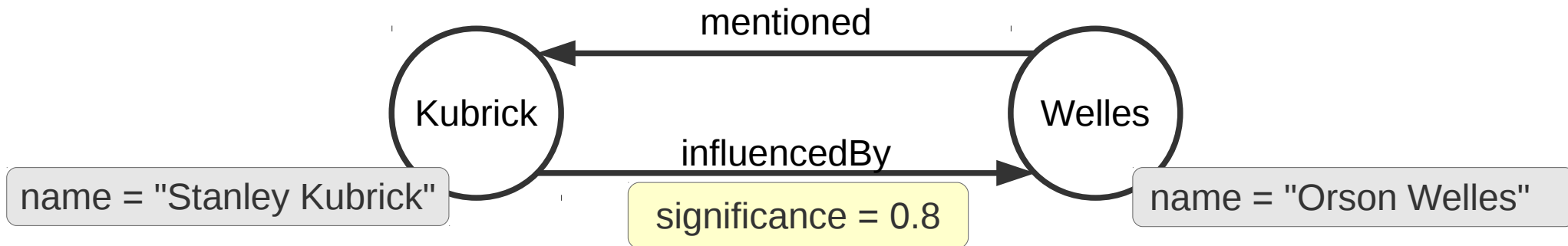
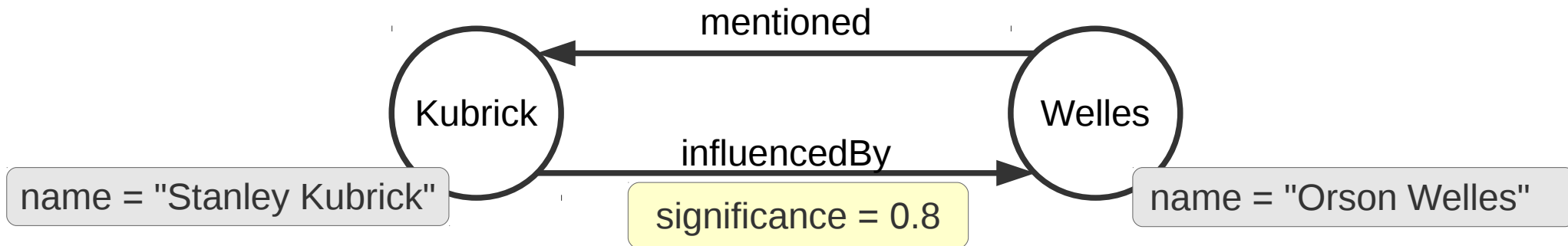


Annotations of Edges in Graph Data

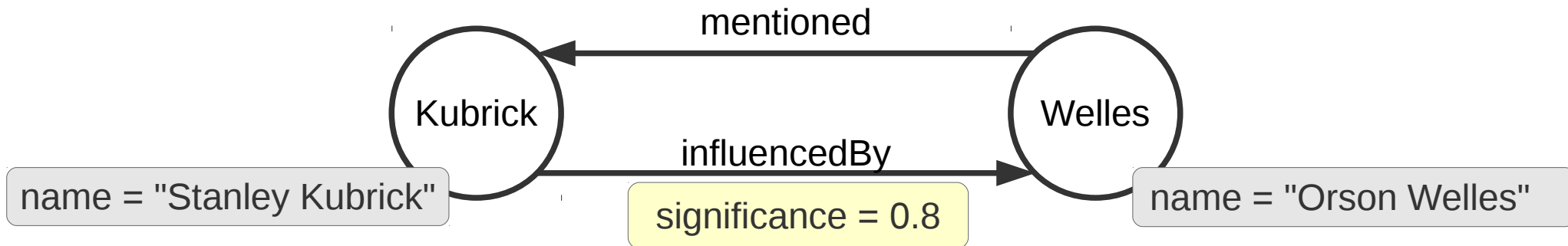


Main Use Case: Statement-Level Metadata



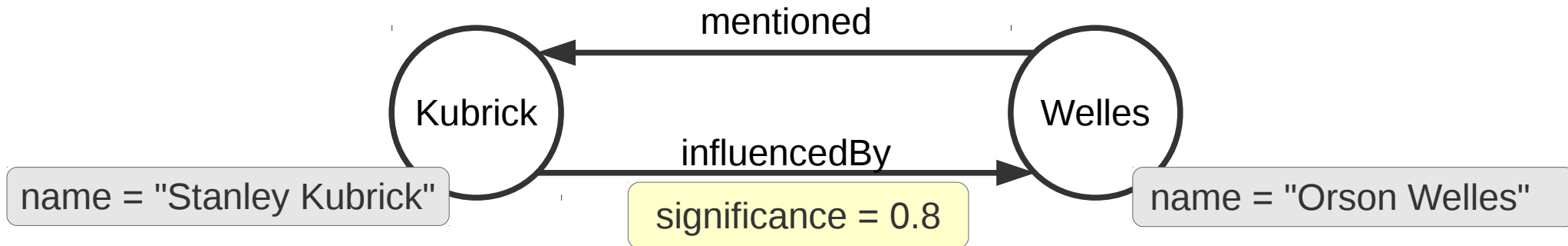
- Certainty scores
- Weights
- Temporal restrictions
- Provenance information
- etc.

Not supported natively in the RDF data model



- RDF triples: Welles name "Orson Welles" .
Welles mentioned Kubrick .
Kubrick name "Stanley Kubrick" .
Kubrick influencedBy Welles .
??? significance 0.8 .

Standard RDF Reification



```
Welles name "Orson Welles" .
Welles mentioned Kubrick .
Kubrick name "Stanley Kubrick" .
Kubrick influencedBy Welles .
s significance 0.8 .
s rdf:type rdf:Statement .
s rdf:subject Kubrick .
s rdf:predicate influencedBy .
s rdf:object Welles .
```

Queries?

Example 1:

List all people that Welles had a significant influence on.

```
SELECT ?x WHERE {  
    ?x          influencedBy  Welles .  
    ?t  significance  ?sig .  
    ?t  rdf:type      rdf:Statement .  
    ?t  rdf:subject   ?x .  
    ?t  rdf:predicate influencedBy .  
    ?t  rdf:object    Welles .  
    FILTER ( ?sig > 0.7 )  
}
```

Queries?

Example 2:

```
SELECT ?x WHERE {  
    ?x influencedBy Welles .  
    Welles influencedBy ?x .  
  
    ?t1 rdf:type      rdf:Statement .  
    ?t1 rdf:subject   ?x .  
    ?t1 rdf:predicate influencedBy .  
    ?t1 rdf:object    Welles .  
  
    ?t1 significance ?sig1 .  
  
    ?t2 rdf:type      rdf:Statement .  
    ?t2 rdf:subject   Welles .  
    ?t2 rdf:predicate influencedBy .  
    ?t2 rdf:object    ?x .  
  
    ?t2 significance ?sig2 .  
  
    FILTER (?sig1 > 0.7 && ?sig2 > 0.7)  
}
```



Other Proposals: Single-Triple Named Graphs

- Example: `g1 { Kubrick influencedBy Welles }`
`g1 significance 0.8 .`

– Query:

```
SELECT ?x WHERE {  
    GRAPH ?g {  
        ?x influencedBy Welles  
    }  
    ?g significance ?sig .  
    FILTER ( ?sig > 0.7 )  
}
```

Other Proposals: Singleton Properties

- Example:

```
Kubrick    influencedBy    Welles .  
Kubrick    p1              Welles .  
p1         singletonPropertyOf influencedBy .  
p1         significance    0.8 .
```

- Query:

```
SELECT ?x WHERE {  
  ?x    influencedBy    Welles .  
  ?x    ?p              Welles .  
  ?p    singletonPropertyOf influencedBy .  
  ?p    significance    ?sig .  
  FILTER ( ?sig > 0.7 )  
}
```


The RDF* and SPARQL* Approach to Annotate Statements in RDF and to Reconcile RDF and Property Graphs

Olaf Hartig

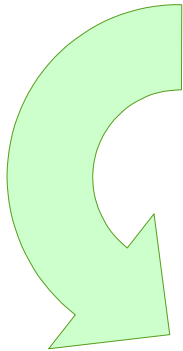
@olafhartig



Our Proposal: Nested Triples



~~Kubrick influencedBy Welles .
s rdf:type rdf:Statement .
s rdf:subject Kubrick .
s rdf:predicate influencedBy .
s rdf:object Welles .
s significance 0.8 .~~



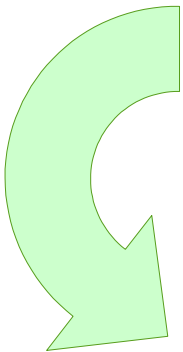
<<Kubrick influencedBy Welles>> significance 0.8

subject *predicate* *object*

... and Nested Triple Patterns



```
SELECT ?x WHERE {  
  ?x influencedBy Welles .  
  ?t significance ?sig .  
  ?t rdf:type rdf:Statement .  
  ?t rdf:subject ?x .  
  ?t rdf:predicate influencedBy .  
  ?t rdf:object Welles .  
  FILTER ( ?sig > 0.7 )  
}
```



```
SELECT ?x WHERE {  
  <<?x influencedBy Welles>> significance ?sig  
  FILTER (?sig > 0.7)  
}
```

Grouping of Patterns with the Same Subject

- By the standard SPARQL syntax, we may write:

```
SELECT ?x WHERE {  
  ?x influencedBy Welles .  
  ?t significance ?sig ;  
  rdf:type rdf:Statement ;  
  rdf:subject ?x ;  
  rdf:predicate influencedBy ;  
  rdf:object Welles .  
  FILTER ( ?sig > 0.7 )  
}
```

- Hence, we may easily query for multiple metadata triples:

```
SELECT ?x ?sig ?src WHERE {  
  <<?x influencedBy Welles>> significance ?sig ;  
                                   source      ?src . }
```

Extension of the BIND Clause

- Assign matching triples to variables:

```
SELECT ?x ?sig ?src WHERE {  
  BIND(<<?x influencedBy Welles>> AS ?t)  
  ?t  significance ?sig ;  
      source       ?src .  
}
```

```
SELECT ?x ?sig ?src WHERE {  
  <<?x influencedBy Welles>> significance ?sig ;  
                                source       ?src . }
```

Extension of the BIND Clause

- Assign matching triples to variables:

```
SELECT ?x ?sig ?src WHERE {  
  BIND(<<?x influencedBy Welles>> AS ?t)  
  ?t    significance ?sig ;  
       source       ?src .  
}
```

- Now, we may even output triples in query results:

```
SELECT ?t ?c WHERE {  
  BIND(<<?x influencedBy Welles>> AS ?t)  
  ?t    certainty    ?c .  
}
```

Example Query 2 Revisited

```
SELECT ?x WHERE {  
  ?x influencedBy Welles .  
  Welles influencedBy ?x .  
  
  ?t1 rdf:type      rdf:Statement .  
  ?t1 rdf:subject   ?x .  
  ?t1 rdf:predicate influencedBy .  
  ?t1 rdf:object    Welles .  
  
  ?t1 significance ?sig1 .  
  
  ?t2 rdf:type      rdf:Statement .  
  ?t2 rdf:subject   Welles .  
  ?t2 rdf:predicate influencedBy .  
  ?t2 rdf:object    ?x .  
  
  ?t2 significance ?sig2 .  
  
  FILTER (?sig1 > 0.7 && ?sig2 > 0.7)  
}
```

Example Query 2 Revisited

```
SELECT ?x WHERE {  
  ?x influencedBy Welles  
  Welles influencedBy ?x .  
  ?t1 rdf:type rdf:Statement .  
  ?t1 rdf:subject ?x .
```

```
SELECT ?x WHERE {  
  <<?x influencedBy Welles>> significance ?sig1 .  
  <<Welles influencedBy ?x>> significance ?sig2 .  
  FILTER (?sig1 > 0.7 && ?sig2 > 0.7)  
}
```

```
?t2 rdf:predicate influencedBy .  
?t2 rdf:object ?x .  
?t2 significance ?sig2 .  
FILTER (?sig1 > 0.7 && ?sig2 > 0.7)  
}
```


Two Perspectives on RDF* and SPARQL*

1. Syntactic sugar on top of standard RDF and SPARQL

- Can be parsed directly into pure RDF and SPARQL
- Can be implemented easily by a small wrapper on top of any existing RDF DBMS

2. A logical model in its own right

- Extension of the RDF data model and of SPARQL to capture the notion of nested triples
- May be implemented using a dedicated physical schema (e.g., Blazegraph)



Existing Contributions in Detail

1. Syntactic sugar on top of standard RDF and SPARQL

2. A logical model in its own right

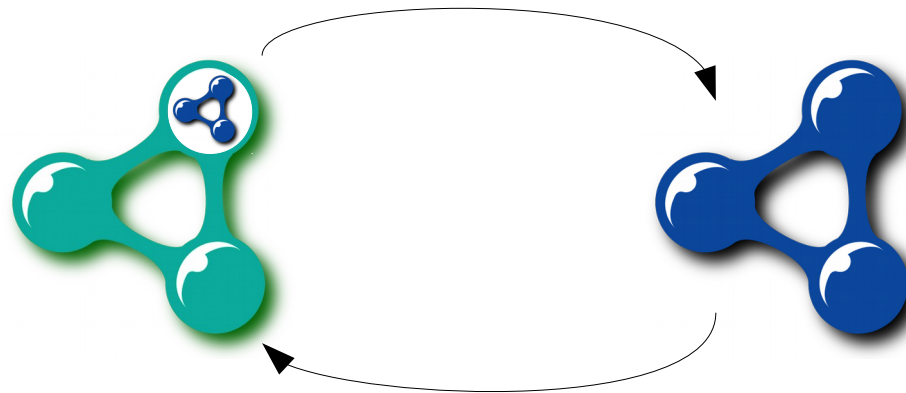


- Definition of the Turtle* file format
 - extends standard Turtle format with support for nested triples
- Definition of the SPARQL* syntax
 - extends the SPARQL syntax with nested triple patterns

Contributions in the Context of Perspective 1

1. Syntactic sugar on top of standard RDF and SPARQL

2. A logical model in its own right



- Definition of desirable properties of RDF*-to-RDF mappings
 - *information preservation* and *query result preservation*
- Definition of RDF reification related mappings and proof that they possess the desirable properties

Contributions in the Context of Perspective 2

1. Syntactic sugar on top of
standard RDF and SPARQL

2. A logical model
in its own right



- Formalization of the RDF* data model
 - extends the RDF data model with the notions of an *RDF* triple* and an *RDF* graph*
- Definition of formal semantics of SPARQL*
 - both the query part of the language and the data manipulation part (SPARQL Update)
- Results regarding redundancy in RDF* graphs
 - example:

```
<<Kubrik influencedBy Welles>> certainty 0.8 .  
Kubrik influencedBy Welles ←.....redundant
```

References

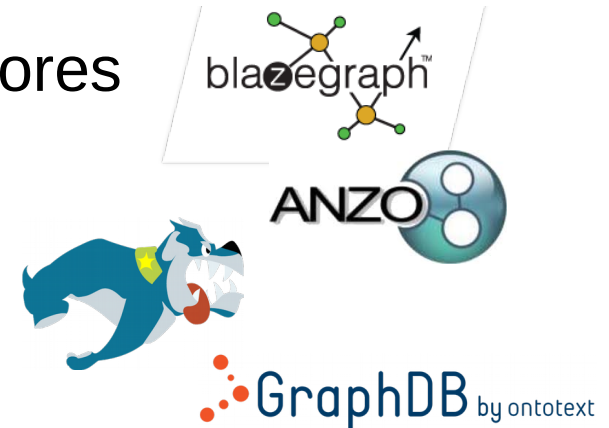
- [1] Olaf Hartig and Bryan Thompson: **“Foundations of an Alternative Approach to Reification in RDF.”** CoRR abs/1406.3399, 2014
 - Specifies Turtle*
 - All relevant extensions of the SPARQL spec
 - [2] Olaf Hartig: **“SPARQL* Update.”** 2014, Online at <http://blog.liu.se/olafhartig/documents/sparql-update/>
 - All relevant extensions of the SPARQL Update spec
 - [3] Olaf Hartig: **“Foundations of RDF* and SPARQL* - An Alternative Approach to Statement-Level Metadata in RDF.”** In Proc. of 11th Alberto Mendelzon Int. W. on Foundations of Data Mgmt (AMW), 2017
 - RDF*-to-RDF mappings and their properties,
 - Redundancy-related results
 - [4] Olaf Hartig: **“Reconciliation of RDF* and Property Graphs.”** In CoRR abs/1409.3288, 2014
 - Formal definition of direct mappings b/w RDF* and Property Graphs
 - [5] Olaf Hartig: **“Foundations to Query Labeled Property Graphs using SPARQL*.”** In Proc. of the 1st Int. Workshop on Approaches for Making Data Interoperable (AMAR), 2019
-

Hold on!

... is this all theory only???

Implementations

- Support in multiple commercial triple stores
 - Blazegraph (“Reification done right”)
 - Cambridge Semantics’ AnzoGraph
 - Stardog
 - OntoText’s GraphDB
- Other vendors have indicated interest in supporting and standardizing RDF* & SPARQL*
- Support in open source tools
 - Apache Jena (Java)
 - Eclipse RDF4J (Java)
 - RDF.rb (Ruby)
 - N3.js (Javascript)
 - RDFLib (Python)



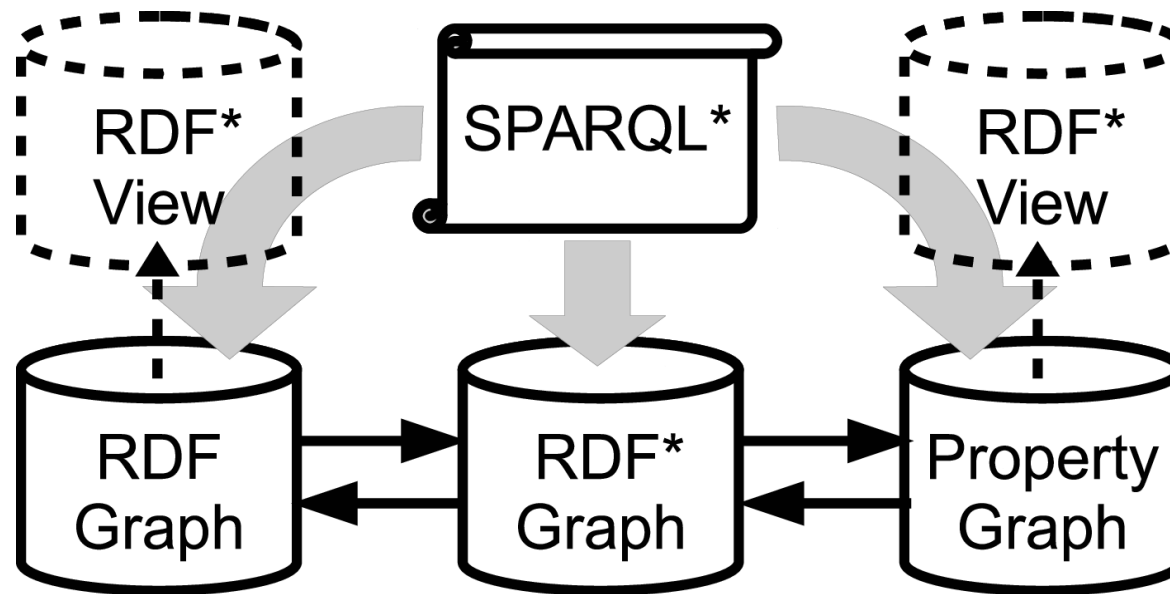
Ongoing Work 1: Modes of Usage

- **Property Graph mode (*PG mode*)**: every nested triple implicitly captures also all the triples contained in it
 - like in a PG, you cannot have an edge property w/o an edge
- As defined so far
- Example: `SELECT ?x WHERE { Kubrik influencedBy ?x }`
 - data `<<Kubrik influencedBy Welles>> certainty 0.8 .`
 - query result *one mapping with ?x → Welles*
 - data `<<Kubrik influencedBy Welles>> certainty 0.8 .`
`Kubrik influencedBy Welles .`
 - query result *one mapping with ?x → Welles*

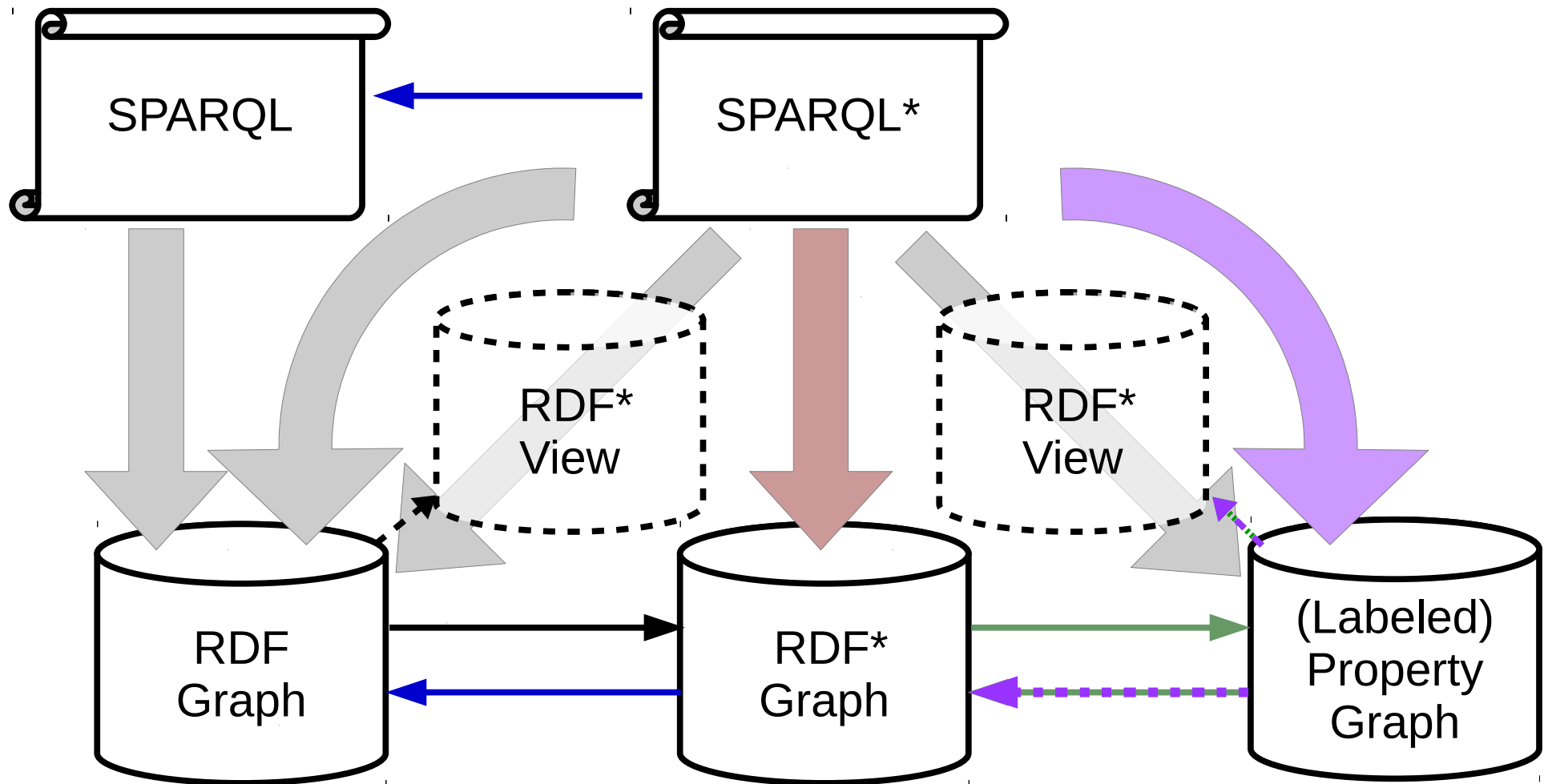
Ongoing Work 1: Modes of Usage (cont'd)

- **Separate-assertions mode (SA mode)**: nested triples do *not* implicitly capture the triples contained in them
 - any triple must be added explicitly to be in the graph
 - allows for making statements about statements that are not required to be “true”
- Example: `SELECT ?x WHERE { Kubrik influencedBy ?x }`
 - data `<<Kubrik influencedBy Welles>> certainty 0.8 .`
 - query result *no mapping*
 - data `<<Kubrik influencedBy Welles>> certainty 0.8 .`
`Kubrik influencedBy Welles .`
 - query result *one mapping with ?x → Welles*

Ongoing Work 2: SPARQL* for RDF and PGs



Ongoing Work 2: SPARQL* for RDF and PGs



Olaf Hartig and Bryan Thompson: “**Foundations of an Alternative Approach to Reification in RDF.**” CoRR abs/1406.3399, 2014.

Olaf Hartig: “**Foundations of RDF* and SPARQL* - An Alternative Approach to Statement-Level Metadata in RDF.**” In Proc. of AMW 2017.

Olaf Hartig: “**Reconciliation of RDF* and Property Graphs.**” In abs/1409.3288, 2014

Olaf Hartig: “**Foundations to Query Labeled Property Graphs using SPARQL*.**” In Proc. of AMAR 2019.



RDF* and SPARQL* in a Nutshell

`<<Kubrik influencedBy Welles>> significance 0.8`

subject *predicate* *object*

```
SELECT ?x WHERE {  
  <<?x influencedBy Welles>> significance ?sig  
  FILTER (?sig > 0.7)  
}
```

1. Syntactic sugar on top of
standard RDF and SPARQL

2. A logical model
in its own right



www.liu.se