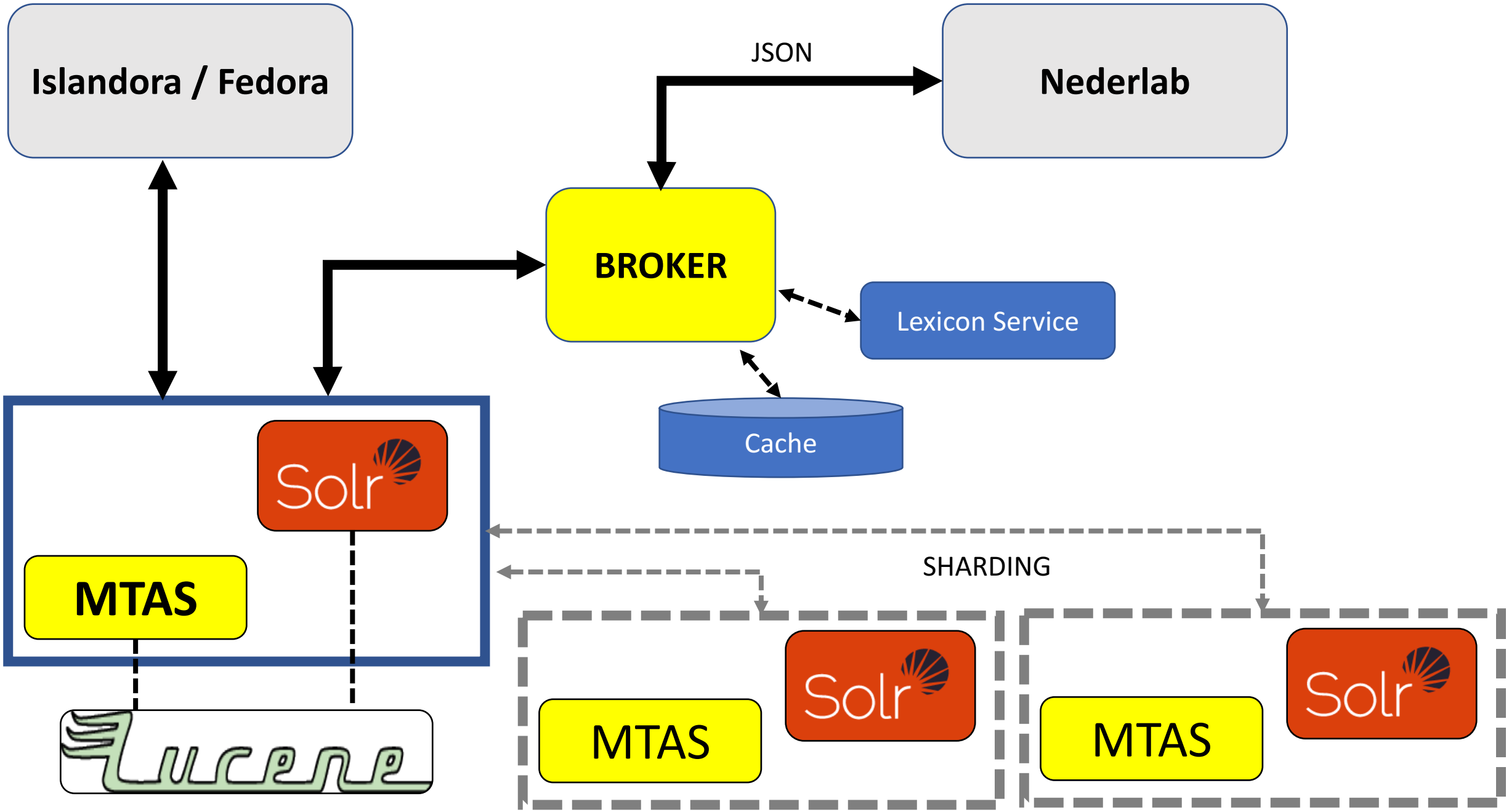
MTAS

Multi Tier Annotation Search

introduction

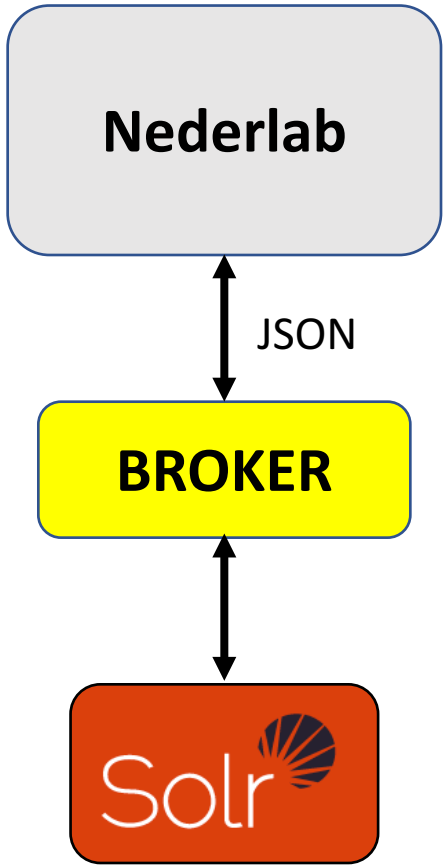
Matthijs Brouwer


Islandora / Fedora
JSON
Nederlab
BROKER
Lexicon Service
Cache
Solr
MTAS
SHARDING
Solr
MTAS
Solr
MTAS

- Scalable
- Open Source
- Search in Metadata and Text
- Indexation process easy
- Good performance

```
<text xml:id="WR-P-E-J-0000000001.text">
    <div xml:id="WR-P-E-J-0000000001.div0.1" class="chapter">
      <head xml:id="WR-P-E-J-0000000001.head.1">
        <s xml:id="WR-P-E-J-0000000001.head.1.s.1">
          <w xml:id="WR-P-E-J-0000000001.head.1.s.1.w.1">
            <t>Stemma</t>
            <pos class="N(soort,ev,basis,onz,stan)"/>
            <lemma class="stemma"/>
          </w>
        </s>
      </head>
      <p xml:id="WR-P-E-J-0000000001.p.1" class="firstparagraph">
        <s xml:id="WR-P-E-J-0000000001.p.1.s.1">
          <w xml:id="WR-P-E-J-0000000001.p.1.s.1.w.1">
            <t>Stemma</t>
            <pos class="N(eigen,ev,basis,zijd,stan)" />
```

```
Groni_ge_ groni_ge_ groningen groningen 020 3 - -
doen doen doen doen 204 0 - -
kundich kundich kundich kondig 100 4 - -
allen allen allen al 324 2 - -
luden luden luden man 014 2 - -
myt myt myt met 700 0 - -
dessen dessen dessen deze 414 3 - -
opene_ opene_ openen open 104 3 - -
breue breue breue brief 001 3 - -
dat dat dat dat 810 0 - 8
```



- **Annotations & Structure**
- **Search and Analysis**
- **Scalable**
- Lucene based
- Solr plugin with support sharding

MTAS

ALTERNATIVES

MTAS

- Lucene Index Structure
- Query Parsers
- Parsers Resources
- Solr integration

CQL

```
<text xml:id="WR-P-E-J-0000000001.text">
    <div xml:id="WR-P-E-J-0000000001.div0.1" class="chapter">
      <head xml:id="WR-P-E-J-0000000001.head.1">
        <s xml:id="WR-P-E-J-0000000001.head.1.s.1">
          <w xml:id="WR-P-E-J-0000000001.head.1.s.1.w.1">
            <t>Stemma</t>
            <pos class="N(soort,ev,basis,onz,stan)"/>
            <lemma class="stemma"/>
          </w>
        </s>
      </head>
      <p xml:id="WR-P-E-J-0000000001.p.1" class="firstparagraph">
        <s xml:id="WR-P-E-J-0000000001.p.1.s.1">
          <w xml:id="WR-P-E-J-0000000001.p.1.s.1.w.1">
            <t>Stemma</t>
            <pos class="N(eigen,ev,basis,zijd,stan)" />
```

Index Structure

title: Max Havelaar
year: 1860
text: Ik ben makelaar in koffie, en woon op de Lauriergracht …

Document	Field	Term	Position	Payload
1	title	Max Havelaar	-	-
1	year	1860	-	-
1	text	Ik	0	-
1	text	ben	1	-
1	text	makelaar	2	-
1	text	in	3	-
1	text	koffie	4	-
1	text	,	5	-
1	text	en	6	-
1	text	woon	7	-
…	…	…	…	…

Index is stored **sorted by field and term**, allowing fast retrieval of documents (and positions) for a given term

Multiple terms on the same position are allowed

Payload available to store additional information

MTAS

POS, lemma, features

Sentences, paragraphs

Named entities, dependencies

- Encode prefix and postfix as term
- Use payload to
  - Provide each token with an id
  - Allow non-single positioned items (sets of positions / ranges)
  - Store hierarchical relations

Document	Field	Term	Position	Payload

Document	Field	TokenId	Prefix	Postfix	Position	ParentId	Payload
1	text	0	s	-	0 - 10	-	-
1	text	1	t	lk	0	0	-
1	text	2	pos	VNW	0	0	-
1	text	3	t	ben	1	0	-
…	…	…	…	…	…	…	…

MTAS

- Extended Lucene Codec
- Forward index on position, parent and id

- Keep existing functionalities
- Add, delete and update documents
- Merge segments
- Merge cores

```
_nt.fdt
_nt.fdx
_nt.fnm
_nt.nvd
_nt.nvm
_nt.si
_nt_Lucene50_0.doc
_nt_Lucene50_0.pay
_nt_Lucene50_0.pos
_nt_Lucene50_0.tim
_nt_Lucene50_0.tip
_nt_MtasCodec_0.doc
_nt_MtasCodec_0.mtas.doc
_nt_MtasCodec_0.mtas.field
_nt_MtasCodec_0.mtas.index.doc.id
_nt_MtasCodec_0.mtas.index.object.id
_nt_MtasCodec_0.mtas.index.object.parent
_nt_MtasCodec_0.mtas.index.object.position
_nt_MtasCodec_0.mtas.object
_nt_MtasCodec_0.mtas.prefix
_nt_MtasCodec_0.mtas.term
_nt_MtasCodec_0.pay
_nt_MtasCodec_0.pos
_nt_MtasCodec_0.tim
_nt_MtasCodec_0.tip
```

PARSERS

RESOURCES

```
Groni_ge_ groni_ge_ groningen groningen 020 3 - -
doen doen doen doen 204 0 - -
kundich kundich kundich kondig 100 4 - -
allen allen allen al 324 2 - -
luden luden luden man 014 2 - -
myt myt myt met 700 0 - -
dessen dessen dessen deze 414 3 - -
opene_ opene_ openen open 104 3 - -
breue breue breue brief 001 3 - -
dat dat dat dat 810 0 - 8
```

```
<text xml:id="
  <div xml:i
    <head xm
      <s xml:
        <w x
          <t
          <p
          <l
        </w>
      </s>
    </head>
    <p xml:id="WR-P-E-J-0000000001.p.1" class="firstparagraph">
      <s xml:id="WR-P-E-J-0000000001.p.1.s.1">
        <w xml:id="WR-P-E-J-0000000001.p.1.s.1.w.1">
          <t>Stemma</t>
          <pos class="N(eigen,ev,basis,zijd,stan)" />
```

QUERY

<s/> containing ([pos="ADJ"] [lemma="Amsterdam"])

CQL

MTAS

Document	Field	TokenId	Prefix	Postfix	Position	ParentId	Payload
1	text	0	s	-	0 - 10	-	-
1	text	1	t	lk	0	0	-
1	text	2	pos	VNW	0	0	-
1	text	3	t	ben	1	0	-
…	…	…	…	…	…	…	…

CQL - Corpus Query Language

PARSERS

[pos = “VNW”]

- Single position
- Prefix equals “pos”
- Postfix equals “VNW”

<s/>

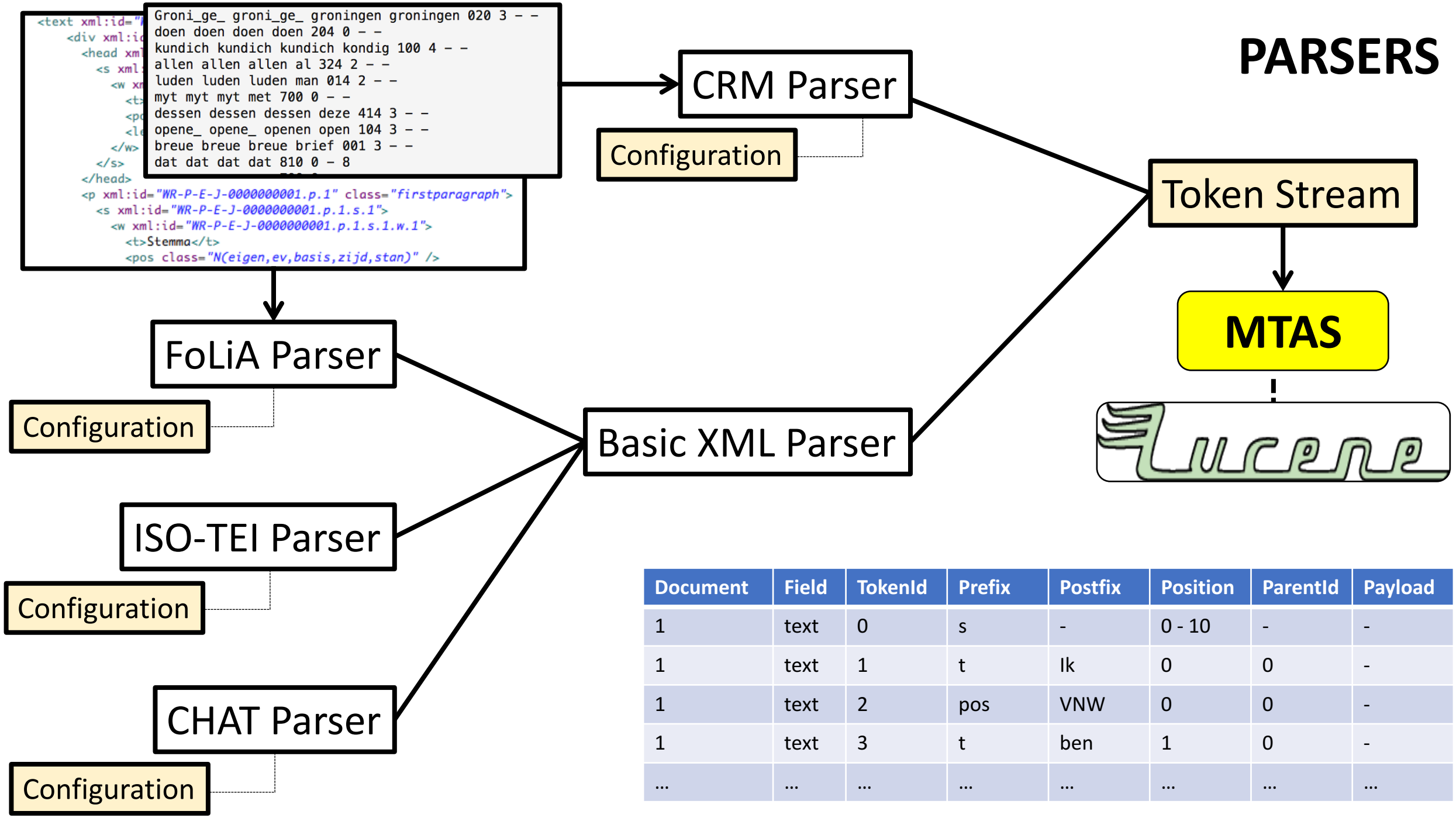
- Range
- Prefix equals “s”

<entity=“loc”/> within (<s/> containing [lemma=“amsterdam”])

Document	Field	TokenId	Prefix	Postfix	Position	ParentId	Payload
1	text	0	s	-	0 - 10	-	-
1	text	1	t	Ik	0	0	-
1	text	2	pos	VNW	0	0	-
1	text	3	t	ben	1	0	-
…	…	…	…	…	…	…	…

ParentId not covered by CQL

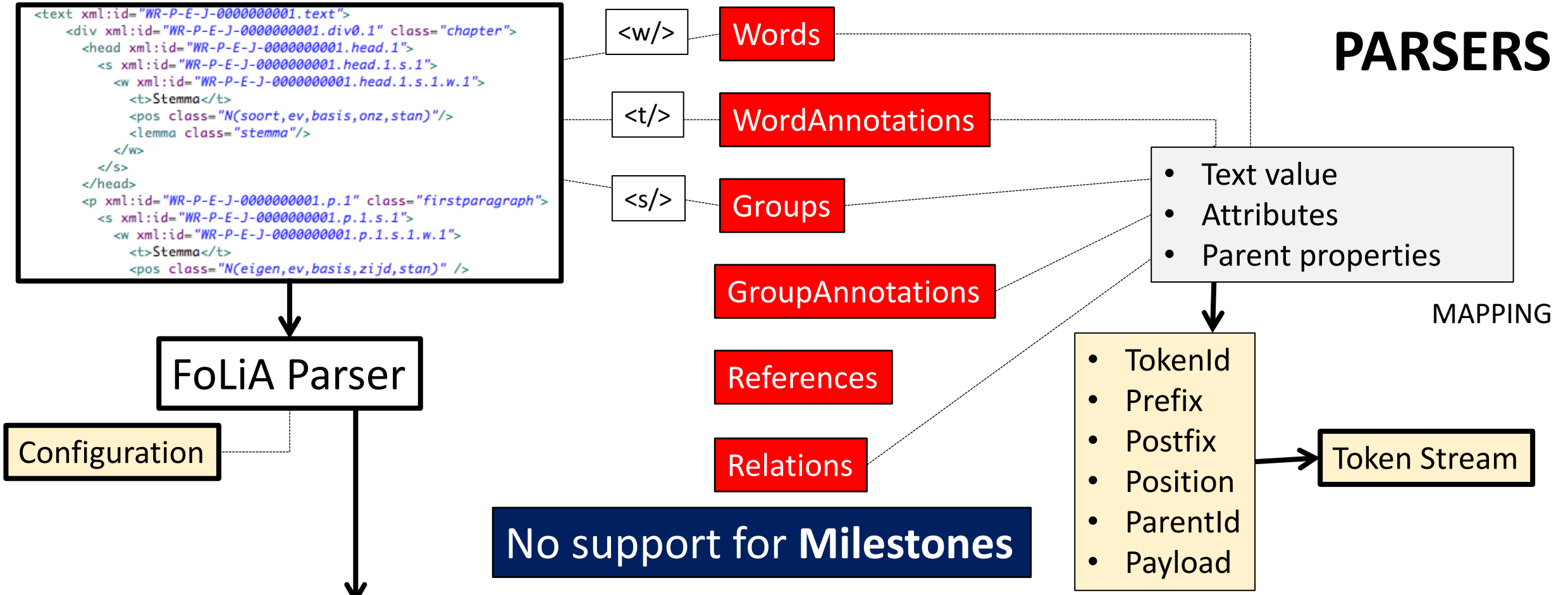
PARSERS



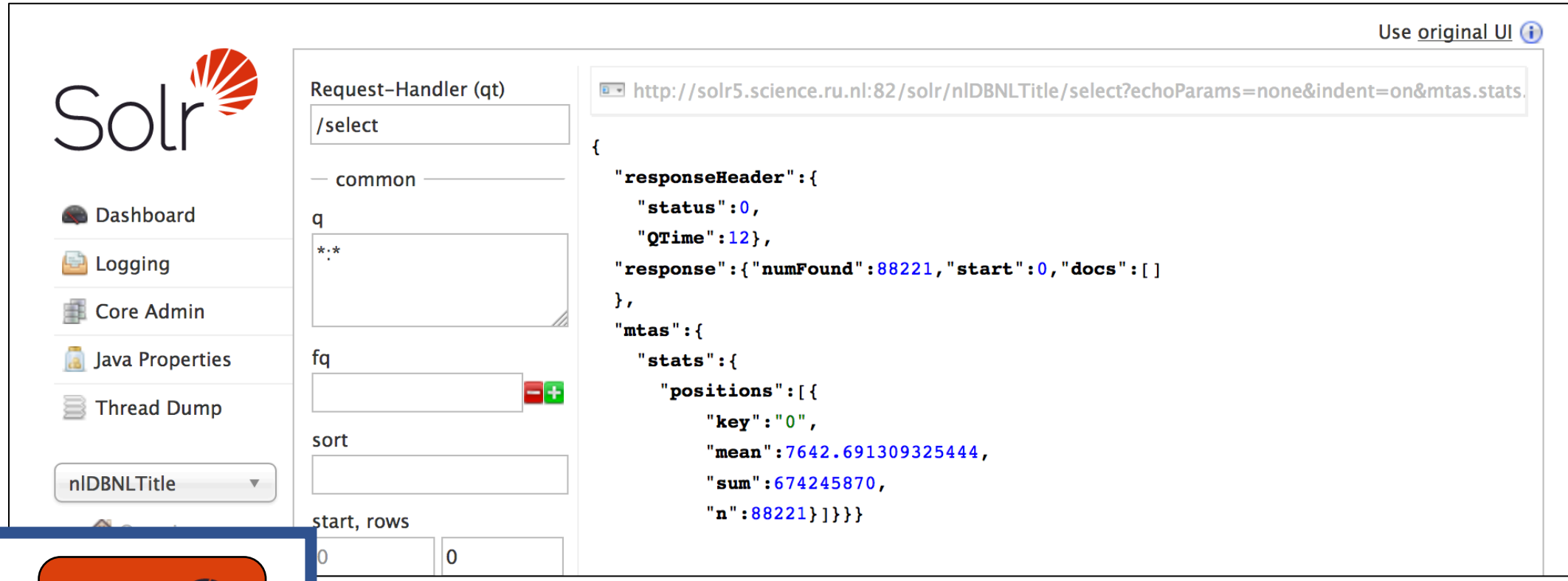
Document	Field	TokenId	Prefix	Postfix	Position	ParentId	Payload
1	text	0	s	-	0 - 10	-	-
1	text	1	t	lk	0	0	-
1	text	2	pos	VNW	0	0	-
1	text	3	t	ben	1	0	-
…	…	…	…	…	…	…	…

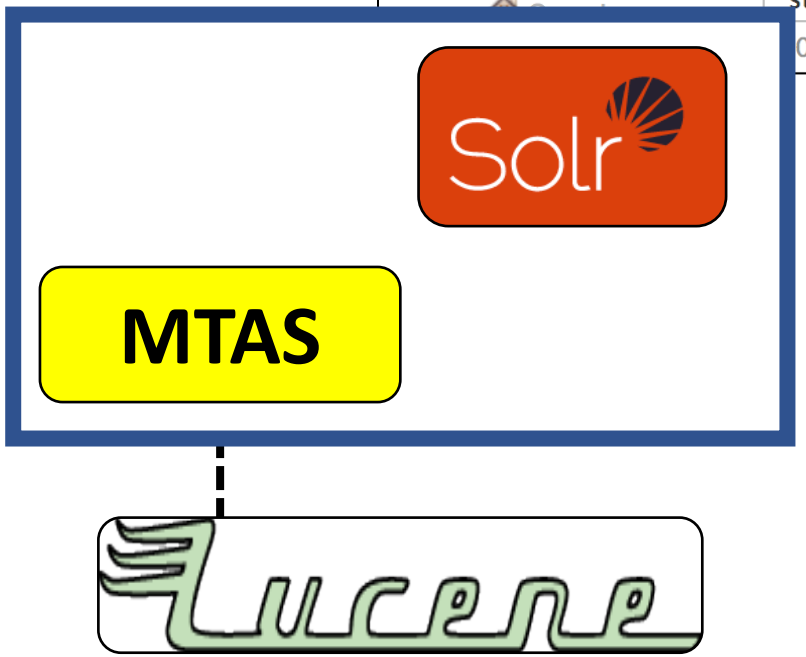
PARSERS

```
<text xml:id="WR-P-E-J-0000000001.text">
    <div xml:id="WR-P-E-J-0000000001.div0.1" class="chapter">
      <head xml:id="WR-P-E-J-0000000001.head.1">
        <s xml:id="WR-P-E-J-0000000001.head.1.s.1">
          <w xml:id="WR-P-E-J-0000000001.head.1.s.1.w.1">
            <t>Stemma</t>
            <pos class="N(soort,ev,basis,onz,stan)"/>
            <lemma class="stemma"/>
          </w>
        </s>
      </head>
      <p xml:id="WR-P-E-J-0000000001.p.1" class="firstparagraph">
        <s xml:id="WR-P-E-J-0000000001.p.1.s.1">
          <w xml:id="WR-P-E-J-0000000001.p.1.s.1.w.1">
            <t>Stemma</t>
            <pos class="N(eigen,ev,basis,zijd,stan)" />
```



Document	Field	TokenId	Prefix	Postfix	Position	ParentId	Payload
1	text	0	s	-	0 - 10	-	-
1	text	1	t	lk	0	0	-
1	text	2	pos	VNW	0	0	-
1	text	3	t	ben	1	0	-
…	…	…	…	…	…	…	…





- Use Mtas as a Solr Plugin
- Add field(s) to Solr schema for Mtas content
- Define mapping

POST variables to http://localhost:8983/solr/core0/select

- q=*:*
- rows=0
- wt=json
- indent=on
- echoParams=none
- mtas=true
- mtas.termvector=true
- mtas.termvector.0.field=mtasTextField
- mtas.termvector.0.prefix=t
- mtas.termvector.0.sort.type=sum
- mtas.termvector.0.sort.direction=desc


```
{
  "responseHeader":{
    "status":0,
    "QTime":3346},
  "response":{"numFound":88221,"start":0,"docs":[]
  },
  "mtas":{
    "termvector":[{
        "key":"0",
        "list":[{
            "mean":5030.202361607627,
            "sum":46433798,
            "n":9231,
            "key":","},
          {
            "mean":3395.675161987041,
            "sum":31443952,
            "n":9260,
            "key":"."},
          {
            "mean":2336.4308818708337,
            "sum":21380679,
            "n":9151,
            "key":"de"},
          {
            "mean":1613.0381202760434,
            "sum":14725425,
            "n":9129,
            "key":"van"},
```


Solr

MTAS

POST variables to http://localhost:8983/solr/core0/select

```
q=year:[* TO 1925] AND {!mtas_cql field="mtasTextField" query="[lemma=\"amsterdam\"]"}
```

For the selected set of documents

- Statistics on number of positions and annotations
- Statistics on spans matching a (CQL-)query
- KWIC representation
- Termvector (Frequency list)
- Facets (statistics over values metadata)
- Grouping

Combination with Metadata

Support sharding

Solr

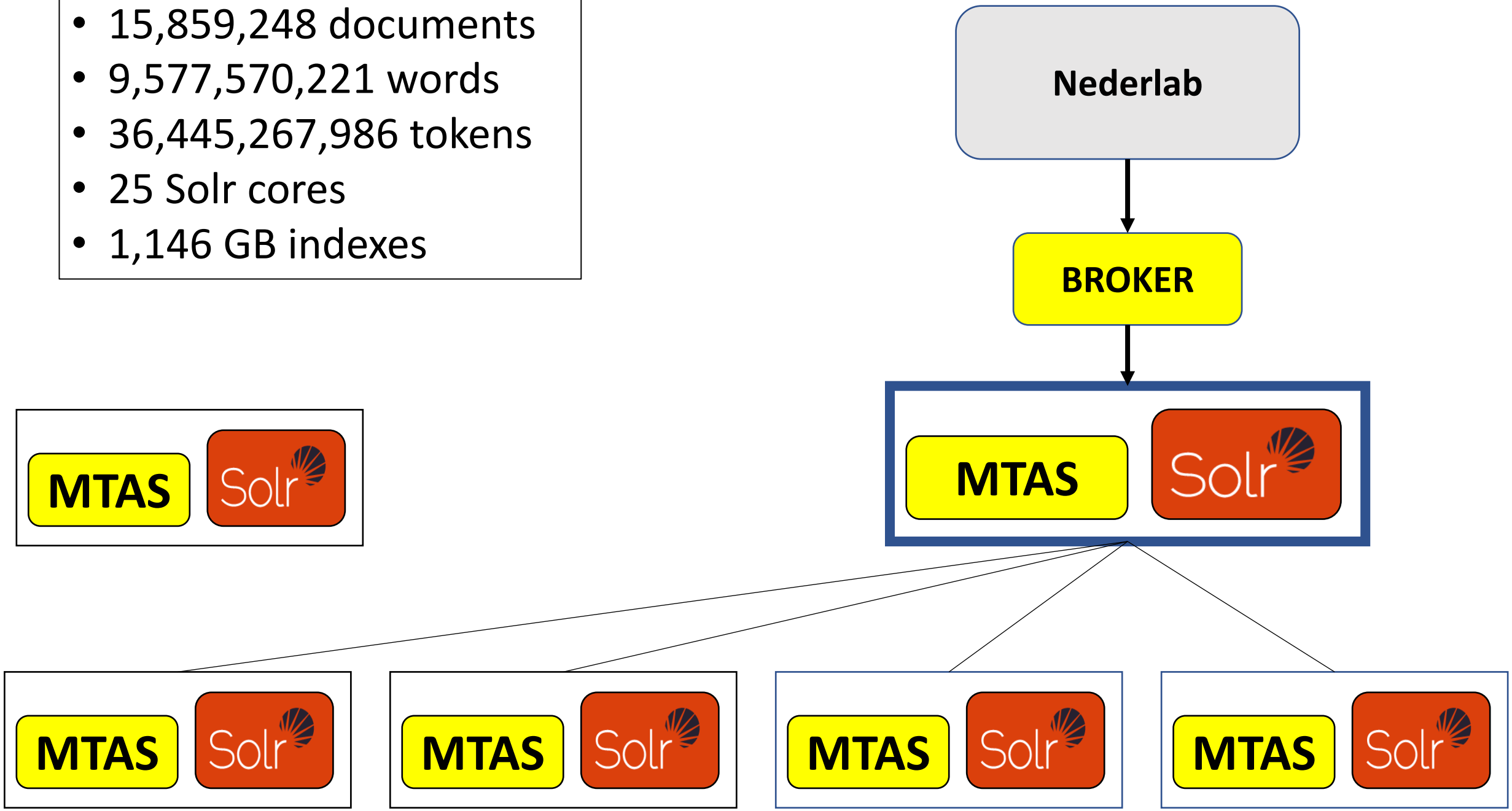
MTAS

Index documents / resources :

Support multiple resource types

Optimizing and merging cores

- 15,859,248 documents
- 9,577,570,221 words
- 36,445,267,986 tokens
- 25 Solr cores
- 1,146 GB indexes



Future work and interests

- Efficient topic modelling techniques
- Apply results for further analysis
- Ranking / score
- Local alignment
- Termvector over multiple layers
- Sort termvector by TFIDF
- Query language to explore hierarchical relations
- Loosely coupled tokenizations

Create matrices instead of huge amounts of frequency lists

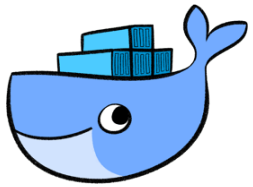
Use factorization to refine document set / allocate weights

Adjustments index structure

Source and documentation on GitHub

https://github.com/meertensinstituut/mtas

Docker image with demo scenarios available



QUESTIONS ?