The keys to the management of time-evolution signals on HDF5 files in ITER

Rodrigo Castro [1], Yury Makushok [2], Lana Abadie [3], Jesús Vega [1]

[1]Laboratorio Nacional de Fusión, CIEMAT. Madrid. Spain
[2]MINSAIT - INDRA, Madrid, Spain
[3]ITER Organization, St Paul lez Durance, France

2025 HDF5 Workshop

This work was partially funded by the Spanish Ministry of Science and Innovation under the Project No. PID2022-137680OB-C31

iter china eu india japan korea russia usa

MINISTERIO DE EDUCACIÓN Y CIENCIA

Ciemat Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas

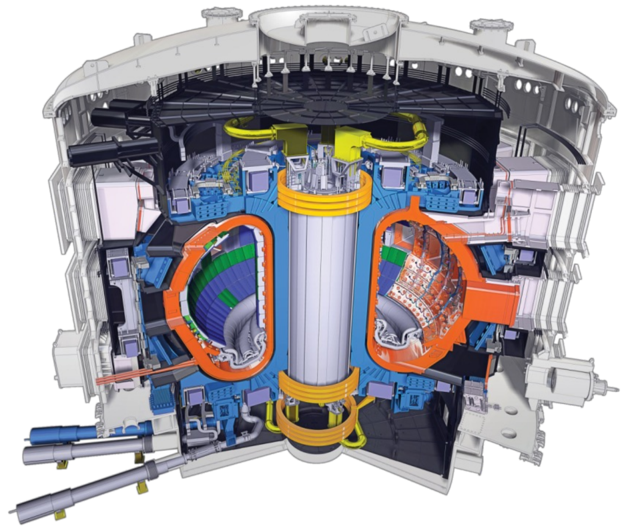
indra

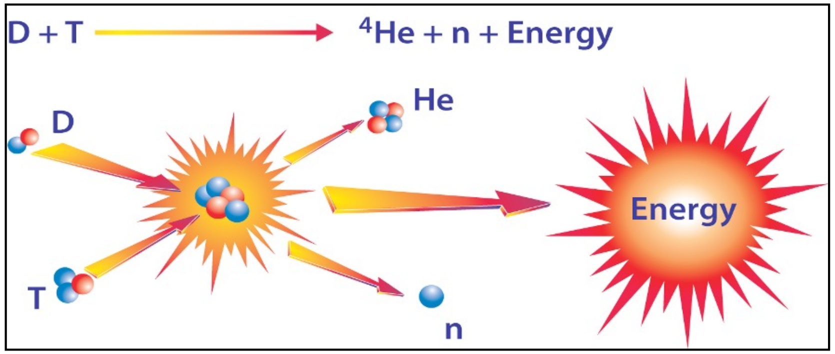
Index

- ❑ Introduction
- ❑ Fast Data Archiving
- ❑ Data indexing
- ❑ Real-time data processors
- ❑ Data access service

ITER machine

- ❑ Experimental fusion energy device
- ❑ Demonstrate feasibility of fusion energy
- ❑ Understand construction and operation challenges of a future real generator
- ❑ **Data handling system ready for continuous operation**





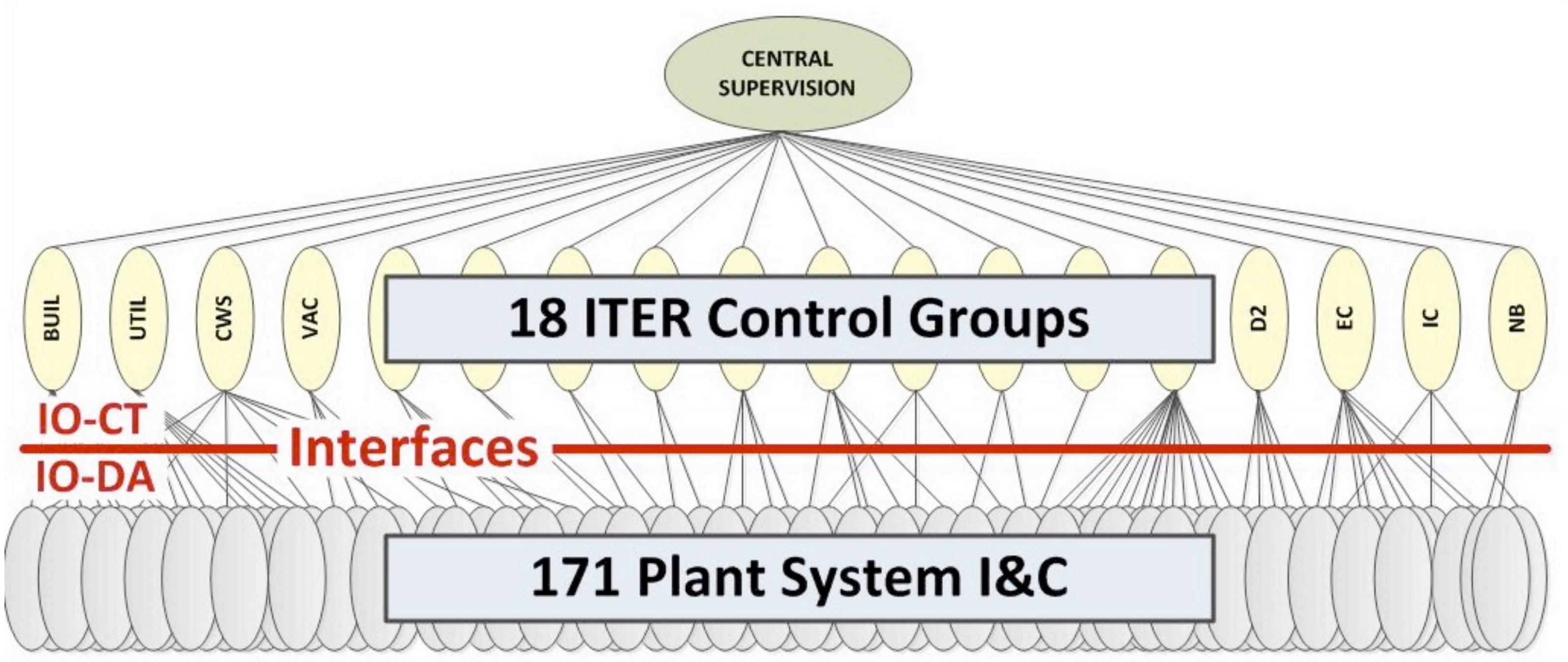
iter china eu india japan korea russia usa

MINISTERIO DE EDUCACIÓN Y CIENCIA

Ciemat Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas

indra

ITER control system organization



- Types of Data
  - DAN : Fast data acquisition, diagnostics and fast control (200K variables)
  - PON : Slow control variables, EPICS channelAccess and pvAccess (over 1M variables)
  - SDN : Real-time network (2000 variables)

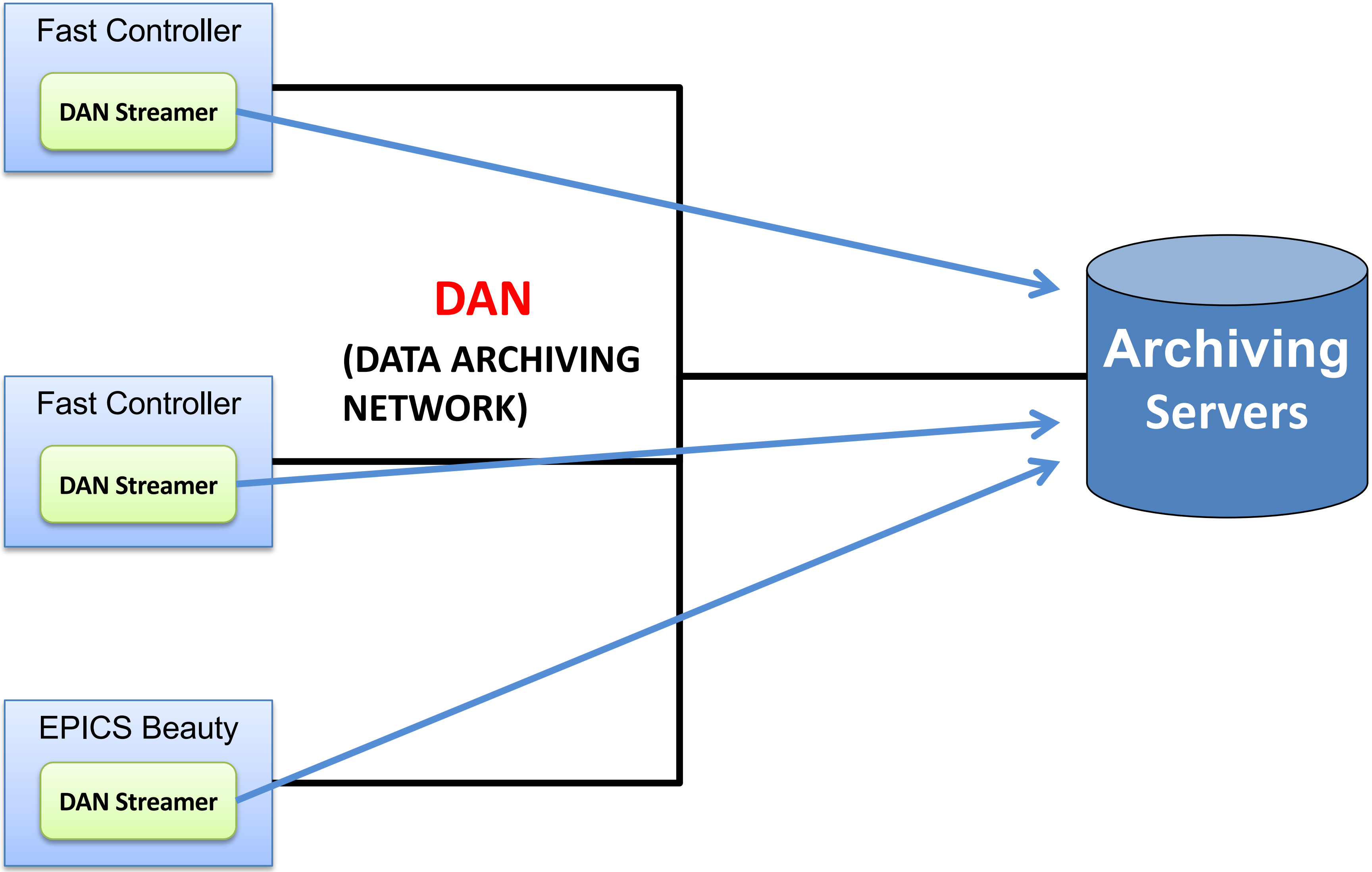
ITER data characteristics

- Very complex infrastructure
  - Huge number of variables and signals to measure
  - Different types of data
- Chaotic nature of plasma
  - Diagnostics include very high sampling rate DAQ systems
  - High data throughput

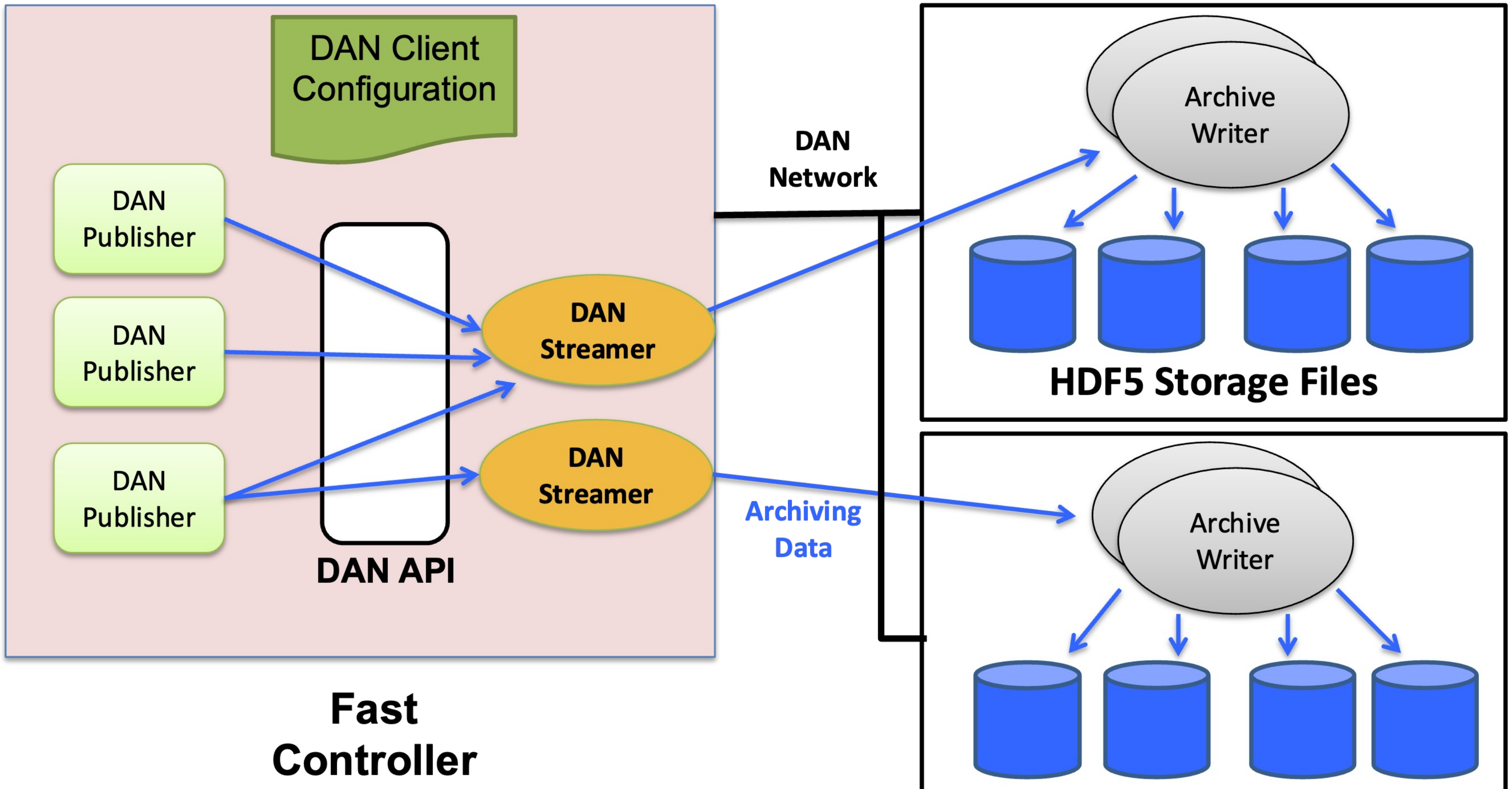
+

- **Long pulse / continuous operation**
  - **Time evolution signals**
  - **On fly change on data properties (sampling rate, dimensionality)**
  - **Real-time data archiving and handling**

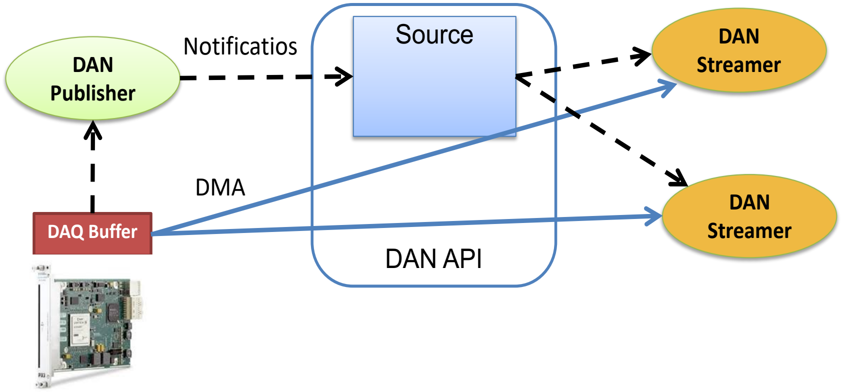
Fast DAQ Archiving



DAN Architecture

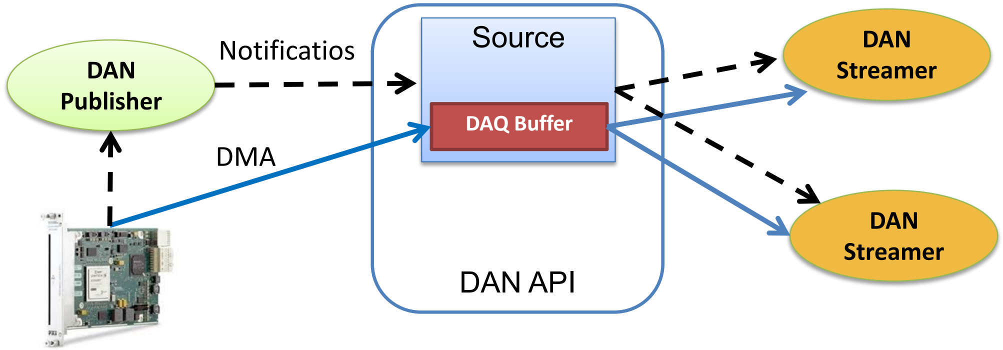


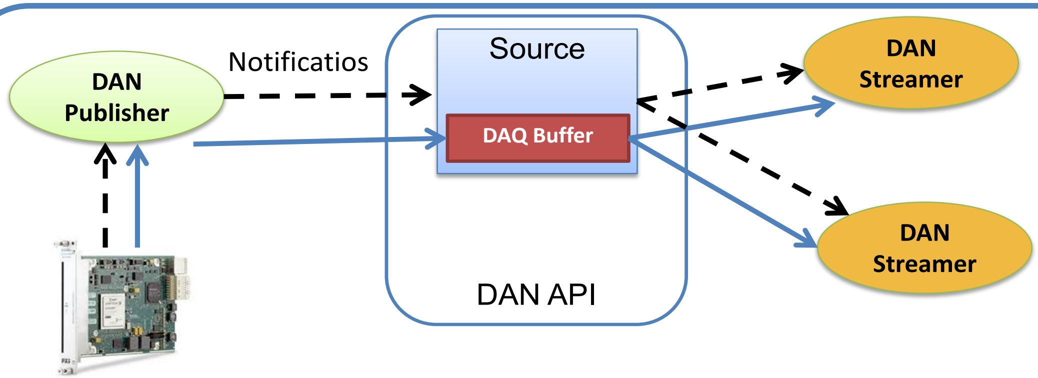
DAN API modes



Notifies when and where new data is available

Provides the buffer for DAQ data



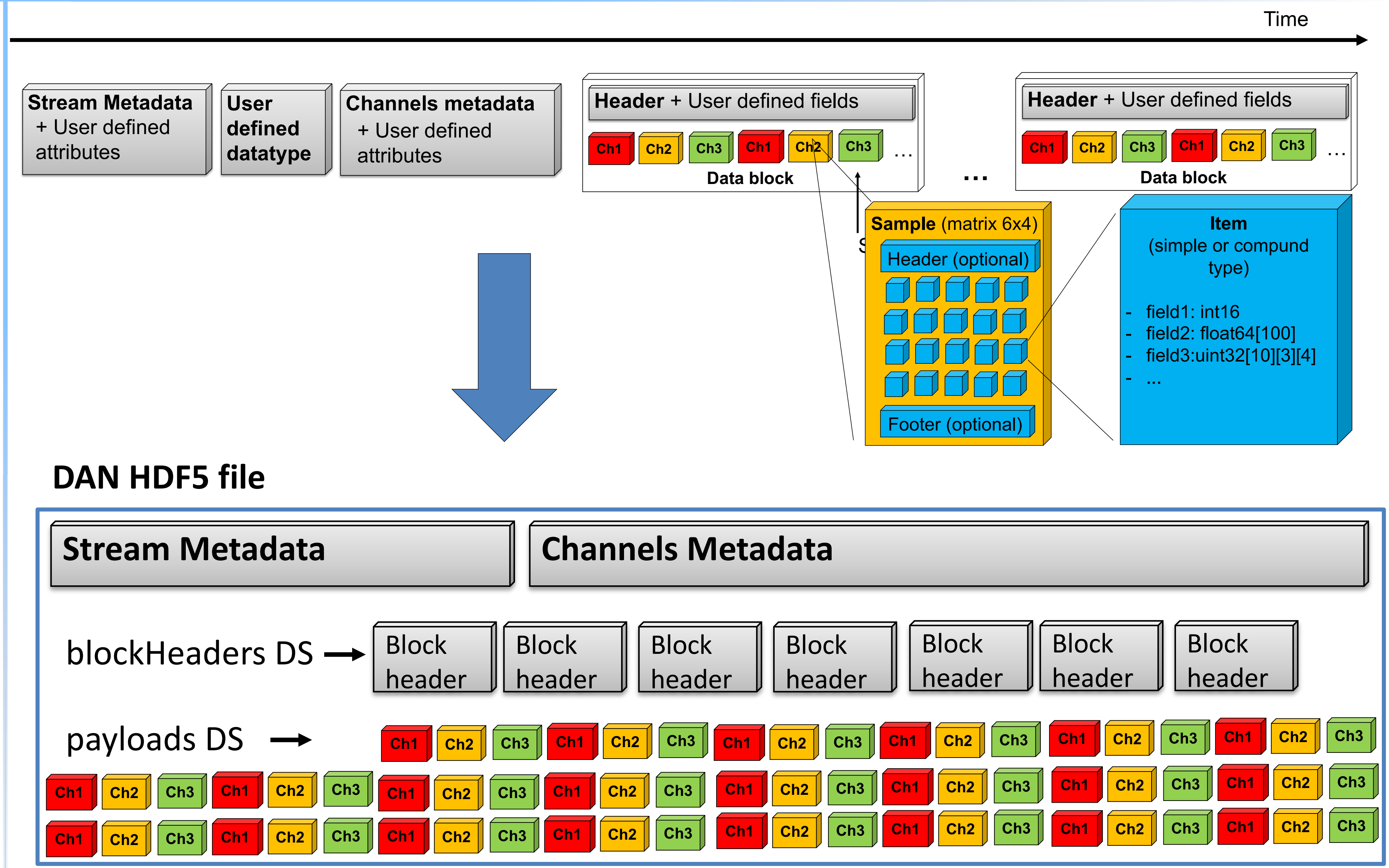


Provides everything

FAST Data archiving keys

- Fast data acquisition data (2KHz -> GHz)
  - DAQ speeds avoids one timestamp per sample
  - Multichannel DAQs
    - Support of multichannel interleaved data is required
- **Datablock as basic container**
  - Metadata (header)
  - Same sampling rate
  - Same dimensionality
  - One timestamp per datablock
  - Contains complete samples
- Different datablocks can have different sampling rate and different dimensionality

DAN protocol data model



FAST Data archiving keys

- Two datasets: datablock and payload
  - Speeds up data location
  - Flexibility in sampling rate and dimensionality along the DAQ

- Data block and payload must be archived as they are received
  - No time to reformat
  - Buffering is fundamental in case of non optimum datablock sizes
  - Small datablocks produces high CPU usage

FAST Data archiving keys

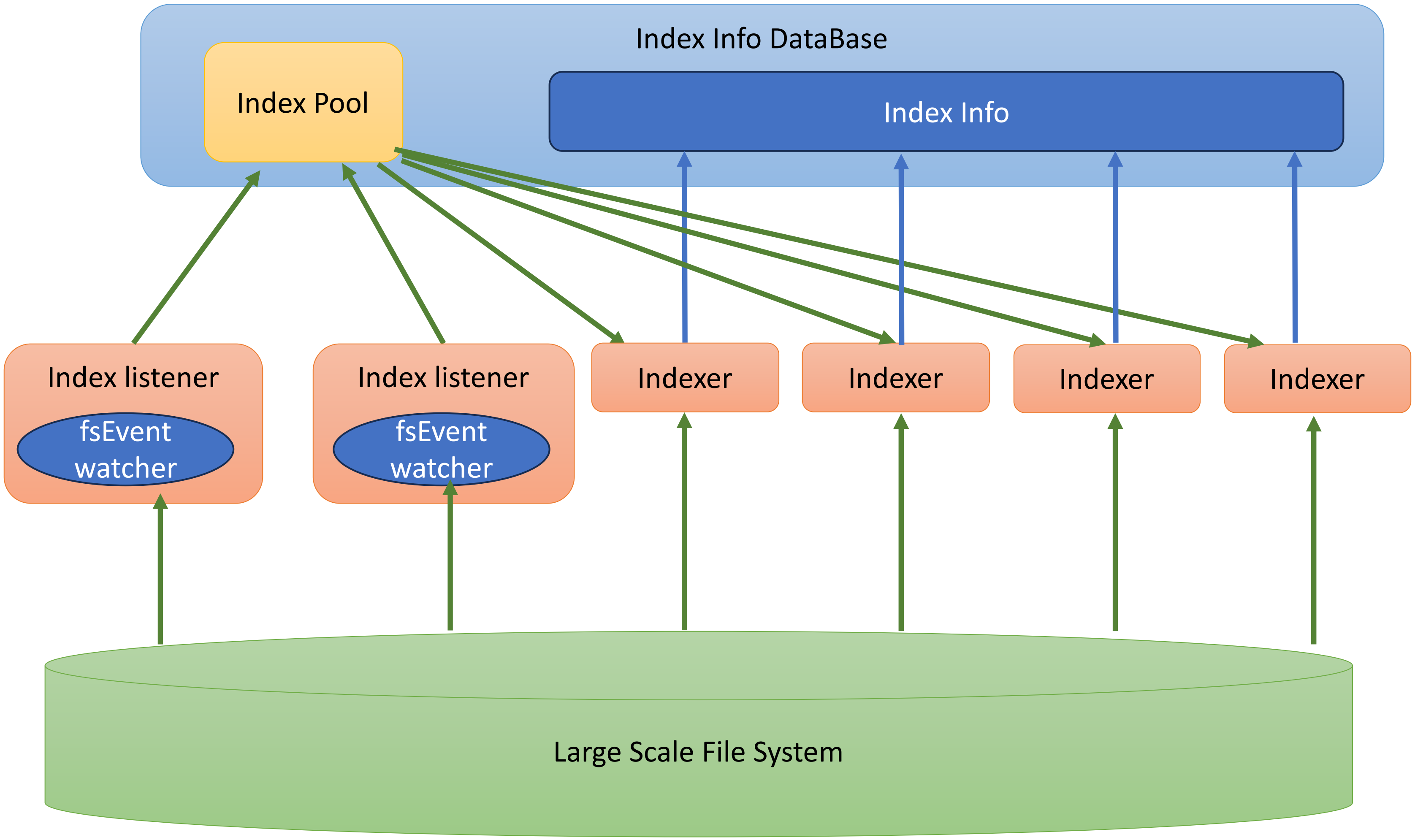
- Special cases: DAQ hardware that produces extremely coded data
  - HDF5 filters have been proved as a very useful solution
  - Readers watch regular datatypes

- Regular data (sampling rate and structure) speeds up data access
  - Definition of time sections and structure sections along the file
  - Fast data access mechanism for same section data

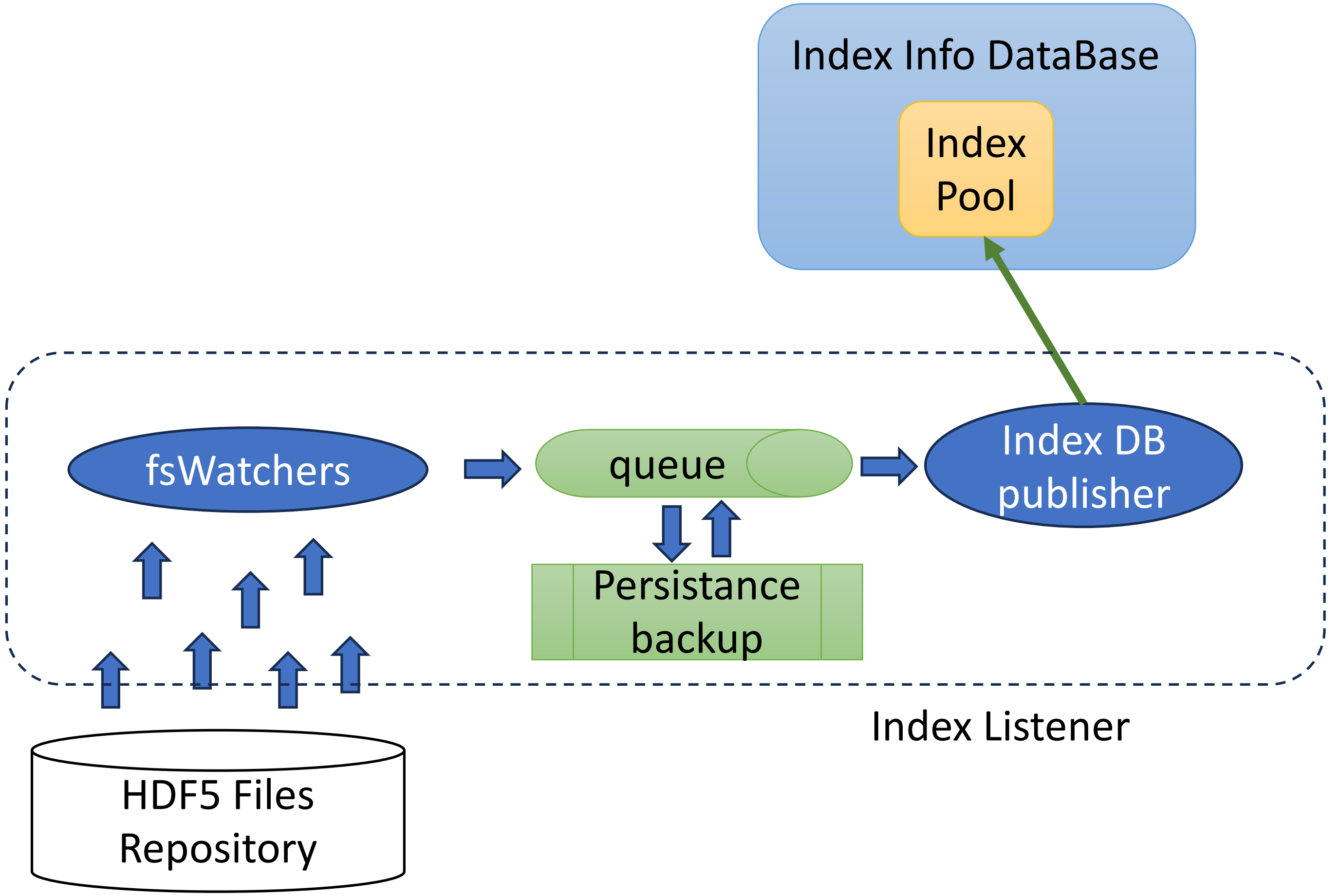
Data Indexing

- First level catalogue of archived data
- Index information is only got from archiving files
  - Variables
  - Storage location
  - Timestamps
  - File state
  - Sampling rate
  - Data quality
- Responsible of
  - Detection of data storage events
  - Index data extraction from archiving files
  - Index data injection into Index Info Database

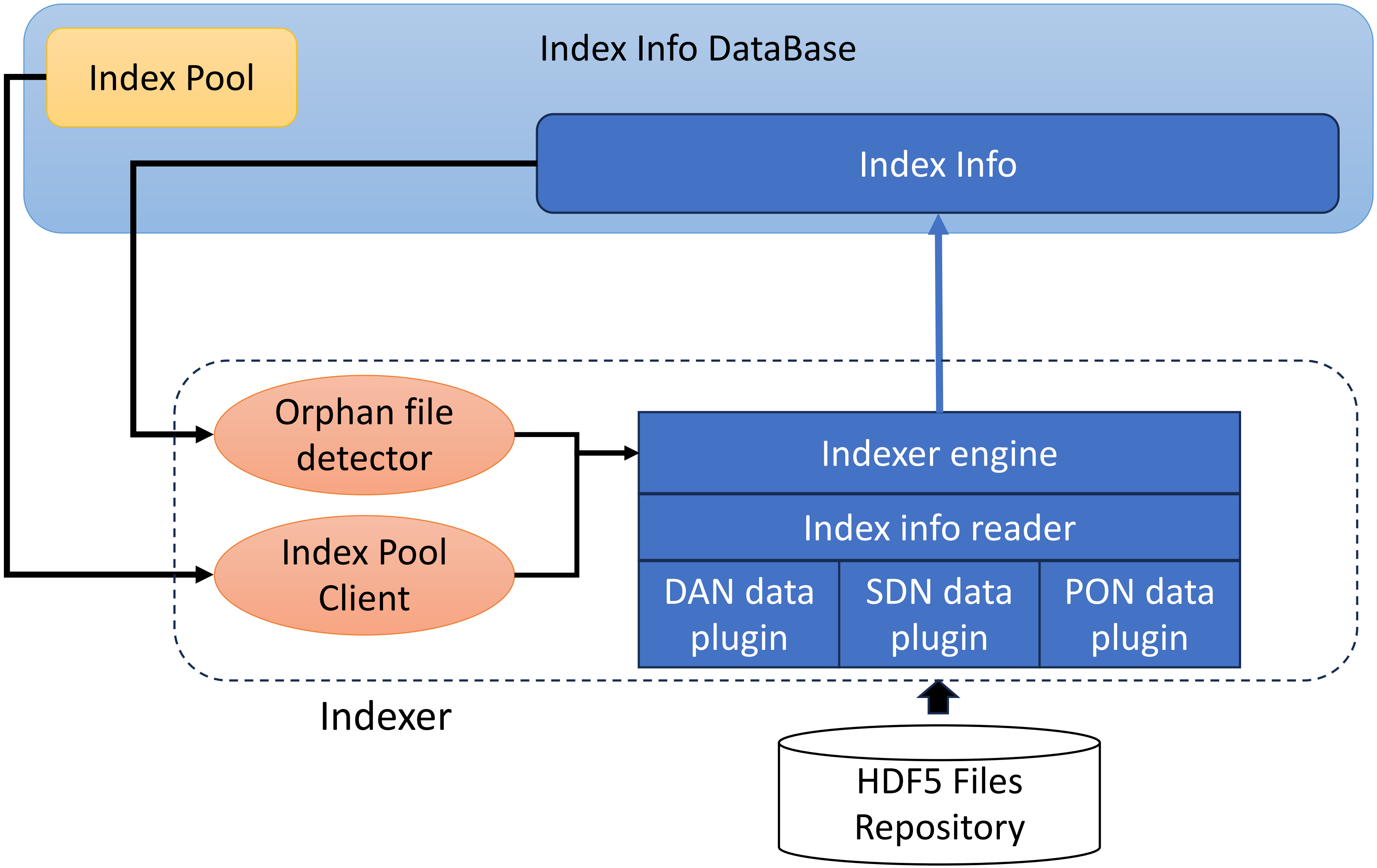
Real-time indexing system



Index listener



Indexer


Index Info DataBase
Index Pool
Index Info
Orphan file detector
Index Pool Client
Indexer engine
Index info reader
DAN data plugin
SDN data plugin
PON data plugin
Indexer
HDF5 Files Repository

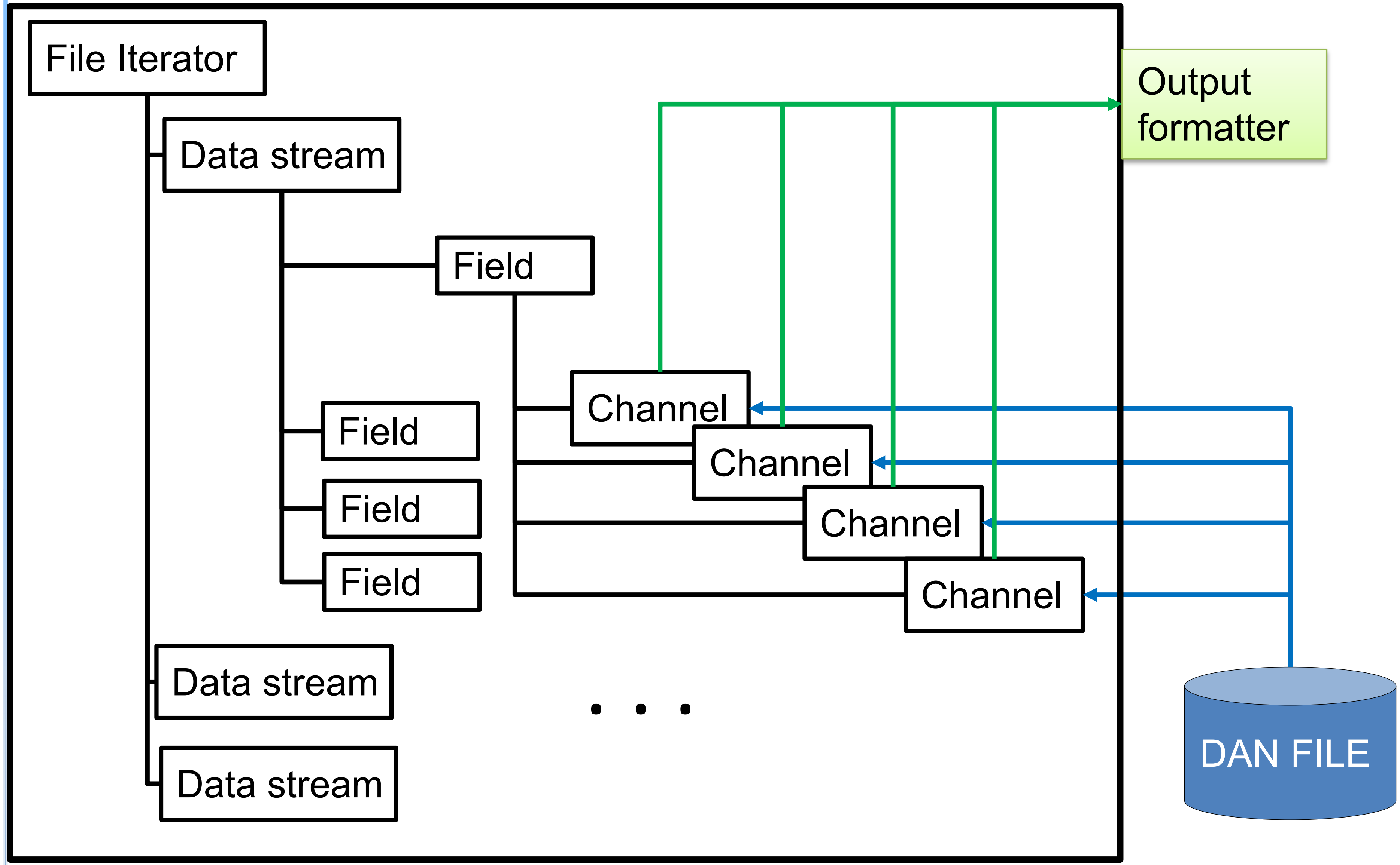
Real-time indexing keys

- Critical system for long pulse and continuous data acquisition
- File system events is a good notification mechanism
  - More robust notification mechanism will help
- Very good reaction policies for non ready files
- Good management of wrong data files
- Good management of orphan open files (not correctly closed)

Real-time processing

- Process archived data in real-time
- Follows the same high scalable model of indexer
  - Multiple instances in parallel
- Implements a generic architecture for real-time data processing developments
- Supports calibrated data

Processor architecture



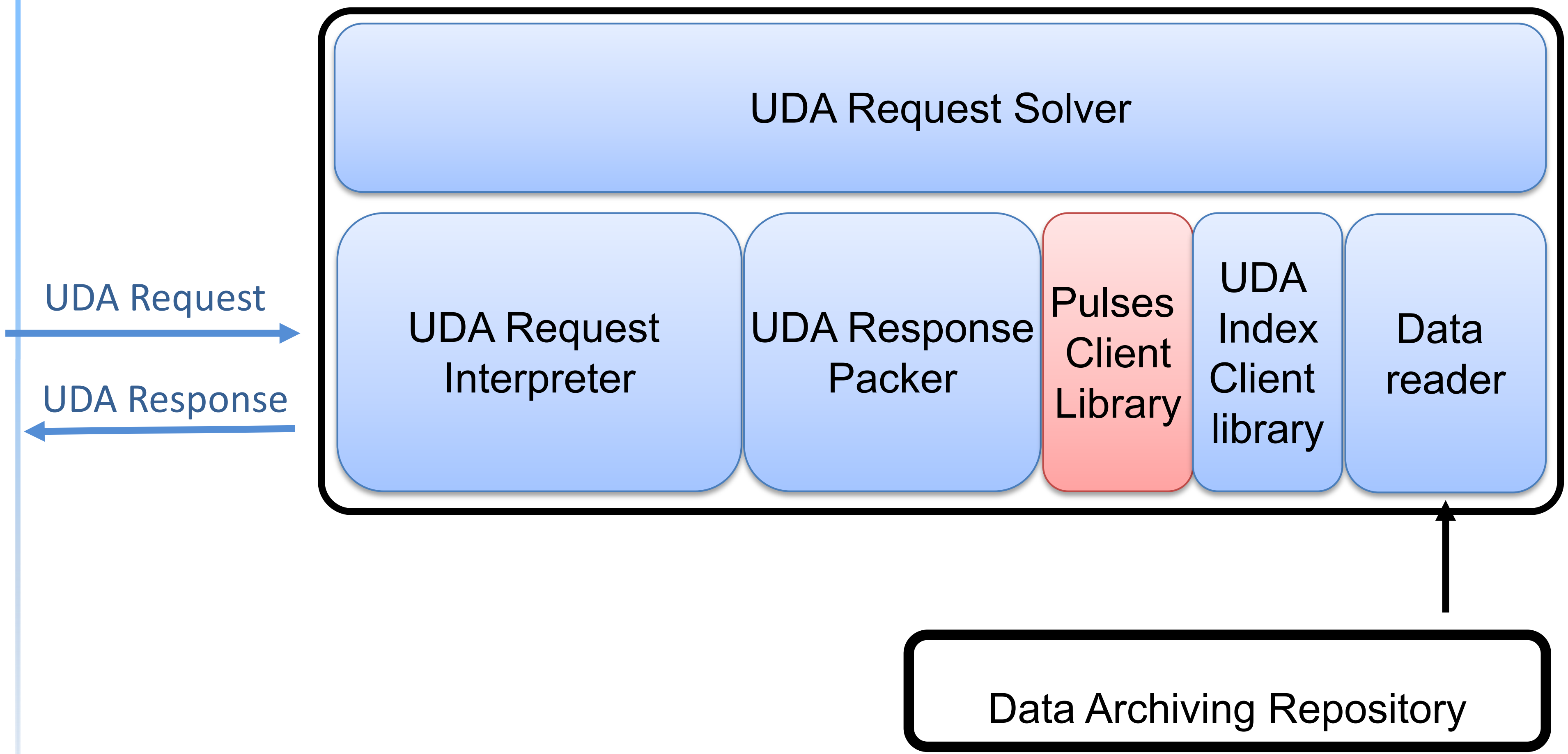
Current real-time processors

- Replicator
  - Replicates archived files in remote locations
  - A direct copy is not valid
  - HDF5 model must be coherent in replica file
- Donwsampler
  - Decimation is a critical operation for big data access
  - It is an expensive operation to be done on fly
  - The objective is to produce look ahead decimated data
  - Files with different decimation levels

Data Access Service

- Support of all ITER data types
- Request based on **timestamp interval** or pulse number
- Metadata and payload
- Multidimensional data and complex data types
- Data chunking
- Supports load-balancers

UDA server architecture



Index database request

- Key service for efficient and robust data access
- Detection of the best files to cover the requested variable and Interval
  - Location of involved files
  - Selection of files that best fit requested data resolution and nature
  - Management of files overlap
  - Integration of already open files

UDA Request Syntax

- **UDA::getData(**<variable>, <interval>, <optional_parameters>**)**
- <variable>
  - **variable=/ITER/**<variable_name>/<field>
- <interval>
  - **startTime=**<time>**, endTime=**<time>
  - **refTime=**<time>**, startTime=**<relTime>**, endTime=**<relTime>
  - **shot=**<pulse>
  - **shot=**<pulse> **, startTime=**<relTime> **, endTime=**<relTime>
- <optional_parameters>
  - Return timestamps and data type
  - Decimation
  - Chunking parameters
  - Calibration

UDA Request Syntax

- UDA::getData("**variable**=/ITER/MAG-CYSI-FCT0:sensor1D-1[4], **startTime**=1484264684735399104 , **endTime**= 1484264714735399104, **decSamples**=10000, **retType**=double")

- UDA::getData("**variable**=/ITER/MAG-CYSI-FCT1:event3/conv1, **pulse**=3542121, **startTime**=10S , **endTime**= 500S, **retType**=long")

UDA Request

- It has demonstrated to be a very powerful mechanism for time evolution data requests
- It is able to implement complex time-evolution data requests
- Intuitive syntax
- Extensible

Next elements

- Data sanity check improvement
- File versions support
  - Zero copy of original data (if it is posible)
  - Use of HDF5 external links
- Improved data access protocol for long intervals
- Future advanced massive data access mechanisms
  - Automatic detection of POIs
  - Advanced decimation mechanisms
  - Automatic patterns search

Thank you

Rodrigo Castro [1], Yury Makushok [2], Lana Abadie [3]

[1]Laboratorio Nacional de Fusión, CIEMAT. Madrid. Spain
[2]MINSAIT - INDRA, Madrid, Spain
[3]ITER Organization, St Paul lez Durance, France

2025 HDF5 Workshop

This work was partially funded by the Spanish Ministry of Science and Innovation under the Project No. PID2022-137680OB-C31

iter china eu india japan korea russia usa

MINISTERIO DE EDUCACIÓN Y CIENCIA
Ciemat Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas

indra