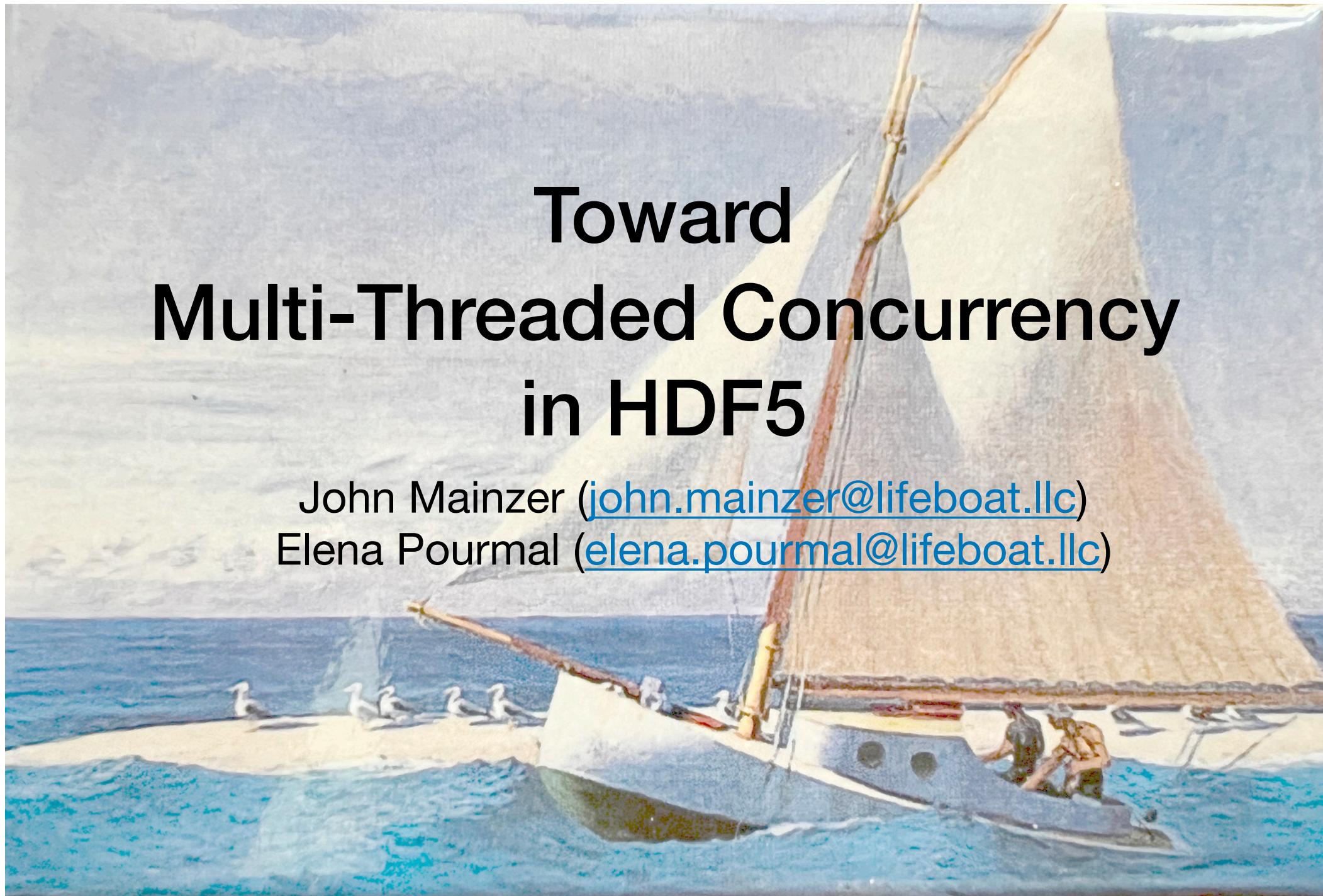# Toward
Multi-Threaded Concurrency in HDF5

John Mainzer (john.mainzer@lifeboat.llc)
Elena Pourmal (elena.pourmal@lifeboat.llc)

Outline

- Introduction to Lifeboat, LLC

- Motivation for multi-threaded* (MT) HDF5 and use cases

- Technical approach

- Next steps

- Acknowledgement

*We interchangeably use "multi-threaded", "concurrent", "thread-safe", 'thread-safety" terms with the meaning of "concurrent threads are allowed in the HDF5 library without corrupting data in memory and in storage".

Introduction to Lifeboat, LLC

- Formed in August 2021

- Mission

  Contribute to HDF5 software sustainability

- Received Phase I DOE funding for exploring multi-threaded concurrency in HDF5

Motivation for MT HDF5

- MT feature in HDF5 is a long-time request from the community and The HDF Group customers

  ‣ Thread-safe software is an industry standard

  ‣ Convenience when used with MT applications

  ‣ Performance

- Current workaround for multi-threaded HDF5 applications:

  Use thread-safe builds of HDF5 to access data

  ‣ Only one thread at a time is allowed into the library creating an I/O bottleneck.

  Read HDF5 files directly by-passing the library

  ‣ Custom solution supports a limited set of HDF5 features and do not provide a general solution.

Use Cases for MT HDF5

- Most requests are for reading data from HDF5 files:

  - Multiple threads open different HDF5 files read-only, and read them.

  - Multiple threads open the same HDF5 file read-only, and read different datasets.

  - Multiple threads open the same HDF5 file in read-only, and read disjoint subsets of the same dataset.
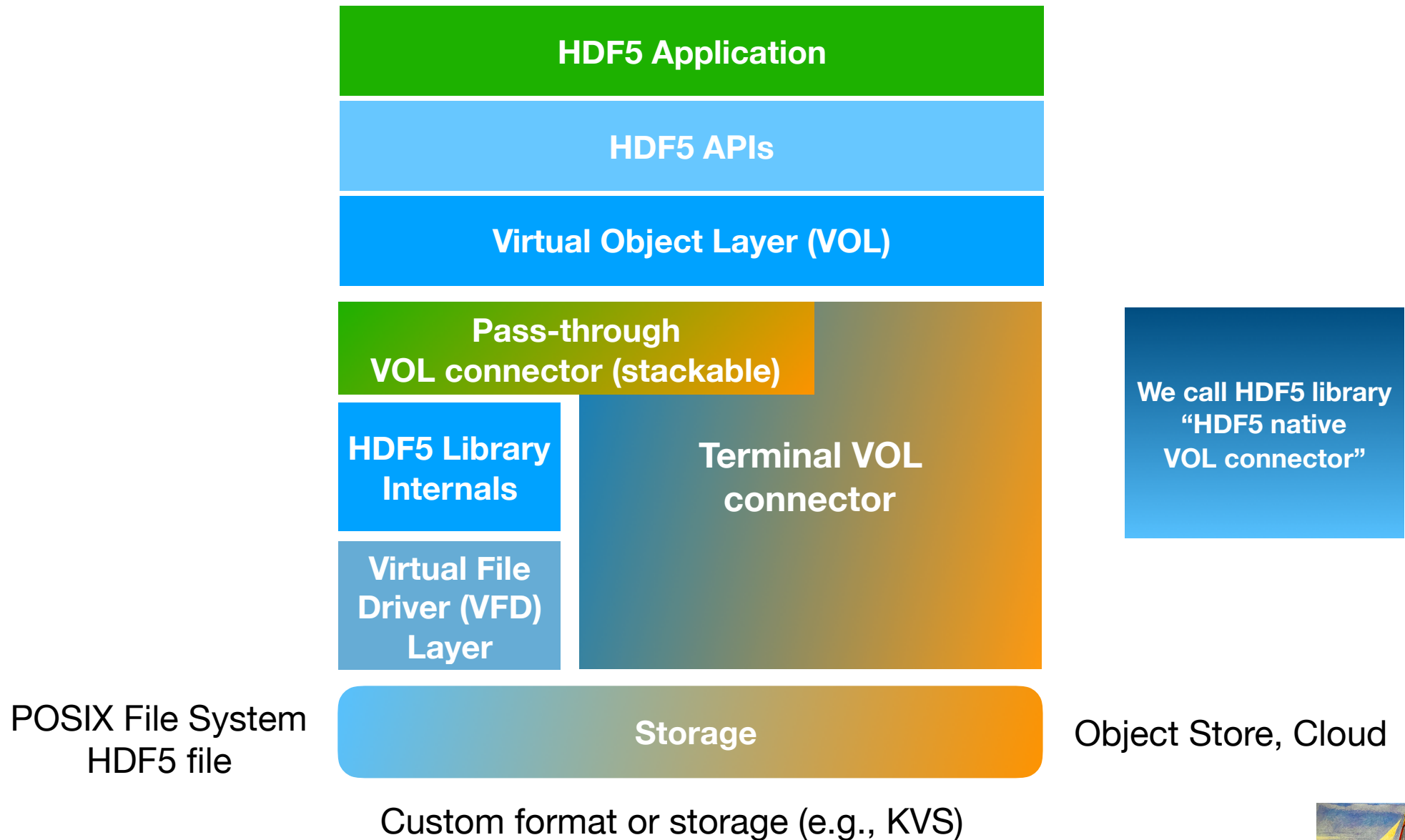
Technical Approach

- Retrofitting HDF5 to make it MT is HARD!

  - HDF5 is still under active development and is used in mission critical software; it cannot be unstable

  - Legacy code with scarce developer-level documentation

- Is it even possible?

  - "Yes" due to the HDF5 library organization by packages, e.g., H5FD, H5VL, H5I, H5E, H5AC, etc.

    - Some packages do not have complex dependencies on other packages and can be converted; some do have and have to be reworked.

  - Going "bottom up" and reworking necessary parts of the library is still doable but a daunting task.

HDF5 Library Architecture

HDF5 Application

HDF5 APIs

Virtual Object Layer (VOL)

Pass-through VOL connector (stackable)

HDF5 Library Internals

Virtual File Driver (VFD) Layer

Terminal VOL connector

We call HDF5 library "HDF5 native VOL connector"

Storage

POSIX File System HDF5 file

Object Store, Cloud

Custom format or storage (e.g., KVS)
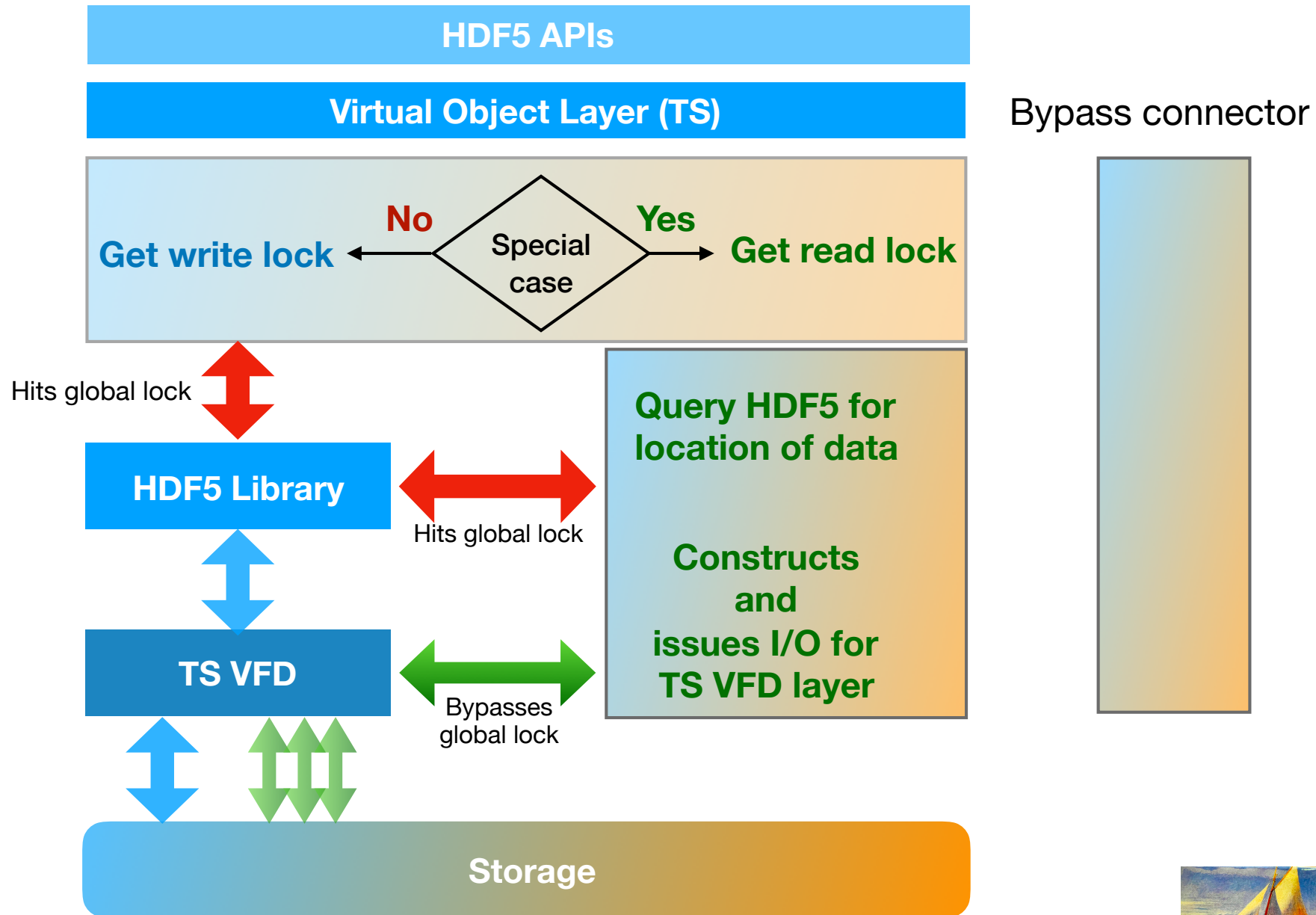
Proposed Approach
"Proof of Concept with Immediate Return"

- Make the following packages TS and contribute to HDF5 source
  - ‣ H5VL - VOL layer
  - ‣ H5E - Error handling
  - ‣ H5CX - Context
  - ‣ H5I - Identifier
  - ‣ H5P - Property lists
  - ‣ H5S - Selections (later)

- Make the HDF5 Virtual Driver layer (VFD) thread-safe
  - ‣ Sub-filing feature will benefit from MT VFD when selection I/O is implemented in the future

- Create MT bypass VOL connector to handle limited use cases
  - Reading from contiguous arrays of scalar datatypes without support for VL types
  - No datatype conversion, etc. TBD

Bypass VOL Connector

HDF5 APIs

Virtual Object Layer (TS)

Bypass connector

No — Special case — **Yes**

Get write lock — **Get read lock**

Hits global lock

HDF5 Library

Hits global lock

Query HDF5 for location of data

Constructs and issues I/O for TS VFD layer

TS VFD

Bypasses global lock

Storage

Pros and Cons of Proposed Approach

- Pros:

  - Creates minimal disruption to HDF5 and applications

    - ‣ No API changes

    - ‣ Threads not employing the multi-thread enable API calls interact with the HDF5 library as usual (i.e, no effect on BLOSC compression)

    - ‣ Pathway for a limited expansion beyond the initial MT support

  - Enables third-party MT VOL connectors

  - Provides feasibility and estimate on converting HDF5 using "bottom up package-by-package" approach

- Cons:

  - Expanding bypass VOL to have full HDF5 functionality is time consuming and is equivalent to re-writing HDF5

Next Steps

- Deliver bypass VOL connector, updated selected HDF5 packages, and documentation, and expand HDF5 TS tests

- Based on Phase I lessons learned, propose direction for future work — most likely one of:

  ‣ A complete rewrite of HDF5

  ‣ Continue with the "bottom-up package by package" conversion approach

  ‣ Combination of the above approaches

- Secure funding for Phase II and Phase III

Acknowledgements

- The HDF Group development team

- Edward Hopper , "The Martha McKeen of Wellfleet", 1944