

Monitoring Ozone Levels with Postgresql

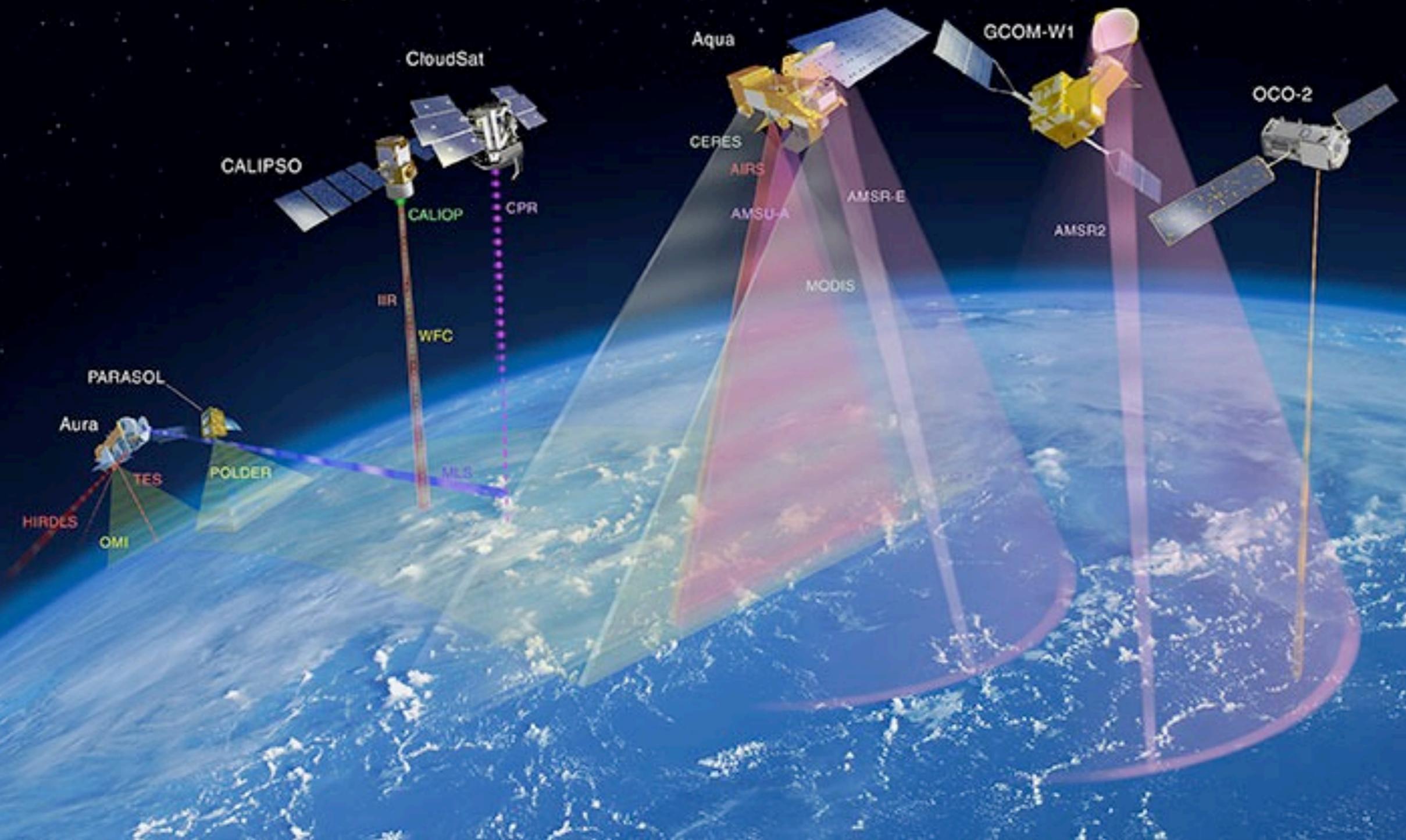
Alex Lai, Marty Brandon
May 2012

Overview

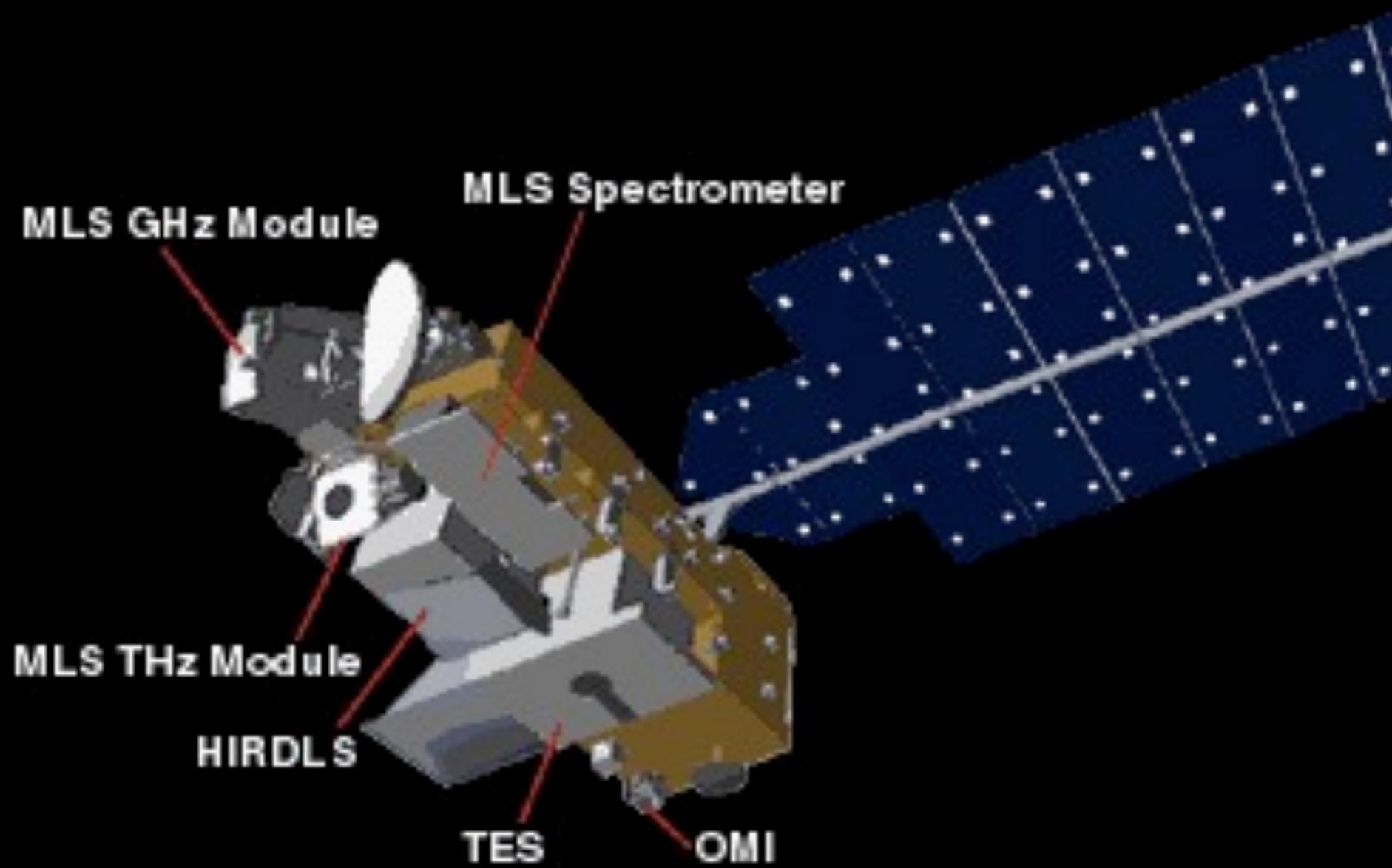
- Background (Marty)
- Ingestion & Processing (Alex)
- Challenges (Alex)
- Ongoing Development (Marty)

Background

A-Train



Aura



Data Downloading



Earth Science Data Types



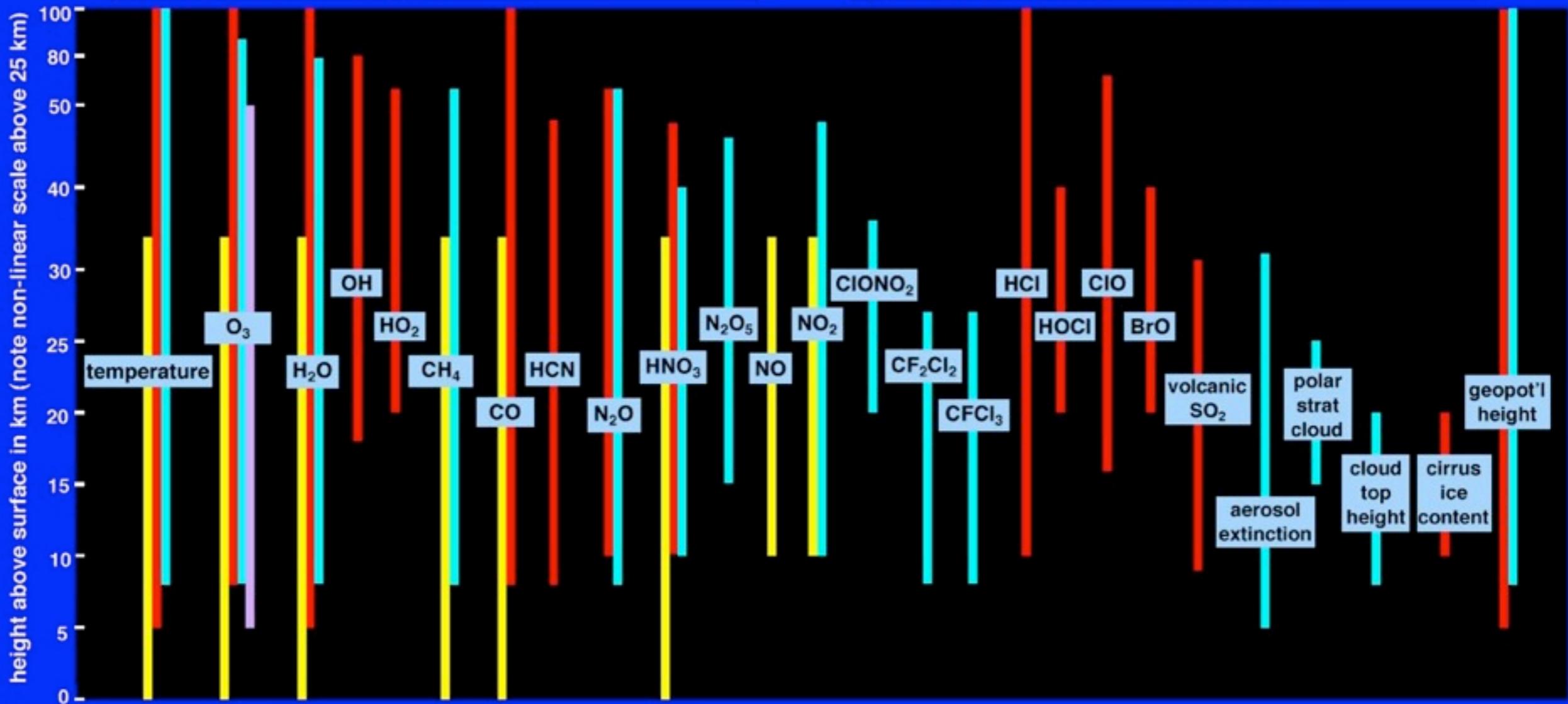
- Interface between the raw science data and the data management system
- > 900

EOS Aura Atmospheric Profile Measurements

OMI also measures UVB flux, cloud top/cover, and column abundances of O₃, NO₂, BrO, aerosol and volcanic SO₂
 TES also measures several additional 'special products' such as ClONO₂, CF₂Cl₂, CFCI₃, N₂O and volcanic SO₂

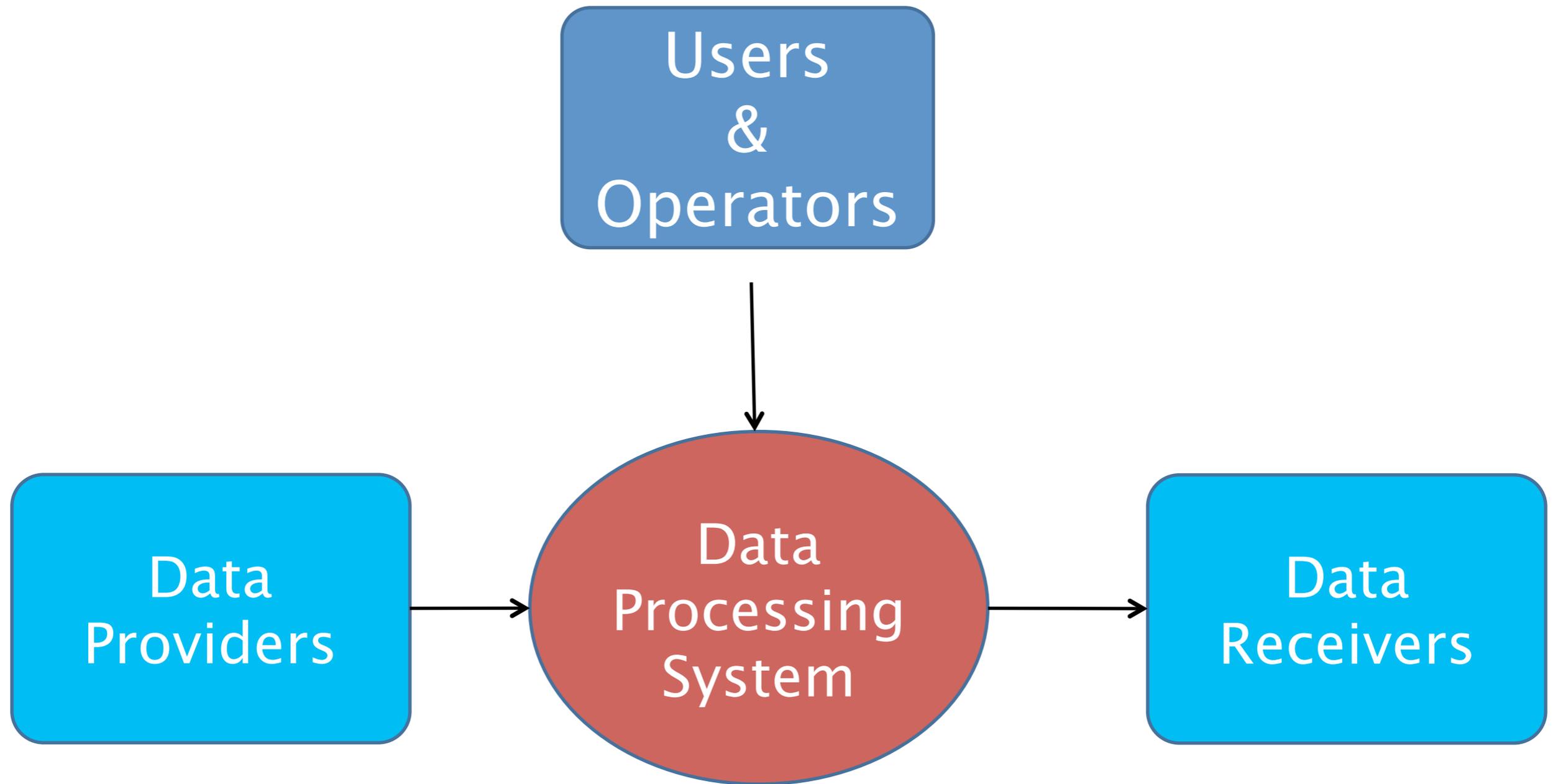
HIRDLS: High Resolution Dynamics Limb Sounder
 OMI: Ozone Monitoring Instrument

MLS: Microwave Limb Sounder
 TES: Tropospheric Emission Spectrometer

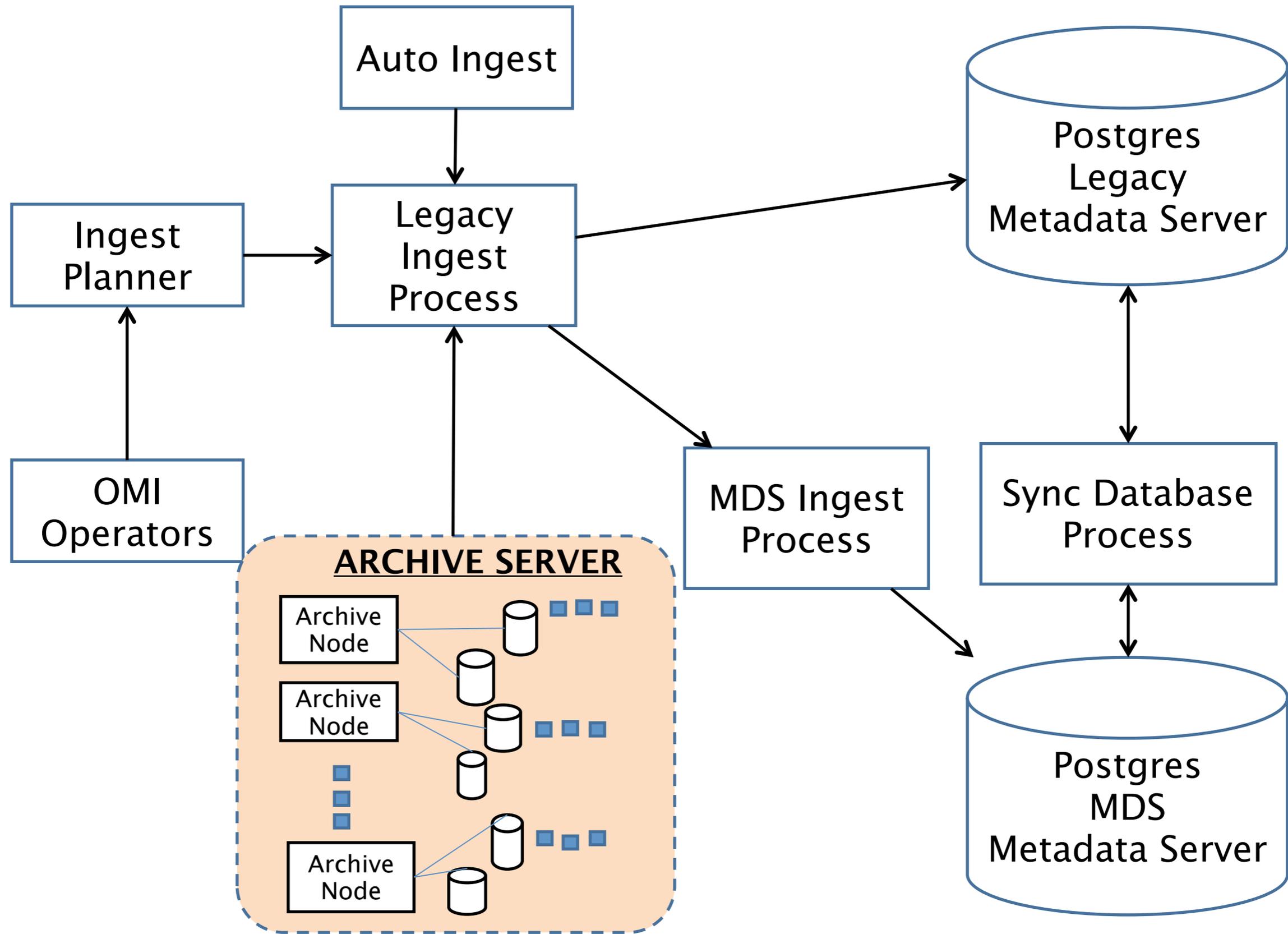


Ingestion & Processing

Science Data Processing System

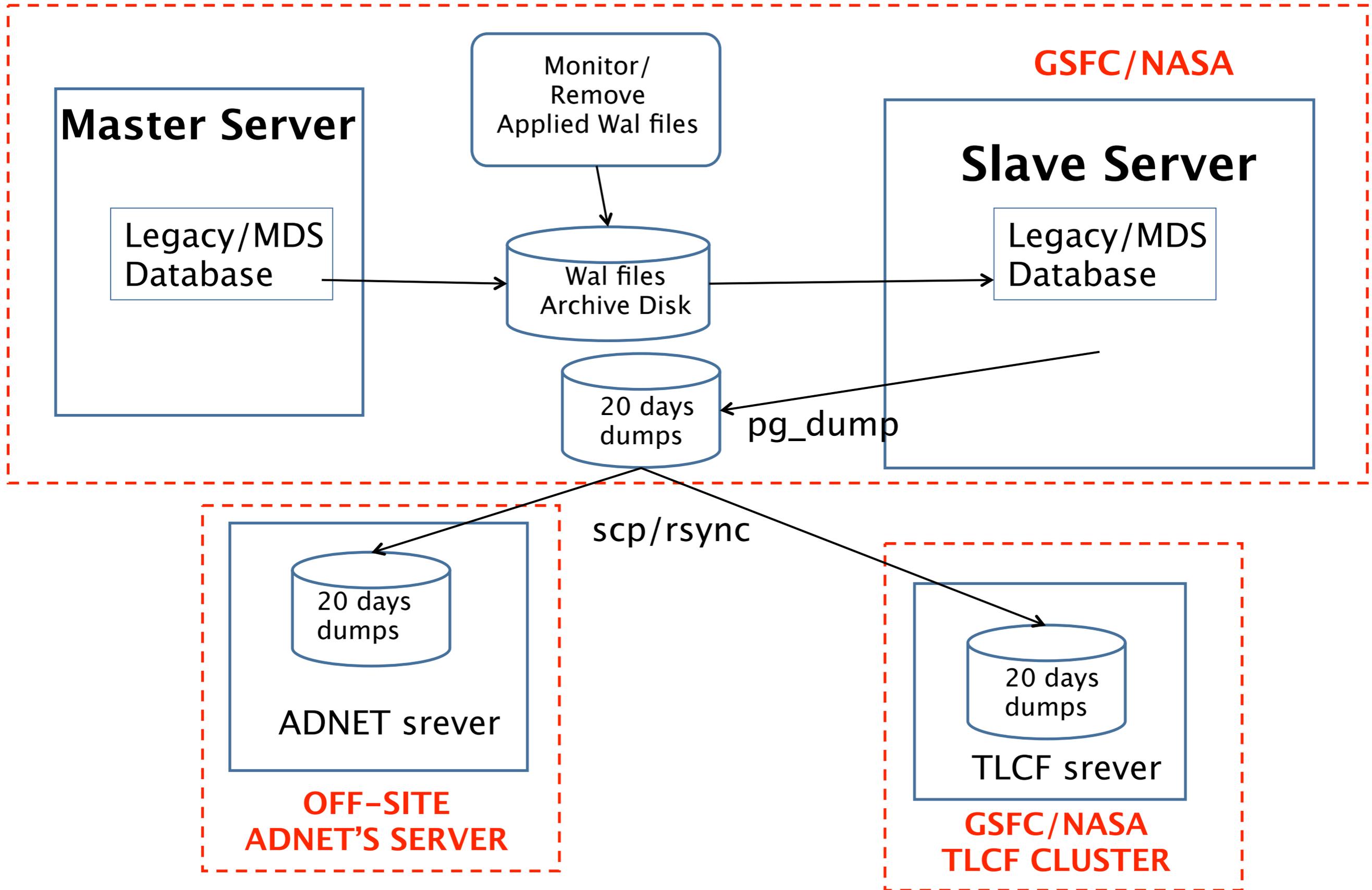


Ingestion Processing



Challenges

Backup and Asynchronous Replication



Monitor WAL is the key to ensure the replication is in good status?

- The cause of WAL not properly apply to Slave
 - Network unavailable
 - WAL files were deleted
 - While COPY took too long to finish that cause the apply WAL file in wait status
(set `Max_standby_streaming_delay = -1`)
 - Slave hangs for any reasons

Monitor WAL is the key to ensure the replication is in good status?

- Slave unable to connect and copy wal file from replication archive disk
- ps aux shows waiting status in slave
- Pg log also show waiting no recovery state for a long period of time

Monitor WAL is the key to ensure the replication is in good status?

- Query using functions
pg_current_xlog_location() (master only)
Pg_last_xlog_receive_location(),
Pg_last_xlog_replay_location()
- When slave hangs for any reasons, an monitoring script will trigger an alert email to DBA

Monitor WAL is the key to ensure the replication is in good status?

- What and how to monitor the WAL files?
 - Monitor the disk space available for WAL archive
 - Monitor the last applied WAL file in Slave
 - Monitor number of purging old applied WAL file
 - Monitor any wait states
 - Monitor dead lock situation

Monitor WAL is the key to ensure the replication is in good status?

- Email Notification to DBA for
 - Disk storage down to warning level
 - The last applied WAL file not changed longer than the `statement_timeout` in SLAVE
 - Network unavailable between master and slave
 - Log the long queries that exceed a long time like 20 minutes

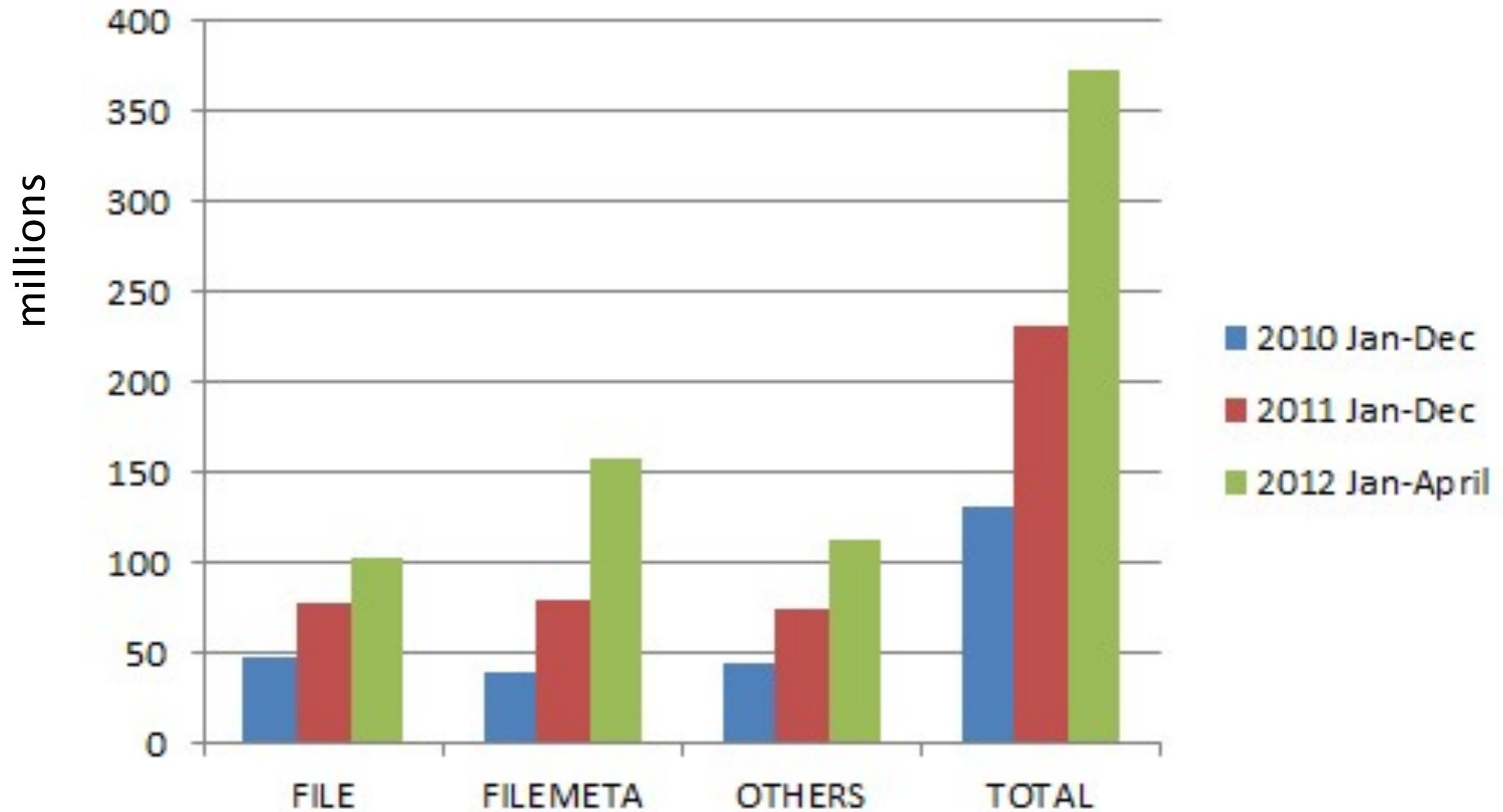
Monitor WAL is the key to ensure the replication is in good status?

- Too many processes in wait state
- When Database has slow performance for a period of time

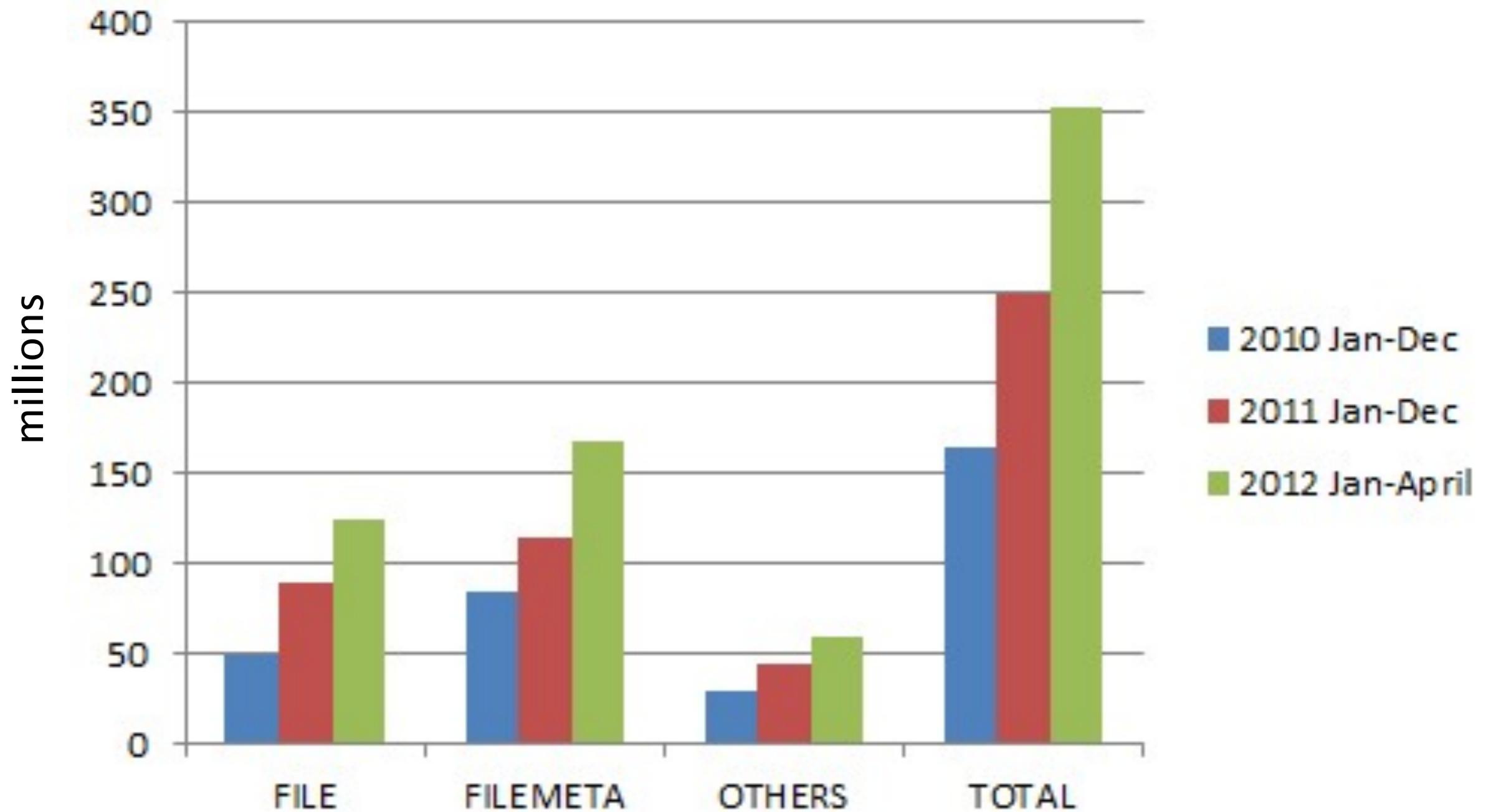
HOW much data do we have?

- Our current database implementation includes 10 clusters, each running Postgres 9.0.6 and divided into three production levels: development, testing, and operations.
- On average about 200 commits per second
- The largest table contains over 160 millions rows
- Total data increase from 40 – 60 % in the last three years.

Number of records (Legacy)



Number of records (MDS)



Synchronization between Legacy and MDS



MDS – Always has long delay
Also has to resolve record locking

Setting of Postgresql.conf

Variables	MASTER	SLAVE
wal_level (enable read-only queries on a standby server)	hot_standby	hot_standby
hot_standby	off (master MUST be off)	on (enable read-only queries on a standby server)
wal_keep_segment	32	32
wal_max_senders	5	5

Setting of Postgresql.conf

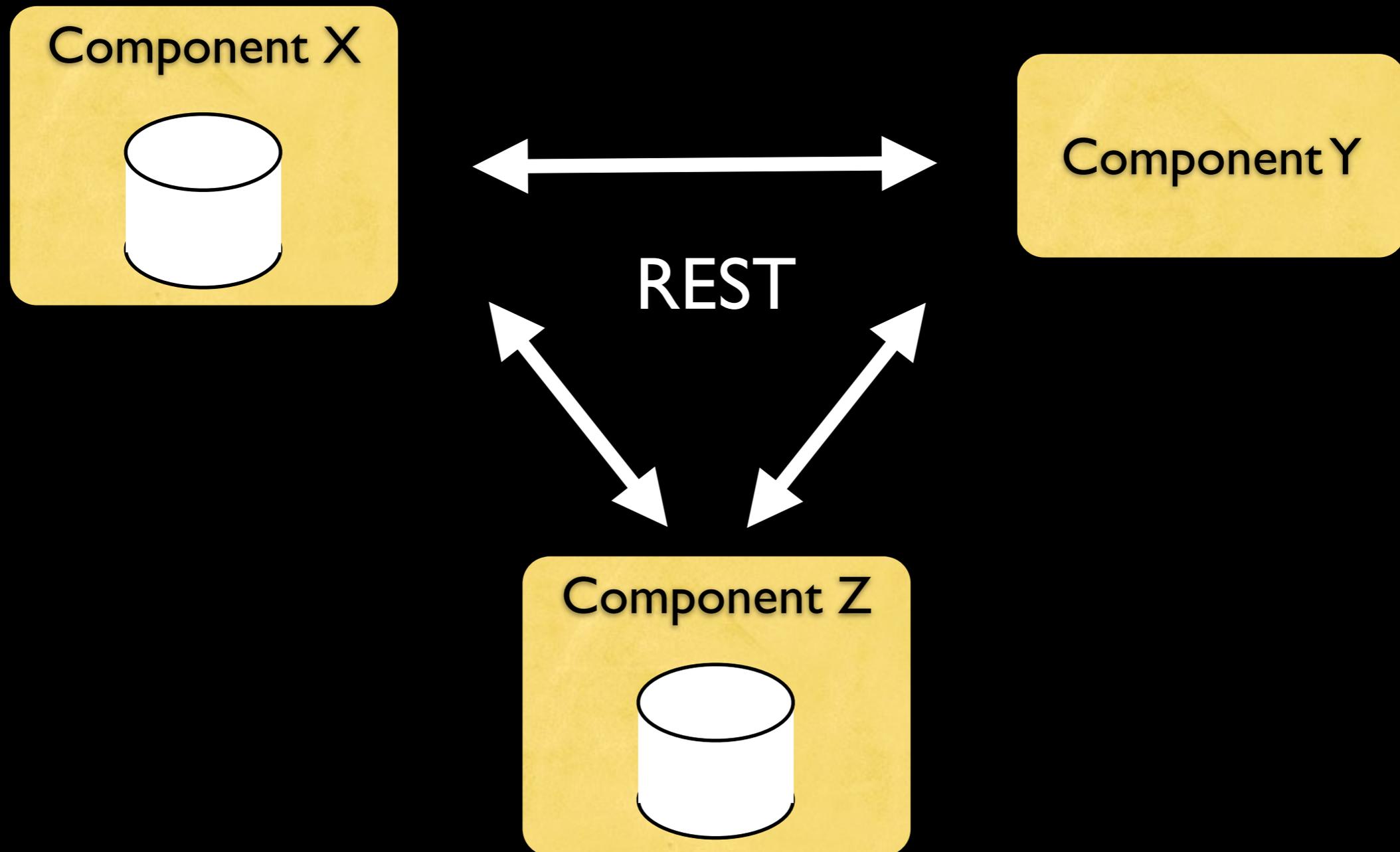
Variables	MASTER	SLAVE
archive_mode	on	off
Archive_timeout	30s	30s
archive_command	'cp %p /your_path/%f'	'cp %p /your_path/%f'
Max_standby_ archive_delay	300s	1200s
Max_standby_ streaming_delay	300s	-1 Wait for any queries before apply for the WAL file This avoid pg_dump COPY fail.

Setting of Recovery.conf on Slave

Variables	SLAVE
standby_mode	'on'
primary_conninfo	'host=ip_of_Master port=5432 user=user_has_all_access_to_all_database pw=user_password'
trigger_file	'/path/to/trigger'
restore_command	'cp /path/to/archive_files/%f "%p"'

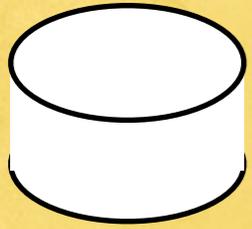
Ongoing Work

Decentralization

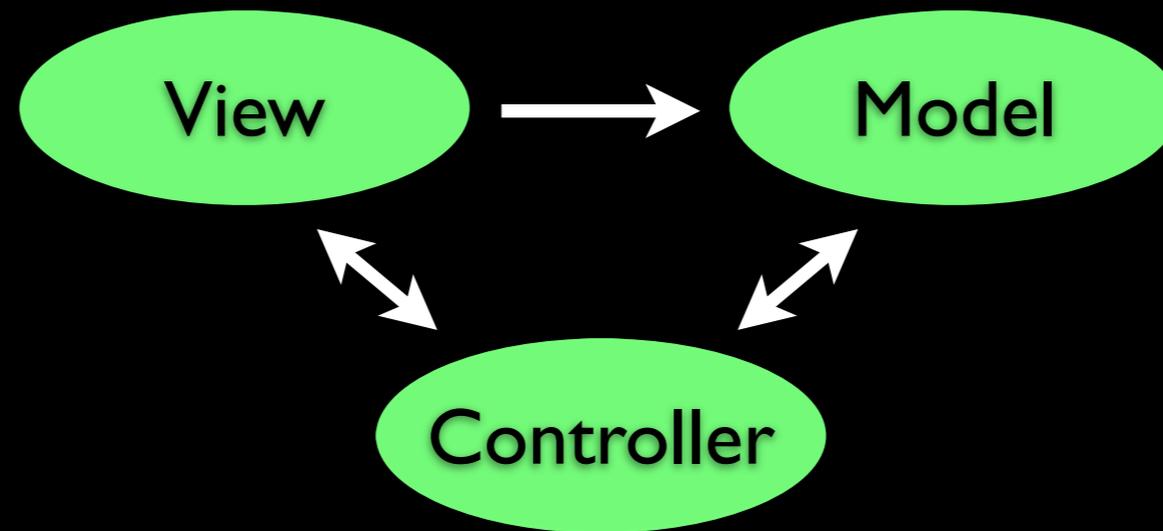


- Web Framework
- ORM
- Database Patches

Component X

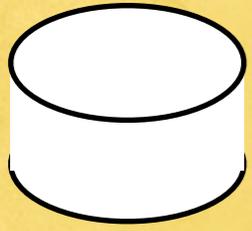


Web Framework



GET /person/*key)
POST /person
DELETE /person/*key)

Component X



ORM

Server Start:

- read config file
- register database
- generate CRUD methods

HTTP Request

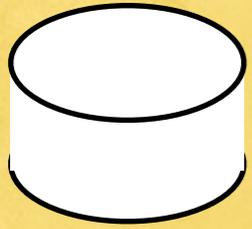


Object Method



Database

Component X



Database Patches

Git Repo -
Component X

code
schema.sql
patch1.sql



Instance -
Component X



patch2.sql

Component X

Database Patch Details

Git Repo

base.sql
patches_applied.txt
patch1.sql
patch2.sql
...

Instance

patches_applied



base.sql = schema dump
+
new patches

Perl Modules

- Mojolicious
- Rose::DB
- Module::Build

Monitoring

The screenshot shows a web browser window titled "Clusters". The address bar contains "127.0.0.1:30" and the search bar contains "mac os x screenshot". The page content is titled "Database Clusters 18:40:27 (checking every 5 minutes)". It displays nine database clusters as 3D cylinders arranged in two rows. The clusters are: acpsdb1 (yellow), acpsas1 (white), devarch11 (grey), testdb2 (grey), testmds1 (red), devdb2 (green) in the top row; testarch1 (white), devmds1 (grey), acpsdb2 (red) in the bottom row. A mouse cursor is visible over the bottom row. The browser's status bar at the bottom right shows "731x263" and system icons.

Cluster Name	Color
acpsdb1	Yellow
acpsas1	White
devarch11	Grey
testdb2	Grey
testmds1	Red
devdb2	Green
testarch1	White
devmds1	Grey
acpsdb2	Red

cluster

← → ☁ ☆ ▼ ↻ mac 🔍 🏠 🐛 🐼

BIG RAT Bigrat

 **18:34:48** (checking every 60 seconds)

devmds1

ssh -N -L3002:acpsdb1:5433 acpsdb1.omisips1.eosdis.nasa.gov
databases: omiops, metamine, policy
roles: omiops, omiopsdba

■ Connectivity

- ssh connection verified
- psql connection verified

■ Load

- CPU load is %10

■ Space

- 10 GB of 100 GB is unused

■ Replication

- master WAL file: AF4322BADD3
- slave WAL file: AF4322BADD2

✕ 400x51

Questions?