Implementing HDF5 in MATLAB

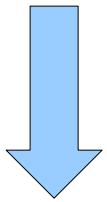
Jeff Mather & Alec Rogers

The MathWorks, Inc.

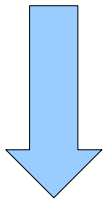
29 November 2006

HDF4

1-1 mapping of C API first. (1998)



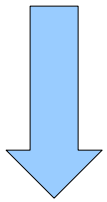
Customer requests for high-level functions.



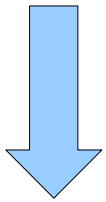
HDFREAD, HDFWRITE, HDFINFO. (2000)

HDF5

High-level first. (2003)

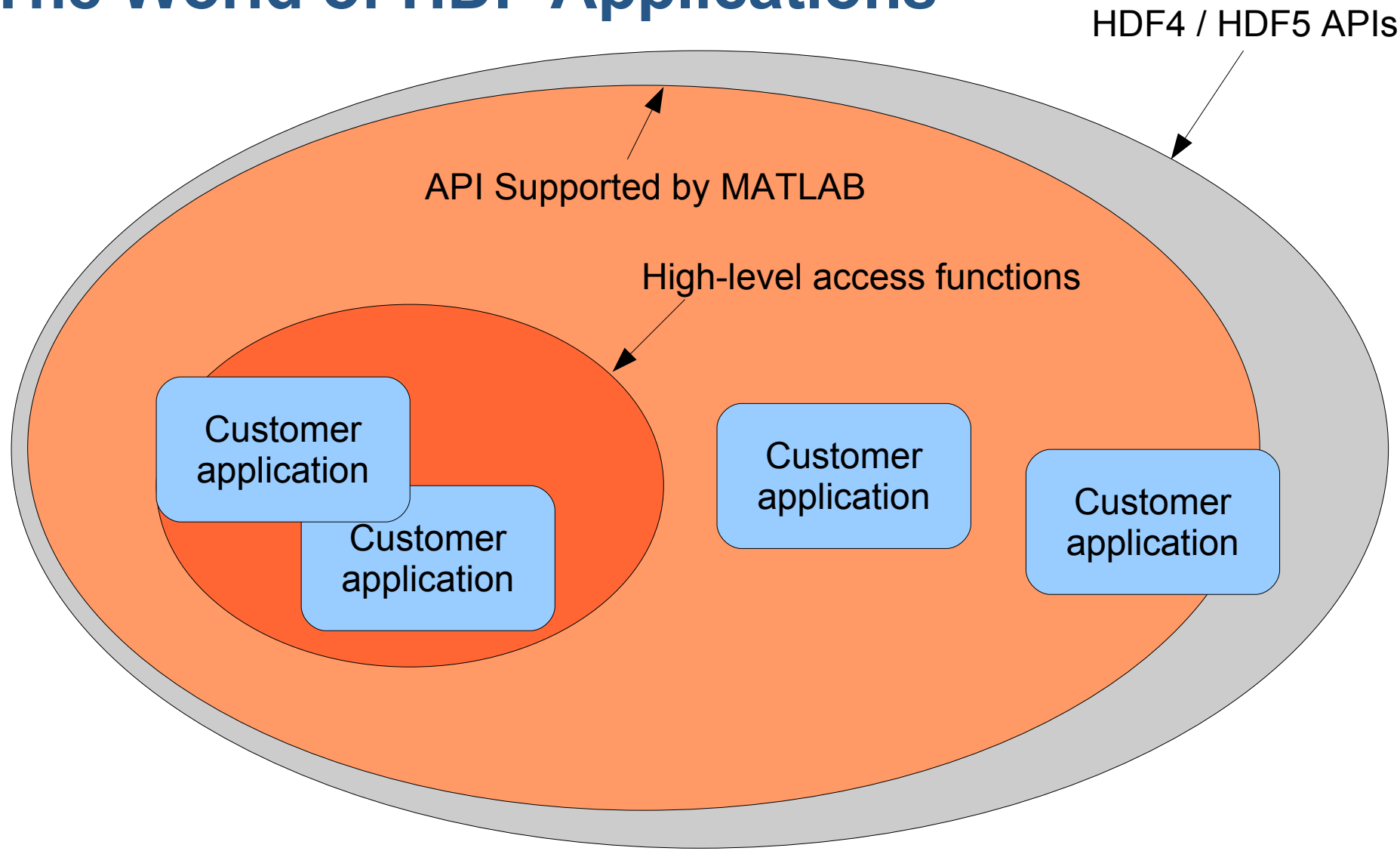


Customer requests for lower-level functionality.



1-1 mapping of C API. (2006)

The World of HDF Applications



HDF5READ

`DATA = HDF5READ(FILENAME,DATASETNAME)` returns in the variable `DATA` all data from the file `FILENAME` for the data set named `DATASETNAME`.

`DATA` has to be extremely general because of the wide variety of datatypes that HDF5 accomodates.

Simple access only:
- No subsetting.
- Limited datatype control.

More control needed to match the uniqueness of customer datasets and files.

HDF5INFO

`FILEINFO = HDF5INFO(FILENAME)` returns a structure whose fields contain information about the contents of an HDF5 file. `FILENAME` is a string that specifies the name of the HDF file.

HDF5WRITE

`HDF5WRITE(FILENAME, LOCATION, DATASET)` adds the data in `DATASET` to the HDF5 file named `FILENAME`. `LOCATION` defines where to write the `DATASET` in the file and resembles a Unix-style path. The data in `DATASET` is mapped to HDF5 datatypes using the rules below. . .

HDF5WRITE is completely symmetric with HDF5READ.

→ Objects disambiguate datatypes.

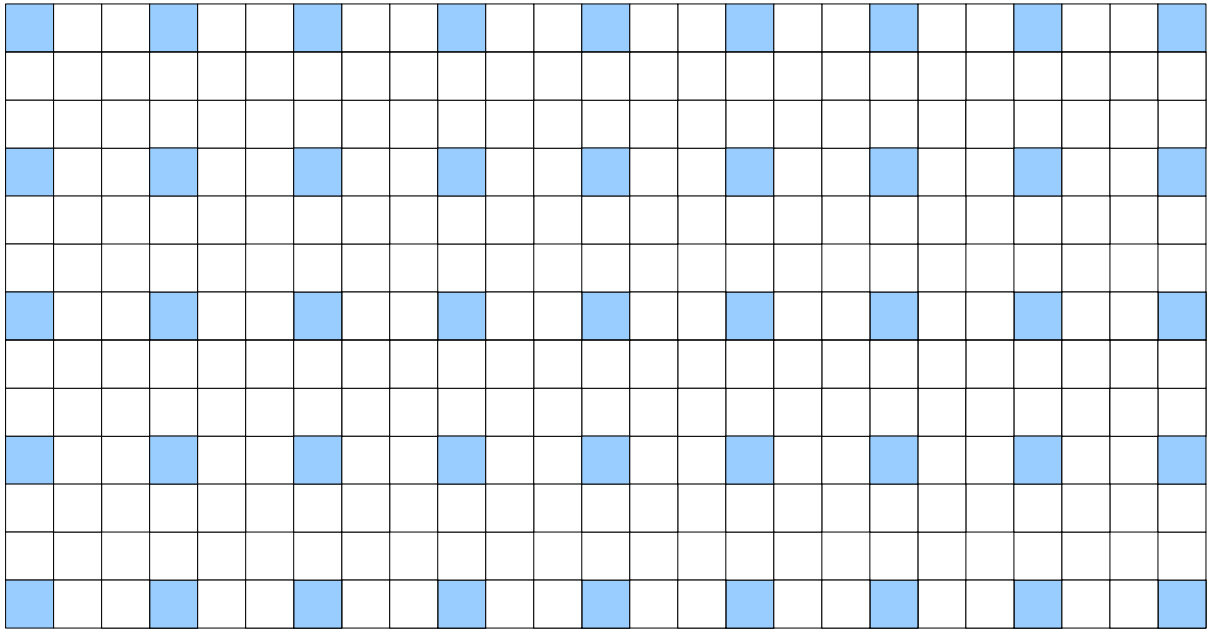
↓ The values in `DATASET` are cumbersome for non-native MATLAB types (e.g., arrays, compound, and references).

Customer HDF5 Requests

- Library upgrades (1.4.5, 1.6.4, 1.6.5, 1.8)
- Better support for large data
- Hyperselection, chunking
- New platform support (Solaris 64, MacIntel)
- GZIP, SZIP compression
- HDF5 file interrogation
- Bitfield, date/time datatypes
- Data translators: HDF5 --> MATLAB

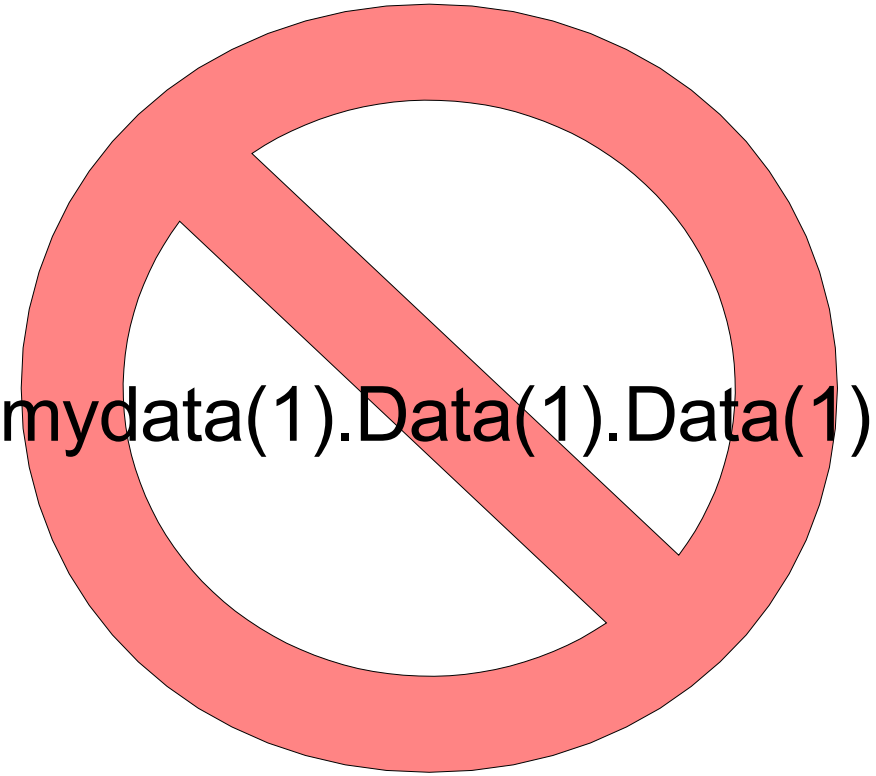
Use Cases

Read parts of an HDF5 dataset (a hyperslab).



Use Cases

Read complicated datatypes without the overhead of MATLAB objects for datasets.



Use Cases

Allow users to extend our HDF5 functionality without waiting for us.

Use Cases

Be able to drop in new versions of the HDF5 library when they become available.

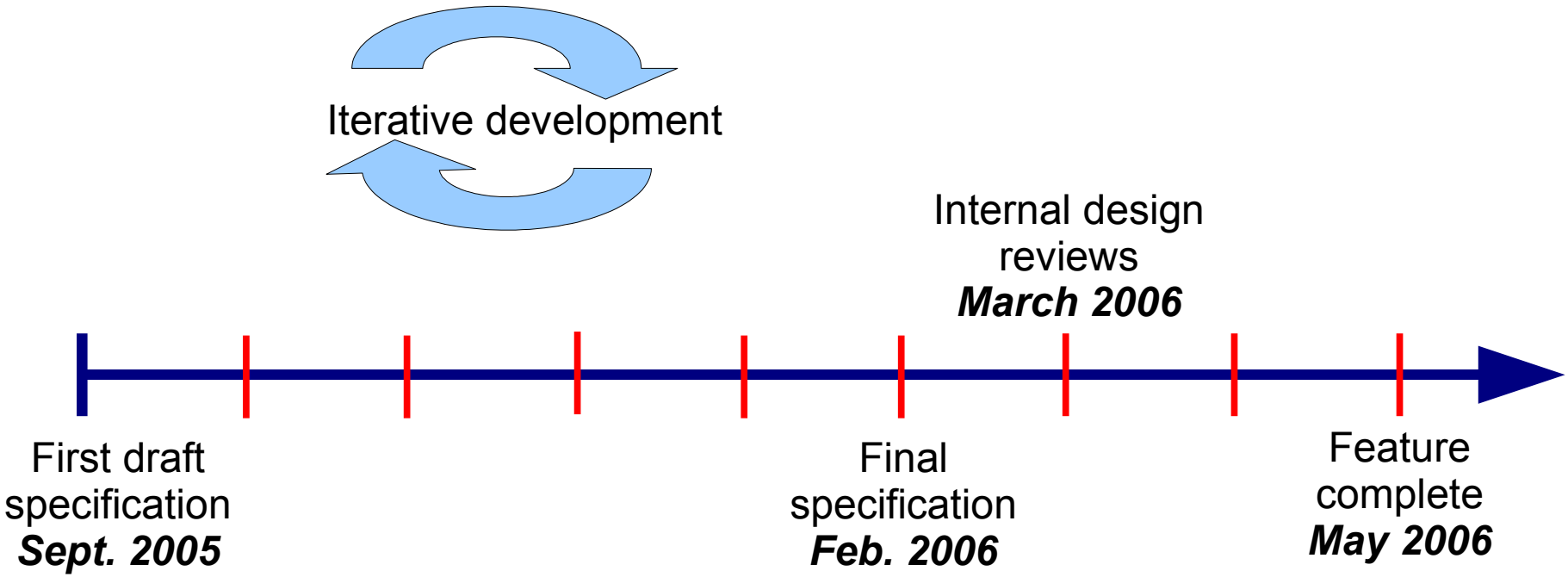
HDF5 1.8

Use Cases

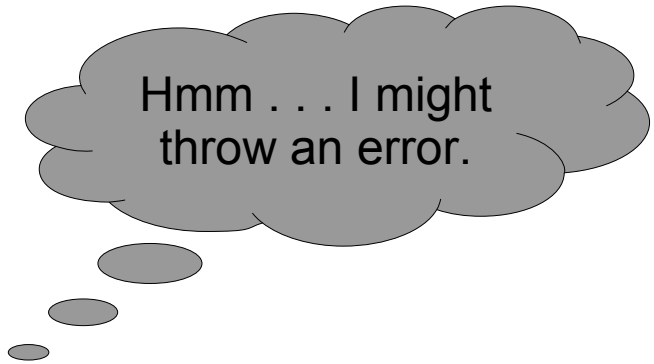
Use a variety of esoteric HDF5 features at once:

> “I'm trying to use HDF5 files [with] grouping features like compound data types, group links, and reference data types.”

Schedule


Iterative development
Internal design reviews
March 2006
First draft specification
Sept. 2005
Final specification
Feb. 2006
Feature complete
May 2006

MATLAB is not C



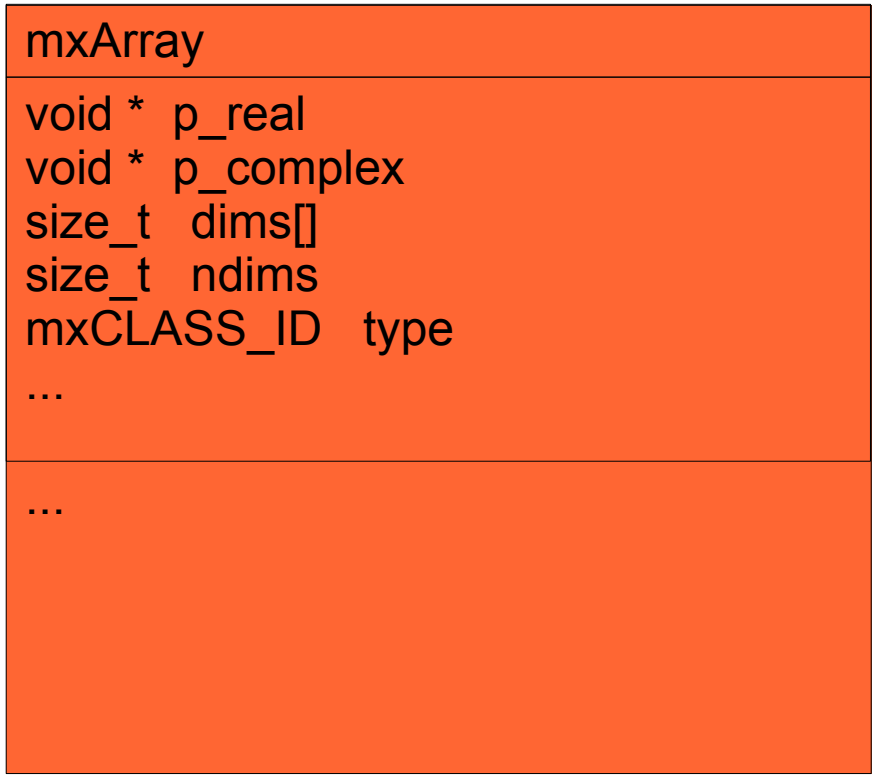
MATLAB

```
[out1, out2] = function(in1, in2);
```

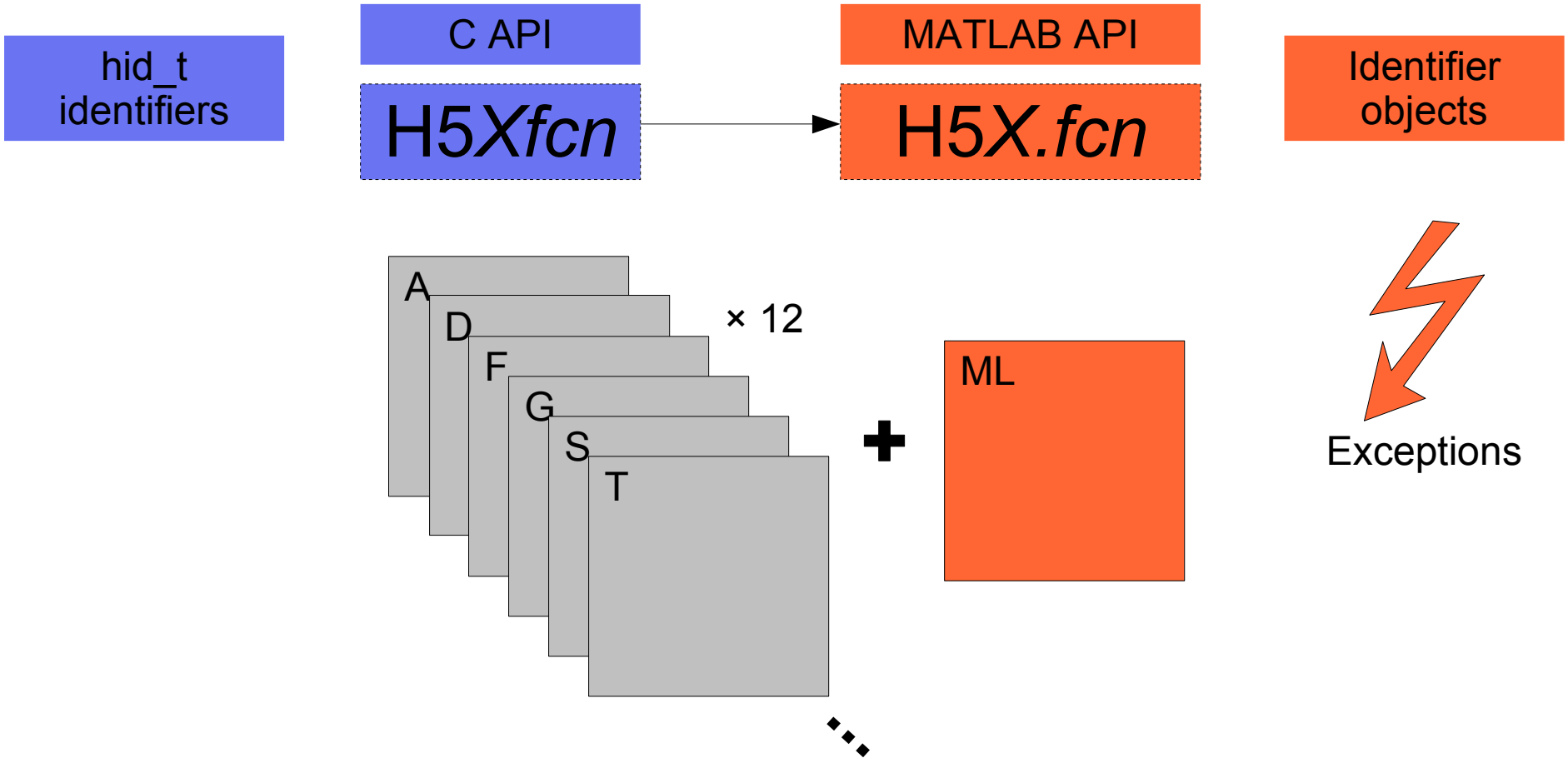
C

```
status = function(in1, in2, &out1, &out2);
```

MATLAB is not C



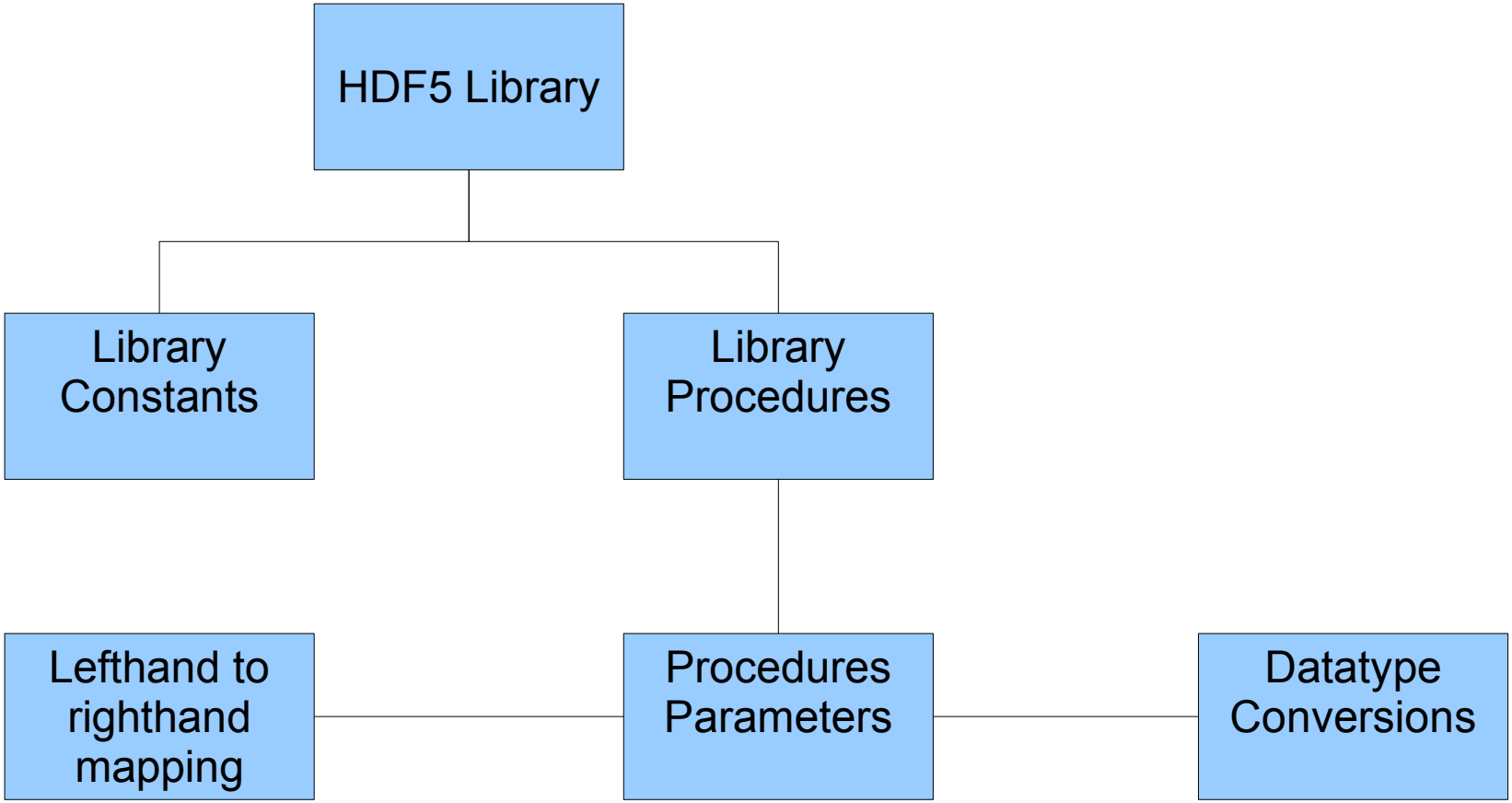
The Interface



Special MATLAB Functions

```
H5ML.compare_values

H5ML.get_constant_names

H5ML.get_constant_value

H5ML.get_function_names

H5ML.get_mem_datatype

H5ML.hoffset

H5ML.sizeof
```

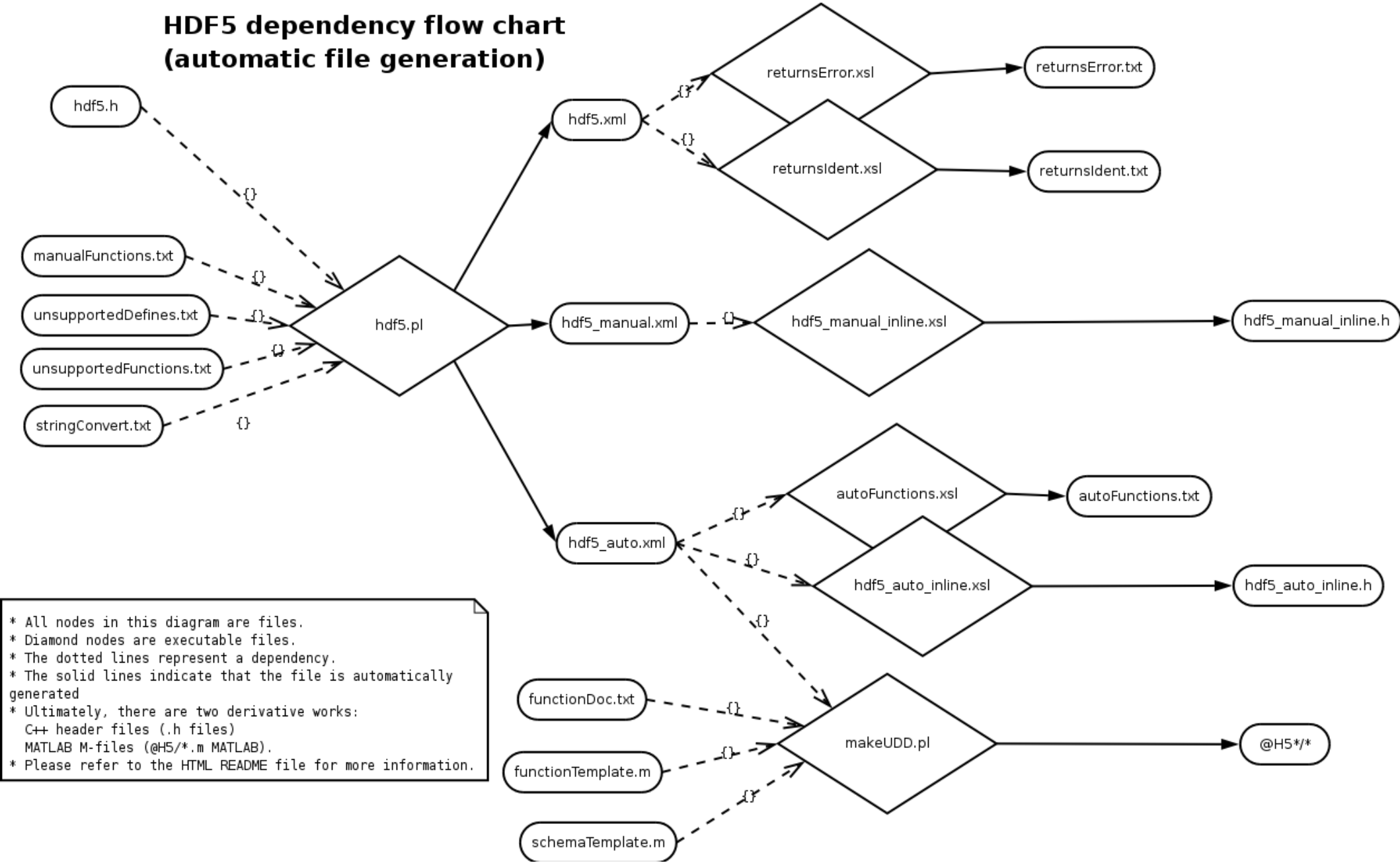
Library Model



Implementing the HDF5 Library

- Step 1: Determine auto vs. manual conversion
- Step 2: Convert .h to .xml
- Step 3: Convert XML to C++
- Step 4: Code manual functions
- Step 5: Integrate
- Step 6: Test

The conversion process



Converting XML to C++

```
// Definition
#define ADD_PROCEDURE_1_5(name,pfn,ret,a1,a2,a3,a4,a5) \
    addMethod(new LibraryProcedure_1_5< LibraryParameter_T<ret>, \
                                        LibraryParameter_T<a1>,  \
                                        LibraryParameter_T<a2>,  \
                                        LibraryParameter_T<a3>,  \
                                        LibraryParameter_T<a4>,  \
                                        LibraryParameter_T<a5> > \
                                      (name, atts, pfn));
```

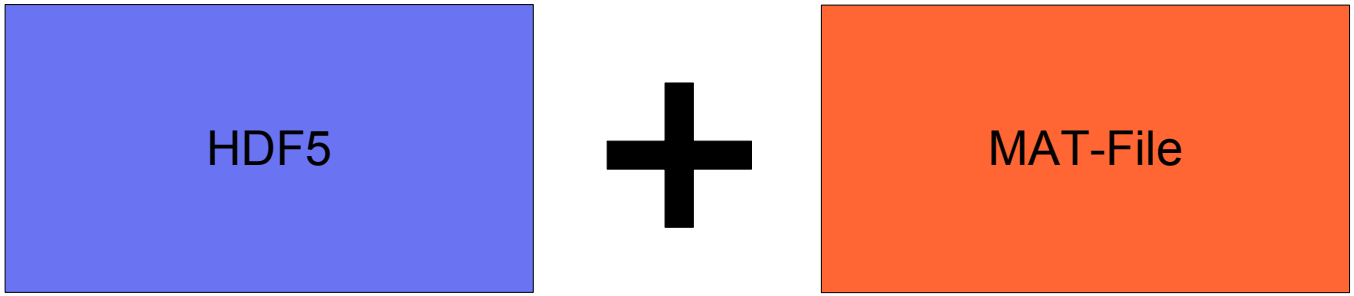
Converting XML to C++

```
// Usage (x ~220 functions)
atts.init(0,1,5,5);
atts.setParamFlags(0, ParameterAttributes::OUTPUT, 1);
atts.setParamFlags(1, ParameterAttributes::INPUT |
                      ParameterAttributes::STRING_CONVERT, 1);
atts.setParamFlags(2, ParameterAttributes::INPUT, 1);
atts.setParamFlags(3, ParameterAttributes::INPUT |
                      ParameterAttributes::STRING_CONVERT, 1);
atts.setParamFlags(4, ParameterAttributes::INPUT |
                      ParameterAttributes::STRING_CONVERT, 1);
atts.setParamFlags(5, ParameterAttributes::INPUT |
                      ParameterAttributes::STRING_CONVERT, 1);
ADD_PROCEDURE_1_5("H5Acreate", H5Acreate, hid_t,
                  hid_t, const char *, hid_t, hid_t, hid_t);
```

The HDF Group and The MathWorks

- Continue to communicate future directions.
- Don't change the existing API functions.
- Communicate API functionality changes.
- Produce a machine parsable version of hdf5.h.

The Future



- 64-bits for large arrays
- Data subsetting on load
- Type conversion on load
- Parallel I/O?