

# Carbon efficient high density enterprise workloads with scale-to-zero WebAssembly

Stuart Harris

Civo Navigate, Berlin, 10th Sep 2024

# Data Centres in Ireland

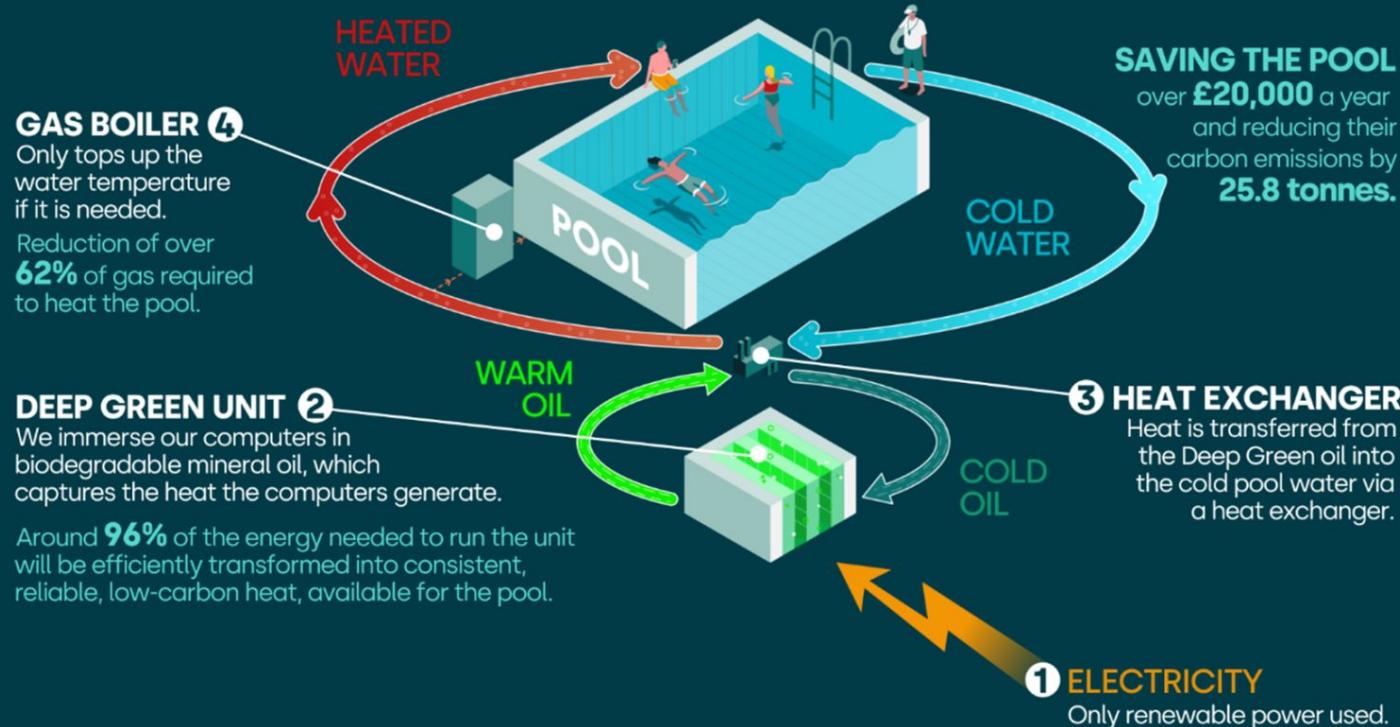
- **82** operational  
**14** under construction  
**52** planned [ref](#)
- Consume **21%** of the nation's electricity, that's more than all urban homes (18%), and projected to rise to **33%** within **3** years [ref](#)
- **50.7%** fossil fuels in 2023, including coal, peat and oil, and not including imports [ref](#)





DEEP  
GREEN

## HOW DEEP GREEN HEATS A POOL



# Equinix PA10



That's great,  
but why create so much heat in the first place?

Let's look at how we can do better.

# Hi, I'm Stu

- Software engineer
- Founder and Chief Scientist  
at  **RED BADGER**

@stuartharris



# Enterprise applications

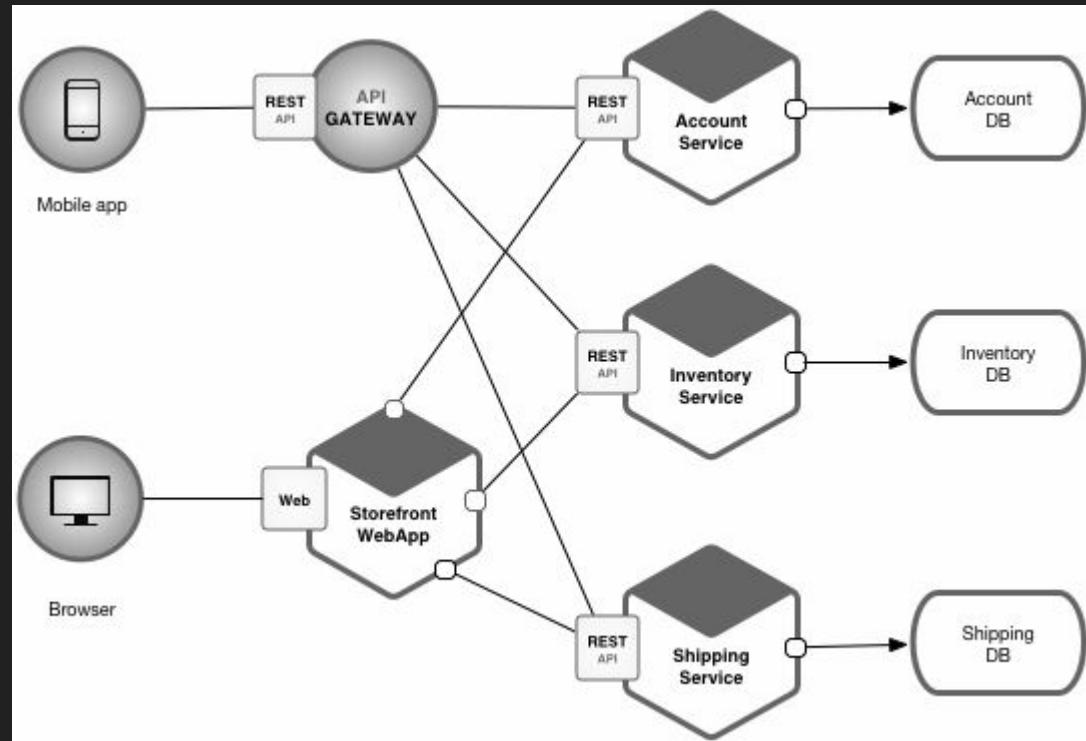
- Microservices today
- Carbon footprint
- Vertical scaling and scale-to-zero
- Next generation WebAssembly platforms
- A real-world example
- How do we get there?

# Microservices

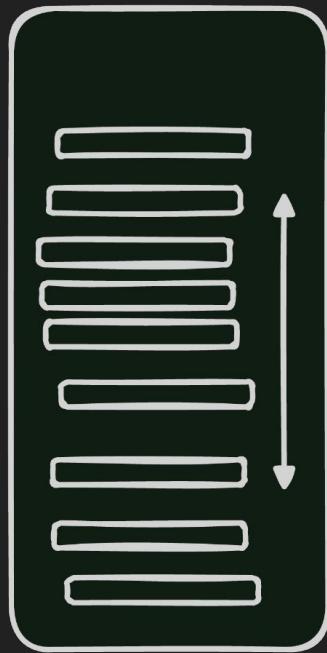
The good

The bad

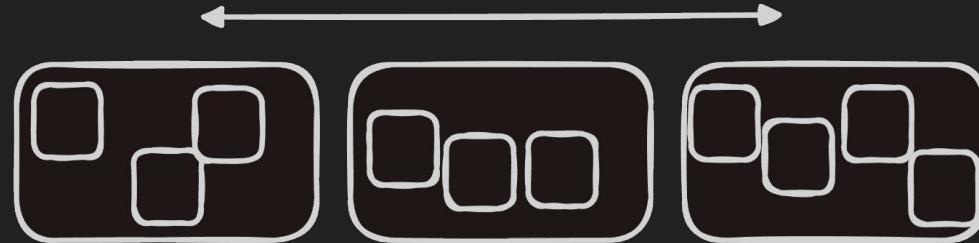
The ugly



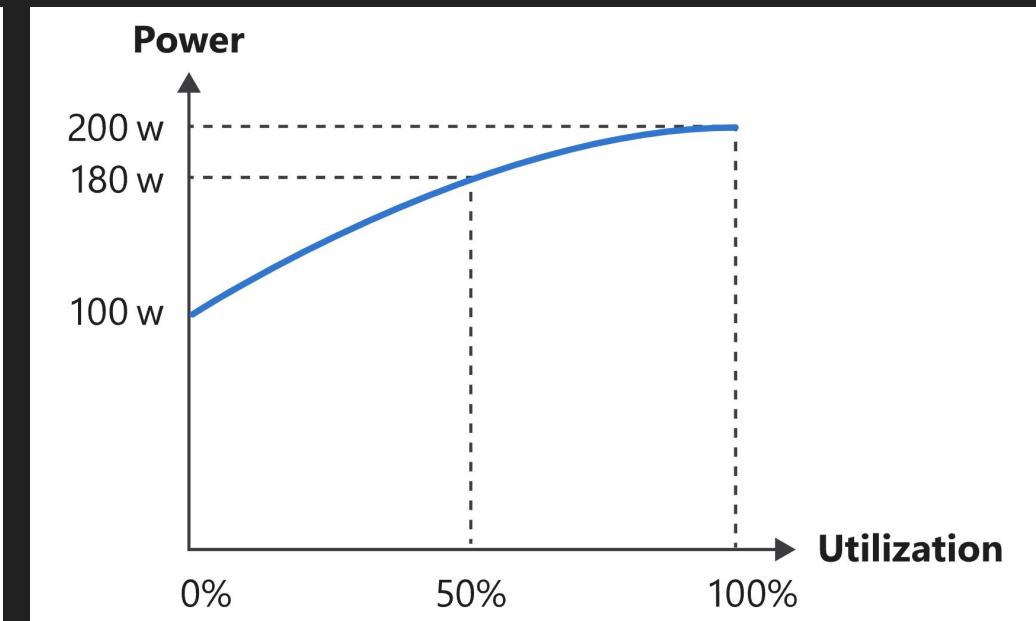
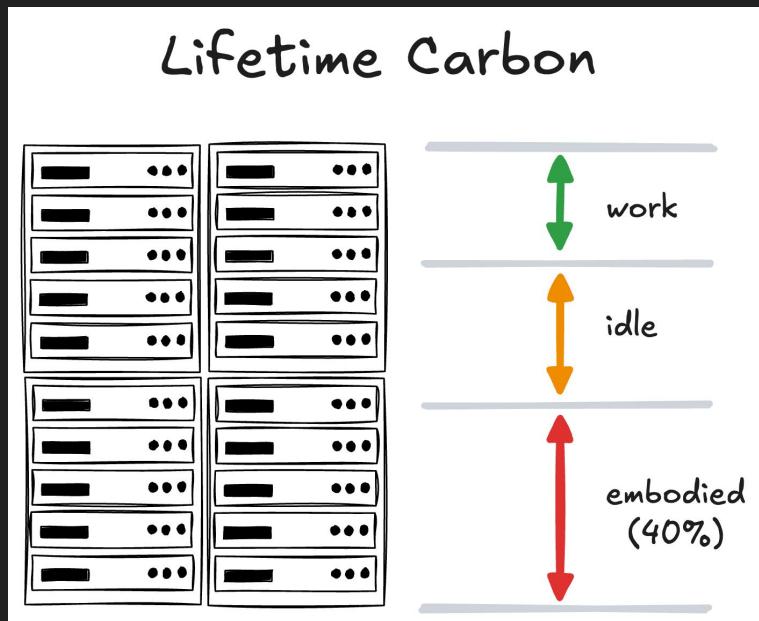
# Scaling Vertically and Scale-to-zero



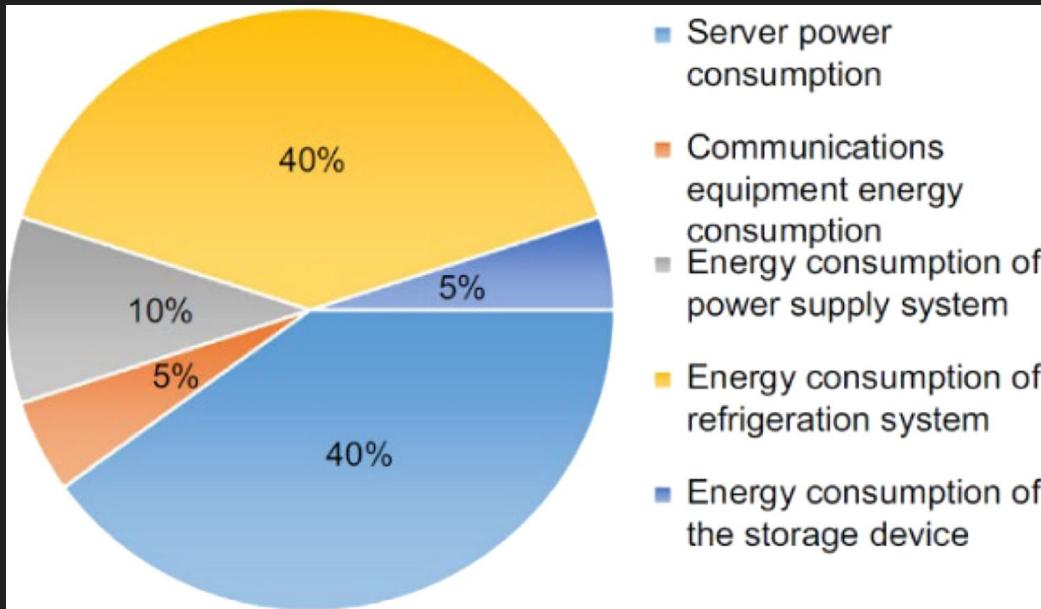
vs



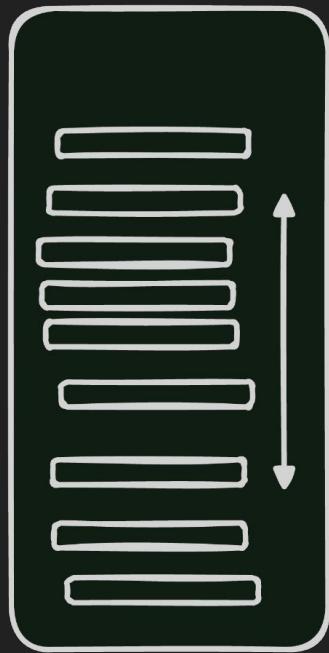
# Carbon



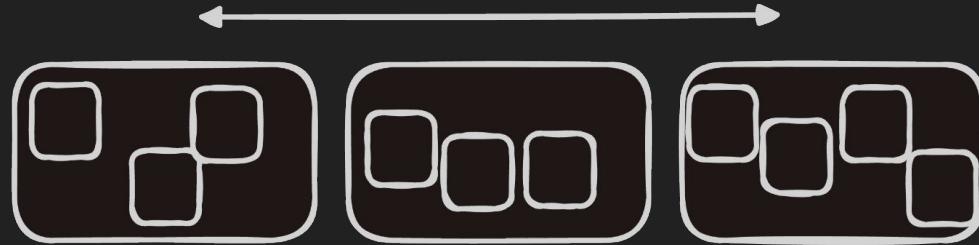
# Cooling



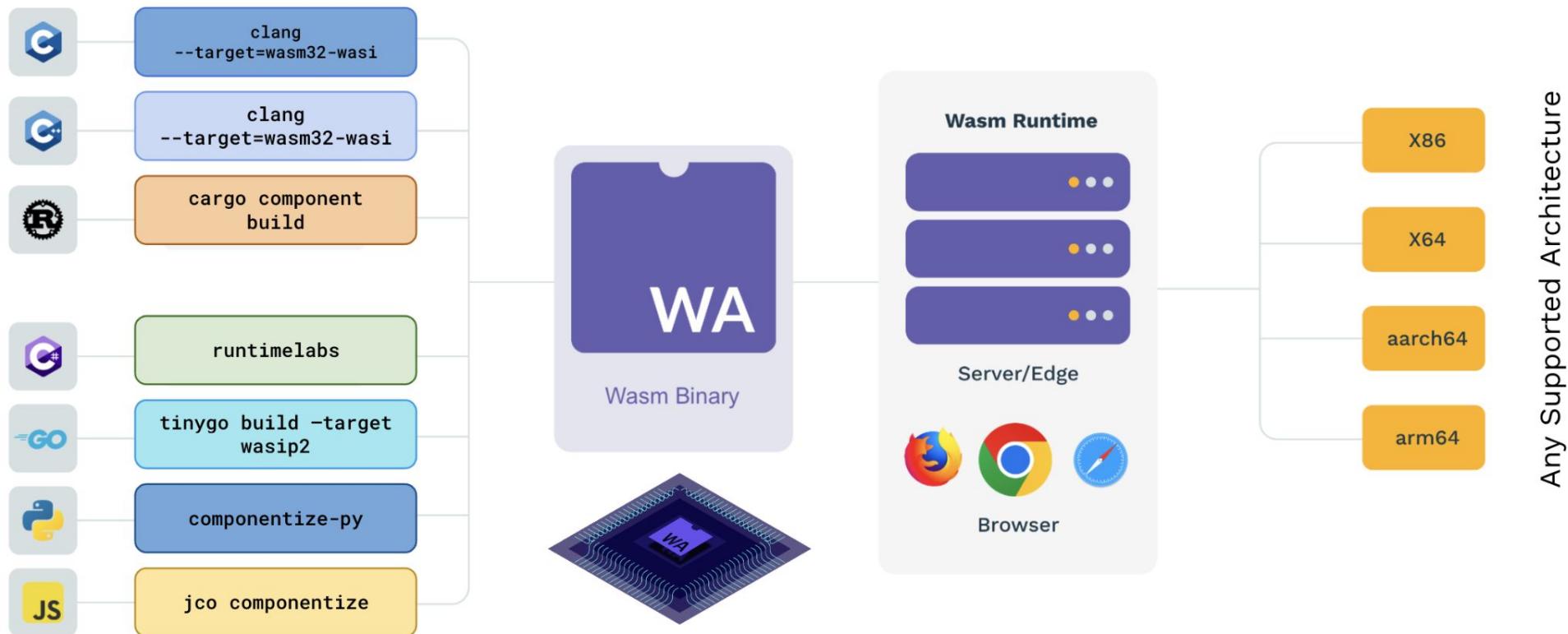
# Scaling Vertically and Scale-to-zero



vs

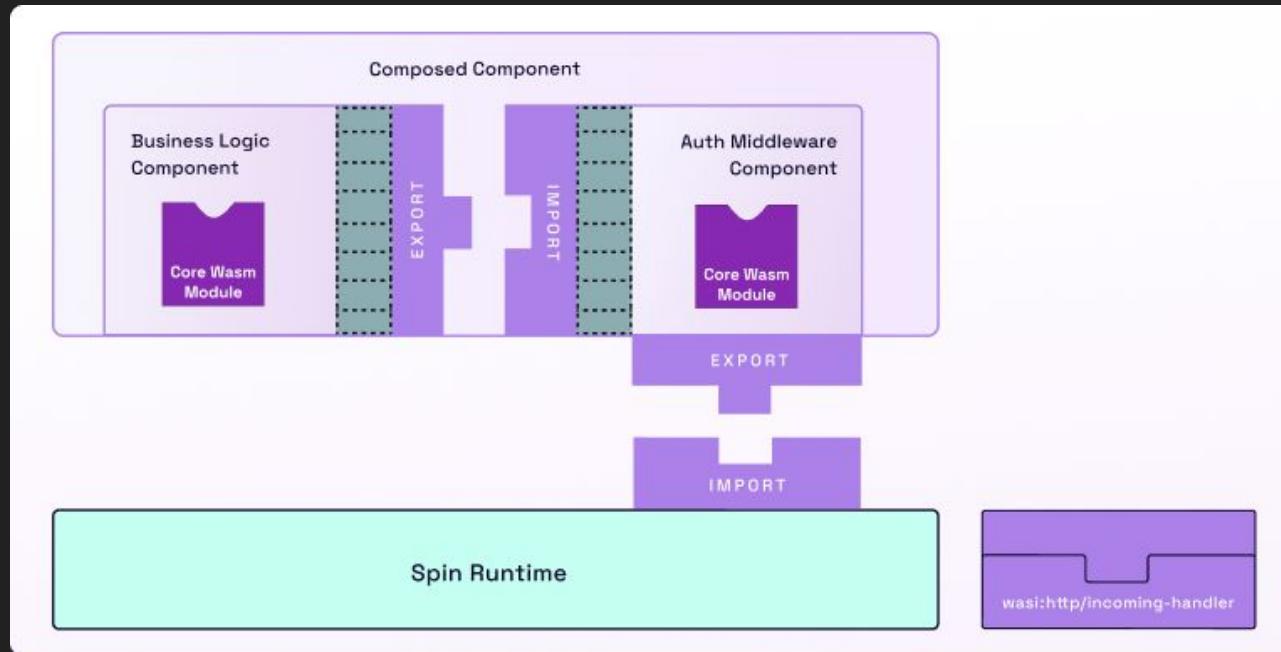


# WebAssembly

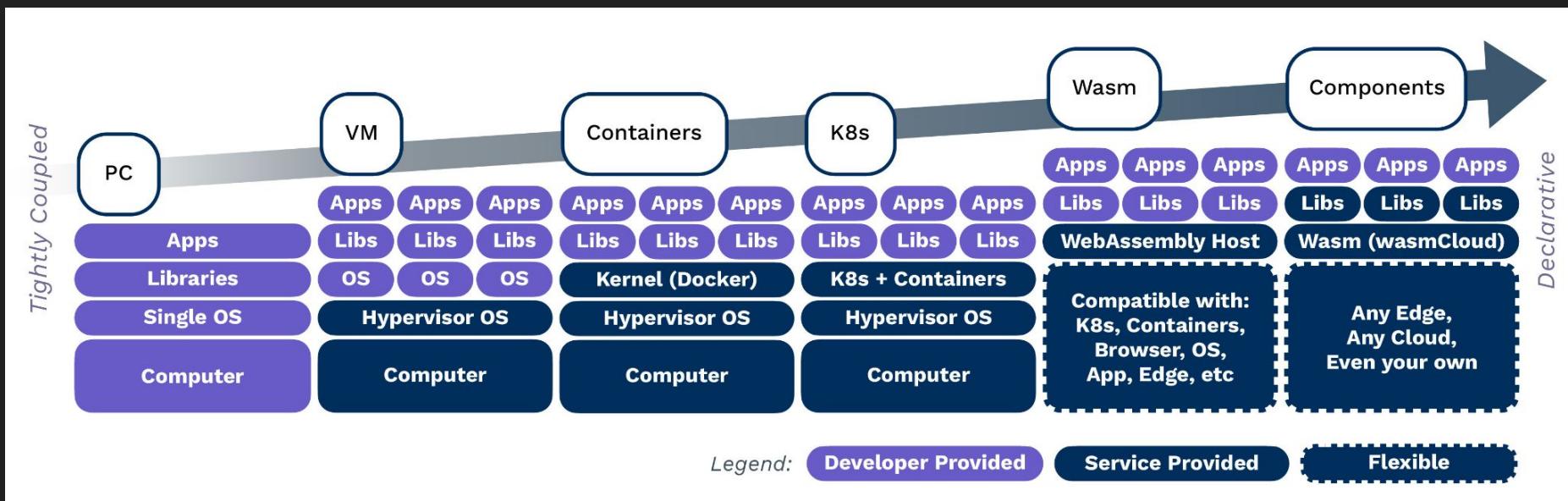


# WASI 0.2 and the Component Model

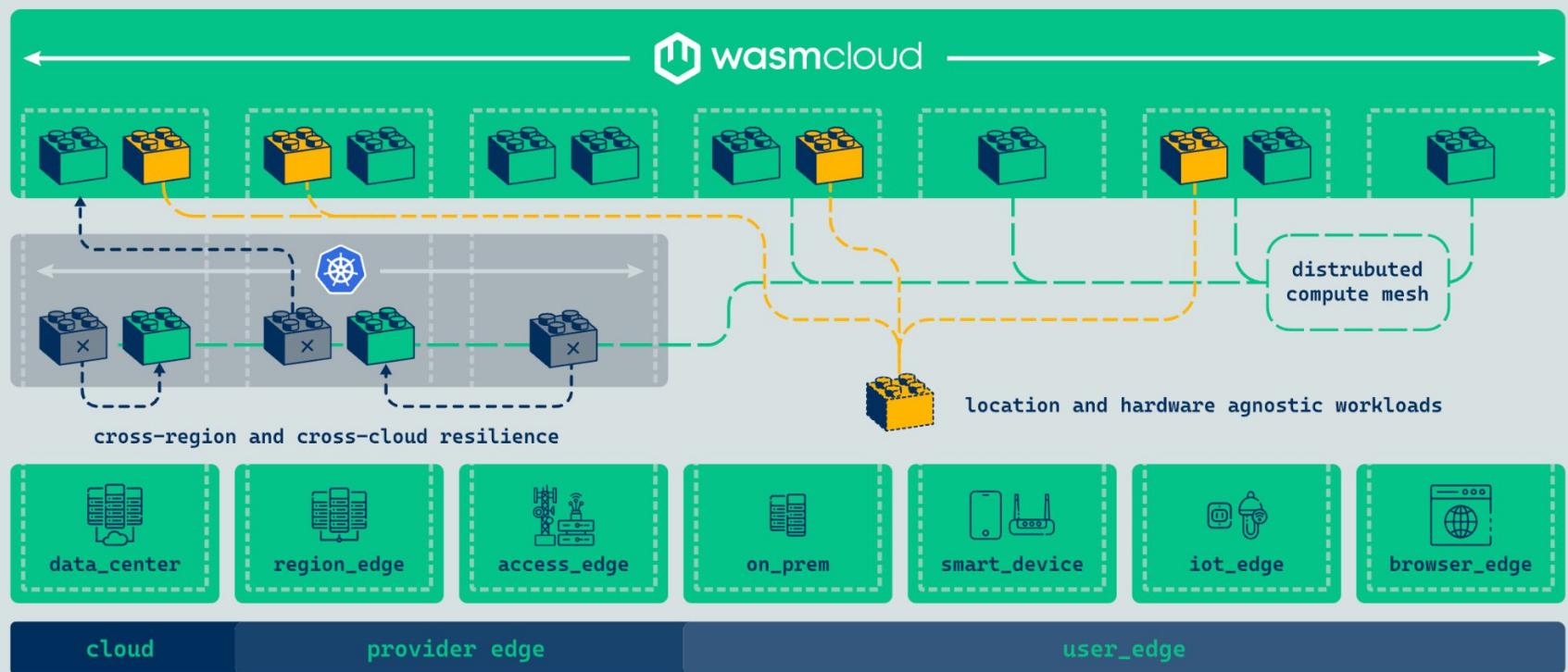
WA SI



# Platform evolution



# wasmCloud





# Build

## Faster Development Cycles

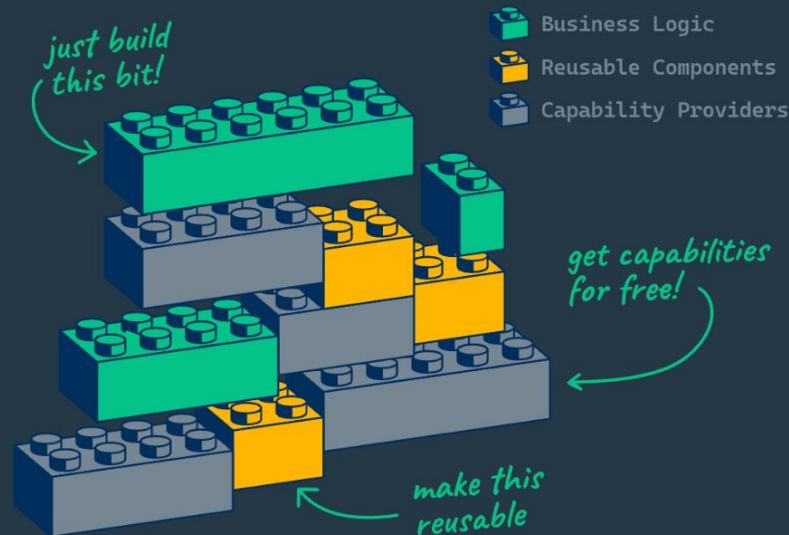
Leverage reusable, polyglot, Wasm components on a reliable, distributed platform.

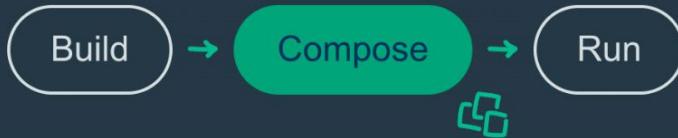
## Centrally Maintainable Apps

Reusable, version-controlled components empower platform teams to maintain thousands of diverse apps centrally.

## Integrate with Existing Stacks

wasmCloud has first-tier support for Kubernetes, AWS, Azure, GCP, Jenkins, Github Actions, ArgoCD, Backstage, Chainguard, Databases, Messaging, and more.





## Compose

### Development Without Lock-In

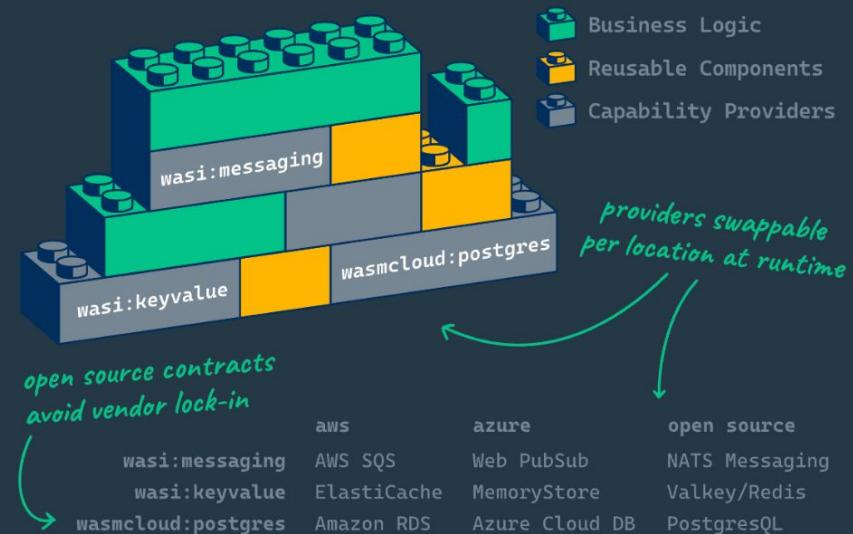
Define application dependencies at runtime via contract driven interfaces leveraging different vendors across deployments, dev, QA, or prod.

### Truly Portable Apps

Run the same Wasm application across operating systems and architectures—no new builds required. Linux, MacOS X, Windows, ARM, x86, and more.

### Custom Capabilities

Easily extend the secure wasmCloud host at runtime to support custom dependencies, hardware, or business contracts.





## Run

### Scale-to-Zero with Zero Cold Starts

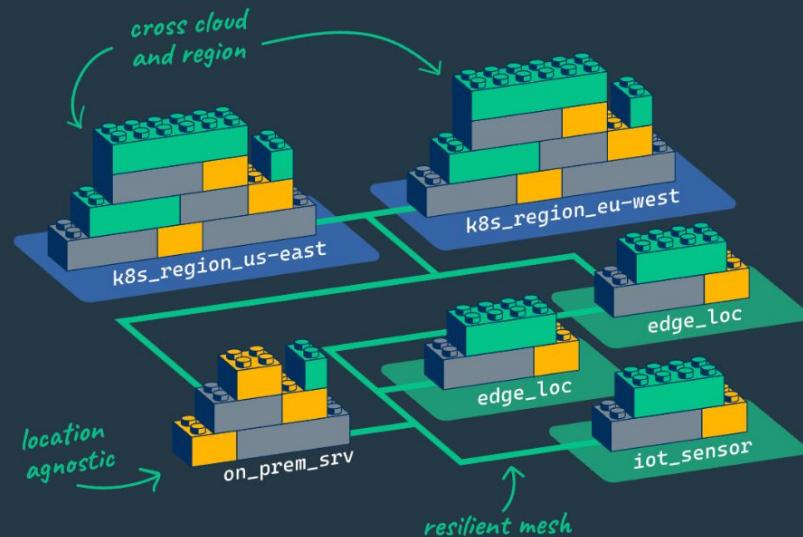
Sub-millisecond start times and vertical autoscaling means workloads scale to the demand.

### Reliable, Fault-Tolerant Apps

Horizontal scaling with automated fail-over gives apps capability-level resiliency, reliability, and scalability.

### Deploy Across Clouds

Close to your users, with local-first routing and at-most-once delivery, wasmCloud delivers cross-region, cross-cloud, and cross-edge capability-level resiliency to every deployment



# Spin from Fermyon



“We've been able to take a Kubernetes batch process of tens of thousands of orders and cut the compute cost by 60%, without trading off performance”



**Kai Walter**

Distinguished Architect, ZEISS Group

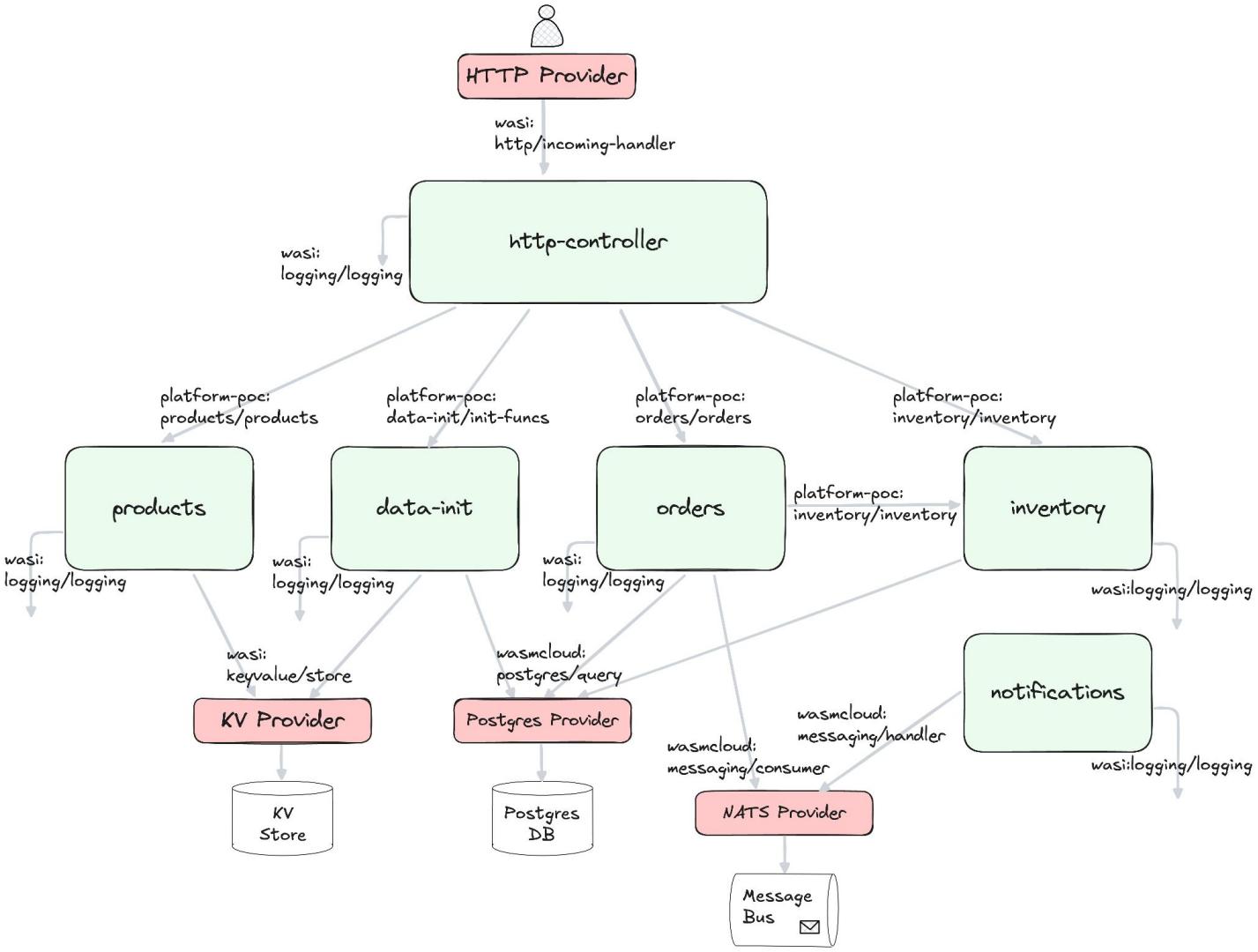
“I'm getting some hands on time with WebAssembly by playing around with Fermyon and their Spin framework. I think it's extra dope.”



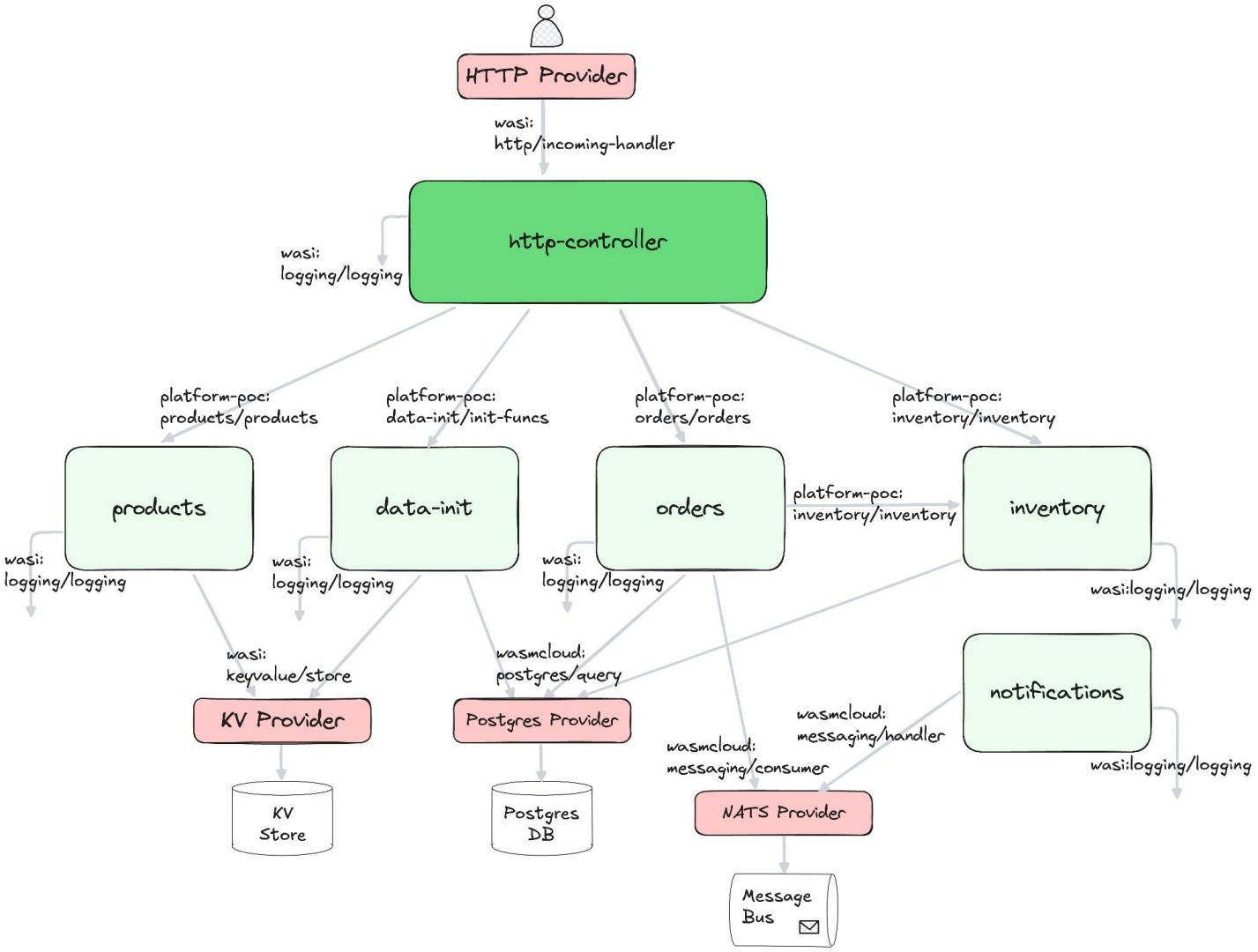
**Kelsey Hightower**

Former Distinguished Engineer, Google

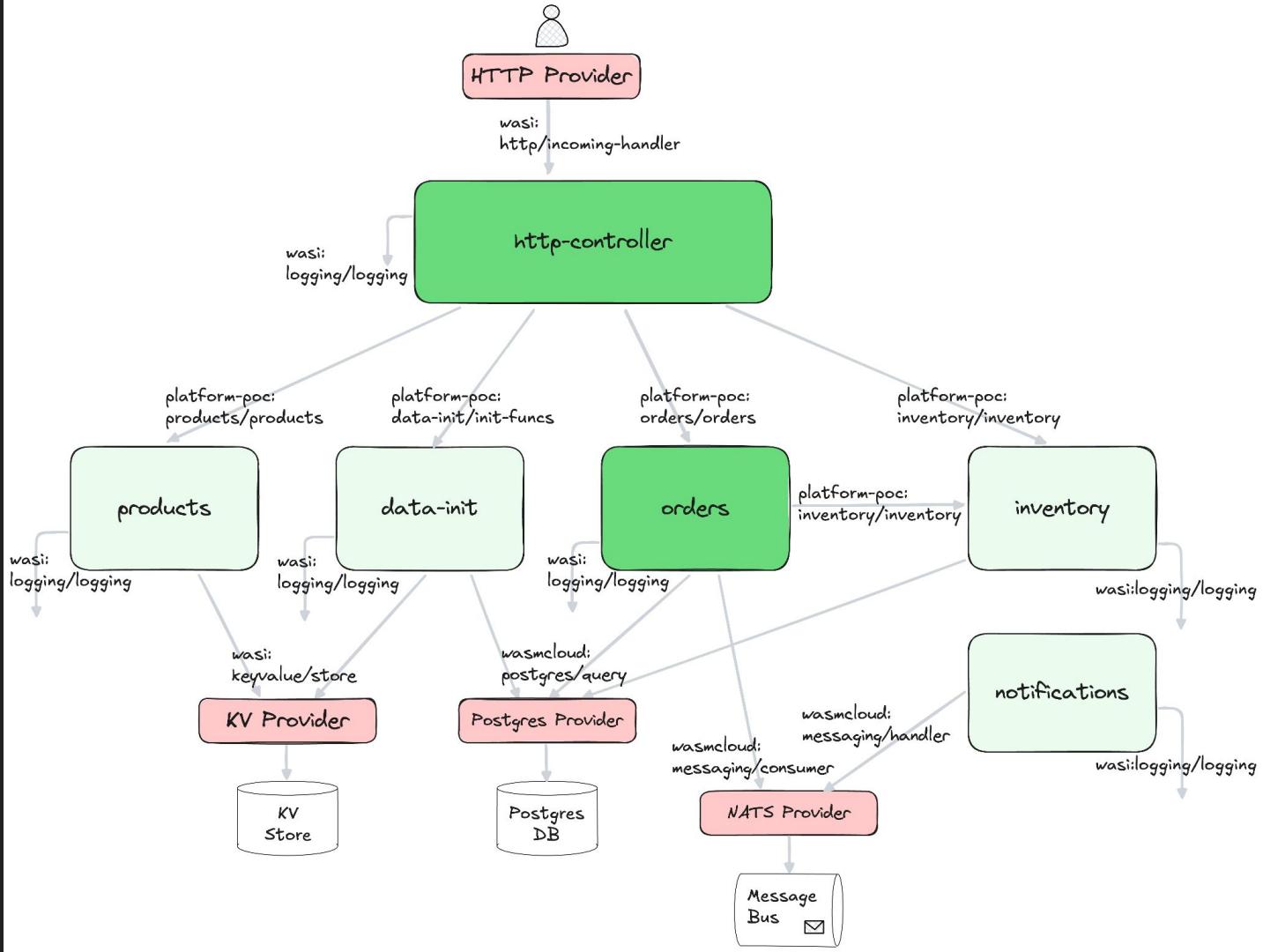
# Demo



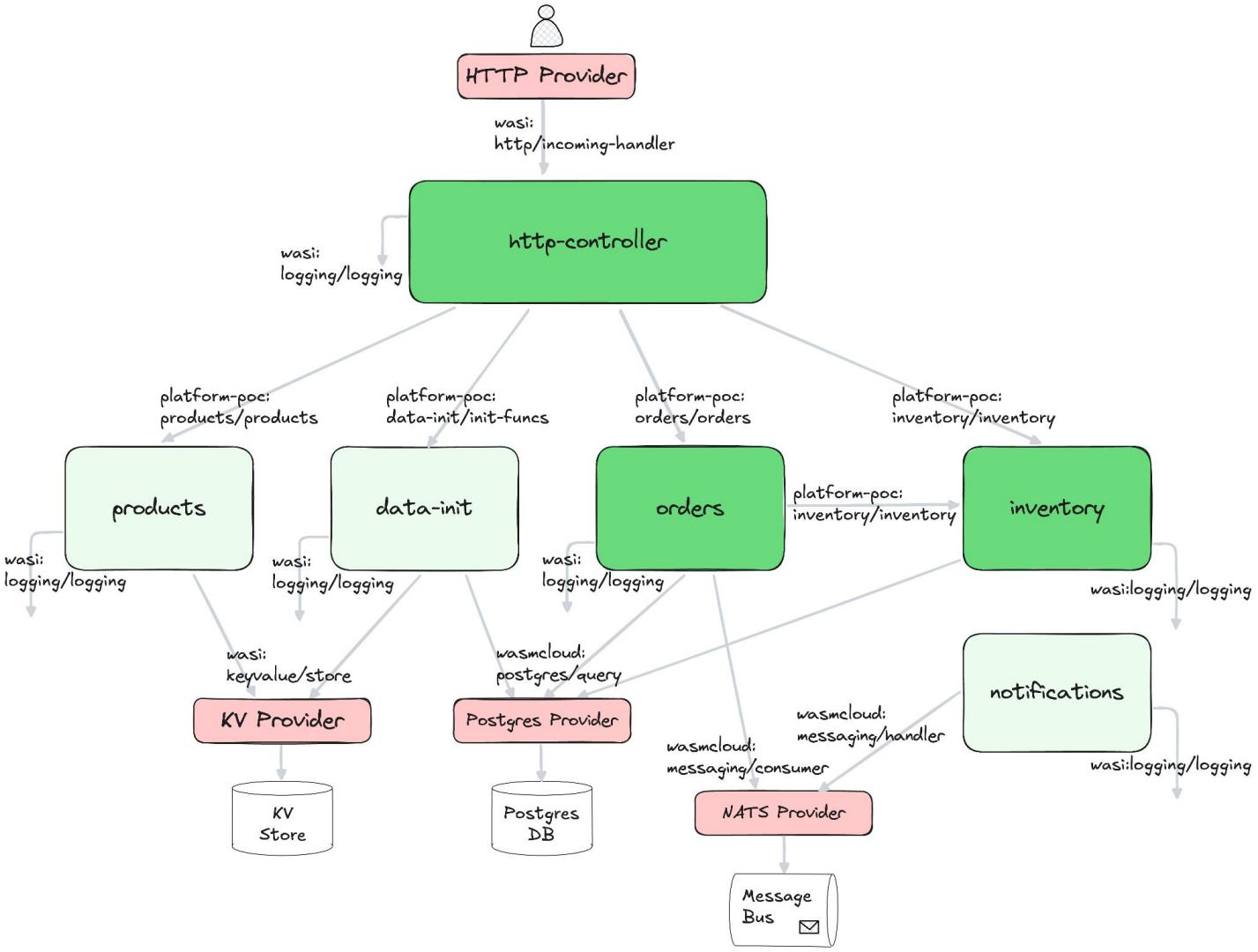
# Demo



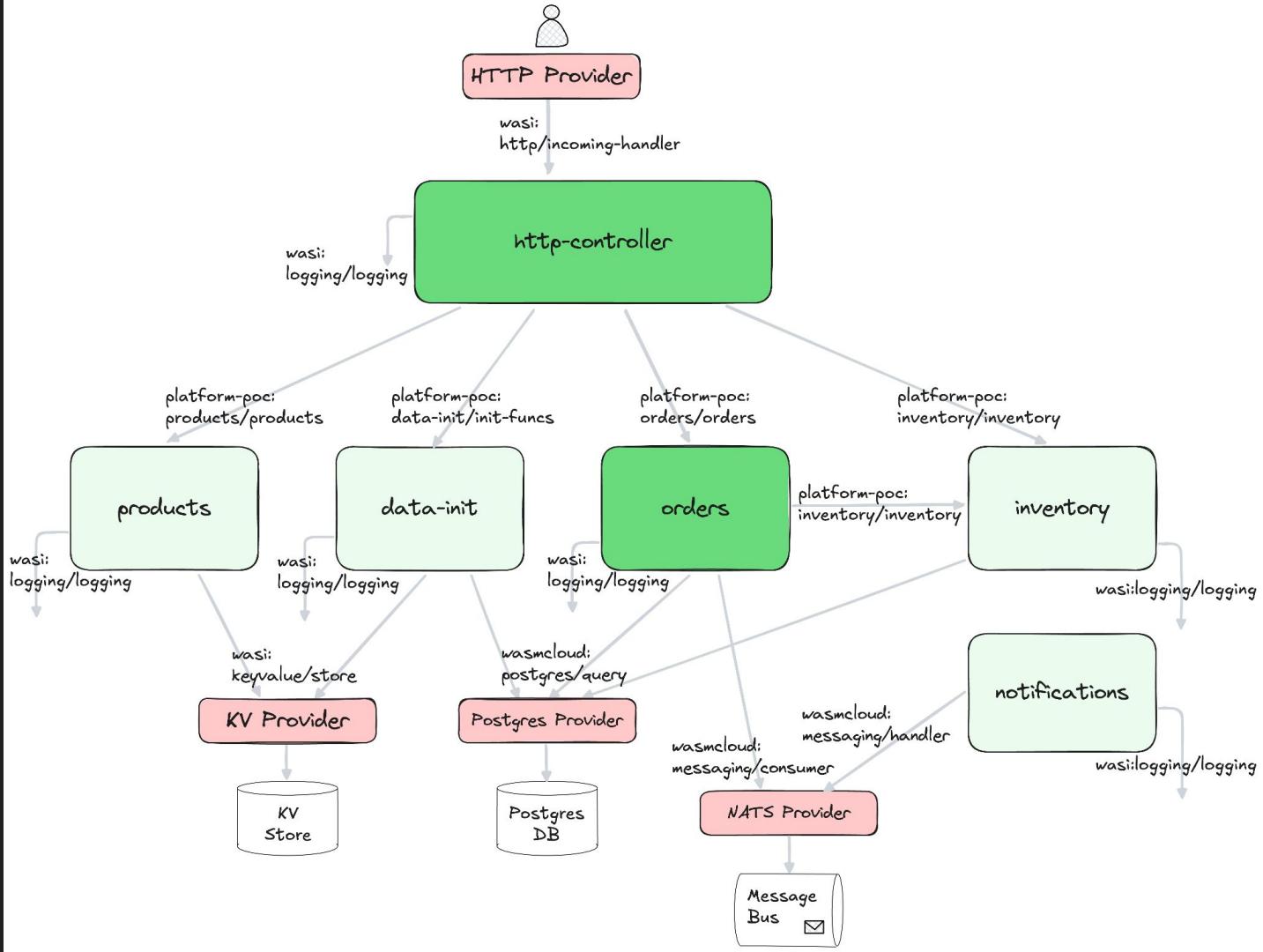
# Demo



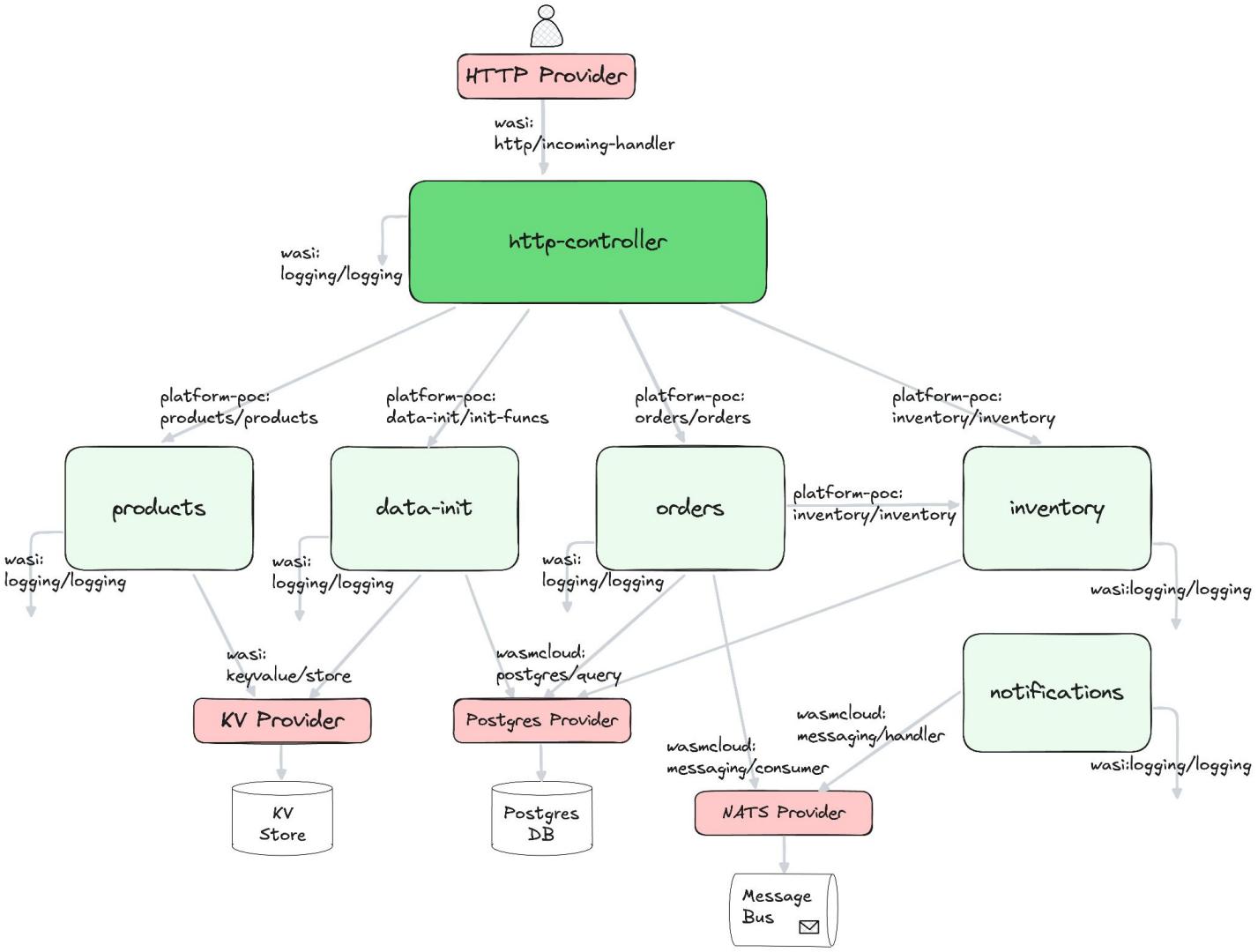
# Demo



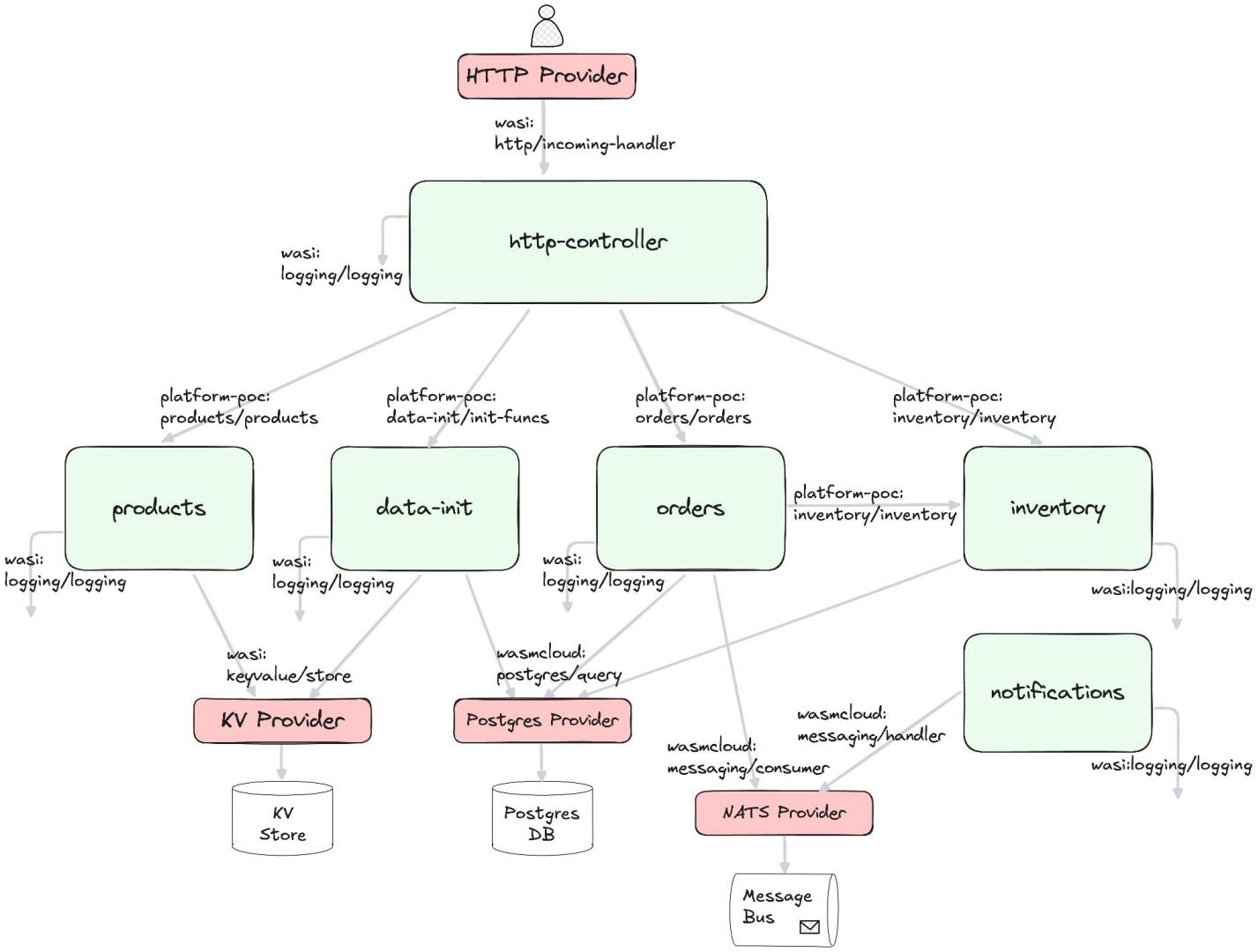
# Demo



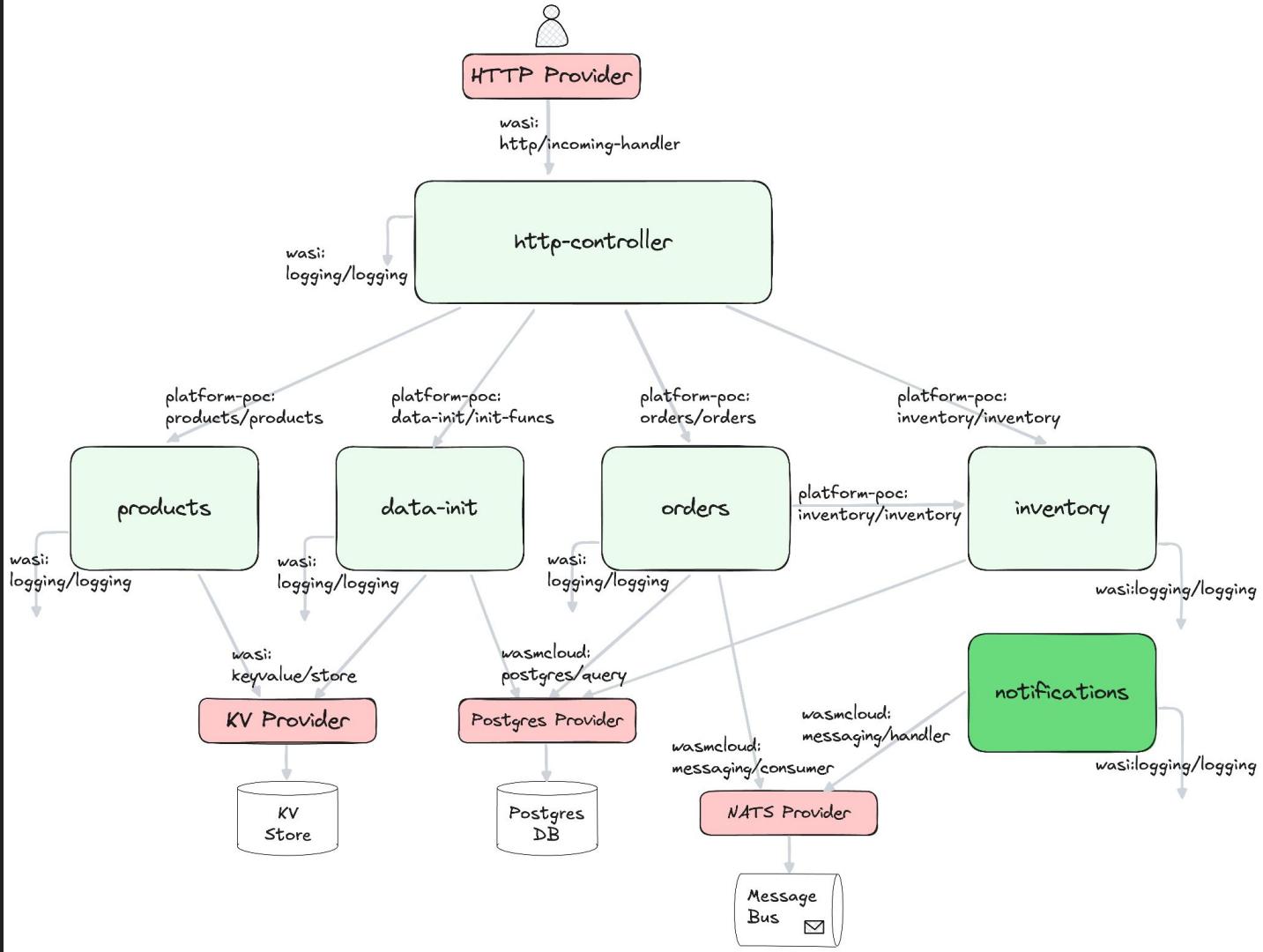
# Demo



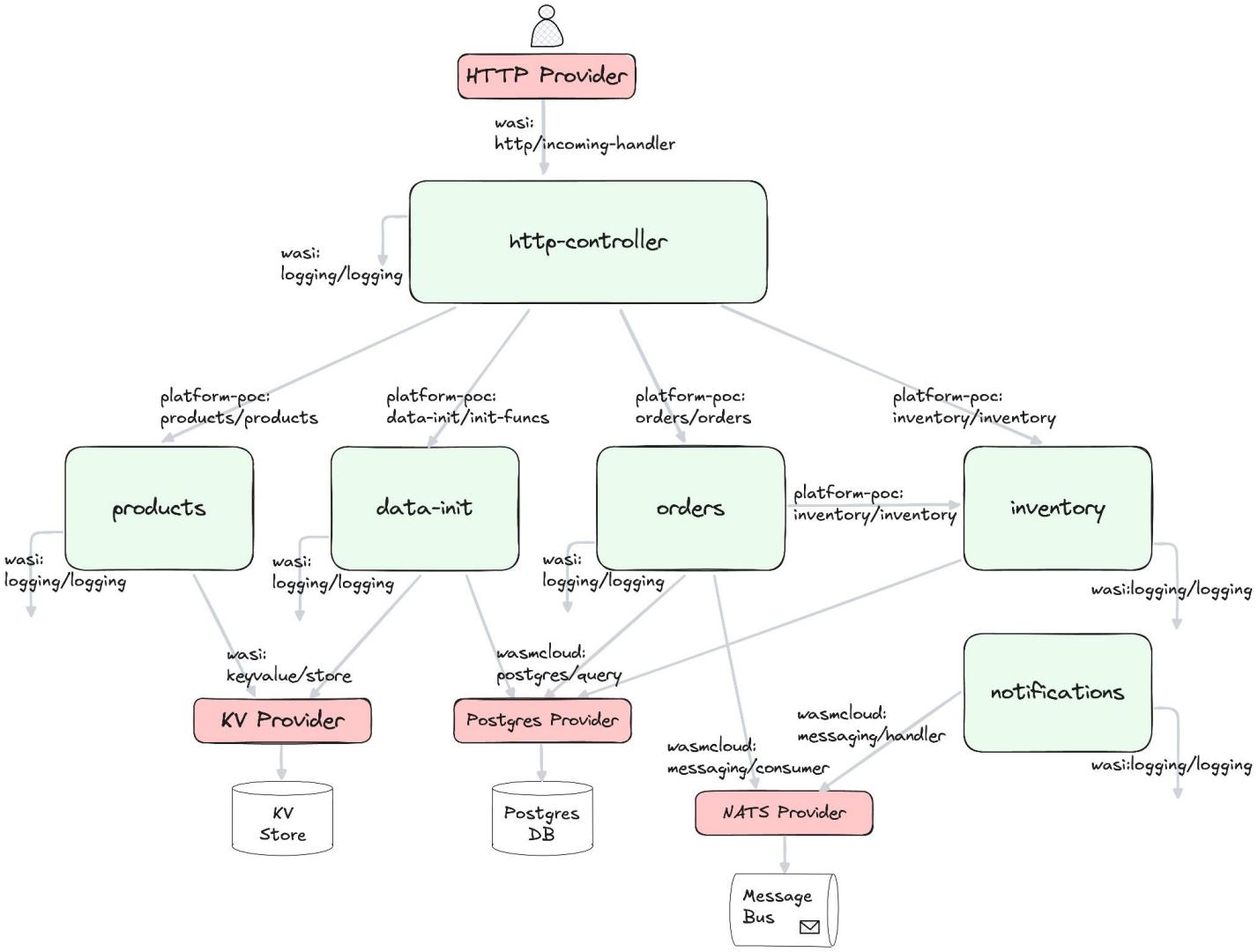
# Demo



# Demo



# Demo



# Placing 10K orders

```
→ oha 'http://localhost:8080/orders' \
      -m POST \
      -d '[{"sku": "ENG-V8-500", "price": 1000, "quantity": 1}]' \
      -c 20 \
      -n 10000
```

## Summary:

```
Success rate: 100.00%
Total:        4.3168 secs
Slowest:      0.0484 secs
Fastest:      0.0037 secs
Average:      0.0086 secs
Requests/sec: 2316.5356
```

# Wasm binaries are small

About 1000 times smaller than their equivalent Java Spring Boot container image.

```
139k data_init_s.wasm
351k http_controller_s.wasm
 99k inventory_service_s.wasm
148k notification_service_s.wasm
159k orders_service_s.wasm
165k products_service_s.wasm
```

## Fermyon.com on Spin

*“A fresh page load of Fermyon.com may result in starting, executing, and shutting down 30+ concurrent WebAssembly processes. Yet the result is a page that loads so fast that it scores a 99% on Google’s page speed ranking.”*

*“Not long ago, we had a very busy day. Our traffic jumped to ten times its normal number of page requests. At a few moments, we were seeing over 2,500 requests coming in at a time (90 concurrent users each fetching around 30 resources). In our model, we run three workers, each on a small-sized VM with 3 CPUs and 300Mb of memory per worker.”*

# Rust

**Safety and speed.**

Increased productivity, really.

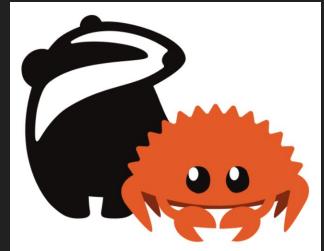
Shift left on quality.

Rust loves WebAssembly:

```
→ cargo build --target wasm32-wasip1
```

Build WebAssembly components on nightly:

```
→ cargo +nightly build --target wasm32-wasip2
```



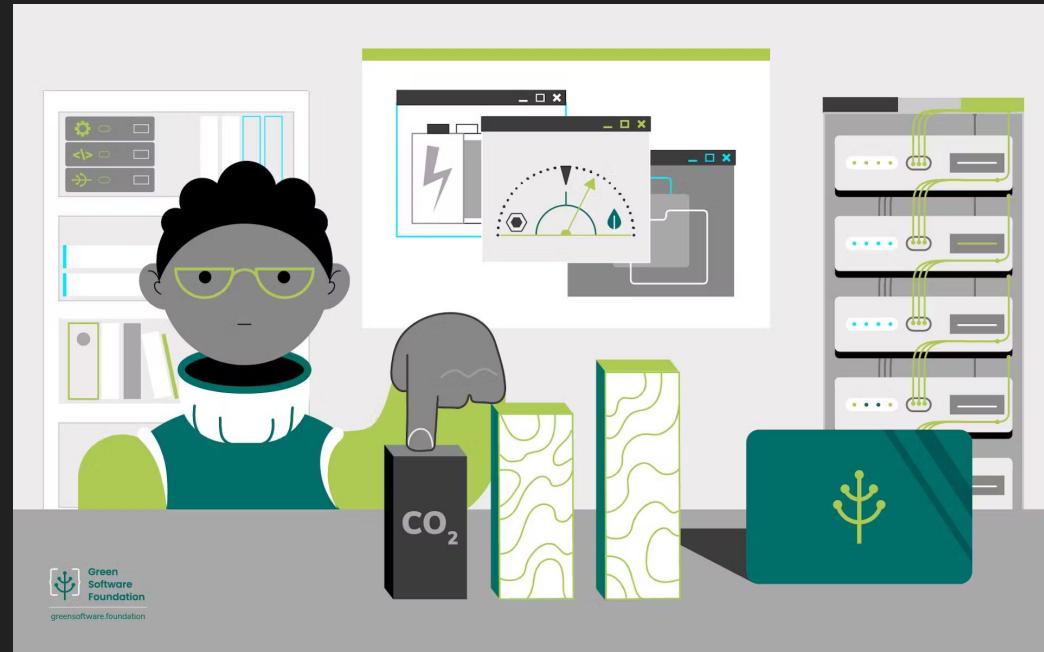
# Roadmap

- Production ready — build greenfield scale-to-zero applications today
- Create a hybrid Kubernetes cluster using
  - ◆ wasmCloud Operator
  - ◆ SpinKube
- Gradually migrate, service by service, to new architecture
  - ◆ Remove all side effects, leaving just business logic

*Let a decade of containers give way to a decade of WebAssembly components!*

# Let's build eco-friendly enterprise applications

- The Green Software Foundation [ref](#)
- The Principles of Sustainable Software Engineering [ref](#)



Thank you! 🙏

[slides](#)



Stuart Harris  
[@stuartharris](https://twitter.com/stuartharris)  
[stuart.harris@red-badger.com](mailto:stuart.harris@red-badger.com)

<https://github.com/rebadger/platform-poc>