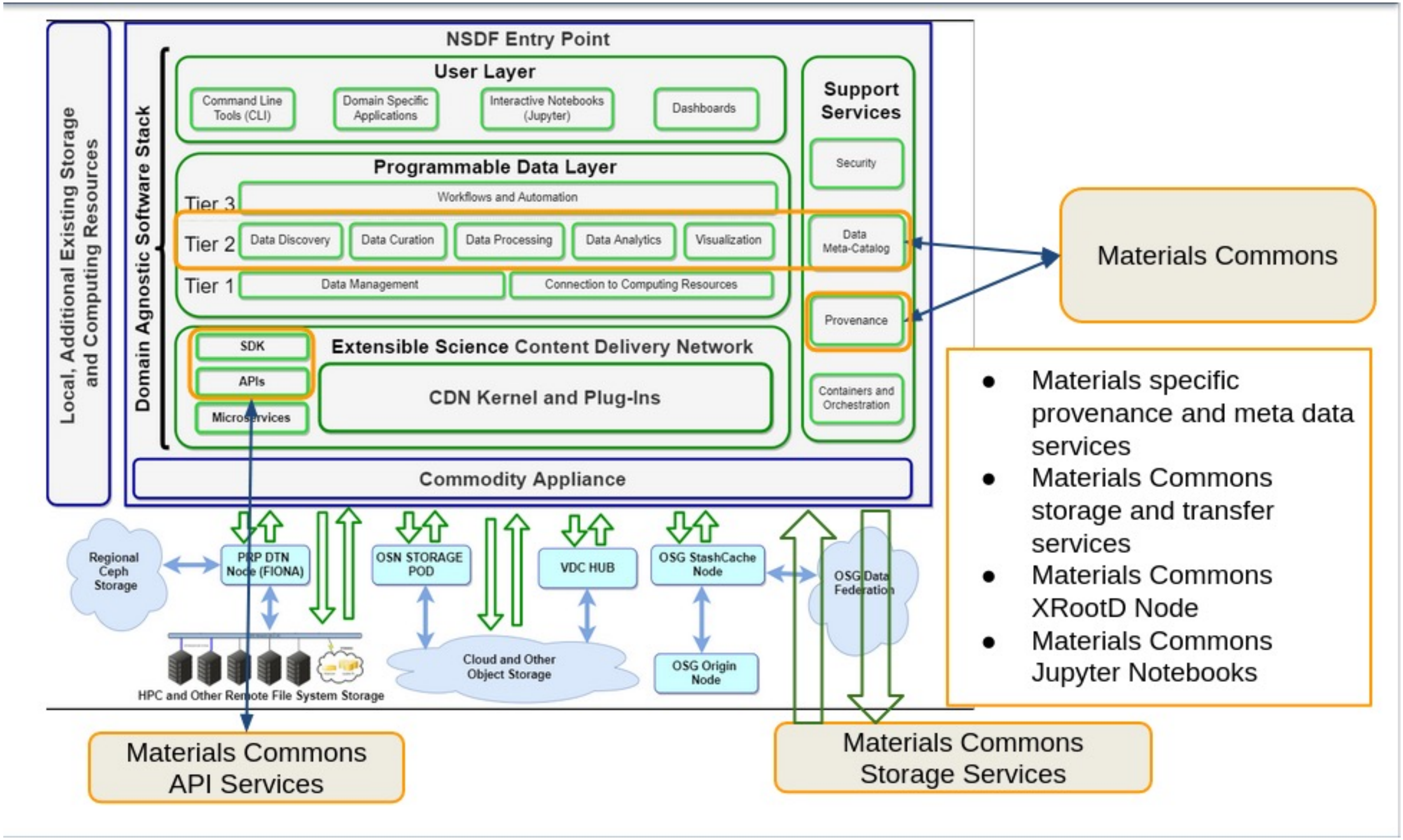
The SCI Institute

Scientific Computing and Imaging Institute

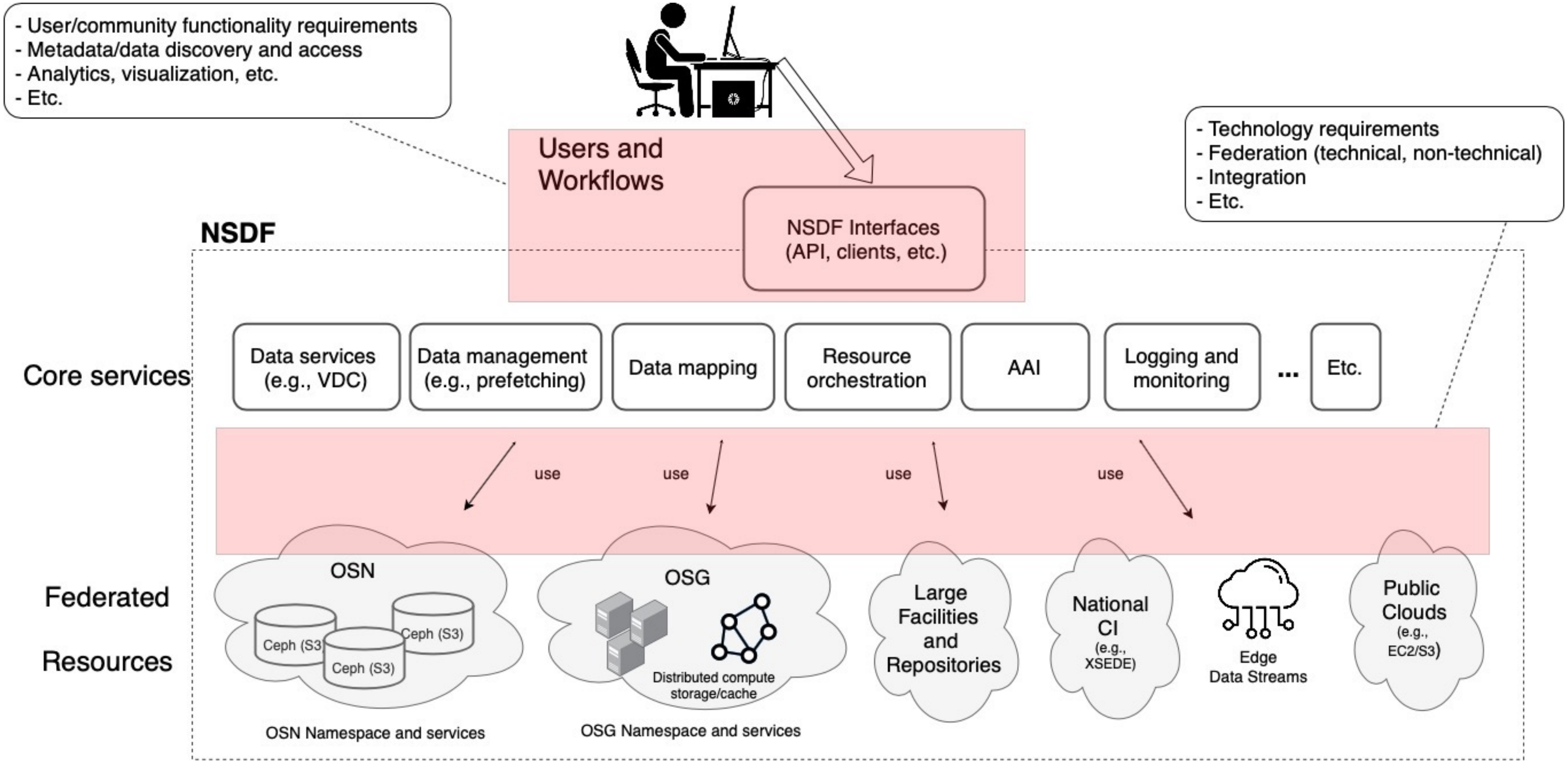
NSDF AHM, February 2022

NSDF Architecture



- Materials specific provenance and meta data services
- Materials Commons storage and transfer services
- Materials Commons XRootD Node
- Materials Commons Jupyter Notebooks

Layered Architecture



- User/community functionality requirements
- Metadata/data discovery and access
- Analytics, visualization, etc.
- Etc.

- Technology requirements
- Federation (technical, non-technical)
- Integration
- Etc.

Users and Workflows

NSDF Interfaces (API, clients, etc.)

NSDF

Core services

Data services (e.g., VDC) | Data management (e.g., prefetching) | Data mapping | Resource orchestration | AAI | Logging and monitoring | ... | Etc.

Federated Resources

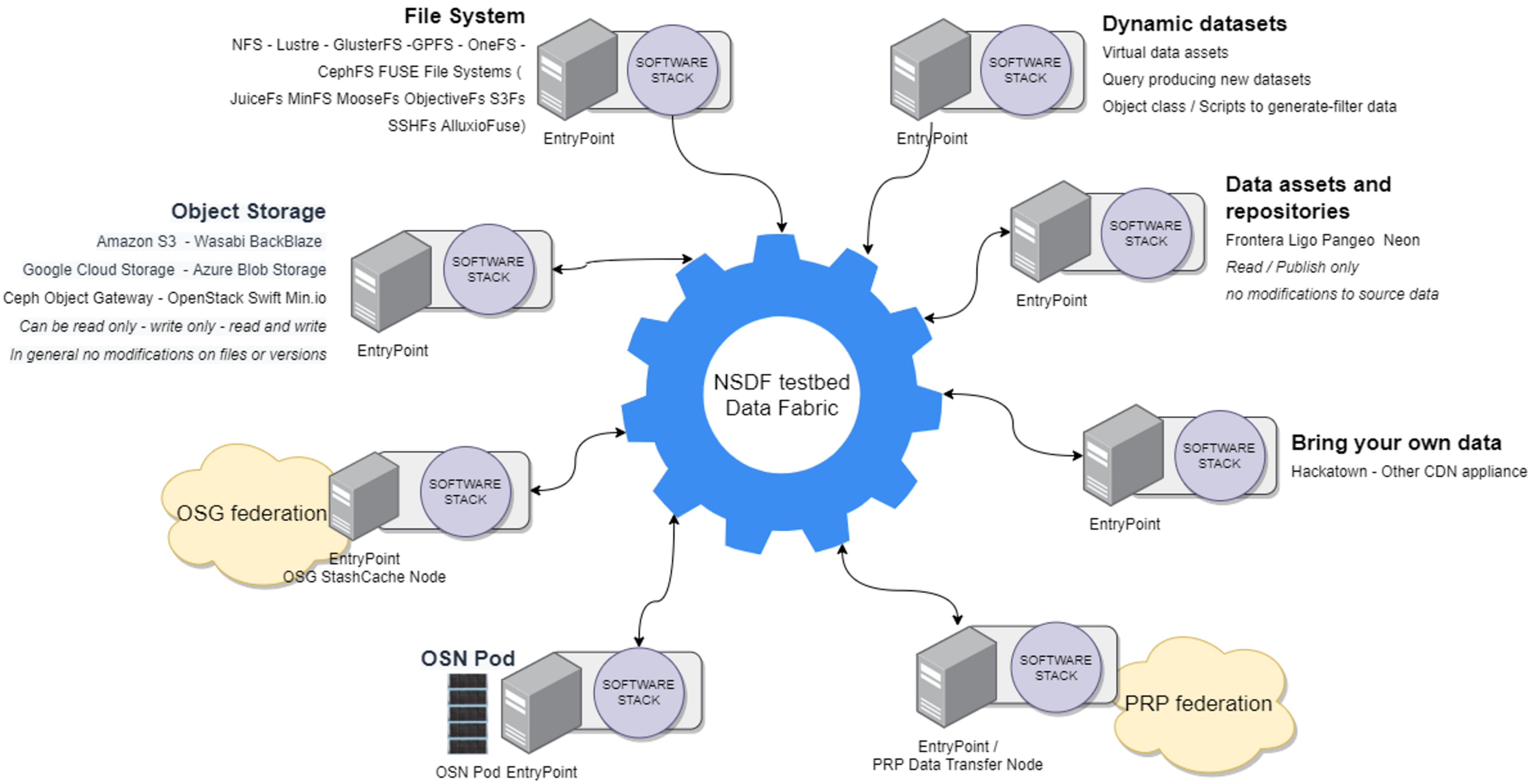
OSN Namespace and services

OSG Namespace and services

use
use
use
use
OSN
Ceph (S3)
Ceph (S3)
Ceph (S3)
OSG
Distributed compute storage/cache
Large Facilities and Repositories
National CI (e.g., XSEDE)
Edge Data Streams
Public Clouds (e.g., EC2/S3)

*Proposal

NSDF Data Fabric - Origins



File System

NFS - Lustre - GlusterFS -GPFS - OneFS -
CephFS FUSE File Systems (
JuiceFs MinFS MooseFs ObjectiveFs S3Fs
SSHFs AlluxioFuse)

Dynamic datasets

Virtual data assets
Query producing new datasets
Object class / Scripts to generate-filter data

Object Storage

Amazon S3 - Wasabi BackBlaze
Google Cloud Storage - Azure Blob Storage
Ceph Object Gateway - OpenStack Swift Min.io
Can be read only - write only - read and write
In general no modifications on files or versions

Data assets and repositories

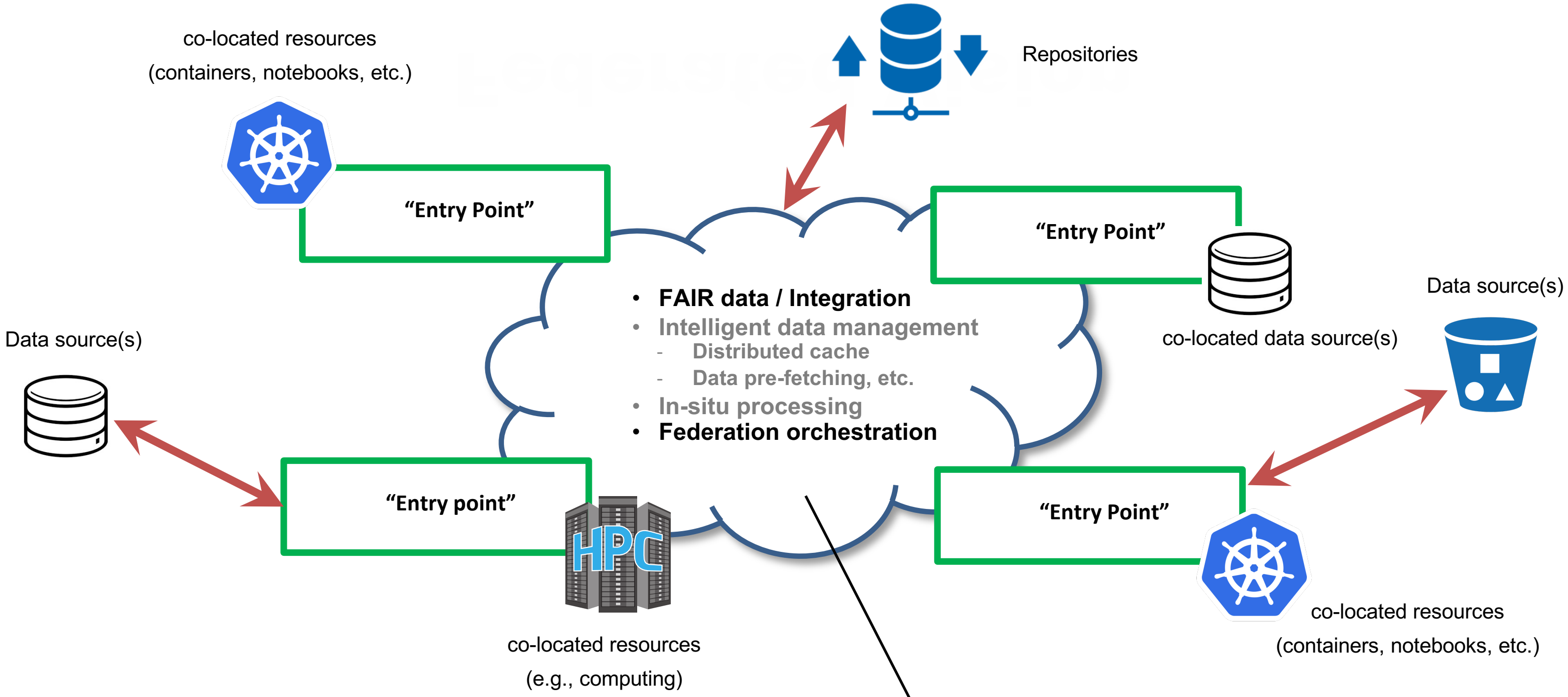
Frontera Ligo Pangeo Neon
Read / Publish only
no modifications to source data

Bring your own data

Hackatown - Other CDN appliance

OSN Pod

Federated Vision



co-located resources
(containers, notebooks, etc.)

"Entry Point"

Repositories

"Entry Point"

co-located data source(s)

Data source(s)

- FAIR data / Integration
- Intelligent data management
  - Distributed cache
  - Data pre-fetching, etc.
- In-situ processing
- Federation orchestration

Data source(s)

"Entry point"

co-located resources
(e.g., computing)

"Entry Point"

co-located resources
(containers, notebooks, etc.)

- Data storage, discovery and sharing
- Workflow composition
- Simplify/reduce "entry cost" (democratization)

- Resource discovery and monitoring
  - Data, computing, network, etc.
- SLA, brokering system, etc.
- **"Better" outcome as the system learns!**

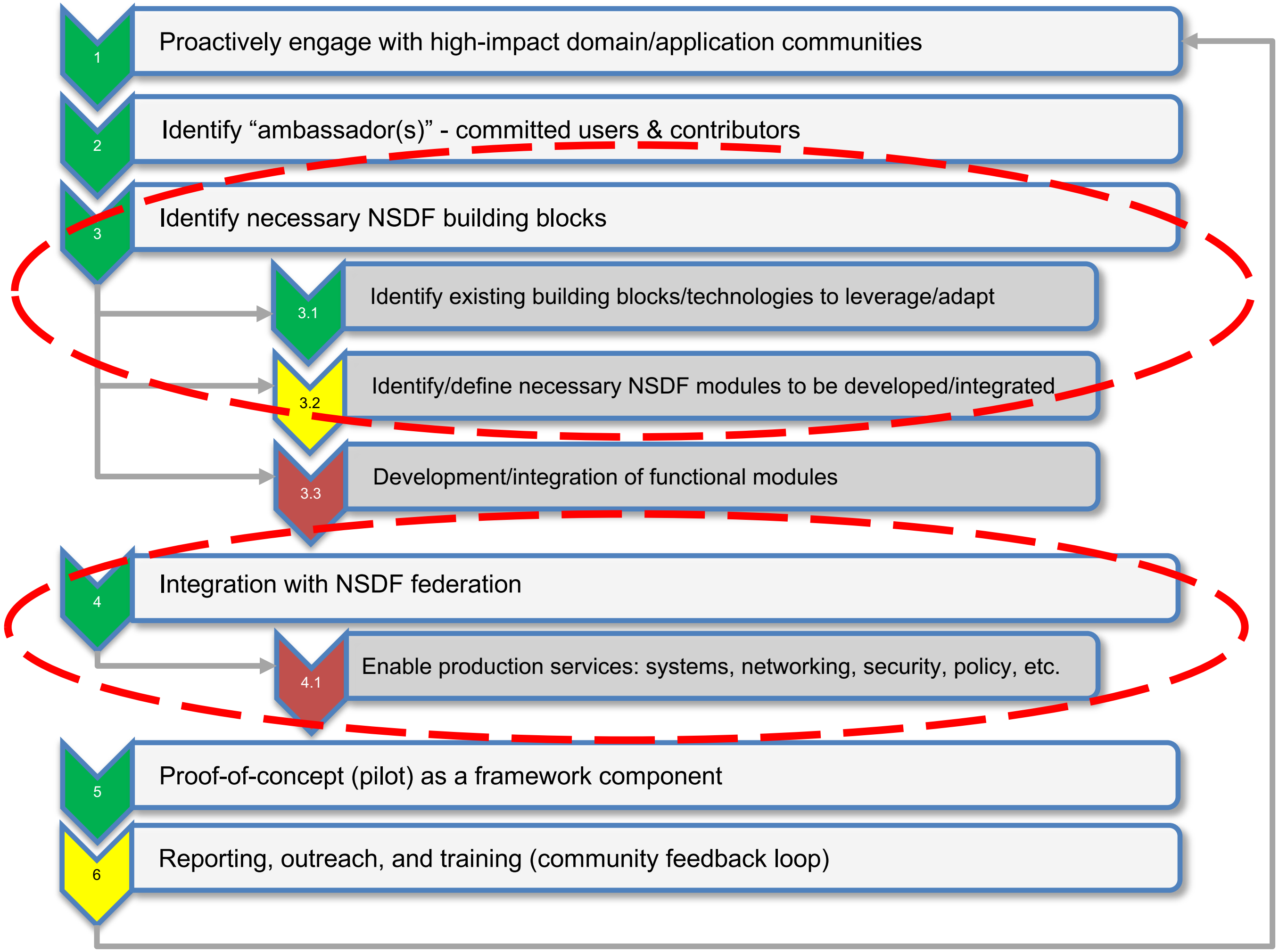
*Proposal

CI Architecture – Year 1

- **Federated data access**
  - OSN, OSG, XSEDE, Material Commons, Commercial Clouds
  - POSIX vs Object storage APIs: unified model? (file vs object, bucket vs directory)
  - In memory metadata caching? (e.g. Redis)
  - Namespace mapping
- **Federated data discovery**
  - (Meta)data services (e.g., object repository)
  - Based on NSDF API (e.g., initial implementation leveraging VDC data services)
- **NSDF Compute Entry Points** based on
  - virtual appliances vs physical rack (OSN)
  - Core services
  - Hardware specifications

*Proposal

Process



1 Proactively engage with high-impact domain/application communities

2 Identify “ambassador(s)” - committed users & contributors

3 Identify necessary NSDF building blocks

3.1 Identify existing building blocks/technologies to leverage/adapt

3.2 Identify/define necessary NSDF modules to be developed/integrated

3.3 Development/integration of functional modules

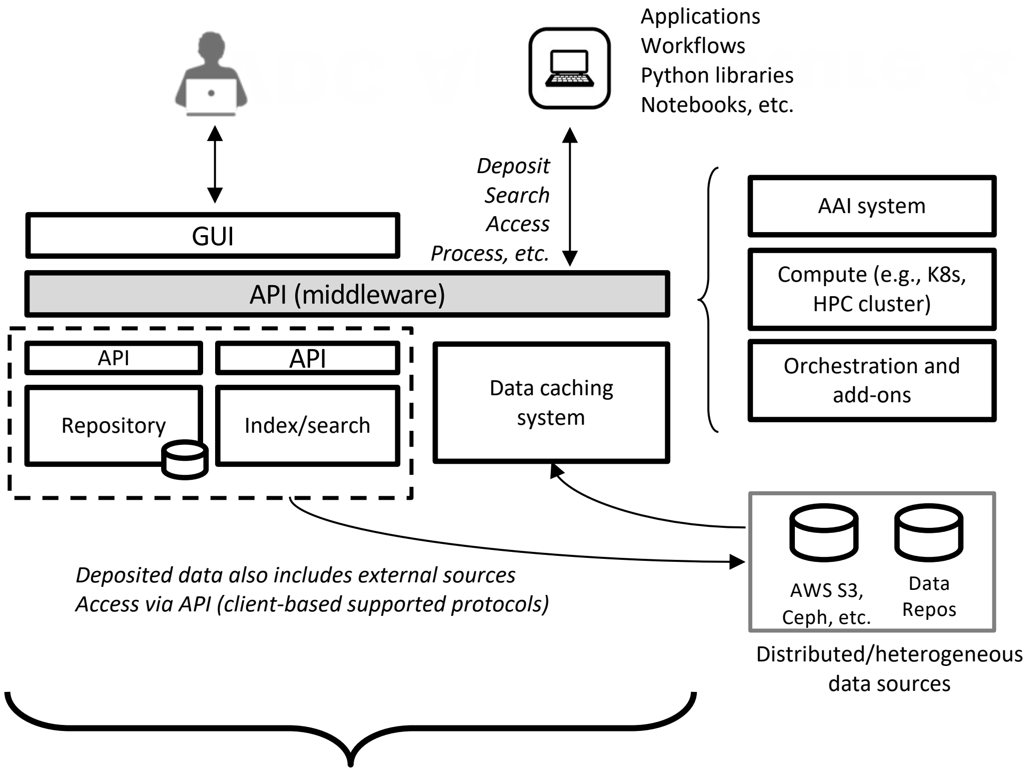
4 Integration with NSDF federation

4.1 Enable production services: systems, networking, security, policy, etc.

5 Proof-of-concept (pilot) as a framework component

6 Reporting, outreach, and training (community feedback loop)

VDC Architecture & Data Services


Applications
Workflows
Python libraries
Notebooks, etc.
Deposit
Search
Access
Process, etc.
GUI
API (middleware)
API
API
Repository
Index/search
Data caching system
AAI system
Compute (e.g., K8s, HPC cluster)
Orchestration and add-ons
AWS S3, Ceph, etc.
Data Repos
Deposited data also includes external sources
Access via API (client-based supported protocols)
Distributed/heterogeneous data sources

Part of the software stack (entry points)

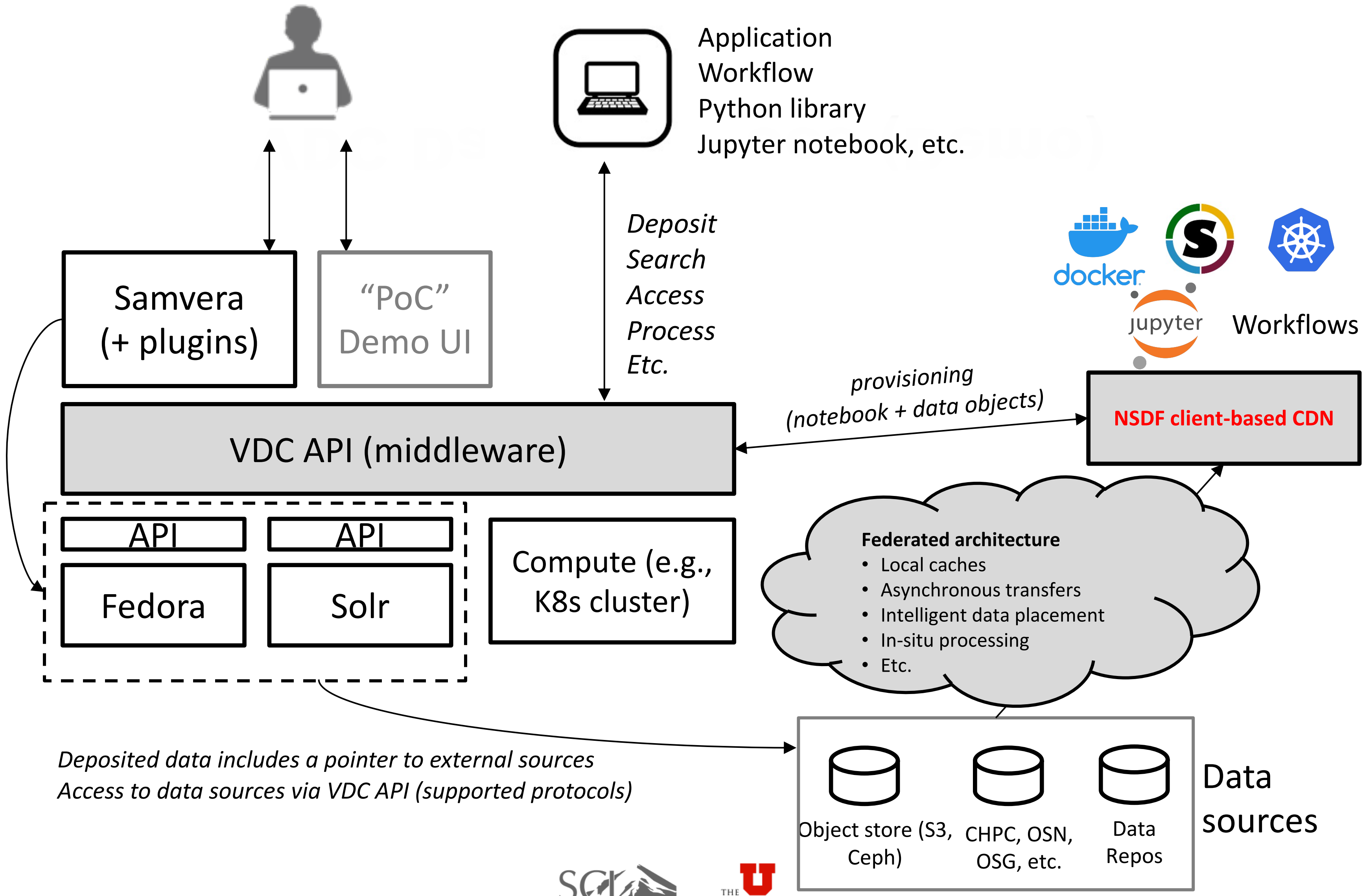
Capabilities

- Provide tools and services to work with datasets, data products, etc.
- Register objects (collections/files/links)
- Discovering and sharing
- Store/edit metadata
- Create PIDs/DOI
- Deriving data, storing provenance
- AAI integration

Key features

- FAIR data technology stack
- Provides content as linked data (RDF)
- Semantic support: data and metadata using any ontologies and vocabularies
- Advanced search: content indexed into an index/search engine
- Advanced query: integration with triple store applications (e.g., Jena Fuseki) - SPARQL query language support

VDC Data Services (Demo)



Application
Workflow
Python library
Jupyter notebook, etc.

Deposit
Search
Access
Process
Etc.

Samvera
(+ plugins)

"PoC"
Demo UI

VDC API (middleware)

API
Fedora

API
Solr

Compute (e.g.,
K8s cluster)

Workflows

provisioning
(notebook + data objects)

NSDF client-based CDN

Federated architecture

- Local caches
- Asynchronous transfers
- Intelligent data placement
- In-situ processing
- Etc.

Object store (S3,
Ceph)

CHPC, OSN,
OSG, etc.

Data
Repos

Data
sources

Deposited data includes a pointer to external sources
Access to data sources via VDC API (supported protocols)

CI Architecture Discussion (1)

- Goals (*Proposal)
  - Testbed for data fabric research
  - Production-level capability for user communities
- Requirements: how do we build it / integrate building blocks?
  - Data and metadata search/access
  - Data sources and repositories integration
    - Immutable vs. mutable data
    - File/directory vs. data streams
  - Computing provision/capabilities (e.g., K8s, HPC)
  - Networking: major fabrics, comprehensive (e.g., perfsonar-like) monitoring?
  - Cybersecurity: AAI (SSO, OpenID, CILogon integration?), cyber-security plan (TrustedCI)?
  - Integration with services, middleware, science gateways, virtual labs
    - Initial pilots + Globus, National Data Service (workbench), Hubzero, Airavata, etc.
- Gaps
  - Heterogeneity in protocols, available capabilities (e.g., object access vs. available rich interfaces, authentication/authorization mechanisms)
    - E.g., AWS S3 bucket, xrootd, Materials Commons, etc.
  - Different maturity levels (for integration)

CI Architecture Discussion (2)

- Challenges
  - Different mechanisms for implementing the same capabilities (e.g., access a dataset/collection)
    - Model: client based vs. CDN-like (e.g., AWS CloudFront)
    - How do we handle this?: multiple clients, standardized mechanisms/protocols, etc.
  - Orchestration (scalability) model
    - Regional/community based + federation (smaller entry points talk each other)?
  - Metadata in a federated ecosystem
    - Registration/deposit (e.g., semi-automated via APIs), metadata harvesting (e.g., OAI-PMH), etc.
    - Interoperability across domains, semantic queries, etc.
  - AAI + Accounting
    - User-based, community-based?
    - Accounting and SLA (storage, transfer, and computing)
  - Optimizations
    - Usability (e.g., "time to science"), minimize/hide latency, resources consumption, etc.
- What are the limitations/trade-offs for a successful pilot?

CI Architecture Discussion (3)

1) Integrate robust, well-proven technologies (all layers)

2) VDC architecture integration
   - Federated data collaboration middleware
     - Model, APIs, services

3) Innovative services and new workflow models
   - Intelligent data discovery and access
     - User-based data caching and pre-fetching, recommender system, etc.
   - Management of long-running computations based on new data products or trends
     - Containerized agent for virtual and physical resources
     - Exposes data producers and computing resources through dynamic profiles (R-Pulsar)
     - Orchestrates computations based on user-defined rules
     - From the edge to the core, etc.

NSDF

Thank you

ivan.rodero@utah.edu