Inventory Manager and Granule Metadata Services

Evan McQuinn*, David Fischman**, Chris Esterlein*, Semere Ghebrechristos*, Arianna Jakositz*

*Cooperative Institute for Research in Environmental Sciences
**National Centers for Environmental Information

NOAA Enterprise Data Management Workshop, 2018
Where does it fit?

Collection Manager
- CEdit
- CoMET
- Metaserver
- Collection M/D
- DocuComp

Inventory Manager
- M/D Event Log

OneStop API/Search Engine
- Index
- C
- G

Order Fulfillment + Access Services
- Storage Services
  - Tape
  - Disk
  - Cloud

External Systems
- OneStop UI
- Thin Portals

Data Producers and Operators

Data Managers
- Common Ingest
- Data Streams
- S2N + ATRAC

Public

Ingest Services

MD Services

Search Services

Storage/Access Services
Disclaimer

Everything you’re about to see is still under development.
What is the Inventory Manager?

A single, canonical database of NCEI’s granule-level metadata

• Discovery information
  • Space, time, keywords, collection membership, domain-specific attributes...
  • For user discovery
• Access information
  • FTP, HTTP, THREDDS, S3…
  • Where can users get the data?
• Inventory information
  • Checksums, archive locations, public/private flags, retention schedules…
  • For internal use.
What is the Inventory Manager?

Challenge #1:

Flexibly storing metadata for billions of files as they are today and as they continue to stream in the door.

Requires scalable, distributed storage technologies.
What is the Inventory Manager?

A flexible workflow system to process metadata

- Normalize raw metadata
  - e.g. transform NetCDF headers into well-formed ISO(Lite) metadata
- Validate metadata
  - Verify links, enforce requirements, collect rubric scores, etc.
- Metadata enrichment
  - e.g. custom SME script to extract domain-specific information
- Populate downstream systems
  - e.g. automatically push collection and granule information into OneStop
  - but also… map services? internal analytics and/or reporting?
What is the Inventory Manager?

Challenge #2:

Managing, deploying, configuring, and scaling a wide variety of SME-provided functions to operate on each granule.

Requires scalable, distributed processing technologies.
Approach: Event Sourcing

$$fn(\ input\ ) \rightarrow output$$
Approach: Event Sourcing

- Facts: data + time.
- “A granule with attribute X was ingested at time Y”
- *Only part that must be preserved*

\[
\text{fn( input )} \rightarrow \text{output}
\]
Approach: Event Sourcing

- Facts: data + time.
- “A granule with attribute X was ingested at time Y”
- Only part that must be preserved

\[ \text{fn( input )} \rightarrow \text{output} \]

- Applied to series of input facts
- Can be re-executed as needed
- Therefore, can evolve over time
Approach: Event Sourcing

- **Facts**: data + time.
- "A granule with attribute X was ingested at time Y"
- Only part that must be preserved

\[ \text{fn} \left( \text{input} \right) \rightarrow \text{output} \]

- Applied to series of input facts
- Can be re-executed as needed
- Therefore, can evolve over time

- Deterministic result of input + function
- Never modified directly
- Recalculated by re-executing function
- Can be sent to multiple downstream services
Approach: An Example

\texttt{goes\_extract( goes\_granules ) } \rightarrow \texttt{goes\_discovery}
Approach: An Example

Same input

\[
goes\_extract( \textit{goes\_granules} ) \rightarrow goes\_discovery
\]
\[
\text{iso\_rubric}( \textit{goes\_granules} ) \rightarrow goes\_rubric
\]
\[
\text{link\_checker}( \textit{goes\_granules} ) \rightarrow goes\_link\_info
\]
\[
\text{ingest\_stats}( \textit{goes\_granules} ) \rightarrow goes\_ingest\_stats
\]

Varied functions

Custom, single-purpose output
What it looks like

- Common Ingest
- External M/D Systems
- Collection Manager

API LAYER

- Persistent, distributed, event stream storage
- Workflows

UNIDIRECTIONAL DATA FLOW

- Key / Value
- Lucene Index
- RDBMS

- OneStop
- Map Service
- Clients

Reference Architecture: Kappa
But How?

- Distributed, ordered storage of facts
- Horizontally scalable
- Replicated for durability
- Facilitates distributed processing

+ Orchestration system for containers
- Container = complete, packaged program
- Dynamically, horizontally scalable
- Cloud optimized
- Deploys, manages distributed processing
In Summary

Inventory Manager is:

A log of every granule metadata event in NCEI
+ A set of algorithms applied to each event
+ A set of outputs to support querying
Questions?