Long-term preservation at BnF

Scalable Preservation and Archiving Repository (SPAR)

June 19th, 2014

{BnF

Agenda

- BnF and digital preservation
- SPAR
- The Storage Abstraction Service
- Limitation of the Abstraction
- Conclusion

The National Library of France

An autonomous public establishment

2200 agents and dozens of professions, a budget of aprox. 230 M€

Local, national and international missions

About 1M readers per year, 300 000 visitors to the exhibitions

Large collections to manage

- More than 14M books
- More than 30M posters and photos
- More than 250.000 manuscripts
- More than 1M audiovisual material
- “French” Web legal deposit

The National Library of France (BnF)

- Main missions:
  - to build up the collections
  - to preserve them forever
  - to communicate them to the public
- Legal deposit:
  - legal deposit since 1537 for printed materials
  - 1648: engravings and maps
  - 1793: musical scores
  - 1925: photos
  - 1938: phonograms
  - 1941: posters
  - 1975: videograms and multimedia documents
  - 1992: audiovisual and electronic documents
  - 2006: Web legal deposit







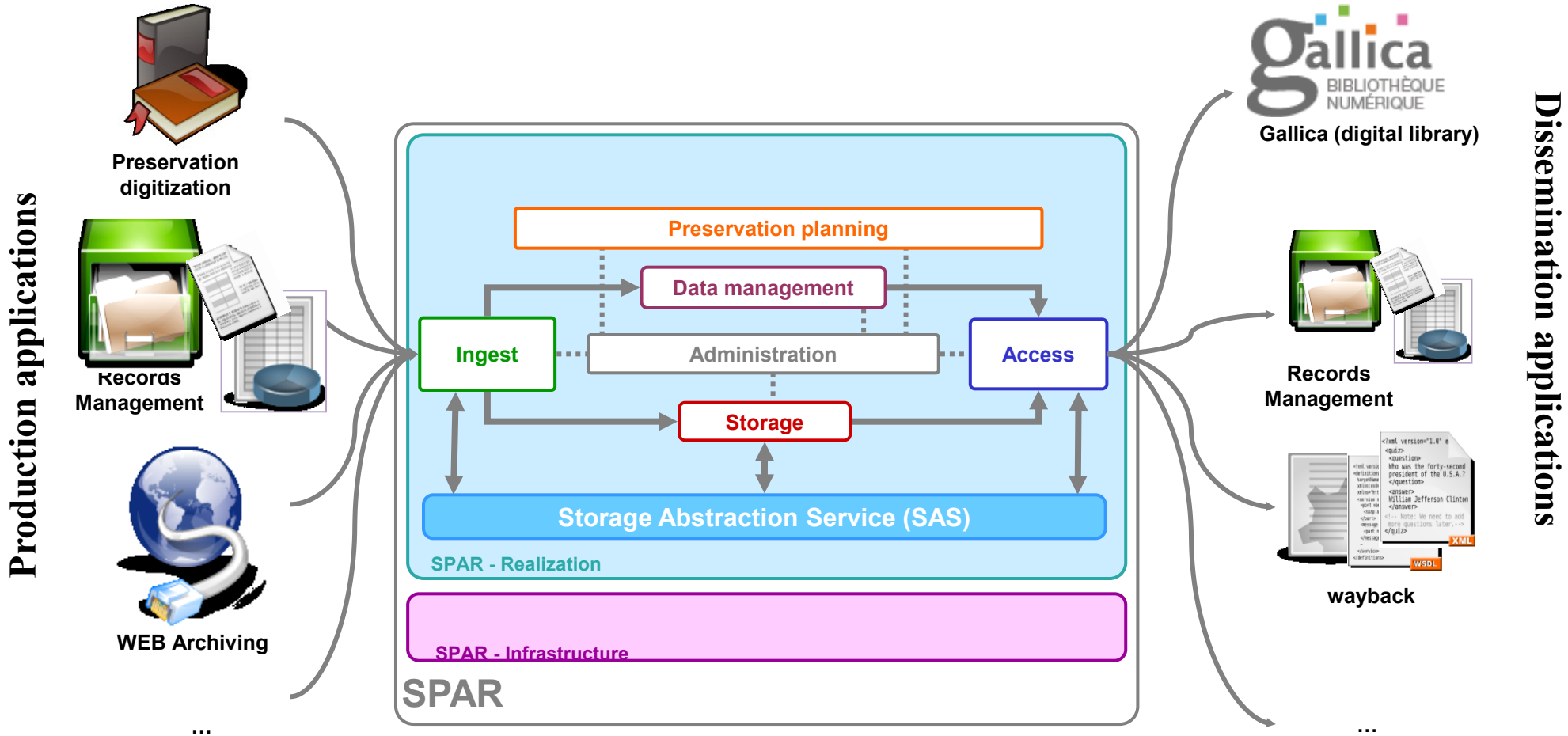
Institutional issue: missions

- Preservation is at the heart of BnF's missions
- Digital preservation is a direct extension of the preservation of BnF's collections. Two different approaches can be considered:

Decree n°94-3 of january 3rd of 19994 – record 2 : La Bibliothèque nationale de France a pour missions de collecter, cataloguer, **conserver** et enrichir dans tous les champs de la connaissance, **le patrimoine national dont elle a la garde**, en particulier le patrimoine de langue française ou relatif à la civilisation française

1. The digital form as a mean to preserve an analog document that degrades. It then enters in a digitization process (as it's the case for paper but also for analog audiovisual collections) and the digital surrogate (which may be a substitution) needs to be preserved along side with the analog
2. The born-digital document (in the case of the web legal deposit since august 1st, 2006, or of the audiovisual legal deposit). This document then belonging to the heritage collections has to be preserved.

Digital archiving at BnF



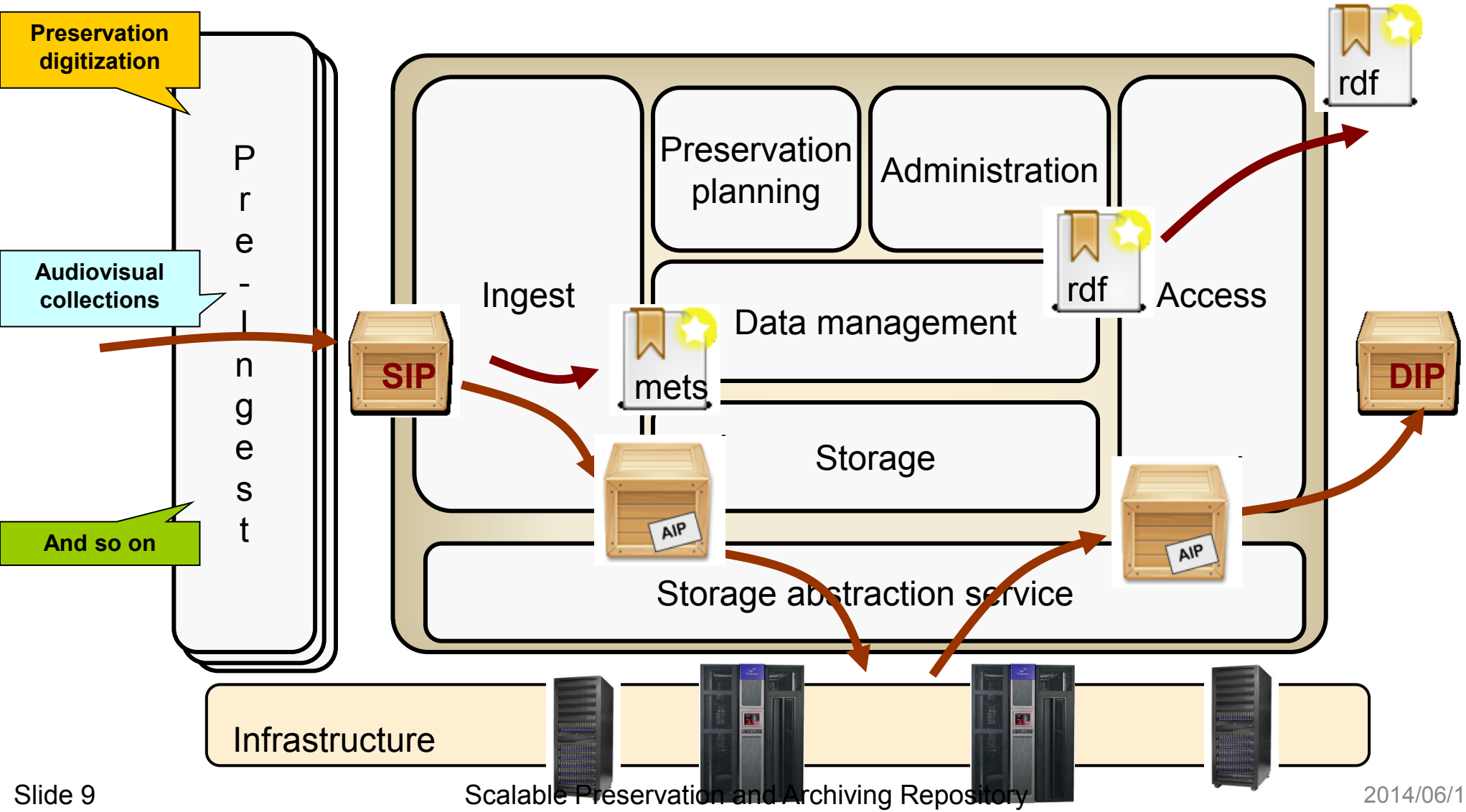
Agenda

- BnF and digital preservation
- SPAR
- The Storage Abstraction Service
- Limitation of the Abstraction
- Conclusion

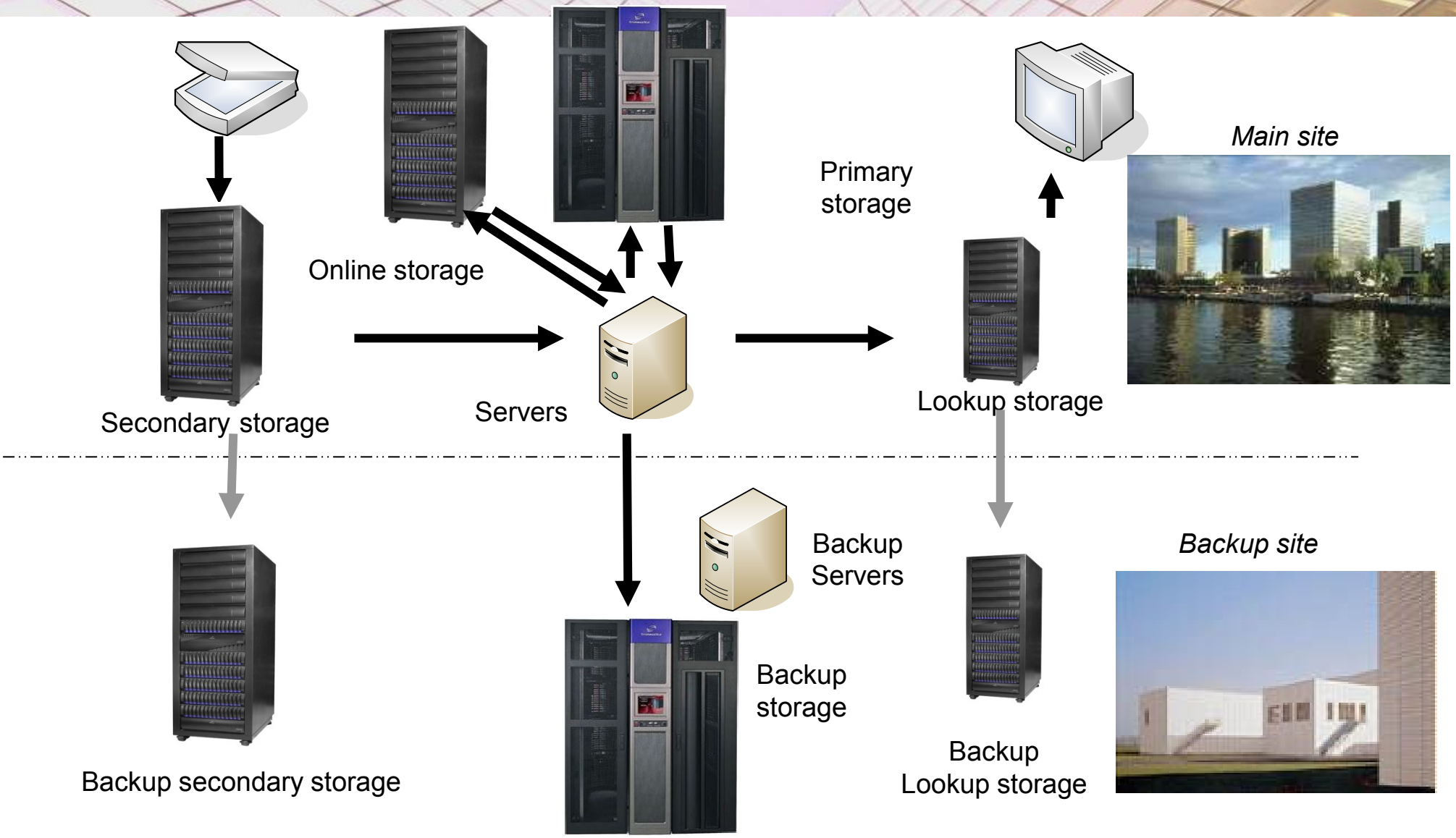
Some facts about SPAR (at 2014/06/16)

- Operation begins: may 2010
  - Current workflow of BnF's mass digitization program
  - Distributed storage over 2 sites
- 2 668 000 packages archived
- 266 703 000 data-objects (files)
- > 1,36PB (=1000TB) of raw data
- > 1 billion triples (elementary metadata)
- Very large manuscripts have been ingested (packages with size of 100GB)
- First technological migration made: new tape generation
- Available tracks: monographs, periodicals, still images, audio, video, web archiving, office documents, third-party archiving

SPAR : modular, OAIS compliant repository



Infrastructure



Infrastructure – Kind of hardware

- **Tapes**
  - Less costly
  - High capacity
  - Energy efficient
  - Slow (reach the tape, mount it in the drive, load the information…)

    ⇒ For archival documents and those that don't need quick access
- **Disks**
  - Fast access
  - Most expensive

    ⇒ For documents in the workflow

    ⇒For documents ready for quick dissemination

Decomposition in tracks

- To deal with the variability and heterogeneity of the data, definition of **tracks**
- build on the relation between the digital objects and the archival system, independently of any given organization:
  - Preservation digitization
  - Audiovisual legal deposit
  - Negotiated legal deposit (e-books, large posters,…)
  - Automatic legal deposit (surface Web)
  - Administrative production
  - Third party archiving
  - Acquisition / Donation

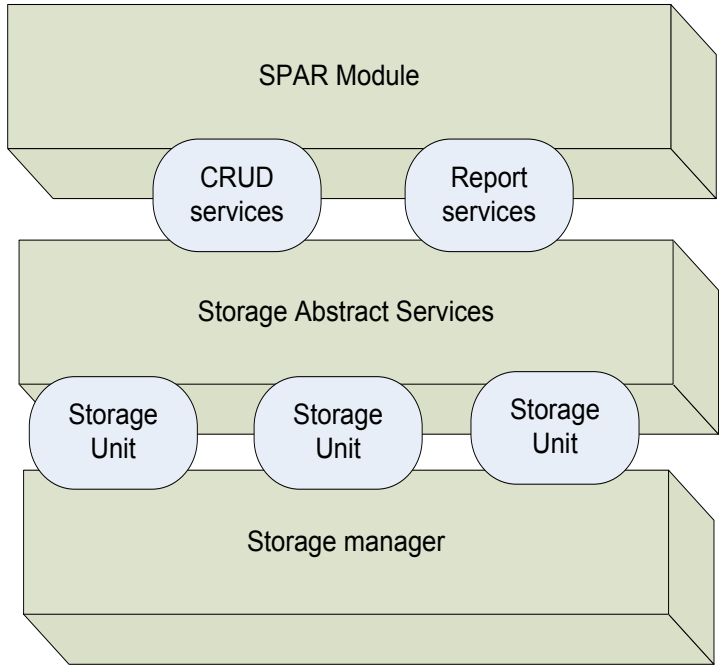
  + reference track

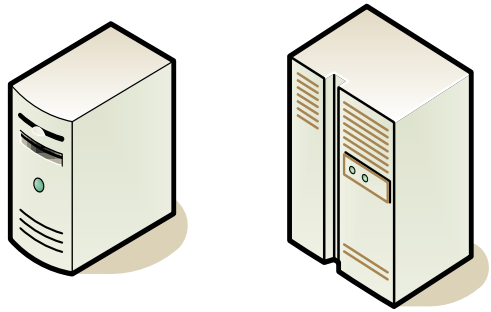
Agenda

- BnF and digital preservation
- SPAR
- The Storage Abstraction Service
- Limitation of the Abstraction
- 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 bit stream (no semantics)
    - some properties : checksum, logical name, lastAuditDate





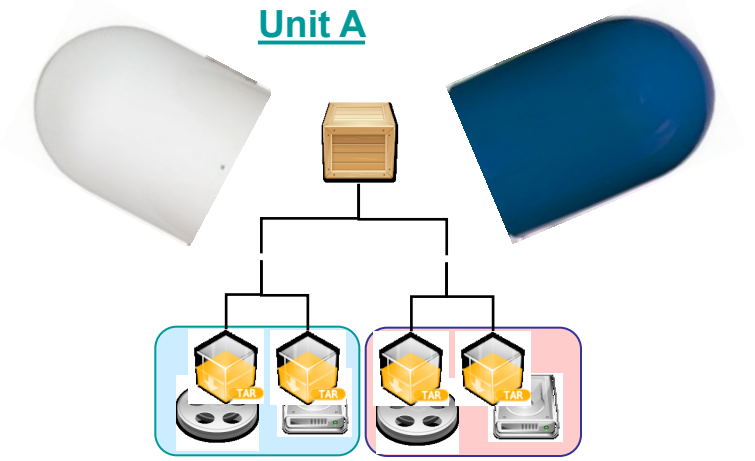
Main concepts for the SAS

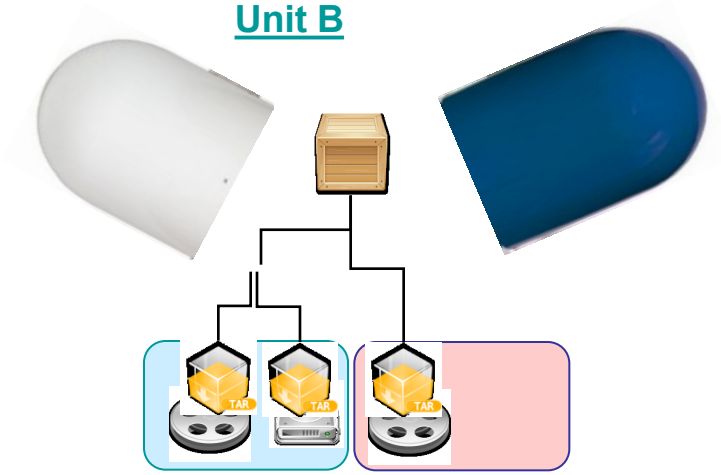
• **A record** is a information managed by the SAS. Each record is hosted on one and only one storage unit.

• **A copy (or replica):** Depending on the needs, a record can have multiple copies on the SAS. All copies are strictly identical (integrity control). The SAS only exposes one record with multiple copies. Each copy is written physically in a storage element.

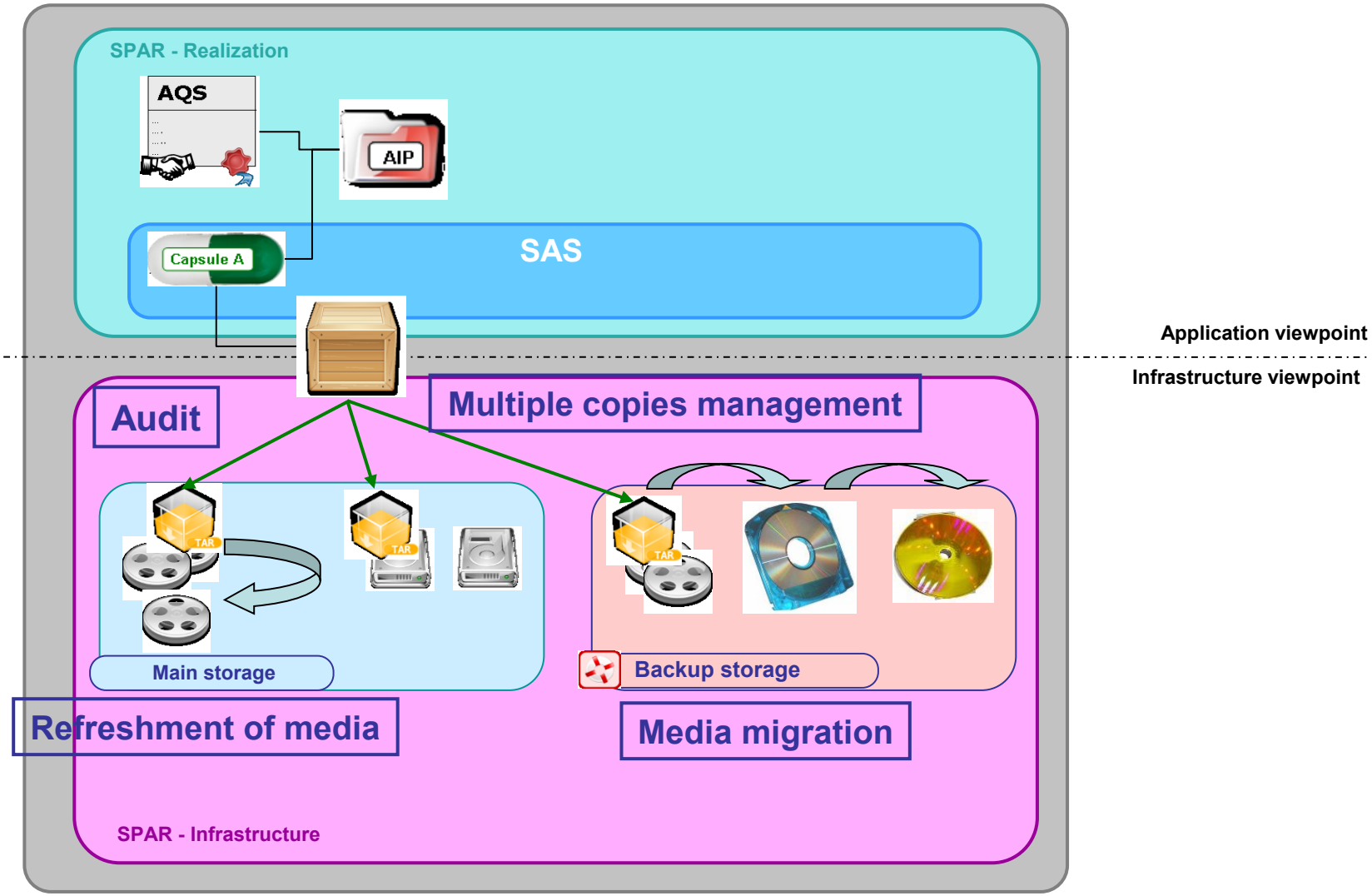
• **A storage unit:** It handles an entity to capture the characteristics of a particular storage. Each unit is declared by an administrator who defines which elements of storage are linked as well as the number of copies. Every record in one storage unit has the same characteristics of storage.

• **A storage element:** it is an entity very closed to the hardware (media, disk, tapes), except that it represents a part of a physical media. For example: a partition of a disk array or a pool of tapes…

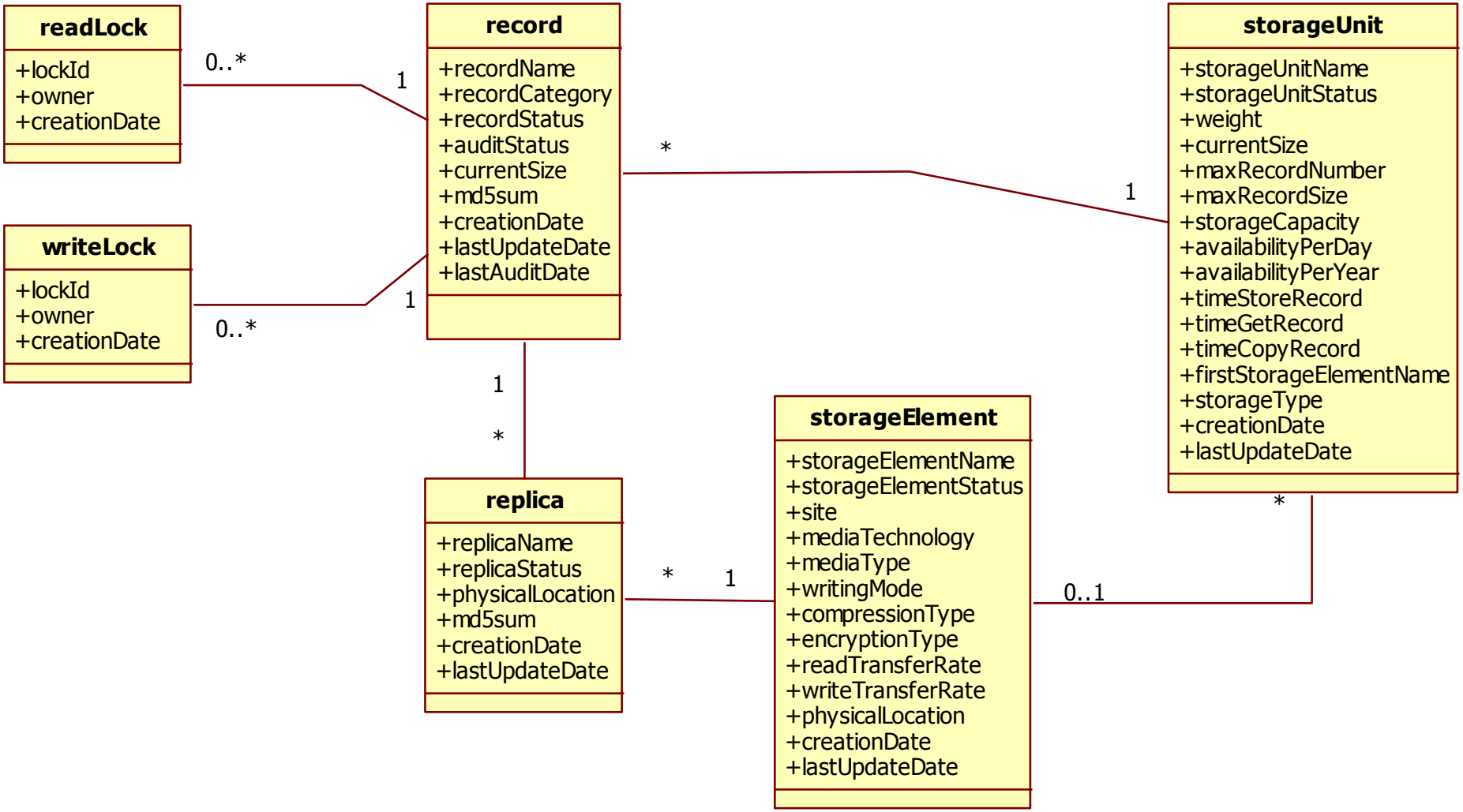




Storage abstraction



Logical Data Model

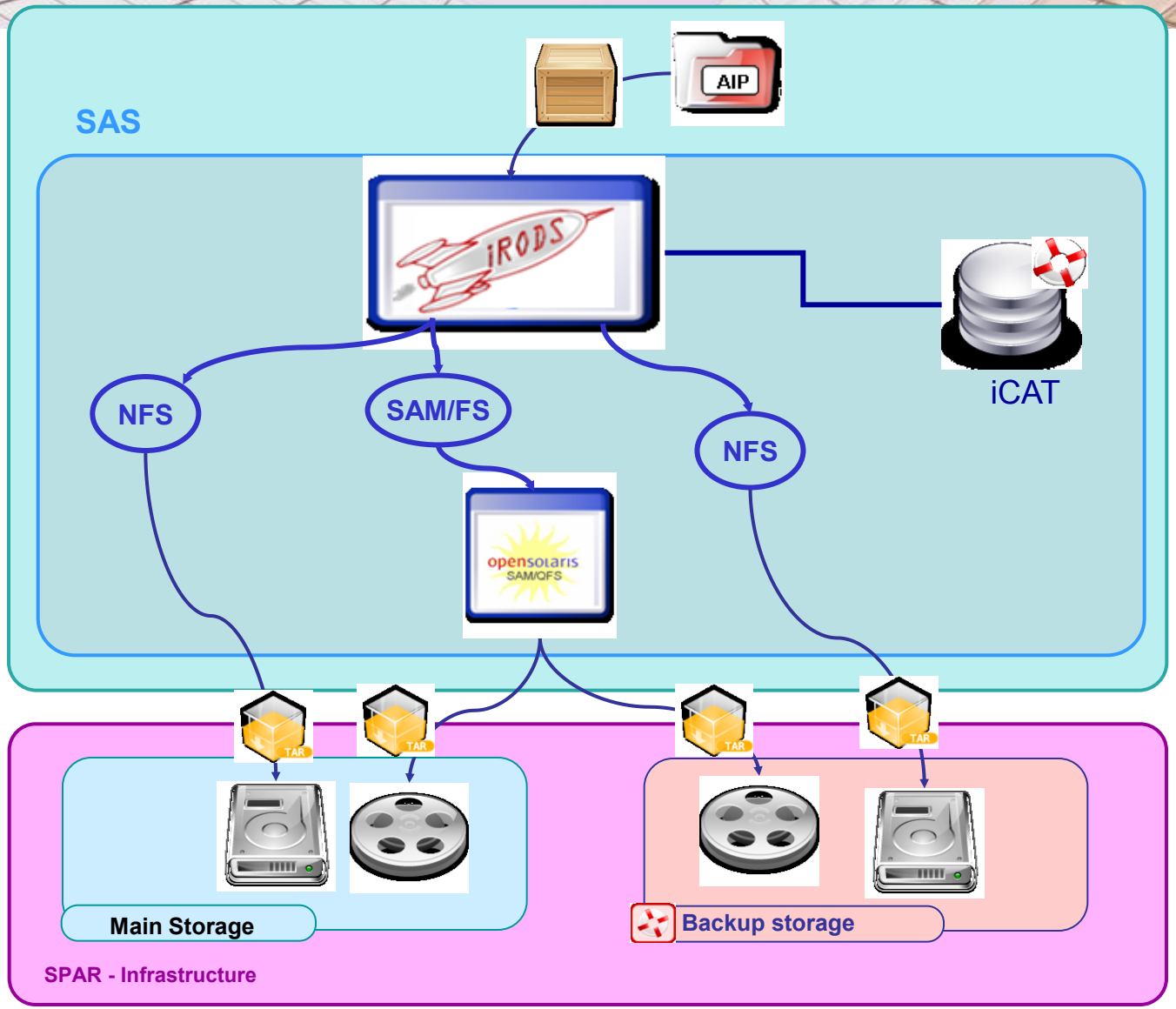


Implementation choice

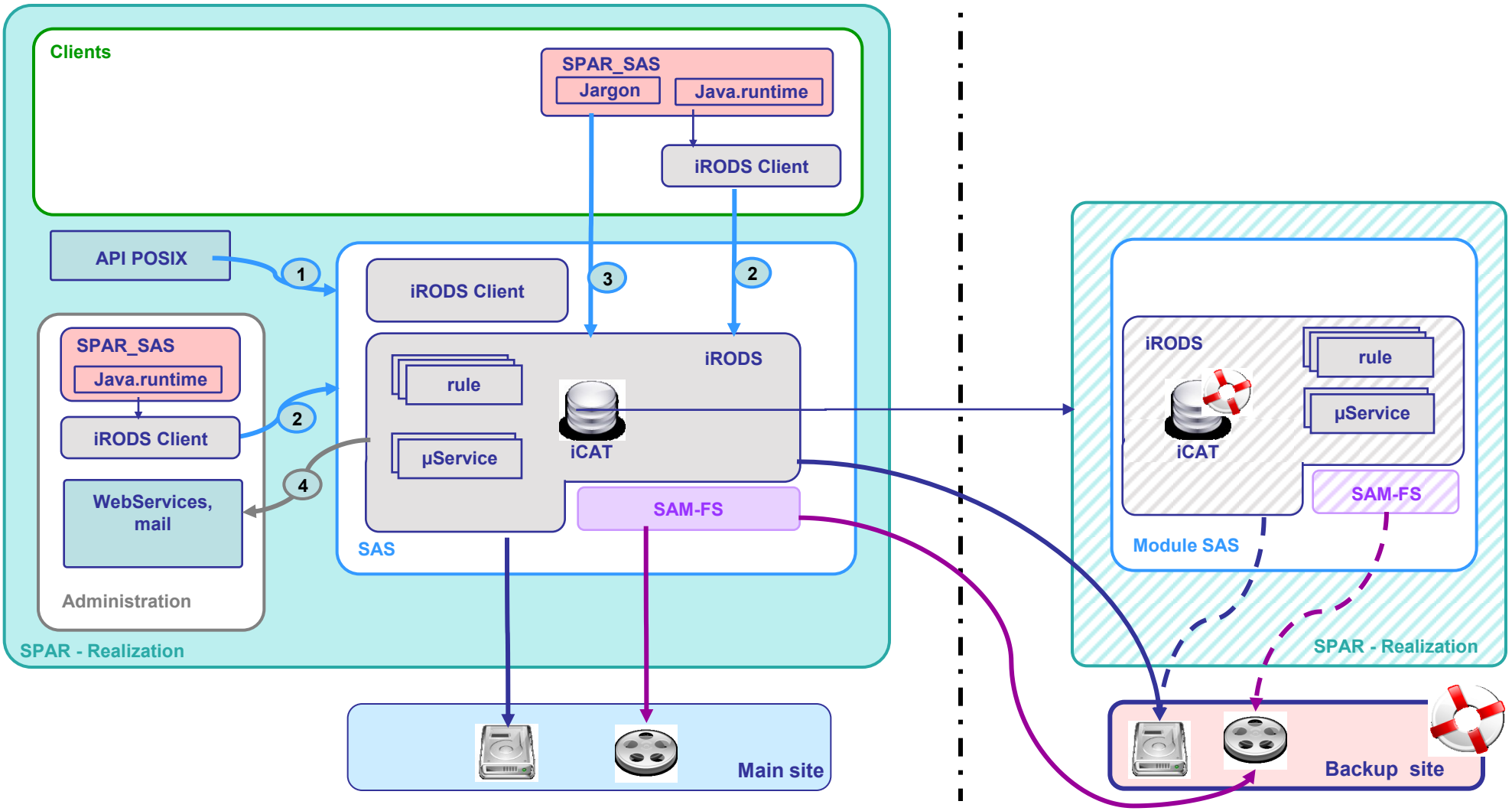
- iRods
  - scalable and proven
  - rules based

- 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

Technical implementation: iRODS



iRODS : Logical objects ⇔ Physical copies



The CRAUD interface

- To the classic CRUD verbs, we add a Audit verb
- Implemented through rules:
  - sparPut
  - sparGet
  - sparAudit
  - sparUpdate
  - sparDElete

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

Storing an AIP

```
sasPUT(*rodsName,*cPath,*localFilePath,*mainRsrc,*inChksum) {
  *putStatus = -1;
  *rodsPath = "*cPath/*rodsName";
  acObjPutWithDateAndChksumAsAVUs(
    *rodsPath,*mainRsrc,*localFilePath,
    *inChksum,*putStatus);
  if (*putStatus == 0) {
   acGetResourceMetaAttribute(*mainRsrc,"replicaResources",*replList);
    if (*replList != "" && *replList != "null") {
      acSetDataObjAttribute(
        *rodsName,*cPath,"recordStatus","WRITE_SCHEDULED");
      delay("<PLUSET>1m</PLUSET>") {
        acReplicateFile(*replList,*rodsPath,*cPath,*rodsName);
      }
    } else {
      acSetDataObjAttribute(
        *rodsName,*cPath,"recordStatus","OK");
    }
  }
}
```

Save replica 1 with checksum test

Query for replica resources

Get the list

Plan the creation of each replica

Audit an AIP

```
sasAUDIT(*rodsName,*parentColl,*stageDir,) {
  *replicasToReplace = list("");
  *rodsPath = "";
  acGetFullPathFromParentCollection(*rodsName, *parentColl, *rodsPath);
  *goodReplicaEncountered = 0;
  acGetDataObjMetaAttribute(*rodsName, *parentColl, "MD5SUM", *objStoredChksum);
  acGetDataObjLocations(*rodsName, *parentColl, *matchingObjects);

  foreach (*obj in *matchingObjects) {
    msiGetValByKey(*obj, "DATA_PATH", *objPhysicalPath);
    msiGetValByKey(*obj, "RESC_NAME", *currRescName);
    *objPhysicalMD5 = "";
    msiExecCmd("rodsMD5sum", "*objPhysicalPath", "null", "null", "null", *cmdOut);
    msiGetStdoutInExecCmdOut(*cmdOut, *objPhysicalMD5);
    if (*objStoredChksum == *objPhysicalMD5) {
      *goodReplicaEncountered = 1;
    } else {
      *replicasToReplace = cons(*replicasToReplace,"*currRescName");
    }
  }
  if (*goodReplicaEncountered == 1) {
    if (*replicasToReplace != "") {
      acSetDataObjAttribute(*rodsName, *parentColl, "recordStatus", "DIRTY");
      *allReplicaReplacementOk = 1;
      foreach (*replica in *replicasToReplace) {
        acReplaceStaleReplica(*rodsPath, "*replica", *replStatus);
        if (*replStatus != 0) {
          *allReplicaReplacementOk = 0;
        }
      }
      if (*allReplicaReplacementOk == 1) {
        acSetDataObjAttribute(*rodsName, *parentColl, "recordStatus", "OK");
      }
    } else {
      acSetDataObjAttribute(*rodsName, *parentColl, "recordStatus", "OK");
    }
  } else {
    acSetDataObjAttribute(*rodsName, *parentColl, "recordStatus", "ERROR");
  }
}
```

Retrieve all the replicas

Calculate the checksum

Replace the bad replica

Update the record status

Retrieving an AIP

```
sasGET(*rodsName,*parentColl,*stageDir,*getStatus) {
  *replicasToReplace = list("");
  *rodsPath = "";
  acGetFullPathFromParentCollection(*rodsName, *parentColl, *rodsPath);
  *goodReplicaEncountered = 0;
  acGetDataObjMetaAttribute(*rodsName, *parentColl, "MD5SUM", *objStoredChksum);
  acGetDataObjLocations(*rodsName, *parentColl, *matchingObjects);
  foreach (*obj in *matchingObjects) {
    msiGetValByKey(*obj, "DATA_PATH", *objPhysicalPath);
    msiGetValByKey(*obj, "RESC_NAME", *currRescName);
    *objPhysicalMD5 = "";
    msiExecCmd("rodsMD5sum", "*objPhysicalPath", "null", "null", "null", *cmdOut);
    msiGetStdoutInExecCmdOut(*cmdOut, *objPhysicalMD5);
    if (*objStoredChksum == *objPhysicalMD5) {
      *goodReplicaEncountered = 1;
      break;
    } else {
      *replicasToReplace = cons(*replicasToReplace,"*currRescName");
    }
  }
  if (*goodReplicaEncountered == 1) {
    if (*replicasToReplace != "") {
      acSetDataObjAttribute(*rodsName, *parentColl, "recordStatus", "DIRTY");
      foreach(*replica in *replicasToReplace) {
        delay("<PLUSET>1m</PLUSET>") {
          acReplaceStaleReplica(*rodsPath, *replica, *replStatus);
        }
      }
    }
    *localFilePath = "*stageDir/*rodsName";
    *getContext = "getting *rodsName to  *localFilePath from resource : *currRescName ...";
    msiDataObjGet(*rodsPath, "localPath=*localFilePath++++rescName=*currRescName++++forceFlag=",
    *getStatus);
  } else {
    acSetDataObjAttribute(*rodsName, *parentColl, "recordStatus", "ERROR");
  }
}
```

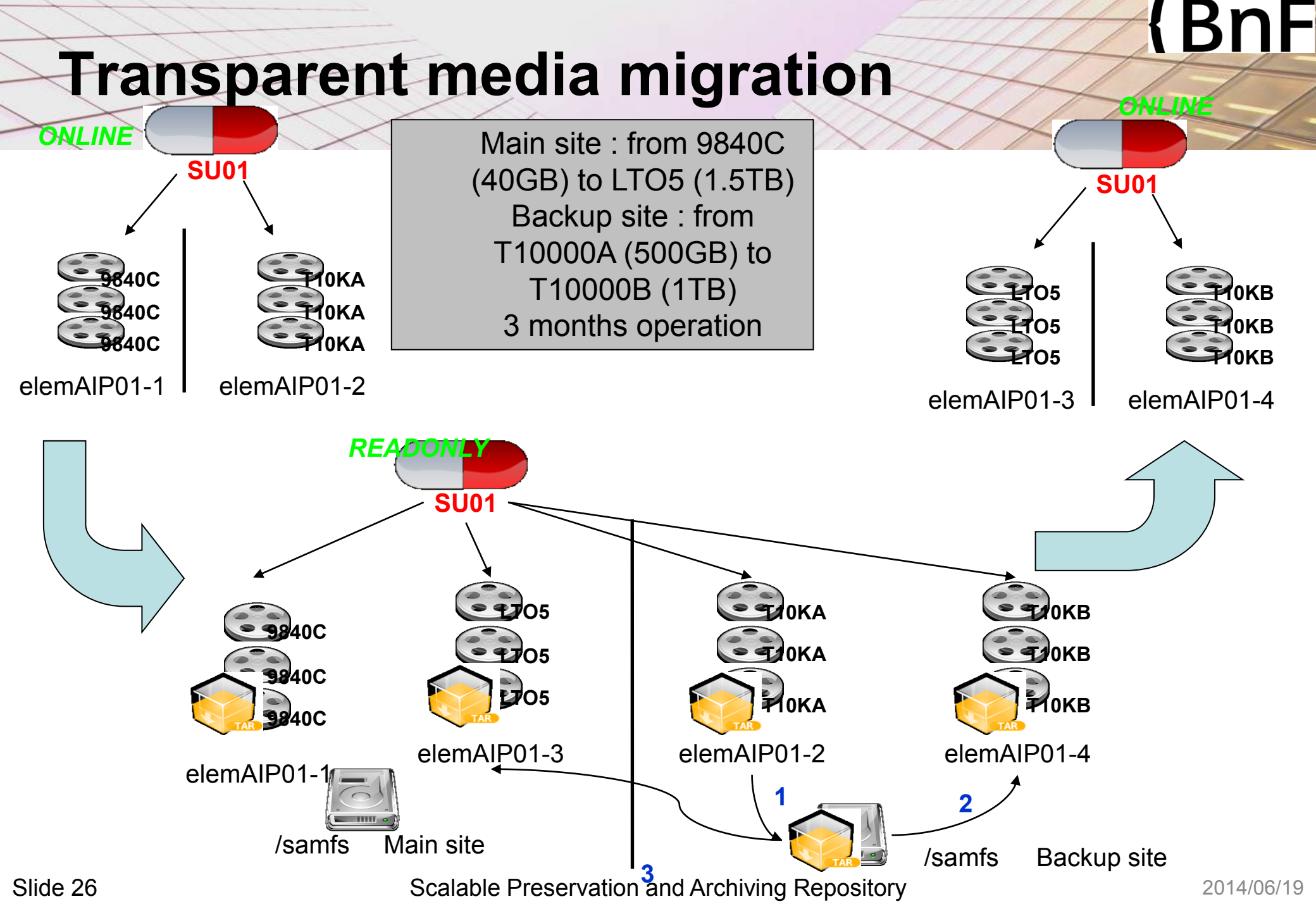
No good replica yet found

Calculate the checksum

Plan the replacement of the bad replica

Get the first good replica

Transparent media migration



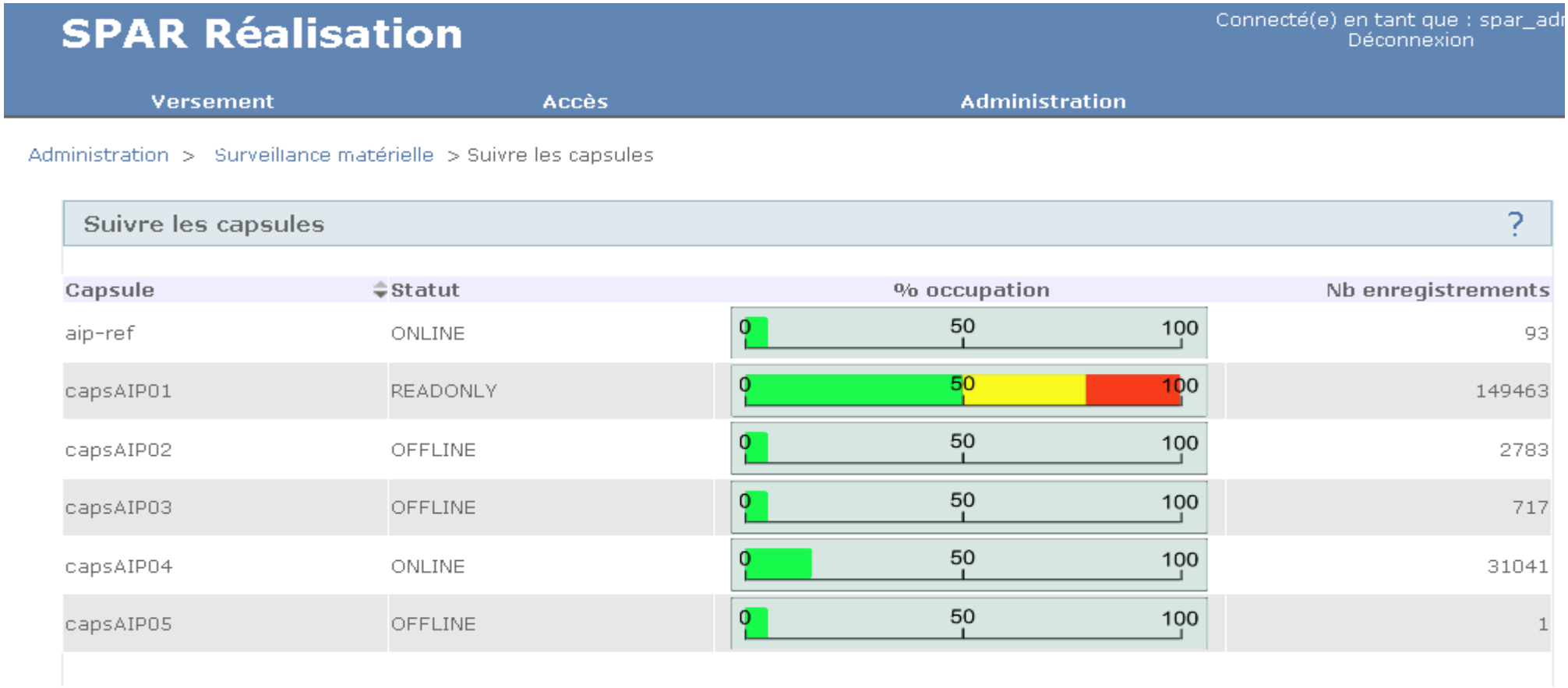
Agenda

- BnF and digital preservation
- SPAR
- The Storage Abstraction Service
- Limitation of the Abstraction
- Conclusion

Issues of storage and SAS

- Provide a intelligent storage system:
  - Driven by the services it should supplied
  - Auto-healing
  - Able to make regular audits
  - Able to take into account new hardware, using replication migration
  - Able to migrate from equivalent hardware (tape change), using refreshment migration

Storage resources management



Limits on the abstraction

- With non-random access ressources, need to know where the records are
- Batch ‘vsn2irods’ to inform on where are located the record
- => allow for optimization of retrieval and auditing

Technical metadata

- **Additionnal metadata linked to the archival package**
  - **sparprovenance:recordLastUpdateDate**: date of the last update of a record
  - **sparprovenance:recordLastAuditDate**: date of the last audit of a record
  - **sparreference:replicaLocations:** serial numbers of the tapes where the replicas are located

Notion of plans

- The [Administration] module executes plans over a large number of archival packages
- In order for the plans to be efficient, need to know where some of the copies are located => register the location in the data management index (triplestore)

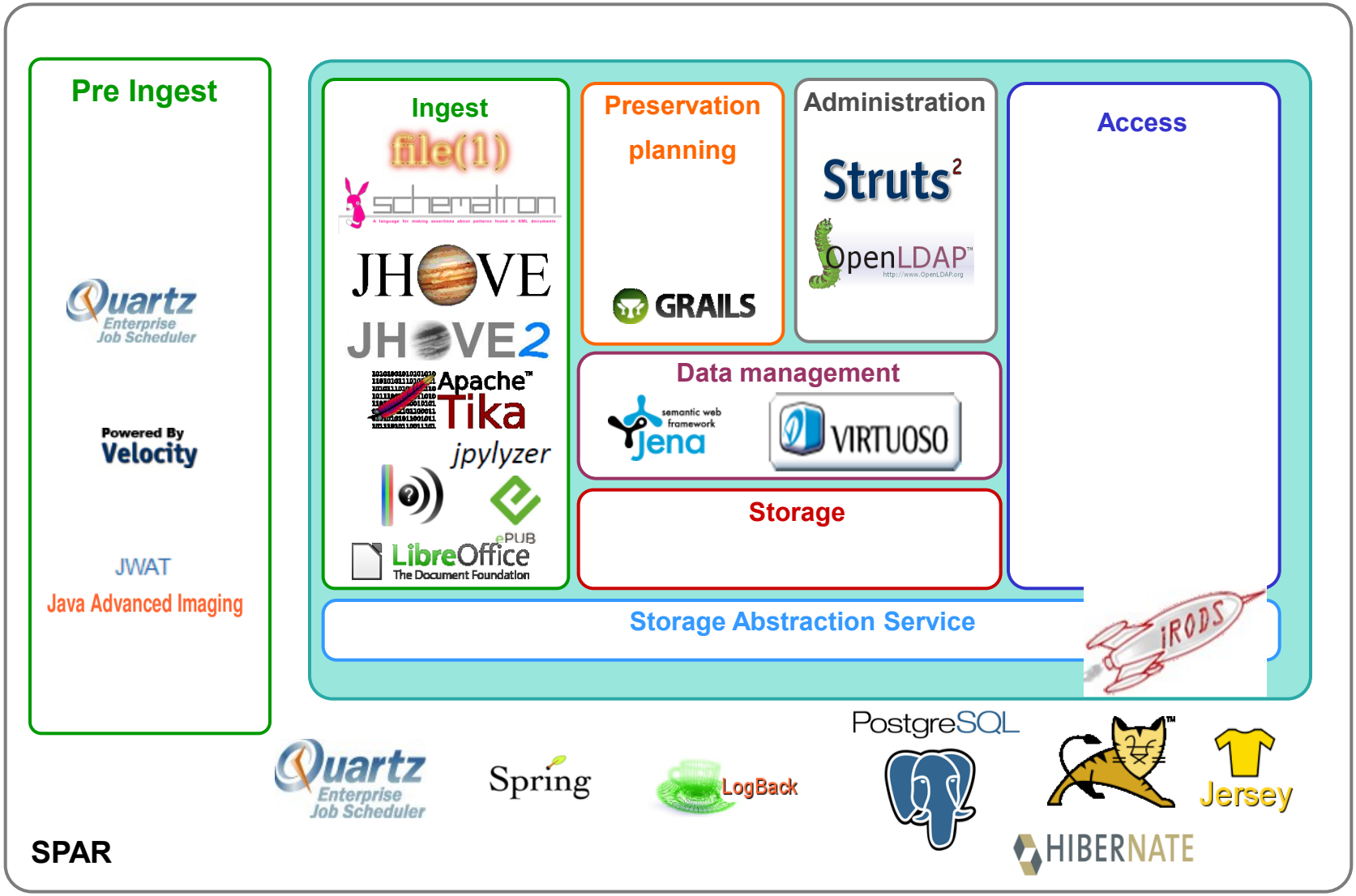
Agenda

- BnF and digital preservation
- SPAR
- The Storage Abstraction Service
- Limitation of the Abstraction
- Conclusion

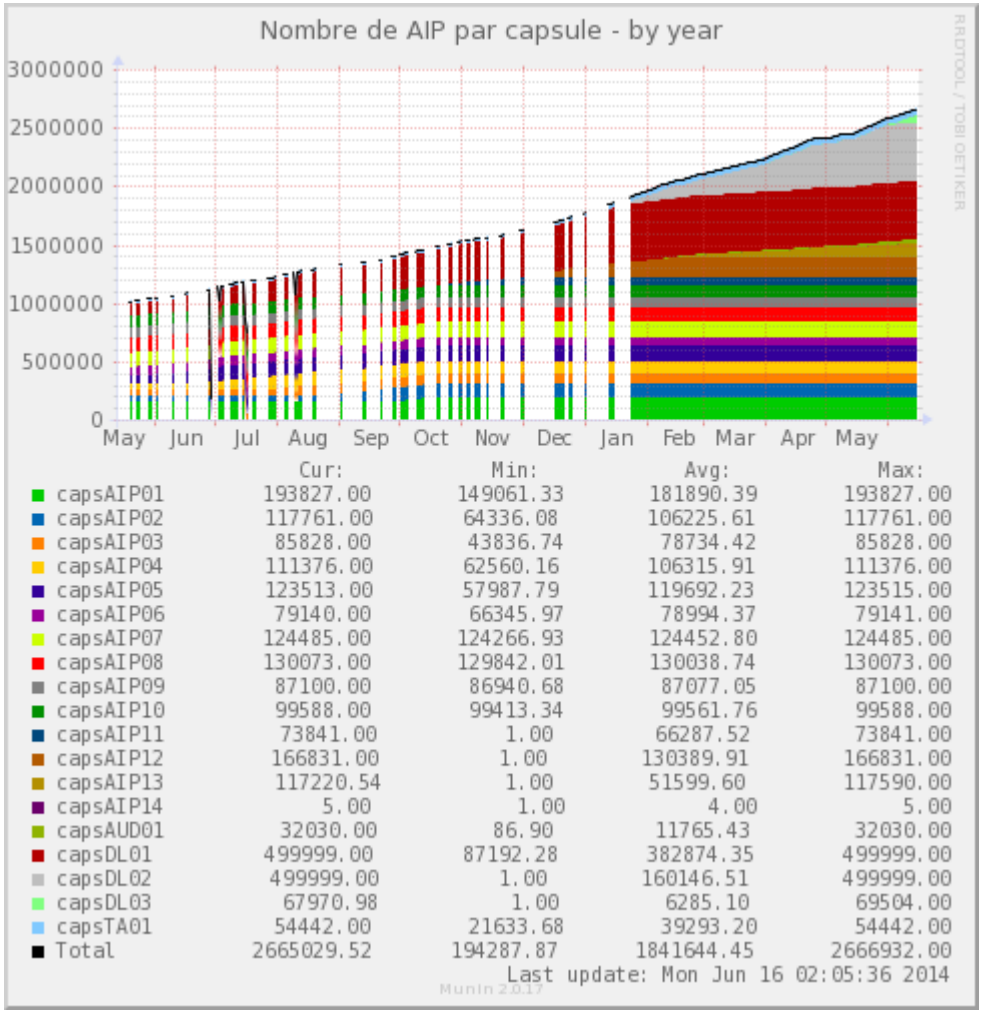
Conclusion

- iRods provides a great way to abstract the application from the work of the administrators
- Work to do:
  - Use of Jargon
  - Migration to iRods 4.0 and use of composable resources
  - Better integration with “tapes” resources

Open source usage



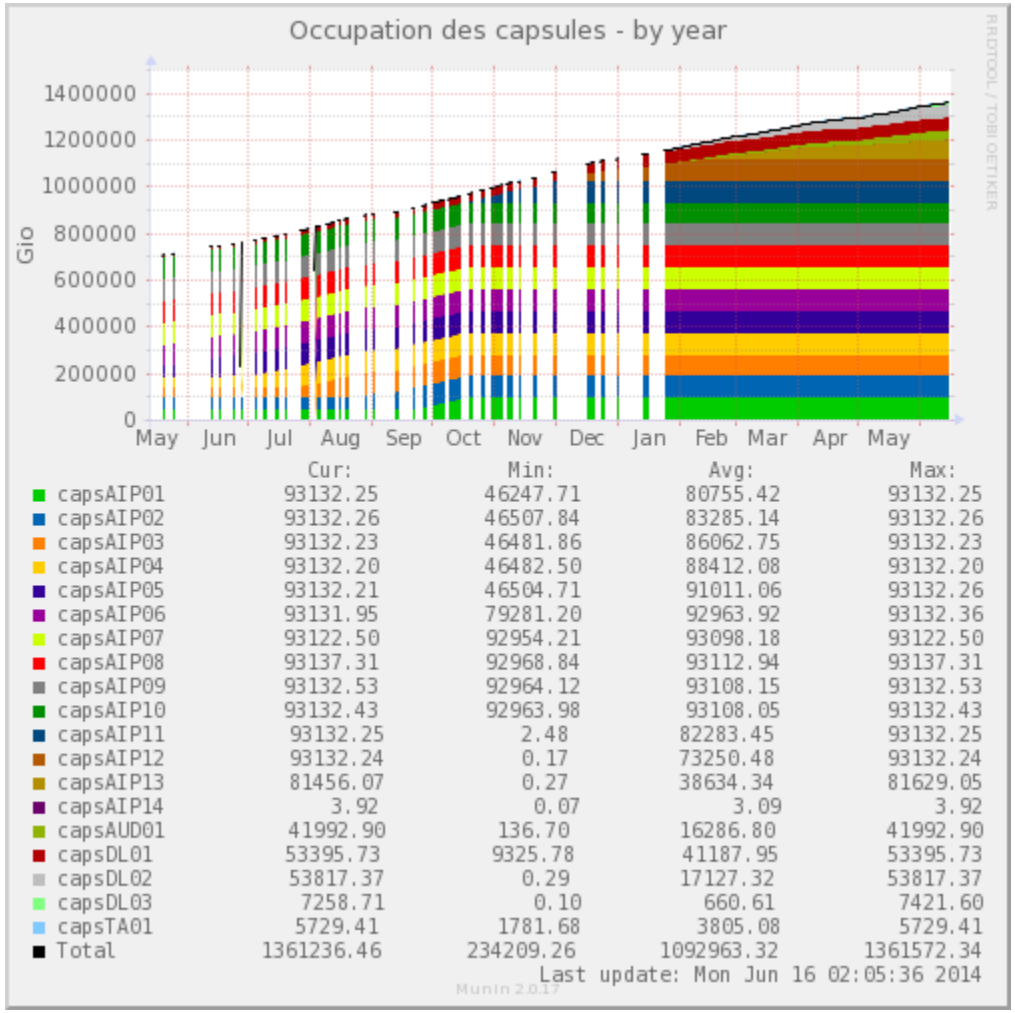
Volume



	Cur:	Min:	Avg:	Max:
capsAIP01	193827.00	149061.33	181890.39	193827.00
capsAIP02	117761.00	64336.08	106225.61	117761.00
capsAIP03	85828.00	43836.74	78734.42	85828.00
capsAIP04	111376.00	62560.16	106315.91	111376.00
capsAIP05	123513.00	57987.79	119692.23	123515.00
capsAIP06	79140.00	66345.97	78994.37	79141.00
capsAIP07	124485.00	124266.93	124452.80	124485.00
capsAIP08	130073.00	129842.01	130038.74	130073.00
capsAIP09	87100.00	86940.68	87077.05	87100.00
capsAIP10	99588.00	99413.34	99561.76	99588.00
capsAIP11	73841.00	1.00	66287.52	73841.00
capsAIP12	166831.00	1.00	130389.91	166831.00
capsAIP13	117220.54	1.00	51599.60	117590.00
capsAIP14	5.00	1.00	4.00	5.00
capsAUD01	32030.00	86.90	11765.43	32030.00
capsDL01	499999.00	87192.28	382874.35	499999.00
capsDL02	499999.00	1.00	160146.51	499999.00
capsDL03	67970.98	1.00	6285.10	69504.00
capsTA01	54442.00	21633.68	39293.20	54442.00
Total	2665029.52	194287.87	1841644.45	2666932.00

Last update: Mon Jun 16 02:05:36 2014

Munin 2.0.17



	Cur:	Min:	Avg:	Max:
capsAIP01	93132.25	46247.71	80755.42	93132.25
capsAIP02	93132.26	46507.84	83285.14	93132.26
capsAIP03	93132.23	46481.86	86062.75	93132.23
capsAIP04	93132.20	46482.50	88412.08	93132.20
capsAIP05	93132.21	46504.71	91011.06	93132.26
capsAIP06	93131.95	79281.20	92963.92	93132.36
capsAIP07	93122.50	92954.21	93098.18	93122.50
capsAIP08	93137.31	92968.84	93112.94	93137.31
capsAIP09	93132.53	92964.12	93108.15	93132.53
capsAIP10	93132.43	92963.98	93108.05	93132.43
capsAIP11	93132.25	2.48	82283.45	93132.25
capsAIP12	93132.24	0.17	73250.48	93132.24
capsAIP13	81456.07	0.27	38634.34	81629.05
capsAIP14	3.92	0.07	3.09	3.92
capsAUD01	41992.90	136.70	16286.80	41992.90
capsDL01	53395.73	9325.78	41187.95	53395.73
capsDL02	53817.37	0.29	17127.32	53817.37
capsDL03	7258.71	0.10	660.61	7421.60
capsTA01	5729.41	1781.68	3805.08	5729.41
Total	1361236.46	234209.26	1092963.32	1361572.34

Last update: Mon Jun 16 02:05:36 2014

Munin 2.0.17

Main computer room