EOSDIS

NASA'S EARTH OBSERVING SYSTEM
DATA AND INFORMATION SYSTEM

Cloud Surprises in Moving NASA EOSDIS Applications into Amazon Web Services

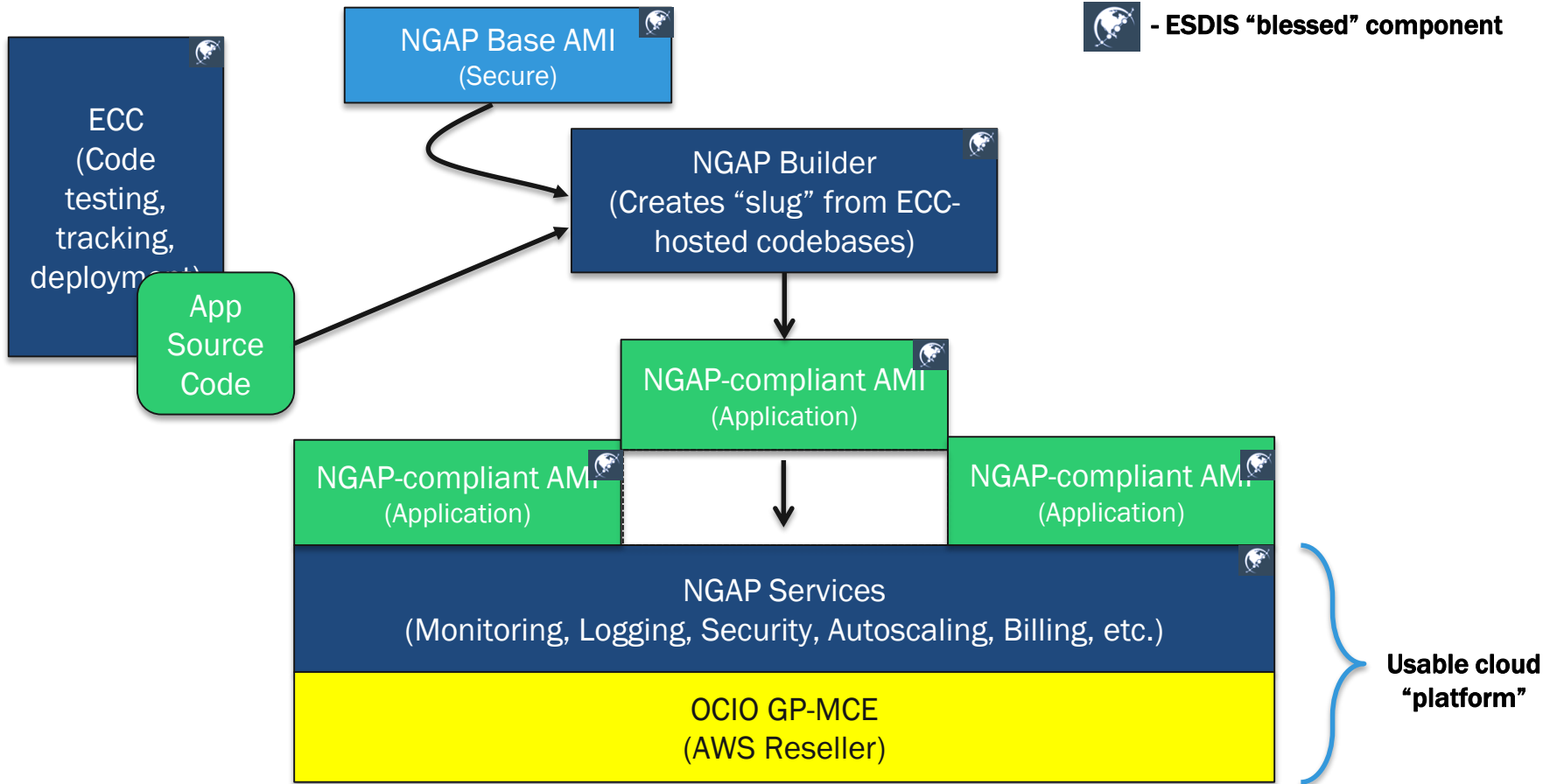Brett McLaughlin, NASA EED2 Program

1. NASA is not a Heroku or PaaS world

- *We thought* NGAP would primarily be *"Heroku for Earth Science"*
  - Hosting for web applications
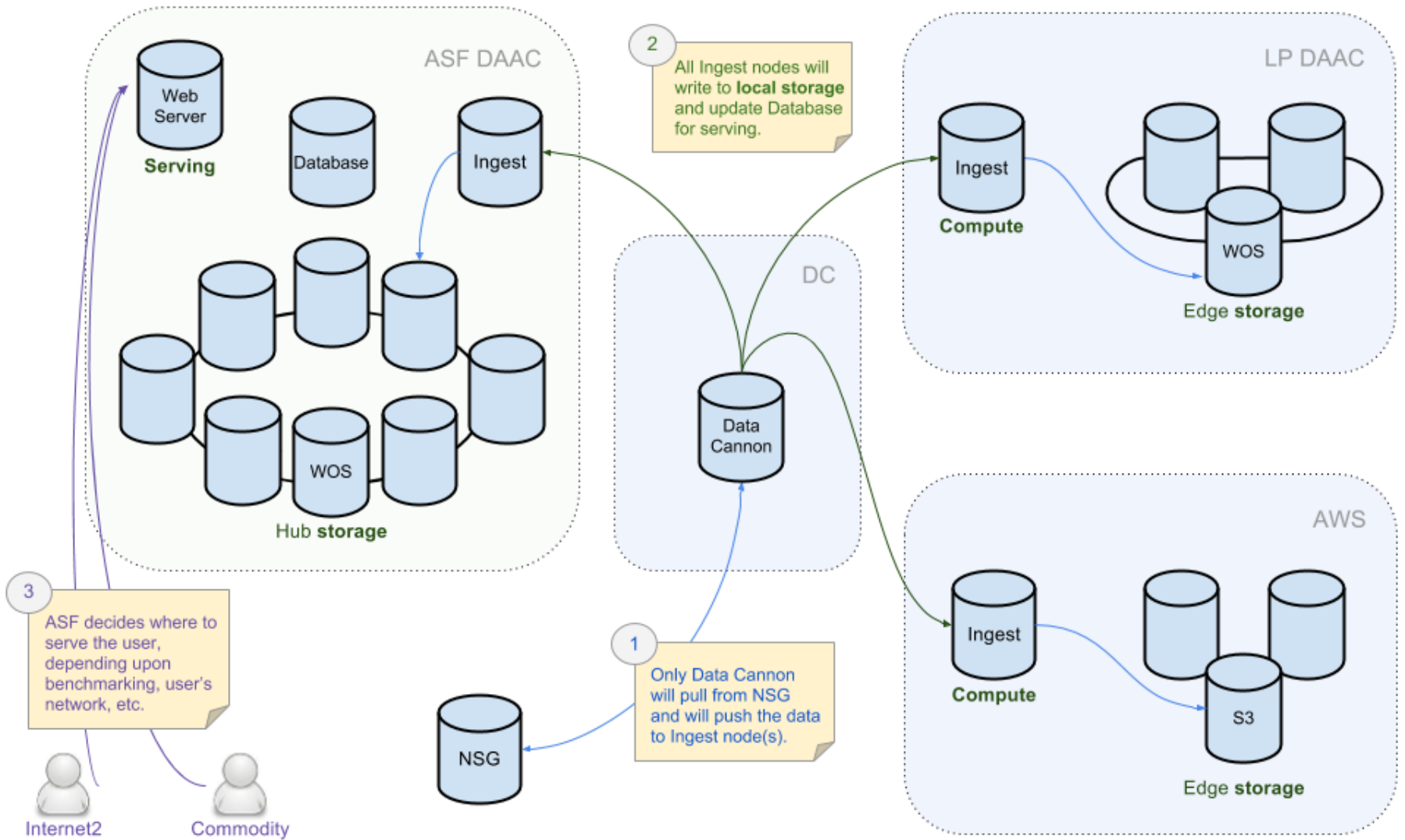 - Limited application profiles
  - Ease of Use

EOSDIS

"The needs of the many…"

- We were asked for a lot of different things
    - Hosting for ~~web applications~~ all the things
    - ~~Limited~~ Broad application profiles
    - Ease of Use (??)
- As the de facto cloud platform, NGAP had to evolve to meet these broader needs

EOSDIS

* https://pixabay.com/p-1458869   3

NGAP as a PaaS

AWS and WOS: Benchmarking at the Edges

ASF DAAC

Web Server
Serving

Database

Ingest

WOS

Hub **storage**

2 All Ingest nodes will write to **local storage** and update Database for serving.

LP DAAC

Ingest
Compute

WOS

Edge **storage**

DC

Data Cannon

AWS

Ingest
Compute

S3

Edge **storage**

3 ASF decides where to serve the user, depending upon benchmarking, user's network, etc.

1 Only Data Cannon will pull from NSG and will push the data to Ingest node(s).

NSG

Internet2   Commodity

DC refers to a Datacenter in lower 48 that is not yet defined.

The evolution from PaaS to (more) IaaS

- **NGAP 0.1**: 100% PaaS

- **NGAP 1.0**: 80% PaaS

- **NGAP 1.1**: 60% PaaS

- **NGAP Sandbox**: <50% PaaS

EOSDIS

2. Managed Services are the shizzle

- *We thought* NGAP would primarily be a *"hosting platform"*
  - Reduce hardware buys
  - Provide operational support for apps
- AWS *does* lots of cool stuff
  - Supplies resources (instances, networks, etc.)
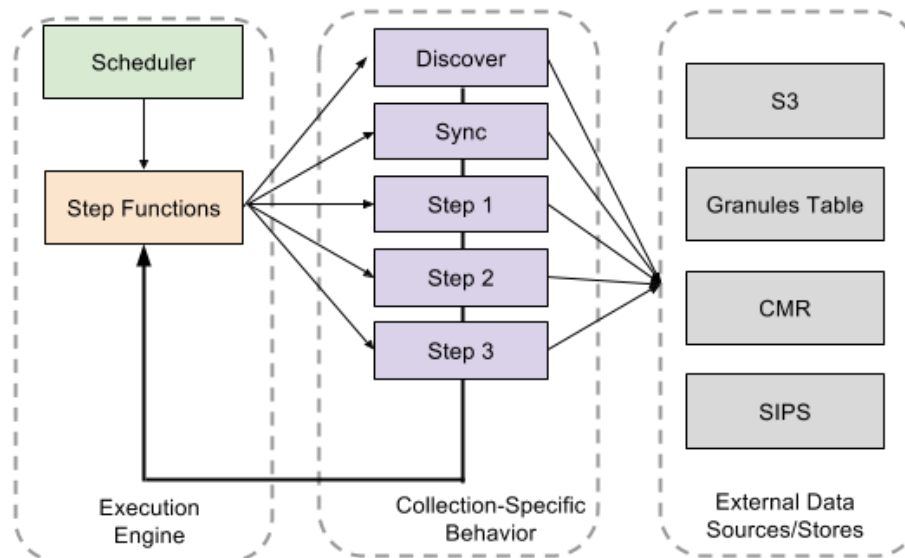 - Monitors and keeps those resources running

EOSDIS

Prefer Services Over Custom Code

- But AWS also *provides* some cool stuff ™
  - Lambda
  - Step Functions
  - AWS Batch
  - API Gateway
  - Something(s) since I wrote this presentation

Ingest/Processing Components without Step Functions
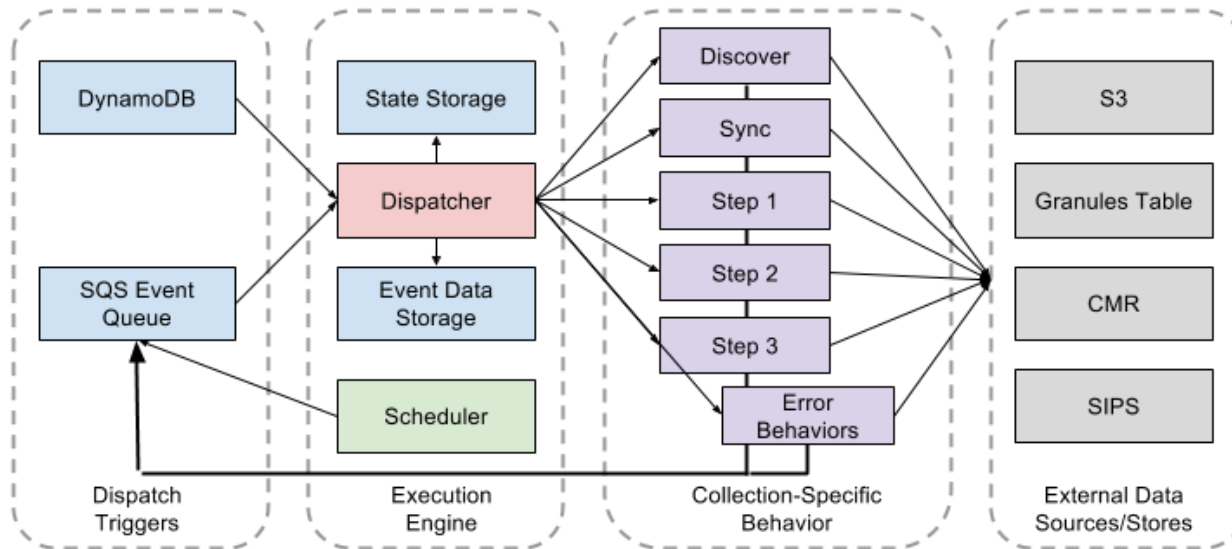
**Ingest/Processing Components using Step Functions**

3. EOSDIS Applications are as Sophisticated as AWS allows

- *We thought* that most of the applications we'd support would be *"web applications"*
  - Think Rails + database + S3
  - Think buildpacks
  - Think well-constrained technical problems

Managed Services Drive Innovation

- **Instances to Containers**
  - Greater segregation of functionality
  - Movement toward services over monoliths
- Software on an Instance to AWS Service
  - ElasticSearch to AWS ElasticSearch
  - RabbitMQ to AWS SQS
  - Etc.
- And bigger changes… *(more on that later)*

EOSDIS

GIBS to GIBS in the Cloud
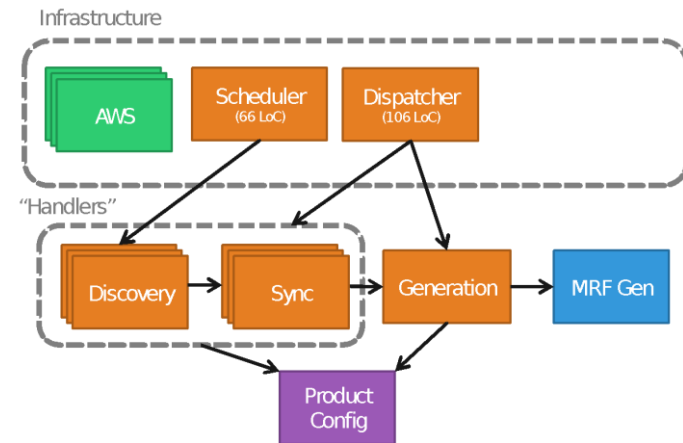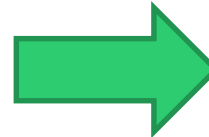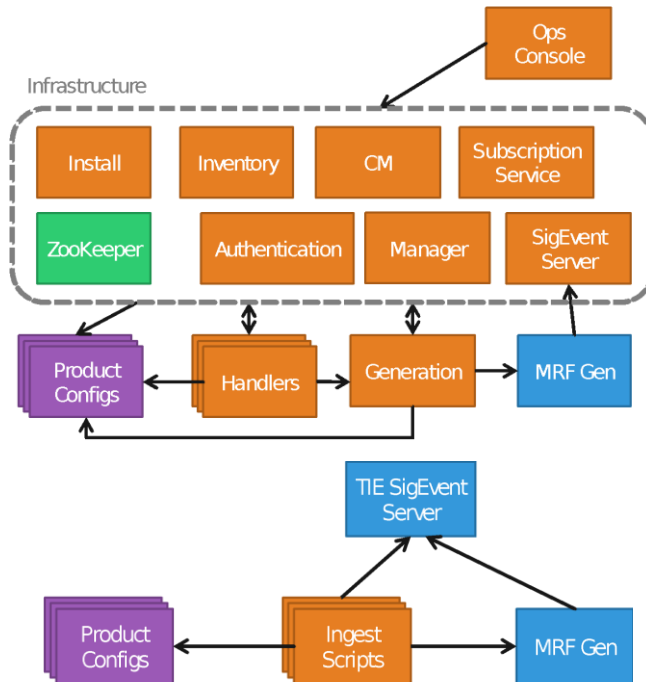
4. NGAP is not as easy as AWS to operate

- *We thought* that NGAP would basically offer the ease of operation that AWS offers to a typical application
  - Low-effort monitoring
  - Low-effort logging
  - Low-latency response times from operations

EOSDIS

Generated via http://wordle.net

5. AWS uses an open-ended spending model

- *We thought* we'd just turn on Amazon's billing controls and be A-OK.
  - Set spending limits
  - Produce granular billing reports
  - Limit egress at predetermined thresholds

Amazon provides the information and empowers the user

- ## Amazon wants to inform but not limit
    - AWS is happy to email you
    - AWS is happy to let you know what you're spending
    - AWS (reasonably) cannot force action, because "the action" is not standard

Egress (in particular) is a big deal

- When data leaves your application, service, data store, etc. …
  - …and goes to another region
  - …and goes outside of AWS
- Egress is expensive
  - Rack Rates: $0.08/GB after first 150TB
  - In other words, a significant portion of total monthly cloud-associated costs

EOSDIS

Cost isn't even the biggest issue

- A huge bill is bad…
  …but jail is worse.

- The Anti-Deficiency
  Act (ADA) disallows
  unbounded costs

- We need a means of
  absolutely limiting
  egress costs

![EOSDIS logo]

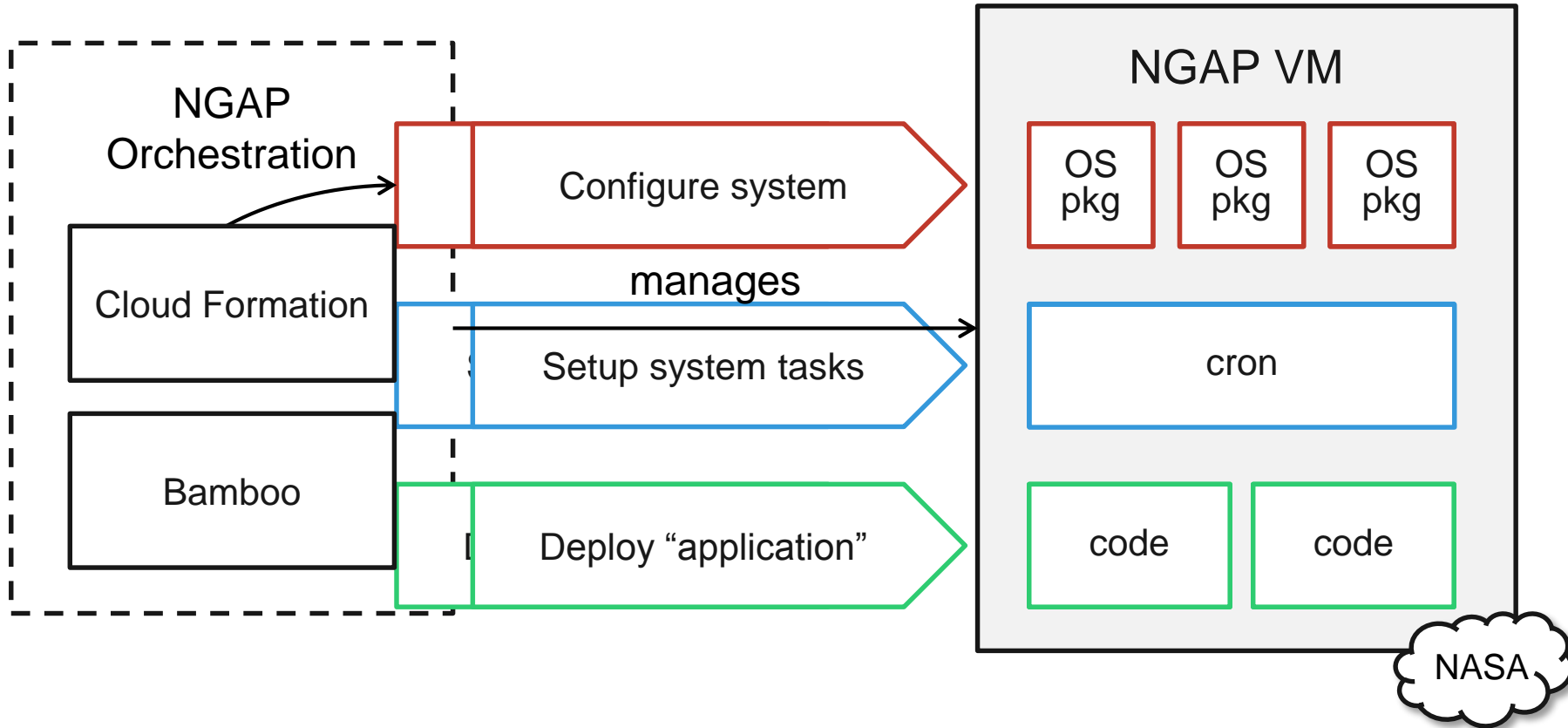** http://bit.ly/2AoJcsZ*

DM_PPT_NP_v01

6. Favor Re-architecture over "just getting into the cloud"

- *We thought* that many applications would simply move their architecture to NGAP and (more or less) call it a day

EOSDIS

DM_PPT_NP_v01

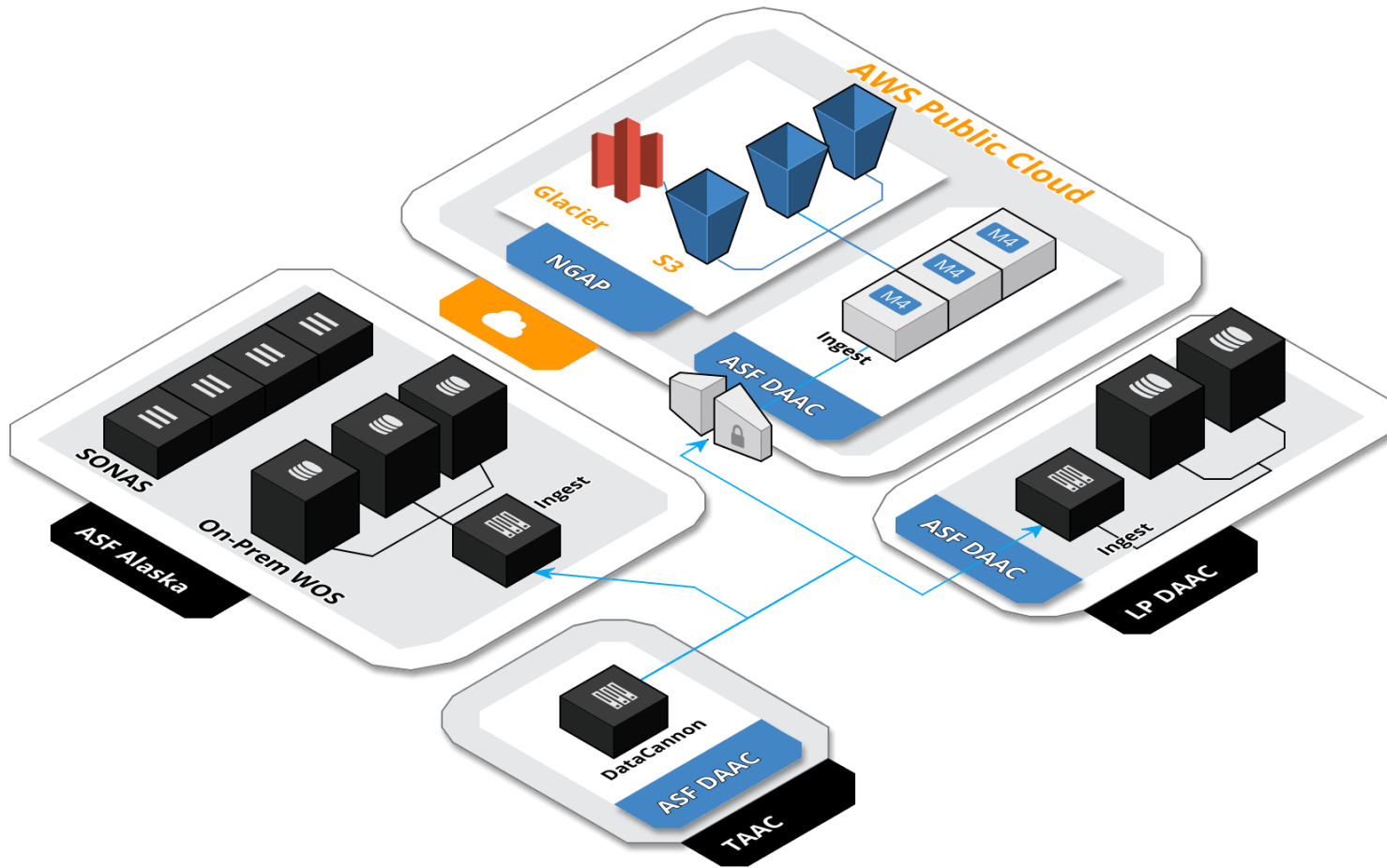# Case Study: ASF

EOSDIS
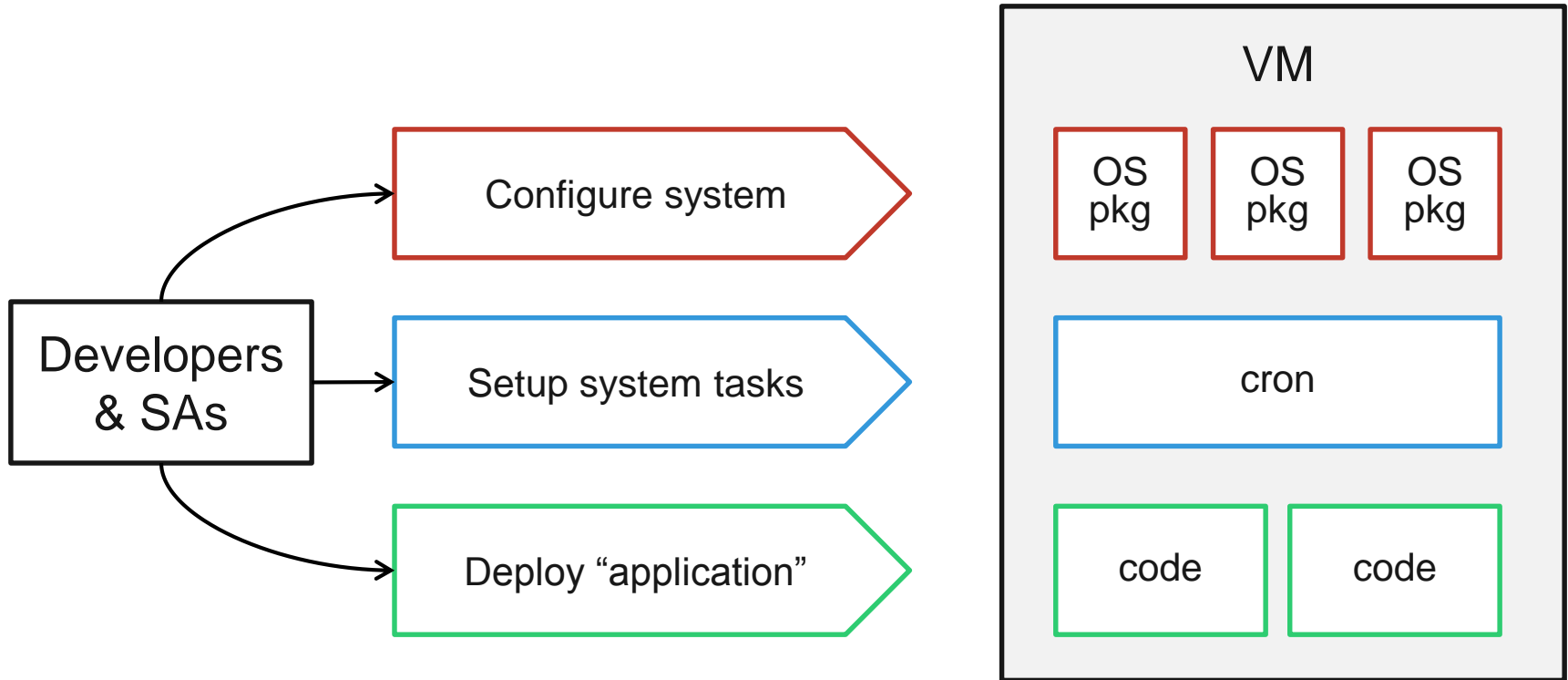
"Direct" Forklift onto NGAP

DM_PPT_NP_v01

But… it turns out…

"We wish we'd re-architected." – ASF

Why?
- Managed Services
- Natural Inflection Point
- Opportunity for Innovation

EOSDIS

ASF, Rearchitected

Created with https://cloudcraft.co/

DM_PPT_NP_v01

Ingest, Rearchitected

Created with https://cloudcraft.co/

EOSDIS

DM_PPT_NP_v01
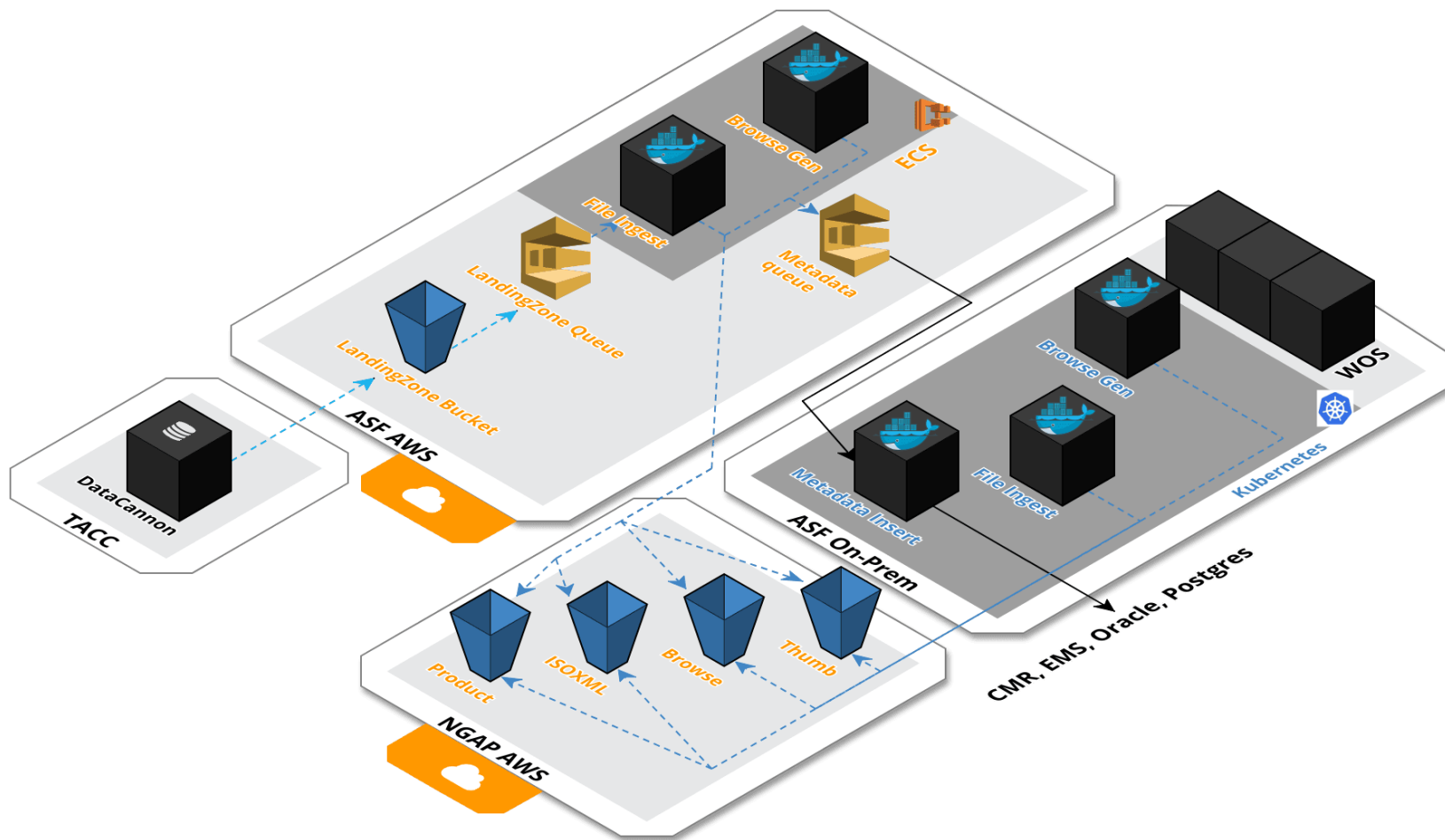
> *One of the great beauties of architecture is that each time, it is like life starting over again.*

-Renzo Piano

Cloud Surprises in Moving NASA EOSDIS Applications into Amazon Web Services

This material is based upon work supported by the National Aeronautics and Space Administration under Contract Number **NNG15HZ39C.**

Raytheon

EOSDIS