

10 Do's and Don'ts of sustainable systems design

01

Don't think your
application doesn't
doesn't make a
difference

Global carbon emissions by industry

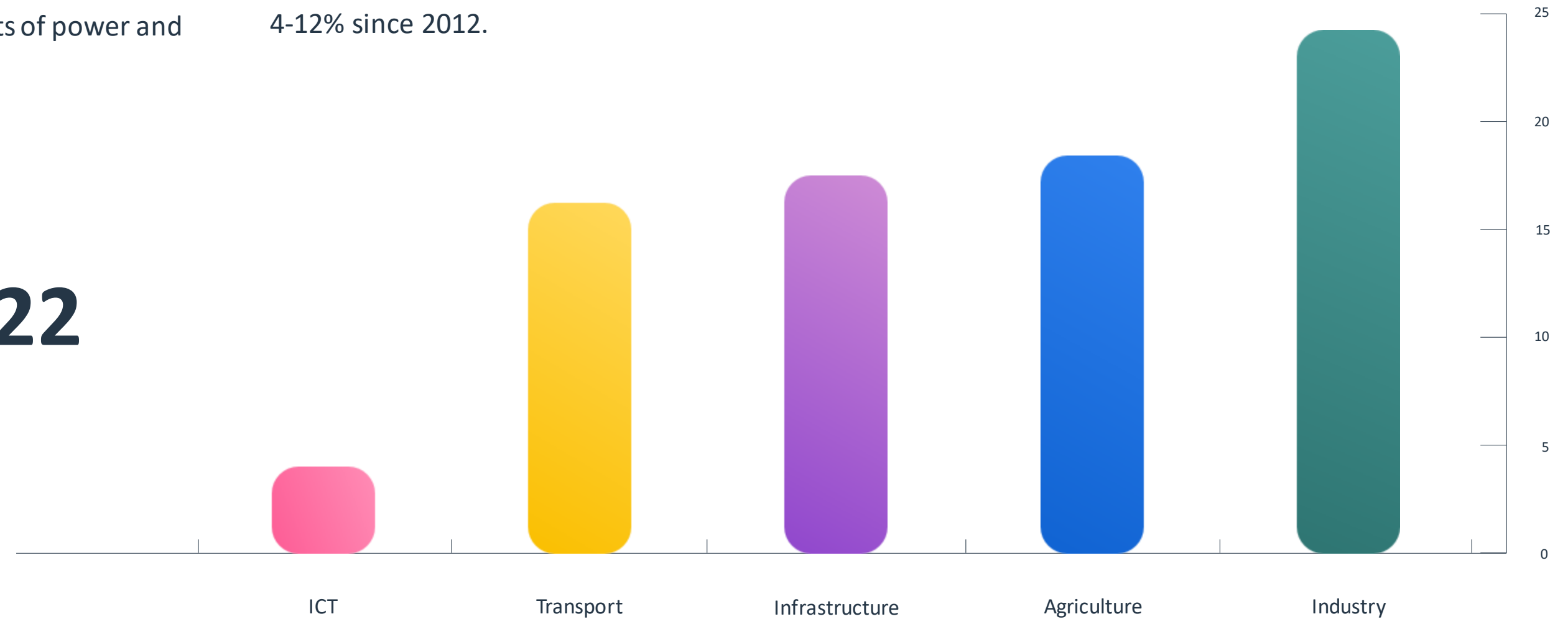
ICT (Information and Communications Technology) isn't the problem its traditional agriculture, industry, infrastructure and transports.

Data centres use huge amounts of power and water.

ICT makes up about 3-4% of global emissions. - Prof. Mike Berners-Lee, 2022

Internet users have continually increased, with year-on-year growth from between 4-12% since 2012.

2022



02

Do take a green approach to systems design as it's not a mutually exclusive benefit

Performance, cost and carbon

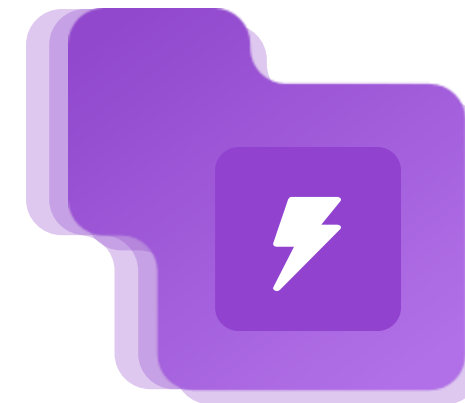
Performance increases, Cost Savings and reduction of carbon emissions are linked.



cost savings



sustainability



performance

03

Don't assume that your
website is carbon
neutral

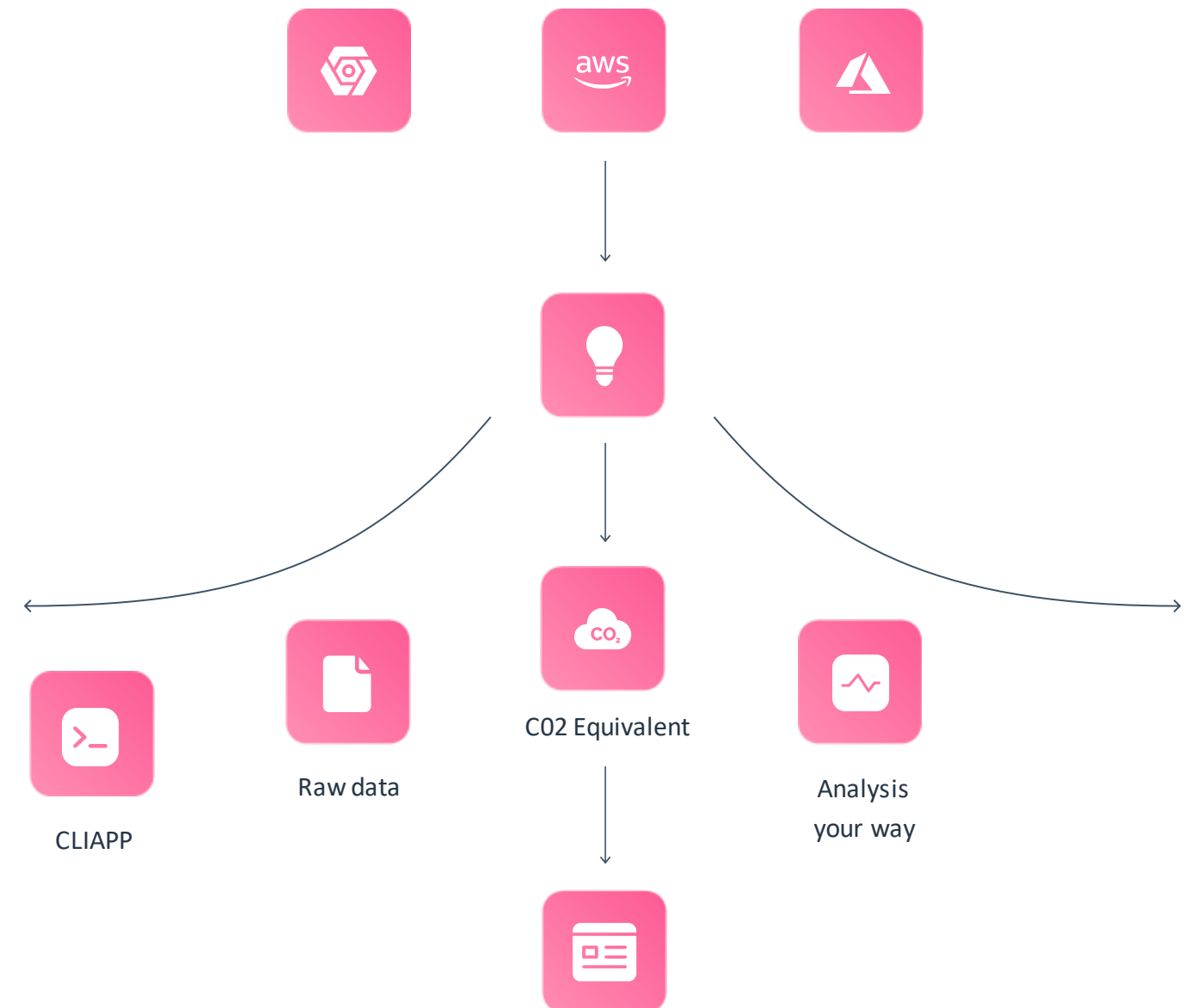
Reporting

How to report on carbon emissions for the major cloud providers.

Cloud Carbon Footprint



Cloud Carbon Footprint
<https://www.cloudcarbonfootprint.org/>



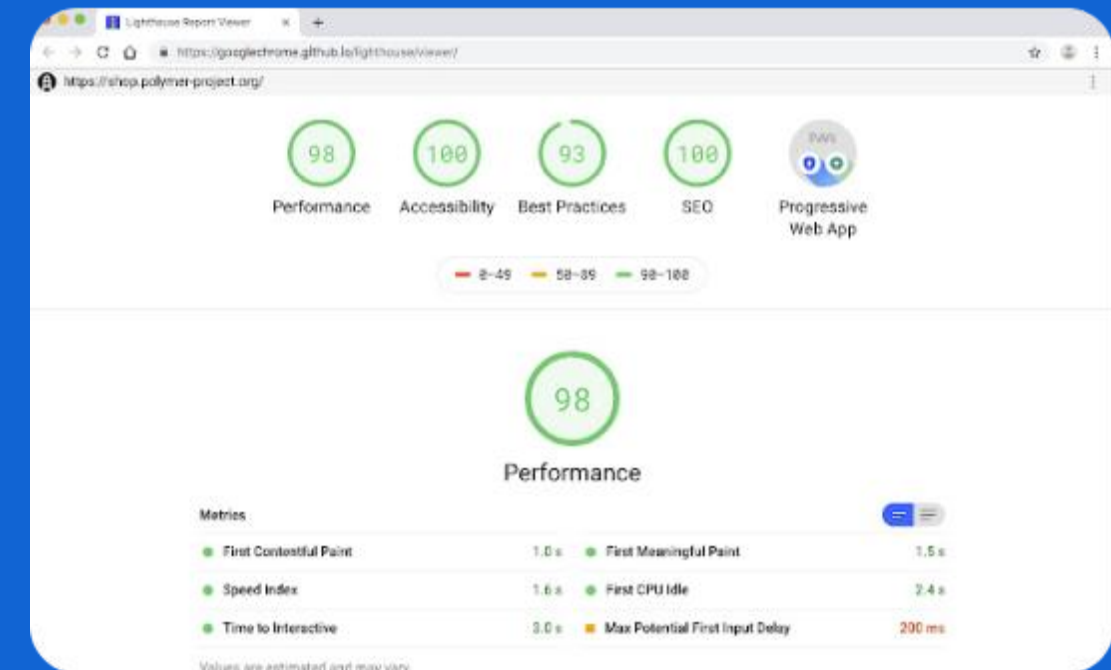
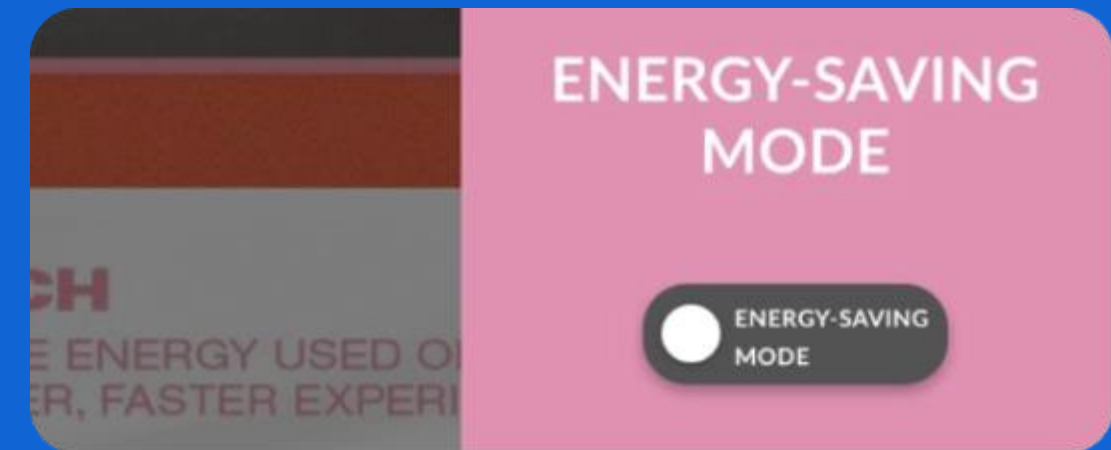
04

Do optimise the UI and
client-side scripts

UI & frontend optimisations

From a frontend perspective we should for example

- 01 ———— Take a mobile-first approach
- 02 ———— Use of carbon aware design
- 03 ———— Consider “energy-saving” toggles
- 04 ———— Optimise images (e.g. WebP or AVIF)
- 05 ———— Bundle and minimise all assets
- 06 ———— Optimise JavaScript
- 07 ———— Limit number of third-party libraries and fonts
- 08 ———— Turn off auto-play for video assets
- 09 ———— Evaluate load in dependencies on click events, rather than page load

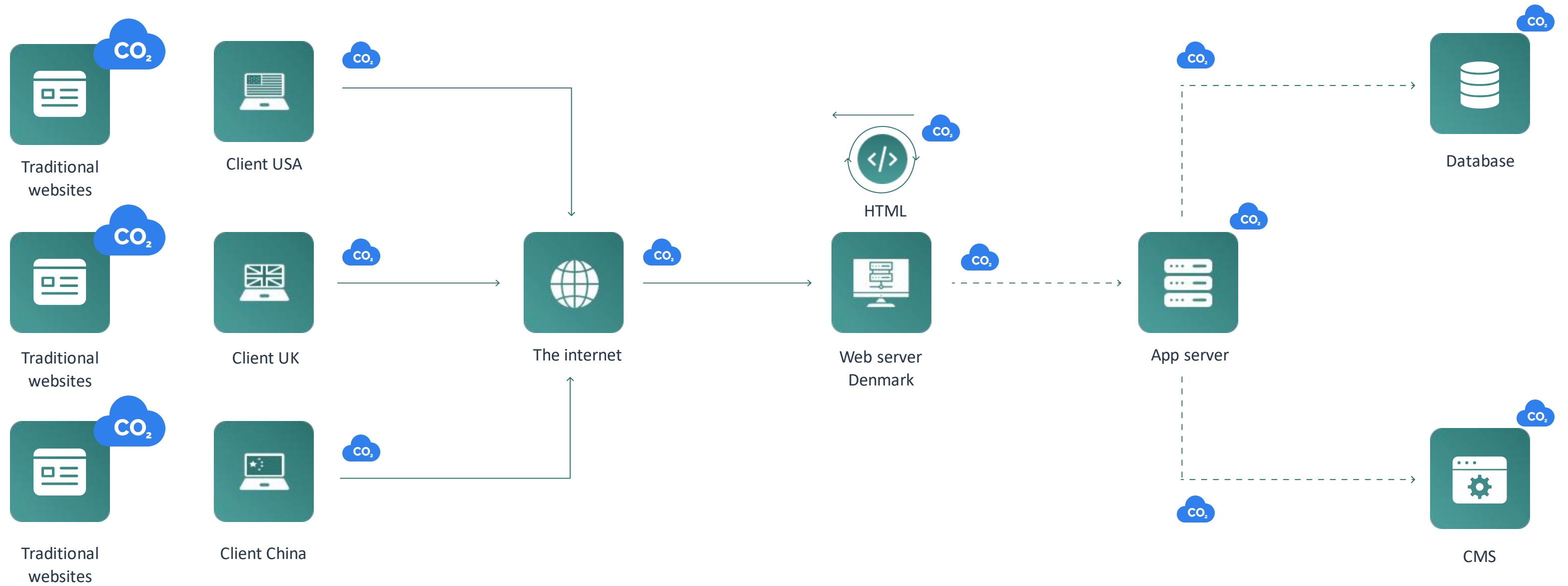


05

Don't ignore user data,
volume and location

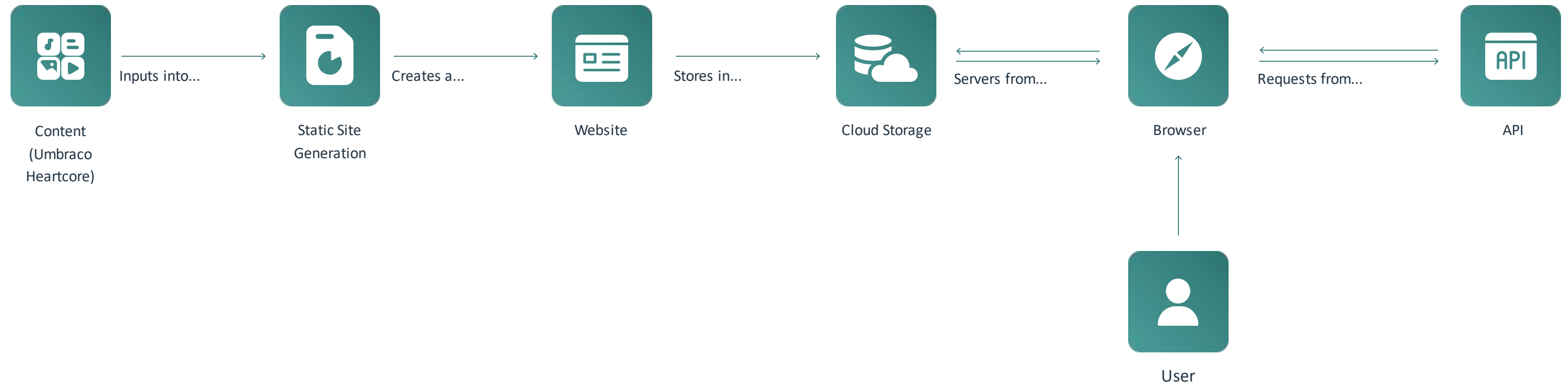
Monolithic applications

End-to-end flow of a monolithic application carbon emissions.



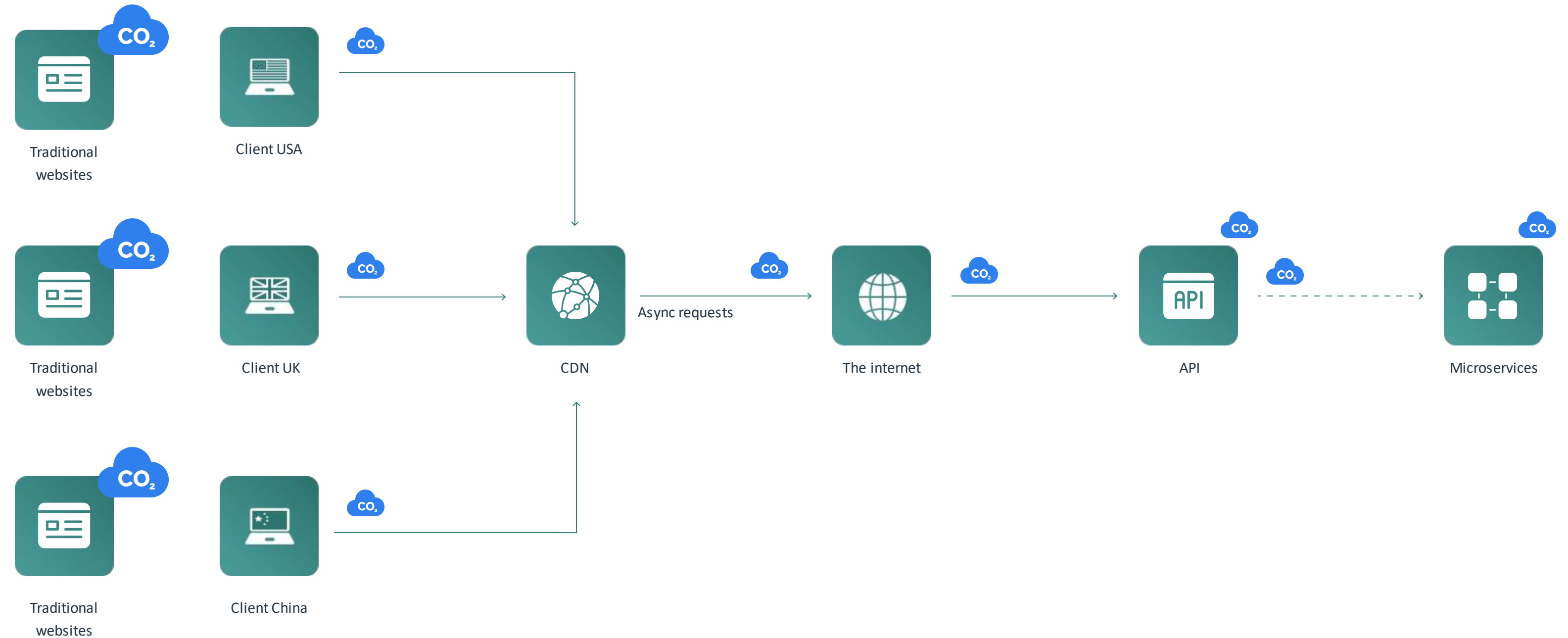
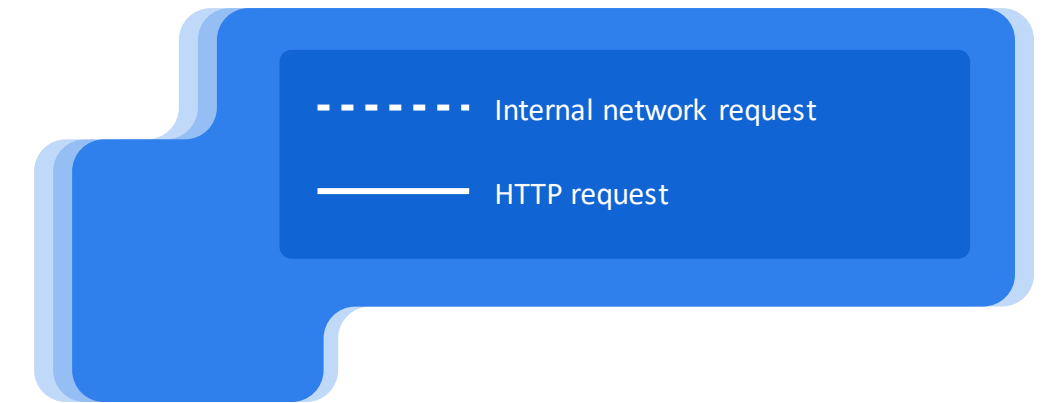
Static site generation

Here is the core principle of SSG; which is a feature of Vercel's Next.js amongst other packages.



JamStack applications

JAMStack applications end to end HTTP request journey and carbon emissions.



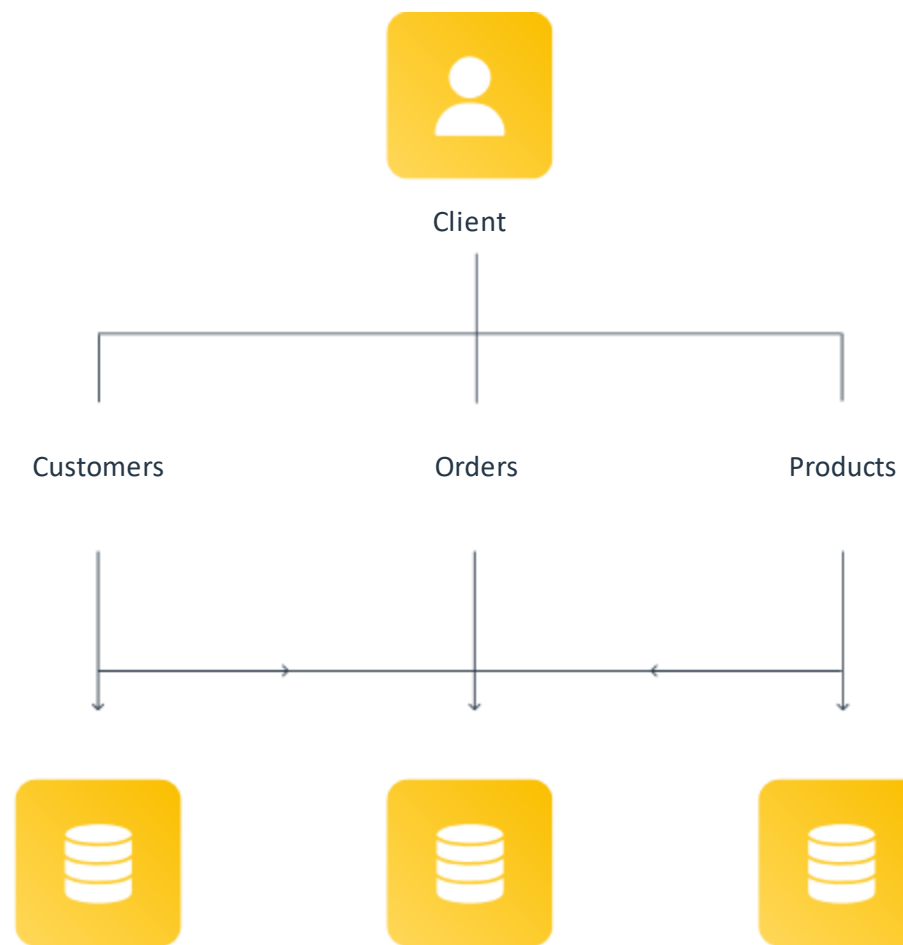
06

Do optimise the number of HTTP requests and size of data transferred

API optimisation

Improve API efficiency (size of response, GraphQL vs REST)

Rest API



GraphQL API



07

Do migrate to SaaS or PaaS

On-Prem or IaaS to PaaS, SaaS or Cloud Native

Sustainable design considerations for infrastructure:

Migration from On-Prem or IaaS to PaaS or SaaS

Take a decouple, service or microservice architecture

Right-size resources, using combination of load tests and FinOps tools like Nordcloud Klarity

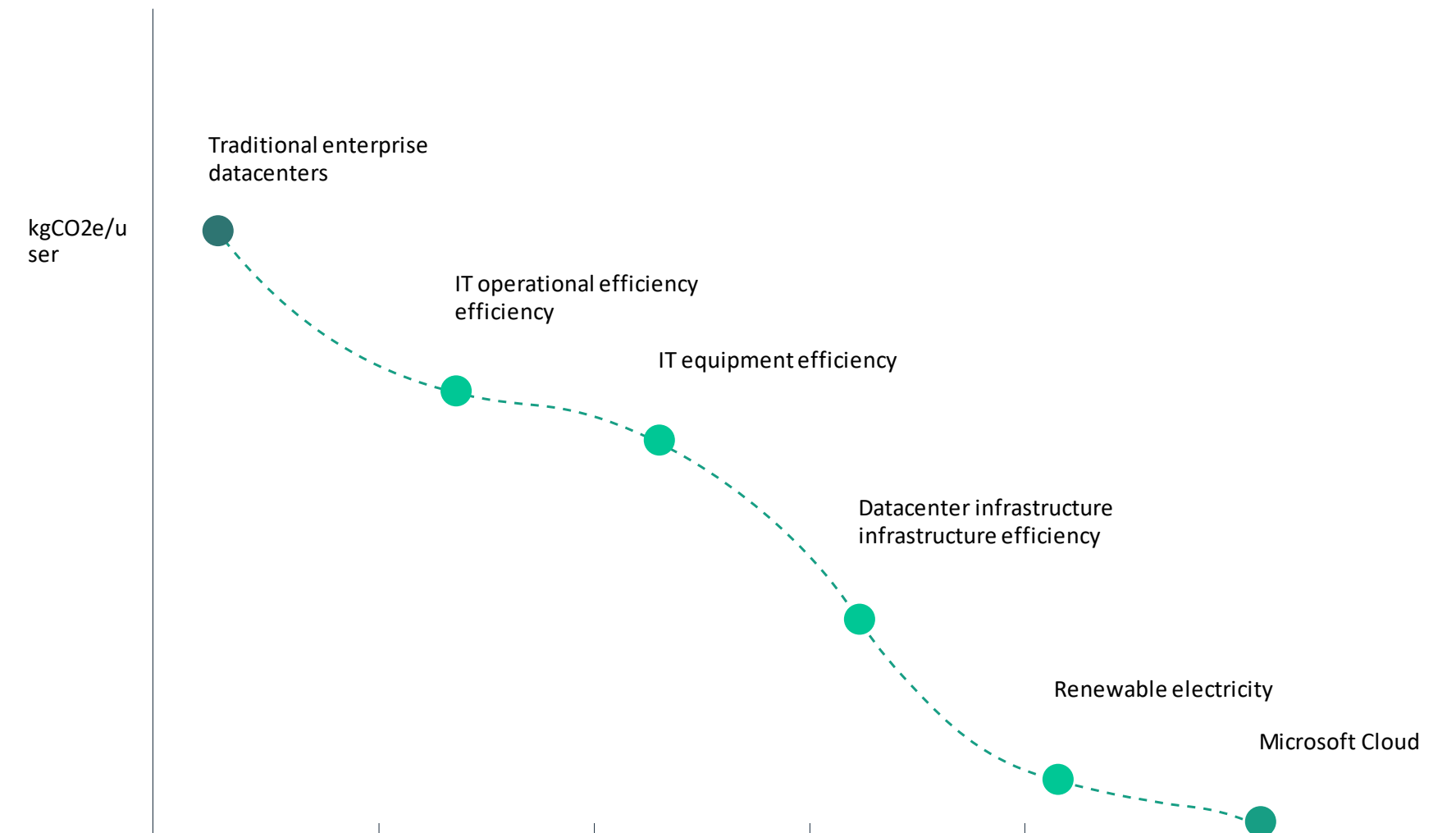
Utilise auto-scaling

Distribute assets from CDN

Block malicious /bad Bots

Turn off non-production environments out of working hours

Adopt cloud-native design patterns to reduce resource utilisation



08

Don't forget to patch

Update dependencies

```
npm-check-updates  
npm update || [package-name]
```

```
pip list  
pip install [package-name] -upgrade
```

```
nuget Get-Package -updates  
nuget Update-Package || [package-name]
```

09

Do use design patterns
patterns

Design Patterns – Circuit Breaker Pattern

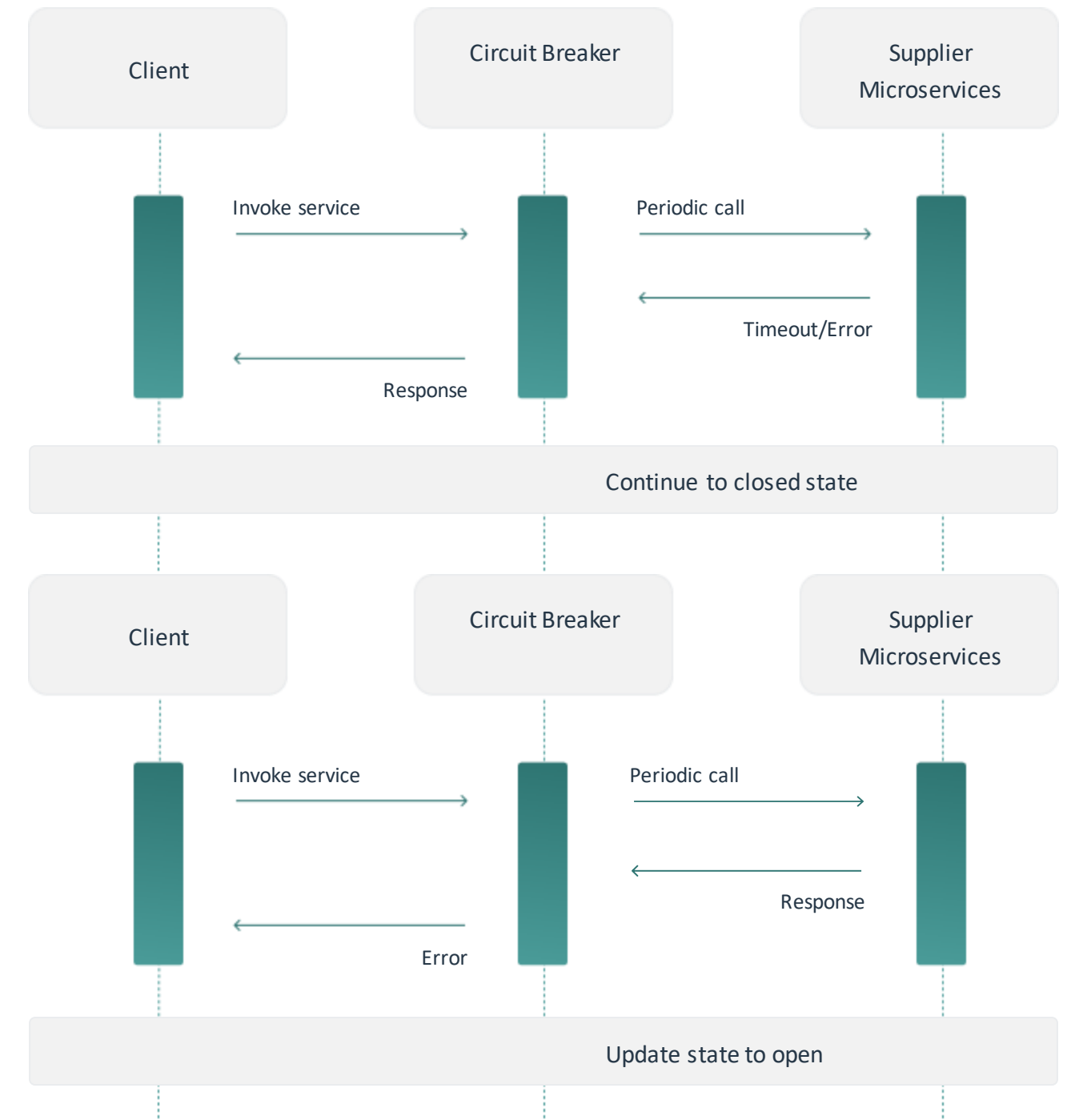
The Circuit Breaker pattern is a popular design pattern used in Microservices Architecture.

It falls under the Sustainable Design Patterns category.

In Microservices architecture, a service usually call other services to retrieve data. Downstream issues like slow network connection, timeouts, or temporal unavailability can be solved retrying calls.

However, severe issue where a service is unavailable for a longer time can result in the network resources being exhausted. Enter the Circuit Breaker Design Pattern to overcome this problem.

- Closed state
- Open state
- Half-open state



10

When and where you
process data makes a
difference

Not all datacentres are created equal

Even though Microsoft cloud may be net-zero, it is still important to note that placing your resources in certain regions may decrease the effect on the environment.

- This is known as carbon aware computing
- Some regions have a Zero waste certification
- Zero Waste certification shows that a region “diverts at least 90% of our food, office, and construction waste away from landfills through employee-driven reuse, recycling and composting programs and sustainable community partnerships” (Microsoft).
- Compare Microsoft Cloud regions via the Azure Global Infrastructure Map
- Process/batch when carbon intensity is low

The screenshot displays the Azure Global Infrastructure Map interface. It features a map of Europe with colored markers for different regions. Below the map, two region profiles are shown: West Europe and North Europe. Each profile includes a title, a subtitle, and several key attributes: Location, Data residency, Year opened, Availability Zones, Products, and Disaster recovery options. A 'Sustainability' section is highlighted with a red box, containing information about the Microsoft Circular Center and Zero-waste certification. At the bottom, there are tabs for 'Global compliance' and 'Industry compliance'.

West Europe	
Region with Availability Zones	
Location	Data residency
Netherlands	Stored at rest in Europe Learn more
North Europe	
Region with Availability Zones	
Location	Data residency
Ireland	Stored at rest in Europe Learn more
Year opened	Availability Zones
2009	Available with three zones
Products	Disaster recovery
See products in this region	Cross-region disaster recovery options: Azure Site Recovery Region Pairing
	In-region disaster recovery options: Zonal DR with Azure Site Recovery
Sustainability	Global compliance
Microsoft Circular Center coming soon Zero-waste certified	Industry compliance

11

Do remember these
three things and act
today!

Three key takeaways

01

_____ The systems we design have an impact on the environment and the time to act is now

02

_____ HTTPS requests and processing results in carbon emissions

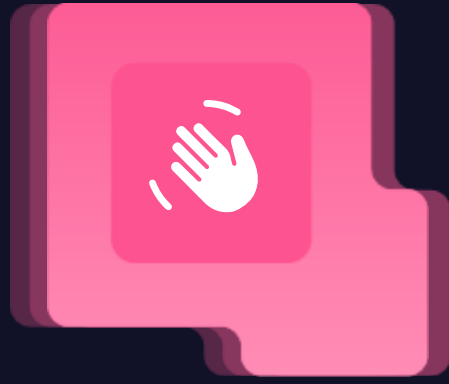
03

_____ Optimise your systems to increase performance, save costs
save costs and reduce emissions



Download the report





Now, I'm
Andy Eva-Dale!
Let's connect.

