MDD

MOGREPS-G DATA DELIVERY

A Presentation for AWS Big Data Team

INTRO to MDD

4 QUESTIONS

Why Are We Here?

We're here to talk about a data delivery problem and a proposed solution I have to solve it.

Who Am I?

I'm Andrew. I've been solving these types of problems for about 12 years.

How Long Is This?

About 30 minutes + 15 for questions and answers.

What Will It Cover?

Big picture solution + primary concerns.

Detailed Architecture. Summary.

email: agmartin@gmail.com
source: github.com/amartin/mogreps/

BIG PICTURE

GOALS PROPOSED

Deliver the Data

Create a pipeline to convert the file data to queryable form

High Availability

Data should always be available to query

Fast Response

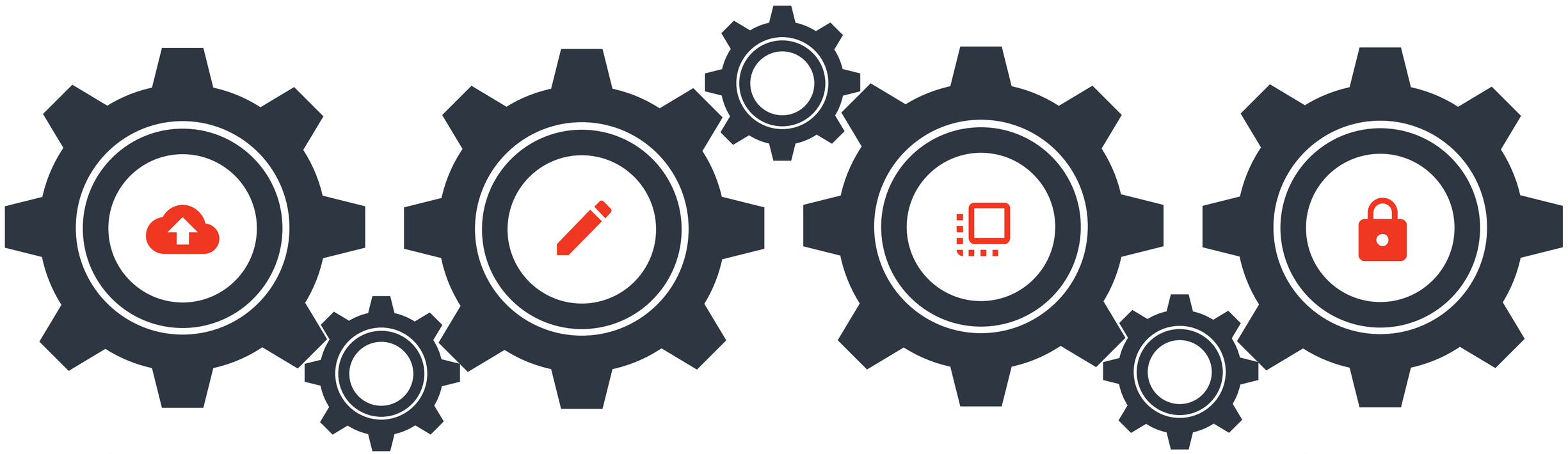
Queries should be fast

Timely

New data should be available for query as soon as it arrives

email: agmartin@gmail.com
source: github.com/amartin/mogreps/

BIG PICTURE



S3 Delivery

AWS S3

ETL Process

AWS Lambda

Data Layer

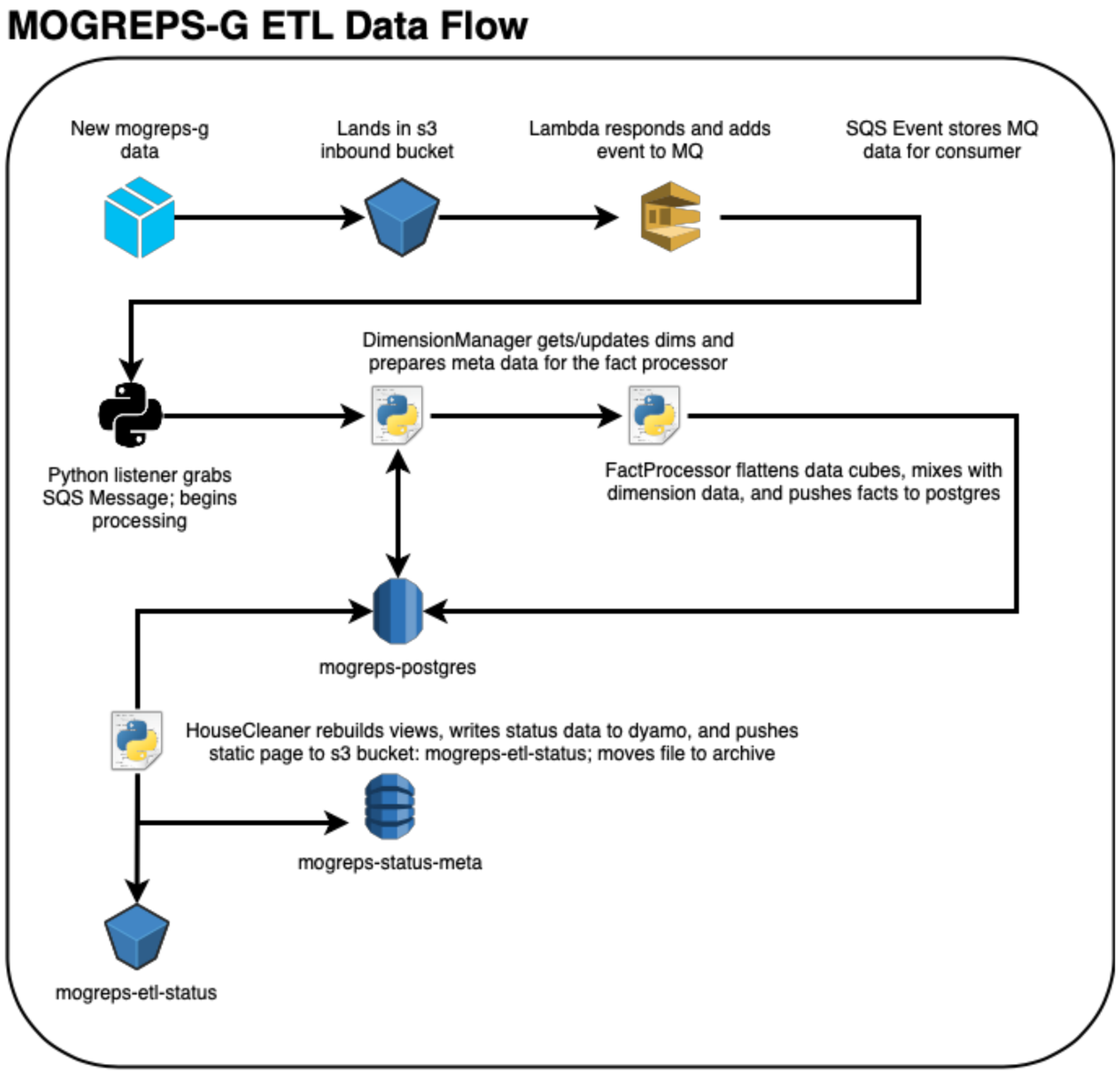
AWS RDS Postgres

Query Access

AWS API Gateway

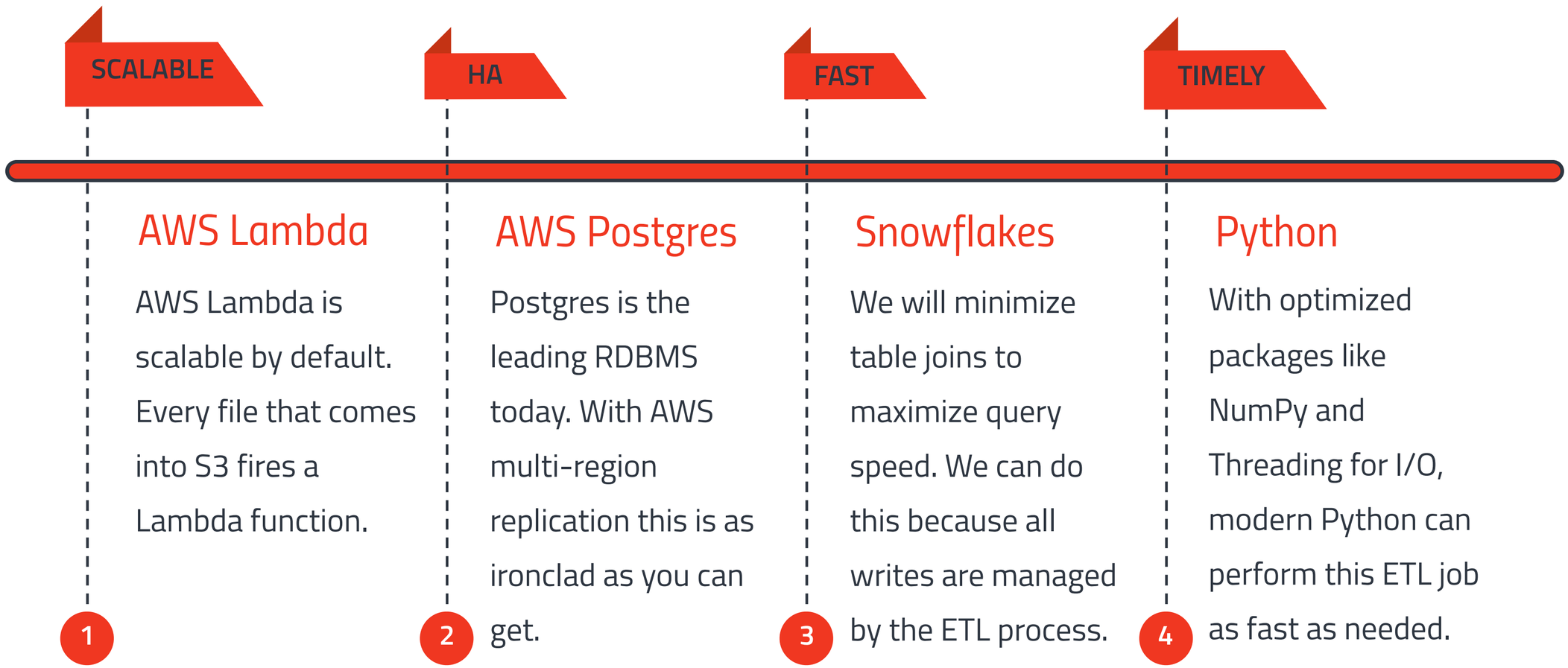
email: agmartin@gmail.com
source: github.com/amartin/mogreps/

BIG PICTURE



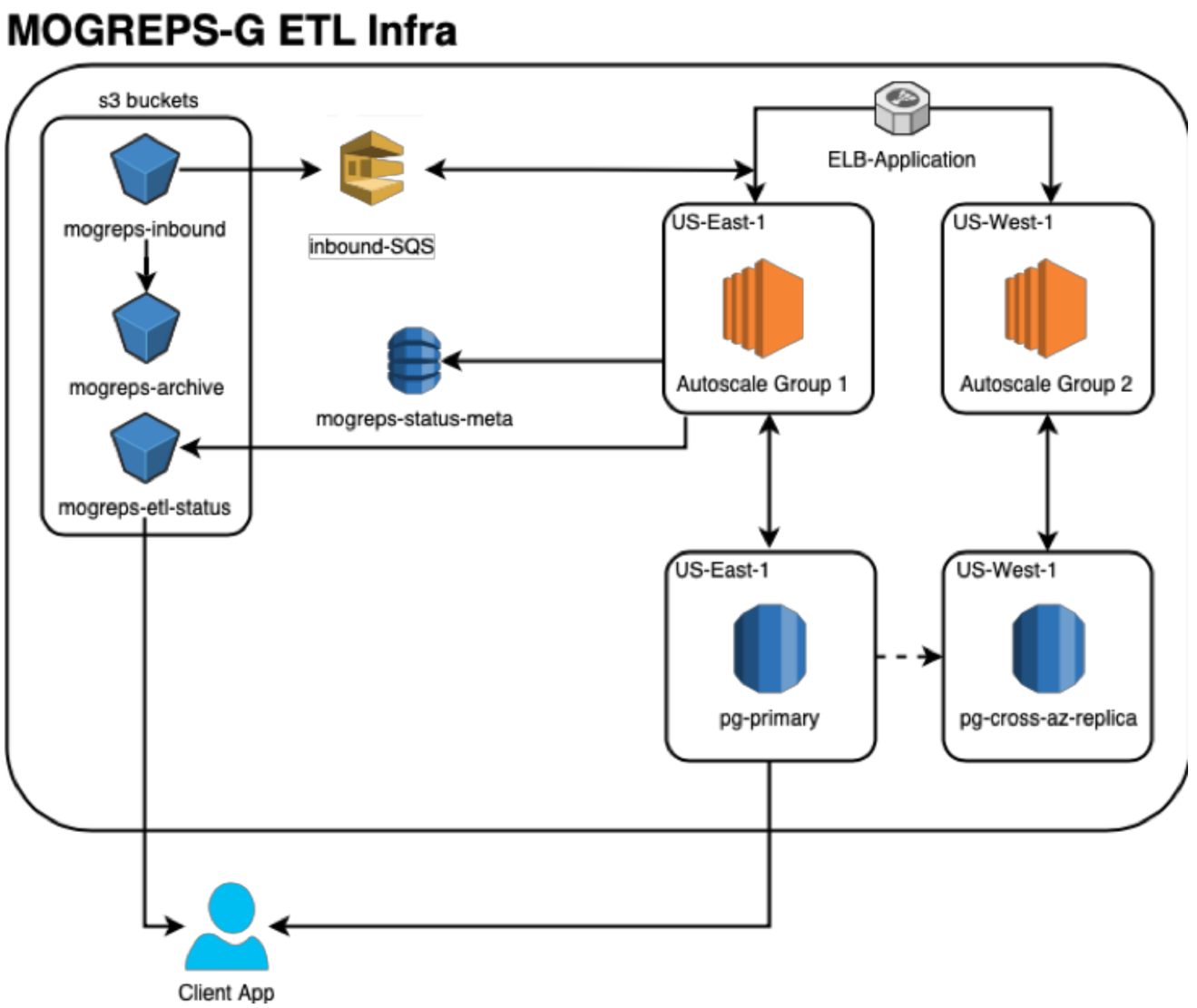
email: agmartin@gmail.com
source: github.com/amartin/mogreps/

BIG PICTURE



email: agmartin@gmail.com
source: github.com/amartin/mogreps/

BIG PICTURE



email: agmartin@gmail.com
source: github.com/amartin/mogreps/

DATA ARCHITECTURE

THE DATA

Predictive Weather

Comes from the UK Met Office. Similar to models we see in hurricane forecasting

Multiple Versions

Models are initialized with real conditions and also perturbed. Models should be combined

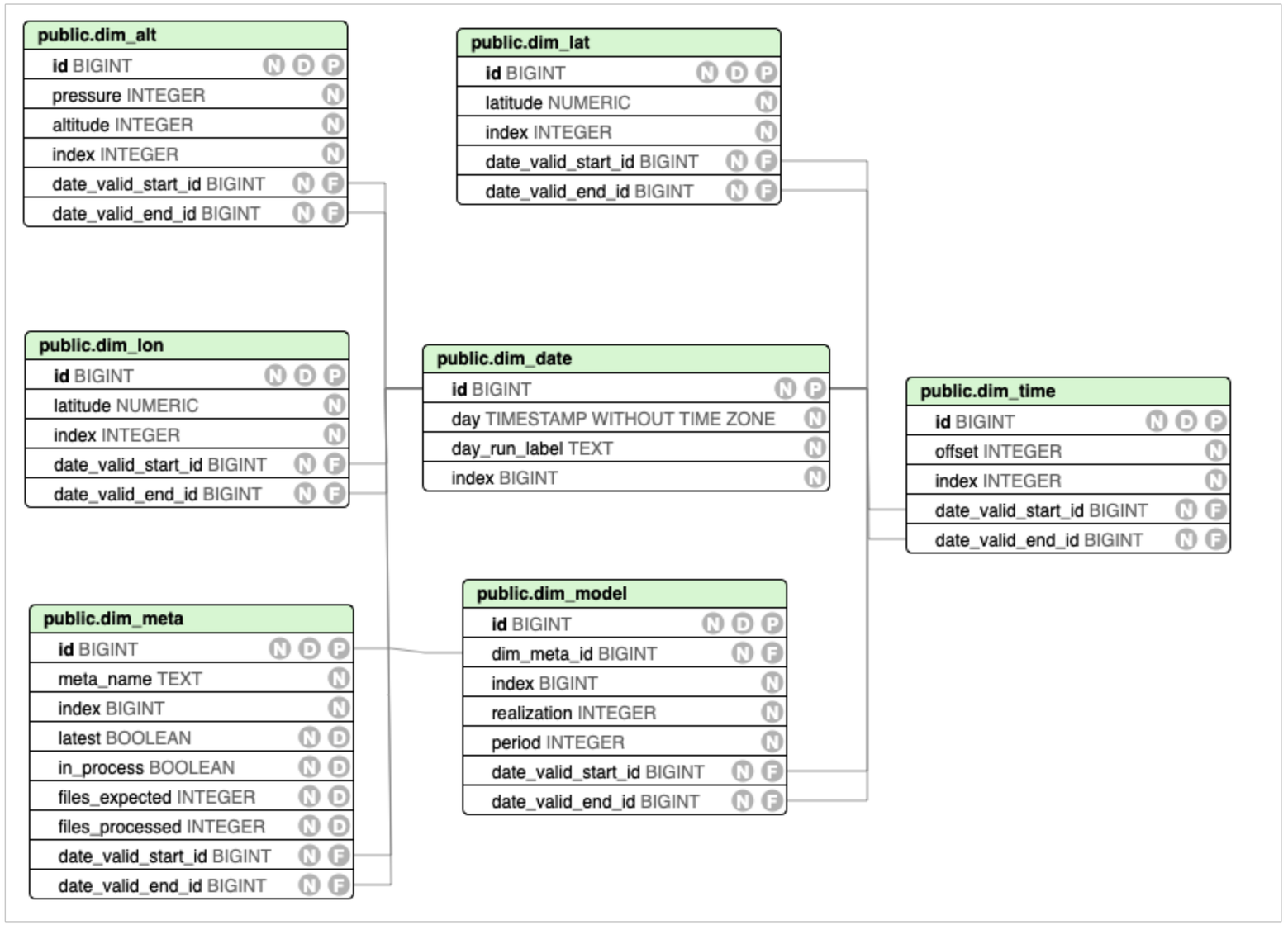
Files up to 7 days

One file per model per perturbation per 3 hour window

Total Dataset = Big

Individually, these files are easy to handle. Total model = ~650GB/day

email: agmartin@gmail.com
source: github.com/amartin/mogreps/



email: agmartin@gmail.com
source: github.com/amartin/mogreps/

TABLE DETAILS

Meta Dimension

Allows ETL application to reflect on itself and be aware of its status

Version Dimension

Allows ETL app to have a granular history of how data was produced

Date Dimension

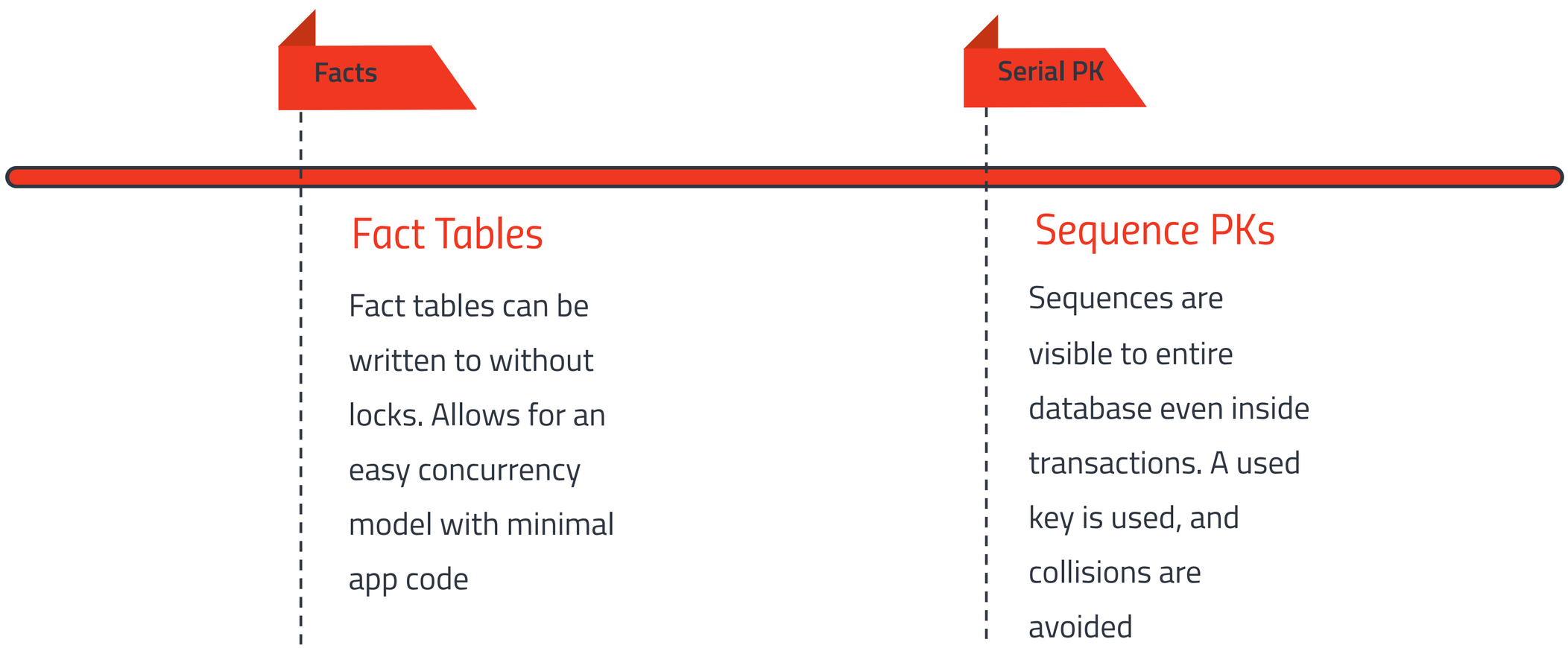
Allows ETL app to gradually change over time while preserving past truth

Geo. Dimension

Allows for fast lookups with built-in GIS functions

email: agmartin@gmail.com
source: github.com/amartin/mogreps/

BULK INSERTS



Fact Tables

Fact tables can be written to without locks. Allows for an easy concurrency model with minimal app code

Sequence PKs

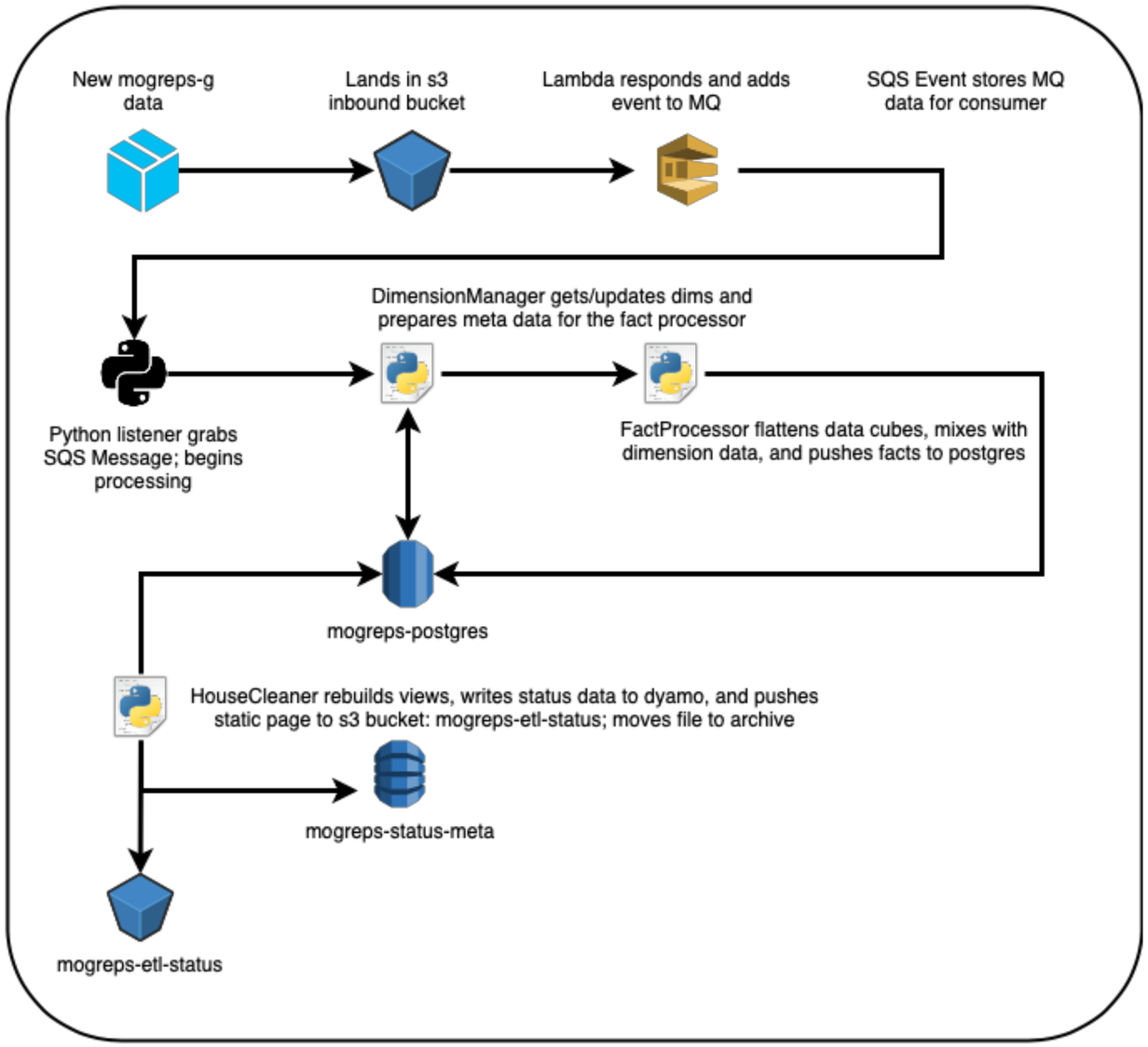
Sequences are visible to entire database even inside transactions. A used key is used, and collisions are avoided

email: agmartin@gmail.com
source: github.com/amartin/mogreps/

PROCESS

DETAILS

MOGREPS-G ETL Data Flow



email: agmartin@gmail.com
source: github.com/amartin/mogreps/

PROCESS DETAILS

Data Inbound

netCDF file lands in S3 bucket that our Lambda is listening to. Could be concurrent

Data Processing

Lambda runs a data cleaning process, written in Python, that outputs a CSV to memory object

Data Loading

Python runs a CSV load operation on the memory object to the Postgres tables

Data Delivery

Data becomes accessible to queries via API Gateway

email: agmartin@gmail.com
source: github.com/amartin/mogreps/

LAMBDA DETAILS

Dimension Manager

Updates meta tables with information about what's happening now

Version Manager

Handles versioning information for current file in process

Fact Manager

Actually does the work of unrolling the data cubes, formatting data, and loading it to PG

House Cleaner

Handles errors, requeues files, checks for outstanding issues

email: agmartin@gmail.com
source: github.com/amartin/mogreps/

FACT MANAGER

Naive Version

Operates at the highest level of abstraction. Iris and SQLAlchemy packages. Too slow.

Improved Version

Uses Numpy directly, conn.cursor, CSV to /tmp file. Still too slow.

Viable Solution

Writes to CSV memory, bulk insert to PG. Good enough for 1 variable.

Proof of Concept

Threaded model to handle multiple variables inside a single lambda time constraint.

email: agmartin@gmail.com
source: github.com/amartin/mogreps/

WRAP UP

GOALS ACHIEVED

Deliver the Data

AWS API Gateway delivers data on-demand

High Availability

Multi-AZ read replicas deliver data no matter what, even if stale

Fast Response

Data Architecture guarantees read-optimization

Timely

AWS Lambda ETL processor moves data from file to RDS at guaranteed rates

email: agmartin@gmail.com
source: github.com/amartin/mogreps/

THE

END