HDF5 Roadmap and New Features

July 22, 2022

The HDF Group

Dana Robinson
The HDF Group

What we'll cover…

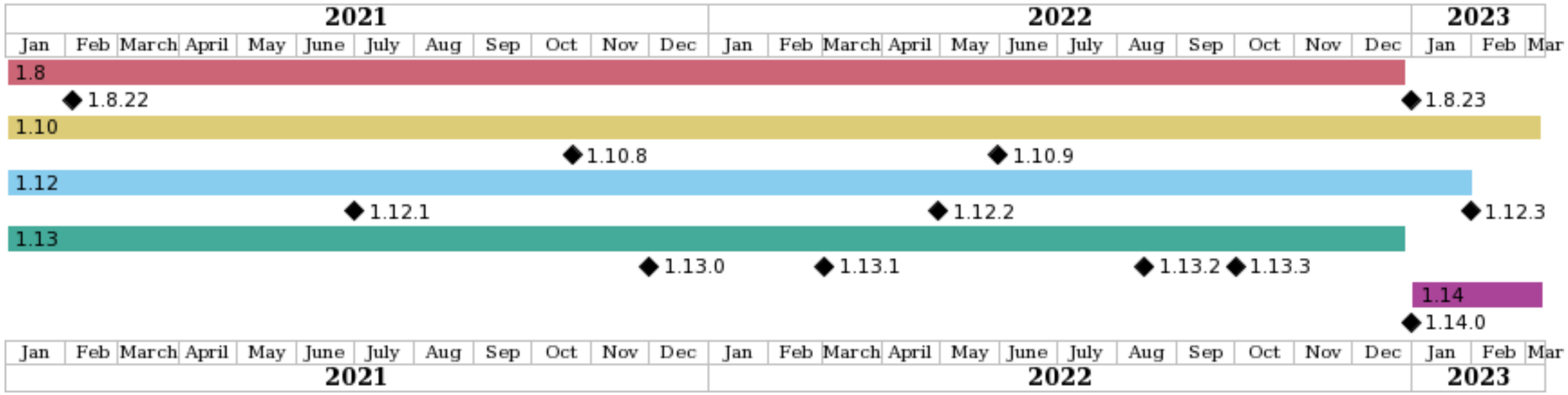
- HDF5 release schedule
- Virtual Object Layer
- Onion VFD
- VFD SWMR
- "HDF5 2.0"

HDF5 Release Schedule

The Current Roadmap

- Published on GitHub in README.md (current as of today)
- Delays 1.13.2 by ~1 month so we can release the subfiling feature earlier
- 1.8 and 1.12 will be retired at the end of the year

HDF5 Release Schedule



1.13.x: New Features

HDF5 1.13.2 (August)	• Selection I/O	(vector I/O at the VFD level)
	• Onion VFD	(versioned HDF5 files)
	• VFD SWMR	(improved SWMR feature)
	• Subfiling	(I/O concentrators in parallel HDF5)
HDF5 1.13.3 (October)	• Multi-Dataset I/O	(I/O that spans multiple datasets)

Experimental vs Maintenance Releases

The HDF Group

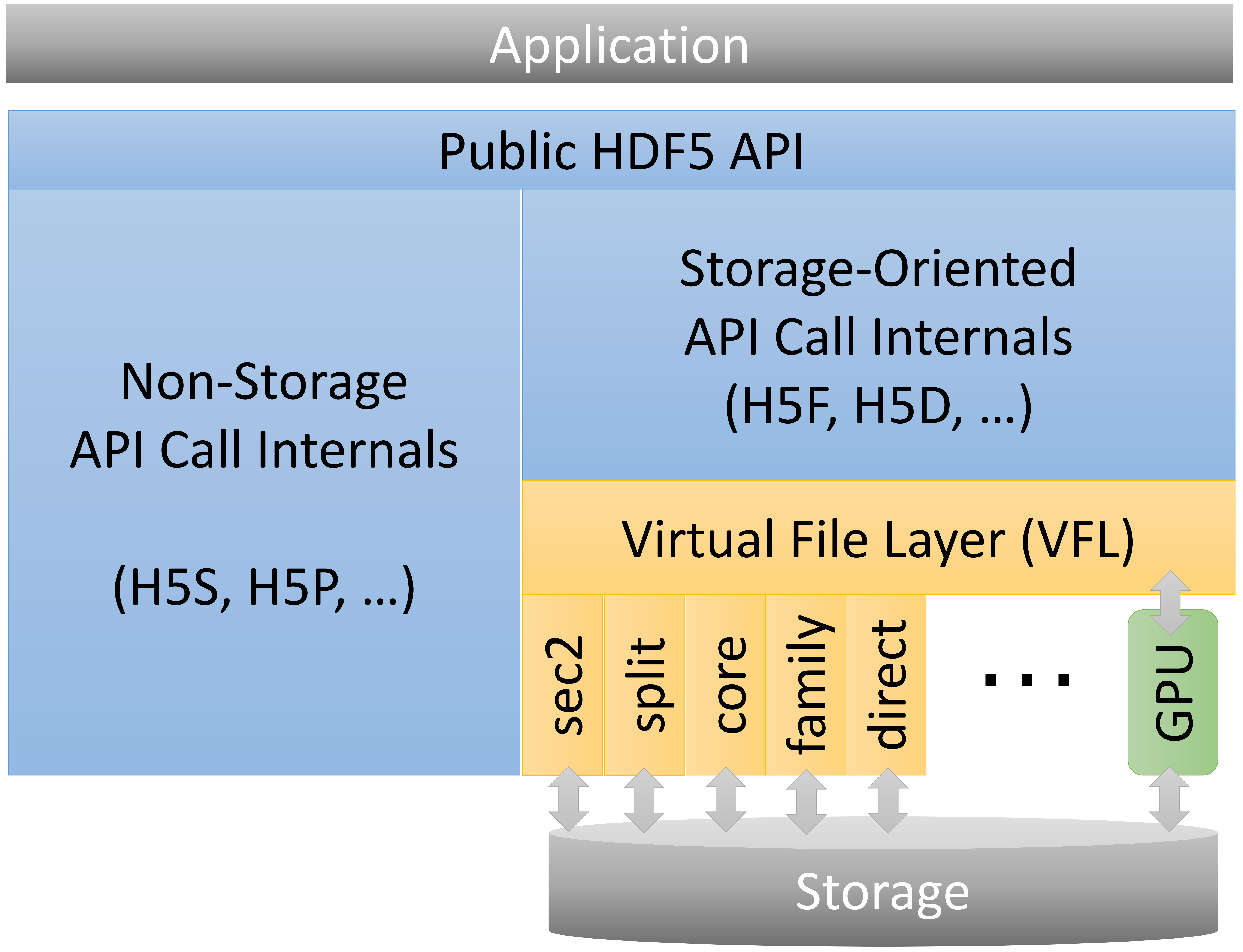
- Even number minor releases are maintenance releases (e.g., 1.12.x)
  - Usual HDF5 maintenance branches you know and love
  - Binary compatibility
  - Stable file format

- Odd number minor releases are experimental releases (e.g., 1.13.x)
  - No binary compatibility guarantees
  - API calls can change
  - File format can change
  - Unready features may be dropped
  - Used to try out new features as we prepare for the next maintenance release

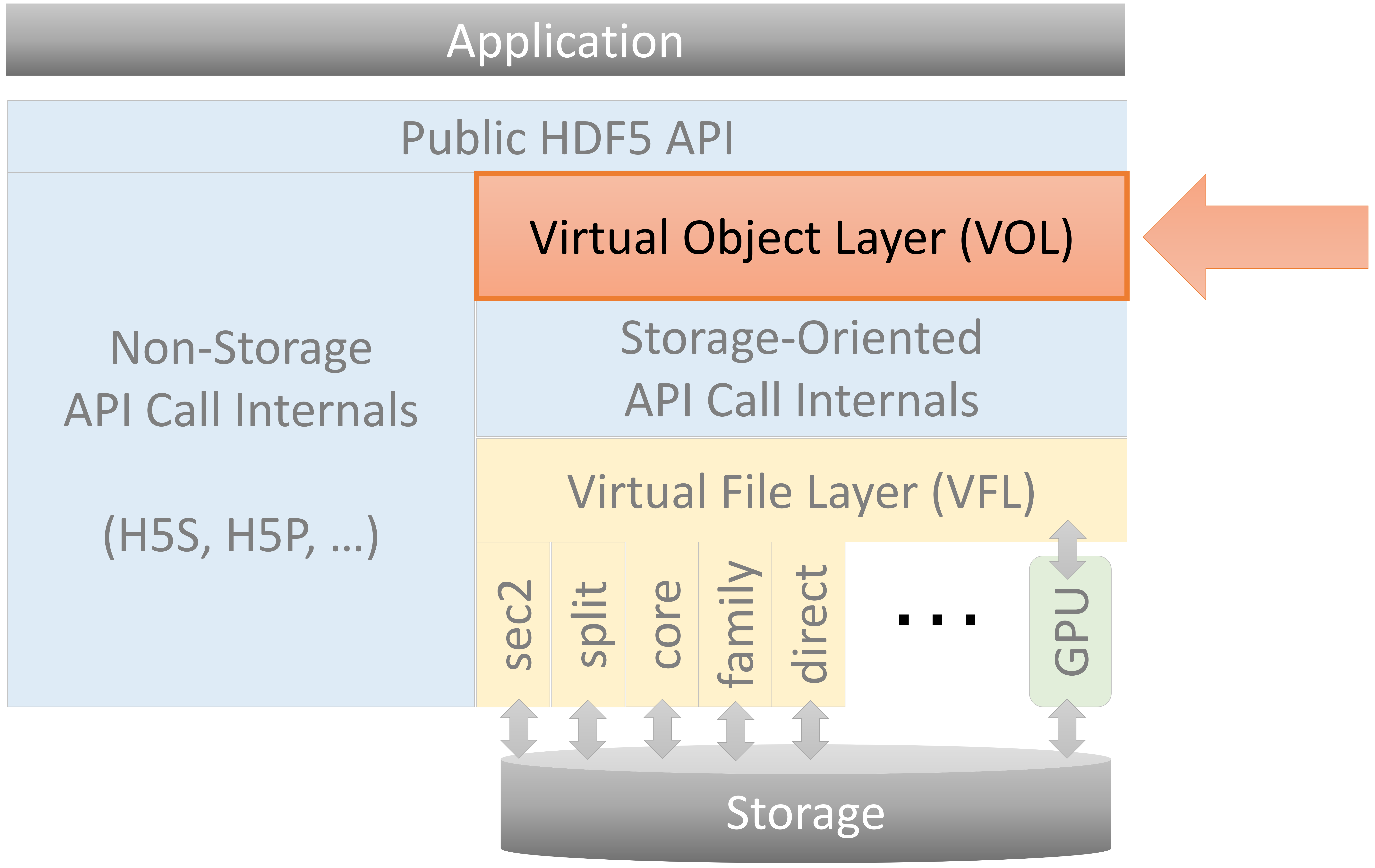
- See the blog posts on this for more clarity
  - https://www.hdfgroup.org/2021/12/hdf5-1-13-0-introducing-experimental-releases

Virtual Object Layer

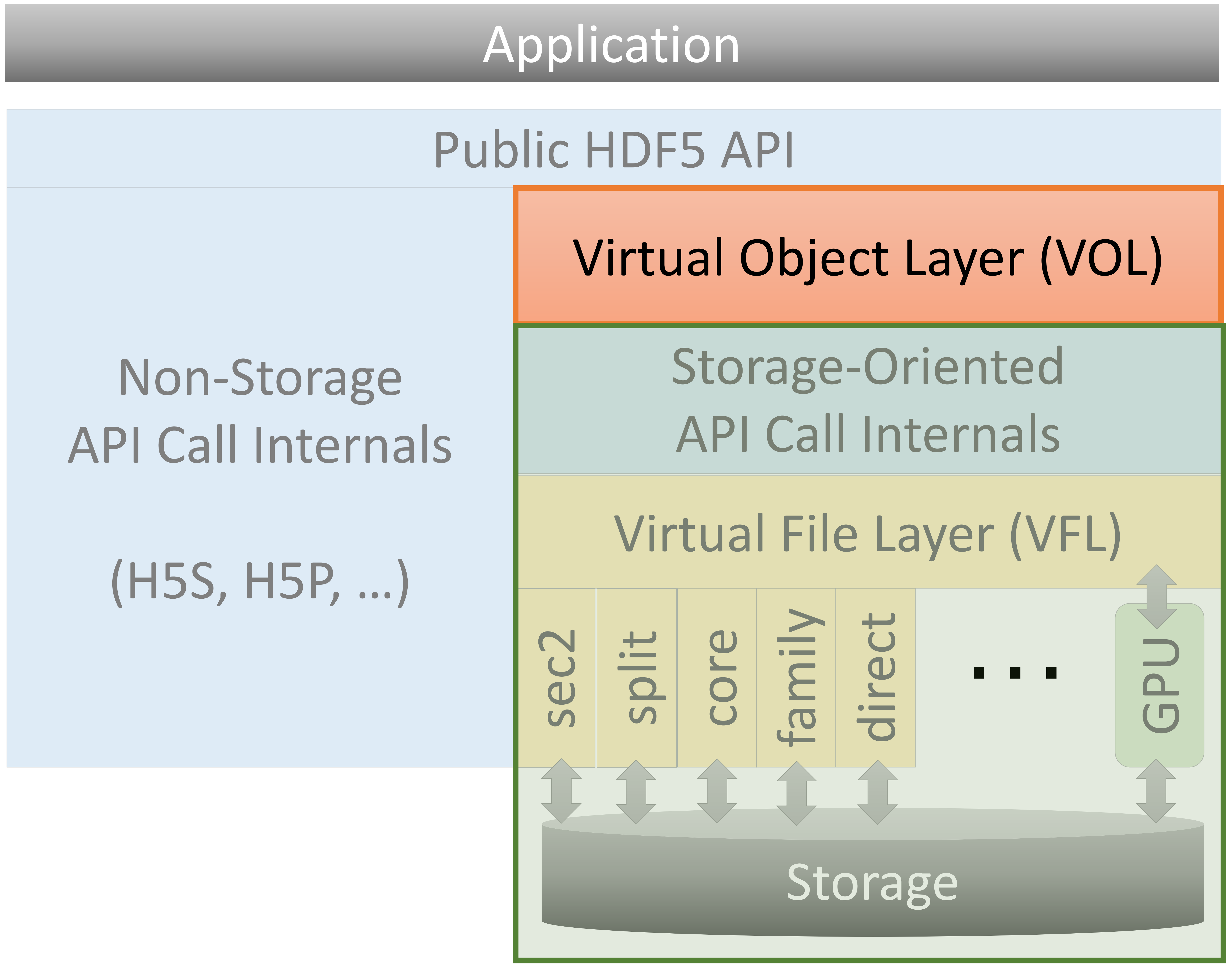
Original HDF5 Architecture (pre-1.12.0)



Current HDF5 Architecture (1.12.0+)



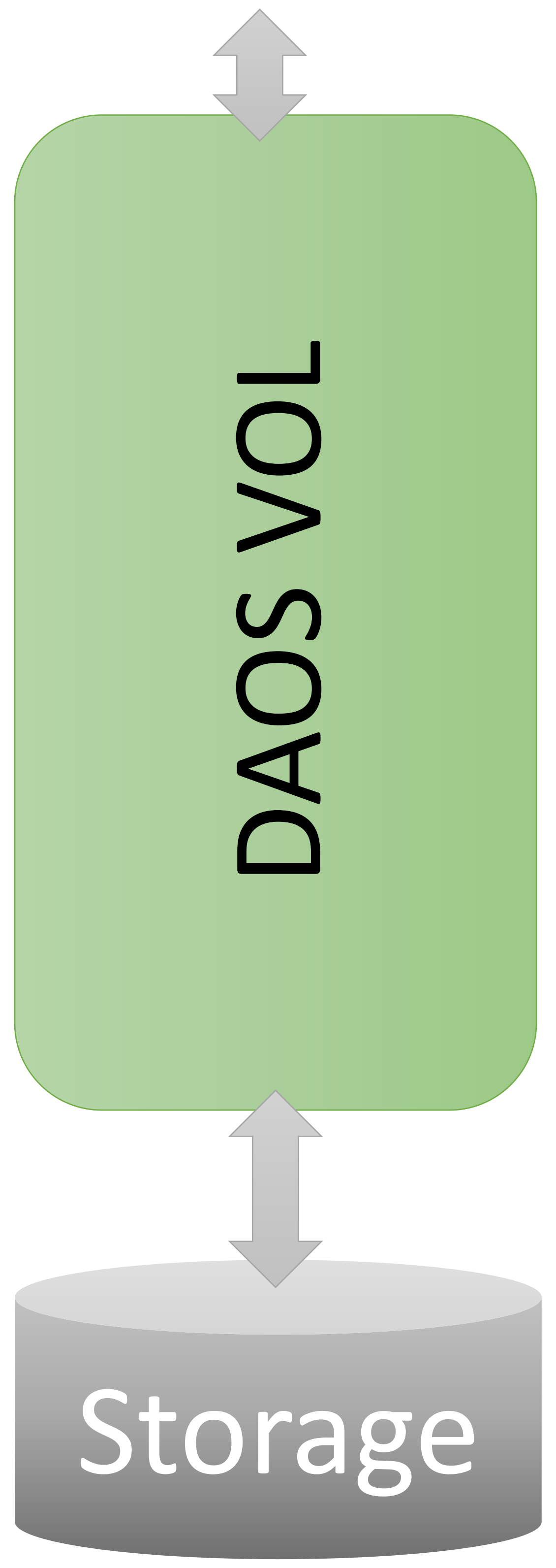
Current HDF5 Architecture (1.12.0+)



New Layer

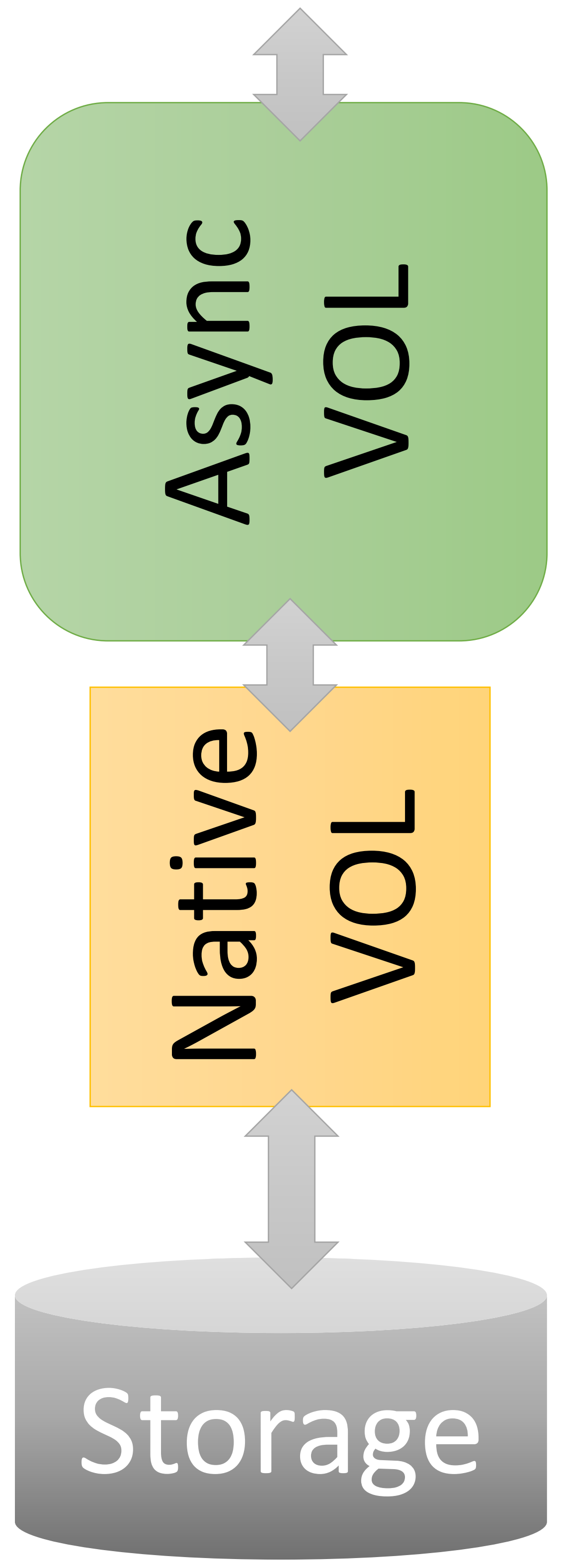
Allows you to replace this with your own code!

Two Kinds of VOL Connector



Terminal

Maps HDF5 objects to arbitrary storage schemes

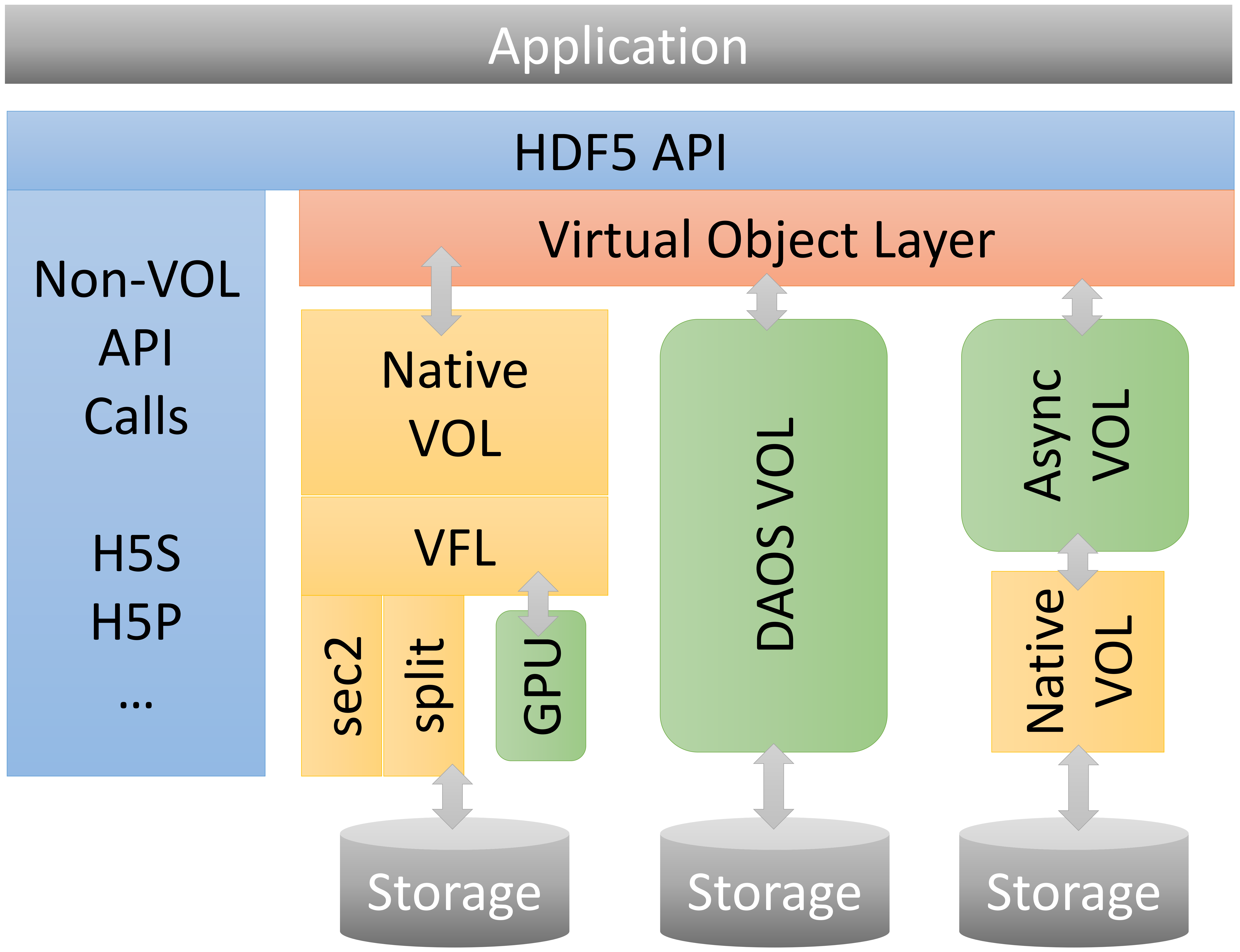


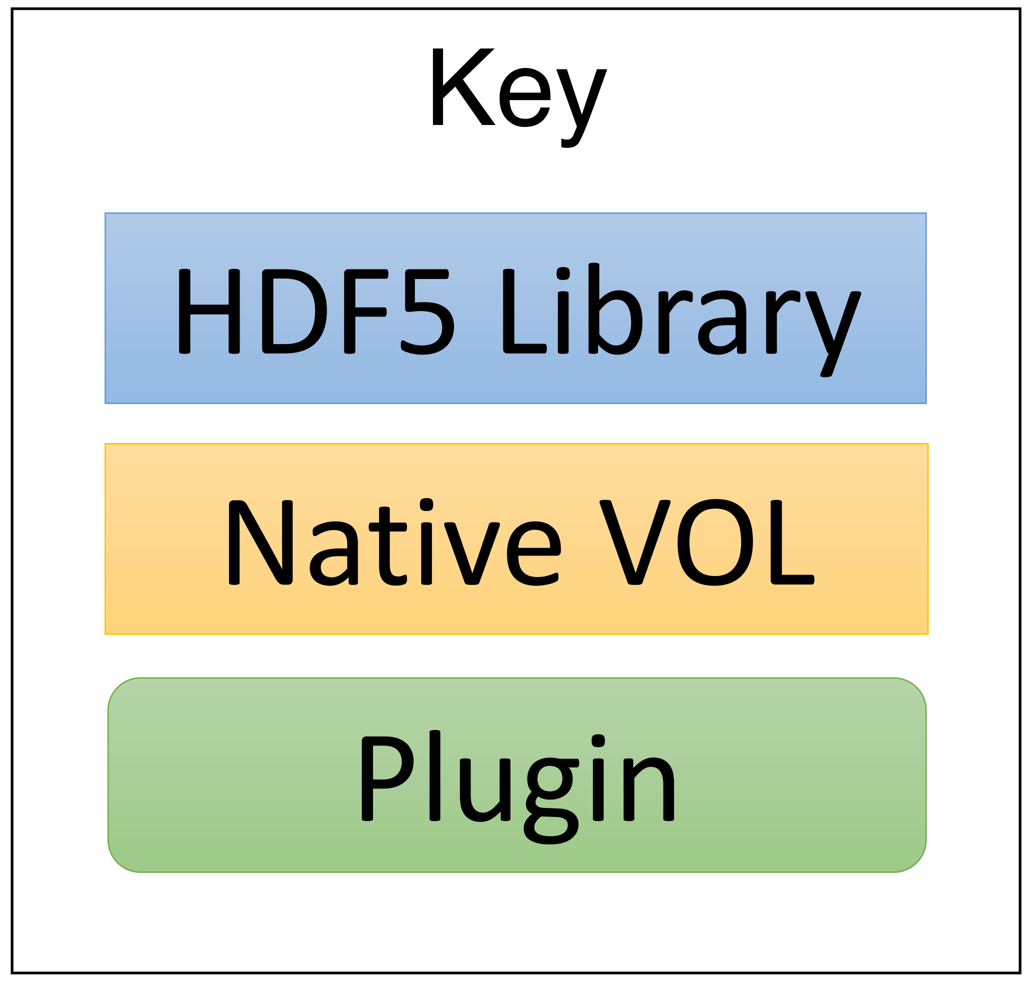
Pass-Through

Perform operations (e.g. caching, logging) before passing the data on to the next connector.

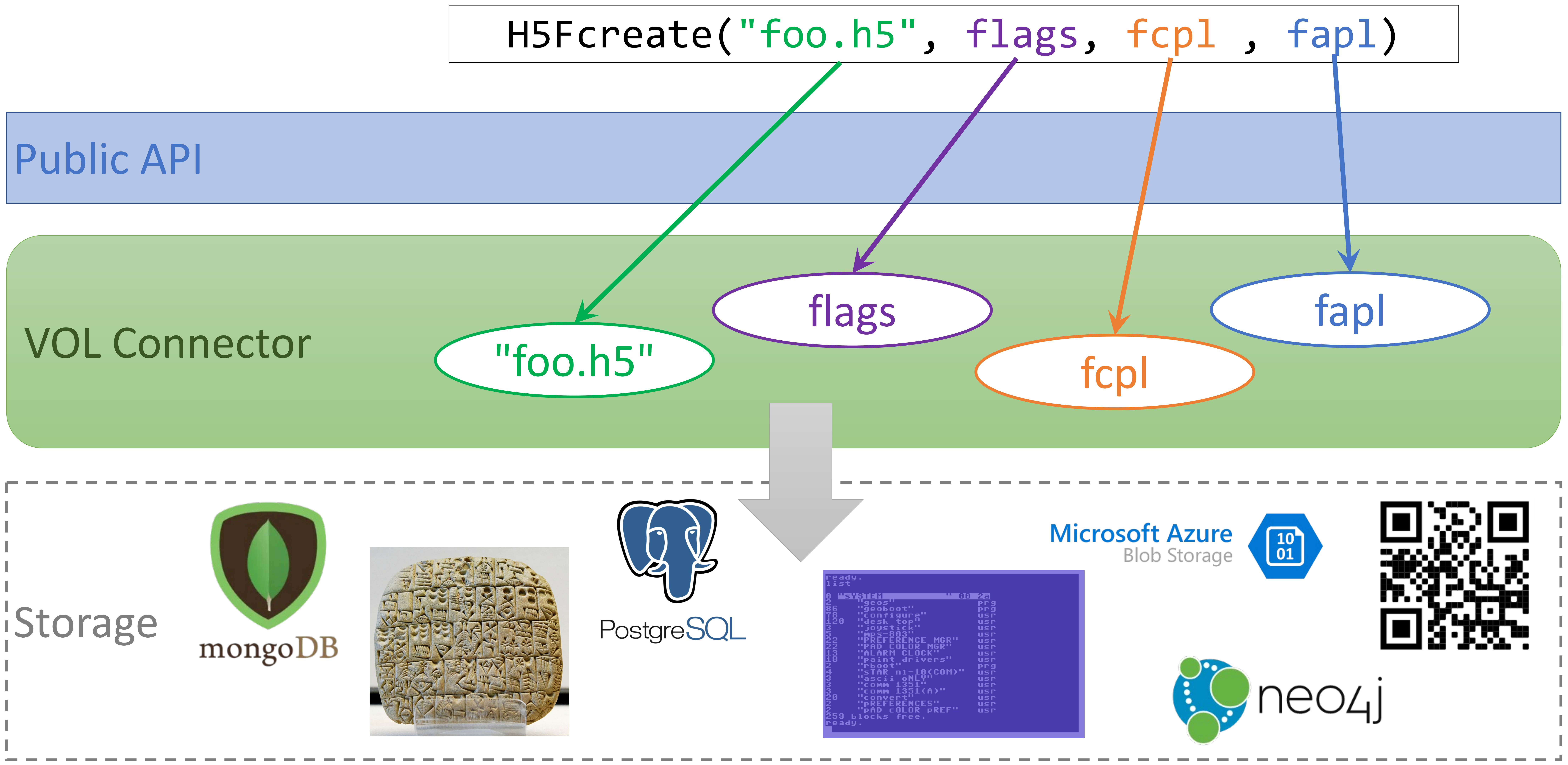
Current HDF5 Architecture (1.12.0+)

The HDF Group





Terminal VOL Connectors



```
ready.
list

0 "sYSTEM           " 00 2a
2     "geos"                 prg
86    "geoboot"              prg
78    "configure"            usr
120   "desk top"             usr
3     "joystick"             usr
5     "mps-803"              usr
22    "PREFERENCE MGR"       usr
22    "PAD COLOR MGR"        usr
13    "ALARM CLOCK"          usr
18    "paint drivers"        usr
2     "rboot"                prg
4     "sTAR nl-10(COM)"      usr
3     "ascii oNLY"           usr
3     "comm 1351"            usr
3     "comm 1351(A)"         usr
20    "convert"              usr
2     "pREFERENCES"          usr
2     "pAD cOLOR pREF"       usr
259 blocks free.
ready.
```

VOL Toolkit Repository

- Location: https://github.com/HDFGroup/vol-toolkit
- All your VOL construction needs in a single location
- Does not contain original content
- Designed to bring important content from other repositories together with consistent versioning
- Content is mainly included as git submodules, though the docs are currently copied in
- Tags will identify "HDF5 1.13.0", etc. versions of the toolkit
- Includes a VOL construction tutorial

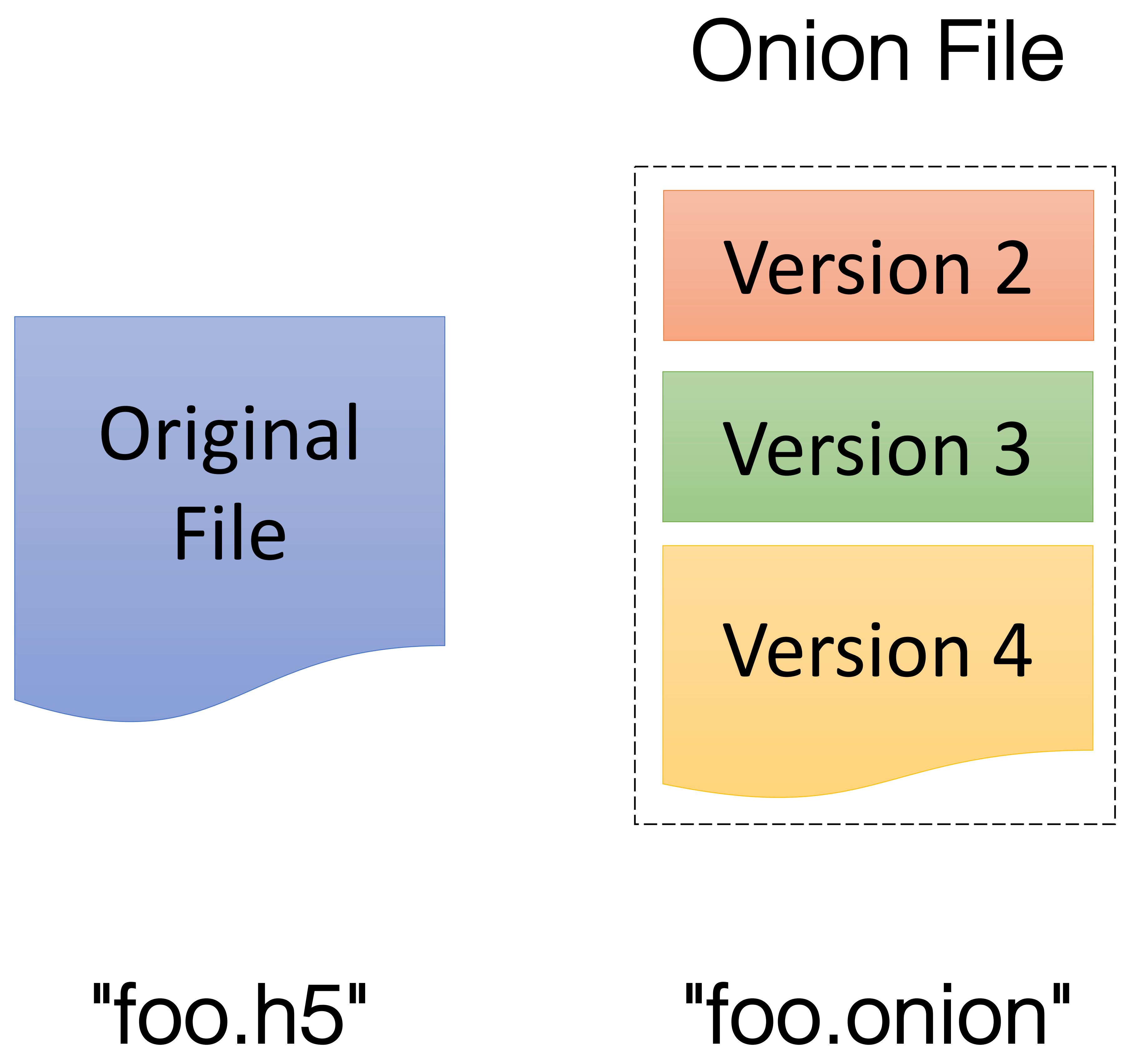
HDF5 1.13.x / 1.14.0 VOL Changes

- **NOTE: ALL VOL CONNECTOR DEVELOPMENT SHOULD TARGET HDF5 1.13.x**
- Do **NOT** use HDF5 1.12.x
- There are important changes to the VOL interface in HDF5 1.13.x that cannot be moved to 1.12.x without breaking binary compatibility
- As mentioned earlier, HDF5 1.13.x are experimental releases
  - It is possible that the VOL interface could be changed in the 1.13.x versions that will be released as HDF5 1.14.0
  - Example: VOL connector feature flags being introduced this summer that describe HDF5 API capabilities
  - Should be fairly stable, and updating is straightforward

Onion VFD

Onion VFD

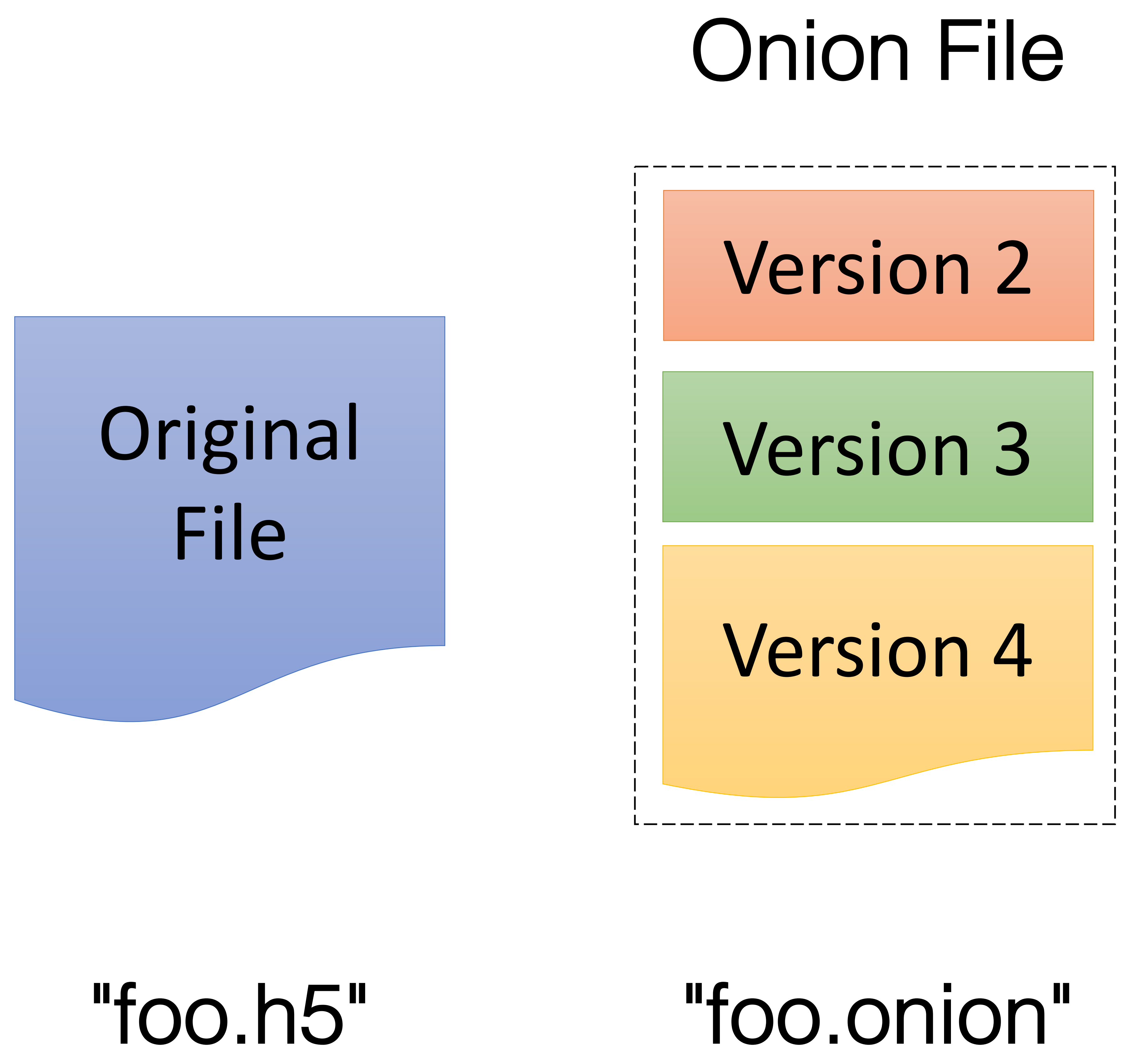
- Enables creating "versioned" HDF5 files
- "version" defined by file open/write/close cycle
- Data and metadata for additional versions stored in a separate "onion" file
- Only one onion file, regardless of number of versions
- Can open onion files by version
- New API calls to get the number of versions, etc.
- Command-line tools support



Linear only! No trees!

Onion VFD

- Will be released in HDF5 1.13.2 (August)
- Not coming to 1.12.x or earlier due to incompatible VFL changes

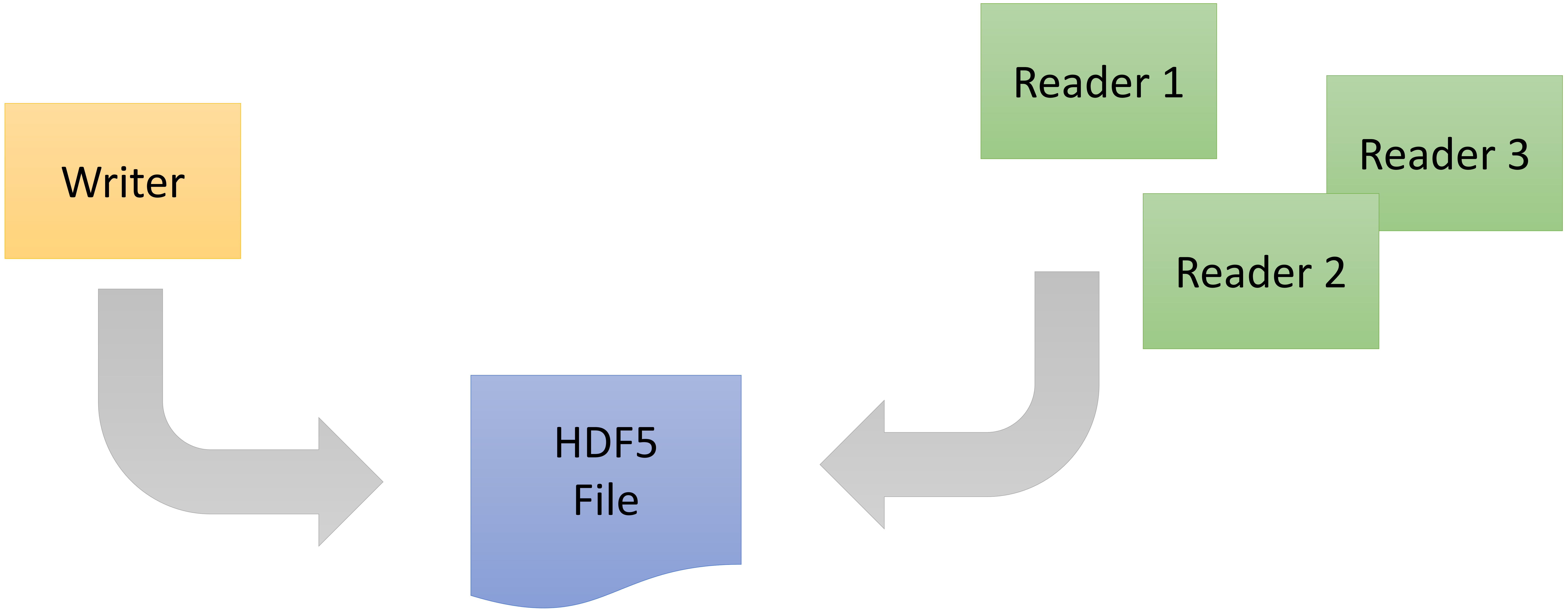


Linear only! No trees!

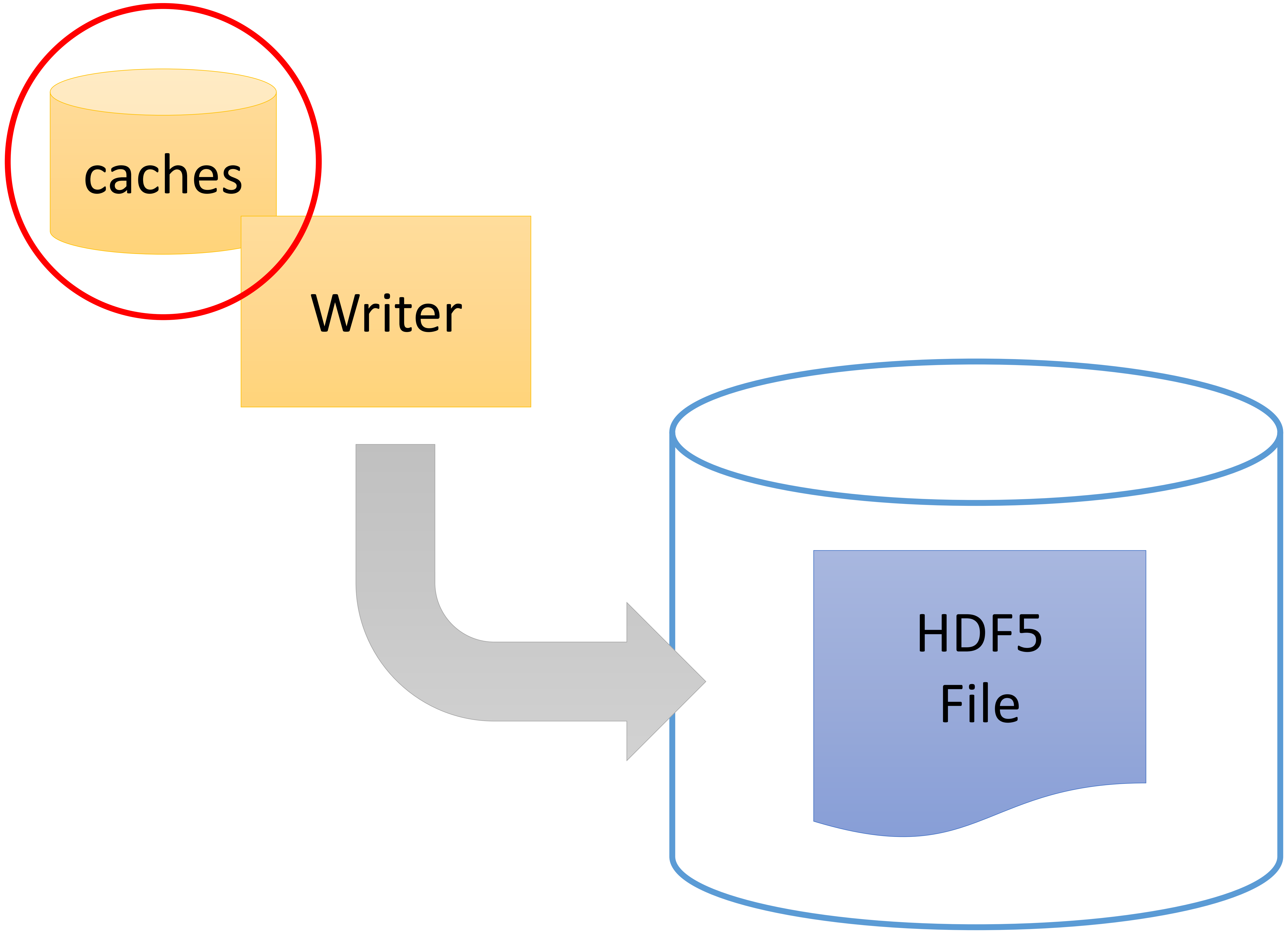
VFD SWMR

SWMR

SWMR = Single Writer / Multiple Readers



The Problem: State



State of an HDF5 file =

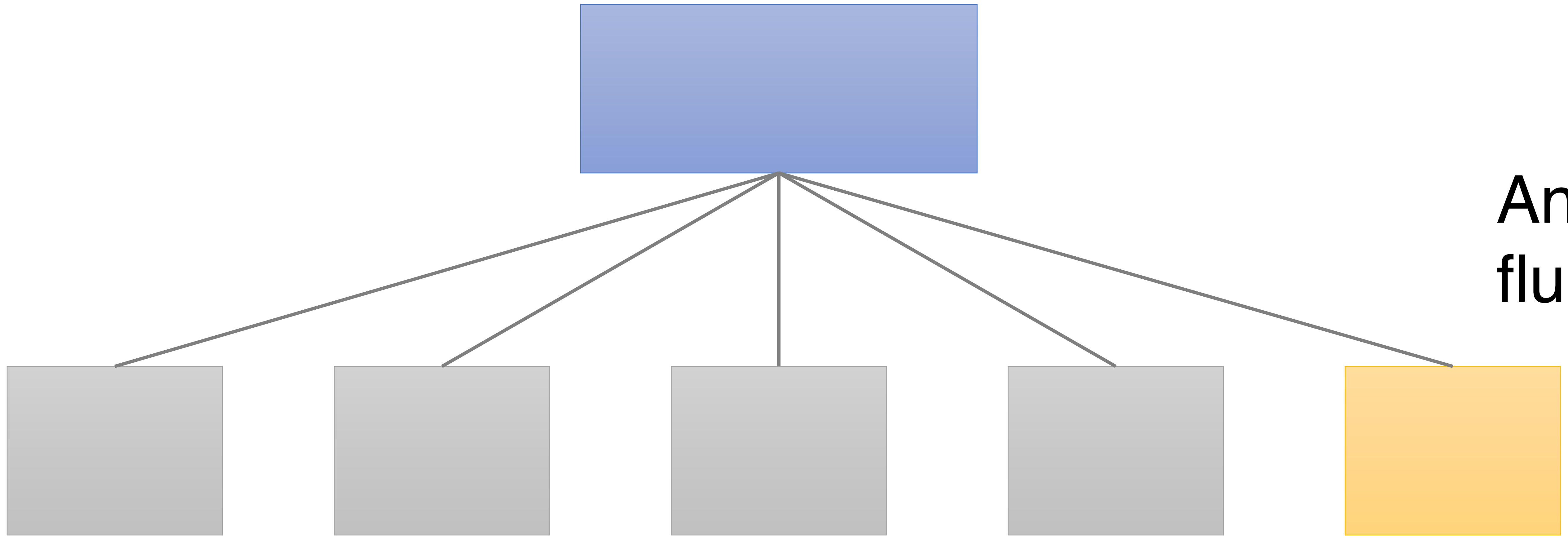
things written to disk

+

what's in the writer's caches

The Problem: State

If this tree node is on disk



And this node has not been flushed from the cache

Any reader that tries to follow the link to the leaf node will read garbage

"Legacy" SWMR (1.10.0+)

- Relies on "flush dependencies" in the metadata cache
- We ensure that children are flushed before parents
- File is always consistent, but not necessarily 100% up to date wrt the writer
- Very limited! – Not implemented for all HDF5 operations
    - Dataset appends only
    - Can't create new file objects, no variable-length type support
- High maintenance cost as SWMR pervades the metadata cache code

VFD SWMR (1.13.2+)

The HDF Group

- Uses external, temporary metadata files
- Implemented at the VFL level
- Like legacy SWMR, the file is always consistent, but not necessarily 100% up to date wrt the writer
- Works with almost all HDF5 operations
- Lower maintenance cost as most SWMR code is isolated to a few internal modules

VFD SWMR (1.13.2+)

- Will be released in HDF5 1.13.2 (August)
- Not coming to 1.12.x or earlier due to incompatible VFL changes
- Legacy SWMR remains in place for now

HDF5 2.0

"HDF5 2.0"

HDF5 1.14.0 will release at the end of the year. What then?

I feel it's time for a (medium-size) API reset

- HDF5 is a 25-year-old codebase that has been managed conservatively
- Not all our choices have been good ones
- Let's fix what is broken!
- Currently in the planning stage, so statements here come with forward-looking caveats

My intention is NOT to rework the API in such a way that causes pain for the community. We want to make any changes in such a way that it's as easy as possible for HDF5 software to adapt.

HDF5 2.0 – Possible Changes

Smaller, more manageable things:

- Semantic versioning
- API improvements
  - Deprecated functions will be removed
  - Signature changes (e.g. all API calls will return herr_t, etc.)
- Renaming (e.g. "sec2" VFD --> "posix" VFD)
- Reworked error mechanism
- Drop the multi VFD (but keep the split VFD)
- Sanitize the metadata I/O layer in the library (source of most of our CVE issues)

If resources are available:

- **Improved variable-length storage**

HDF5 2.0 – Suggestions

We're a community, so let's decide this together!

Join the discussion on our forum: https://forum.hdfgroup.org/

References

Links to documentation

While we work on release documentation, the latest RFC documents can be found in the hdf5doc repository on GitHub:

Selection I/O
https://github.com/HDFGroup/hdf5doc/blob/master/RFCs/HDF5_Library/SelectionIO/selection_io_RFC_210610.docx

Multi-Dataset I/O
https://github.com/HDFGroup/hdf5doc/blob/master/RFCs/HDF5_Library/MultiDataset/H5HPC_MultiDset_RW_IO_RFC_v7_20220523.pdf

Subfiling
https://github.com/HDFGroup/hdf5doc/blob/master/RFCs/HDF5_Library/VFD_Subfiling/RFC_VFD_subfiling_200424.docx

Links to documentation

While we work on release documentation, the latest RFC documents can be found in the hdf5doc repository on GitHub:

Onion VFD
https://github.com/HDFGroup/hdf5doc/blob/master/RFCs/HDF5_Library/OnionVFD/Onion_VFD_RFC_v5.docx

VFD SWMR
https://github.com/HDFGroup/hdf5doc/blob/master/RFCs/HDF5_Library/VFD_SWMR/VFD_SWMR_RFC_220519.pdf

Links to documentation

HDF5 Youtube channel
https://www.youtube.com/channel/UCRhtsIZquL3r-zH-R-r9-tQ

Onion VFD demonstration
https://www.youtube.com/watch?v=cfshkgr05mA&ab_channel=hdf5

VOL connector construction tutorial
https://www.youtube.com/watch?v=7XEbm-__QuM&t=2s&ab_channel=hdf5

THANK YOU!

Questions & Comments?