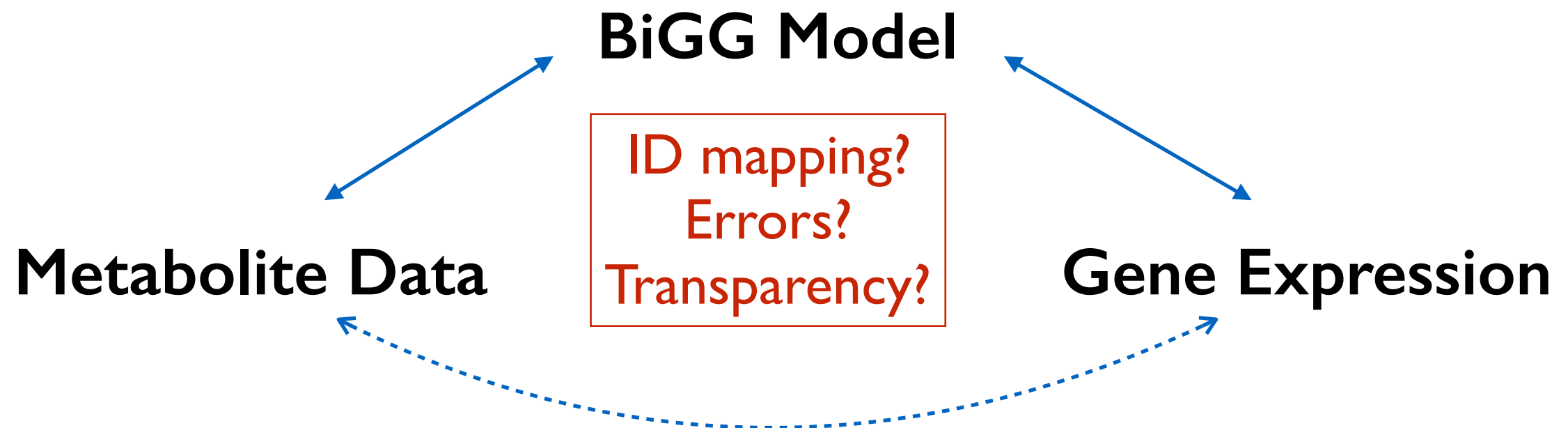
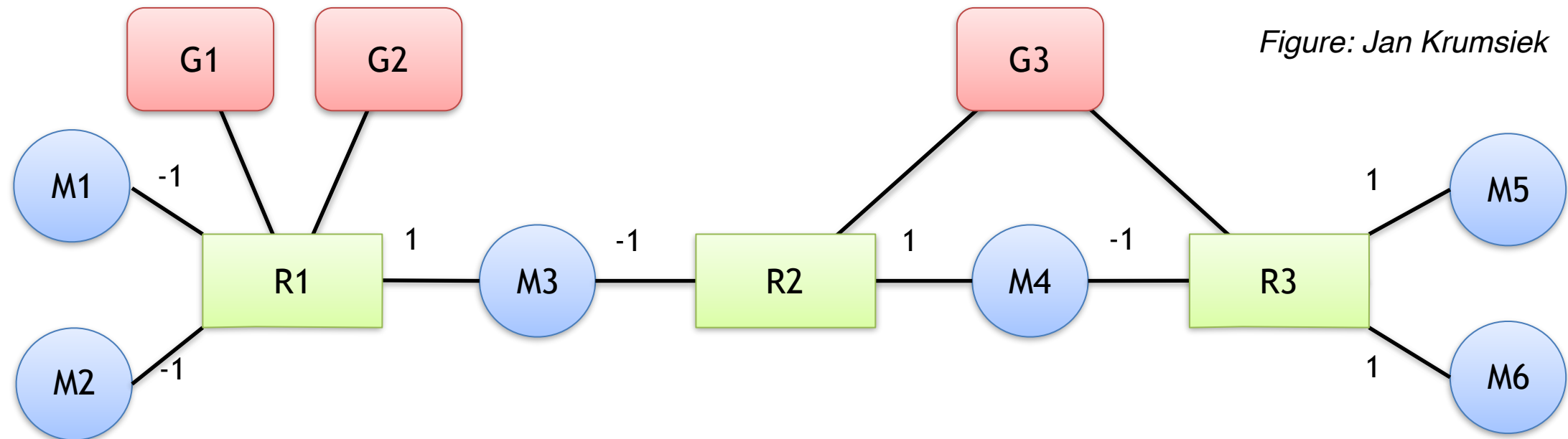


Neo4j - a graph DB

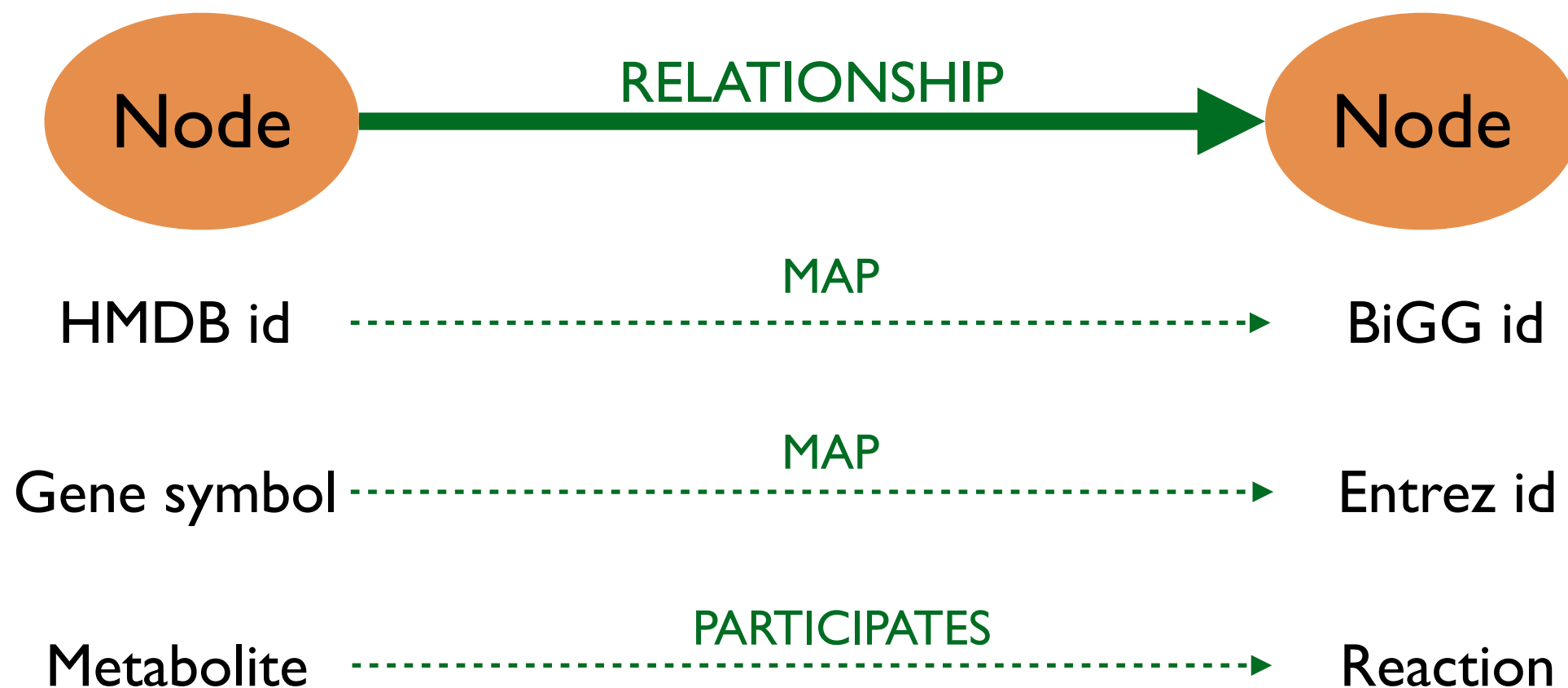
Maria Wörheide January 19, 2017

Problematics of data integration



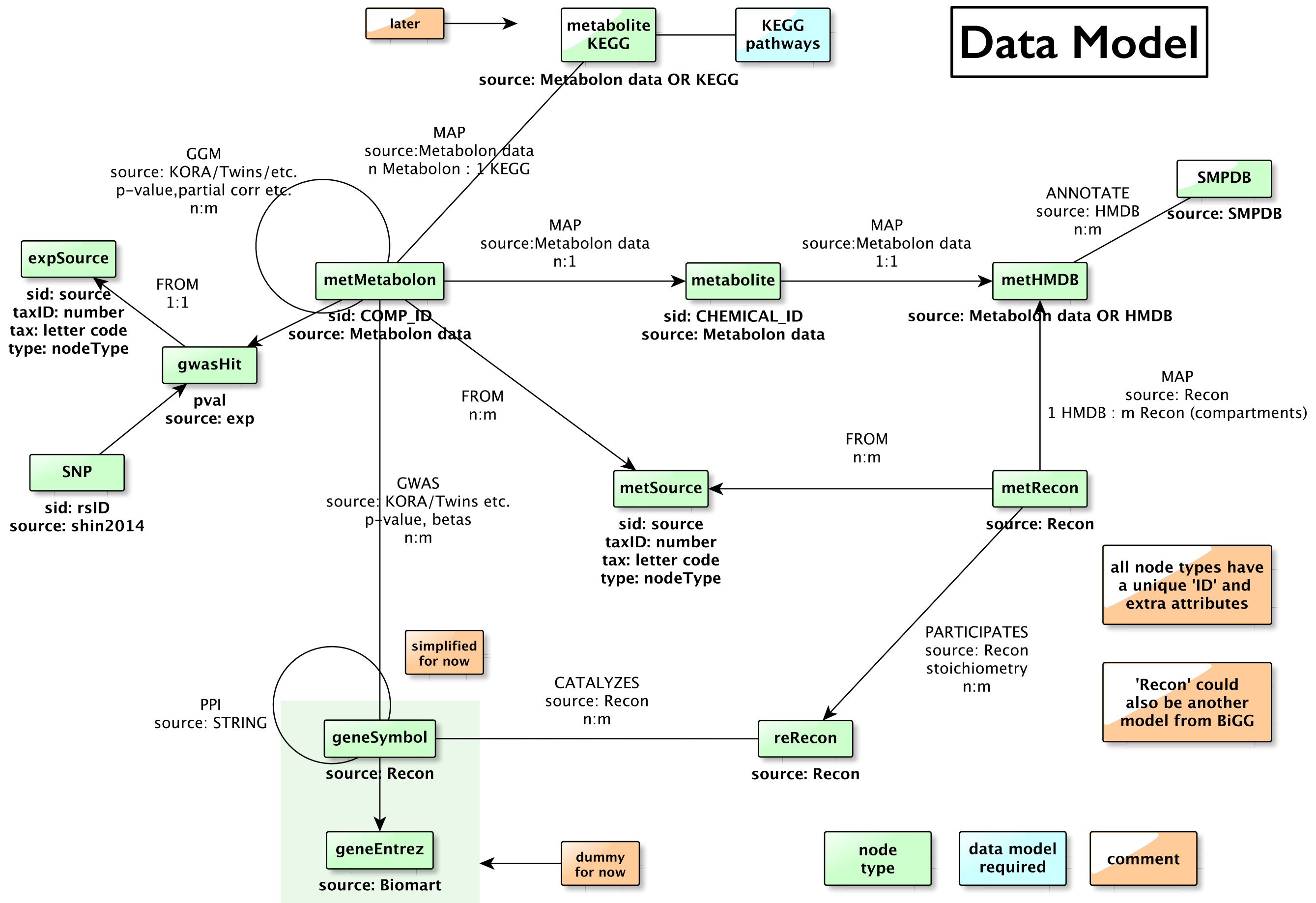
neo4j - a native graph DB

“Today's world is no longer driven by data – it's driven by the connections between them¹⁾. “



¹⁾<https://neo4j.com/product/>

Data Model



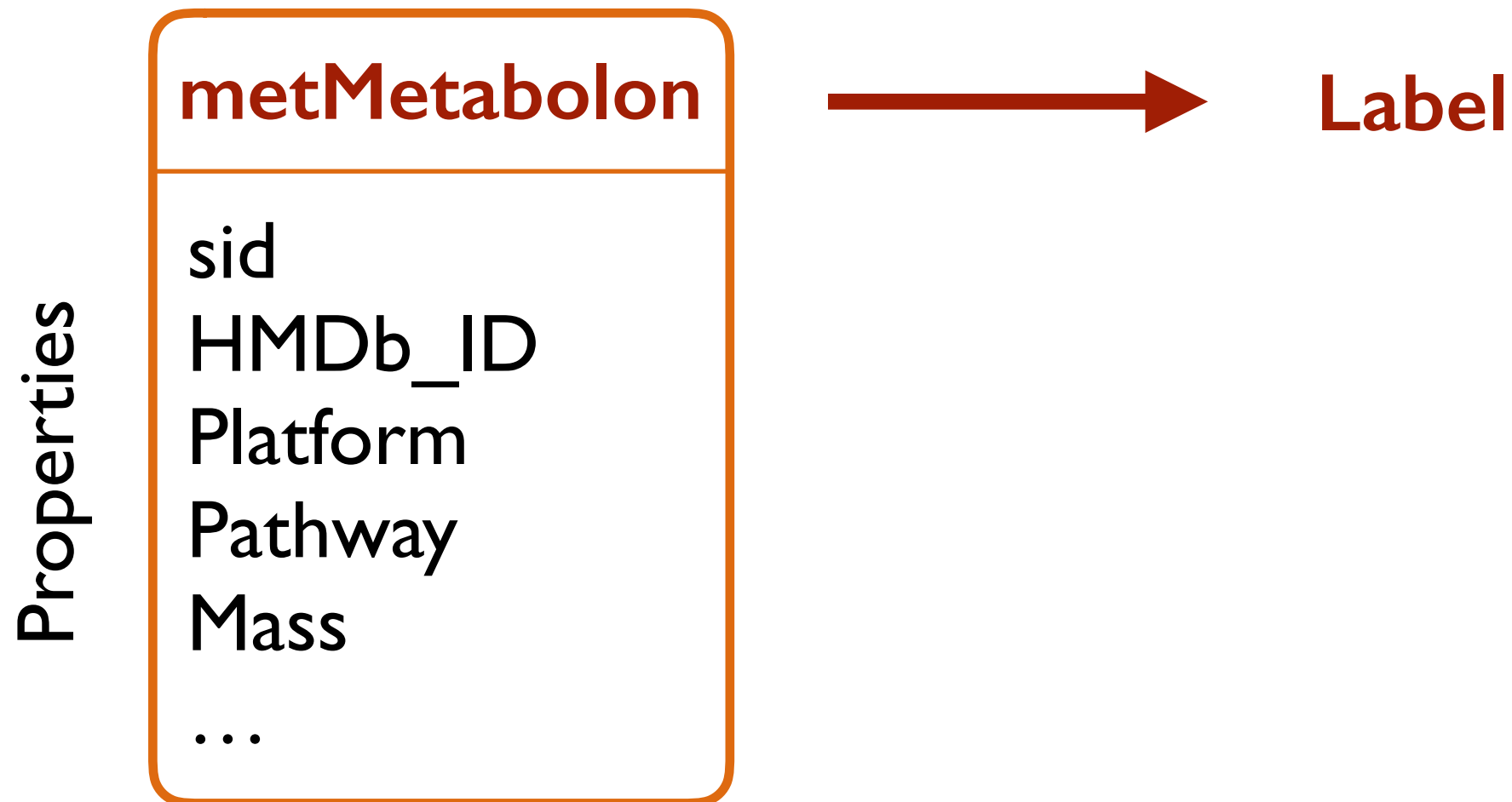
Neo4j - Nodes

Properties

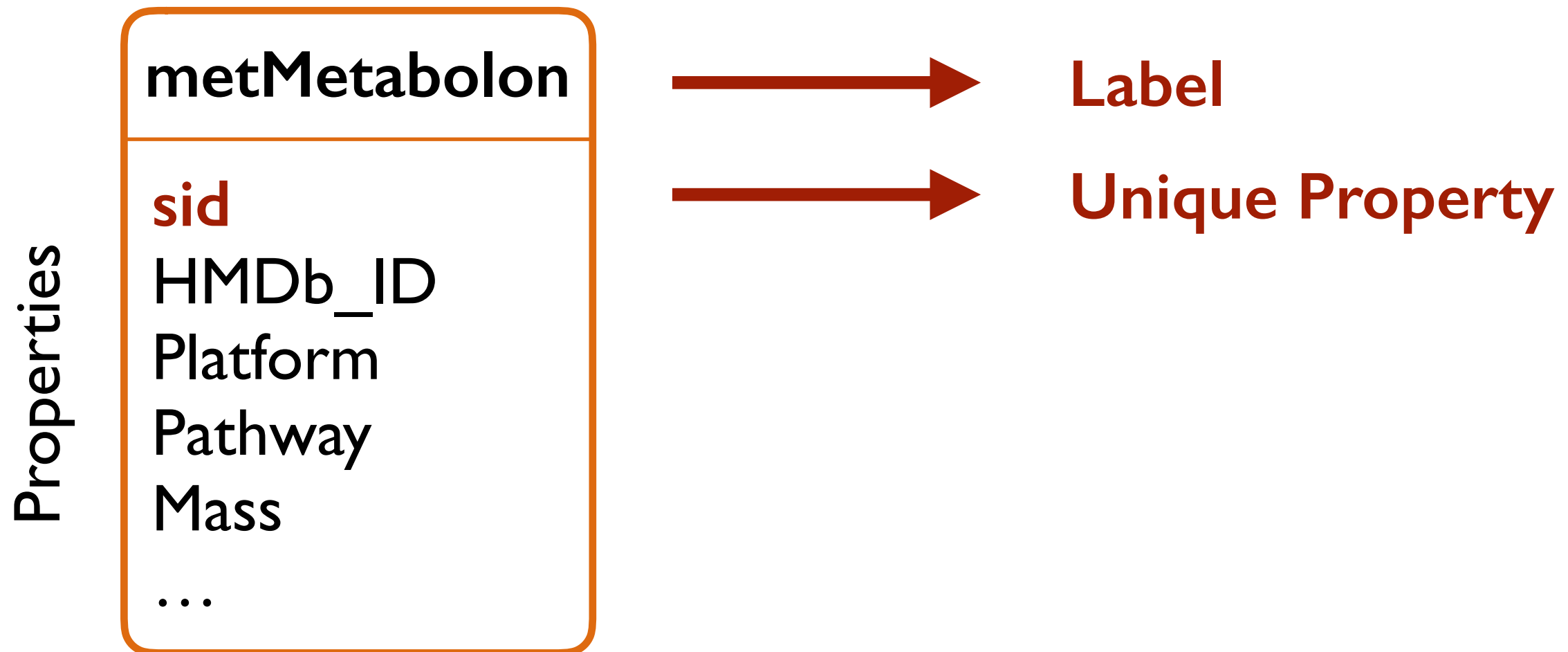
metMetabolon

sid
HMDB_ID
Platform
Pathway
Mass
...

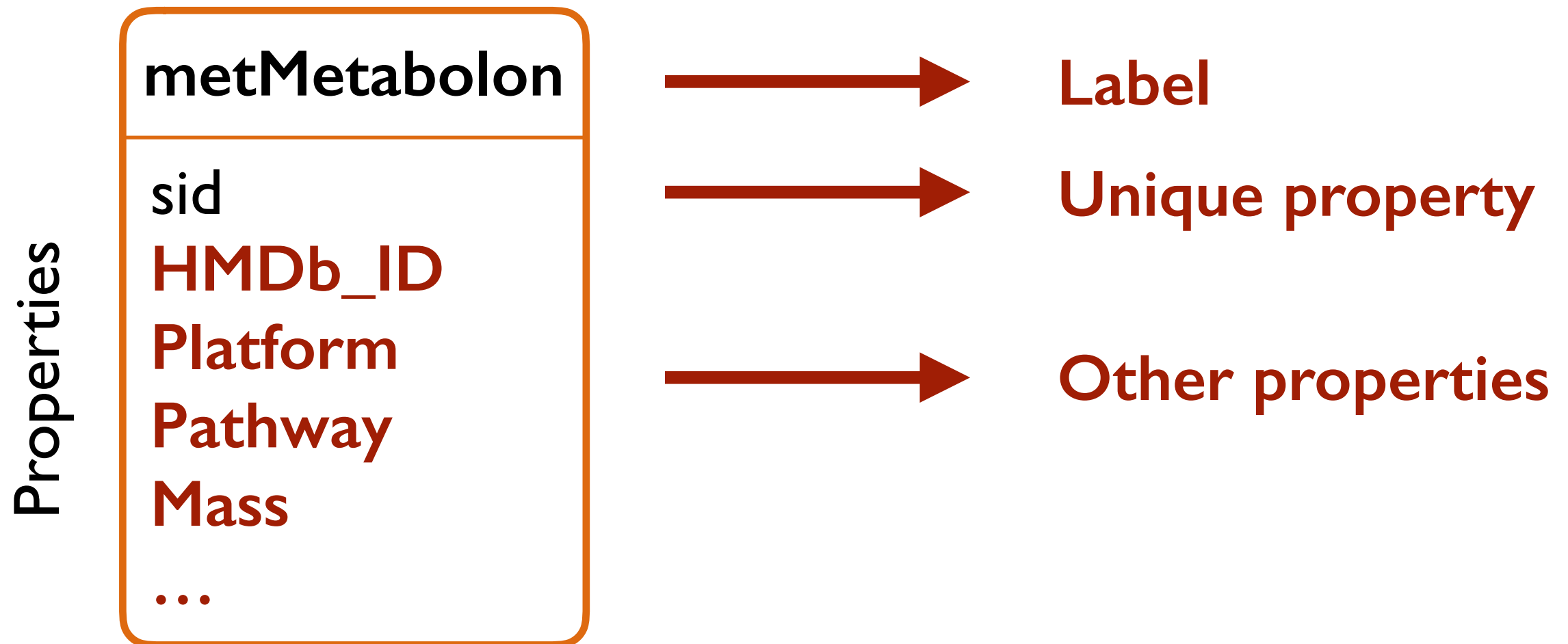
Neo4j - Nodes



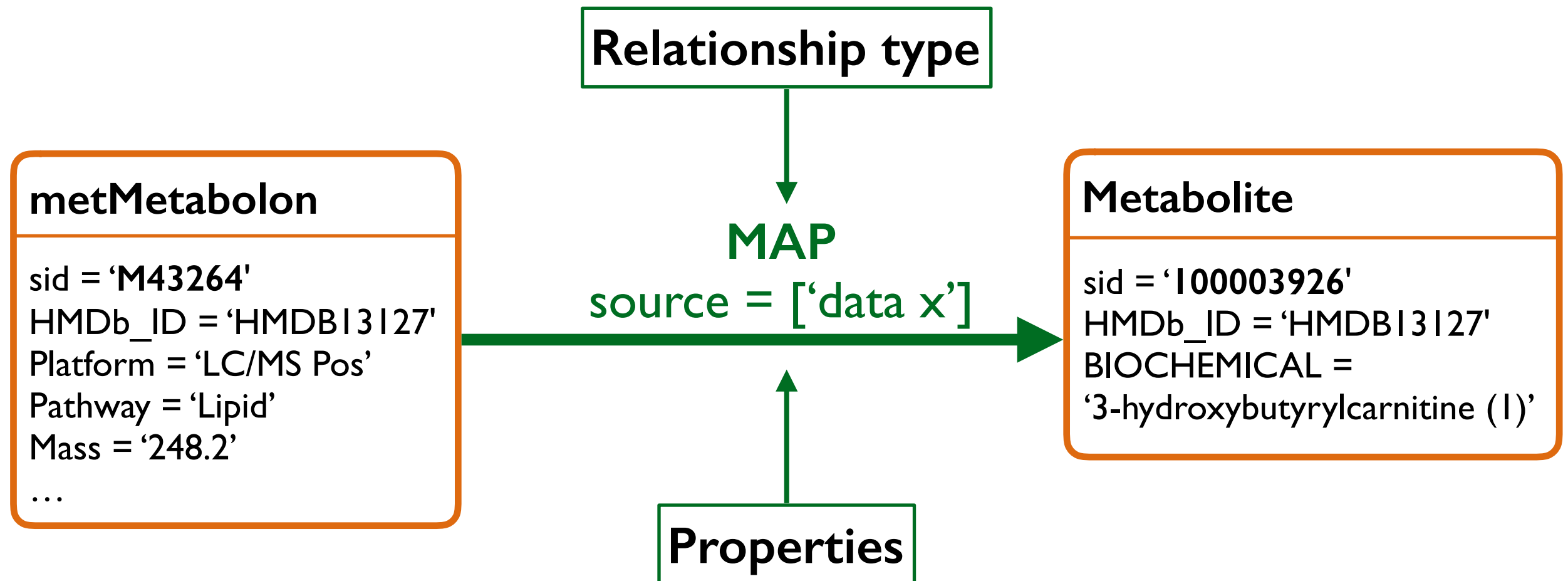
Neo4j - Nodes



Neo4j - Nodes

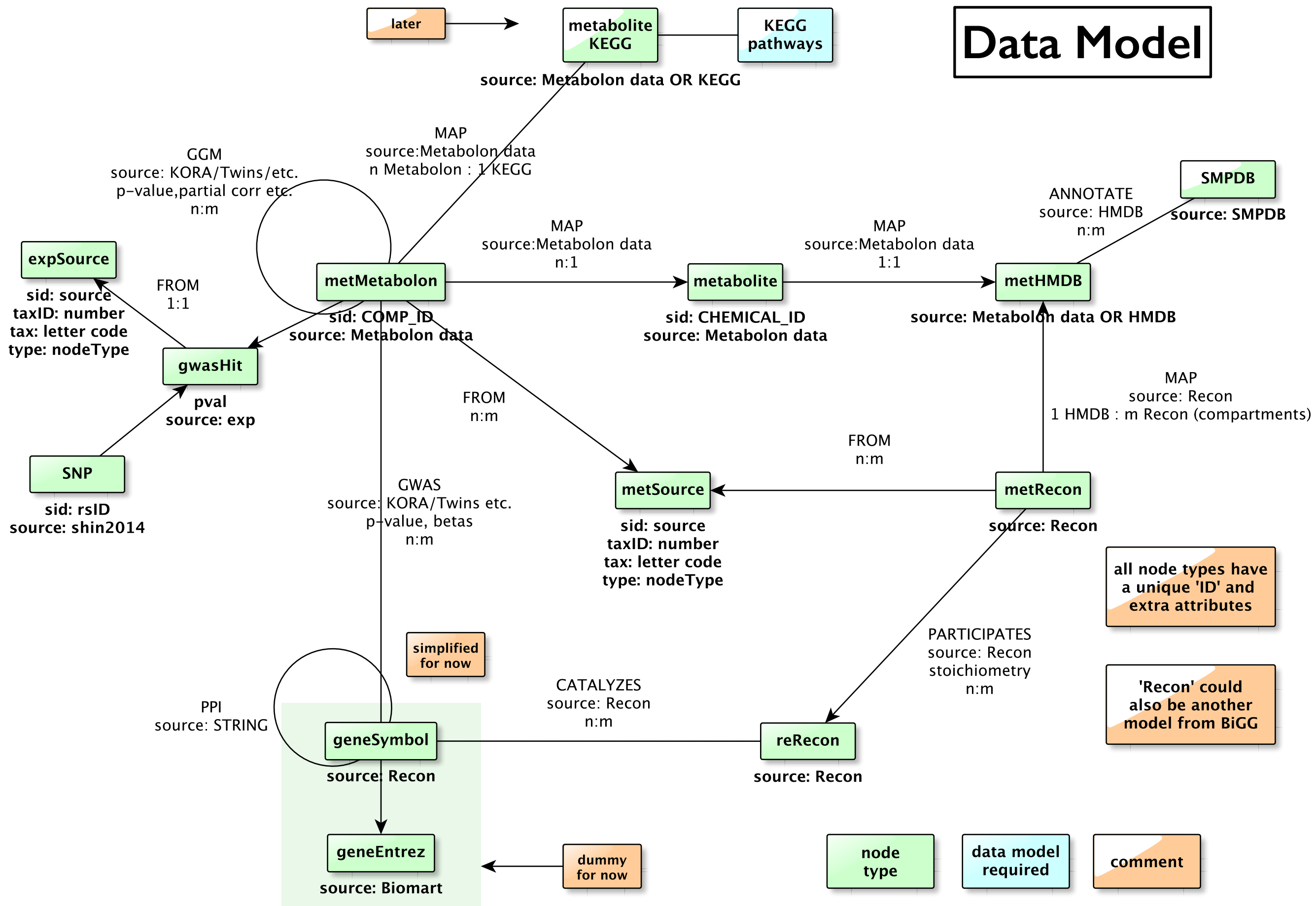


Neo4j - Relationships



- ➡ relationships always have a direction
- ➡ direction can be ignored in queries

Data Model



Cypher

- declarative graph query language
- uses patterns to describe graph data
- can be used for querying and updating
- queries built up using various clauses
 - ➡ familiar, SQL-like
 - ➡ can be chained together
 - ➡ intermediate results will be context for next clause

Queries - Simple

Match a specific node and return it:









MATCH (   **RETURN** 

Table with all reactions that a metabolite participates in:

MATCH (  **RETURN**  **AS** 

Queries - Advanced

Return all metabolites that:

- a) participate in multiple(>1) reactions,
- b) have been measured and
- c) belong to the 'Amino acid' pathway

```
MATCH (:reRecon)-[p:PARTICIPATES]-(m:metRecon)
WITH count(p) AS nr, m AS metabo
WHERE nr>1
MATCH (metabo)-[:MAP*3]-(m:metMetabolon)
WHERE m.SUPER_PATHWAY='Amino Acid'
RETURN distinct(m.sid)
```

Queries - Advanced

Return all metabolites that:

- a) participate in multiple(>1) reactions,
- b) have been measured and
- c) belong to the 'Amino acid' pathway

```
MATCH (:reRecon)-[p:PARTICIPATES]-(m:metRecon)
WITH count(p) AS nr, m AS metabo
WHERE nr>1
```

```
MATCH (metabo)-[:MAP*3]-(m:metMetabolon)
WHERE m.SUPER_PATHWAY='Amino Acid'
RETURN distinct(m.sid)
```

RNeo4j and Local DB

#install Rneo4j

```
install.packages("RNeo4j")
```

```
library(RNeo4j)
```

#connect to db

```
db = startGraph("http://ibisdb02:7475/db/data/")
```

#sample cypher query

```
query = "MATCH (h:metHMDB)-[m:MAP]-(:metabolite)
        WITH h AS hmdb, count(m) AS nr
        WHERE nr > 1
        MATCH (hmdb)-[:MAP]-(m:metabolite)
        RETURN hmdb.sid AS metabolite, collect(m.sid) AS multipleID"
```

#get results of query

```
result <- cypher(db, query)
```

#open browser view

```
browse(db)
```

	metabolite	multipleID
1	HMDB01976	100001181, 100001580

More information, tutorials etc.

Browser: <http://ibisdb02:7475/browser/>

Neo4j/Graph DB:
<https://neo4j.com/why-graph-databases/>

Cypher:
<https://neo4j.com/docs/cypher-refcard/current/>
<https://neo4j.com/docs/developer-manual/current/cypher/>

Online tutorial:
<https://neo4j.com/graphacademy/online-training/introduction-graph-databases/>

