Smarter automotive experiences

Privacy-enhancing technologies with WebAssembly

P J Łaszkowicz - Omnifi Foundation (2025)

Introduction

Omnifi Foundation

- Non-profit cooperative foundation since 2019
- Focused on human-centric solutions for safer technologies
- Predominantly focused on privacy-preserving approaches
- Self-funded through advisory and consulting projects
- 94% of services built use existing WebAssembly frameworks (e.g. Spin and wasmCloud)

Introduction

Omnifi Foundation landscape (2024)

Taira

Rai

Naamio

Introduction

The automotive data problem

- Automotive OEMs can spend between €10m to €100m on data collection & retention.
- Data is not cleaned between drivers & owners.
- Data is typically sold, and rarely used for the driver’s benefit.
- Data is portable.

Introduction

The automotive data solution

- Data should be local-first
- Zero-knowledge proofs (ZKPs) and model weights can move
- WebAssembly as a tool for retrieval augmentation
- Personalization, fine-tuning, and training happen on-device

Introduction

Taira landscape

Secure and portable tooling to protect data and intelligence workloads:

- Taira Wallet (secure agentic wallet)
- Taira Shield (secure private network)
- Taira Vault (secure private data space)

Built on Naamio components.

In-cabin intelligence

Flight use case

- A family owned vehicle with knowledge from personal devices
- Family tickets for flight detected
- Vehicle warms, checks road conditions, and sets up for trip, reminding driver with a notification

In-cabin intelligence

Flight use case

- Family enter vehicle and journey is already mapped
- As journey begins, vehicle asks whether parking is required
- Parking is requested by voice and vehicle reserves a suitable space, updating the route

In-cabin intelligence

Flight use case

- Family arrive on time at security gate
- Whilst waiting for flight, vehicle asks whether cleaning is desired
- Cleaning is reserved during trip

In-cabin intelligence

Trip use case

- Family arrive at destination and collect rental vehicle
- Rental vehicle picks up preferences and is set up as original vehicle
- Insurance, toll passes, and parking are added to wallet automatically based on accommodation and plans

In-cabin intelligence

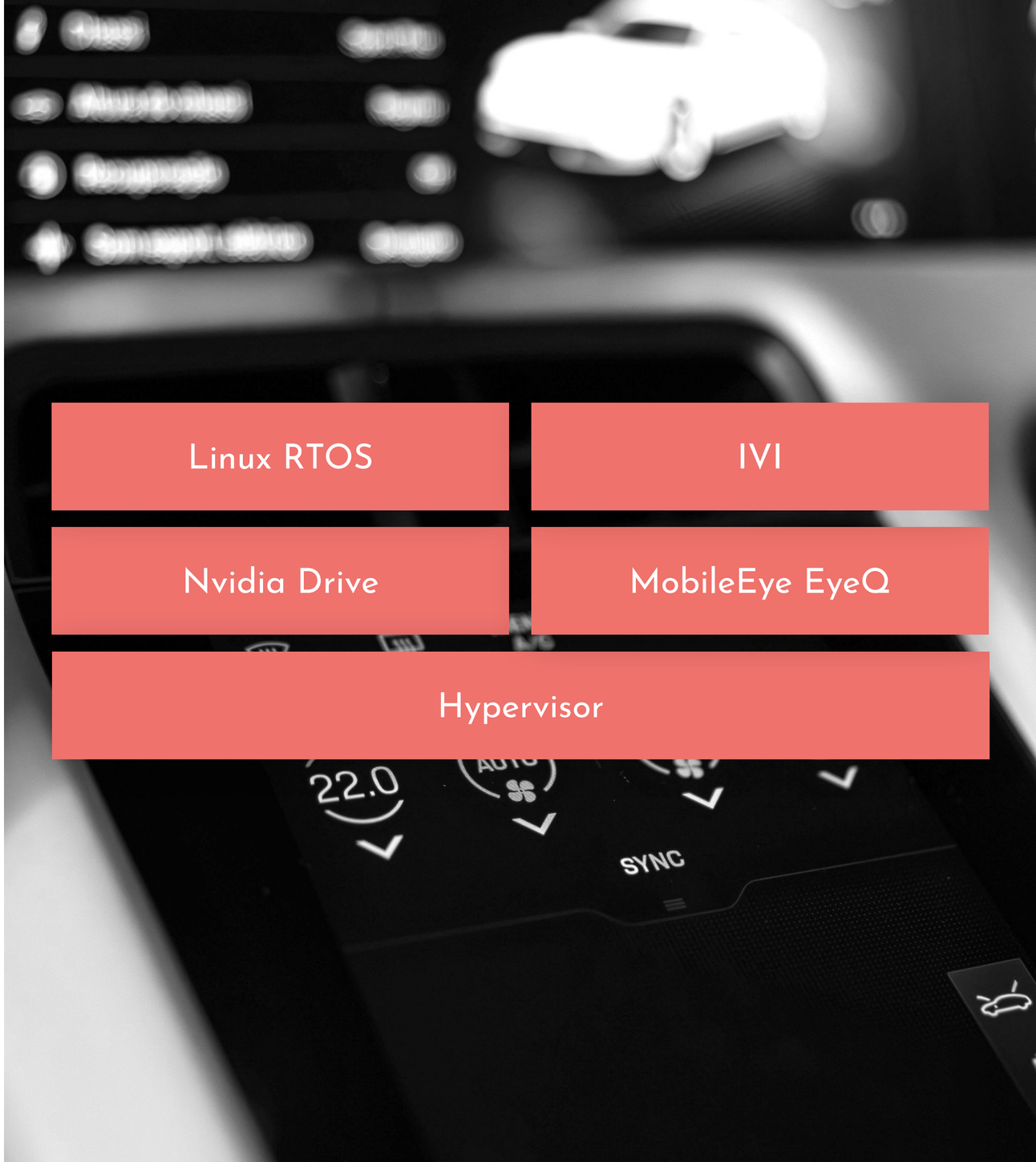
Trip use case

- Family are notified the original vehicle is being cleaned
- Temporary digital key is issued for duration of clean
- During road trip charging is pre-planned
- Food and drink are preordered based on family preferences

Automotive architecture

Automotive architecture

- Hypervisor
- Linux RTOS
- In-Vehicle Infotainment / Android Automotive (not Auto)
- Nvidia Drive
- Or Intel MobileEye (EyeQ)



Why WebAssembly?

- **Sandboxed environment** is a good starting point for a zero-trust security model
- **Polyglot tooling** enable great access to tooling and easier onboarding
- **Efficient resource use** means more sustainable and manageable workloads
- **Near-native performance** enables fluid experiences to be built
- **WAC and WIT** enable contract-first application development with uniform tooling

“You're currently at the bleeding edge of where we're taking Spin”

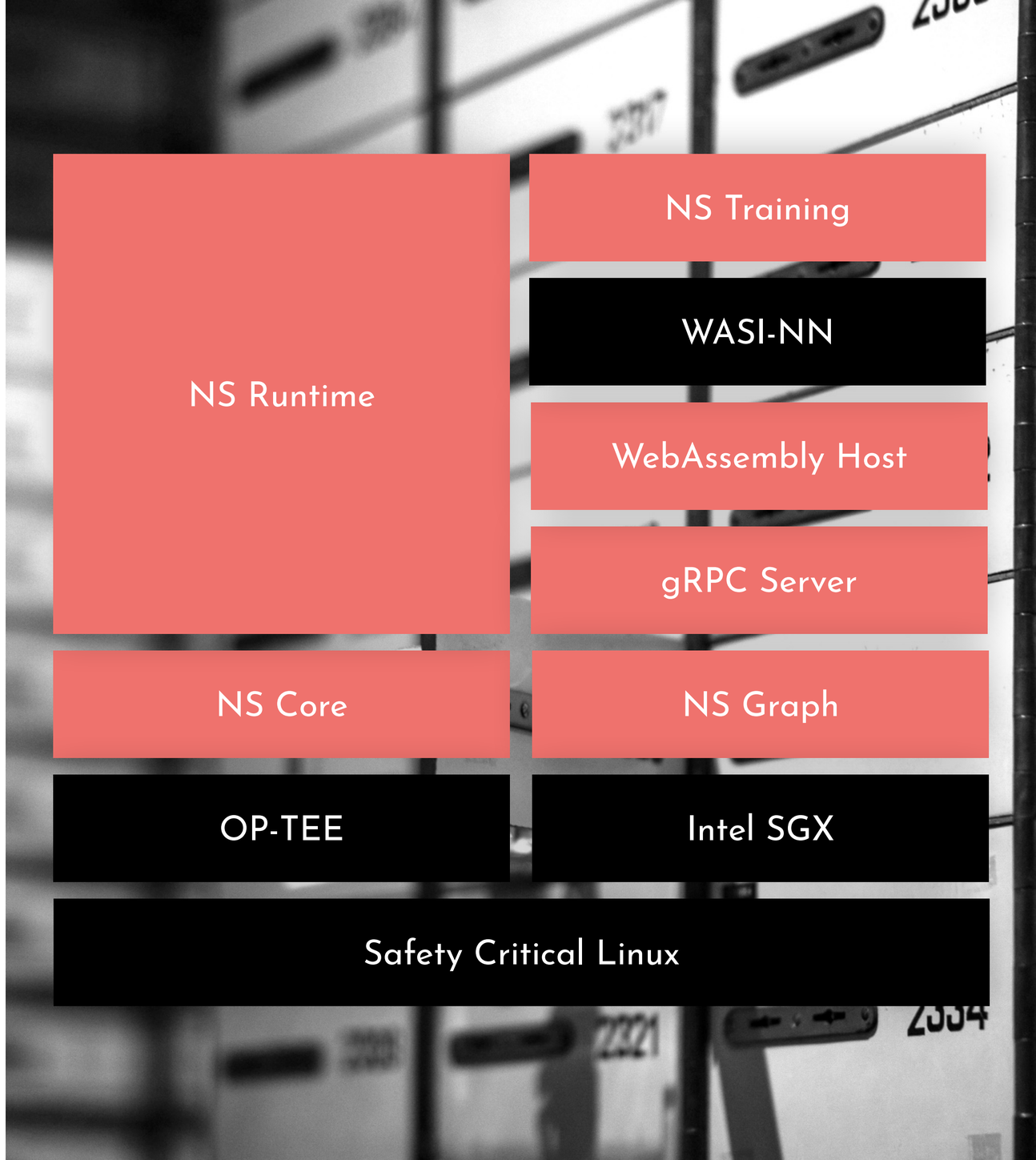
Fermyon (October 2024)

Naamio Space

Naamio Space architecture

Automotive PDS stack

- Open Portable Trusted Execution Environment / OP-TEE & Intel SGX
- gRPC & Protocol Buffers
- WebAssembly Host (Wasmtime / WAMR)
- Component Model (WAC / WIT)
- WASI-NN (for limited inference)
- Native training runtime



Rai

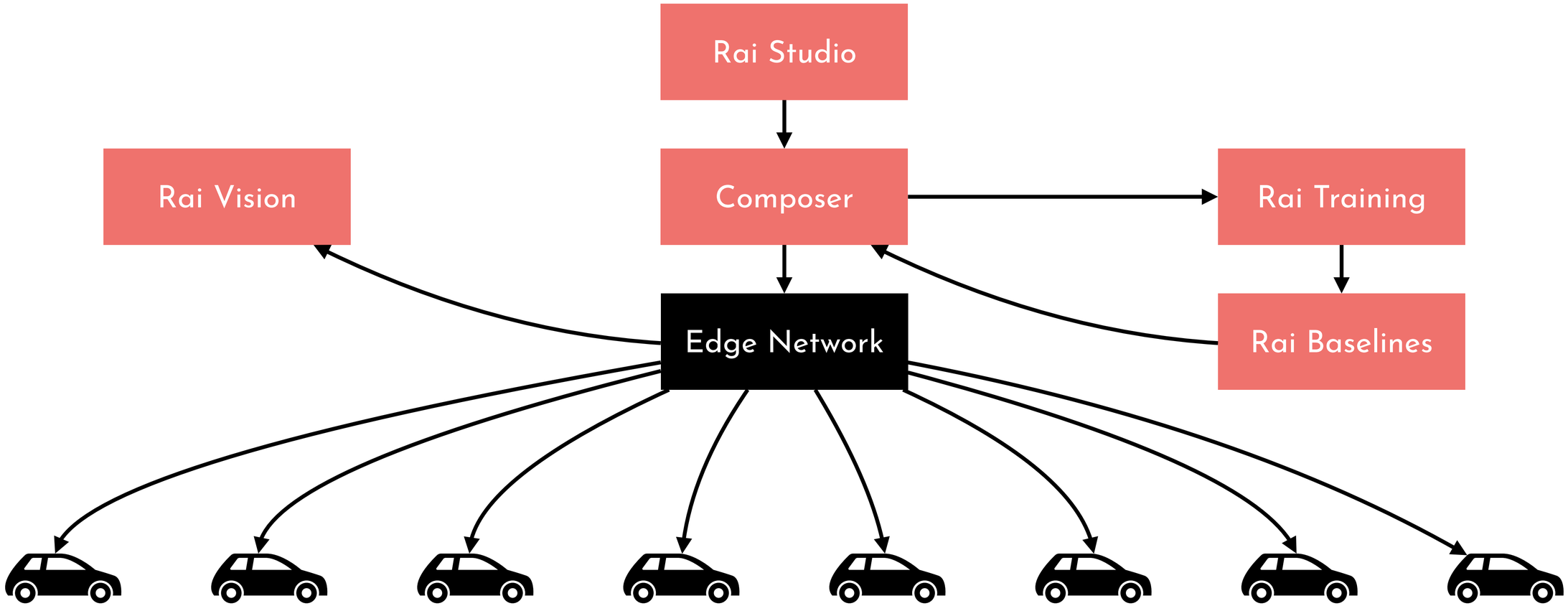
Rai

Responsible intelligence

- Safety-by-design intelligence tools
- Supports MLOps patterns
- Orchestrates training workloads across global edge network
- Provides guardrails out-of-the-box
- Supports JAX, PyTorch, and TensorFlow (so far)

Rai architecture

PDS fleet intelligence orchestration



Conclusions

- Fastly provides FaaS-like approach to starting service development
- Spin and wasmCloud provide strong tooling for customising service architecture
- Code re-use with the Component Model and WIT between edge and cloud is good
- Privacy-first inference is fine, but training is powerful
- It’s rarely necessary to collect data in order to provide intelligence-powered services
- WebAssembly is efficient, safe, and is now incredibly easy to use practically anywhere

Licensing

We’re fully committed to building ethical, open source tooling, including:

- LGPL for applications and services
- MPL for components and libraries

Thanks

Feel free to reach out