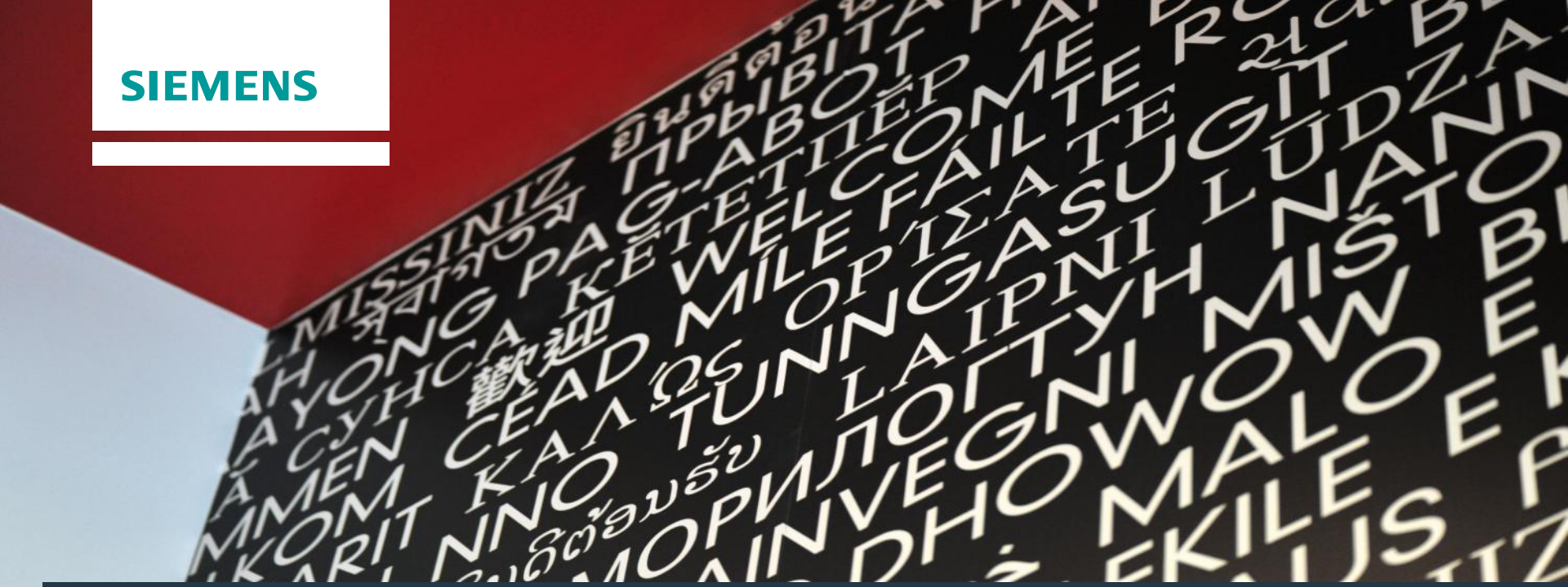


Claudia Bretschneider[1,2], Heiner Oberkampf[2], Sonja Zillner[2,3]

UIMA2LOD: Integrating UIMA Text Annotations into the Linked Open Data Cloud

[1] University Munich, Center for Information and Language Processing, Munich, Germany
[2] Siemens AG, Corporate Technology, Munich, Germany
[3] School of International Business and Entrepreneurship, Steinbeis University, Berlin, Germany

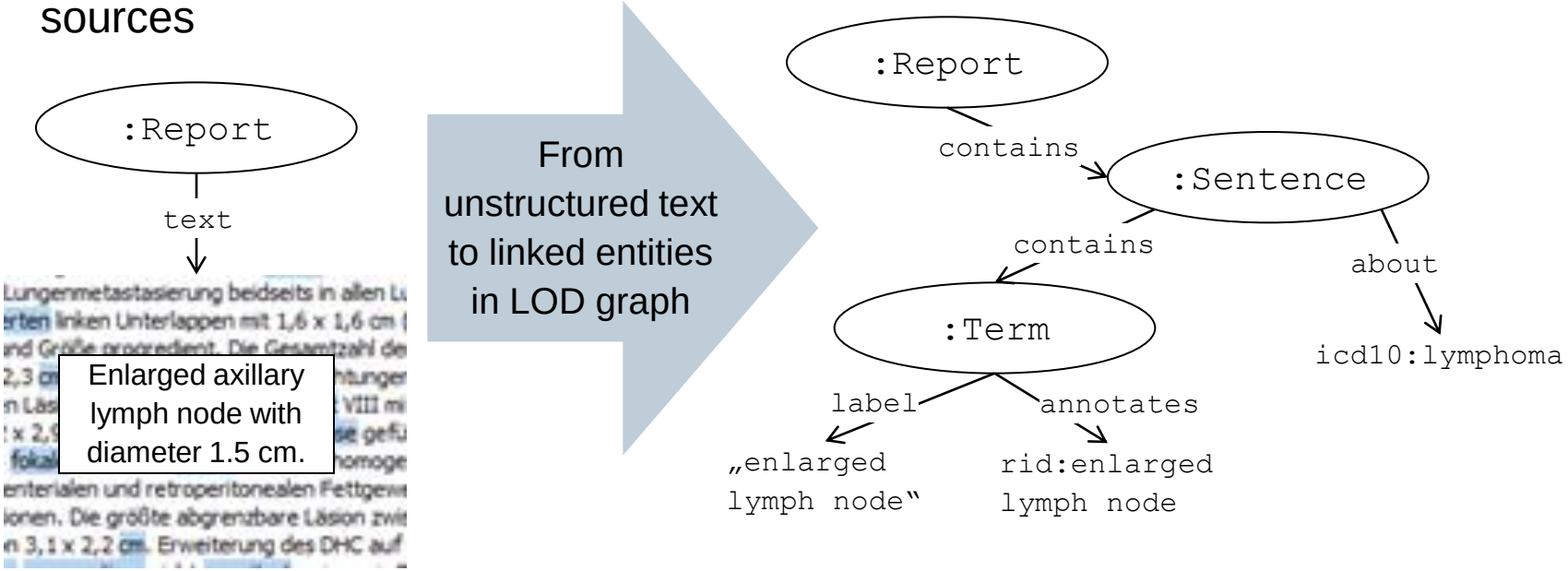
SIEMENS

Agenda

1. Problem of Texts as Unused Resources in LOD Resources
2. UIMA Pipeline for Creating and Integrating Semantic Text Annotations
3. Conceptual Representation of Semantic Text Annotations
4. Case Study on Integrating Annotations into the Model for Clinical Information (MCI)
5. Comparison of Approach with Existing Gold Standard

Problem of Texts as Unused Resources in LOD Resources

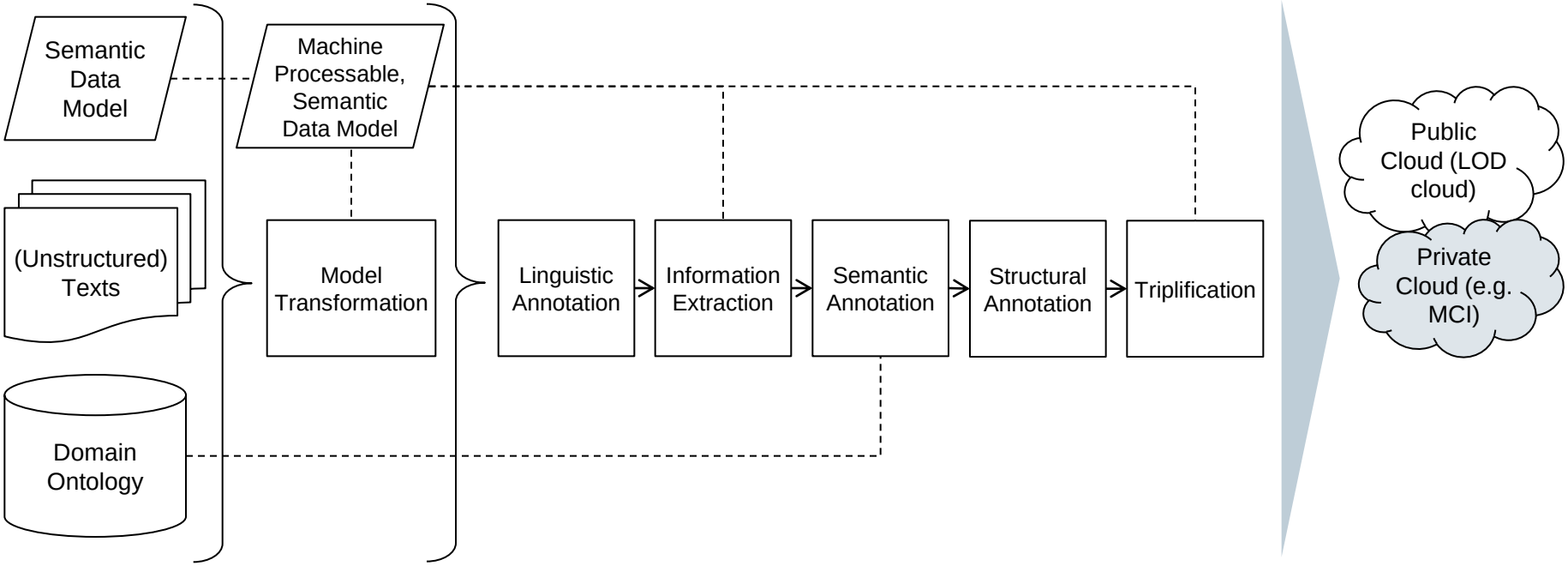
- About 80% of all information is encapsulated in unstructured format
- Content enclosed in unstructured texts is not available for structured analysis
- Employ Natural Language Processing (NLP) methods for the extraction of textual information
- Goal of this work to deliver a framework and pipeline for automatically extracting structured information from texts as linked entities, integrated with the LOD cloud, thus extending the LOD cloud with information from textual sources



Overview of Integration Process: From Semantic Annotation to RDF-Transformation

SIEMENS

Integration of Text into LOD



Semantic Enrichment of Text with LOD information

Linguistic Annotation

UIMA-based Information Extraction pipeline operating on recognized linguistically meaningful units

Semantic Annotation

Alignment of text with semantic resources and additional identification of domain-specific concepts

Structural Annotation

Internal representation of textual annotations using defined semantic structures

Information Extraction (IE) framework: High-Level UIMA Component Architecture from Source to Sink

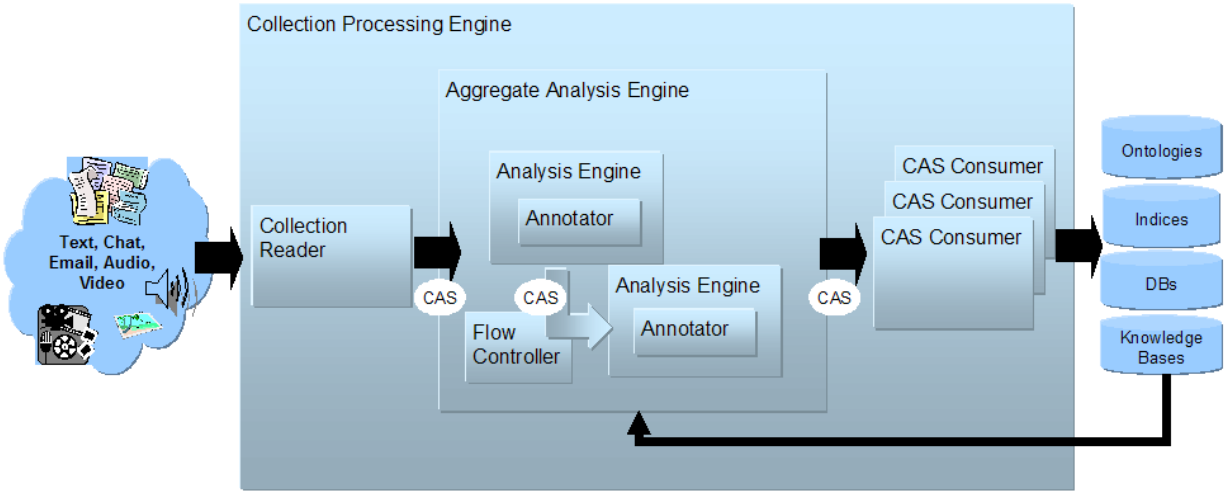
SIEMENS

Characteristics

- Unstructured information management architecture
- Former IBM project matured to Apache project
- Architectural framework (incl. tools, annotators) to manage and facilitate analysis of unstructured content
- Helps enriching texts with metadata (so called annotations)

Advantages of UIMA for the medical annotation process

- Modularity of components
- UIMA ships with a list of implemented standard components (FileSystemReader, ConceptMapper, LuceneIndexConsumer, etc.)
- Numerous medical text analysis systems that are built on UIMA
- Exchangeability and reuse of existing (external) components
- Implementation support enables focus on functional requirements rather than technical

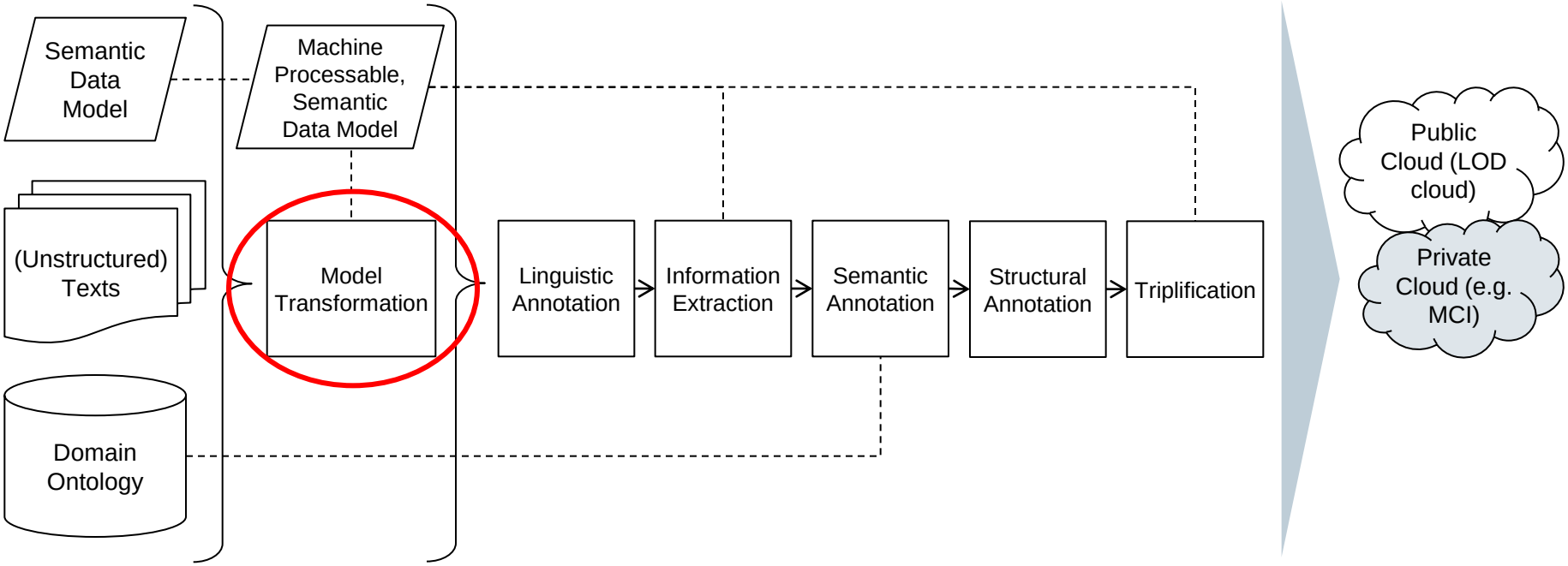


Source: http://uima.apache.org/d/uimaj-2.4.2/overview_and_setup.html

Overview of Integration Process: From Semantic Annotation to RDF-Transformation

SIEMENS

Integration of Text into LOD



Semantic Enrichment of Text with LOD information

Linguistic Annotation

UIMA-based Information Extraction pipeline operating on recognized linguistically meaningful units

Semantic Annotation

Alignment of text with semantic resources and additional identification of domain-specific concepts

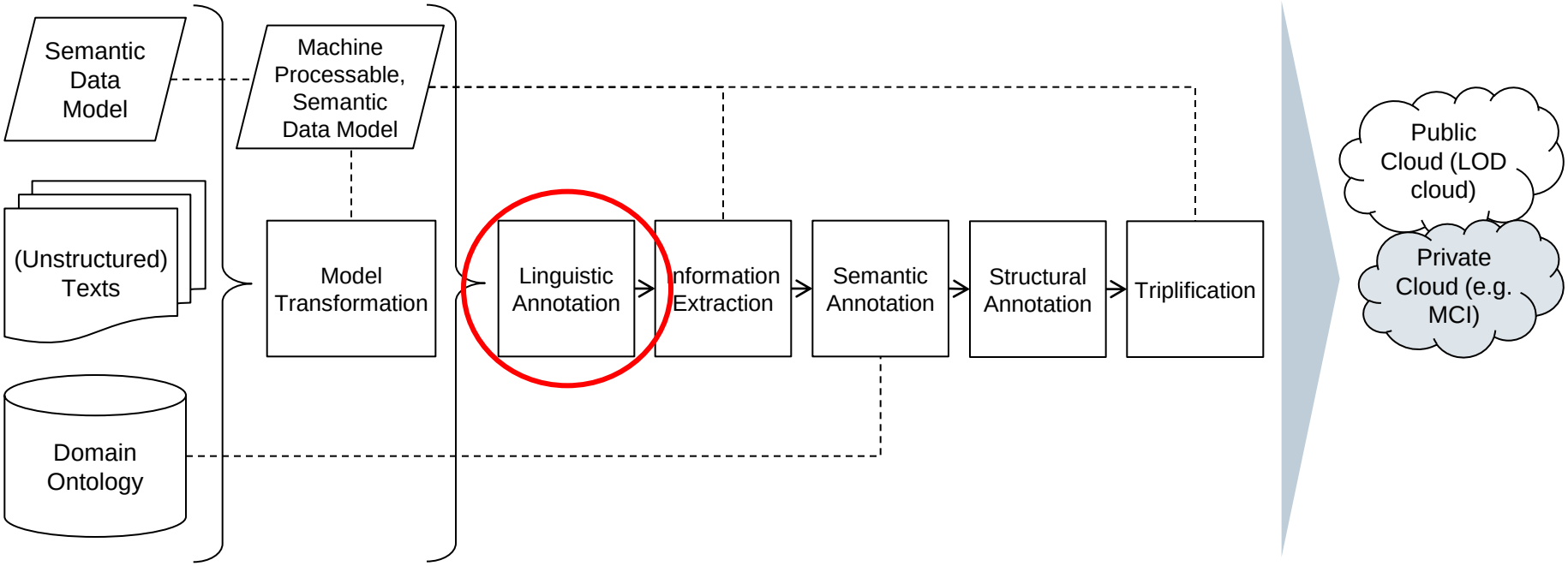
Structural Annotation

Internal representation of textual annotations using defined semantic structures

Overview of Integration Process: From Semantic Annotation to RDF-Transformation

SIEMENS

Integration of Text into LOD



Semantic Enrichment of Text with LOD information

Linguistic Annotation

UIMA-based Information Extraction pipeline operating on recognized linguistically meaningful units

Semantic Annotation

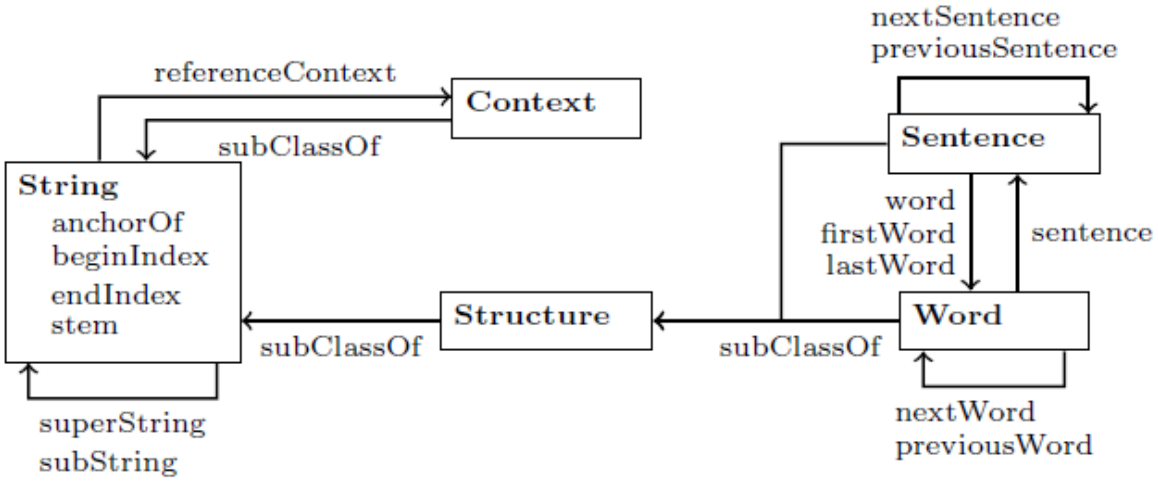
Alignment of text with semantic resources and additional identification of domain-specific concepts

Structural Annotation

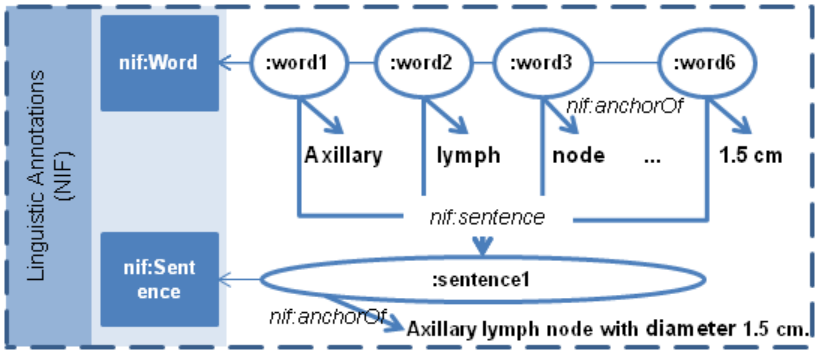
Internal representation of textual annotations using defined semantic structures

Linguistic Annotation

- Three linguistic annotators creating the annotations that reflect the basic linguistic units in the texts:
  (1) sentence splitting, (2) tokenization and (3) compound splitting
- NLP Interchange Format (NIF) ontology for representation of resulting linguistic annotations and their relations
- We are able to support adaptation of *already existing* UIMA annotators – no need for annotators designed specifically for this task
- Requirement for distinct representation of linguistic information in text



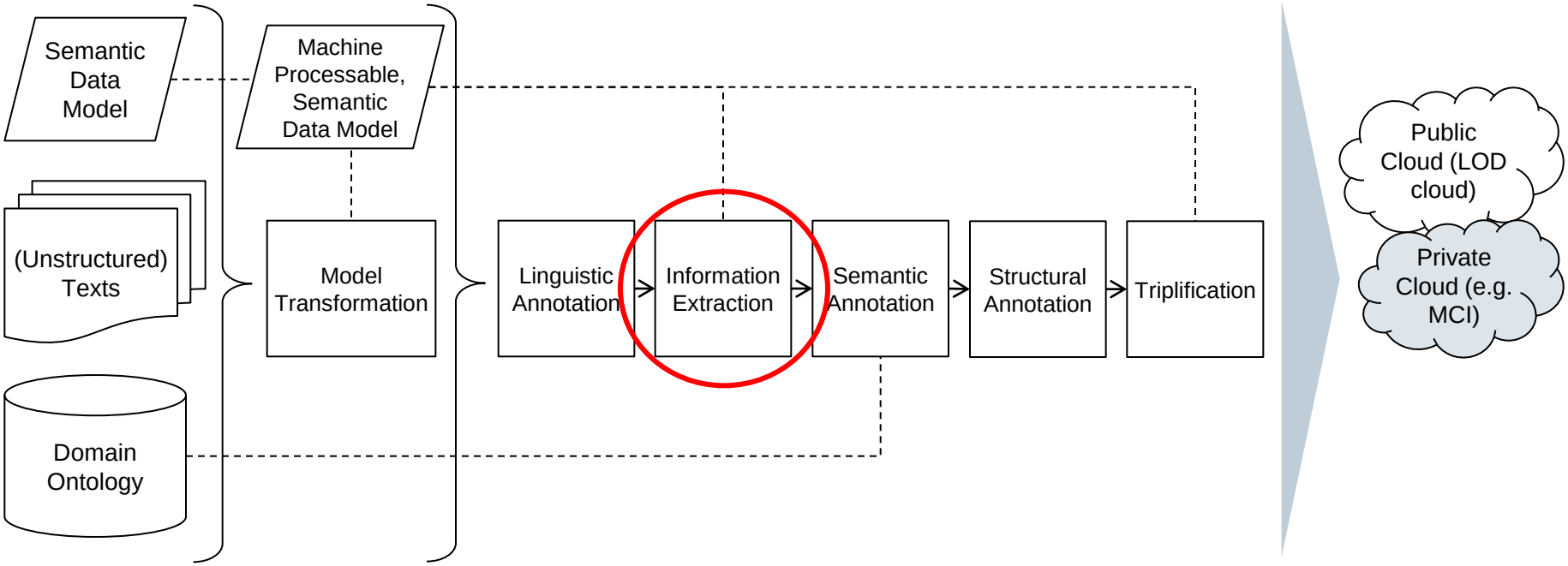
Axillary lymph node with diameter 1.5 cm.
Linguistic Annotations



Overview of Integration Process: From Semantic Annotation to RDF-Transformation

SIEMENS

Integration of Text into LOD



Semantic Enrichment of Text with LOD information

Linguistic Annotation

UIMA-based Information Extraction pipeline operating on recognized linguistically meaningful units

Semantic Annotation

Alignment of text with semantic resources and additional identification of domain-specific concepts

Structural Annotation

Internal representation of textual annotations using defined semantic structures

SIEMENS

Information Extraction (IE)

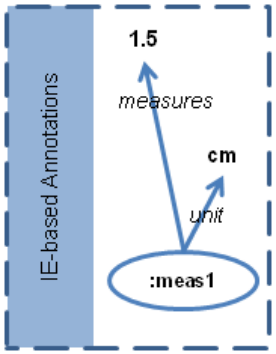
- Semantic Annotations that represent useful information of the target domain
- Domain defines annotators necessary – multitude of NLP algorithms can be integrated (including regular expressions, entity lists, and other more sophisticated algorithms)

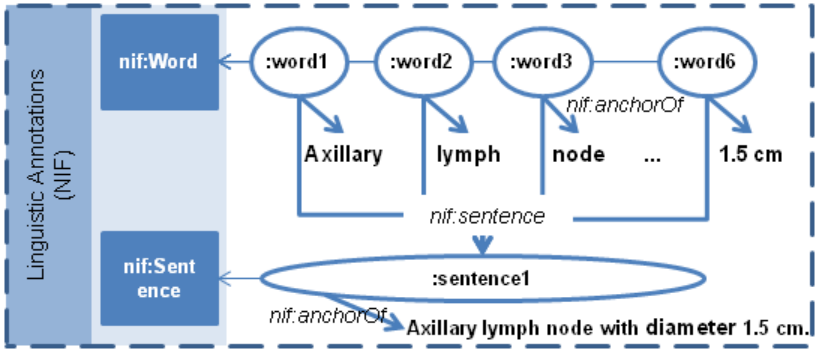
- Annotations specific to the use case tackled
  e.g. medical measurements describing the organs' size status
  - *Representation specific to the use case*

```
@prefix ex: <http://example.org/stuff/1.0/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema/> .

<http://example.org/stuff/1.0/MeasurementAnnotation124112>
              ex:begin    20;   ex:end   26;
              ex:text     "1.5 cm"^^xsd:string;
              ex:measures 1.5;
              ex:unit     "cm"^^xsd:string .
```

Axillary lymph node with diameter 1.5 cm.
IE-based Annotations

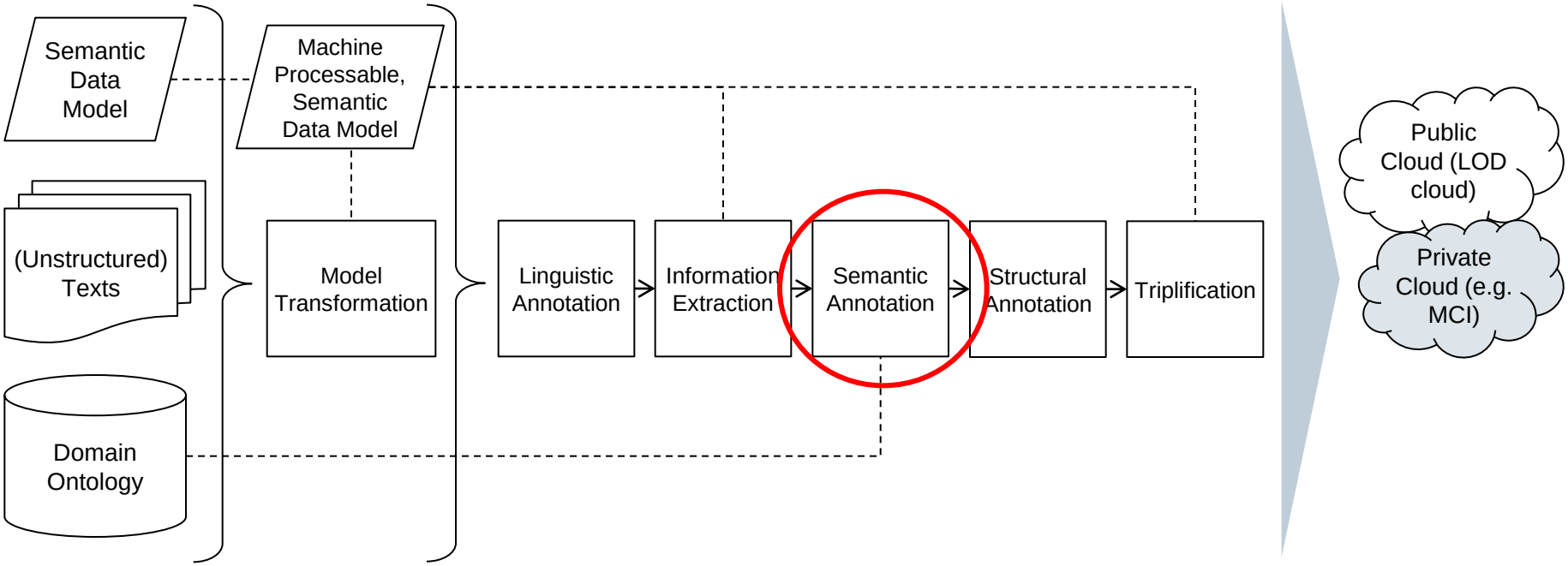




Overview of Integration Process: From Semantic Annotation to RDF-Transformation

SIEMENS

Integration of Text into LOD



Semantic Enrichment of Text with LOD information

Linguistic Annotation

UIMA-based Information Extraction pipeline operating on recognized linguistically meaningful units

Semantic Annotation

Alignment of text with semantic resources and additional identification of domain-specific concepts

Structural Annotation

Internal representation of textual annotations using defined semantic structures

SIEMENS

Semantic Annotation

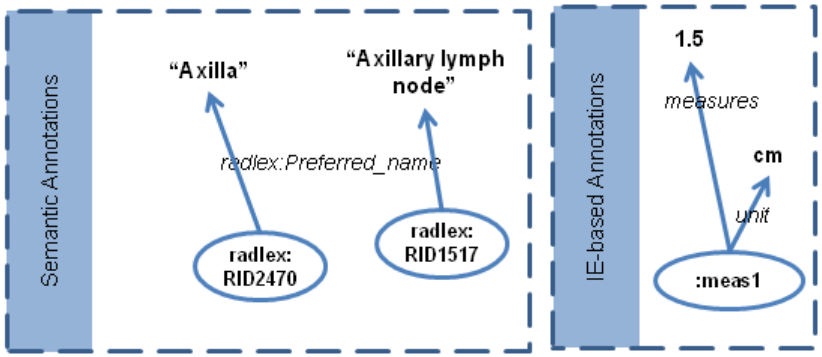
- Semantic Annotations that represent useful information of the target domain
- Created using vocabulary from domain ontologies
- Two possible application scenarios:
  1. Identify domain-specific semantically classified concepts + interconnect ontology's knowledge to the textual information
  2. Link to other existing LOD datasets

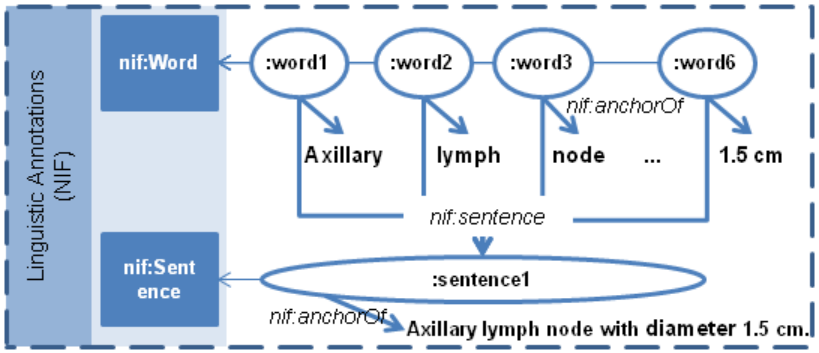
- Implementation based on UIMA Concept Mapper
- Requires transformation of ontology vocabulary into defined XML structure

```
<token RID="RID1301" URI="http://www.owl-ontologies.com/Ontology1392225293.owl#RID1301"
          pn="lung" semanticClass="anatomical">
          <variant base="lung"/>
          <variant base="Lunge"/>
          <variant base="pulmo"/>
</token>
```

- Annotations are created by mapping stemmed UIMA variants from the XML dictionary to the text's tokens

Axillary lymph node with diameter 1.5 cm.
Semantic Annotations

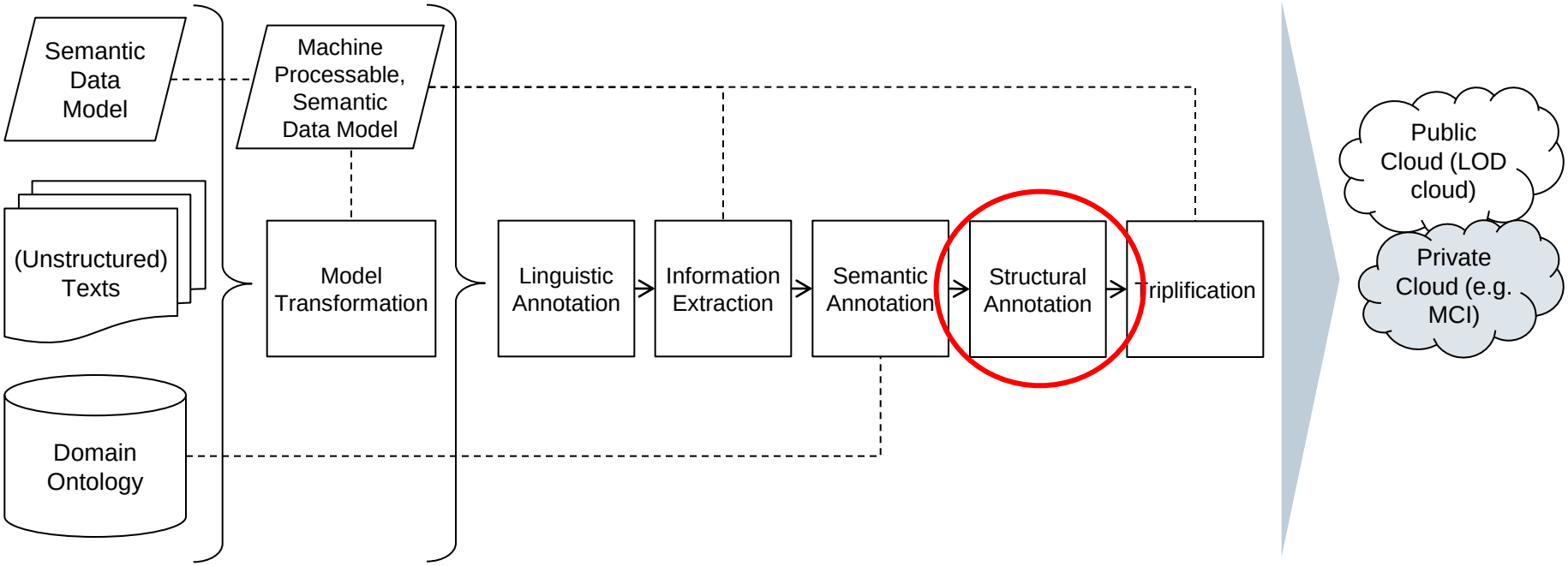




Overview of Integration Process: From Semantic Annotation to RDF-Transformation

SIEMENS

Integration of Text into LOD



Semantic Enrichment of Text with LOD information

Linguistic Annotation

UIMA-based Information Extraction pipeline operating on recognized linguistically meaningful units

Semantic Annotation

Alignment of text with semantic resources and additional identification of domain-specific concepts

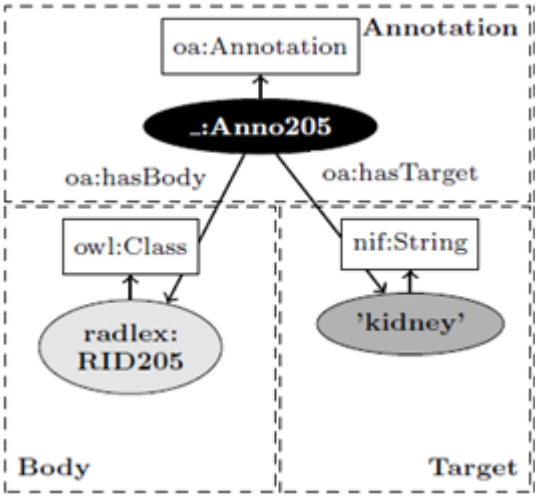
Structural Annotation

Internal representation of textual annotations using defined semantic structures

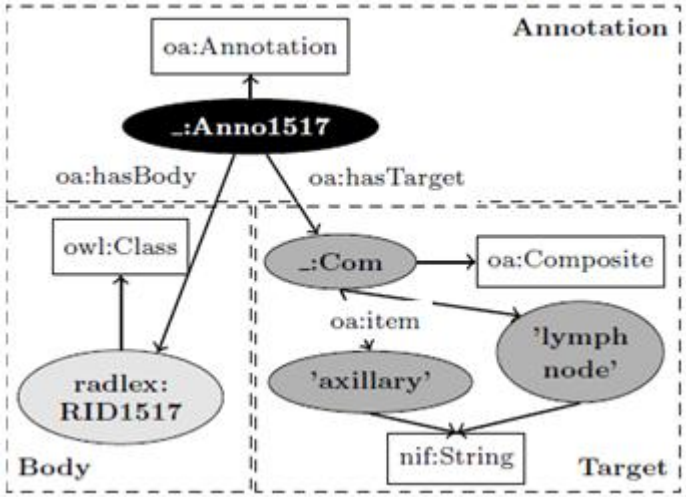
Open Annotation (OA) Structures

- Structural annotations that interconnect the linguistic and the semantic world of the text annotations
- Show how semantics is mapped to linguistic annotation using semantic annotations
- Generic approach by employing basic elements
  - *annotation* – annotation element
  - *target* – linguistic annotation
  - *body* – reference to ontology concept

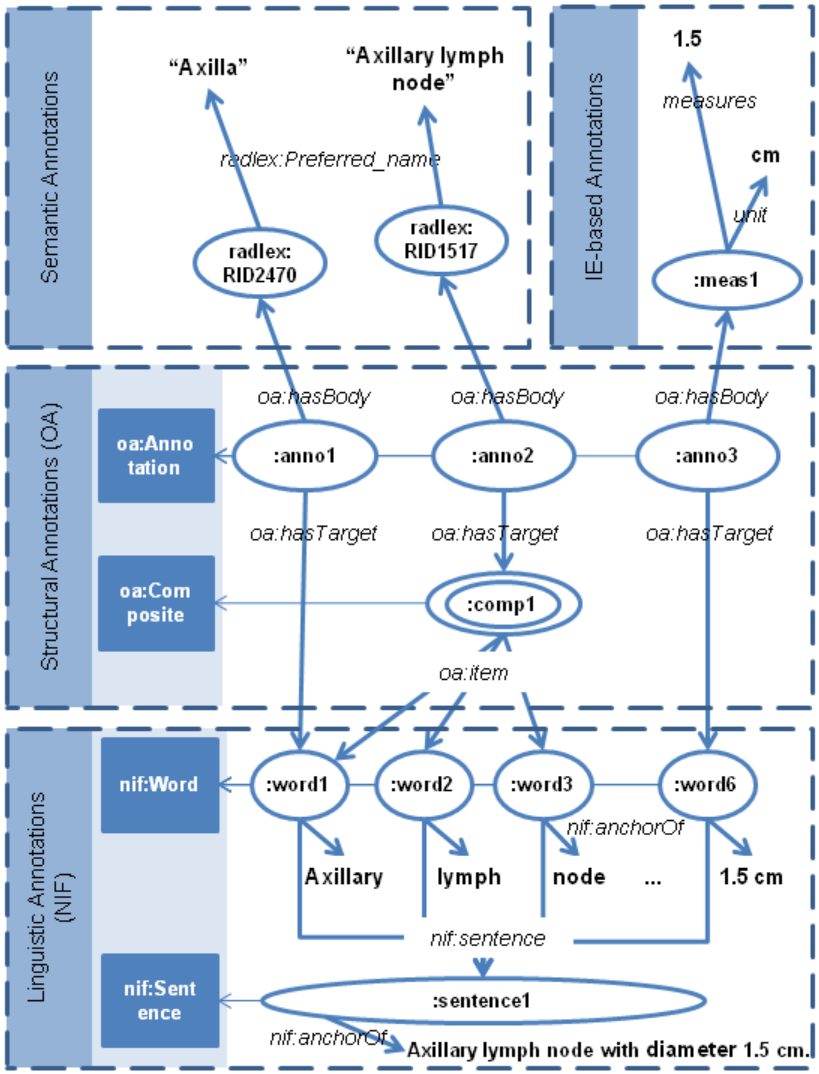
Single-Target = One Token annotated



Multiple Targets = Multiple Tokens annotated



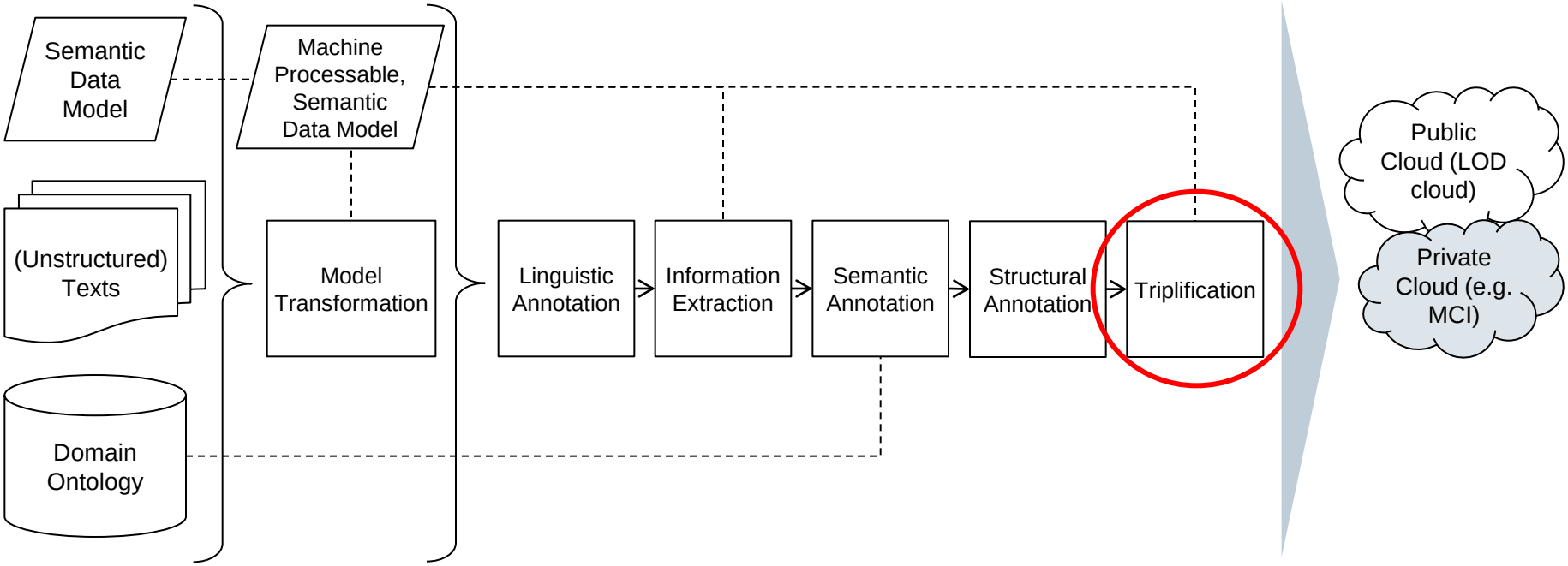
Axillary lymph node with diameter 1.5 cm.
Structural Annotations



Overview of Integration Process: From Semantic Annotation to RDF-Transformation

SIEMENS

Integration of Text into LOD



Semantic Enrichment of Text with LOD information

Linguistic Annotation

UIMA-based Information Extraction pipeline operating on recognized linguistically meaningful units

Semantic Annotation

Alignment of text with semantic resources and additional identification of domain-specific concepts

Structural Annotation

Internal representation of textual annotations using defined semantic structures

SIEMENS

Triplification

- Correct representation of the text annotations created by the NLP pipeline into RDF graph
- Gold Standard: UIMA RDF Consumer
  - 5 limitations with respect to triplification of
  1. Data Properties
  2. Plain Literals
  3. Ambiguities in Unique IDs
  4. Object Properties
  5. Non-Functional Properties

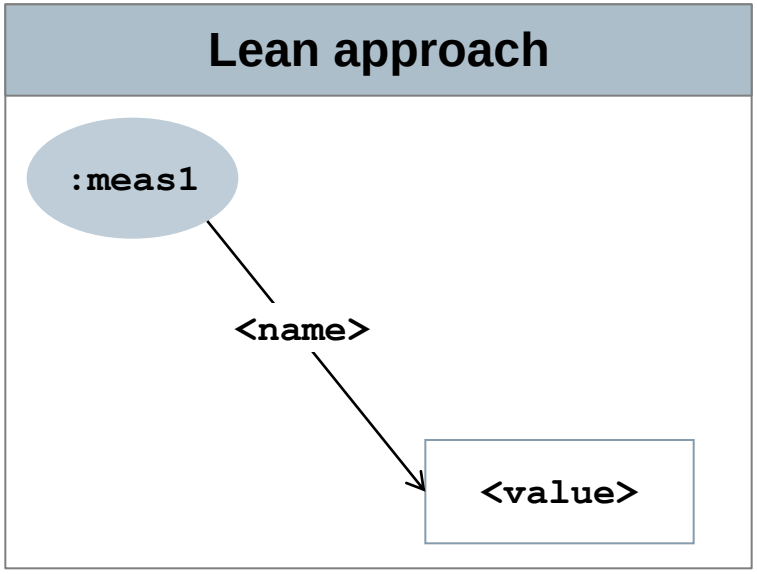
Triplification –
(1) Declarative Modeling of Data Properties

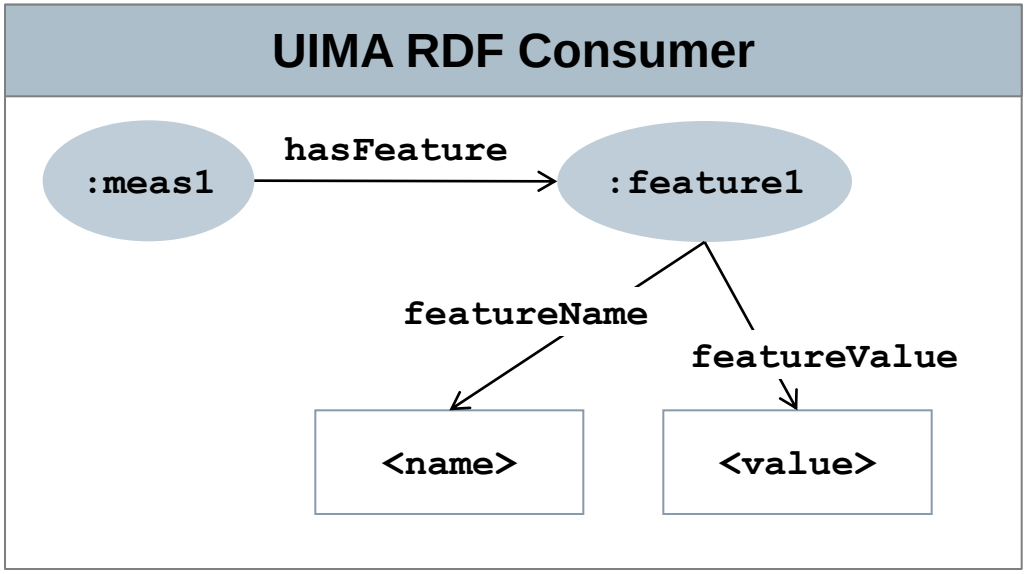
SIEMENS

- Each annotation feature is represented using three triples
- Our approach: Lean and intuitive representation of data properties

Triplification –
(1) Declarative Modeling of Data Properties

- Each annotation feature is represented using three triples
- Our approach: Lean and intuitive representation of data properties





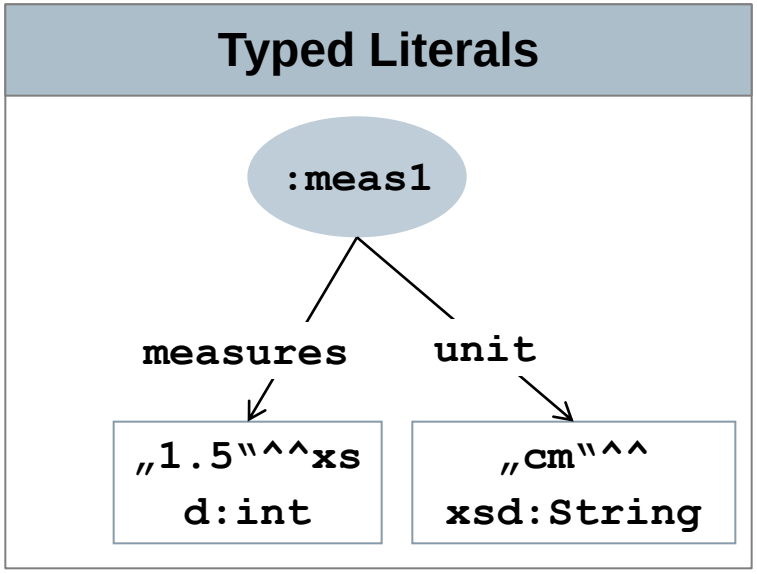
Triplification –
(2) Typed instead of Plain Literals

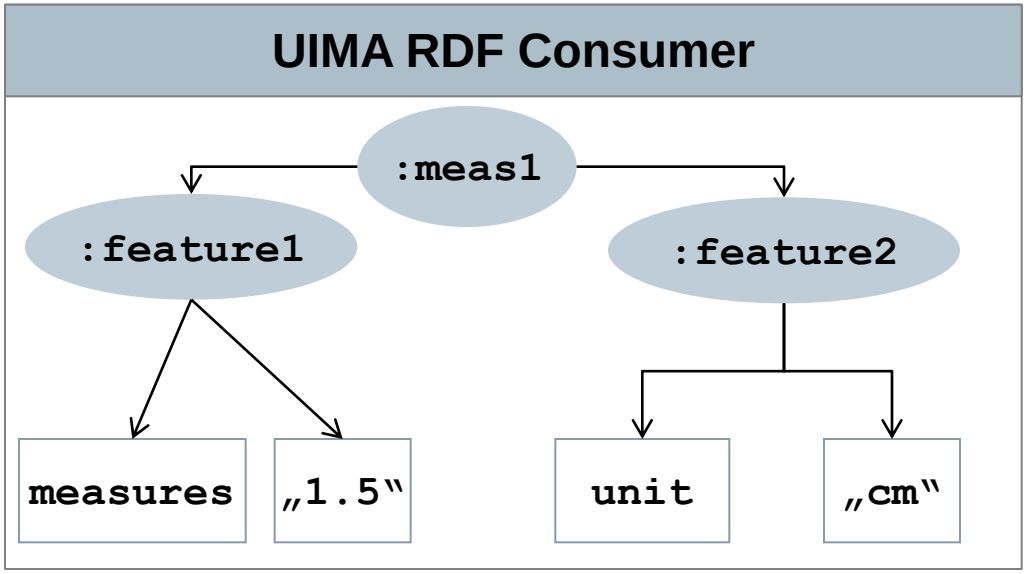
SIEMENS

- Plain literals cannot be interpreted with their correct data type
- Our approach: Use the information from the UIMA type system and assign correct data type
- Enable automated analysis of the values

Triplification – (2) Typed instead of Plain Literals

- Plain literals cannot be interpreted with their correct data type
- Our approach: Use the information from the UIMA type system and assign correct data type
- Enable automated analysis of the values





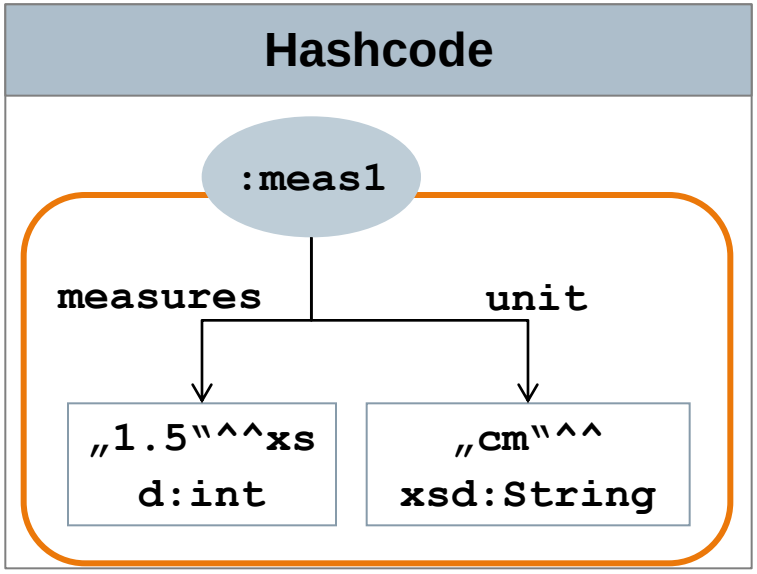
Triplification – (3) Resolution of Ambiguities with Unique IDs

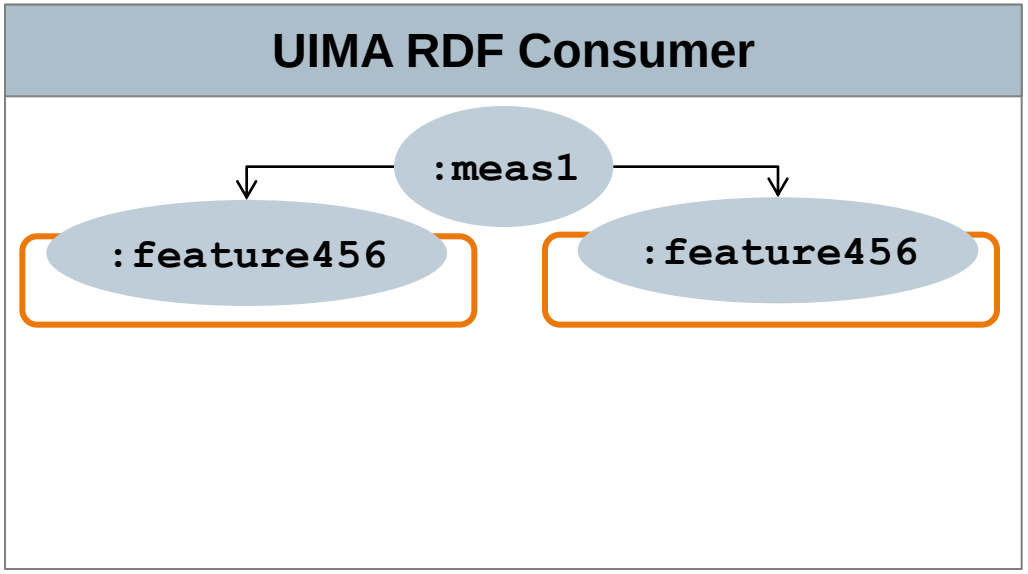
SIEMENS

- Incorrect calculation of identifiers for resources leads to ambiguities in assignment of `featureName` and `featureValue`
- Our approach: Hash code representing all features (names and values)
- Combined into URL as unique identifier for resource

Triplification – (3) Resolution of Ambiguities with Unique IDs

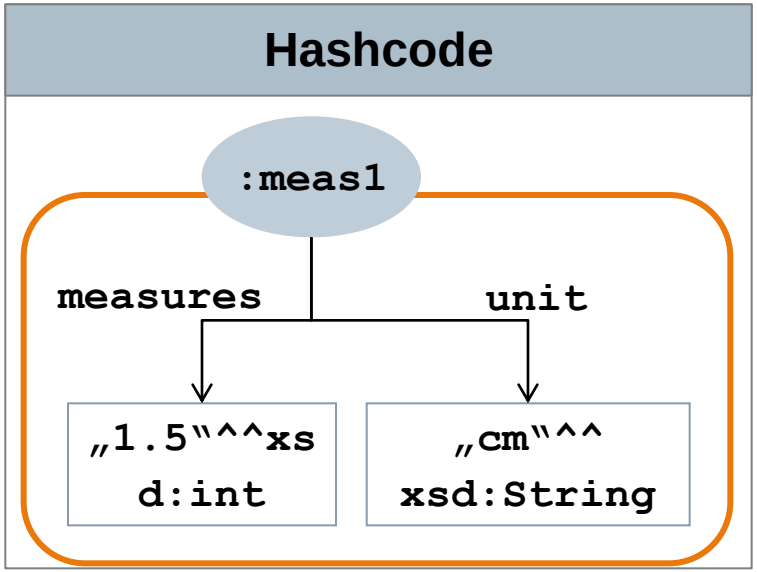
- Incorrect calculation of identifiers for resources leads to ambiguities in assignment of `featureName` and `featureValue`
- Our approach: Hash code representing all features (names and values)
- Combined into URL as unique identifier for resource

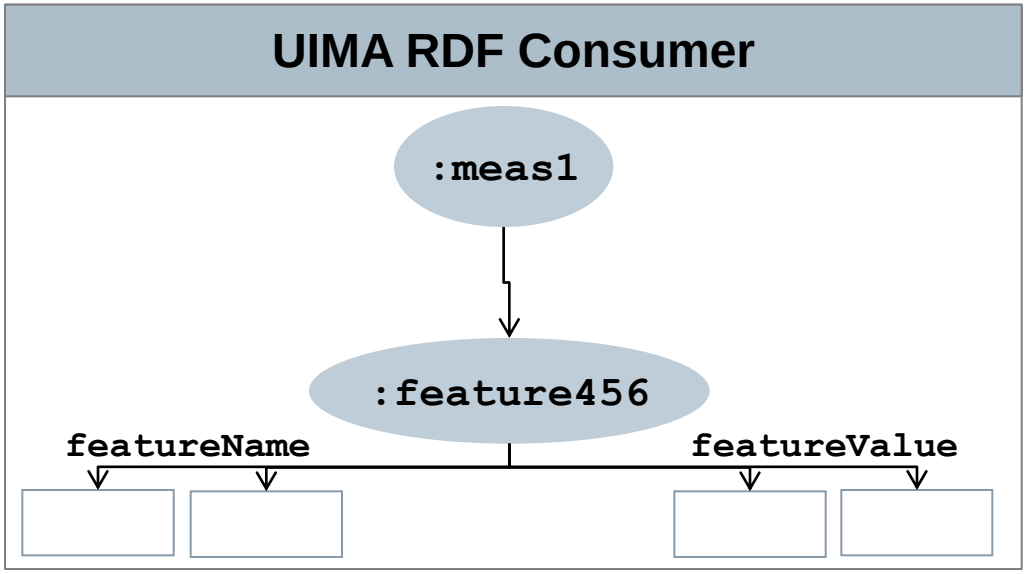




Triplification –
(3) Resolution of Ambiguities with Unique IDs

- Incorrect calculation of identifiers for resources leads to ambiguities in assignment of `featureName` and `featureValue`
- Our approach: Hash code representing all features (names and values)
- Combined into URL as unique identifier for resource





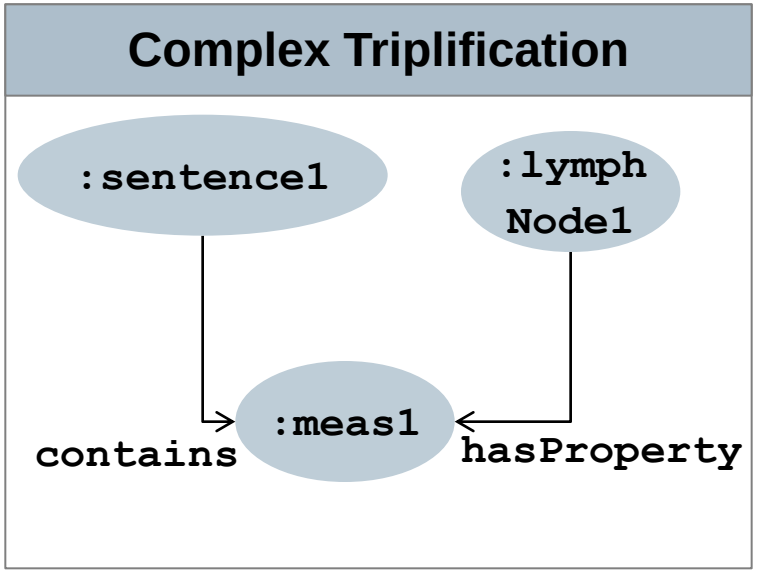
Triplification –
(4) Triplification of Object Properties

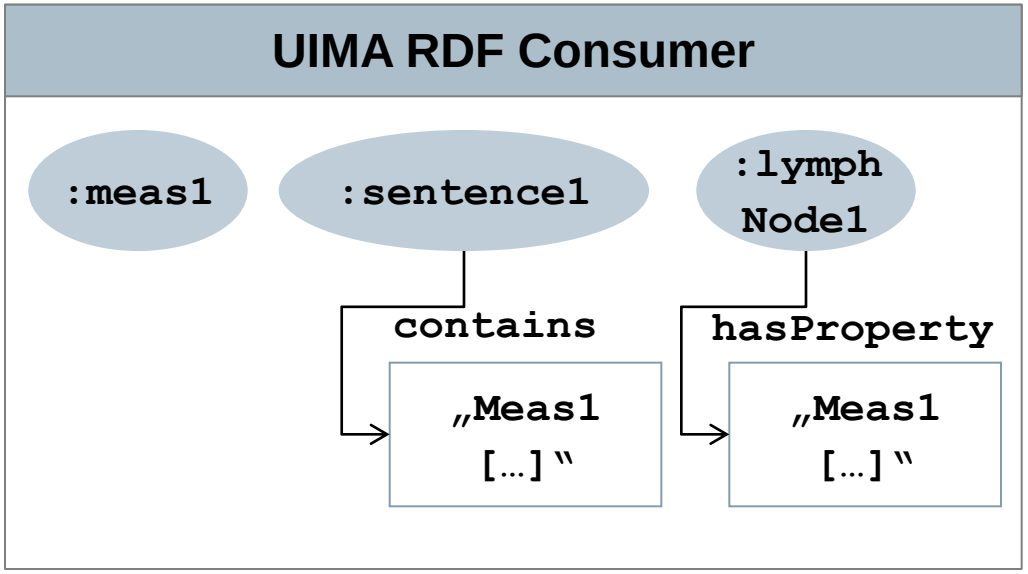
SIEMENS

- Linked resources are only triplified to their string representation
- Entities are isolated and links are missing
- Our approach: Deep triplification (recursion) of referenced resources with their features
- Reoccurring resources are identified by their unique id and referenced accordingly

Triplification –
(4) Triplification of Object Properties

- Linked resources are only triplified to their string representation
- Entities are isolated and links are missing
- Our approach: Deep triplification (recursion) of referenced resources with their features
- Reoccurring resources are identified by their unique id and referenced accordingly





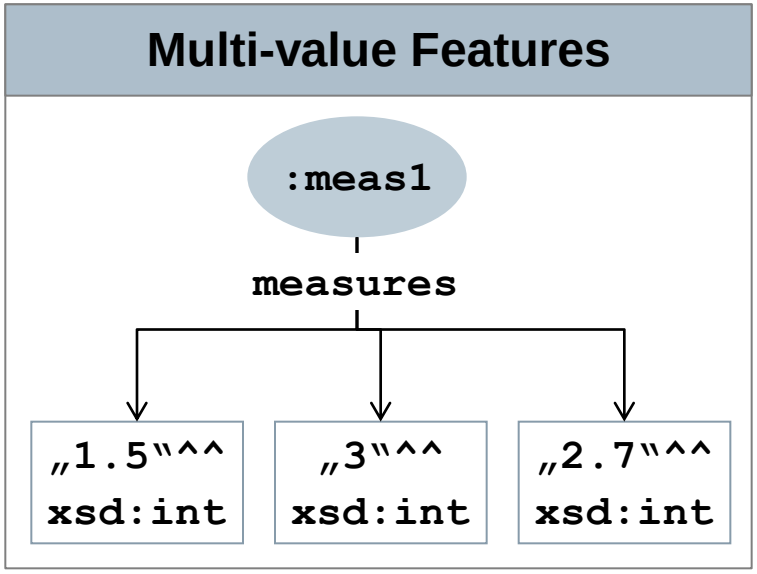
Triplification –
(5) Triplification of Non-Functional Properties

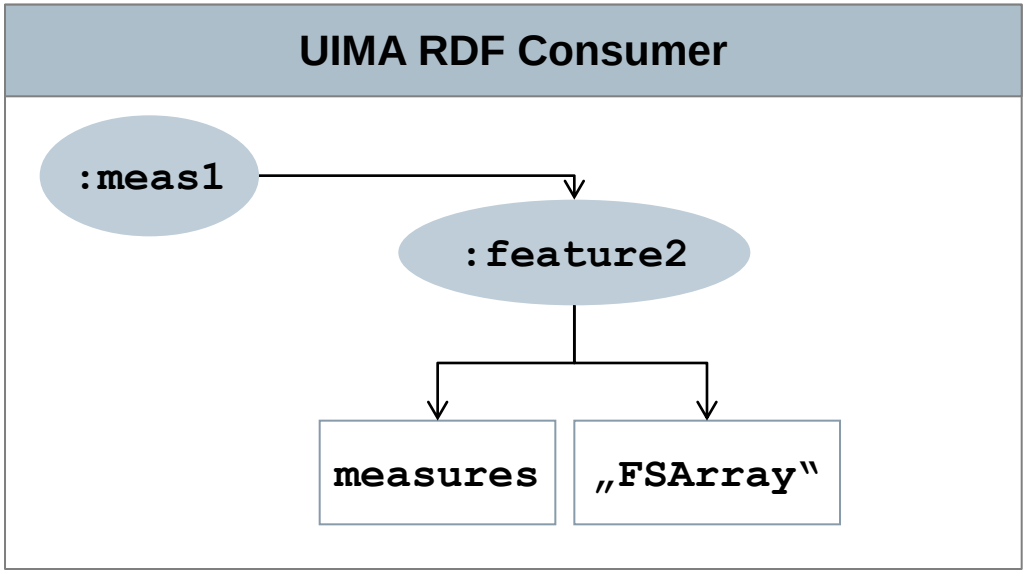
SIEMENS

- All multi-value features should be triplified to non-functional properties, however the used string representation masks the existing values (for both data and object properties)
- Our approach: Identification of value dimension and consideration of multi-value features by deep triplification (recursion) of referenced values

Triplification –
(5) Triplification of Non-Functional Properties

- All multi-value features should be triplified to non-functional properties, however the used string representation masks the existing values (for both data and object properties)
- Our approach: Identification of value dimension and consideration of multi-value features by deep triplification (recursion) of referenced values





Qualitative Comparison of Approach with Existing Gold Standard

SIEMENS

- Irregular triplifications of the UIMA RDF Consumer leads to
  - Irreversible loss of object properties and non-functional properties
  - Unavailability of linking between object properties not available
- Incomplete RDF graph with high lacks in information is replaced by full and correct representation of the text annotations for further analysis in Semantic Web-based applications

	CAS2RDF	UIMA2LOD
# NIF Annotations	1,506,029	
# Semantic Annotations	416,251	
# OA Annotations	486,425	
# Triples	144,215,917	47,700,368
# Triples with wrong serialization of		
Non-Functional Properties	681,745	–
Object Properties	8,302,645	–
Runtime of Annotation Pipeline	24h for 180 docs	9 min for all 2713 docs

SIEMENS

Conclusion and Future Work

- Pipeline to automatically triplify the text annotations created in a UIMA pipeline and integrate the text annotations into the LOD cloud
- Consumer fully applicable for any pipeline
- Some annotators have to be exchanged by existing or new implementations to adapt for the texts‘ domains

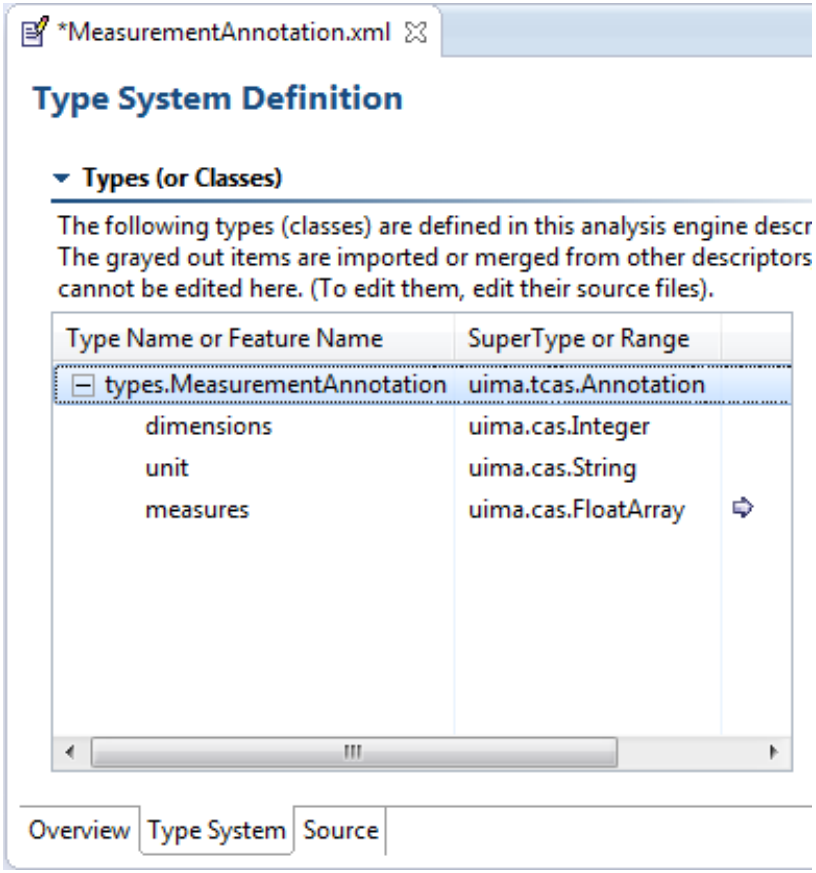
- Future Work
  - Automated process to transform a semantic model into the UIMA-internal representation of the type system
  - Extract information from Big Data corpus (10,000 patients from examinations over a period of up to 30 years)

SIEMENS

BACKUP

Model Transformation

- Alignment of Semantic Model and UIMA type system
- Annotations are stored in UIMA's CAS in addition to their predefined type systems (structure of annotations)
- Annotator uses type system to instantiate created text annotations
- Annotations are assigned features of different types and their triplification equivalences
  - **Primitive data types**
    → `owl:DataProperty`
  - **Complex data** types (i.e. Referecing other annotation types)
    → `owl:ObjectProperty`
  - Features with **multiple instances**
    → <u>not</u>
    `owl:FunctionalProperty`

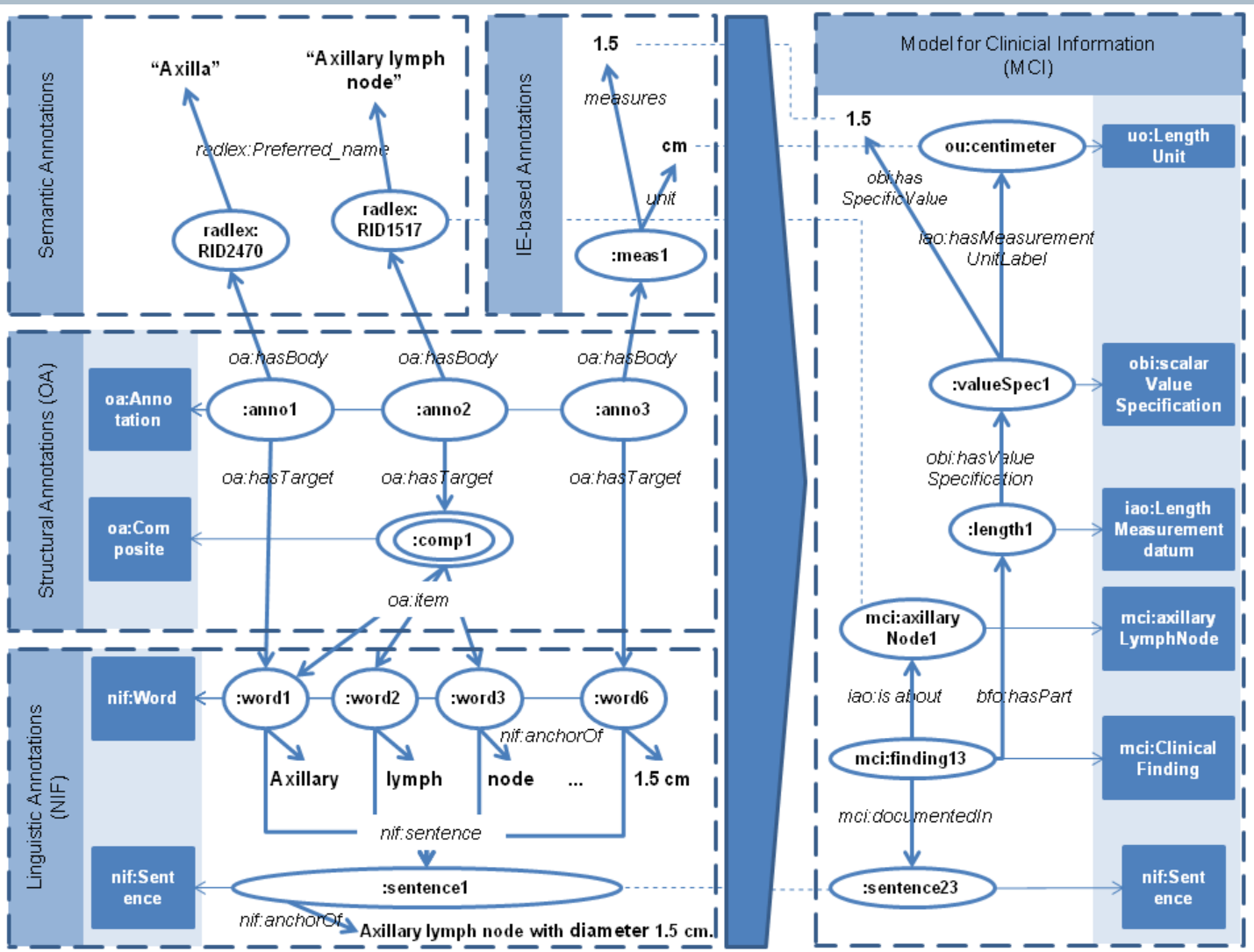


BERNERS-LEE

For the resulting RDF graph to be a valid LOD dataset the requirements imposed by Berners-Lee have been taken into account:

1. **Use URIs as names for things**
2. **Use HTTP URIs so that people can look up those names**
   During the Triplication step each annotation instance gets assigned a unique ID represented as HTTP URI, so that the first two requirements are fulfilled.
3. **When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)**
   For extraction of useful information we included a number of steps (Linguistic Preprocessing, Information Extraction, Open Annotation (OA) Creation), which at the same time support the envisioned conceptual representation. Again here, for the correct structural representation of the resulting triples, the Triplication step is implemented to use the defined standards.
4. **Include links to other URIs, so that they can discover more things.**
   Finally, to enhance the LOD cloud with additional information entities that are also interconnected with existing datasets, we included the Named Entity Recognition step.

Case Study on Integrating RDF Annotations into the Model for Clinical Information (MCI)



Case Study on Integrating RDF Annotations into the Model for Clinical Information (MCI)

SIEMENS

- Resources:
  - Corpus of 2,713 German medical radiology reports
  - RadLex ontology v3.12 containing 74,875 terms for NER
  - Model for Clinical Information (MCI)
- Tranformation process for text annotation RDF graph to MCI
  1. Transformation of RDF Graph into MCI Schema
  2. Transformation and Normalization of Measurement Annotations
  3. Disambiguation of Anatomical Entities
  4. Inference