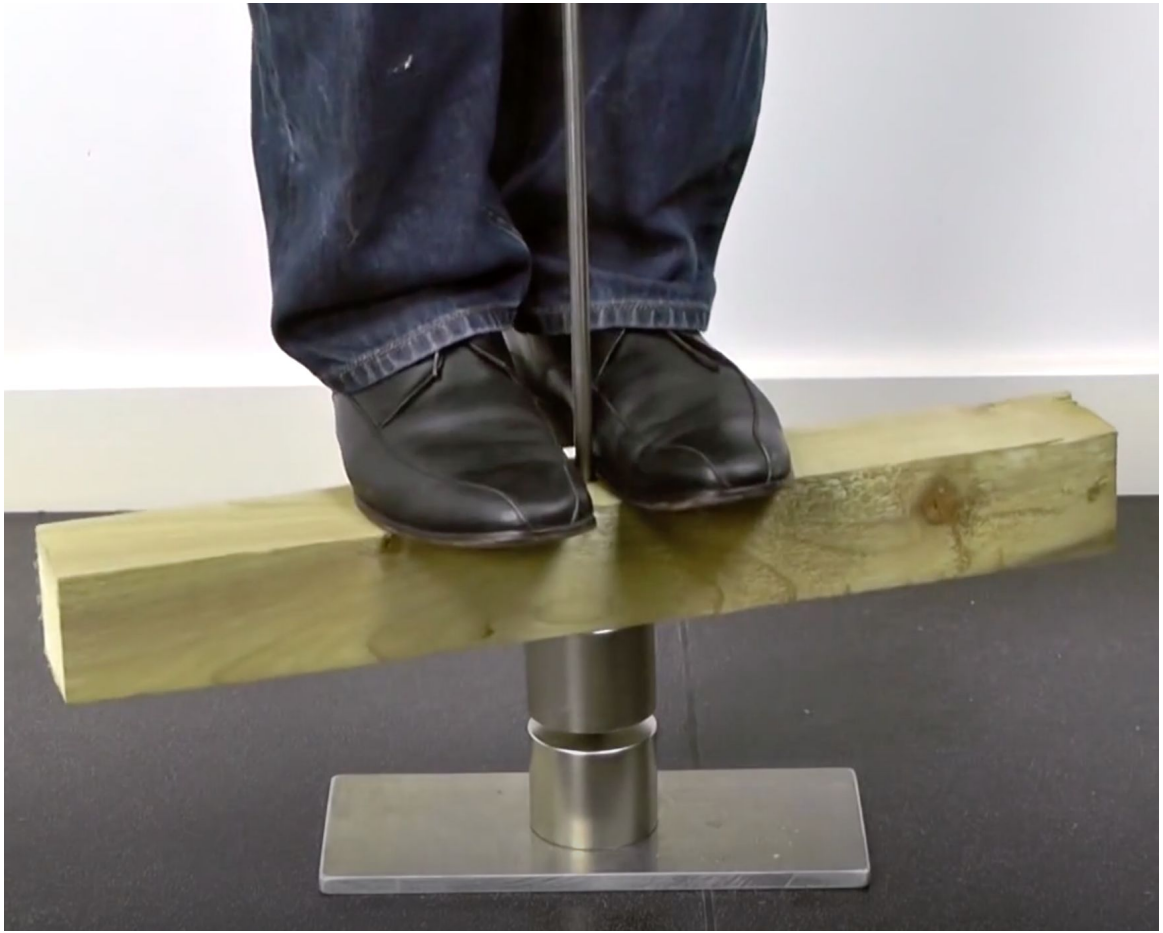
Interoperation & Co-operation

W3C Workshop, 4-6 March 2019, Berlin **Graph Data Management Standards**

Alastair Green Query Languages Standards and Research, Neo4j

Like poles repel

Two 300kg pull power magnets cannot be forced together by the weight of a man



Like poles repel

Two 3c
pull po
magne
cannot
forced
by weig
a man

Three large software vendors found it hard to agree on how to handle XML in SQL

Like poles repel

Two 3c
pull po
magne
cannot
forced
by weig
a man

RDF

- openCypher
- SQL/PGQ
- PGQL
- G-CORE

Gremlin

Creating effective software standards is hard

Useable

Adopted

Uncontested

Useable

Sufficient features
Fit requirements
Ergonomic for users
Fit for implementation

Adopted

Timely
Community
Accessible

But above all ...

Uncontested

Just the one

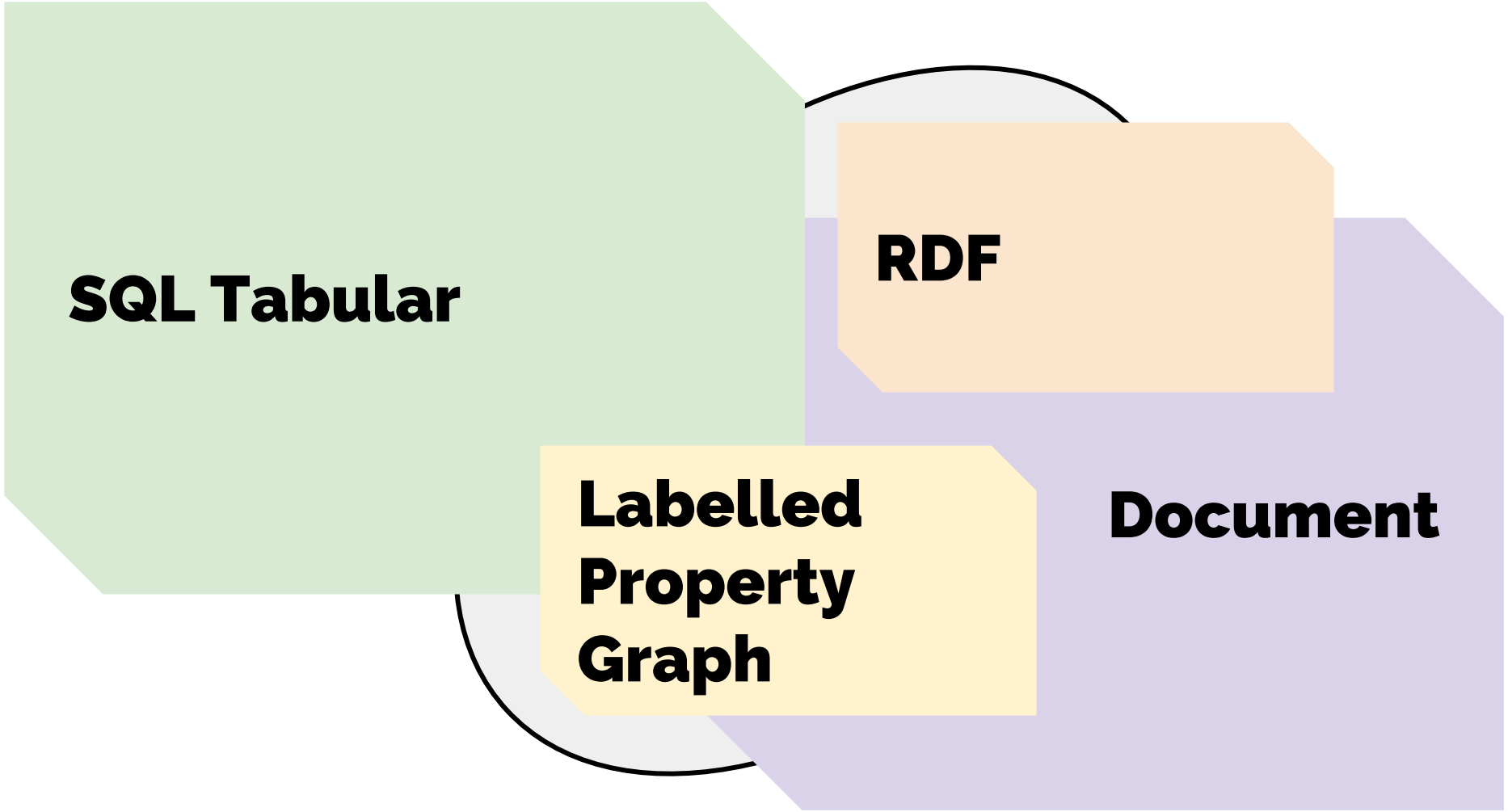
Just the one,
for each ...

Domain
Community

Why do data communities arise?

Data model

Uses of data


SQL Tabular
RDF
Labelled Property Graph
Document

Algorithms

Traversals
Gremlin

DeclarativeQueries

Interchange

Results
GraphQL

Schema

Domain standards and interchange standards

Data communities are not islands

Users either want or have to deal with multiple models and uses

Bridging SQL, RDF & Property Graphs

An opportunity to create co-operative approaches to interoperation in graph data management

Minimal Interoperation

Each community has to gets its house in order

Property graph world seems to lag the RDF world in this respect

“GQL face-to-face”

In theory . . .

SQL could be extended to do everything for graphs

SPARQL could be extended to do everything for PG and tables

A property graph GQL that handles tables and graphs could do everything SQL can do

In practice . . .

That would lead to paralysis, or endless war

Data communities have very deep social and product roots, and large to huge user bases

Like humans, they can’t get personality transplants

In practice . . .

That would lead to paralysis, or endless war

Data communities have very deep social and product roots, and large to huge user bases

Like humans, they can’t get personality transplants

Co-operate to define reasonable interoperation standards

SQL/PGQ

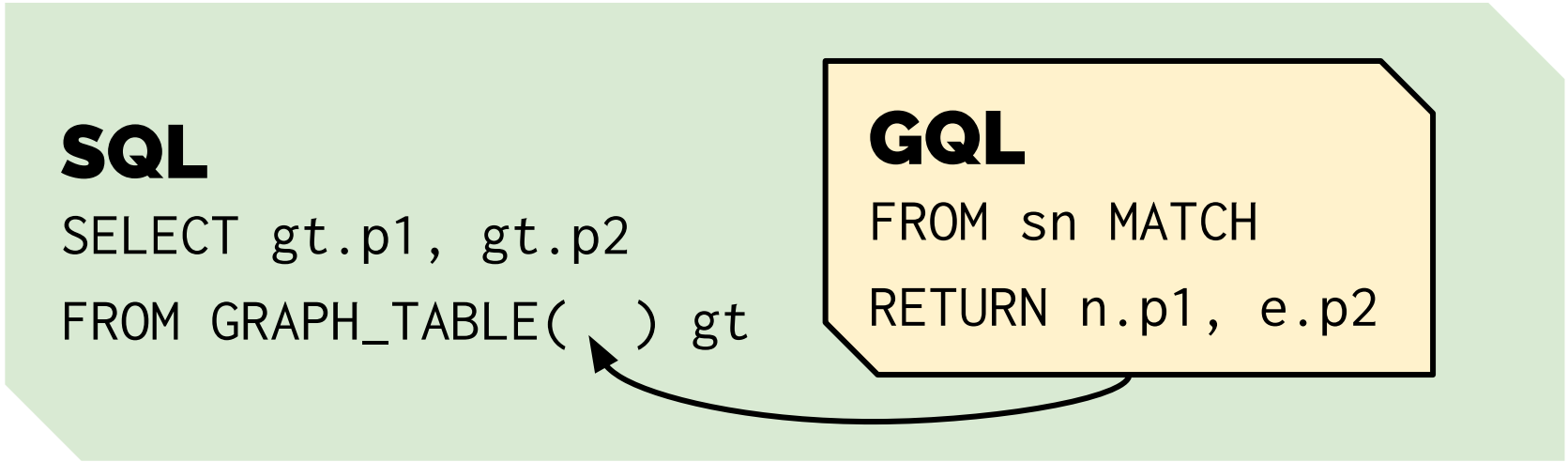
SQL-GQL shared substrate

sparql-gremlin + CfoG

GQL-SPARQL* views

role of graph typing

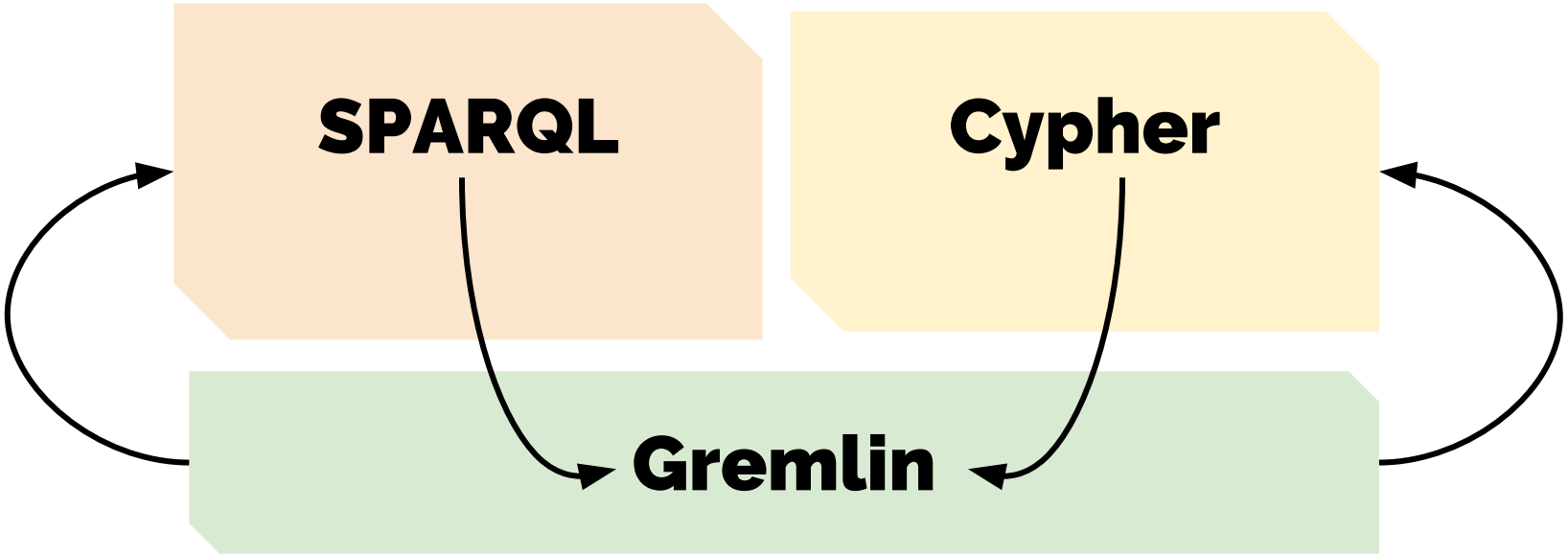
data interchange


SQL
SELECT gt.p1, gt.p2
FROM GRAPH_TABLE( ) gt
GQL
FROM sn MATCH
RETURN n.p1, e.p2

SQL

GQL

scalar datatypes
expressions
tabular post-project clauses
catalog
authN model ...


SPARQL
Cypher
Gremlin

GQL

```
FROM sfoo MATCH
```

SPARQL* named query

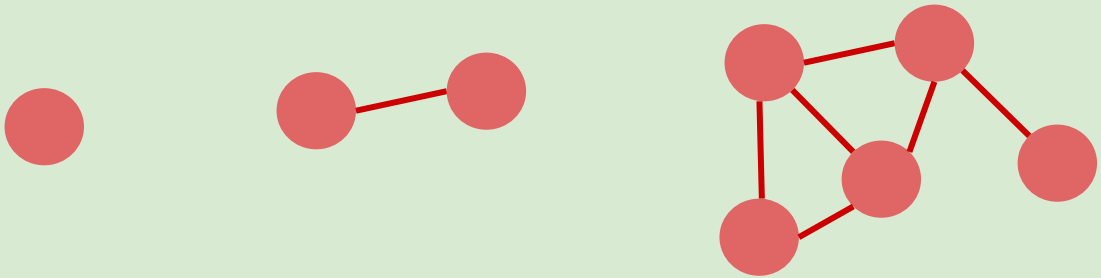
```
sfoo # that can CONSTRUCT
```

Graph program

Query

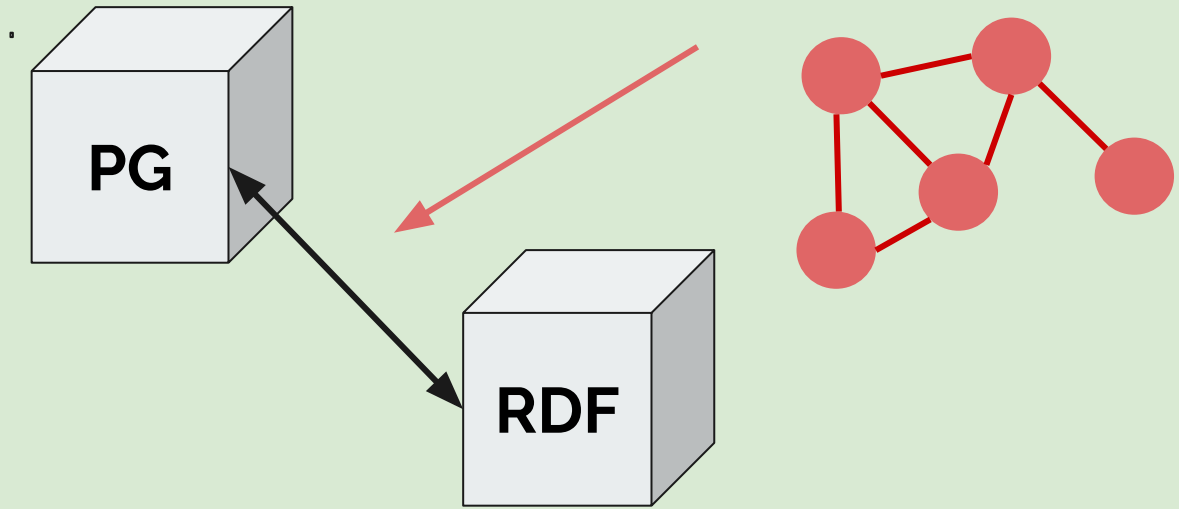
Imperative/Traversal

Compute



X-model
Data interchange

Graph Schema/Type



Co-operate to define reasonable interoperation standards