SeL4 Summit 2023:
Microkernel OS, TPMs, and WASM in IIoT Environments

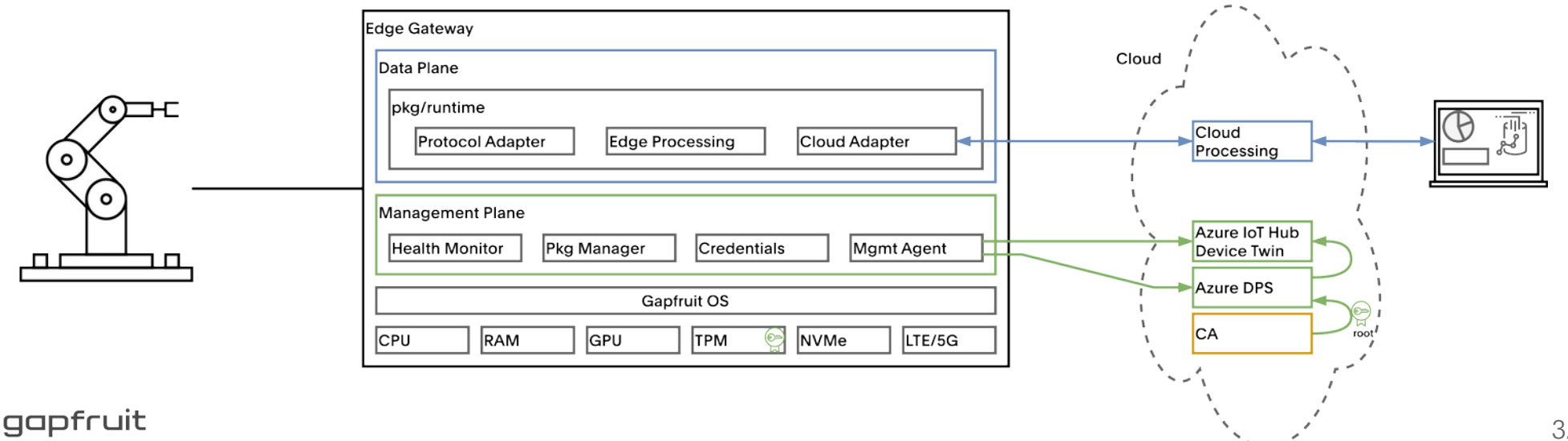
Sid Hussmann, CTO and Co-Founder, Gapfruit

About Gapfruit

- 2012: Team starts developing with microkernel and capability-based security for the governmental sector
  - Genode contributor since then
- 2017: Rollout secure notebook (HW/SW co-design)
- 2018: Founding of Gapfruit in Switzerland
- 2019: Pivot & Partnership with Toradex
- 2020: HSM vendors run Gapfruit OS for the banking sector
  - First use of WASM/WASI for attested transactional TEE
- 2022: Partnership with Bechtle (and others) for the IIoT sector

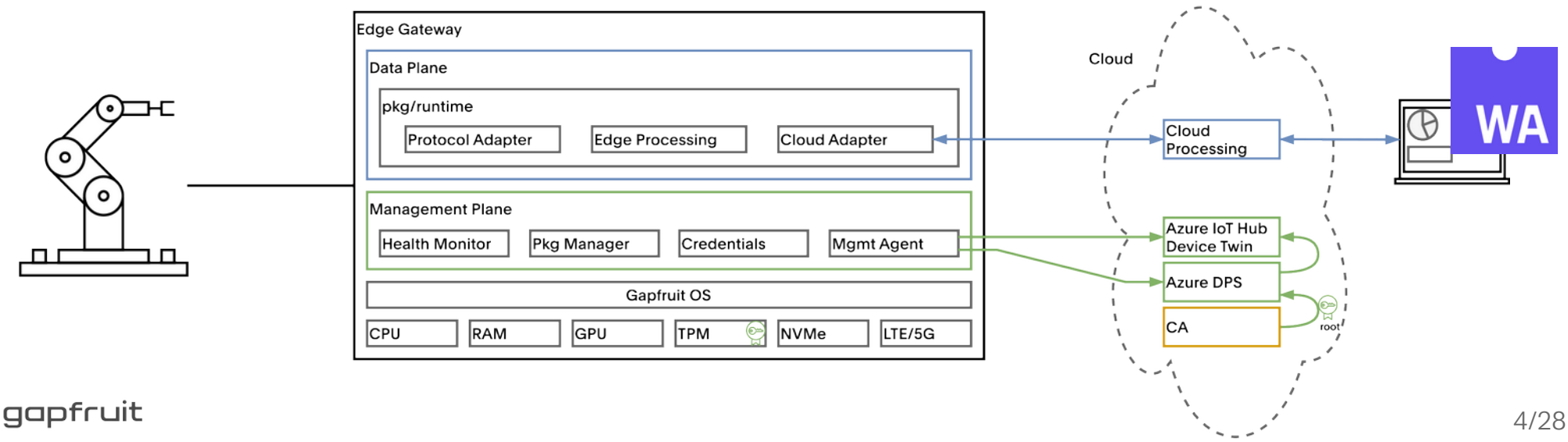
Use-Case Industrial IoT Gateway

- 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



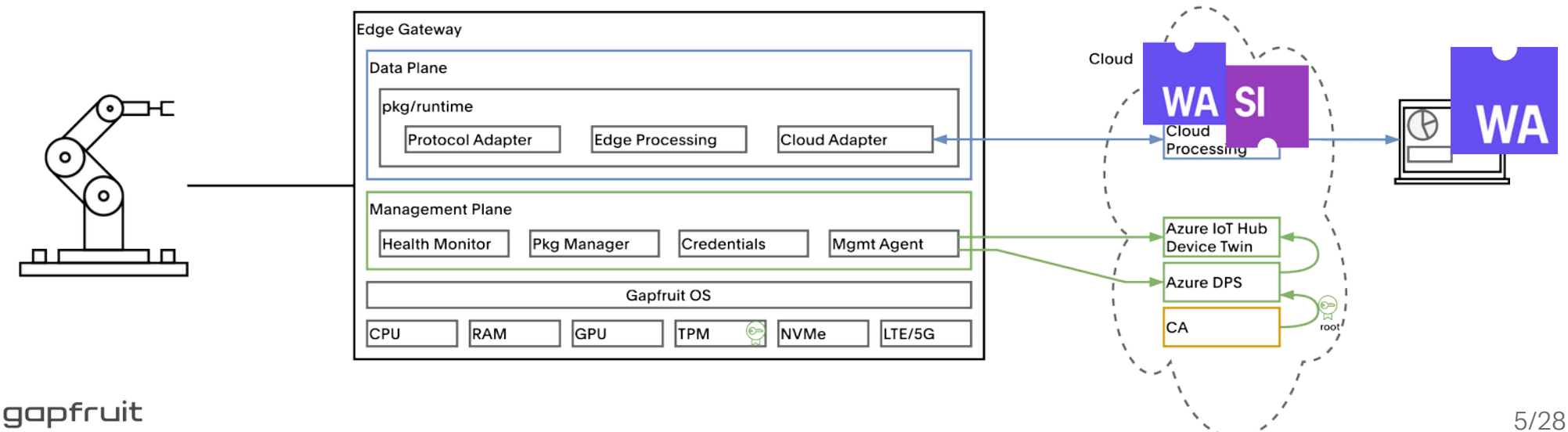
WebAssembly: Why?

- WASM in the browser for performance and language support



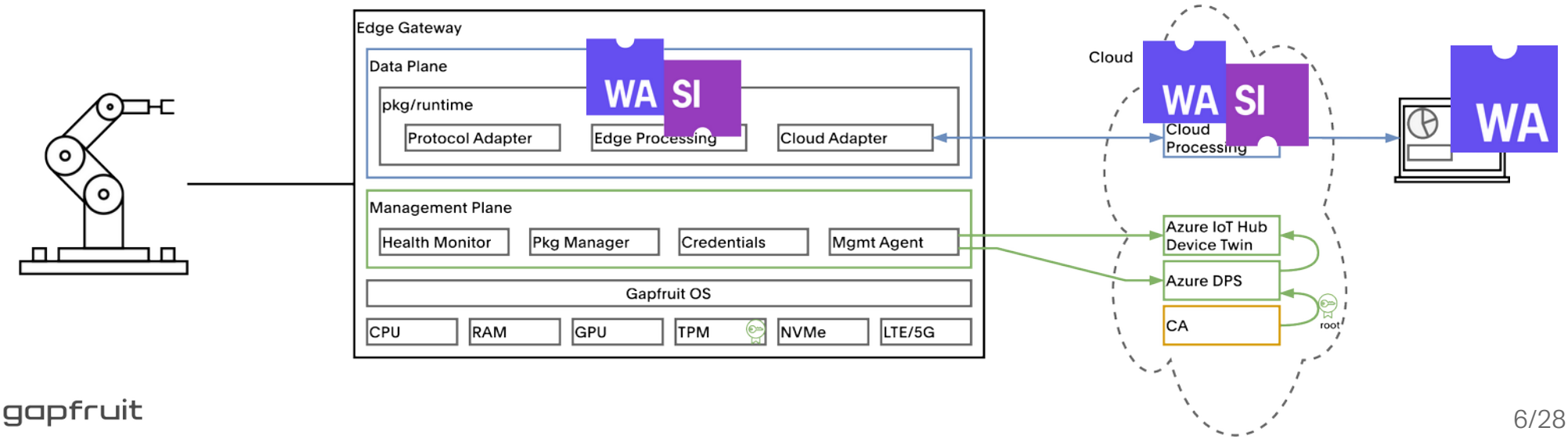
WebAssembly: Why?

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

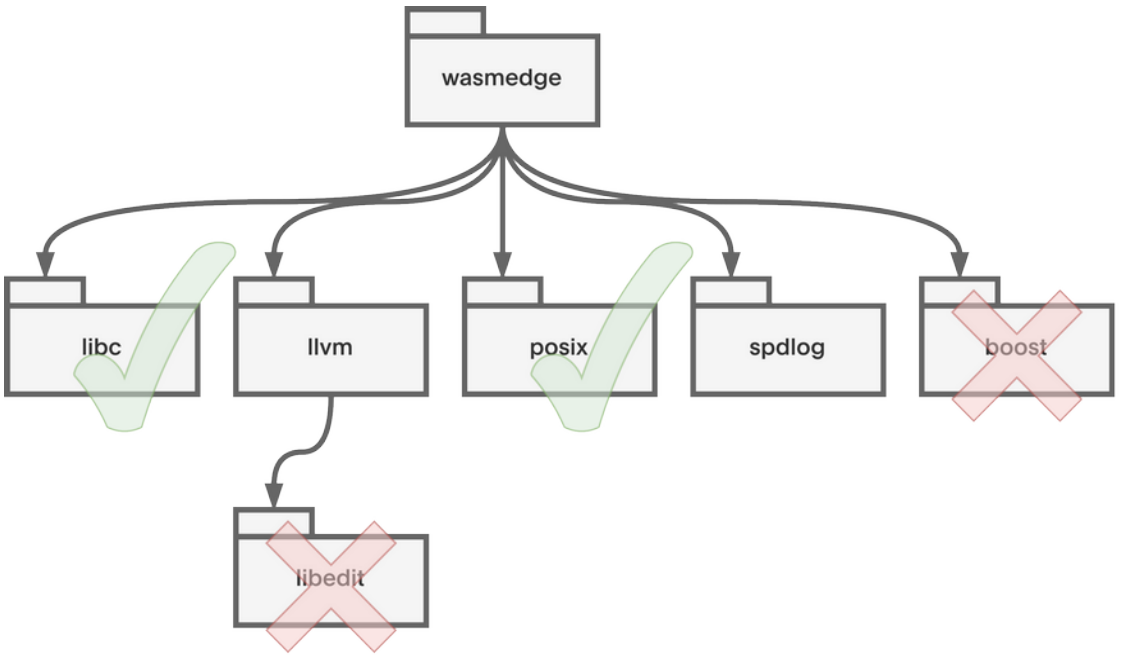


WebAssembly: Why?

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

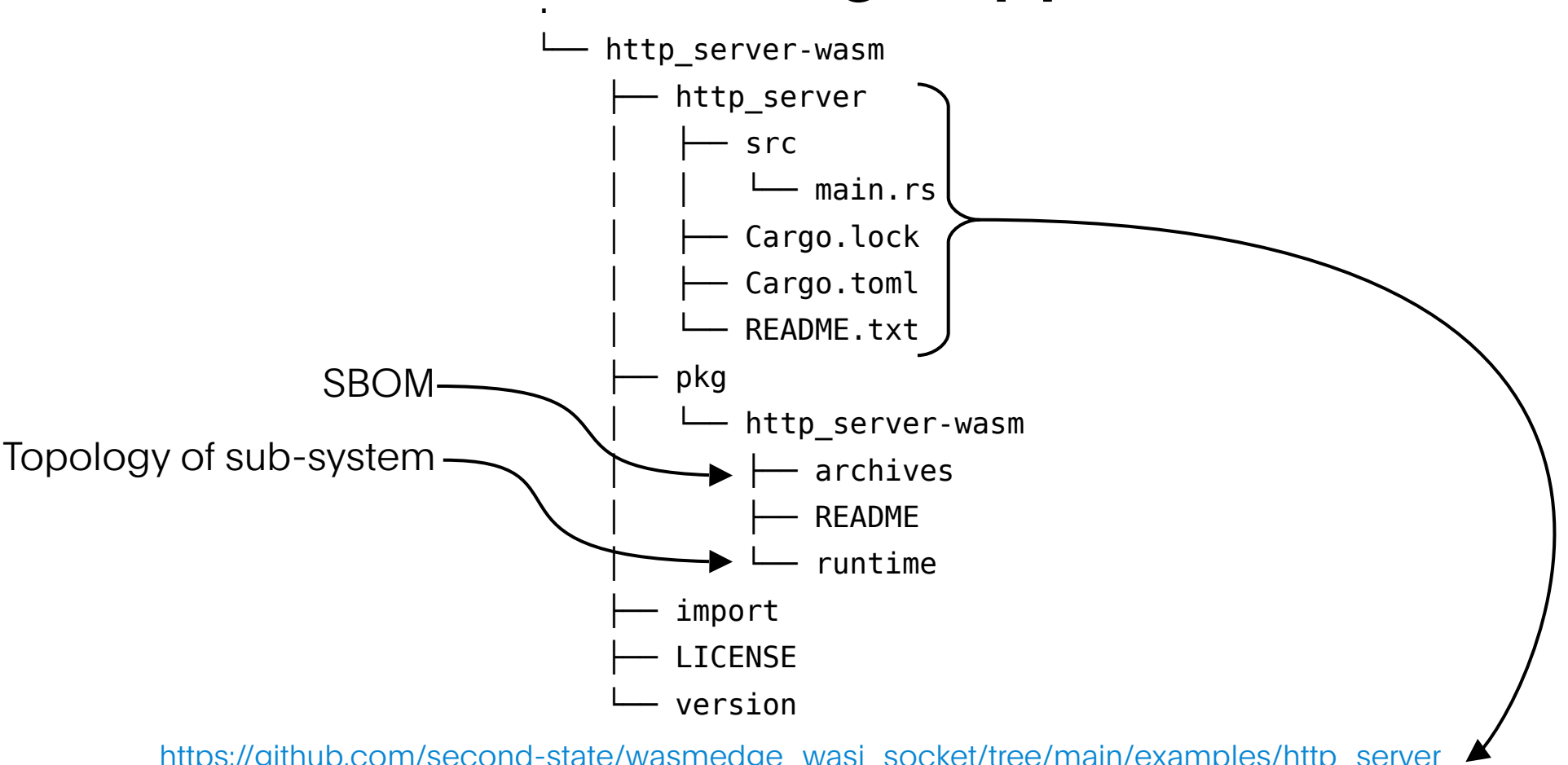


WebAssembly: How?



https://github.com/WasmEdge/WasmEdge

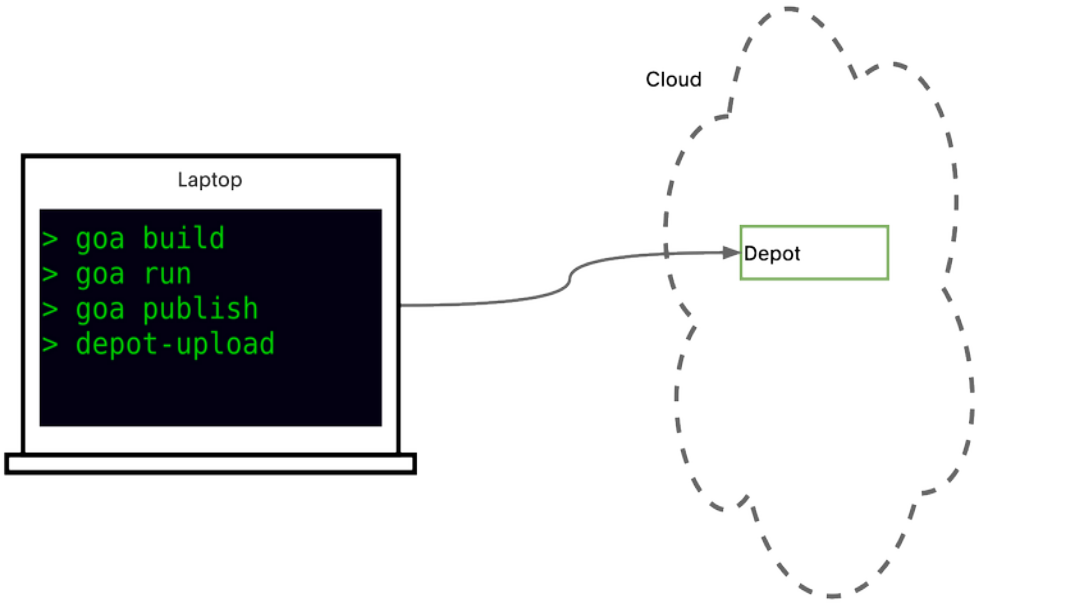
Demo: Build WasmEdge App with Goa



https://github.com/second-state/wasmedge_wasi_socket/tree/main/examples/http_server

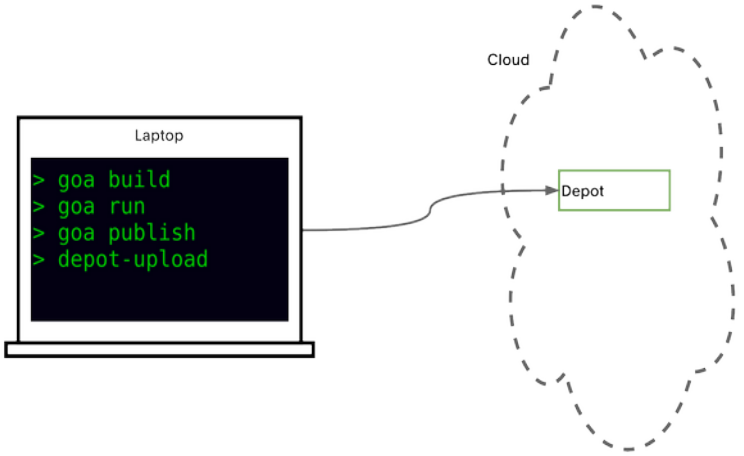
Publish WasmEdge App with Goa

```
.
└── http_server-wasm
    ├── http_server
    │   ├── src
    │   │   └── main.rs
    │   ├── Cargo.lock
    │   ├── Cargo.toml
    │   └── README.txt
    ├── pkg
    │   └── http_server-wasm
    │       ├── archives
    │       ├── README
    │       └── runtime
    ├── import
    ├── LICENSE
    ├── README.md
    └── version
```



depot_user/pkg/http_server-wasm/version

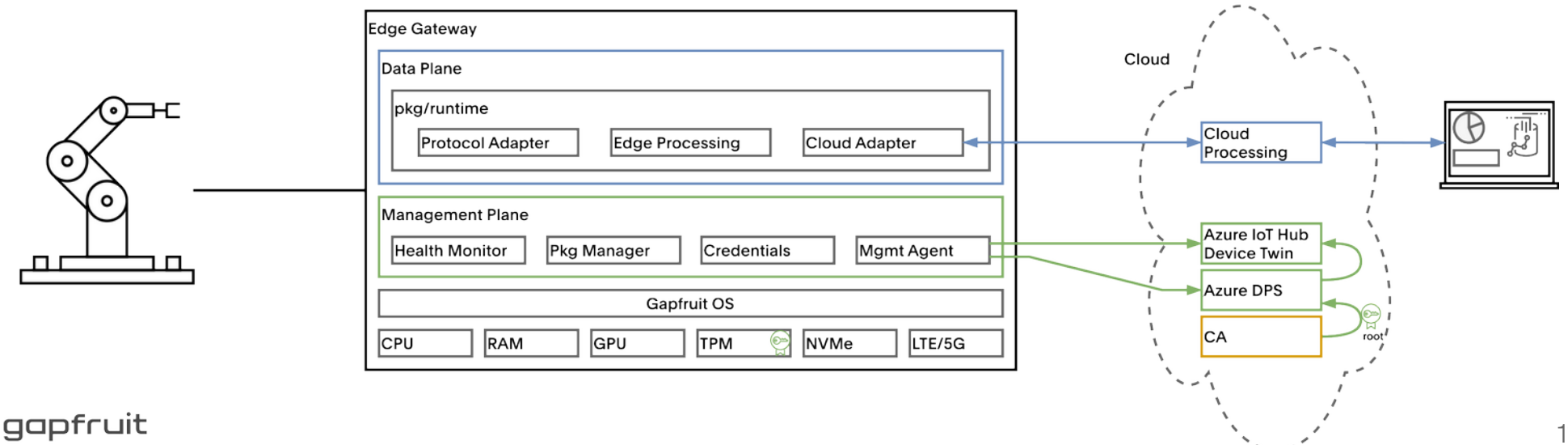
```
└── http_server-wasm
    ├── http_server
    │   ├── src
    │   │   └── main.rs
    │   ├── Cargo.lock
    │   ├── Cargo.toml
    │   └── README.txt
    ├── pkg
    │   └── http_server-wasm
    │       ├── archives
    │       ├── README
    │       └── runtime
    ├── import
    ├── LICENSE
    ├── README.md
    └── version
```



depot_user/pkg/http_server-wasm/version

Gapfruit OS for IIoT

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

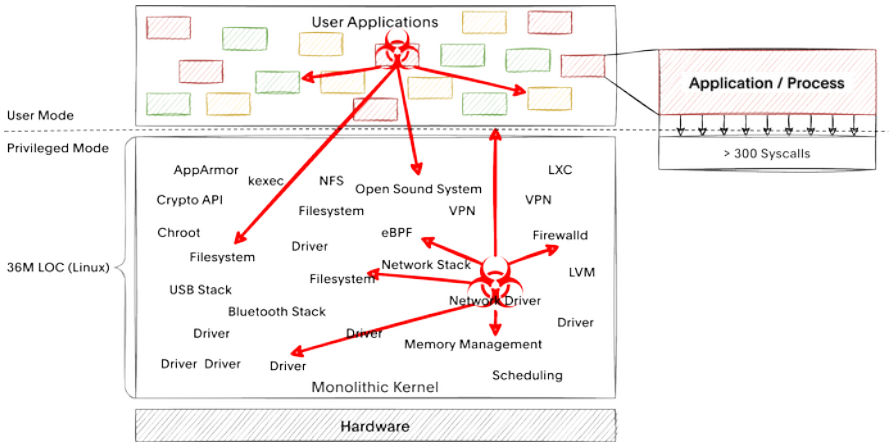


Gapfruit OS

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

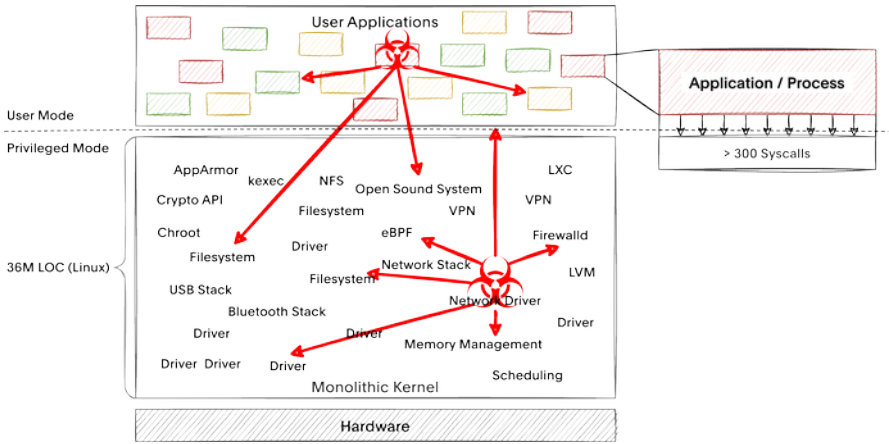
>99% Reduction of Attack Surface

Linux

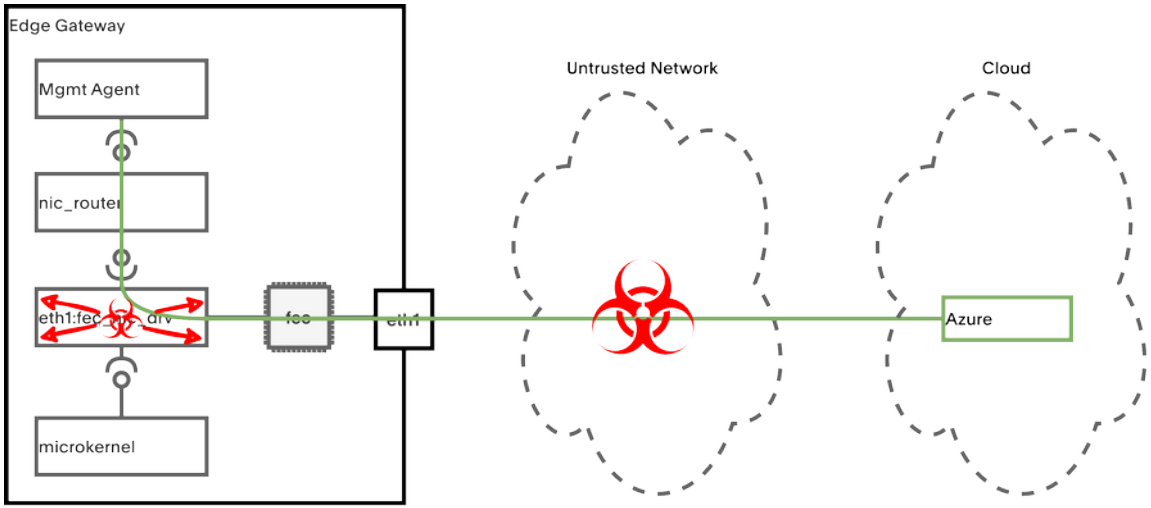


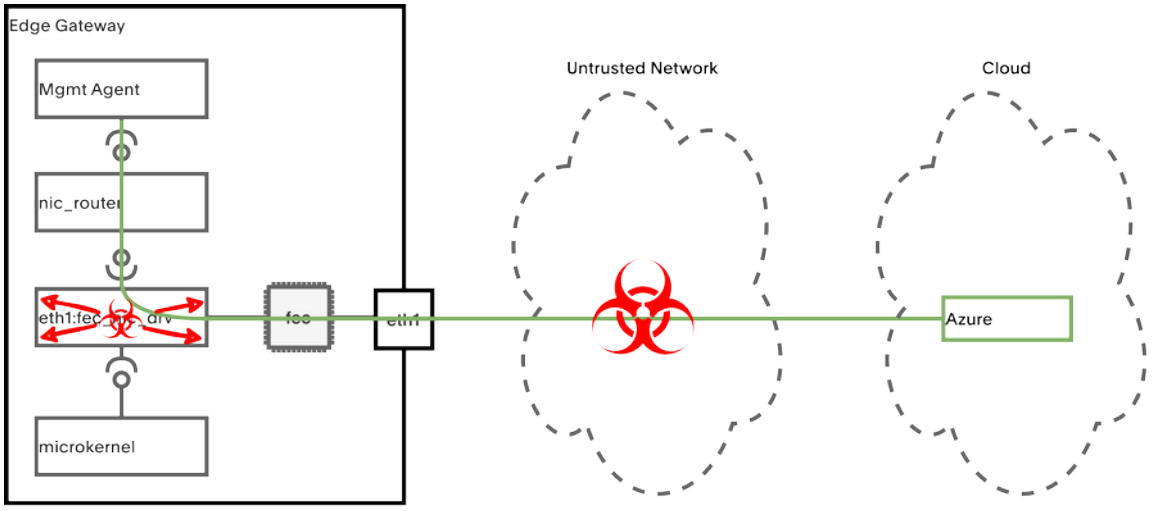
>99% Reduction of Attack Surface

Linux



Gapfruit OS

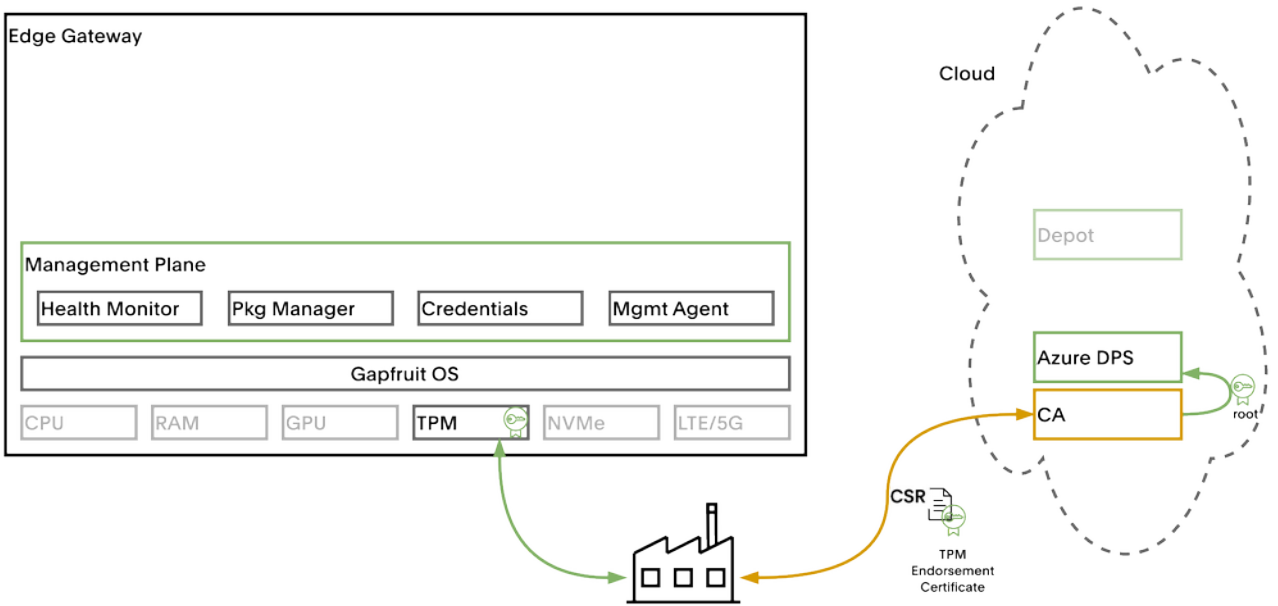



Edge Gateway
Mgmt Agent
nic_router
eth1:fec_drv
microkernel
fec
eth1
Untrusted Network
Cloud
Azure

TPM's: Why?

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

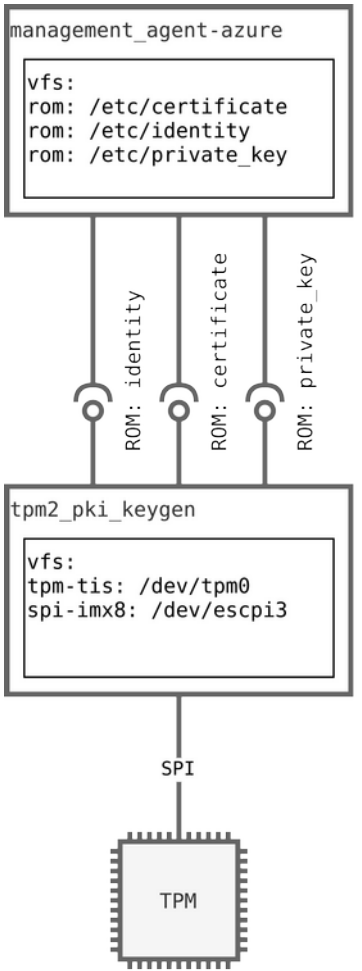
Utilizing TPMs for Mass-Provisioning



TPM Support for Gapfruit OS

- Port **`tpm2-tss`**
- Implement drivers as VFS plugins:
  - CRB driver for fTPM (x86_64)
  - SPI driver for dTPM (i.MX8)
- **`tpm-tis`** VFS plugin that adapts TPM commands to SPI
- Update **`openssl3`** with TPM2 provider
- Create short-lived key/certificate from TPM-backed credentials

TPM's: How?



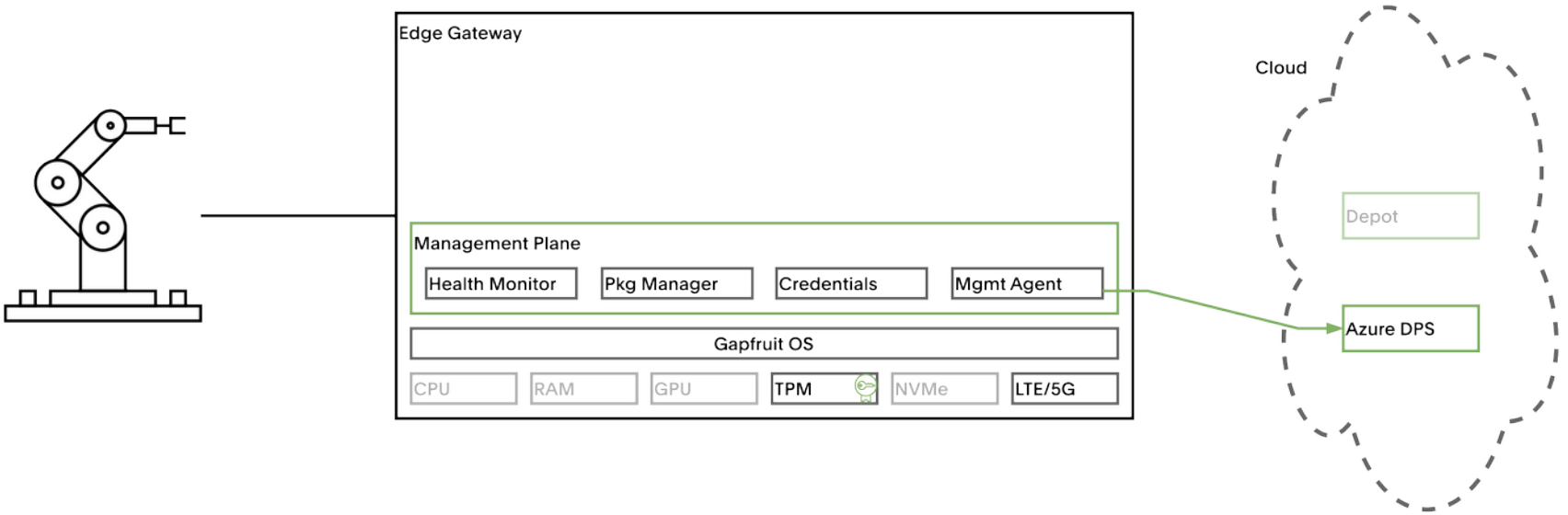
simplified sub-system:
- generate shortlived credentials
- provide shortlived credentials

uses:
libcrypto3
tpm2-tss
tpm2-openssl
vfs_spi
vfs_tpm-tis

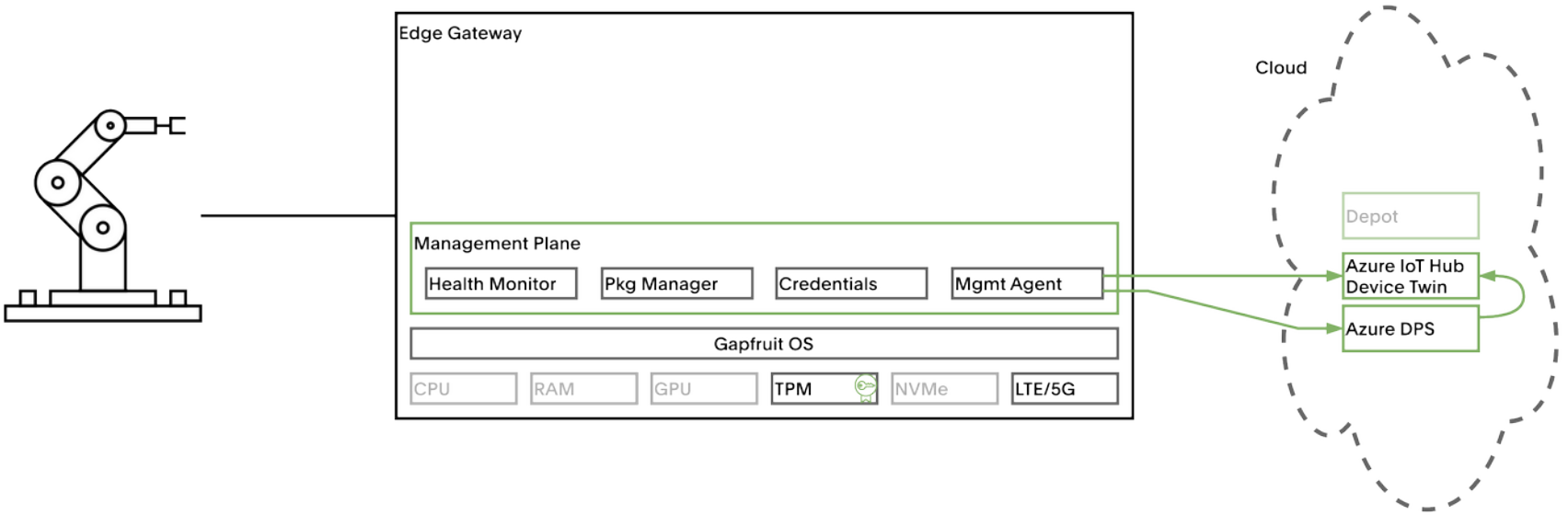
Demo: Zero-Touch Provisioning

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

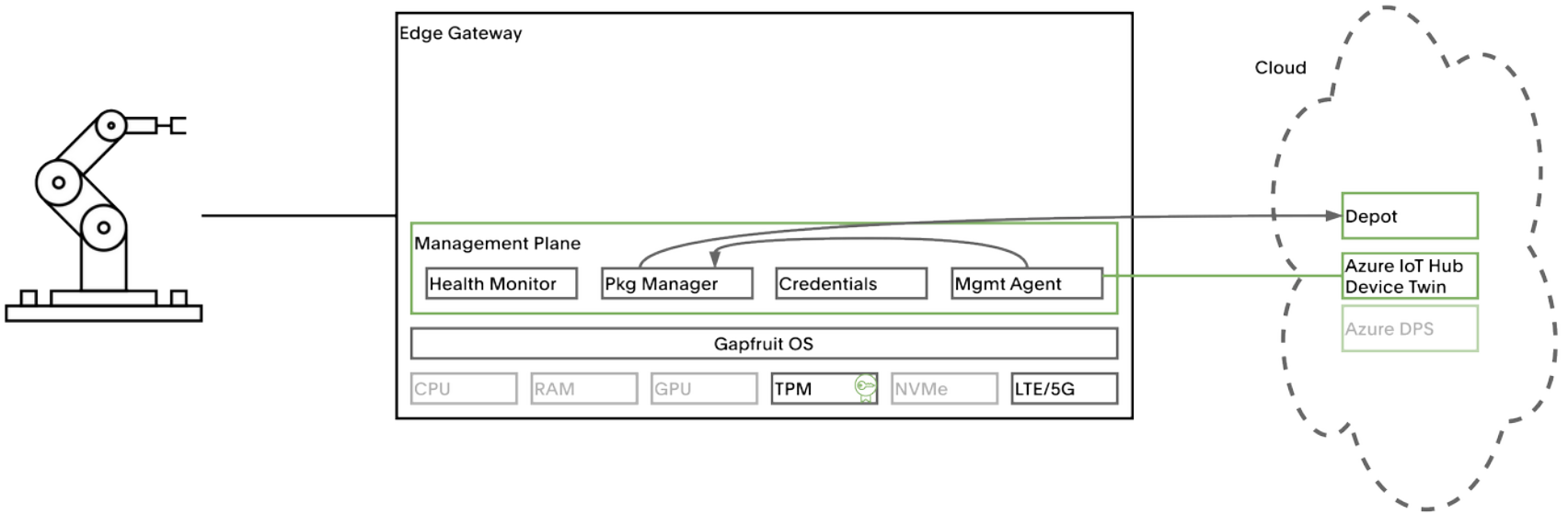
Step 1: Azure Device Provisioning Service



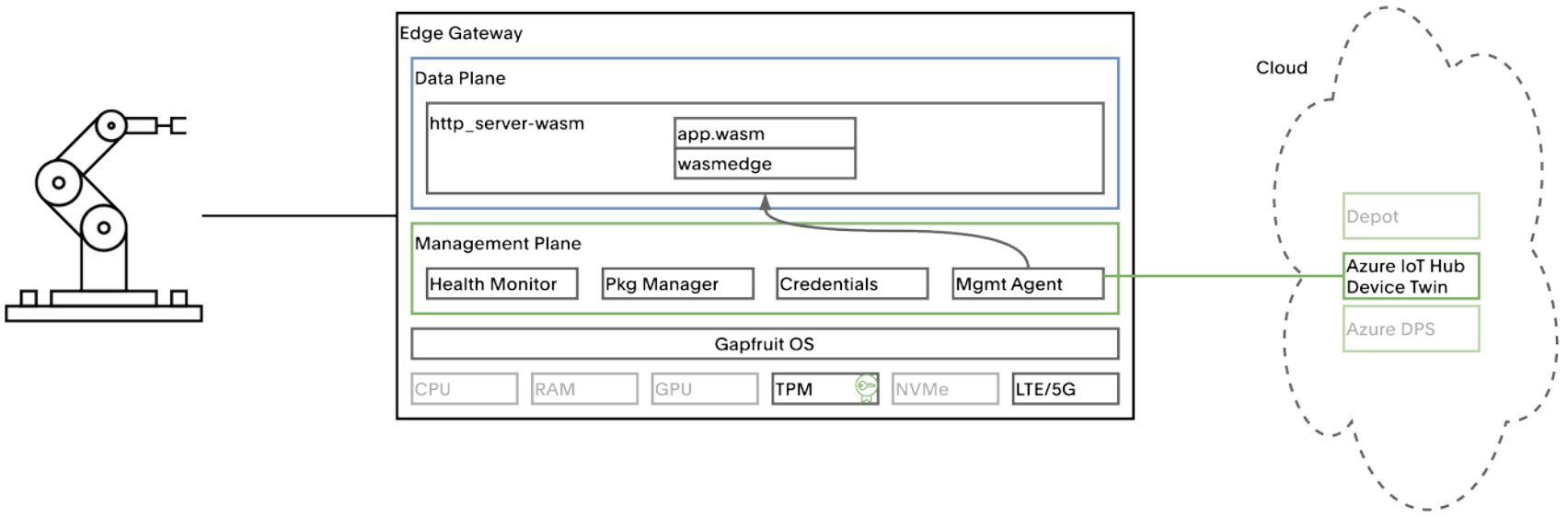
Step 2: Azure IoT Hub Device Twin



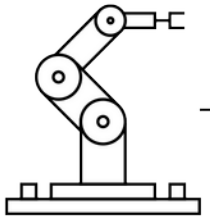
Step 3: Fetch WasmEdge App

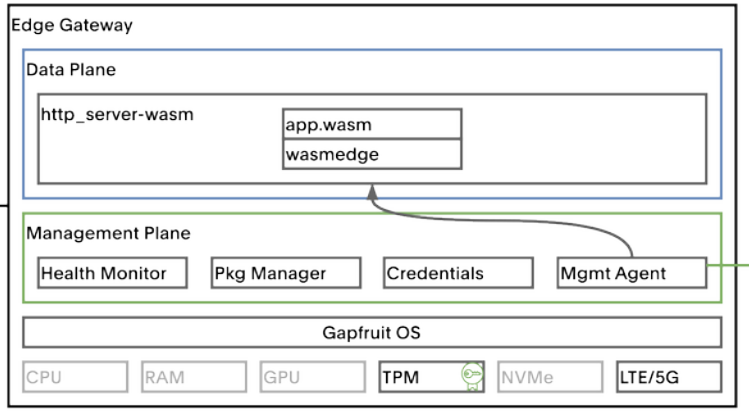


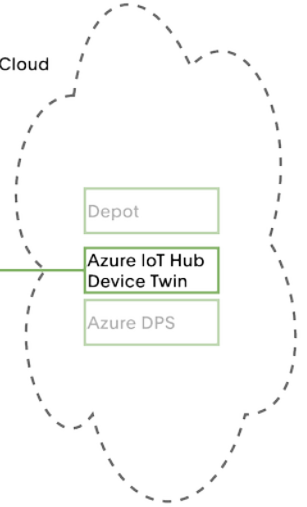
Step 4: Deploy WasmEdge App



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







Next Steps

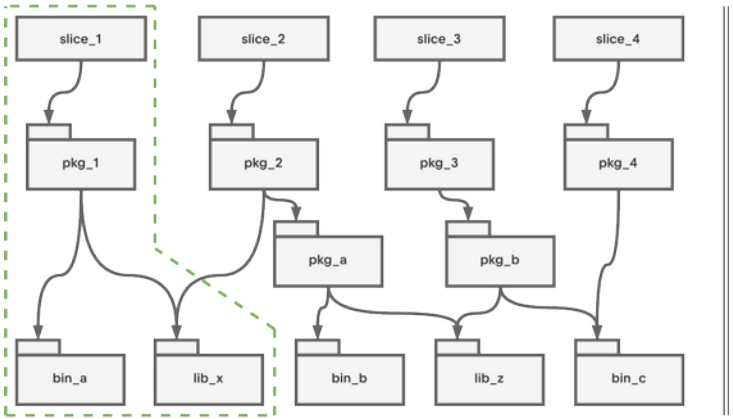
- WebAssembly/WASI standardization
- Better tooling for application developers
  - Support of OCI images?
- Enabling Gapfruit OS on i.MX8 with seL4
- ACME Device Attestation Extension [RFC]
- Finish work on the TPM resource manager **`tpm_abrmd`**

Sid Hussmann
CTO & Founder
sid.hussmann@gapfruit.com

https://gapfruit.com
Twitter: @sidhussmann
LinkedIn: https://linkedin.com/in/sidhussmann
Mastodon: @sidhussmann@infosec.exchange

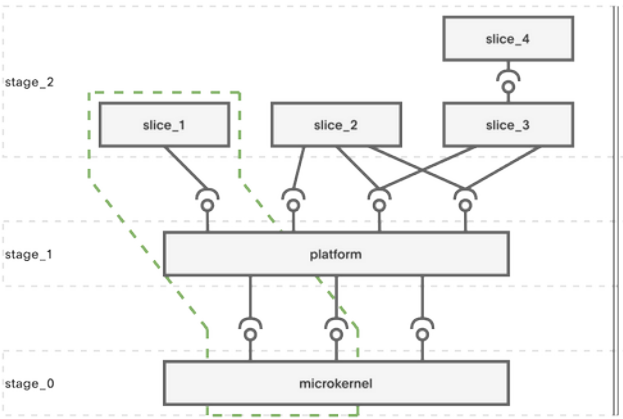
Control over Dependencies

Software Dependencies



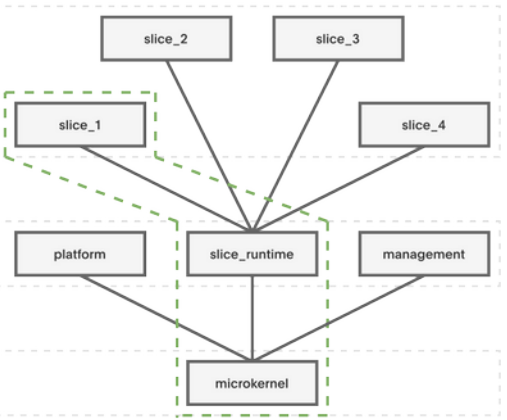
Lightweight Package Management

Service Topology



Service Oriented Architecture

Resource Distribution



Parent-Child Relationship