

EU HUG 2022: HSDS v0.7 New Features



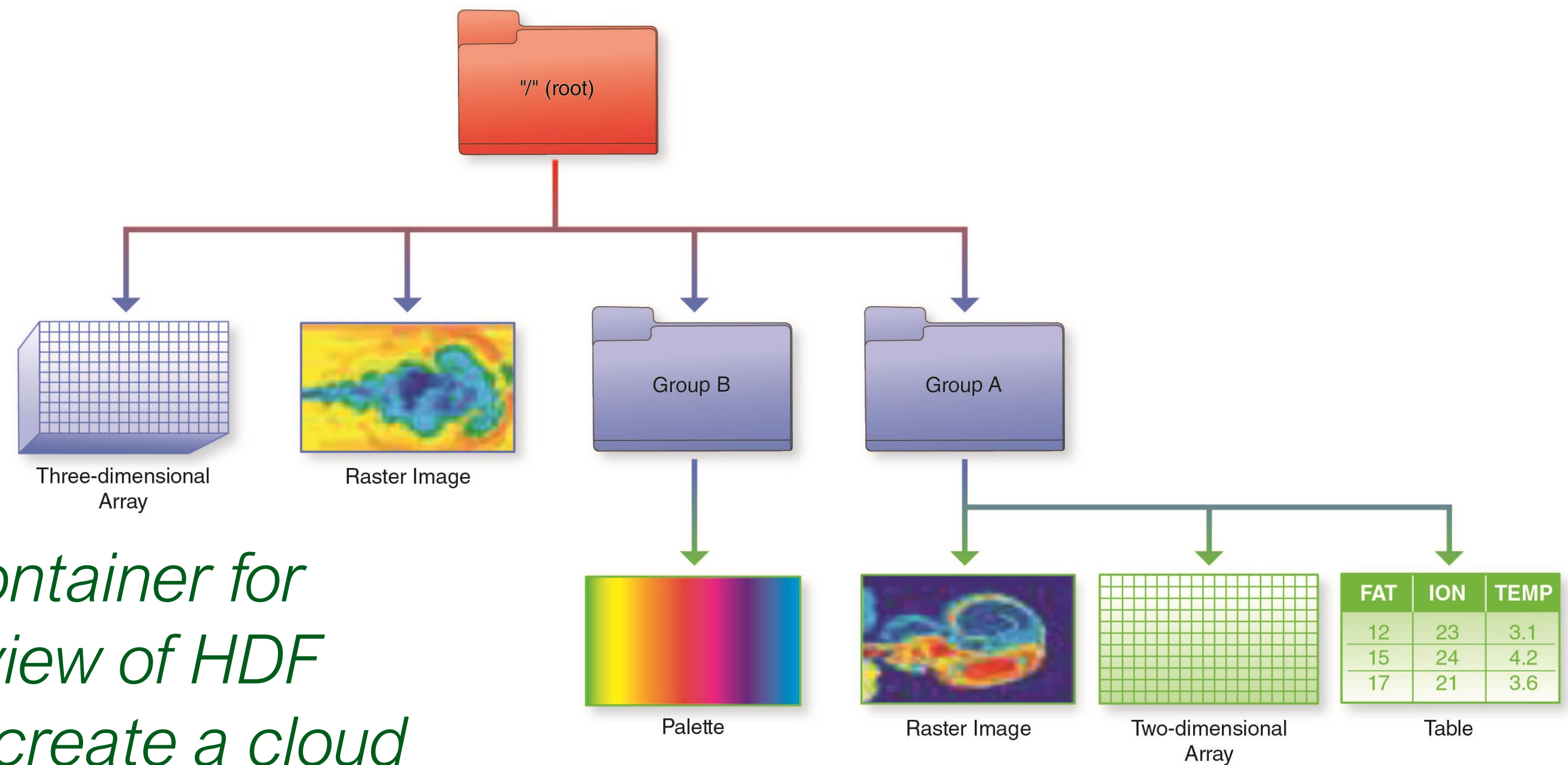
Outline

- Brief Overview of HSDS
- HSDS Deployment Smorgasbord
 - Docker
 - (Improved) Kubernetes
 - (New) Serverless 1: h5pyd direct mode
 - (New) Serverless 2: Lambda
- Streaming support
- Fancy indexing
- Case Studies

What is HDF5?

Depends on your point of view:

- a C-API
- a File Format
- a data model



*The File format is just a container for
The data. Dropping this view of HDF
allows us to more flexibly create a cloud
version of HDF.*

HDF for the Cloud

- **Ideas**
 - **Web based - provide a RESTful API that is feature compatible with HDF5 Lib API**
 - **Utilize object storage – cost effective, scalable throughput, redundant**
 - **Elastic compute – scale throughput by autoscaling compute clusters**
 - **Compatibility - Provide client SDK so existing HDF applications can just work**

HDF Cloud Schema

How to store HDF5 content in S3?

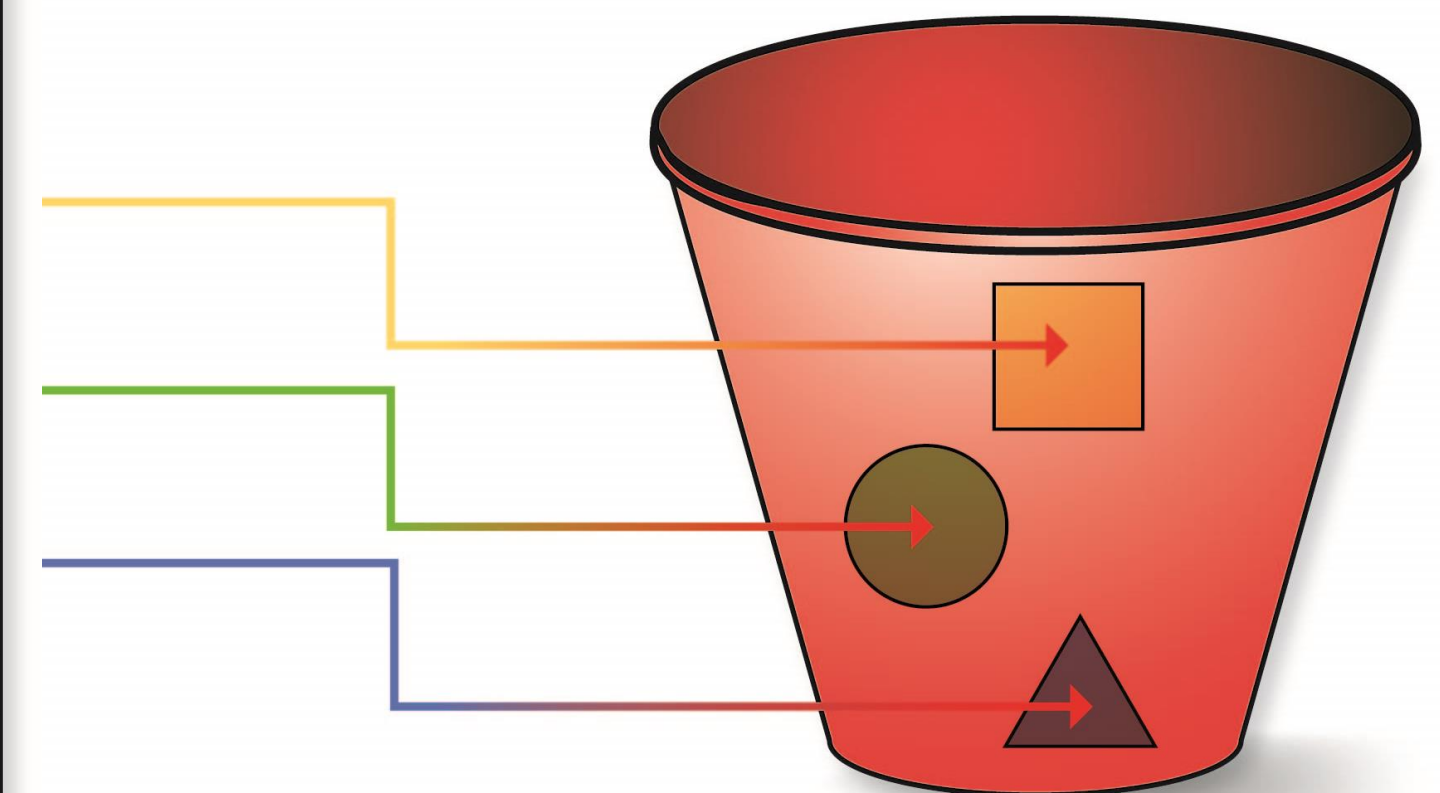
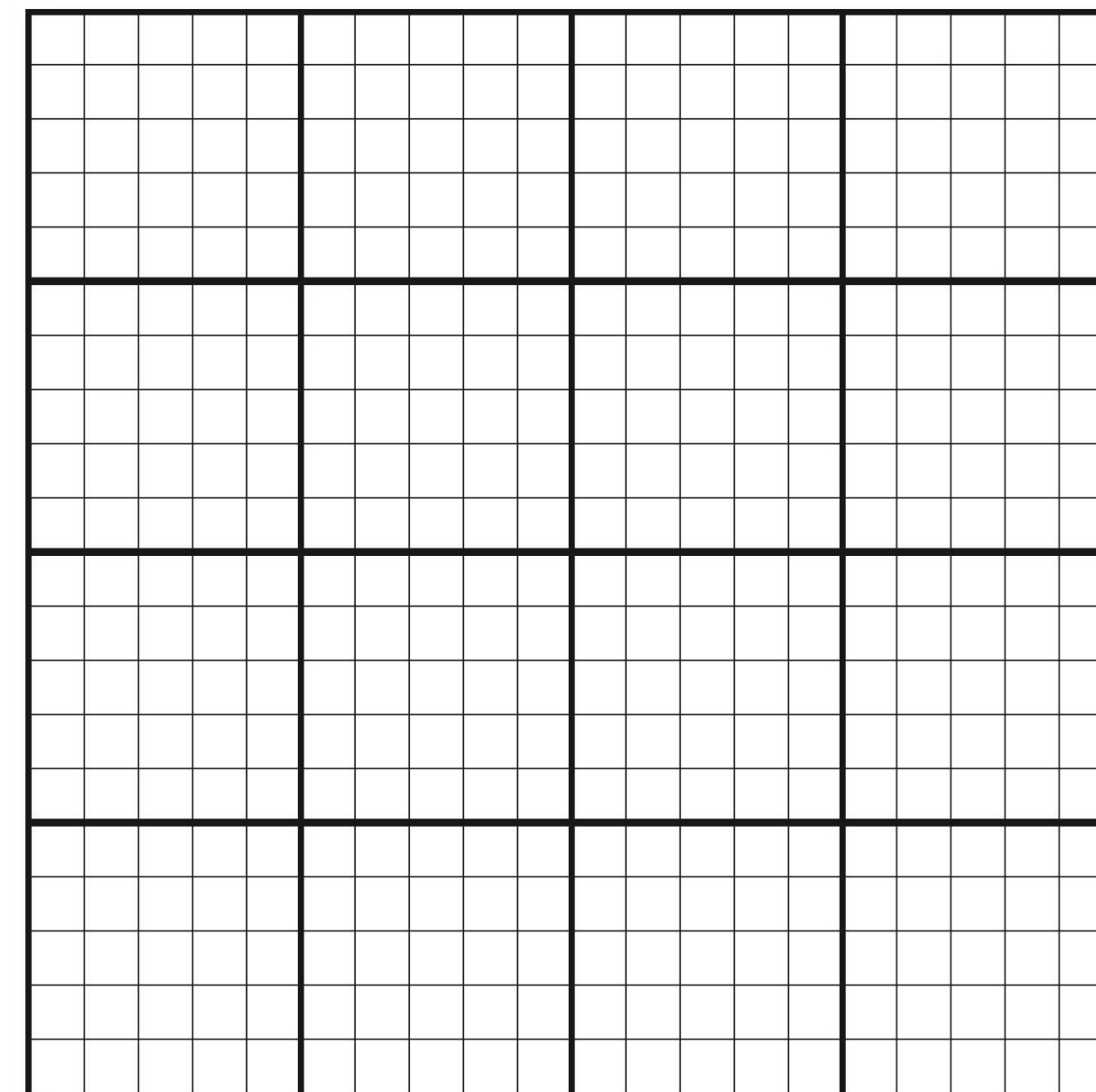
- Limit maximum storage object size
- Support parallelism for read/write
- Only data that is modified needs to be updated
- Multiple clients can be reading/updating the same “file”

Big Idea: Map individual HDF5 objects (datasets, groups, chunks) as Object Storage Objects

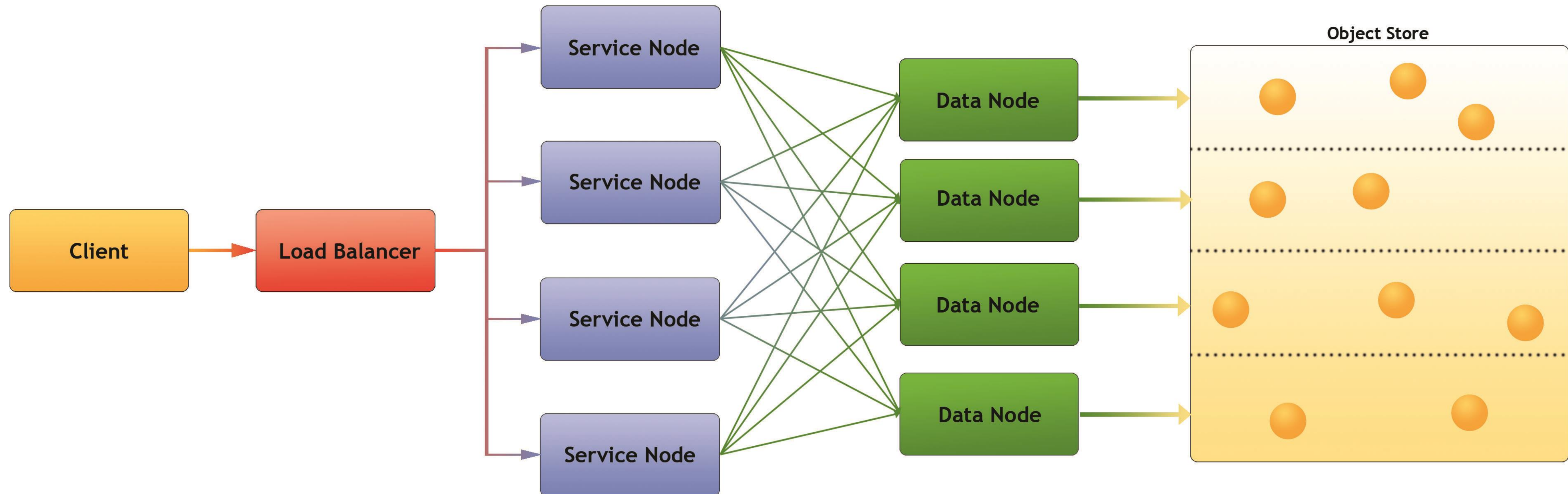
Each chunk (heavy outlines) get persisted as a separate object

Legend:

- *Dataset is partitioned into chunks*
- *Each chunk stored as an S3 object*
- *Dataset meta data (type, shape, attributes, etc.) stored in a separate object (as JSON text)*



Architecture



- Client: Any user of the service
- Load balancer – distributes requests to Service nodes
- Service Nodes – processes requests from clients (with help from Data Nodes)
- Data Nodes – responsible for partition of Object Store
- Object Store: Base storage service (e.g. AWS S3)

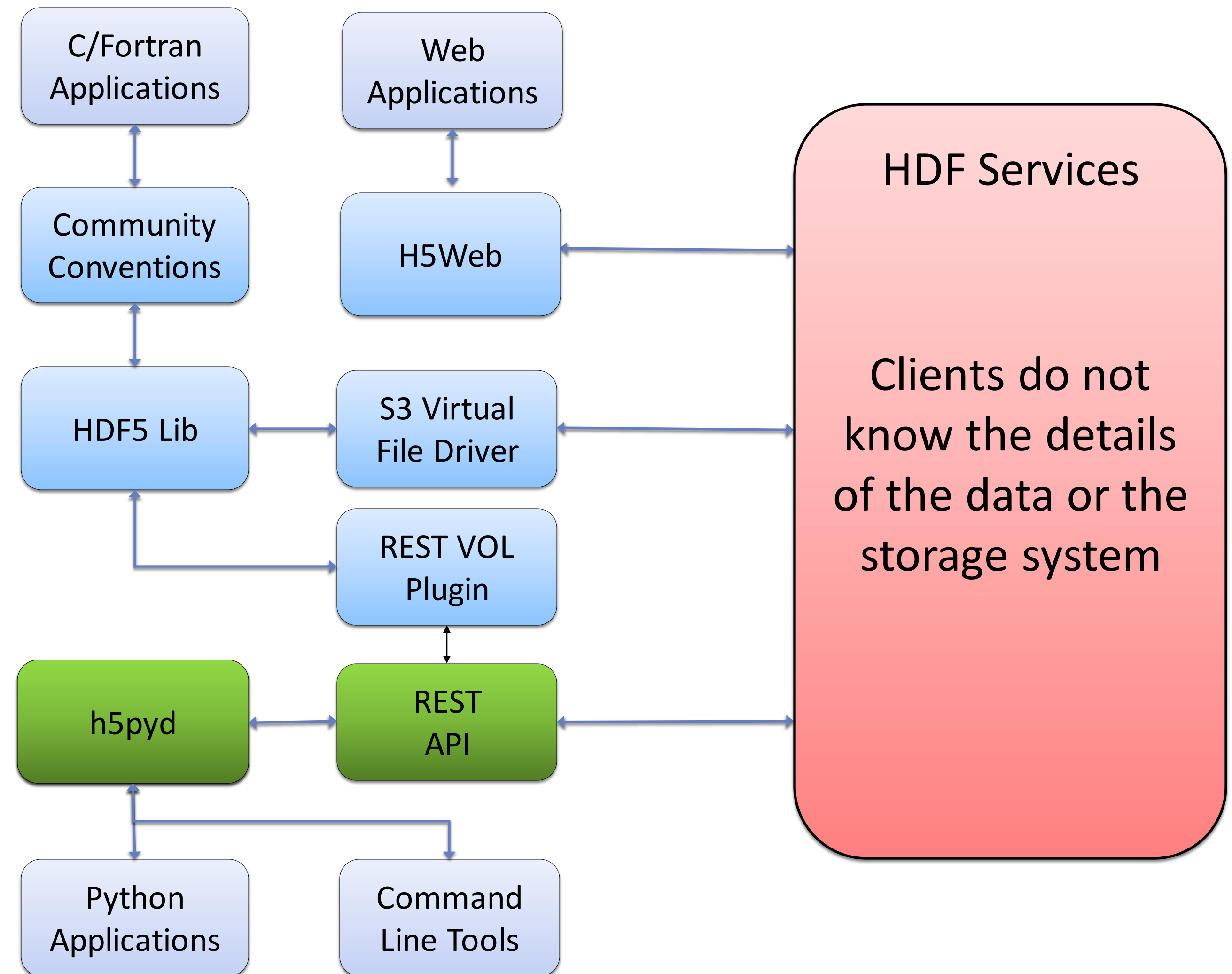
Server Features

- **Simple + familiar API**
 - Clients can interact with service using REST API
 - SDKs provide language specific interface (e.g. h5pyd for Python)
 - Can read/write just the data they need (as opposed to transferring entire files)
 - Support for compression
- **Scalable performance:**
 - Can cache recently accessed data in RAM
 - Can parallelize requests across multiple nodes
 - More nodes → better performance
 - Cluster based – any number of machines can be used to constitute the server
 - Multiple clients can read/write to same data source
 - No limit to the amount of data that can be stored by the service

Architecture

h5pyd for Python and REST VOL plugin are drop-in replacements for libraries used with local files.

No significant code change to access both local and cloud based data.



H5pyd – Python client

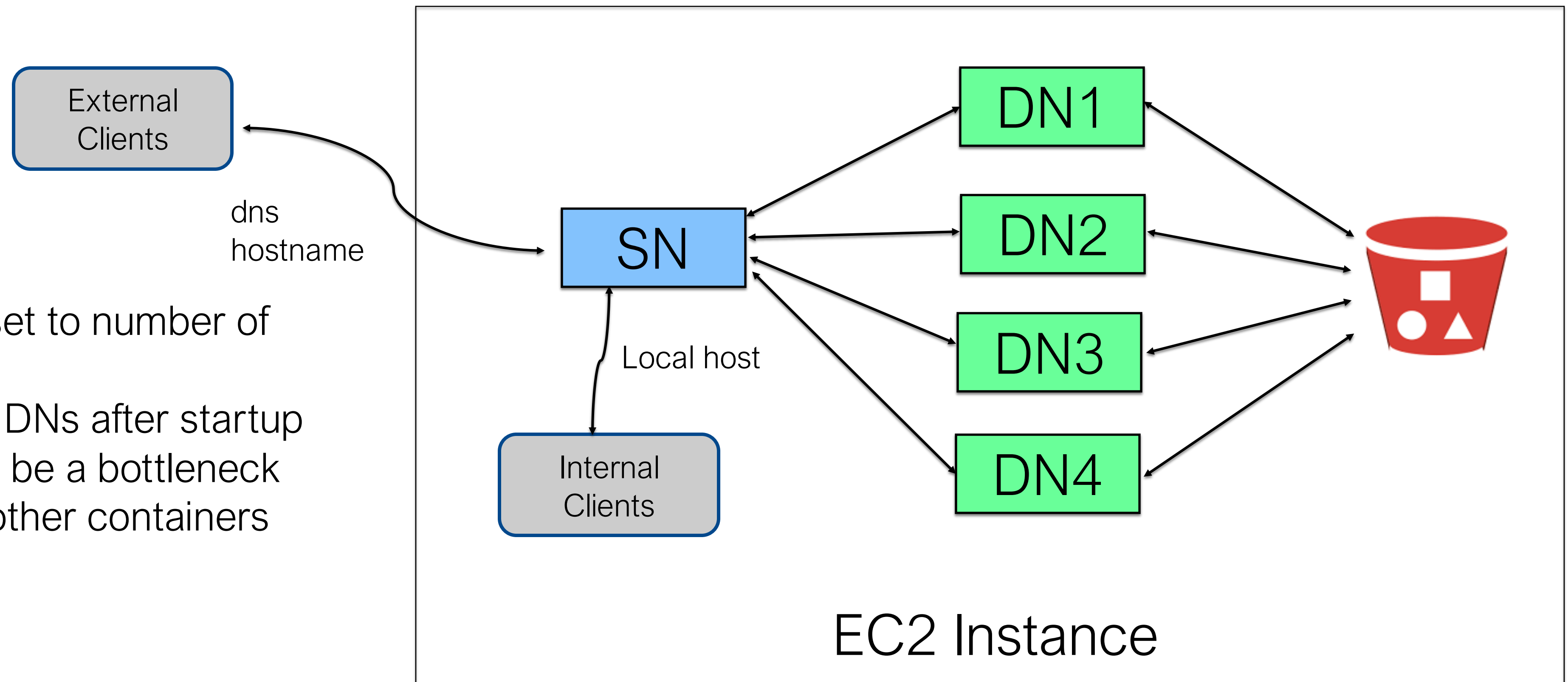
- H5py is a popular Python package that provide a Pythonic interface to the HDF5 library
- H5pyd (for h5py distributed) provides a h5py compatible h5py for accessing the server
- Pure Python – uses requests package to make http calls to server
- Include several extensions to h5py:
 - List content in folders
 - Get/Set ACLs (access control list)
 - Pytables-like query interface
- H5netcdf and xarray packages will use h5pyd when <http://> is prepended to the file path
- Installable from PyPI: \$ pip install h5pyd

HSDS Deployment Types

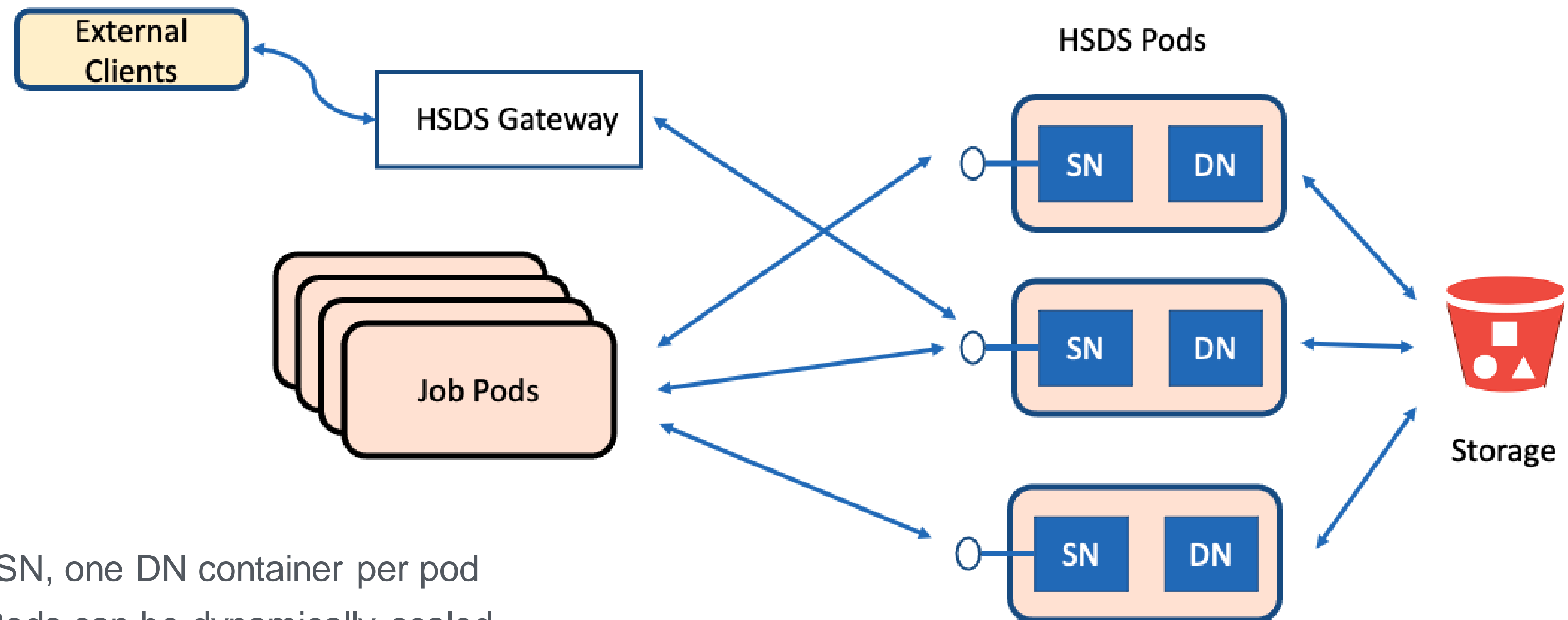
- There's quite a few ways to deploy HSDS:
 - Docker -- +simple to setup, - Limited to one machine
 - Kubernetes -- + dynamically scale, +run in cluster, +complicated to setup
 - AWS Lambda -- + no server, +super elastic, -slow startup
 - H5pyd direct – + no server, -no coordination between apps

HSDS on Docker

- Typically # of DN set to number of cores
- Can't change # of DNs after startup
- Only 1 SN, so can be a bottleneck
- Not easy to have other containers talk to SN node



HSDS On Kubernetes

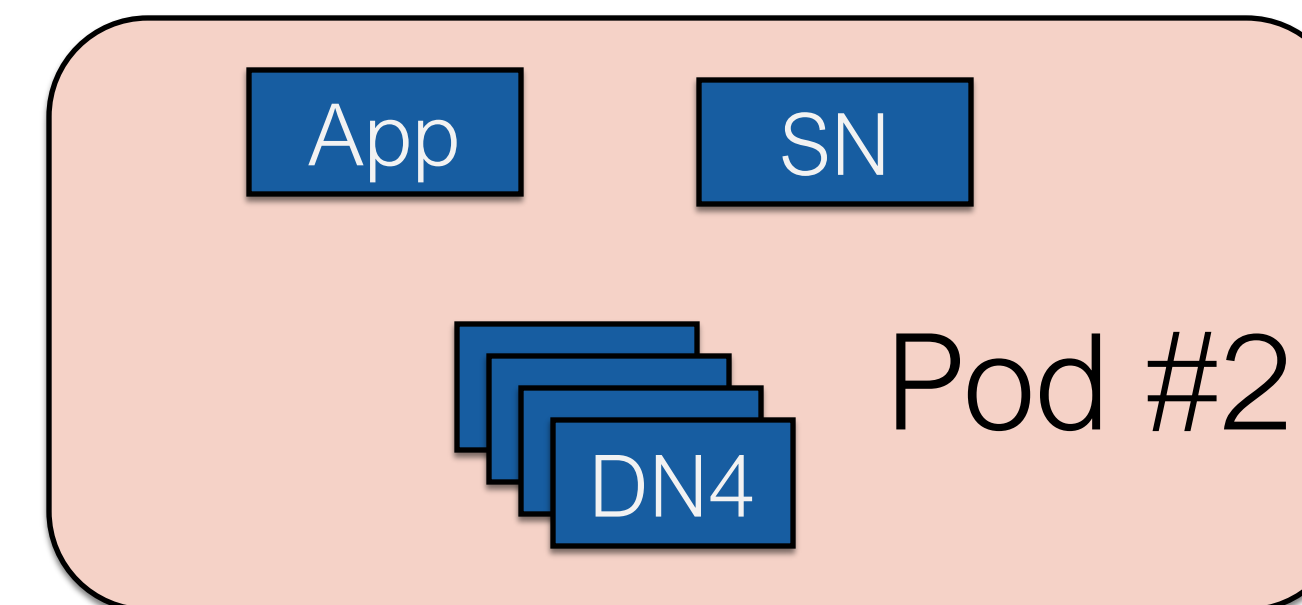
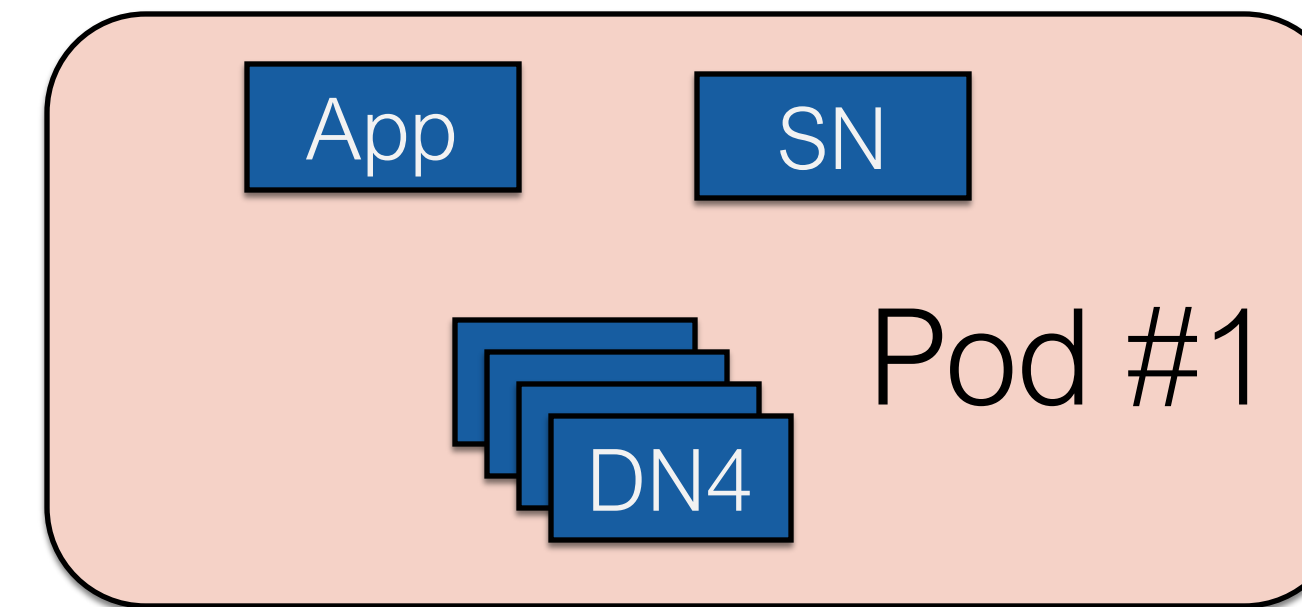


- HSDS Pods – one SN, one DN container per pod
- Number of HSDS Pods can be dynamically scaled
- Kubernetes Job pods (user application) can be scaled to increase throughput
- Ability to run more pods can be scaled by increasing number of servers in Kubernetes cluster
- All pods agnostic as to which machine they are running on

Ability to scale up and down is improved in v0.7!

HSDS on Kubernetes Alternative Deployment

- Configure each pod with containers for app, SN, DNs
- App just uses localhost endpoint to talk to SN
- SN just talks to DNs in its pod
- No worries about scaling HSDS
- But DN's in different pods won't coordinate (may overwrite each other – need to have app pods write to different chunks)



Etc...

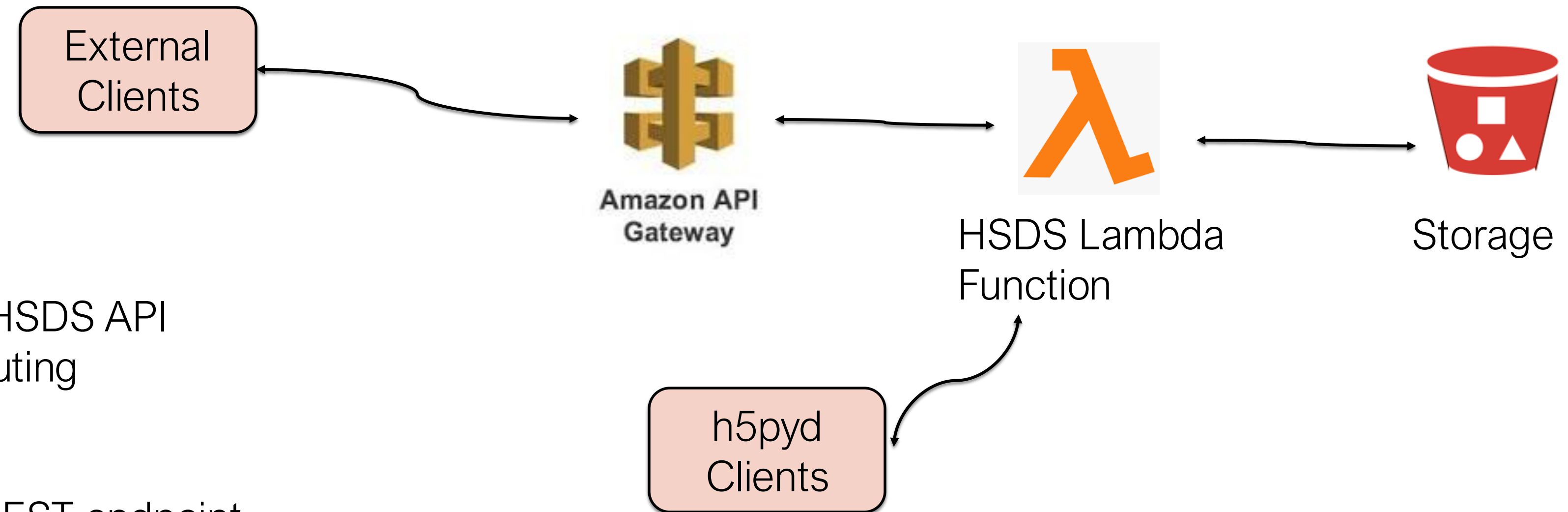
HSDS on Lambda

The Good

- Lambda function provides complete HSDS API
- Pay just for time function spends executing
- Up to 1000 simultaneous invocations
- H5pyd can invoke lambda directly
- API Gateway can be used to provide REST endpoint

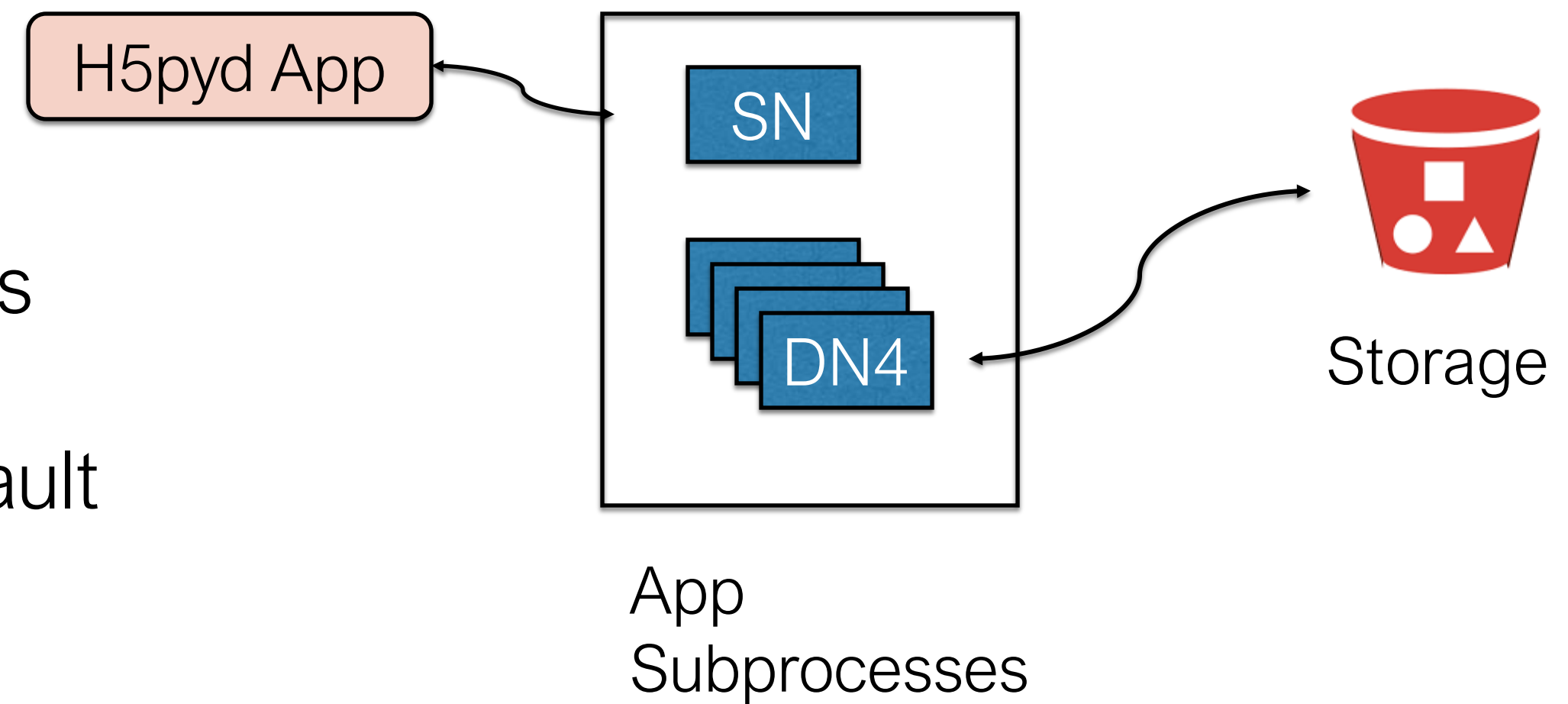
The Bad

- Startup cost per invocation (up to 2 sec)
- Limitations on size of req/response
- Maximum run time
- Only JSON supported (no binary req/response)

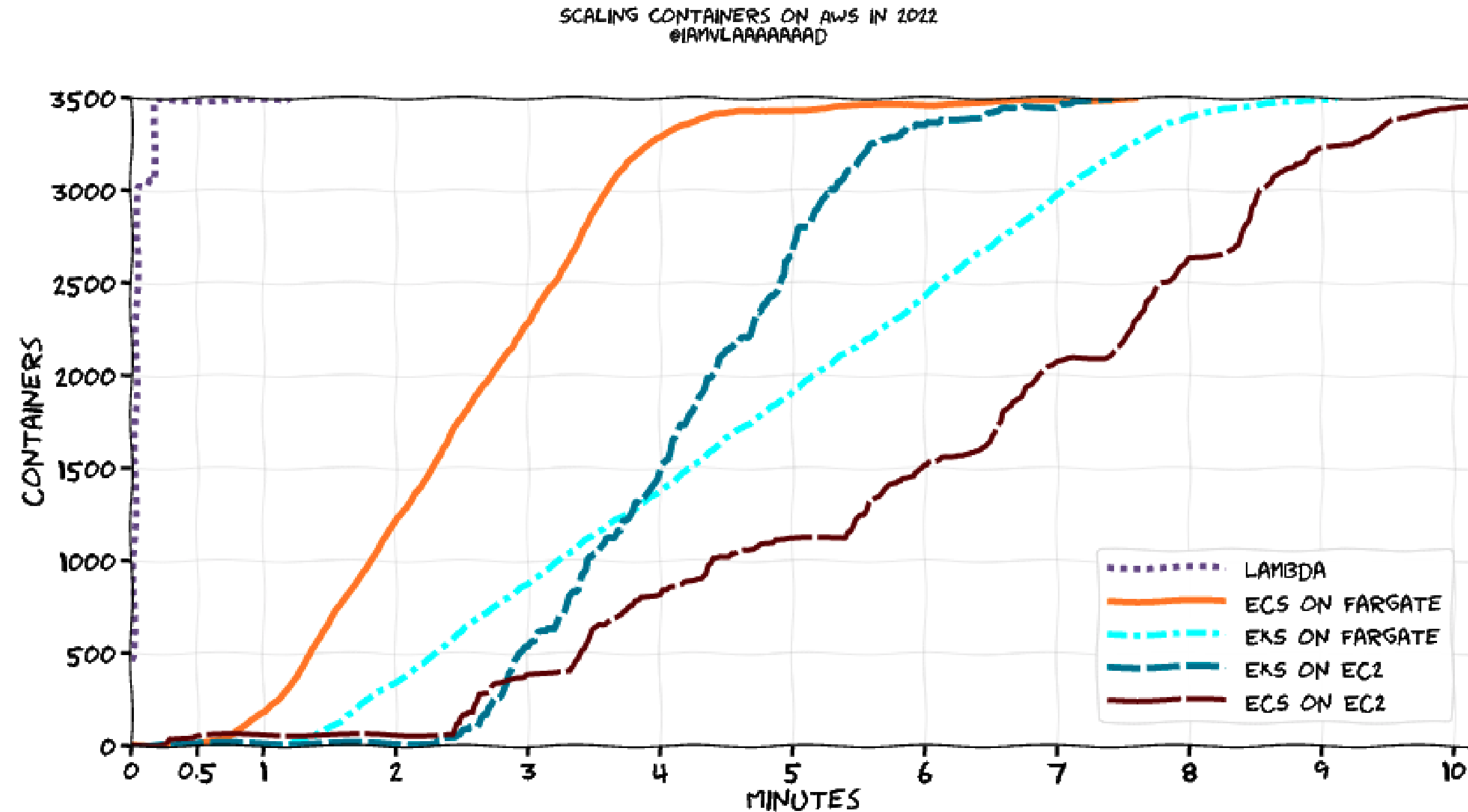


H5pyd Direct

- In direct mode, h5pyd instantiates SN/DN nodes as sub-processes on File open
- Number of DN nodes can be specified (will default to # of cores)
- On file close, subprocesses are terminated
- Some overhead setting up subprocesses, not a good choice if app will be doing lots of file opens



When to use Lambda?



Use Lambda when handling quickly changing workloads or sporadic usage.
(otherwise you'll be better off with EC2 or Kubernetes)

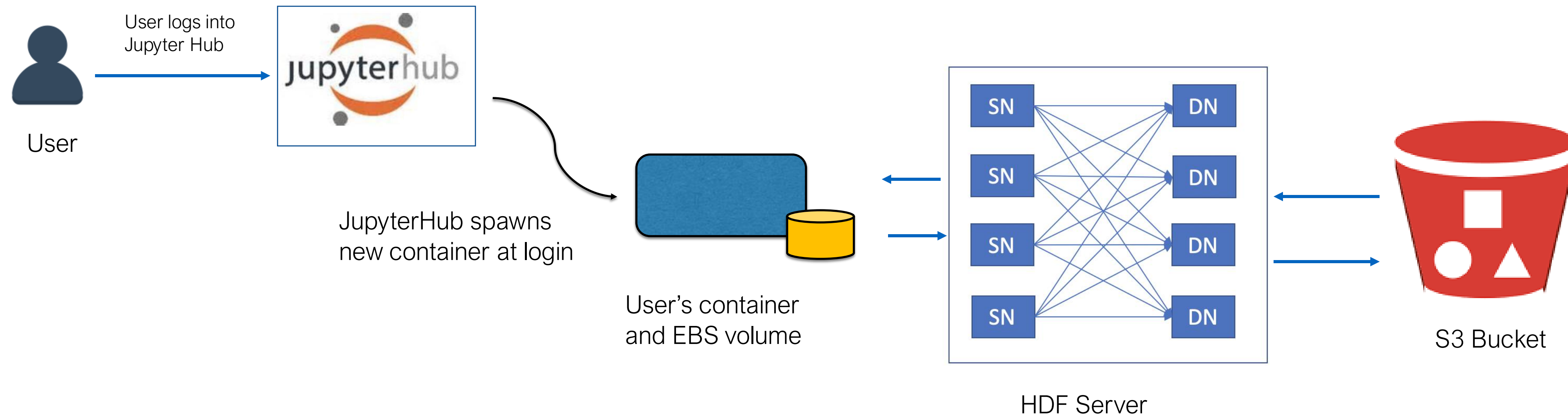
Streaming Support

- Previously HSDS had limits on the number of chunks that could be accessed in one request and the number of bytes that could be read or written (server would send a 413 – Payload too large response)
- Problem was the HSDS would need to allocate memory buffer to store the response (or request)
- Too large of request -> out of memory death!
- In v0.7, HSDS now internally paginates larger request
- So:
 - No limit on the size of a binary request (+1GB/req is fine)
 - Memory load on server is managed
 - Client will start getting bytes back when the server is still processing latter chunks

Fancy Indexing

- Fancy indexing enables dataset selections based on start+stop+stride or set of indexes
- Example. `N(0:100, (1,4,63,92))`
- In v0.7 fancy indexes can be done in one request vs (a series of hyperslab selections)
- Some selections see 8x speed increase vs getting on index per request
- Example: retrieving 4000 random columns from a 17520 by 2018392 dataset demonstrating good scaling as the number of HSDS nodes was increased:
 - 4 nodes: 65.4s
 - 8 nodes: 35.7s
 - 16 nodes: 23.4 s

- HDF Lab is a JupyterLab and HDF server environment hosted by the HDF Group on AWS
- HDF Lab users can create Python notebooks that use h5pyd to connect to HDF Server
- Each user gets equivalent to 2-core Xeon Server and 10GB local storage
- Users can use up to 100GB of data on HDF Server



Case Study - NREL

NREL (National Renewable Energy Laboratory) uses HDF Cloud to make PB's of environmental data accessible to the public.

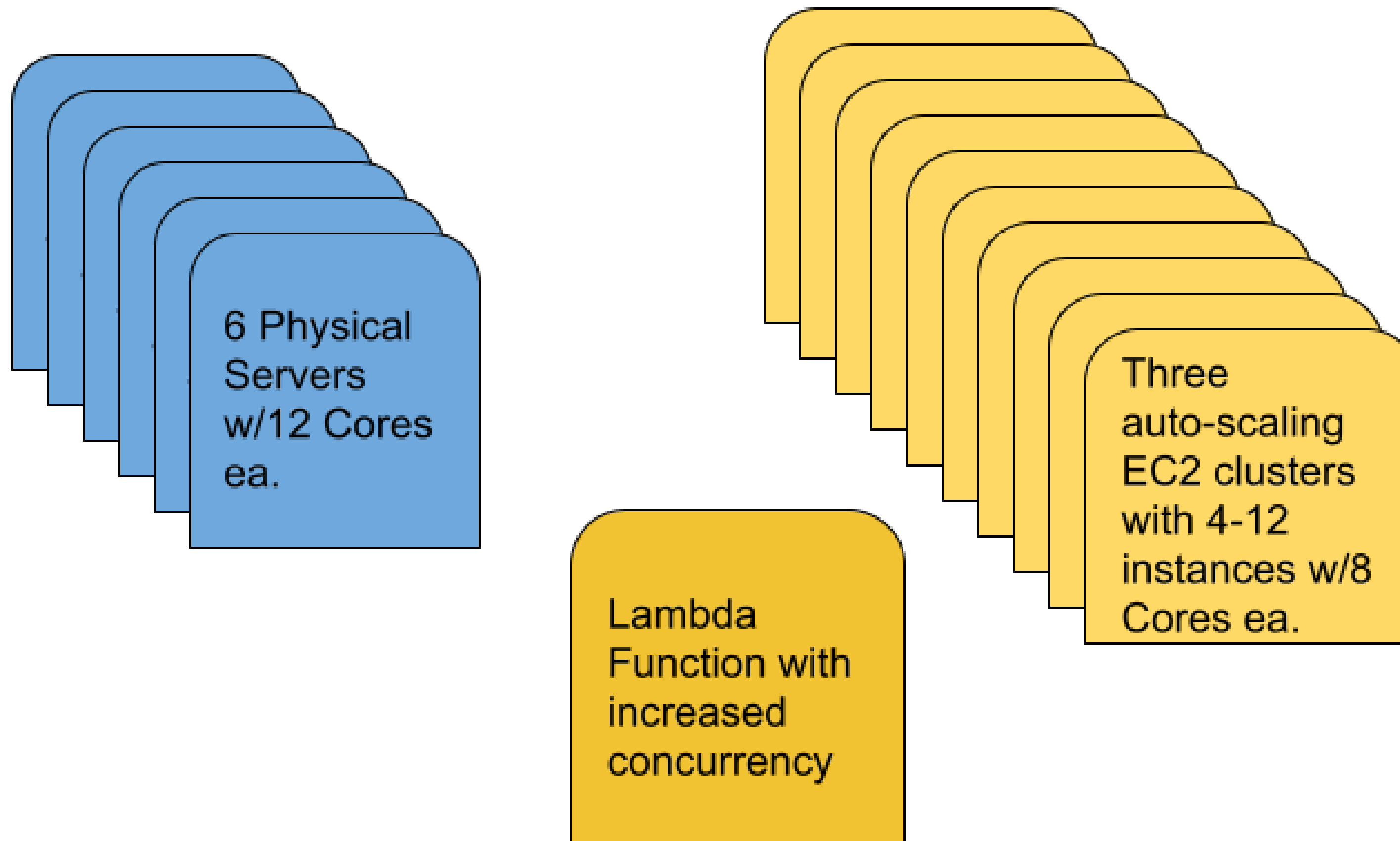
HSDS enables users to slice and dice just the data they need (say given time range, or geo range)

Use of HSDS has expanded over the last 5 years

NREL has started to use the HSDS Lambda functions (request load is highly variable driven mostly by external uses)



NREL Data Download API - supporting HSDS installations



Case Study - BioSimulations

Using HSDS for biological simulations with Vega for visualizations

Running on HSDS with Kubernetes

Read the blog article at:

<https://www.hdfgroup.org/2022/02/biosimulations-a-platform-for-sharing-and-reusing-biological-simulations/>

Or attend Jonathan's presentation tomorrow!

Case Study – NSF UVA-ARC

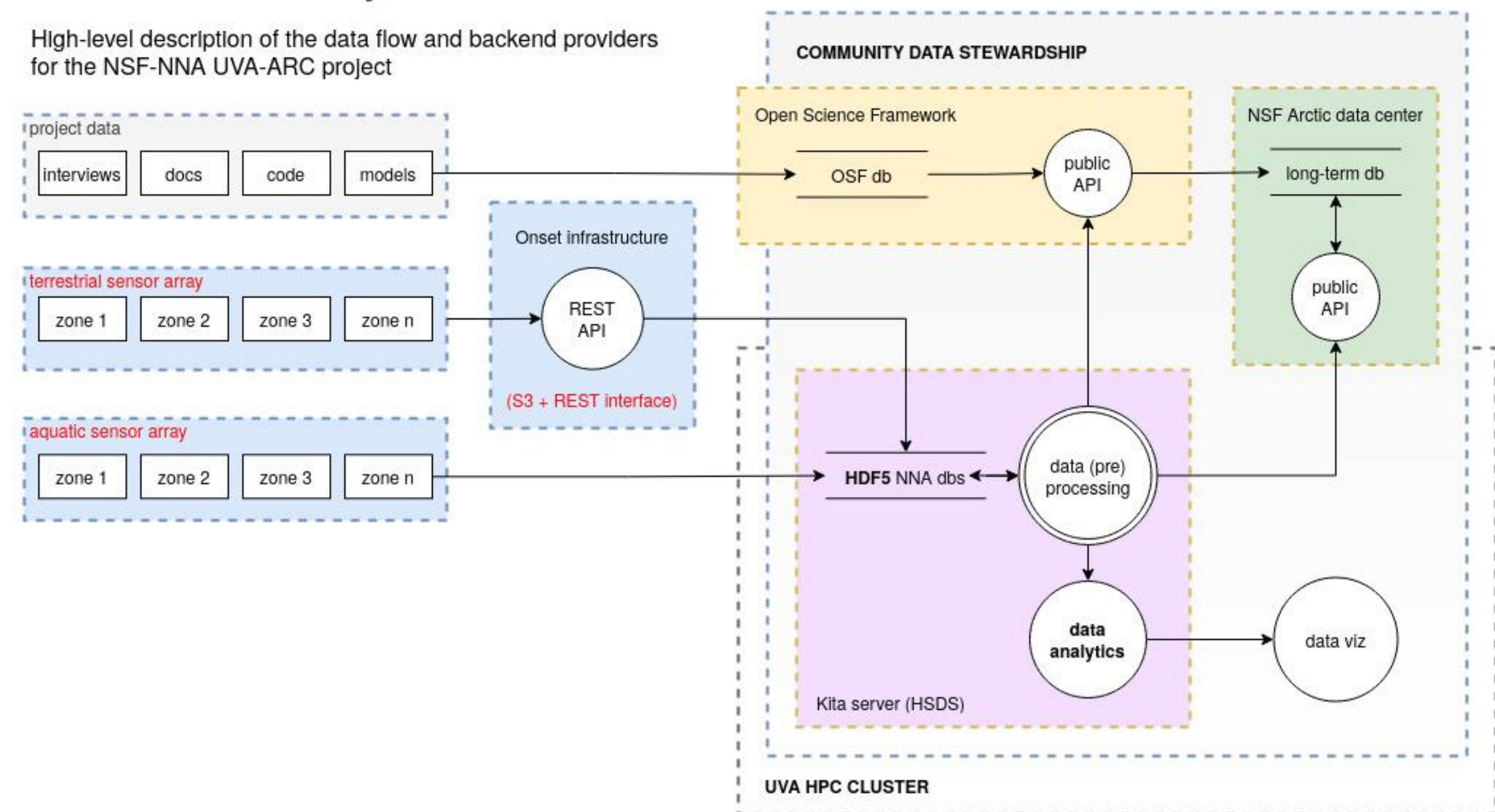
- NSF Project to collect environmental data
- Data is streamed from hundreds of sensors to HSDS
- Have collected two years of data from testbed site at U. of Virginia
- North Alaska data collection starts this week!



Figure 1. Utqiagvik, Alaska, with the four research target areas highlighted: 1) Town center, which is identified as a region of possible infilling and development 2) Shoreline interface, which could undergo some development and is a region of particular vulnerability to climate change impacts, such as sea-ice reduction, coastal erosion, and sea level rise 3) Lagoon-tundra urban edge, where there is an existing gradient from dense urban to tundra ecosystems, and may undergo significant development by 2035 and 4) Tundra-sewage lagoons, which targets important infrastructure and a transect from coast to tundra

NSF UVA-ARC: Understanding the Changing Natural-Built Landscape in an Arctic Community

High-level description of the data flow and backend providers for the NSF-NNA UVA-ARC project



References

- HSDS Design Doc:
https://github.com/HDFGroup/hsds/blob/master/docs/design/hsds_arch/hsds_arch.md
- HSDS Design Schema:
https://github.com/HDFGroup/hsds/blob/master/docs/design/obj_store_schema/obj_store_schema_v2.md
- SciPy2017 talk:
https://s3.amazonaws.com/hdfgroup/docs/hdf_data_services_scipy2017.pdf
- AWS Big Data Blog article: <https://aws.amazon.com/blogs/big-data/power-from-wind-open-data-on-aws/>