T and T Consulting Services, Inc.

Storefront Technology Assessment

T and T Consulting Services, Inc.
www.tatcs.com

Agenda

- Overview
- Recommendation
- Output
- Use Case
- Two Approaches
- Summary & Next Steps

OVERVIEW

Consideration

As part of our efforts to make a product recommendation for the NESDIS Storefront, Team T&T has been conducting a technology assessment for the experience. While the Storefront is not yet built, we wanted to explore some of the various data points that might feed into the Storefront.

Objective: Organize NOAA's 25 web service endpoints by data type (e.g., Atmospheric, Oceanic, etc.) to enable efficient access for 12 personas, supporting NESDIS Storefront goals.

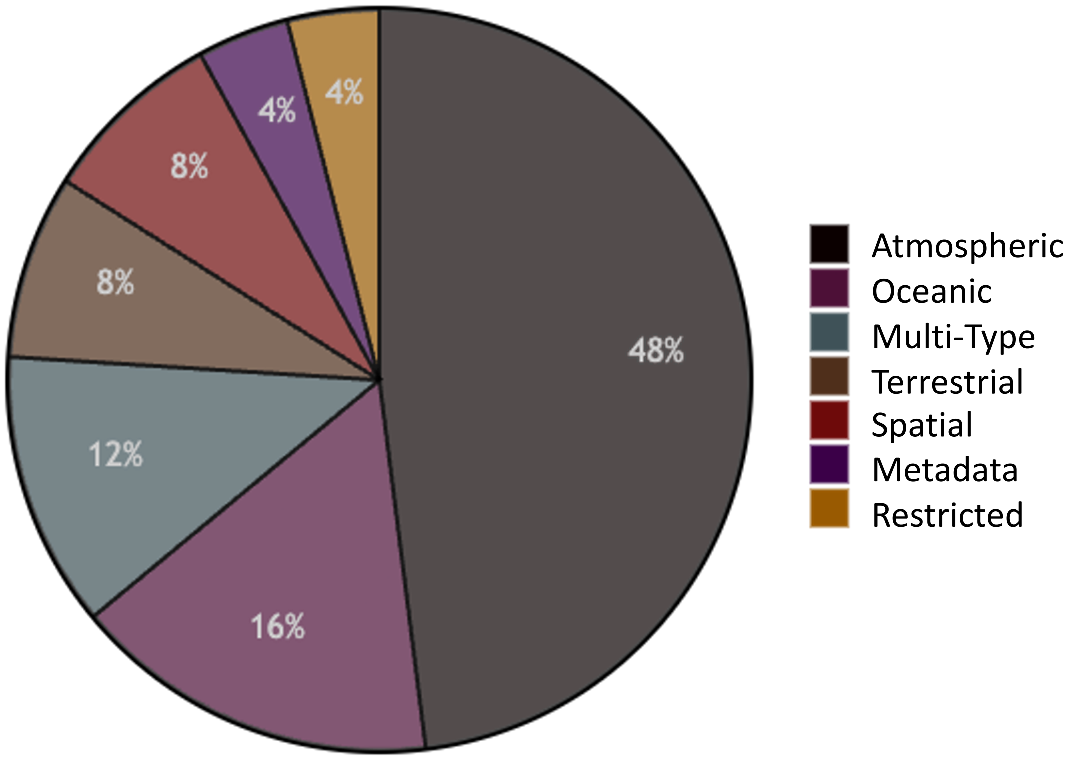
Background

The near-term solution should not rely on NCCF.

Reports Reviewed

- **25 Endpoints (NESDIS, OSPO, NWS, NOS, NCEI, OAR):**
  - Atmospheric (12): CDO API, NWS API, GOES-19, MRMS, EMWIN, HURSAT, etc.
  - Oceanic (4): Tides & Currents API, OISST, MPSR, Jason Series
  - Terrestrial (2): VIIRS, Argos DCS
  - Spatial (2): CO-OPS GIS, SARSAT
  - Multi-Type (3): NCEI API, GEONETCast
  - Metadata (1): Repository API
  - Restricted (7, overlapping): GOES DCS, SARSAT, EMWIN, etc.
- **Formats:** JSON (1–10 KB), NetCDF (10 MB–1 GB), GeoJSON, BUFR, proprietary
- **Traffic:**
  - Ingress: ~Tens TB/day (e.g., GOES-19 to Atmospheric Pond)
  - Inter-Pond: ~1–10 TB/day
  - Egress: 10–100 GB (Ortiz), 1–10 KB (Wright)
- **Integration:** Supports NCCF and APIs
- **Personas:** Points of pain and needs

NOAA Endpoints Assessed



RECOMMENDATION

Recommendation

Create a federated system of data ponds with a unified API, aligning with FAIR principles and human-centered design.

What is a data pond?

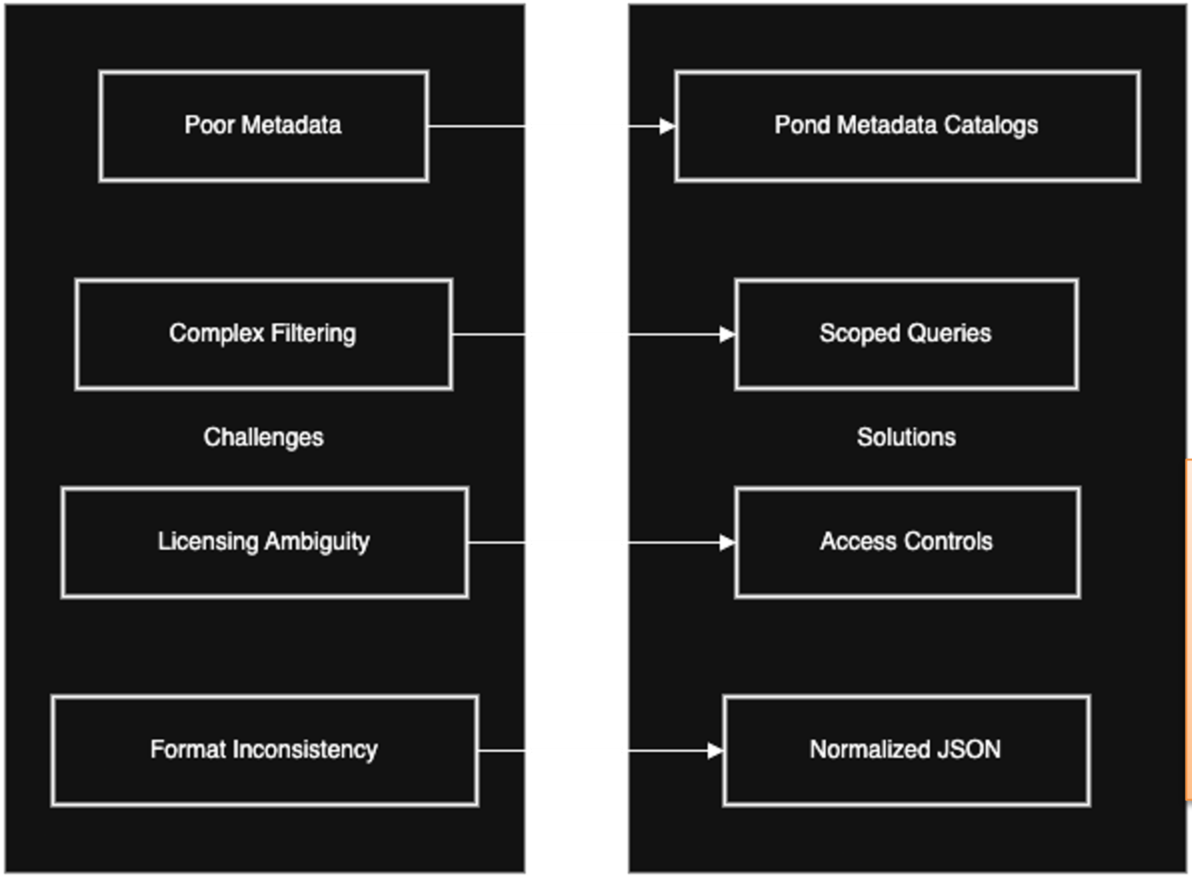
A data pond is a small, focused, or temporary data storage area, especially in contrast to larger structures like a data lake or data warehouse.

Term	Description
Data Pond	A small, specific, often project-level data store
Data Lake	A large, centralized repository storing raw, unstructured or semi-structured data
Data Warehouse	A structured, cleaned, and integrated data store optimized for analytics and reporting

Why a Federated System of Data Ponds?

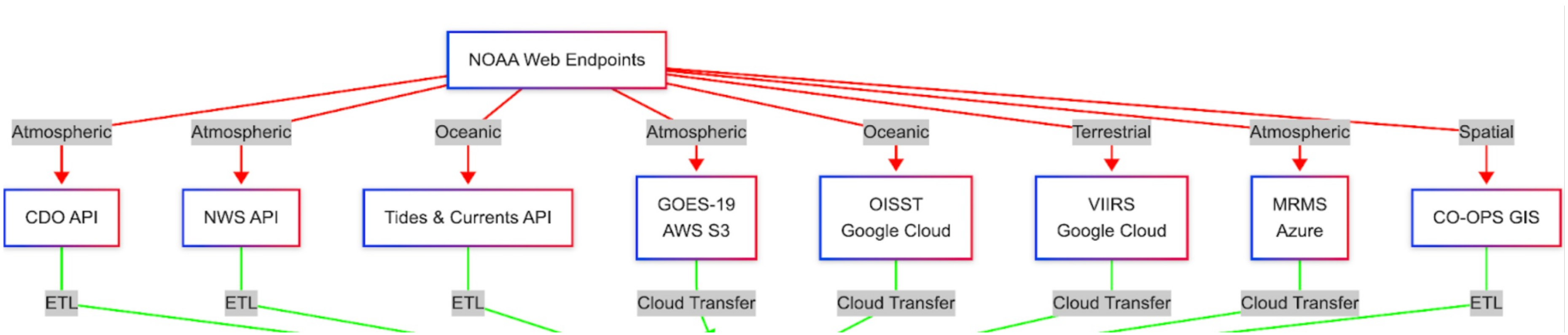
- **Poor Metadata:**
  - Inconsistent schemas (e.g., CDO vs. NCEI)
  - Solution: Pond-specific metadata catalogs
- **Complex Filtering:**
  - Diverse formats
  - Solution: Scoped queries to ponds
- **Licensing Ambiguity:**
  - Unclear commercial terms
  - Solution: Pond-level access controls
- **Format Inconsistency:**
  - JSON vs. NetCDF vs. GeoJSON
  - Solution: Normalized JSON via API
- **Other considerations**
  - Fragmented endpoints (CDO API, GOES-19, SARSAT) with inconsistent formats (JSON, NetCDF, GeoJSON)
  - A single data lake is complex (ponds offer modularity and performance
  - Diverse persona needs

Alignment to Challenges



Ty to VD

How It Works



Ty to VD

Second Step

Layer in an AI Assisted API.

What Does This Mean

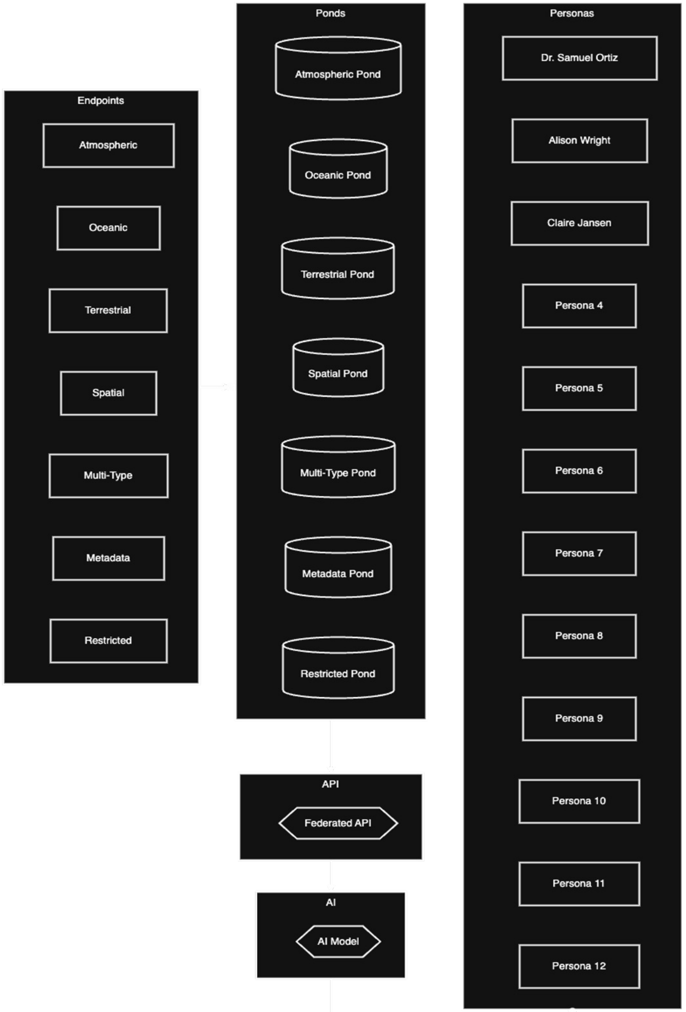
XYZ

Ty to write to

How API Layering Will Work

- **Federated Data Ponds:**
  - 6+ ponds: Atmospheric, Oceanic, Terrestrial, Spatial, Multi-Type, Metadata, Restricted
  - ETL normalizes data (e.g., JSON/NetCDF → Parquet) with metadata catalogs (REQ-NS-156)
- **Unified API:**
  - https://api.noaa-ponds.org/v1/data queries across ponds
  - Example: /data?type=storm®ion=CA → Atmospheric + Oceanic
- **JSON API:**
  - Default JSON output, options for NetCDF, GeoJSON
- **AI Layering:**
  - BERT-based NLP routes queries (e.g., "urban heat" → Oceanic + Terrestrial)
  - Personalizes responses (e.g. NetCDF)
- **Security:** 2FA, OAuth for Restricted Pond

How It Will Look



Ty to VD

OUTPUT

Unified API with Data Personalization

- **Federated API:**
  - Queries ponds, delivers JSON (e.g., /data?type=urban_heat)
  - Options: NetCDF, GeoJSON, imagery
- **Layering:**
  - AI routing (e.g., Jansen: Atmospheric + Oceanic)
  - Personalization: Ortiz (NetCDF), Wright (GeoJSON)
  - Cross-Pond Aggregation (e.g., Myers: GOES-19 + OISST)
- **Benefits:**
  - Metadata resolution
  - Simplified filtering
  - Licensing clarity
- **Security:**
  - OAuth for Restricted Pond

JSON Snippet Example

json{
"data": { "precipitation": [],
"water_levels": [] },
"metadata": { "sources":
["Atmospheric Pond", "Oceanic Pond"]

USE CASE

Lieutenant Colonel Jenna Blackwell (Military Operations Planner)

Persona Background: Lieutenant Colonel Jenna Blackwell is a military operations planner with the U.S. Air Force, specializing in real-time environmental data for mission planning and execution. She requires rapid access to NOAA's weather and geospatial data to support operations, including disaster response and tactical missions. Her role demands high-availability data, secure access to restricted endpoints (e.g., EMWIN, SARSAT), and integration with military systems. She has NOAA public trust clearance, enabling access to sensitive datasets.

Scenario: Lt. Col. Blackwell is tasked with planning a humanitarian aid mission in response to a Category 4 hurricane approaching the southeastern U.S. coast. The mission involves coordinating Air Force logistics, including airlifts and reconnaissance flights, to deliver supplies and assess damage. She needs real-time weather data (e.g., precipitation, wind speed), geospatial data (e.g., flood zones), and satellite imagery to ensure safe flight paths and prioritize aid distribution. The data must be accessible securely, quickly, and in formats compatible with military planning tools (e.g., JSON for real-time updates, GeoJSON for mapping).

Current State

Blackwell accesses NOAA's endpoints separately: NWS API for weather forecasts, GOES-19 for satellite imagery, and CO-OPS GIS for coastal flood data.

Challenges include:

- **Fragmented Access**: Navigating multiple endpoints (e.g., NWS API, GOES-19) is time-consuming.
- **Format Inconsistency**: JSON (NWS API) and NetCDF (GOES-19) require manual conversion for integration.
- **Security**: Restricted endpoints like EMWIN (emergency weather data) require separate authentication, slowing her workflow.
- **Metadata Gaps**: Unclear data sources and update frequencies hinder decision-making under tight deadlines.

Proposed Solution

System Response:

- **AI Routing:** A BERT-based NLP model interprets the query and routes it to relevant ponds:
  - **Atmospheric Pond:** Pulls NWS API (JSON) and GOES-19 (NetCDF) for weather and imagery.
  - **Spatial Pond:** Retrieves CO-OPS GIS (GeoJSON) for flood zones.
  - **Restricted Pond:** Accesses EMWIN (real-time alerts) using her NOAA clearance via Oauth.
  - **Data Aggregation:** The API normalizes data into JSON by default, with an option for GeoJSON to support mapping tools.

Personalization:

- **Tailoring:** The AI tailors the response to Blackwell's needs:
- **Prioritization:** Prioritizes real-time JSON for quick integration into mission planning software.
- **Metadata:** Includes metadata (e.g., source, latency) to ensure reliability for critical decisions.

Security: OAuth authentication verifies her clearance for EMWIN, ensuring compliance with security requirements

JSON REQUEST


```
{
  "data": {
    "precipitation": [...],
    "wind_speed": [...],
    "flood_zones": [...],
    "satellite_imagery": [...]
  },
  "metadata": {
    "sources": ["Atmospheric Pond", "Spatial Pond", "Restricted Pond"],
    "last_updated": "2025-05-23T11:20:00CDT"
  }
}
```

Outcome

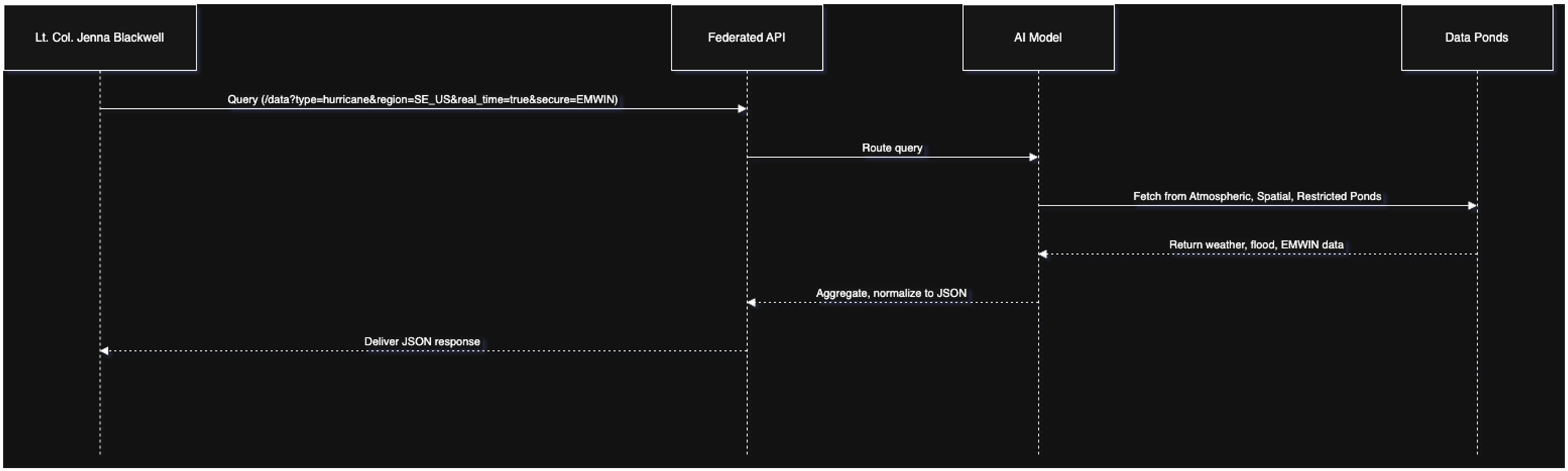
Efficiency: Blackwell accesses all necessary data through one API call, reducing planning time from hours to minutes.

Accuracy: Normalized JSON and clear metadata ensure data compatibility and reliability, enabling precise flight path planning and aid prioritization.

Security: Secure access to restricted data (EMWIN) supports mission-critical decisions without delays.

Impact: The mission successfully delivers aid to affected areas, with safe and optimized flight operations, enhancing disaster response effectiveness.

Use Case Diagram



Ty to VD

How It Might Function

- This is built by AI
- Not designed or branded
- Simply a reflection of how this application could support Lieutenant Colonel Jenna Blackwell

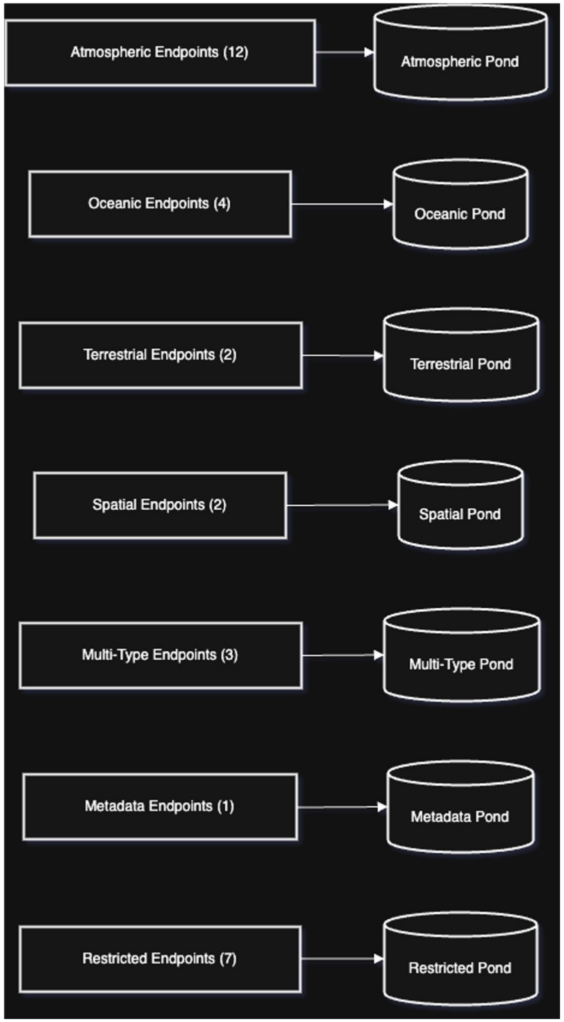
[Use Case Experience]

TWO PHASES

Phase #1: Endpoints to Data Ponds

- **ETL:** JSON/NetCDF → Parquet
- **Restricted Pond:** Clearance-based access
- **Metadata:** Catalogs for routing
- **Traffic:** ~Tens TB/day ingress

Approach #1: Endpoints to Data Ponds

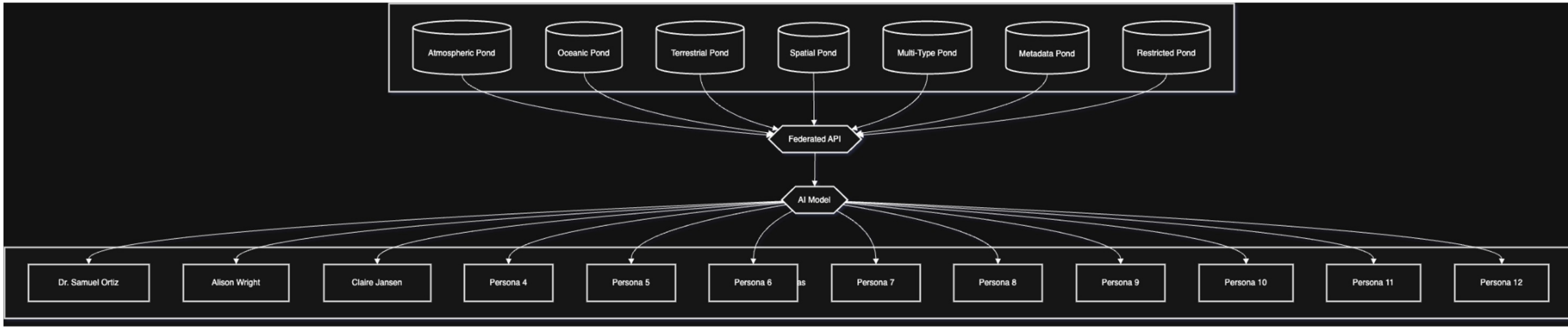


Ty to VD

Phase #2: Data Ponds to Personas

- **Queries:** Multi-pond queries (e.g., Lin: Oceanic + Terrestrial)
- **Routing:** AI
- **Delivery:** Personalized
- **Traffic:** 10–100 GB (Ortiz), 1–10 KB (Wright)

Approach #1: Endpoints to Data Ponds



Ty to VD

SUMMARY & NEXT STEPS

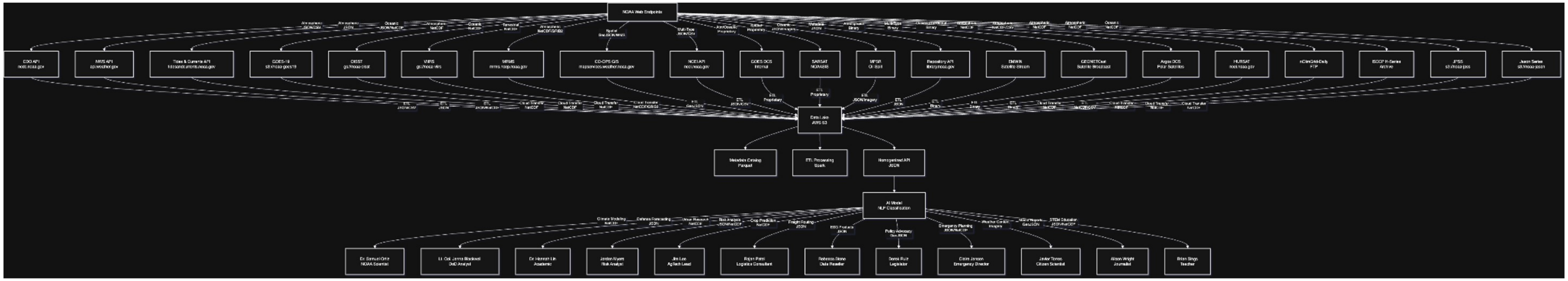
Scalable & Personalized Solution

1. **Data Ponds:** Modular repositories for 25 endpoints

2. **Federated API:** Unified, AI-driven access

3. **Benefits:**
    - Resolves metadata/filtering/format issues
    - Clarifies licensing
    - Scales for personas

Approach #1: Endpoints to Data Ponds



Ty to VD

Next Steps

- AWS S3, Spark for ponds (or internal / standardized)
- AWS Lambda, BERT for API/AI (or custom)
- NSOF authentication for restricted endpoints (REQ-NS-174)

THANK YOU