TERRA-REF Pipeline:
A Containerization Story

Max Burnette

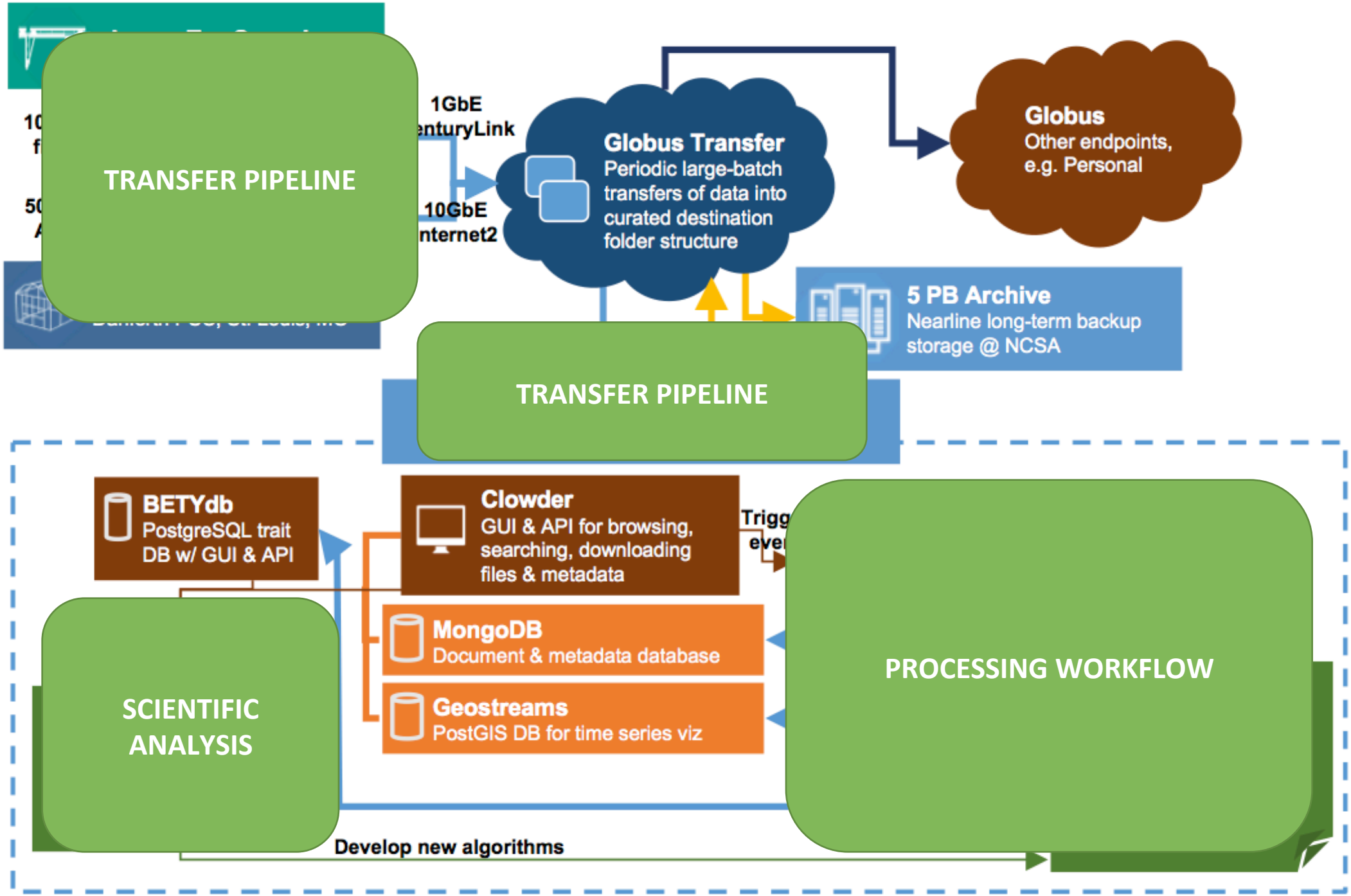
August 14, 2017

TERRAPHENOTYPING
REFERENCE PLATFORM

terraref.org

outline

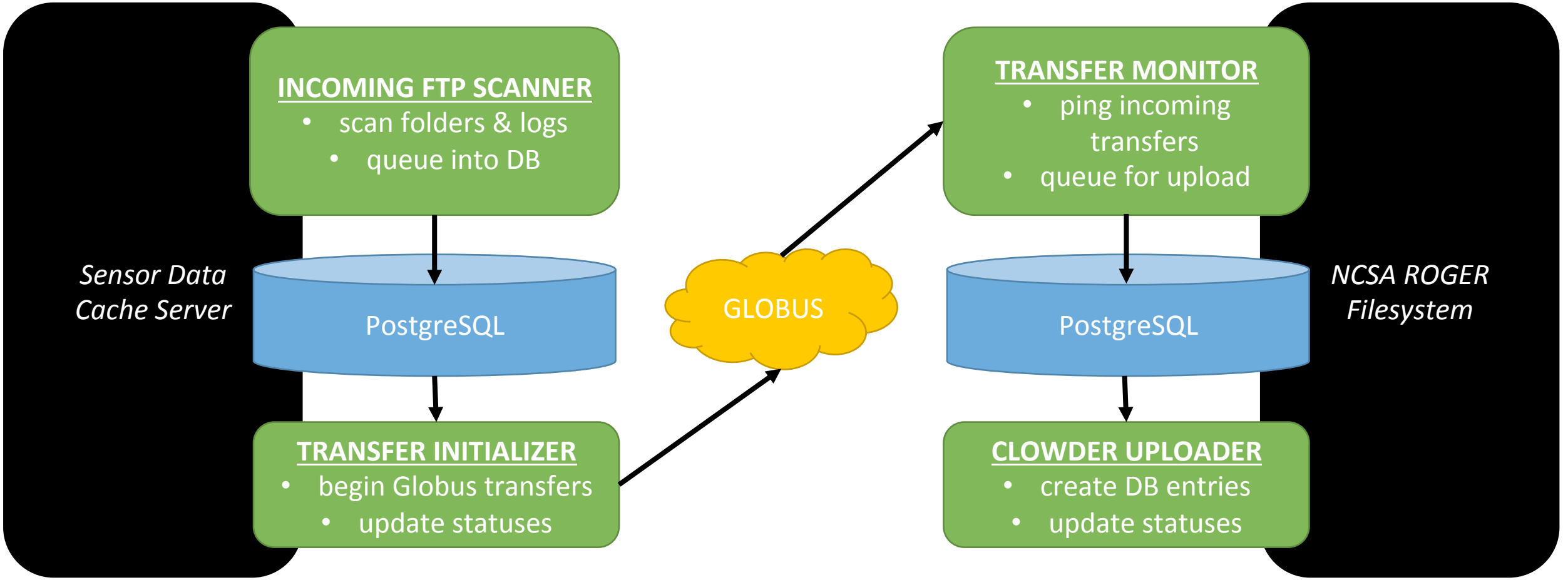
- Brief project overview
- Containers I - **transfer pipeline**
- Containers II - **processing workflow**
- Containers III - **scientific analysis**
- Q/A?


TRANSFER PIPELINE
1GbE
enturyLink
10GbE
nternet2
Globus Transfer
Periodic large-batch transfers of data into curated destination folder structure
Globus
Other endpoints, e.g. Personal
5 PB Archive
Nearline long-term backup storage @ NCSA
TRANSFER PIPELINE
BETYdb
PostgreSQL trait DB w/ GUI & API
Clowder
GUI & API for browsing, searching, downloading files & metadata
Trigg
eve
MongoDB
Document & metadata database
Geostreams
PostGIS DB for time series viz
SCIENTIFIC ANALYSIS
PROCESSING WORKFLOW
Develop new algorithms

TERRA-REF project goals

- **transfer pipeline**
  - move data from Arizona to Illinois efficiently
- **processing workflow**
  - handle large variety of sensor data in standardized scalable way
- **scientific analysis**
  - make tools, software & data easily accessible without large downloads

containers I - transfer pipeline



containers I - transfer pipeline

INCOMING FTP SCANNER
TRANSFER INITIALIZER
TRANSFER MONITOR
CLOWDER UPLOADER

THE GOOD

- **version/environment control**
  - avoid dependency desynchronization
  - hardware can change with minimal impact
  - same setup between development/testing/production
- **minimize downtime**
  - install & deploy container with software updates and new libraries with effectively zero switchover time
  - administration on deployment side is easy

containers I - transfer pipeline

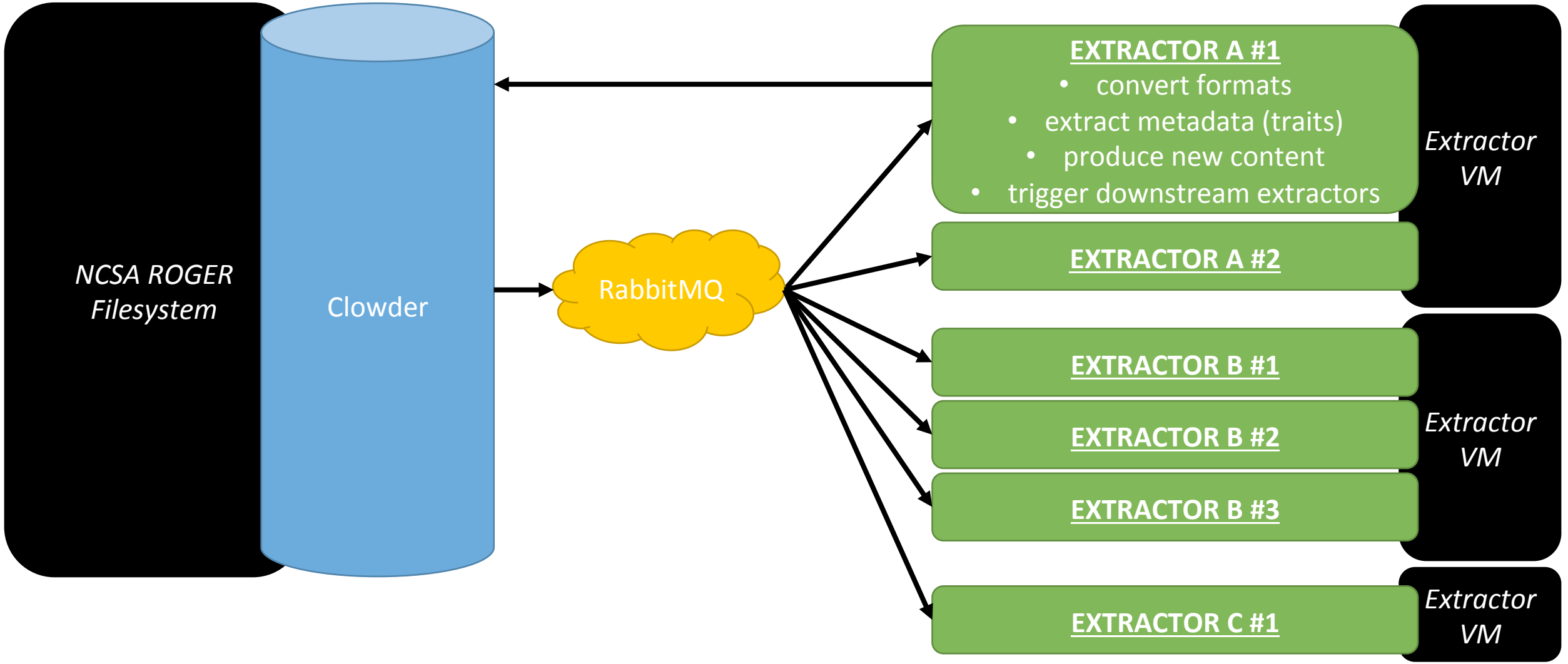
THE BAD

- **one more deployment step**
  - hotfixes and such become more strict

THE UGLY

- **permission control on mounted storage**
  - complex user/group rules need special accommodation
  - users with permission can run container and lose permission
  - can require manhandling of UIDs and GIDs of user

containers II - processing workflow



containers II - processing workflow

THE GOOD

- **highly scalable**
  - dynamically spin up or down full stacks instantly
  - separate temp space and logging*

EXTRACTOR A #1

- convert formats
- extract metadata (traits)
- produce new content
- trigger downstream extractors

THE UGLY

- **port exposure to outside world**
  - APIs and other HTTP services require additional forwarding
- **making sure logs aren't lost**
  - some container environments (Docker) will happily overwrite

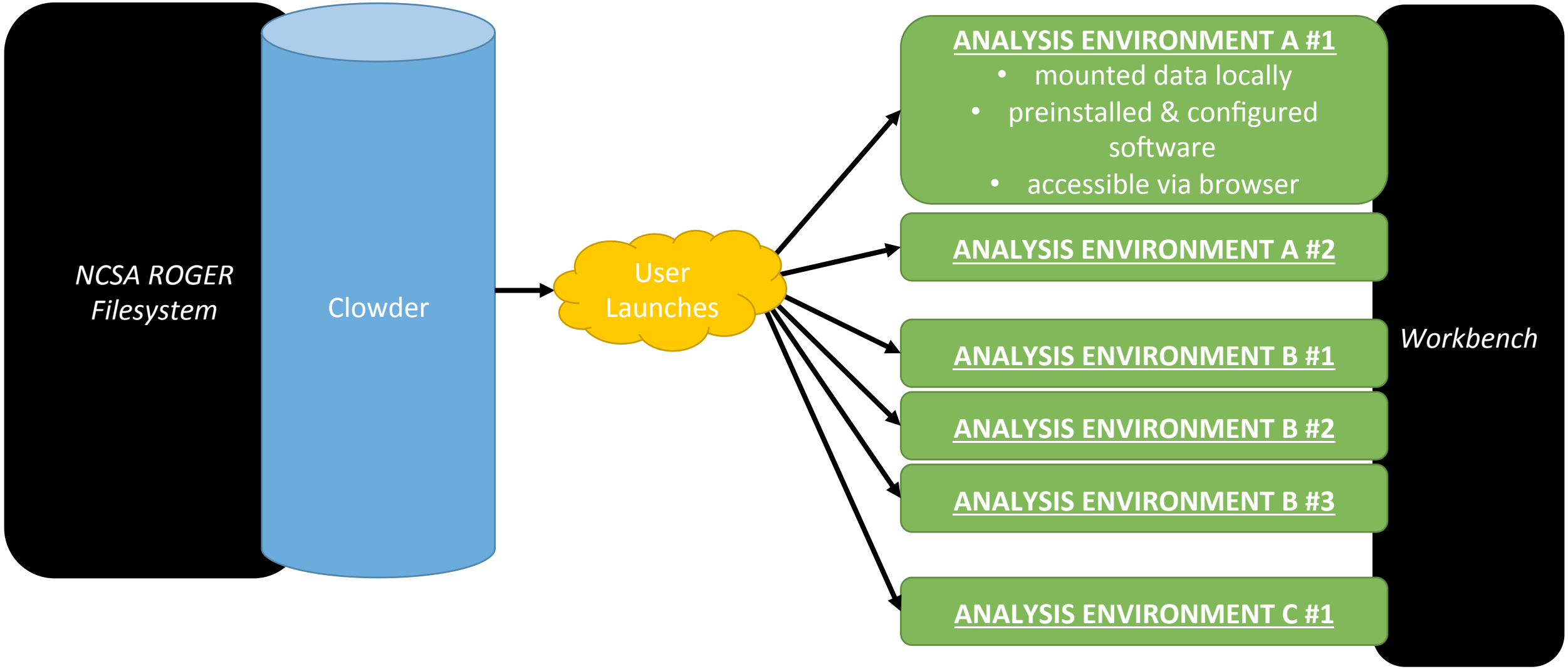
containers II - processing workflow

EXTRACTOR A #1
- convert formats
- extract metadata (traits)
- produce new content
- trigger downstream extractors

THE BAD

- **recreating external file structures**
  - some code written against expected paths
  - even mounted directories must replicate those, or account for them
  - “local” files will not appear local unless correctly configured

containers III - scientific analysis



containers III - scientific analysis

ANALYSIS ENVIRONMENT A #1

- mounted data locally
- preinstalled & configured software
- accessible via browser

THE GOOD

- **easy to use**
  - no installation or tracking versions
  - identical environments across users
- **sharable**
  - simply sharing URLs and credentials is sufficient

THE BAD

- **ephemeral by nature**
  - Expectations of saving data between sessions?
  - How to move data in and out of a remote container?

Q/A?

- Brief project overview
- Containers I - transfer pipeline
- Containers II - processing workflow
- Containers III - scientific analysis
- “huh?”