# Implementing a Storage Abstraction Service with iRODS

## iRODS User Group Meeting 2018

Jordan de la Houssaye

June 7, 2018

BnF | Bibliothèque nationale de France

# Table of contents

# Introduction

# The national library of France (BnF)

## Some facts

- a public institution
- ~2200 agents and dozens of professions
- ~1M readers/year

## Some figures (December 31th 2016)

- ~15 000 000 books
- ~15 000 000 posters and photographs
- ~1 930 000 audiovisual material
- ...

## Some dates for legal deposit

- 1537: printed material
- 1648: engravings and maps
- 1793: musical scores
- 1925: photographs
- 1938: phonograms
- 1941: posters
- 1975: videograms and multimedia documents
- 1992: audiovisual and electronic documents
- 2006: web

# The BnF – institutional stakes

## preservation is at the heart of BnF's missions

> decree #94-3, January 3, 1994: *The National Library of France has for mission to collect, preserve, enrich and make available in every field of knowledge the national heritage of which it has the guardianship (...)*

## digital preservation is the direct continuity of BnF's collections preservation

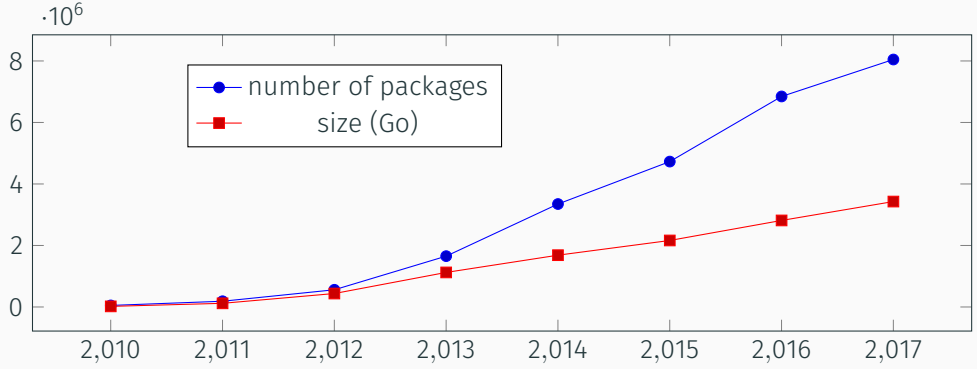
- digitization as a mean to preserve,
- born digital documents

# The BnF – technical stakes

## loss of data is an evermore worrying risk

1. from valorization digitization to preservation digitization
2. legal deposit of substitution
3. born digital documents

## a mass to manage



# SPAR – a digital preservation system

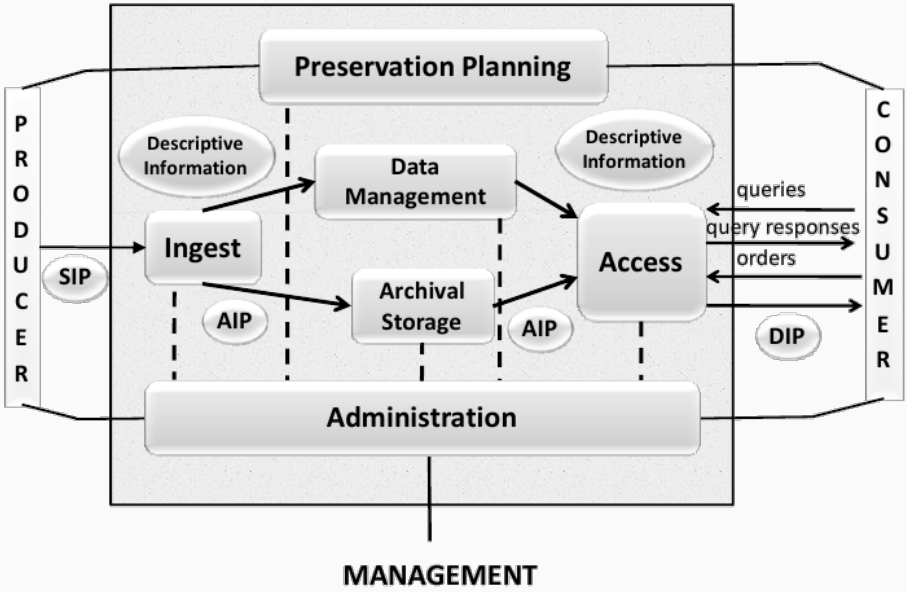
## OAIS (Open Archival Information System)

> *An OAIS is [...] an organization of people and systems that has accepted the responsibility to preserve information and make it available for a* ***Designated Community****.*

## SPAR (Scalable Preservation and Archiving Repository)

- an implementation of OAIS,
- the tool of digital preservation at the BnF
- in operation since may 2010
- replicated on two sites (operations and storage)

# SPAR and OAIS



# SPAR and OAIS – notions

### Information packages

It is a normalized way to present data, ensuring it has a contour and is addressable and findable.

### First job of an OAIS

- normalize data that enters,
- verify it conforms to quality standards,
- augment it with different kind of metadata,
- index it and securely store it,
- ...

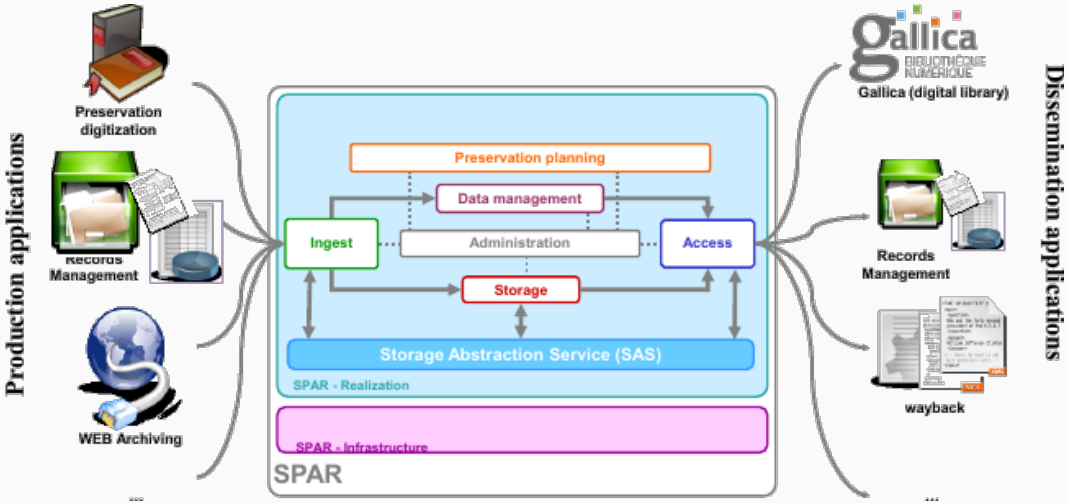
# Approach

# A Storage Abstraction Service – divide and conquer

We divided the storage problematic in two parts

1. a [Storage] module that understands business and is able to apply preservations policies,
2. a [Storage Abstraction Service] modules that know nothing about business but reliably exposes offers of services on storage.

# SPAR and OAIS



# A Storage Abstraction Service – stakes

Abstract the technical complexity for the [Storage] module

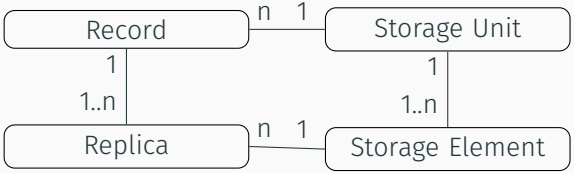
- notion of storage unit, records, ...
- application of a policy based on an offer of services

Abstract the business complexity for the storage administrators

- migrate records with no impact on information packages,

# SAS – model and notions

## Objects




Abstract notions

Concrete notions

Data notions

Containers notions

## Principles

- the SAS exposes storage units where we put records
- it manages automatically storage, replications, retrievals, ...

# iRODS – model and notions

## Virtual file system

- data-objects
- collections
- replicas

Not concerned with physical location of data-objects.

## Resources/Storage devices

Concerned with physical location of data-objects.

## Zones, servers

- iCat (iRODS metadata catalog)
- IES (iCat Enabled Server)
- Resource servers

Concerned with the system's deployment.

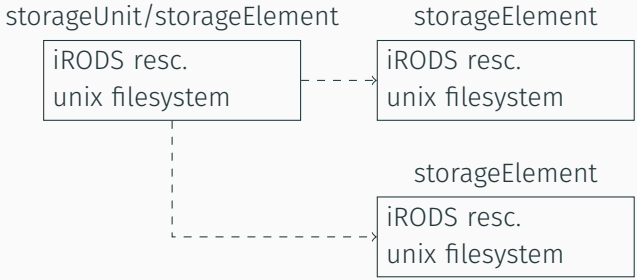
# Implementation

# Recent past — SAS with iRODS 3 i

## CRAUD rules

Create a record, Read it, Audit it (verify and repair its integrity), Update it, Delete it.

## Homemade hierarchical resources

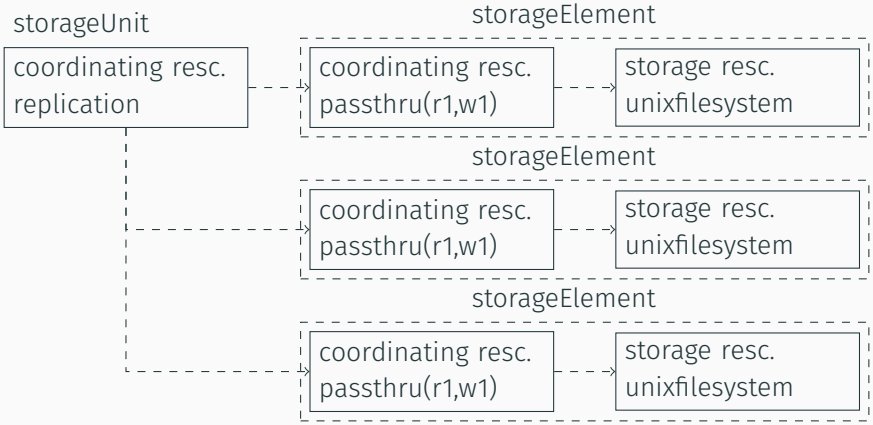


# Recent past — SAS with iRODS 3 ii

View of the resources

```
> ilsresc capsCONSA01
capsCONSA01
> ilsresc elemCONSA01-2
elemCONSA01-2
> ilsresc elemCONSA01-3
elemCONSA01-3
```

## iRODS 4 hierarchical resources



View of the resources

```
> ilsresc capsCONSA01
capsCONSA01:replication
├--- vanneCONSA01-1:passthru
|     └--- elemCONSA01-1:unix file system
├--- vanneCONSA01-2:passthru
|     └--- elemCONSA01-2:unix file system
└--- vanneCONSA01-3:passthru
      └--- elemCONSA01-3:unix file system
```

# Migration from iRODS 3 to iRODS 4

## Context

- r_data_main: approx. 16 million entries
- r_meta_main: approx. 24 million entries
- backend database is postgresql
- development started with iRODS 4.1.7
- migration of the production system with iRODS 4.1.10 (then upgrade to 4.1.11)

## Steps

1. upgrade iCat schema from v3 to v4
2. rename some of our meta_attr_name
3. migrate SAS implementation to v4

# Migration i — upgrade iCat schema to v4

### Intent

Because of huge ”row update” we need to drop index and perform full vacuum and recreate index.

### Actions

1. drop all index
2. upgrade-3.3.xto4.0.0.sql
3. perform vacuum
4. recreate index

# Migration ii — rename some of our meta_attr_name

## Intent

Because of huge ”row update” we need to drop index and perform full vacuum and recreate index.

## Actions

1. drop index
2. update metadata
3. perform vacuum
4. recreate index

# Migration iii — old storageUnits to storageElements

Retrieve all storage element from attribute 'replicaResources'

```
> iquest %s "SELECT META_RESC_ATTR_VALUE WHERE META_RESC_ATTR_NAME
= 'replicaResources' AND RESC_NAME = '${UNIT}'"
```

Get name of storageElement from a storageUnit (v3)

```
> ilsresc -l ${UNIT} | grep "^vault"
```

Homebrew rename resource with clause where with sql in iCAT

```
> resc_id="select resc_id from irods.r_resc_main where
resc_name='${old_name}' and zone_name='SAS' limit 1"
```

```
> update irods.r_resc_main set resc_name='${new_name}' where
resc_id=${resc_id}
```

```
> update irods.r_data_main set resc_name='${new_name}',
resc_hier='${new_name}' where resc_name='${old_name}'
```

# Migration iv — new storageUnits (replication)

Remove useless AVU from storageElement

```
> imeta rm -R ....
```

Create replication resource storageUnit

```
> iquest %s "SELECT RESC_LOC WHERE RESC_NAME = '$ELEMENT_1'"
```

```
> iadmin mkresc $UNIT replication $UNIT_HOST:'FAKE_CAPS_PATH'
```

Transfer AVUs from storageElement to storageUnit

```
> imeta cp -R "${ELEMENT_1}" "${UNIT}"
```

Remove storageElement AVUs from storageUnit

```
> imeta rm -R ....
```

Remove storageUnit AVUs from ELEMENT_1

```
> imeta rm -R ....
```

# Migration v — new storageUnits (replication)

Attach floodgate (passthru) + storageElement

```
> iadmin mkresc $GATE_NAME passthru $UNIT_HOST:'FAKE_GATE_PATH'
'read=1.1;write=1.1'
```

```
> iadmin addchildtoresc $GATE_NAME $ELEMENT_NAME
```

```
> iadmin addchildtoresc $UNIT_NAME $GATE_NAME
```

Proceed with others storageElements

# Conclusion

# Summary

Our Storage Abstraction Service allows SPAR to enforce its daily operations without stopping.

iRODS is its central element.

Migration from iRODS 3 to iRODS 4 was not an easy task.

We are now ready to investigate an upgrade to iRODS 4.2, in particular study what it has to offer in terms of rebalance (we need fine grain capacities).

Questions?