

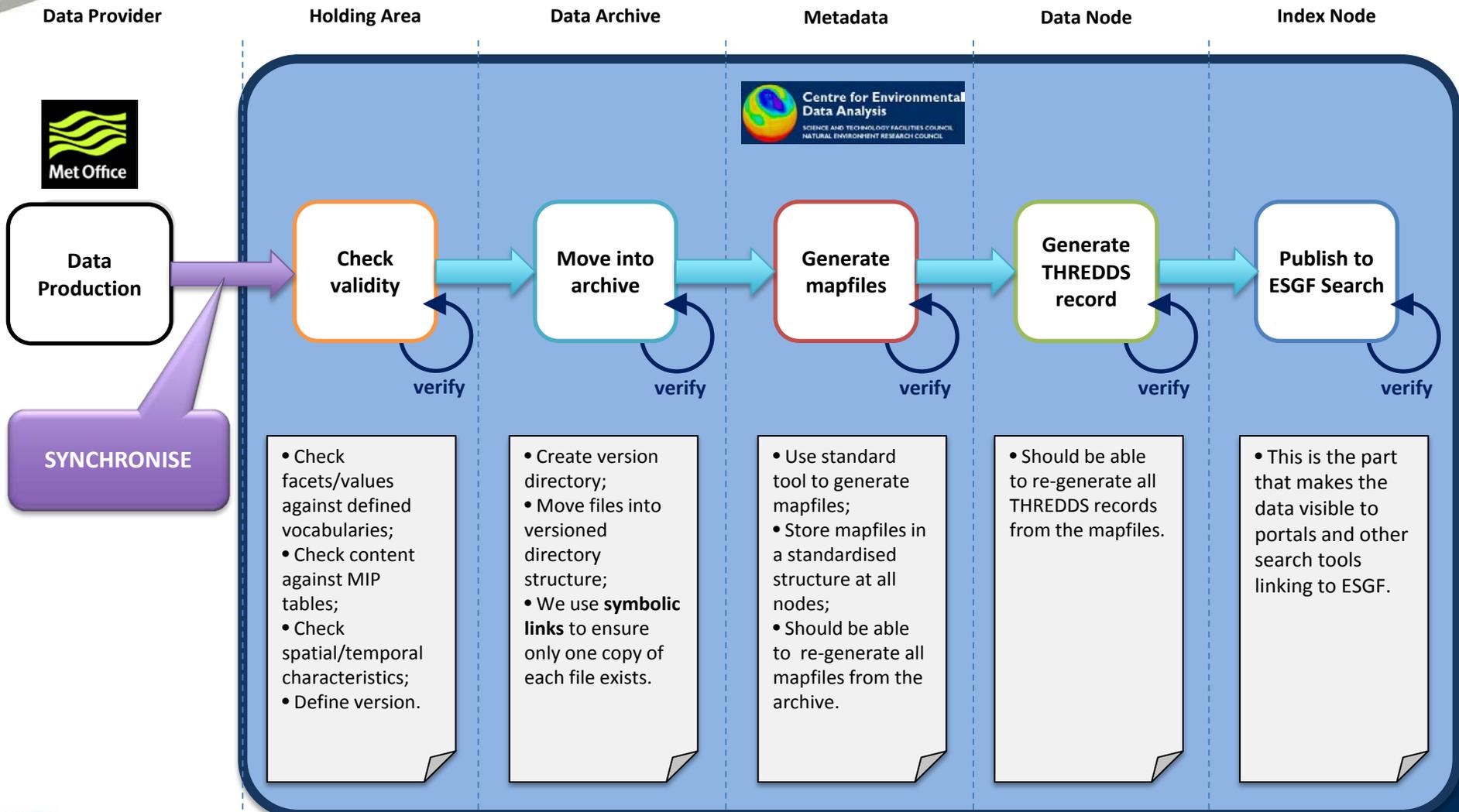


Automating Data Synchronization, Checking, Ingestion and Publication for CMIP6

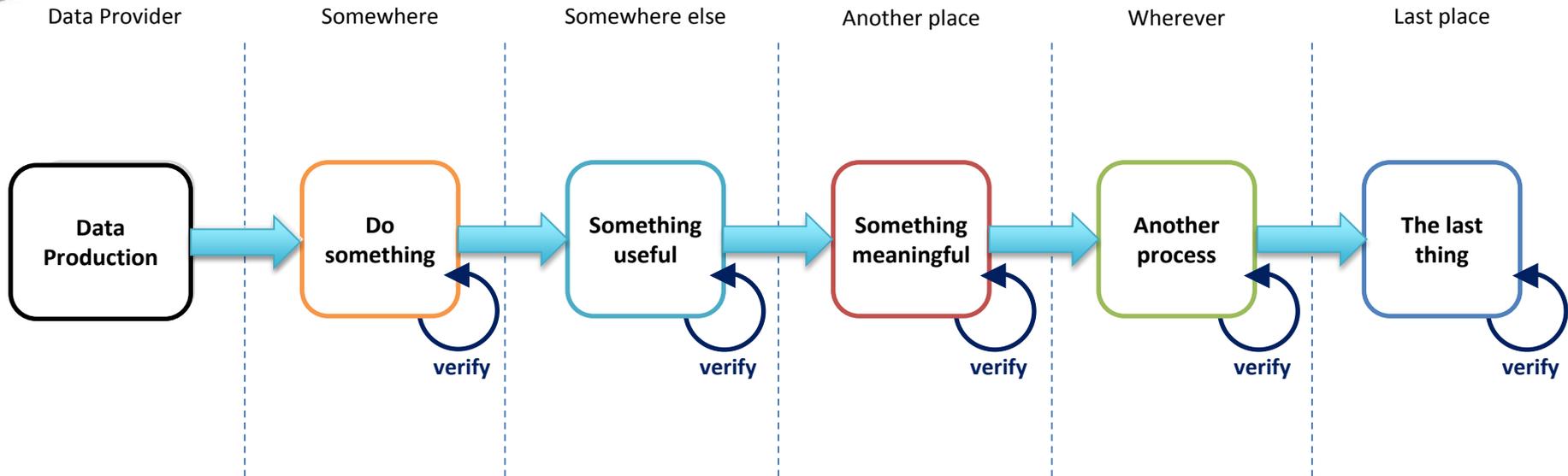
Ag Stephens & Alan Iwi
(STFC Centre for Environmental Data Analysis)
Thanks to Emma Hibling & Mark Elkington
(Met Office)



Ingest-publish workflow: Requirement

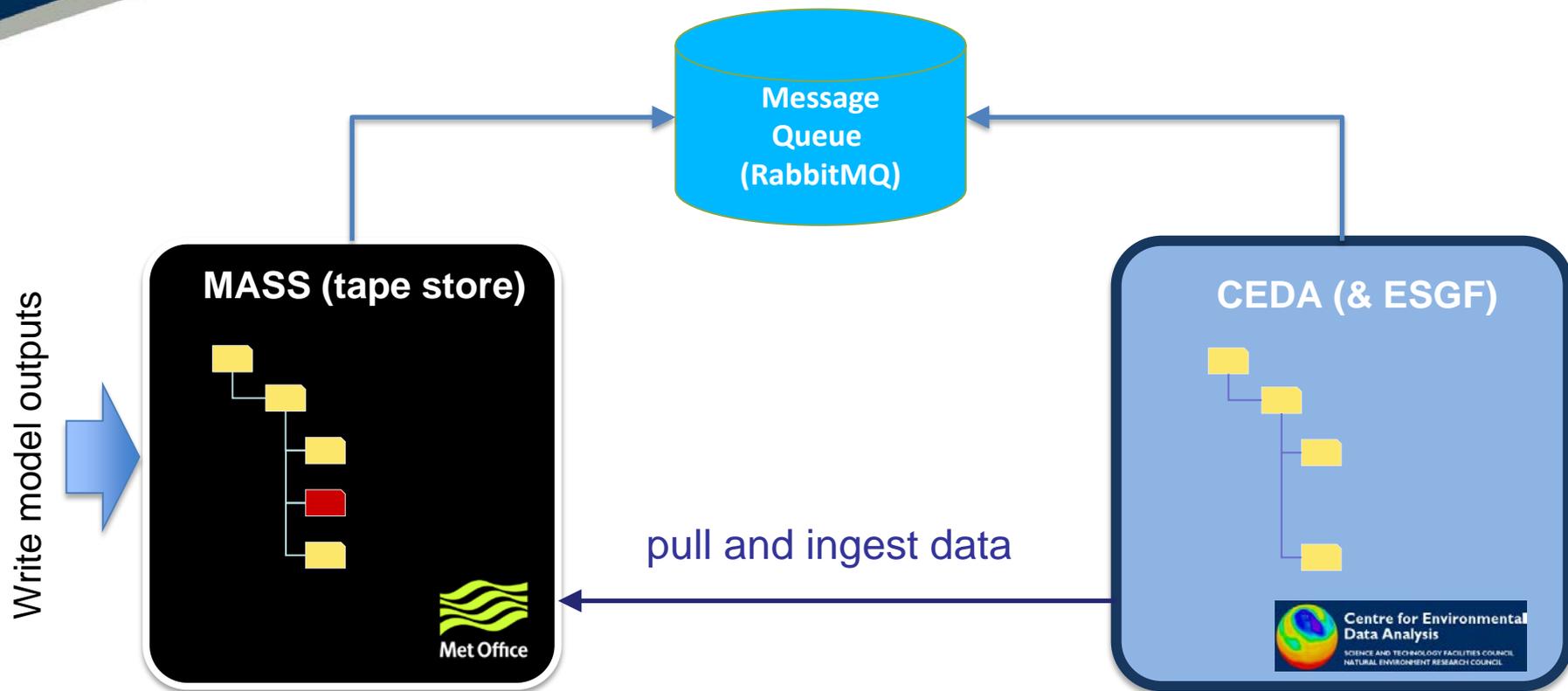


The generalised requirement



- For different datasets (ESGF or *other*) we can imagine there are different processes.
- Common pattern:
 - Each task is isolated.
 - "UNDO" behaviour may be desirable (e.g. "remove files", "unpublish")
 - Tasks may be managed by:
 - Different (Unix) users
 - With different access levels
 - On different servers.

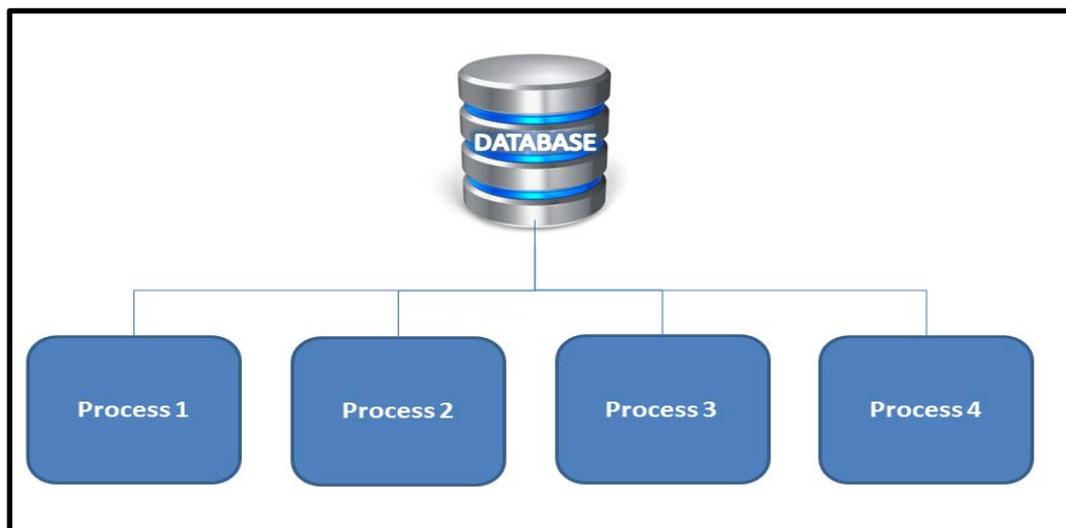
MOHC-to-ESGF pipeline - CMIP6



With special thanks to
Emma Hibling and **Mark
Elkington** (Met Office)

CEDA REceive-to-Publish Pipeline (CREPP)

We have called it "CREPP" - currently an internal ("cedadev") GitHub project.



Client-server architecture:

- "Server" is actually just a DB.
- "Clients" are any number of Controllers on any number of machines.

CREPP: Key concepts

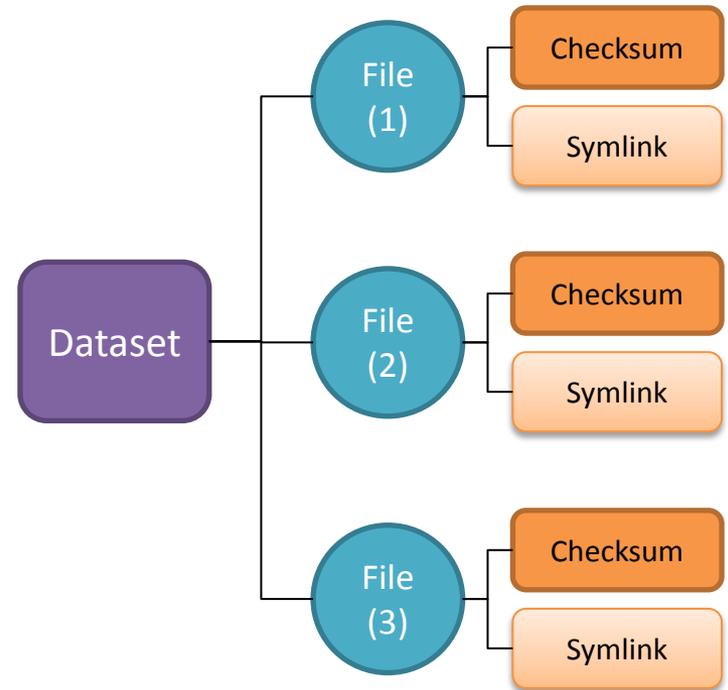
- **Dataset ID (ESGF)** is the unit of granularity across the system, e.g.:

```
cordex.output.CAS-44.MOHC.ECMWF-ERAINT.evaluation.r1i1p1.  
MOHC-HadRM3P.v1.mon.clt.v20150608
```

- All Controller actions should be **atomic** wherever possible. This will maximise the chances of “DO” and “UNDO” being possible for each Process Stage.

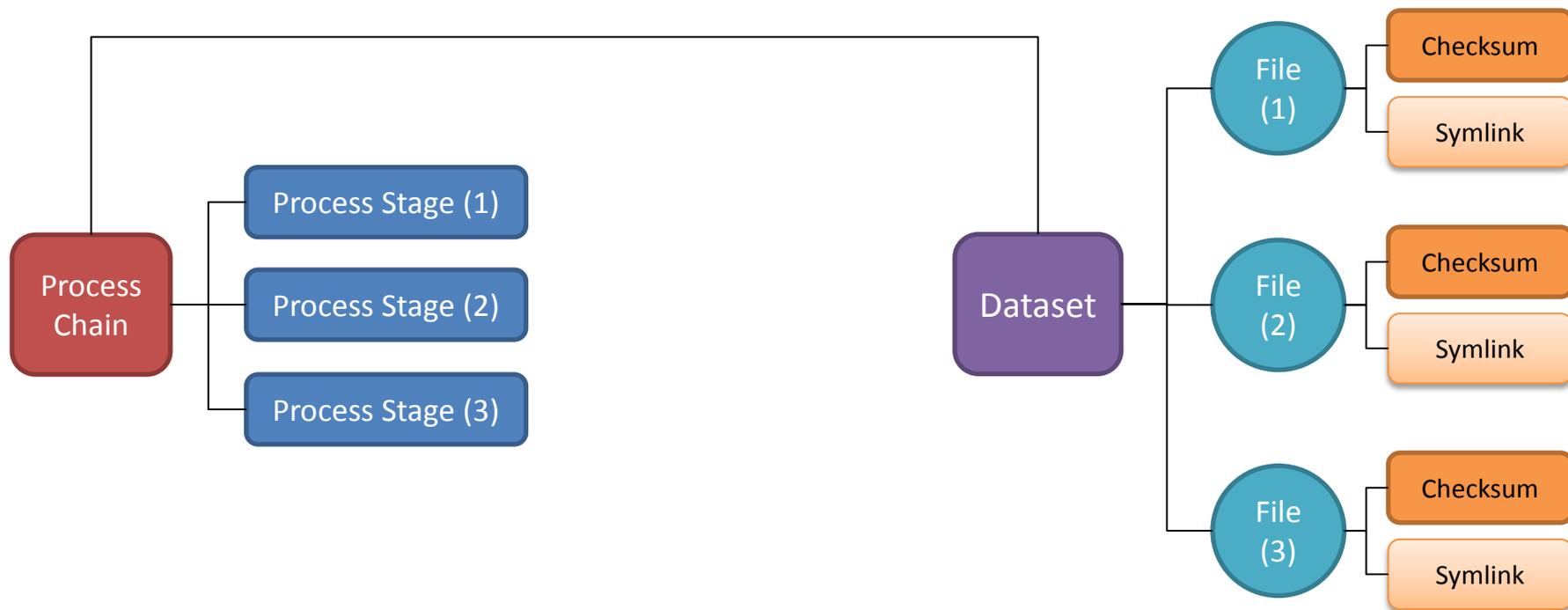
Data Model

The Data Model is built in Django...each type of box is a relational DB table.



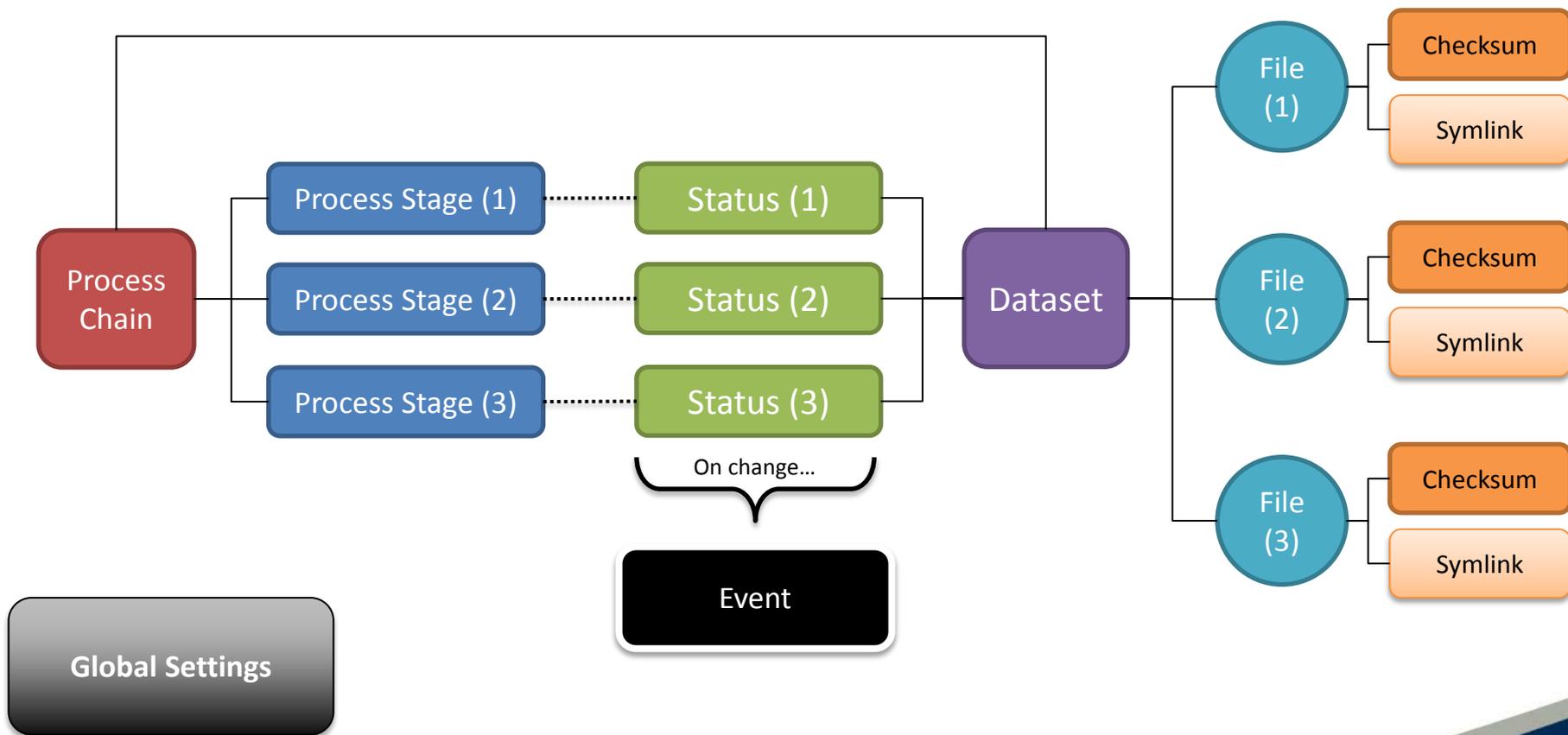
Data Model

The Data Model is built in Django...each type of box is a relational DB table.



Data Model

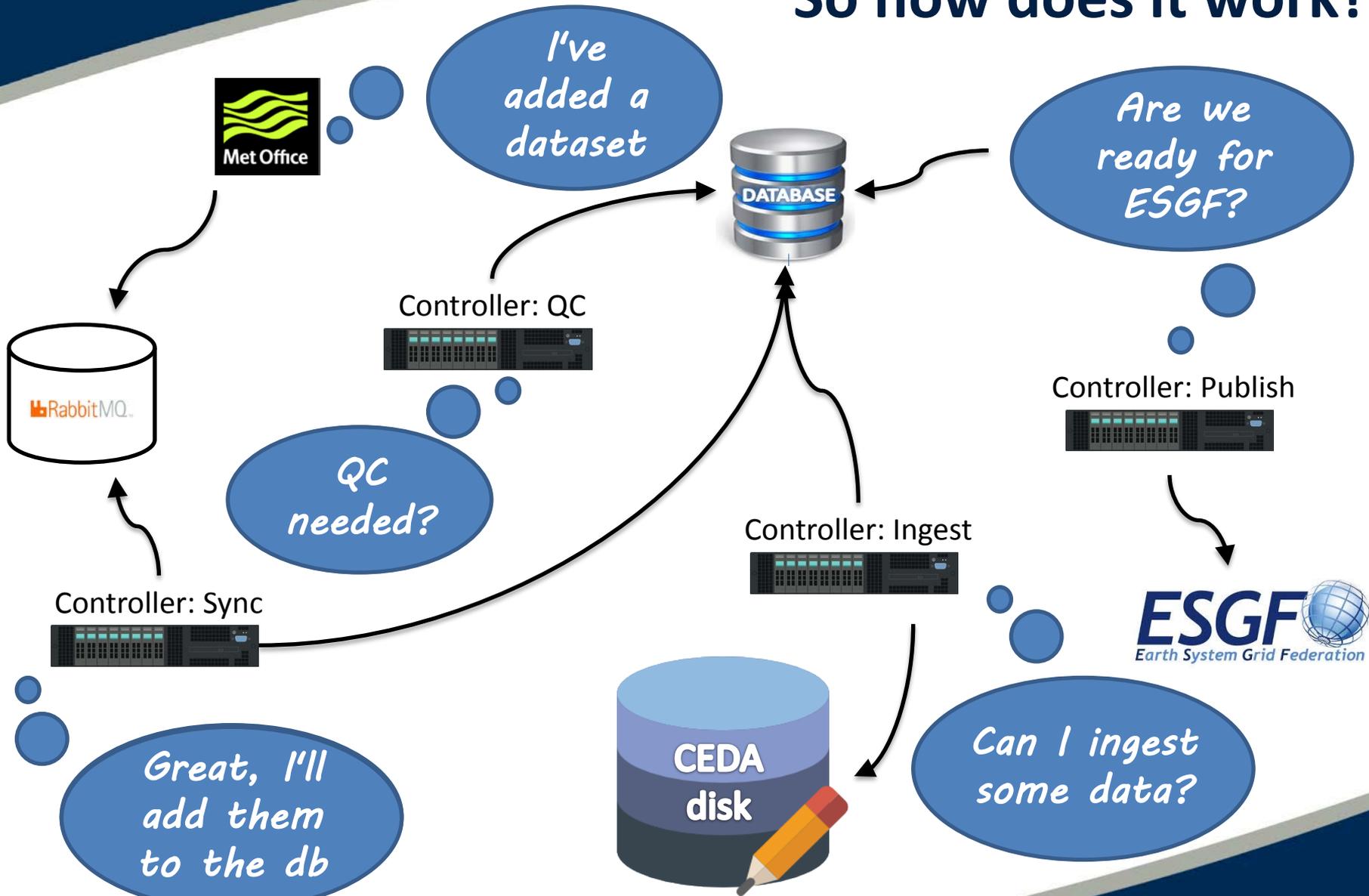
The Data Model is built in Django...each type of box is a relational DB table.



"DO" and "UNDO" actions

Action Type	QC	Ingest	Publish (TDS)
DO	Run QC	Create directories; move files	Publish to THREDDS
UNDO	-	Move files back to cache; remove directories	Unpublish from THREDDS

So how does it work?





Key components



Controller: QC



Controller: Publish



Controller: Ingest

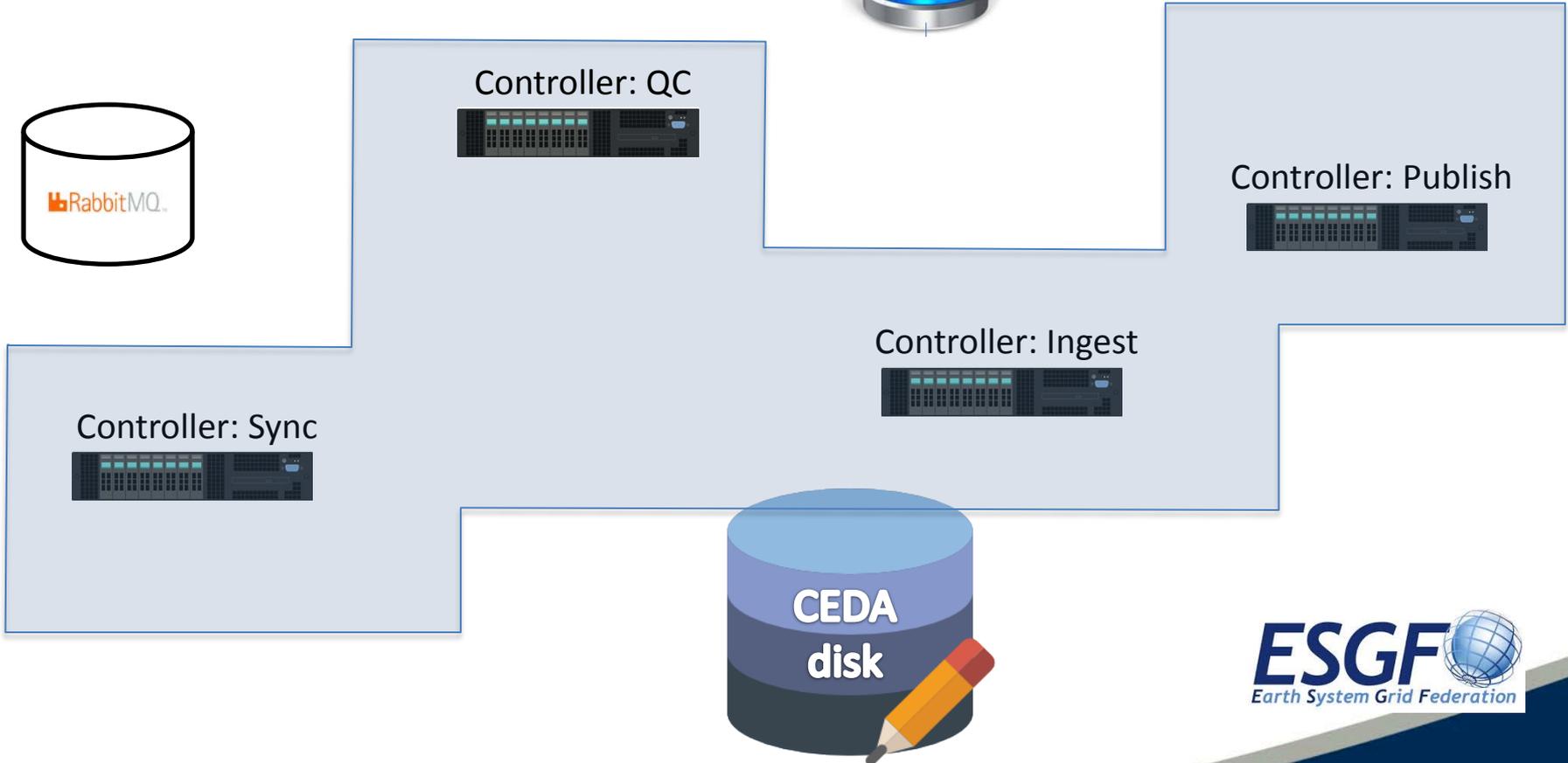


Controller: Sync





A Process Chain





I've added a dataset



Controller: QC



Controller: Publish



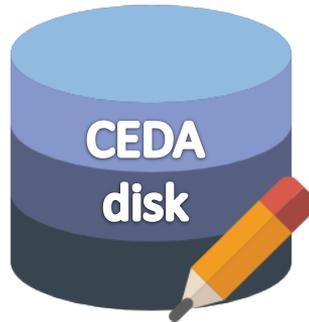
Controller: Ingest



Controller: Sync



Is there any work for me?





I've added a dataset



DATABASE

Controller: QC



Controller: Publish



Controller: Ingest



RabbitMQ

Controller: Sync



Great, I'll add them to the db



CEDA disk



Controller: QC



Controller: Publish

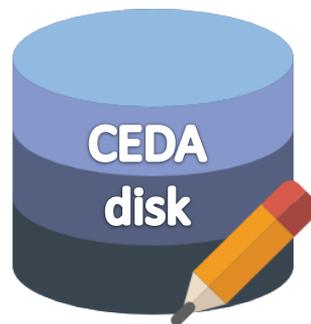
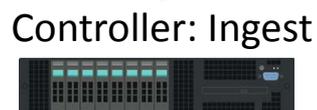
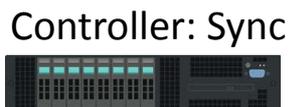


Controller: Ingest



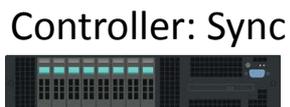
Controller: Sync





Can I ingest some data?





Controller: QC

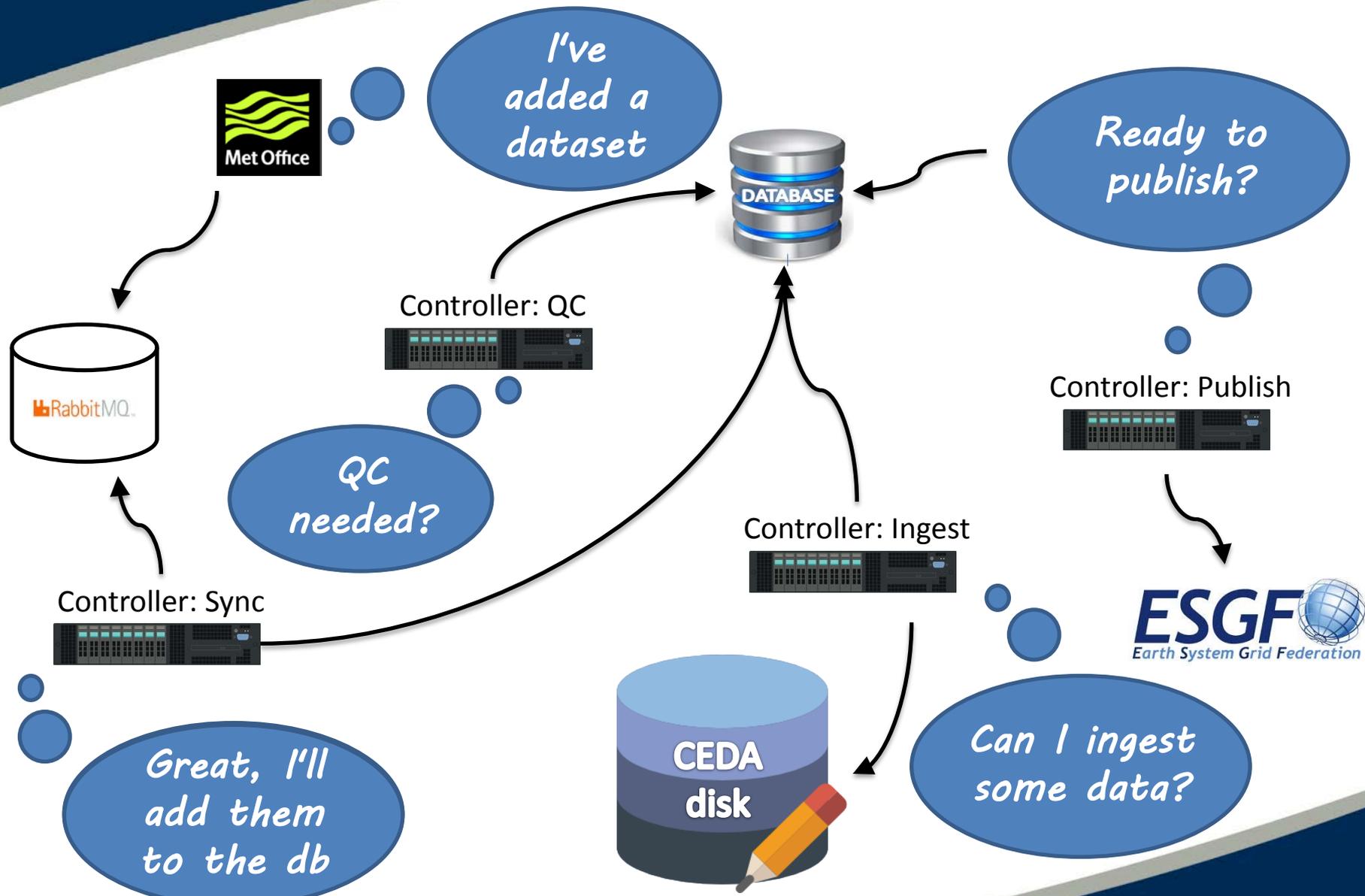


Controller: Publish



Controller: Ingest





Example Process Chain: CMIP6

#	Process Stage	<user>@<host>
1	Message Queue	cedauser@mass-cli1
2	QC	badc@ingest1
3	Ingestion	controller: badc@ingest1 workers: badc@ingest_cluster
4	Mapfile generation	badc@ingest1
5	CEDA metadata records	badc@ingest1
6	ES-DOC record generation	badc@ingest1
7	ESGF publication: DB	root@esgf-data1
8	ESGF publication: TDS (without r	
9	ESGF publication: TDS main catal	
10	ESGF publication: index	
11	Consume RabbitMQ Message	

Many Process Stages are re-usable. The Controller can be re-deployed in a Process Stage of multiple Process Chains (for different projects).



Technology choices



PostgreSQL





Django front-end

CEDA Dataset Pipeline App

Home Views Admin

The CEDA Dataset Pipeline App

Many datasets have an ingest "pipeline" that requires multiple processing stages that run on various servers under different user IDs. This app provides a front-end to manage and monitor the processes as each dataset travels through the system.

The first project to test this system is "CMIP6-MOHC" (Met Office Hadley Centre climate simulations for the 6th Coupled Model Intercomparison Project). Select listings from the Views menu above to monitor existing datasets being processed. If you log in you will also be able to pause or resume the system and run other management functions.

Welcome to our superb app, it will revolutionise your life. Try some links:

- [View Datasets](#)
- [View Files](#)
- [View Chains](#)
- [View Events](#)

Events view - allows real-time monitoring

CEDA Dataset Pipeline App

Home

View

CEDA Dataset Pipeline App

Home

Views ▾

Sign in

Ad

Events

An Event is logged each time a Controller has completed (i.e. succeeded or failed to process) a [Dataset](#). This page allows you to view Events in the system based on a set of search criteria.

Show Search Filters

Dataset:

Apply Filters

Time	Dataset	Process Stage	Message	Action Type	Succeeded?	Withdrawn?
Aug. 16, 2016, 10:55 a.m.	cmip5.output1.MOHC.HadGEM2-ES.rcp45.day.atmos.cfDay.r1i1p1.v20120114	Publish		do	SUCCEEDED	False
Aug. 16, 2016, 10:55 a.m.	cmip5.output1.MOHC.HadGEM2-ES.rcp45.day.atmos.cfDay.r1i1p1.v20120114	Ingest		do	SUCCEEDED	False
Aug. 16, 2016, 10:55 a.m.	cmip5.output1.MOHC.HadGEM2-ES.rcp45.day.atmos.cfDay.r1i1p1.v20120114	QC		do	SUCCEEDED	False



Special action: "withdraw"

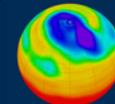
If the MOHC spots a problem with a dataset they may change the state (via a RabbitMQ message) from "available" to "withdrawn".

This triggers action at CEDA:

1. If already ingested/published:
 - hide?/unpublish
2. If not ingested/published:
 - **DO NOT** ingest/publish.
 - Acknowledge/consume the message with "available" state.



Integration issues





Integration with CIM2

- We intend to adopt the **cdf2cim** tool in the publication workflow to extract *Simulation* records.
- These will be automatically generated and pushed to the ES-Doc server.

Will CREPP handle replication?

NO:

- Replication nodes are being developed to work with Synda (using GridFTP where possible).
- We expect a different set of recipes/rules to be managing replication.

(or) YES:

- Replication is a set of tasks on different servers that (can) use the *ESGF Dataset* as their unit of granularity.
- The publish/unpublish components *could* be managed using CREPP.

Need to understand more about Synda post-processing workflows before we decide on this.



Recovery response

1. The (only) database goes offline:
 - All Controllers *wait...fail...undo...stop*.
 - Manual recovery to backup db.
 - Re-start all Controllers.
2. Individual Controller (or server it runs on) fails:
 - Some datasets are in "claimed" state
 - Remove claims
 - Let them be re-run by new instance of the Controller
3. Urgent software upgrade required
 - Switch on Global Pause
 - Current tasks will run to completion; then all will pause; it is safe to stop all and roll out new software before re-starting



CREPP Status

- Code base developed
- Being tested on operational platform - with first MOHC test simulations
- Individual Controllers being written to handle specific process stages
- Needs to be ready soon!



Further information

- Centre for Environmental Data Analysis
 - <http://www.ceda.ac.uk>
 - support@ceda.ac.uk
- CREPP code (currently internal to CEDA):
 - <https://github.com/cedadev/crepp>
- Met Office pipeline (climate-dds) code (internal):
 - <https://code.metoffice.gov.uk/trac/cdds/>

If you are interested in finding out more please contact me on:

ag.stephens@stfc.ac.uk



Science & Technology Facilities Council
Rutherford Appleton Laboratory

CREPP Terminology

Term	Meaning	Comments
Queue	RabbitMQ instance of a queue of Dataset held in MASS with a Met Office status associated with them.	
Message	A message in the Queue that specifies the Met Office status related to a single Dataset.	Status can be: available withdrawn embargo superseded. We only respond to “withdrawn” “superseded” or “available”. We treat “superseded” as the same as “available”. Each message will trigger stage 1 of the processing chain for Met Office data.

CREPP Terminology

Term	Meaning	Comments
Dataset	A set of files that have a complete ESGF DRS description including the version component.	Note that this term has a specific meaning throughout the system.
Controller	A process running on a node that communicates with the DB and manages Workers running locally or remotely.	A Controller can run as a daemon process, might be invoked by cron or other means. The key aspects are that it routinely polls the DB for its next set of tasks and manages Workers to perform Tasks.

CREPP Terminology

Term	Meaning	Comments
Worker	A process running on a node that performs a distinct task and then terminates.	Workers may be invoked locally or via a scheduler (e.g. on the “Ingest” cluster).
Task	A single complete Action undertaken by a Worker when operating on a single Dataset at a given Process Stage.	Each completed Task will result in the DB being updated. Where successful, this will update the status to trigger the next Controller. An entry will also be made in the Event Log.

CREPP Terminology

Term	Meaning	Comments
Event Log	A table in the DB that records ALL outcomes from Tasks per Dataset.	
Process Chain	An ordered set of Process Stages.	Multiple ESGF “projects” may use the same chain and a single project might use multiple chains for different data providers.
Process Stage	A component of the processing chain that is managed by a Controller.	E.g. “Run QC”, “Create mapfiles”, etc.
Action Type	The attribute of the Task that specifies the type of behaviour.	Can be: “Do”, “Undo”