# SEVEN KEYS FOR PRACTICAL UNDERSTANDING AND USE OF CGNS

Marc Poinot, SAFRAN
Christopher Rumsey, NASA LaRC

AIAA SciTech 2018

# Motivation

CGNS is well known in the CFD community, but:
- often mismatched with its mid-level library
- only understood as mesh storage on disk

We have seen some initiatives to develop a new standard to:
- define an implementation-independent data model
- add Computer-Aided Design (CAD) related features
- implement in HDF5

➔ This looks quite close to CGNS

Common comments:

CGNS is a library
CGNS cannot fit my own data structure
There are too many ways to describe the same feature in CGNS
CGNS cannot handle parallel processing
CGNS is only for archival data
CGNS files are too big
The recommended implementation is HDF5 but only ADF is available
There are few useful tools for CGNS manipulation
I already have HDF5 and I do not need CGNS
CGNS inefficiently stores time data and statistics

We propose to gather initiatives rather than creating entropy: let us refresh our CGNS comprehension…

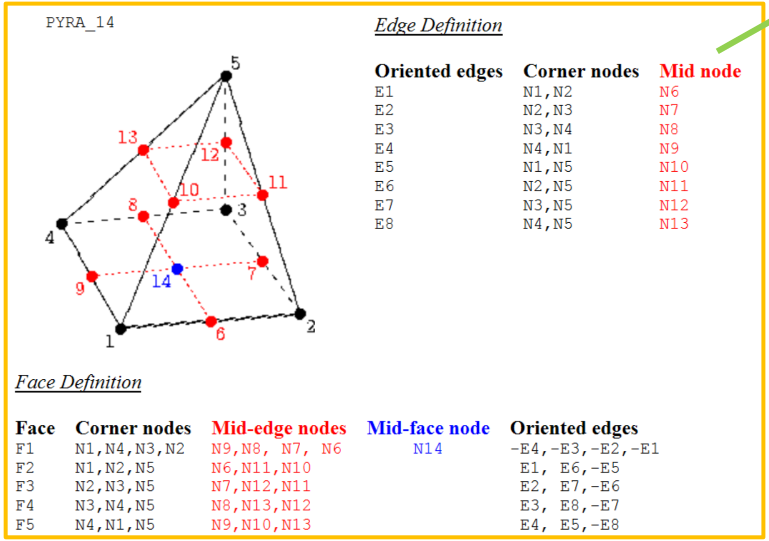
# CGNS entities and their acronyms

| Acronym | Meaning | Entity |
|---|---|---|
| CGNS | CFD General Notation System | Identifier |
| CGNS/SC | Steering Committee | Group of people |
| CGNS/SIDS | Standard Interface Data Structure | Textual document |
| CGNS/CPEX | CGNS Proposal for EXtension | Procedure and textual document |
| CGNS/FMM | File Mapping Manual | Textual document |
| CGNS/HDF5 | HDF5 implementation | Textual document |
| CGNS/Python | Python implementation | Textual document |
| CGNS/MLL | Mid-Level Library | Software library |

# Seven keys

1. Abstract model
   - What is the data I want to define?
   - How do I define it?
2. Interoperability
   - How to go from CAD to mesh, to post-processing, visualisation and then back to CAD?
   - How to exchange data during CFD/CSM, CFD/CAA or any multiphysics workflows?
3. Functional coverage
   - Do I have enough functions to describe my data?
   - What is the remaining volume data I need to define/store?
4. High performance
   - Is there a High Performance Computing (HPC) efficient representation for my data model?
   - Who is going to support it?
5. Versatile implementation
   - Can I manipulate my data in a flexible way?
   - Do I have to re-invent all the functions I need for my data processing?
6. Extensibility
   - How can I describe out-of-CFD data, such as chemistry, high order, CSM features, new CFD trends?
   - Does the standard belong to NASA?
7. Open System
   - Can I connect the simulation elements together as black boxes?
   - Is there a mandatory implementation?

# Abstract model



*Edge Definition*

| Oriented edges | Corner nodes | Mid node |
|---|---|---|
| E1 | N1,N2 | N6 |
| E2 | N2,N3 | N7 |
| E3 | N3,N4 | N8 |
| E4 | N4,N1 | N9 |
| E5 | N1,N5 | N10 |
| E6 | N2,N5 | N11 |
| E7 | N3,N5 | N12 |
| E8 | N4,N5 | N13 |

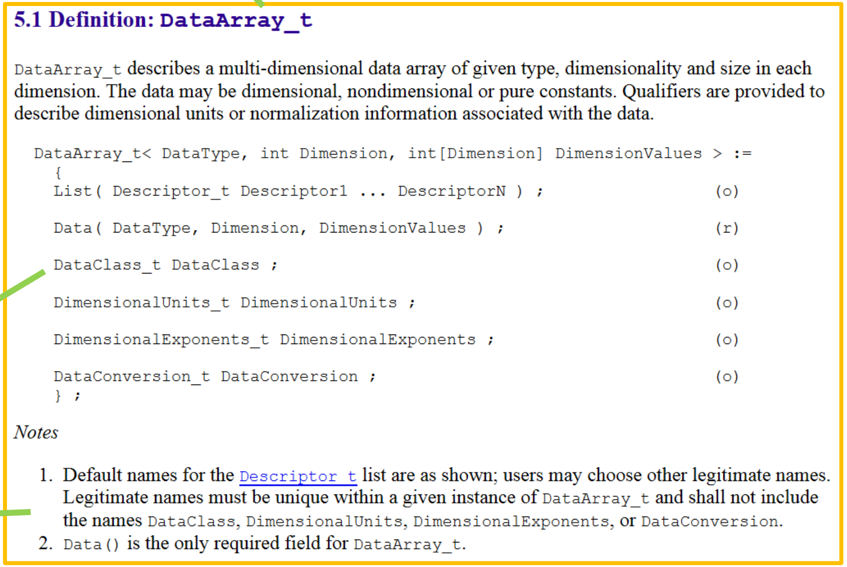
*Face Definition*

| Face | Corner nodes | Mid-edge nodes | Mid-face node | Oriented edges |
|---|---|---|---|---|
| F1 | N1,N4,N3,N2 | N9,N8, N7, N6 | N14 | -E4,-E3,-E2,-E1 |
| F2 | N1,N2,N5 | N6,N11,N10 | | E1, E6,-E5 |
| F3 | N2,N3,N5 | N7,N12,N11 | | E2, E7,-E6 |
| F4 | N3,N4,N5 | N8,N13,N12 | | E3, E8,-E7 |
| F5 | N4,N1,N5 | N9,N10,N13 | | E4, E5,-E8 |

Types and Indexing conventions

SIDS type definitions

A model is what you want to specify but not how you want to implement it



### 5.1 Definition: DataArray_t

DataArray_t describes a multi-dimensional data array of given type, dimensionality and size in each dimension. The data may be dimensional, nondimensional or pure constants. Qualifiers are provided to describe dimensional units or normalization information associated with the data.

```
DataArray_t< DataType, int Dimension, int[Dimension] DimensionValues > :=
  {
  List( Descriptor_t Descriptor1 ... DescriptorN ) ;                 (o)

  Data( DataType, Dimension, DimensionValues ) ;                     (r)

  DataClass_t DataClass ;                                            (o)

  DimensionalUnits_t DimensionalUnits ;                              (o)

  DimensionalExponents_t DimensionalExponents ;                      (o)

  DataConversion_t DataConversion ;                                  (o)
  } ;
```

*Notes*

1. Default names for the Descriptor_t list are as shown; users may choose other legitimate names. Legitimate names must be unique within a given instance of DataArray_t and shall not include the names DataClass, DimensionalUnits, DimensionalExponents, or DataConversion.
2. Data() is the only required field for DataArray_t.

Formal definition, supported using a grammar

Informal definition, textual non-exhaustive description

# Interoperability

The highest level of interoperability is the user's meaning
The lowest is the implementation

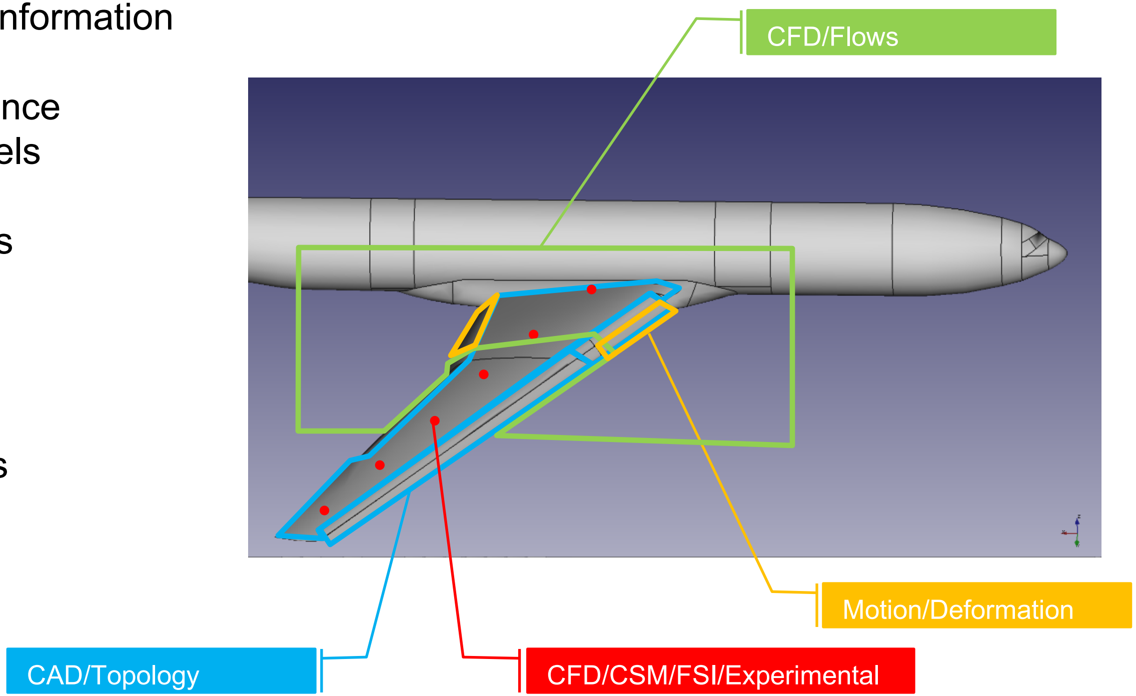
Families are the means to hold user's information

- Topology entities, CAD reference
- Same entity for different models
- Entities with special features
- Simulation specialized entities

Simultaneous families allowed
Hierarchy of families

➔Insure traceability through processes

➔ Allows high level interoperability

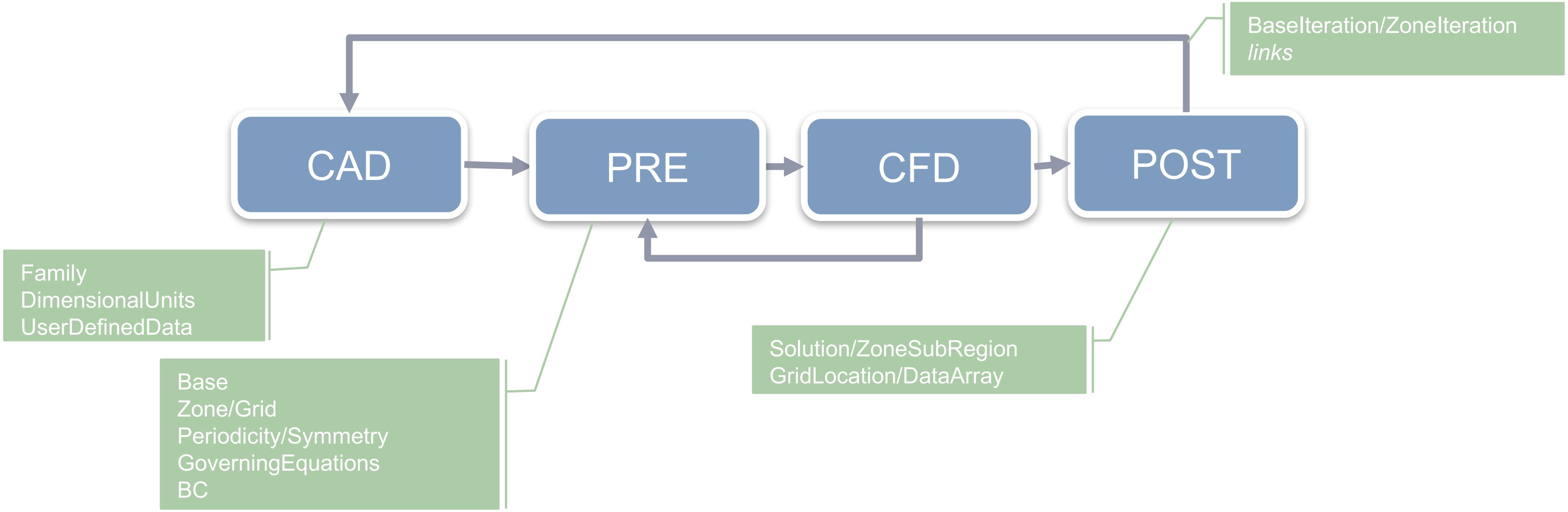


# Functional coverage

99% of the volume of the data already is defined

## The CGNS tree changes all along the simulation workflow

- It covers our CFD needs and more
- It doesn't cover everything, but it could eventually fulfill any needs: CPEX



CAD

PRE

CFD

POST

BaseIteration/ZoneIteration
*links*

Family
DimensionalUnits
UserDefinedData

Base
Zone/Grid
Periodicity/Symmetry
GoverningEquations
BC

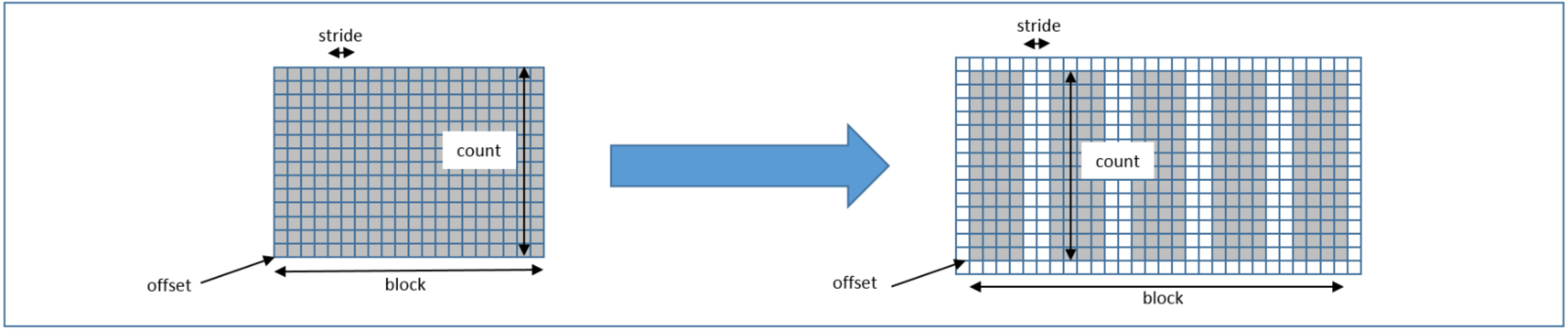
Solution/ZoneSubRegion
GridLocation/DataArray

# High performance

## HPC implementation

- HDF5
  - HDF Group support for portage/tuning
  - Very large data but limited number of types
  - Fast memory/disk swap
  - Smart memory mapping
  - Parallel features

- The HDF5 implementation specification is public
  - Write your own application
  - Specific node orientation
  - Specific memory mapping (interlaced, non-contiguous, partial, out-of-core…)



# Versatile implementation

## Gluing/prototyping the easy way

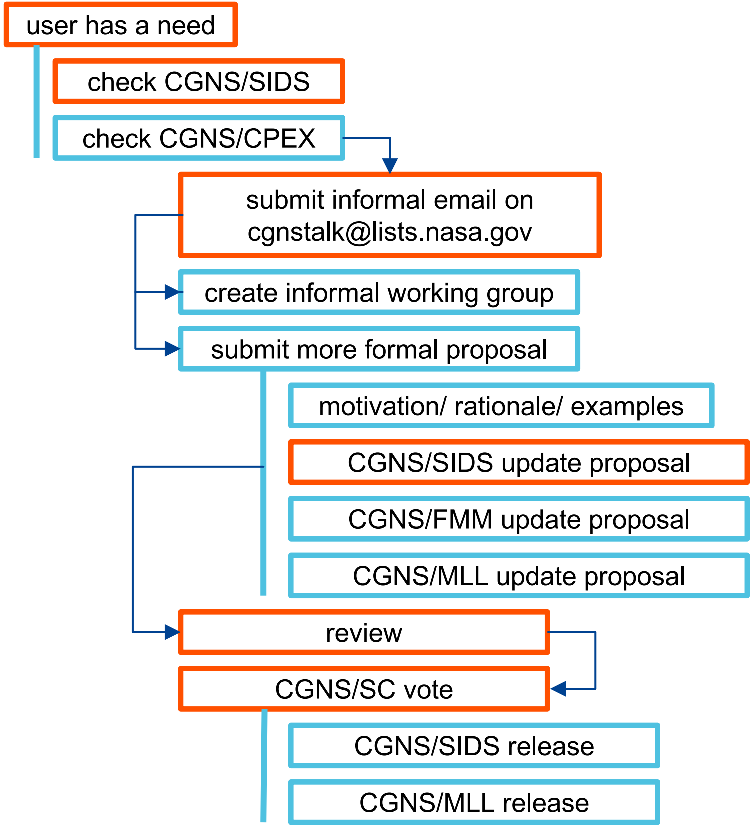
- Python/numpy
  - Your C/Fortran array IS your python array
  - Reuse of the very large python libraries
  - No-class paradigm, easy serialization/deserialization
- Memory to memory exchange (same process, inter-process, inter-nodes, inter-hosts…)
- Fast creation/modification of CGNS trees

```
import CGNS.MAP as CGM
import CGNS.PAT.cgnsutils as CGU
import CGNS.PAT.cgnskeywords as CGK
import CGNS.PAT.cgnslib as CGL

(T,L,P)=CGM.load('HL-CRM.cgns')

families=set()
pathlist=CGU.getALlNodesByTypeOrNameList([CGK.CGNSTree_ts,'Base',CGK.Zone_ts,CGK.ZoneBC_s,CGK.BC_ts])

for path in pathlist:
  bctype=node[1]
  family=bctype[2:]
  node=CGU.setStringByPath(T,path,CGK.FamilySpecified_s)
  CGL.newFamilyName(node,family)
  families.add(family)

base=CGU.getNodeByPath(T,'/Base')
for family in families:
  fam=CGL.newFamily(base,family)
  CGL.newFamilyBC(fam,'BC'+family)

CGM.save('HL-CRM.cgns',T)
```

Change all BCs to family defined

# Extensibility

- **A support for user needs**
  - CPEX
    - Propose your own extension
    - Do not break already used features
  - Examples
    - NGON HPC redefinition
    - Arbitrary High order
    - Enhanced Chemical data model
    - Arbitrary Reference Frame
  - PROS
    - Arguing often leads to better ideas
    - Once adopted, everyone uses your data model / implementation
  - CONS
    - Lengthy process
    - You have to explain your ideas



user has a need

check CGNS/SIDS

check CGNS/CPEX

submit informal email on cgnstalk@lists.nasa.gov

create informal working group

submit more formal proposal

motivation/ rationale/ examples

CGNS/SIDS update proposal

CGNS/FMM update proposal

CGNS/MLL update proposal

review

CGNS/SC vote

CGNS/SIDS release

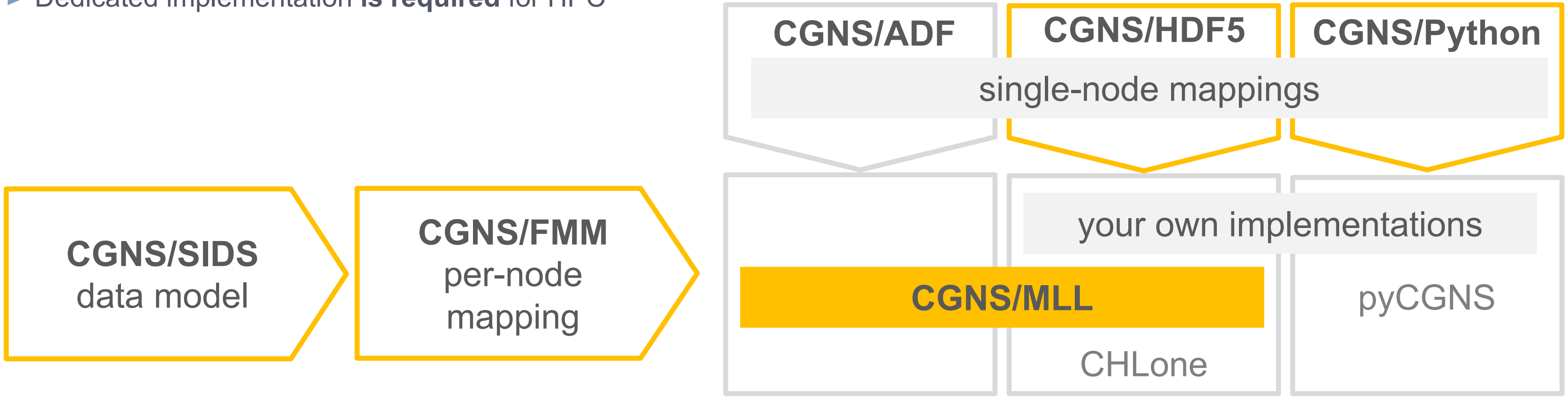
CGNS/MLL release

# Open system

## Specification & connection only through public interfaces

- High Level: CGNS/SIDS
- Low Level: CGNS/FMM
- Implementation: CGNS/HDF5, CGNS/Python

- Any compliant implementation can 'connect' to others
- Dedicated implementation **is required** for HPC



# Answers to remarks

- ☒ CGNS is a library
- ☒ CGNS cannot fit my own data structure
- ☒ XML is the way to describe data
- ☑ **Extension process is too long for me**
- ☑ **There are too many ways to describe the same feature in CGNS**
- ☒ CGNS cannot handle parallel processing
- ☒ HPC cannot waste time on data model
- ☒ CGNS is only for archival data
- ☑ **CGNS files are too big**
- ☒ The recommended implementation is HDF5 but only ADF is available
- ☑ **There are few useful tools for CGNS manipulation**
- ☒ I already have HDF5 and I do not need CGNS
- ☑ **CGNS inefficiently stores time data and statistics**

- □ CGNS is only for CFD

# Conclusion

## You now have a broader view of CGNS

## CGNS is far more than a library ➔ it is a means for CFD workflow interoperability

Do you want to add CAD traceability?
Do you have such a standard for CSM, FSI, CAA, or other field?
Do you want more data structure, more dedicated implementations?

**Join CGNS**

www.cgns.org
cgnstalk@lists.nasa.gov

CGNS