Primary author: **Hansen, Lars** (DMI - Danish Meteorological Institute, Data and Climate Division), lsh@dmi.dk

Abstract ID: 3O2

The challenges of a Metadata database

A metadata database holds information about meteorological stations. This information changes over time; as an example, a station may change location. If you simply update the corresponding information in the database, you loose the information about the previous conditions. And climatologists don't like that. So you need a *temporal database* in which each piece of information is augmented with a *valid-time* interval. But what happens when two such augmented tables are joined?

StatDB

This talk

- A bit about how StatDB fits into our system

- A bit about StatDB as a temporal database

- Positions in StatDB

The new databases
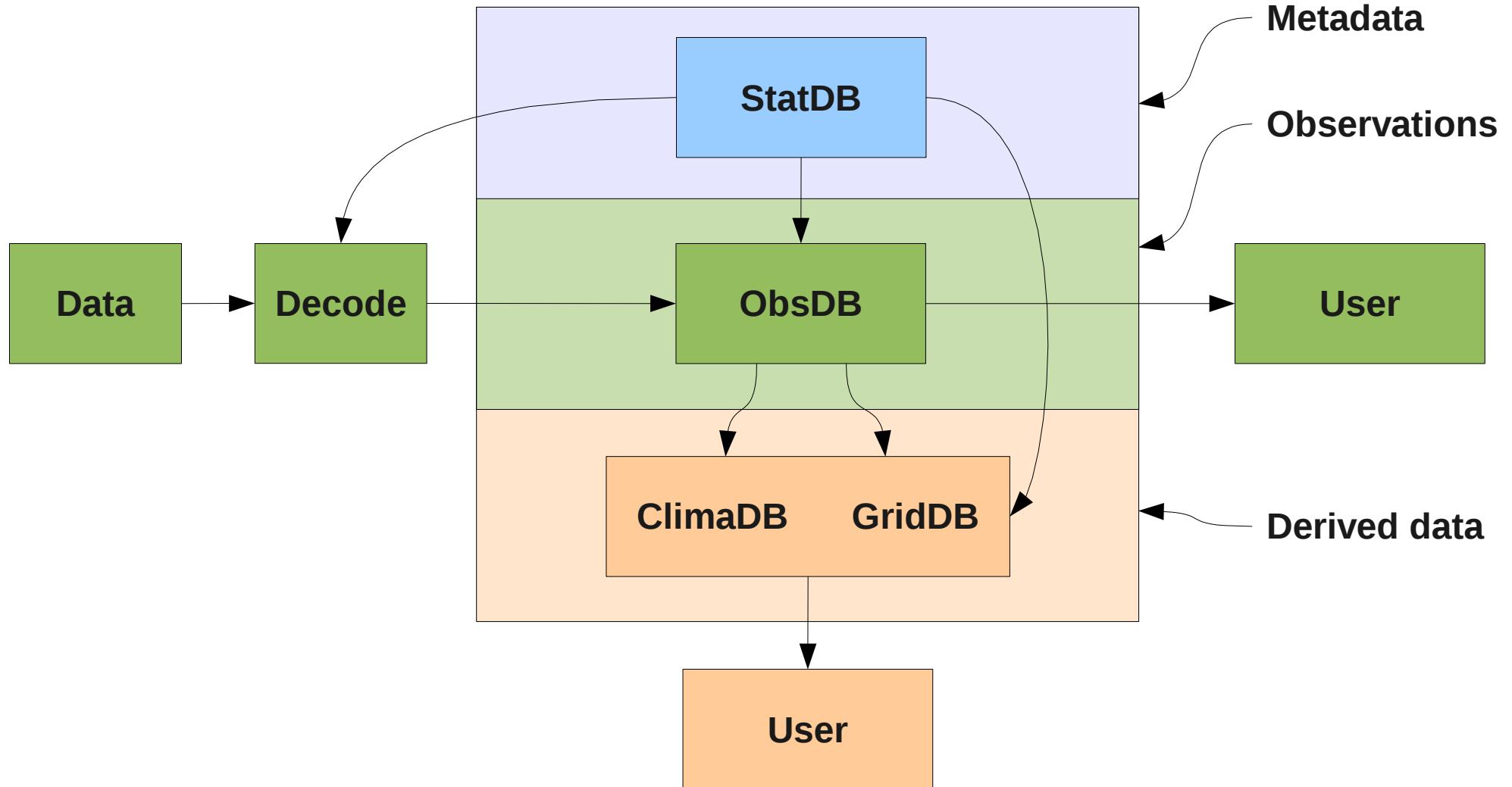
What's was missing in the picture?

The new databases

Metadata

Observations

Derived data

StatDB

Data → Decode → ObsDB → User

ClimaDB   GridDB

User
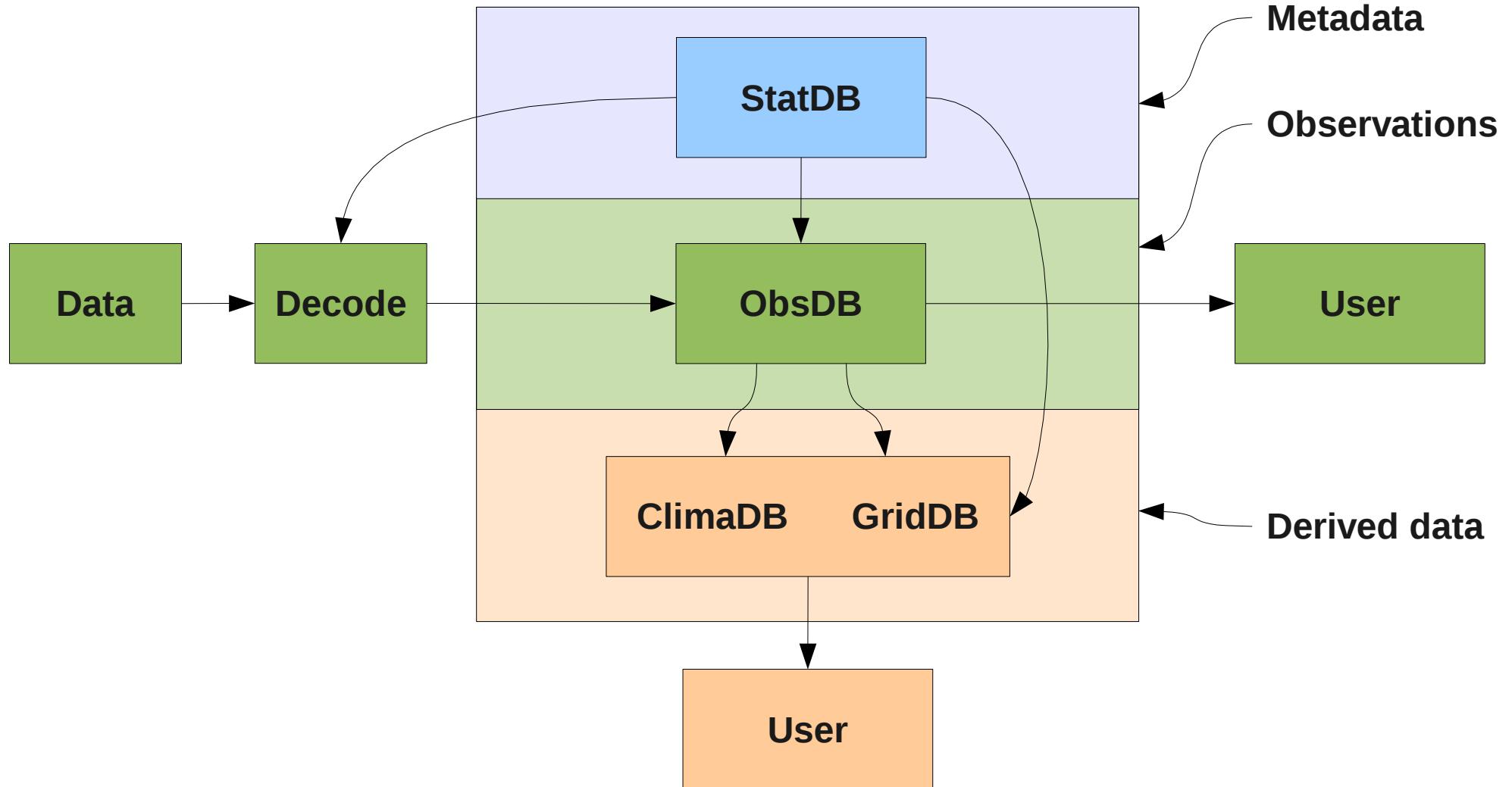
No arrow going into StatDB !

StatDB environment

Old UI

New UI

WMO

Ingres

StatDB

Data transfer

Programs

ObsDB

ClimaDB

Decode

Existing system, to be shut down

Existing system, maintained and further developed

To be designed :-)

External system
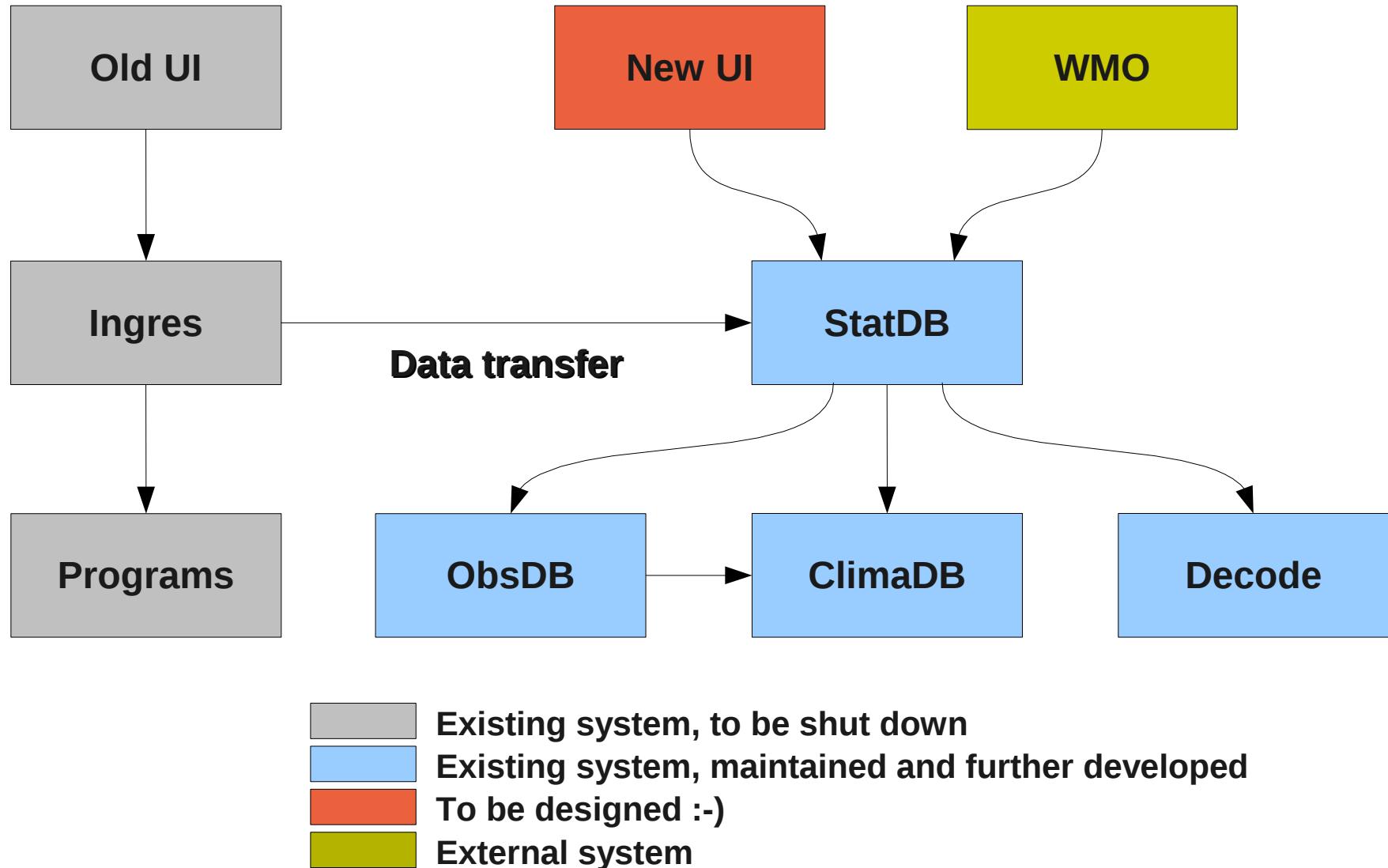
StatDB as a temporal database

Temporal database

Wikipedia:

A temporal database is a database with built-in time aspects.
More specifically the temporal aspects usually include valid-time.

StatDB design
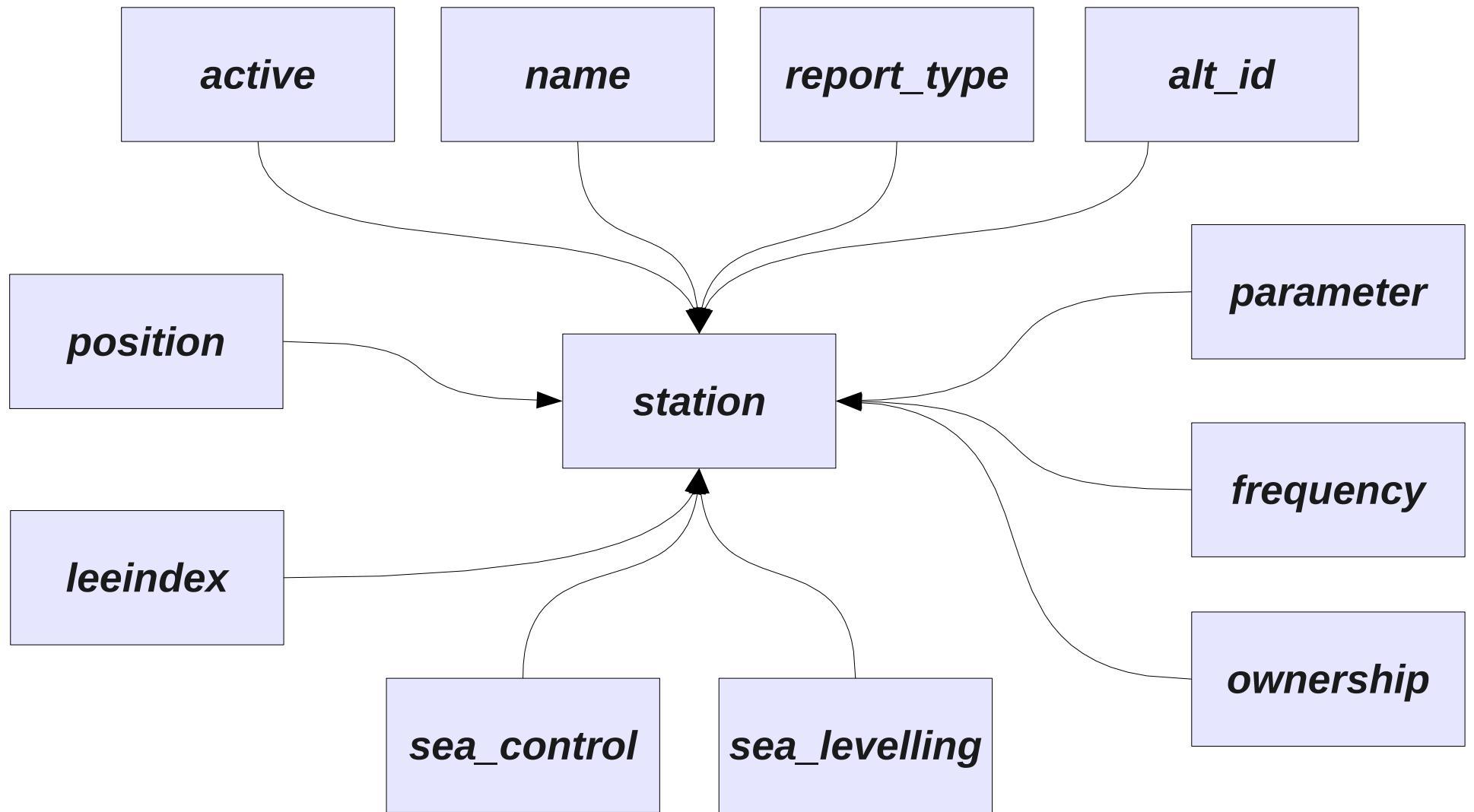
active

name

report_type

alt_id

position

station

parameter

leeindex

frequency

ownership

sea_control

sea_levelling

Time intervals

name

statid
☛ **start_time**
☛ **end_time**
name
...

report_type

statid
☛ **start_time**
☛ **end_time**
report_type
...

station

statid
...
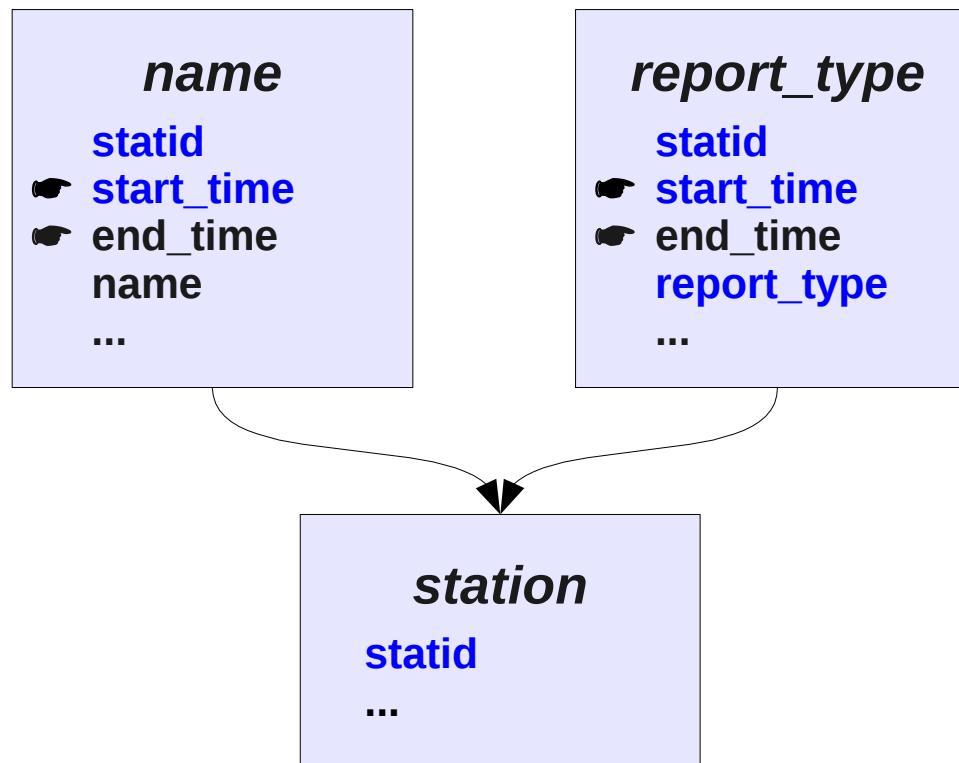
Problem: Time-joins

name

id	name
02	Smallville

_report_type_

id	report
02	synop
02	metar

Problem: Time-joins

name

id	name
02	Smallville

report_type

id	report
02	synop
02	metar

join on id

id	name	report
02	Smallville	synop
02	Smallville	metar

Problem: Time-joins

name

id	name	start	end
02	Smallville	1990	2001
02	Metropolis	2001	2009

report_type

id	report	start	end
02	synop	1990	2009
02	metar	2003	2009

Problem: Time-joins

name

id	name	start	end
02	Smallville	1990	2001
02	Metropolis	2001	2009

report_type

id	report	start	end
02	synop	1990	2009
02	metar	2003	2009

Join on id

id	name	start	end	report	start	end
02	Smallville	1990	2001	synop	1990	2009
02	Smallville	1990	2001	metar	2003	2009
02	Metropolis	2001	2009	synop	1990	2009
02	Metropolis	2001	2009	metar	2003	2009

Problem: Time-joins

name

id	name	start	end
02	Smallville	1990	2001
02	Metropolis	2001	2009

report_type

id	report	start	end
02	synop	1990	2009
02	metar	2003	2009

Join on id

id	name	start	end	report	start	end
02	Smallville	1990	2001	synop	1990	2009
02	Smallville	1990	2001	metar	2003	2009
02	Metropolis	2001	2009	synop	1990	2009
02	Metropolis	2001	2009	metar	2003	2009

Smallville, metar ?!

Problem: Time-joins

name

id	name	start	end
02	Smallville	1990	2001
02	Metropolis	2001	2009

report_type

id	report	start	end
02	synop	1990	2009
02	metar	2003	2009

Time-join on id

id	name	report	start	end
02	Smallville	synop	1990	2001
02	Metropolis	synop	2001	2009
02	Metropolis	metar	2003	2009

This is what we want :-)

Solution

Nice exercise in computer science :-)
Solve it correctly, but do it only once!

create_interval_join(xname, yname)

Time-joins has same type

name

id	name	start	end
02	Smallville	1990	2001
02	Metropolis	2001	2009

report_type

id	report	start	end
02	synop	1990	2009
02	metar	2003	2009

Time-join on id

id	name	report	start	end
02	Smallville	synop	1990	2001
02	Metropolis	synop	2001	2009
02	Metropolis	metar	2003	2009

... so we can cascade

name

report_type

position

create_interval_join

name_report_type

create_interval_join

name_report_type_position
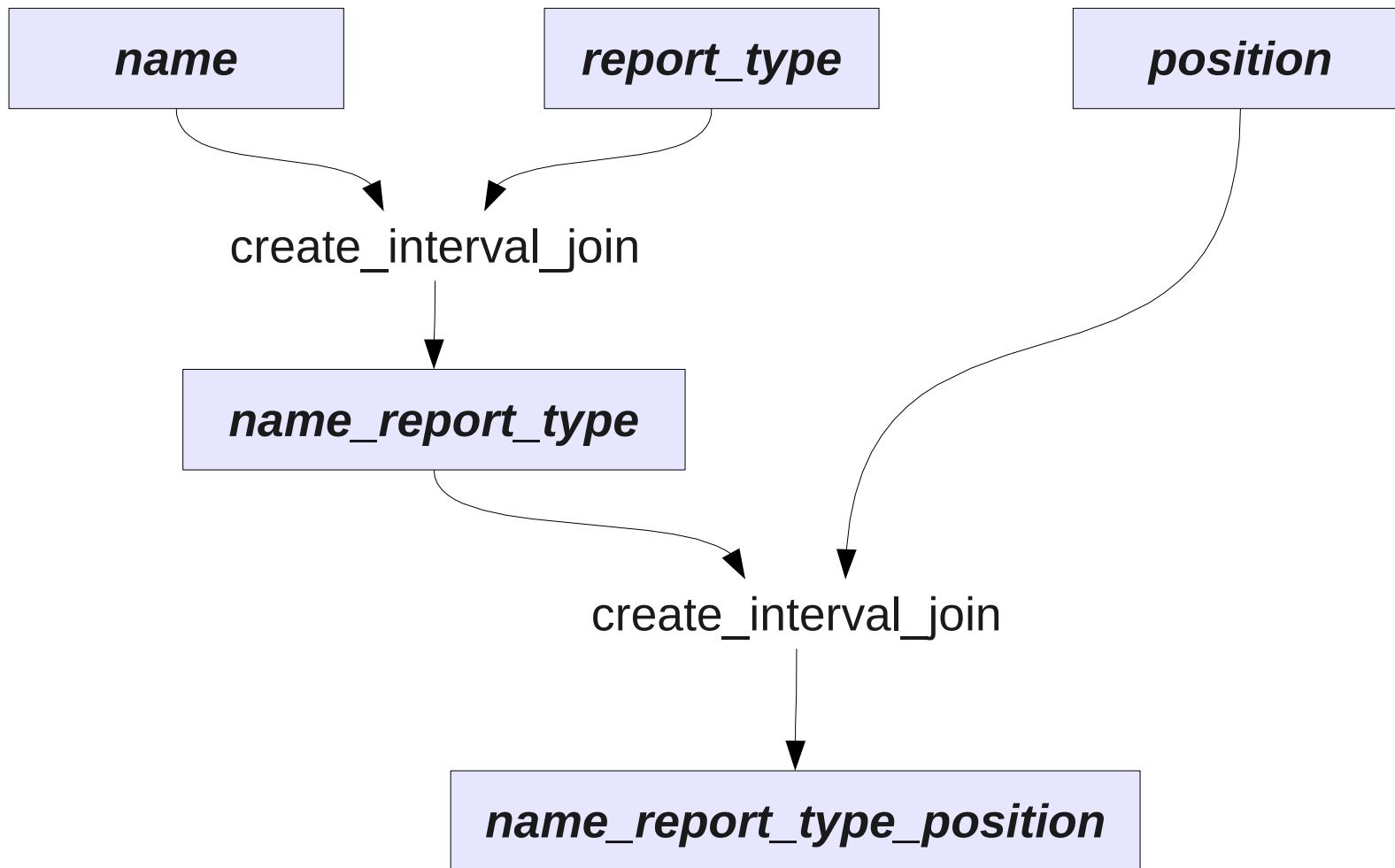
Positions in StatDB

Positions

- Saved as lat, long, height (wgs 84)

- Updatable views provide eg. UTM ed50 using conversion functions from the library Proj4

That's all folks!