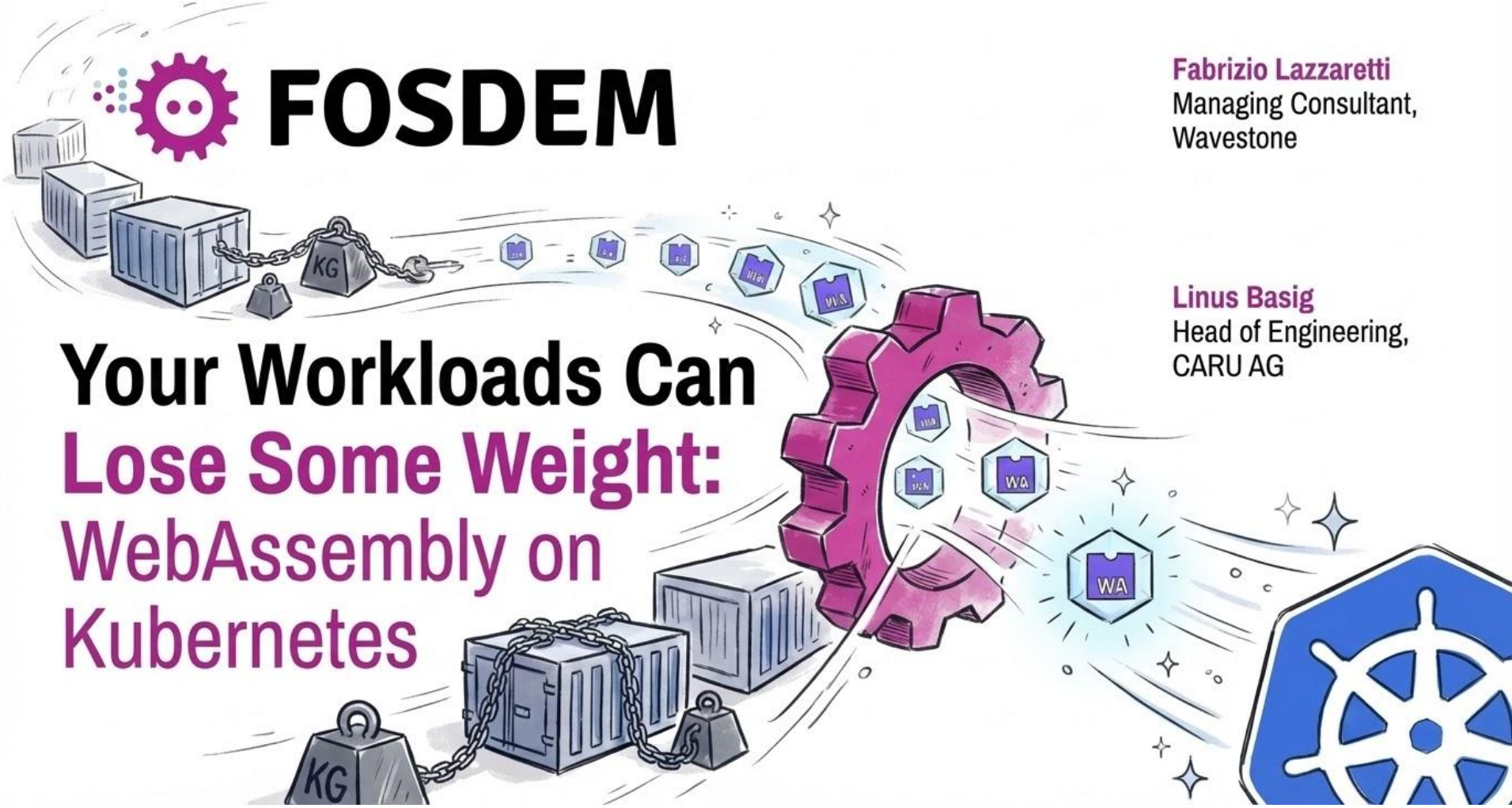
FOSDEM

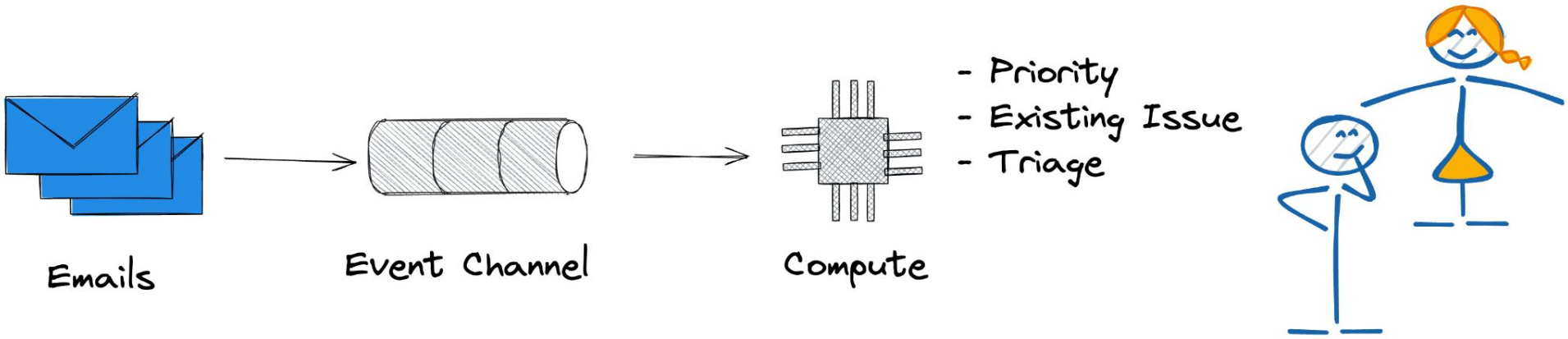
Fabrizio Lazzaretti
Managing Consultant,
Wavestone

Linus Basig
Head of Engineering,
CARU AG

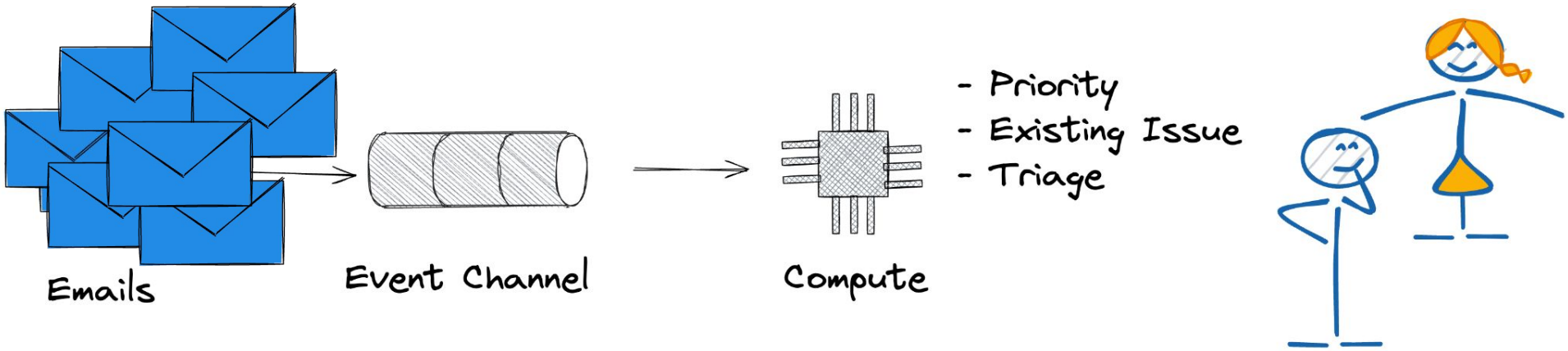
Your Workloads Can Lose Some Weight: WebAssembly on Kubernetes



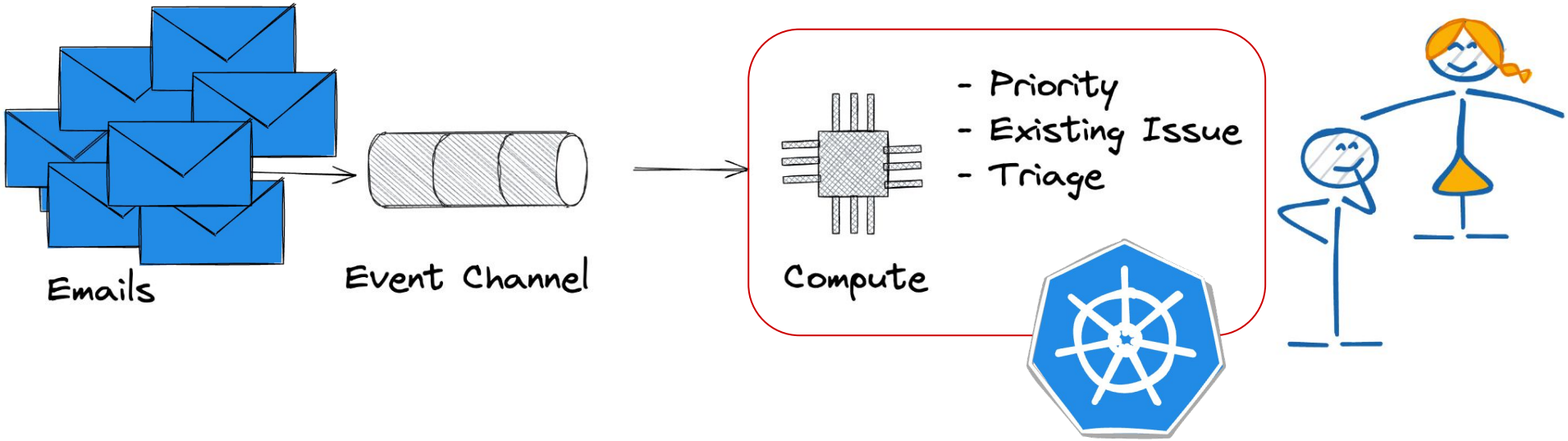
You got Mail!


- Priority
- Existing Issue
- Triage
Emails
Event Channel
Compute

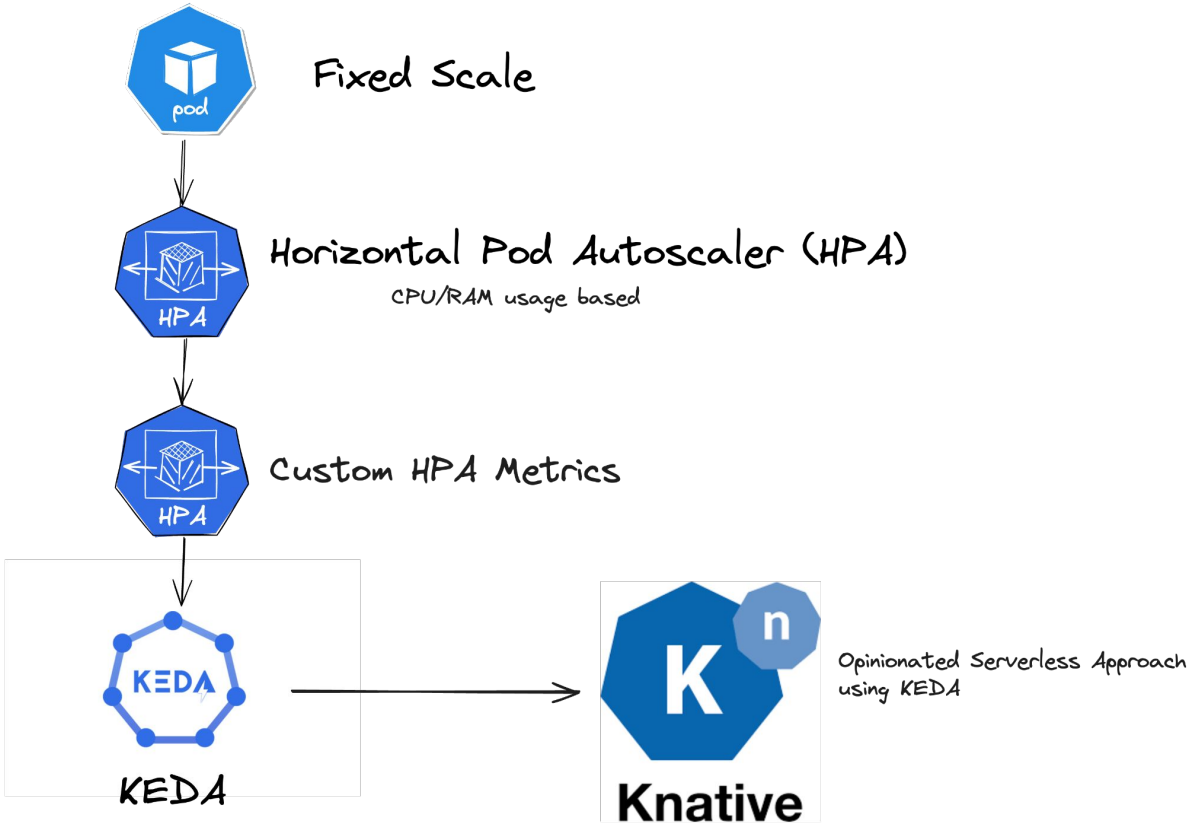
You got Mail!


Emails
Event Channel
Compute
- Priority
- Existing Issue
- Triage

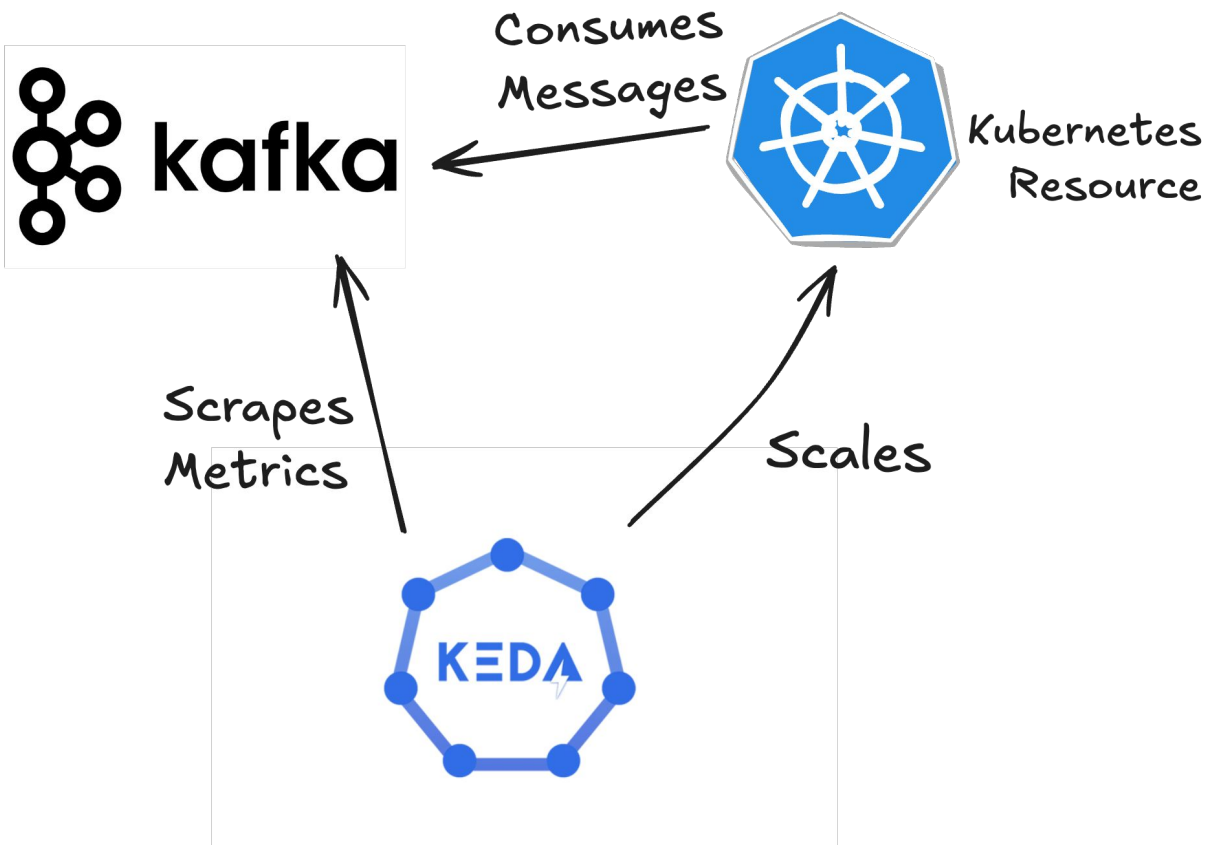
You got Mail!


How do we scale?
Emails
Event Channel
- Priority
- Existing Issue
- Triage
Compute

Scaling Event-Driven Architecture on Kubernetes



How does KEDA work?



Now that we have scaling

How do we make it fast?

DISCLAIMER: Cutting Edge

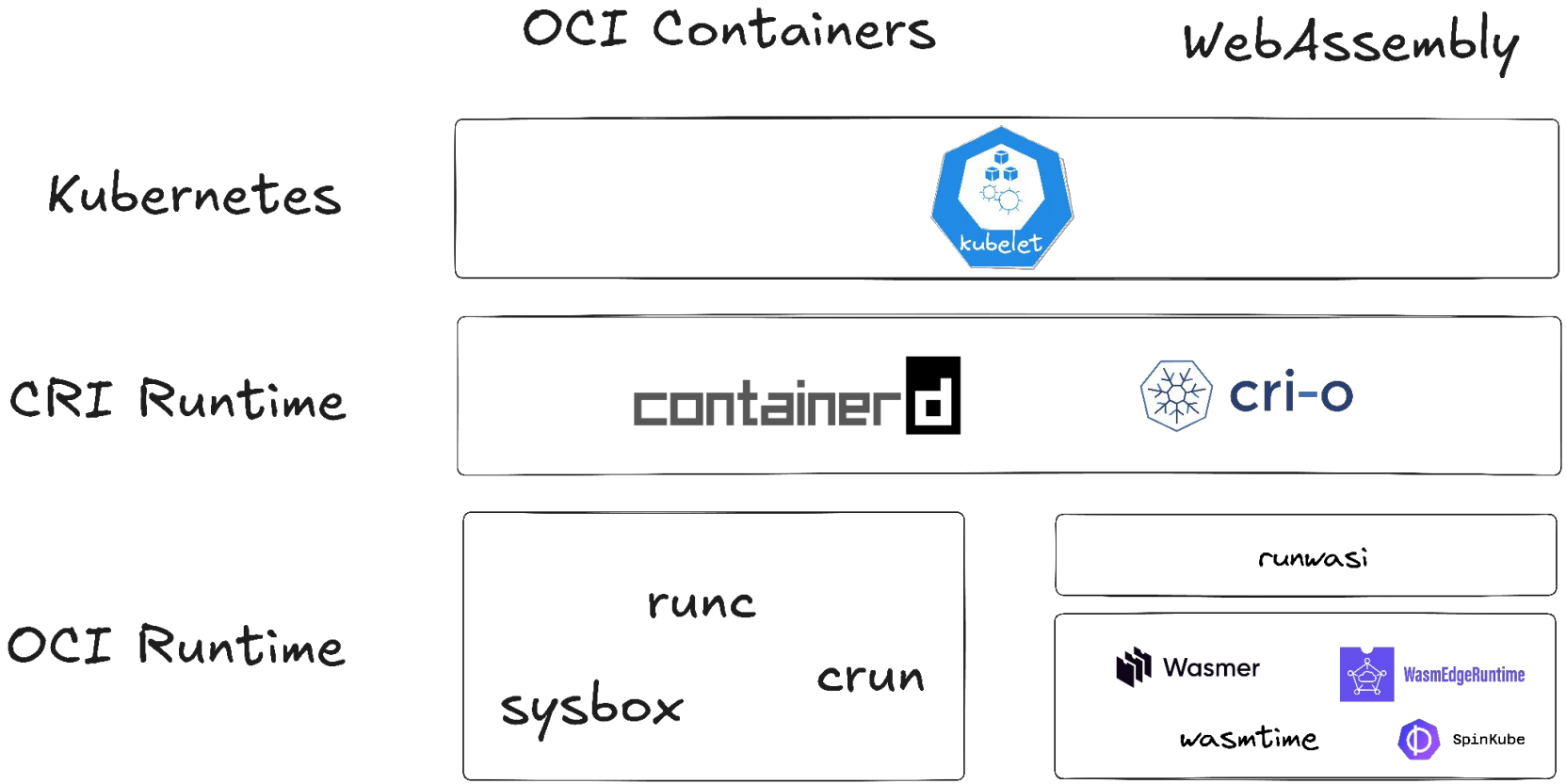
Working with WebAssembly on Kubernetes may lead to unexpected emotional distress. Side effects may include, but are not limited to:

- Frustration from complex configurations
- Anxiety due to compilation issues
- Despair when debugging across multiple layers of abstraction
- Euphoria upon successful deployment (followed by crashes)

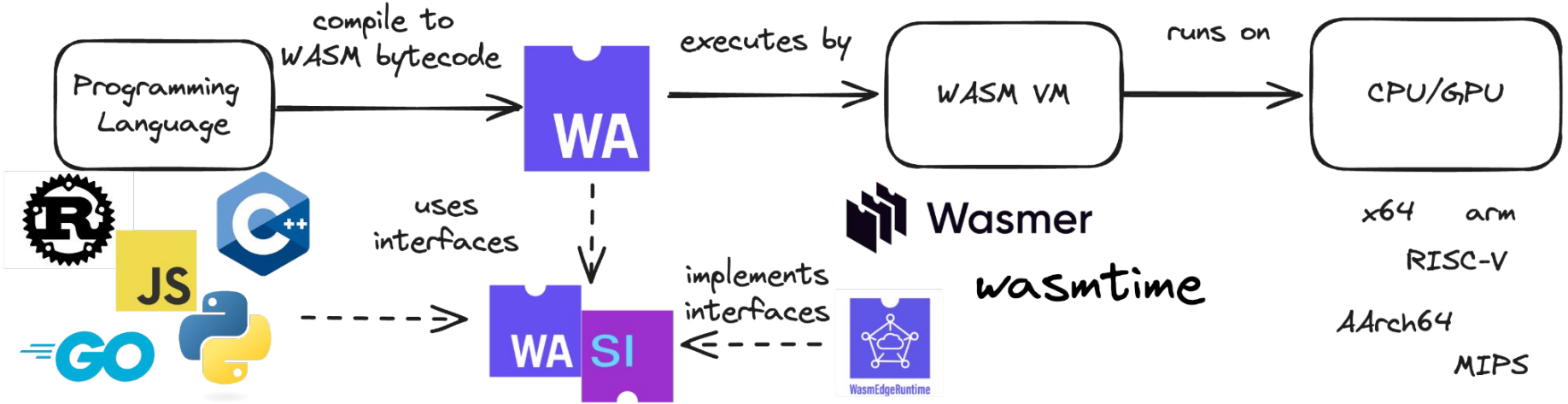
Users are advised to proceed with caution and maintain a sense of humor. Remember: it's just code, not a reflection of your worth as a human being.

Consult your nearest DevOps therapist if symptoms persist for more than two sprints.

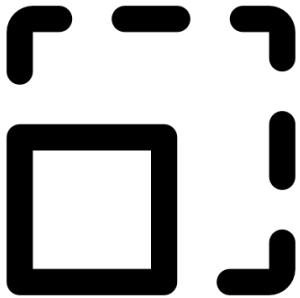
WebAssembly on Kubernetes



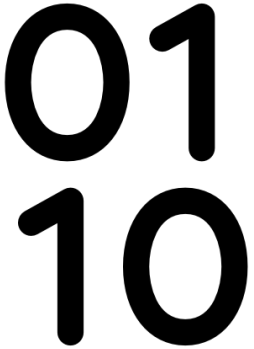
WebAssembly in a nutshell



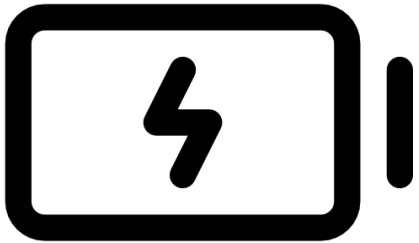
How does WebAssembly (Wasm) help with speed?



Tiny Images

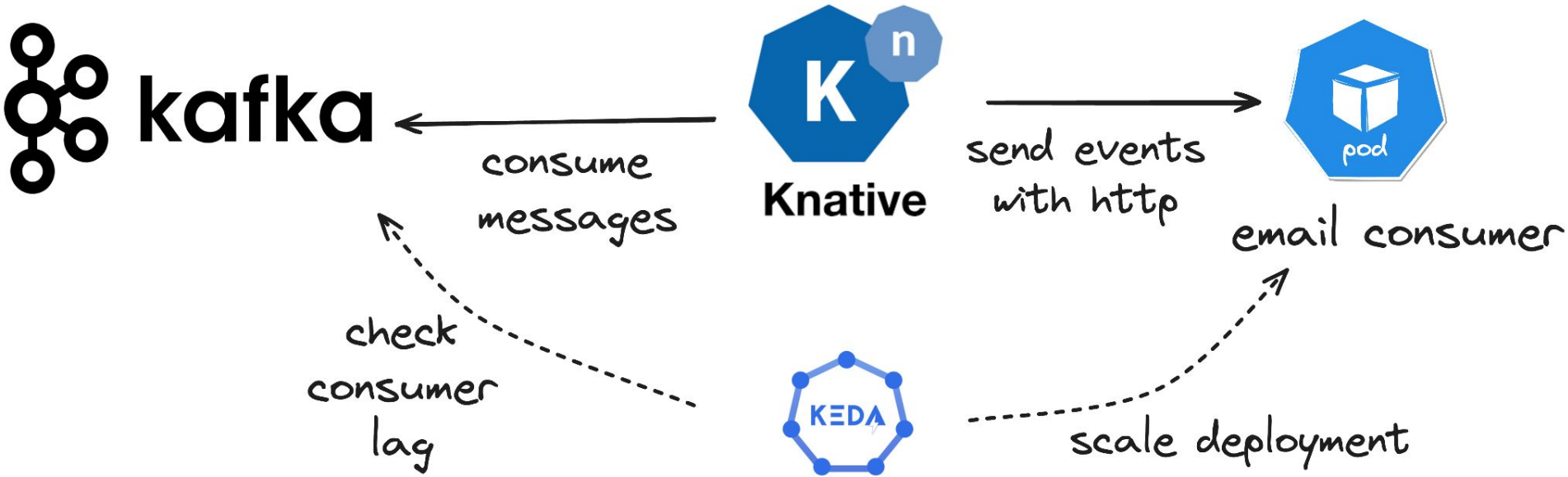


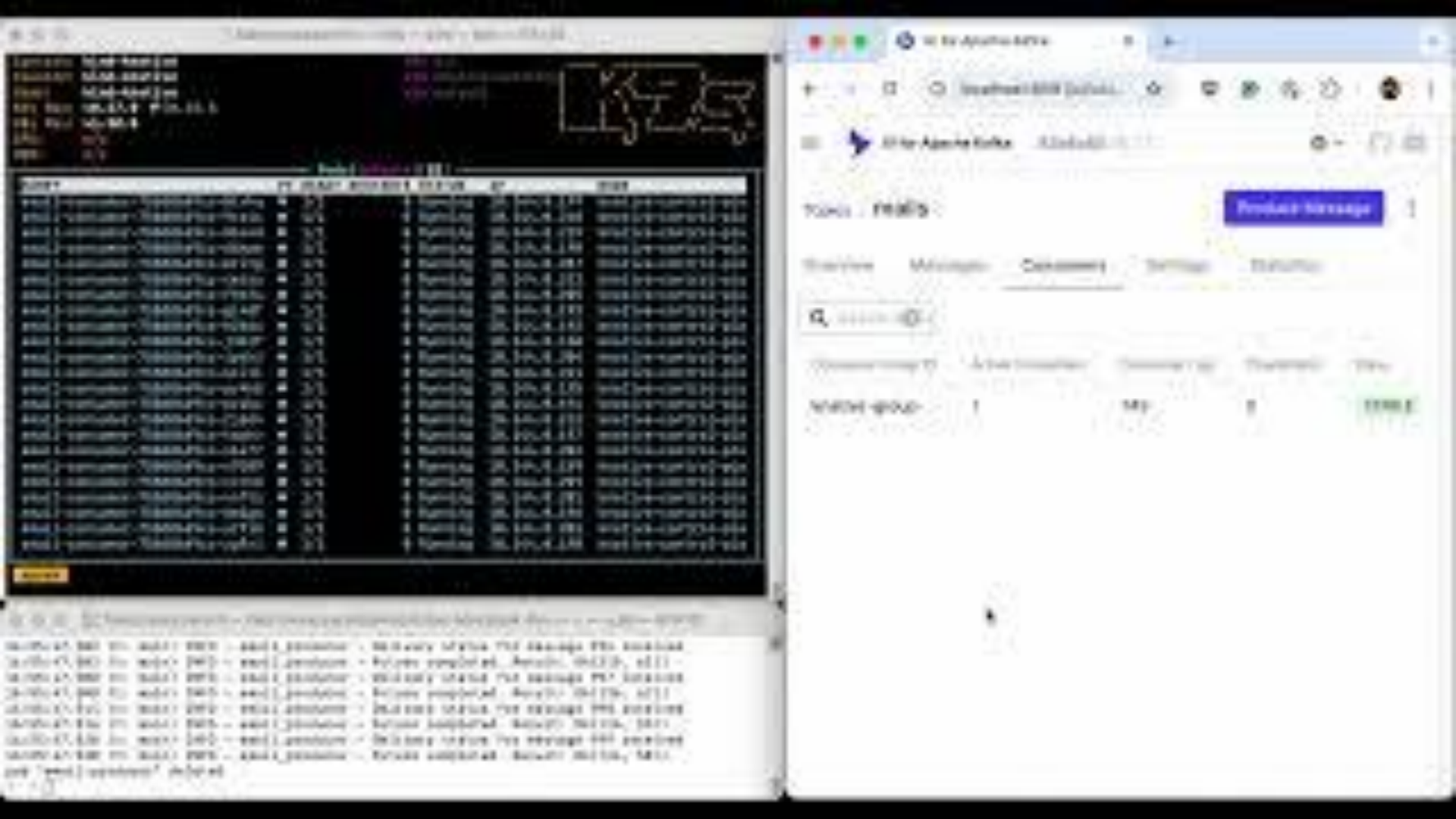
Bytecode



Batteries Included

Demo Architecture





Productive usage

WASI is in v0.2

- v0.3: expected around February 2026
  - Adds native async support
- v1: sometime in the future

CNCF Usage of WASM

allows extensions with custom logic to compile to WASM using the http-handler

allows filtering of requests with the Proxy-Wasm sandbox API

added support for Wasm plugins in Helm 4

Dapr usage of WASI (simplified)

Wasm http-handler in TinyGo

```go
package main

import (
    "strings"

    "github.com/http-wasm/http-wasm-guest-tinygo/handler"
    "github.com/http-wasm/http-wasm-guest-tinygo/handler/api"
)

func main() {
    handler.HandleRequestFn = handleRequest
}

// handleRequest implements a simple HTTP router.
func handleRequest(req api.Request, resp api.Response) (next bool, reqCtx uint32) {
    // If the URI starts with /host, trim it and dispatch to the next handler.
    if uri := req.GetURI(); strings.HasPrefix(uri, "/host") {
        req.SetURI(uri[5:])
        next = true // proceed to the next handler on the host.
        return
    }

    // Serve a static response
    resp.Headers().Set("Content-Type", "text/plain")
    resp.Body().WriteString("hello")
    return // skip the next handler, as we wrote a response.
}
```

https://docs.dapr.io/reference/components-reference/supported-middleware/middleware-wasm/#generating-wasm

Run WASM (host process)

```go
import "github.com/tetratelabs/wazero"

// ...

r := wazero.NewRuntime(ctx)
defer r.Close(ctx)
mod, _ := r.Instantiate(ctx, wasmAdd)
res, _ := mod.ExportedFunction("add").Call(ctx, 1, 2)
```

https://wazero.io/

How to get started?

1. Familiarize yourself with the WebAssembly Component Model.
2. Familiarize yourself with the WebAssembly System Interface.
3. Implement a Hello World in Rust
4. Choose a WebAssembly Runtime
   a. Wasmer
   b. WasmEdge
   c. WAMR
   d. Wasmtime
5. Prepare Kubernetes Nodes
   a. runwasi
   b. crun

FOSDEM

Fabrizio Lazzaretti
Managing Consultant,
Wavestone

Linus Basig
Head of Engineering,
CARU AG

https://lazzaretti.me/fosdem26/

Credits & License

Used Work

Excalidraw images (MIT License)

- Comms Platform Icons @Adam K Dean https://libraries.excalidraw.com/#adamkdean-comms-platform-icons
- Kafka Streams Topology Design @Hartmut Armbruster https://libraries.excalidraw.com/#hartmut-co-uk-kafka-streams-topology-design
- Computer parts @Bruno Rocha https://libraries.excalidraw.com/#rochacbruno-computer-parts
- Stick people @David Hurt https://libraries.excalidraw.com/#dhtoran-stick-people
- Kubernetes icons @Nikola Markovic https://libraries.excalidraw.com/#boemska-nik-kubernetes-icons

Phosphor Icons (MIT License): https://phosphoricons.com/

References & Links

- https://keda.sh/
- https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/
- https://webassembly.org/
- https://wasi.dev/
- https://containerd.io/
- https://github.com/containerd/runwasi
- https://wasmedge.org/
- https://wasmtime.dev/
- https://wasmer.io/
- https://runwasi.dev
- https://archive.fosdem.org/2024/schedule/event/fosdem-2024-2769-the-jvm-vs-webassembly-an-in-depth-comparative-analysis/
- https://helm.sh/docs/overview/#plugin-system-overhaul
- https://wasi.dev/roadmap
- https://docs.dapr.io/reference/components-reference/supported-middleware/middleware-wasm/
- https://istio.io/latest/docs/concepts/wasm/