CLARIN Federated Content Search Specification and Software Components

Oliver Schonefeld
Institut für Deutsche Sprache, Mannheim
schonefeld@ids-mannheim.de

GEFÖRDERT VOM
Bundesministerium
für Bildung
und Forschung

Overview

- Overview of CLARIN-FCS architecture
- (Brief) Introduction to SRU/CQL
- (Brief) CLARIN-FCS interface specification
- Available Software components

Overview of CLARIN-FCS architecture

Goals of CLARIN-FCS

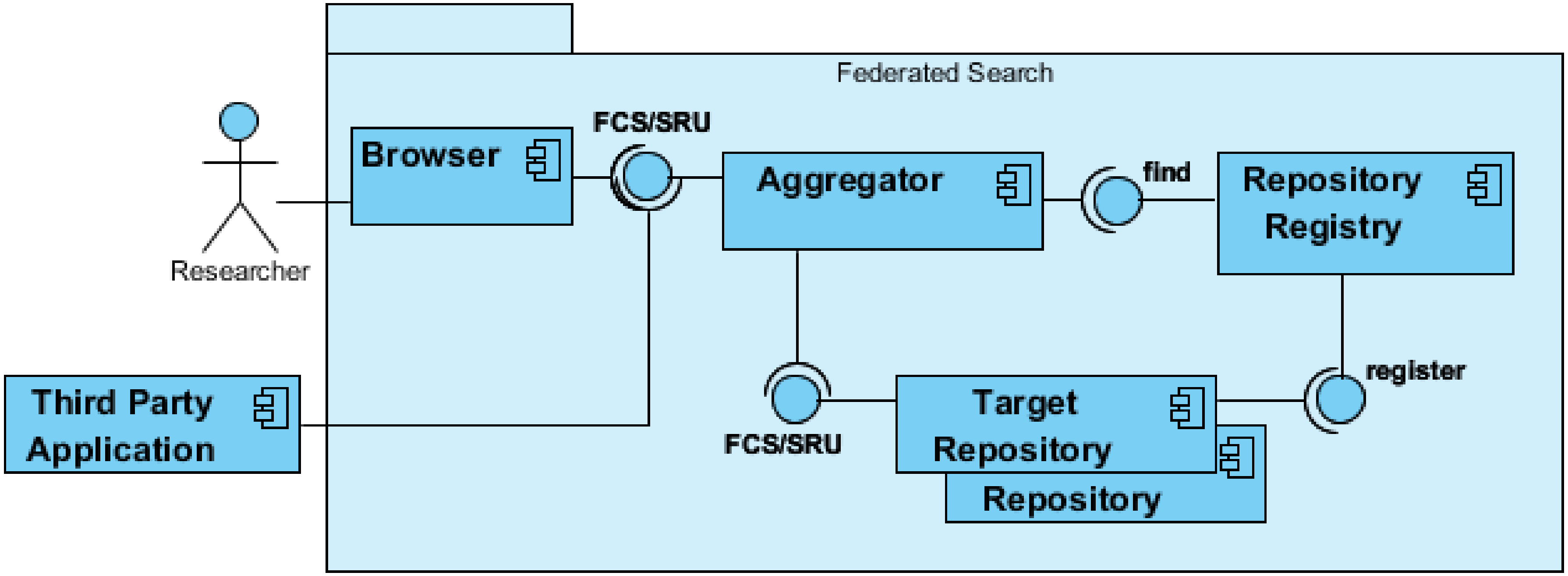
“The *main goal* of the Federated Content Search (FCS) is to introduce a common protocol, to decouple the search engine functionality and its exploitation (user-interfaces, third-party applications) and to allow (composite) services to access the search engines in an uniform way, leading to a truly distributed SOA environment as a *federative web of (search) services*.”

(from the initial Federated Search document)

CLARIN-FCS key components

- Interface Specification
  - common harmonized interface (= lingua franca) that repositories need to implement
- Aggregator
  - module or service to dispatch queries to repositories and collect results
- Endpoints
  - repository that implements the interface specification
- Repository Registry
  - A separate service that allows registering endpoints and provides information about these to other components, e.g. the Aggegator

CLARIN-FCS key components



schematic overview of the components

(Brief) Introduction to SRU/CQL?

What is SRU/CQL?

- ***S***earch and ***R***etrieve via ***U***RL
- Originates from library world
  - Developed as web service replacement of the Z39.50 protocol
- Allows (meta-)data format agnostic searching and retrieval of hits
- Originally intended to search in library catalogs

What is SRU/CQL?

- Defines an abstract data and processing model
- Supports three operations
  - explain = info about endpoint capabilities
  - scan = enumerate index values
  - searchRetrieve = (actual) search and retrieve operation
- Operations have several mandatory and optional arguments
- extensible though custom *request arguments*, *search contexts* and *record formats*.
- Protocol Bindings
  - SRU (REST/CGI style binding) and SRW (SOAP bindings)

What is SRU/CQL?

- Version 1.1 and 1.2 “standardized” by Library of Congress
- Former Version 2.0 (now searchRetrieve Version 1.0) standardized through OASIS (February 2013)
- Unfortunately, OASIS standardization may changed some XML serialization details in Version 1.2 (= XML namespaces changed)

What is SRU/CQL?

- ***C**ontextual **Q**uery **L**anguage*
- Formal language for representing queries to information retrieval systems
- Design objective
  - human readable and writable
  - Intuitive while maintaining expressiveness of complex languages

- Fun Fact: In SRU/CQL 1.1 CQL stands for “Common Query Language”

What is SRU/CQL?

- context sets (thus the name)
  - permit CQL users to create their own *indexes*, *relations*, *relation modifiers* and *boolean modifiers* without fear of choosing the same name as someone else and thereby having an ambiguous query
  - ≈ “namespaces” for query languages
- General Syntax:

```
searchClause ::= index relation searchTerm
                 | searchTerm
```

What is SRU/CQL?

- SRU/CQL defines several levels of conformance
  - Level 0 (Base Profile): term only query
    - Must be able to process a term-only query
    - Respond with diagnostic message to unsupported queries
    - Note: With term-only query the server (= endpoint) decides which index to use
  - Level 1 (Indices and Boolean Operators)
    - Ability to *parse* both:
      - (a) search clauses consisting of
        `index relation searchTerm`
      - (b) queries where term-only queries are combined with boolean operators, e.g.
        `term1 AND term2`
    - *Support* for at least one of (a) and (b)
  - Level 2 (Parse any CQL)
    - Ability to *parse* all of CQL and respond with appropriate diagnostics
    - Level 2 does not require support for all of CQL, just be able to parse it

NB: Higher levels include features of lower levels

What is SRU/CQL?

Some CQL queries (default context set)

- Level 0

  -
    ```
    system
    "language acquisition"
    "She said \"Yes\""
    ```
- Level 1

  ```
  dc.creator = anderson
  title adj "wonderful feelings"
  bib.dateIssued < 1998
  wonderful OR feelings
  ```
- Level 2

  ```
  title contains Herz and date within "1910 1920"
  ```

What is SRU/CQL?

CQL default context set supports ...

- Relations

  `= >= <= == adj all any within encloses`

- Modifiers

  ```
  /stem /relevant /fuzzy /exact /respectCase
  /isoDate /oid (...)
  ```

- Sorting with **`sortBy`** clause defined by a dedicated context set

What is SRU/CQL?

New features in SRU 2.0 (that may be interesting for CLARIN-FCS)

- Facets
  - provides means to supply faceted results, i.e. the analysis of how the search results are distributed over various categories
- Search result analysis
  - provide information for some or all of the sub queries of a complex query
- resultCountPrecision
  - allows the server to indicate or estimate the accuracy of the result count as reported (controlled vocabulary)
- window
  - Be able to formulate a multi-term query within a defined window
- … and some misc features

(Brief) CLARIN-FCS interface specification

CLARIN-FCS

- CLARIN-FCS extends SRU/CQL ...
  - Contexts Set
    - **`isocat`** (for DCs defined in ISOcat Data Category Registry)
    - **`fcs`** (for content, including annotation tiers)
    - **`cmd`** (for metadata)
    - (This is still rather underspecified)
  - Record format
    - generic and extensible structure for returning results
  - Behavior
    - Enumeration of extended information about resources at an endpoint using scan operation

CLARIN-FCS

- CLARIN-FCS record format
  - One record shall represent one hit in the result set
  - Allows encoding of resource fragments (full text vs. a single sentence)
  - Allows encoding persistent and non-persistent links (@pid and @ref)
    - Endpoints are required to provide proper PIDs (if available)
  - Actual hit is encoded as one or more *DataViews*
    - Keyword-In-Context DataView is mandatory

CLARIN-FCS

- Currently defined DataViews ...
  - Keyword-In-Context (KWIC)
    - a keyword-in-context view, where each hit should be presented within the context of a complete sentence (if possible) or any other reasonable unit of context
  - CMDI metadata
    - CMDI metadata record applicable to the specific context
  - Geolocation (KML)
    - geographic location encoded in the Keyhole Markup Language
- DataView format is deliberately kept open to allow further extensions in the future

CLARIN-FCS

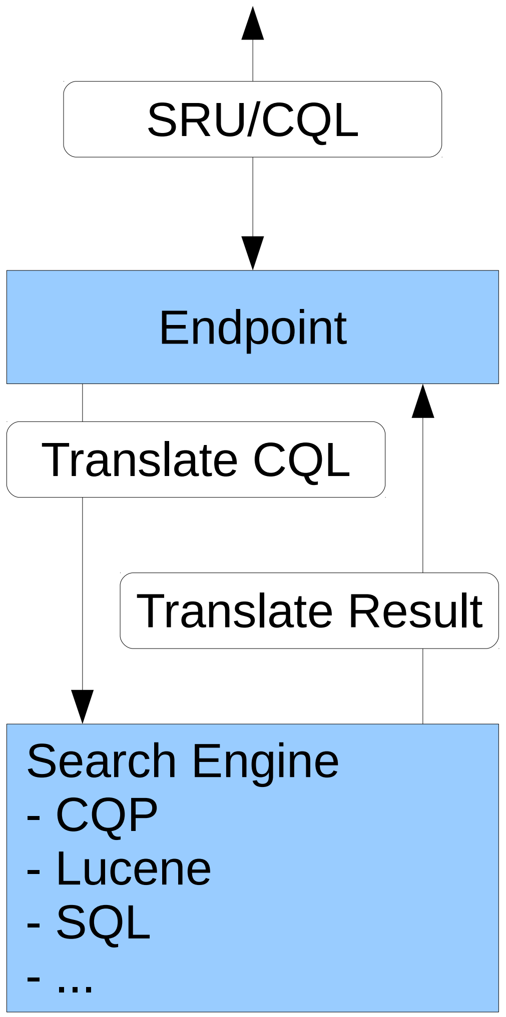
CLARIN-FCS *endpoints*

- Software, that implements the CLARIN-FCS interface specification and acts as bridge between (resource specific) search engine and CLARIN-FCS
- CLARIN centers may implement an arbitrary number of endpoints
- endpoints may act as a “portal” to an arbitrary number of search engines

CLARIN-FCS

A CLARIN-FCS *endpoint* basically needs to ...

- … transform the CQL query to the search engine specific query
- ... serialize the results from the search engine in the CLARIN FCS record format (i.e. DataViews)



CLARIN-FCS specification

- Current CLARIN-FCS specification is maintained at CLARIN EU Trac Wiki
  - https://trac.clarin.eu/wiki/FCS-specification
- Readers are expected to have understanding of SRU/CQL
  - ➔ *Don't expect to get all required SRU/CQL background from FCS spec!*
- If you plan to implement CLARIN-FCS, please read specification and provide feedback, if things are unclear
- NB: public export in CLARIN EU web-page is (slightly) outdated and some information is conflicting!

Available Software components

SRUServer

- Java Web Application
  - Servlet implements REST binding for SRU
- Generic SRU implementation
  - Build from scratch with strict protocol conformance in mind
  - Users need to implement a few interfaces to connect their search engine
  - Package **`eu.clarin.sru.server.*`**
- Supports SRU/CQL Version 1.1 and 1.2
- Uses existing CQL Parser
  - **`org.z3950.zing.cql.CQLParser`**
- Complete JavaDoc for public API

FCSSimpleEndpoint

- Small library that provides convince classed and methods for implementing CLARIN-FCS endpoints
  - Provides support for extended resource enumeration
    - Either through a statically configured list of resources or by implementing a set of interfaces
  - Provides support for serializing CLARIN-FCS record format and KWIC dataview
  - Package **`eu.clarin.sru.server.fcs.*`**
  - Of course, less flexible than plan SRUServer ...
- Complete JavaDoc for public API

SRUClient

- Java library to build SRU consuming applications
- Generic implementation conforming to SRU 1.1 and 1.2
- Several interfaces exported to client applications
  - Either SAX-like (= streaming mode) or POJO interface to record
  - Provides simple and threaded (= asynchronous) mode
- Support for parsing CLARIN-FCS record format into POJOs
  - Will be separated from generic SRUClient into another library in the future
  - Some bits are missing (e.g. recursive records)
- Complete JavaDoc for public API

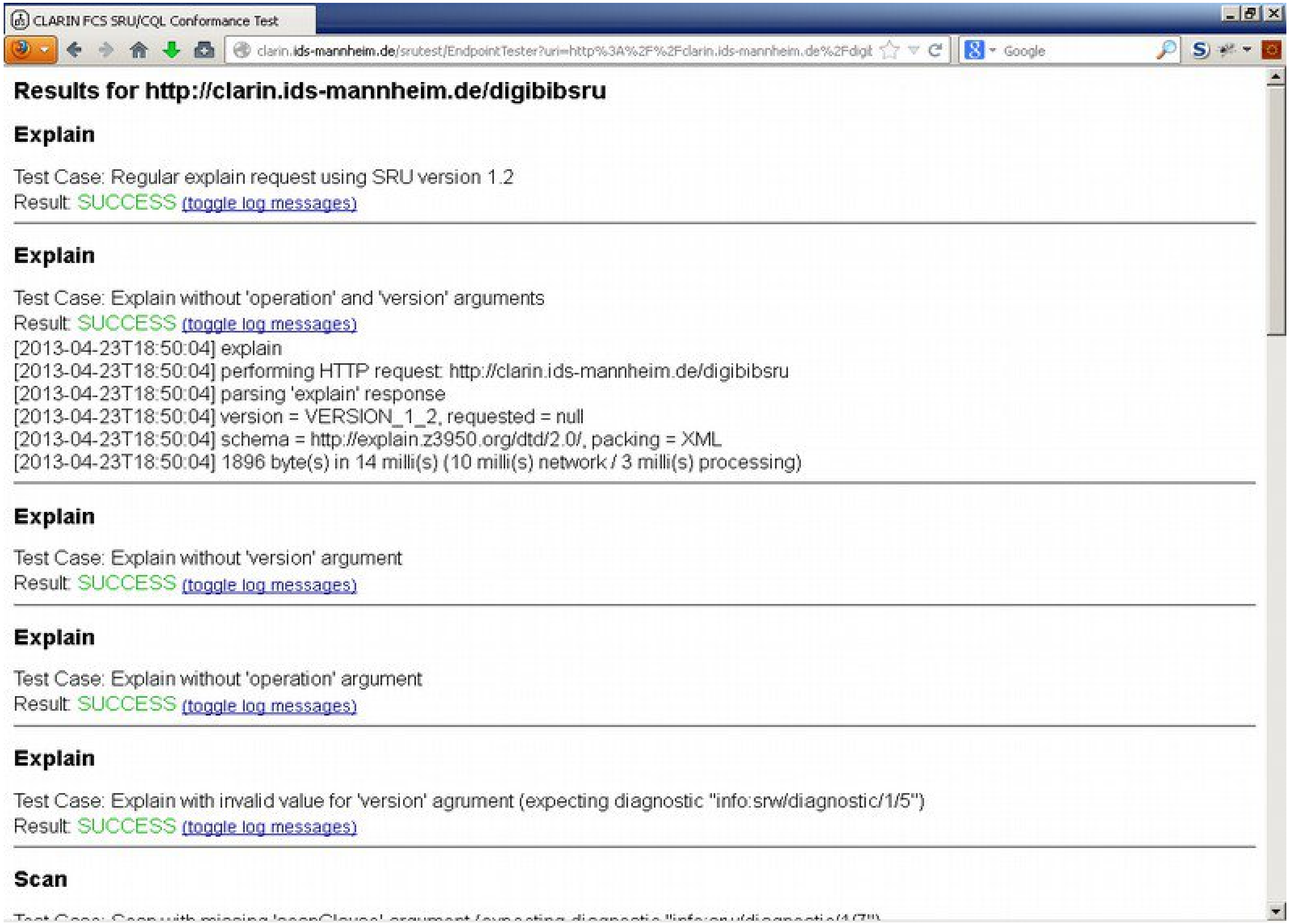
Software availability

- Current Versions
  - SRUServer 1.5.0
  - FCSSimpleClient 1.2.0
  - SRUClient 0.9.0
- All GPL licensed
- Available from ...
  - CLARIN EU Trac (source)
  - CLARIN Maven repository (binaries, source, javadoc)

Conformance Test

- A simple web-service to check if endpoints conform to specification
  - Currently 20 tests
  - Mostly basic SRU/CQL tests
  - A few CLARIN-FCS tests
  - Will be extended for more CLARIN-FCS specific tests
- http://clarin.ids-mannheim.de/srutest/ (requires authentication)

Conformance Test


CLARIN FCS SRU/CQL Conformance Test
clarin.ids-mannheim.de/srutest/EndpointTester?uri=http%3A%2F%2Fclarin.ids-mannheim.de%2Fdigit
Google

Results for http://clarin.ids-mannheim.de/digibibsru

Explain

Test Case: Regular explain request using SRU version 1.2
Result: SUCCESS (toggle log messages)

Explain

Test Case: Explain without 'operation' and 'version' arguments
Result: SUCCESS (toggle log messages)

```
[2013-04-23T18:50:04] explain
[2013-04-23T18:50:04] performing HTTP request: http://clarin.ids-mannheim.de/digibibsru
[2013-04-23T18:50:04] parsing 'explain' response
[2013-04-23T18:50:04] version = VERSION_1_2, requested = null
[2013-04-23T18:50:04] schema = http://explain.z3950.org/dtd/2.0/, packing = XML
[2013-04-23T18:50:04] 1896 byte(s) in 14 milli(s) (10 milli(s) network / 3 milli(s) processing)
```

Explain

Test Case: Explain without 'version' argument
Result: SUCCESS (toggle log messages)

Explain

Test Case: Explain without 'operation' argument
Result: SUCCESS (toggle log messages)

Explain

Test Case: Explain with invalid value for 'version' agrument (expecting diagnostic "info:srw/diagnostic/1/5")
Result: SUCCESS (toggle log messages)

Scan

Test Case: Scan with missing 'scanClause' argument (expecting diagnostic "info:srw/diagnostic/1/7")

CLARIN-FCS further steps

- Revisit specification of CLARIN-FCS context set (indices and relations)
  - How to link content and metadata search?
  - How to map other linguistic annotation tiers?
- Revisit extended resource enumeration
- Define more DataViews?
- Authentication and Authorization?
- Organizational issues (Which committee is in charge of FCS spec)?

What about your use-cases?

- With current state of CLARIN-FCS most use-cases are not feasible, because they ...
  - … require access to annotation tiers
    … require searching on metadata
    - Not yet sufficiently specified
  - … require aggregation of results
    - SRU 2.0 facets could help here

What about your use-cases?

However, CLARIN-FCS is not and cannot be the panacea to solves all query needs

- It's basically a bridge to *specialized search engines* in a highly *heterogeneous environment*
- It's limited by the power search engines at the endpoints
- It's limited by the features of resources at the endpoints (e.g. annotation tiers, tag sets used, …)

Conclusion

- CLARIN-FCS ...
  - … defines an interface and several components to enable a federated search infrastructure
  - … is based on SRU/CQL
  - … defined a flexible return format to encode different views on data
- Some software components are already publicly available for centers to build endpoints
- Still a lot of work to do ...

CLARIN-D

Thank you for your attention. Questions?

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

Links

- SRU/CQL spec
  http://www.loc.gov/standards/sru/specs/
- searchRetrieve 1.0 spec (via LOC)
  http://www.loc.gov/standards/sru/oasis/
- CLARIN Wiki federated search home page
  https://trac.clarin.eu/wiki/FederatedSearch
- CLARIN Wiki federated search specs
  https://trac.clarin.eu/wiki/FCS-specification

Links

- SRU endpoint tester
  http://clarin.ids-mannheim.de/srutest/
- SRUServer
  https://trac.clarin.eu/browser/SRUServer
- FCSSimpleEndpoint
  https://trac.clarin.eu/browser/FCSSimpleEndpoint
- SRUClient
  https://trac.clarin.eu/browser/SRUClient