

BRIDGING WASM AND KUBERNETES

David Justice

Principal Engineer at Microsoft

Co-chair of CNCF Wasm Working Group



What are we going to learn today?

What makes WebAssembly so interesting

How to naively run Wasm in Kubernetes

The beautiful future of Wasm in Kubernetes

How Wasm is going to change platform engineering

What you can do to help build the future



WASM IS NOT JUST FOR THE BROWSER



Why server- side Wasm?

Operating system and architecture agnostic

Near native execution speed / fast start up

Secure by default

Language agnostic



Redefines interoperability and platform engineering




NAÏVE CONTAINER IMAGE

Let's start with something familiar



example1 >  hello.rs >  main

```
1 fn main() {  
2     println!("Hello, world!");  
3 }
```

example1 >  Makefile

```
1  
2 .PHONY: run  
3 run:  
4     rustup target add wasm32-wasi  
5     rustc hello.rs --target wasm32-wasi  
6     wasmtime hello.wasm
```

EVERYTHING STARTS WITH "HELLO WORLD!"




```
2
3 FROM rust:1.74.0
4 WORKDIR /app
5 COPY . /app
6 RUN rustup target add wasm32-wasi && rustc hello.rs --target wasm32-wasi
7 RUN curl https://wasmtime.dev/install.sh -sSf | bash
8 CMD [ "/root/.wasmtime/bin/wasmtime", "hello.wasm" ]
```

NAÏVE CONTAINER IMAGE EXAMPLE



⏮ ⏭ 🔍 kcd-2023

📄

EXPLORER

⋮

⌵ KCD-2023

⌵ example1

🔍 .gitignore

🚢 Dockerfile

🌐 hello.rs

📄 hello.wasm

📄 Makefile

⌵ example2

🔑 LICENSE

📄 README.md

⌵ OUTLINE

⌵ TIMELINE

⌵ RUST DEPENDENCIES

! kind-config.yaml

M Makefile example1

🌐 hello.rs

🚢 Docker

⋮

example1 > 🌐 hello.rs > 📦 main

1 fn main() {

2 | println!("Hello, world!");

3 }

⏮

main ↺

⊗ 0 ⚠ 0 ⓘ 2

🔊 0

🔄 Live Share

⊗ rust-analyzer

Ln 3, Col 2

Spaces: 4

UTF-8

LF

Rust

✓ Spelling



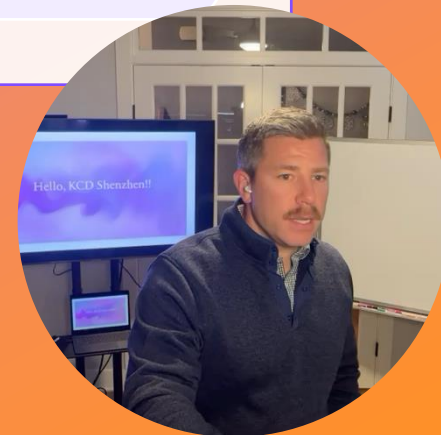
What's wrong with that image?



OS / Arch
specific image

Contains the
Wasmtime
runtime

2.69 GB!!



WE CAN DO BETTER!

Let's think about what "better" looks like.



Kubernetes + Wasm of the Future



Current



OS / Architecture specific OCI images



Bloated images



Slow start up



Increased memory consumption

Future



OS / Arch agnostic OCI images



Images only contain app code



Runtimes ready and running on node



Platform virtualization / less m



**LET'S GO FOR A SPIN IN THE
FUTURE!**



SPIN OCI RUNTIME DEMO



⋮

📄

🔍

🔗 1

🔧

📁 1

🖥️

⋮

👤

⚙️

EXPLORER

⌵ KCD-2023

> example1

⌵ example2

⌵ future

> .spin

> dist

> node_modules

⌵ src

TS index.ts

> target

💎 .gitignore

{} package-lock.json

{} package.json

📘 README.md

⚙️ spin.toml

TS tsconfig.json

📦 webpack.config.js

! deployment.yaml

> OUTLINE

> TIMELINE

> RUST DEPENDENCIES

Makefile example2 M

TS index.ts ×

! kind-config.yaml

Makefi

example2 > future > src > TS index.ts > ...

1 import { HandleRequest, HttpRequest, HttpResponse } from "@fermyon"

2

3 export const handleRequest: HandleRequest = async function (request)

4 return {

5 status: 200,

6 headers: { "content-type": "text/plain" },

7 body: "Hello from TS-SDK"

8 }

9 }

10

main*

0 0 0 3

0

Live Share

rust-analyzer

Spaces: 2

UTF-8

LF

{ } TypeScript

1 Sp



CONTAINERD/RUNWASI



```
pub trait Instance {  
    /// The WASI engine type  
    type Engine: Send + Sync + Clone;  
  
    /// Create a new instance  
    fn new(id: String, cfg: Option<&InstanceConfig<Self::E>>) -> Self;  
    /// Start the instance  
    /// The returned value should be a unique ID (such as a PID) for the instance.  
    /// Nothing internally should be using this ID, but it is returned to the containerd when  
    fn start(&self) -> Result<u32, Error>;  
    /// Send a signal to the instance  
    fn kill(&self, signal: u32) -> Result<(), Error>;  
    /// Delete any reference to the instance  
    /// This is called after the instance has exited.  
    fn delete(&self) -> Result<(), Error>;  
    /// Wait for the instance to exit  
    /// The waiter is used to send the exit code and time back to the caller.  
    /// Ideally this would just be a blocking call with a normal result,  
    /// because of how this is called from a thread it causes issues with  
    fn wait(&self, waiter: &Wait) -> Result<(), Error>;  
}
```

EXTENDING THE NODE RUNTIME FOR WASM



Related interesting work in the space

- [deislabs/krustlet](#)
- [deislabs/containerd-wasm-shims](#)
- [kuasar-io/kuasar](#)



PLATFORM ENGINEERING

Why have network boundaries when you can embed?



WebAssembly Standard Interface (WASI)



- Files / file systems
- Sockets
- HTTP
- Clocks
- Random numbers
- Other OS like behaviors



WASI Cloud Core



Key Value



Messaging



Runtime
Configuration



Distributed
Locking



SQL



Blob Storage




```
use types.{bucket, error, incoming-value, key, outgoing-value}

/// Get the value associated with the key in the bucket. It returns a incoming-value
/// that can be consumed to get the value.
///
/// If the key does not exist in the bucket, it returns an error.
get: func(bucket: bucket, key: key) -> result<incoming-value, error>

/// Set the value associated with the key in the bucket. If the key already
/// exists in the bucket, it overwrites the value.
///
/// If the key does not exist in the bucket, it creates a new key-value pair.
/// If any other error occurs, it returns an error.
set: func(bucket: bucket, key: key, outgoing-value: outgoing-value) -> result<_, error>

/// Delete the key-value pair associated with the key in the bucket.
///
/// If the key does not exist in the bucket, it returns an error.
delete: func(bucket: bucket, key: key) -> result<_, error>

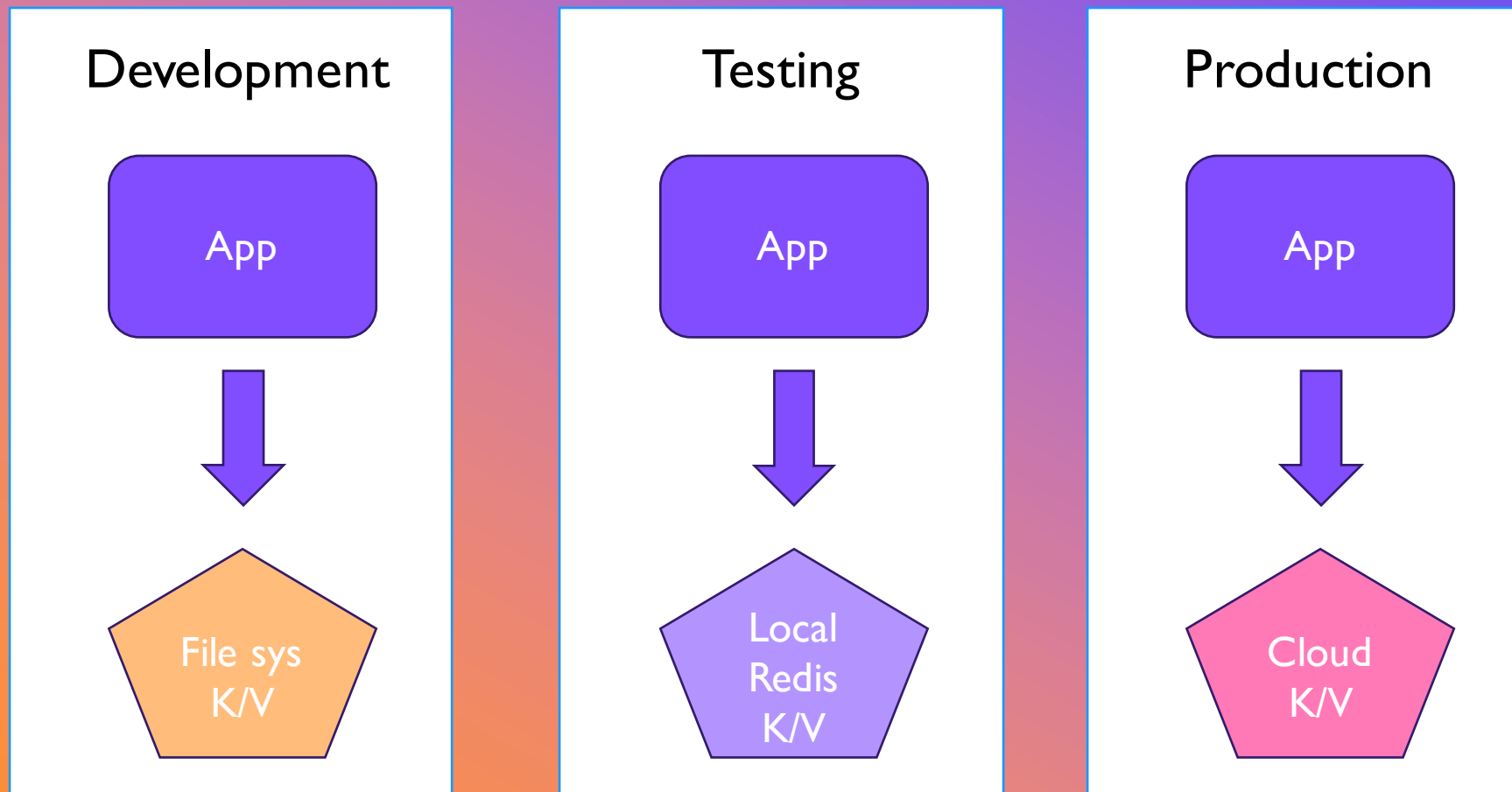
/// Check if the key exists in the bucket.
///
/// If the key does not exist in the bucket, it returns an error.
exists: func(bucket: bucket, key: key) -> result<bool, error>
```

CLOSER LOOK AT WASI- KEYVALUE



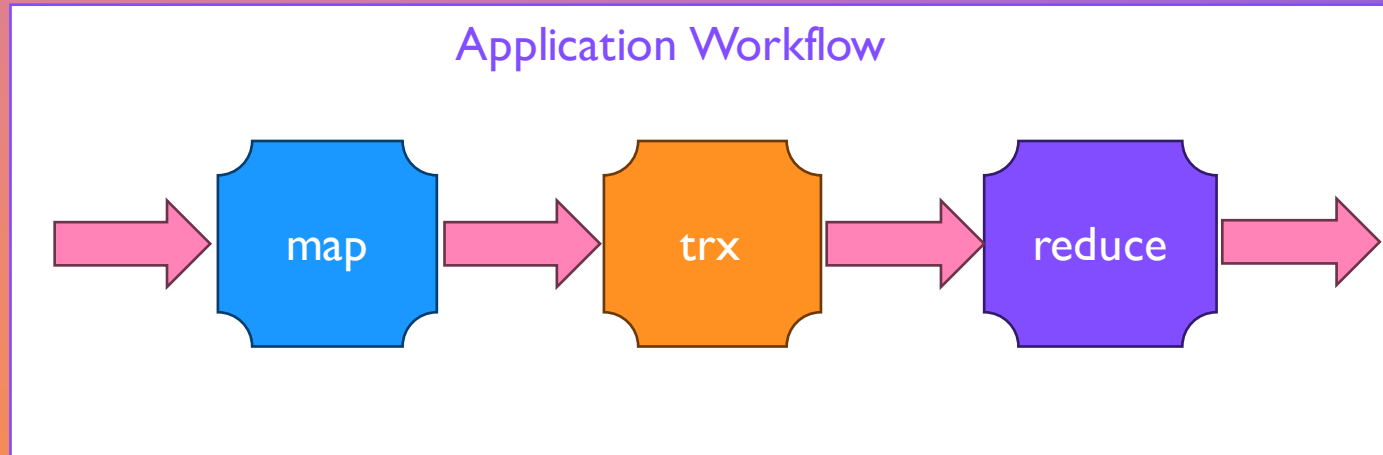
+
•
○

PLATFORM VIRTUALIZATION



+
•
○

DEPENDENCY INJECTION



BUILDING A STACK

Dev stack

anonymous
auth

disabled cache

....

Application

file system

Sqlite

....

Prod stack

oauth

enabled cache

....

Application

blob storage

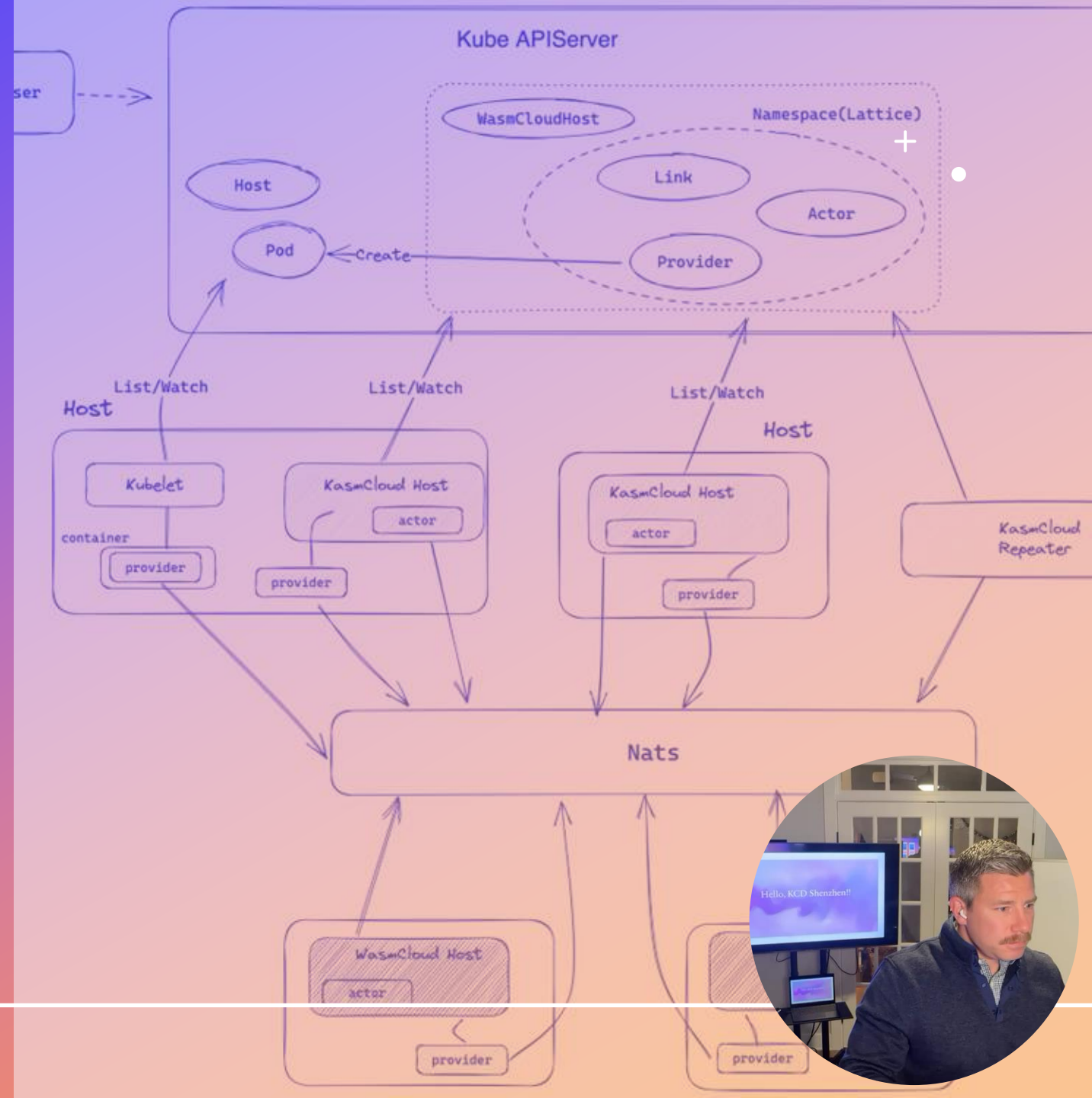
cloud storage

....



ABSTRACTIONS BEYOND PODS

credit to kasmCloud



HELP BUILD THE FUTURE

Let's build the future of Kubernetes together!



Some places to get involved

TAG-Runtime Wasm Working Group #wg-wasm

Bytecode Alliance

- (<https://bytecodealliance.zulipchat.com>)

bytecodealliance/wasmtime

containerd/runwasi

wasmCloud/wasmCloud

fermyon/spin

Kwasm/kwasm-operator

deislabs/containerd-wasm-shims



CNCF WASM LANDSCAPE

Runtimes



Application Frameworks



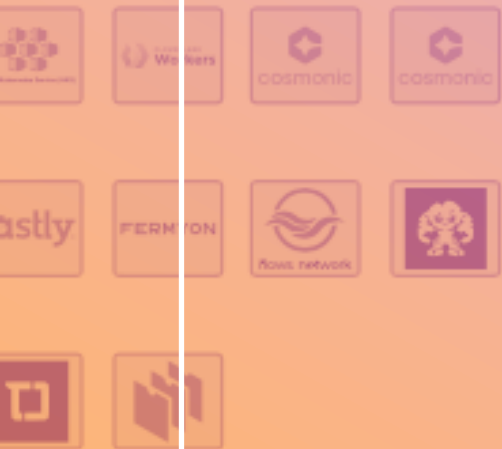
Edge/Bare metal



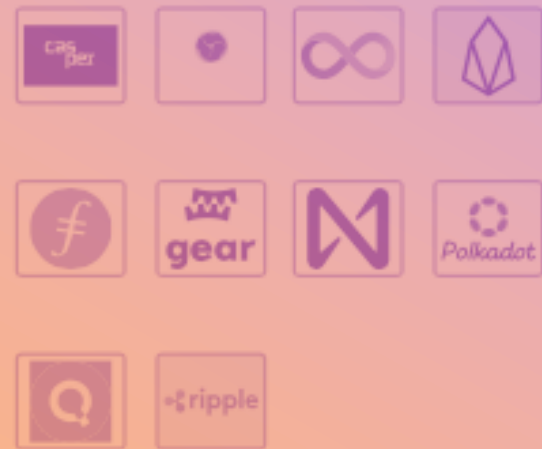
AI/Machine Learning



Hosted Platforms



Decentralized Platforms



Debugging & Observability



THANK YOU

I look forward to building the
future of Kubernetes with you!

