Ontology-Based XQuery'ing of XML-Encoded Language Resources on Multiple Annotation Layers

Georg Rehm[1], Richard Eckart[2], Christian Chiarcos[3], Johannes Dellert[1]

University of Tübingen[1]
SFB 441: Linguistic Data Structures
Tübingen, Germany

TU Darmstadt[2]
Dept. of English Linguistics
Darmstadt, Germany

University of Potsdam[3]
SFB 632: Information Structure
Potsdam, Germany

Context

- Long-term availability of linguistic resources
- Joint Project "Sustainability of Linguistic Data"
- Consolidation of the corpora and data formats
  - Tusnelda SFB 441 "Linguistic Data Structures"
  - Exmaralda SFB 538 "Multilingualism"
  - Paula SFB 632 "Information Structure"

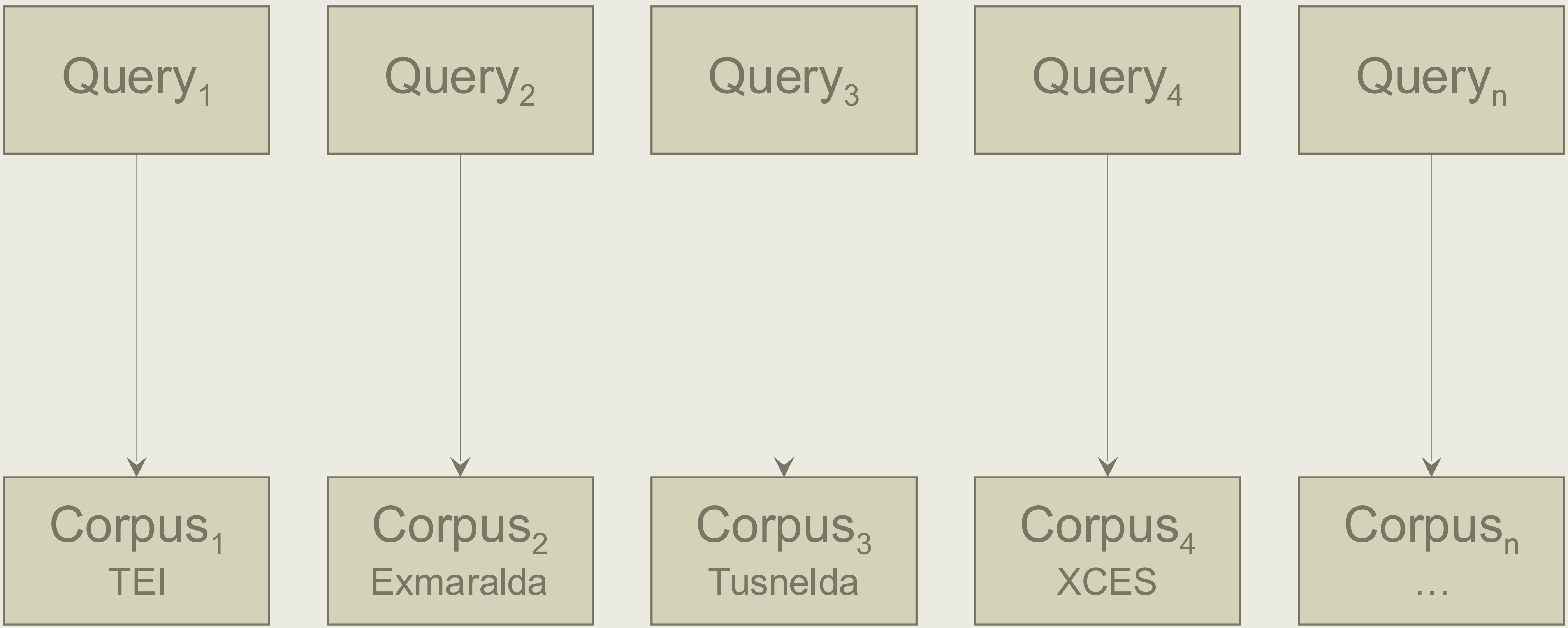
SPLICR

- Sustainability Platform for Linguistic Corpora and Resources
  - ~60 highly heterogeneous linguistic resources
- Goals
  - Centralized corpus platform
  - Homogeneous means of accessing and querying
  - Generalisation over
    - Format (Tusnelda, Exmaralda, etc.)
    - Semantics (various tag-sets)
  - Web-based user interface
    - Intuitively usable for linguists

Linguistic Corpora

- Corpus specific queries

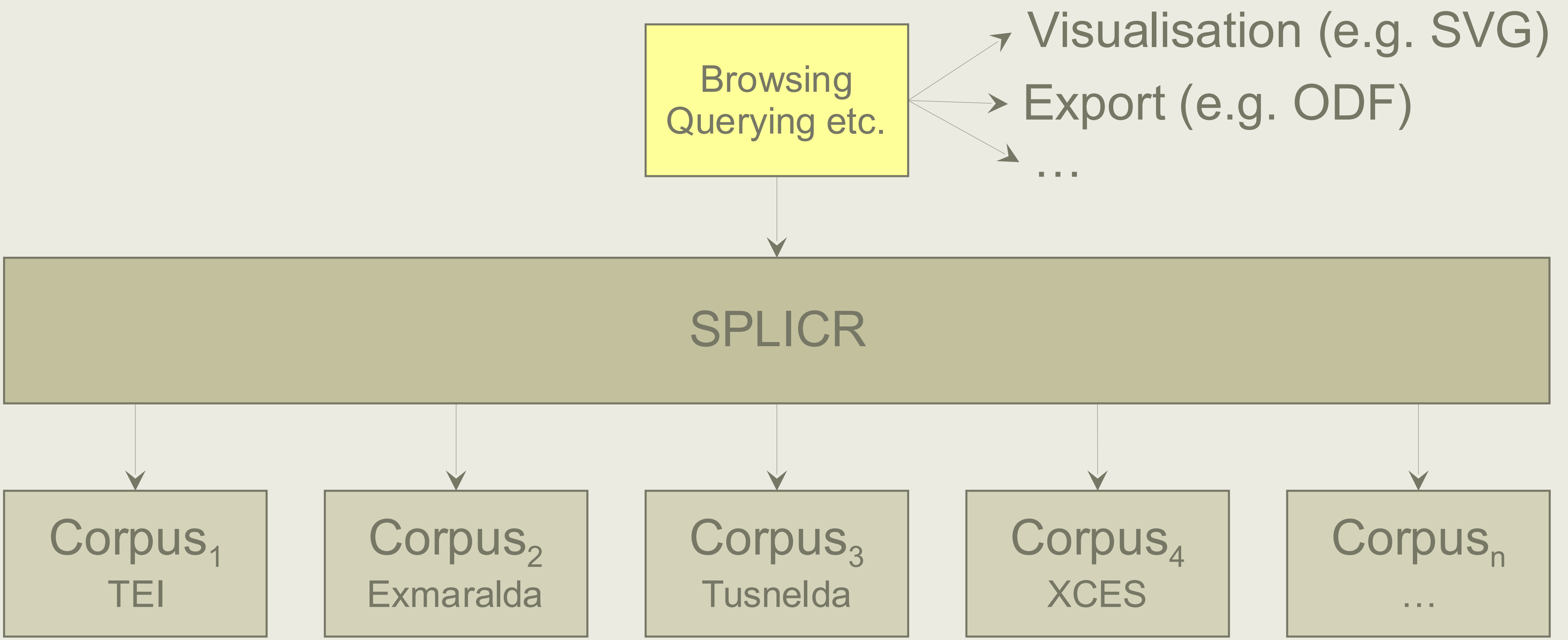
status quo



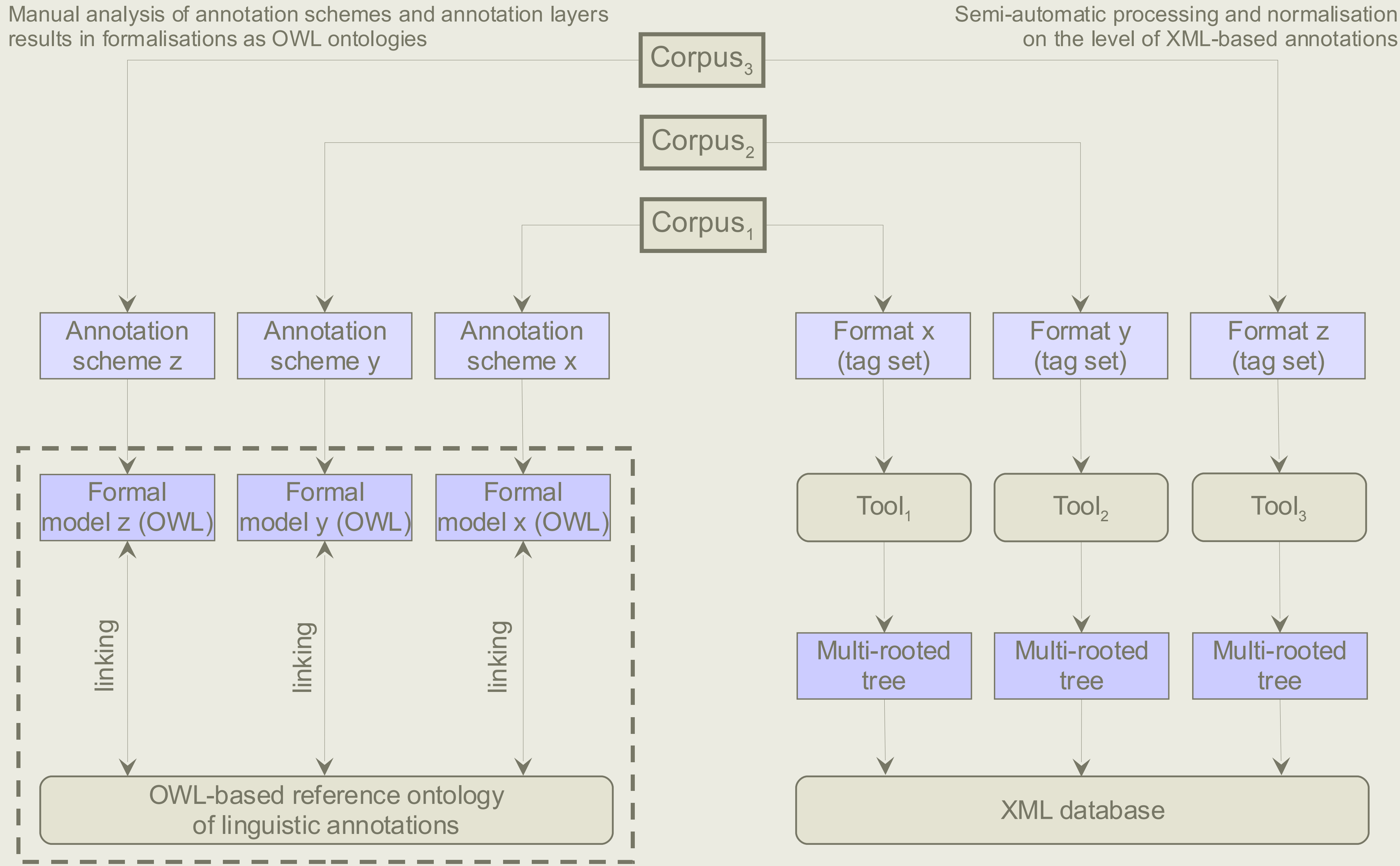
Linguistic Corpora

best case scenario

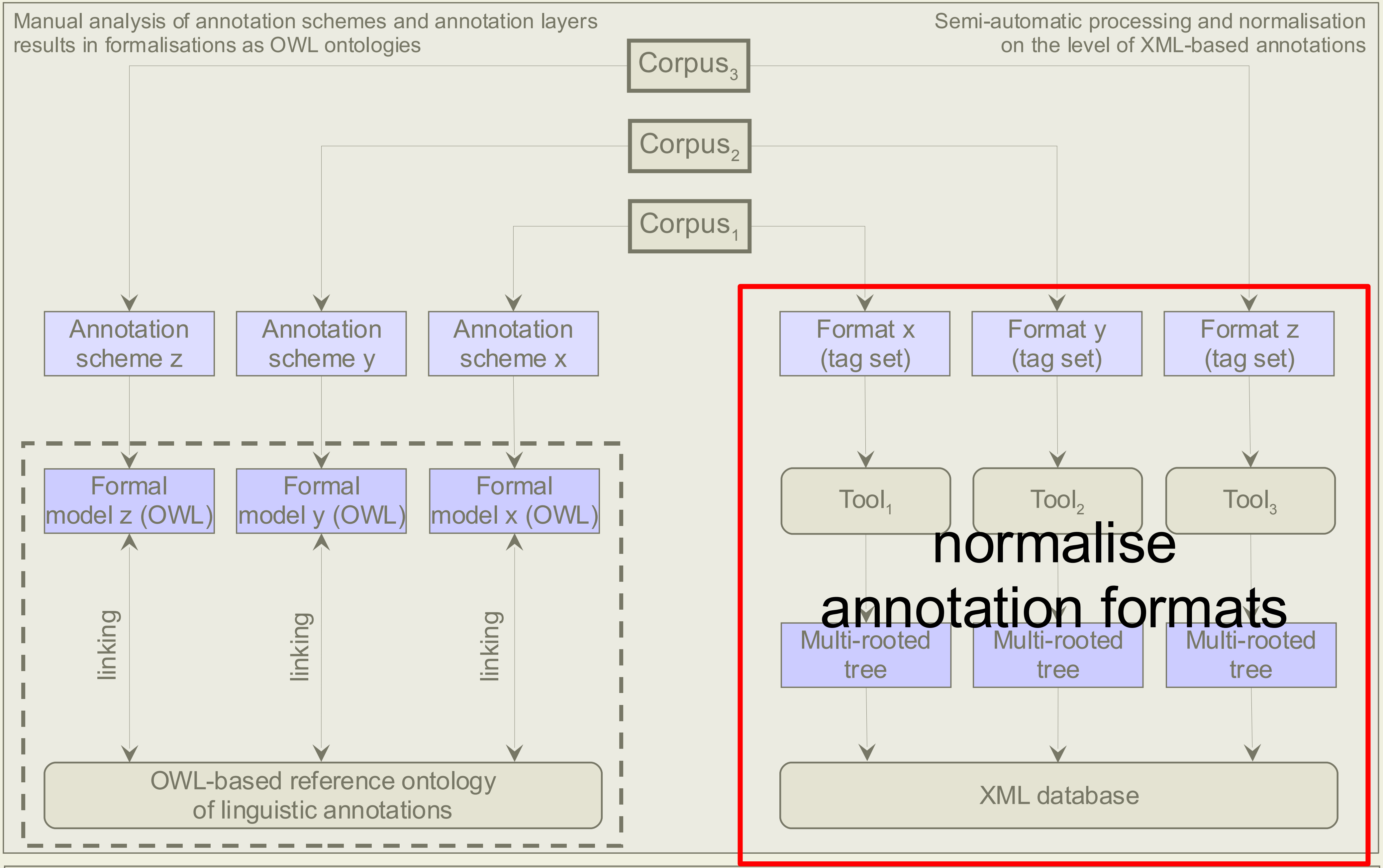
- Query against SPLICR
- SPLICR generalises over corpora
- Common visualisation/export modules



Processing and Normalisation of Corpus Data



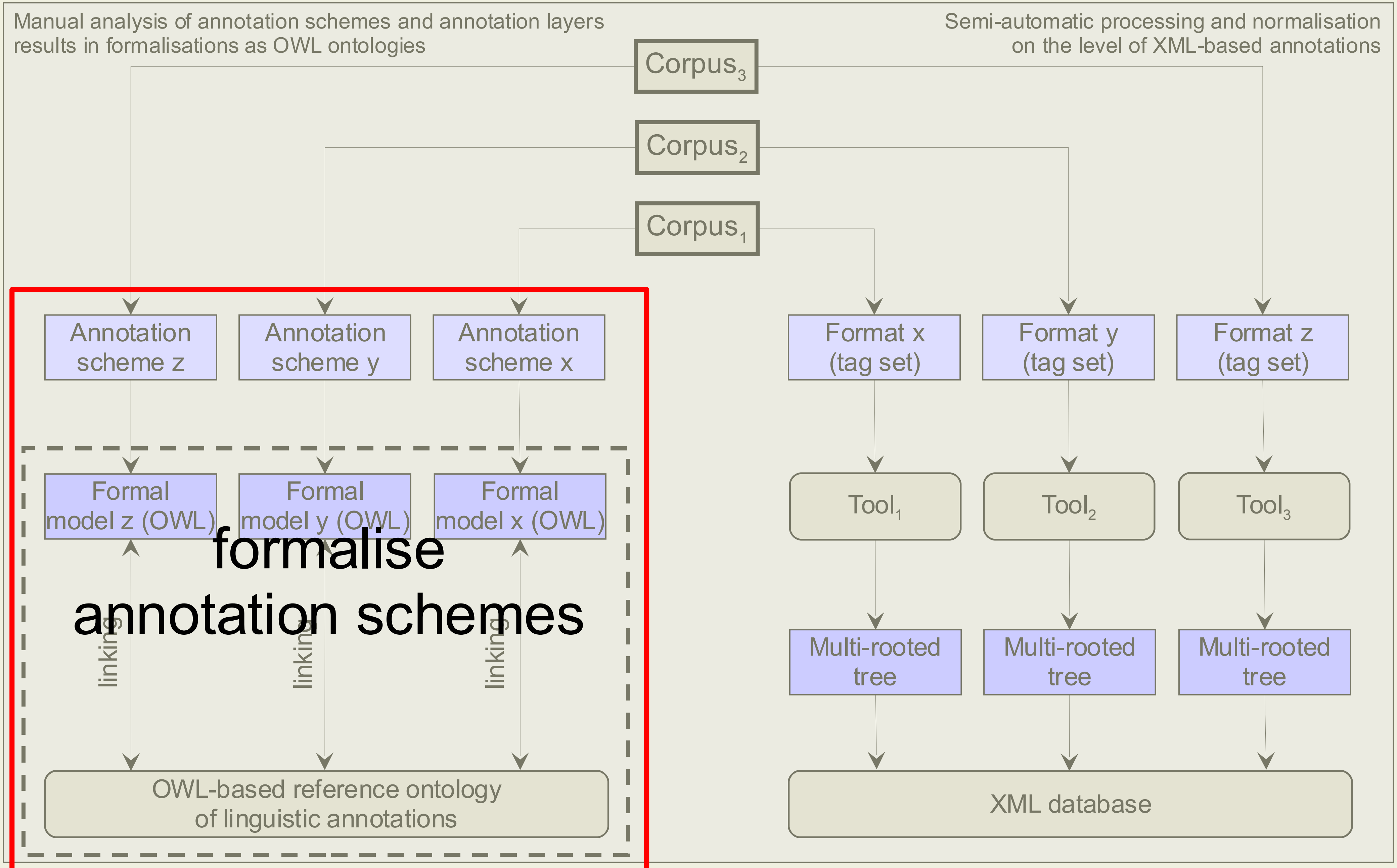
Processing and Normalisation of Corpus Data



Normalising Annotation Format

- Model: multi-rooted trees
- XML-encoded corpora split into multiple layers (trees)
  - One XML file per annotation layer
  - All are identical with regard to their primary data
- Normalizing the XML elements and attributes
  - Tool supported and flexibly configurable (Splitter, Leveler)
- Single layer can be queried with standard XML methods
- Multiple layers cannot be queried with standard methods
  - Introduce custom XQuery functions

Processing and Normalisation of Corpus Data



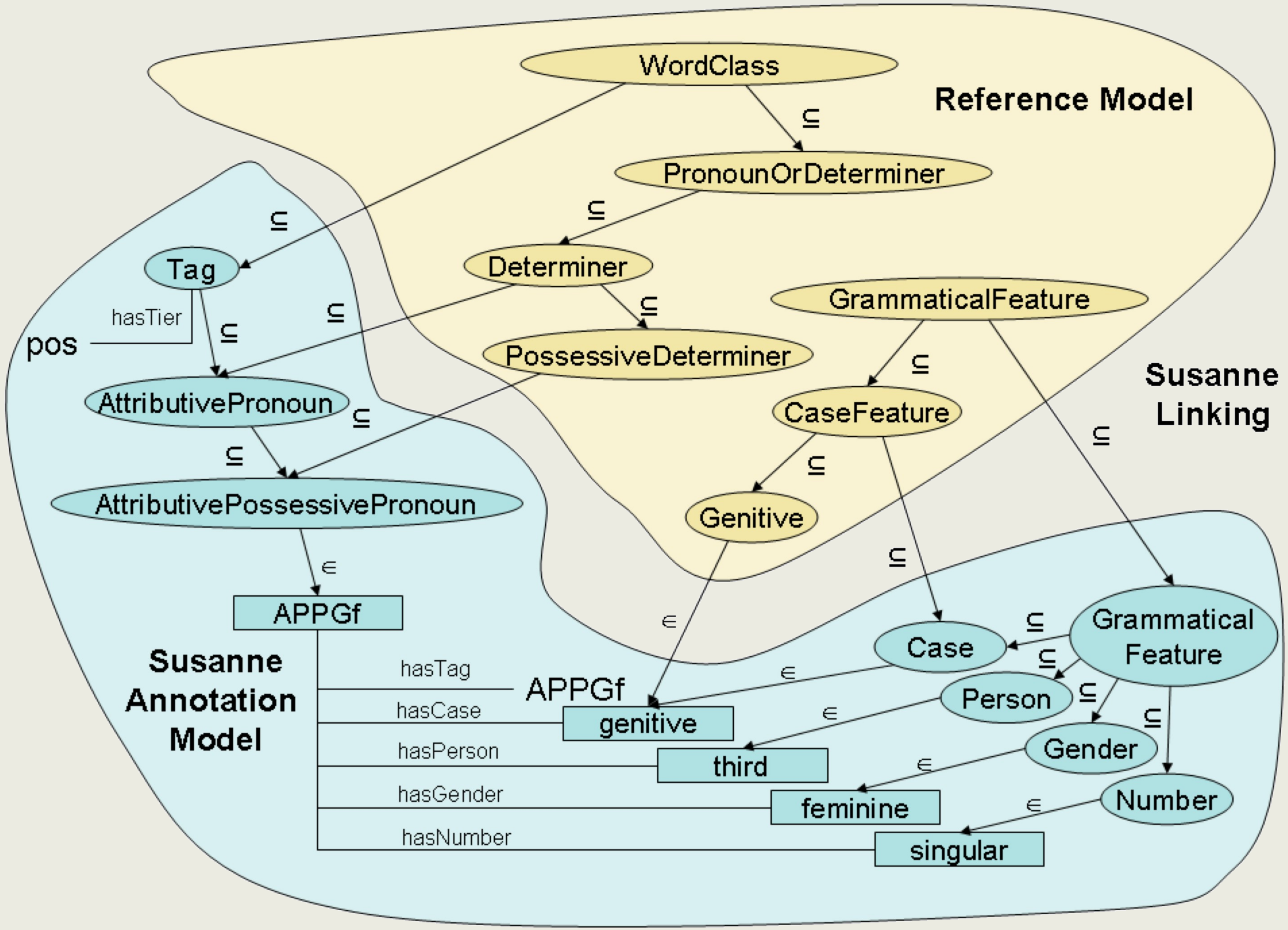
Formalising Annotation Semantics

- Corpora differ in their annotation schemes
- Integrated treatment of heterogeneous resources requires
  - Annotation specifics documented using a formal language
  - Integrated access to resources with different annotations
- Ontology-based approach
  - Ontological formalisation of annotation schemes
  - Standard format (OWL/DL)
  - Supported by several tools (Protégé, Pellet)

OLiA: Ontology of Linguistic Annotations

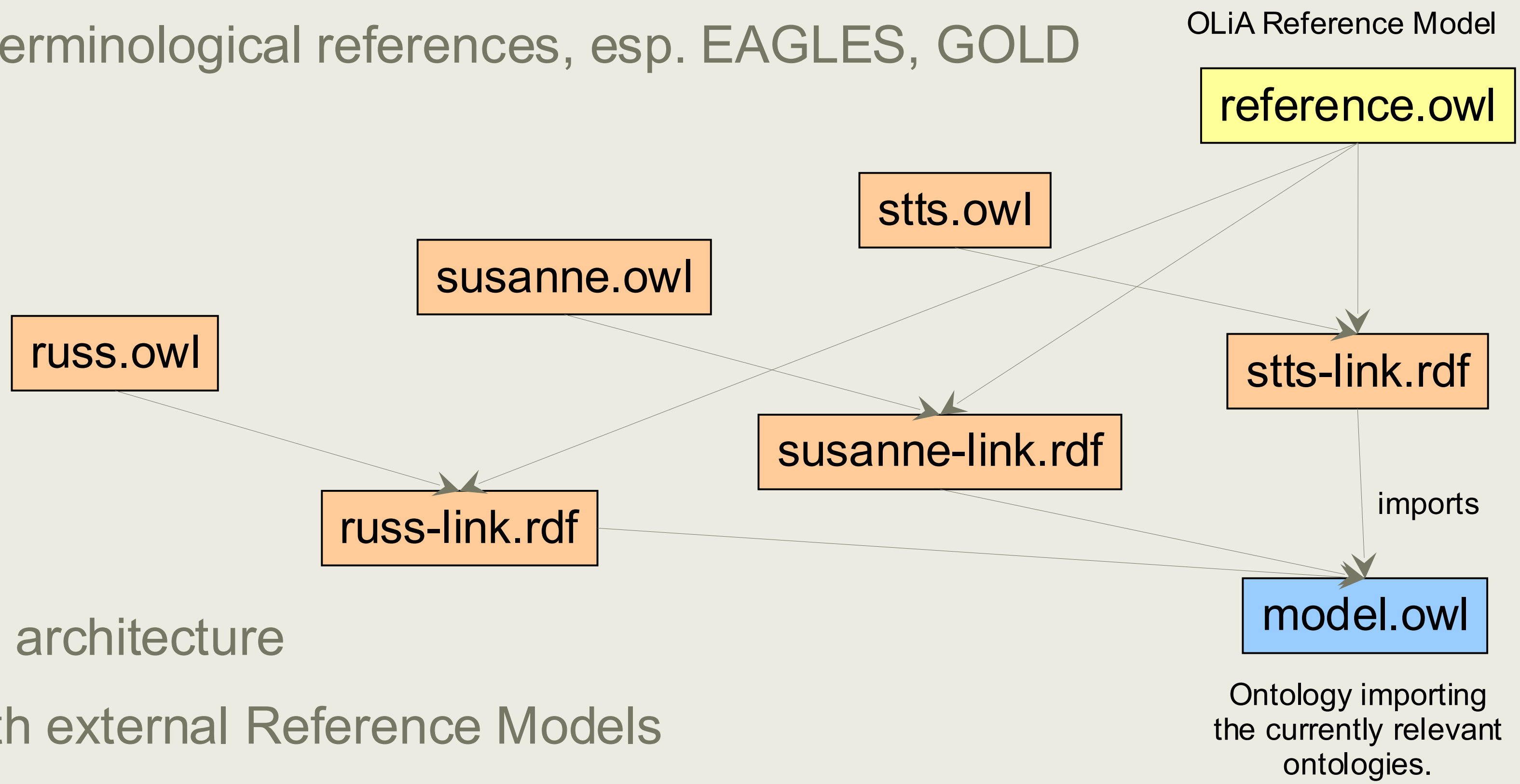
- Annotation Model
  - Ontological formalization of one particular annotation scheme
- OLiA Reference Model
  - Ontological formalization of reference terminology
- Linking
  - Concepts (and tags) of an annotation model are defined with reference to the OLiA Reference Model
    - Sub-concepts/sub-properties $\subseteq$ $\in$
    - Complex expressions $\cap$ $\cup$
- An example
  - POS tag APPGf “her” [Susanne Tagset]

OLiA: Ontology of Linguistic Annotations



OLiA: Ontology of Linguistic Annotations

- Annotation model
  - 10 models for European and non-European languages
  - POS, morphology, syntactic labels, co-reference, information structure
- OLiA Reference Model
  - Based on terminological references, esp. EAGLES, GOLD



- Linking
  - Extensible architecture
  - Linking with external Reference Models
  - (GOLD, OntoTag, Data Category Registry) supported

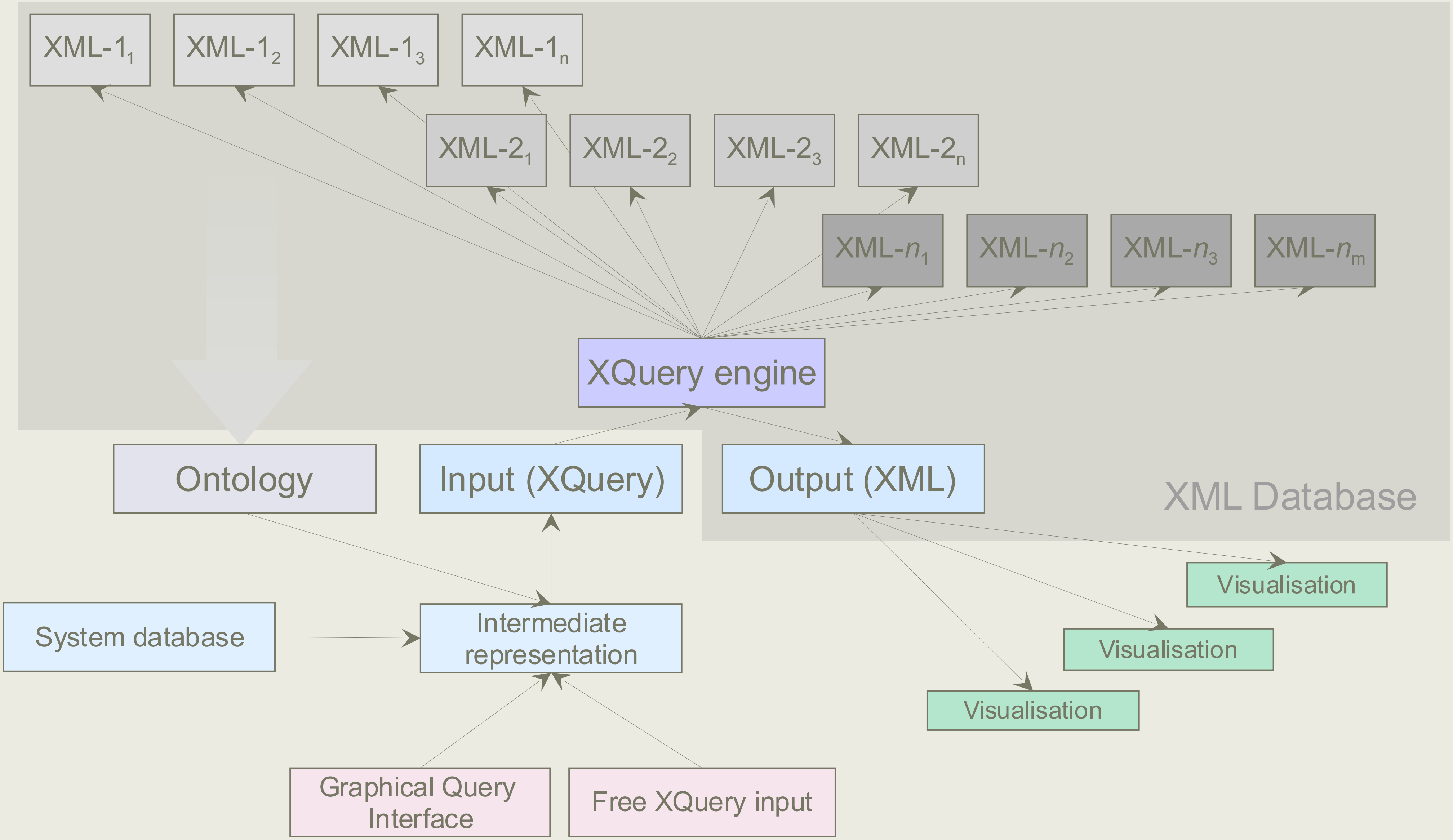
Graphical Query Interface Requirements

- Intuitively usable graphical query interface
- Work with multi-rooted trees
- Include the ontology of linguistic annotations into queries
- Work with open standards, i.e., XQuery, OWL

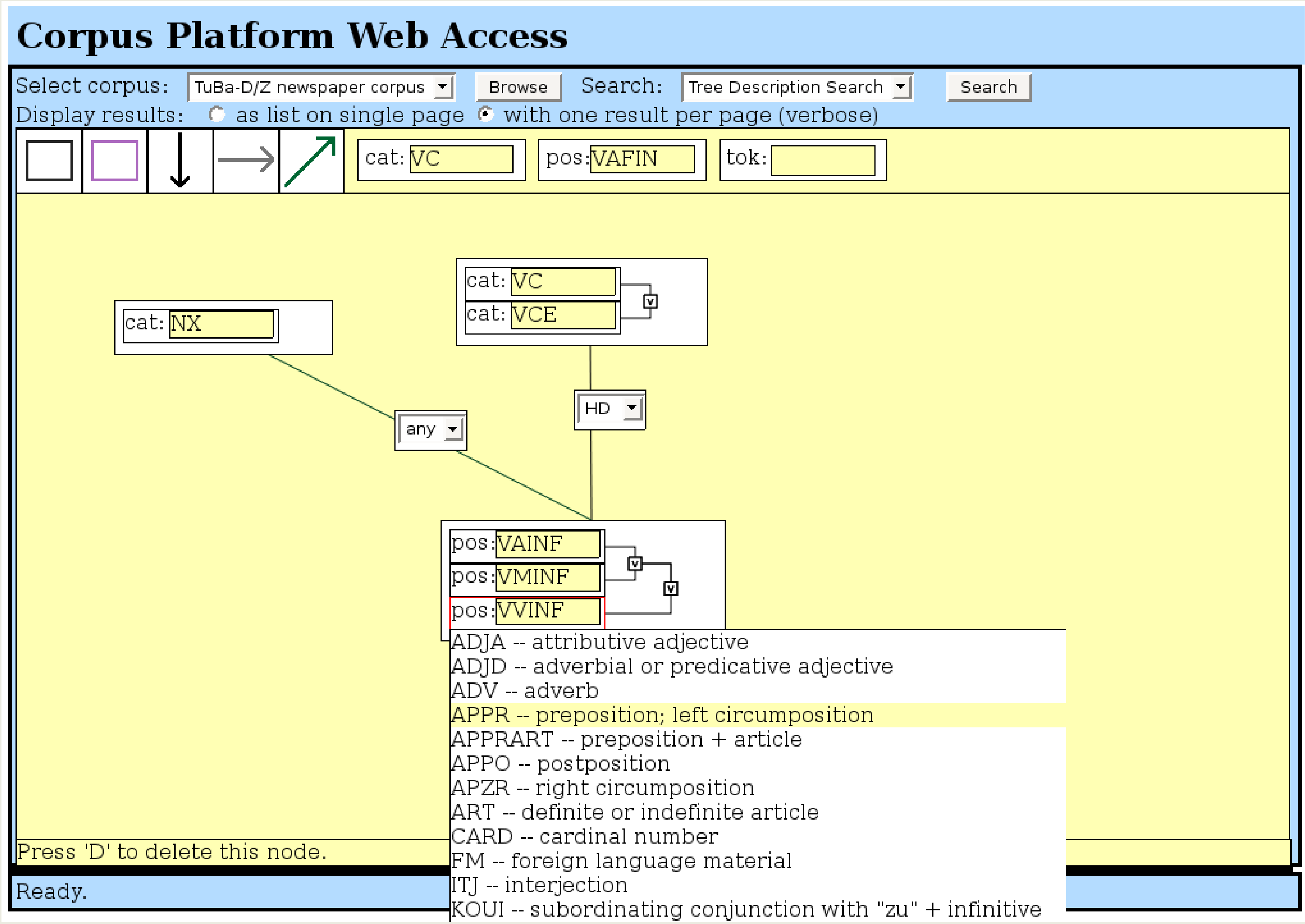
SPLICR Graphical Query Interface

- SPLICR has an intuitive graphical query interface
- Generalises over the underlying data structures and querying
- Tree fragment query editor
  - Ontology-supported abstraction of linguistic concepts
  - Operands glue together concepts to construct complex queries
- Multiple display and visualisation modes
  - plain text view
  - XML view
  - graphical tree view
  - time-line view
- Ajax (Asynchronous JavaScript and XML)
- Query and visualisation extensible through modules

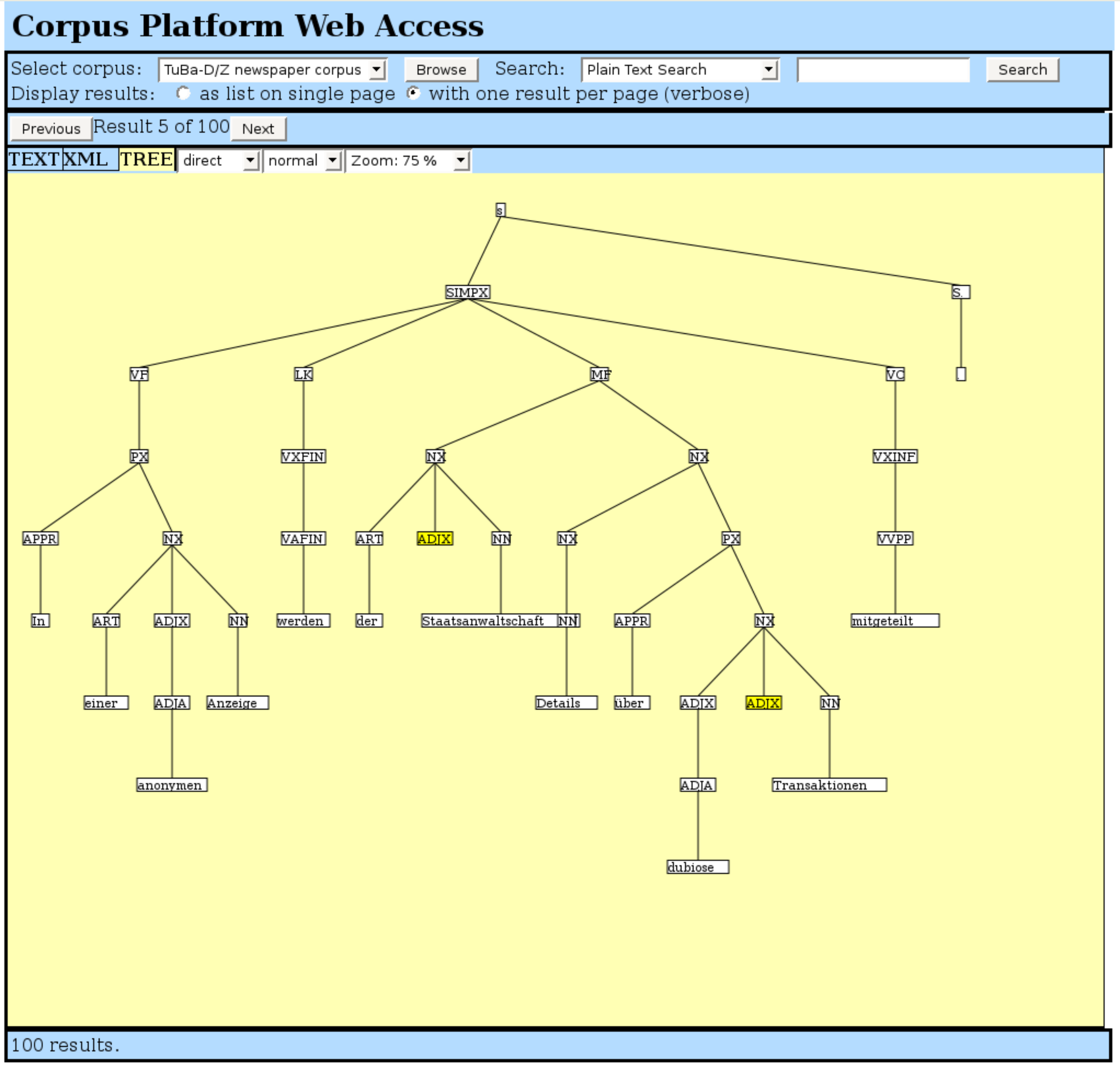
Querying


XML-1$_1$
XML-1$_2$
XML-1$_3$
XML-1$_n$
XML-2$_1$
XML-2$_2$
XML-2$_3$
XML-2$_n$
XML-n_1
XML-n_2
XML-n_3
XML-n_m
XQuery engine
Ontology
Input (XQuery)
Output (XML)
XML Database
Visualisation
Visualisation
Visualisation
System database
Intermediate representation
Graphical Query Interface
Free XQuery input

Tree Fragment Query Editor



Graphical Tree Visualisation



AnnoLab Multi-layer Query Example

- Lexical layer - find the verb *will* ('V')
- Field layer - find Vorfelds ('VF')
- Coordination - keep those Vorfelds containing *will* as a verb (seq:containing)

```
let $verb := ds:layer('Lexical')//tok
     [starts-with(pos/text,'V')]
     [.//orth = 'will']

let $vf := ds:layer('Field')//ntNode
     [category='VF']

return seq:containing($vf, $verb)
```

TUEBA1: Find the verb *will* in the Vorfeld

AnnoLab Multi-layer Query Example

- Lexical layer - find the verb *will* ('V')
- Field layer - find Vorfelds ('VF')
- Coordination - keep those Vorfelds containing *will* as a verb (seq:containing)

```
let $verb := ds:layer('Lexical')//tok
     [starts-with(pos/text,'V')]
     [.//orth = 'will']

let $vf := ds:layer('Field')//ntNode
     [category='VF']

return seq:containing($vf, $verb)
```

TUEBA2: Find the verb *will* in the Vorfeld

AnnoLab Multi-layer Query Example using OLiA

- Lexical layer - find the verb *will* ('V')
- Field layer - find Vorfelds ('VF')
- Coordination - keep those Vorfelds containing *will* as a verb (seq:containing)

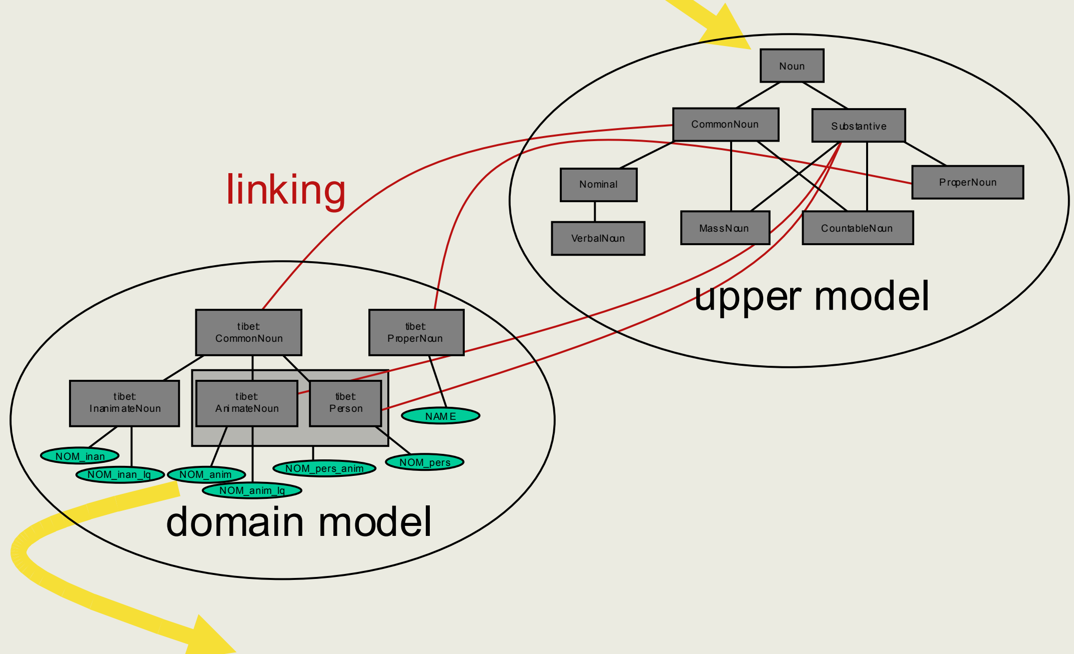
```
let $verb := ds:layer('Lexical')//tok
      [pos/text = oc:expand('Verb')]
      [.//orth = 'will']

let $vf := ds:layer('Field')//ntNode
      [category='VF']

return seq:containing($vf, $verb)
```

TUEBA2: Find the verb *will* in the Vorfeld using OLiA corpus query `... oc:expand('Noun') ...`

ontology lookup:
1. instance retrieval
2. application of set operators

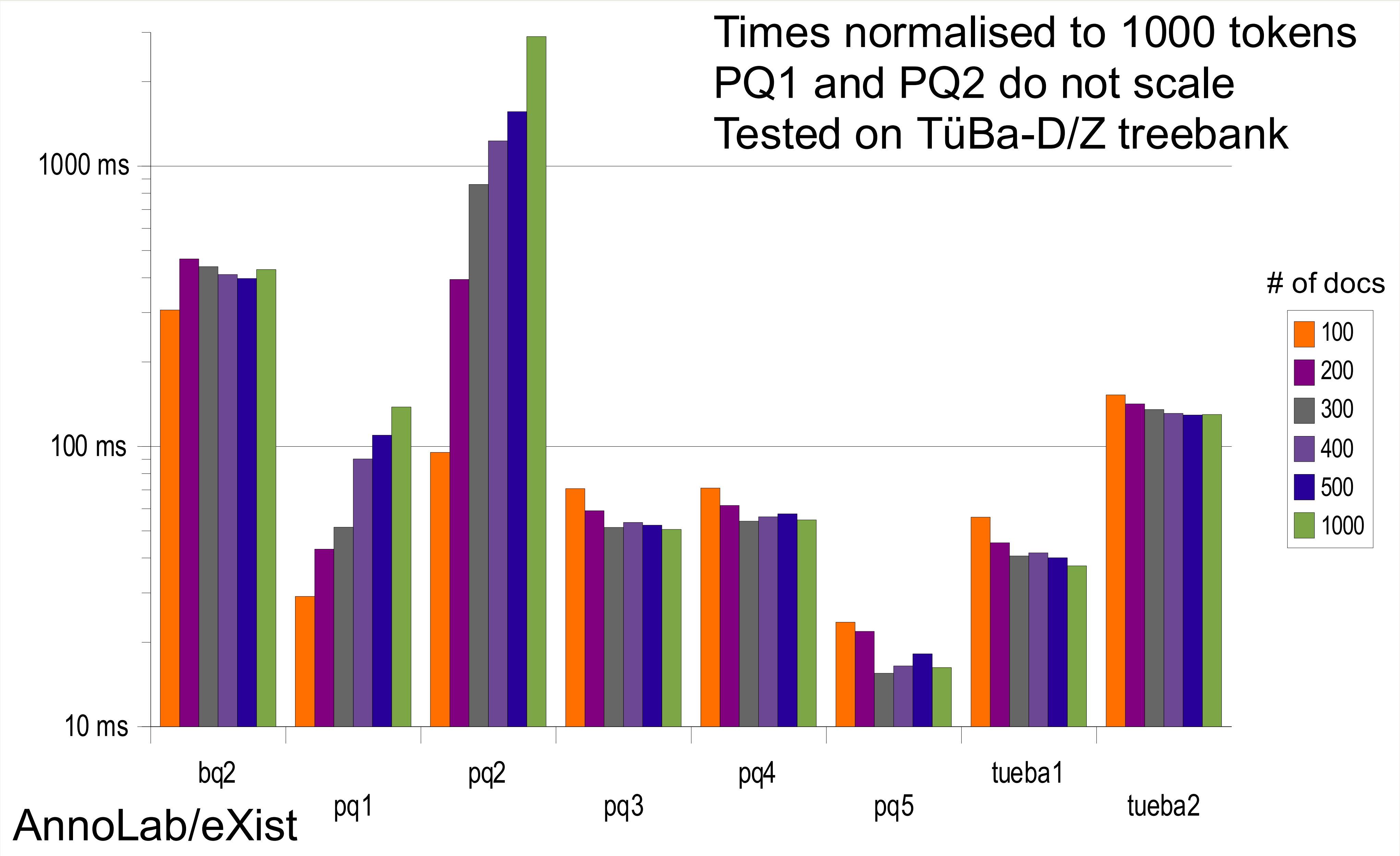


`... NOM_inan | NOM_inan_lq | NOM_anim | NOM_anim_lq | NOM_anim_pers | NOM_pers | NAME ...`

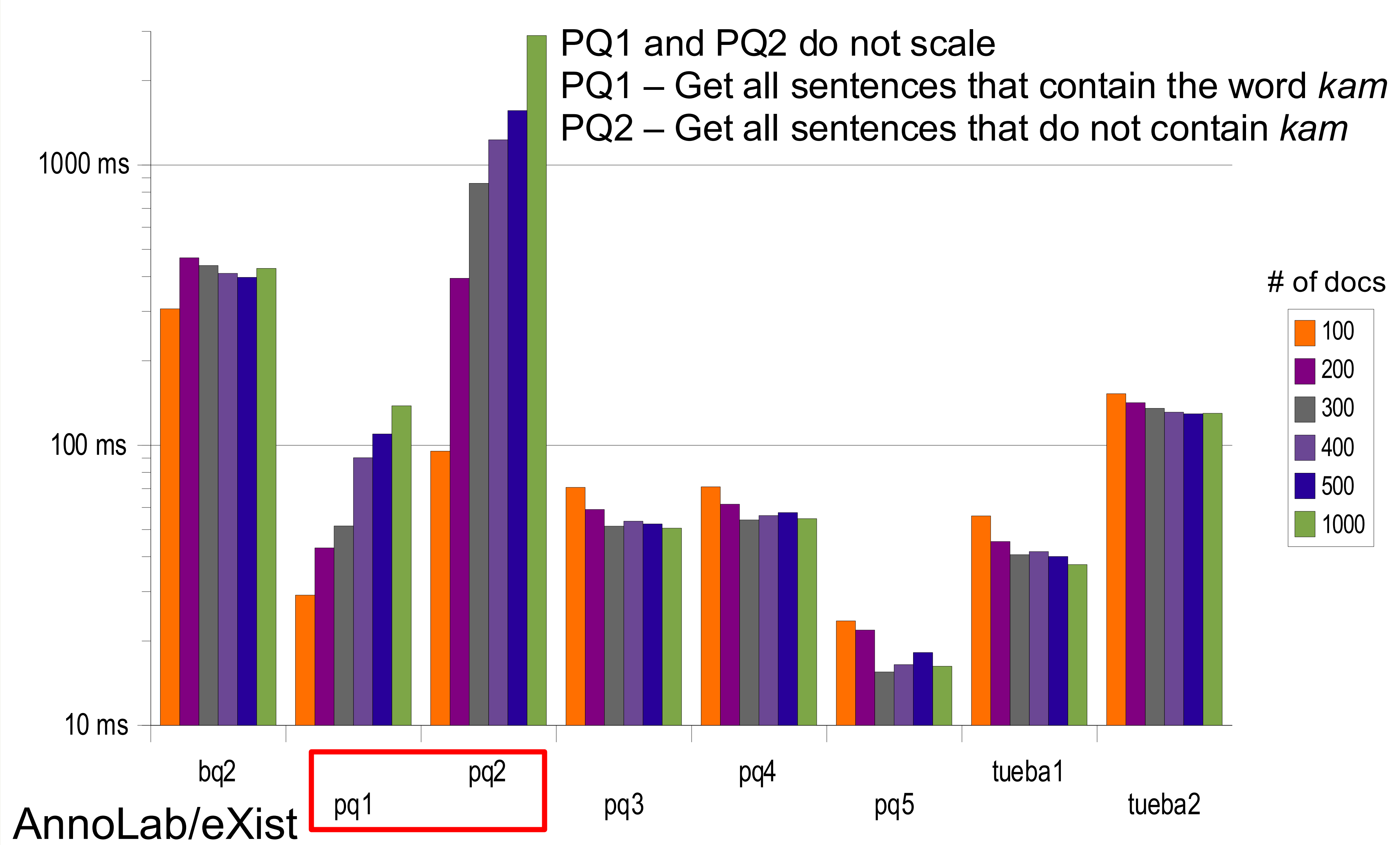
Experimentation queries

- PQ1 – Get all sentences that contain the word *kam*
- PQ2 – Get all sentences that do not contain *kam*
- PQ3 – Get references to all NPs
- PQ4 – Get all subtrees dominated by NPs
- PQ5 – Get all NPs subtrees dominated by a VP
- TUEBA1 – Find all occurrences of the verb *will* in the Vorfeld
- TUEBA2 – TUEBA1 using OLiA
- BQ2 – Get NPs that are immediate following siblings of a verb

Average Query Run-Time (logarithmic)



Average Query Run-Time (logarithmic)



Summary

- Approach to querying XML-annotated corpora using standard techniques such as XPath and XQuery
- Extended an XML database to query multi-rooted trees
- Built an OWL ontology of linguistic annotations generalising over annotation schemes and tag sets
- OWL ontology can be used for query expansion
- Implemented an intuitive and flexible graphical query interface

Conclusions and Future Work

- Work on SPLICR is ongoing
- Building the GUI to explore and to query meta-data
- Extended query interface functionality (e.g. saved searches)
- Working on benchmark queries for evaluating XML databases with respect to linguistic corpora