Implementing the SPAR architecture
Agenda

- Objective and key requirements
- The implementation
  - data management module
  - Storage abstraction service
- The organizational impact on the library
- Conclusion
Key requirements

- OAIS compliance (ISO 14721:2003)
- modularity and distributivity
- abstraction
- use of well known formats and standards
- use of Open Source technical building blocks
Objective and key requirements
The implementation
  → data management module
  → Storage abstraction service
The organizational impact on the library
Conclusion
The “Data management” module

In order to query the metadata, those have to be extracted from the manifest.

The METS files contain also reference metadata. In case of problem, metadata can be retrieved from the AIP.

Ingest
Preservation
Administration
Rights management
Access
Storage Abstraction Service (SAS)

SPAR - Infrastructure

Data management

Storage

AIP

The METS files contain also reference metadata.
Implementation choice

- BnF chooses RDF technology
  - high expressiveness
  - standard language query

- Questioning about scalability
  - benchmark
  - technical choice of Virtuoso
What are the advantages of RDF for SPAR?

- Data model: mapping from METS to RDF
- Modularity and flexibility:
  - Information is represented as simple sentences
    
    the « x » package - hasEvent - « y » event
    « y » event - isA - « ingest »
    « y » event - hasDate - « 22 avril 2009 »
  - Each « sentence » is queryable in the RDF store
  - Request can be build as needed
- Example of query (« SPARQL » syntax)

```sparql
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX sioc: <http://rdfs.org/sioc/ns#>
PREFIX sb: <http://some/ontology#>
PREFIX b: <http://example.org/>
PREFIX dcterms: <http://purl.org/dc/terms/>

?event a <info:bnf/spar/provenance#ingest>.  
FILTER ( ?date > "2009-04-01T00:00:00Z"^^xsd:dateTime )  
FILTER ( ?date < "2009-07-01T00:00:00Z"^^xsd:dateTime )  
}
```
In RDF, any « item » (resource) needs an identifier (URI)

BnF chooses ARK identifiers: this gives a permanent identifier but limited to packages (concrete objects)

For abstract « items », we still need identifiers: concepts, metadata elements, properties, object classes

BnF chooses ‘info’ URI
Ontology for provenance information

Provenance Information: info:bnf/spar/provenance#
Channel description (sample)
Same Channel description in RDF

Channel level requirements

applicableFor

ingestSLA

definesRequirement

Channel

channelRequirement

allow

name

schedule

processingOrder

deny,allow

workingSchedule

startDate

endDate

startTime

endTime

monday

sunday

20:00:00

08:00:00
Objective and key requirements

The implementation
  → data management module
  → Storage abstraction service

The organizational impact on the library

Conclusion
Requirements for the Storage abstraction service

- Abstract the infrastructure with:
  - storage unit
    - aggregates storage element to defines an abstract storage defined by a class of service
      - mean time (read/write)
      - number of copies …
  - record
    - simple bitstream (no semantics)
    - some properties: checksum, logical name, lastAuditDate
Implementation choice

- iRods
  - scalable and proven
  - rules based

- Use of the rules to configure a storage unit
  - rule for put
  - rule for audit
  - rule for get

- General rules for
  - refreshment migration
  - duplication migration
Dialog between Storage module and SAS

→ Storing an AIP
  - corresponding SLA search (storage requirements)
  - retrieval of available storage units with compatible class of service
  - choose of the least expensive storage unit
  - store the AIP as a record

→ Audit of an AIP
  - if audit time has expired,
  - asked the SAS for an audit of the copies
  - retrieved the package itself for internal audit

→ Retrieve of an AIP
  - get back the first ok copy
  - if one is found bad, launch an audit
Agenda

- Objective and key requirements
- The implementation
  - data management module
  - Storage abstraction service
- The organizational impact on the library
- Conclusion
Impact on professions

- Extension of the existing professions
  - track manager
  - digital collection manager
  - digital storekeeper
  - preservation expert

- Enforcement of the relationship with technology
  - the storage hardware keeps the DATA
  - migration operations under the control of system administrator
Define a channel and managing it

- The « track manager » defines the commitment of the institution
- The « collection manager » builds the SLAs (precise definition of the channel) and takes care of its fulfilment
- Depending on
  - Technical possibilities of the system, known by the SPAR administrator,
  - formats allowed by the Archive, known or defined by the preservation expert.
Objective and key requirements

The implementation
  - data management module
  - Storage abstraction service

The organizational impact on the library

Conclusion
Modularity of SPAR

- Pre Ingest
- Ingest
- Preservation
- Administration
- Rights management
- Data management
- Access
- Storage
- Storage Abstraction Service (SAS)
Goal for the archived objects
- definition of an open model
- completeness of the description
- self-supporting package

Ways of dealing with the permanency
- modularity
- abstraction
- use of well known formats and standards
- use of Open Source technical building blocks
Thank you for your attention

Questions?

Thomas Ledoux
thomas.ledoux_AT_bnf.fr