Towards a scalable semi-structured data platform for COPE*

Drasko Vasiljejic,

Lars Isaksen, Tomas Kral, Piotr Kuchta, Joaquín Muñoz Sabater

and

Anne Fouilloux

*Continuous Observation Processing Environment

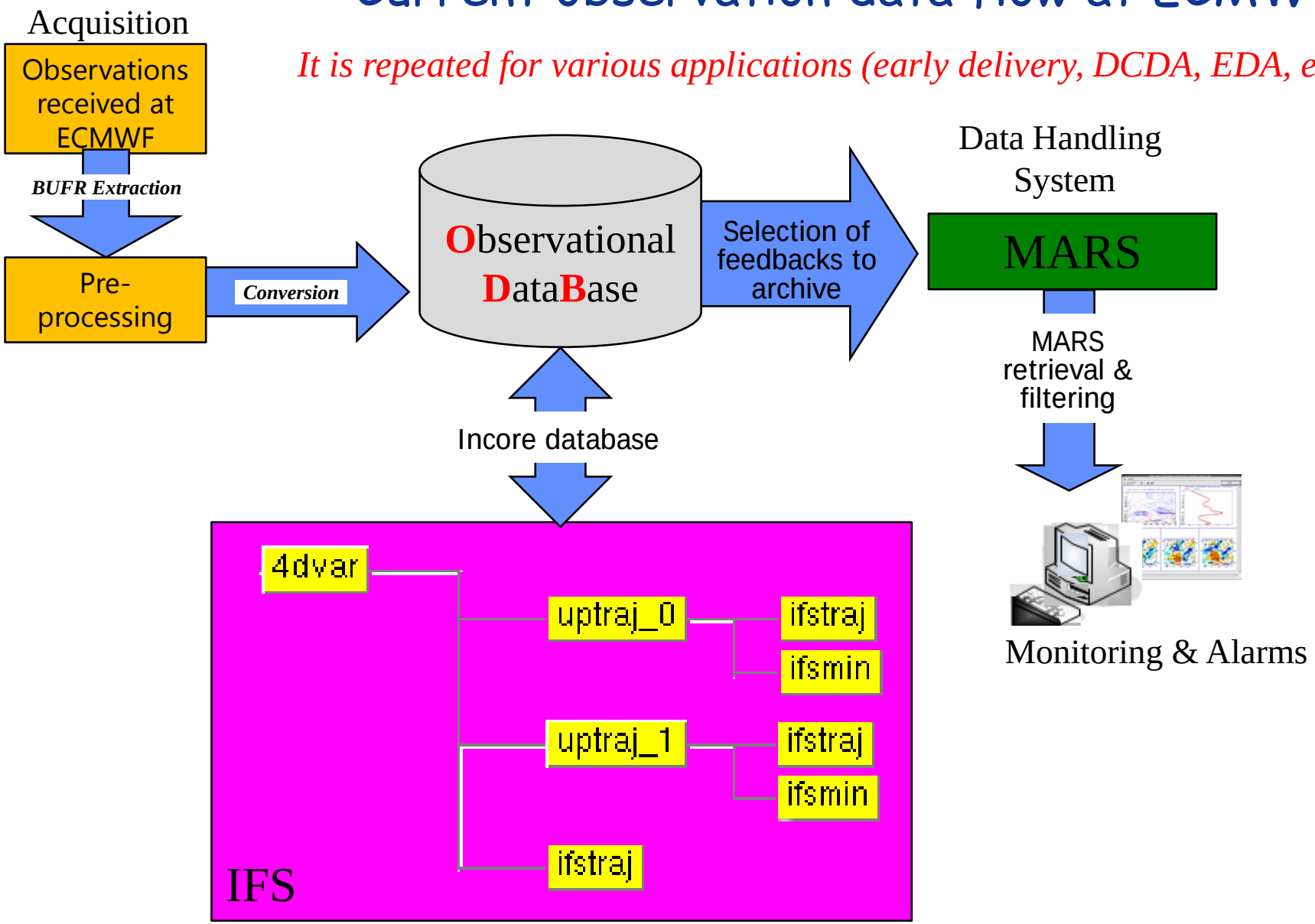
ECMWF

Outline

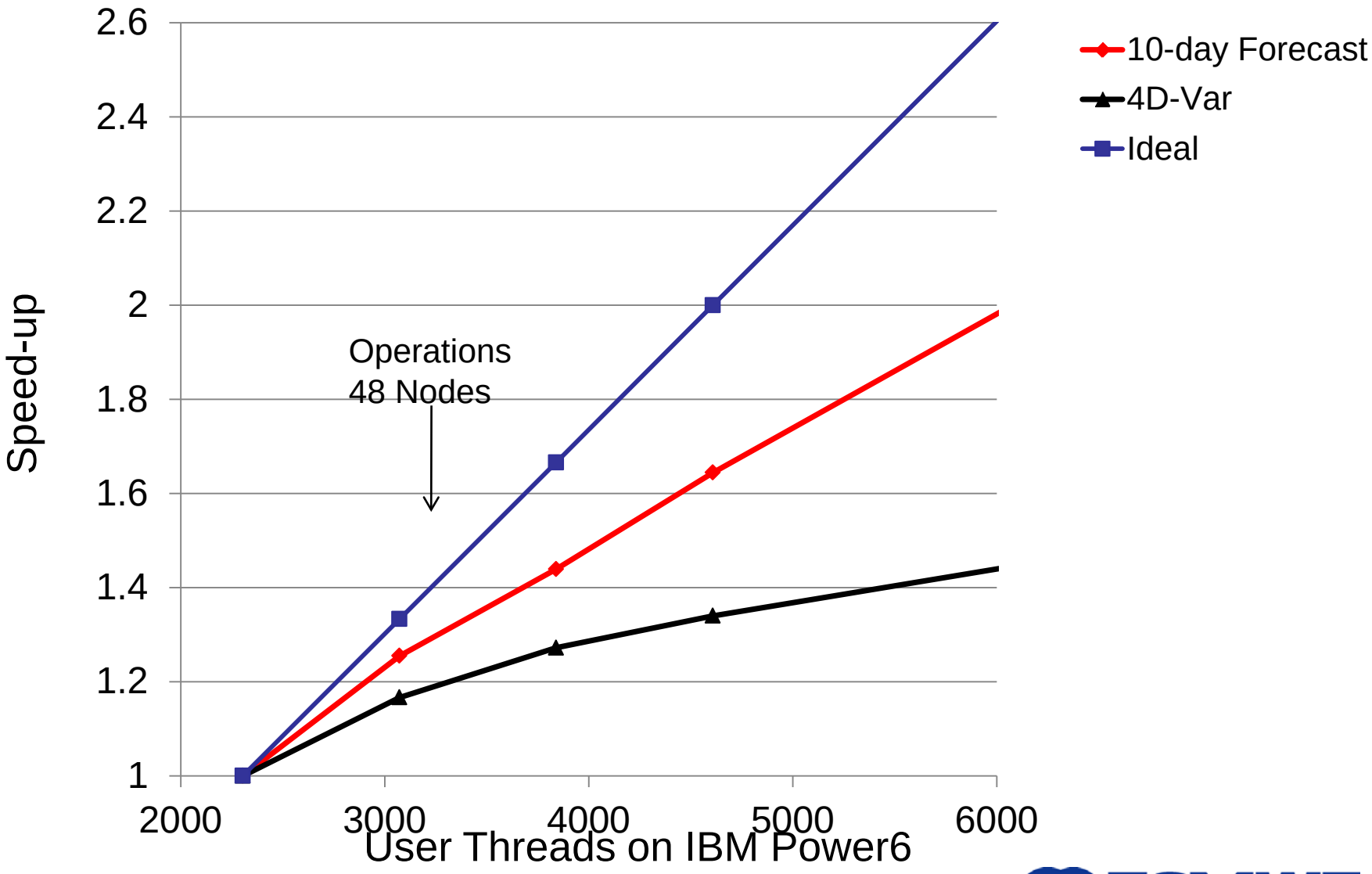
- Current observation data flow at ECMWF
- Bottleneck in our current system
- How to tackle these issues?
- COPE: Continuous Observation Processing Environment
- A semi-structured data platform for COPE: why and how?
- Preliminary results…

Current observation data flow at ECMWF

It is repeated for various applications (early delivery, DCDA, EDA, etc.)

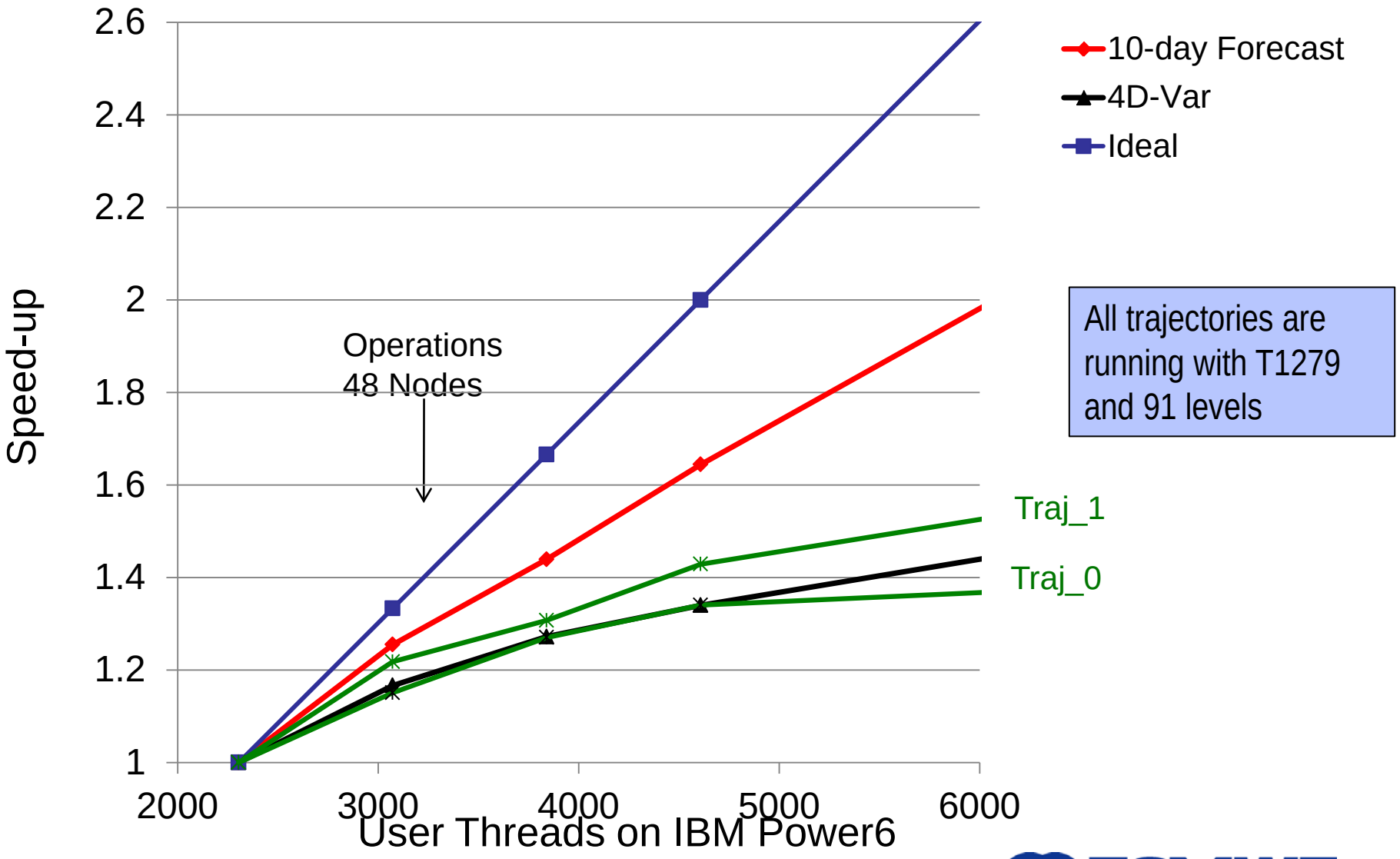


Can we still cope with it?



ECMWF

Can we still cope with it?



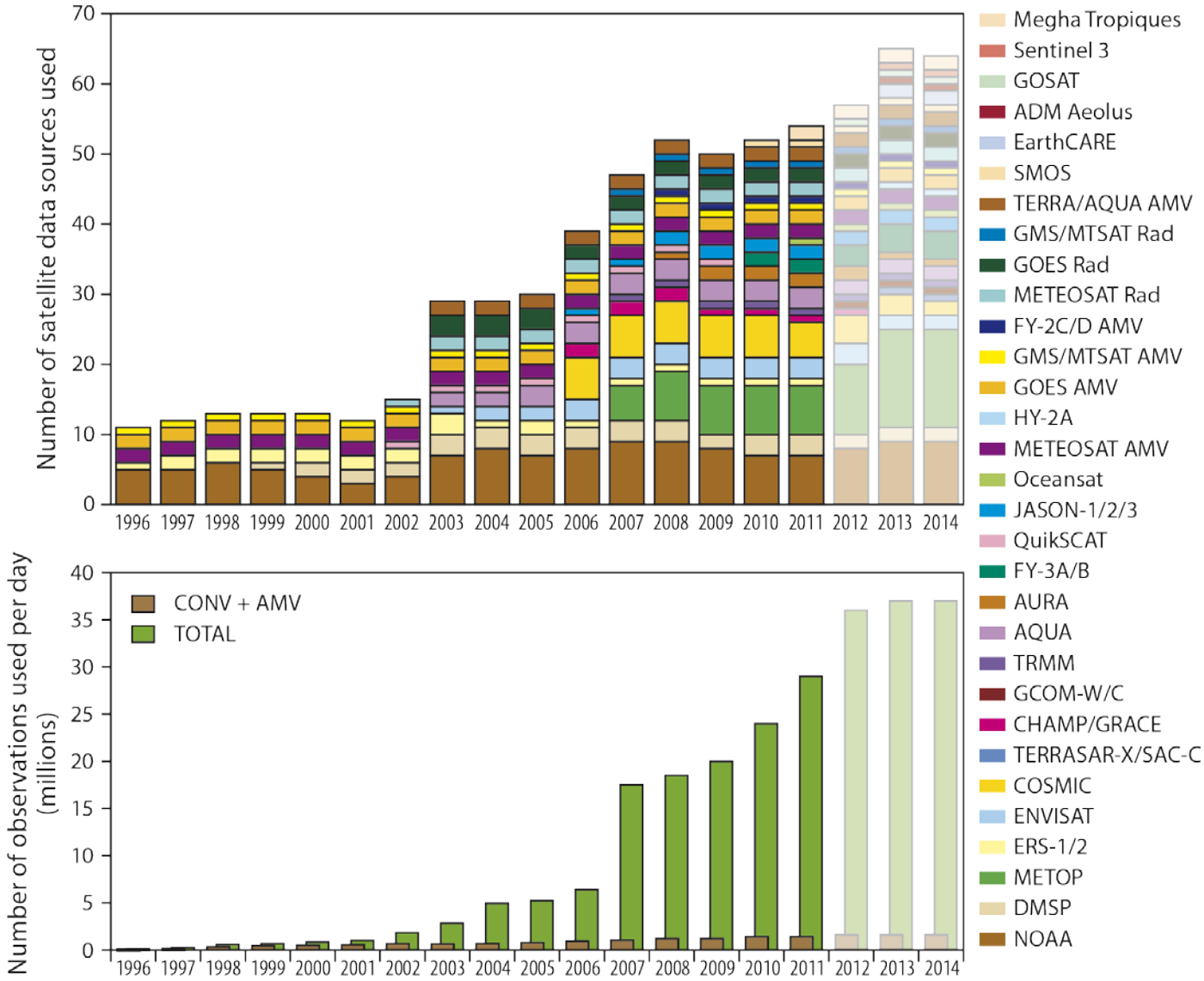
All trajectories are running with T1279 and 91 levels

ECMWF

What are the bottlenecks in our system?

- Traj_0 uses 80% more observations than Traj_1 and does quality control, blacklisting, thinning, screening, monitoring,…
- The execution of the 4Dvar requires running many sub-tasks consecutively at different resolutions with unnecessary IOs
- No fault tolerance because issues with a single observation can delay the whole operational suite in case of failure
- The same observation can be processed several times (4D-VAR, EDA,…)
- ODB and our HPCF system are very efficient: we can store without thinking…
- Individual components of our system have been analysed and optimised separately but the whole system has not

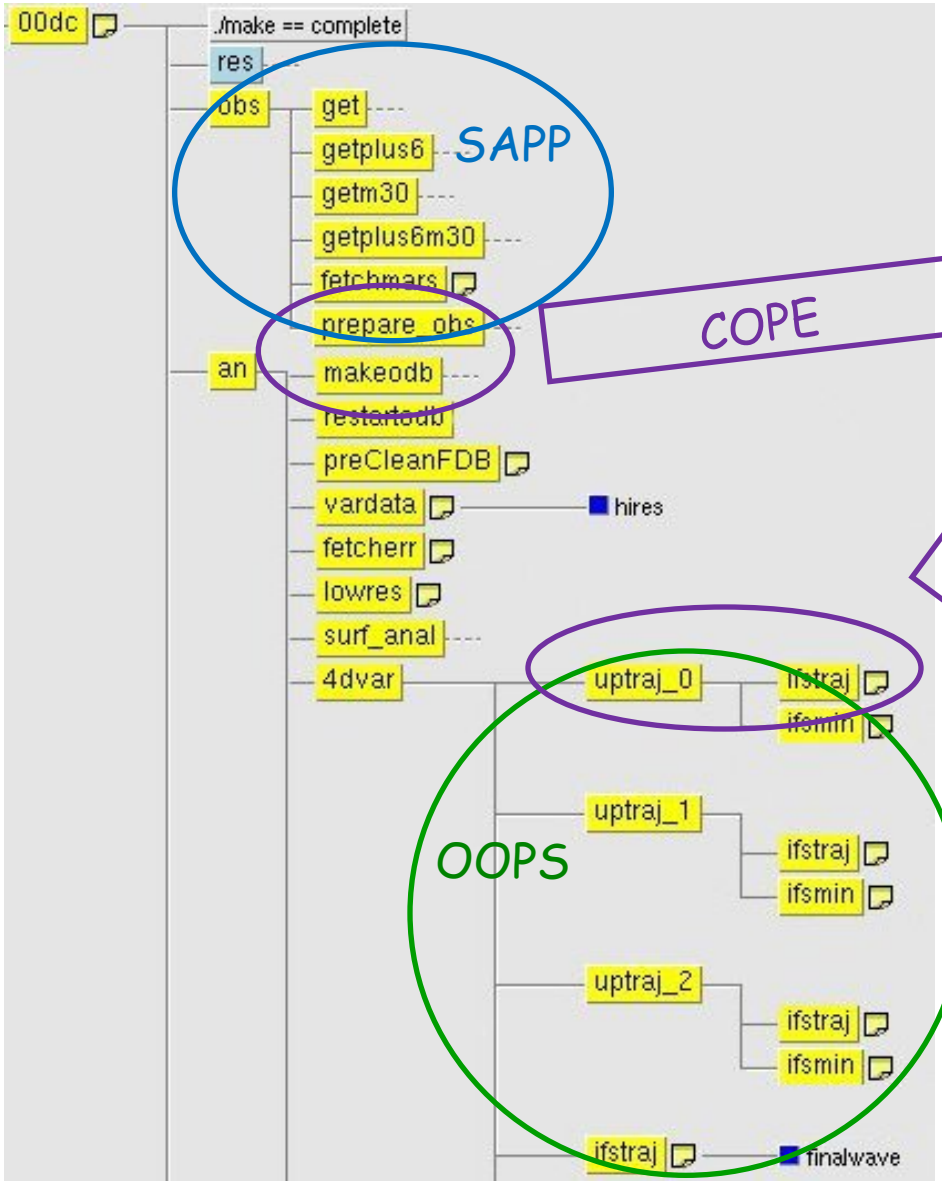
Satellite data used by ECMWF



How to tackle these issues?

- Externalize as many Observation Processing activities (quality control, blacklisting, etc.) , from the core of IFS and perform them before the actual analysis and once only
- Enhance error detection and handling
- Consolidate quality control activities
- Think carefully about what to store and how to organise our observational system
- Modularize our system to improve flexibility, validate components "independently" of IFS and reduce maintenance costs

ECMWF future NWP system



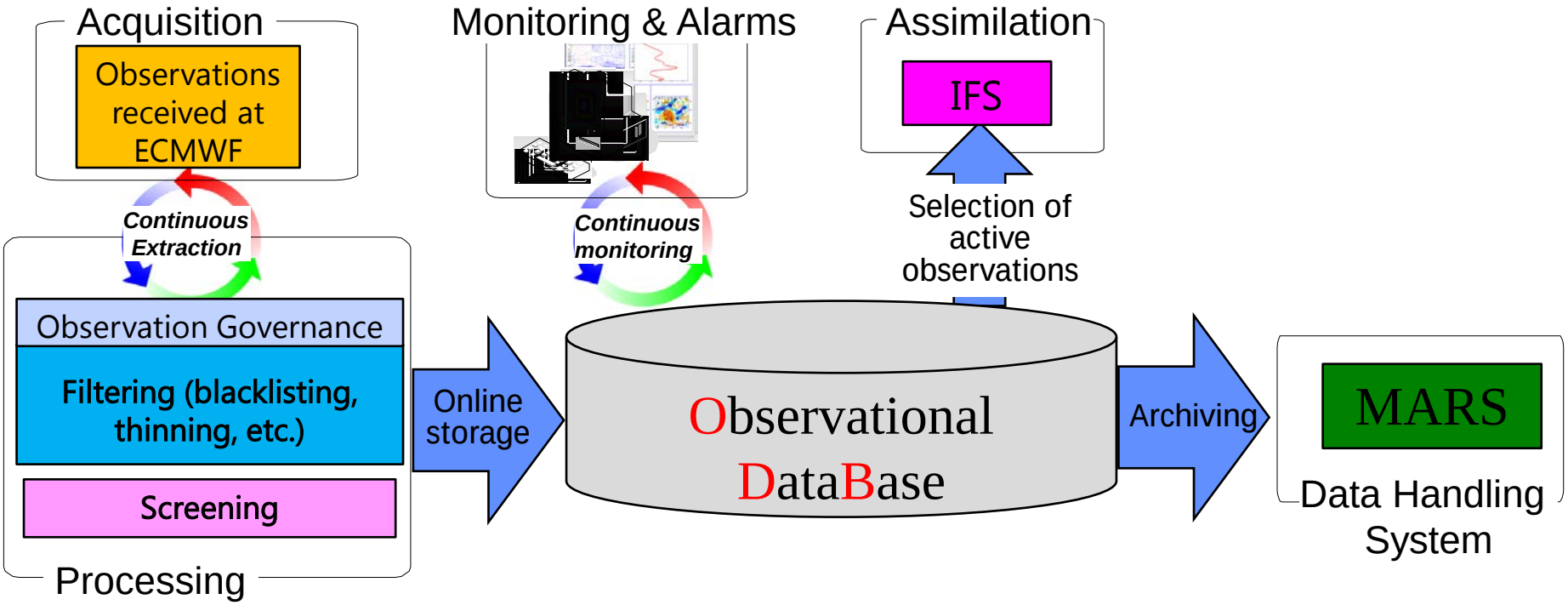
SAPP: *Scalable Acquisition and Pre-Processing*

COPE: *Continuous Observation Processing Environment*

- Remove observation pre-processing from time-critical path
- Pre-process observations only once (for all our operational suites) and keep ODBs online
- Perform screening outside IFS

OOPS: *Object-Oriented Prediction System*

Continuous Observation Processing Environment (COPE)



→ 2 years project initiated by Drasko Vasiljevic and done in collaboration with Meteo-France and the HIRLAM consortium (IFS/Arpege)

COPE components

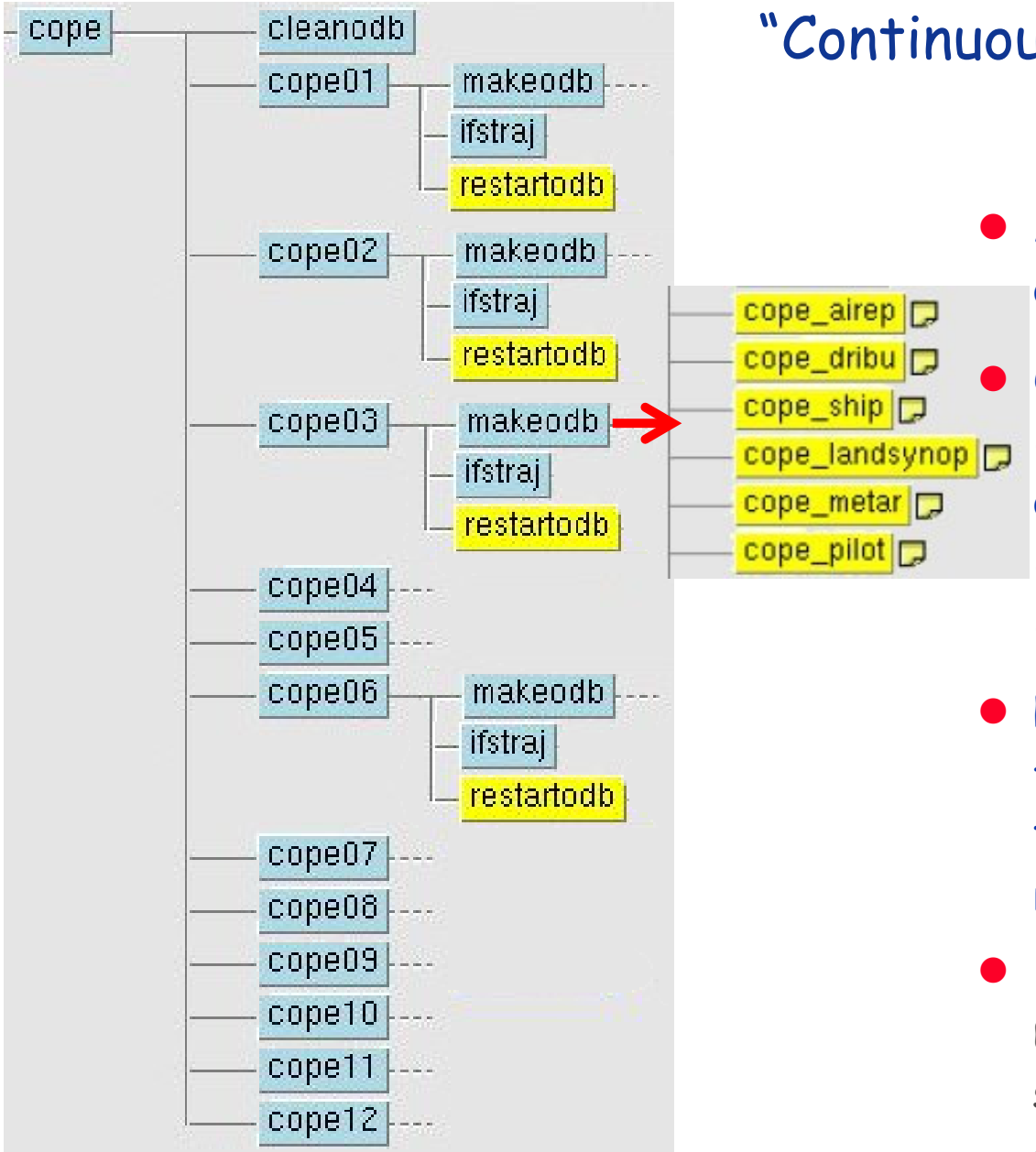
- Continuous extraction (SAPP project)
- COPE filters (quality control, blacklisting, etc.)
- First guess check, thinning, screening (use and adapt OOPS framework)
- Observational Database: a new data platform for COPE
- Monitoring & Alarm systems

COPE filters (Tomas Kral)

- A sequence of transformations (quality checks, unit conversions, computation of derived parameters e.g. wind speed and direction to u and v components, height to pressure coordinates, dumb thinning, bias correction, blacklisting) is applied to each observation
- IOs are minimal because these transformations are chained one after another

- Run the filter chain as soon as the data is available and run it once!
- Exploit intrinsic scalability of observation processing

"Continuous" observation processing



- A COPE suite for all our observation processing work
- Output from filters are written in different ODBs (one ODB per observation group but we could increase the granularity to increase parallelism)
- Run several "ifstraj" to compute first guess departures according to the arrival time (rdb_date, rdb_time)
- Keep several ODBs online and merge the ODBs for the screening and select active observations for the analysis

A new data platform for COPE: why and how?

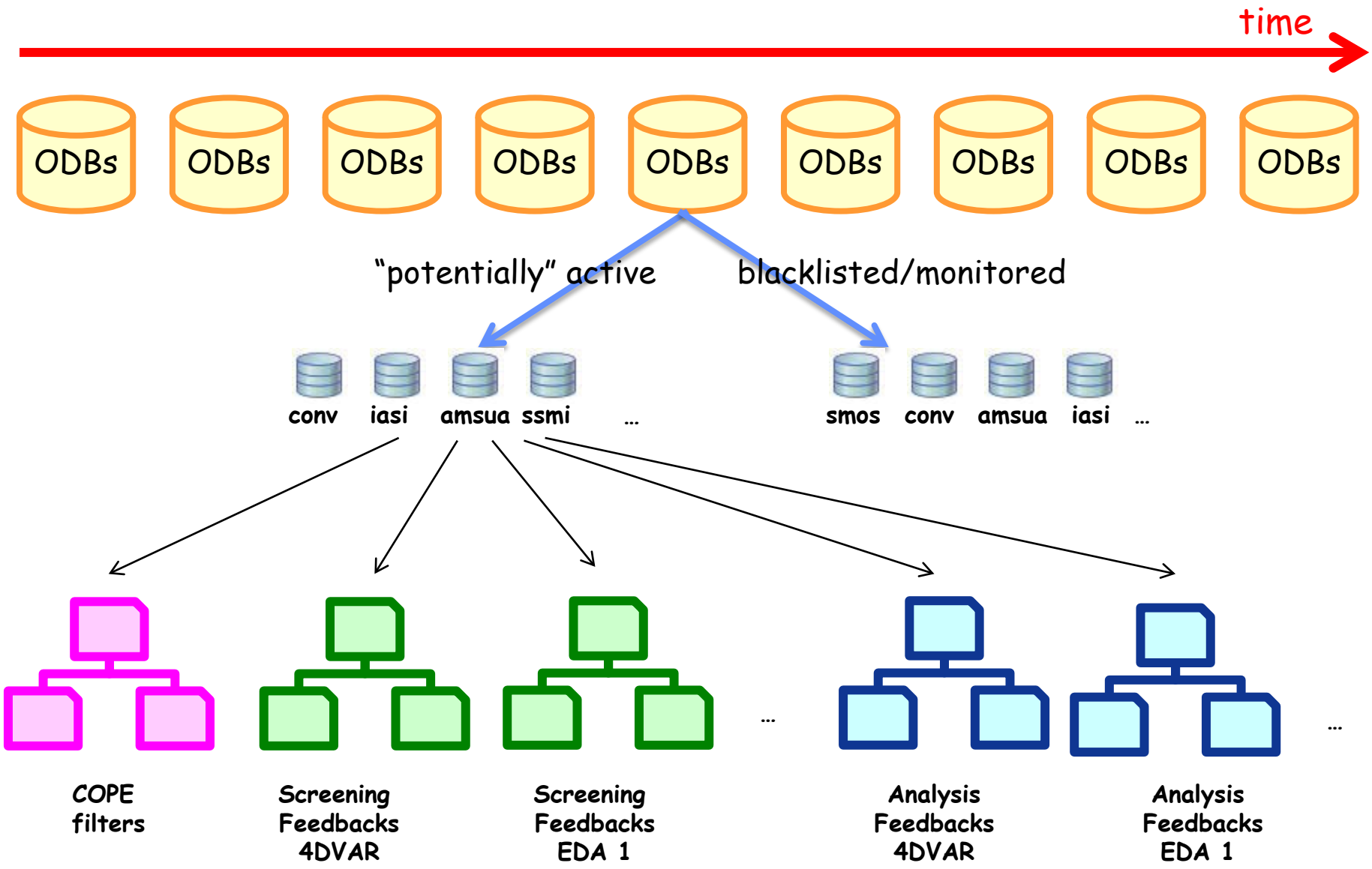
→ If we process observations earlier, where and how do we store them?

- It depends on the outcome of the COPE filters:
  - An observation is flagged as "potentially" active or blacklisted or monitored,…

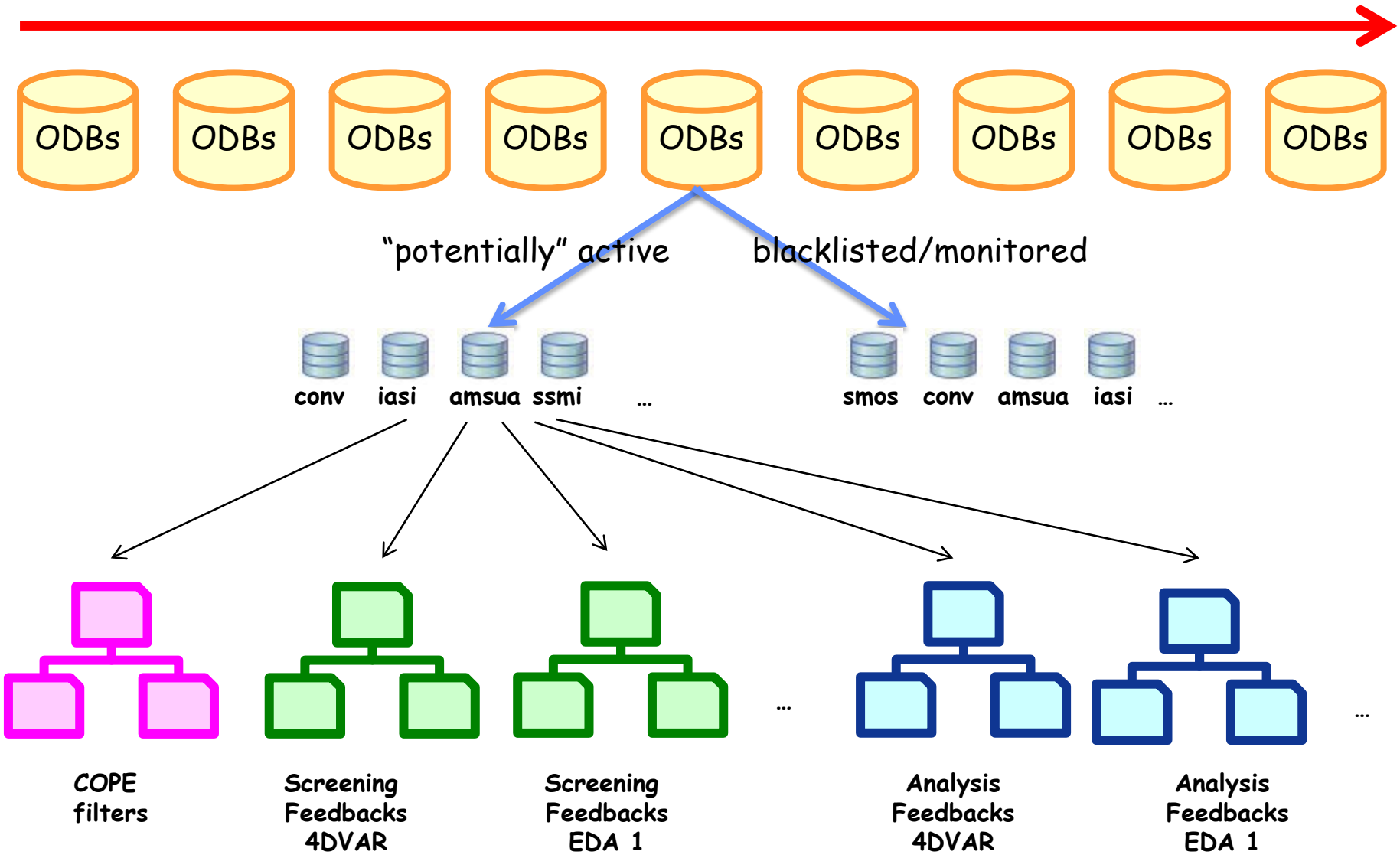
- Only "potentially" active observations are "presented" to the first guess check, thinning, screening
  - This step requires IFS and can be run several times for different set of observations and for various "applications" (4D-VAR,EDA,…)
  - The number of MPI tasks of the target application drive the observation distribution

- First guess departures of monitored or blacklisted observations can be computed separately (outside the time critical path)

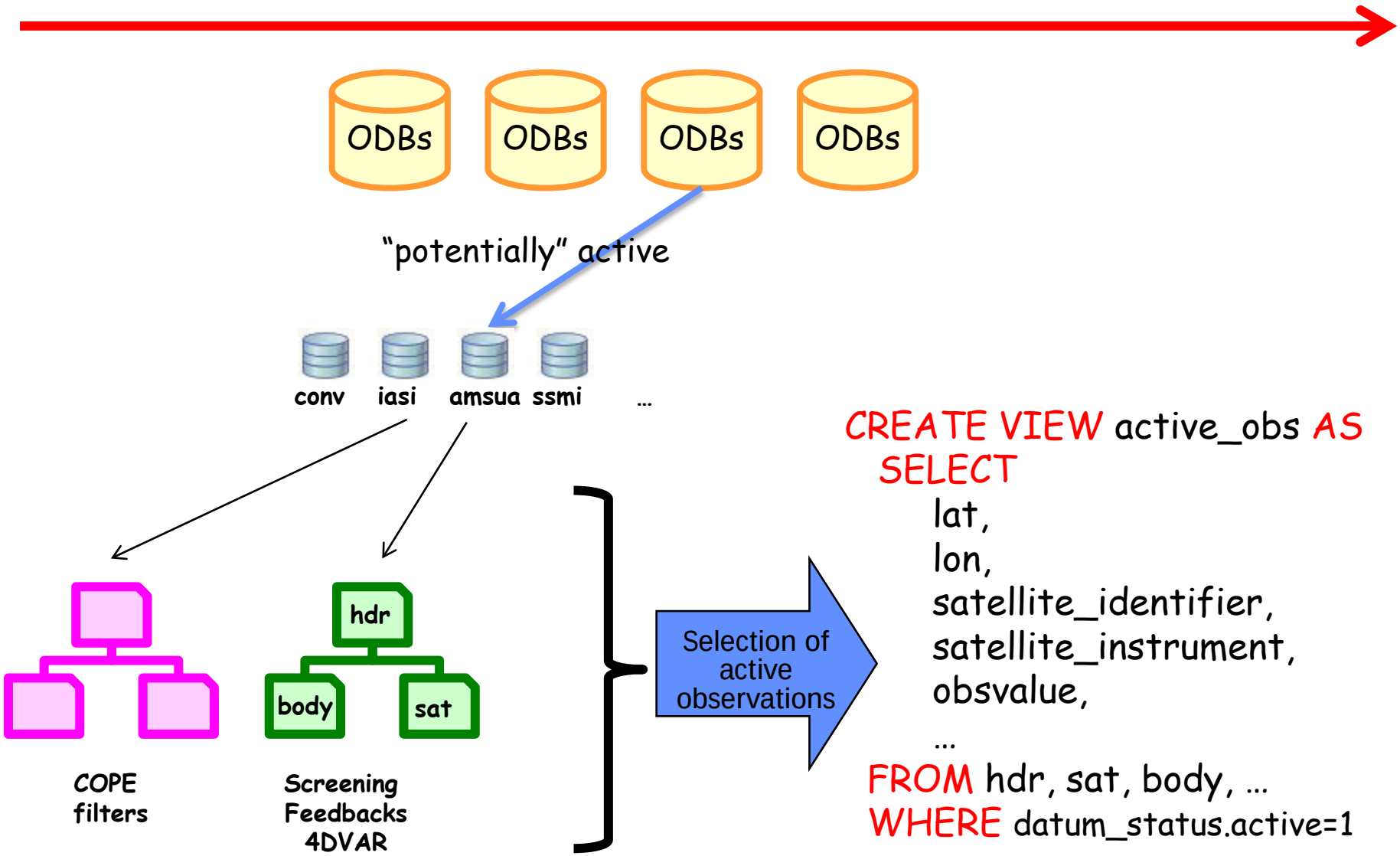
Our new data platform: no more than ODBs…



Create a view for each target application



Create a view for each target application

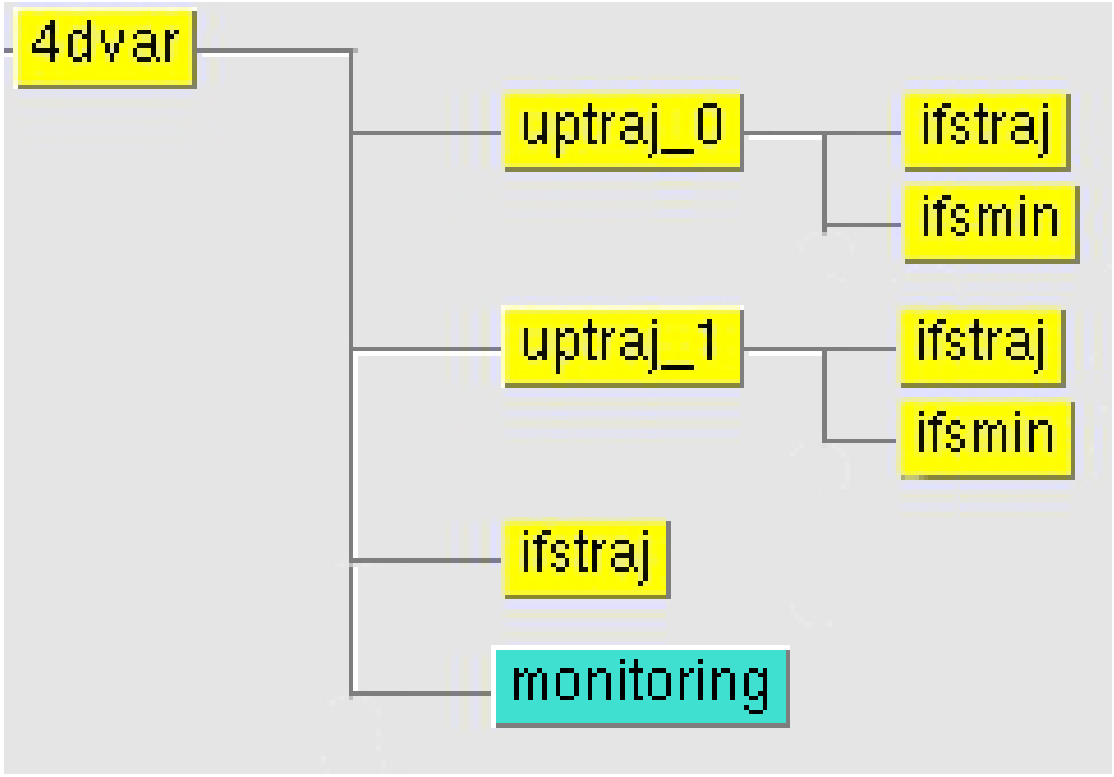



```
CREATE VIEW active_obs AS
  SELECT
      lat,
      lon,
      satellite_identifier,
      satellite_instrument,
      obsvalue,
      ...
FROM hdr, sat, body, ...
WHERE datum_status.active=1
```

Some facts about our "new" data platform

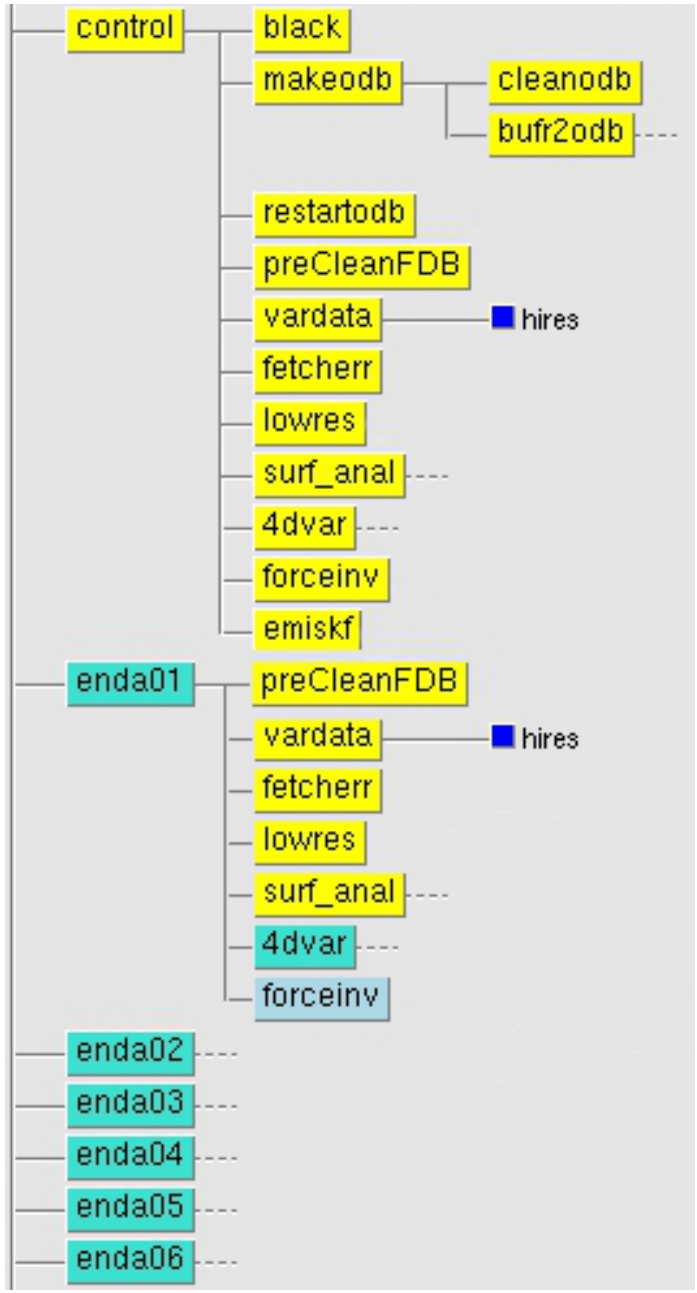
- Once written, ODBs become READONLY, "feedbacks" (from the COPE filters, screening, analysis,…) are written in new separate files and for each target application
- Metadata provides an up-to-date view of one or several (merged) ODBs for a given task (screening, minimisation) and a given operational suite (EDA, deterministic 4DVar short cutoff or dcda,…)
- The same database can be seen differently depending on the task or the operational suite

Operational "monitoring"



- Passive data can be decoupled from the main atmospheric analysis.
- Monitoring of "passive" data should be out of the critical path.
- SMOS will be the first (CY38R2), but it can include other "passive" data.
- Advantage for SMOS: supported by operations

➢ Included in our next operational cycle CY38R2



Ensemble Data Assimilation

- ODBs are created only once (by the control)
- Each member reads ODBs from the control and write its own "feedbacks" in different files (enda_1, enda_2, etc.)
- Each member (+ control) has its own local view of ODBs
- For archiving ODBs in MARS, a global view is "created" (metadata of each individual members + control are merged together)

How far are we from an operational implementation?

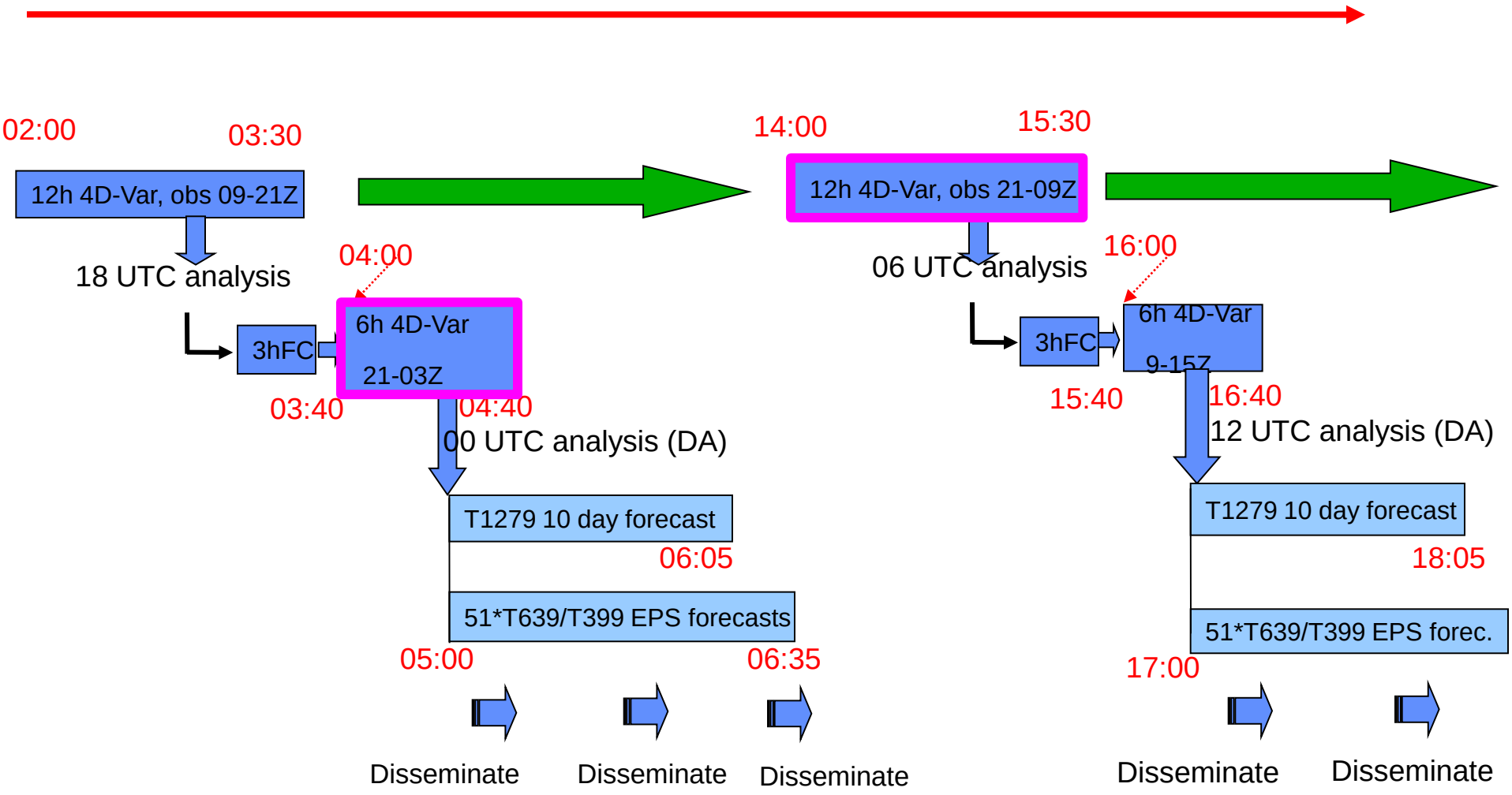
➢ Very far…

- Still not able to start running with active observations only without creating an intermediate ODB (and then matchup analysis feedbacks in the "global" ODB view): it involves "unnecessary" IOs.
- Still some efforts to be done to improve the robustness of our system (fault tolerance)
- There are still many unanswered questions:

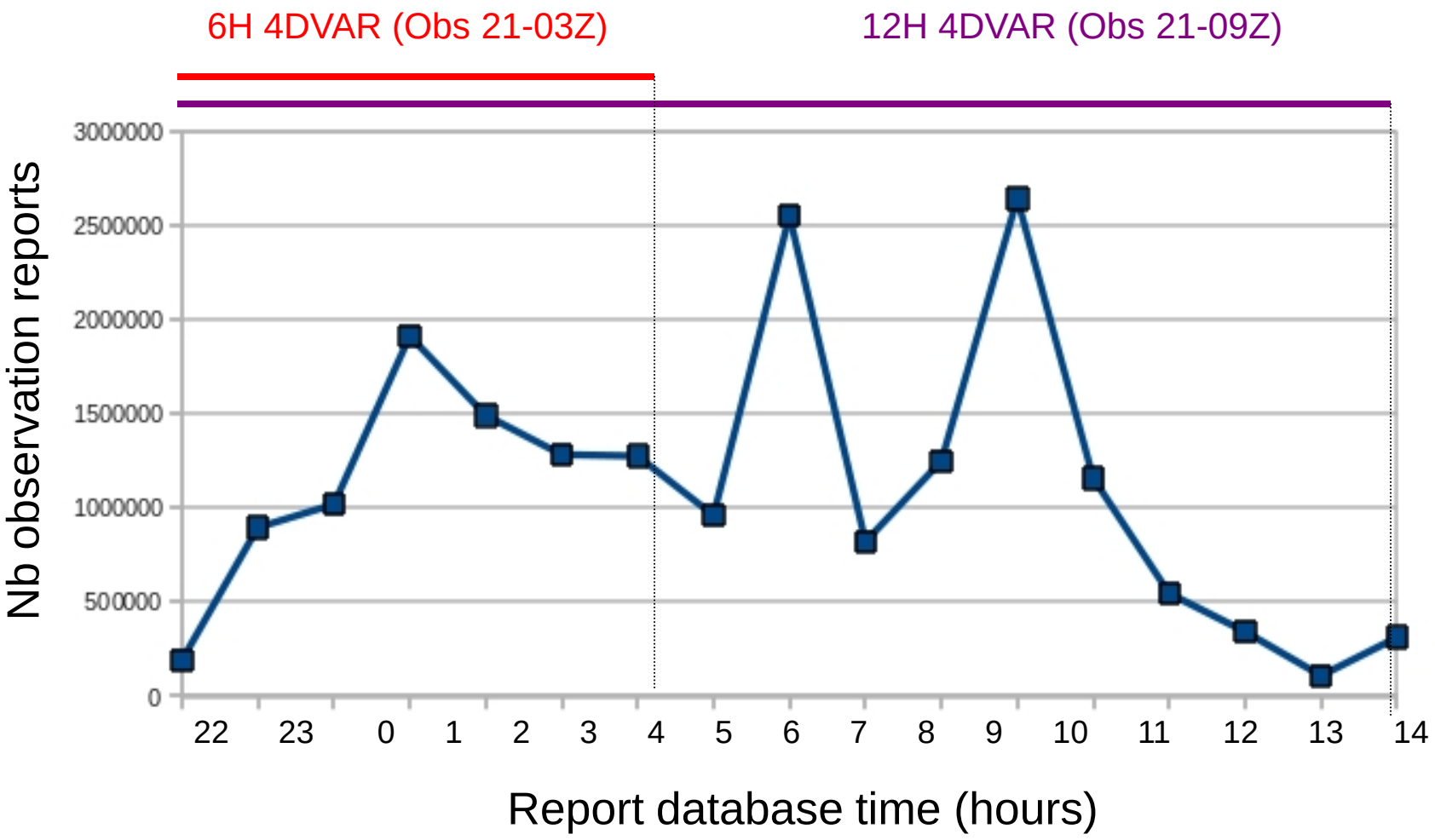
  - Screening, thinning can they be done incrementally?

  - How does it fit with our tight operational schedule?

Operational schedule for early delivery

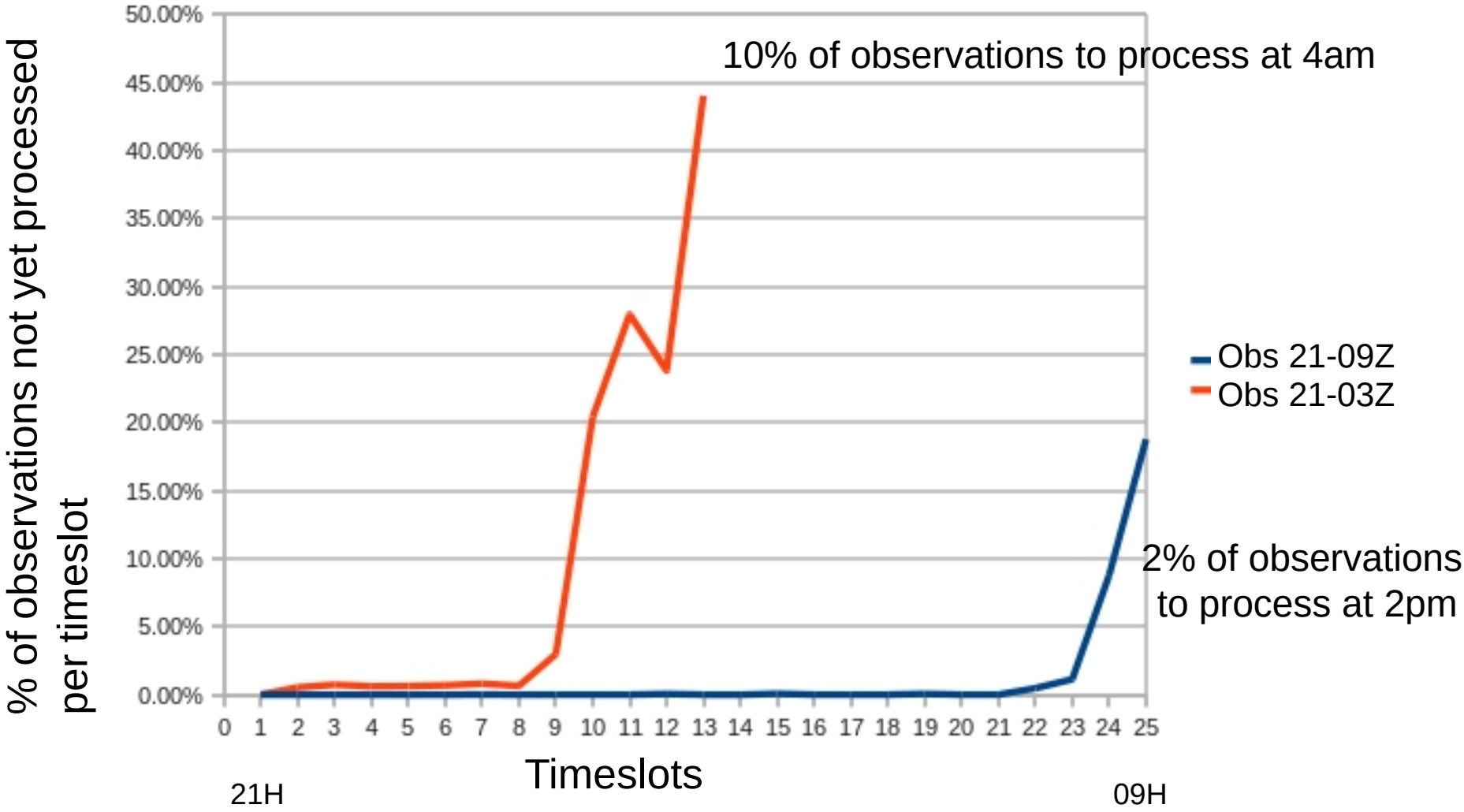


Number of observations for 12H 4DVAR (Obs 21-09Z)



Report database time is the time an observation "arrives" at ECMWF

Report DataBase time and observation time



1 timeslot ≈ half an hour

ECMWF

Summary and Conclusions

- Our approach seems valid… but to be really effective from an operational point of view it relies on many other components:
  - SAPP project,
  - usage of OOPS component for computing first guess departures
  - Extend OOPS for the screening and implement new "screening" methods
- It also relies on our ability to "understand" our current observation processing framework (make sure we don't forget anything!)
- And our ability to **work efficiently together!**

Questions?