# Simulating Carbon Opportunity Cost at Grid Sites

Dr Dwayne Spiteri et al

Discussion Evening: Sustainability in ErUM-Data - 18/11/2024

RESEARCH FACILITY
TOWARDS A MORE ENERGY-EFFICIENT AND SUSTAINABLE PATH

Scottish Grid Service

# Contents

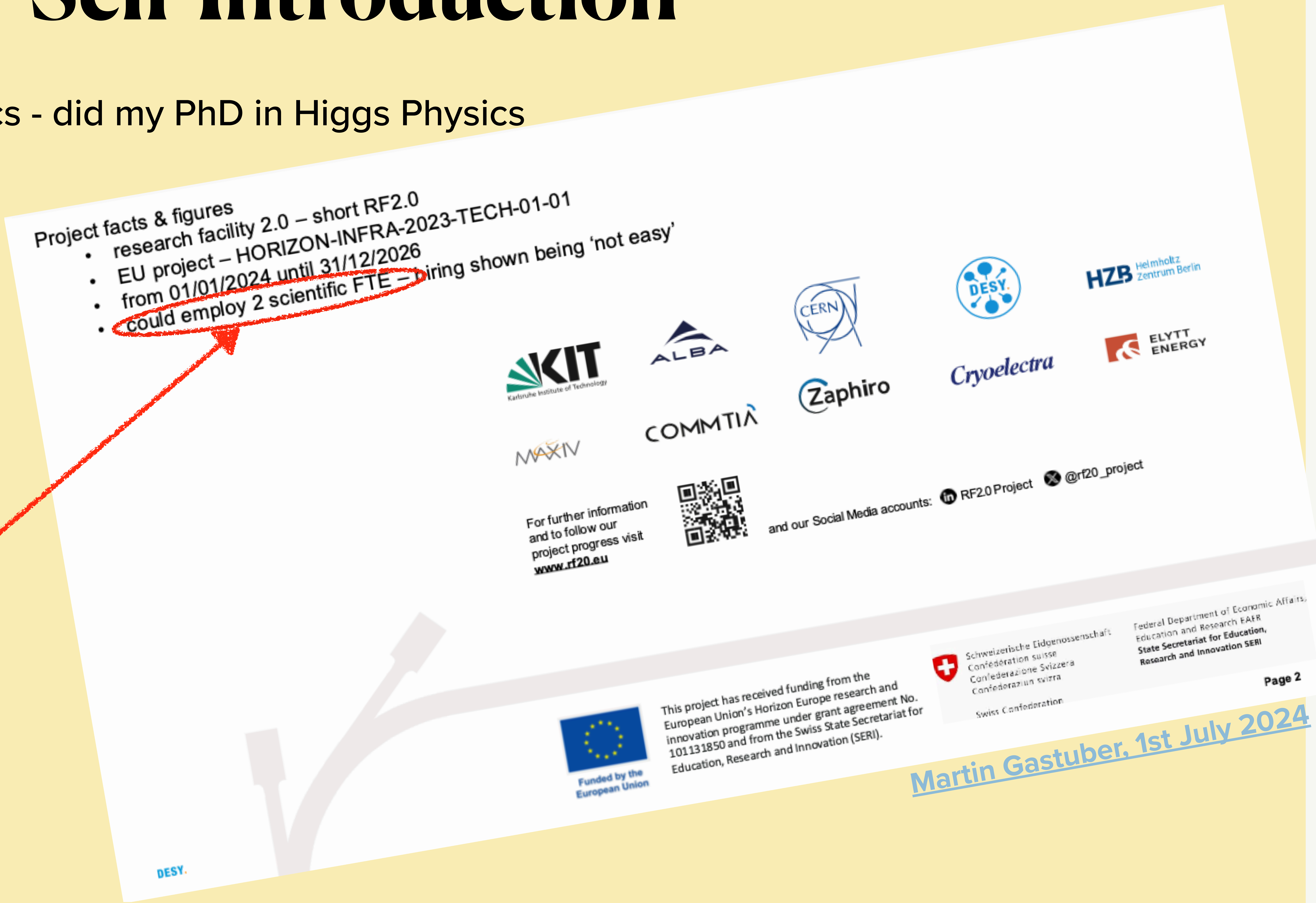
- Self-introduction
- Project Re-introduction
- What I plan to do on this project - ***I created a simulation of the computing site at Glasgow*** with the aim to test the carbon consumption of various running methods.
- Motivation
- Structure of the simulation
- Challenges
- Conclusion

# Self-Introduction

- My name is Dr Dwayne Spiteri. Former Particle physics - did my PhD in Higgs Physics

- Used to work in Scotland at the University of Glasgow

- I started working at DESY in October (one of this is me). Working in the IT group with Martin Gastuber and Kilian Schwartz




Martin Gastuber, 1st July 2024

- Funded by EU Project RF 2.0

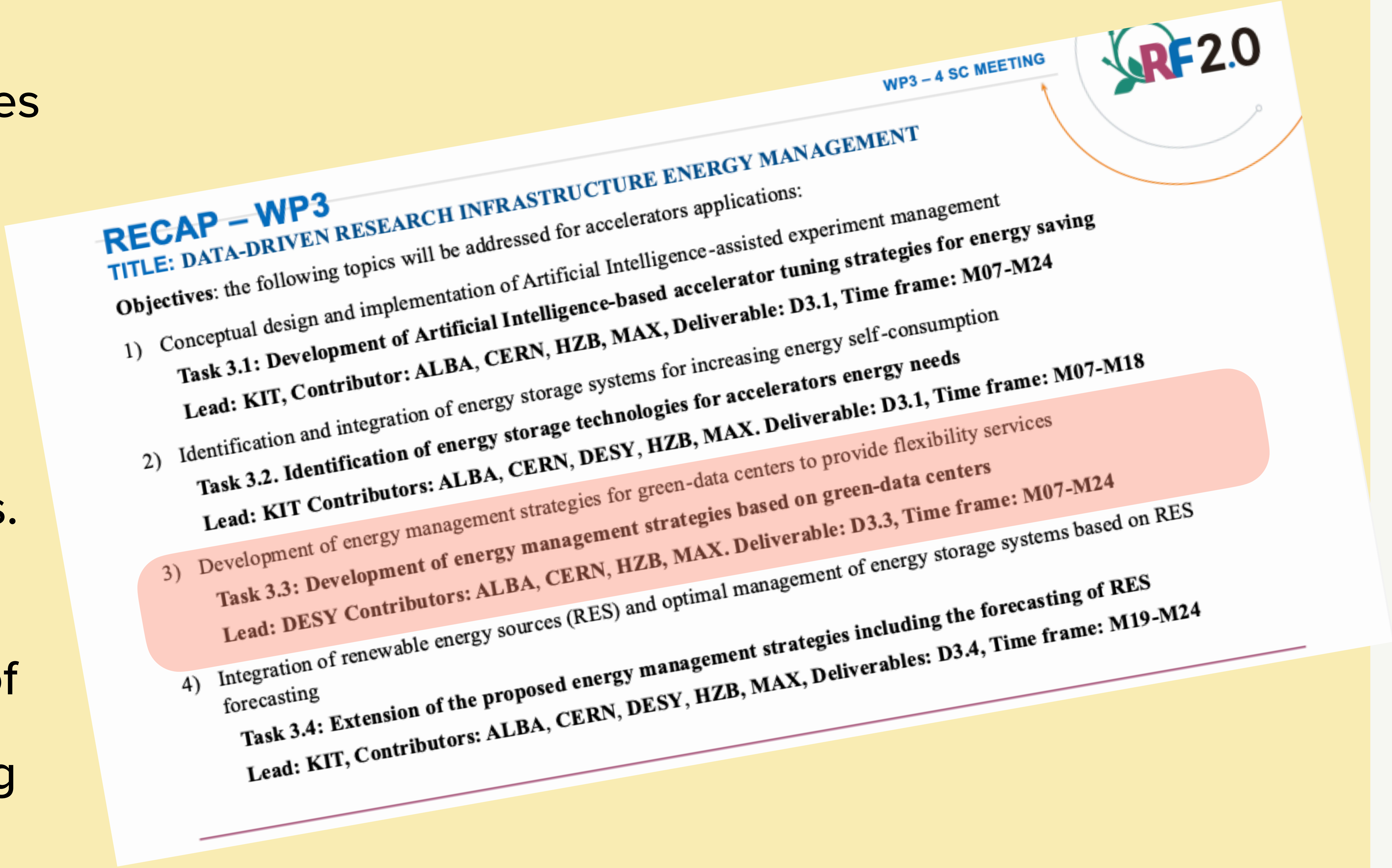
# What is R.F 2.0

- An EU-funded project whose remit covers the design and use of technologies for use at future accelerators; and the approaches we can take to manage energy at support infrastructures.

- It has more than 10 partners each responsible for leading work packages in various sections.

- Work Package 3 is about energy management at Research infrastructures, and **DESY is the only institution on this project** looking to develop strategies for the energy management of "green" data-centres.

- My idea was to create a digital twin of the a datacenter and use that to try and investigate energy/carbon saving strategies.



# Motivation

- There are a lot of good options for sites to choose from for future kit, especially as HEP workloads are running well on ARM architectures.

- Can't always try before you buy - but you often can either get performance markers from your own tests or from the community

- Wanted to see if we can take the information that can be found about machines:
***frequency options - F | power - p(F) | HEPScore - s(F)***
and simulate grid performance - especially with perceived benefits of running clocked down - and run weeks worth of work in minutes without affecting delivery of "Tier2" service provision

- A dataset (like UK National Grid ESO or German C02 Monitor ) is fed in to get an idea of real-time and forecasted carbon intensity. Estimates for power and Carbon Use are calculated per time-step, and produce metrics at the end of the simulation.

- I created a simulation framework while at the University of Glasgow to do this and I aim to try it at DESY.

# The Simulation Framework

### 1 Grid Carbon Intensity Data

### 2 DataCenter Inventory

- *What kind of machines are there*
- *How many of them there are*
- *when were they installed.*

### 3 Machine Schematics

- *(Frequency-dependent) Power measurements*
- *Benchmarks to determine how good at doing work a machine is*

### 4 Simulation

### 5 RESULTS

- ***To be then used to generate recommendations***

# Simple Simulation Schematic

## Simulation.py

1 ***Specify variable parameters of the simulation*** mainly:

- The number and type of nodes your cluster is made from (ampere, dell, grace)
- The amount of starting jobs and how many jobs are submitted per hour
- Maximum length of the simulation

## WorkerNode.py

2 ***Create different kinds of worker nodes***

- Different types of worker node Attributes like hostnames, cores, memory, max power consumed, frequency
- Formulas for scaling power consumption
- Methods for automatically clocking up and down nodes
- Updates with whether the job is finished per timestep

## JobFactory.py

3 ***Create different kinds of jobs from different VO's***

- Assume jobs run for samples amount of time drawn from previously measured distributions (for testing all jobs are set to be 5hrs long)
- Require amounts of memory and cores to be used

## JobScheduler.py

4 ***Create a programme of work to be run on a cluster***

- Initialises jobs from ones requested from types of ones available
- Updates with jobs to be submitted to the cluster per time-step

## Cluster.py

5 ***Spins up a cluster to run specified workloads***

- Defines things like amount of memory, cores available to outside sources from input worker nodes
- Define how you run the cluster in the event you want to try and run it differently - clock down nodes at certain times of day for example

6 ***Run Simulation***

- Calculates the total power used and CO2e emitted per timestep (10 minutes)
- Takes Jobs from the scheduler if able
- Passes data from the worker nodes to the DataLogger
- Ends when you run out of work, or out of time

## DataLogger.py

7 ***Formats output statistics***

- Total (and average): CPU used, time elapsed, jobs started/completed, (peaktime) power used and estimated C02e emissions.

# Current Output

## DataLogger.py

7 ***Formats output statistics***

- Total (and average): CPU used, time elapsed, jobs started/completed, (peaktime) power used and estimated $CO_2$e emissions.

```
========
Summary
========

Total Simulated-time Duration      :  5.4 days
Total Real-time Duration           : 10.2 minutes

Jobs Started                       :  50000
Jobs Finished                      :  50000

Total CPU duration                 : 2000000.0 hours
Average CPU duration               : 5.00 hours

Total energy consumed by compute   : 1428.75 kWh
Peaktime (5-9pm) energy consumption: 256.65 kWh
Average energy consumption per job : 28.57 Wh

Estimated CO2e emissions           : 112.188 kg
Estimated Peaktime CO2e emissions  : 21.009 kg
Average CO2e emissions per job     : 2.244 g
Peaktime CO2e emissions percentage: 18.726 %
```

- Data here is an example from when I ran on the Glasgow Data-centre
- Each time the simulation is called, a file gets produced with the following information

**Simulated and Real-time duration of the simulation**

**Job information**

**Total and Average CPU duration**

**Estimated energy used in total, during peak times and job-average**

**Estimated $CO_2$ (e)quivalent emissions for said work**

# What do different procurements look like?

- An example type of recommendation

**No Changes (2022 Running)**

```
Total Simulated-time Duration       : 27.8 hours
Total Real-time Duration            :  1.0 minutes

Jobs Started                        : 50000
Jobs Finished                       : 50000

Total CPU duration                  : 250451.5 hours
Average CPU duration                : 5.01 hours

Total energy consumed by compute    : 1362.10 kWh
Peaktime (5-9pm) energy consumption: 292.48 kWh
Average energy consumption per job : 27.24 Wh

Estimated CO2e emissions            : 94.188 kg
Estimated Peaktime CO2e emissions : 19.462 kg
Average CO2e emissions per job      : 1.884 g
Peaktime CO2e emissions percentage: 20.663 %
```

**Replacing Older Nodes w/Sienna**

```
Total Simulated-time Duration       : 20.0 hours
Total Real-time Duration            :  0.6 minutes

Jobs Started                        : 50000
Jobs Finished                       : 50000

Total CPU duration                  : 259273.7 hours
Average CPU duration                : 5.19 hours

Total energy consumed by compute    : 969.80 kWh
Peaktime (5-9pm) energy consumption: 211.61 kWh
Average energy consumption per job : 19.40 Wh

Estimated CO2e emissions            : 66.048 kg
Estimated Peaktime CO2e emissions : 13.810 kg
Average CO2e emissions per job      : 1.321 g
Peaktime CO2e emissions percentage: 20.909 %
```

**Replacing older nodes w/AltraMax**

```
Total Simulated-time Duration       : 18.0 hours
Total Real-time Duration            :  0.5 minutes

Jobs Started                        : 50000
Jobs Finished                       : 50000

Total CPU duration                  : 252801.8 hours
Average CPU duration                : 5.06 hours

Total energy consumed by compute    : 939.53 kWh
Peaktime (5-9pm) energy consumption: 217.55 kWh
Average energy consumption per job : 18.79 Wh

Estimated CO2e emissions            : 63.599 kg
Estimated Peaktime CO2e emissions : 14.197 kg
Average CO2e emissions per job      : 1.272 g
Peaktime CO2e emissions percentage: 22.323 %
```

# Embedded Carbon Discussion

- The improvements listed are only on the carbon opportunity cost of **RUNNING** work. Assume an total operational carbon cost of Y.

- A significant component of carbon in a servers lifetime is in the embedded carbon. How we account for it will change the significance conclusions we have.

- Estimates of embedded carbon range from from 50-50 to 20-80 with operation costs.

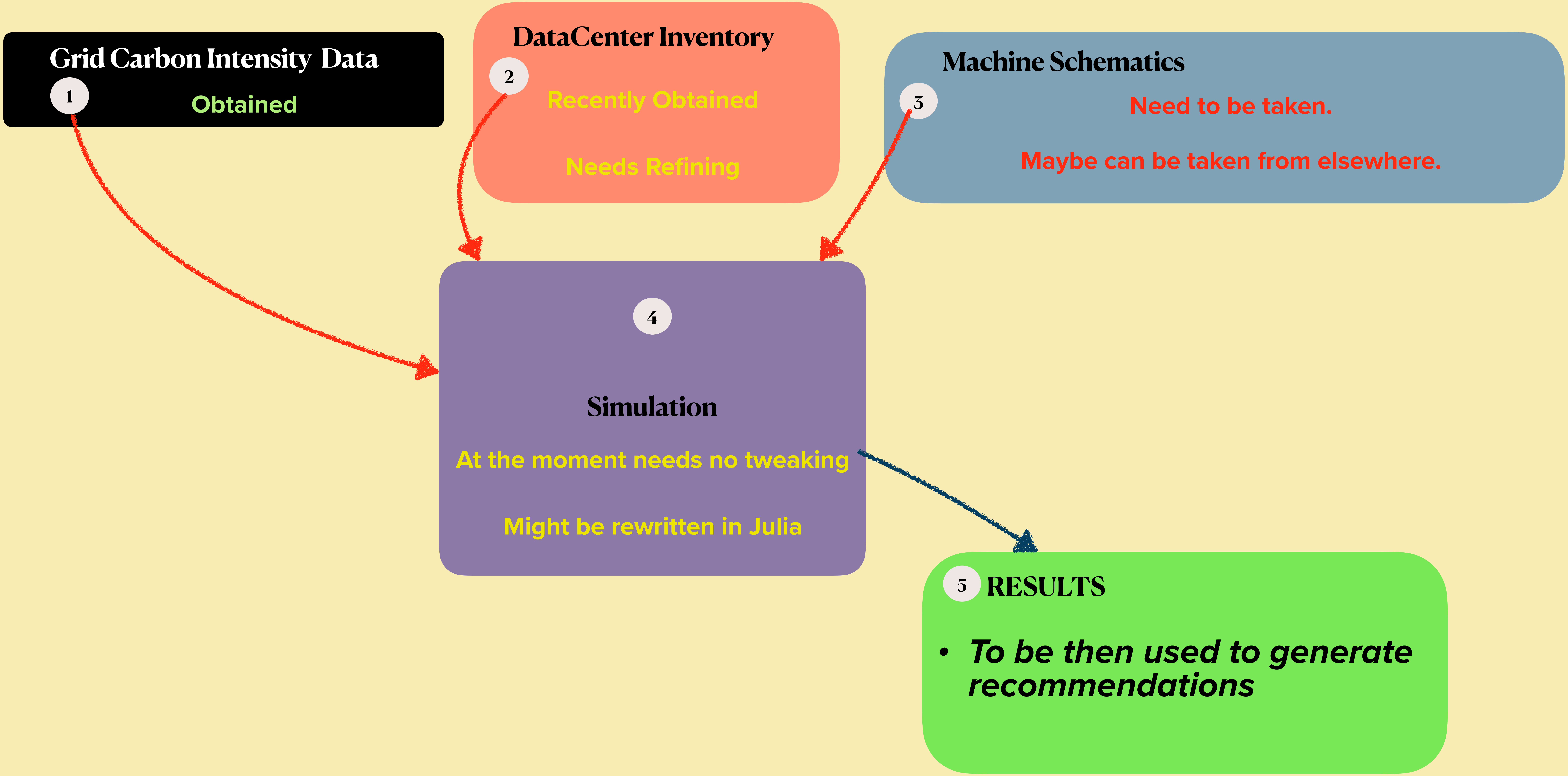
- If a machine we purchase has an embedded carbon cost of X. Do we

  - Attribute it all to purchase and treat operational carbon as independent?
    Total Carbon 2025 = Y(2025) -> Run in a way that reduces carbon

  - Assume a set lifetime of operation (5 years) and split the cost for each year - X/5?
    Total Carbon 2025 = X/5 + Y(2025) -> Optimisations of Y(2025) less impactful

  - Split the embedded carbon cost over every job you run?
    Total Carbon 2025 = X(2025) + Y(2025) -> Reduction in Jobs wastes embedded carbon

# The Simulation Framework - DESY Progress



# Conclusions and Future Work

- A simulation has been created to try and test different kinds of operation of Tier2 sites. It's modular, so different types and amounts of machines can be span up and run.

- Improvements will be tempered by how we treat embedded carbon in the future.

- The team working on this will be expanding shortly, with some students and hopefully one new FTE.

- Work will continue at DESY and hopefully the first provisional results will come out soon!