# J-TEXT DB Cloud (JDC) White Paper

Abstract

This document is not intended to be a design document of the JDC nor a requirement statement. You can take this as an inspiration for the requirement and design of the JDC. This document describes the aspect that must be taken account in the design of the JCD and nothing more.

## Introduction

I have a dream, that is to build a system for scientist to acquire, preserve, analysis the data and most important to collaborate easily without considering the infrastructure that handles the data. The key to this system is a data base that not only preserves the data, but manages it, so the data can be preserved, organized, discovered, understood, analyzed and shared.

This document is greatly inspired by the NASA OAIS[1] design. So If you are going to design the JDC, I order you to read *REFERENCE MODEL FOR AN OPEN ARCHIVAL INFORMATION SYSTEM (OAIS)[1]* first, it’s mandatory.

## The Scope of JDC

Before we start, we have to make clear that what we are going to achieve.

Yes, we are going to make a cloud database that is for scientists. But what is a cloud data base for scientist, is it software or, service or a deliverable product? What does it include and exclude?

First we must define scientific research and scientific data. I believe that scientific research is always data centric. That is to say without data, this cannot be called science. So let’s give the following definition: Scientific research is an activity that by studying existing data scientists will come up with hypothesis, then they either design experiments or design data analysis method; by running the experiment or data analysis they will get result data; by analysis and visualization of the result data, the will get conclusive data to prove or deny the hypothesis, and generates report and publication. As show in Fig. 1 every step is associated with data, either accessing or generating data. This, as said in the previous section, it needs means of preserving, organizing, discovering, understanding, analyzing and sharing of the data. The red line represents the most intensive data interaction, and that what we will focus on.



Fig. 1 Scientific research process

So the following is what will be covered, data model that contains the information that is needed to be preserved; preservation technology that keep the contained information; query and deliver of the date so it can be accessed; analysis work flow so data can be analyzed; data organization; data sharing; data migration and backup; data acquisition that will populate the database; data visualization; and the management of the whole system.

## The Data Model

Data model is how the data is contained throughout the JDC system. Data model is the key of most aspect of JDC design. It defines how we interpret and understand the information with in it, and how we preserve and manage the information.

Take MDSplus for an example. The data model in early MDSplus is simple: a hierarchical structure of signals which is multi-dimensional array. This limits the information that can be contained, mainly wave forms, although many others can be sterilized into arrays. And it also limits the way of accessing the information. You can access the whole signal only. Later the data model changed bringing the segment concept. This changed the way of accessing the data by allowing read part of the data. The MDSplus model also limits the way of interpretation of the information since it has no way of preserve the metadata.

The HDF5 however is a much more complex compared to MDSplus in data model. It has built in metadata for the information in the file, and allowing flexible access to any part of the information. But he metadata in the HDF5 file is not query able and not enough.

### OPEN ARCHIVAL INFORMATION SYSTEM (OAIS)

This is an archive system designed by NASA. It’s more like a design guide for archive system. It’s becoming ISO standard and ITER DB is considering adopt some of its design. After reading it, I found many things we can adopt in our design.

Some basics of the OAIS first. In OAIS the actual information we want to preserve is called content information. In order to understand the content information, we need two thing: Preservation Descriptive Information (PDI) and the knowledge base. Let have an example: “你是个傻子” is the content information we want to preserve, and “text in Chinese” is the PDI, you have to be able to read in Chinese this is the knowledge base, clear? The PDI is not complete here, there may be like encodings information (GB2312 or UFT8) and many more. The above to is encapsulated in a package called information package, as in Fig. 2. There are many other definition in OAIS which I think is less significant.

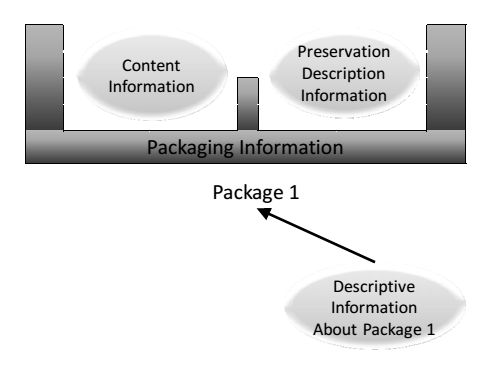


Fig. 2 OASI information package

Next we have Fig. 3, the data interaction. The producer generates information and packs it as stated before as an Information Package, this is called Submission Information Package (SIP). The SIP is handled by the archive system and preserved as Archive Information Packages (AIP). The consumer that needs information is delivered with Dissemination Information Packages (DIP).

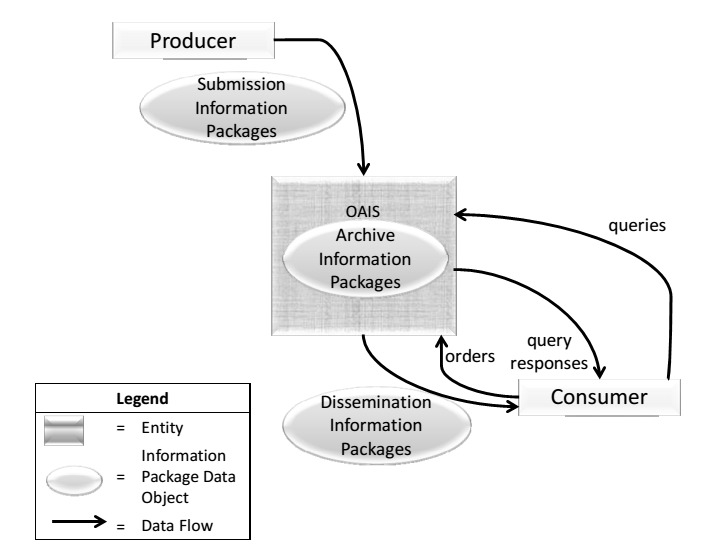


Fig. 3 Data interaction

SIP, AIP and DIP are all information packages. Why distinct them? Here is an example to show. In long pulse fusion experiment like ITER, the data is streamed continuously to the database. Each time a data of a signal is send to the database it will be appended to the HDF5 files for this signal. The each package streamed to database is an SIP, and the HDF5 file is an AIP. You can see that there are multiple SIPs that will make up a bigger AIP. Each SIP may contain very little PDI, since they are for the same signal. However the AIP contains a complete PDI. The AIP gets updated when each SIP came including its PDI. When the user want some data of the signal, it may not want the whole signal, but part of it, maybe a section of it or decimated signal. Then the database will read part of the AIP and assembles a DIP for the consumer.

## The Variety of Data

As said MDSplus is best suited for array types of data, HDF5 is for much various data type but still best for array. Data in science comes in different shapes, but somehow they all resembles the array type.

## PDI

For PDI, it is the metadata; it should contain complete information to understand the content information. OAIS said that PDI *must include information that will support the trust in, the access to and context of the Content Information over an indefinite period of time. The PDI must include information that is necessary to adequately preserve the particular Content Information with which it is associated. It is specifically focused on describing the past and present states of the Content Information, ensuring it is uniquely identifiable, and ensuring it has not been unknowingly altered.*

I totally agree with OAIS’s design, so I quite the types of information that PDI will contain:

* **Reference Information** identifies, and if necessary describes, one or more mechanisms used to provide assigned identifiers for the Content Information. It also provides those identifiers that allow outside systems to refer, unambiguously, to this particular Content Information. Examples of these systems include taxonomic systems, reference systems and registration systems. In the OAIS Reference Model most if not all of this information is replicated in Package Descriptions, which enable Consumers to access Content Information of interest.
* **Context** Information documents the relationships of the Content Information to its environment. This includes why the Content Information was created and how it relates to other Content Information objects existing elsewhere.
* **Provenance** Information documents the history of the Content Information. This tells the origin or source of the Content Information, any changes that may have taken place since it was originated, and who has had custody of it since it was originated, providing an audit trail for the Content Information. This gives future users some assurance as to the likely reliability of the Content Information as it contributes to evidence supporting Authenticity. Provenance can be viewed as a special type of context information.
* **Fixity Information** provides the Data integrity checks or validation/verification keys used to ensure that the particular Content Information object has not been altered in an undocumented manner. Fixity Information includes special encoding and error detection schemes that are specific to instances of Content Objects. Fixity Information does not include the integrity preserving mechanisms provided by the OAIS underlying services, error protection supplied by the media and device drivers used by Archival Storage. The Fixity Information may specify minimum quality of service requirements for these mechanisms.
* **Access Rights** Information identifies the access restrictions pertaining to the Content Information, including the legal framework, licensing terms, and access control. It contains the access and distribution conditions stated within the Submission Agreement, related to both preservation (by the OAIS) and final usage (by the Consumer). It also includes the specifications for the application of rights enforcement measures.

I will explain some. Those I didn’t explain is straight forward enough.

**Reference Information** is like an Id to the data, e.g. a file path, or an ID to a MongoDB document.

**Context** is about how the content information is created, like the sample rate.

**Provenance** is used mainly in work flow system. It’s about the origin of this information. So if the origin of this information cannot be trusted, neither should the derived information be trusted.

## Conclusions

For Data model, we need to preserve content information and PDI;

For content information, we have to consider the variety of the data and the fact that they all somehow have related to array types;

For PDI, it is the metadata; it should contain complete information to understand the content information.

## Preserve Data

Data preservation comes immediate after the data model is designed. It is about how to store the data using a specific technology.

HDF5 is mostly data model, but is covers part of preservation. First the HDF5 is based on files, second it comes with APIs that manipulate the HDF5 files. That’s the preservation ends with HDF5. If you are using HDF5 for preservation you will work out the file system NTFS or EXT. Of course you have to decide to run the database on RAM or SSD or hard disk. And the disk arrays, servers, clusters, storage network they are or preservation issues.

To determine what preservation we need, we have to consider the following:

The cost we can afford, RAM is the fastest but the most expensive, disks are ok but if the data does not need to be online all the time tapes are a cheap choice.

The performance requirement, budget will determine the preservation technology choices. You may have to come up a hierarchical solution like using a full stack of ram, sdd, raid and tapes.

## Access Data

### Query Data

Data query is to let the scientist to find information they need based on the information they provide and the PDI stored in the database.

The query should be able to find the AIPs (yes, a query may return multiple AIPs) and which parts of an AIP is needed by the query.

Query depends heavily on the preservation of the PDI. The PDI preservation should be properly indexed to enable fast query. To achieve effective indexing, the PDI data model and the content of the user query must be considered.

### Deliver Data

Data delivery is to user the result from a query to generate DIPs. This is not a very complex process. When the AIP is picked out by the query, the delivery process will assemble the AIPs or parts of AIPs into a DIP, mainly based on the requested data type and the required representation format. E.g. when a decimated waveform is required, you will get lots of chunks of points from the raw signal; then you put them in an array and generate timestamps for them because this is a waveform; then if the user required a binary format, just send the user the bits, if the user required a JSON, the you will serialize it into JSON and the JSON is the DIP delivered to the user.

## Performance

We are initially doing this for ITER. ITER will generate data at 50GB/s rate continuously which is large enough to give serious consideration. And if you are host large data for large consumers, deliver performance is also critical. So performance is a hard issue. The following aspect must be considered for performance.

### Preservation I/O

The read and write speed of a storage device is the first thing that come to you mind I guess. Yes tape will not handle any online data reading or writing. Disk is ok for small data, but is this is not enough, you get yourself a RAID. If RAID is not fast enough, you may have to build a SAN. Cluster is a good way which provides good scalability. Load balancing in the cluster is a great factor affecting the performance.

### Ingest Performance

Outside the preservation, the ingest function is another performance limiter. User will submit information directly to the ingest function, which handles the user submitted information and prepare proper AIPs and then write into the preservation. So the Ingest should be able to buffer the user request and be able to simultaneously handle vast user requests. Ingest is often archived by using a cluster, goo load balancing and work schedule strategy is essential for the performance.

#### Access Performance

The main limiter for user to get the data they want is the query performance. When you have a large amount of data entity, a fast and flexible indexing in the most important. Other performance factor may be similar to the ingest aspect.

### Protocol Overhead

How users ingest their data to the database and how you deliver data to them affect the performance hugely. Protocol is how you format your information when sending it though the network. If not properly handled, it may slow down the transmitting by a factor of 10 or 100 times. E.g. if you have a waveform and you encapsulated it in a binary format, it may be only 1/10 the size of the JSON serialized format, or even smaller. The same information but the JSON wasted 10 times more bandwidth! But basic encapsulation is necessary since you need basic PDI with you content information to form a SIP.

### Network performance

Another overhead is related to network protocol, this is similar to the protocol overhead. Some network protocols include many overhead messages in each package like http. And some need to perform lots if handshakes each connection is made, if the connection is closed on each package and reopen on next one, a lot of bandwidth is wasted on handshake. So is a connection is made and data can be constantly streamed without reconnection is efficient.

Here is an example: in ITER DAN design, the fast controller many used as high speed DAQ are the data producers will generate large data constantly during a pulse[2]. Instead buffer data on limited local memory and send to database on their own, they send the data to a transmit accelerator. The accelerator has many 10G network interfaces to connect the fast controllers and an enormous amount of RAM. As shown in Fig. 4, it buffers the data from many fast controllers and sends to the database with is 40G Ethernet connection. It buffers the data and generates a big enough package (SIP) to sustain its 40G connections, thus achieve best network efficiency.

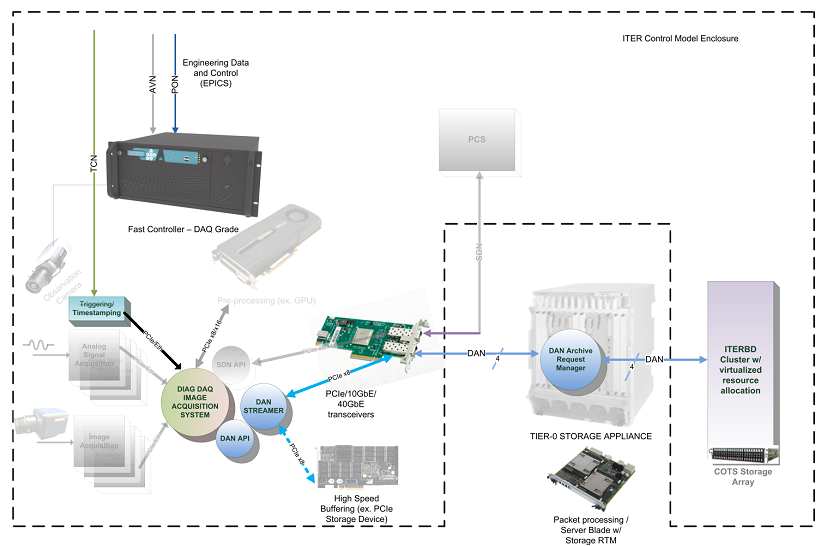


Fig. 4 ITER DAN accelerator

## Data Analysis Workflow

To analysis the data automatically the database have to support some sorts of workflow system. Besides data ingest and access, the workflow requires two more thing, provenance and a work scheduler.

Scheduler is something that starts and organizes the workflow. User can submit their data processing programs to the workflow system, may be a virtual machine. Multiple processing programs can that executed at a sequence is a workflow. The scheduler will get the required data and feed to a program, get the result and feed to the next program, until the final result is generated. The result data as well as some of the intermediary data is written into the database. This requires the work scheduler is able to read and write data from the database and handles event or generate events.

The provenance function is important for data quality. When a result is based on calculation using un-trustable data then the result is un-trustable, if the algorithm generates the result is prove to be wrong, the result and any deriving result is wrong. And some algorithm may bring errors the error may be propagated. So the workflow should be able to track the origin of the data so when something is bad, all the massed up data can be picked out.

## Organize Data

Many teams, institutes, scientist will use the system. How to manage their data and workspace? There should be a structure to organize all data in a team or in on experiment.

Access control is part of the data organize function. User may grant other to access the data that the owned. The data access can be part of the data model problem.

## Share Data

Here is something you may underestimate. It’s easy to share the data in the database you just give them a reference to the data (id) and make sure they have right to access it.

But what if the data is not preserved in the database? Yes here is a situation that never been considered before. It is not cheap to store everything in a centralized database. The storage and the bandwidth that transfer the data to the database are expensive.

The Scibox[3] is a scientific data sharing tool. Instead of uploading all data to the cloud database, it only transfer the data needed to the consumer to the cloud. Let’s look how this solution evolved. First, in science community, there is a tool called GridFTP, it is like a FTP server, scientist send data to the server .Whoever needs the data can download from it. Then there is Globus, it is like the dropbox for scientist. The scientist marks the data that need to be shared, it will be sent to the cloud. At home or another lab, the data will be downloaded.

The GridFTP and Globus is not something special but Scibox is. When user generate data from the experiment, the data is not sent to the cloud but save in local database. However the complete metadata (PDI) is sent to the cloud. When user from elsewhere access the cloud, they can see all the metadata, and query the data. When the query is returned the Scibox will ask the server hosting the real data in the institute to upload the data that the user need, then deliver it to the users. The Scibox also keeps the data so later someone ask for the same data no more upload is needed.

Beside what mentioned above, to speed up data sharing CDN can be used. And we can place the hot data which is frequently accessed on fast servers.

## Migration and Backup

### Migration

Storage tech goes out of date fast. When new servers arrives and replace the old ones the data should not be affected. To move the data from the old server to the new sever is on kind on migration, but the simplest one.

When new database software arrives it not that easy like copy and paste to migrate data. Data format may be changed. Maybe HDF5 data is saved into a non-file-system database. The process must not lose any information that is preserved before.

The NIF store all the data in an Oracle relational database. Some old data is stored in HDF5 files. These HDF5 files are stored as Binary Large Objects in the DB. This is kind of migration be a bad one.

Also to move MDSplus data into JtextBD is migration. And more likely ones are user but the data in their own db and want to move them to the cloud.

### Backup

Back is a kind of migration, two-way migration: Move to backup and recover from backup.

## Data Acquisition

Data acquisition seems outside the scope of this paper. But without proper support for the DAQ, how is there going to be a data producer?

The DAQ front end should work seamlessly with the data ingest APIs. Functions like backup in local storage and auto metadata generation (including provenance) is important.

## Data Visualization

Data visualization is a vast project. We never know how the scientist is going to visualize the data. Many will write their own program to do this. But if we want to promote JtextDbCloud in a certain community, we have to research on the visualization needs. We must provide basic tools so they can have a try without any huge effort.

Scope in MDSplus is a good example. It is a simple tool, just display waveform in MDSplus. But that is enough. The wave manipulation (zoom, pan, scale) is fast, setting can be saved. We can learn many from it. But currently, fusion data has become more diverse then just wave. And tide integration with the db allow fluent query and data management is also very important.

Also as mentioned before the data in the database has PDI which include repetition information, this can be used by visualization. But it should not be the only mean of visualize this data. Moreover, multiple data can be used in a single visualization widget.

To support more client and reduce development effort, web technology (e.g. HTML, JS) is consider the best choice for data visualization.

## References

1. NASA, *REFERENCE MODEL FOR AN OPEN ARCHIVAL INFORMATION SYSTEM (OAIS).* 2012.

2. P., M., *ITER Control Model Technical Specification.*

3. Huang, J., et al., *Scibox: Online Sharing of Scientific Data via the Cloud*, in *2014 Ieee 28th International Parallel and Distributed Processing Symposium*. 2014, Ieee: New York.