Booz
Allen

A Study to Determine Natural Language Processing Capabilities with NCCF Open Knowledge Mesh (KM/NLP)

A Technical Report Concluding NOAA NESDIS Discoverability Enhancer for Context/Quality of KM Using AI *[Deck-AI]*

Acknowledgement:

This project was funded through the Joint Venture Partnerships Program, Office of Systems Architecture and Engineering (SAE), NOAA/NESDIS

Table of Contents

Table of Figures

Table of Tables

Authorship

Report Principal Investigators – Booz Allen

Karolyn Babalola, Senior AI/ML Engineer
Prachi Sukhatankar, Vice President

Report Co-Authors – Booz Allen

Rebecca Kurz, Lead AI Solution Architect
Elizabeth Redfearn, Senior Digital Transformation Specialist

Contributors

Ryan Berkheimer, NOAA
Jennifer Webster, NOAA
Justin Dennison, NOAA
Kevin Bartlett, NOAA
Beau Backus, NOAA
Joseph Conran, NOAA
Jonathan Mote, NOAA
Austin Sandler, NOAA
Jessica Hung, AWS ProServe
Chris Mattioli, AWS ProServe
Tracy Rouleau, TBD Economics
Marie Nguyen, Booz Allen
Demi Gray, Booz Allen

Disclaimer

This material is based upon work supported by the Dept of Commerce under Contract No. 1332KP24C0027. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Dept of Commerce.

1. Executive Summary

The National Oceanic and Atmospheric Administration (NOAA) issued a Broad Agency Announcement (BAA) in pursuit of the National Environmental Satellite, Data, and Information Service (NESDIS) mission to improve democratization of its information, data and processes. The targeted outcome of this democratization is empowering NOAA stakeholders across key strategic initiatives, such as enhancing access to environmental intelligence for weather, climate and environmental resilience. The BAA study to Determine Natural Language Processing Capabilities within the NESDIS Common Cloud Framework (NCCF) Open Knowledge Mesh (KM) explored innovative approaches to **develop information processing systems that improve discovery, trust, and usability of data assets and processes** with the backing of an interoperable, semantic knowledge mesh.

Improving democratization and usability of datasets across NOAA while promoting open science and open data is a priority for NESDIS. With a strategic focus on linking disparate datasets, expanding open access, and strengthening contextual understanding of NOAA data, this study aligns with the NOAA and NESDIS missions by investigating how large language models (LLMs), natural language processing (NLP), and knowledge mesh frameworks can enable dynamic, persona-driven information access. By supporting highly contextual query resolution and enhancing semantic linkages across multi-domain data sources and processes, **this study aimed to demonstrate how AI technologies can help NOAA scale insight generation, serve a broad spectrum of users, and guide the evolution of future data architecture strategies.**

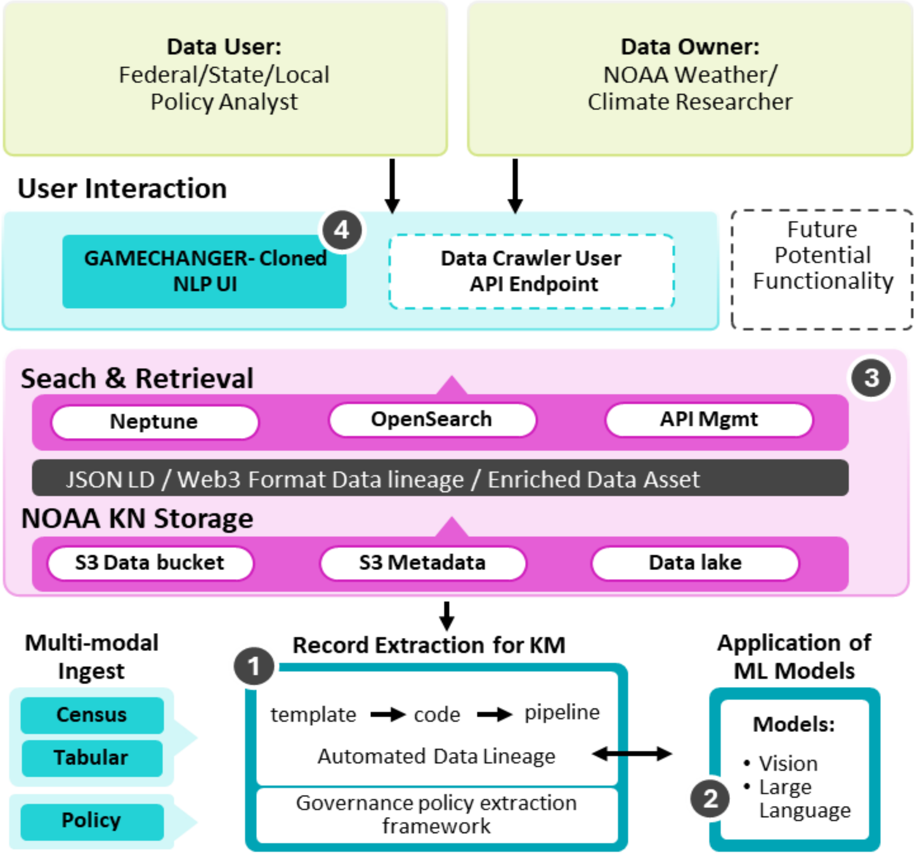
Performers were to choose from three different phases of study, including:

- **Phase A:** Knowledge Mesh Construction: Capabilities related to automating the ingest,enhancement, or quality control of information meant to come into or actively cominginto the system.
- **Phase B:** Knowledge Mesh Grooming: Capabilities related to continual sustainment ofthe knowledge mesh resource in terms of optimization, standardization, and qualitycontrol.
- **Phase C:** Knowledge Mesh Exploitation: capabilities related to the (foundation modelbased) access side interface of the system.

Booz Allen chose Phase A to provide a more foundational study of the knowledge mesh; in recognition of its nascent form, Phase A would develop capabilities that would enable Phases B and C.

This project leveraged Booz Allen's open-source tools, GAMECHANGER and aiSSEMBLE™ to demonstrate how advanced visualizations can speed adoption by multiple end users and to show how one could rapidly prototype AI-assisted metadata pipelines that support multi-modal data types, such as, tabular collection records, unstructured grant reports and policy documents. As depicted in Figure 1, the prototype features four (4) partially overlapping steps, 1) automated record extraction for metadata lineage, 2) application of LLMs to enrich and structure metadata, 3) synchronization with existing KM via a multi-service data lake, and 4) dynamic visualization of enriched content. The development team ("dev team") took a methodological approach to integrating data and AI-enhancing processing into a transformative knowledge mesh ecosystem.

Figure 1: High-Level Approach



This study's findings suggest that the knowledge mesh ecosystem has great potential for enabling more context based semantic searches across siloed datasets and accelerating the generation of structured insights from distributed content. This project delivered an adaptable open architecture that could be easily adopted with the continued development of the foundational components of the NOAA knowledge mesh. The dev team used domain-relevant, yet generalized approaches to governance, automatic template generation to rapidly integrate different data types, and LLM-assisted metadata enrichment and provenance pipelines. To build on current progress, the dev team recommends: engaging more users by adopting a standard user interface and developing governance training standards, scaling search and LLM-integration through a more defined knowledge graph and extended infrastructure, simplifying onboarding, and exploring additional use cases by embracing a more comprehensive deployment roadmap. These steps will be essential for ***scaling the impact of NOAA's knowledge mesh and advancing NESDIS's broader vision for a connected, intelligent, and adaptive environmental data ecosystem.***

2. Introduction and Objectives

2.1 Introduction

Imagine a policy analyst in the state of Florida, charged with analyzing funding for coastal resilience projects related to harmful algae blooms (HABs) for key legislative decision-makers. The blooms, which negatively affect fish ecosystems and can cause local beach closures, impact the local tourism and fishing economies, and subsequently may lead to constituent inquiries. The analyst needs access to relevant data across multiple domains to inform their work—from qualitative data on economic activity (i.e. restaurant revenue, lodging bookings) to public health statistics about gastrointestinal illnesses. While they are not a scientist by trade, the analyst may struggle to identify the most recent research, relevant observations, and spend time wading through highly technical language.

Now, envision a NOAA researcher who is completing a study on the economic impact of algae blooms along the Florida coastline. The researcher may rely on manual and ad hoc processes to load data to their repositories, leading to inconsistent metadata. In identifying existing data to leverage for benchmarking and analysis, they may struggle to locate and aggregate sources across disconnected data repositories or must spend time tracking down individual data owners across the organization.

These sample scenarios set the stage for the KM prototype solution and demonstrate a common challenge at NOAA: how to make its vast stores of disparate data usable, accessible, and discoverable to many different types of consumers. The challenges that the policy analyst and data owner personas face represent a larger fragmented approach that slows insight generation and increases the cost and complexity of science and decision-making. However, the opportunity to make this data searchable is likewise powerful; to inform mission-critical decisions around environmental related issues, such as economic resilience, ecosystem recovery, and disaster response.

With this study, the Booz Allen team aimed to address these challenges by integrating machine learning, NLP and generative AI techniques into NOAA’s knowledge mesh (KM) framework, Open Information Stewardship Service (OISS). Rather than relying on manual and time-consuming processes to standardize and improve metadata, the study approach leveraged AI technology to do it more efficiently, and created repeatable processes to make it easier for data owners to add their data to the KM. Additionally, the study approach provided data consumers with the interface needed to complete a search with a simple natural language query and visualize and rapidly discover data from distributed systems.

2.1 Background

This BAA is a follow up to the recent NESDIS Earth Observation Digital Twin (EO-DT) BAA, which sought to demonstrate how EO-DTs could serve as a next generation enterprise system for ground processing in the NOAA Earth System modeling effort[1] [2].[3] One of the many outcomes of that study was the urgency to move disparate data into findable, accessible, interoperable and reusable (FAIR) frameworks that embrace Web 3.0 standards. As an answer to this mandate, NESDIS began developing an open knowledge mesh capability to form the basis of data governance within NCCF. **This open knowledge mesh, called the Open Information Stewardship Service, or OISS, is an API that allows users to subscribe, or onboard, data and algorithms into a shared metadata management system.**

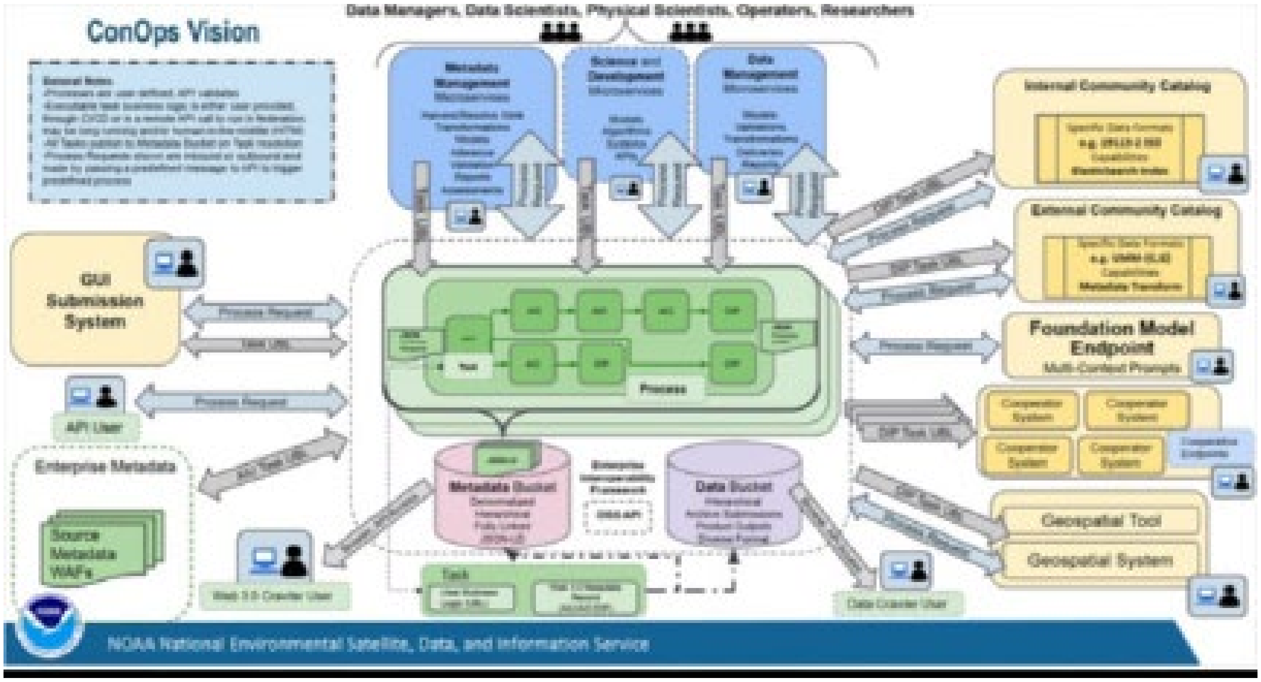
OISS enables the discovery of multi-source data, such as physical, economic, demographic, etc., in the same context as algorithms that may transform that data. This shared context between data and algorithms enables generating new knowledge while automatically tracking its provenance. OISS does this by semantically encoding all metadata into a common knowledge graph, the OISS Core and Framework ontologies. Integration into OISS requires registering datasets, assets or archives as canonical units called archival information units (AIUs). Processes or algorithms that transform or augment AIUs are registered as Archival Information Collections (AICs). For AIUs and AICs, provenance and usage data may be captured in dissemination information units (DIPs), which the KM can leverage for reporting or distributing data based on access privileges. DIPs can also be used to manage context for metric reporting, which this project leverages to share feedback metrics. Figure 2 shows a high-level view of the OISS concept.

[1] https://www.nesdis.noaa.gov/s3/2025-01/LM-NVIDIA-EODT-FinalReport-dmg-final-20250110.pdf

[2] https://www.nesdis.noaa.gov/s3/2025-01/NOAA-EODT---Orion-Final-Report---FINAL-dmg-final-20250110.pdf

[3] https://www.nesdis.noaa.gov/s3/2024-02/STC_EODT_202311302023_FINAL_REPORT_v3.pdf

Figure 2: NESDIS Open Information Stewardship System (OISS) Open Knowledge Mesh Concept



Though OISS was still in development at the time of this study, the dev team was able leverage its AIU, AIC and DIP frameworks to successfully demonstrate the feasibility of integrating AI with OISS to help NESDIS enforce the FAIR standards emphasized by the EO-DT BAA as a requirement for effective information stewardship.

2.1 Goals and Objectives

The primary goal of this study and demonstration was to test the efficacy of using NLP and foundational models (e.g. LLMs) technology to enhance context and discoverability of disparate data sets for NOAA's knowledge mesh construction.

To achieve this, the dev team outlined three major objectives for the study:

- **Accommodate multi-modal data acquisition** (e.g., algal bloom data, project grants,policy, ecological effects) by streamlining the build and integration process for newdata pipelines, expanding the utility of the KM to multiple personas and use cases.
- **Enrich the KM and extract metadata more efficiently** for analysis of various data sources by integrating foundational models (e.g., LLMs) for use with the KM.
- **Provide context for the uncertainty of results** (e.g., extraction efficiencies, uncertainty by extraction type) by integrating tools that support data lineage and searchability.

Throughout the course of the study, a fourth objective emerged. It became clear that searching the KM was a clear mechanism to demonstrate the first three objectives and achieve the overall goal to improve the context and discoverability of disparate datasets. Hence, the dev team added a fourth objective to demonstrate their concept:

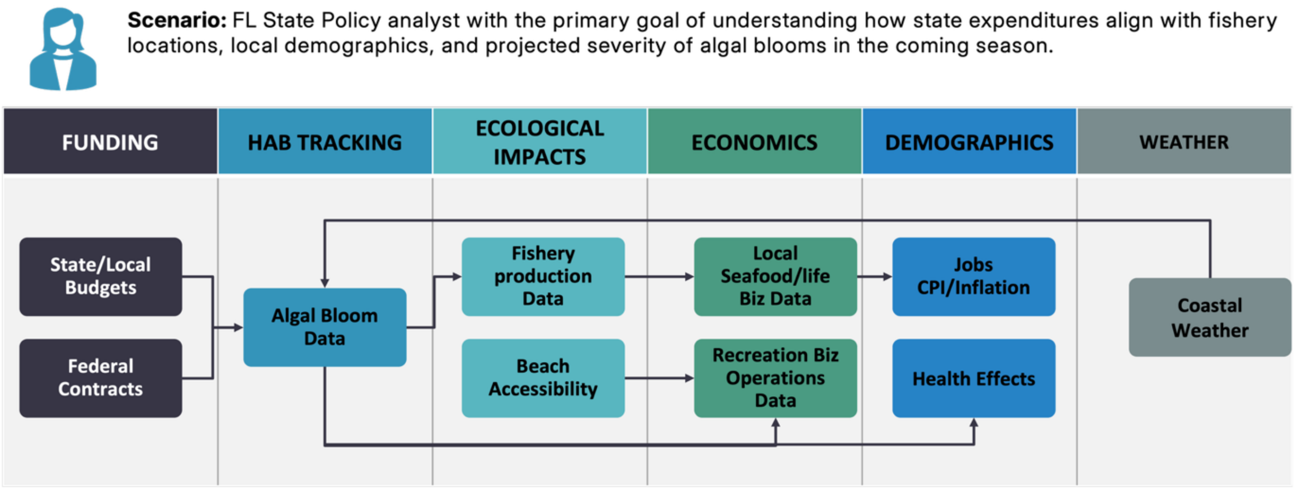
- **Demonstrate the prototype capabilities through an enhanced multi-modal search.** To achieve these objectives through a 12-month rapid prototyping effort, the study leveraged Booz Allen open-source tools, i.e., aiSSEMBLETM and GAMECHANGER as accelerators, to enable 1) automated data pipeline development for an ML-enriched KM; and 2) a visualization interface to simplify user adoption (UI). The prototype was designed and built for future scalability and expanded use cases, keeping in mind compatibility with OISS and the NCCF.

3. Project Scope and Methodology

3.1 Demo Scenario and User Personas

To demonstrate the goals outlined above and define the project scope, the dev team outlined the following scenario: *State policy analysts from a local government in Florida are trying to assess appropriate funding levels for projects to address how algal blooms impact fisheries and their local economies*. The team selected this scenario in response to recent federal mandates that increase demand for authoritative climate, geospatial, and earth science data for use in decision and policy making. Using principles of human-centered design (HCD), the team started by identifying target personas, conducting a literature review to understand the use case (HAB) domain, and conducted a mapping exercise of key questions that each user persona might be interested in. The team then validated the personas, key data sources, and user search journey with several subject matter experts through interviews, including NOAA fisheries SMEs, economists, and researchers. The scenario required data from multiple domains, including financial, environmental, recreational demographic and physical science (Figure 3); therefore, providing an ideal test bed for multi-modal acquisition and search.

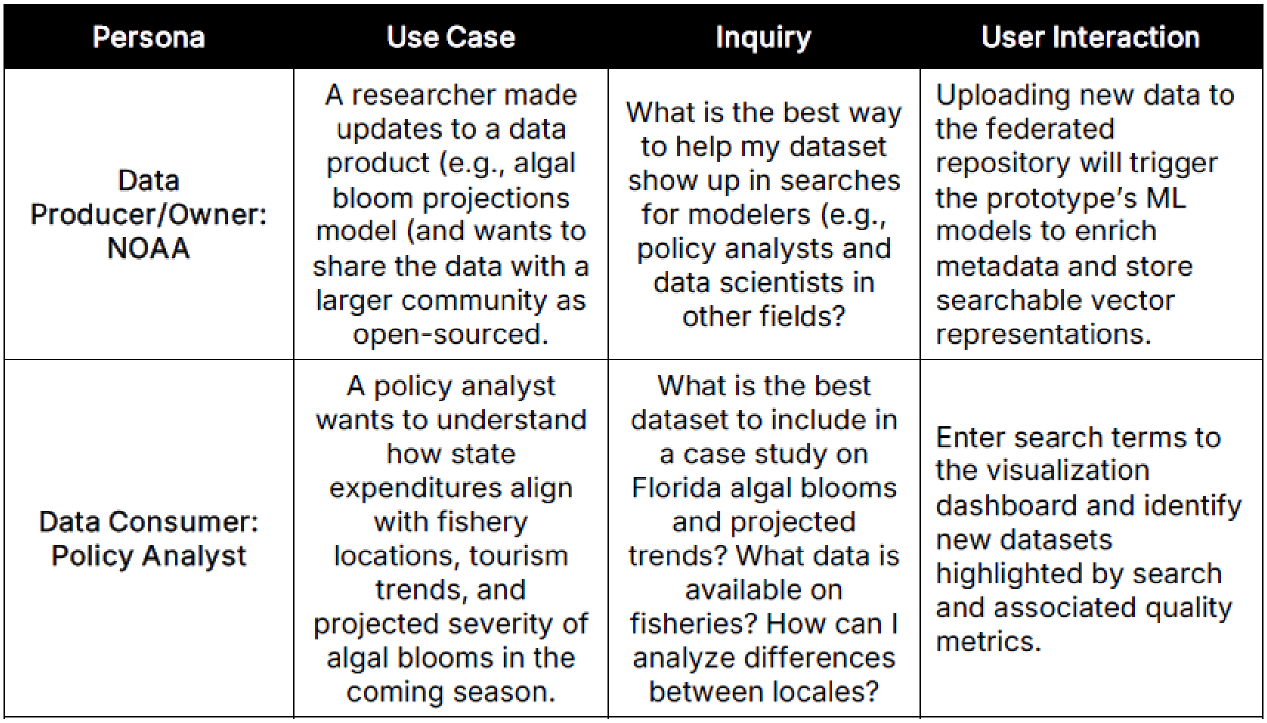
Figure 3: Motivational Scenario



Scenario: FL State Policy analyst with the primary goal of understanding how state expenditures align with fishery locations, local demographics, and projected severity of algal blooms in the coming season.

The prototype was tailored to address the project scenario from the perspective of two user personas, i) Data Owners (e.g., NOAA researchers) and ii) Data Consumers (e.g., policy analysts). User personas are detailed in Table 1. A comprehensive list of the data sets used can be found in Appendix A.

Table 1:User Personas, Use Cases, Inquiries and Interactions.



Persona	Use Case	Inquiry	User Interaction
Data Producer/Owner: NOAA	A researcher made updates to a data product (e.g., algal bloom projections model (and wants to share the data with a larger community as open-sourced.	What is the best way to help my dataset show up in searches for modelers (e.g., policy analysts and data scientists in other fields?	Uploading new data to the federated repository will trigger the prototype's ML models to enrich metadata and store searchable vector representations.
Data Consumer: Policy Analyst	A policy analyst wants to understand how state expenditures align with fishery locations, tourism trends, and projected severity of algal blooms in the coming season.	What is the best dataset to include in a case study on Florida algal blooms and projected trends? What data is available on fisheries? How can I analyze differences between locales?	Enter search terms to the visualization dashboard and identify new datasets highlighted by search and associated quality metrics.

3.2 Booz Allen Acceleralors

To accelerate the project timeline, the dev team integrated code baselines and patterns from two of Booz Allen's proven open-sourced engineering baselines with high technical readiness levels (TRLs) to rapidly prototype a scalable solution (see Appendix B for opensource repos). GAMECHANGER, an AI-powered information extraction, search, retrieval and analysis platform provided a user-tested and vetted UI and multiple AI-enhanced data extraction pipeline patterns to be able to quickly visualize the AI enhancements in a user experience demonstration. aiSSEMBLE™, Booz Allen's lean manufacturing approach to AI/ML operations, provided patterns for scaling and deploying AI-enhanced data transformation pipelines. The final prototype used elements from both solutions to enable future scalability and expandability for new use cases while maintaining compatibility with OISS and NCCF.

3.2.1 GAMECHANGER

Booz Allen's GAMECHANGER is a mature, production-grade platform designed to transform unstructured policy and knowledge assets into structured, actionable intelligence. Initially developed for Department of Defense (DoD) policy analysts, GAMECHANGER's modular architecture has been replicated across several use cases including contracts, budget and health. To date, GAMECHANGER is deployed for eight different federal enterprise use cases. It offers advanced Natural Language Processing (NLP) and search capabilities, providing a centralized environment to explore, compare, and analyze vast amounts of policy, guidance, and documentation. For the NOAA NESDIS Knowledge Mesh initiative, GAMECHANGER’s open-source codebase was cloned to provide an out-of-the box user-friendly interface for the KM prototype. It is available on https://github.com/dod-advana/gamechanger.

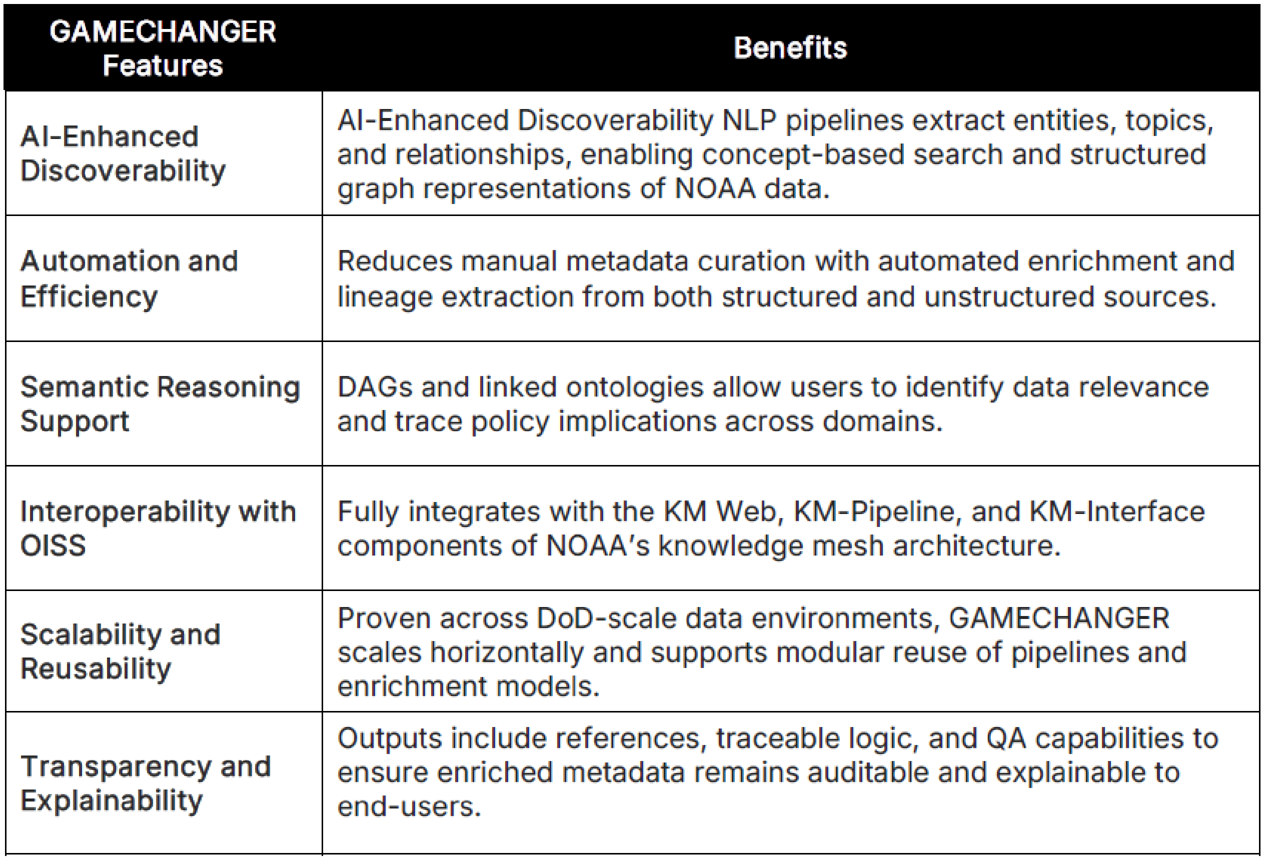
GAMECHANGER ingests static, semi-structured, and unstructured documents and applies a suite of NLP pipelines that include summarization, named entity recognition (NER), relationship extraction, ontology mapping, and knowledge graph creation. These capabilities are exposed through a fully developed web application that supports advanced semantic search, policy comparison, and visualization of metadata-driven connections between disparate datasets.

The platform supports integration with LLMs through agentic APIs, allowing it to dynamically extract structured knowledge representations from NOAA archival records. The enriched metadata is stored alongside the source data and indexed for multimodal search using OpenSearch, Amazon Neptune, and AWS Athena.

GAMECHANGER offerings and benefits are laid out in Table 2. It provides a modular microservice-based architecture compatible with containerized deployment and Kubernetes environments. Its plug-and-play enrichment services allow seamless integration with NOAA’s OISS, extending the agency’s metadata infrastructure with explainable, machine-readable insights.

Another key strength of GAMECHANGER is its user-centered design, offering personas, such as policy analysts and data stewards, intuitive access to OISS for both distribution and discovery. Within the UI multiple users can access datasets, processes, assess metadata quality, and filter by domain context or relevance. For NOAA’s environmental intelligence mission, this translates to actionable discovery across climate, oceanographic, demographic, and socioeconomic datasets.

Table 2: GAMECHANGER offerings and NOAA benefits



GAMECHANGER Features	Benefits
AI-Enhanced Discoverability	AI-Enhanced Discoverability NLP pipelines extract entities, topics, and relationships, enabling concept-based search and structured graph representations of NOAA data.
Automation and Efficiency	Reduces manual metadata curation with automated enrichment and lineage extraction from both structured and unstructured sources.
Semantic Reasoning Support	DAGs and linked ontologies allow users to identify data relevance and trace policy implications across domains.
Interoperability with OISS	Fully integrates with the KM Web, KM-Pipeline, and KM-Interface components of NOAA's knowledge mesh architecture.
Scalability and Reusability	Proven across DoD-scale data environments, GAMECHANGER scales horizontally and supports modular reuse of pipelines and enrichment models.
Transparency and Explainability	Outputs include references, traceable logic, and QA capabilities to ensure enriched metadata remains auditable and explainable to end-users.

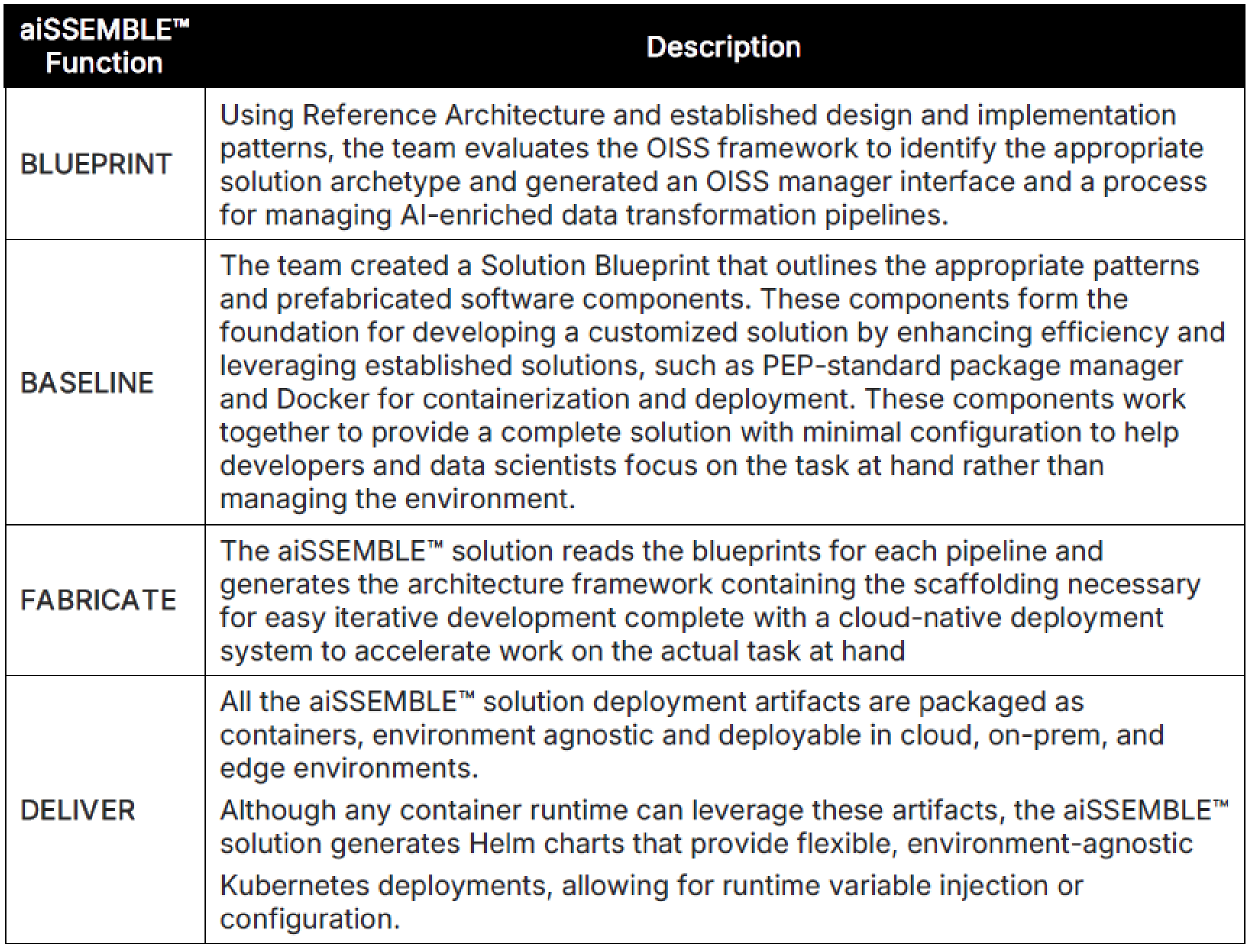
GAMECHANGER is interoperable with the NESDIS Common Cloud Framework (NCCF) and supports NOAA's long-term goal of developing resilient, AI-powered open science architecture.

3.2.2 aiSSEMBLE™

aiSSEMBLE™ is Booz Allen's lean manufacturing approach for holistically designing, developing and operating AI solutions across the engineering lifecycle. aiSSEMBLE™ leverages an intelligent automation engine that auto-generates over 50% of the code for data pipeline continuous integration and includes pre-wired capabilities for collecting metadata (e.g., provenance, lineage) and data quality. aiSSEMBLE™ provides configuration for Jenkins templates, leveraging a predefined Continuous Integration/Continuous Delivery (CI/CD) pipeline to build and deploy the pipelines. aiSSEMBLE™ was previously deployed for the NESDIS Ground Processing Demonstration (GPD) BAA to demonstrate successful cloud-optimized data processing and dissemination for use cases such as the MIDAPS-AI algorithm for processing satellite data. Documentation and a user guide were delivered to NESDIS, and public information can be found here https://github.com/boozallen/aissemble.

aiSSEMBLE™ standardizes and scales AI delivery through a framework built on open-source products. It empowers users, evolves constantly by embedding lessons learned, and provides the necessary components for comprehensive AI software systems and solutions (Table 3). The aiSSEMBLE™ framework integrates well with different relevant data producers to provide an enterprise-grade, secure, and ethical AI solution delivered with full government purpose rights. The aiSSEMBLE™ solution defines four primary functions to establish an AI baseline and drive engineering consistency:

Table 3: aiSSEMBLE™ Functions



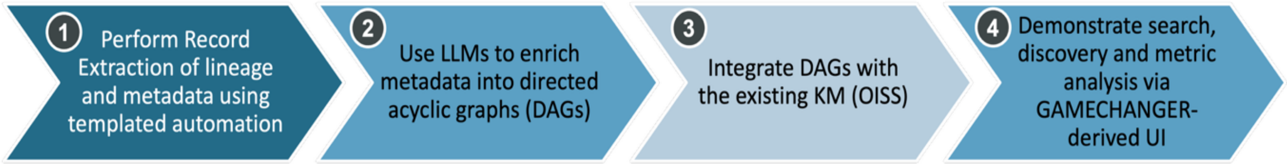
aiSSEMBLE™ Function	Description
BLUEPRINT	Using Reference Architecture and established design and implementation patterns, the team evaluates the OISS framework to identify the appropriate solution archetype and generated an OISS manager interface and a process for managing AI-enriched data transformation pipelines.
BASELINE	The team created a Solution Blueprint that outlines the appropriate patterns and prefabricated software components. These components form the foundation for developing a customized solution by enhancing efficiency and leveraging established solutions, such as PEP-standard package manager and Docker for containerization and deployment. These components work together to provide a complete solution with minimal configuration to help developers and data scientists focus on the task at hand rather than managing the environment.
FABRICATE	The aiSSEMBLE™ solution reads the blueprints for each pipeline and generates the architecture framework containing the scaffolding necessary for easy iterative development complete with a cloud-native deployment system to accelerate work on the actual task at hand
DELIVER	All the aiSSEMBLE™ solution deployment artifacts are packaged as containers, environment agnostic and deployable in cloud, on-prem, and edge environments. Although any container runtime can leverage these artifacts, the aiSSEMBLE™ solution generates Helm charts that provide flexible, environment-agnostic Kubernetes deployments, allowing for runtime variable injection or configuration.

For this short-term, exploratory project, the dev team used the aiSSEMBLE™ blueprint and baseline patterns to develop the solutions. These patterns set a foundation to be able to later mature the solution rapidly into Fabricate and Deliver functions for a production-grade deployment.

3.3 Study Methodology

The KM prototype development effort was broken down into four main tasks to accomplish the three defined study objectives to: 1) accommodate multi-modal data, 2) enrich and extract metadata for efficiency, 3) provide context for uncertainty, and achieve expected outcomes outlined in Figure 4 below.

Figure 4: Tasks to Achieve Study Goals



1 Perform Record Extraction of lineage and metadata using templated automation	2 Use LLMs to enrich metadata into directed acyclic graphs (DAGs)	3 Integrate DAGs with the existing KM (OISS)	4 Demonstrate search, discovery and metric analysis via GAMECHANGER-derived UI
EXPECTED OUTCOMES			
• Show that automation expedites multi-modal metadata extraction	• Show that LLM-enrichment can contextual multi-modal data types • Demonstrate the early-stage extraction of data quality metrics and provenance data	• Show how LLM-generated DAGS augment the KM	• Demonstrate how LLM enrichment and integration of DAGs into the KM enhances discoverability • Show that metrics can be displayed to inform users
VALUE DELIVERED			
• Expedites adding multi-modal data and supports new use cases and personas to grow KM **adoption and scale**	• **More efficient** metadata enrichment and uncertainty allows for benchmarking improvements	• Context and lineage generate **more robust KM records**	• UI enables user interaction with KM to iterate and **build trust** in its context and bring validity to search results

3.3.1 Automated Record Extraction

The initial task was integration between the baseline solutions and the NOAA KM. While the KM came packaged with example metadata models of various data types, including satellite assets and structured data, the dev team recognized the need to expand on those patterns to enable rapid integration of multi-source datasets for different domains. This required outlining a generalized governance schema for AIU records and allowing data owners to subscribe to new datasets by specifying additional domain-relevant features. With a KM management framework, the dev team could then automate dataset template generation and record metadata extraction.

3.3.2 Application of LLM for Metadata Enrichment

Next, the dev team used LLMs to generate domain-relevant information about the extracted record metadata. Using the KM process deployment and invocation mechanism, the AIC, the dev team interfaced its dynamic AI pipeline APIs, making them available to process pre-determined data types (AIUs) as directed acyclic graphs (DAGs). For simplicity, this study processed structured and unstructured documents. The AI enrichment processes included text summarizations and embedding generation, image extraction and auto-captioning, entity extraction, and linking related ontology classes. The API used aiSSEMBLE™ patterns to enforce modularity and configurability to enable model swapping for different use cases and environments, and scalability to extend to larger record loads. The team tested the capability with multiple open-source LLMs, including models available in Ollama, and on proprietary models available in AWS Bedrock. In adherence to NOAA's open-science principles, the AI APIs generated quality metrics for downstream reporting (see *Capability Deep Dive* for a description of the models and quality metrics used in this study).

3.3.3 Knowledge Mesh Integration

The dev team deployed the OISS service in their internally provisioned AWS enterprise environment to successfully test metadata integration and search. Out of the box, OISS has the mechanisms to generate AIU, AICs and DIPs and stores their templates (dataset descriptions) in Neptune, making dataset descriptions and processes searchable through graph query languages, such as SPARQL or Cypher. However, records and AIC-generated metadata are stored in S3, which is challenging to search in lieu of having access to the S3 metadata during object creation. Therefore, to be able to demonstrate the enhanced metadata in a search user flow, the team expanded the KM ecosystem to include the vector and text search index capabilities of AWS OpenSearch.

The full KM implementation is thus comprised of the dev team's developed governance management implemented in AIU templates, automated metadata extraction, AI-enhanced augmentation of metadata, and an extension of the native OISS storage infrastructure of S3 and Neptune with OpenSearch. The dev team further linked the knowledge graph representation of records with LLM-enriched data to their reference record node IDs. This resulted in an interoperable, searchable data ecosystem.

3.3.4 Visualization UI: Dynamic KM Visualization Tool

The dev team used GAMECHANGER's out-of-the-box UI and visualization tools to accelerate development. GAMECHANGER's keyword and intelligent search backend enabled the rapid integration of the updated KM ecosystem, developed for full KM integration. This ecosystem includes AI-assisted multi-modal search components including RAG-based search leveraging OpenSearch and text-to-SPARQL graph querying using Neptune. Integrating both an index search, i.e. OpenSearch, and a graph database, i.e. Neptune, were key to demonstrate how NOAA users can explore relationships between datasets, entities, and topics within the KM to connect disparate datasets while discovering them with contextual natural language queries. In addition, the UI supported seamless data uploads for data owners, allowing them to observe data quality metrics and explore processes available for data transformation. This capability provided multiple interfaces to engage broad spectrum of users.

4. Capability Deep Dive

4.1 Solution Microservices

To deliver NOAA’s data processing and dissemination solutions, the dev team implemented three microservice solutions, KM Web, KM-Interface and KM-Pipeline. The KM Web UI enables seamless integration for data owners to subscribe to new datasets, such as archives or real-time systems, to OISS and allows data consumers to perform context-based queries for aggregation of information across multiple domains. KM-Interface is a control and monitoring system for OISS. KM-Interface defines and enforces governance standards, creates new templates for new datasets and triggers record uploads and metadata extractions to be handled by the third microservice KM-Pipeline, the NLP/LLM pipeline API. Table 4 outlines the key features and technology integrated into each microservice. The phased execution approach is detailed below.

Table 4: NOAA KM Microservice Components and Capabilities

Microservice	Description	Technology	Key Features
KM Web	Booz Allen’s NLP-driven frontend interface for semantic metadata search	• GAMECHAN GER • PGSQL	• Policy-aligned UI • Conceptual queries
KM-Interface	OISS command and control interface designed to simplify integration of new data types and track data lineage.	• OISS API • Python/Fast API • Docker • Elastic Container Service	• Simplified multi-modal data integration into the OISS KM • Automatic lineage tracking

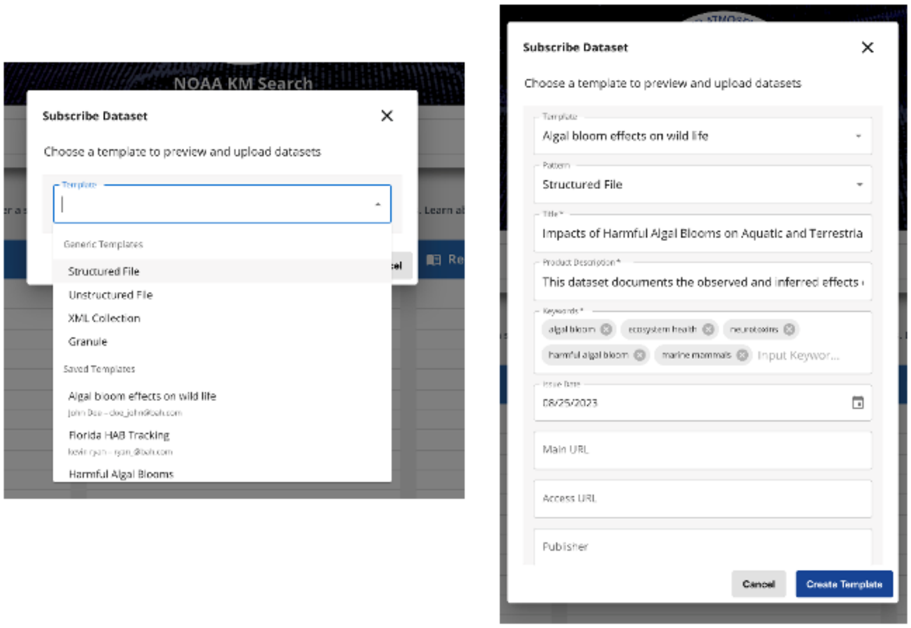
Microservice	Description	Technology	Key Features
KM-Pipeline	An AI API that enriches metadata records and implements multi-modal search using.	• Python/Fast API • Bedrock/AWS Bedrock • Celery • OpenSearch • Neptune • Docker • Elastic Container Service	• DAG-framework implementation • AI-Enrichment(LLM-based text/image summarization, entity extraction) • Ontology-linking • LLM-based text-to-SPARQL query • S3 metadata search

4.2 User Persona Workflows

4.2.1 Data Owner Persona

Dataset Subscription, Registration and Record Upload & Extraction: Here data owners may initiate their first interaction with the KM by subscribing, or registering, archival datasets by sharing general access, origin and usage-related metadata.

Figure 5: KM-Web Dataset subscription and record upload view



NOAA KM Search

Subscribe Dataset

Choose a template to preview and upload datasets

Template

Generic Templates

- Structured File
- Unstructured File
- XML Collection
- Granule

Saved Templates

- Algal bloom effects on wild life
John Doe – doe_john@bah.com
- Florida HAB Tracking
kevin ryan – ryan_@bah.com
- Harmful Algal Blooms

Subscribe Dataset

Choose a template to preview and upload datasets

Template: Algal bloom effects on wild life

Pattern: Structured File

Title*: Impacts of Harmful Algal Blooms on Aquatic and Terrestria

Product Description*: This dataset documents the observed and inferred effects

Keywords*: algal bloom, ecosystem health, neurotoxins, harmful algal bloom, marine mammals

Issue Date: 08/25/2023

Main URL

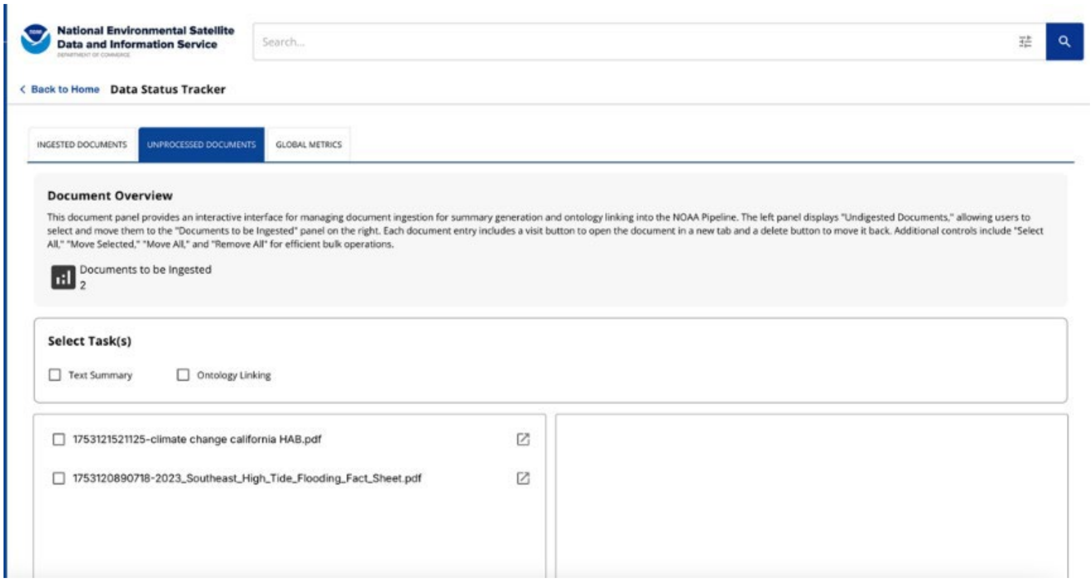
Access URL

Publisher

Behind the scenes, these registered NOAA and non-NOAA datasets, such as the Harmful Algal Blooms Observing System (HABSOS), Florida Fish and Wildlife HAB Tracker, Florida Climate Action Plan and U.S. Census data, are converted to OISS AIU templates that are stored as OISS Core and OISS Framework ontology graphs in Neptune. Once a data set is subscribed, data owners may upload records or in a future version, register an access location such as a link or s3 bucket. Upon submission: 1) Template metadata is generated with specification for the record data (i.e. issue date, checksum value); and 2) Record metadata graphs are stored in s3.

Data Status Validation and LLM Meta Data Enrichment: The data owner, or in a future version, any user with access rights, accesses the Data Status Tracker (Figure 6) to view their uploaded records.

Figure 6: Data Status Tracker Unprocessed Documents View



Document Overview

This document panel provides an interactive interface for managing document ingestion for summary generation and ontology linking into the NOAA Pipeline. The left panel displays "Undigested Documents," allowing users to select and move them to the "Documents to be Ingested" panel on the right. Each document entry includes a visit button to open the document in a new tab and a delete button to move it back. Additional controls include "Select All," "Move Selected," "Move All," and "Remove All" for efficient bulk operations.

Documents to be Ingested
2

Select Task(s)

□ Text Summary □ Ontology Linking

□ 1753121521125-climate change california HAB.pdf

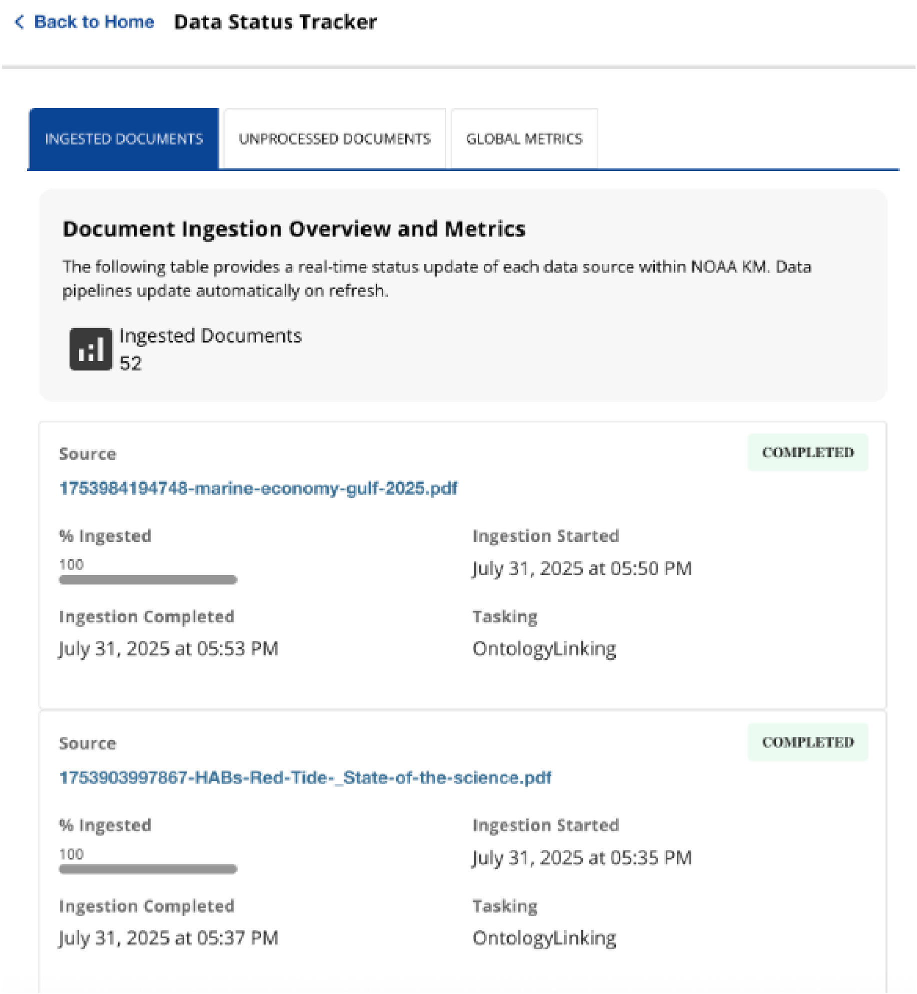
□ 1753120890718-2023_Southeast_High_Tide_Flooding_Fact_Sheet.pdf

Data owners can use the status tracker to choose one or more of the processes available to transform their data records based on the data type (the section below on Knowledge Mesh Integration details the AI-enrichment and ontology-related processes implemented in this study).

Once they select their processes, behind the scenes this launches user business logic that has been registered to OISS through its AIC framework to run a DAG-like AI process. The AIC process transforms the AIU metadata, and any accessible related archive data into supplemental information that is added to the knowledge mesh as member description edges to the associated AIU record.

These enrichment steps enhance discoverability and support initial provenance capture by relating the namespace or metadata of the AI-generated member description to an OISS AIU record ID, which persisted in Amazon S3, and, in the prototype, indexed into OpenSearch for semantic and graph-based search. This supports NOAA's FAIR/CARE compliance and aligns with NESDIS's goals for scaling access to authoritative environmental data.

Figure 7: Data Status Tracker Ingested Documents View



Document Ingestion Overview and Metrics

The following table provides a real-time status update of each data source within NOAA KM. Data pipelines update automatically on refresh.

Ingested Documents
52

Source
1753984194748-marine-economy-gulf-2025.pdf
COMPLETED

% Ingested
100

Ingestion Started
July 31, 2025 at 05:50 PM

Ingestion Completed
July 31, 2025 at 05:53 PM

Tasking
OntologyLinking

Source
1753903997867-HABs-Red-Tide-_State-of-the-science.pdf
COMPLETED

% Ingested
100

Ingestion Started
July 31, 2025 at 05:35 PM

Ingestion Completed
July 31, 2025 at 05:37 PM

Tasking
OntologyLinking

4.2.2 Data User Persona

User Query: The workflow begins with a data user, such as a state policy analyst who endeavors to access the level of funding needed to address HABs impact on local public health, business and recreational effects. A user's interactions are numbered below.

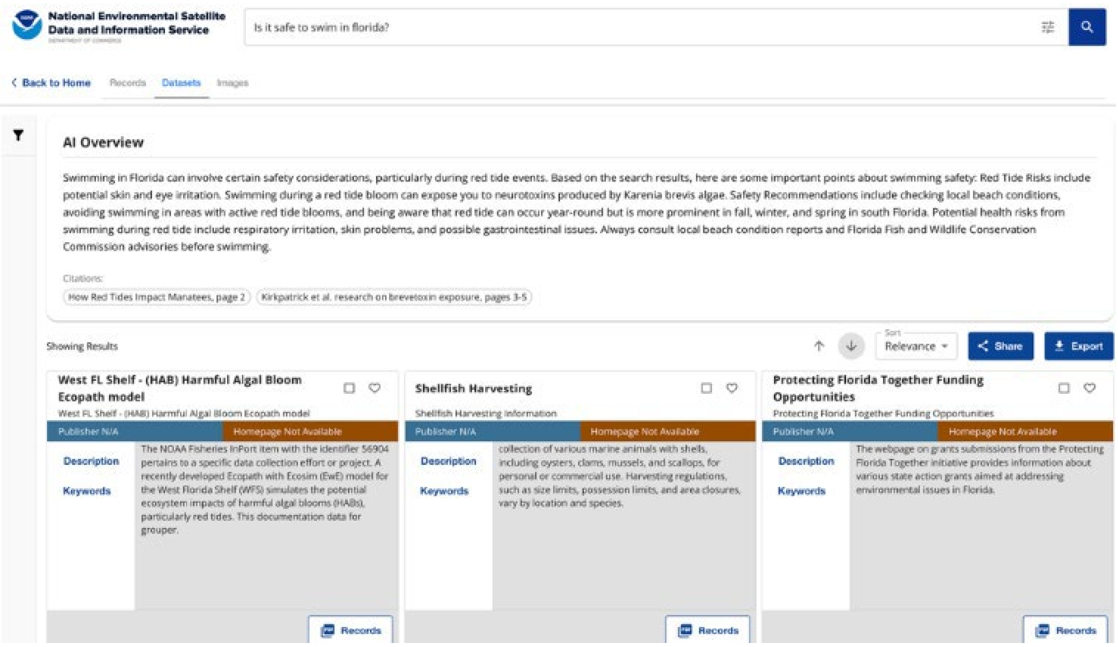
1. The User may submit a contextual query such as: "What is the best dataset to include in a case study for Florida on algal blooms and projected climate trends?"
2. This query is then processed through the KM Web's front-end interface, which targets subscribed datasets from both NOAA and external publishers relevant to fisheries, climate trends, equity impacts, and demographics.
3. The front-end UI leverages components from the GAMECHANGER platform,

which provides interactive search, topic navigation, and exploratory graph-based visualizations across structured and unstructured data

Query Routing: The search query is routed to the KM-Pipeline Search API. Depending on the context of the query and filters, KM-Pipeline forwarded the query to one or more pipelines within its multimodal search orchestration stack. This includes:

- Text-to-SPARQL Query Builder for accessing AIU (or AIC in a future version) templates stored in Neptune to enable dataset search (Figure 8).

Figure 8: KM-Web App Dataset Search View



Is it safe to swim in florida?

AI Overview

Swimming in Florida can involve certain safety considerations, particularly during red tide events. Based on the search results, here are some important points about swimming safety: Red Tide Risks include potential skin and eye irritation. Swimming during a red tide bloom can expose you to neurotoxins produced by Karenia brevis algae. Safety Recommendations include checking local beach conditions, avoiding swimming in areas with active red tide blooms, and being aware that red tide can occur year-round but is more prominent in fall, winter, and spring in south Florida. Potential health risks from swimming during red tide include respiratory irritation, skin problems, and possible gastrointestinal issues. Always consult local beach condition reports and Florida Fish and Wildlife Conservation Commission advisories before swimming.

Citations:
How Red Tides Impact Manatees, page 2
Kirkpatrick et al. research on brevetoxin exposure, pages 3-5

Showing Results

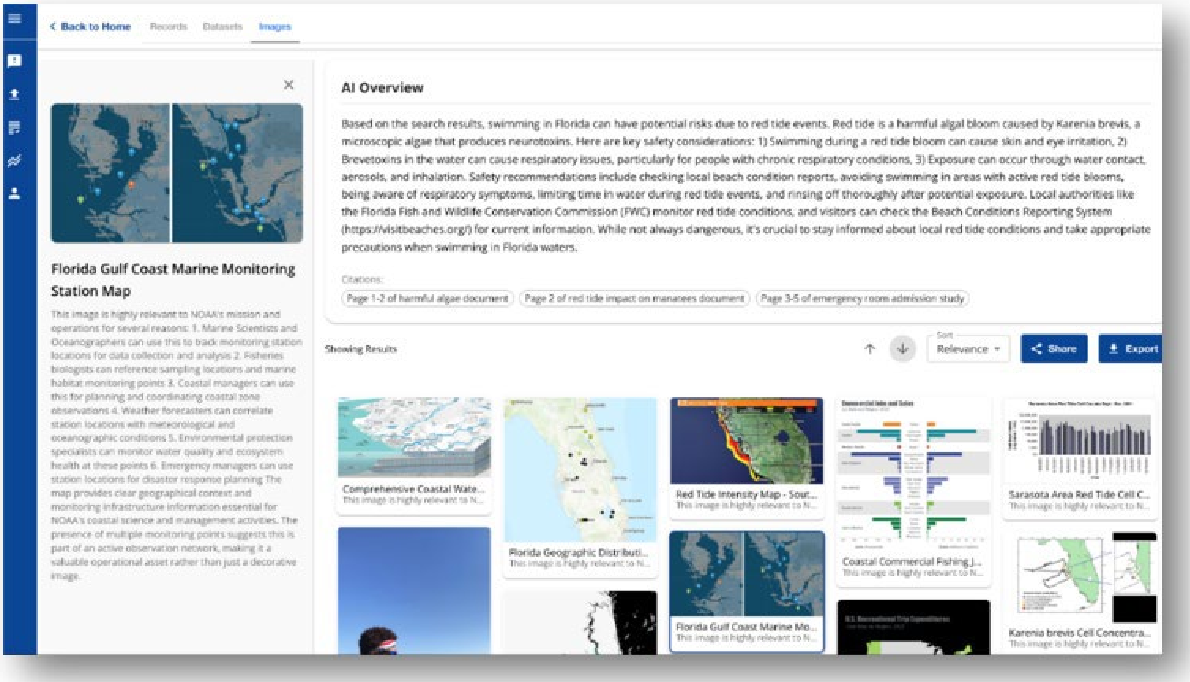
West FL Shelf - (HAB) Harmful Algal Bloom Ecopath model
West FL Shelf - (HAB) Harmful Algal Bloom Ecopath model
Publisher N/A | Homepage Not Available
Description
Keywords
The NOAA Fisheries InPort item with the identifier 56904 pertains to a specific data collection effort or project. A recently developed Ecopath with Ecosim (EwE) model for the West Florida Shelf (WFS) simulates the potential ecosystem impacts of harmful algal blooms (HABs), particularly red tides. This documentation data for grouper.

Shellfish Harvesting
Shellfish Harvesting information
Publisher N/A | Homepage Not Available
Description
Keywords
collection of various marine animals with shells, including oysters, clams, mussels, and scallops, for personal or commercial use. Harvesting regulations, such as size limits, possession limits, and area closures, vary by location and species.

Protecting Florida Together Funding Opportunities
Protecting Florida Together Funding Opportunities
Publisher N/A | Homepage Not Available
Description
Keywords
The webpage on grants submissions from the Protecting Florida Together initiative provides information about various state action grants aimed at addressing environmental issues in Florida.

- A Q/A chatbot that takes question-like queries and performs Retrieval-Augmented Generation (RAG)-based summarization (see “AI Overview” in Figure 8).
- Text and vector similarity matching across embeddings stored in OpenSearch to retrieve generative AI-captioned images and records (Figure 9).

Figure 9: KM-Web App Image Search View



Back to Home
Records
Datasets
Images
Showing Results
Sort
Relevance
Share
Export
Comprehensive Coastal Wate...
Florida Geographic Distributi...
Red Tide Intensity Map - Sout...
Coastal Commercial Fishing J...
Sarasota Area Red Tide Cell C...
Florida Gulf Coast Marine Mo...
Karenia brevis Cell Concentra...
This image is highly relevant to N...

Florida Gulf Coast Marine Monitoring Station Map

This image is highly relevant to NOAA's mission and operations for several reasons: 1. Marine Scientists and Oceanographers can use this to track monitoring station locations for data collection and analysis 2. Fisheries biologists can reference sampling locations and marine habitat monitoring points 3. Coastal managers can use this for planning and coordinating coastal zone observations 4. Weather forecasters can correlate station locations with meteorological and oceanographic conditions 5. Environmental protection specialists can monitor water quality and ecosystem health at these points 6. Emergency managers can use station locations for disaster response planning The map provides clear geographical context and monitoring infrastructure information essential for NOAA's coastal science and management activities. The presence of multiple monitoring points suggests this is part of an active observation network, making it a valuable operational asset rather than just a decorative image.

AI Overview

Based on the search results, swimming in Florida can have potential risks due to red tide events. Red tide is a harmful algal bloom caused by Karenia brevis, a microscopic algae that produces neurotoxins. Here are key safety considerations: 1) Swimming during a red tide bloom can cause skin and eye irritation, 2) Brevetoxins in the water can cause respiratory issues, particularly for people with chronic respiratory conditions, 3) Exposure can occur through water contact, aerosols, and inhalation. Safety recommendations include checking local beach condition reports, avoiding swimming in areas with active red tide blooms, being aware of respiratory symptoms, limiting time in water during red tide events, and rinsing off thoroughly after potential exposure. Local authorities like the Florida Fish and Wildlife Conservation Commission (FWC) monitor red tide conditions, and visitors can check the Beach Conditions Reporting System (https://visitbeaches.org/) for current information. While not always dangerous, it's crucial to stay informed about local red tide conditions and take appropriate precautions when swimming in Florida waters.

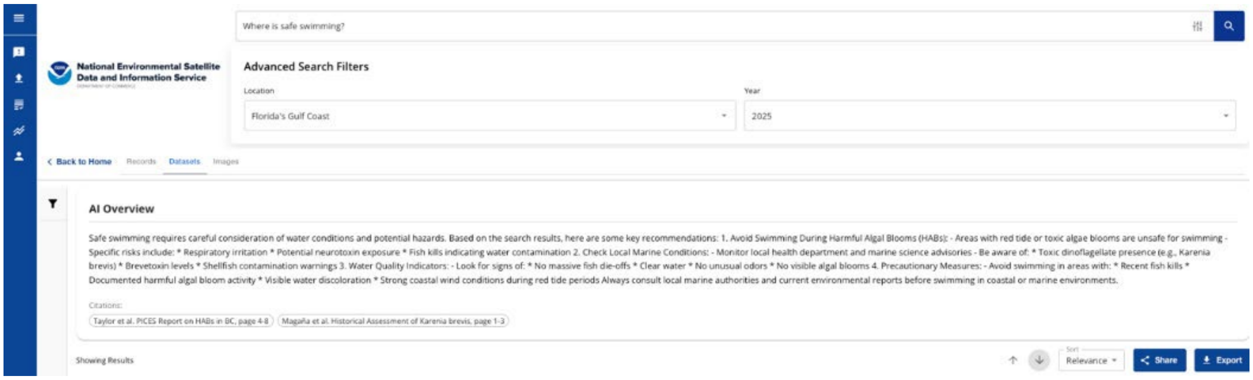
Citations:

Page 1-2 of harmful algae document | Page 2 of red tide impact on manatees document | Page 3-5 of emergency room admission study

This phase operationalized NOAA's open-science goals by enabling intuitive semantic search across domain-linked metadata.

Search Execution, Graph Filtering, and Visual Exploration: A user can view results through the KM Web dashboard, where GAMECHANGER-powered visualization tools display datasets, records and related media (i.e. images) generated by the AI-enhanced AICs. Faceted filters facilitated advanced searching, enabling a user to drill into datasets by date, time and location (Figure 10). The user can refine their search iteratively via a feedback loop—enabling users to favorite key datasets or flag those of low utility.

Figure 10: KM-Web App Search Filters



Location: Florida's Gulf Coast

Year: 2025

AI Overview

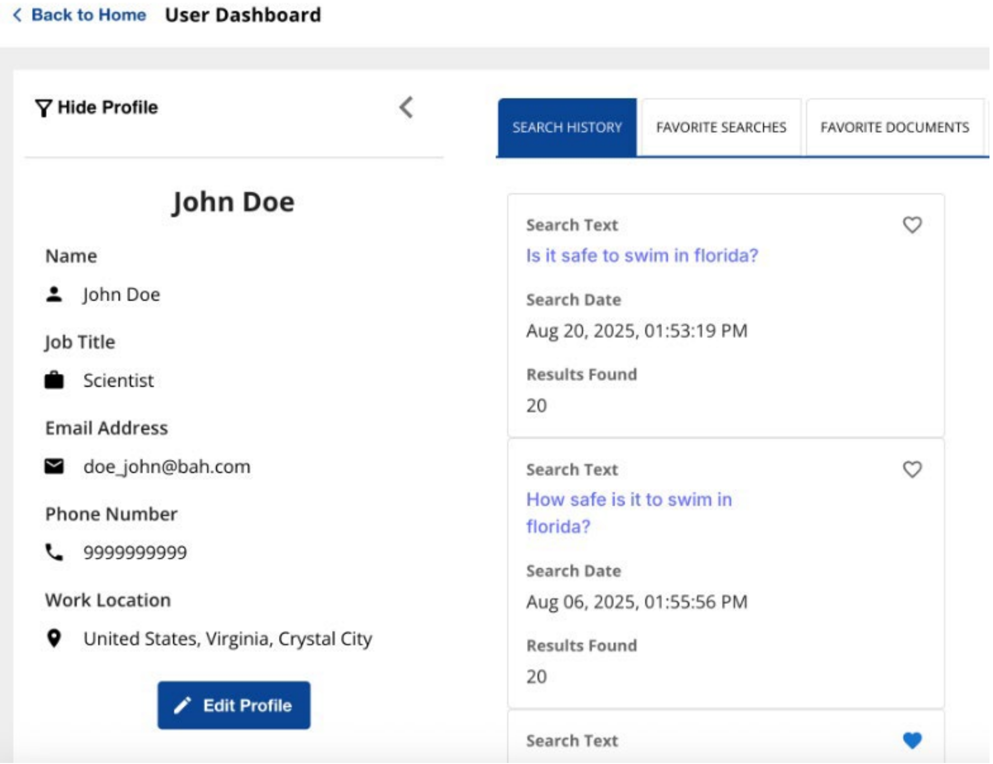
Safe swimming requires careful consideration of water conditions and potential hazards. Based on the search results, here are some key recommendations: 1. Avoid Swimming During Harmful Algal Blooms (HABs): - Areas with red tide or toxic algae blooms are unsafe for swimming - Specific risks include: * Respiratory irritation * Potential neurotoxin exposure * Fish kills indicating water contamination 2. Check Local Marine Conditions: - Monitor local health department and marine science advisories - Be aware of: * Toxic dinoflagellate presence (e.g., Karenia brevis) * Brevetoxin levels * Shellfish contamination warnings 3. Water Quality Indicators: - Look for signs of: * No massive fish die-offs * Clear water * No unusual odors * No visible algal blooms 4. Precautionary Measures: - Avoid swimming in areas with: * Recent fish kills * Documented harmful algal bloom activity * Visible water discoloration * Strong coastal wind conditions during red tide periods Always consult local marine authorities and current environmental reports before swimming in coastal or marine environments.

Citations:
Taylor et al. PICES Report on HABs in BC, page 4-8
Magaña et al. Historical Assessment of Karenia brevis, page 1-3

Sort: Relevance

User Profile and Search Support: The User Profile helps the user analyze previous searches and select documents, thus enabling a common interface to link disparate, cross-domain data alignment fulfilling NOAA's goal of supporting climate services and environmental intelligence (Figure 11). The integrated historical view improves the likelihood of adoption of the knowledge mesh by lowering the complexity of the typical analyst content generation workflow. Users can manage their search portfolio through their user profile, enabling search history tracking, search and document favorites, and export tracking.

Figure 11: KM-Web App User Profile



John Doe

Name: John Doe

Job Title: Scientist

Email Address: doe_john@bah.com

Phone Number: 9999999999

Work Location: United States, Virginia, Crystal City

Search Text: Is it safe to swim in florida?
Search Date: Aug 20, 2025, 01:53:19 PM
Results Found: 20

Search Text: How safe is it to swim in florida?
Search Date: Aug 06, 2025, 01:55:56 PM
Results Found: 20

Search Text

This supports a data user's ability to compile content. Future features may include:

- Annotation tools for policy memos or reports
- Integration with a scenario generator (e.g., for economic modeling of HABs and equity impacts across Florida)
- Linking favorite documents or datasets with AI-enhanced processing pipelines implemented as AICs.

This end-to-end HAB use case demonstrates how the prototype supports data owners by automating uploads and enriching metadata, while also empowering policy analyst data users to perform flexible, AI-enhanced discovery. It is designed to scale with the KM-Web, KM-Interface, and KM-Pipeline microservices, incorporating patterns from aiSSEMBLE™ and GAMECHANGER visualization. Ultimately, the prototype advances NESDIS and NOAA's open science and equity-driven decision-making mandates.

4.3 Record Extraction

4.3.1 AIU Template Generation and Deployment

Integration into OISS, the NOAA knowledge mesh, requires adoption of its AIU pattern/template infrastructure. AIU patterns are the mechanism through which organizations define the canonical metadata schema of a data type. The pattern schema defines information relevant the specific data type, such as the sample resolution of structured data, or the pixel resolution of images; domain relevant information, such as data provider, or wind speed for buoy-based sea state data; and general governance data to capture access rights, ownership and provenance.

AIU templates define datasets of specific AIU patterns by assigning default values to the data type patterns. Templates may also remain generic. Within OISS, users can define their own patterns, defining schema field types within the specifications of the OAIS standards. Users may then deploy templates of their defined patterns to a specified graph database where OISS encodes them as semantic graphs within the Core and Framework ontologies. Users may also add references to other domain-relevant ontology classes, though the most effective means of adding domain ontologies is still a research item. The dev team adopted this generic template to speed initial integration with OISS.

The dev team later updated KM-Interface integration with OISS to enable dynamic template generation by defining patterns via a configuration file. This configuration file structure is designed to be expanded for more dynamic use cases; however, its current iteration enables OISS managers to rapidly define patterns for new datatypes that can then be used by data owners to generate and deploy a template through KM-Web.

4.3.2 Record Uploads

Record uploads occur once a template is deployed. Whether a generic template for a specific data type (e.g. pdf file, CSV file, satellite image) or a named dataset template, such as any of the datasets outlined in Appendix A, records represent an instance or archive of a template. When a record is uploaded, its metadata is stored as OISS Ontology graphs uploaded to S3 at JSON-LD files. The JSON-LD stores the individual resource identifier (IRI) of its template.

4.4 LLM Enrichment

4.4.1 AIC Template Generation and Deployment

In OISS, AIC Templates are generated as secondary processes to storing AIUs. For this effort, AICs were decoupled from their respective AIUs and launched by generic pass-through AIUs to simplify automation. The pass-through AIUs were populated with the corresponding AIUs reference information. AIC configuration files define the APIs inputs and accepted API response as a single output, giving OISS managers working with KM-Interface flexibility in how they choose to handle data returned from KM-Pipeline. The outputs are also routed through the Data and Integration Process (DIP), discussed in section 5.5.

OISS currently launches AIC User Business Logic (UBL) processes through AWS Step functions. Step functions can then reference an AWS Lambda function to enable functionality. This project used the Step functions and Lambda functions as conduits to the KM-Pipeline API. The API then managed scaling and compute requirements through a well-defined token and computer configuration within its container deployment. LLM processes were implemented in a modular way to allow for flexible model context protocol management. The final implementation used Bedrock to enable controlled scaling.

4.4.2 Gen AI Summarization

KM-Pipeline implements two main AI-assisted enhancement pipelines. The first performs supplemental document archive summarization. KM-Pipeline implements this through a single API with parameterization to select among from summarization options, (i.e. document summarization and embedding; image extraction and auto-captioning, named Entity Recognition (NER)).

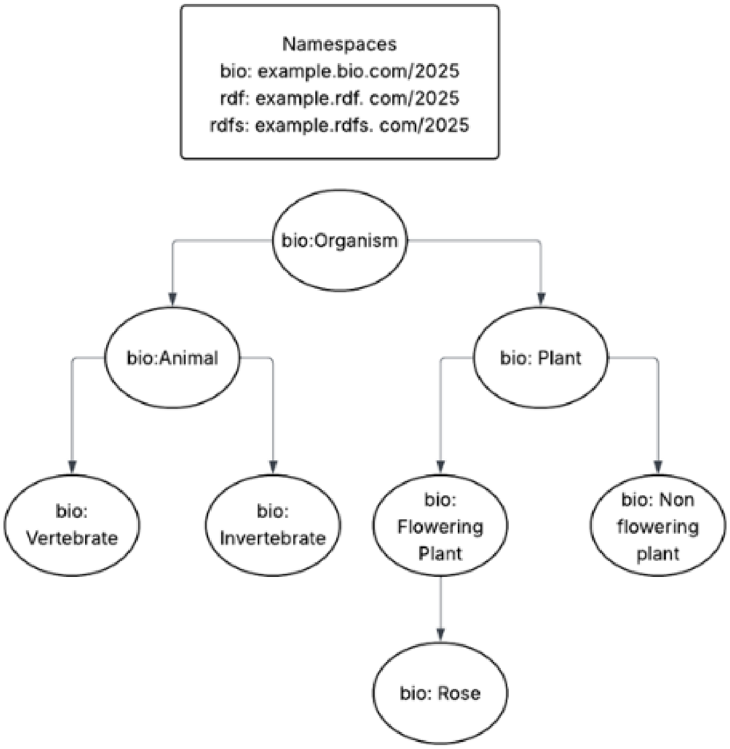
To perform comprehensive document summarization, KM-Pipeline uses AWS BedRock to invoke an Anthropic LLM. The model was able to generate subsection document chunks and perform individual summarizations for each section that were then routed to an OpenSearch index with their LLM-generated vector embeddings (also LLM generated).

By selecting the multi-modal option for the KM-Pipeline summarization API, KM-Pipeline performs document image extraction, summarization and domain-level relevancy assessment. The current implementation uses Anthropic’s multi-modal model to perform these image tasks with additional classification for domain-level relevancy. Subsection-level document summary and comprehensive document summary reason over data and generate entities as is.

4.5 LLM-Assisted Ontology Linking

Domain-specific ontologies are valuable in knowledge mesh construction and exploitation for helping classify and identify relationships between metadata. Augmenting the KM with multiple ontologies and linking them to records adds breadth and depth to searching a KM. A namespace is a fundamental concept in computing and programming that helps in organizing and managing identifiers such as variables, functions, classes, and other entities. It ensures that all identifiers within a given set are unique, which promotes high data governance and quality to support data sharing. A simple ontology can be found in Figure 12.

Figure 12: Simple Ontology Linking



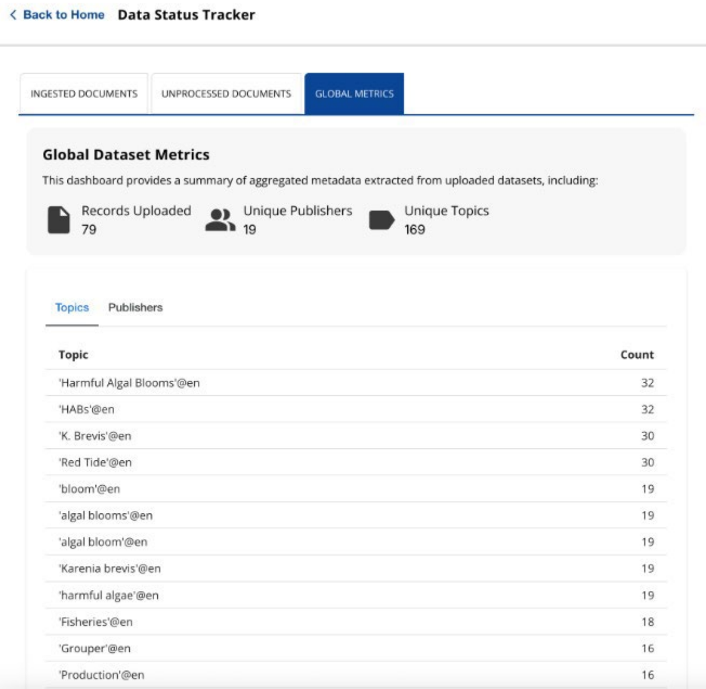
In this effort, the dev team created an exploratory AIC process to automatically associate record metadata, including keywords, entities, and schemas, with the most relevant namespace and class in a predetermined set of ontologies. While this capability was implemented as an AIC to add additional edges to the AIU's graph, in the future it could be incorporated as an augmentation to the AIU template during template generation and deployment. Ideally, namespaces would be associated with datasets at the template creation stage. Class and property assignments could then occur at template creation for keywords and schemas in structured data, or during the record upload stage for entities and other context extracted from documents. Currently, ontology linking is performed at the record upload stage, but the process should be updated to enable initial namespace association during template creation.

Additionally, ontology data associated at the record upload stage must be made searchable. One approach is to push this data to S3 object metadata and enable search via Athena, Redshift, or Apache Spark. The KM provides a viable testbed for investigating this approach.

4.6 Data Status Tracker

The Data Status Tracker is the primary interface KM-Web uses to inform data owners of their AIU upload status. From the Tracker, users can view and transform their uploaded records as described above. Users may also visualize global stats that are accessed through the KM-Interface Global Stats API. The Global Stats API is an extendable, parameterized data analyzer that uses DIPs generated from AIU (and potentially AIC) records uploads. In this project, DIPs store the schema data for AIUs and additional data outlined by the governance provenance standards to estimate upload statistics and generate uncertainty metrics. If, for instance the ontology-linking function were moved to the AIU level, uncertainty estimates would be propagated to the DIPs where they can be made accessible through the KM-Interface API.

Figure 13: Data Status Tracker



Data Status Tracker

Global Dataset Metrics

This dashboard provides a summary of aggregated metadata extracted from uploaded datasets, including:

Records Uploaded 79

Unique Publishers 19

Unique Topics 169

Topic	Count
'Harmful Algal Blooms'@en	32
'HABs'@en	32
'K. Brevis'@en	30
'Red Tide'@en	30
'bloom'@en	19
'algal blooms'@en	19
'algal bloom'@en	19
'Karenia brevis'@en	19
'harmful algae'@en	19
'Fisheries'@en	18
'Grouper'@en	16
'Production'@en	16

The current Global Stats view provides a summary of publishers, topics, and uploads as a proof of concept. These are generated by running a persistent backend task that pulls metadata from multiple DIPs and saving summary statistics to S3. This approach was chosen to improve scale since attempting to calculate global stats during a KM-Web request would cause too much latency. This API can be redesigned however KM Managers like to move some or all statistical analysis processes from the backend for real time processing purposes.

A similar API could be used to support estimating uncertainty or confidence during searches, either using global stats, benchmarking data or error estimates for linked ontology classes and properties.

4.7 KM Integration and Search

While building a comprehensive search capability was not the focus of this work, it was an inevitable prerequisite to demonstrate that AI enhancements improve content discoverability. Search is done at two levels, 1. OISS templates and 2. OISS record metadata. Template search (1) enables discovery of named datasets (AIUs) and algorithms (AICs). Record metadata search (2) enables discovery of records that are uploaded as instances of a named dataset (AIU) template. Templates are stored in the Neptune graph database and, therefore, searched using graph query language, such as SPARQL. Record metadata are stored in S3 and, when enhanced by LLMs through AICs, indexed in OpenSearch.

This project successfully implemented a text-to-SPARQL to enable template/dataset searches. For record search, the dev team combined NLP and LLM-based search techniques to provide different types of results based on the user's interaction with KM-Web. For simple text search, the dev team used benchmarking to determine an optimal way to combine LLM and OpenSearch results into a *Hybrid* Search. Each search method is implemented as an API in KM-Pipeline and described in detail below.

4.7.1 AIU Templates and Dataset Search

With proper configuration of governance, the OISS knowledge mesh is designed to allow users different access levels based on the user type. In this project, the dev team has acquired the role of knowledge mesh managers generating new data type AIUs and processor AICs through KM-Interface. The subscriber users, the Data Owner and the Data Consumer/Analyst, have access to the data type patterns and may create new dataset templates; they may also access the AIC templates to transform specific AIU records into member descriptions, contextual data that extend the knowledge mesh.

The knowledge mesh must make AIU patterns and templates discoverable for subscribing users to use them. This project makes them discoverable through three KM-Web interfaces, the Data Upload modal, the Data Status Tracker process selection interface and the Dataset Search interface. Because they have been registered through KM-Web, AIU and AIC templates can be saved, accessed and triggered through the Data Upload and Data Status Tracker interfaces by direct access to their OISS IDs. The

Dataset Search interface, however, uses AI-assisted text to SPARQL queries.

The text-to-SPARQL search leverages a configurable Claude Sonnet model enhanced with few-shot prompting, a technique that significantly improves the model's accuracy in generating domain-specific SPARQL queries

This API has a primary invocation when running a dataset search, then a secondary process links dataset results with their related reports and member descriptions using the process namespace ID. A similar process could enable discovery of AIC processes, allowing for searches for processes based on data type or direct retrieval of processes based on a pre-defined schema field.

4.8 Multi-Modal Record Search and Optimization

Content search on KM-Web is supported by KM-Pipeline, which integrates the OISS knowledge graph structure, and APIs supporting AI and other NLP-based search techniques to find the optimal results for a contextual query. The modular and swappable search components in KM-Pipeline combined with the multiple interfaces available in KM-Web make an ideal testbed for benchmarking different search techniques to determine the best approach for any domain and use case. In this project, the Booz Allen team analyzed and benchmarked several search paradigms combining contextual and structural techniques.

4.8.1 Text Search

KM-Pipeline uses the standard OpenSearch BM25 algorithm to perform text-based search on the document and image summary chunks. The relevance ranking of BM25 is often useful for corpuses of relatively verbose documents with a plurality of terms. The text content generated by the AIC processing pipelines work well on AIUs that have access to contextual archive data as those are the items on which the AIC LLM-based text summarization API operates. In the case there the content of archives is limited, KM-Pipelines other extractions may prove more useful.

4.8.2 Retrieval-Augmented Generation (RAG)

Embeddings search uses the document chunk vector embeddings generated by the KM-Pipeline AIC text summarization API. Generated by Claude Haiku as numerical vectors, the embeddings encode semantic reasoning into a Euclidean space and can therefore leverage proximity measures to search documents to support queries. The two main search approaches use K-Nearest Neighbor to find predetermined number of result documents and either 1) serve the document results directly to the user or 2) contextualize a LLM in a retrieval-augmented generation (RAG) framework. The project tests both approaches.

The dev team tested the RAG with an internal rag optimization framework that used synthetic data generation to produce a set of questions from ingested documents.

These questions were then evaluated against the search system to determine the precision and mean-reciprocal rank of the search results to feed into the RAG, to determine the best search and ingestion method to retrieve the correct document to answer user query.

4.8.3 Hybrid Search

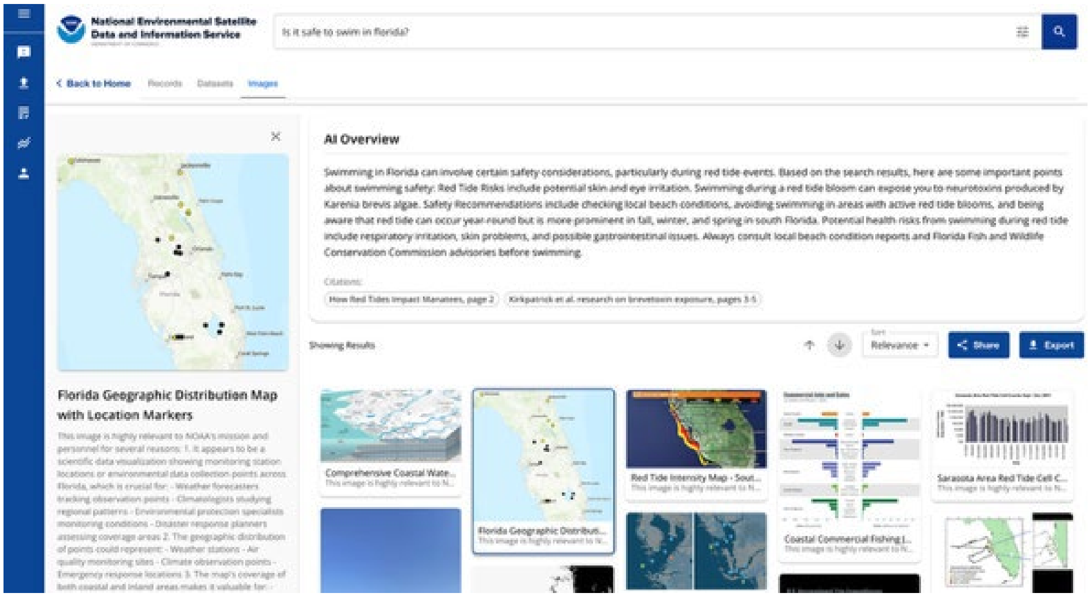
Hybrid search combined text, semantic, or RAG search by using one approach to enhance the ranking or filter the other. The team performed cross-validation and benchmarking of each of the two above methods with various parameters, including maximum k-value, different ranking weights, and combinations of the two methods with filtering or rankings updates. Hybrid search has higher performance in precision when retrieving correct documents against a user query to give the large language model appropriate context. Section 5.1.6 outlines the benchmarking results.

4.8.4 Image Search

Image search used the summaries generated by choosing the multi-modal option for the KM-Pipeline AIC text summarization API.

Image summaries were generated by the multi-modal Claude model as an image auto-captioning output and a relevancy classification generated by prompting the model to assign relevancy to the image in the context to the specific domain, in this case of or relating to Florida state policy analysts studying harmful algal blooms. The results of the image caption were stored in its own index and searched using a hybrid approach (Figure 14).

Figure 14: KM-Web Image Search Results Page with AI Overview



National Environmental Satellite Data and Information Service
Is it safe to swim in florida?
< Back to Home
Records
Datasets
Images
Showing Results
Sort
Relevance
Share
Export

AI Overview

Swimming in Florida can involve certain safety considerations, particularly during red tide events. Based on the search results, here are some important points about swimming safety: Red Tide Risks include potential skin and eye irritation. Swimming during a red tide bloom can expose you to neurotoxins produced by Karenia brevis algae. Safety Recommendations include checking local beach conditions, avoiding swimming in areas with active red tide blooms, and being aware that red tide can occur year-round but is more prominent in fall, winter, and spring in south Florida. Potential health risks from swimming during red tide include respiratory irritation, skin problems, and possible gastrointestinal issues. Always consult local beach condition reports and Florida Fish and Wildlife Conservation Commission advisories before swimming.

Citations:
How Red Tides Impact Manatees, page 2
Kirkpatrick et al. research on brevetoxin exposure, pages 3-5

Florida Geographic Distribution Map with Location Markers

This image is highly relevant to NOAA's mission and personnel for several reasons: 1. It appears to be a scientific data visualization showing monitoring station locations or environmental data collection points across Florida, which is crucial for: - Weather forecasters tracking observation points - Climatologists studying regional patterns - Environmental protection specialists monitoring conditions - Disaster response planners assessing coverage areas 2. The geographic distribution of points could represent: - Weather stations - Air quality monitoring sites - Climate observation points - Emergency response locations 3. The map's coverage of both coastal and inland areas makes it valuable for: -

Comprehensive Coastal Wate...
This image is highly relevant to N...

Florida Geographic Distributi...
This image is highly relevant to N...

Red Tide Intensity Map - Sout...
This image is highly relevant to N...

Coastal Commercial Fishing (...
This image is highly relevant to N...

Sarasota Area Red Tide Cell C...
This image is highly relevant to N...

4.8.5 Query Routing and AI Summarization

The KM-Pipeline provides a query-routing API that analyzes query format and provides an AI summary if the user enters a question. This capability uses a Claude Haiku model to analyze query structure, then analyzes intent and performs a RAG search on the text summarization indexes to generate a summarized answer. KM-Web displays the summary results with their reference records above the standard results interface.

5. Study Finding and Recommendations

Several key lessons were revealed throughout this 12-month effort that can inform future development of OISS, prioritization of next steps, and creating a vision for how to scale and grow adoption.

5.1. Assessment of Approach and Technology Under Test

The overall approach for the study proved successful and the expected outcomes were achieved, with many lessons learned along the way. The study validated the value and feasibility of automation and the importance of enriching the KM metadata to enable search and discovery. We identified some key challenges with OISS usability and adoption, recognizing several ways OISS will need to improve its scalability and performance before it can mature into a production-level capability for NESDIS and the enterprise.

5.1.1 Assessment of OISS Ease of Use

OISS provides a unique and comprehensive solution for archiving and managing data, algorithms to transform data, user access rights and provenance. However, onboarding OISS proved to be non-trivial.

For instance, deploying OISS into a new AWS environment proved challenging and slightly cumbersome because the process required elevated account permissions in lieu of an unrestricted environment. Though time consuming, the team was able to circumvent permissions issues by using CloudFormation (CF) Macros. The CF Macro applied, among several other exceptions, a mandated permissions boundary to every IAM Role before CloudFormation creates the role. Without the Macros, CF often failed and rolled back the OISS application deployment, leaving the dev team to step through long Cloud Watch stack traces to determine points of failure. OISS could adopt similar CF Macros to avoid challenges for future users by producing a flexible and readily instantiable system.

Furthermore, the OISS deployment configuration layer was distributed across several YAML files, and, in some cases, appear to be hardcoded. This translated into a challenging learning curve for the OISS API due to challenges tracing the naming conventions for its backend components. For example, the team found they ran into fewer record upload issues when they limited all microservice deployments to a specific AWS region as they found that the specific region appeared to be hardcoded throughout the OISS API. Removing hardcoded values for all configurable components would be simple update for OISS, as most of its infrastructure already uses a well-defined environment naming convention.

Another challenge related to the current OISS API is its approachability for new developers. Setting up the correct Python version and respective packages to support OISS across different operating systems can be confusing and time. consuming. The setup instructions in the README are very helpful; however, having two scripts, one for Unix-based operating systems, such as macOS, Ubuntu, and CentOS, and another for Windows-based operating systems could improve the set-up experience for prospective users. Another solution is to containerize the OISS API deployment such that dedicated image(s) and containers run the deploy commands necessary to instantiate the operating system. This remediates any challenges from working with different underlying operating systems. The required packages could simply be saved as images and then run on any system capable of supporting containerization, such as Docker or Podman.

Overall, OISS has the potential to gain wide adoption across NOAA and beyond, especially once it more clearly defines its target users and tailors its APIs, documentation, and start-up scripts. Currently OISS requires a technical, developer skillset; but in the future back-end processes could be abstracted with a well-crafted UI that supports non-technical users. More on this is covered under the recommendations and next steps section.

5.1.2 Assessment of Baseline Accelerators

The GAMECHANGER baseline code saved valuable time and resources related to UX design and the KM-Web front-end application development. Cloning and configuring the open-source codebase took less than a month to stand-up, enabling the team to divert more resources to deployment, development, and testing.

We estimate savings or value-add of an additional 2-3 months of development time for the project. The UI uses a React framework with branding that was customized for NOAA, and can be deployed in a NOAA environment, so it can be used and enhanced in the future with minimal specialized resources to maintain. If the metadata schema is changed, it would require some configuration and development updates to the front-end, but changes are relatively straightforward.

The GAMECHANGER UI has been vetted and tested across several different user groups for different use cases. Adapting KM-Web from GAMECHANGER was quick and efficient to show different functionalities. However, search experiences can be very persona-driven and additional user testing would be needed for mass adoption and to ensure the search meets expectations.

The team used aiSSMEBLE™ patterns to develop codebase for KM-pipeline, but full adoption of the aiSSEMBLE™ codebase was not necessary. In the future, if this application was moved to production, the aiSSEMBLE™ baseline simplifies unit and integration testing, and has an extensive continuous integration and development pipeline for managed packaging and scaled deployment to Kubernetes. Next steps for scaled production deployment would be to integrate the existing KM-Pipeline codebase with the aiSSEMBLE™ baseline.[4]

5.1.3 Assessment of Record Extraction in KM-Interface

The implemented solution worked well for the prototype’s limited use case; however, more data types should be assessed to determine the feasibility of scaling. Furthermore, engaging data governance experts and potentially define a user role for “governance user” could improve overall data management strategy.

This project limited the archive schema in KM-interface to minimal set of fields that the team defined with some input from subject matter experts. These included Archive ID, Publisher, Issue Date, Keywords, Checksum, Title, and Description. Different data types could then add fields needed to distinguish their types, such as Schema for structured data, or Temporal Resolution for time series data. The dev team simplified the process for defining new data types with a JSON configuration file, which made updating and accessing data types more accessible to less technical users; however, lacked the robustness available in the OISS API.

Engaging more governance professionals could inform a happy medium between the configuration process engaged in this project and the highly configurable data type/pattern definition in the OISS API. As a next step, governance experts could develop a standard template for most common data types, and possibly a “cookbook” of how to build effective governance system to accelerate adoption. OISS should work with Chief Data Officers and Data Stewards across the Lines of Business to define governance templates and config patterns to pre-define these fields.

5.1.4 Assessment of Uploads and Record Management

Through the tasks, the dev team defined two different uploads. First there were the initial template uploads, which include named data types with hardcoded values like “Maxar”. These were stored in Neptune as a semantic graph. The second upload came after a user updates an instance of a template, a record. The dev team encountered several challenges with this process. First, the records were stored in S3 object storage, which was not easily searchable. Second, to extract context from records using LLM enrichment in KM-Pipeline, OISS needed to generate a new record to trigger the LLM enrichment AIC instead of using the record that already existed in S3. Third, those new contexts generated from LLM enrichment needed to be linked to the original records template to enable provenance capture and simplify search.

To address the challenge of searching S3, the dev team implemented a search index, OpenSearch, but was unable to avoid generating extraneous records, and had to build a work around to address linking the additional metadata to the original template. Further, OpenSearch had an associated cost, and broader implementation should be carefully considered prior to scaling. To integrate the second upload step, the dev team

[4] https://github.com/boozallen/aiassemble

created a dummy record that references the original record, to trigger the AIC. This was not an ideal approach; Sections 6.2.3 and 6.2.4 outline suggestions for improving this limitation.

5.1.5 Assessment of LLM Enrichment

The use of LLMs provided immediate discoverability enhancement OISS. Whereas in the baseline version of OISS a user could only search Neptune for templates, this step enabled searching records. The LLMs extracted more natural language-like context which proved helpful, especially for data types that lacked text context, such as images. This simplified the process for a hypothetical data owner to share their data and make them discoverable for users to search.

Scalability of LLM Enrichment: In this project, records uploads triggered through AIUs and LLM enrichment triggered through AICs were decoupled to enable a dynamic view in KM-Web and allow a one-to-many record enrichment approach; this allow enables future scalability. KM-Pipeline is set up with parallel celery workers to run asynchronously in the background when users select LLM Text Summary or Ontology Linking for their reports; the progress is then shown in the Data Status Tracker. Currently, upload and enrichment are selected manually for each record, but a more scaled approach would automate this process. For example, users could instead subscribe to their data catalog, say for a satellite, through their initial upload on KM-Web and share their access URL. KM-Interface would then periodically retrieve satellite image data from the user-provided URL, generate a record AIU in OISS, and schedule LLM enrichment keeping provenance records of each update. This type of automated process at scale would be necessary for OISS to connect to entire data lakes and update real time information, not just individual static datasets.

Selecting and Evaluating Language Models: The team selected different versions of Anthopic's Claude LLM as the foundational model due to their general performance for conducting their designated NLP task. One consideration for the future is not to assume that bigger models are always better for performance. Recent studies have suggested that smaller language models, such as some BERT-based models or pruned LLMs, perform as well or better than LLMs for discrete repetitive tasks, such as in agentic AI networks.[5] The benefit is using such models in a hybrid approach to LLM enrichment is lower compute requirements and search complexity, thus scaling much faster and more efficiently. In the future, NESDIS could adopt MLOps processes with continual experiment tracking of different models to determine the best models for their enrichment needs and maintain performance over time.

5.1.6 Assessment of Search

To demonstrate that the LLM enrichment was successful at improving discoverability, the team was required to add search infrastructure and implement search algorithms to

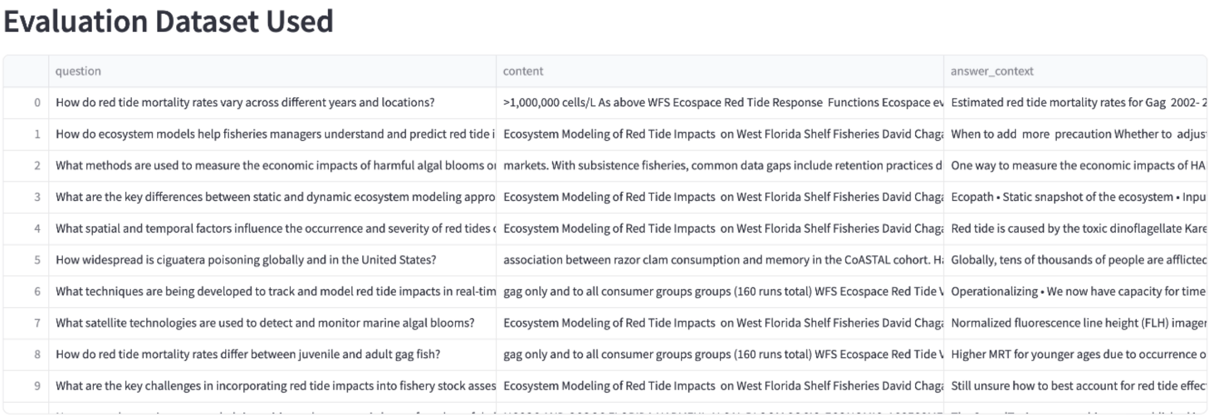
[5] https://arxiv.org/pdf/2506.021353

demonstrate improved discoverability in KM-Web. The team developed several search approaches to support notional interactions that a policy user would likely have within the UI as was outlined in section 5.7. To confirm success, each search technique needed to be evaluated to guarantee the output was accurate based on its ability to retrieve the correct related documents for a given query. The team ran several experiments to evaluate each search approach and chose which approach best demonstrated the project goals.

Search Optimization: The dev team tracked experiments with three main text search techniques for the LLM-generated context. Those search techniques included the standard BM25 implemented in OpenSearch, semantic search using LLM-generated vector embeddings, and a hybrid approach that combined BM25 and semantic search.

Synthetic Query Generation: Each experiment used search queries synthetically generated by a large-language model with tailored prompts based on NOAA mission profiles and the datasets uploaded for this study (see Appendix A). The questions were then mapped to specific areas of the documents that contained the answers. This mapping was used to evaluate the performance of each search method.

Figure 15: Synthetically generated search queries used for evaluating different search techniques



Evaluation Dataset Used

	question	content	answer_context
0	How do red tide mortality rates vary across different years and locations?	>1,000,000 cells/L As above WFS Ecospace Red Tide Response Functions Ecospace ev	Estimated red tide mortality rates for Gag 2002- 2
1	How do ecosystem models help fisheries managers understand and predict red tide i	Ecosystem Modeling of Red Tide Impacts on West Florida Shelf Fisheries David Chaga	When to add more precaution Whether to adjus
2	What methods are used to measure the economic impacts of harmful algal blooms o	markets. With subsistence fisheries, common data gaps include retention practices d	One way to measure the economic impacts of HA
3	What are the key differences between static and dynamic ecosystem modeling appro	Ecosystem Modeling of Red Tide Impacts on West Florida Shelf Fisheries David Chaga	Ecopath • Static snapshot of the ecosystem • Inpu
4	What spatial and temporal factors influence the occurrence and severity of red tides (	Ecosystem Modeling of Red Tide Impacts on West Florida Shelf Fisheries David Chaga	Red tide is caused by the toxic dinoflagellate Kare
5	How widespread is ciguatera poisoning globally and in the United States?	association between razor clam consumption and memory in the CoASTAL cohort. Ha	Globally, tens of thousands of people are afflicted
6	What techniques are being developed to track and model red tide impacts in real-tim	gag only and to all consumer groups groups (160 runs total) WFS Ecospace Red Tide V	Operationalizing • We now have capacity for time
7	What satellite technologies are used to detect and monitor marine algal blooms?	Ecosystem Modeling of Red Tide Impacts on West Florida Shelf Fisheries David Chaga	Normalized fluorescence line height (FLH) imager
8	How do red tide mortality rates differ between juvenile and adult gag fish?	gag only and to all consumer groups groups (160 runs total) WFS Ecospace Red Tide V	Higher MRT for younger ages due to occurrence o
9	What are the key challenges in incorporating red tide impacts into fishery stock asses	Ecosystem Modeling of Red Tide Impacts on West Florida Shelf Fisheries David Chaga	Still unsure how to best account for red tide effec

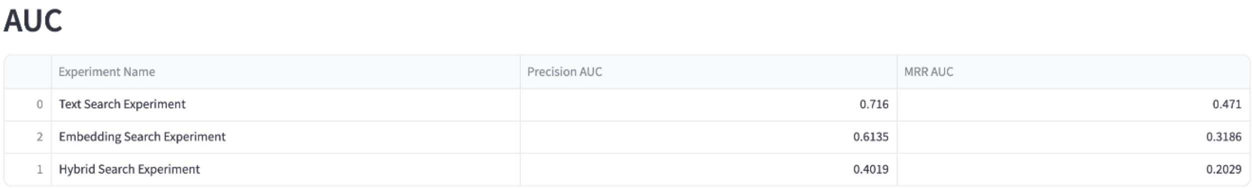
Search Metrics: Search functionality was evaluated using three key metrics for search results ranging from one to 10:

1. **Precision@N:** Determines if the mapped documents is in the results for each cutoff of *N* search results. Precision@1 means “did the right document show up first?”;Precision@10 means “did it show up anywhere in the first 10 results?”
2. **MRR:** Looks at the exact rank of the first correct document, no matter if it’s at 1 or 10.Averaging across questions shows how high correct documents tend to appear.
3. **AUC:** Instead of choosing one cutoff, evaluates ranking quality across all

possible thresholds. It measures how well relevant documents are consistently ranked above irrelevant ones throughout the list.

Experiment Tracking: The team developed an experiment-tracking tool benchmark search performance across many runs and visualize them in an integrated dashboard. This allowed the team to continually test and iterate upon different methods with different parameters to improve search capabilities and justify the approaches used. Figures 16 and 17 show the experiment results across the three different search methods. As is shown, each method demonstrated a reasonable precision above nine results, but the BM25 text and embedding searches operated most effectively on their own. This justified using BM25 to enable the text search behind the record and image content searches and using the embeddings to enable the RAG search behind the AI summarization.

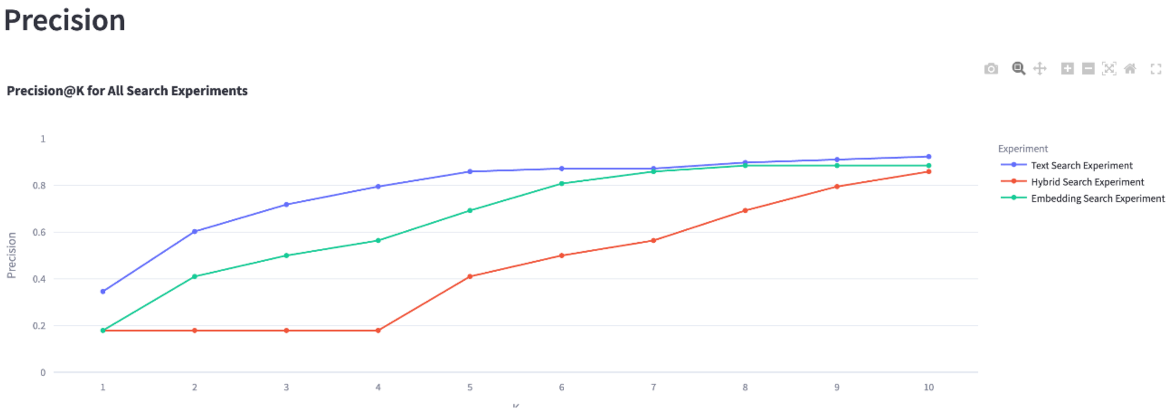
Figure 16: Precision and MRR AUC scores for each search technique



AUC

	Experiment Name	Precision AUC	MRR AUC
0	Text Search Experiment	0.716	0.471
2	Embedding Search Experiment	0.6135	0.3186
1	Hybrid Search Experiment	0.4019	0.2029

Figure 17: Precision tracking for result sets of N=1 to N=10



5.1.7 Ontology Linking Exploration

While not part of the original scope, the dev team pursued a supplemental exploratory task to augment the KM vocabulary by introducing additional ontologies. By expanding the ontological framework, the team aimed to create a larger semantic network that would increase the likelihood of successful natural language search matches. Working with feedback from NOAA, the team identified over 12000 ontology classes to automatically link.

Technical Implementation: The process employs a two-stage approach combining k-nearest neighbor (KNN) vector similarity search with LLM reasoning. Keywords and named entities from AIC records are first converted to 1536-dimensional embeddings using Amazon Titan models, then matched against pre-indexed ontology classes in OpenSearch using KNN search to retrieve the top 100 candidates. Claude (via AWS Bedrock) evaluates these candidates based on semantic relevance, domain appropriateness, and specificity, rather than accepting the highest vector similarity score. This hybrid approach addresses pure vector similarity limitations, which often produce high scores for similarly distributed but semantically different terms. Finally, RDF predicates are assigned as connections between the keywords and ontology classes based on three tiers: owl:sameAs[6] for high confidence (≥0.8), schema:about[7] for moderate confidence (0.5-0.8), and ex:potentialMatch[8] for low confidence (<0.5).

Results and Recommendations: The team successfully demonstrated ontology linking through the KNN+Claude pipeline, though confidence scores varied across different domains. While the hybrid approach showed improvement over pure vector similarity, achieving consistently high confidence across diverse ontologies proved challenging. Two key recommendations emerged from this exploration: First, ontology linking should be incorporated during AIU template generation rather than after records exist, associating namespaces with datasets at creation time. This would enable users to directly populate the mapped ontology fields and classes during data upload, ensuring more accurate and complete semantic annotations from the start. Second, confidence scores should be leveraged for search to be truly useful. The current implementation stores mappings as JSON-LD in S3 as OISS does with records, which lacks efficient search capabilities. Future iterations should index these scores and mappings directly in OpenSearch to enable filtering and ranking based on link quality.

5.1.8 Assessment of Uncertainty, Metrics, and the Data Status Tracker

The status tracker feature is an additional feature added during development that proved vital for sharing global statistics and keeping data owners apprised of LLM enrichment and ontology linking status.

Sharing global metrics, including measures of uncertainty, such as those captured curing ontology linking requires effective provenance capture. The dev team’s primary tool for provenance capture is the OISS dissemination information package (DIP), an additional archive reporting task that follows every AIU and AIC task. DIPs capture any data preprogrammed by the AIU or AIC to propagate to the DIP, including upload times, metadata, and any validations the program performs, including confirming successful execution. The team then samples multiple DIPs to estimate global statistics, such as aggregate metadata summaries.

[6] http://www.w3.org/2002/07/owl#
[7] https://schema.org
[8] http://example.com

The Data Status Tracker is a UI feature that can evolve significantly over time to more effectively leverage uncertainty metrics for specific use cases. When faced with the challenge of contextualizing how to use uncertainty metrics to augment the Scenario use case, the team wanted an approach aside from refining the search results and the ontology linking assignments. Both could be leveraged in the UI as processed DIP data. This would have required implementing search as an AIC in OISS, which would have slowed performance down significantly. The team found that adding a search endpoint to the OISS API could potentially absolve the issue. This would be an area of expansion for a future use case, in addition to tracking overall data usage through more extensive use of DIP functionality.

5.2. Study Recommendations

Based on the study findings and assessment of the capabilities developed in this study and the dev team's understanding of the broader objectives of the NESDIS KM effort, the next section outlines actionable next steps to mature OISS and the KM prototype implementation toward a production-grade capability that can be used for knowledge management beyond just archiving and closer to enabling an EO DT. These recommendations are loosely sequential, focusing first on the most critical technical improvements, then enhancing user experience and functionality, in alignment with product strategy.

5.2.1. Define Scaled Search Approach

NESDIS could explore approaches to scale search. By using S3 as a data lake, NESDIS could scale the full knowledge graph and enable the search of AIU reports and any related data generated by AICs[9].

To wield more control over S3 objects for search, OISS could enable updating the S3 object metadata to be queried using the fully managed Apache Iceberg tables that are natively created to back S3. In this approach, OISS data owners could add contextual fields in the form of key-value pairs to the S3 metadata during object creation. These fields could be comprised of a subset of the record metadata uploaded during AIU tasks, the LLM context data, or the linked ontology classes generated during AIC tasks. Therefore, while the Iceberg tables have a tabular format, they could still be enriched with textual metadata that would make the objects more easily linked in an extended KM ecosystem. This would simplify the complexity of linking AI-assisted searches back to their respective KM record(s). Implementing this change would require an update to OISS that allows users additional record configuration for S3 metadata updates when creating AIU records or AIC member descriptions updates. Such an update could increase the degrees of freedom when searching the knowledge mesh; providing more avenues to reduce complexity and increase scale.

[9] https://aws.amazon.com/blogs/database/create-a-virtual-knowledge-graph-with-amazon-neptune-and-an -amazon-s3-data-lake/

A simpler approach could use AWS Glue to crawl the S3 records uploaded by OISS and automatically generate Athena tables. However, the output of a Glue crawler would be difficult to configure for multi-modal data and likely oriented towards transactional querying and does not necessarily support contextual natural language queries. Thus, we do not recommend this approach.

5.2.2. Update OISS to Simplify S3 Path Naming Convention

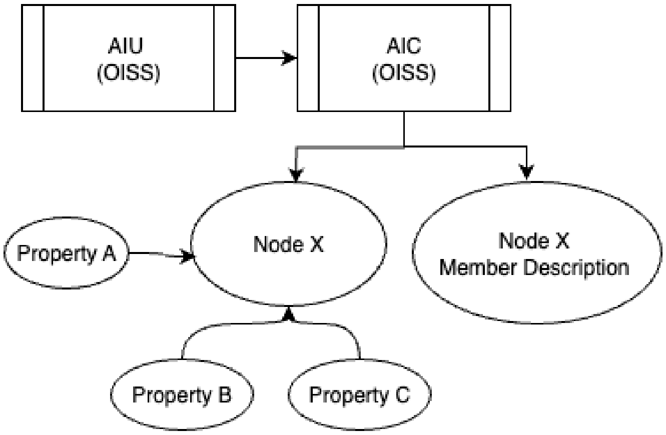
Another means of scaling a knowledge graph through S3 would be to leverage the path naming convention. A simplified, and well-documented naming strategy would enable lower complexity linking through concept normalization. In many cases, this could lead to constant time (O(1)) linking between AIU templates, records and AIC-generated member descriptions. While OISS has a namespace mechanism that likely enables this, it is somewhat difficult to manage in its current form. Making the record uploads entirely configurable by from root IRI to endpoint could extend a level of transparency that enables a more intuitive referencing among templates, records and member descriptions through the entire knowledge graph.

5.2.3. Simplify the OISS AIC Interface

Developing a method for previously generated AIU records to directly trigger new AIC tasks can improve efficiency and scale by reducing the number of new records generated, allowing for asynchronous AIC calls to enable horizontal scaling for complex computing processes like LLMs, and simplifying the knowledge graph by enabling a one-to-many linkage between AIU records and AIC-generated context.

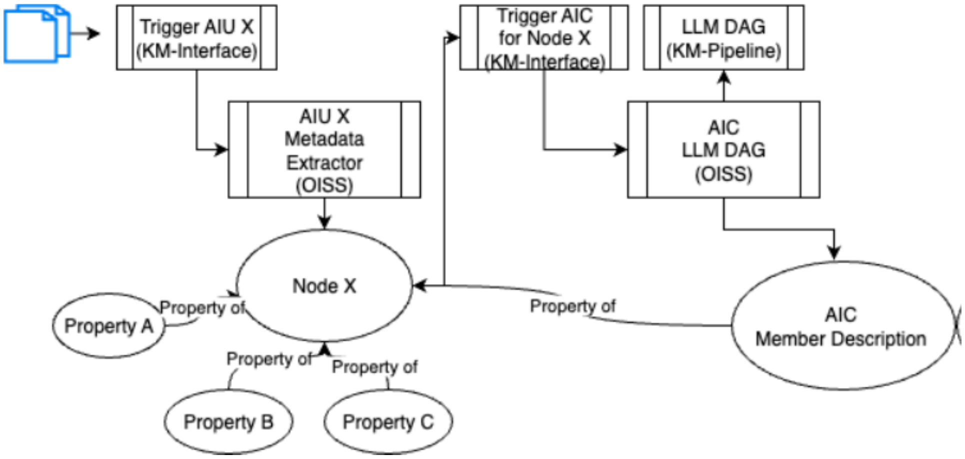
As discussed in the Introduction, AIUs store report metadata as OISS Core and Framework ontology graphs. AICs augment those graphs with member descriptions which essentially add edges to the original AIU record graph. This linkage is managed through an AIC property that references the AIU. Figure 18 illustrates this process.

Figure 18: OISS native AIC trigger process



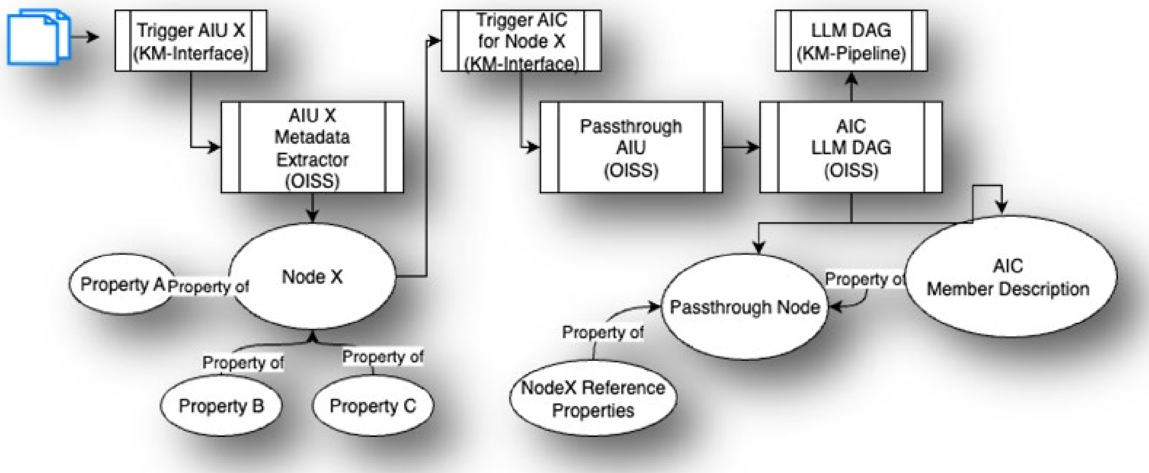
Furthermore, the KM limits the way it allows one to directly link an AIC to an AIU using its native namespace functions because it limits the way one can trigger an AIC. In its current implementation, OISS requires an AIU process to trigger an AIC process, thus requiring a new AIU record to be generated whenever an AIC task runs. To enable the dynamic process selection functionality displayed in KM-Web's Data Status Tracker, the dev team had to decouple AIU report generation from AIC. This was done by calling a passthrough AIU and generating a corresponding AIU record. This results in the member description natively referencing the passthrough namespace rather than the namespace of the record that the AIC is augmenting (Figure 19).

Figure 19: Decoupled AIU to AIC Pattern with Passthrough AIU



An update to OISS that implements native synchronization between AIC processes and previously performed AIU records would help the system maintain a bidirectional linkage graph with one-to-many connections from AIUs to AICs. Currently, linkage between AIU and AIC processes relies on metadata fields generated manually by the application, not the framework. This should be elevated to framework-level enforcement to facilitate mandatory AIU reference. When an AIC workflow initiates, the framework would require explicit reference to previously generated AIU record and automatically generating the proper provenance in all downstream references, such as in the DIPs. This would also significantly reduce the number of additional AIU records generated, this optimizing storage and search complexity through S3 path references (Figure 20).

Figure 20: Recommended Namespace – Linking Strategy for Decoupled AIU to AIC Pattern



How This Supports a Digital Twin Framework

Digital twin technology must have the ability to maintain a concept of a digital record, corresponding to past or present states of a real-time asset, while enabling generation of digital replicas, transformations of the digital record to forecast notional changes or updates[10]. This requires minimizing the complexity of linking data to transformation algorithms and their distributed computing resources, and automatic provenance capture to enable forecasting and impact assessment for digital replicas. The proposed decoupling approach to AIU and AIC generation provides more degrees of freedom in tracking assets and their available algorithms, while maintaining a linkage through their S3 path namespace. This could also scale search with minimal reliance on indexed metadata search as described in 2.2.

5.2.4. Standardize and Scale Search Capabilities

Implement a unified data lake with a semantic search layer across OISS using search indexes, such as OpenSearch in addition to Neptune and S3. With a simple update to AIU record UBLs to update indexes, this approach allows users to immediately perform contextual searches over their record data. Using the AWS service can result in non-trivial costs for scaling, but there are alterative deployment options that can leverage containerized open-source indexes.

[10] https://esto.nasa.gov/files/AIST/ESDT_ArchitectureFramework.pdf

5.2.5. Engage Users with a Standard User Interface

Implement an Optimized Search API

In the project, search was implemented as an API in KM-Pipeline and, rather than using the OISS AIC UBL framework, the API was directly accessed by KM-Web backend. This implementation avoided some of the latency experienced when triggering AICs. While OISS, is likely developing a faster UBL interface, a search API would likely benefit from avoiding the UBL interface entirely and directly accessing the graph database, S3 and any additional search indexes

Improve Search Results by Integrating User Feedback

Incorporate mechanisms into the User Interface for query refinement and crowd-sourced feedback. When users are unsatisfied with results, they should be able to adjust filters, add context, or reframe questions using UI prompts. These adjustments can drive more accurate vector-based search responses and improve model performance over time.

Refine Data Status Visualizations Build Trust in Results

Improve the user interface to support more robust filtering, faceted browsing, and a more extensive display of performance metrics. Provide users with real-time feedback on data lineage, coverage, and provenance so they can confidently evaluate datasets using them for analyses. A data status dashboard would also help engage data owners looking to track usage over time.

Leverage Generative AI to Expedite Data Onboarding

While the current AIU templates work well on individual datasets, to scale dataset onboarding, it would be advantageous to develop an interactive UI component that enables data owners to define, configure, test, and deploy new data types. The UI could leverage LLM-based code generation models and tools to assist with this process. This feature simplifies onboarding of new datasets, supports better data stewardship, and reduces reliance on backend engineering. Enabling self-service through the template builder promotes agility and democratizes access to tools within the KM.

5.2.6 Define Governance Standards Training for Strategy-Focused Users

Developed for archiving disparate data, OISS has specific data onboarding requirements that governance officers, data strategists, or other policy-oriented professionals could adopt to support their team's OISS adoption, possibly for more expansive use cases, such as managing data for an Earth System Digital Twin. As an open archival information system (OAIS), OISS governance is implemented through general concepts for managing data archives, such as the AIU, AIC and DIP (and system information package, or SIP) that are captured in its documentation; however, the documentation is heavily geared towards technical users.

Prior to technical implementation, an organization must adopt a governance strategy that meets the needs of their specific use case. This could involve managing data ownership, access rights, distribution, provenance, and even incorporate AI governance standards[11] to manage how data is assessed or transformed with AICs as was demonstrated in this project. This is where training can be reoriented towards to more strategy focused users rather than technical. This kind of training would emphasize how a team should define their use case and then map it to OISS using its OAIS concepts. A pilot training program for alternative use cases for knowledge management in OISS would present an opportunity to engage non-technical users and expand adoption beyond its current application.

5.3. Develop a Product Strategy and Adoption Roadmap

For the KM to be adopted broadly, NOAA will need to more clearly define its target users, target the value provided to those users, and develop a plan to scale user onboarding. Many of the above recommendations can be prioritized based on who NOAA is targeting and which problems it can most effectively address. For example, it may be more feasible to focus on internal technical users and data owners that only need the API, rather than optimizing a search application for many different NOAA and non-NOAA (e.g. policy analyst) users. Alternatively, a long-term roadmap should consider how OISS integrates into existing search engines and user-facing applications that already enjoy wide adoption and awareness. These decisions will drive the prioritization of UI enhancements and which journey to optimize for, such as subscribing data, managing governance, or searching the KM.

After defining its target users, perhaps just as important will be engaging NOAA OCIO and other business lines to identify KM champions and develop a strategy to onboard additional data owners and datasets. It is unclear how difficult or easy it may be to connect OISS to disparate domain-specific data repositories in NOAA and align on data governance standards and access rules while embedding those policies seamlessly into the KM. This is where pilots can pave the way and build buy-in. There may also need to be an internal and external strategy to integrate valuable data from other Federal science and statistical organizations (e.g., Census, BLS, NSF, NASA) to increase the overall multi-domain data and value.

This section further outlines a notional product development roadmap for OISS over the next 12 months to begin sequencing and implementing these recommendations.

5.3.1 Simplify OISS Deployment Configuration

Aligned with developing a roadmap for OISS, NOAA should consider its long-term plan for deploying OISS outside of NESDIS and NCCF. If the intention is for OISS to be deployable in other environments, both for further development and collaboration with

[11] https://www.boozallen.com/insights/ai-research/responsible-ai-for-mission-innovation/ai-governance-platform.html

external contractors, and/or as an extension of its reuse in other organizations, this project demonstrates that simplifying the deployment prior to ultimately launching an OISS open-source API that others can deploy independently would be beneficial for adoption. More details on experiences with deployment and solutions the team employed with collaboration from NESDIS are outlined in 5.1.1.

5.3.2 Leverage Pilots to Quickly Test and Learn

Pilots are valuable in product development for quickly learning how to improve a system or product and refining where to add value and focus technical development. For OISS, pilots can also serve the purpose of quickly adding volume (new data) to the KM. With an overall strategy in place (e.g., identifying prioritized target users, value proposition, differentiators, and adoption strategy), target user groups can be engaged for multi-phased pilots.

Pilots can focus on advancing critical success factors, such as demonstrating data onboarding at scale, or refining user experiences. Some potential examples include:

- Pilot Governance system: Work with a target data owner or group to refine how governance will be defined and passed through OISS. Define minimum viable governance requirements or fields to be included in the KM templates.
- Pilot Performance: Repeat the study's approach and pipeline at a larger scale, i.e., running a pipeline against an entire data lake, testing asynchronous and autonomous KM uploads. Use the pilot to benchmark performance and costs with larger volumes of data.
- Pilot Enhanced UI Features: Consider enhancements to KM-Web using proven GAMECHANGER UI functionality and test usability and how users find value in the KM results and what features promote trustworthiness. Alternatively, this pilot could involve integrating the OISS KM API into an existing application and finding out how useful it is to surface its results and metadata.
- Pilot Training and User Guides: Develop persona-based (e.g., governance user, data owner, search consumer) training, content and/or UI features (glossary, FAQs, tool tips in the GAMECHANGER UI) targeted towards non-technical consumers to reduce friction. Create feedback tools and mechanisms to measure usage through more enhanced analysis of provenance capture through DIPs.

In all, these suggested pilots paired with a phased roadmap make incremental progress toward a production-grade knowledge mesh that can serve the NOAA enterprise and provide valuable data discovery.

6. Addendum

6.1 Appendix A: Datasets

Table 5 identifies the data sources used during the study

Table 5: Data Sources

Data Source	Data Sets and Reference Links
Algal Bloom	Florida Fish and Wildlife Conservation Commission (FWC) Florida Karenia brevis categorical abundance *https://geodata.myfwc.com/datasets/myfwc::florida-karenia-brevis-categorical-abundance-most-recent-8-days/about* FWC Historic Harmful Algal Blooms Events (1960-Present) *HAB Monitoring Database | FWC* HABSOS (Harmful Algal BloomS Observing System) Mapping Systems *https://www.ncei.noaa.gov/maps/habsos/maps.htm*
Fishery Production Data	West FL Shelf - (HAB) Harmful Algal Bloom Ecopath model *https://www.fisheries.noaa.gov/inport/item/56904* HistoricalFLHAB-Basic estimates *https://www.iana.org/assignments/media-types/text/csv* Ecosystem Modeling of Red Tide Impacts on West Florida Shelf Fisheries *https://ncbs.ifas.ufl.edu/modeling-red-tide-impacts-on-the-west-florida-shelf/* NOAA Fisheries Economics of the United States Report 2022 *https://s3.amazonaws.com/media.fisheries.noaa.gov/2024-11/FEUS-2022-SPO248B.pdf*

Data Source	Data Sets and Reference Links
	NOAA Fisheries - TOTALS BY YEAR/SPECIES DEFLATED VALUE Datasets *https://www.fisheries.noaa.gov/foss/f?p=215:200:5431540447813* NOAA Fisheries - TOTALS BY YEAR/STATE/SPECIES Datasets *https://www.fisheries.noaa.gov/foss/f?p=215:200:5431540447813*
Health Effects Data	Aerosolized Red-Tide Toxins (Brevetoxins) and Asthma *https://www.sciencedirect.com/science/article/abs/pii/S0012369215498988* CDC - One Health Harmful Algal Bloom System (OHHABS) across the nation *https://www.cdc.gov/ohhabs/data/summary-report-united-states-2022.html* Gastrointestinal emergency room admissions and Florida red tide blooms *https://www.floridahealth.gov/environmental-health/aquatic-toxins/_documents/kirkpatrick-economic-impacts.pdf* Neurological illnesses associated with Florida red tide (Karenia brevis) blooms *https://pmc.ncbi.nlm.nih.gov/articles/PMC9933543/pdf/nihms-1860788.pdf* Review of Florida Red Tide and Human Health Effects *https://pmc.ncbi.nlm.nih.gov/articles/PMC3014608/pdf/nihms232128.pdf*
Recreation Business Operations Data	HABs: National and Regional Impacts by the Numbers *https://cdn.coastalscience.noaa.gov/page-*

Data Source	Data Sets and Reference Links
	attachments/habs/HABs-National-Impacts-1-pager.pdf NOAA Fisheries Hitting Us Where it Hurts: The Untold Story of Harmful Algal Blooms *https://www.fisheries.noaa.gov/west-coast/science-data/hitting-us-where-it-hurts-untold-story-harmful-algal-blooms*
Local Seafood/Lifestyle/Business Data	Harmful Algal Bloom Event Database (HAEDAT) *https://haedat.iode.org/browseEvents.php* Shellfish Harvesting Area Information *https://shellfish.fdacs.gov/seas/seas_westgulf.htm*
Public Stock Data	Stock Assessments Quarterly Reports Archive - Fish Assessment Report *https://media.fisheries.noaa.gov/2021-04/Annual%20Summary%20PDF%20508-C3.pdf?null* NOAA Fisheries - Stock SMART -Status, Management, Assessments& Resource Trends: Browse by Stock *https://apps-st.fisheries.noaa.gov/stocksmart?app=browse-by-stock* NOAA Fisheries - Stock SMART -Status, Management, Assessments& Resource Trends: Chart Time Series *https://apps-st.fisheries.noaa.gov/stocksmart?app=chart-time-series* NOAA Fisheries - Stock SMART -Status, Management, Assessments& Resource Trends *https://apps-st.fisheries.noaa.gov/stocksmart?app=count-*

Data Source	Data Sets and Reference Links
	assessments NOAA Fisheries - Stock SMART -Status, Management, Assessments& Resource Trends: Download Data Tab *https://apps-st.fisheries.noaa.gov/stocksmart?app=download-data* NOAA Fisheries - Stock SMART -Status, Management, Assessments& Resource Trends: Plot Stock Condition *https://apps-st.fisheries.noaa.gov/stocksmart?app=plot-stock-condition*
Economic Impact Data	Economics: National Ocean Watch - Quick Report Tool for Socioeconomic Data *https://coast.noaa.gov/quickreport/#/ENOW/ENOW //2013,2012,2011,2010,2009,2008,2007,2006,2005* Economic Impacts of 2018 Florida Red Tide: Airbnb Losses and Beyond *https://coastalscience.noaa.gov/news/economic-impacts-of-2018-florida-red-tide-airbnb-losses-and-beyond/* Florida Has Spent Nearly $20M Dealing With Algae Blooms In Last Decade *https://news.wjct.org/first-coast/2020-08-26/florida-has-spent-nearly-20m-dealing-with-algae-blooms-in-last-decade* The Economic Contribution Of Spending In The Florida Keys National Marine Sanctuary To The Florida Economy *https://marinesanctuary.org/wp-content/uploads/2019/07/FKNMS-Report-Final-072819.pdf*

Data Source	Data Sets and Reference Links
	Estimating the Benefits of Ocean Color Data in Mitigating HAB Events *https://repository.library.noaa.gov/view/noaa/56882* US Census Data - AB00MYCSA01AAnnual Business Survey *https://data.census.gov/table/ABSCS2022.AB00MYCSA01A* US Census Data - AB2100CSA05 Annual Business Survey: Urban and Rural Classification of Firm Statistics for Employer Firms by Industry, Sex, Ethnicity, Race, and Veteran Status for the U.S., States, and Metro Areas: 2021 *https://data.census.gov/table/ABSCS2021.AB2100CSA05?y=2021&n=11* US Census Data - S0201SELECTED POPULATION PROFILE IN THE UNITED STATES *https://data.census.gov/table/ACSSPP1Y2017.S0201?t=Business%20and%20Economy:Disability:Health:Health%20Insurance&g=040XX00US01,12,22,28,48* US Census Data - S2503 Financial Characteristics *https://data.census.gov/table/ACSST1Y2023.S2503* Workshop on the Socio-Economics Effects of Marine and Fresh Water Harmful Algal Blooms in the United States *https://hab.whoi.edu/wp-content/uploads/2021/04/HAB-Socioeconomics-Workshop-Proceedings_14.pdf*
State/Local Budget Data	FWC: Harmful Algal Bloom (HAB) Grant Program *https://myfwc.com/research/redtide/taskforce/grant/*
Federal Contracts Data	Grants.gov

Data Source	Data Sets and Reference Links
	https://grants.gov/search-grants
Funding Opportunities Data	NCCOS Funding Opportunities - Current and Historical Opportunities *https://grants.gov/search-results-detail/350259* "Protecting Florida Together" Funding Opportunities *https://protectingfloridatogether.gov/sites/default/files/documents/508%20Compliant_FY2022-23_Innovative_Tech_Grants_1.pdf*

6.2 Appendix B: Technical Documentation

The dev team leveraged the following tooling displayed in Table 6 below.

Table 6: Software Inventory

Tools and services	DevOps Software Inventory
AWS Services	• Elastic Compute Cloud (EC2) • S3 • Elastic Block Storage (EBS) • Amazon Machine Images (AMIs) • IAM Roles, Policies, and Service Accounts • CloudFormation • Virtual Private Cloud (VPC) • Transit Gateway • IPSec VPN • CloudTrail • Key Management Service (KMS) • Secrets Manager • Lambda • Step Functions (from OISS) • Elastic Container Service

Tools and services	DevOps Software Inventory
	(ECS) - Provisioned Mode • Elastic Container Registry (ECR) • Certificate Manager • Neptune (managed graph DB from OISS) • Aurora for PostgreSQL (web application backend) • CloudWatch • EventBridge Rules • Systems Manager (maintenance and patching automation) • Config • Route 53 • Cloud Map • EC2 Image Builder (for golden image creation) • ElastiCache for Redis (for knowledge mesh implementation) • Load Balancers, Target Groups, Listeners, Pathing Rules, and Healthchecks • API Gateway (from OISS)
Additional Services and tools	• Github Enterprise (GHE) • GHE Actions (for CI/CD) • Docker and Docker Compose • Ubuntu 22.04 Jammy Jellyfish • MacOS Sequoia • Windows 11 • Hadolint (linting for Dockerfiles) • Trufflehog3 (secrets scanning) • CodeQL (static code analysis) • Node.js • yarn • ESLint (code linting) • Docker-in-Docker (DinD) builds and Docker Buildx

Tools and services	DevOps Software Inventory
	• Python3 and pip3 • Poetry • SAM CLI (from OISS) • Boto3 • Autoflake (Python imports management) • FastAPI • Celery (task queue system with polling • and worker allocation) • OpenSearch • Vite • Windows Subsystem for Linux (WSL) • Ollama

About Booz Allen

Booz Allen is the advanced technology company delivering outcomes with speed for America's most critical defense, civil, and national security priorities. We build technology solutions using AI, cyber, and other cutting-edge technologies to advance and protect the nation and its citizens. By focusing on outcomes, we enable our people, clients, and their missions to succeed—accelerating the nation to realize our purpose: Empower People to Change the World®.