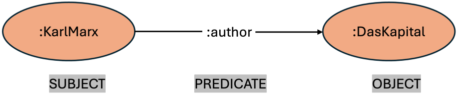
RDF* - Advanced Metadata Modeling

What is RDF?

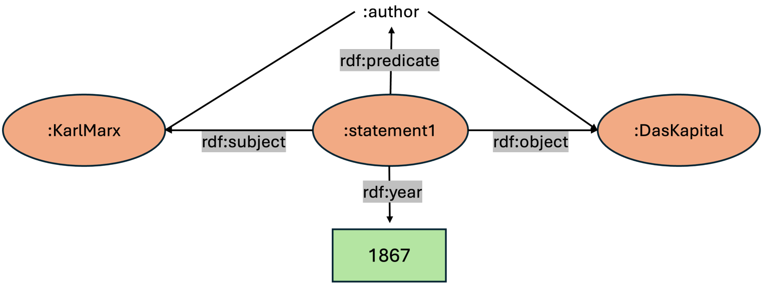
- **Resource Description Framework (RDF)**: A framework to represent data in the Semantic Web.
- **Structure**: Data is structured in *triples* (subject, predicate, object): `:KarlMarx :author :DasKapital`
- **Use Cases**: Knowledge graphs, data integration, and enabling semantic interoperability
- **Limitations**:
  - Cannot easily express metadata about relationships
  - Leads to complexity when representing context, source, or certainty of statements



<http://example.org/KarlMarx> <http://example.org/atuhor> <http://example.org/DasKapital>

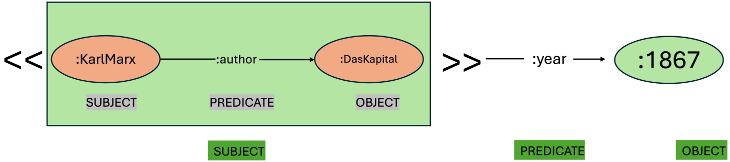
Why Do We Need RDF*?

- **Metadata Challenges in RDF**:
  - RDF cannot natively add context to statements (e.g., source, date, certainty)
  - **Reification**: Standard RDF workaround, but it requires multiple extra triples, increasing complexity
- **Goal of RDF***:
  - Introduces a way to add metadata directly to triples
  - Provides a more efficient, intuitive solution for metadata and contextual information



What is RDF*?

- **RDF-star** (RDF*): An extension of RDF for easier metadata representation
- **Core Idea**: Allows triples to be treated as subjects/objects in other triples
- **Example**:
  - Standard RDF: `:KarlMarx :author :DasKapital`
  - RDF*: `<<:KarlMarx :author :DasKapital>> :year 1867`



<<http://example.org/KarlMarx> <http://example.org/author> <http://example.org/DasKapital>> <http://example.org/year> “1867" .

)

Technical Aspects of RDF*

- **Triple Embedding**:
  - In RDF*, triples can be embedded in other triples using `<<subject predicate object>>`
- **SPARQL***:
  - Extension of SPARQL to query RDF* data

Example Query:

```
sparql
SELECT ?s ?p ?o ?certainty WHERE {
  << ?s ?p ?o >> :year ?year .
}
```

Advantages of RDF*

- **Simplified Data Modeling**:
  - Directly express metadata on statements
- **Increased Performance**:
  - Avoids the need for complex reification structures, reducing triple count
- **Improved Querying**:
  - Queries for metadata become more straightforward and easier to interpret
- **Growing Tool Support**:
  - RDF* is e.g. supported by GraphDB and Blazegraph

Applications of RDF*

- **Provenance Tracking**:
  - Track metadata like source, author, or timestamp of data statements
- **Data Confidence & Uncertainty**:
  - Represent certainty levels, useful in research, finance, and knowledge graphs
- **Data Integration**:
  - Useful in healthcare, open data, government records, where provenance and context are key
- **Knowledge Graphs**:
  - RDF* simplifies complex relationship representation in large datasets

Challenges and Considerations

- **Compatibility**:
  - Not all RDF and SPARQL tools support RDF*, limiting interoperability
- **Standardization**:
  - RDF* is evolving; W3C standardization is still in progress
- **Tooling Requirements**:
  - Support depends on triple stores and query engines implementing RDF* standards

Summary

- **RDF* Overview**:
  - RDF* extends RDF by allowing metadata directly on triples
- **Key Advantages**:
  - Simplifies modeling and querying, especially for metadata-rich use cases
- **Applications**:
  - Useful in provenance tracking, uncertainty representation, and knowledge graphs
- **Future of RDF***:
  - Increased adoption and tool support as RDF* moves toward W3C standardization
- **Consider RDF***:
  - When building applications with complex, metadata-driven data needs

Sources

1. RDF-star Working Group Charter, W3C
2. RDF Working Group Wiki, W3C
3. “What is RDF-star?” Ontotext
4. “The Pros and Cons of RDF-star and SPARQL-star,” Data Science Central