



# RED BADGER

## Mapping the Future of Sustainable Digital Products

2023-11-28

Stuart Harris

Founder & Chief Scientist

Red Badger



**1** Rust

**2** WebAssembly

**3** Next-gen platforms

# Stu

- Software engineer
- Founder of Red Badger

@stuartharris



**RED BADGER**



# 6 principles of sustainable software engineering

1. **Carbon efficiency** — minimize the amount of carbon emitted per unit of work
2. **Energy efficiency** — the more you utilize a computer, the more efficient it becomes at converting electricity to useful computing operations
3. **Carbon awareness** — Shifting demand to times when carbon intensity is low
4. **Hardware efficiency** — reduce embodied carbon by extending the life of hardware
5. **Measurement** — measure the carbon footprint of your software
6. **Climate commitments** — net zero (abate/eliminate) vs. carbon neutral (offset)

| 25 minute overview

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

Energy consumed by software in kWh

Carbon emitted per kWh of energy, gCO<sub>2</sub>/kWh

Carbon emitted through the hardware that the software is running on

Functional Unit; this is how software scales, for example per user or per device



Green  
Software  
Foundation  
[greensoftware.org](http://greensoftware.org)



# Three revolutions

- 1 Rust is a **revolution** in sustainable software engineering
- 2 WebAssembly
- 3 Next-gen platforms

# 1

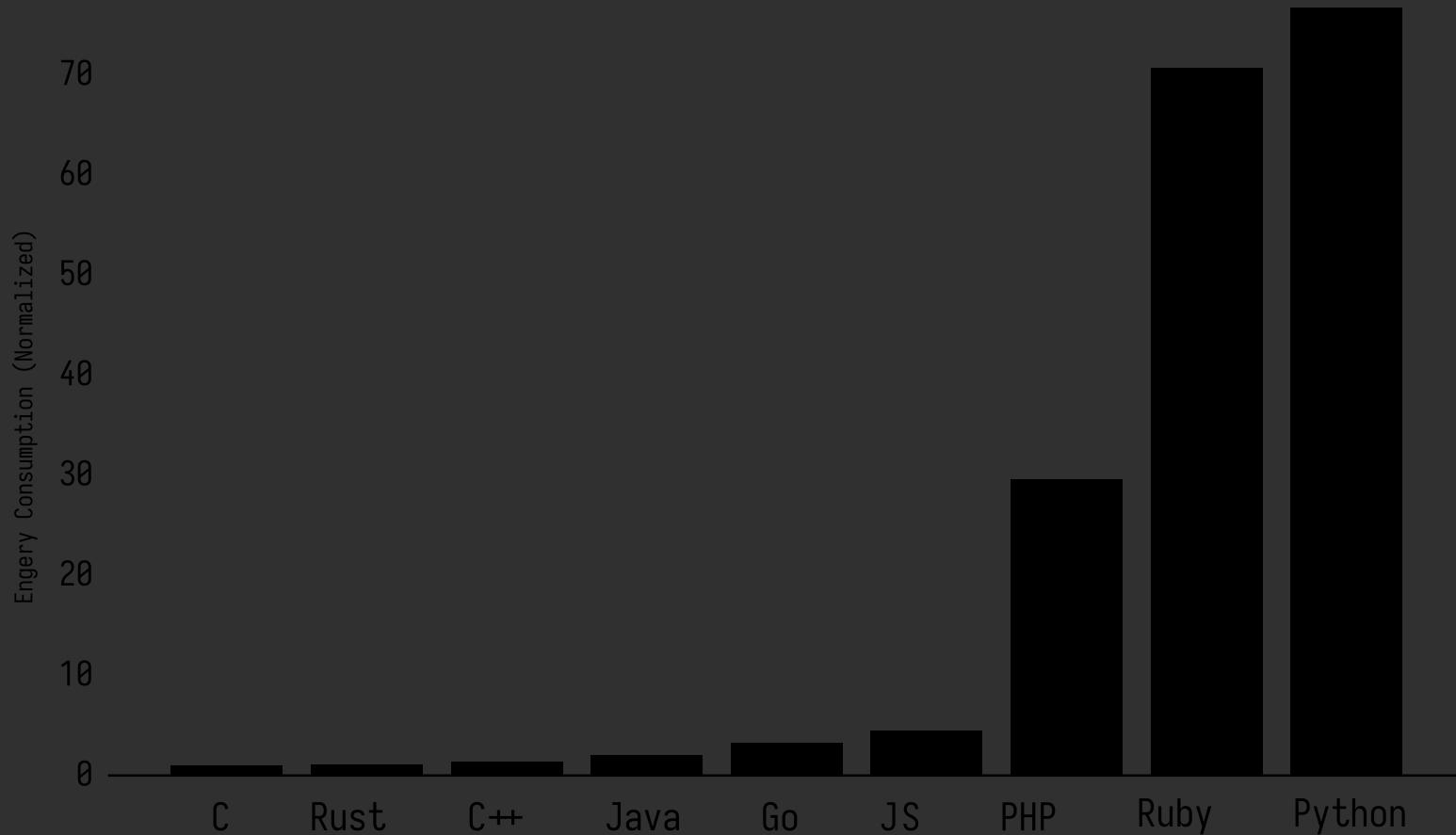
## Rust is a **revolution** in sustainable software engineering

"Everyone has a part to play."

"Sustainability is enough, all by itself, to justify our work."



## Energy Consumption by Programming Language (Normalized)



Why Rust in production?, Sustainability with Rust

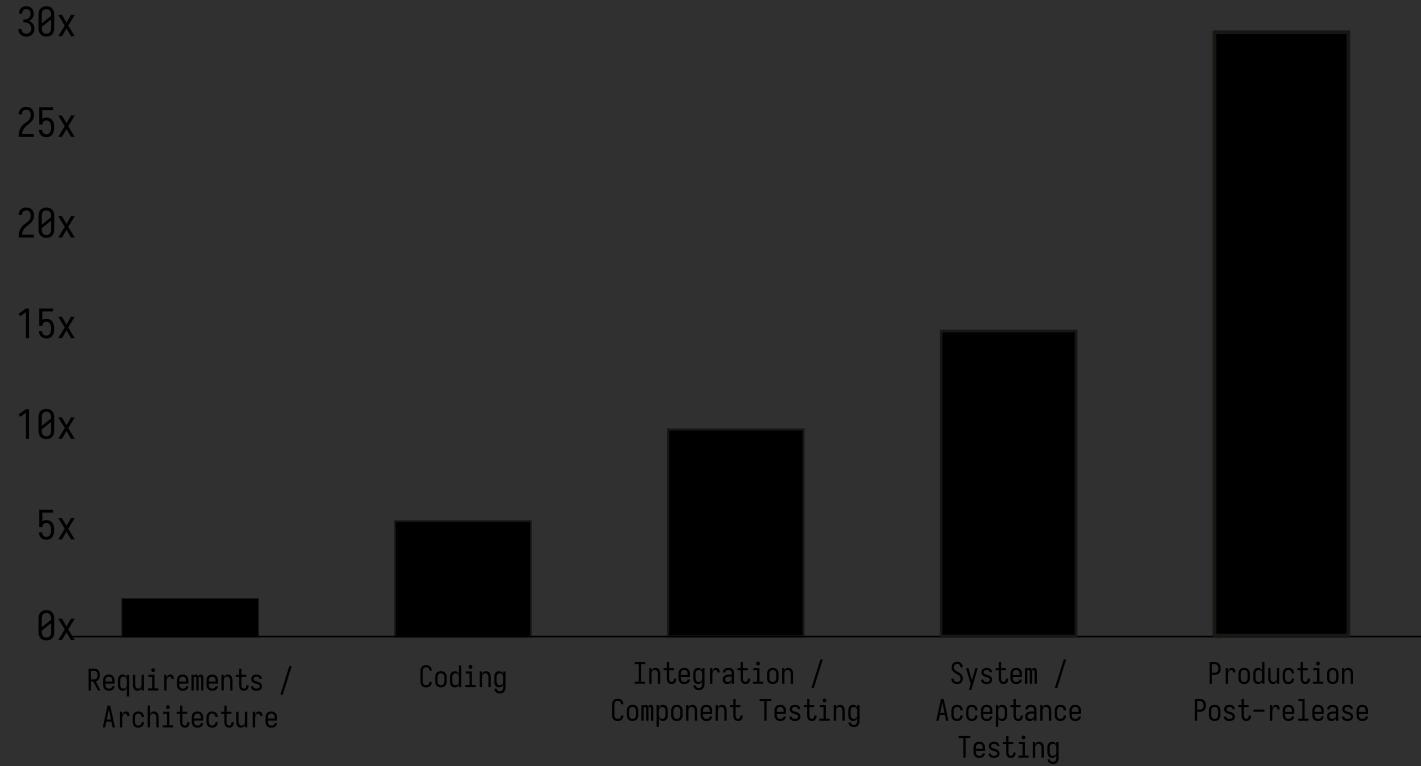
# 1 Sustainable Software Engineering

Find as many "Carbon Proxies" as you can

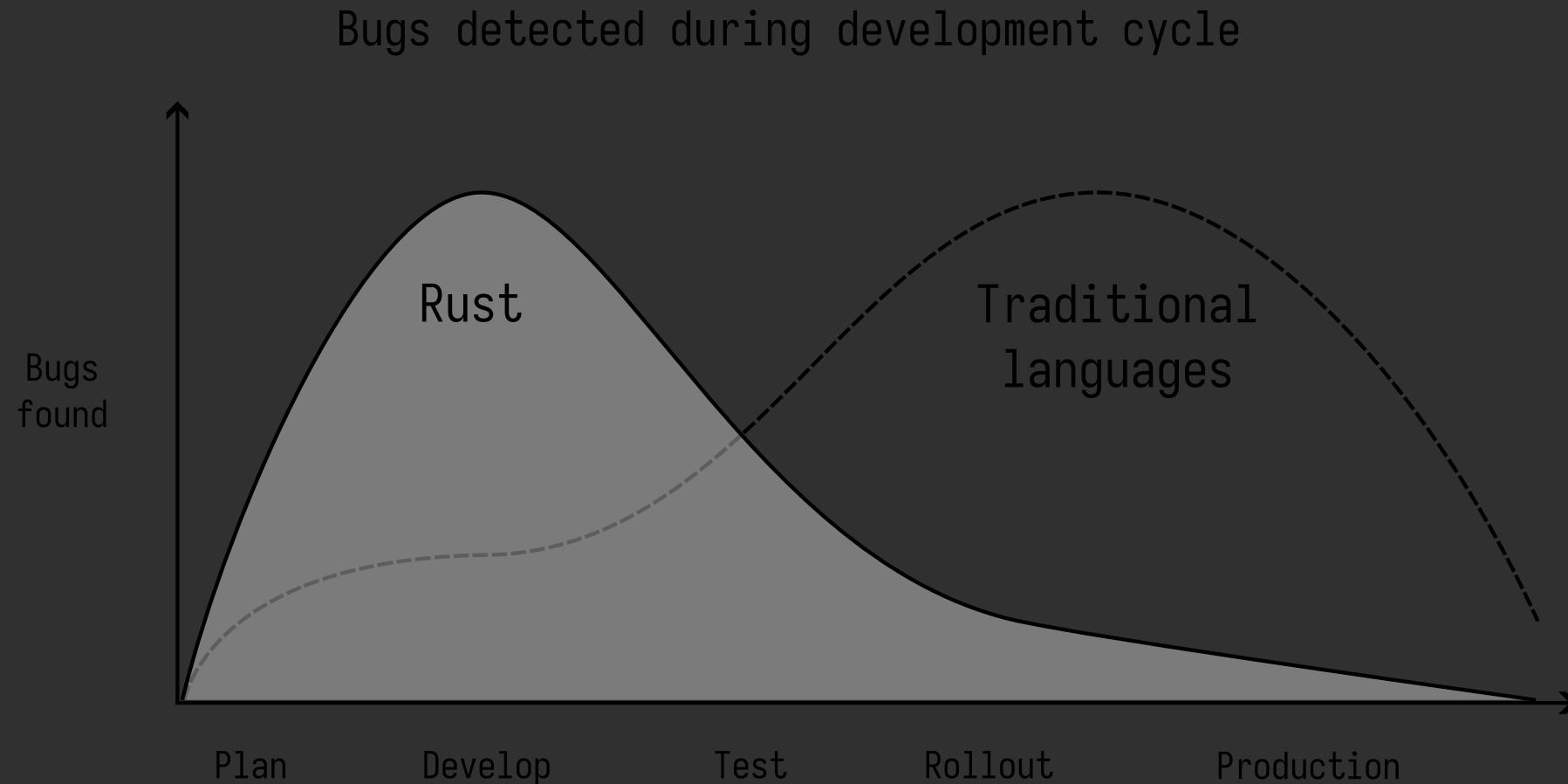
- Time-to-interactive and Page weight
- Average server response time
- Cost of your services
- The utilisation of your servers

Sustainable Software Engineering overview, [How to measure and reduce the carbon footprint of your application](#)

Relative cost to fix bugs, based on time of detection



Why Rust in production?



Why Rust in production?

# Three revolutions

1 Rust

2 WebAssembly

3 Next-gen platforms

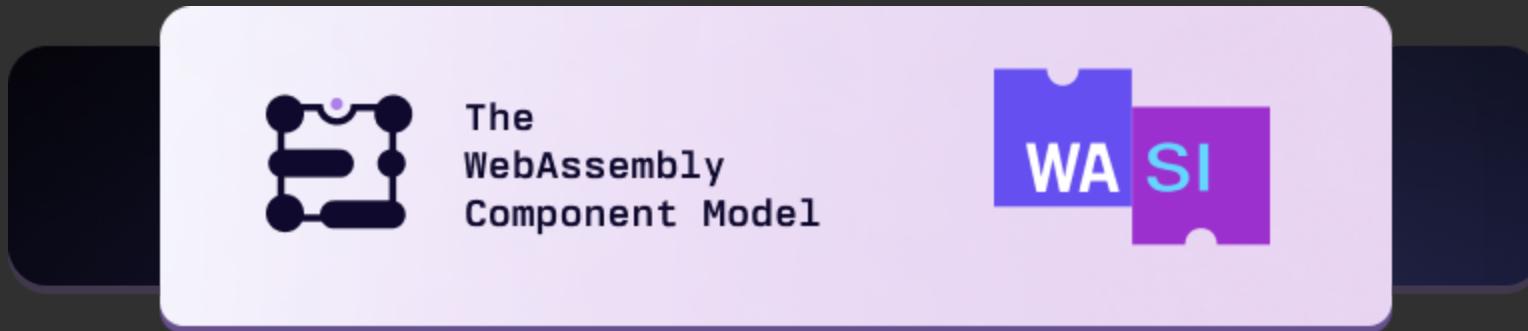
2

**WebAssembly** is a  
**revolution**





## The WebAssembly Component Model





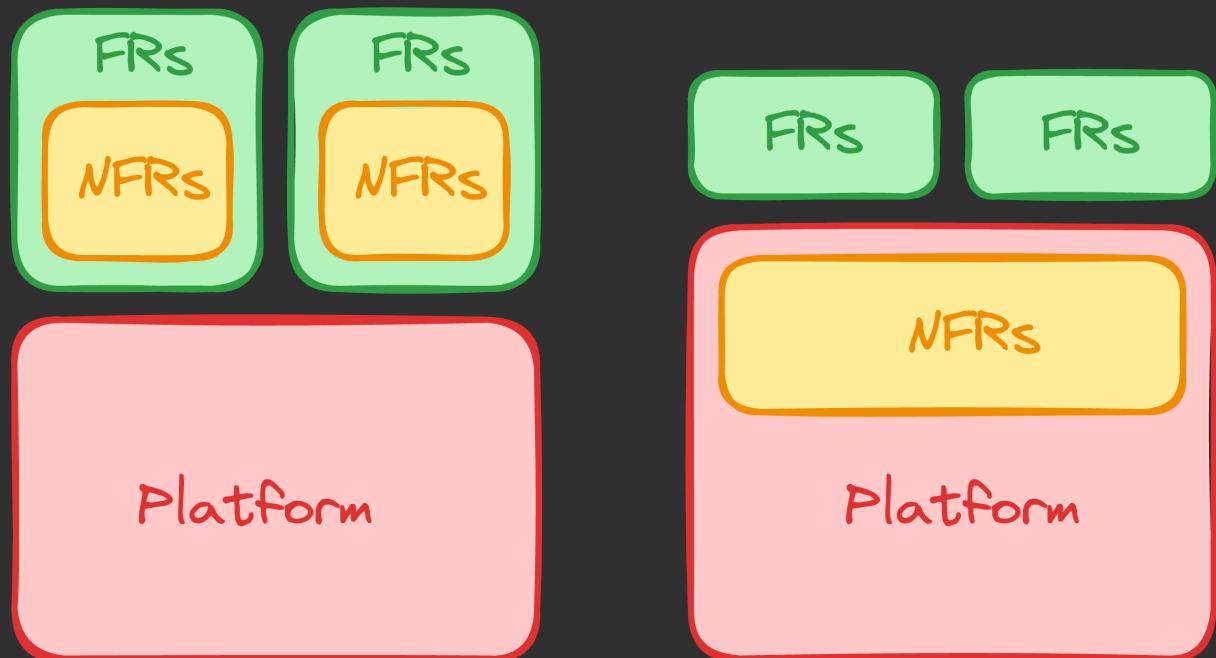
# Three revolutions

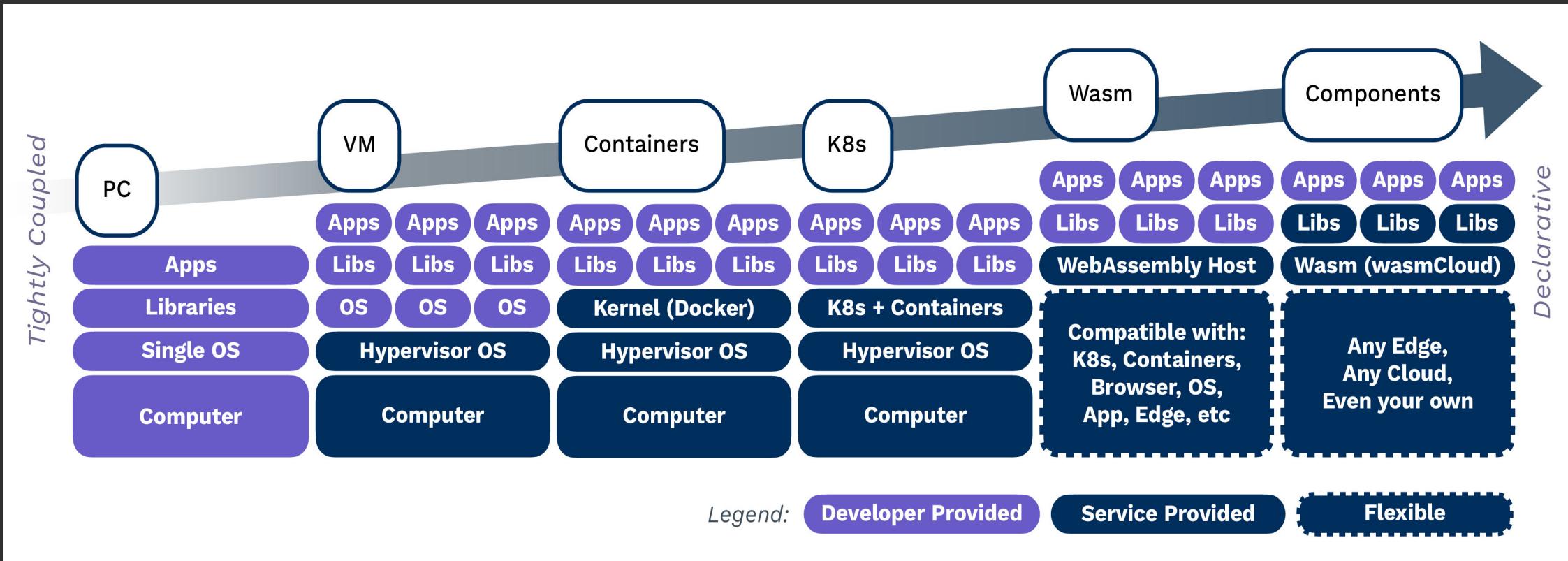
- 1 Rust
- 2 WebAssembly
- 3 Next-gen platforms

# 3

Next-gen platforms are  
a **revolution**

Pushing non-functionals down





## Cosmonic and **wasmCloud**

<https://cosmonic.com/>, <https://wasmcloud.com/>

## Fermyon and **Spin**

<https://www.fermyon.com/>, <https://www.fermyon.com/spin>, <https://www.fermyon.com/cloud>

# Three revolutions

- 1** Rust
- 2** WebAssembly
- 3** Next-gen platforms

# 6 principles of sustainable software engineering

1. **Carbon efficiency** — minimize the amount of carbon emitted per unit of work
2. **Energy efficiency** — the more you utilize a computer, the more efficient it becomes at converting electricity to useful computing operations
3. **Carbon awareness** — Shifting demand to times when carbon intensity is low
4. **Hardware efficiency** — reduce embodied carbon by extending the life of hardware
5. **Measurement** — measure the carbon footprint of your software
6. **Climate commitments** — net zero (abate/eliminate) vs. carbon neutral (offset)

25 minute overview

# Thank you!



@stuartharris