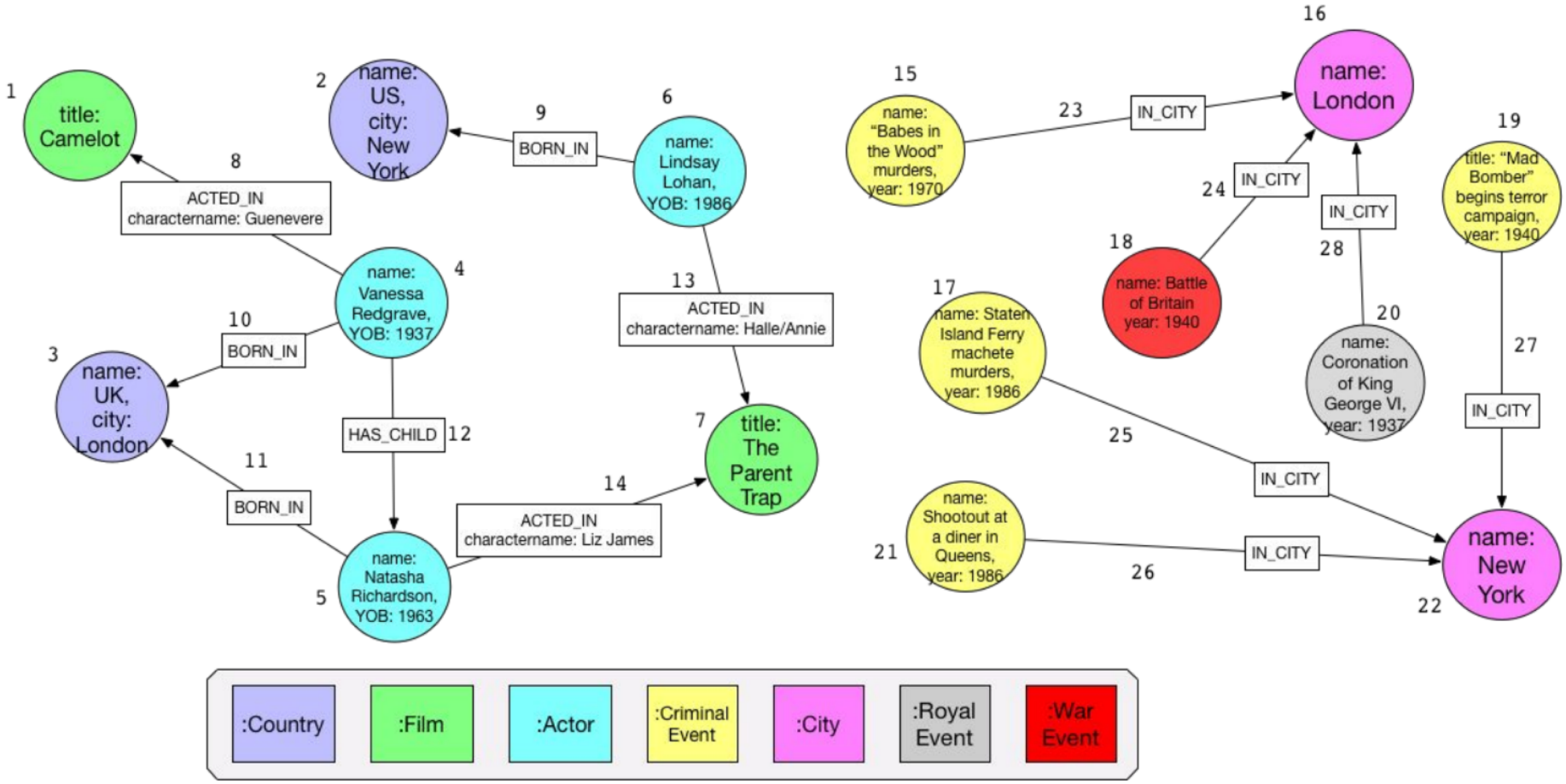
GQL – TECHNICAL DESIGN

- multiple graphs, namespaces
- graph construction, updateable views
- graph abstraction
- regular path queries
- configurable matching semantics

MULTIPLE GRAPHS – WHY?

- Combining and transforming graphs from multiple sources
- Versioning, snapshotting, computing difference graphs
- Graph views for access control

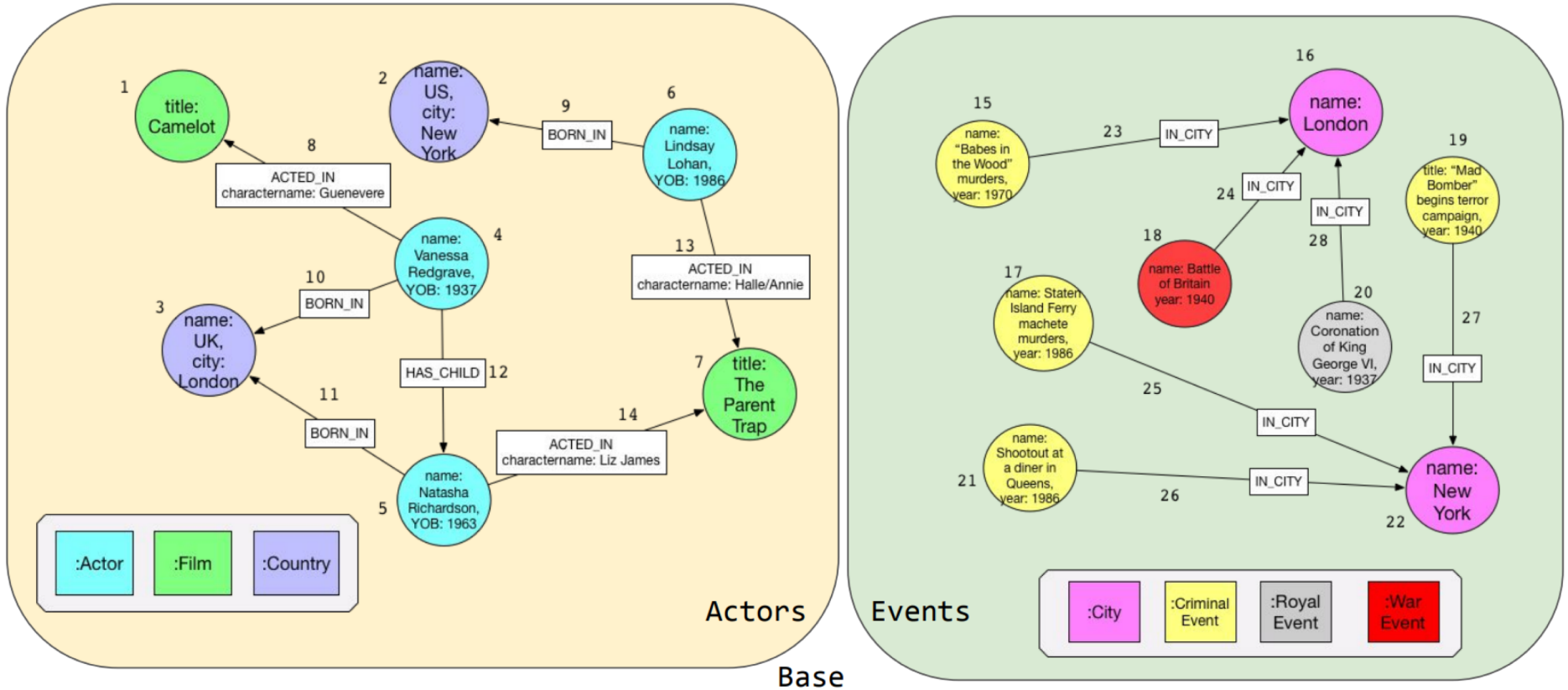
MULTIPLE GRAPHS



MULTIPLE GRAPHS

```
CREATE GRAPH Events {
  FROM GRAPH Base
  MATCH ()-[:IN_CITY]->()
  RETURN GRAPH
}
```

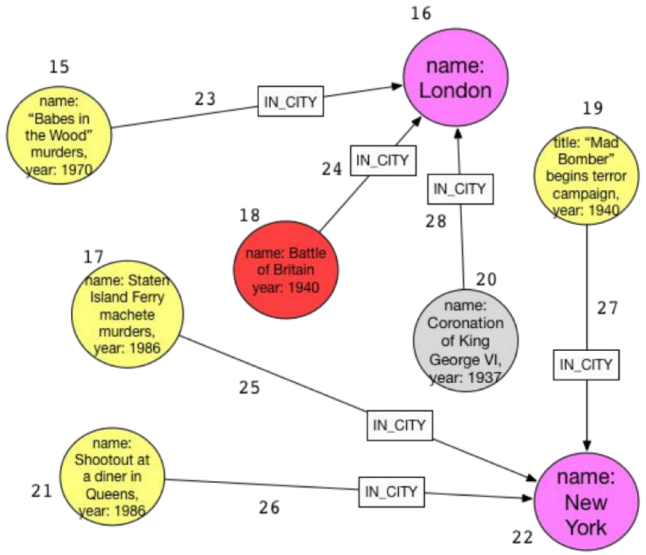
```
FROM Events
MATCH (e)-[:IN_CITY]->(c)
RETURN c.name AS City,
       e.year AS Year,
       count(e) AS NumberOfEvents
```



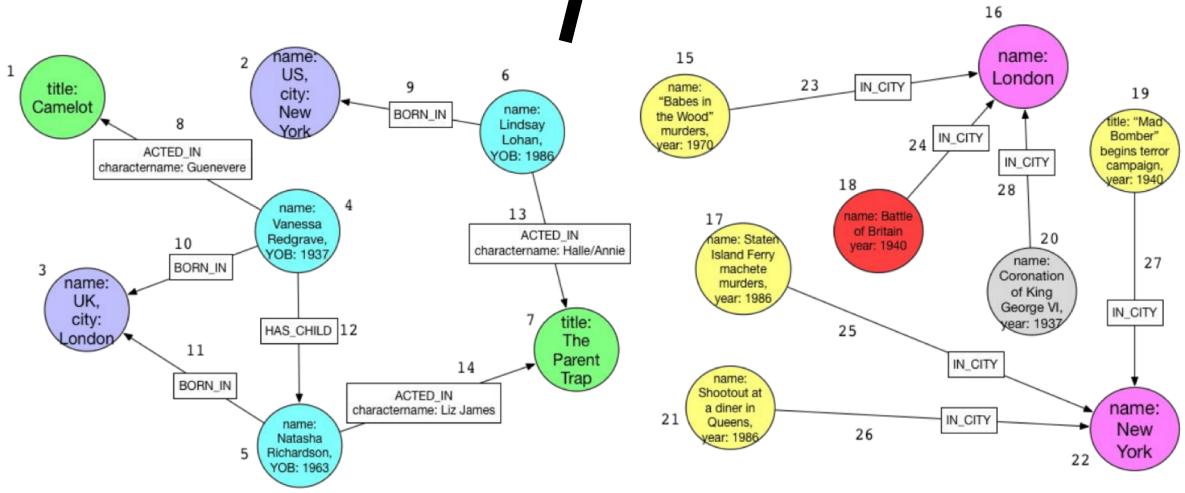
MULTIPLE GRAPHS

```
CREATE GRAPH CriminalEvents {
  FROM Events MATCH
(e:CriminalEvents)-[:IN_CITY]->()
  RETURN GRAPH
}
```

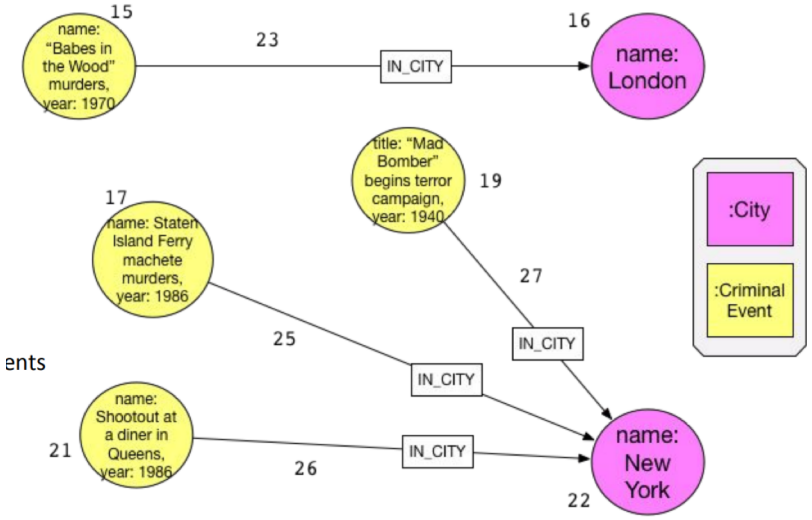
Events



Base



CriminalEvents



MULTIPLE GRAPHS

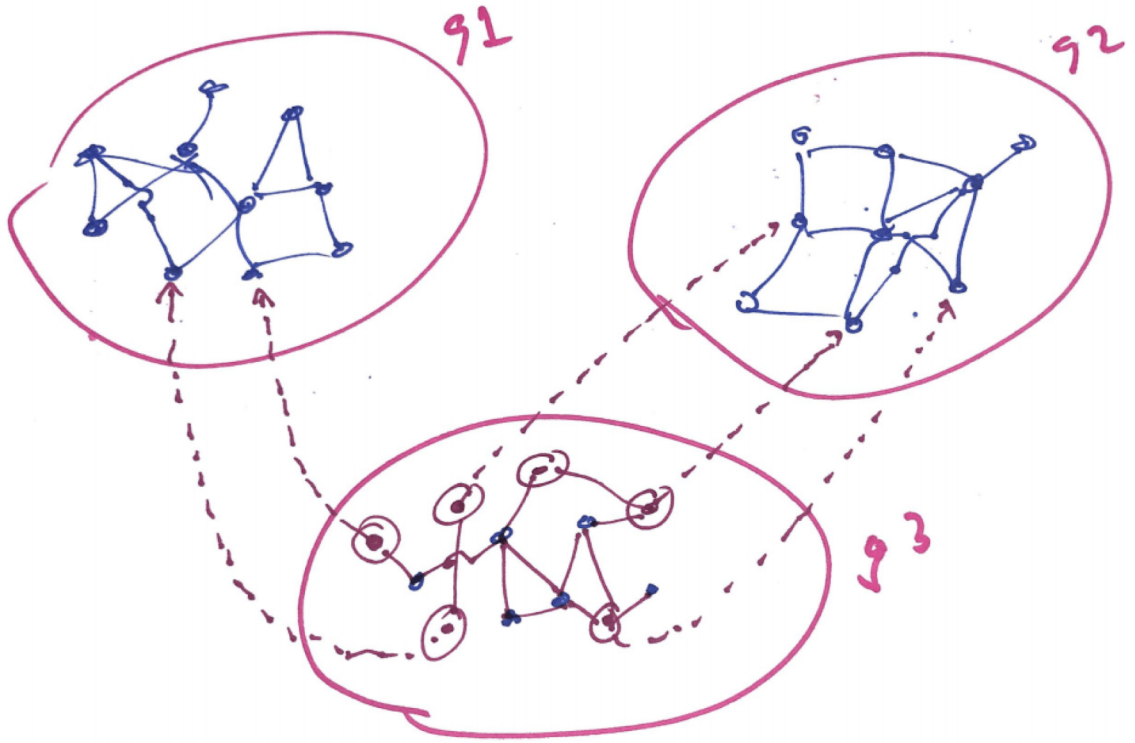
- copy into distinct graph - by value
  - forms new entities in the created graph
- view source from new graph - by reference
  - only ”subtractions” allowed
- mixed
  - powerful but complicated

GRAPH CONSTRUCTION

- dynamically constructs a new working graph
- CAPS already has some support for it
- G-CORE: Entities are references to base data but shared in views
- GQL Proposal:
  - identity and equality semantics
  - formalize provenance tracking
  - construction behavior
  - updatable view behavior

GRAPH CONSTRUCTION

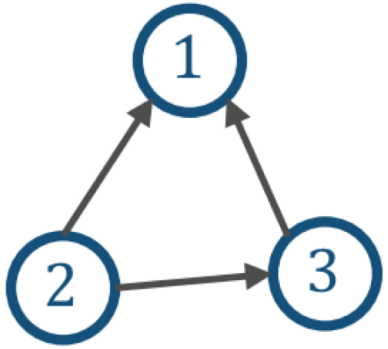
- Provenance tracking via entity sharing
  - entities belong to one and only on graph (ownership)
- Provenance graph
- Entity values:
  - References to a replica group with the same root
- Updatable views



GRAPH ABSTRACTION

- semantics for constructing graphs with patterns

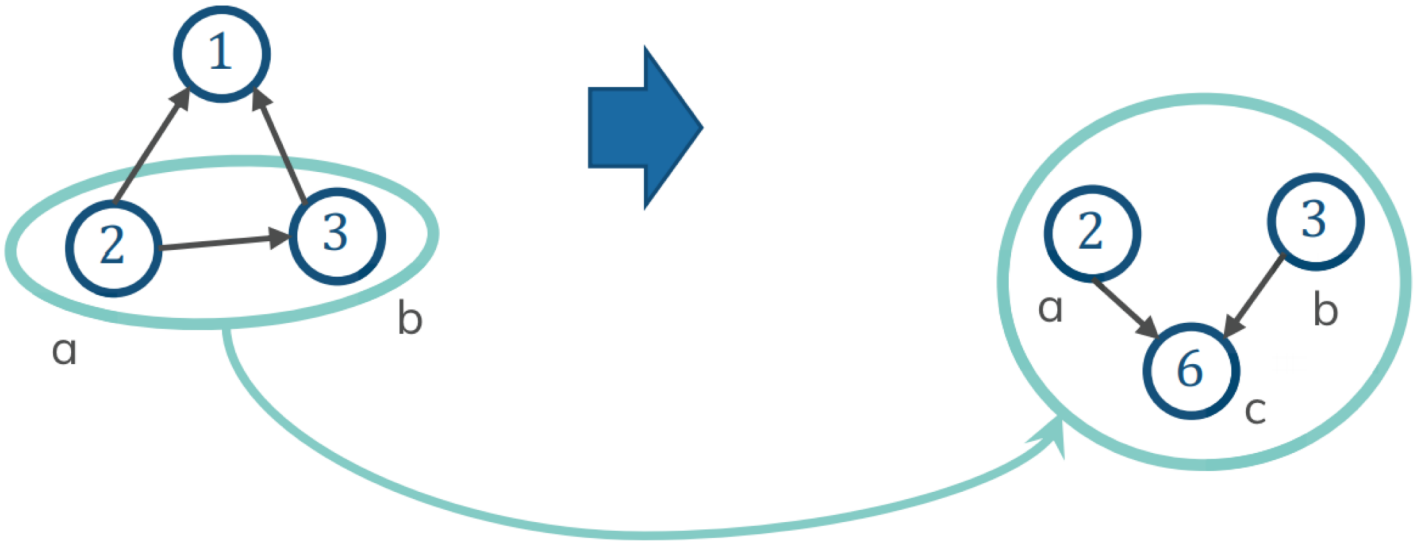
```
MATCH      (a)-->(b)
CONSTRUCT  (a)-->(c)<--(b)
```



GRAPH ABSTRACTION

- semantics for constructing graphs with patterns

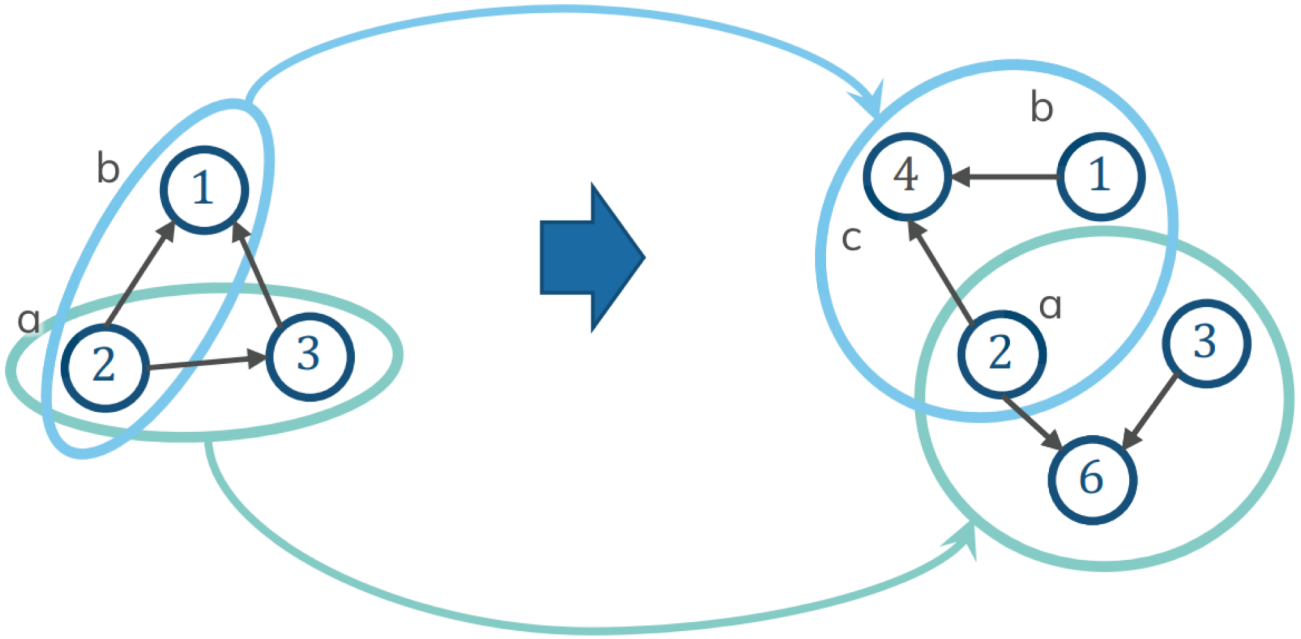
```
MATCH      (a)-->(b)
CONSTRUCT  (a)-->(c)<--(b)
```



GRAPH ABSTRACTION

- semantics for constructing graphs with patterns

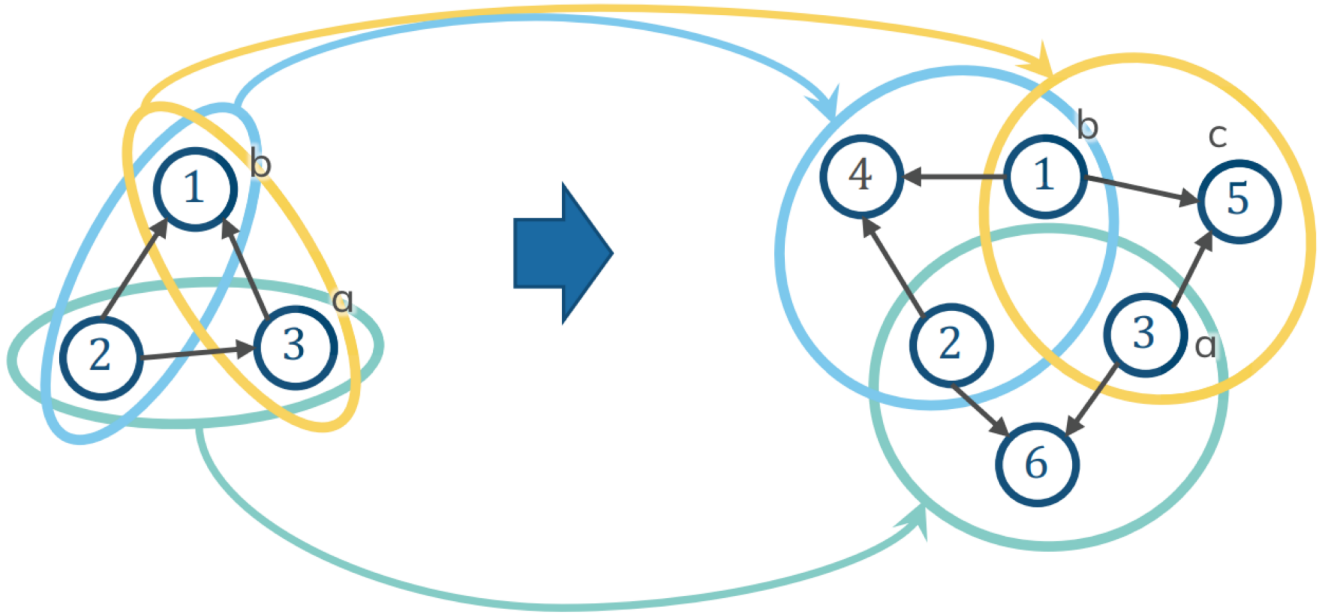
```
MATCH      (a)-->(b)
CONSTRUCT  (a)-->(c)<--(b)
```



GRAPH ABSTRACTION

- semantics for constructing graphs with patterns

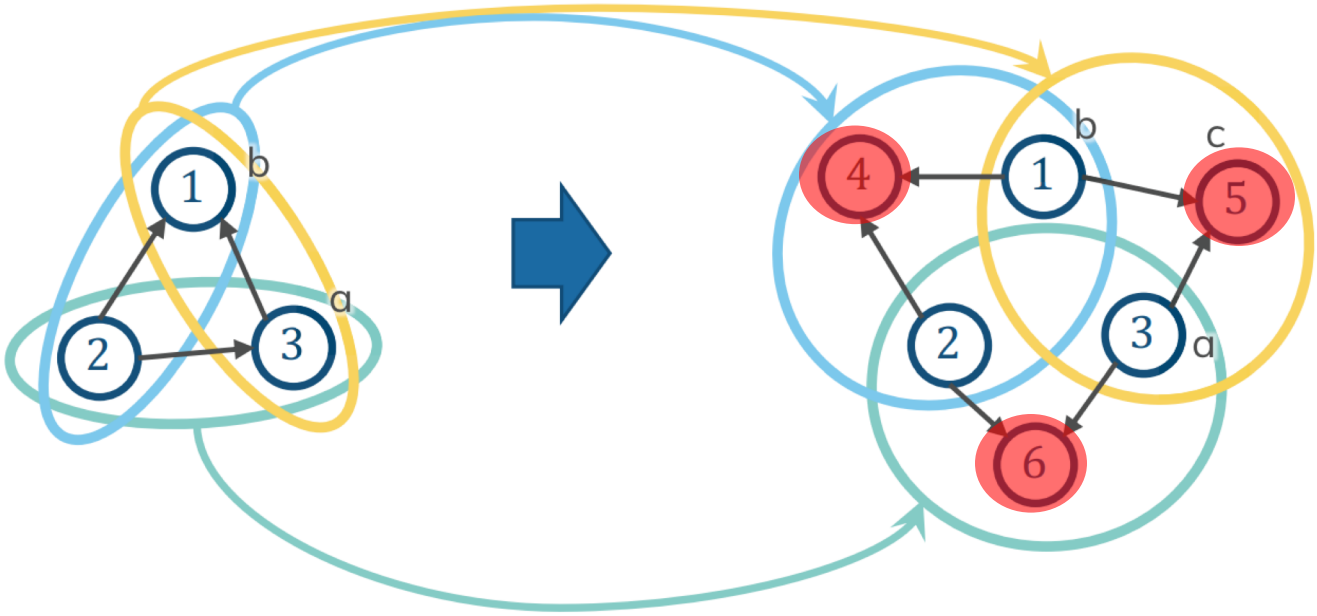
```
MATCH      (a)-->(b)
CONSTRUCT  (a)-->(c)<--(b)
```



GRAPH ABSTRACTION

- semantics for constructing graphs with patterns

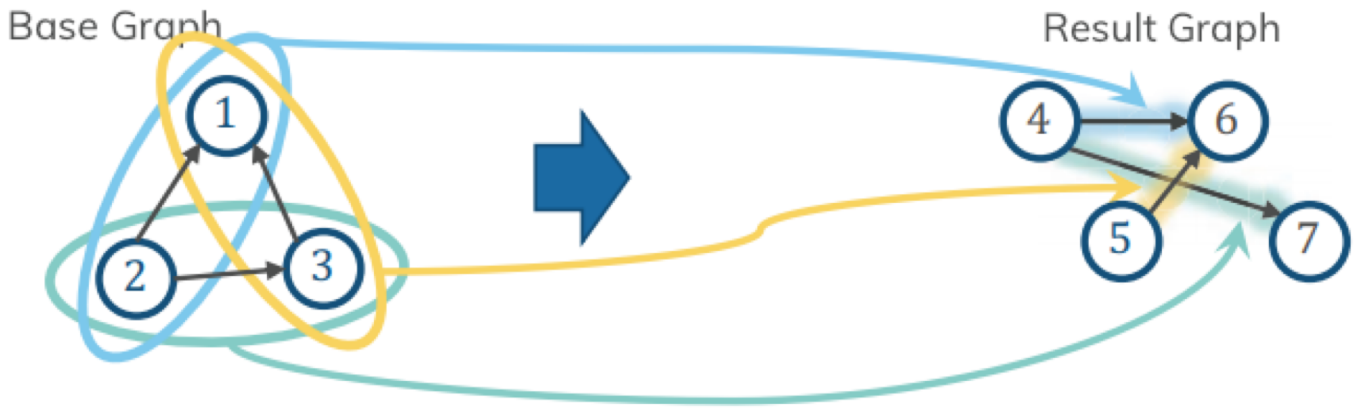
```
MATCH      (a)-->(b)
CONSTRUCT  (a)-->(c)<--(b)
```



GRAPH AGGREGATION

- grouping variables:
  - nodes, edges
  - Grouping variables must be a subset of the set of all bound variables

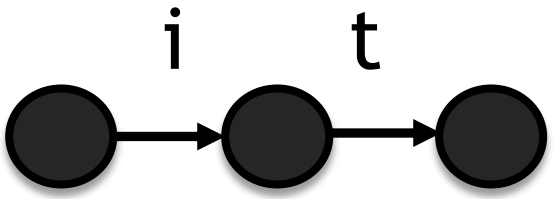
```
MATCH      (a)-->(b)
CONSTRUCT  (c GROUP a)-[e]->(d GROUP b)
```



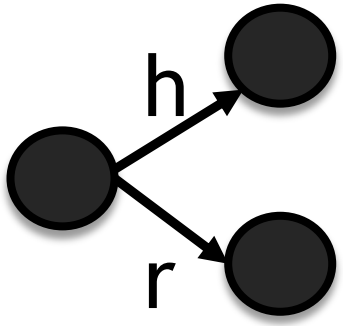
REGULAR PATH QUERIES

```
W[hr]ite\s?p?a*ger?s?
```

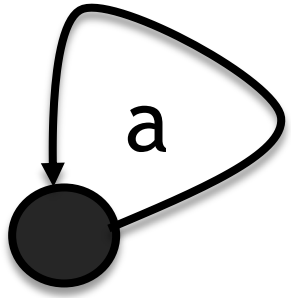
```
Whitepages
Whitepagers
Write pages
Write aaaaages
```



it



[hr]

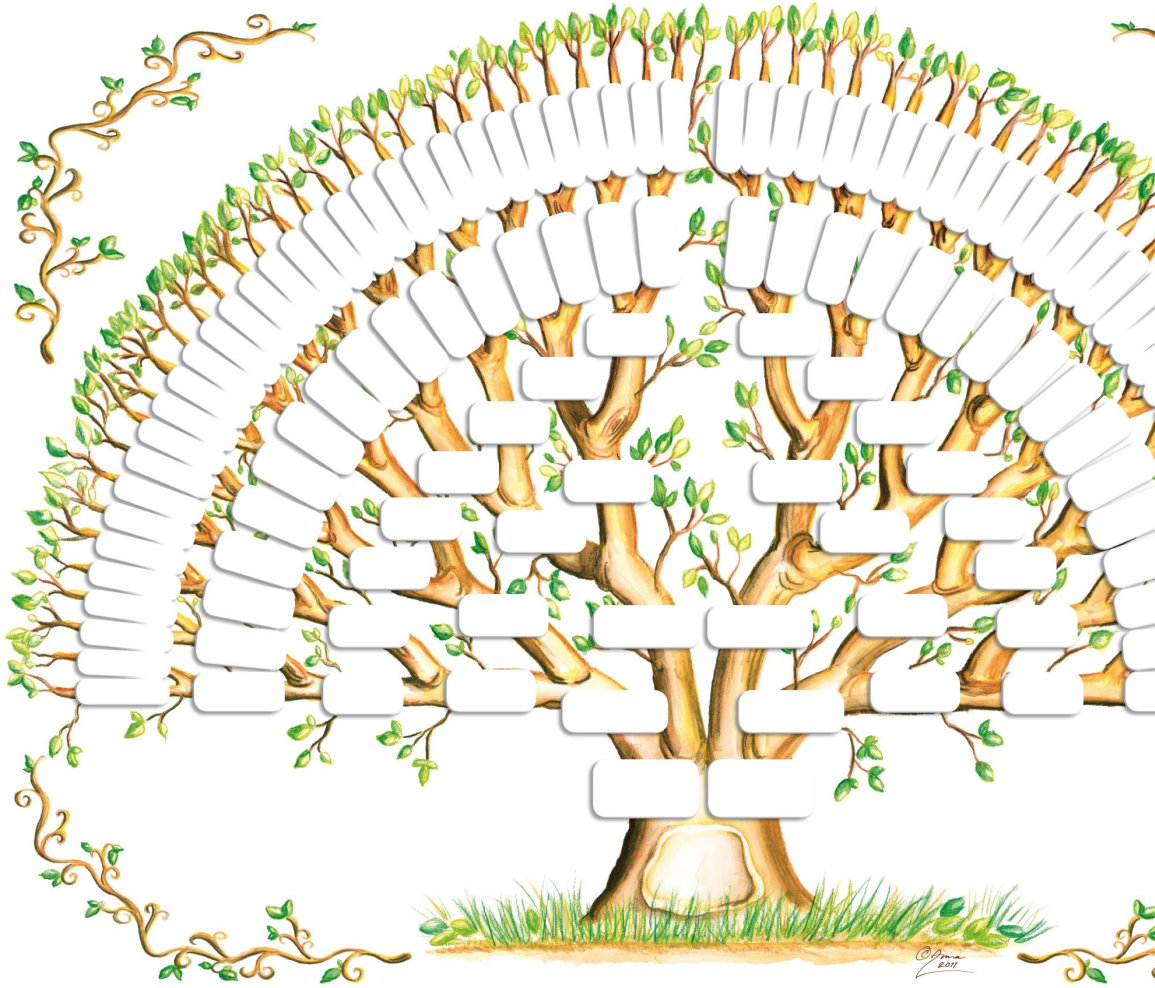


a*

REGULAR PATH QUERIES

```
PATH has_parent := () -[:has_father|has_mother]-> ()
SELECT ancestor
WHERE
  (:Person WITH name = 'Mario')
       -/:has_parent*/->
  (ancestor:Person),

  (:Person WITH name = 'Luigi')
       -/:has_parent*/->
  (ancestor:Person)
```



CONFIGURABLE MATCHING

- Cypher: edge-isomorphic, vertex-homomorphic
- Proposed:
  - path constraint eg. shortest, cheapest
  - homo
  - explicit-iso
  - vertex-iso
  - edge-iso

PROPERTY GRAPHS ARE COMING TO SQL!

SQL

- read-only core of Cypher 3.4 as part of ISO SQL:2020
- SAP/IBM/Microsoft agreed to that
- Timeframe 2019

```
SELECT aName, bName
FROM YourCoolGraph GRAPH_TABLE(
  MATCH (a)(-[b]->)*(c)
  WHERE …
  ONE ROW PER [MATCH|STEP(n,r)]
  COLUMNS(
    a.name AS aName, b.name AS bName
    N.id,
    R.distance,
    ELEMENT_NO(i)
  )
)
```

HONORABLE MENTIONS

HONORABLE MENTIONS

- Incremental View Maintenance [Gábor Szárnyas]
  - Event-driven programming in databases
- Formal Semantics for Cypher Queries and Updates [Martin Schuster]
  - non-deterministic operations in Cypher (SET, MERGE, DELETE)
- Cypher.🇵🇱 [Jan Posiadała]: Cypher formalized in Prolog
- RedisGraph's super fast sparse matrix graphs [Roi Lipman]

QUESTIONS TIME!

http://www.opencypher.org/event/2018/05/22/ocim4/