Disrupting the Downtime Continuum

Brooks Townsend and Taylor Thomas

Who are we?

Brooks Townsend

- Lead Software Engineer at Cosmonic
- wasmCloud maintainer
- Serial open source contributor
- Brewer of elixir, Wasm enjoyer
- Demo enthusiast

Taylor Thomas

- Director of Customer Engineering at Cosmonic
- Rustacean
- Co-creator of Krustlet and Bindle
- Open Source Maintainer
- Emeritus Helm Maintainer

Agenda

- What is WebAssembly?
- What’s this wasmCloud and Cosmonic thingy?
- Demo time
- What can I do right now?
- How to get involved

Neither Web, nor Assembly

Open W3C Standard

Open and widely supported standard

Safe & Secure

Deny by default secure sandbox, featuring capability driven permissions

Efficient and fast

Small size and near-native execution speed

Polyglot

Choice of deployment language means ability to reuse existing libraries

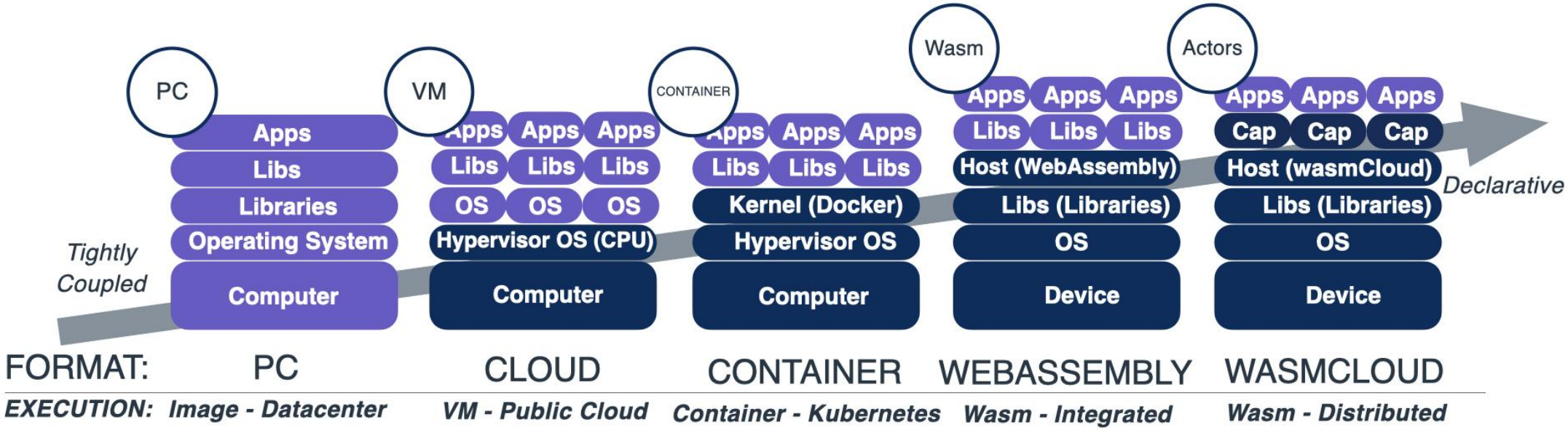
Portable

WebAssembly runs in all major browsers

But there are some gaps

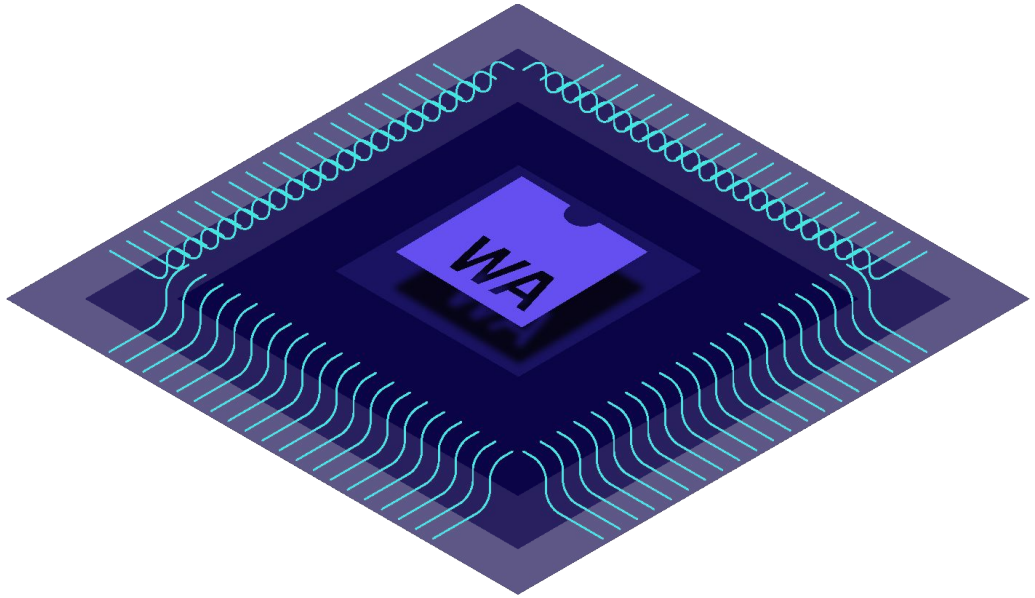
- Language support is still limited (but quickly growing!)
- Networking is rudimentary or non-existent
- Still have to compile your dependencies into the final binary
- Numbers in, numbers out

Modern Computing Env



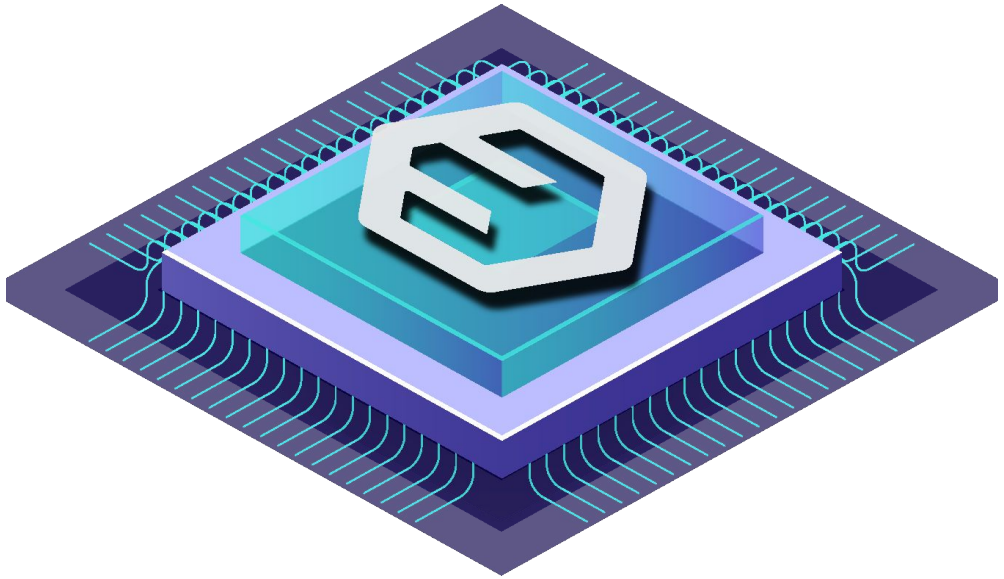
Legend: Developer Provided / Device / Computer

WebAssembly Host Runtime



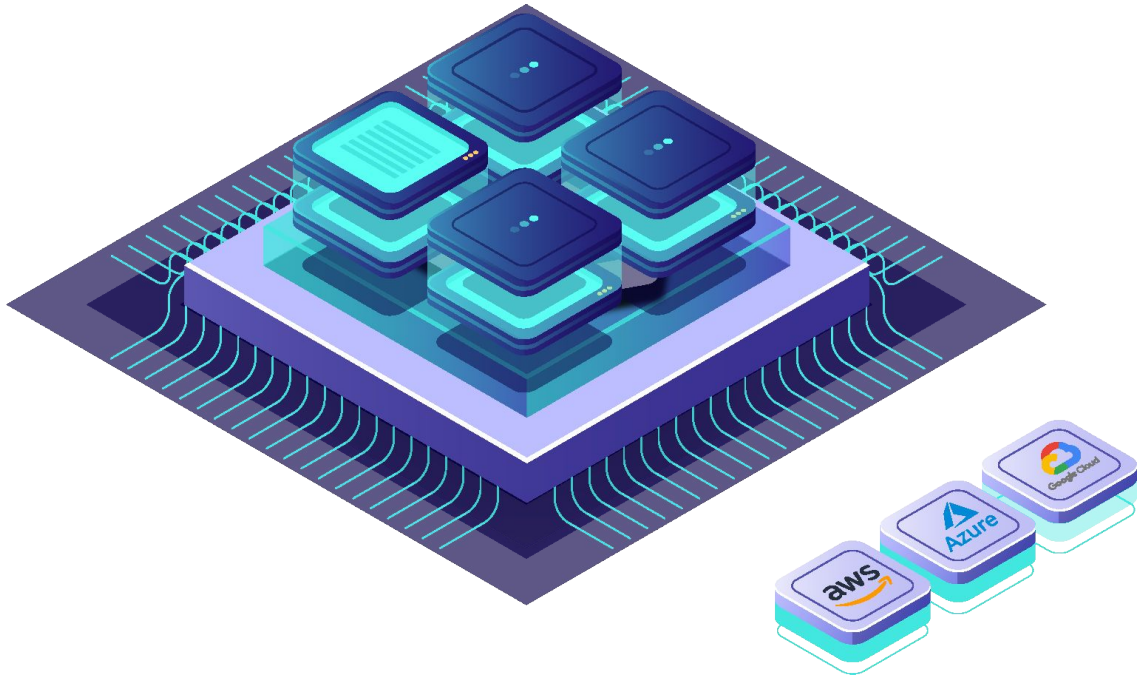
- Portable
- Secure
- Small
- Fast
- Language agnostic

wasmCloud Application Runtime



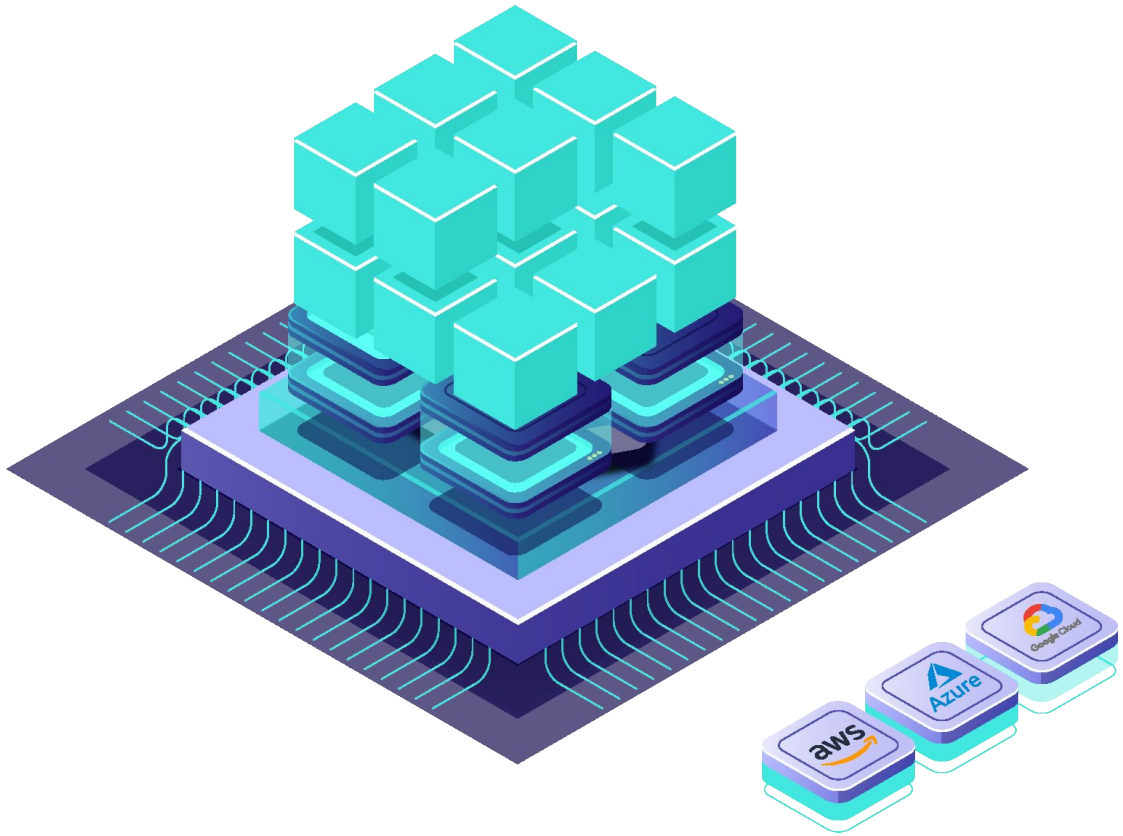
- Removed boilerplate code
- Secure access to capabilities
- Elixir/OTP - Extreme Scalability
- Horizontally and vertically scalable, stateless actors

Capabilities



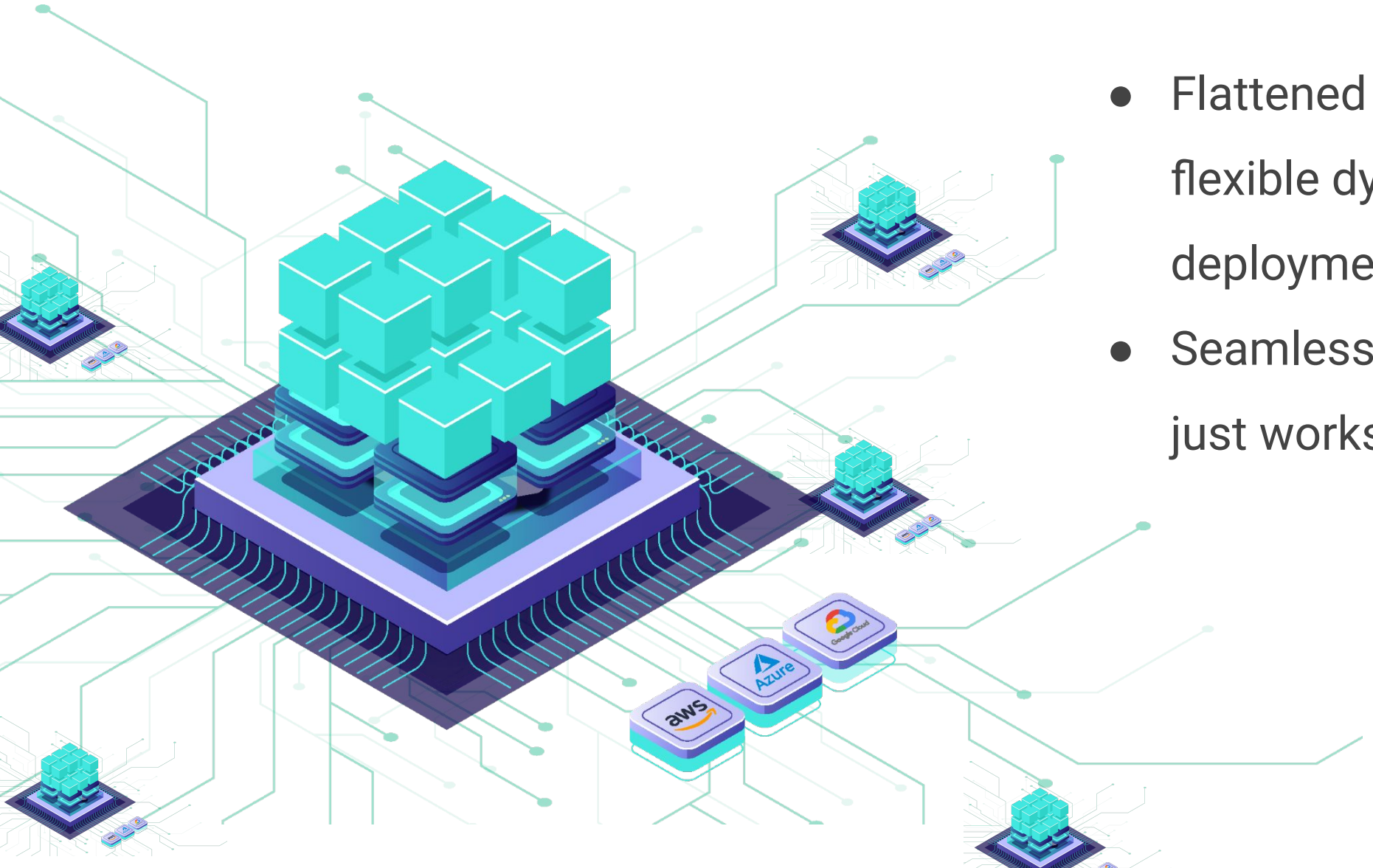
- **Maintain & update centrally - stop distributing vulnerable boilerplate**
- Runtime choice of capabilities, hot swap
- Contract driven design

Composable Actors



- Implement your business logic
- Stateless and reactive
- Easy to develop & low boilerplate
- Tiny footprint, portable & scalable

Lattice Network



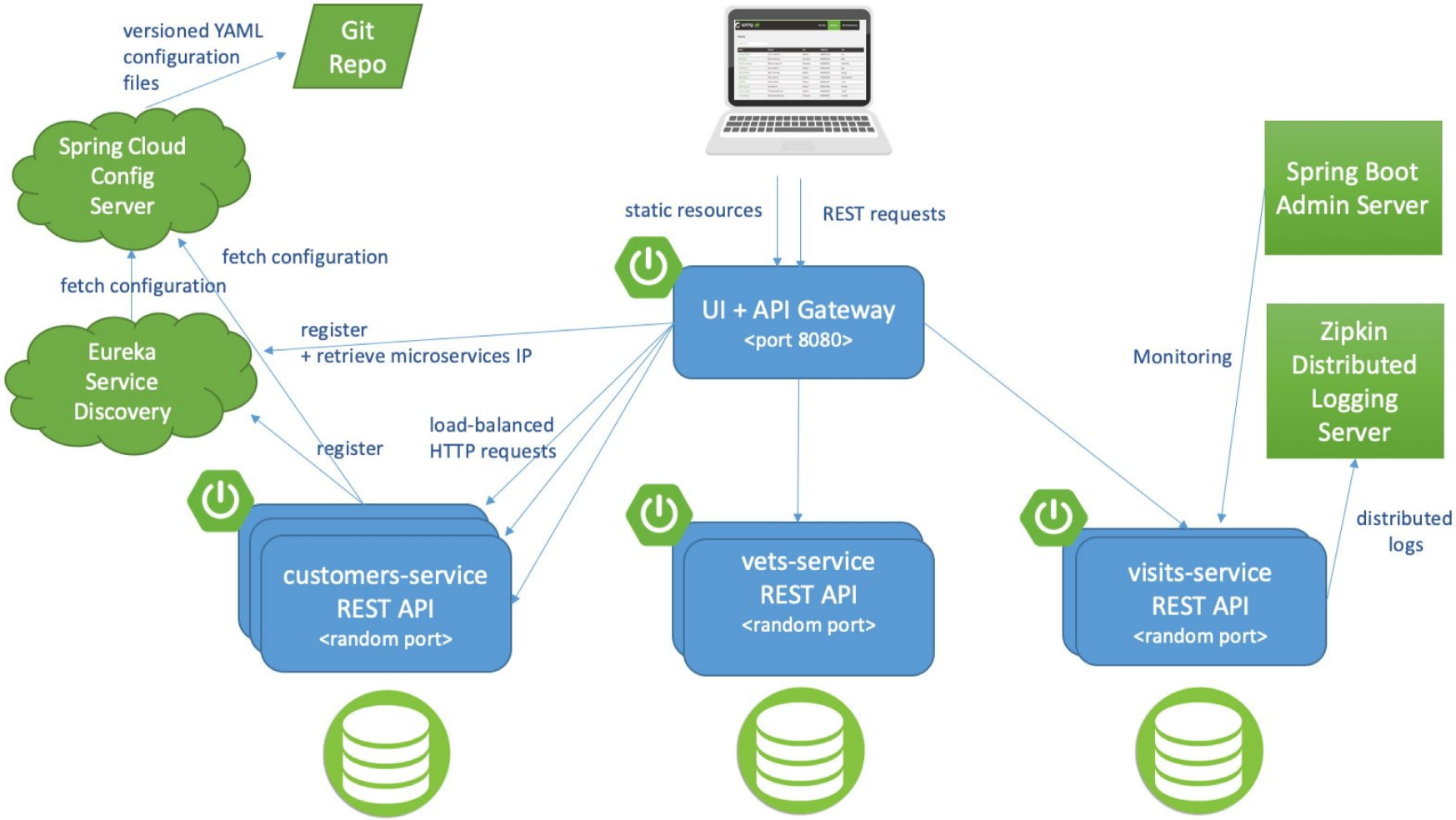
- Flattened topology, enables flexible dynamic deployments
- Seamlessly connected, “it just works”

How does Cosmonic fit into this?

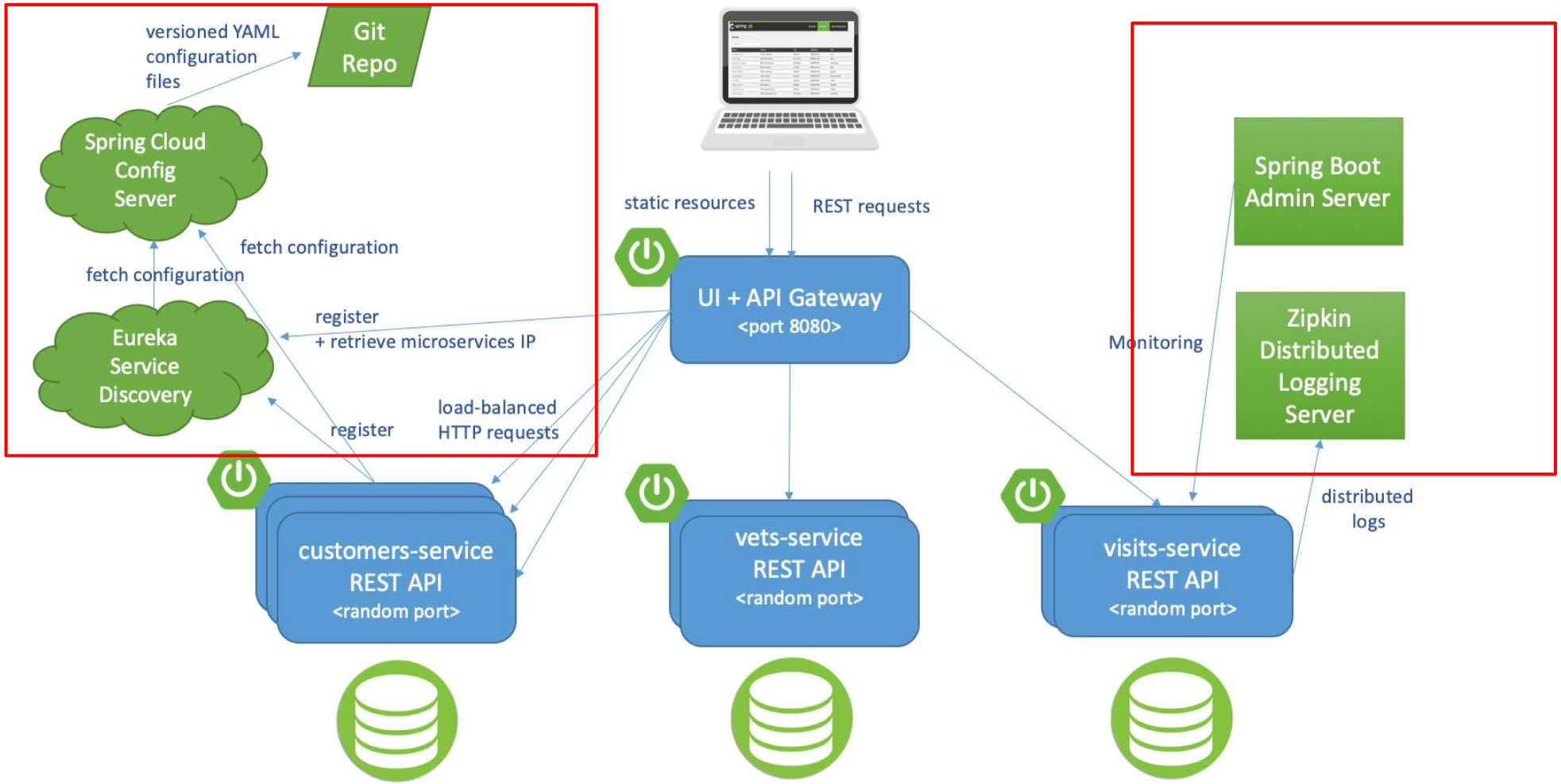
- Making it a painless experience to develop server-side wasm apps
- Easy cross-platform/cloud/device management
- Someone has to manage the thing at some point
- Everything is built on wasmCloud

DEMO TIME

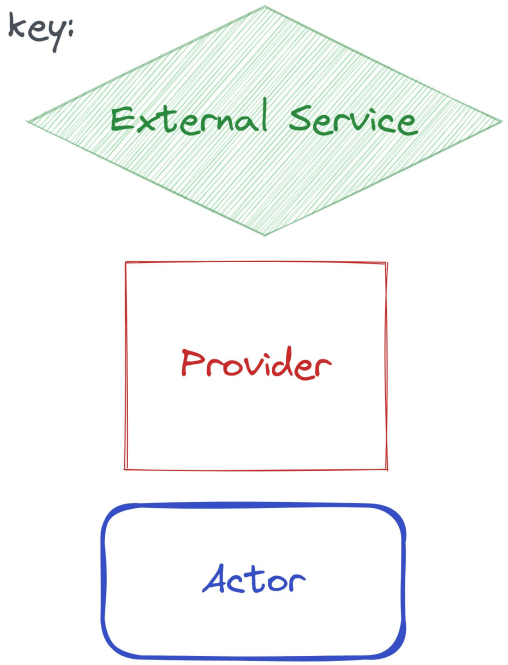
Spring Boot Pet Clinic

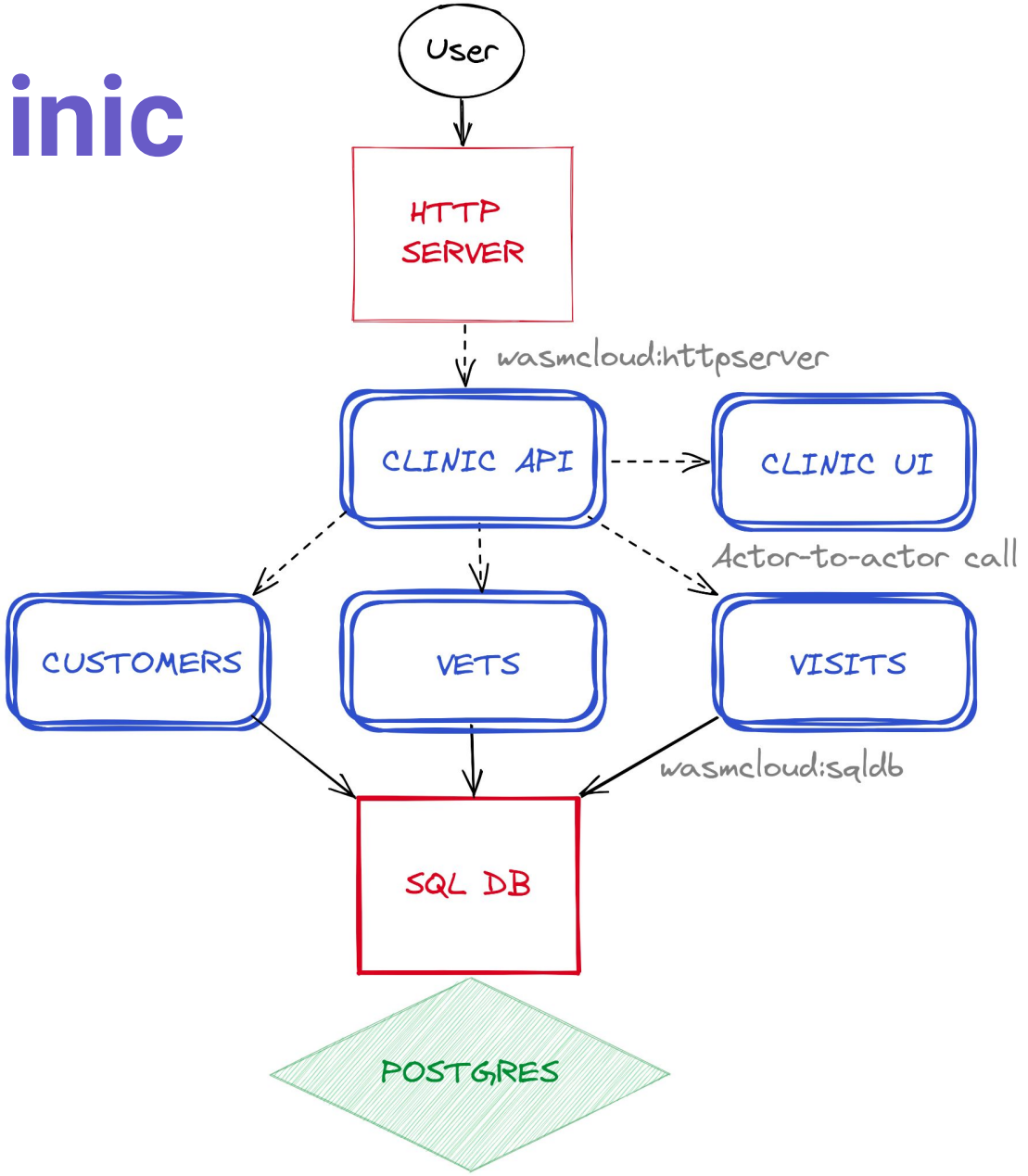


Spring Boot Pet Clinic

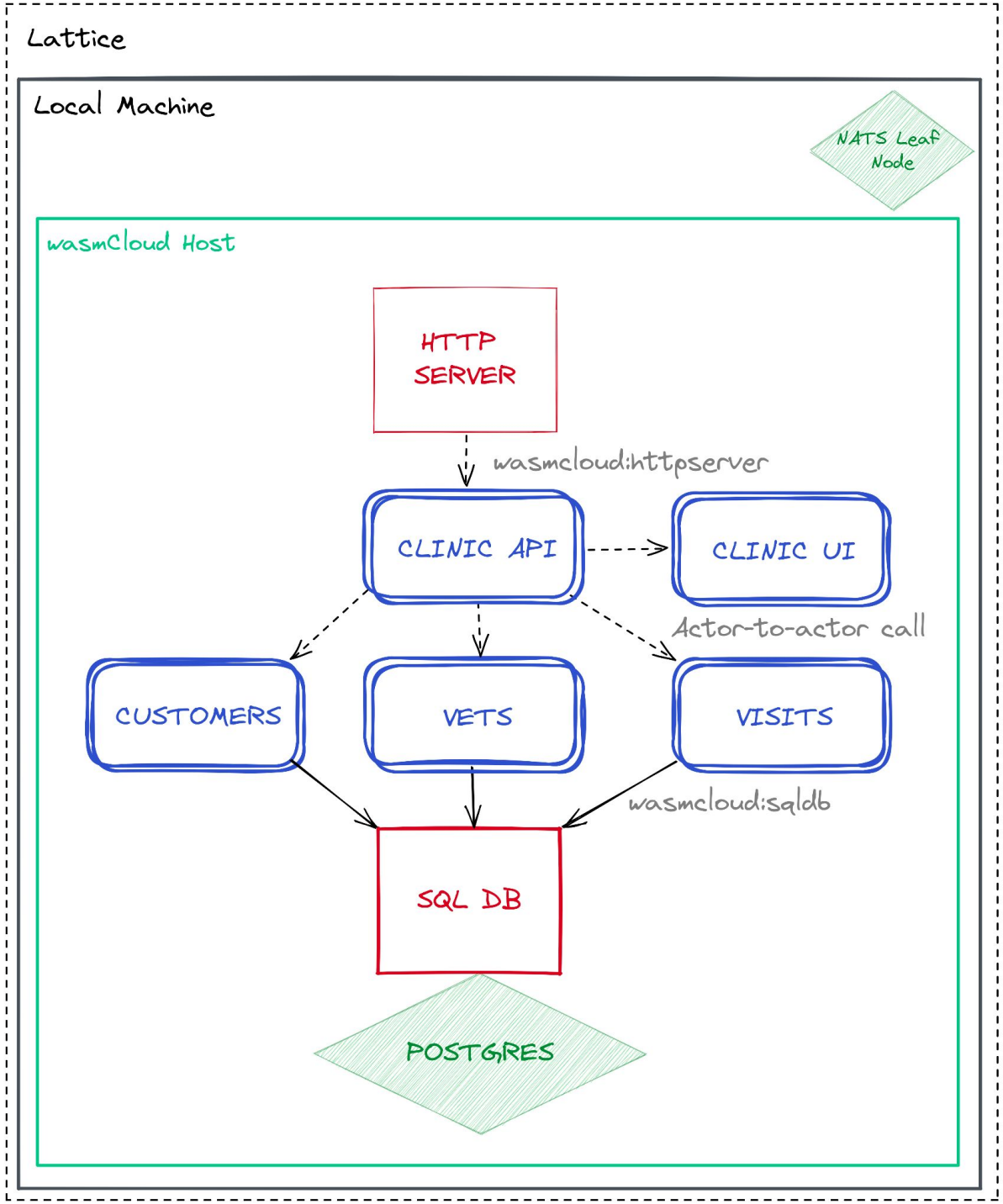


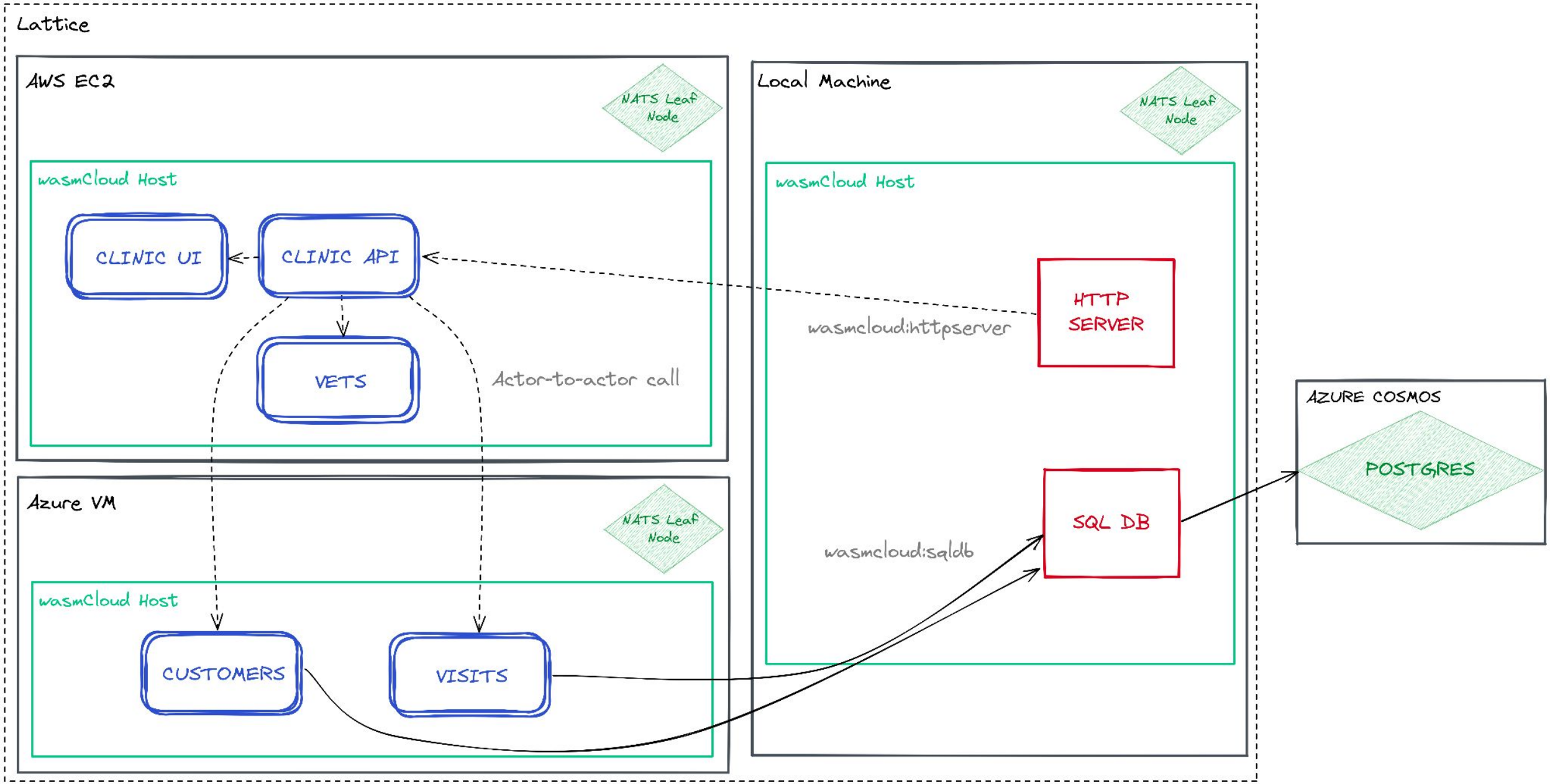
wasmCloud Pet Clinic





DEMO TIME ROUND 2: FIGHT!


Lattice
Local Machine
NATS Leaf Node
wasmCloud Host
HTTP SERVER
wasmcloud:httpserver
CLINIC API
CLINIC UI
Actor-to-actor call
CUSTOMERS
VETS
VISITS
wasmcloud:sqldb
SQL DB
POSTGRES


Lattice
AWS EC2
NATS Leaf Node
wasmCloud Host
CLINIC UI
CLINIC API
VETS
Actor-to-actor call
Local Machine
NATS Leaf Node
wasmCloud Host
HTTP SERVER
wasmcloud:httpserver
SQL DB
wasmcloud:sqldb
AZURE COSMOS
POSTGRES
Azure VM
NATS Leaf Node
wasmCloud Host
CUSTOMERS
VISITS

But how am I gonna use it?

What could you do now?

1. Basic data/image/etc. processing
2. One small part of a service
    - Runs smaller and cheaper
3. A full stateful application
    - Use the various providers to connect to the data sources you need

References

- https://slack.wasmcloud.com/
- https://github.com/wasmCloud/wasmCloud
- Additional resources
  - https://github.com/cosmonic/kubernetes-applier
  - https://github.com/wasmCloud/capability-providers
  - https://github.com/wasmCloud/interfaces

Join our community Slack and check out our GitHub!

https://slack.wasmcloud.com

https://github.com/wasmCloud/wasmCloud