

Lisa Mischer, Lucas Thielmann, Andreas Rau, Nico Hein

Agenda

- Einsatzgebiete
- Vor- und Nachteile
- Einführung des Beispiels
- Datenmodell
- Speicherstruktur
- DBMS Anwendungen und Fun Facts

Einsatzgebiete

- Page Rank bei Suchmaschinen
- Soziale Netzwerke
- Empfehlungssysteme
- Verkehrsinfrastruktur
- Transportplanoptimierung

Vor- und Nachteile

- Beziehungen können ohne Fremdschlüssel modelliert werden
- Effektives Speichern
- Traversierung
- Aggregation aufwändig
- Performance von Java VM
- Partitionierung kompliziert

Bücherdatenbank

Demo

Datenmodell

Graph Datenmodell erstellen

1. Nodes

- Repräsentieren Entitäten des Modells
- Identifizieren Identitäten eindeutig

2. Labels

- Gruppieren Nodes in Sets
- Identifizieren Rollen der Identitäten

Book

Author

Publisher

derGrosseGatsby

fScottFitzgerald

reclam

derGrosseGatsby

fScottFitzgerald

reclam

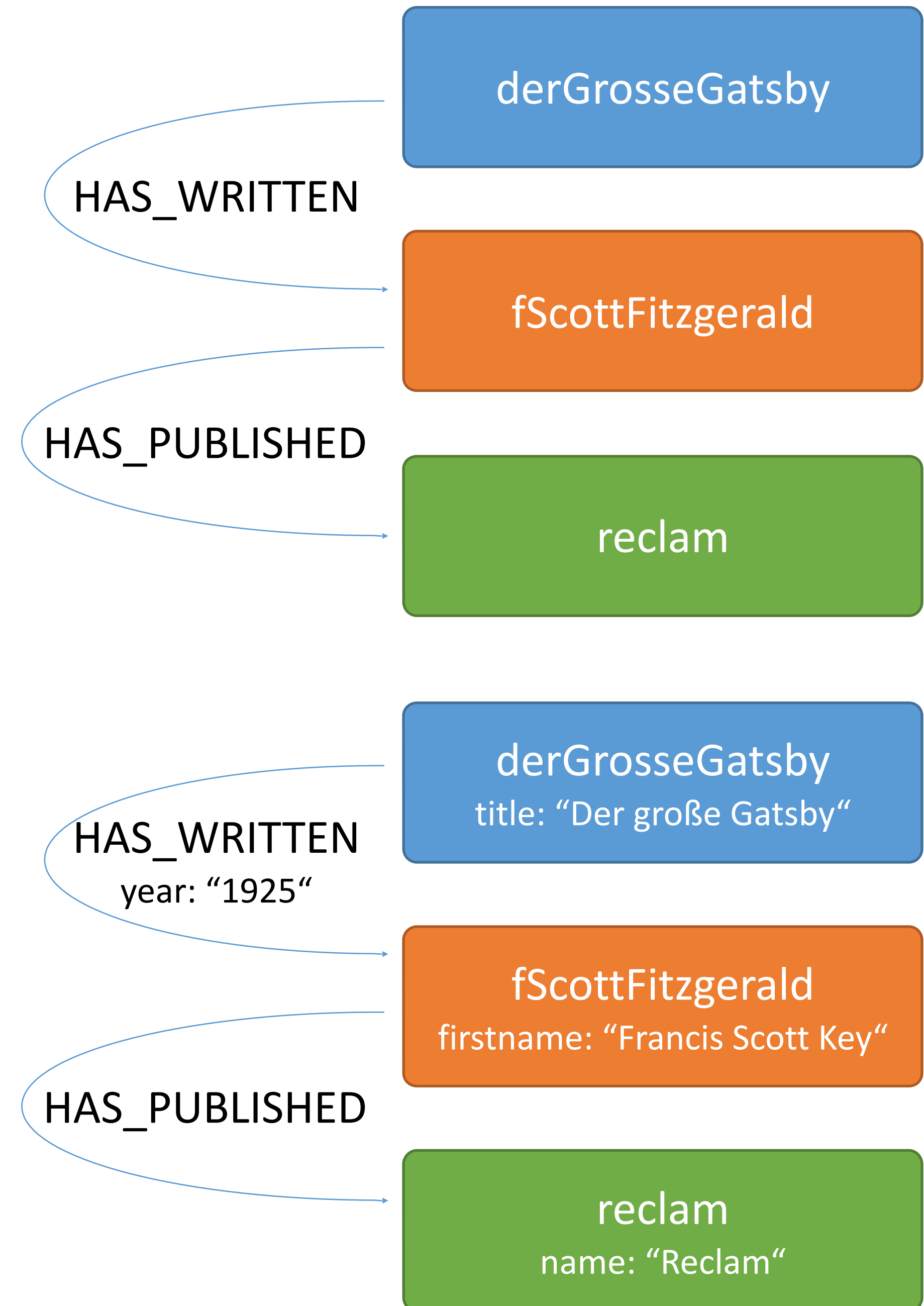
Datenmodell

3. Relationen

- Definieren Relationen zwischen Nodes
- Identifizieren Verbindungen zwischen Nodes

4. Anreichern mit Daten (Properties)

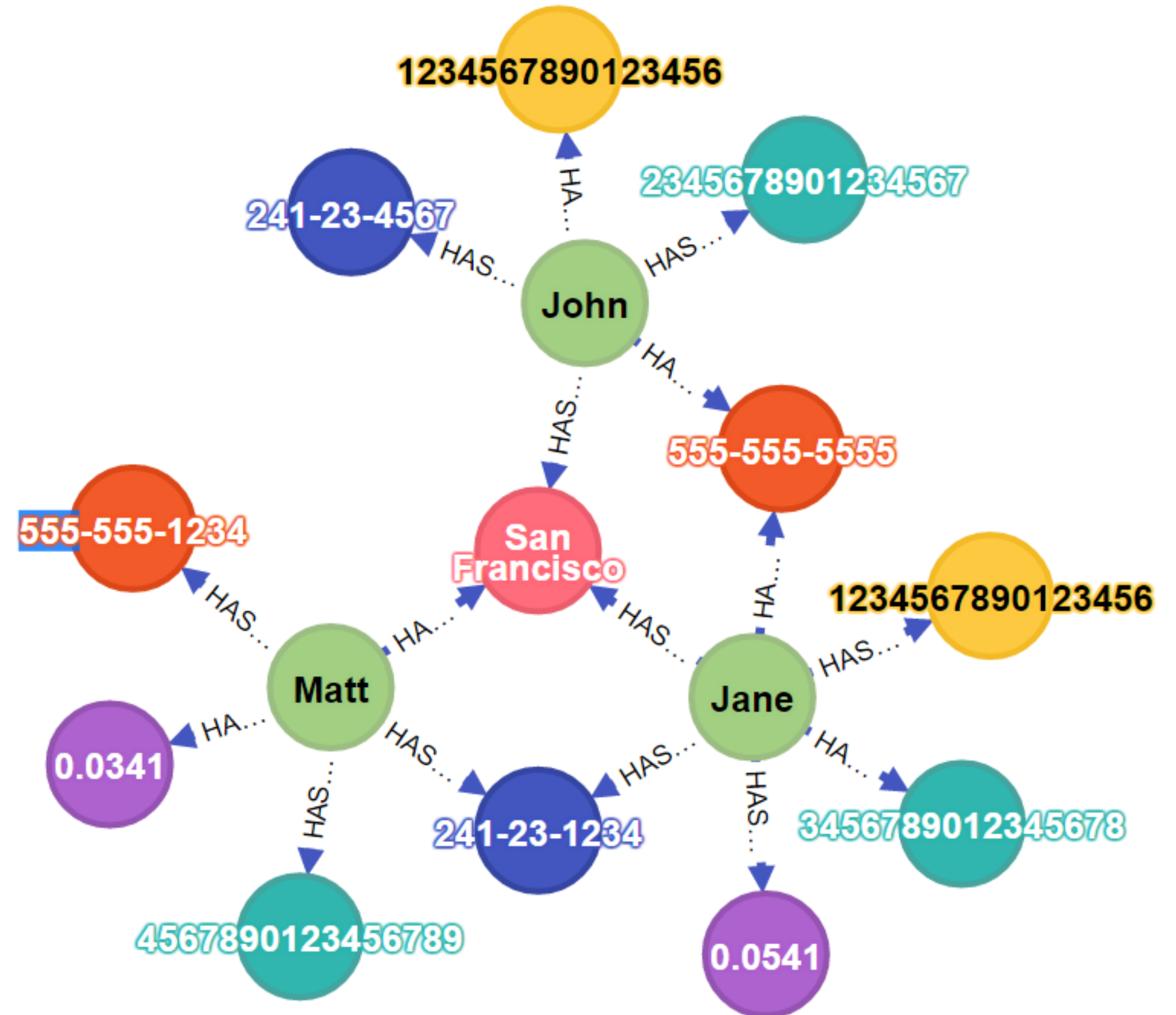
- Daten zur Datenbank hinzufügen
- Können zu Nodes und Relationen hinzugefügt werden



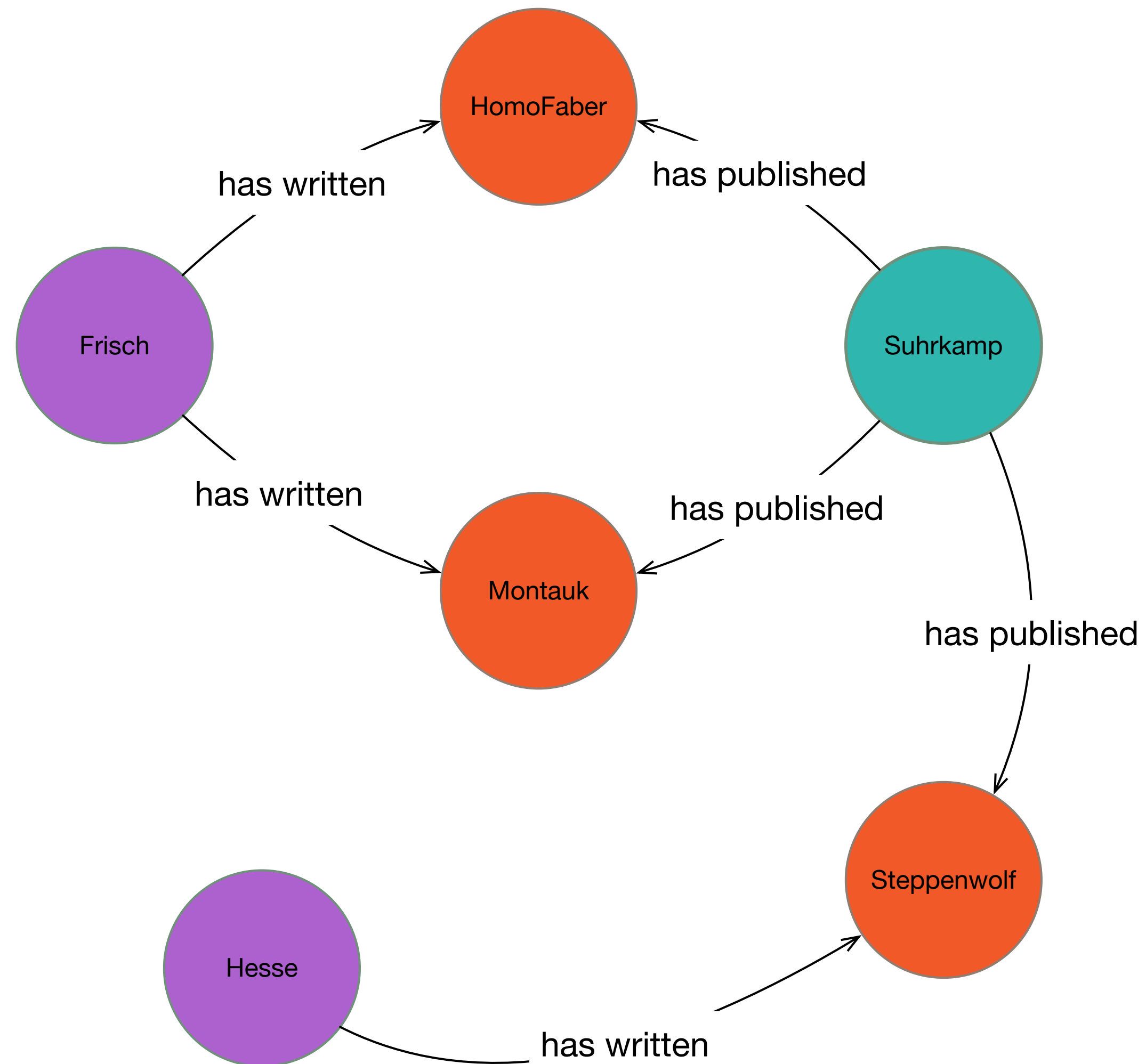
Datenmodell

GraphGist zur Dokumentation des Datenmodells

- Geschrieben in Markup (textbasiert)
- Im Browser gerendert
- Beinhaltet Beispieldaten
- Interaktive Konsole
- Beispiel

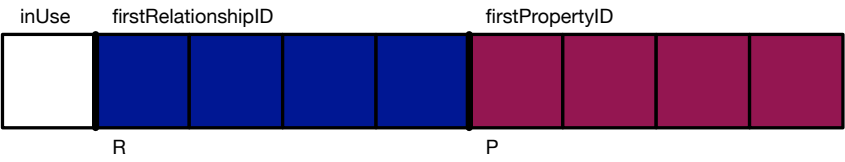


Speicherstruktur

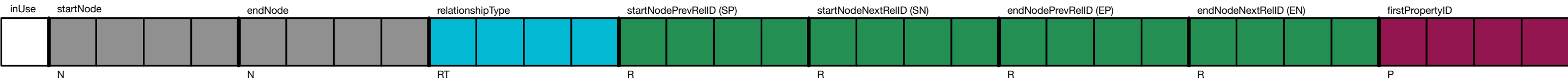


Speicherstruktur

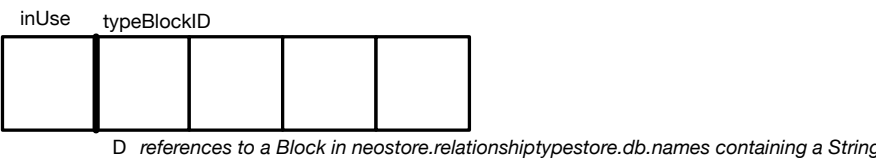
Node *N* (9 bytes) - stored in *neostore.nodestore.db*



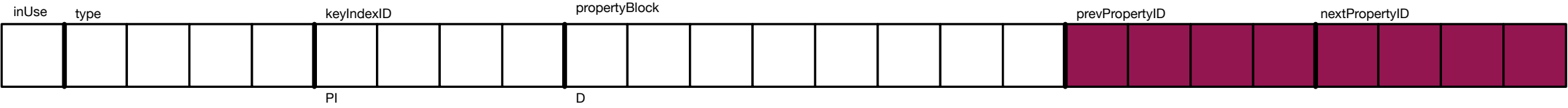
Relationship *R* (33 bytes) - stored in *neostore.relationshipstore.db*



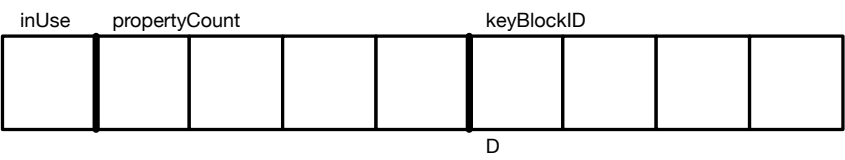
Relationship Type *RT* (5 bytes) - stored in *neostore.relationshiptypestore.db*



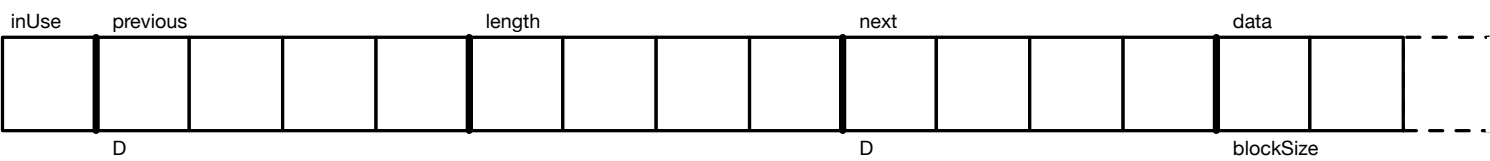
Property *P* (25 bytes)



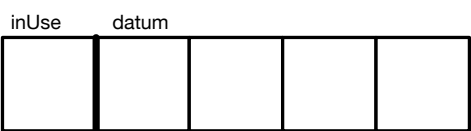
Property Index *PI* (9 bytes)



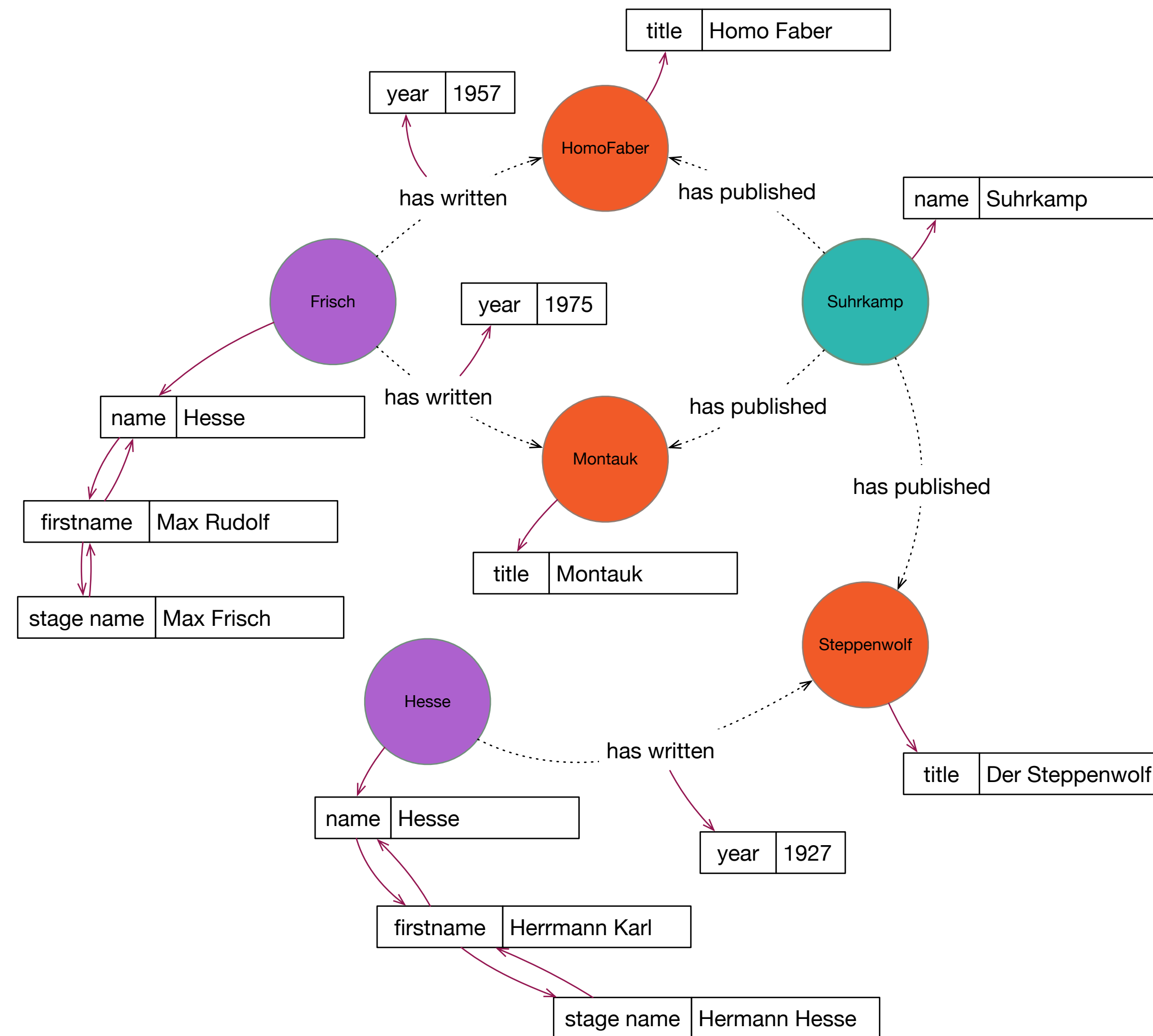
Dynamic Store *D* (13+blockSize bytes)



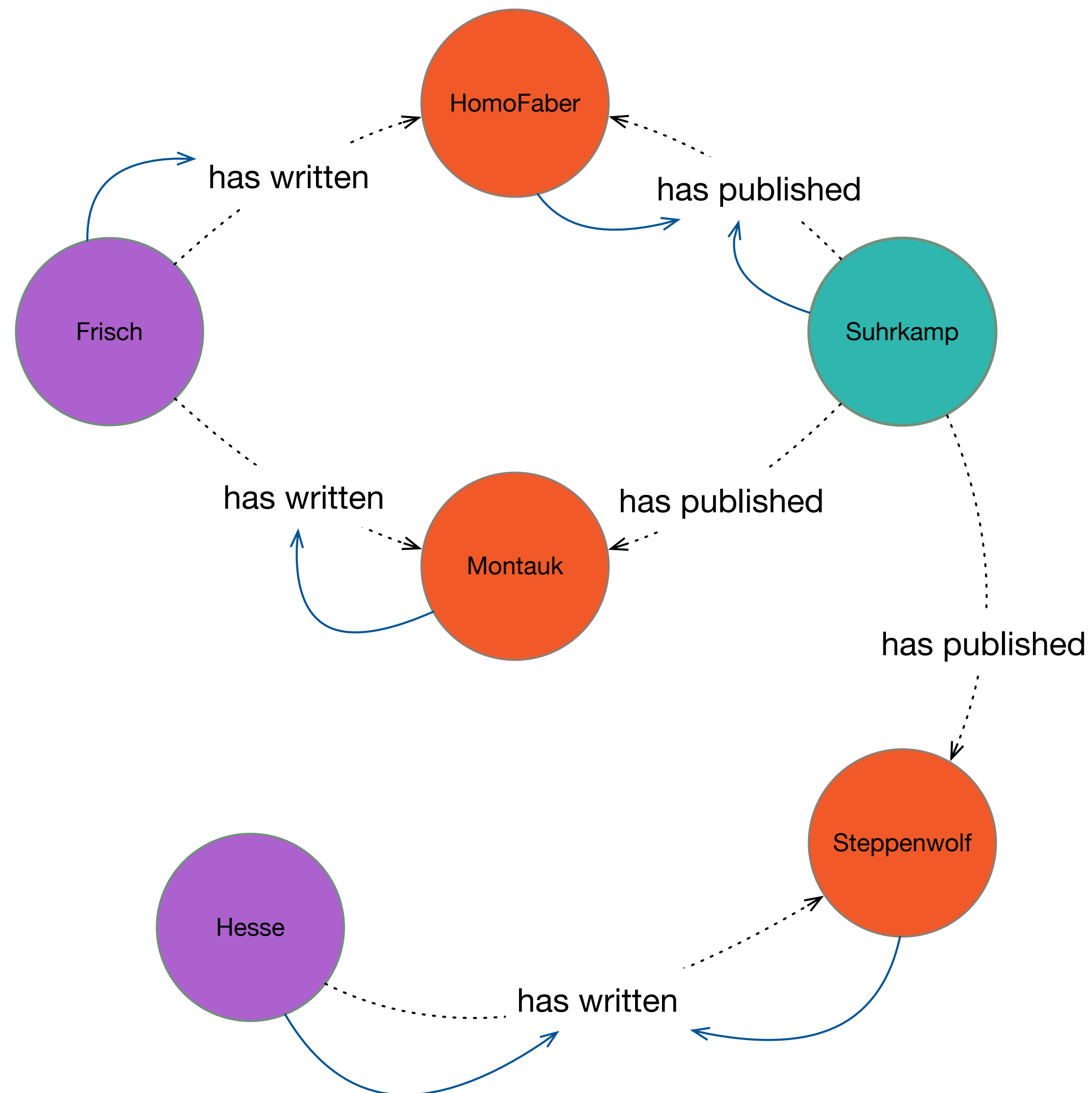
NeoStore (5 bytes)



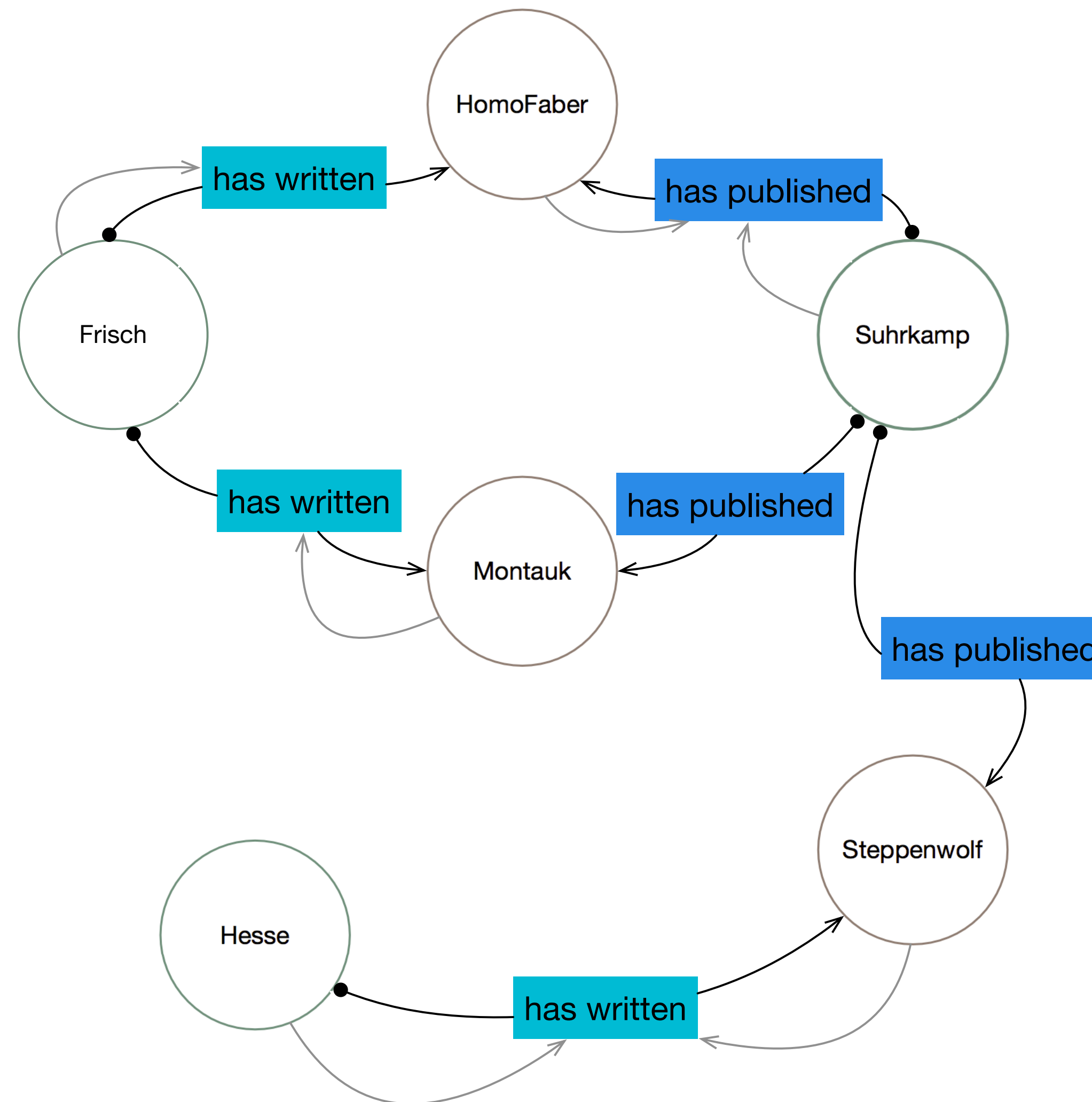
Speicherstruktur



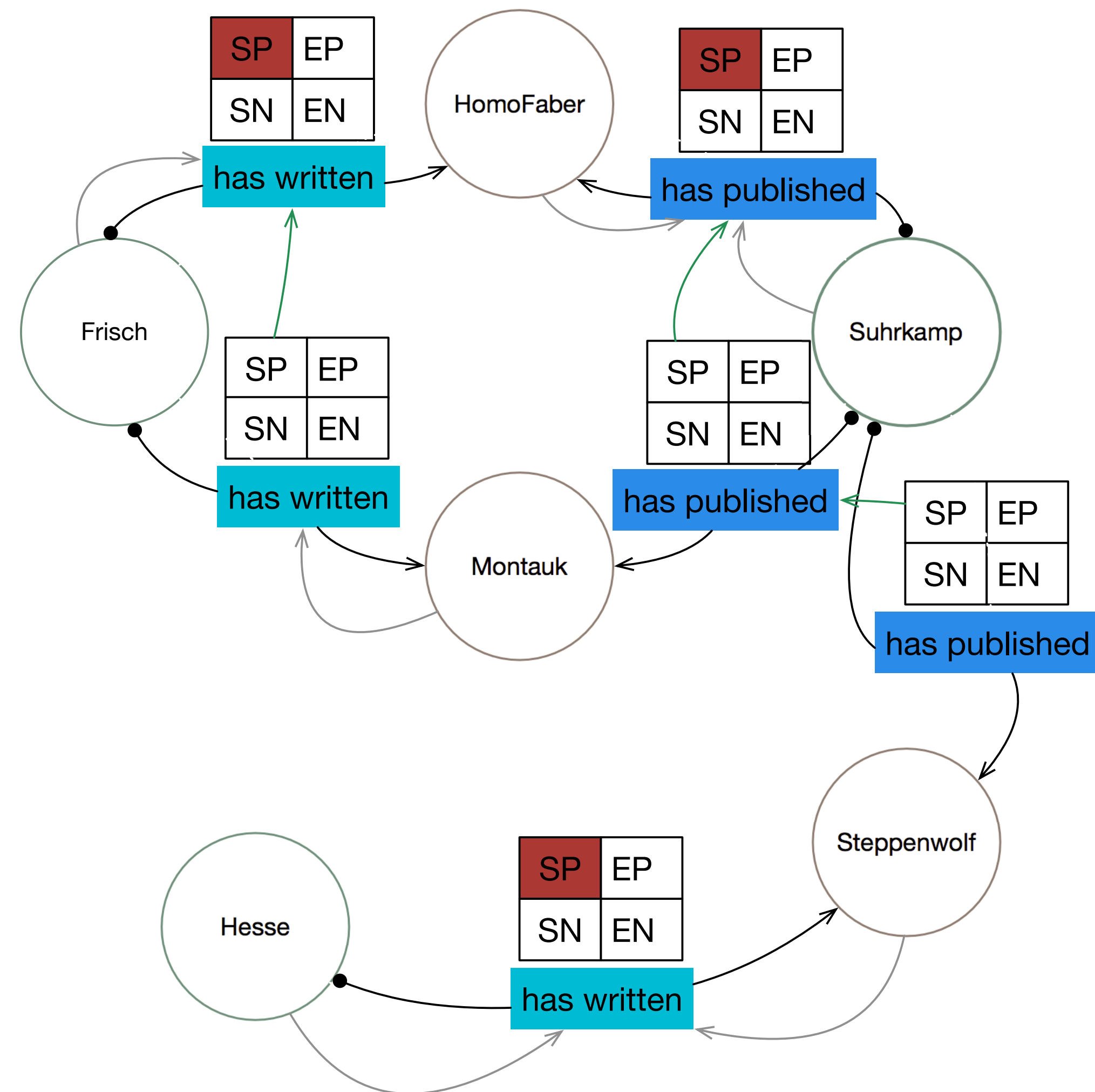
Speicherstruktur



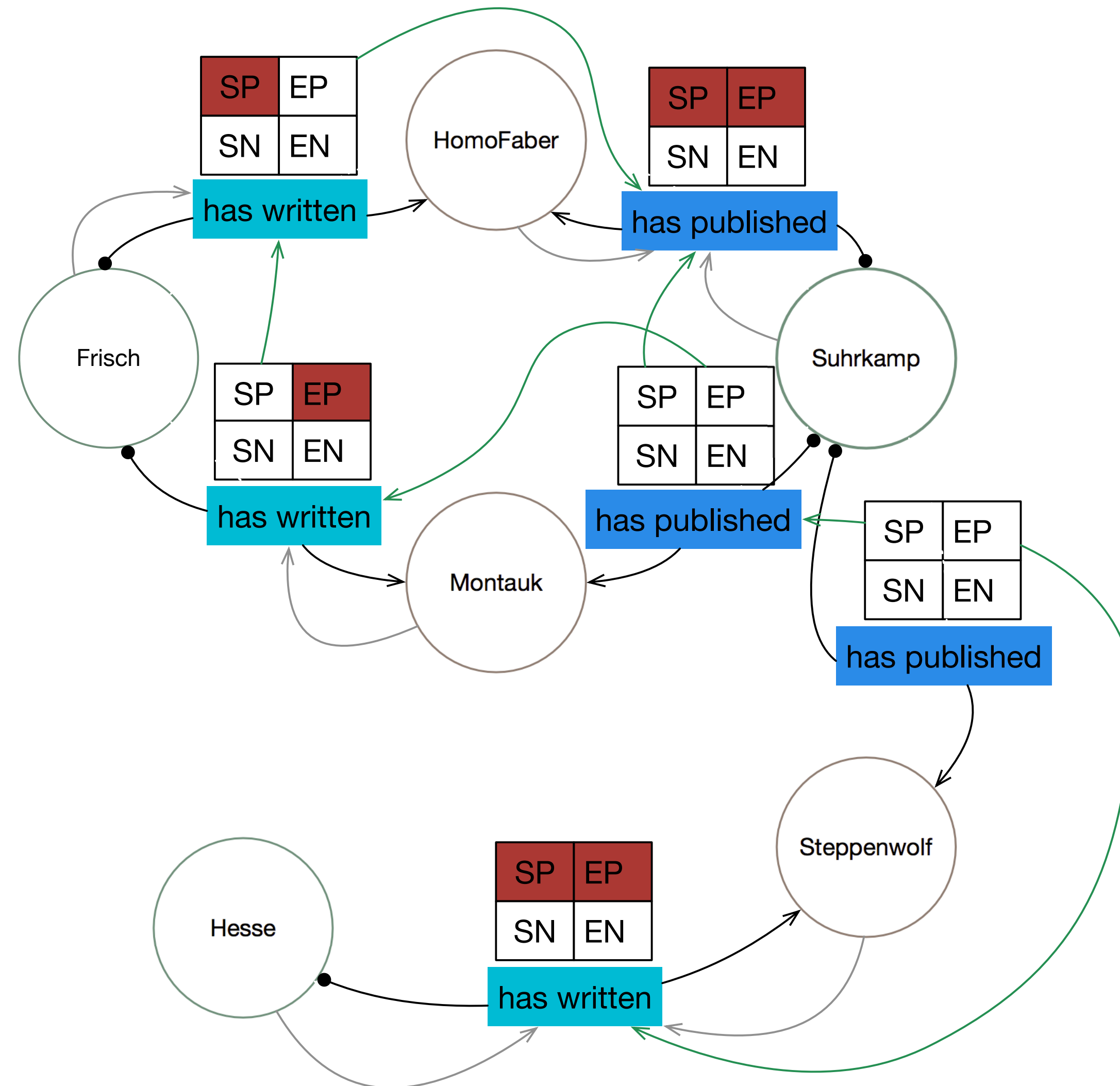
Speicherstruktur



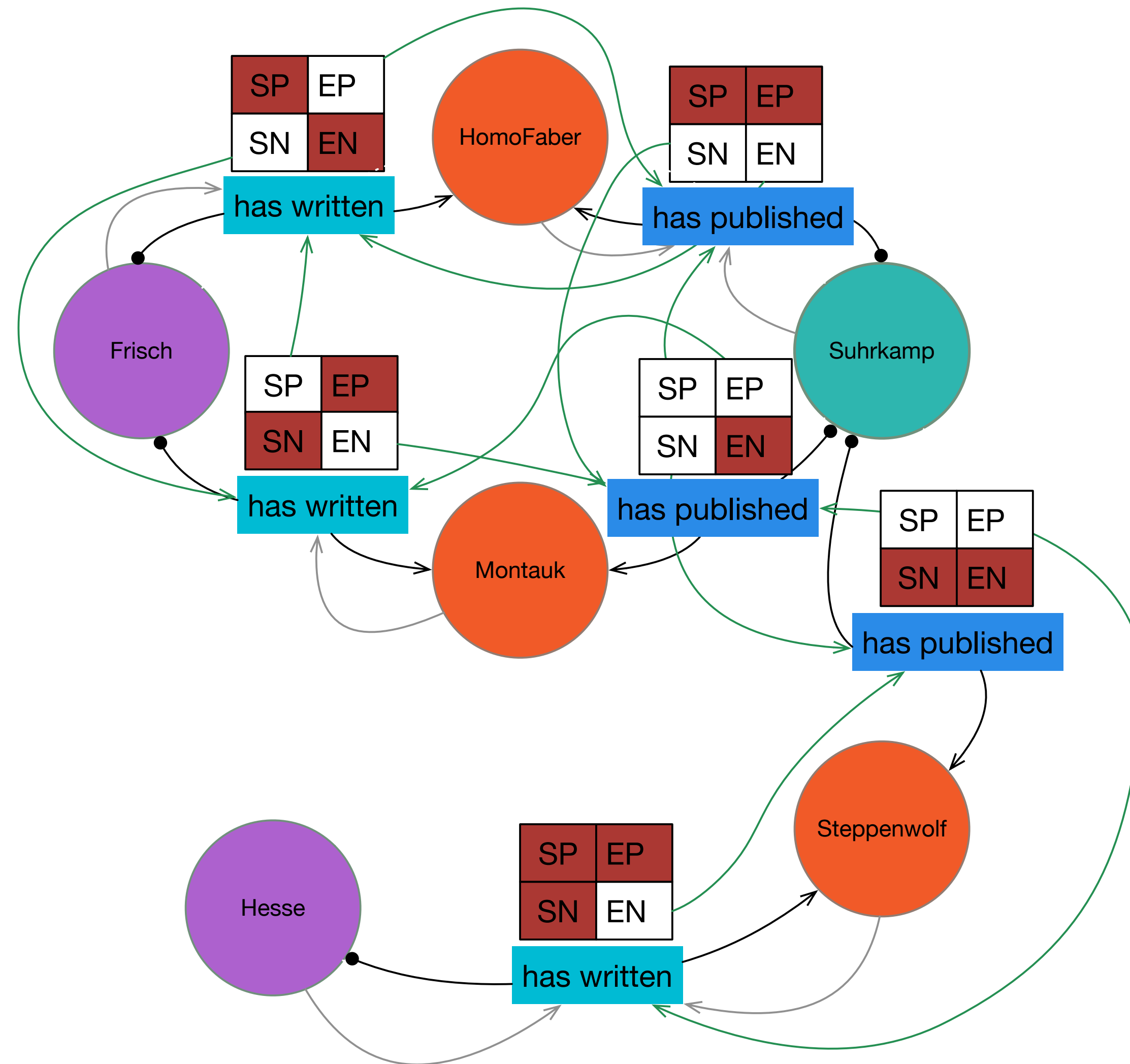
Speicherstruktur



Speicherstruktur

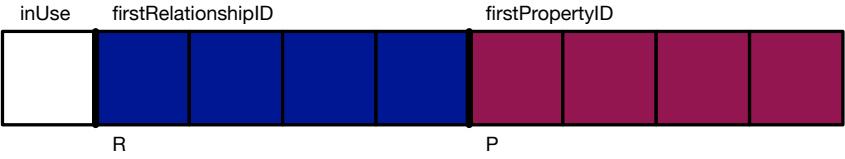


Speicherstruktur

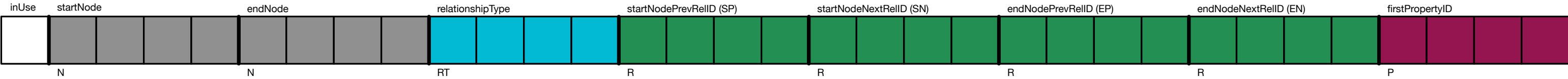


Speicherstruktur

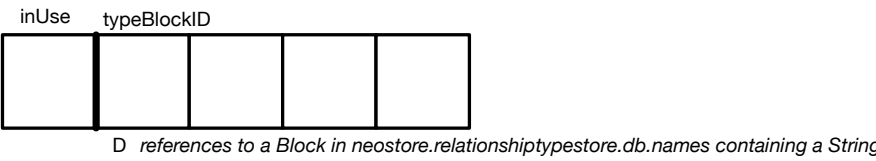
Node *N* (9 bytes) - stored in *neostore.nodestore.db*



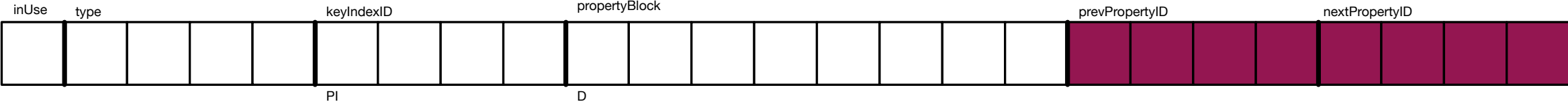
Relationship *R* (33 bytes) - stored in *neostore.relationshipstore.db*



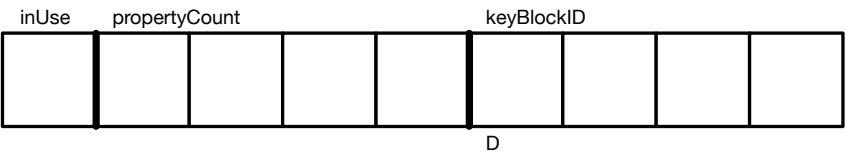
Relationship Type *RT* (5 bytes) - stored in *neostore.relationshiptypestore.db*



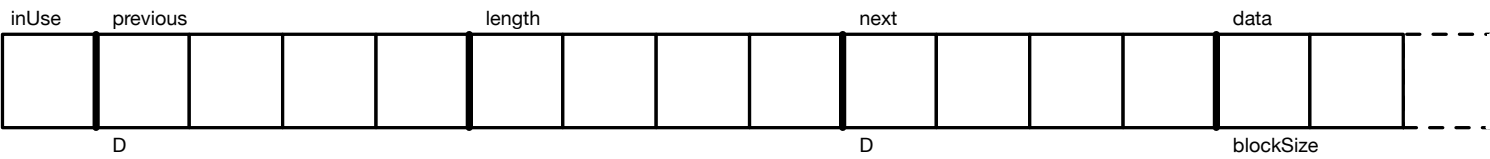
Property *P* (25 bytes)



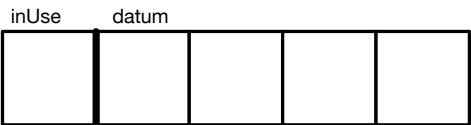
Property Index *PI* (9 bytes)



Dynamic Store *D* (13+blockSize bytes)



NeoStore (5 bytes)

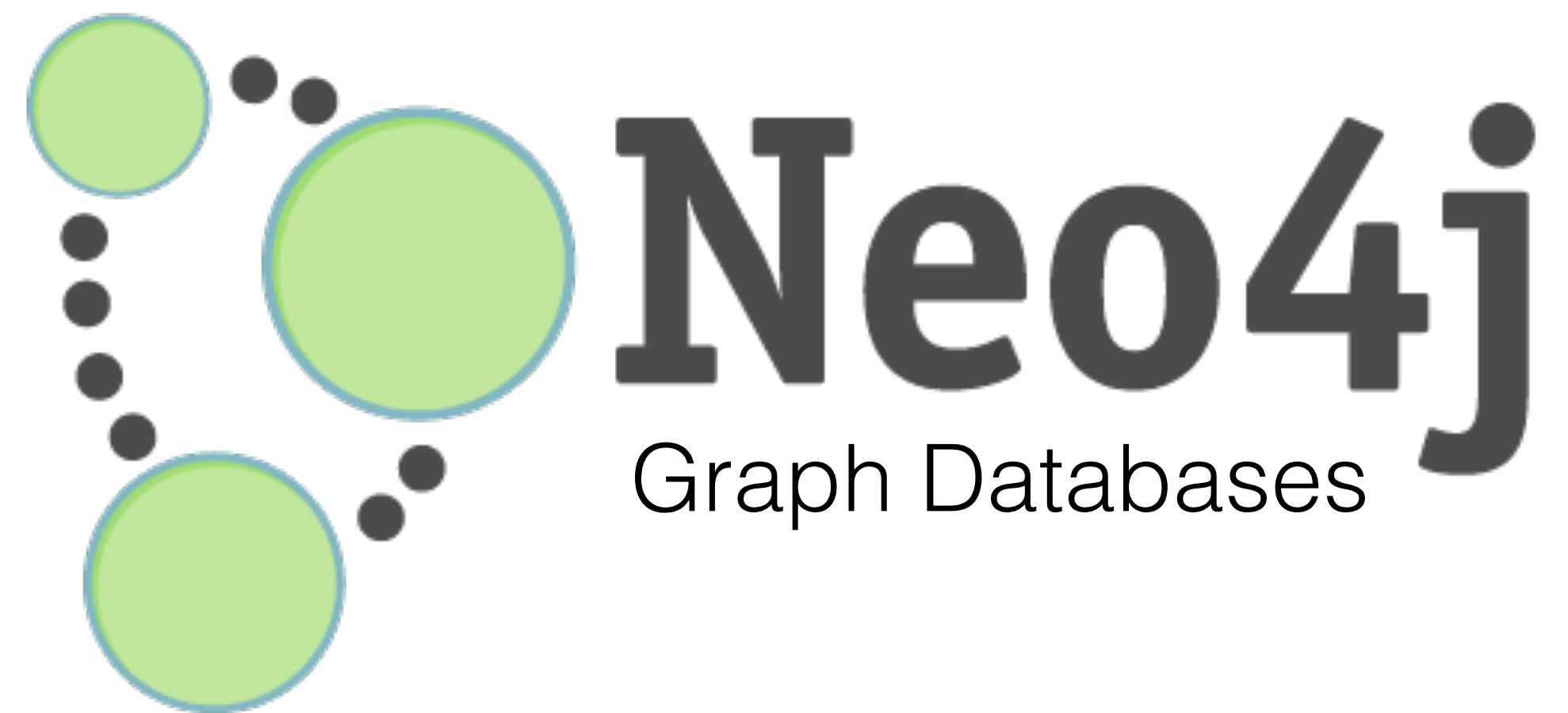


Business Intelligence

- Vor allem nützlich wenn keine anderen Strukturen zum festhalten der Daten möglich sind (hierarchisch, Netzwerke, etc.)
- Graphentheorie kann angewendet werden um analytische Prozesse zu unterstützen
- OLAP-ähnliche Aggregation möglich -> DW

Fun fact

- Ebay und Walmart benutzen Neo4J



Lisa Mischer, Lucas Thielmann, Andreas Rau, Nico Hein