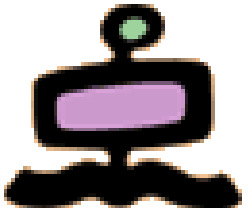
Corso di Biblioteche Digitali



- Vittore Casarosa
  - casarosa@isti.cnr.it
  - Office: 050 621 3115
  - Mobile: 348 397 2168
  - Skype: vittore1201

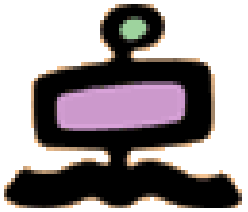
- “Ricevimento” at the end of the lessons or by appointment
- Final assessment
  - 70% oral examination
  - 30% project (development of a small digital library))
- Reference material:
  - Ian Witten, David Bainbridge, David Nichols, How to build a Digital Library, Morgan Kaufmann, 2010, ISBN 978-0-12-374857-7 (Second edition)
  - Material provided by the teacher
- **http://cloudone.isti.cnr.it/casarosa/BDG/**

Servizio di Ascolto

- **Servizio di Ascolto e Consulenza per Studenti Universitari**

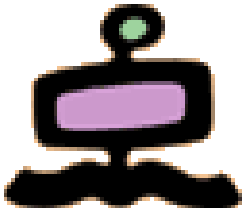
- **https://www.unipi.it/index.php/ascolto**

Modules



- Computer Fundamentals and Networking
- A conceptual model for Digital Libraries
- Bibliographic records and metadata
- Information Retrieval and Search Engines
- Knowledge representation
- Digital Libraries and the Web
- Hands-on laboratory: the Greenstone system

Parallel evolution



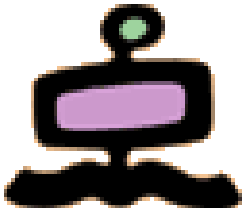
Libraries

- Description (documents)
  - Bibliographic records
  - MARC
- **Interoperability**
  - **Z39-50**
- Conceptual model (classes)
  - FRBR – LRM for Works, Expr., Manif.
- Information Retrieval
  - Full text (catalogue and documents)

The Web

- Description (instances)
  - Metadata
  - Dublin Core
- Interoperability
  - OAI-PMH
- Conceptual model (classes)
  - RDF and RDF Schema for all resources (ontologies)
- Information Retrieval
  - Full text (web pages and resources)

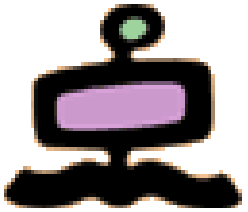
Exchange of information



Interoperability and the exchange of information

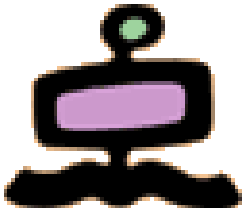
- The Z39.50 protocol
  - protocol defined to support the exchange of bibliographic data (basically MARC records) between libraries
- SRU/SRW queries
  - language defined to support “Z39.50 queries” carried over HTTP
- The OAI-PMH protocol
  - protocol defined to support the exchange of metadata (Dublin Core and others) over HTTP

Z39.50



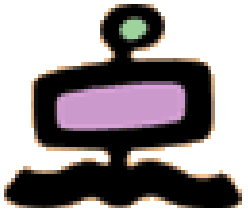
- "Information Retrieval (Z39.50); Application Service Definition and Protocol Specification, ANSI/NISO Z39.50-1995"
- Developed by NISO (National Information Standards Organization), the standards development organization serving libraries, publishing and information services
- NISO was (is) the Z39 Committee of ANSI (American National Standards Institute), and Z39.50 was the 50th standard defined by NISO
- Current version (Version 3) was adopted in 1995, superceding earlier versions adopted in 1992 and 1988 (1984 version was rejected)
  - Another revision, initiated in 2001, is still “work in progress”
- Z39.50 was heavily influenced by OSI, and was an “application layer” protocol that needed a full-duplex reliable OSI connection
  - In Version 3 it runs over TCP/IP
- It is a wide ranging protocol for information retrieval between a client and a database server, which attempts to standardize shared semantic knowledge

Z39.50 architectural model (client/server)



- A server houses one or more databases containing records.
- Associated with each database are a set of access points (indexes) that can be used for searching
- Z39.50 includes a set of “registries” that provide information about the structure and attributes of the data bases (query syntax, attribute fields, content retrieval format, etc.)
- A search (sent from the client/origin to the server/target) produces a set of records, called a "result set", that are maintained on the server
- The client has also functions for search management (e.g. request progress reports for an active search, authorize the server to continue a resource intensive search, abort an active search)

Z39.50 architectural model



- Records from the result set can be retrieved by the client, which has many options for controlling the contents and format of the records that are returned (e.g. sorting a result set, selecting a subset of the result set, using the result set for a new search)
- The client has available also a general mechanism called "extended services" to invoke services on the server, which can survive past the end of the session (e.g. saving result sets across sessions, queuing result sets for print or electronic mail processing at the server, registering queries that would be executed periodically on the server)

Z39.50 : THE BASICS

Fay Turner

National Library of Canada

fay.turner@nlc-bnc.ca

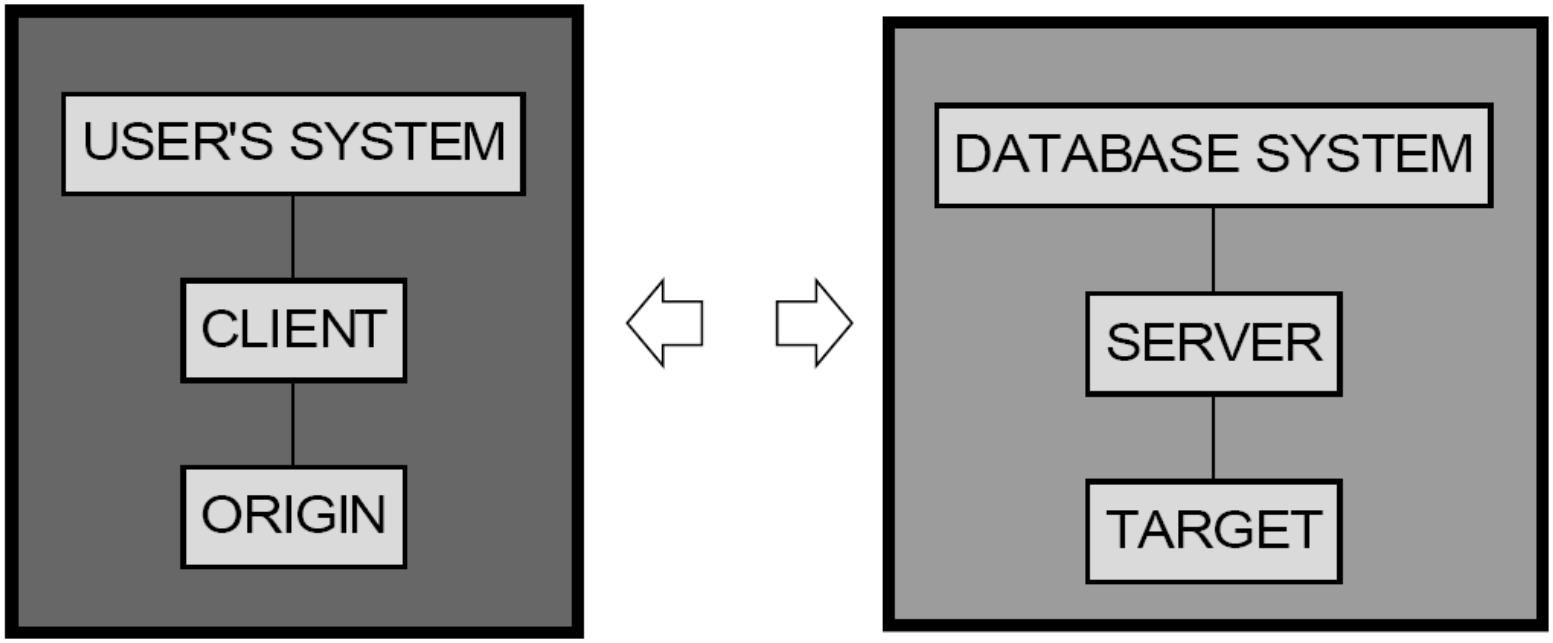
1997 IFLA Conference

Session - Z39.50: Information Retrieval in an Open Networked Environment

Z39.50

- Z39.50 SIMPLIFIES SEARCHING
- ONLY NEED TO KNOW COMMANDS OF LOCAL SYSTEM
- PERMITS COMMUNICATION BETWEEN SYSTEMS USING DIFFERENT HARDWARE AND LIBRARY SOFTWARE

Z39.50 MODEL



DRA, NOTIS, VTLS **GEAC, OCLC, LC**

National Library of Canada/Bibliothèque nationale du Canada

ORIGIN SYSTEM (CLIENT)

- SOFTWARE ON LOCAL SYSTEM TRANSLATES SEARCH QUERY INTO FORMAT OF Z39.50 STANDARD
- CONNECTS TO AND SENDS QUERY TO SYSTEM HOUSING THE DATABASE
- PRESENTS RECORDS/RESULTS OF QUERY TO SEARCHER

SEARCHER OF ORIGIN SYSTEM NEVER INTERACTS DIRECTLY WITH TARGET SYSTEM

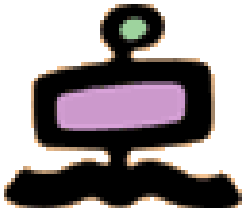
TARGET SYSTEM (SERVER)

- COMPUTER HOUSING THE DATABASE(S)
- TRANSLATES THE Z39.50 QUERY TO SEARCH LOGIC OF DATABASE SYSTEM
- OBTAINS INFO FROM DATABASE, RETURNS IT TO ORIGIN SYSTEM
- RETURNS RECORDS OR REPORTS A RESULT SET

CLIENT AND TARGET ROLES CAN BE CONTAINED IN SAME SYSTEM

National Library of Canada/Bibliothèque nationale du Canada

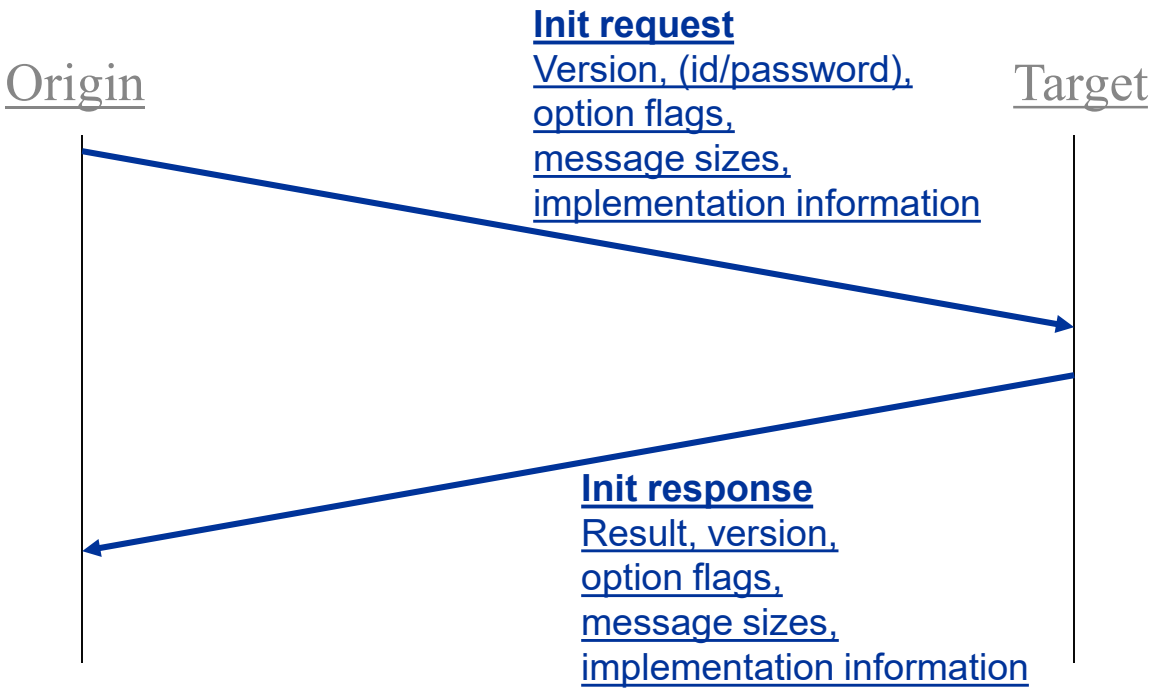
Z39.50 functionality



Z39.50 Facility	Client-side description
Initialization	Establish connection with server and set/request resource limits.
Search	Initiate search using a registered query syntax, generating a result set server-side.
Retrieval	Retrieve a set of records from a specified result set: a large record may be segmented and transmitted piecemeal.
Result-set-delete	Request deletion of server-side result set or sets.
Access Control	Server initiated authentication check.
Accounting & Resource Control	Request status reports of committed server resources and dictate if server is allowed to contact client when agreed limits are reached.
Sort	Specify how a result set should be sorted.
Browse	Access ordered lists such as title and subject metadata.
Explain	Interrogate server to discover supported services, registries, and so on.
Extended Services	Access services that continue beyond the life of this client-server exchange, such as persistent queries and database update.
Termination	Abruptly end client-server session: initiated by either client or server.

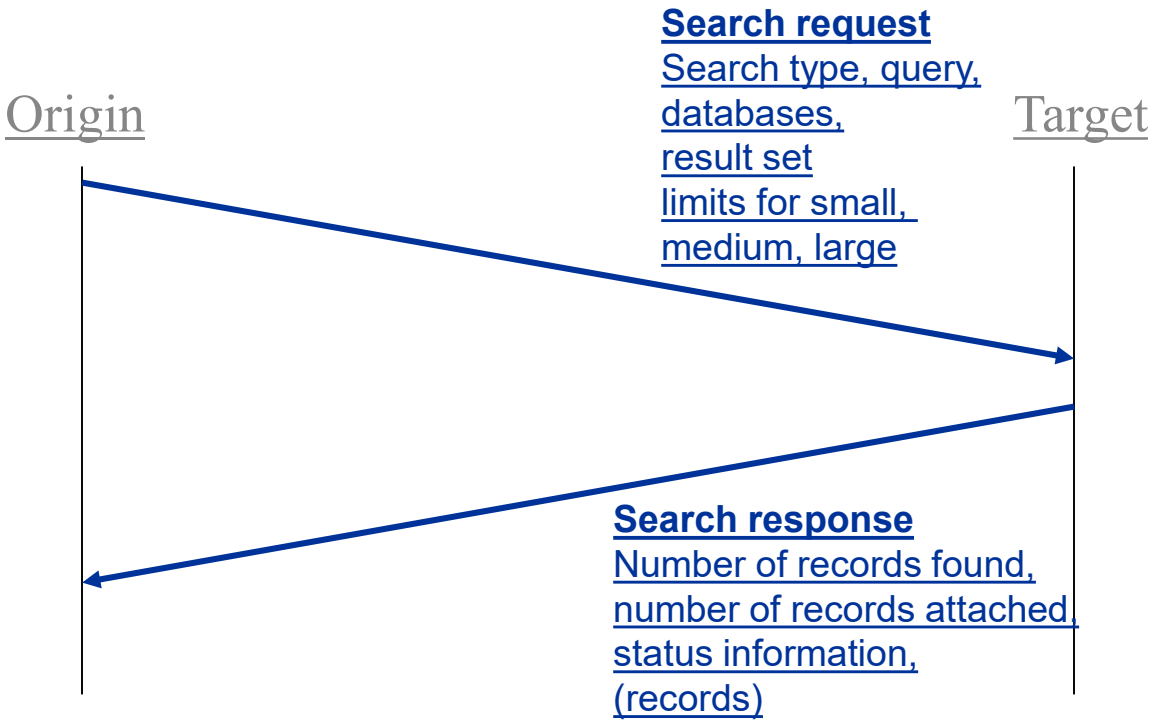
Initialization facility

- Init service: establishes Z-association

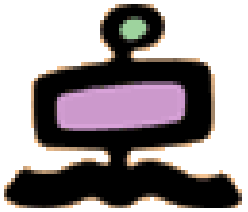


Search facility

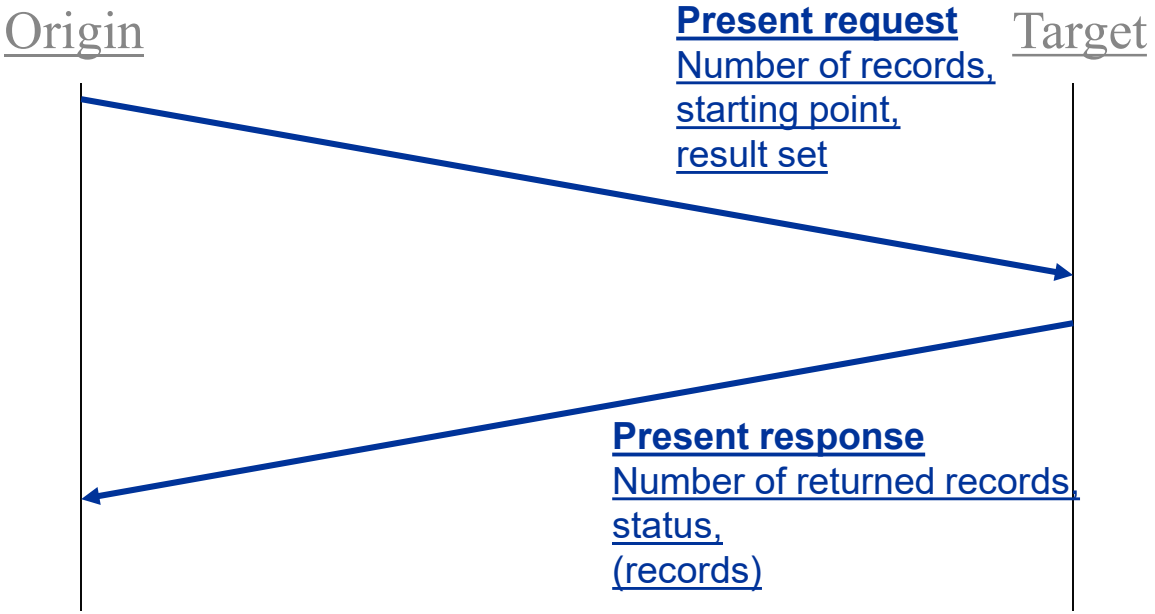
- Search service



Retrieval facility



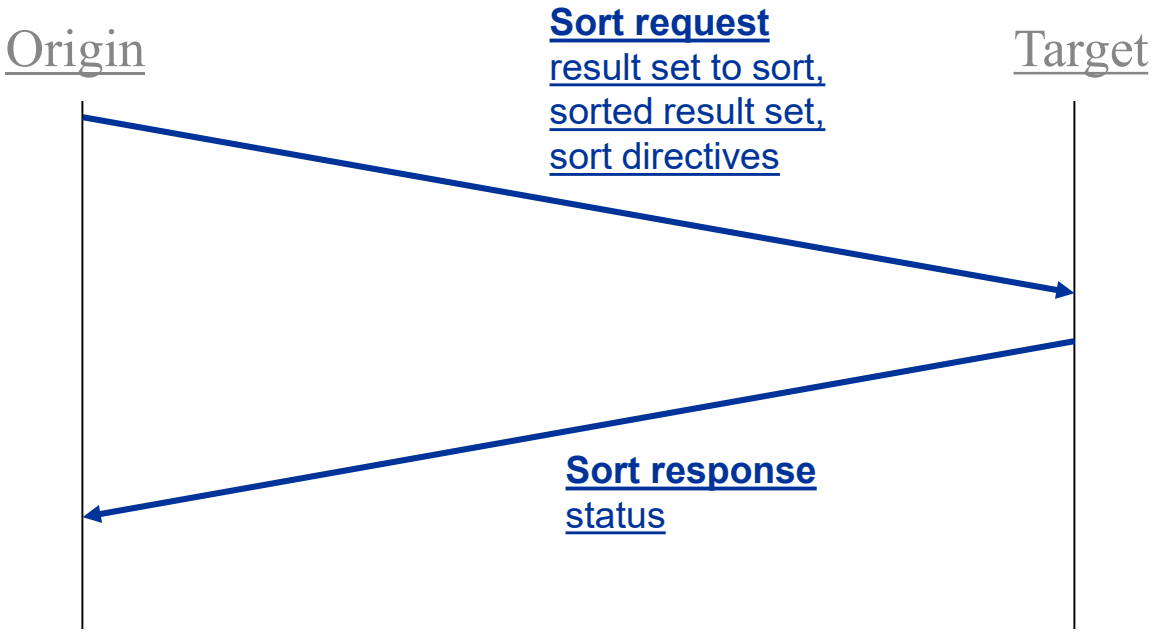
- Present service



- Segment service
  - Allows a “Present response” that is larger than max size to be split in segments

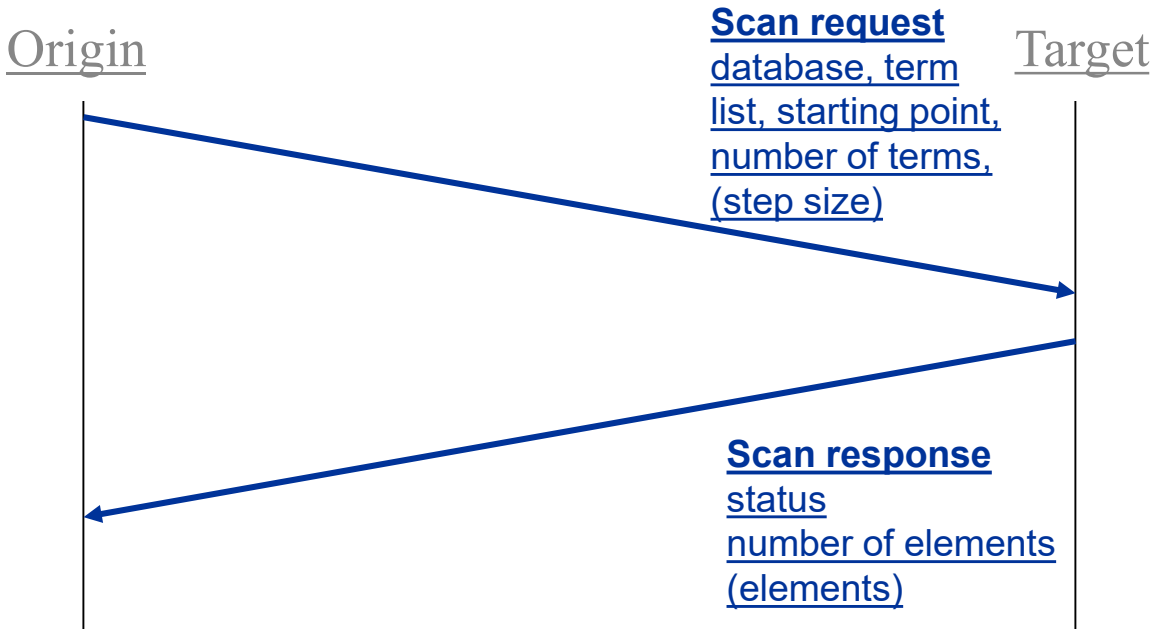
Sort facility

- Sort service



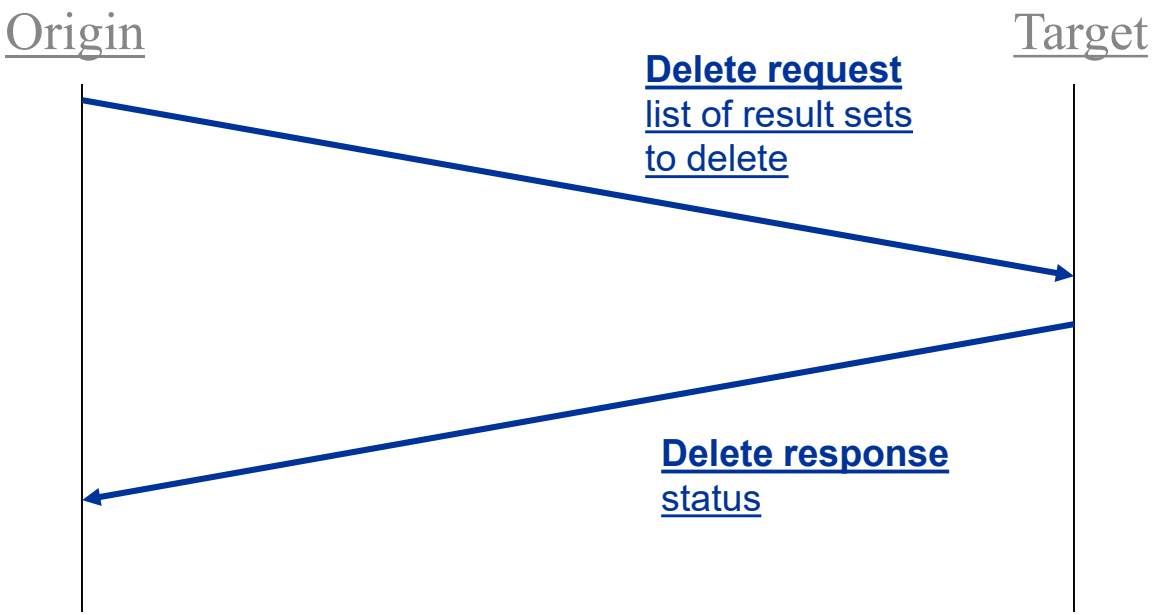
Browse facility

- Scan service

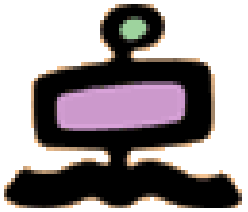


Result-set-delete facility

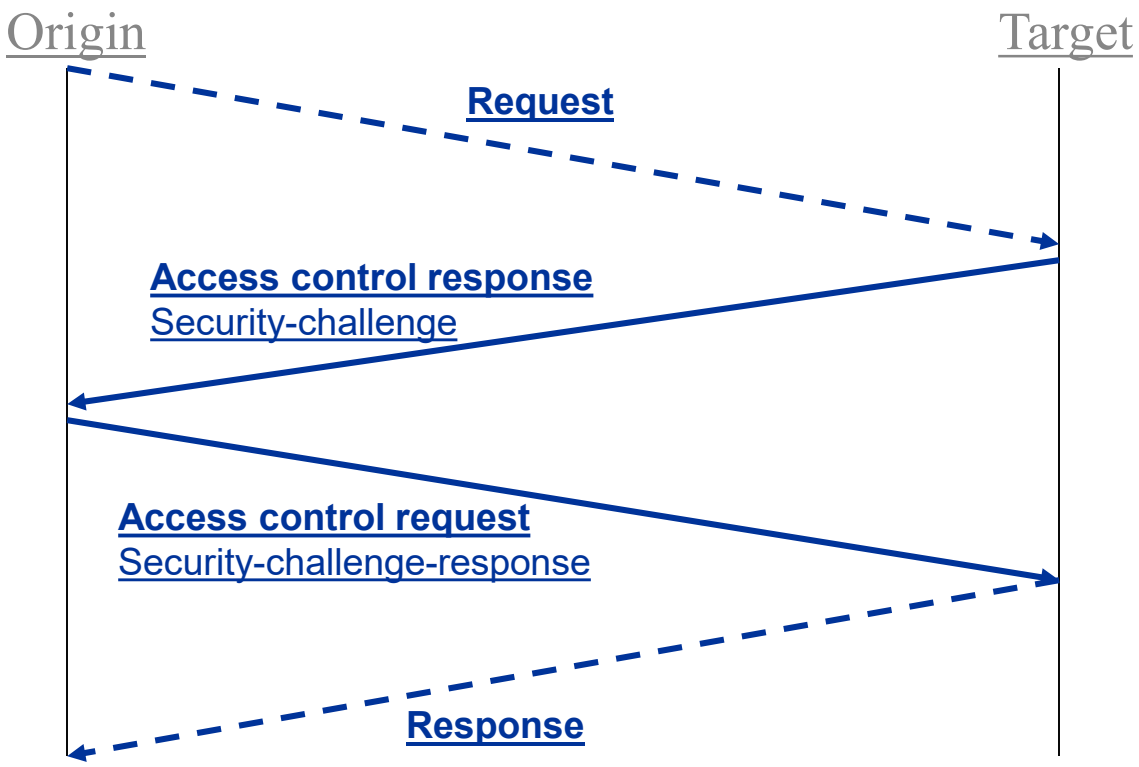
- Delete service



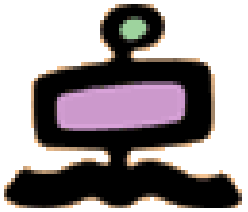
Access control facility



- Access-control service

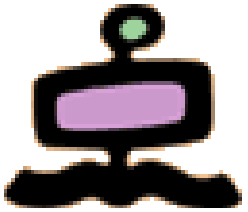


Accounting/Resource control facility



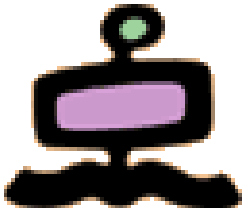
- Resource-control service
- Trigger-resource-control service
- Resource-report service
  - Complex functionality to control and report resource usage
  - Mostly used for fee based operation

Termination facility



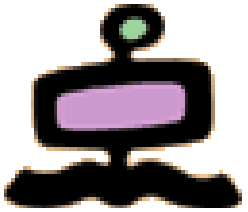
- Close service
  - Terminates a Z-association

Explain facility



- Explain service
  - Gives access to information about the Z39.50 target
    - Databases
    - Access points
    - Query languages
    - Element sets
    - ...
- This information is maintained by the server in a specific data base, and therefore can be accessed using the Search and Retrieve facilities of Z39.50
- The idea was that a (smart) client, when accessing a (unknown) data base, could be able to find its access points, its element sets and other info by querying the ”Explain” data base

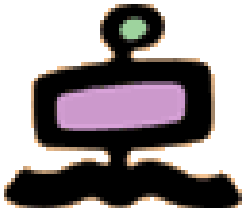
Extended Service facility



- Extended Services service
  - Persistent Result Set Extended Service
  - Persistent Query Extended Service
  - Periodic Query Schedule Extended Service
  - Item Order Extended Service
  - Database Update Extended Service
  - Export Specification Extended Service

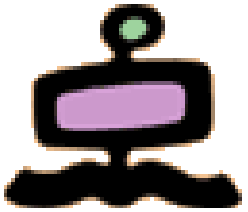
- Task package
  - Used to create, modify or delete an Extended Service Request

Z39.50 queries



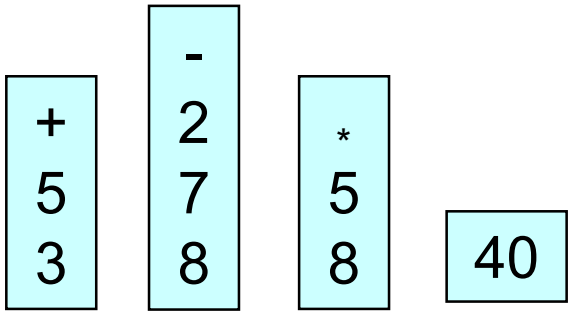
- Query types
  - Type-0: proprietary between 2 parties
  - Type-1: RPN (Reverse Polish Notation)
  - Type-2: ISO 8777
  - Type-100: Z39.58
  - Type-101: Extended RPN (v 2)
  - Type 102: Ranked List query

Type-1 Query
Reverse Polish Notation (RPN)

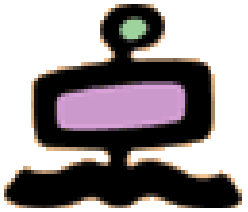


- Consists of
  - One or more operands linked (RPN style) with Boolean operators (AND, OR, AND_NOT)
  - Every operand is a search expression consisting of 7 parts
- Example of query
  - (operand)(operand)operator
  - (“Mark Twain”, 1:1003, 2:3, 3:1, 4:1, 5:100, 6:1) (“Clemence, Samuel”, 1:1003, 2:3, 3:3, 4:101, 5:100, 6:2) AND_NOT

RPN
(3 + 5) * (7 – 2)
3 5 + 7 2 – *

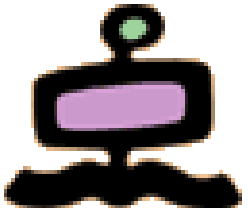


Operands in Type-1 queries



- 0. Term
  - What you are looking for
- 1.Use Attributes
  - Which abstract access point to use (e.g. title, author)
- 2.Relation Attributes
  - Relation between the term and the data in the access point (e.g. less than, equals, phonetic equals)
- 3.Position Attributes
  - Where in the access point should the term be? (e.g. first in field, first in subfield)
- 4.Structure Attributes
  - How is the query term to be treated? (e.g. as phrase, as words, as date, as normalised name)
- 5.Truncation Attributes
  - Should truncation be applied on the match? (e.g. left truncation, right and left truncation, no truncation)
- 6.Completeness Attributes
  - What is the term to be matched against? (e.g. part of subfield, whole subfield, whole field)

Type 2 Query – ISO 8777

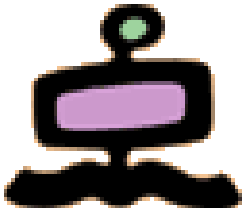


- CCL – Common Command Language

Table 8.2 Common Command Language keywords, with abbreviations.

Keyword	Abbreviation	Keyword	Abbreviation	Keyword	Abbreviation
BACK	BAK	FIND	FIN	SAVE	SAV
CHOOSE	CHO	FORWARD	FOR	SCAN	SCA
DEFINE	DEF	HELP	HEL	SEE	SEE
DELETE	DEL	PRINT	PRI	SET	SET
DISPLAY	DIS	RELATE	REL	SORT	SOR
EXPLAIN	EXP	REVIEW	REV	START	STA
				STOP	STO

From Z39.50 to SRW/U

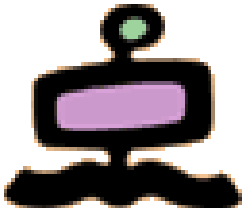


- Need for a generic Information Retrieval capability more suited to the Web Architecture
- Motivation to create an easy to implement protocol with (more or less) the power of Z39.50
- Use existing off the shelf solutions where possible
- Re-evaluate Z39.50, which was “a good idea at the time”
- Avoid library-centric perspective

Solution:

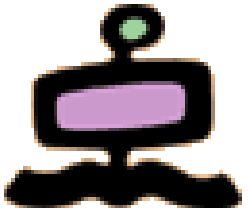
- SRU – Search/Retrieve via URL
- SRW – Search/Retrieve via Web Service

Simple SRU query



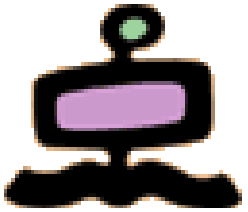
http://sru.miketaylor.org.uk/sru.pl?
version=1.1&
operation=searchRetrieve&
query=dinosaur&
startRecord=1&
maximumRecords=1&
recordSchema=dc

SRU response in XML (1/2)



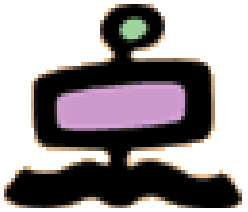
```
<?xml version="1.0"?>
<zs:searchRetrieveResponse
    xmlns:zs='http://www.loc.gov/zing/srw/'>
  <zs:version>1.1</zs:version>
  <zs:numberOfRecords>29</zs:numberOfRecords>
  <zs:records>

   .... details in a moment ....

  </zs:records>
</zs:searchRetrieveResponse>
```

SRU response in XML (2/2)



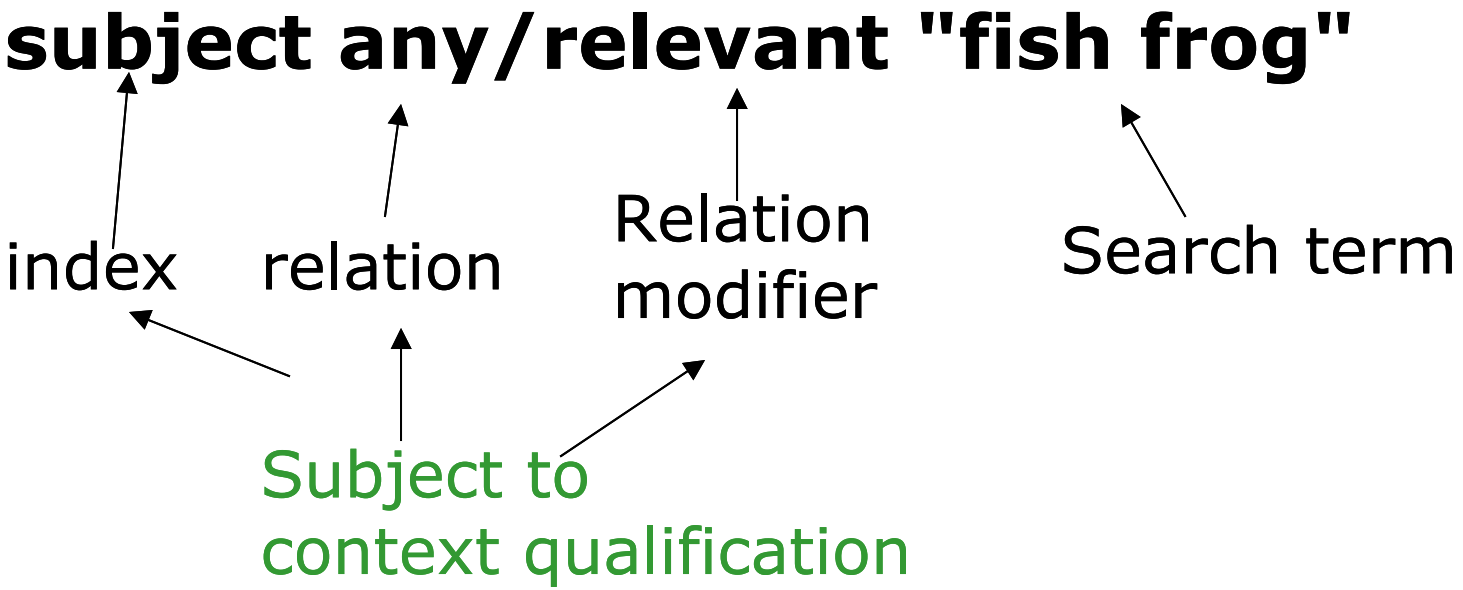
```
<zs:record>
    <zs:recordSchema>info:srw/schema/1/dc-v1.1</zs:recordSchema>
    <zs:recordPacking>xml</zs:recordPacking>
    <zs:recordPosition>1</zs:recordPosition>
    <zs:recordData>
     <srw_dc:dc xmlns:srw_dc="info:srw/schema/1/dc-schema"
            xmlns="http://purl.org/dc/elements/1.1/">
      <title>Fossils</title>
      <creator>Lappi, Megan.</creator>
      <type>text</type>
      <publisher>New York, NY: Weigl Publishers</publisher>
      <date>2005</date>
      <language>en</language>
      <description>Studying fossils -- Fossil facts -- Gone
       forever -- A fossil is born -- From bone to stone --
       Insects in amber -- Dinosaur footprints</description>
      <identifier>
          http://www.loc.gov/catdir/toc/ecip0415/2004004136.html
        </identifier>
      <identifier>URN:ISBN:1590362136</identifier>
     </srw_dc:dc>
    </zs:recordData>
  </zs:record>
```

Contextual Query Language

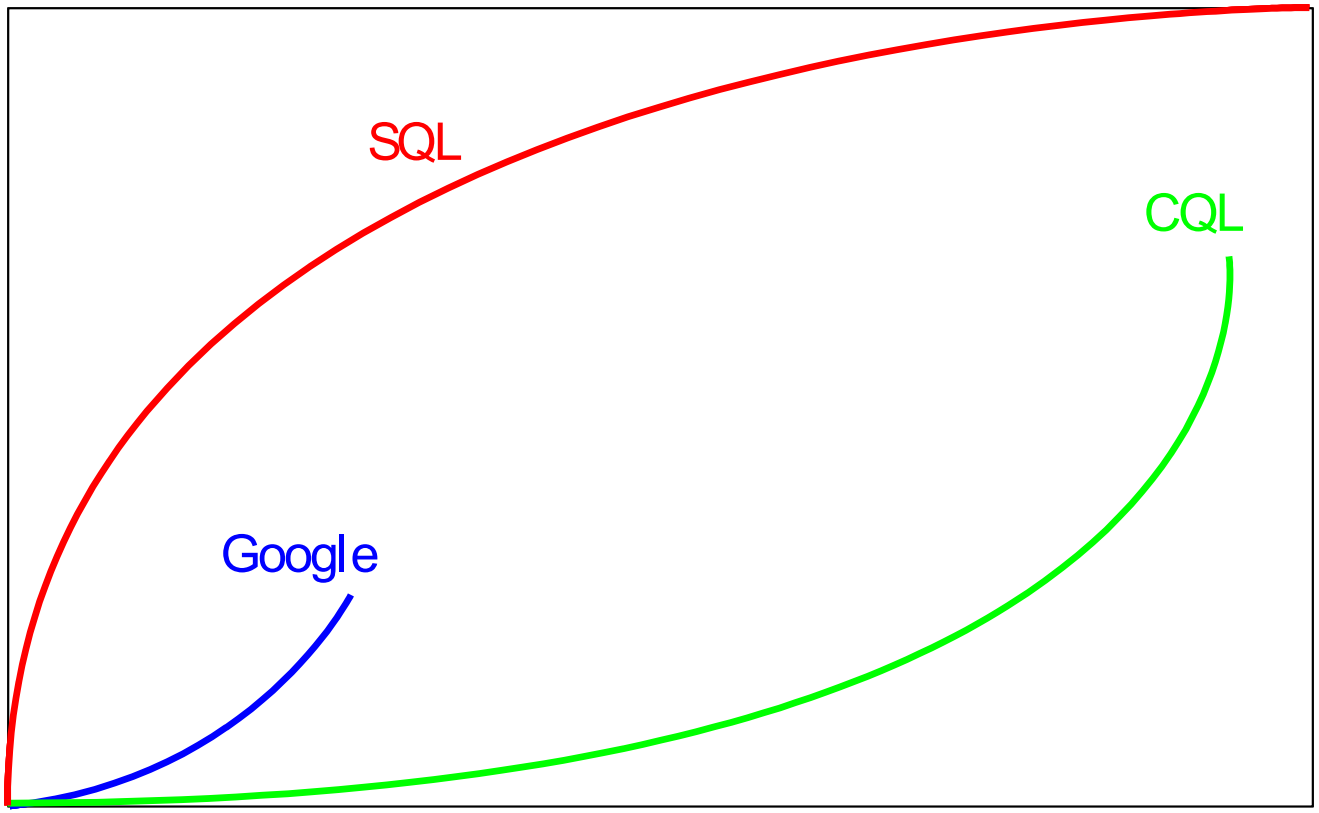


- CQL (formerly known as Common Query Language) is the query language used in SRU
- The conceptual model of CQL is the same as Type 1 in Z39.50
  - The server has one or more databases, containing records
  - The databases can be searched through access points, or indexes
- The language defines a number of defaults to make simple queries really simple
- At the same time it defines a number of Indexes, Relations, Relation Modifiers, Booleans and Boolean Modifiers to increase the expressing power of the language

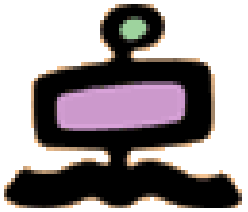
CQL search clause



Learning curves for query languages

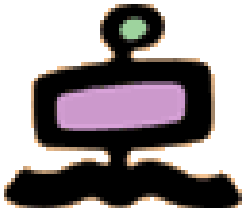


CQL queries



- CQL query
  It consists of either a single search clause or multiple search clauses connected by boolean operators
- Search Clause
  It consists of either an index, relation and a search term or a search term
- Search Term
  Search terms MUST be enclosed in double quotes if they contain any of the following characters: < > = / ( ) and whitespace
- Indexes
  Indexes indicate what part of the records is to be searched (aka field, access point, metadata element, etc)
- Relations
  - In addition to the usual ones (=, <, <=, etc,) include also “any”, “all”, “exact”
  - A relation can be modified by a “relation modifier”, which can be one of the following: **`stem, relevant, fuzzy, phonetic`**
- Boolean Operators
  - Boolean operators are: and, or, not and prox; they all have the same precedence and are evaluated left-to-right
  - The “proximity” operator can be modified according to four parameters: relation (referred to distance), distance (number of units), unit (word, sentence, paragraph, elements), ordering (ordered or unordered)
- Indexes, relations and modifiers can be prefixed with the name of a context set (e.g. dc.title)

CQL examples



- dinosaur
- title=dinosaur
- title=(dinosaur or pterosaur) and author=martill
- dc.title=*saur and dc.author=martill
- heraldry.title=viscount
- title exact "the complete dinosaur" and date < 2000
- telephoneNumber exact/fuzzy "0208 346 6797"
- name=/phonetic "smith"
- fish prox/distance<3/unit=sentence frog

```
dc.author=(kern* or ritchie) and
    (bath.title exact "the c programming language" or
     dc.title=elements prox///4 dc.title=programming) and
    subject any/relevant "style design analysis"
```

Existing standards for catalogues

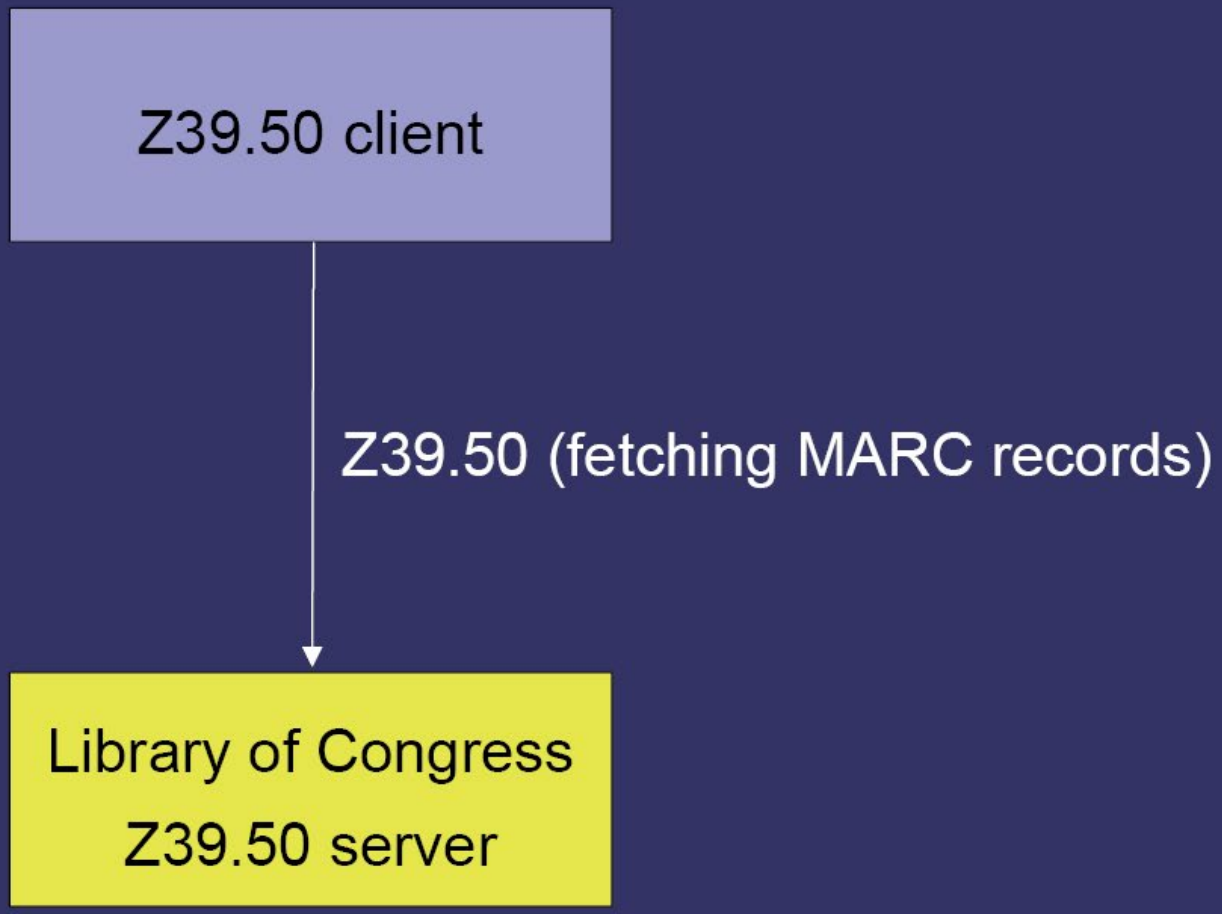
The value of existing standards is well understood:

- MARC (MAchine Readable Catalogue) records
- ISO 2709 (interchange format for MARC)
- ANSI/NISO Z39.50 (search and retrieve on the Internet)

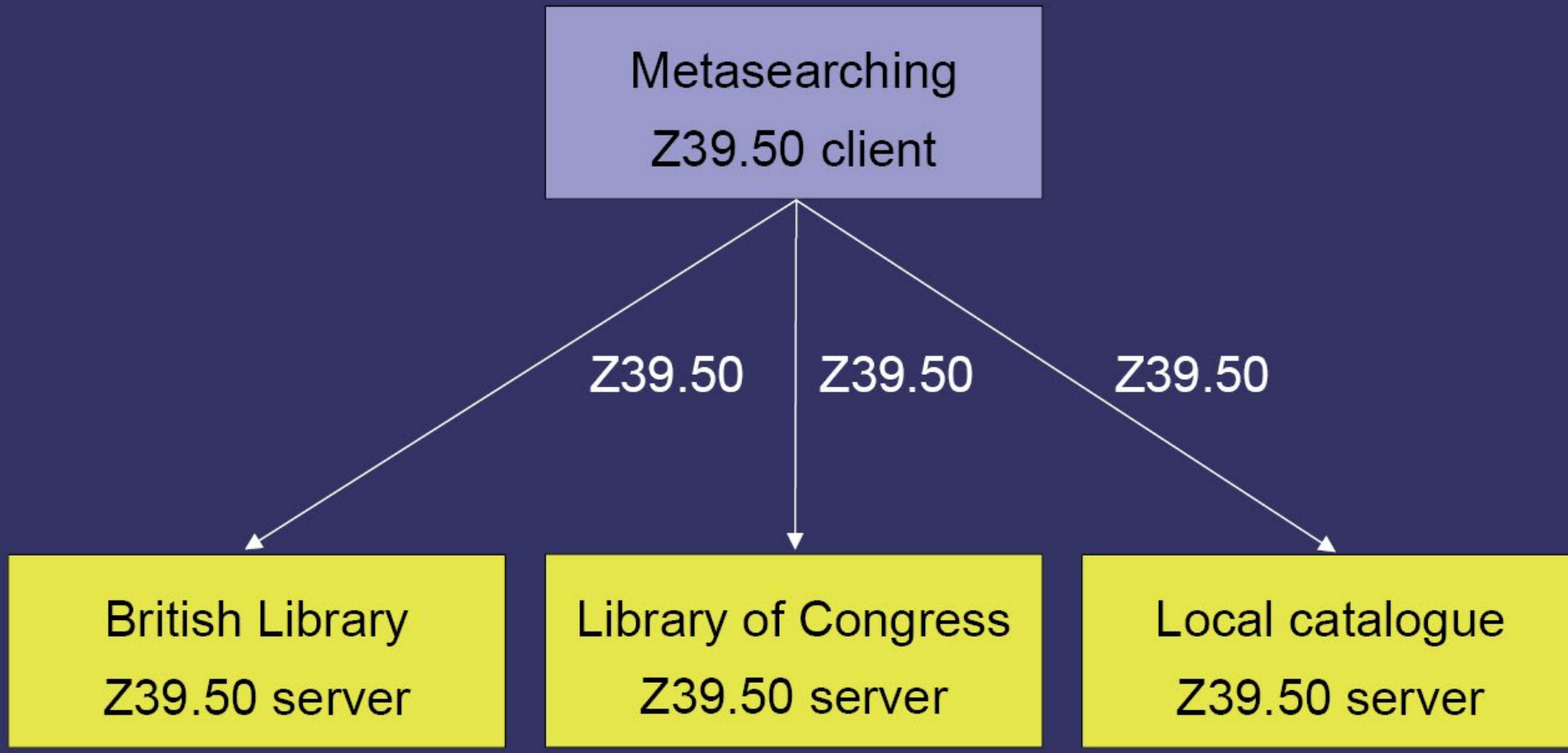
These standards allow interoperability and co-operation between libraries that other fields can only dream about.

(Librarians don't know how lucky they are!)

Z39.50 for searching catalogues



Z39.50 for searching multiple catalogues



Trouble in paradise

Then the serpent saith unto Adam, “Lo, why doth thy catalogue service not use XML?” And Adam saith, “Verily, Z39.50 worketh just fine.” But the serpent, who was subtle of tongue, saith unto him, “But XML is more *fashionable*.” And, behold, Adam was deceived, and did fall.

-- The Book of Standards, ch. 3, v. 4-6.

Welcome to the 21st Century