



# Hafslund

## Hafslunds implementation of EMPS v.10

and how we want to work with SINTEF models

# Agenda

1. FUNdament – how did it all start?
2. EMPS v10 – how far have we come on our journey?
3. Sintef models – what would we like to see?



*FUNdament – how did it all start?*

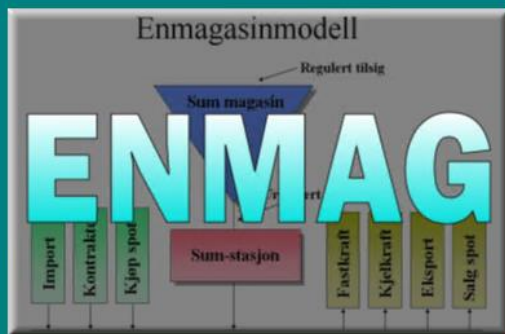
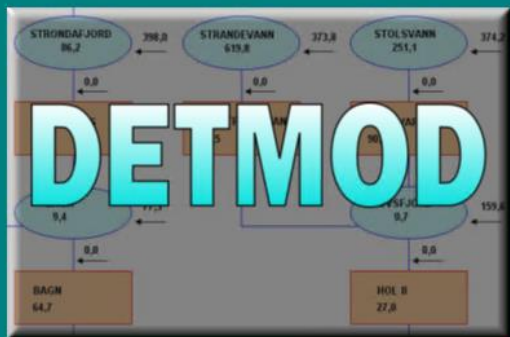


## In autumn 2020,

- It was time-consuming to fetch and check input data every time you wanted to run EMPS.
- We had many ideas to improve our modelling, but it was difficult to make changes to our existing model configuration.
- It was difficult to both spot errors and to debug (lack of transparency).
- Model operations and operative model configuration was person-dependent (single point of failure).
- ...

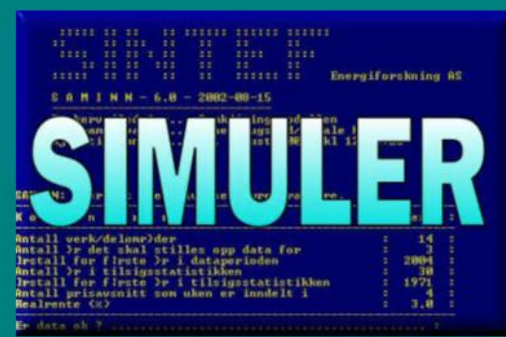
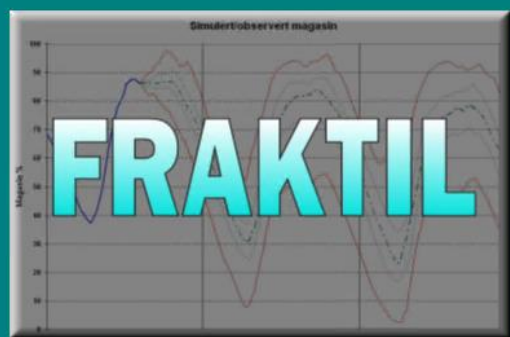


# Samkjøringsmodellen



## INPUT-----

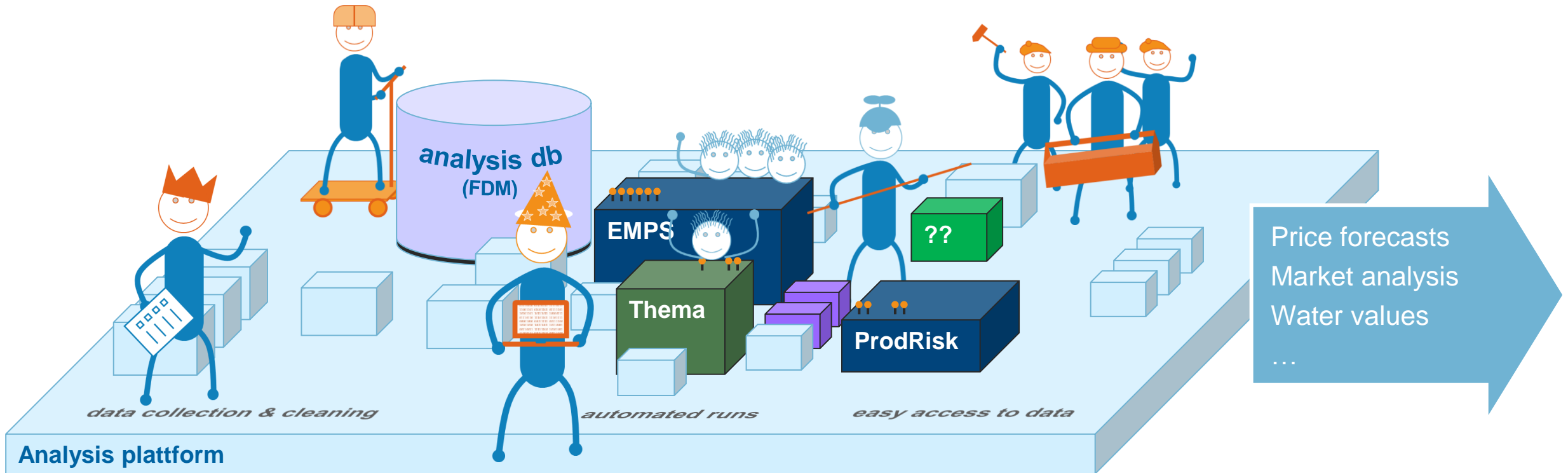
Startmagasiner	Tilslag
Kortid Vind/Forbruk	Eksogene priser
Eksogen kapasitet	Kjernekraft
Brenselpriser	Termisk Kap
Sjekk endringer interne linjer	Alternativ Tilsigsprognose DETMOD



## RESULTATER-----

Res Rapp	Ukeres Rapp	Til FDM priser
----------	-------------	----------------

# Hafslund decided to start FUNdament.



**FUNdament is a strategic initiative to modernize how we do seasonal and long-term analysis work at Hafslund. This applies to EMPS and all other models that we use for that purpose.**

*EMPS v10 – how far have we come  
on our journey?*



In August 2023, we switched over to EMPS v10,  
but we have met some challenges getting there.





# Circular dependencies between files

Data should live in databases, and you should be able to generate all files from scratch.

```
/// <summary>
/// Function that generates a file (maskenett.data) that sets up references for power line connectors between different price areas.
/// The referances is based on the mapping provided within the interconnector objects in the DAL.
/// Maskenett.Data referances need to correnspond to the referances generated in CapFile.h5 (see dynamic data generation).
/// At the time of writing (2021-08-31) the Sintef API for generating CapFile.h5 uses the file model.h5 to generate price area and line referances.
/// An issue with this approach is that model.h5 is not generated until after the model i run.
/// Hence, one need capfile.h5 to generate model.h5 and model.h5 to generate capfile.h5.
/// This recursive conflict needs to be resolved and this is currently being discussed with Sintef.
/// as a temporary workaround to enable testing of other functionality the mapping set up in the DAL corrensponds to that of an operation model.h5 file
/// so that the maskenett.data and capfile.h5 files do point to the same referances.
/// </summary>
/// <param name="jobDefinition"></param>
/// <param name="dummyValues">Whether to insert dummy values for capacities and revisions or insert a reference to CAPFILE. Default is <c>>false</c>.</param>
/// <returns></returns>
public async Task<string> GenerateInterconnectorMap( JobDefinition jobDefinition, bool dummyValues = false )
{
```

Rumours say that the R10.6-release (end of November) will fix this issue. We are very excited.

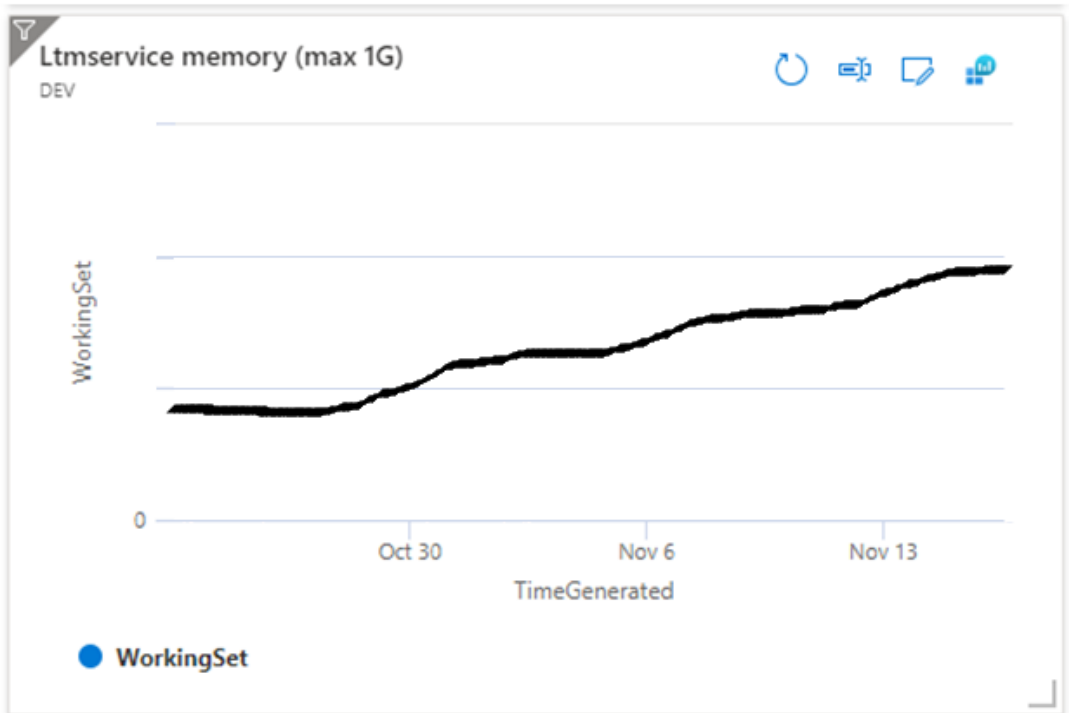


# Linux compatibility

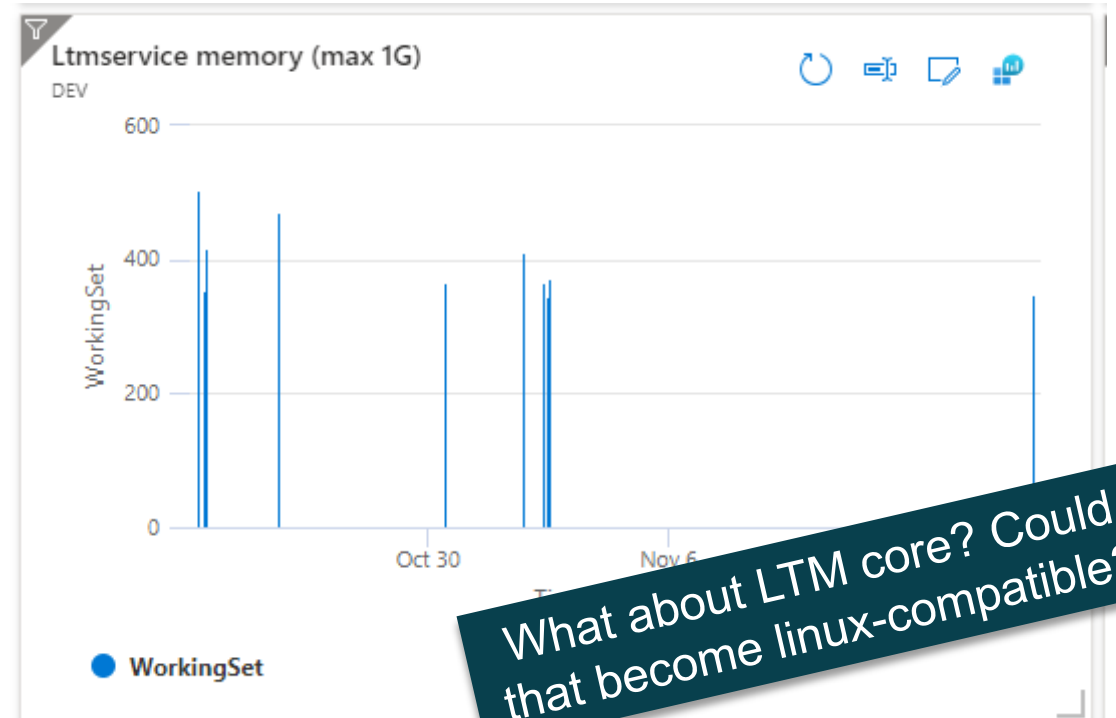
So far, we prefer to host our cloud services in a linux environment 10 out of 10 times.

- We build a service that uses Sintef's LTM API to update various input data, e.g. available capacity for each transmission line in cap\_file.h5.

App service (windows)



Kubernetes scaledJob (linux)



What about LTM core? Could that become linux-compatible?

# Calendar-functionality

**We want to use the same input data consistently across all our models, but models often have slightly different calendar-functionality.**

- When does a year start?
  - Sintef: Monday Week 1
  - Another model provider: «Regelen er at første time i modellen er første mandag i året (CET) selv dersom 31.12 er en mandag.»
- What about the length of a year? 52x weeks or continuous?
- Some input series have to be padded with zeros between the start of the year and the start of the simulation period, while forecasts for inflow and snow can start without padding.
  - Every exception causes extra complexity for which we have to write code.



- **Try to think ahead.** What do you want to achieve?
  - In the beginning, much of our code was written to support only one model configuration, e.g. functional IDs were used as «internal» IDs in the code.
- **Microservice architecture** is working well for us
  - Several services that generate input
  - Several services that write input to files
  - One service that executes ltm-commandos
- **Modelling freeze** to avoid continuous converting of our operative v9 dataset to a working v10 dataset.

# Some practical tips



*Sintef models – what would we like to see?*



# Our technical wish list for future Sintef models

**EMPS v10 was a step in the right direction and we would like to see more like that.**

- Standardisation!
  - File formats – e.g. just hdf5
  - Calendar functionality – ideally standardisation across model providers
  - Definition of a timeseries – no exceptions for specific input. Each exception adds complexity on our side.
- API from Sintef to update all input (files)
- Linux-compatibility for all code and programs



