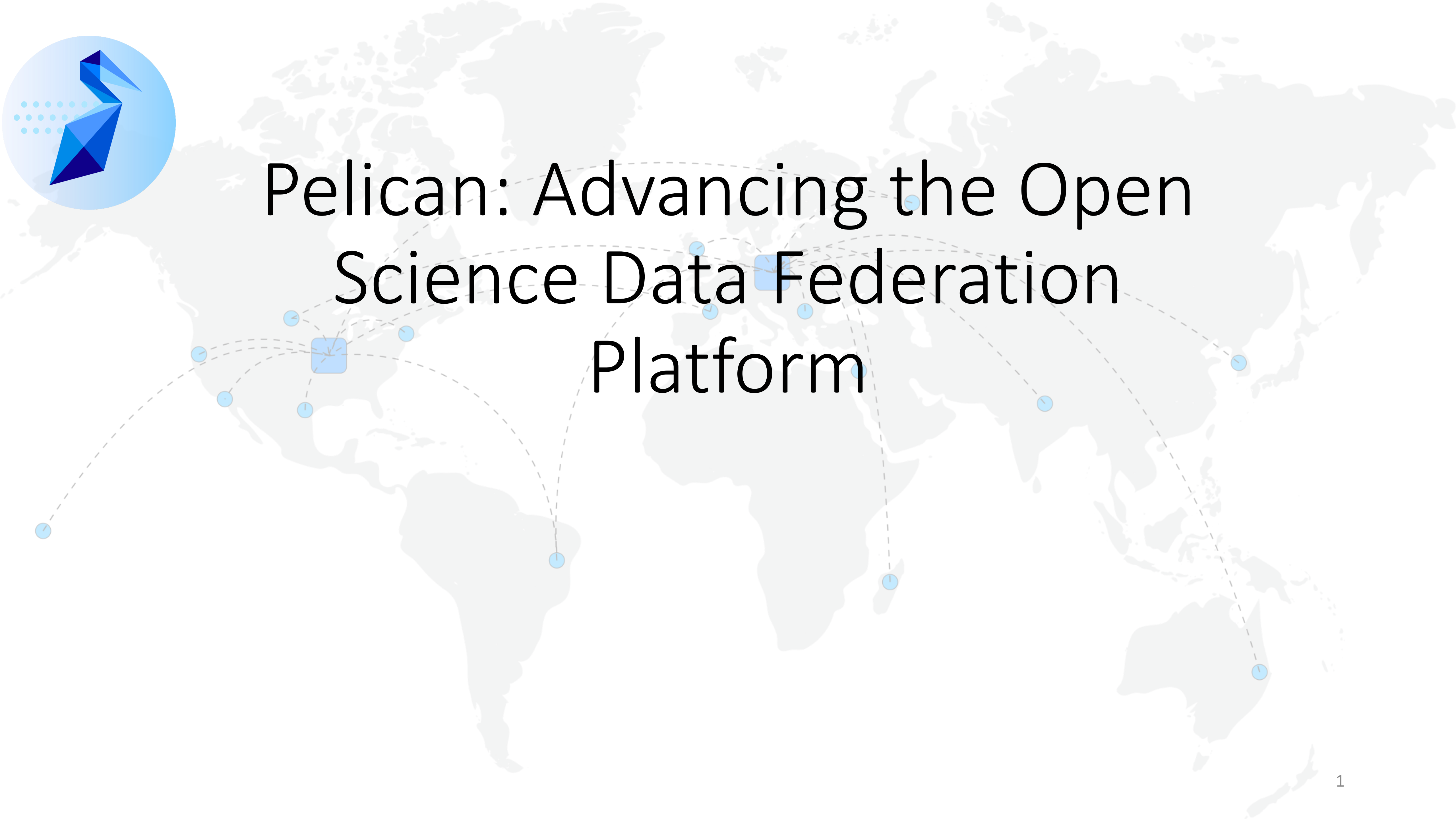
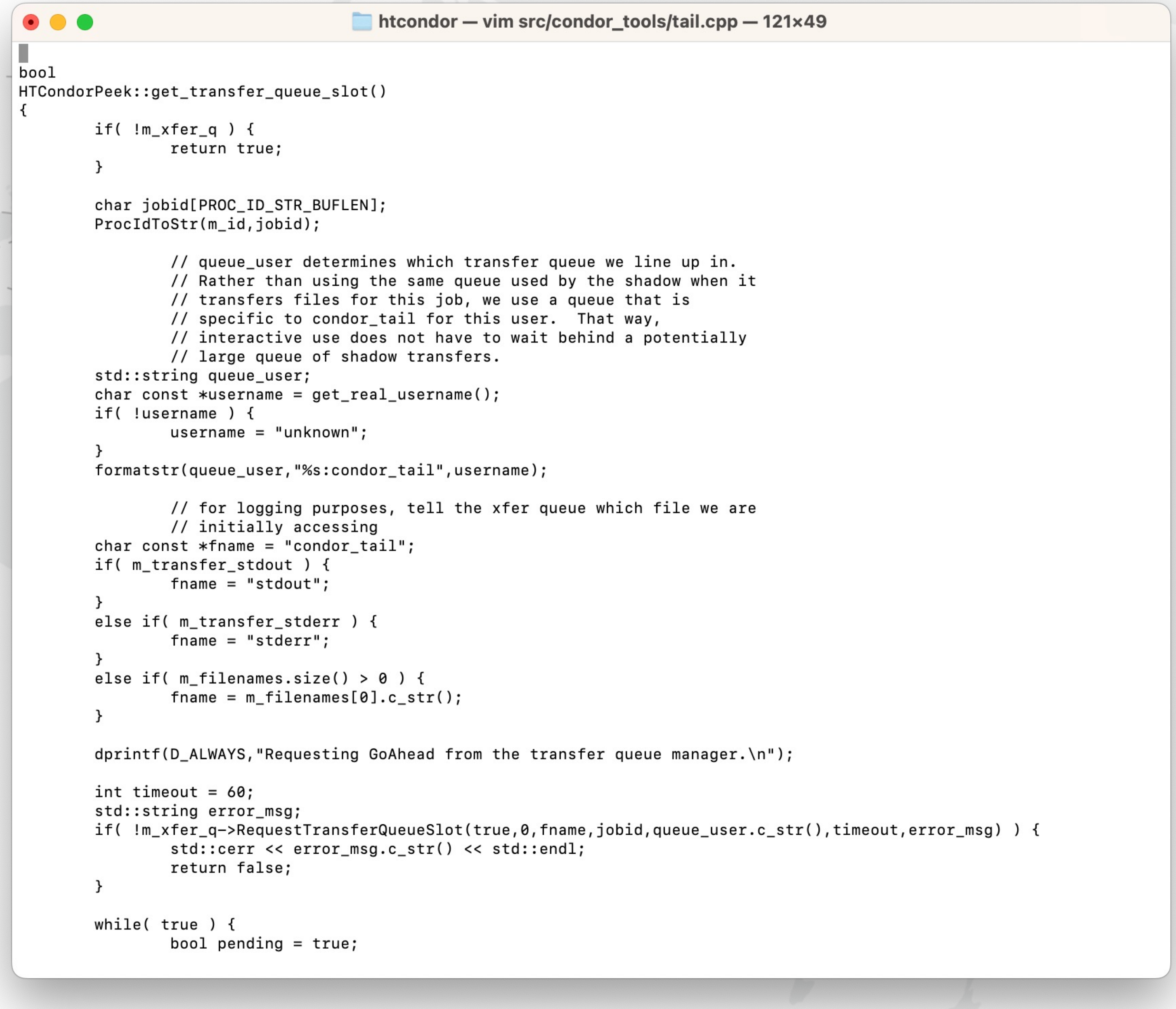


Pelican: Advancing the Open Science Data Federation Platform

Before we start…

One 🍻 Please



```
bool
HTCondorPeek::get_transfer_queue_slot()
{
        if( !m_xfer_q ) {
                return true;
        }

        char jobid[PROC_ID_STR_BUFLEN];
        ProcIdToStr(m_id,jobid);

                // queue_user determines which transfer queue we line up in.
                // Rather than using the same queue used by the shadow when it
                // transfers files for this job, we use a queue that is
                // specific to condor_tail for this user.  That way,
                // interactive use does not have to wait behind a potentially
                // large queue of shadow transfers.
        std::string queue_user;
        char const *username = get_real_username();
        if( !username ) {
                username = "unknown";
        }
        formatstr(queue_user,"%s:condor_tail",username);

                // for logging purposes, tell the xfer queue which file we are
                // initially accessing
        char const *fname = "condor_tail";
        if( m_transfer_stdout ) {
                fname = "stdout";
        }
        else if( m_transfer_stderr ) {
                fname = "stderr";
        }
        else if( m_filenames.size() > 0 ) {
                fname = m_filenames[0].c_str();
        }

        dprintf(D_ALWAYS,"Requesting GoAhead from the transfer queue manager.\n");

        int timeout = 60;
        std::string error_msg;
        if( !m_xfer_q->RequestTransferQueueSlot(true,0,fname,jobid,queue_user.c_str(),timeout,error_msg) ) {
                std::cerr << error_msg.c_str() << std::endl;
                return false;
        }

        while( true ) {
                bool pending = true;
```

Dataset Sharing Powers Open Science

- Datasets are the lifeblood of science. Without them, the cleverest software or algorithms are essentially useless.
- To *utilize* these datasets, they need to be effectively connected to a broad set of computing types and capacity.
- Engineering this access is an immensely difficult problem:
  - Challenges in both software & infrastructure scale.
  - Inherently distributed problem: the archive and the computing may be separate resources.

The **Open Science Data Federation (OSDF)** federates dataset repositories with a reach of all of open science.

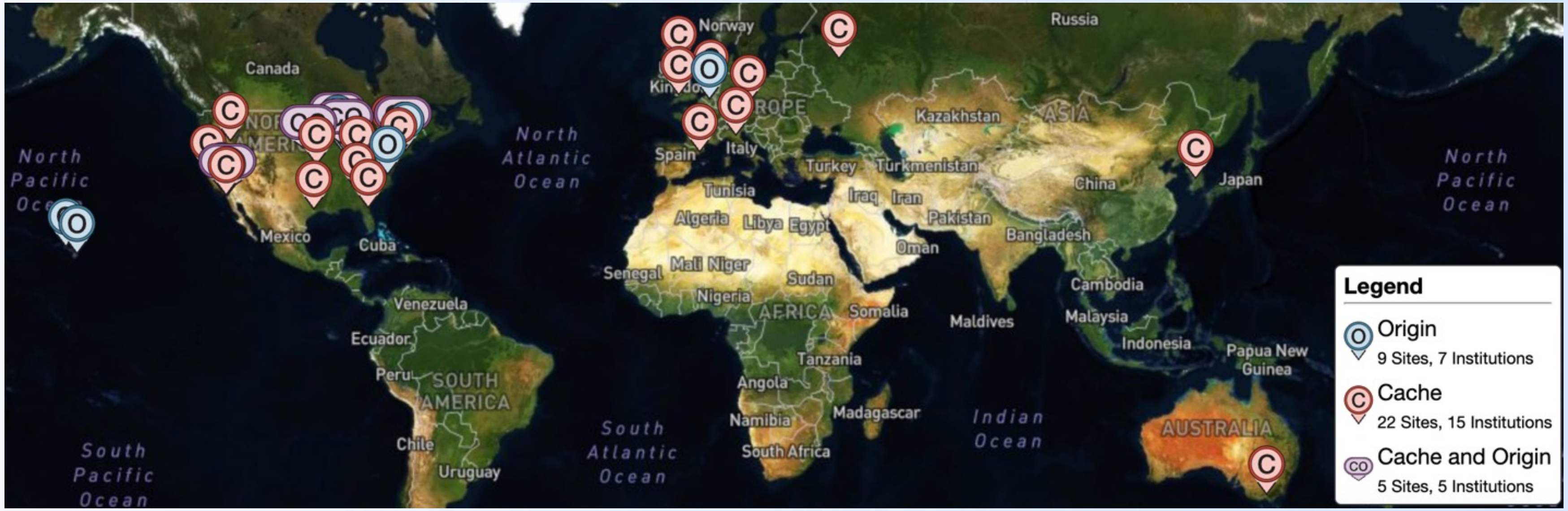
The Open Science Data Federation

The OSDF connects repositories through origin services and scales the dataset access through a series of distributed caches.

The OSDF is operated by PATh

Using hardware from NRP

And integrates a wide range of open science.



OSDF by the numbers

Over the last 12 months, the OSDF transferred

64PB &

62 files/s

Data used by

15 science collaborations &

~120 OSPool users

Boise
Sunnyvale
Pasadena
San Diego
Lincoln
Kansas City
Cincinnati
New York
Newport News
Atlanta
Houston
Jacksonville

Introducing: the Pelican Platform

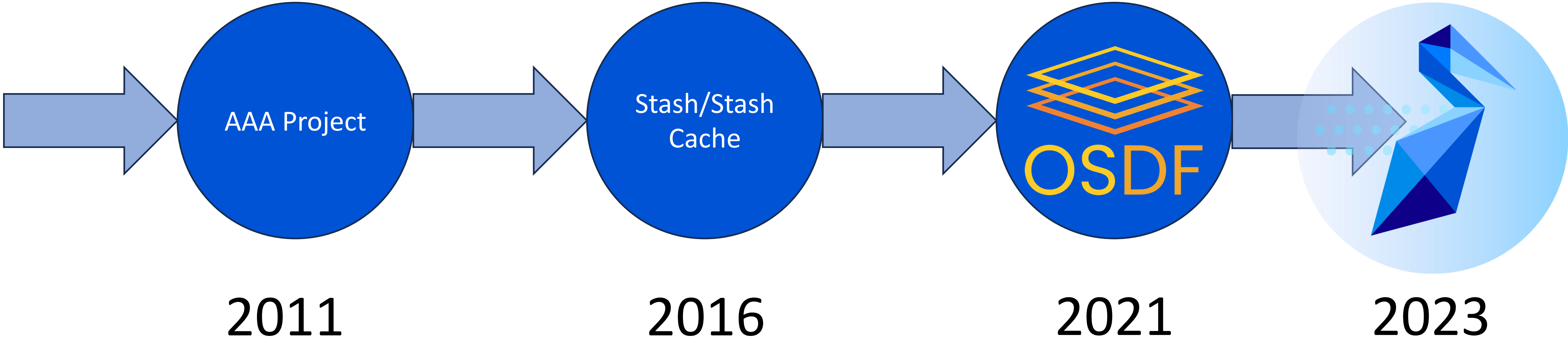
The Pelican Platform is a project recently funded by the NSF to help advance the OSDF. Scale: $7M over 4 years starting 1 September.

Three main goals:

1. Strengthen and Advance the OSDF
2. Expand the types of computing where OSDF is impactful.
3. Expand the science user communities.
   - Particularly, Pelican is funded as part of the “National Discovery Cloud for Climate” (NDC-C) so there will be an early focus on climate datasets.

Project Goals and Approaches

1. Strengthen and Advance the OSDF.
   - Ensure the components of the OSDF – Pelican – functions as a standalone software suite.
   - Re-engineer the central services. Add abstraction layer between the OSDF and other OSG Services.
   - Expand the origin’s functionality, focusing on non-POSIX backends (S3, DataVerse) and easing authorization.
   - Maintain and improve the distributed caching services.
2. Expand the types of computing where OSDF is impactful.
   - Add new browser-based client & python API.
   - Integrate with ML frameworks such as PyTorch
3. Expand the science user communities.
   - Collaboration with NCAR, NationalDataPlatform (https://www.nationaldataplatform.org/)



AAA Project

Stash/Stash Cache

OSDF

2011

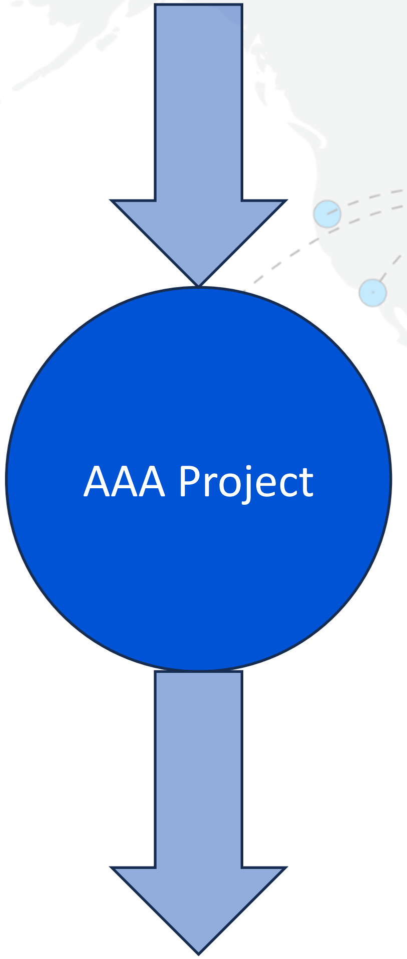
2016

2021

2023

A trip down memory lane...

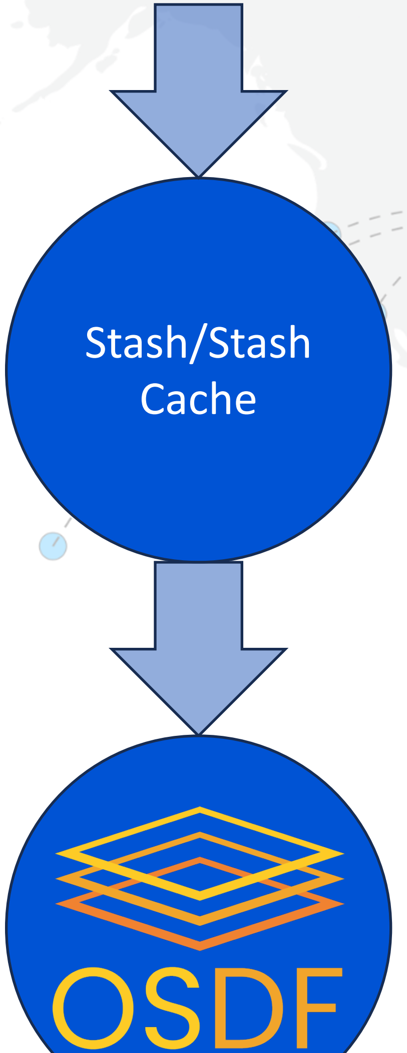
Any time, Any Data, Anywhere



AAA Project

- In ~2009, the teams at Nebraska, Wisconsin, and UCSD started using XRootD to build out a data federation for CMS.
- This turned into a 3-year funded project, AAA (NSF #1104664), starting in 2011.
- Outcomes include:
  - Design & implement AAA, completely based on XRootD software.
  - First generation of XRootD monitoring collectors used by the WLCG.
  - First (and second) implementation of the XRootD caching proxy (later rechristened “XCache”).

Evolution toward the OSDF

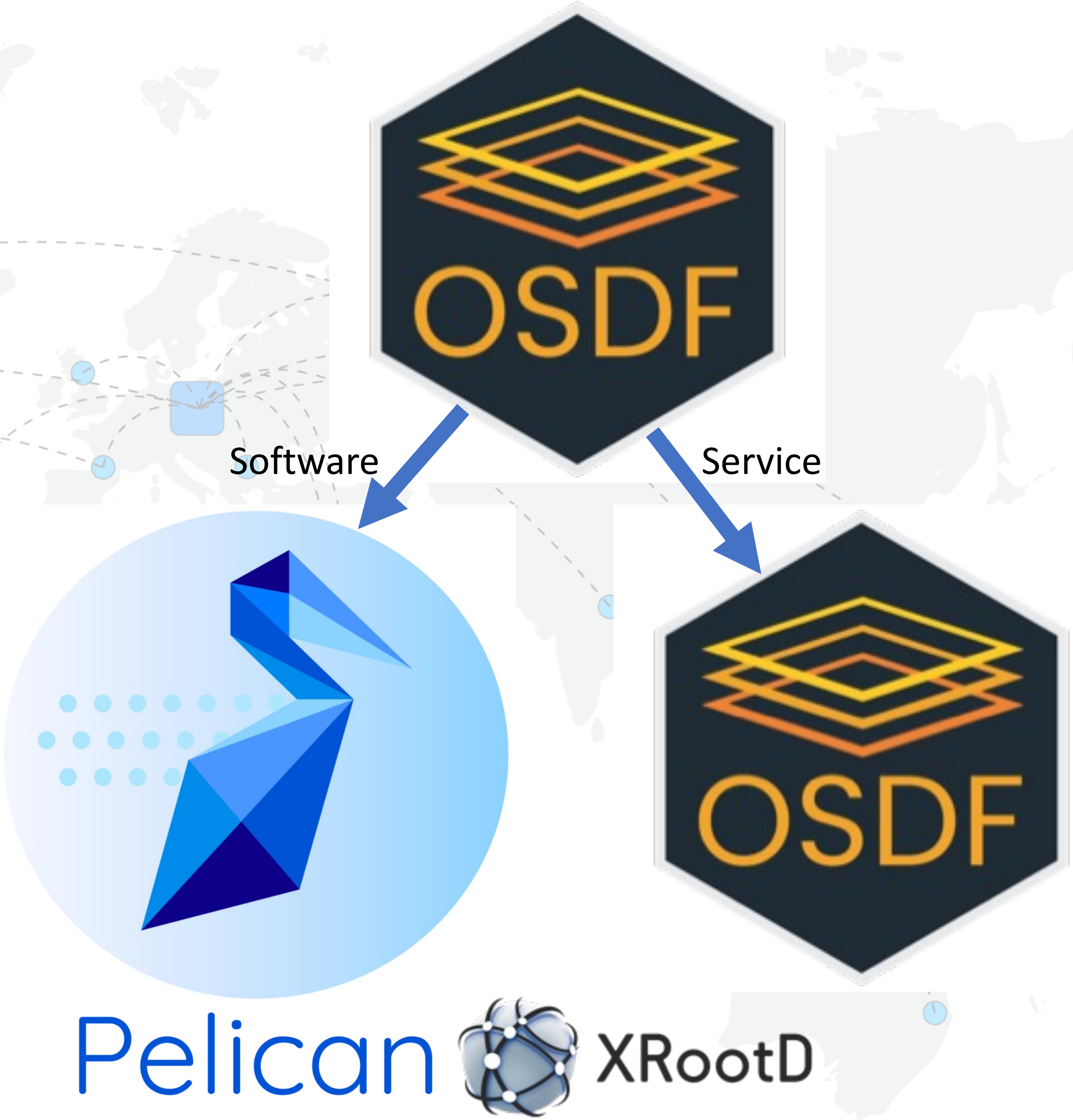


Stash/Stash Cache

- Around 2016, the OSG Consortium built on top of the AAA approach.
  - Originally used to export the “Stash” filesystem at U.Chicago; hence, the caching infrastructure was “StashCache”.
- Evolutions from being CMS-specific:
  - Was used to scale access to Stash via an origins – strong focus on data caching and delivery.
  - Developed a client for cache discovery and robust downloads (Python).
  - Configuration & registration based on OSG’s Topology service.
  - Start switch to HTTP protocol & token-based auth.
- This evolved into the OSDF in ~2021:
  - Client was rewritten into Go.
  - Hardware was placed into the network.
  - Emphasis on Kubernetes-based packaging.
  - Distributed service operations with Kubernetes and the NRP.
  - NSF-funded hardware projects join in the federation.

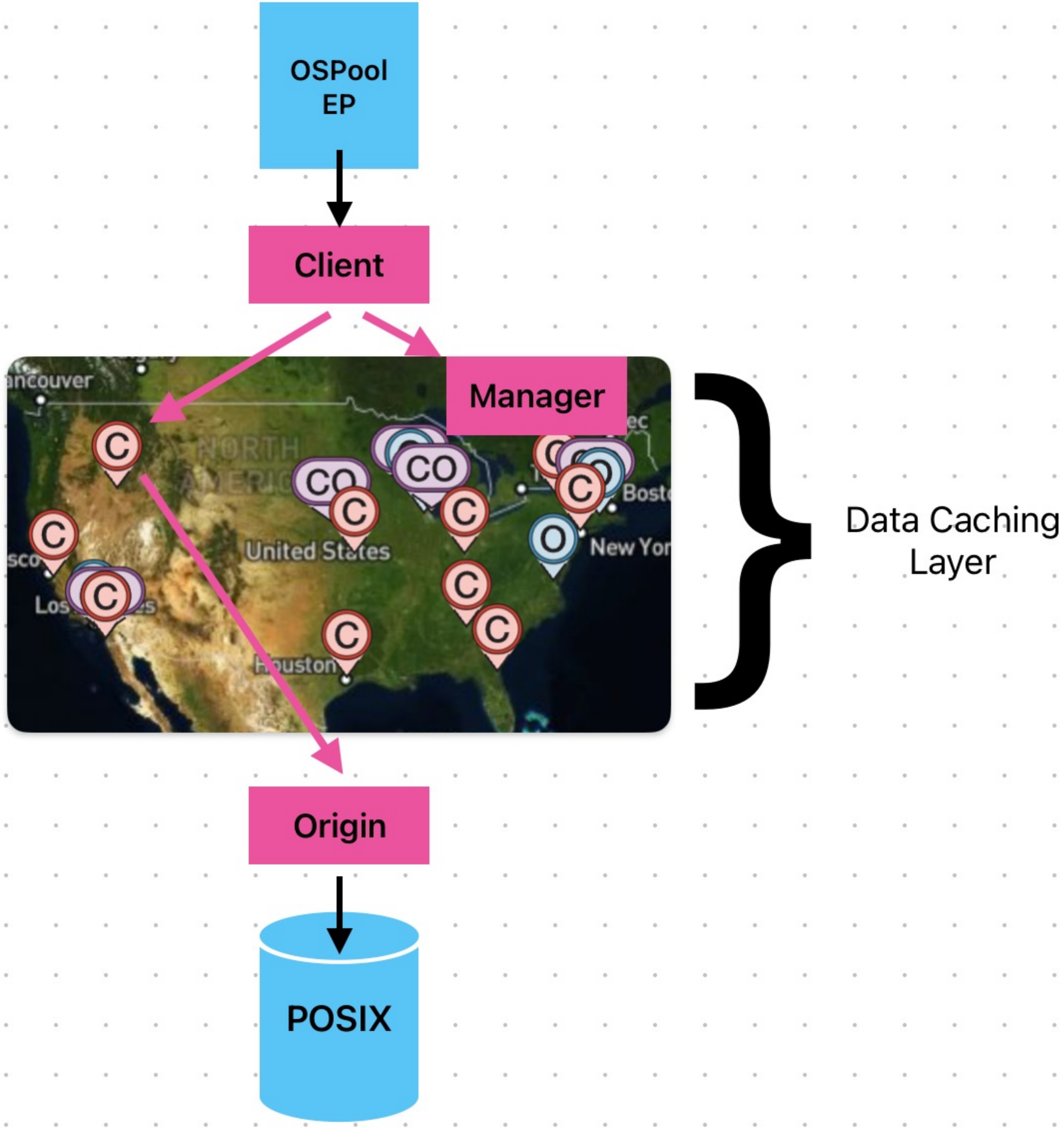
OSDF & Pelican

- We split out the technology powering the OSDF and christened it the “**Pelican Platform**”.
    - The software is the same – integrating technologies like XRootD, SciTokens, OA4MP, and the ‘stashcp’ client utilities.
- The Pelican project provides the effort for a solid engineering foundation.



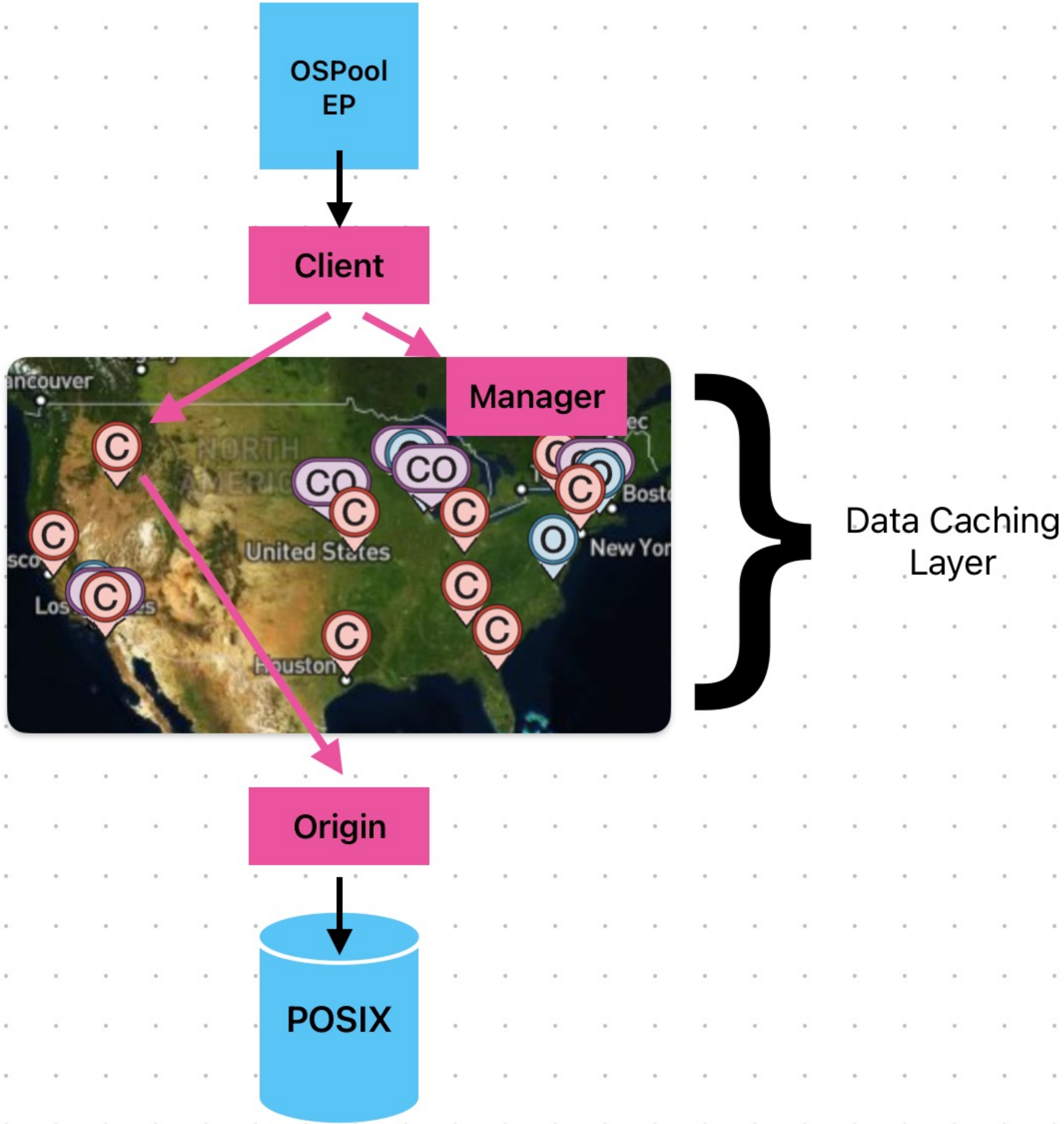
Pelican Architecture

- An **origin service** integrates the object store into the OSDF in the same way a CE integrates a batch system into the OSPool. Interfaces to move data and map authorizations.
- The **cache service** stores and forwards objects, providing scalability to the data access.
- The **manager** selects a source/sink of an object for clients and maintains the namespace.

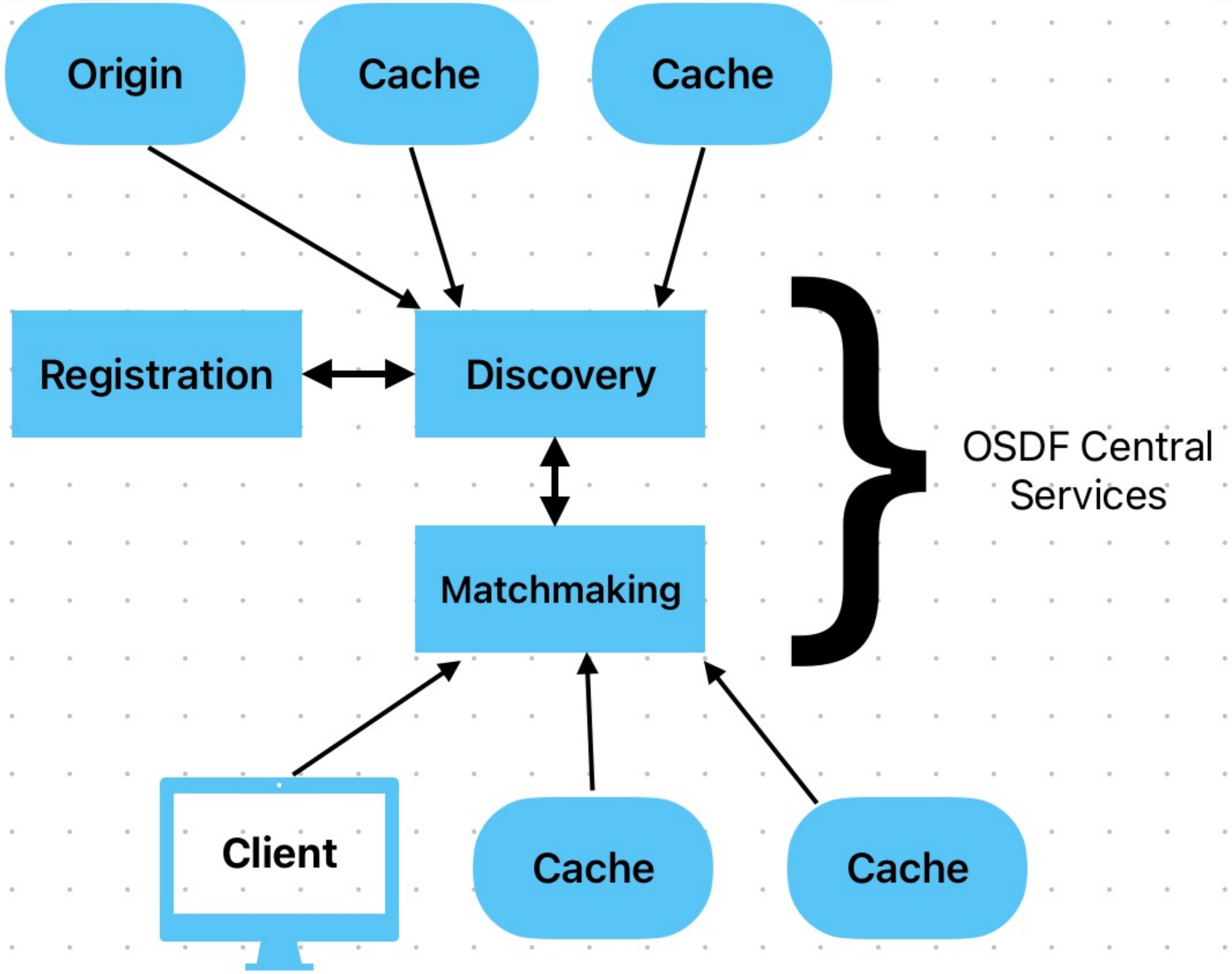


OSDF in Practice

- The OSPool is tightly integrated with the OSDF, giving jobs reliable, scalable means to move data in and out.
  - Same technique used by the CHTC, IGWN, PATh pools.
- OSDF is just starting to integrate more with campuses (often via the NSF) and address non-compute cases.



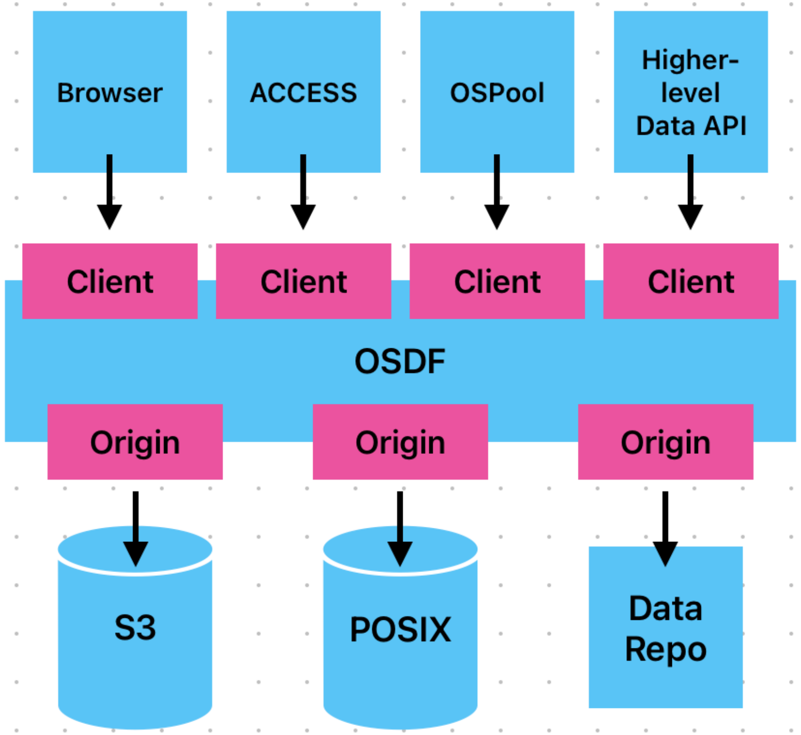
Pelican Architecture – Central Services



The central services are seeing a massive overhaul for Pelican.

- Standalone registration workflow, not tied to OSG Topology.
- Service registration associated with a public key – better protection against spoofing.

OSDF Architecture - Vision



- **Long-term vision**: Pelican Platform, provides a ‘transport bus’, connecting a broad range of dataset providers to consumers.
  - Today, we just integrate POSIX backends which leaves out a wide range of storage types (S3, data repositories like DataVerse)
- Become a platform beyond the CLI-centric, object delivery service it is today.

Pelican as the ‘most favored nation’ of HTCSS

- Pelican and PATh (the project funding the majority of the HTCSS work) share common leadership and principles.
- Unique opportunity to co-design and have Pelican drive new transfer & storage management capabilities in HTCondor.
- Examples:
  - Pelican client ships with HTCondor EP.
  - Pelican drove the development of the new shadow-side plugins.
  - HTCSS is integrating the LotMan component for storage management.
- Even as Pelican broadens the OSDF, we will still leverage this relationship!

+ PATh = ♥

Pelican Platform: making federated storage available to everyone

The OSDF was seeded out of the technologies of the HEP and dHTC communities.

- Already has made a huge impact on science!

Pelican opens this further:

- Anyone can setup their own data federation, not explicitly tied to the OSG Consortium.
- Targets a broader range of science & computing approaches.
  - While keeping its roots with the HTCondor community

Looking forward to changing how you connect to datasets in the next years!

Questions?

This project is supported by the National Science Foundation under Cooperative Agreements OAC-2331480. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.