



EOSDIS

NASA'S EARTH OBSERVING SYSTEM
DATA AND INFORMATION SYSTEM

Cumulus Lessons Learned: Building, testing, and sharing a cloud archive

2019 ESIP Summer Meeting

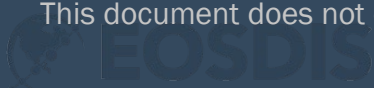
Patrick Quinn

Cumulus Architect

patrick@element84.com

This work was supported by NASA/GSFC under Raytheon Co. contract number NNG15HZ39C.

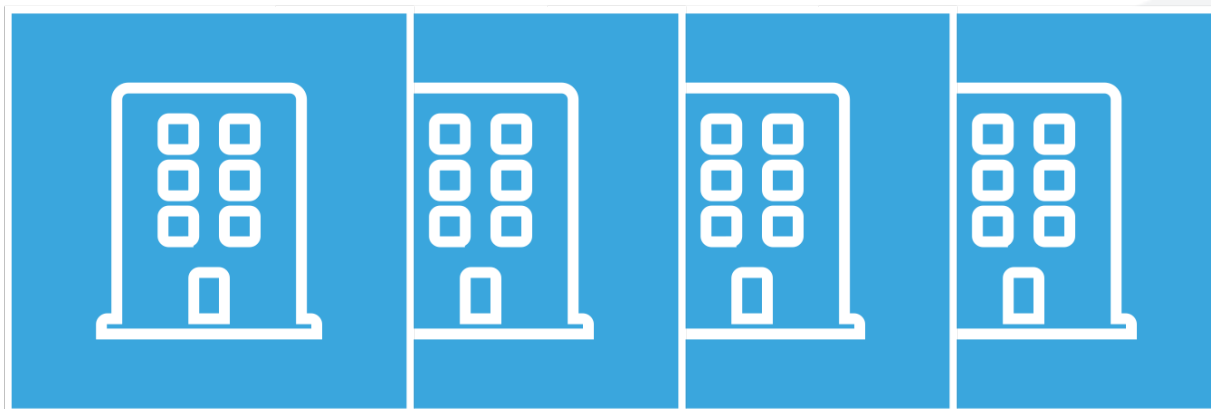
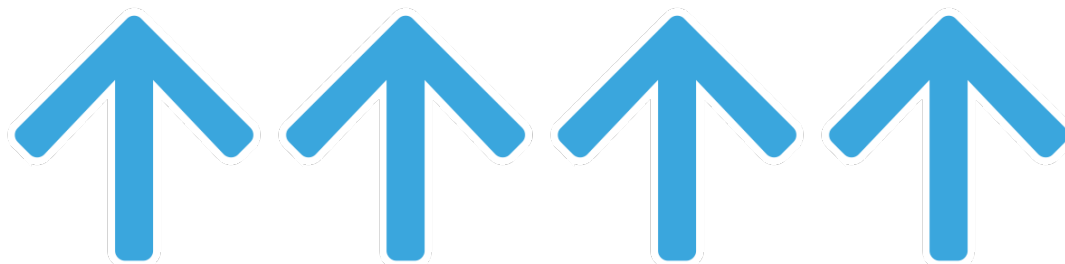
This document does not contain technology or Technical Data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

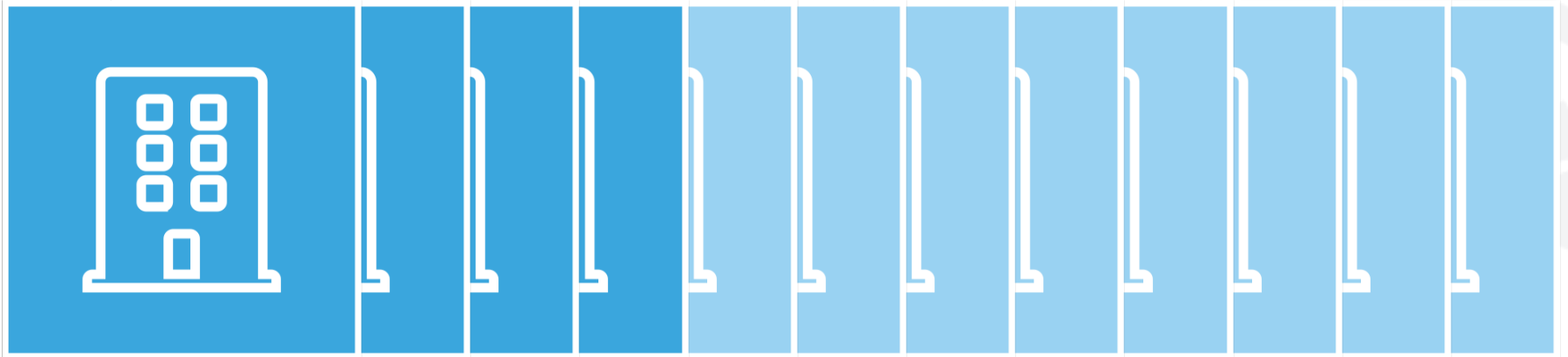
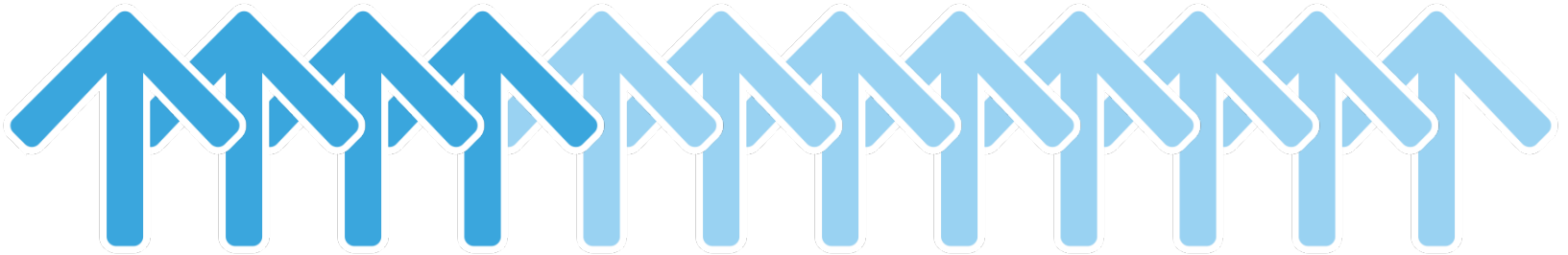
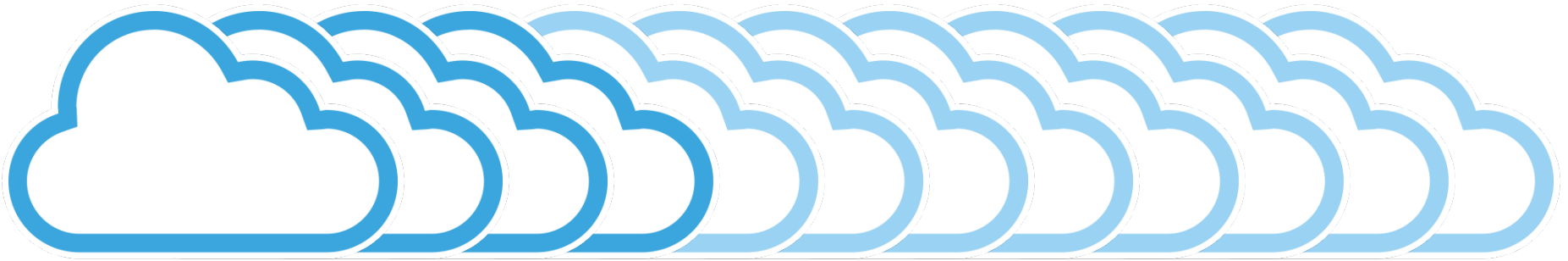


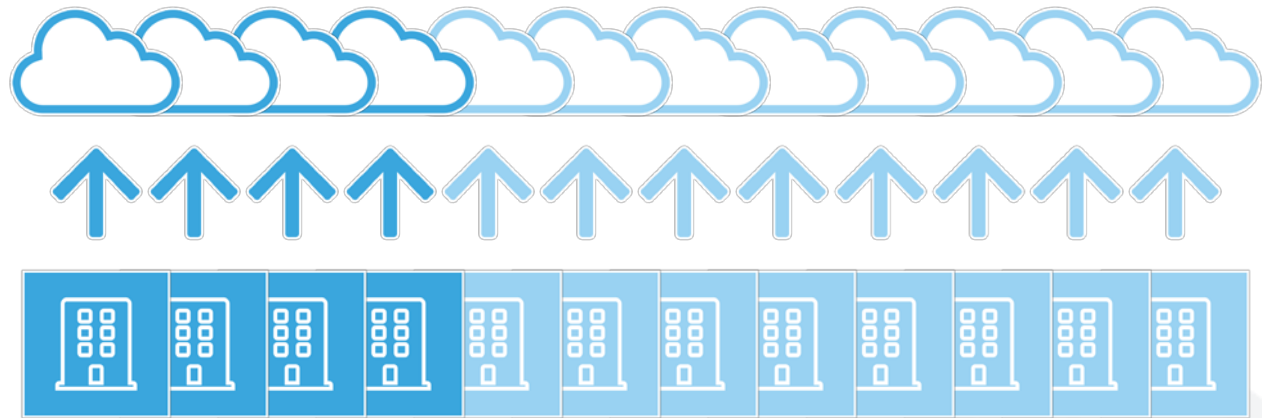
Cumulus Lessons Learned

BACKGROUND & LESSONS ON TEAM DYNAMICS

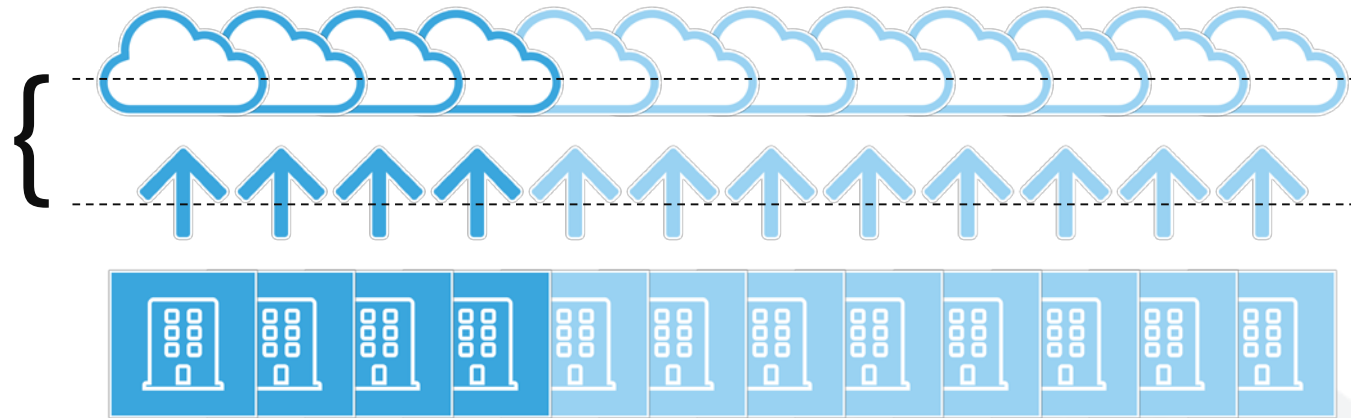




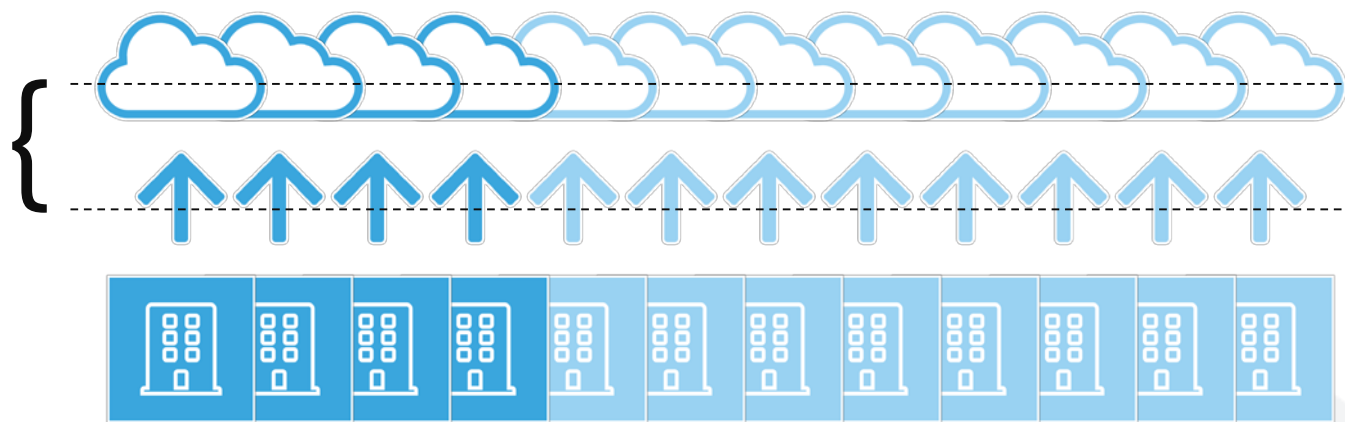




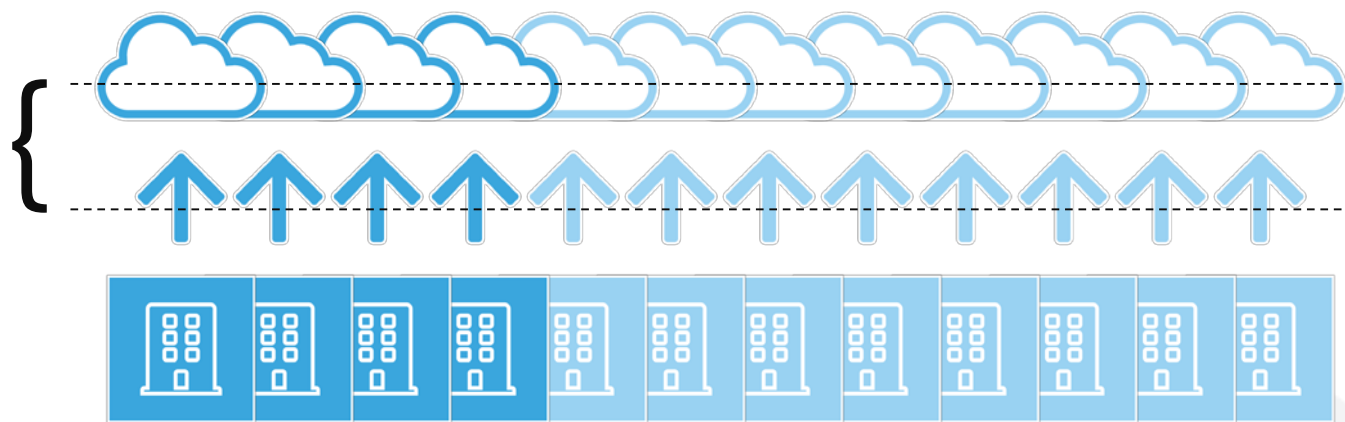
Cumulus



Distribution
Operator Tools
Archive Management
Cumulus
Common Ingest



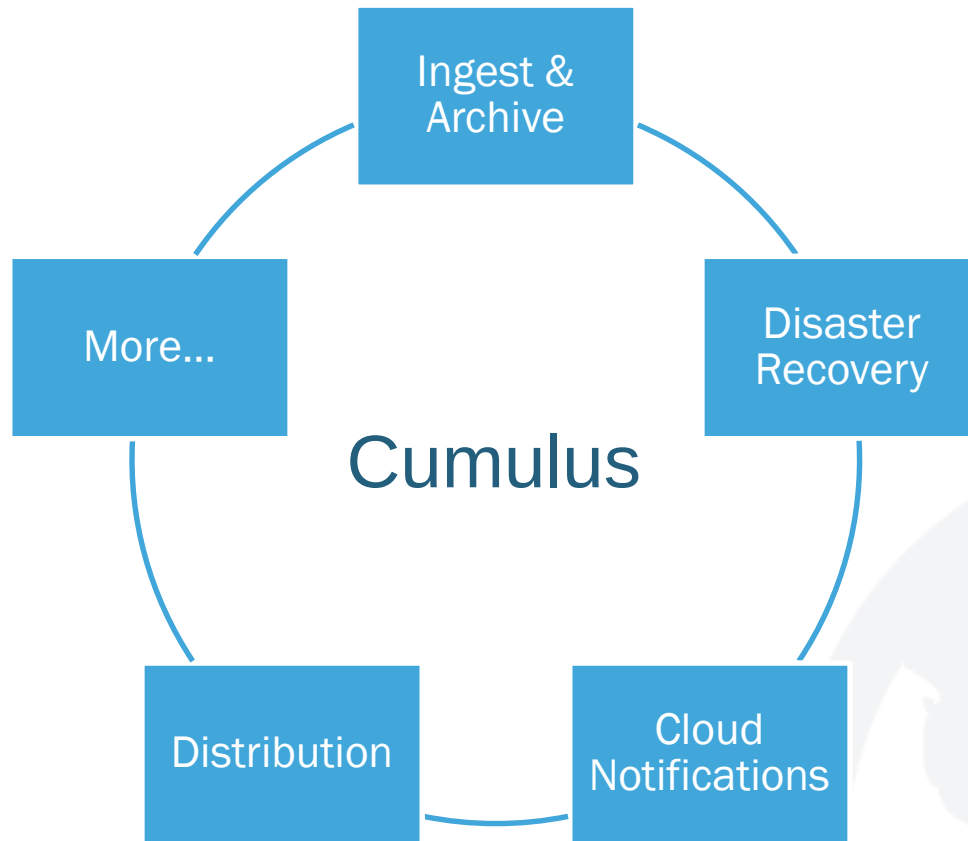
Distribution
Operator Tools
Archive Management
Cumulus
Common Ingest

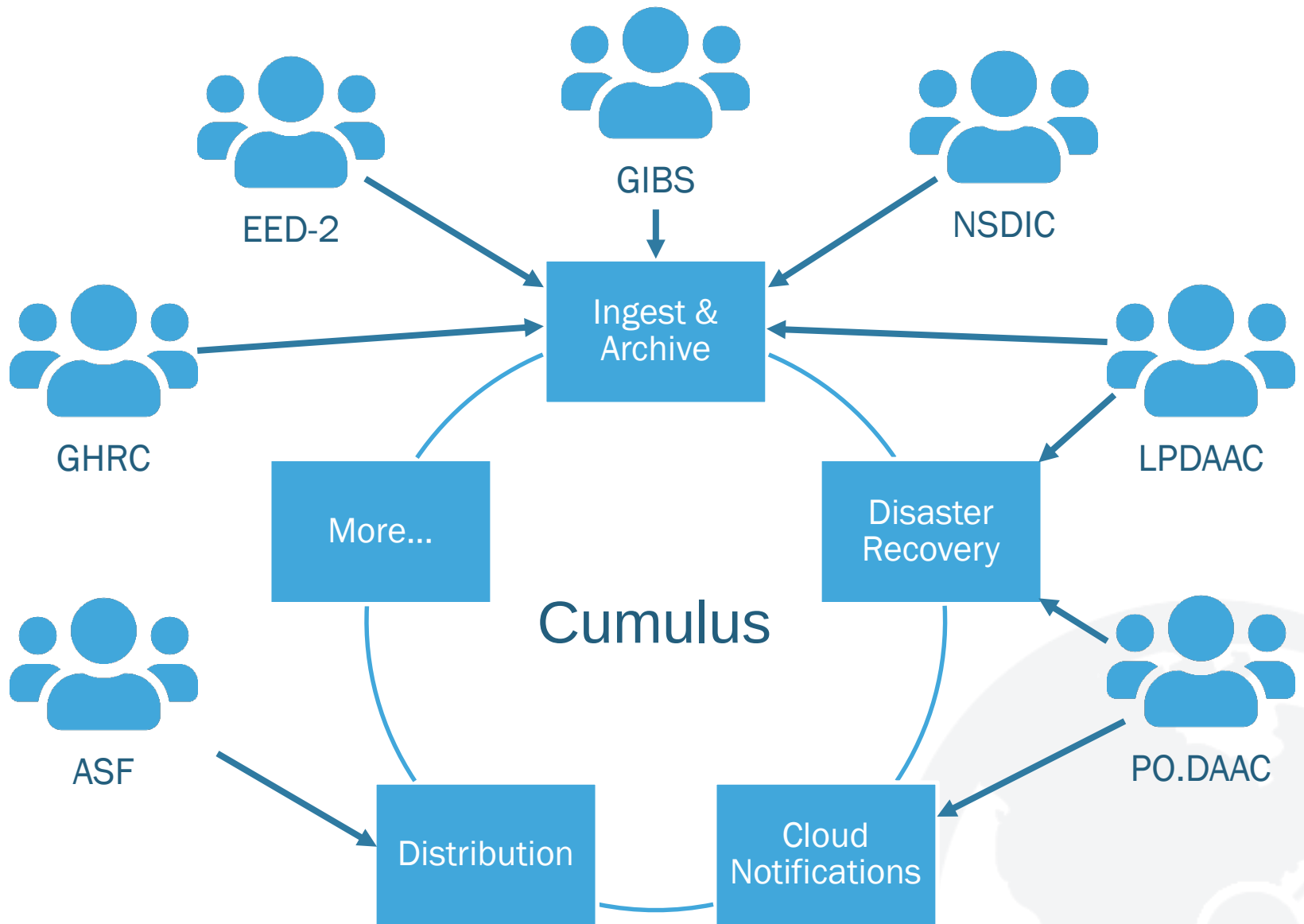


Not Cumulus

Truly unique tooling
Cloud Compliance Infrastructure (Earthdata Cloud)
Data services
Operations
Waffles

...





Takeaways

1. Sharing many separable components is better than trying to be everything to everyone
(This requires clear, consistent interfaces)
2. Archive and migration needs are rarely unique. Default to seeking out the work of others who have encountered similar needs
3. Archive and migration needs are rarely unique. Build tools expecting to share from the start

How we communicate



Slack

Informal,
ephemeral,
ad-hoc,
no permanent
decisions



Daily Scrum

Individual teams
meet/tag-up daily



Video/ Telecons

2x Weekly Tag-ups
“Scrum of Scrums”
Progress Reports



Biweekly Demos

Sprint Reviews



Special Working Groups

Dashboard
Architecture
Integrators

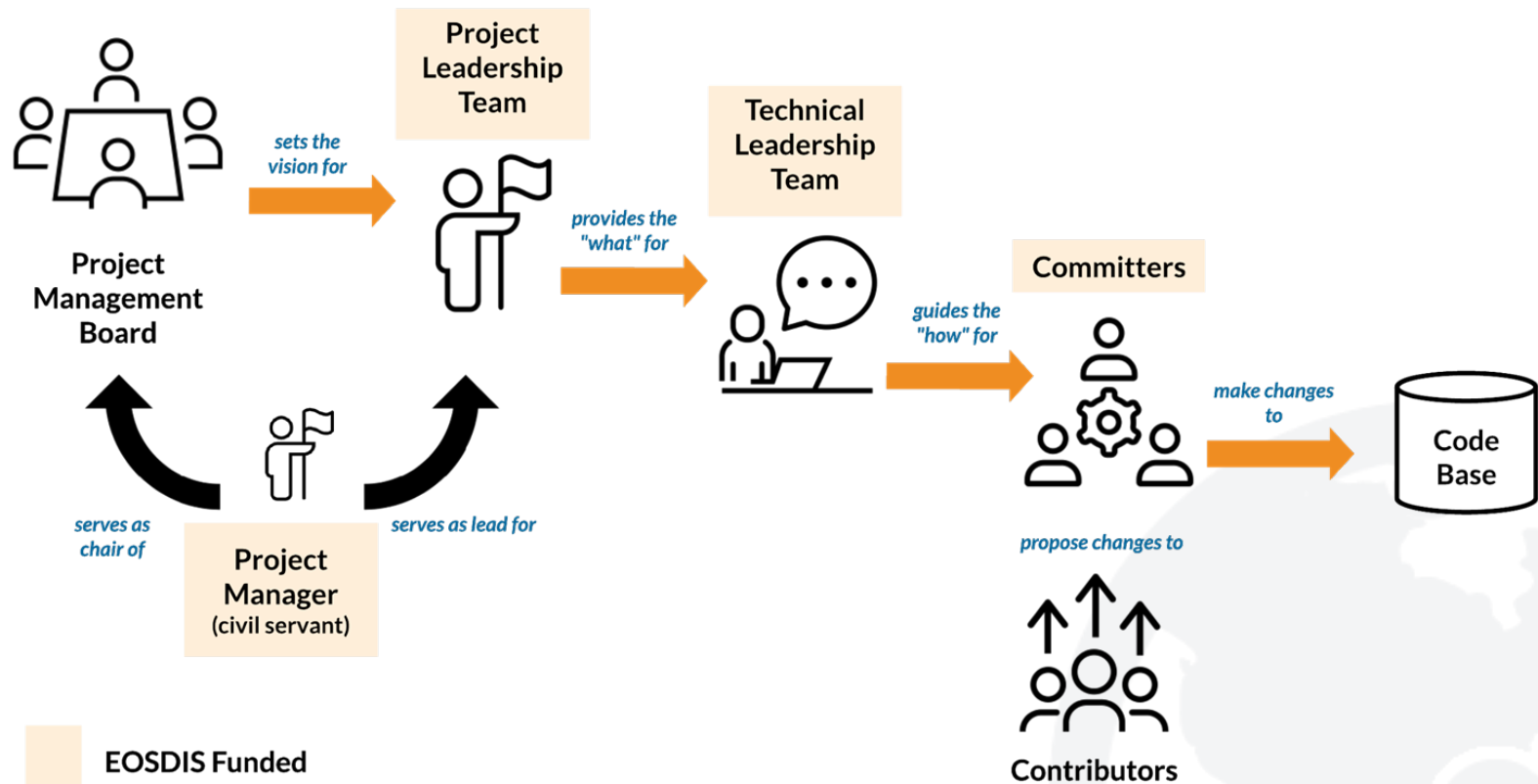
Less formal



Wiki Pages Document Meeting
Notes, Decisions, Action Items



Clear Avenues for Collaboration



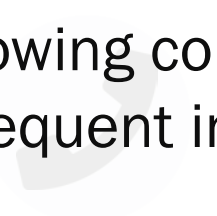
Takeaways

1. Distributed teams need to vigilantly build free-flowing communication while guarding against frequent interruptions
2. Once-a-sprint “architecture” meetings are great!



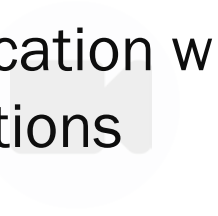
Slack

Informal,
ephemeral,
ad-hoc,
no permanent
decisions



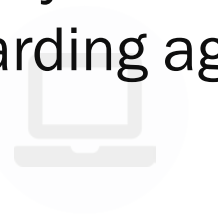
Daily
Scrum

Individual teams
meet/tag-up daily



Telecons

2x Weekly Tag-ups
“Scrum of Scrums”
Progress Reports



Demos

Sprint Reviews



Special
Working
Groups

Dashboard
Architecture
Integrators

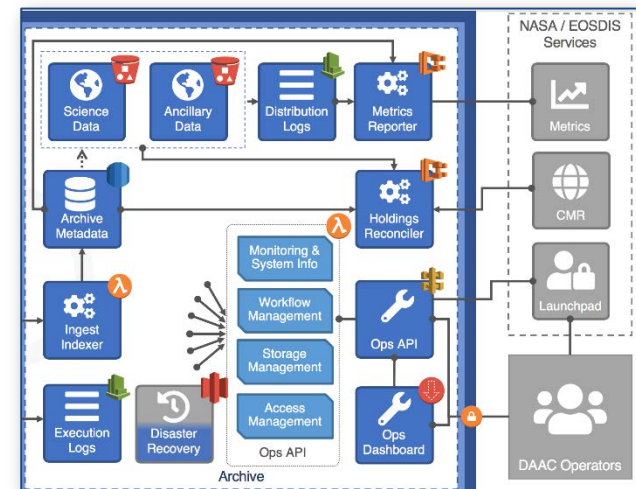
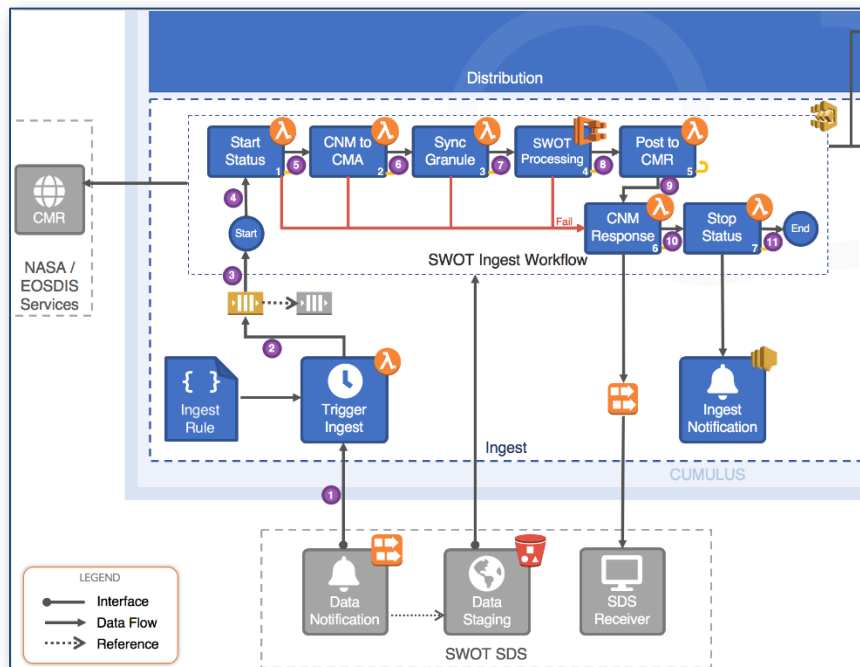
Less formal

Wiki Pages Document Meeting
Notes, Decisions, Action Items

Cumulus Lessons Learned

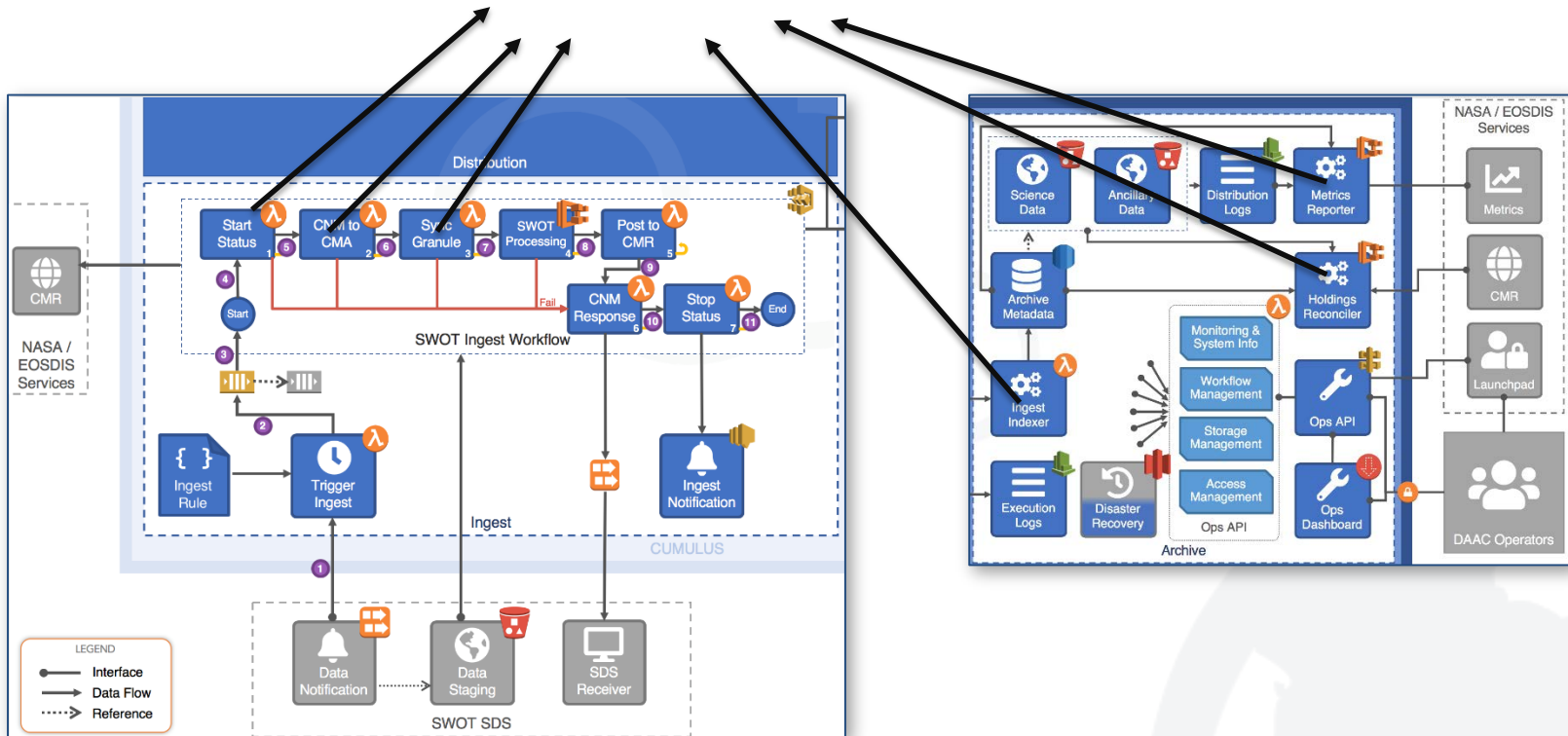
LESSONS ON CLOUD DEVELOPMENT

Testing Microservices



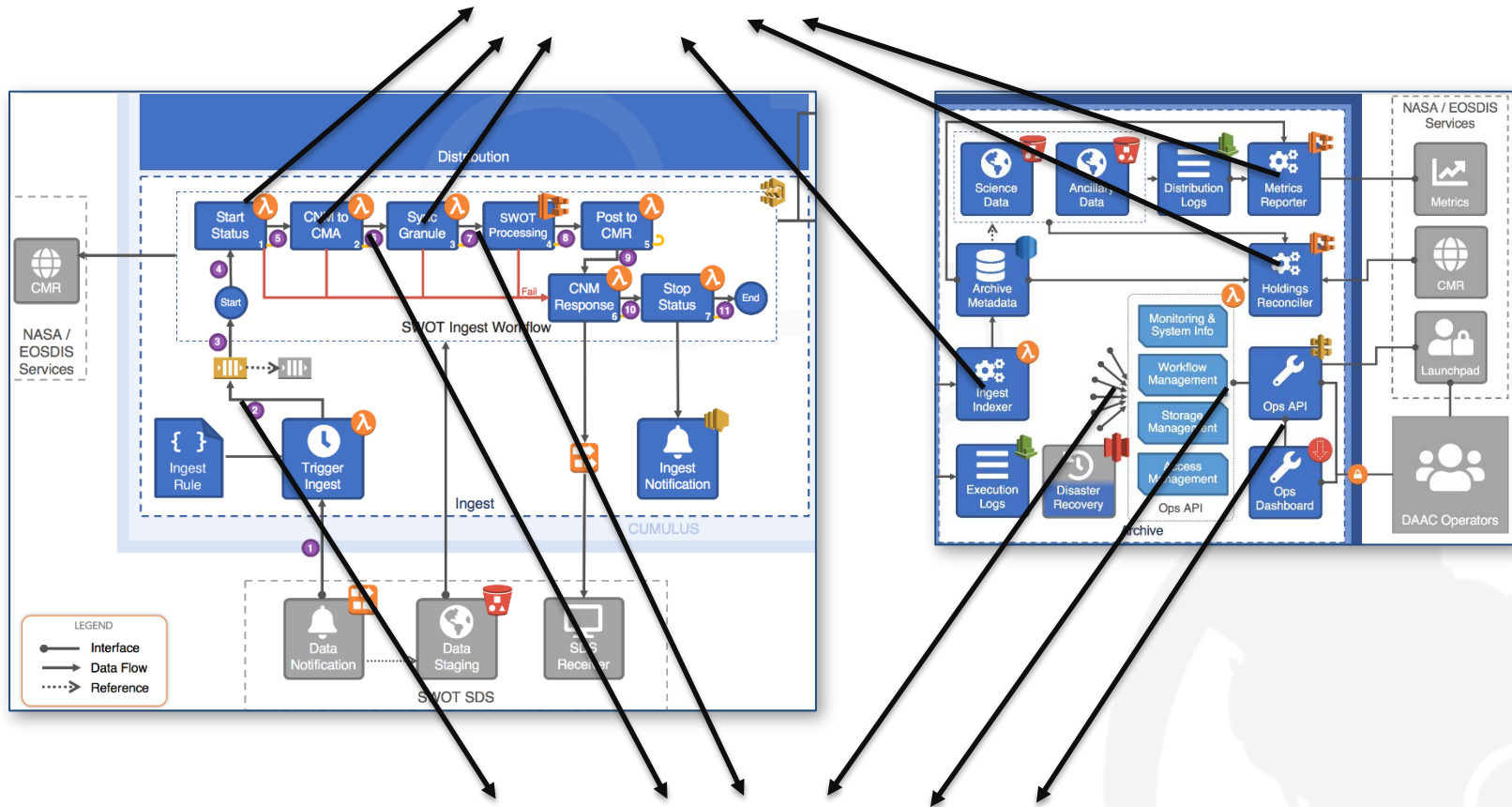
Testing Microservices

Tests for all the boxes are important (Unit tests)



Testing Microservices

Tests for all the boxes are important (Unit tests)



Tests for all the arrows are *critical* (Integration tests)

Integration Testing Practices

- We write acceptance criteria in consistent language conducive to testing
- We write an integration test for every acceptance criterion
- Deploying and running real services is slow. We use LocalStack for rapid dev and test cycles
- We allow the same code to be run in Docker or Lambda and speak to Step Functions and STDIN to maximize flexibility in testing

Beware JSON: Use Explicit, Simple Inputs & Outputs



CopyFiles



```
{
  "meta": {
    "workflow_name": "DiscoverGranules",
    "collection": {
      "files": [
        "s3://cumulus-example-private/example/file.nc",
        "s3://cumulus-example-private/example/file.nc.md5"
      ],
      "name": "http_testcollection_test-test-DiscoverGranules-1559660310213",
      "granuleIdExtraction": "^(.*)\\.(nc|nc\\.md5)$",
      "granuleId": "^.*$",
      "dataType": "L4_test-test-DiscoverGranules-1559660310213",
      "provider_path": "granules/fake_granules",
      "version": "001",
      "duplicateHandling": "replace",
      "sampleFileName": "example.nc",
    },
    "provider": {
      "protocol": "http",
      "globalConnectionLimit": 10,
      "port": 12345,
      "host": "127.0.0.1",
      "id": "myprovider-1234567",
    },
    "buckets": {
      "private": "cumulus-example-private",
      "protected": "cumulus-example-protected"
    }
  },
  "workflow_config": {
    "CopyFiles": {
      "sourceFiles": "${$.meta.collection.files}",
      "destinationBucket": "${$.meta.buckets.protected}"
    }
  }
}
```

Use Explicit, Simple Inputs & Outputs



CopyFiles



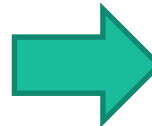
```
{
  "meta": {
    "collection": {
      "files": [
        "s3://cumulus-example-private/example/file.nc",
        "s3://cumulus-example-private/example/file.nc.md5"
      ]
    },
    "buckets": {
      "protected": "cumulus-example-protected"
    }
  },
  "workflow_config": {
    "CopyFiles": {
      "sourceFiles": "${$.meta.collection.files}",
      "destinationBucket": "${$.meta.buckets.protected}"
    }
  }
}
```



Which primitively-typed parameters to pass



Where to find them



```
{
  "sourceFiles": [
    "s3://cumulus-example-private/example/file.nc",
    "s3://cumulus-example-private/example/file.nc.md5"
  ],
  "destinationBucket": "cumulus-example-protected"
}
```

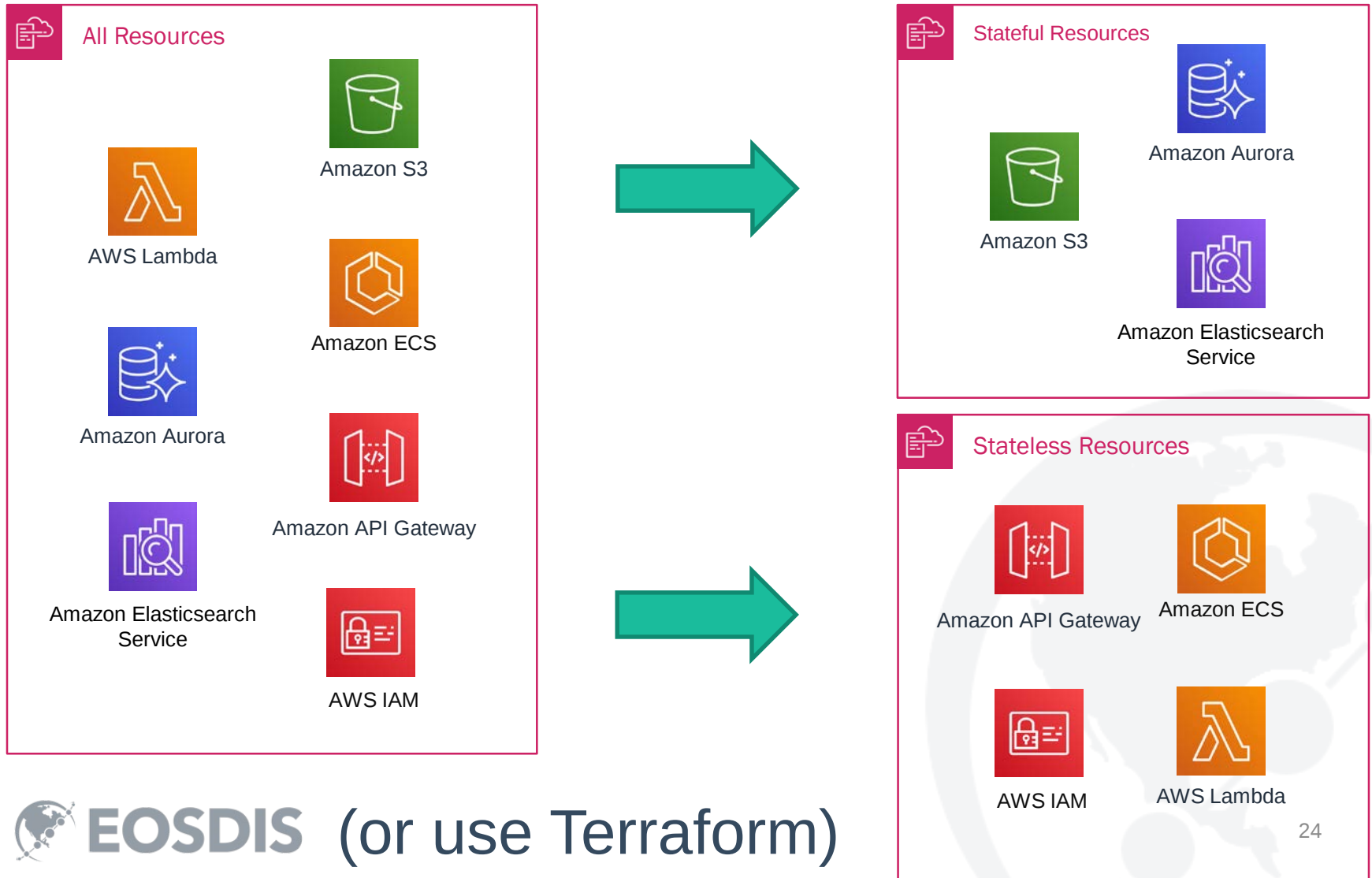
Why?

1. If the big JSON document changes, we update config and don't break code
2. It is clear and verifiable which keys the code does not use

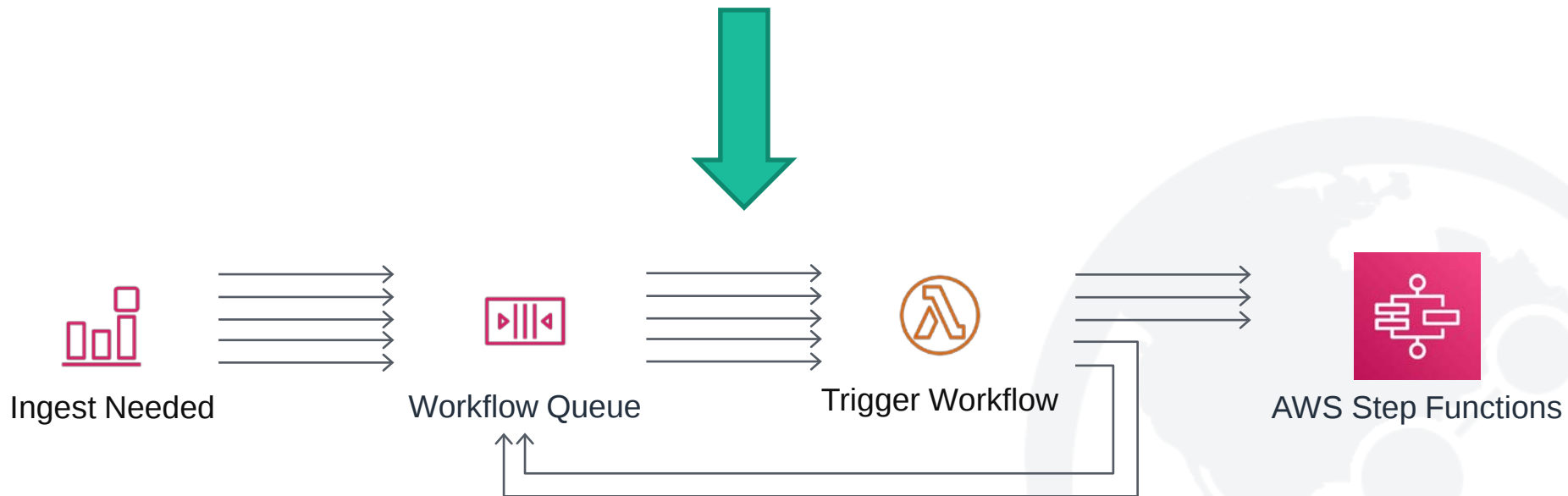
Cumulus Lessons Learned

LESSONS ON AWS USAGE

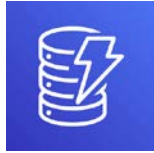
Separate Stateful Resources



Always Queue External Events



Sometimes Data is Relational!

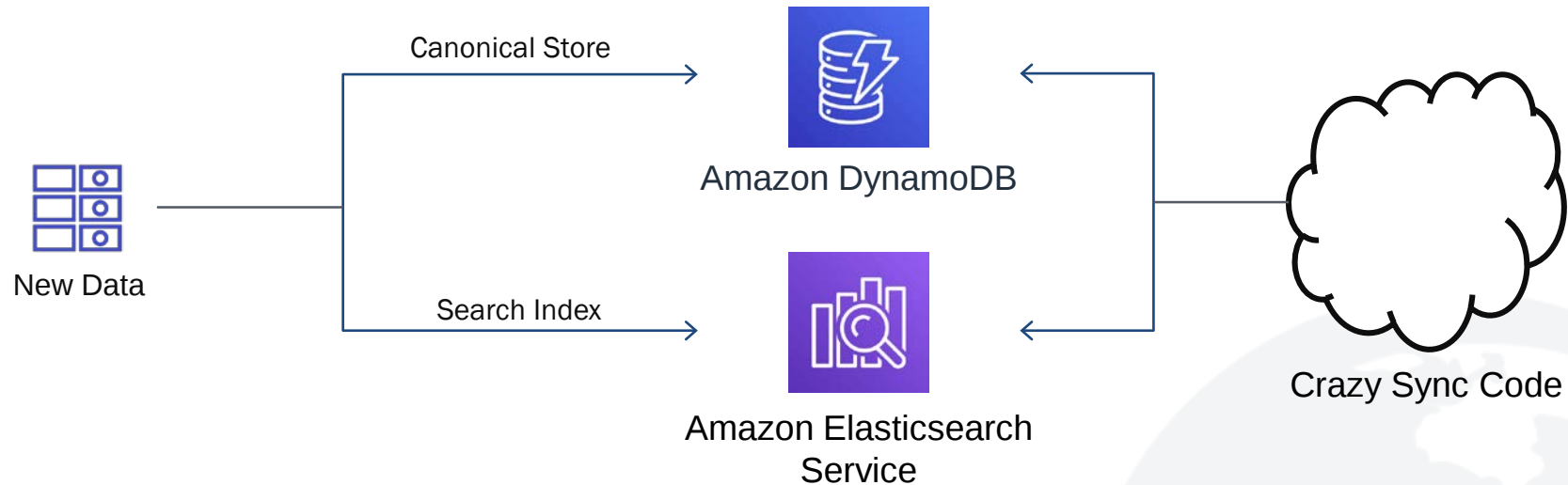


Amazon DynamoDB

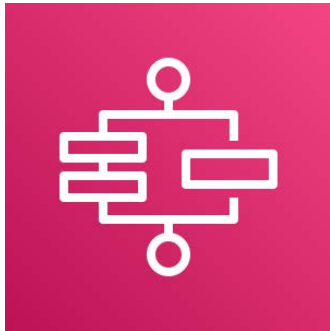


Amazon Aurora

Avoid Multiple Databases if you Need Consistency



Watch Announcements and Plan to Delete Code



AWS Step Functions

Obsolete Cumulus Code
Within 2 Years

- SQS-based workflows
- ECS Service Activity Polling
- Delivery to SQS
- Config from Input Message
- List Iteration

Thank You!

Patrick Quinn

patrick@element84.com

This work was supported by NASA/GSFC under
Raytheon Co. contract number NNG15HZ39C.

Raytheon

*in partnership
with*

