



Capabilities all the way down

A resilient OS for products where safety, security, and privacy matter

Sid Hussmann, CTO and Co-Founder

What should a keynote look like?

1. Feel smug about my past research
2. Sell my latest startup idea
3. Vent my anger and frustration

I'm going for option 3.



ETH zürich Userlix OSDI / ATC 2021

1. Feel smug about my past research

2. Sell my latest startup idea

3. Vent my anger and frustration

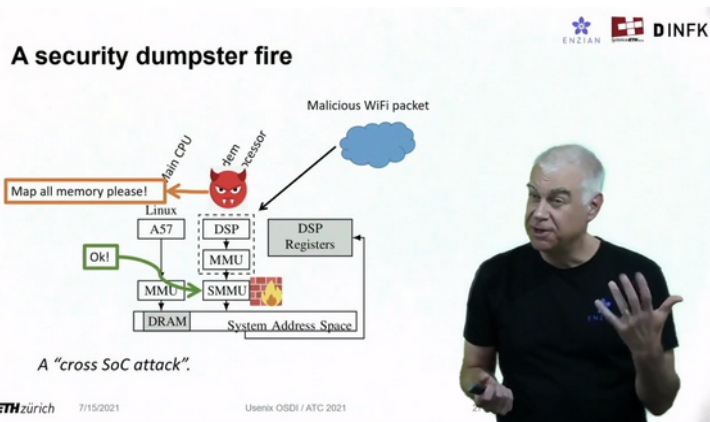
I'm going for option 3.

But what is it?

1. The set of
 - Kernel
 - Daemons
 - Runtime libraries

2. *That body of software* that:

- Multiplexes machine's hardware resources
- Abstracts the hardware platform
- Protects software principals from each other
 - Using the hardware



A security dumpster fire

Diagram illustrating a malicious WiFi packet attack on a mobile device. The packet, labeled "Malicious WiFi packet", is received by the "Main CPU" (labeled "Main CPU" with a red devil icon). The packet contains the instruction "Map all memory please!". This instruction is processed by the "Linux" OS, which then sends a signal to the "A57" processor. The "A57" processor sends a signal to the "MMU", which then sends a signal to the "SMMU". The "SMMU" sends a signal to the "DRAM" (labeled "System Address Space"). The "DSP" (labeled "DSP Registers") is also shown, with a bidirectional arrow between it and the "MMU". A red devil icon is also present near the "DSP".

A "cross SoC attack".

This is such an opportunity!

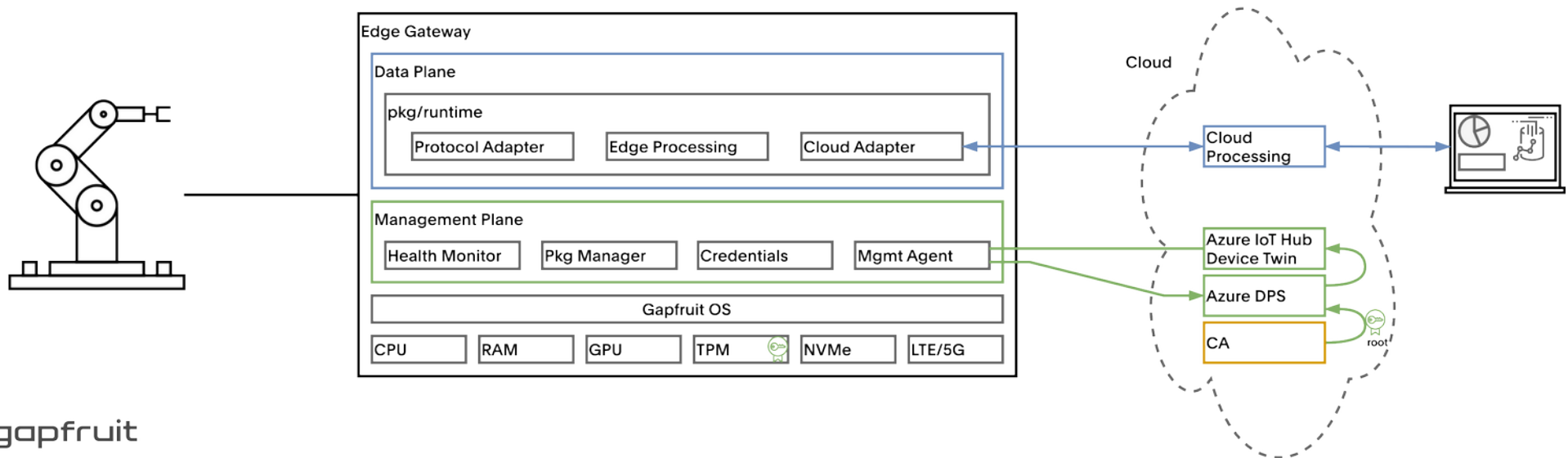
- There is so much **cool, fun, relevant** research that could be done today
 - We can really rethink OS structures for new h/w
- We've got **great tools** that didn't exist before
 - Great new formal techniques to apply
 - New languages
 - Amazing new hardware
- Why waste this on 50-year-old thinking?

About Gapfruit

- 2012: Team starts building real-world products with microkernel and capability-based security for the governmental sector
- 2017: Rollout military-grade notebook (HW/SW co-design)
- 2018: Founding of Gapfruit AG in Switzerland
- 2020: TEE for transactional workloads in finance industry (first use of WASM/WASI in a product)
- 2022: Partnership with Bechtle (and others) for the IIoT sector

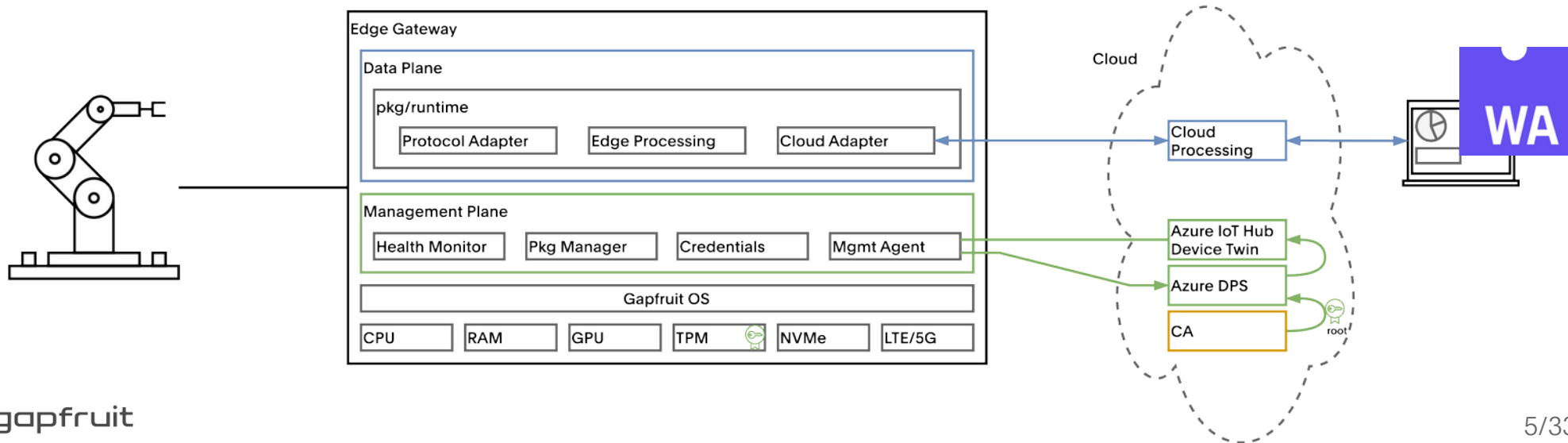
Use-Case Industrial IoT Gateway

- Connect machines/robots securely to the internet
- Pre-process data on the edge
- Manage fleets of devices over a long period of time
- Guarantee availability, integrity, and confidentiality



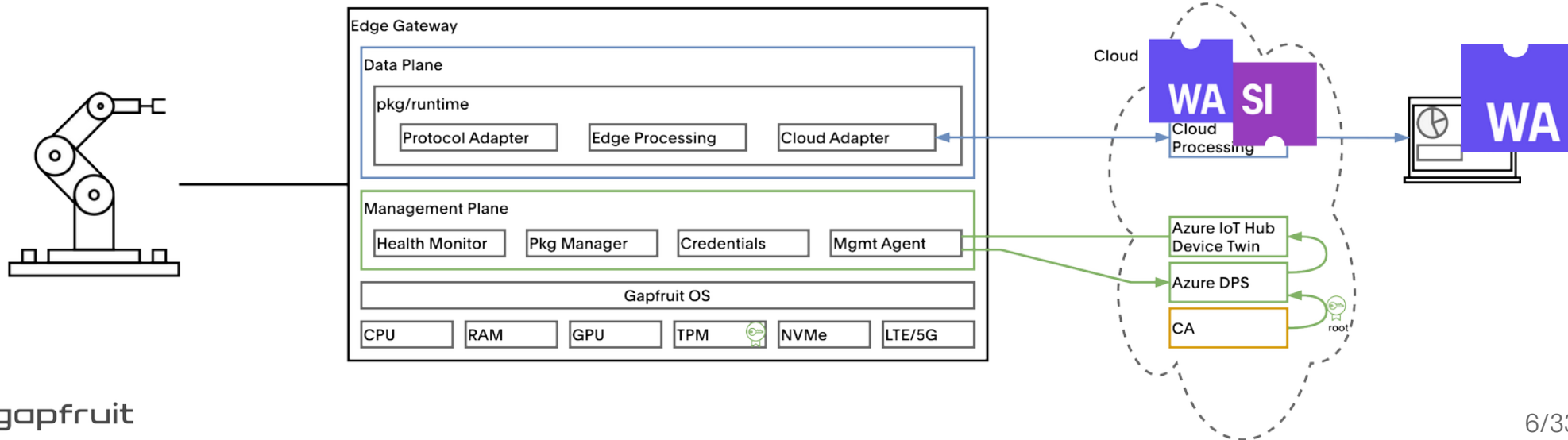
WebAssembly: Why?

- WASM in the browser for performance and language support



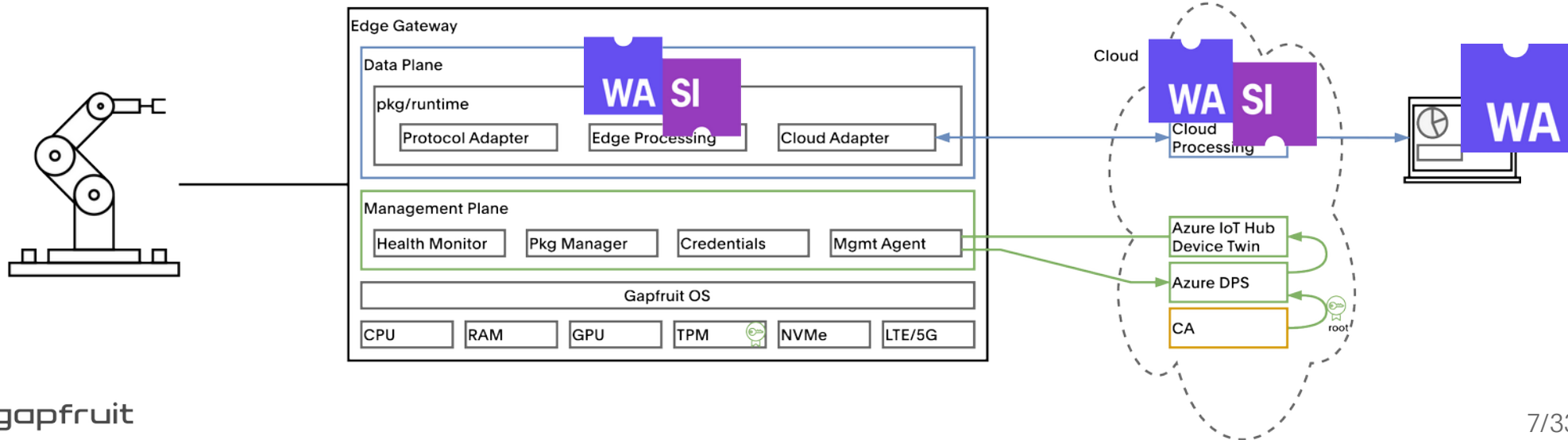
WebAssembly: Why?

- WASM in the browser for performance and language support
- WASM/WASI for portable and lightweight cloud workloads

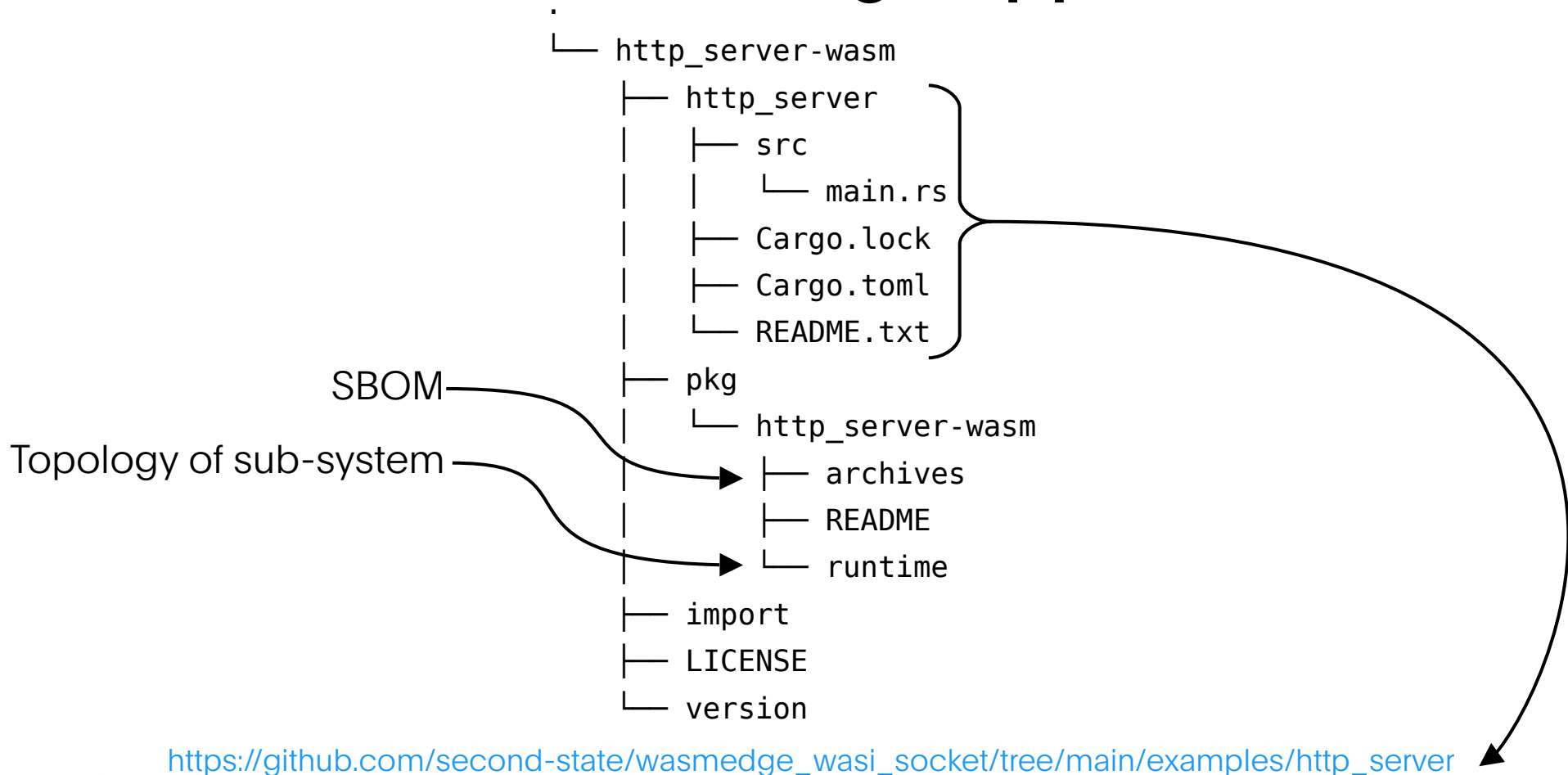


WebAssembly: Why?

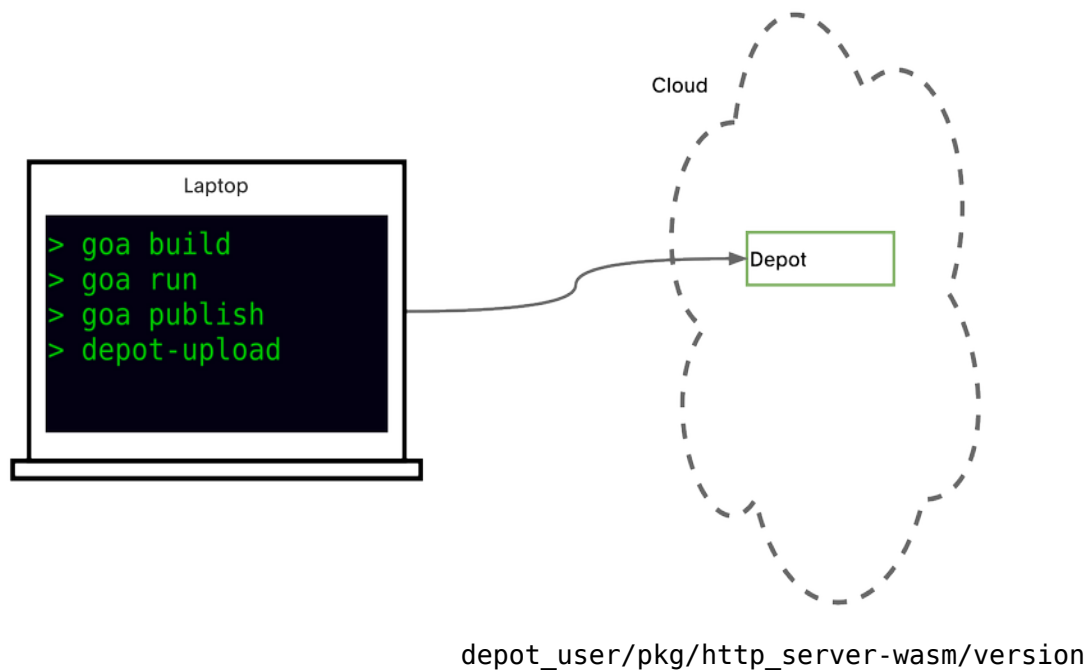
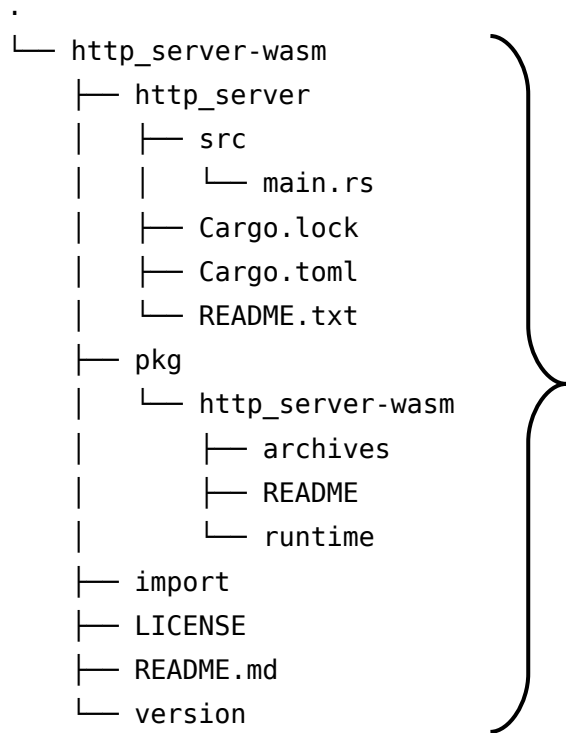
- WASM in the browser for performance and language support
- WASM/WASI for portable and lightweight cloud workloads
- Deploy the same apps to the edge

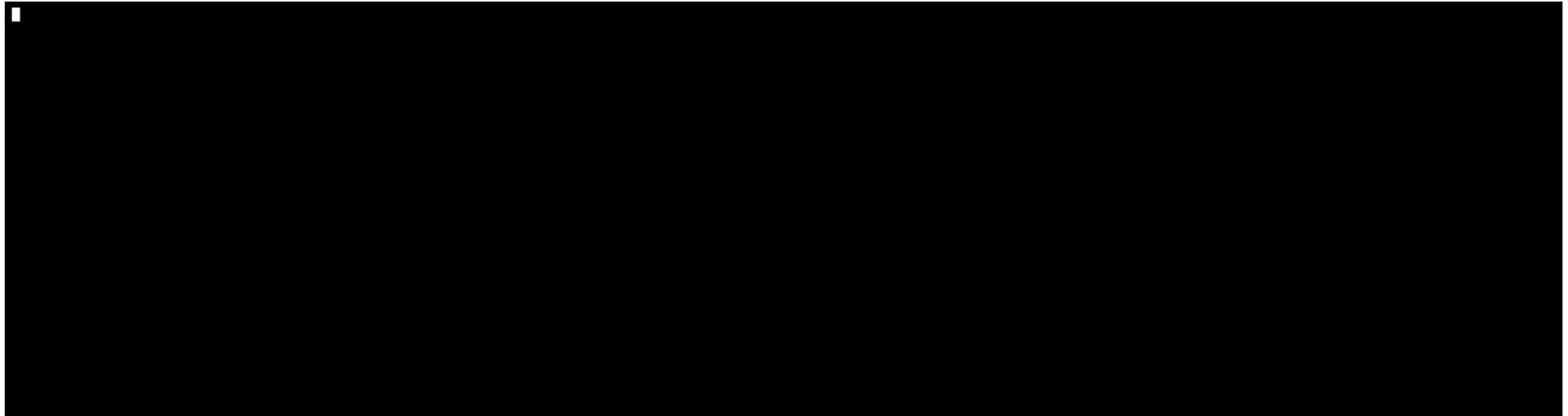
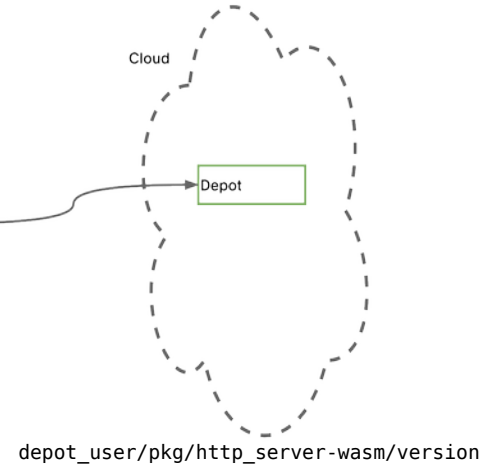
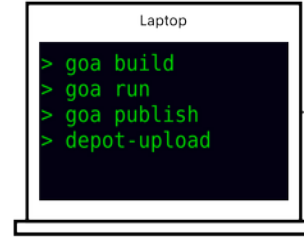
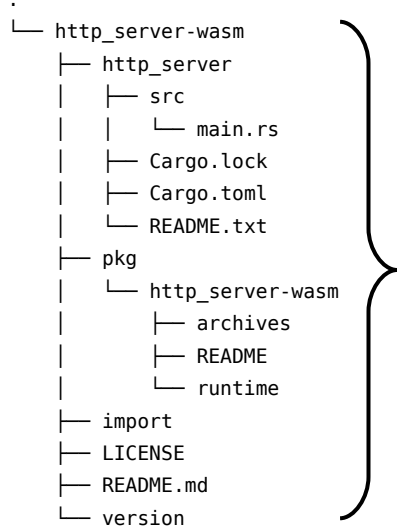


Demo: Build WasmEdge App with Goa



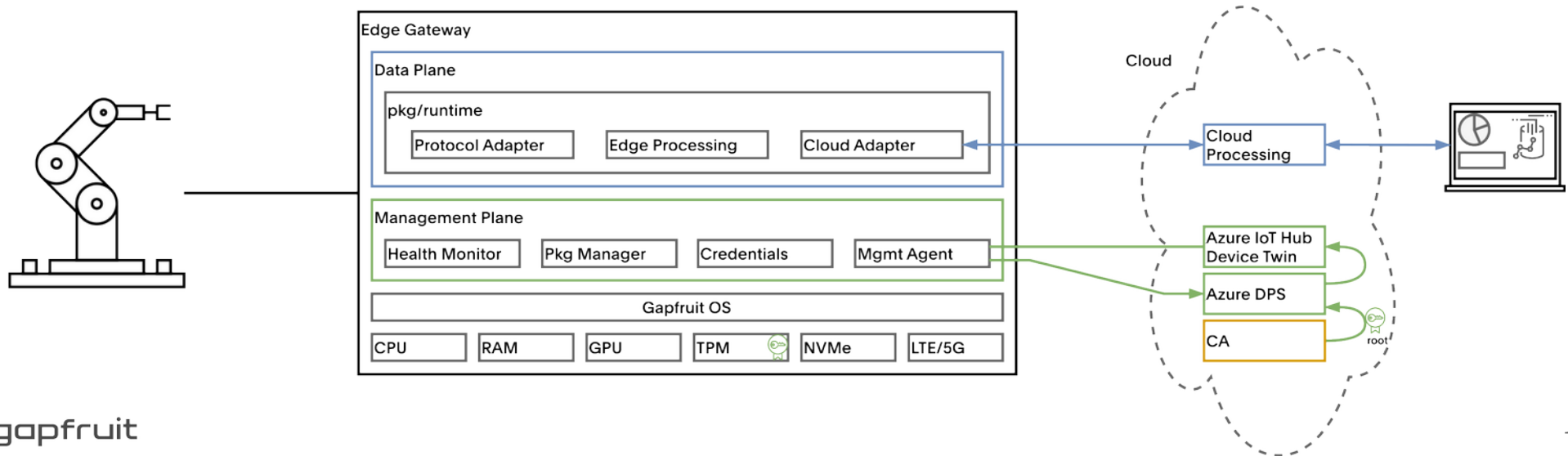
Publish WasmEdge App with Goa





Gapfruit OS for IIoT

- >99% reduction of attack surface
- Mass provisioning of IIoT infrastructure



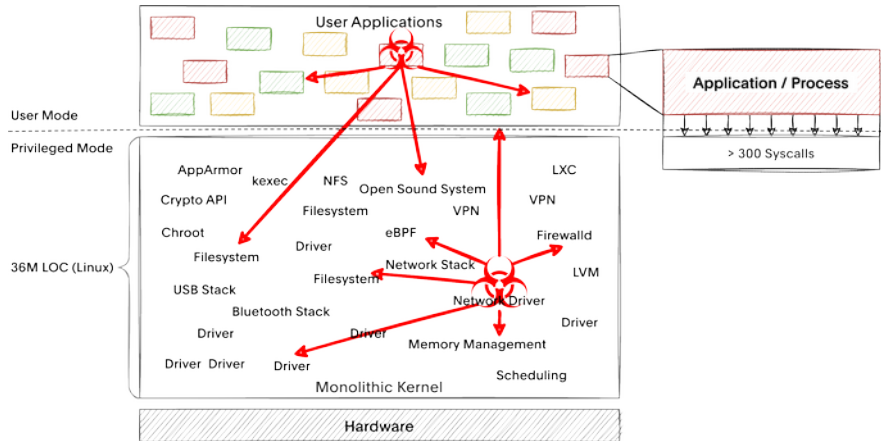
Gapfruit OS

- Microkernel operating system with capability-based security
- Built with the Genode Framework [1]
 - Supporting multiple kernels (seL4, base-hw, nova, Linux)
 - Linux Device Driver Environment
 - Multiple runtimes/languages: Posix, libc, JVM, Rust, WASI, VM, Python, Go, etc.
 - Desired state system composition
- Industrial-grade declarative configuration management
 - TR-369 in collaboration with Axiros
 - Microsoft Azure DPS/IoT in collaboration with Device Insight

[1] <https://github.com/genodelabs/genode>

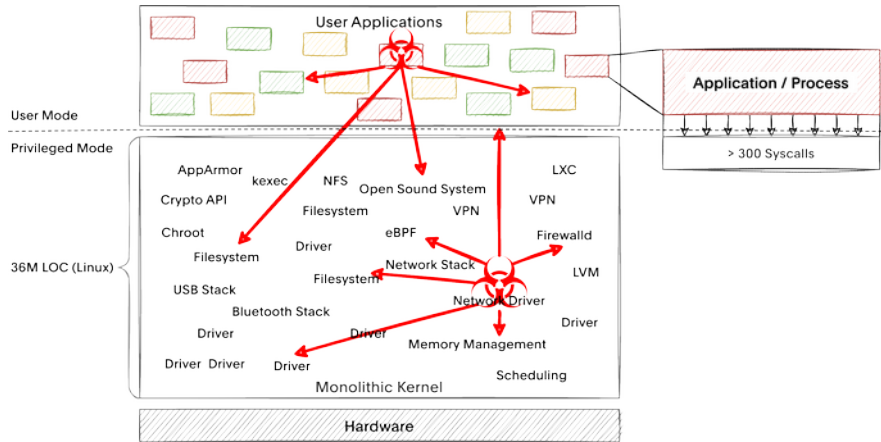
>99% Reduction of Attack Surface

Linux

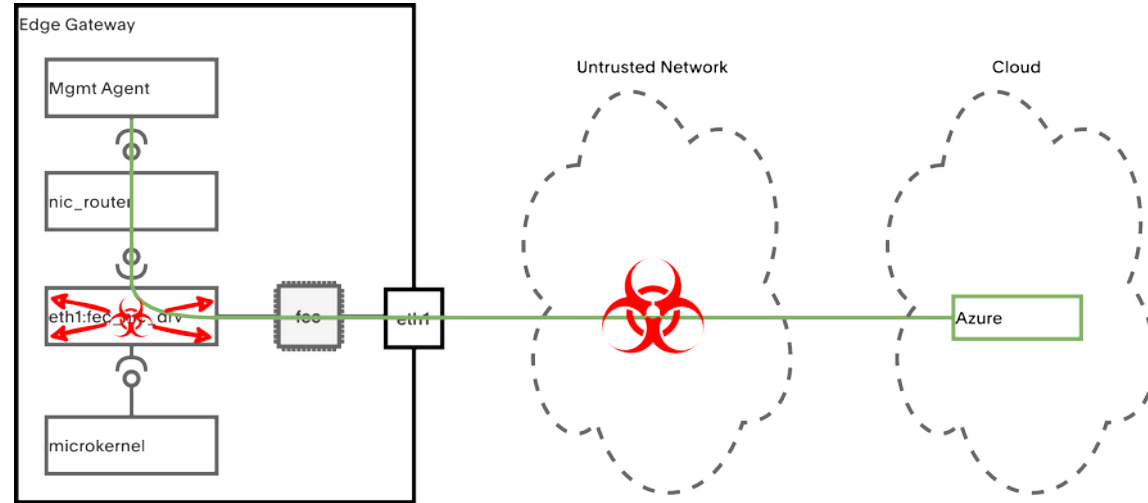


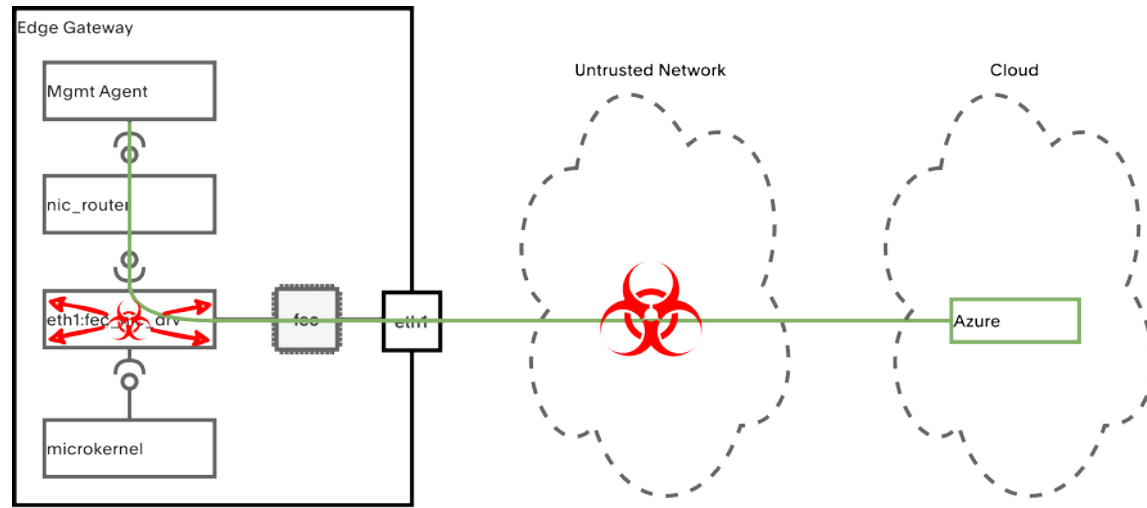
>99% Reduction of Attack Surface

Linux



Gapfruit OS

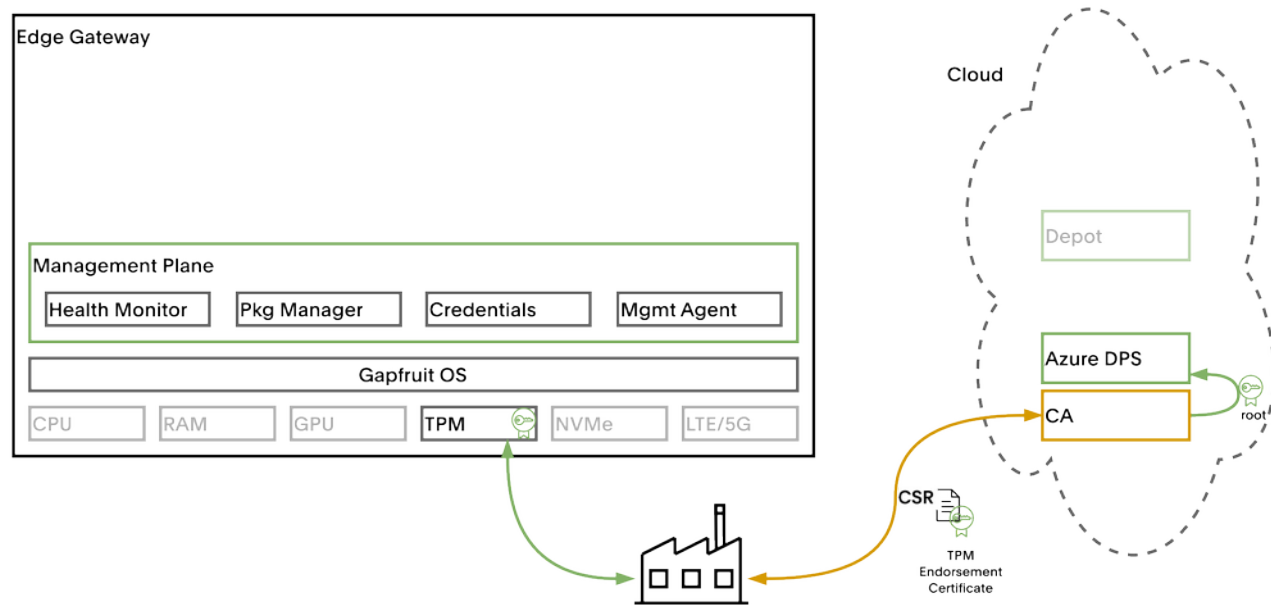




TPM's: Why?

- Trusted Boot
- Attestation
- Hybrid Secure Counters for Dynamic Disk Integrity
- Protect Secrets with Policy
- Strong Digital Identity for Authentication

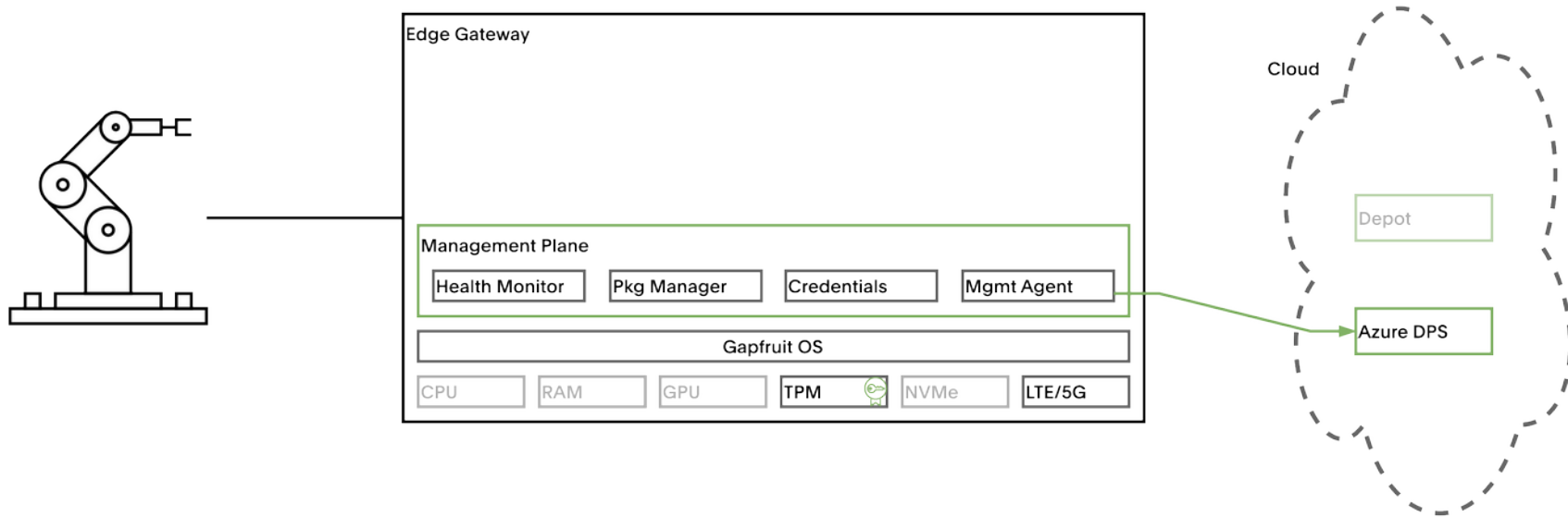
Utilizing TPMs for Mass-Provisioning



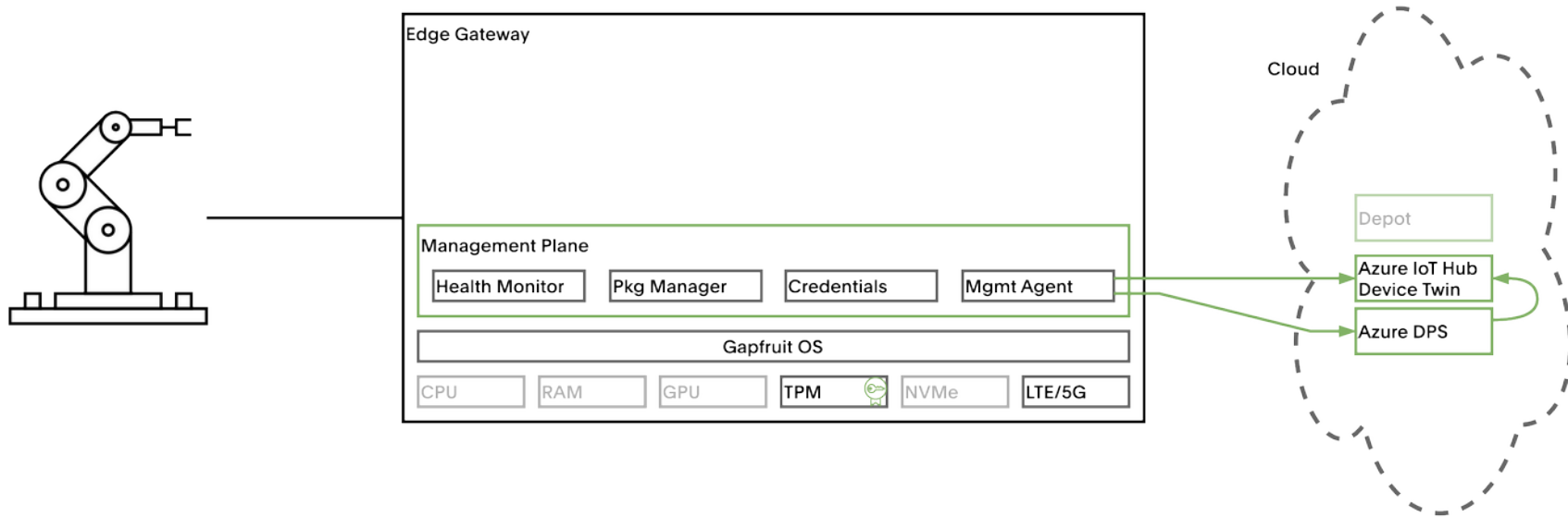
Demo: Zero-Touch Provisioning

- Azure Device Provisioning Service (DPS)
- Declarative Desired State Management
- Authentication with TPM-backed credentials
- Integration with PKI
- Deployment of WasmEdge app to the edge

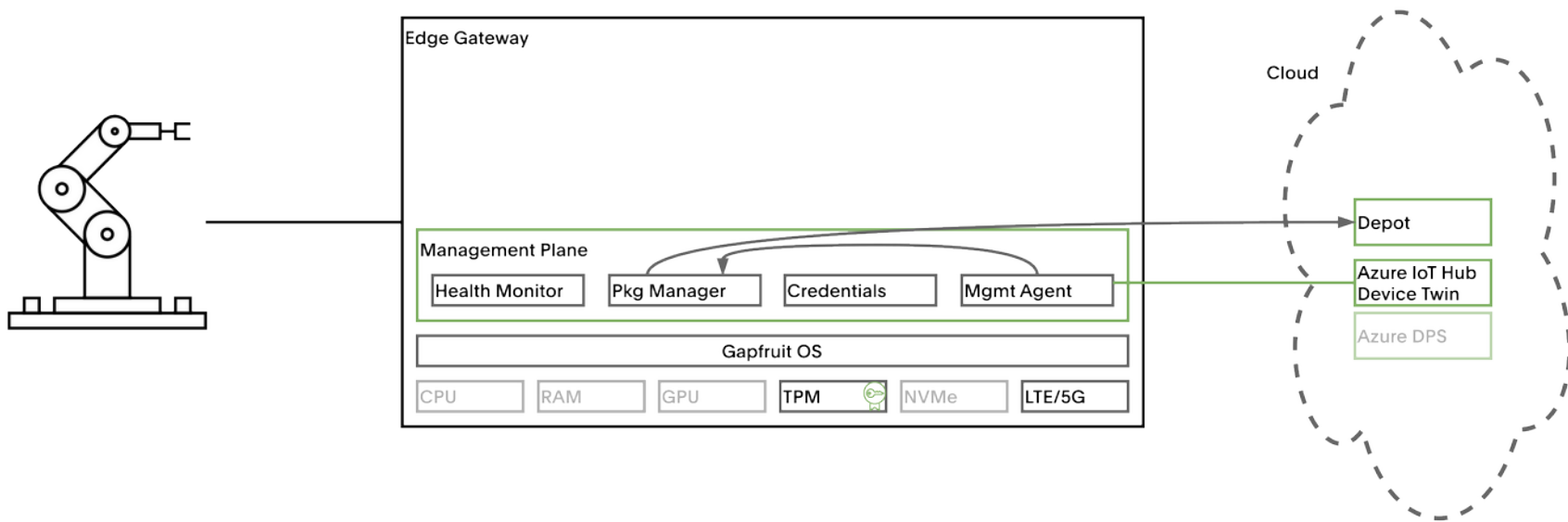
Step 1: Azure Device Provisioning Service



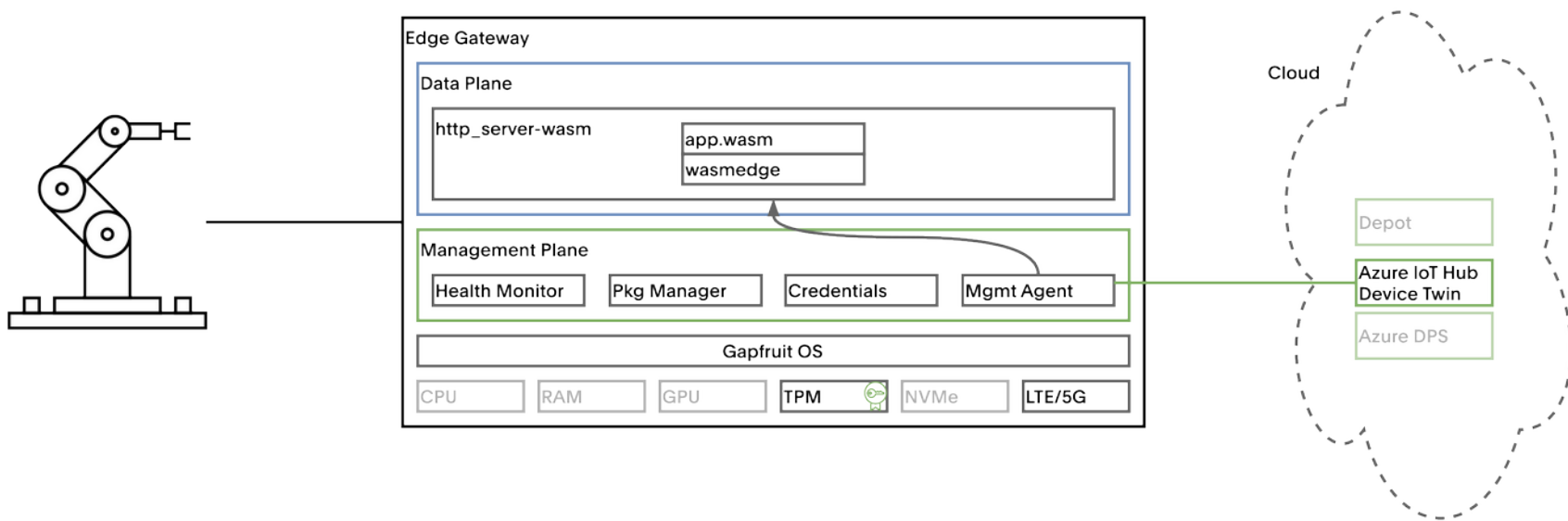
Step 2: Azure IoT Hub Device Twin



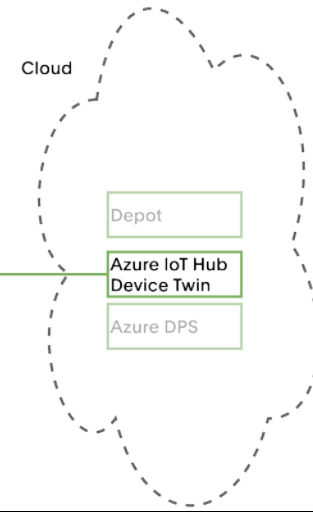
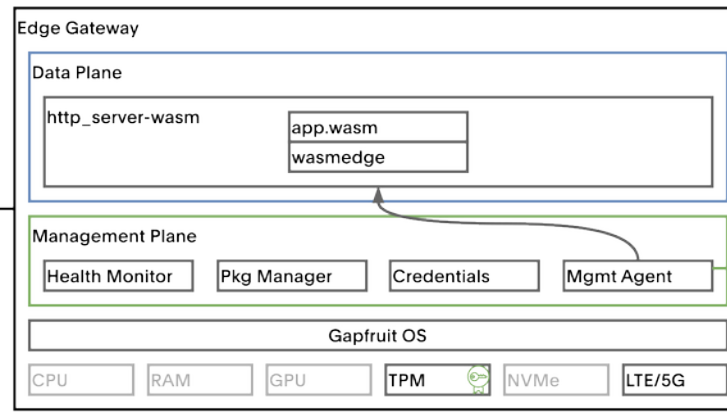
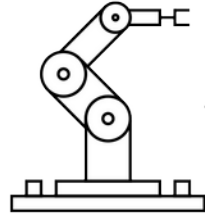
Step 3: Fetch WasmEdge App



Step 4: Deploy WasmEdge App



1. DPS: Desired state
2. DPS: Create Twin
3. Install Pkg
4. Start App/Server
5. Connection:
Robot → Server

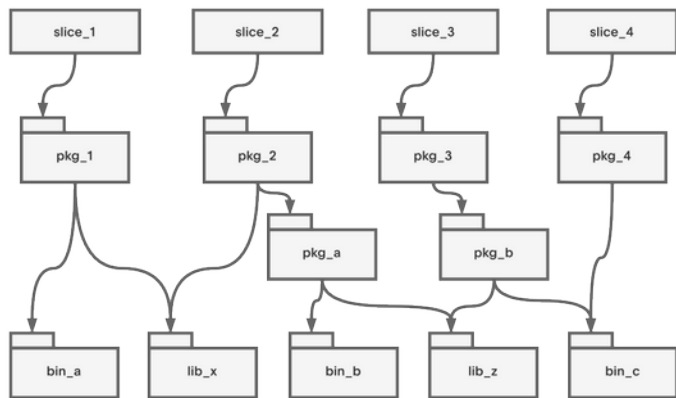


Core Concepts of Gapfruit OS

- Everything runs in user-space
- Strong isolation guarantees from microkernel
- Control over dependencies from capability model

Control over Dependencies

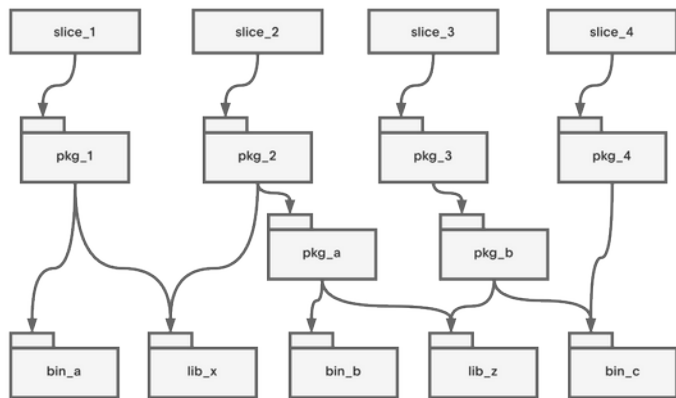
Software Dependencies



Lightweight Package Management

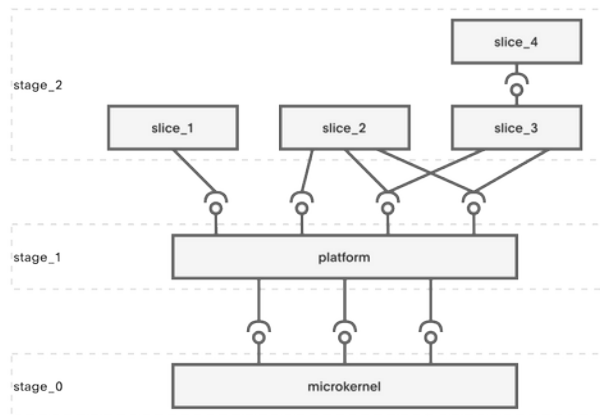
Control over Dependencies

Software Dependencies



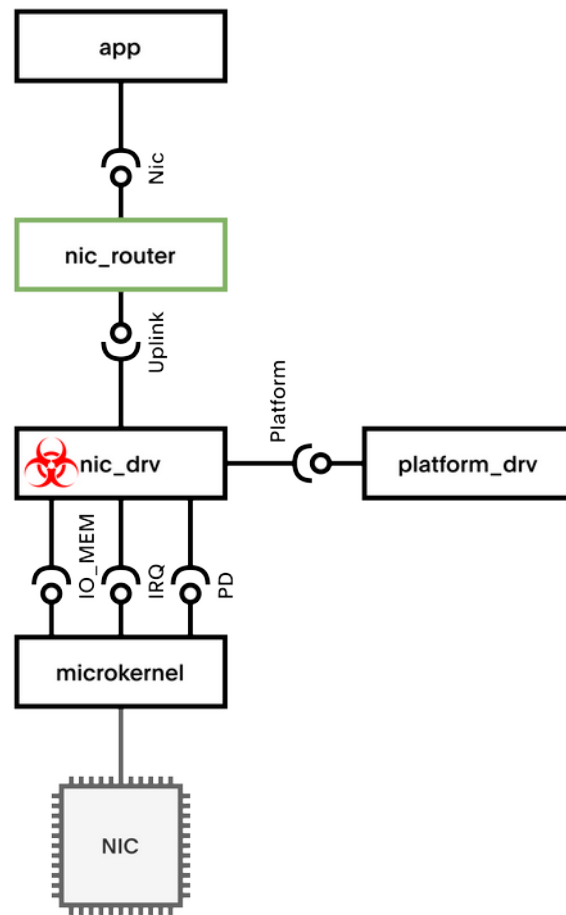
Lightweight Package Management

Service Topology



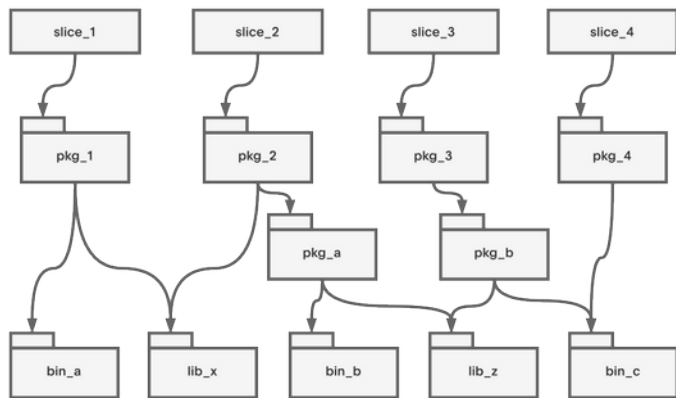
Service Oriented Architecture

Service Topology Example



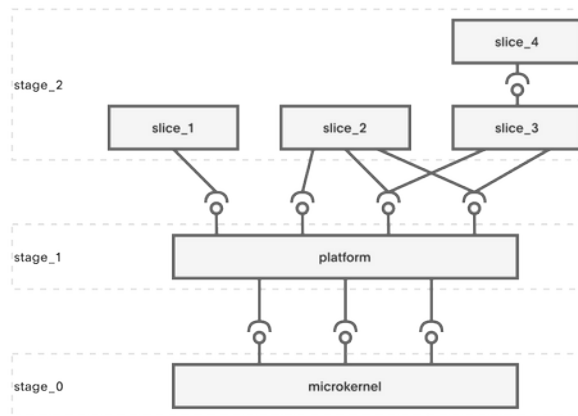
Control over Dependencies

Software Dependencies



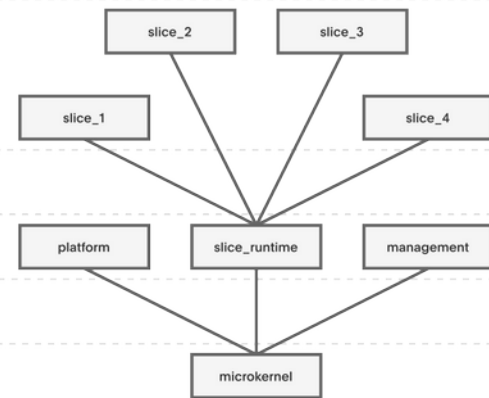
Lightweight Package Management

Service Topology



Service Oriented Architecture

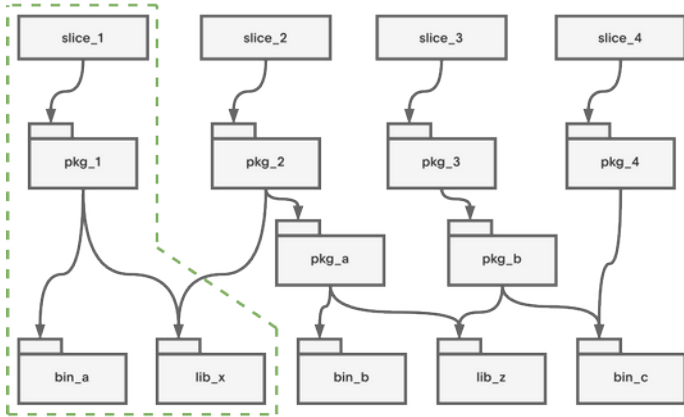
Resource Distribution



Parent-Child Relationship

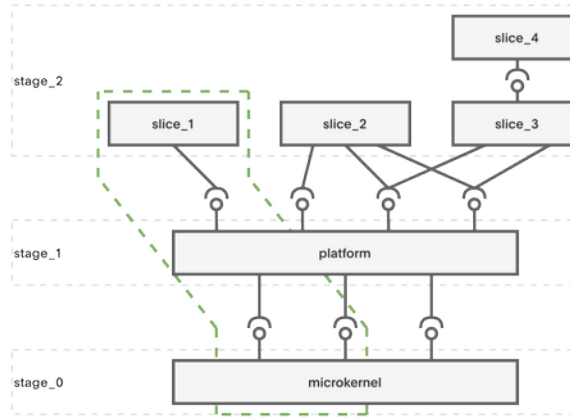
Control over the Trusted Computing Base

Software Dependencies



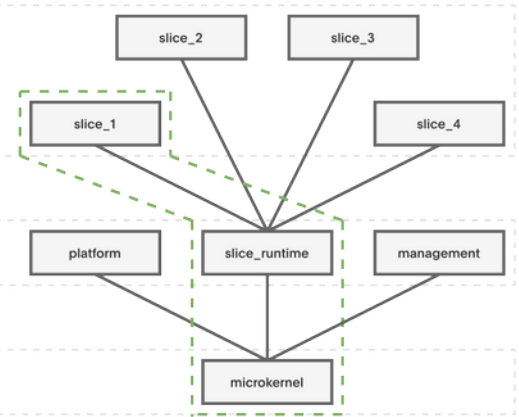
Lightweight Package Management

Service Topology



Service Oriented Architecture

Resource Distribution



Parent-Child Relationship

No Safety without Security

Safety:

- System must not harm the world
- Certification
- Low complexity = better safety, certification cost

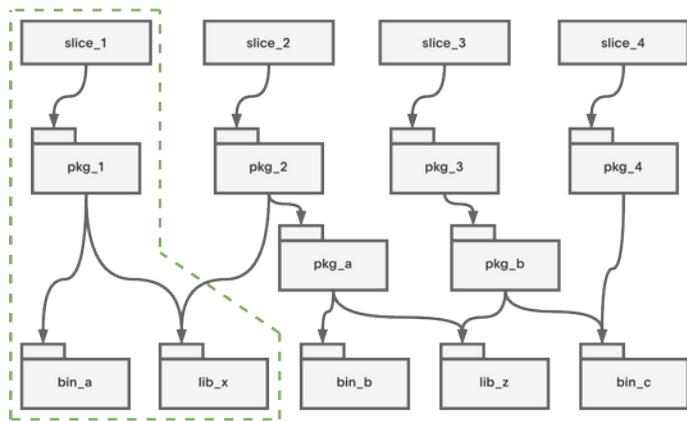
Security:

- The world must not harm the system
- Respond to vulnerabilities
- Low complexity = better security

Solution: Strong separation of concerns

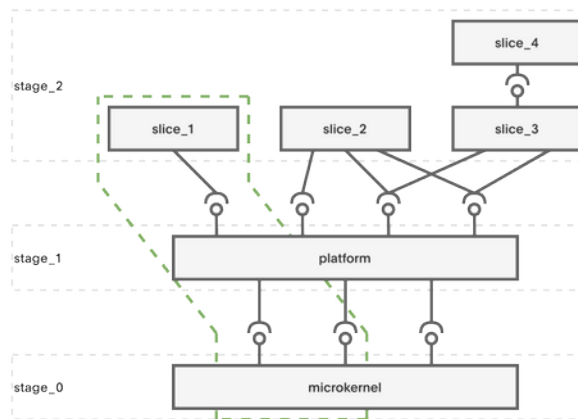
Capabilities all the way down

Software Dependencies



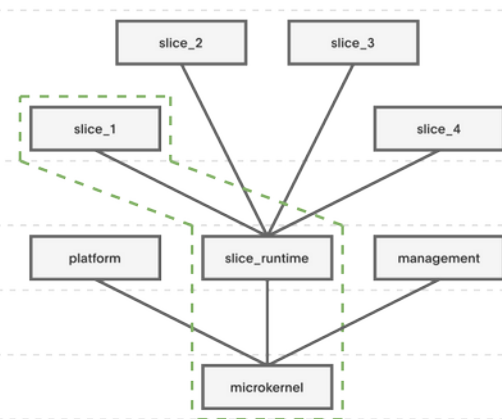
Lightweight Package Management

Service Topology



Service Oriented Architecture

Resource Distribution



Parent-Child Relationship

Use-cases for capabilities all the way down

- Products on the verge between OT and IT
 - Industrial automation
 - Critical infrastructure
- Products on the verge between safety and security
 - Medtec
 - Automotive
 - Aerospace
- Any product where safety, security, and privacy matter

Questions

Sid Hussmann
CTO & Founder

<https://gapfruit.com>

Twitter: @sidhussmann

LinkedIn: <https://linkedin.com/in/sidhussmann>

Mastodon: @sidhussmann@infosec.exchange