Carbon-Aware computing
Reducing the carbon intensity of software

Anders Lybecker | @Lybecker

I am a principal engineering lead at Microsoft

Vestas

Challenges

- Wind Turbine Simulation
  - Optimal placement of turbines in windfarm
  - Structural optimizations
- Existing on-premise HPC cluster
  - Expensive to maintain
  - Industry growth at 300%
  - Complexity of turbine

Requirements Destilled

Scale to 400K cores

Inexpensive

As green as possible


The IT industry emits more
carbon than the aviation industry

What can I as a software engineer do?

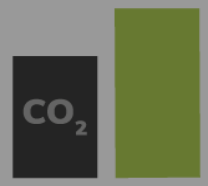
Table 4. Normalized global results for Energy, Time, and Memory

Total					
	Energy		Time		Mb
(c) C	1.00	(c) C	1.00	(c) Pascal	1.00
(c) Rust	1.03	(c) Rust	1.04	(c) Go	1.05
(c) C++	1.34	(c) C++	1.56	(c) C	1.17
(c) Ada	1.70	(c) Ada	1.85	(c) Fortran	1.24
(v) Java	1.98	(v) Java	1.89	(c) C++	1.34
(c) Pascal	2.14	(c) Chapel	2.14	(c) Ada	1.47
(c) Chapel	2.18	(c) Go	2.83	(c) Rust	1.54
(v) Lisp	2.27	(c) Pascal	3.02	(v) Lisp	1.92
(c) Ocaml	2.40	(c) Ocaml	3.09	(c) Haskell	2.45
(c) Fortran	2.52	(v) C#	3.14	(i) PHP	2.57
(c) Swift	2.79	(v) Lisp	3.40	(c) Swift	2.71
(c) Haskell	3.10	(c) Haskell	3.55	(i) Python	2.80
(v) C#	3.14	(c) Swift	4.20	(c) Ocaml	2.82
(c) Go	3.23	(c) Fortran	4.20	(v) C#	2.85
(i) Dart	3.83	(v) F#	6.30	(i) Hack	3.34
(v) F#	4.13	(i) JavaScript	6.52	(v) Racket	3.52
(i) JavaScript	4.45	(i) Dart	6.67	(i) Ruby	3.97
(v) Racket	7.91	(v) Racket	11.27	(c) Chapel	4.00
(i) TypeScript	21.50	(i) Hack	26.99	(v) F#	4.25
(i) Hack	24.02	(i) PHP	27.64	(i) JavaScript	4.59
(i) PHP	29.30	(v) Erlang	36.71	(i) TypeScript	4.69
(v) Erlang	42.23	(i) Jruby	43.44	(v) Java	6.01
(i) Lua	45.98	(i) TypeScript	46.20	(i) Perl	6.62
(i) Jruby	46.54	(i) Ruby	59.34	(i) Lua	6.72
(i) Ruby	69.91	(i) Perl	65.79	(v) Erlang	7.20
(i) Python	75.88	(i) Python	71.90	(i) Dart	8.64
(i) Perl	79.58	(i) Lua	82.91	(i) Jruby	19.84

Gr ...s

Energy Efficiency

Consume the least amou...
of electricity possible



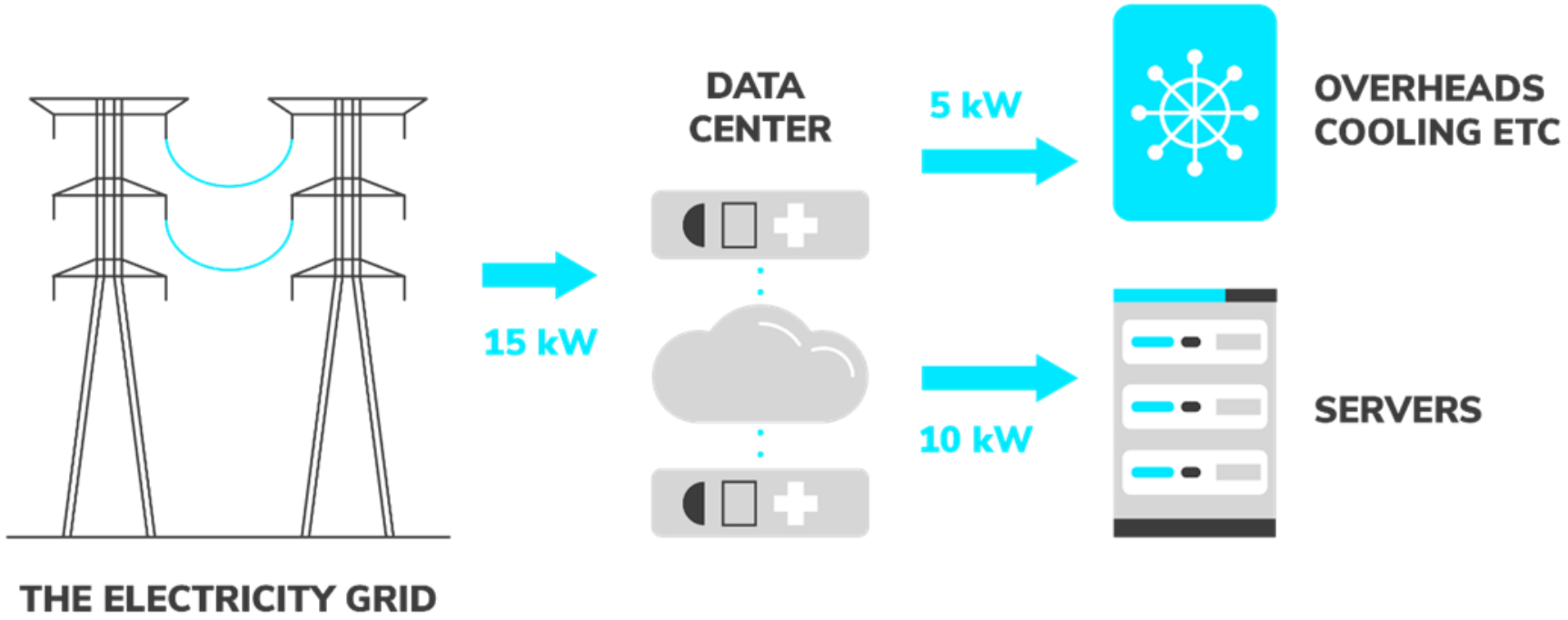
Carbon Awareness

...more when the electricity is
...an and less when it's dirty

Green Software Foundation

greensoftware.org

Power Usage Effectiveness



PUE 1.5

Design Decisions

Saturate CPU	✓	✓
Rightsizing VM	✓	✓
Auto scale up and down	✓	✓
Using unused VMs	✓	✓

Carbon Awareness

CO_2

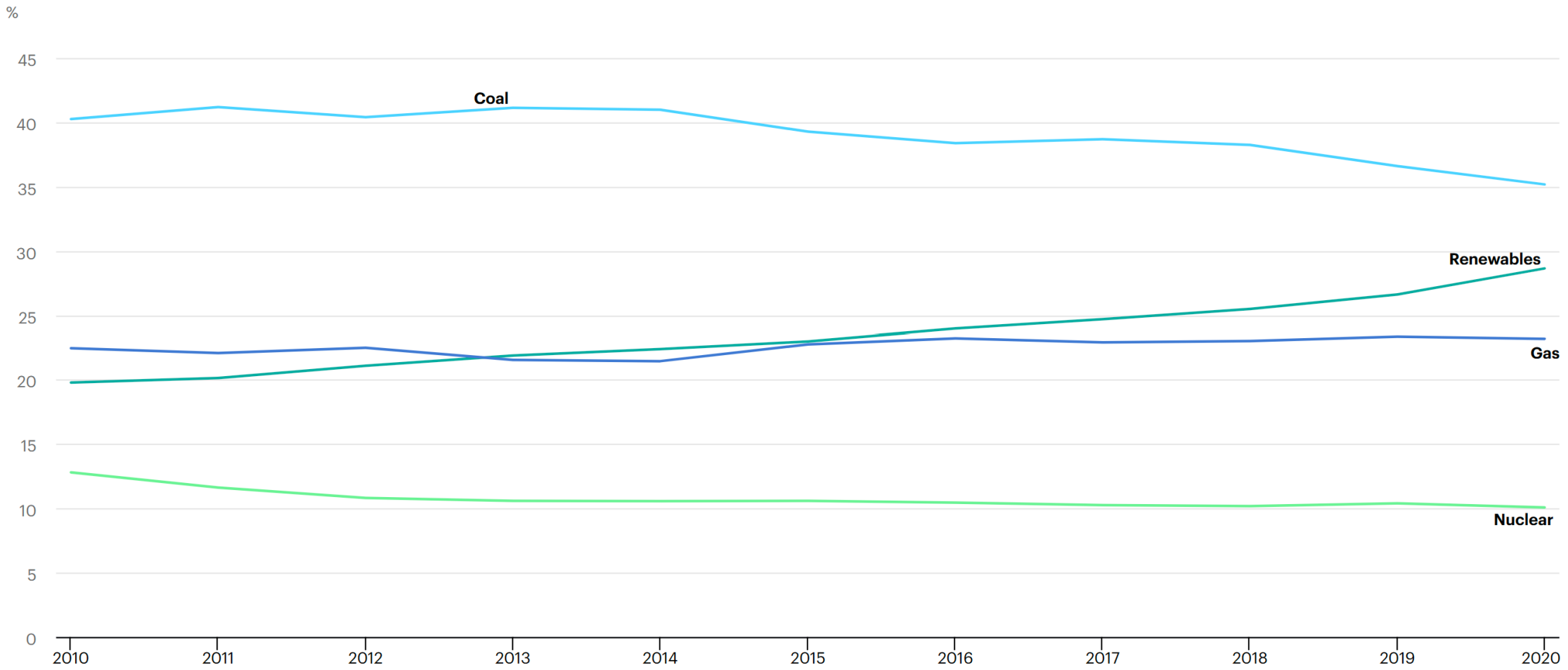
CO_2eq

Carbon intensity

CO_2eq/kWh

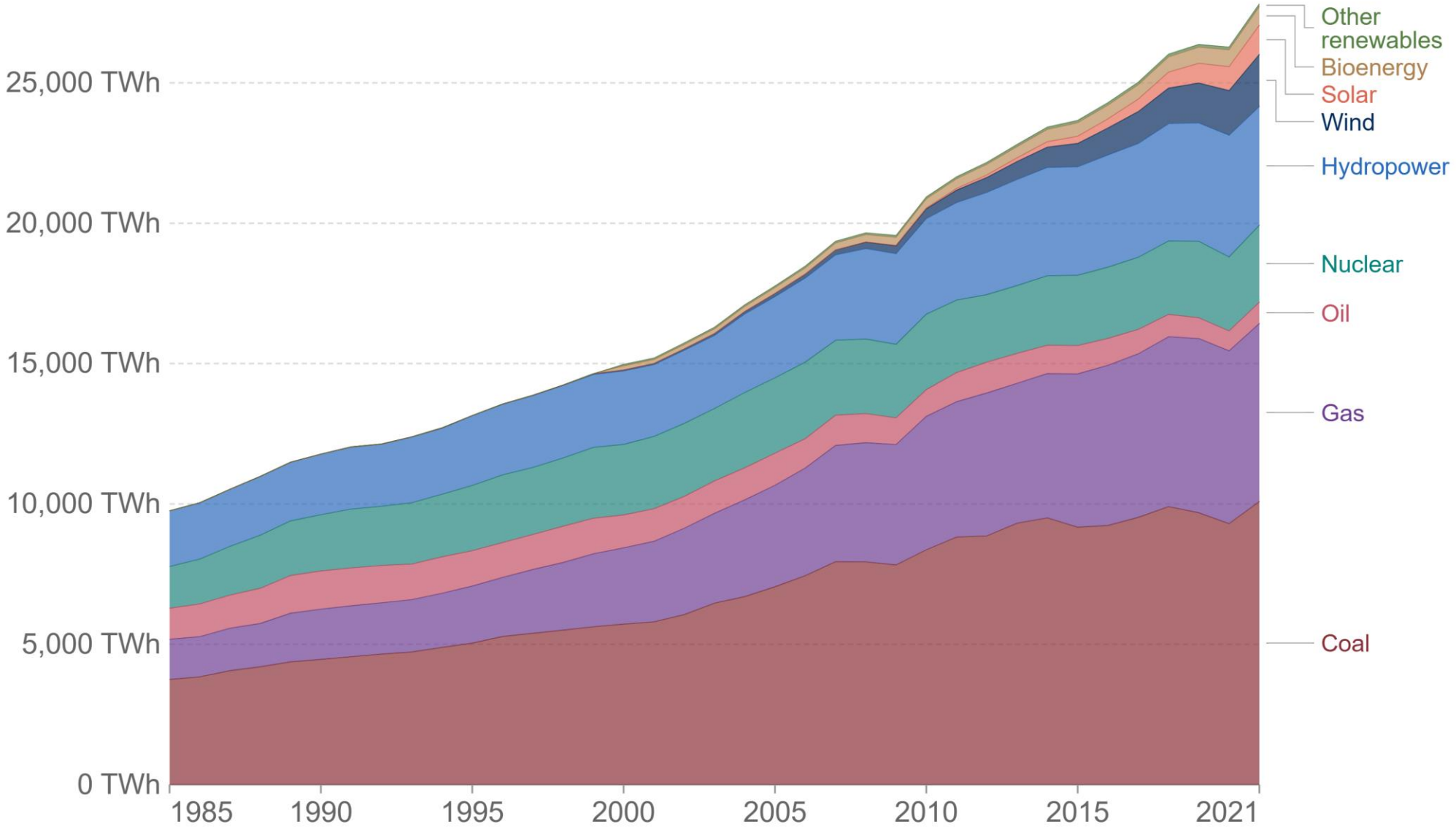
Electricity mix

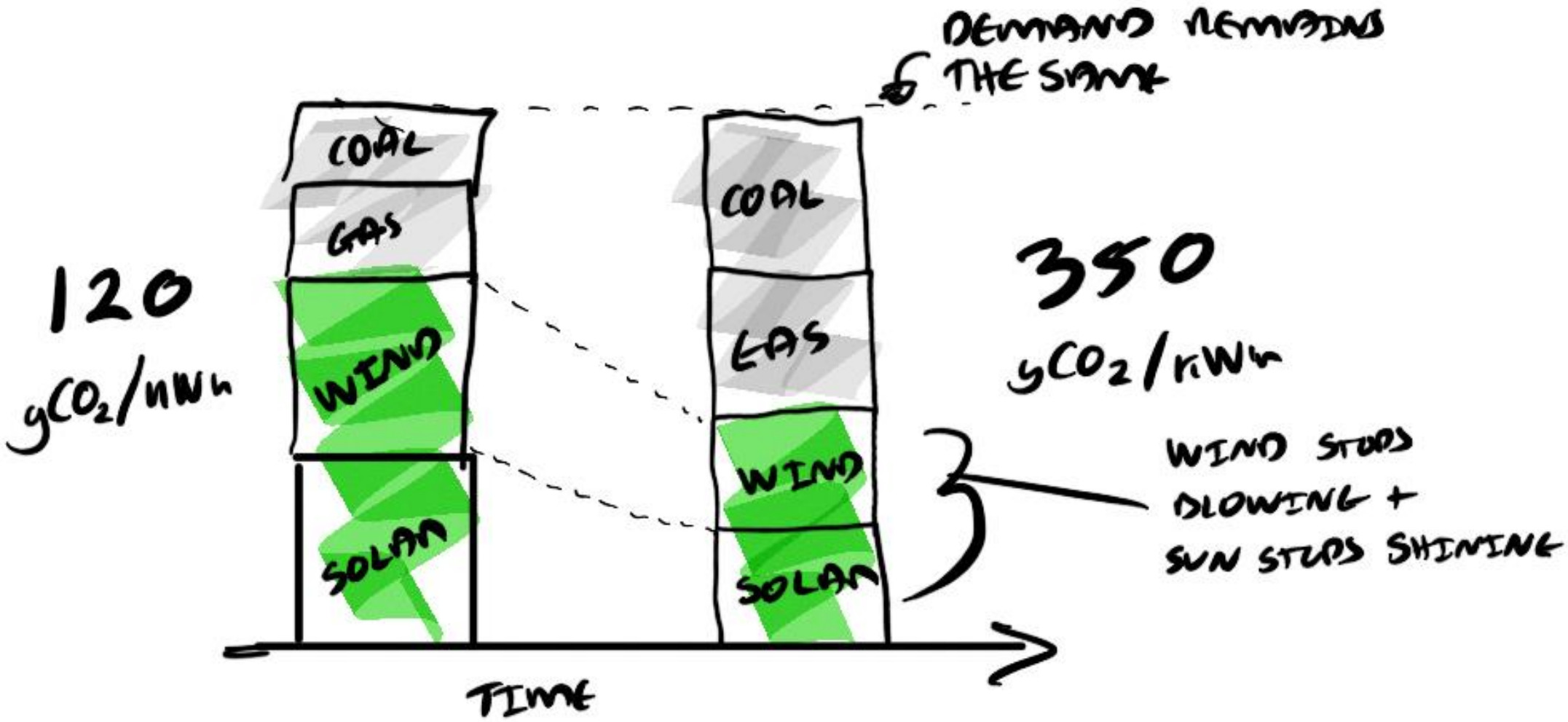
Electricity mix

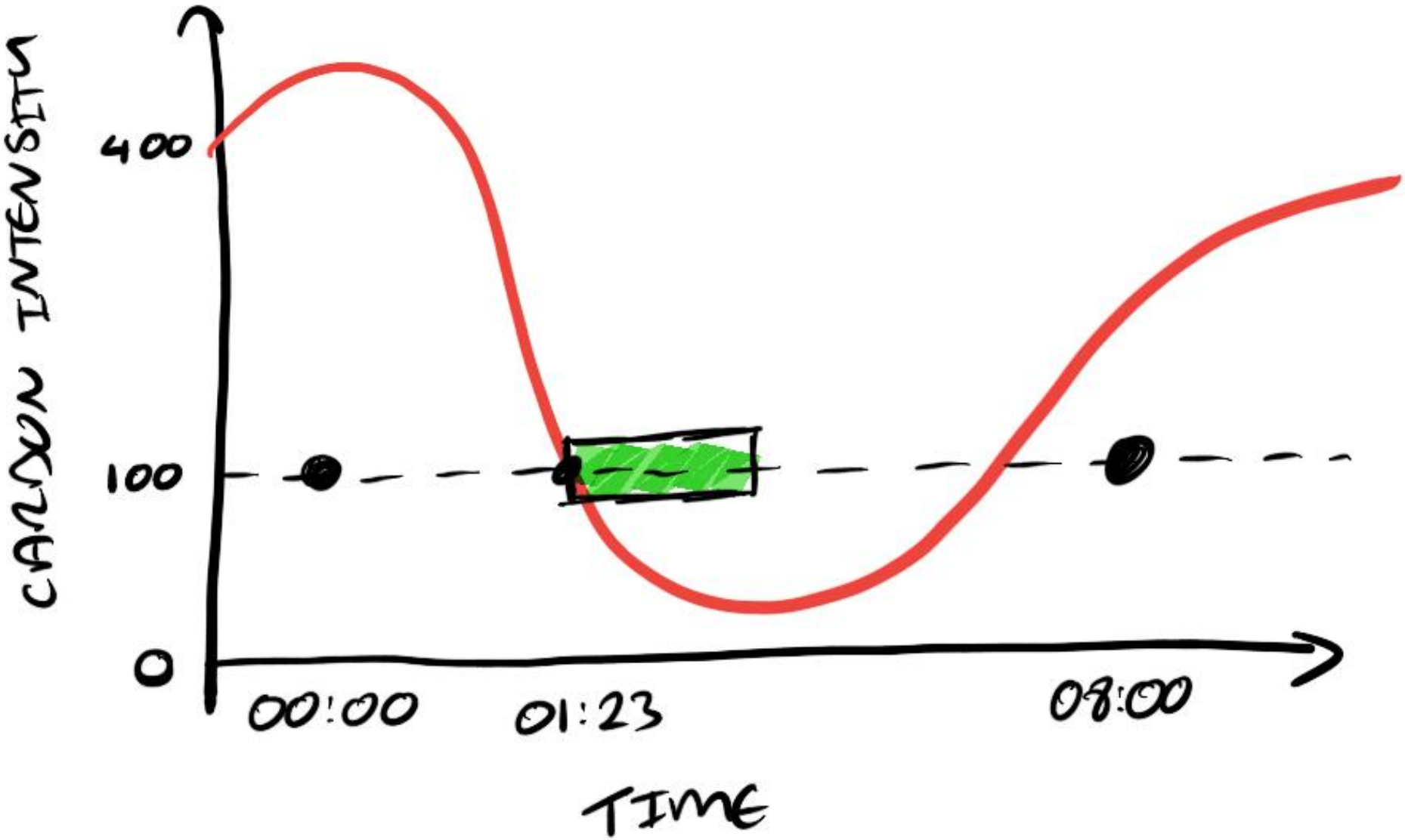


Source: International Energy Agency

Electricity mix




DEMAND REMAINS THE SAME
COAL
GAS
WIND
SOLAR
120
gCO2/kWh
COAL
GAS
WIND
SOLAR
350
gCO2/kWh
WIND STOPS BLOWING + SUN STOPS SHINING
TIME


CARBON INTENSITY
400
100
0
00:00
01:23
08:00
TIME

Netherlands

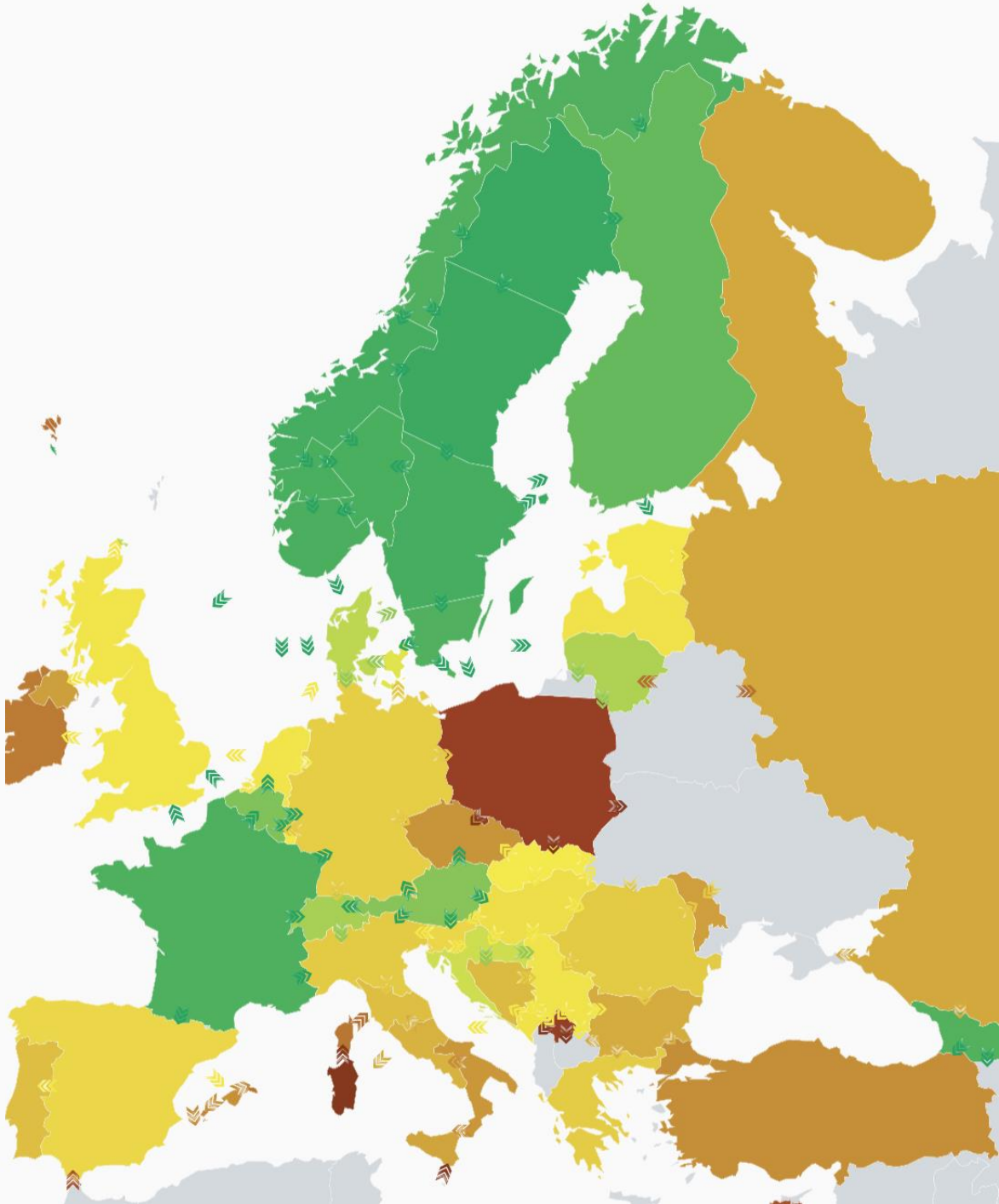
May 30, 2023, 10:00 AM

165 g	76%	71%
Carbon Intensity	Low-carbon	Renewable

South Central Sweden

May 30, 2023, 10:00 AM

22 g	100%	74%
Carbon Intensity	Low-carbon	Renewable



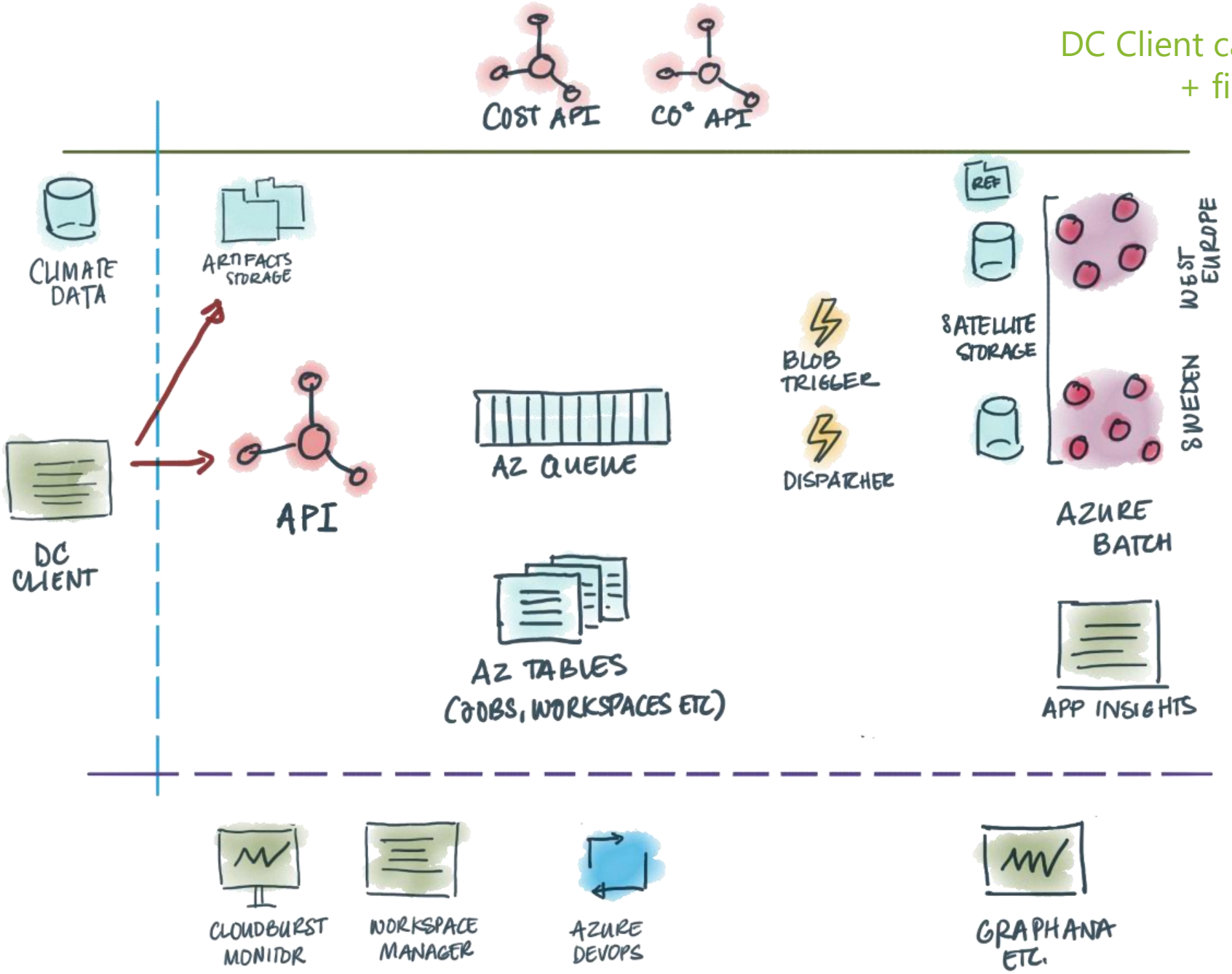
Design Decisions

Saturate CPU	✓	✓
Rightsizing VM	✓	✓
Auto scale up and down	✓	✓
Using unused VMs	✓	✓
Time shifting		✓
Location shifting	✓	✓

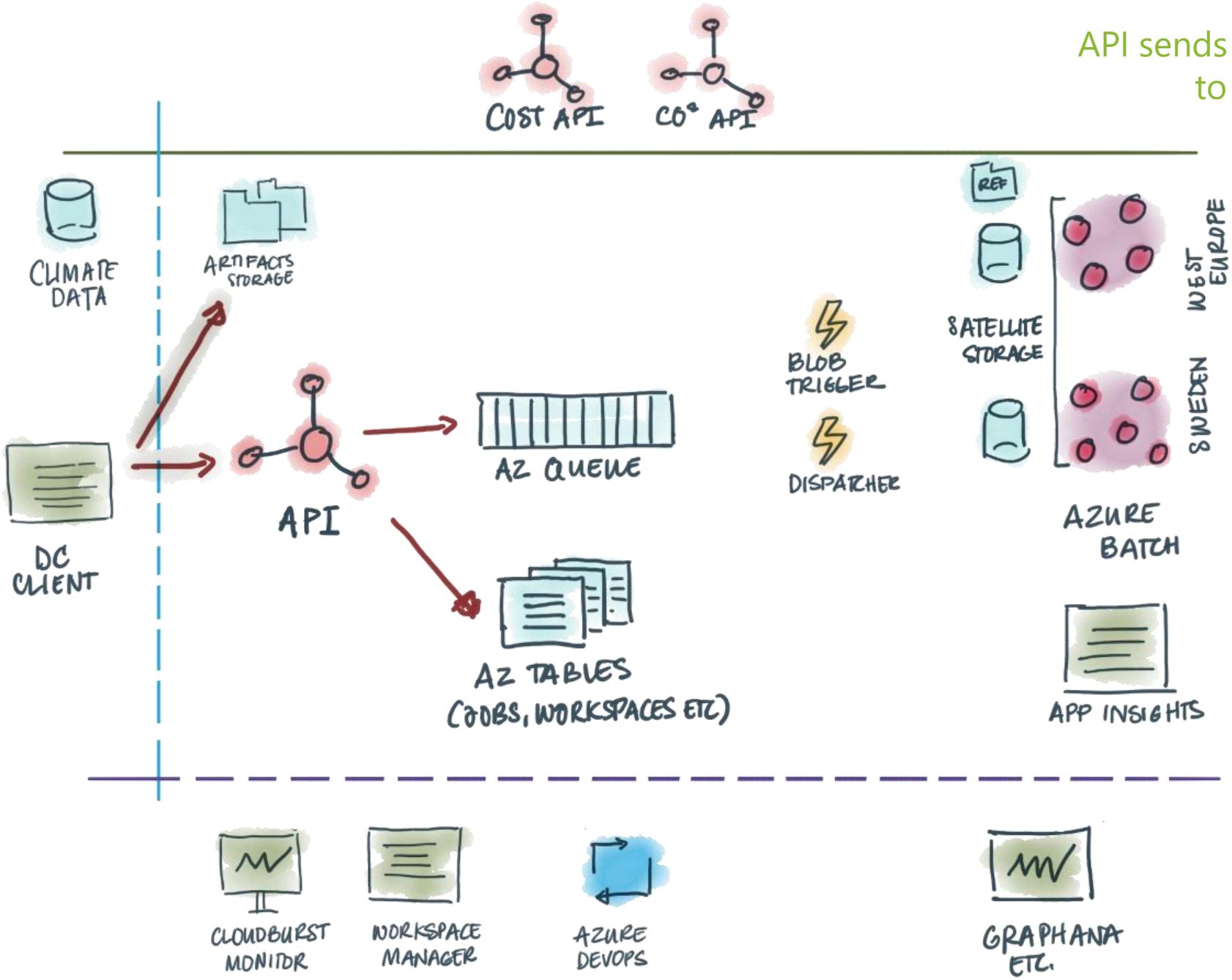
Time & location shifting

Coordination, data & time

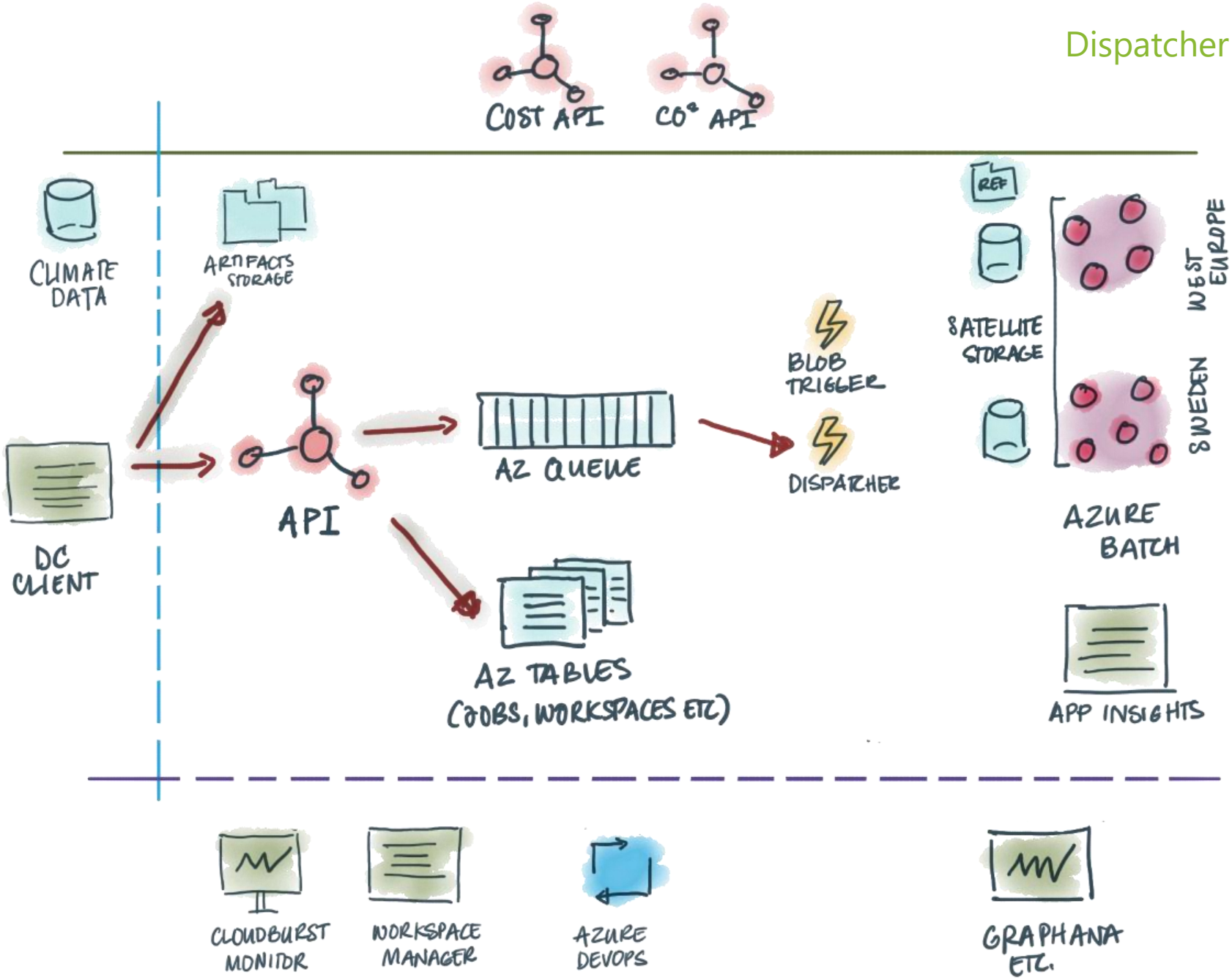
DC Client calls API with job spec
+ files in artifact storage



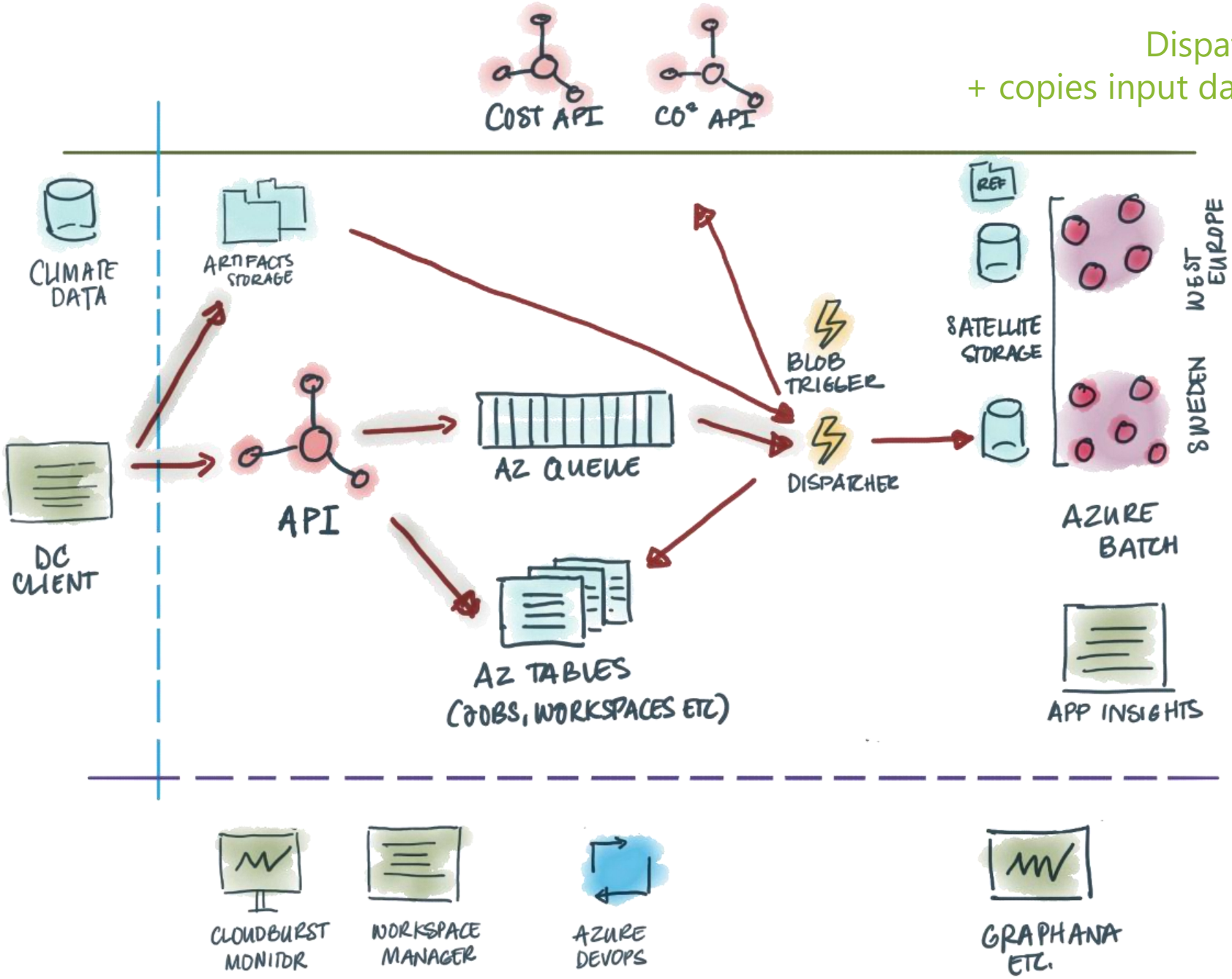
API sends a job create message to the dispatcher queue



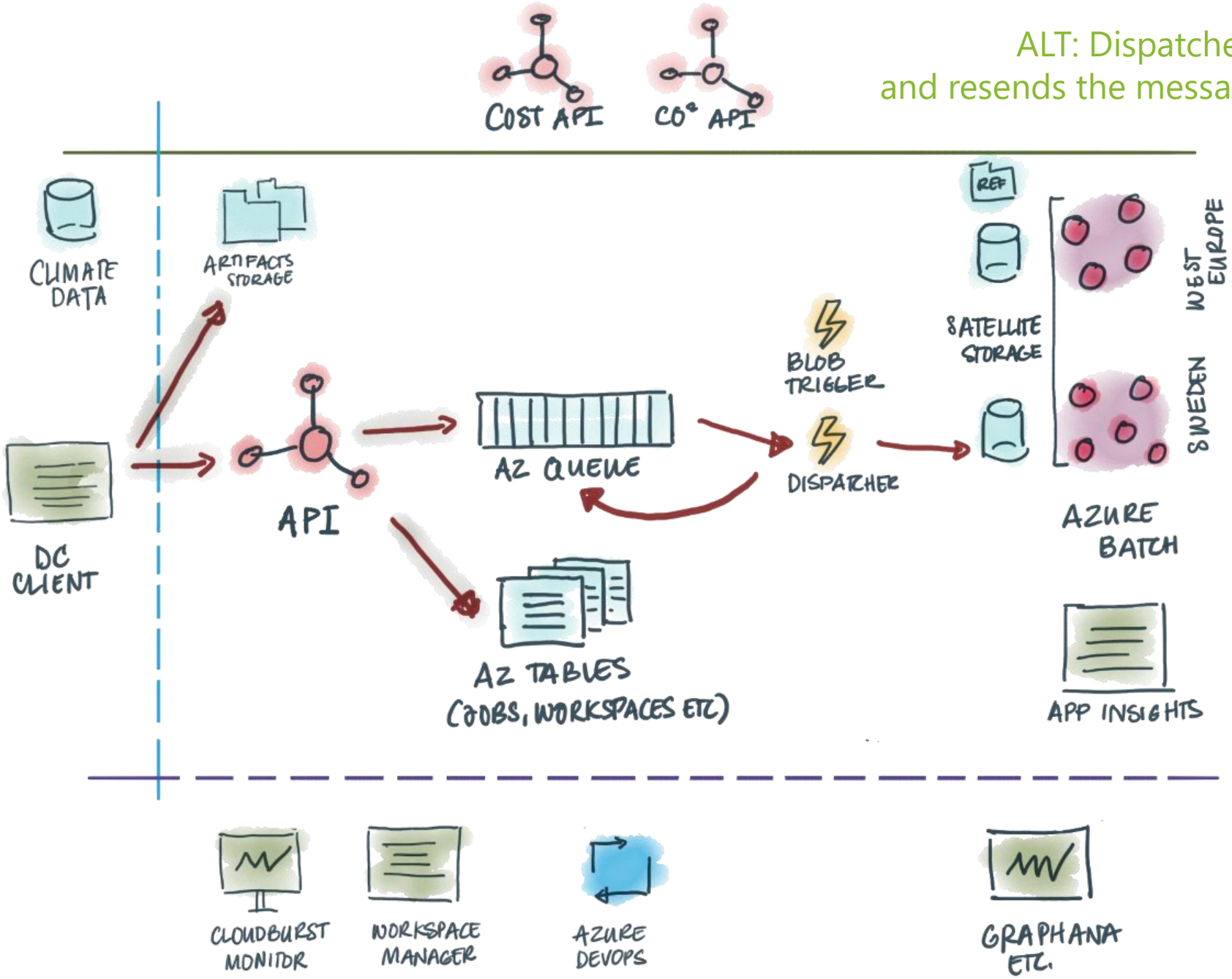
Dispatcher handles the message



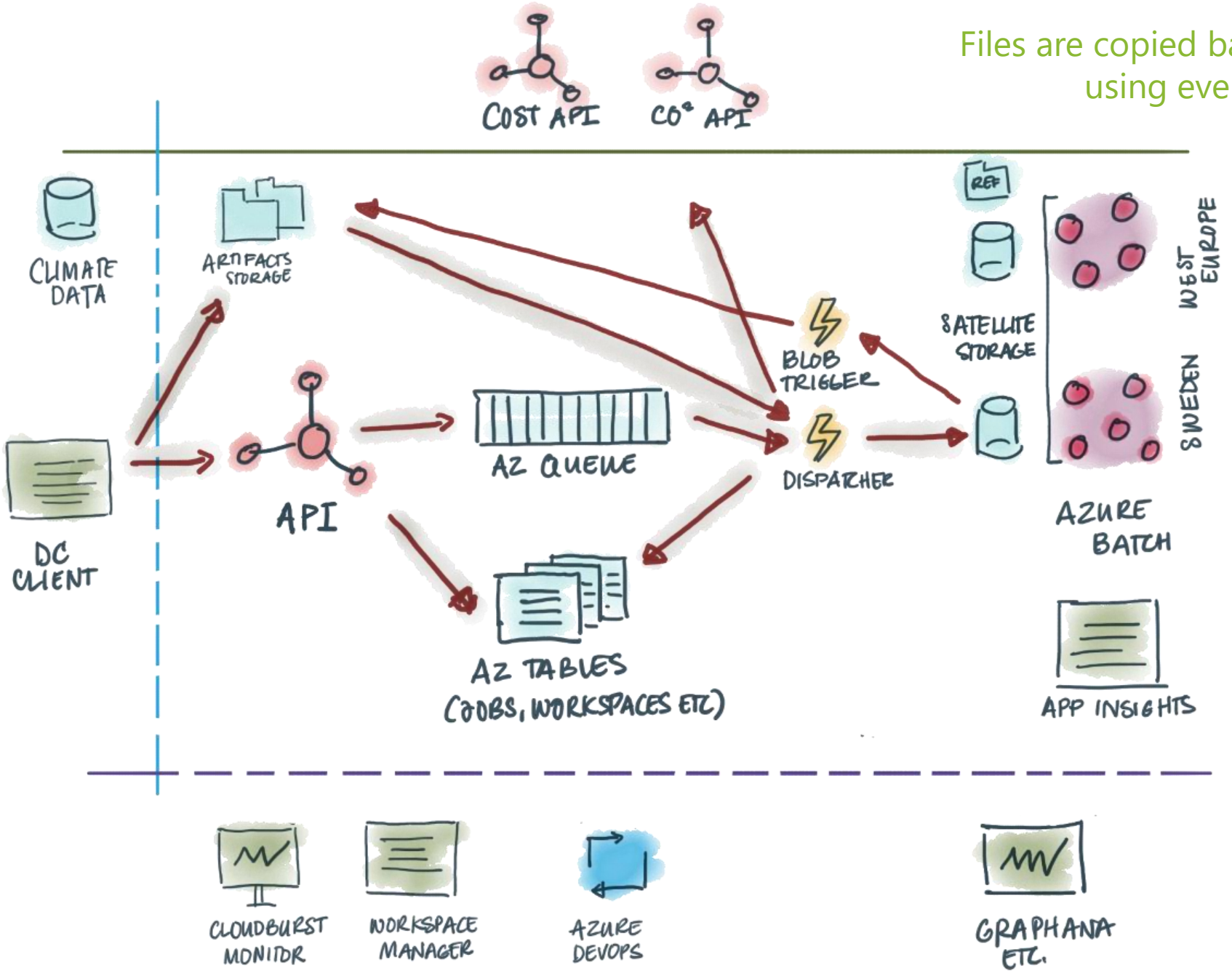
Dispatcher selects the pool
+ copies input data to satellite storage



ALT: Dispatcher needs to scale pool
and resends the message with a 1 min delay



Files are copied back on task complete using event grid (BlobCreated)



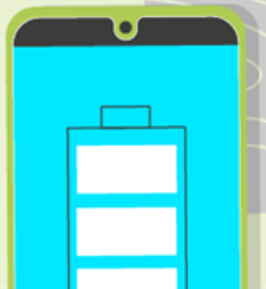

12-16% reduction with time shifting within a day
30-90% reduction with location shifting within EU

Tools

Green Software Foundation

Measuring:

What you can't measure, you can't improve



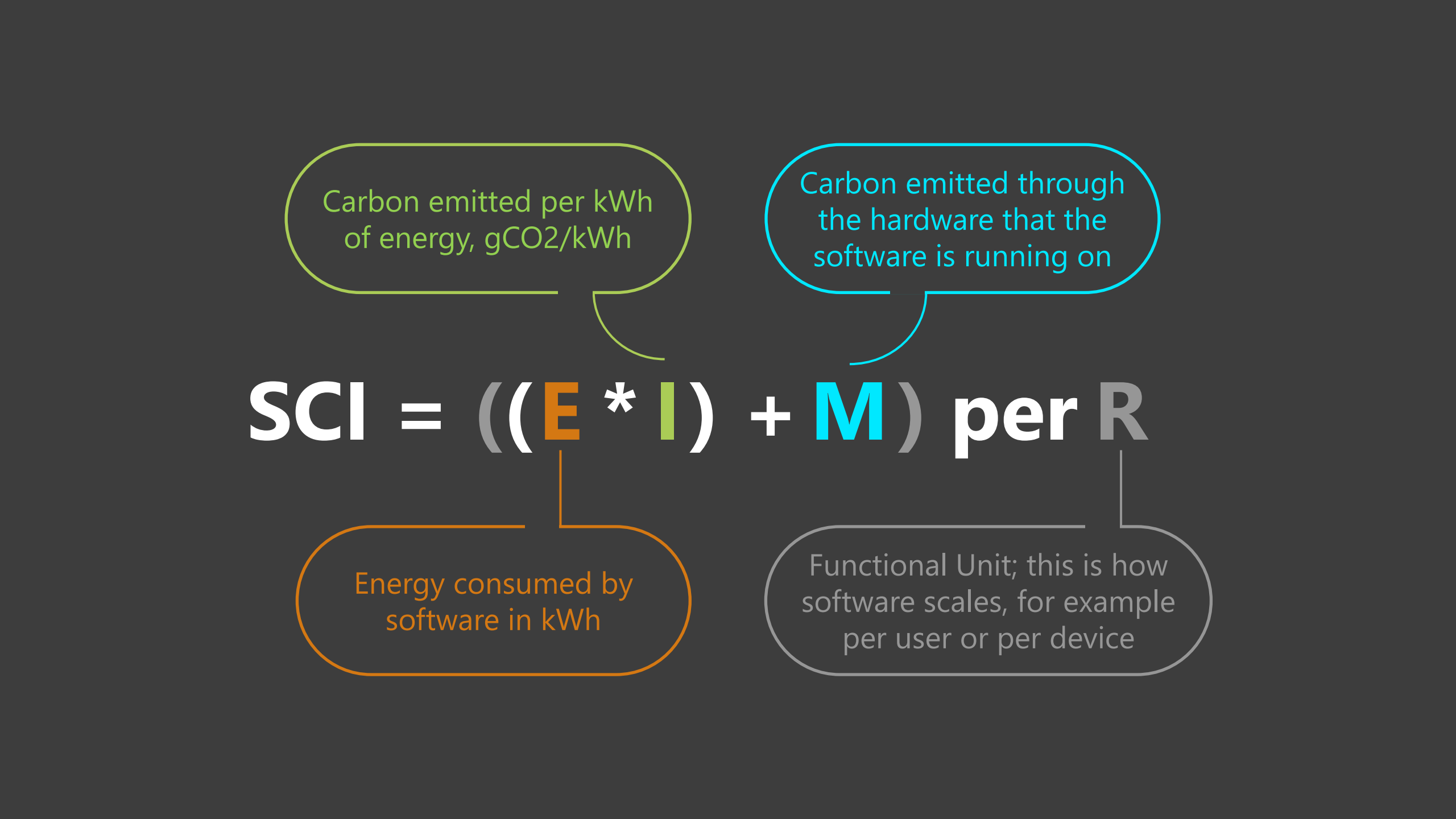
Software Carbon Intensity calculation

One hour carbon emissions

- 400 W server
  400 W/hour = 0.4 kWh
- 1000 kg embodied carbon for server with 4-year lifespan
  1000 / (4 * 365 * 24) = 28g CO2eq/hour
- 216 carbon intensity
- 398 orders

Dell PowerEdge R6515
- AMD EPYC 7313P 3.0GHz
- 16GB RDIMM
- 480GB SSD SATA

* not including cooling, network etc.


Carbon emitted per kWh of energy, gCO2/kWh
Carbon emitted through the hardware that the software is running on
SCI = ((E * I) + M) per R
Energy consumed by software in kWh
Functional Unit; this is how software scales, for example per user or per device

E 0.4 kWh server

I 216 carbon intensity

M 28 g embodied carbon

R **398 orders**

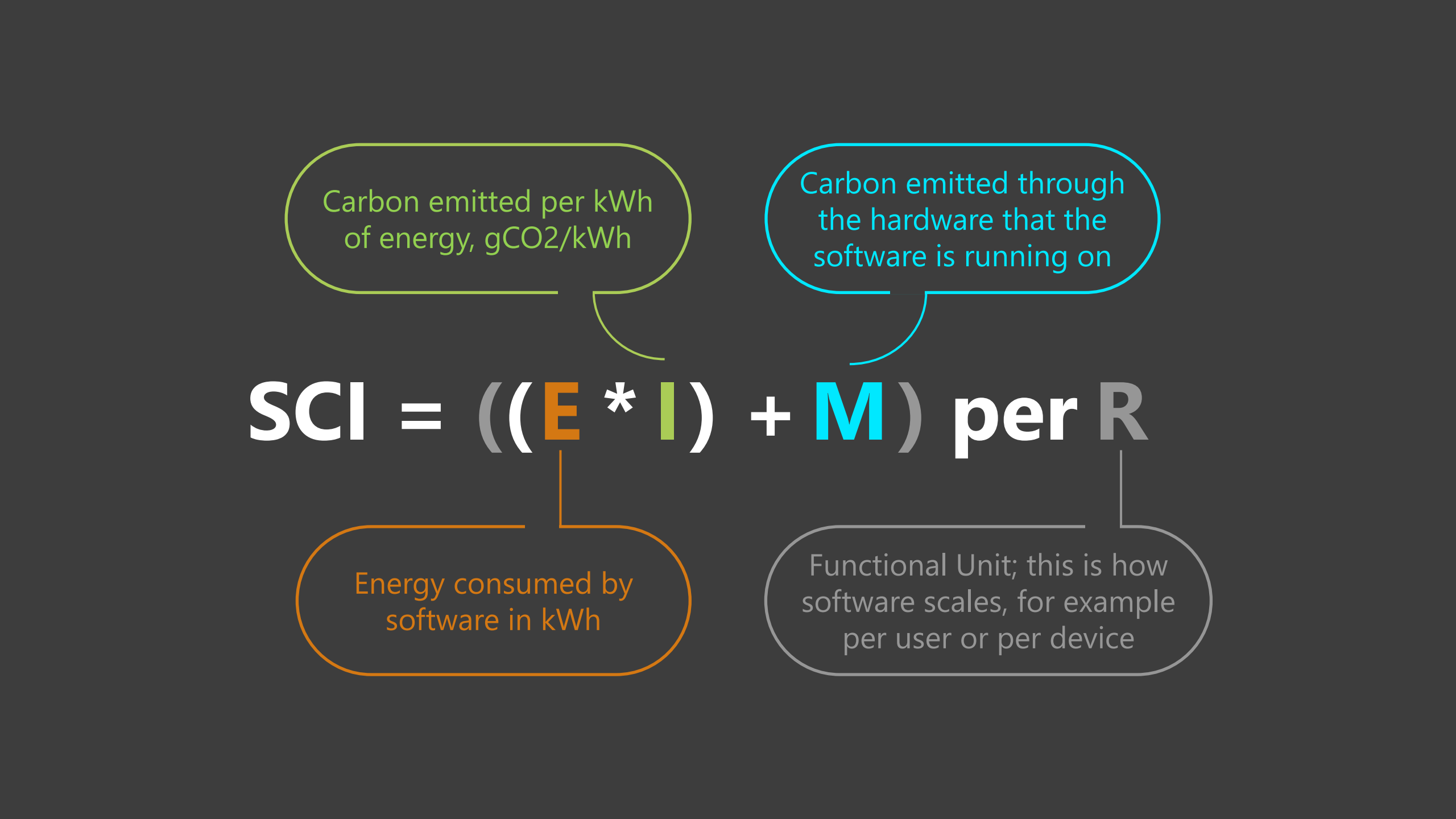
$$SCI = ((E * I) + M) \text{ per } R$$

SCI = ((0.4 *216) +28) per 368

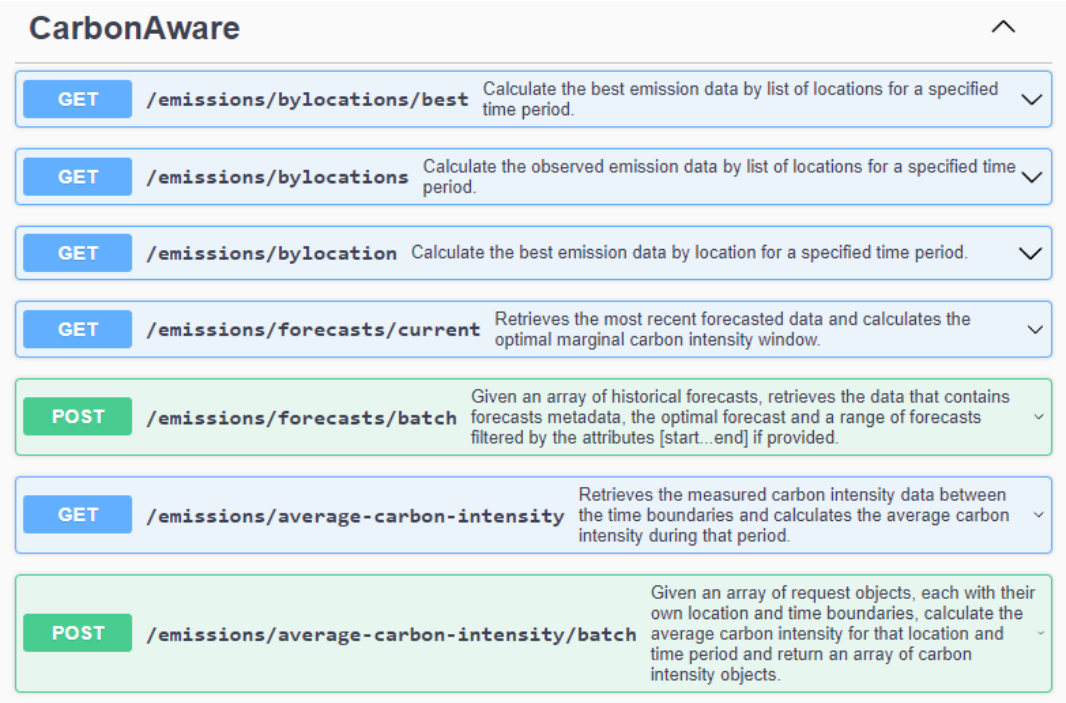
SCI = ((86.4) + 28) per 368

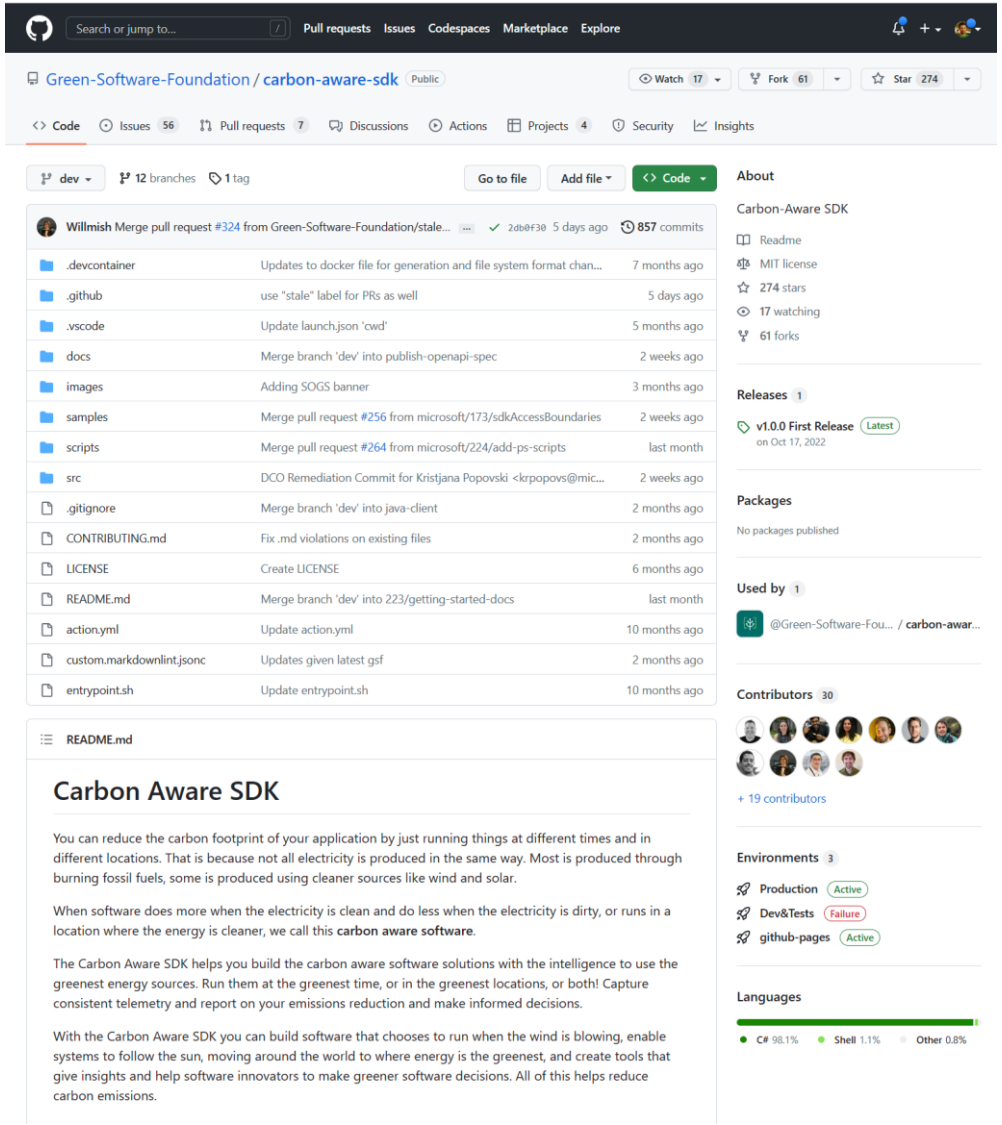
SCI = (114.4 g CO2) per 368

SCI = 0.31 g CO2/Order


Carbon emitted per kWh of energy, gCO2/kWh
Carbon emitted through the hardware that the software is running on
SCI = ((E * I) + M) per R
Energy consumed by software in kWh
Functional Unit; this is how software scales, for example per user or per device

Carbon Aware SDK

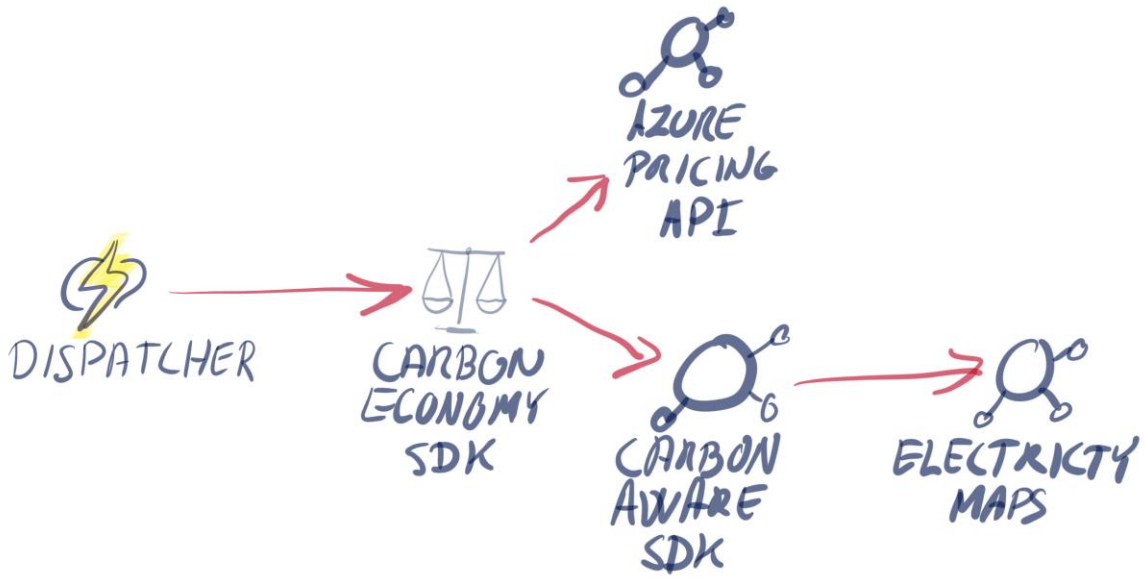


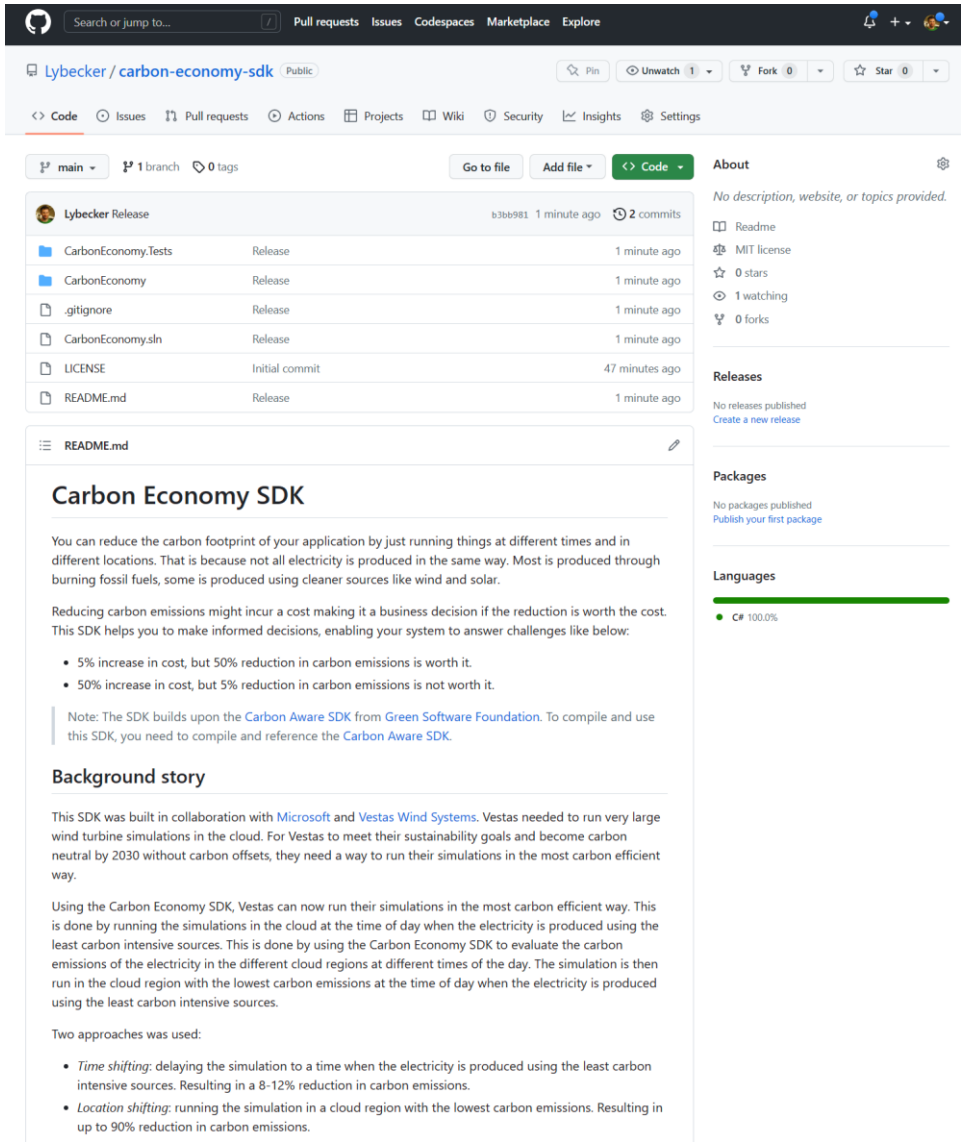


Pay 5% extra for 50% carbon reduction

Pay 50% extra for 5% carbon reduction

Carbon Economy SDK





Carbon Economy SDK Formula

$$\sqrt{((1 - w_e) * price)^2 + (w_e * emissions)^2}$$

Carbon Economy SDK Example

Emission weight: 0.25

Location	Emission rating	Unit price	Norm. rating	Norm. price	Calc weight
Sweden Central	29	$1.932	0	1	**0.750**
Norway East	42	$1.824	0.107	0.870	**0.653**
West Europe	100	$1.103	0.587	0	**0.147**
UK South	150	$1.713	1	0.736	**0.606**

Emission weight: 0.75

Location	Emission rating	Unit price	Norm. rating	Norm. price	Calc weight
Sweden Central	29	$1.932	0	1	**0.250**
Norway East	42	$1.824	0.107	0.870	**0.232**
West Europe	100	$1.103	0.587	0	**0.440**
UK South	150	$1.713	1	0.736	**0.772**

Vestas Green Management Dashboard

- General Settings
- User Settings
- Rulebook
- **Job Processing**

Settings > Job Processing

Choose the balance that works for you

How much should we optimize for greenness?

How much should we optimize for budget?

You are willing to pay 1% more for each 1% CO2 reduction

In the previous month, in only about **4%** of cases, the greener option was cheaper, or priced the same.

Month over Month Numbers

CO2 Reduction	Budget Spending	Queue Lenght	Avg. Execution Time
↗ 12.3%	↘ 9.3%	↘ 1%	↗ 5.4%

Global Settings

100 — **Maximum number of standard instances**
This number limits how many of standard nodes can run in the workspace, and everything else is waiting for Spot instances to become available.

50% — **Minimum discount for a spot instance**
You can set a minimum discount (default 50%) compared to a normal prices node

100 — **Maximum number of cores**
The maximum number of nodes in a Workspace across Azure Regions including spot instances.

YES — **Overule maximum number of standard instances to process High priority jobs**
The admin can set a number of full cost instances in the settings. However this number might get overrule in case high priority job gets in


Our future,
our responsibility
questions?