Australia’s Integrated Marine Observing System (IMOS)

Automated data ingestion for the Australian Ocean Data Network

Marton Hidas, Sebastien Mancini, Roger Proctor, Peter Blain, Leigh Gordon, Angus Scheibner, Laurent Besnard

University of Tasmania

6 November 2018
Outline of the talk
12 minutes, plus 3 minutes for questions

1. About IMOS
2. Data providers versus data users
3. Automated pipeline
4. Future improvements
IMOS Facilities

All data discoverable, accessible, usable and reusable

https://portal.aodn.org.au
Why AODN?

Data providers

Diversity in
- Technologies
- Instruments
- Platforms
- Organisations
- People

Data users/use cases

Diversity in
- Data products required
- Data volume required
- Method of access
- Preferred formats
- Tools used

- Research scientists
- Government departments
- Private industry
- Fisheries
- Managers
- Policy makers
- Shipping
- Recreational fishing/boating
- Etc...
Why AODN?

Each provider has to cater for many users
Why AODN?

Data providers

- University of Tasmania
- IMAS
- Australian Government Geoscience Australia
- CSIRO
- Australian Government Bureau of Meteorology
- Australian Institute of Marine Science
- Australian Government Department of the Environment and Energy
- NSW Office of Environment & Heritage
- SARDI
- Australian Agricultural College

Data users/use cases

Each user may need to obtain data from multiple sources
Why AODN?

Each provider has to cater for many users

That’s a lot of unique connections! Diversity²

Each user may need to obtain data from multiple sources

Data providers

Data users/use cases
Why AODN?

Data providers

Data users/use cases

Standard ingestion

Standard access
IMOS standard ingestion: Key design criteria

• Keep data safe
• Make only high quality data for users
• Do not increase load on front facing systems
• Make data available as quickly as possible
• Robustness
• Transparency
Architecture overview of the pipeline

Data providers

Data providers

FTP
Rsync

Incoming directory

Watch service

Broker (RabbitMQ)

Queue 1 / Task 1
Queue 2 / Task 1
Queue 3 / Task 1
Queue 1 / Task 2

Consumer (Celery worker)

Worker 1 / Task 1
Worker 2 / Task 1
Worker 3 / Task 1

AODN Data Package:
- Specific handler
- Specific tests

AODN Core Package:
- Generic handler
- File State manager
- Logger
- Posthook
- Tests
Pipeline workflow: the state machine

Data providers

Data providers (and AODN staff) are notified via a detailed email report.

Incoming data files

Resolve

Pre-process

Validate

Process

Publish

Post-process

Notify

Handle error

Notify error

Error files

Compliance checker

ETL (Talend)

Storage (AWS S3)

Metadata (GeoNetwork)

Data & Maps (GeoServer)

DB (Postgres)

Data providers

AODN Portal

AODN Portal
Pipeline workflow: the state machine

Data providers

Incoming data files

Resolve

Error files

Handle error

Post-process

Notify error

Notify

Some Numbers:

32 Organisations providing data
60 Data uploaders (+ a handful that we pull from)
24 Handler classes
43 Ingestion pipelines
86 Data/metadata “Harvesters” (95 Harvest jobs)
204 Data collections in the AODN Portal

Data providers (and AODN staff) are notified via a detailed email report.

DB (Postgres)

Data & Maps (GeoServer)

Some Numbers:
Benefits: Python ecosystem

- Pandas
- NumPy
- netCDF-4 python
- AODN Python packages
- Celery
- IOOS Compliance checker
Benefits: Python ecosystem – One outlier

- Pandas
- NumPy
- netCDF-4 Python
- AODN Python packages
- Talend ETL Tool (Java)
- Celery
- Compliance checker

IOOS
Benefits: Test and deploy code in confidence
Benefits: Test and deploy code in confidence
Benefits: Test and deploy code in confidence

627 Tests passed !!!
Benefits:

• Generic code
• Increased confidence:
  • All features can be tested from one version to the next
• Easier to debug
• Better control of the different version
  • Workflow to build, package and deploy
  • What version is deployed on which environment (Release candidate or Production)
• Faster publication of new data
• Improved consistency within data collections
Future improvements

• Reporting, creation of dashboard in Sumologic
• Improve code:
  • Replace Talend (Java) by a similar tool in Python for a consistent environment
• Use of AWS Batch to improve scalability:
  • No conflict between pipelines
  • Multiple queues for different data streams
  • Only running when needed

GitHub
https://github.com/aodn/python-aodncore
https://github.com/aodn/python-aodndata