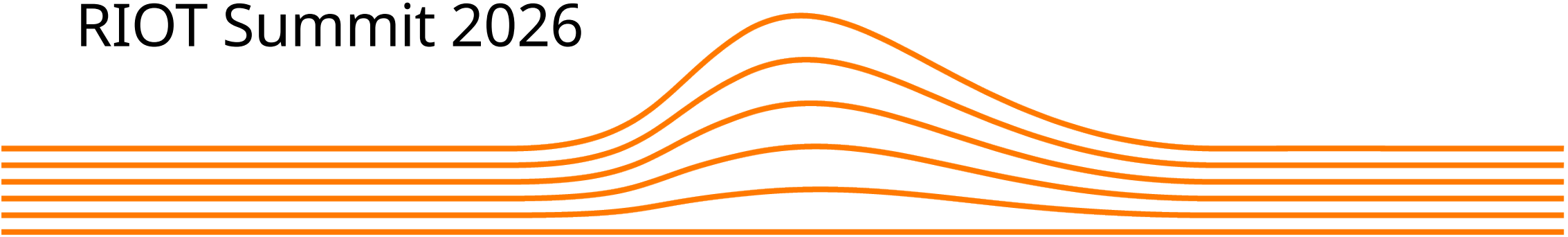
Orange Research

Shape the future and value innovation

Multi-tenancy and TinyML-as-a-Service on RIOT

Bastien BUIL

RIOT Summit 2026





Me



Bastien BUIL

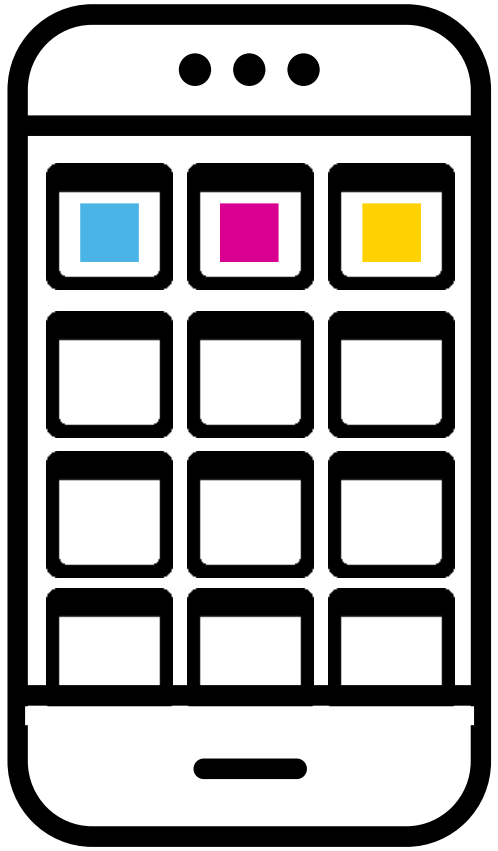
Software & Security Researcher

Last year PhD candidate specialized in cybersecurity and constrained systems with a focus on multi-tenant microcontrollers

End of PhD thesis: December 2026

Orange Innovation
Caen (France)

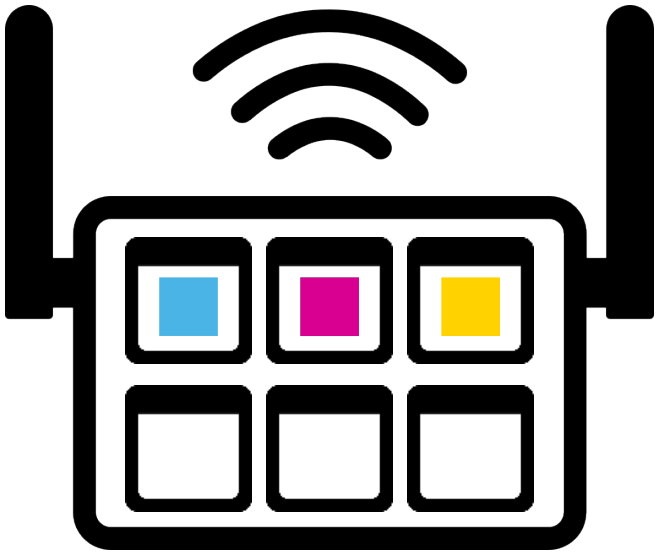
CEDRIC department
Cnam Paris (France)



1 Phone

Multiple isolated
third-party apps

- Multi-tenancy

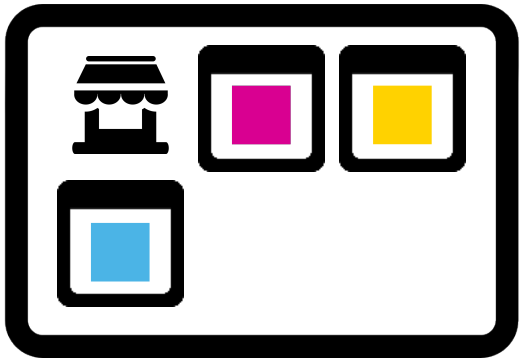


IoT device

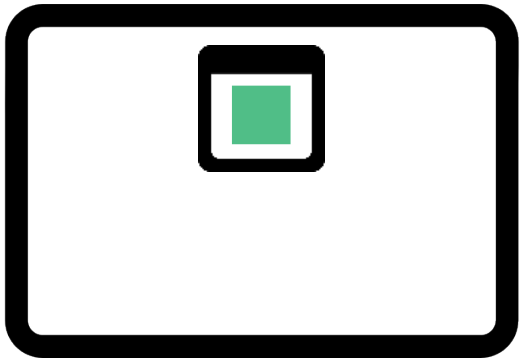
Can we have multi-tenancy on microcontrollers ?

Why?

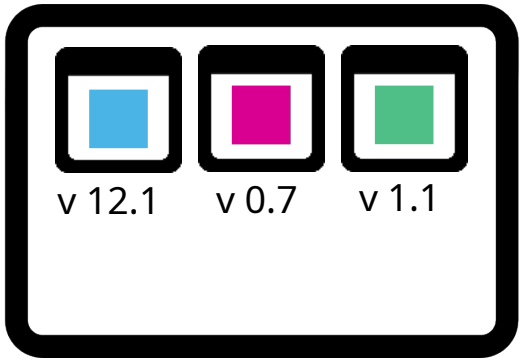
Use cases of multi-tenant microcontrollers



Marketplace



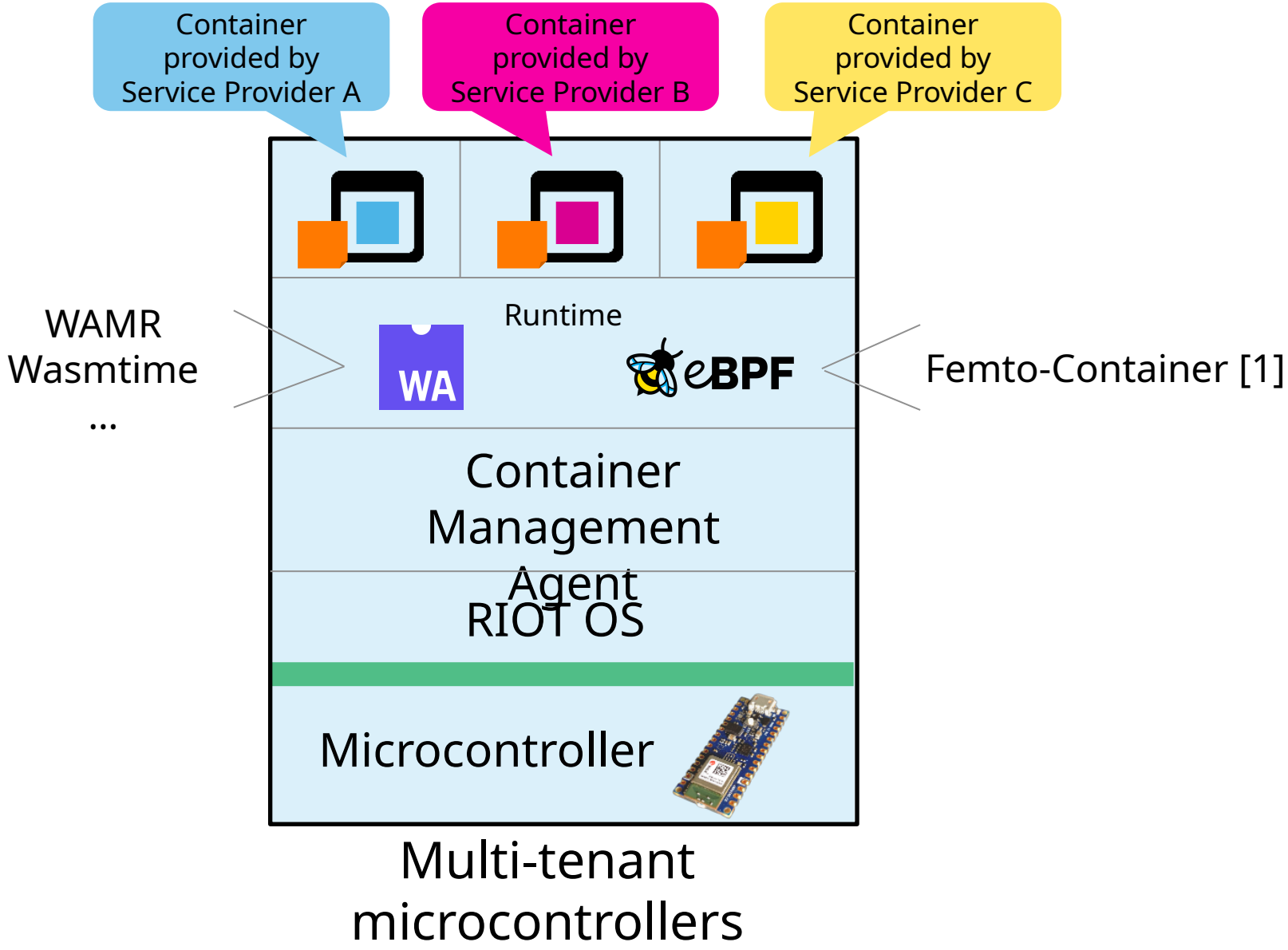
Regularly changing contractor



Composability for Security & DevOps

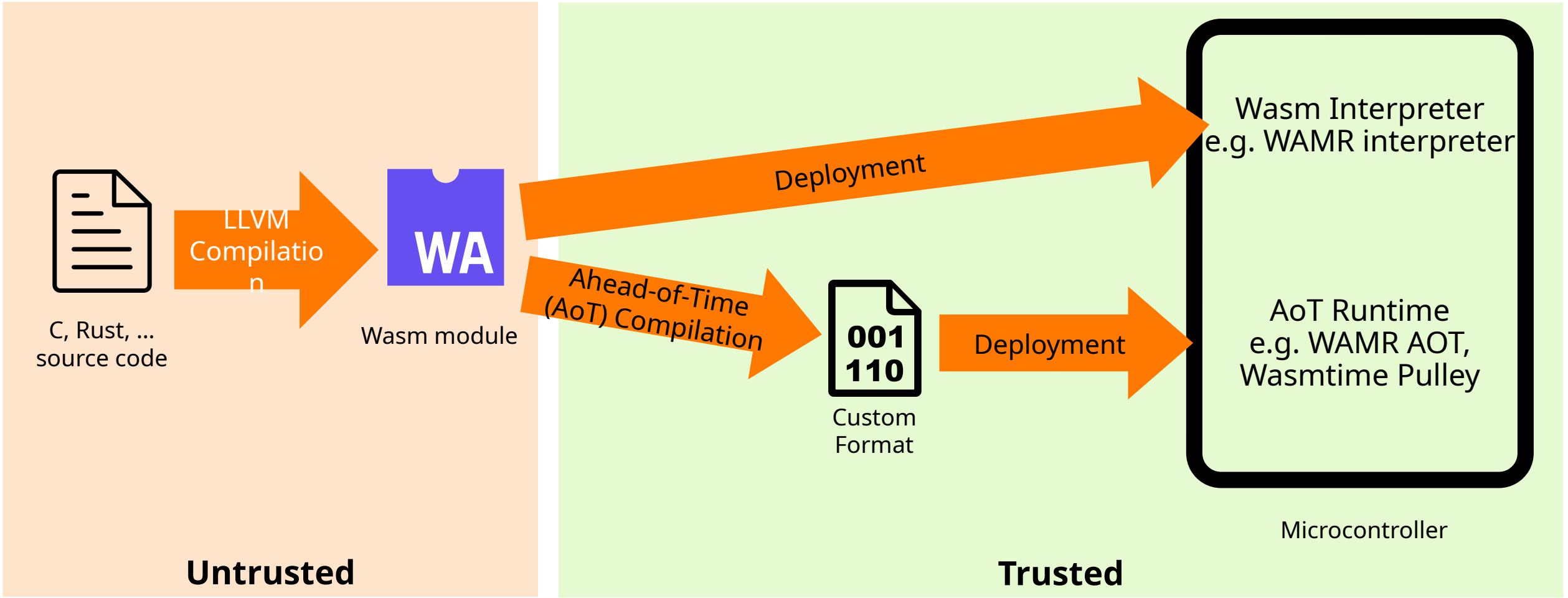
➢ Isolation & ease of deployment

The target: secure multi-tenant microcontrollers



[1] K. Zandberg, E. Baccelli. Minimal Virtual Machines on IoT Microcontrollers: The Case of Berkeley Packet Filters with rBPF. In Proc. of 9th IFIP/IEEE PEMWN, Dec. 2020.

WebAssembly (Wasm) on Microcontrollers



Empowering containerization solutions

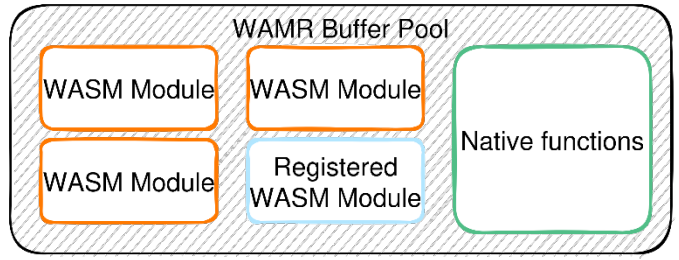
Limitation of current WebAssembly runtimes

- **No dynamic management of permissions**
- **No scheduling of applications**
- **Some runtimes do not control memory usage of applications**

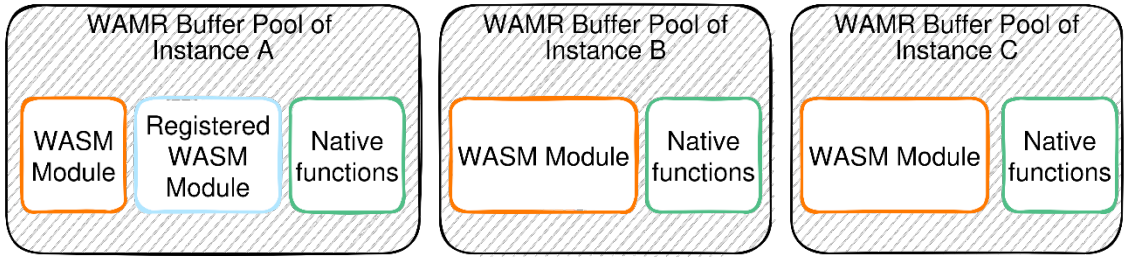
We have developped two solutions

CS4WAMR

Wrapper around WebAssembly Micro Runtime

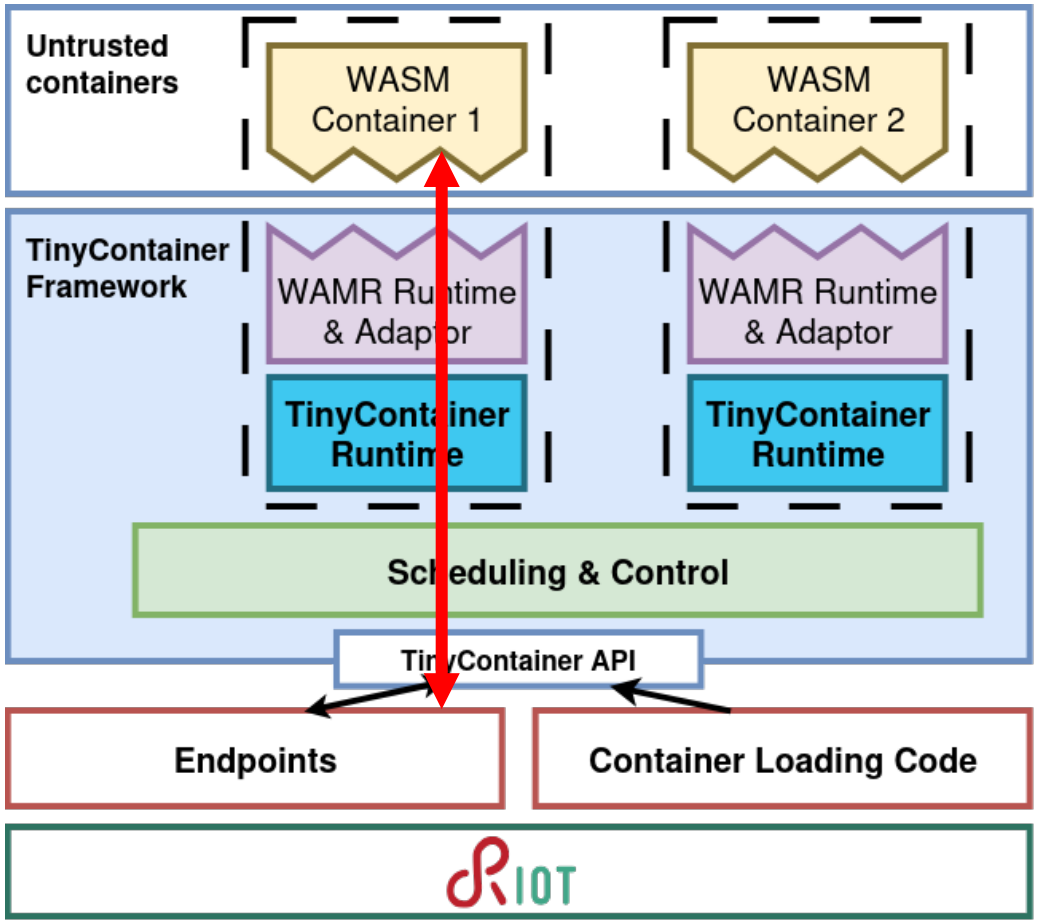


WAMR default memory management



CS4WAMR memory management

We have developped two solutions



TinyContainer

Container management middleware for microcontrollers

We have developped two solutions

CS4WAMR

Wrapper around WebAssembly Micro Runtime

Goals:

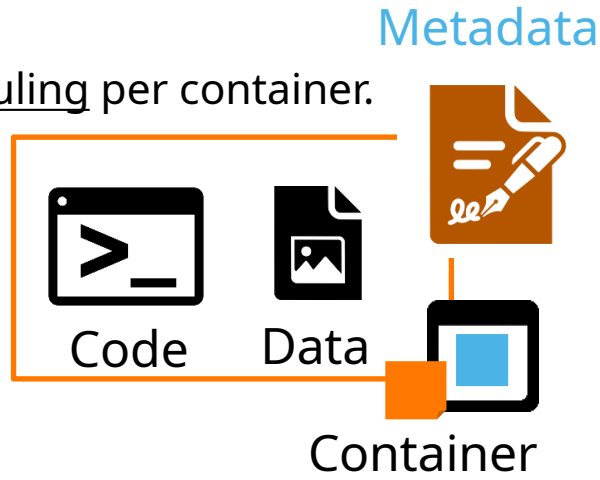
- Control of the memory usage by containers
- Fine-grained permissions by containers
- Scheduling based on RIOT threads and round-robin scheduler
- Container restoration mechanism
- First step for hardware isolation of WAMR with separate memory pool per container

TinyContainer

Container management middleware for microcontrollers

Goals:

- Endpoint system for accessing host resources,
- Permission management per container,
- Dynamically-loaded metadata,
- Configurable execution scheduling per container.



Short conclusion of containerization for microcontrollers

Limitations of our propositions:

- No built-in inter-container communication
- No hardware isolation of memory implemented yet
- No deep integration in the OS

Limitations of our WebAssembly isolation:

- Big overhead when doing interpretation
- RAM overhead

TinyML (Tiny Machine Learning)

Running Machine Learning models on constrained devices

Local models notably allows:

- **Privacy: data are not quitting the device**
- **Filtering/processing prior sending to Cloud**

However TinyML, like any code, can contain malware.

Can we leverage containerization solution ?

TinyML empowered by containerization

TinyMLaaS Use case: two variants

- We propose to integrate TinyMLaaS in the multi-tenant microcontrollers architecture

Variant 1: ML run in Container

- Require AoT for Performance
  - Required trusted third-party
- No in-host integration

Variant 2: Driver-based ML

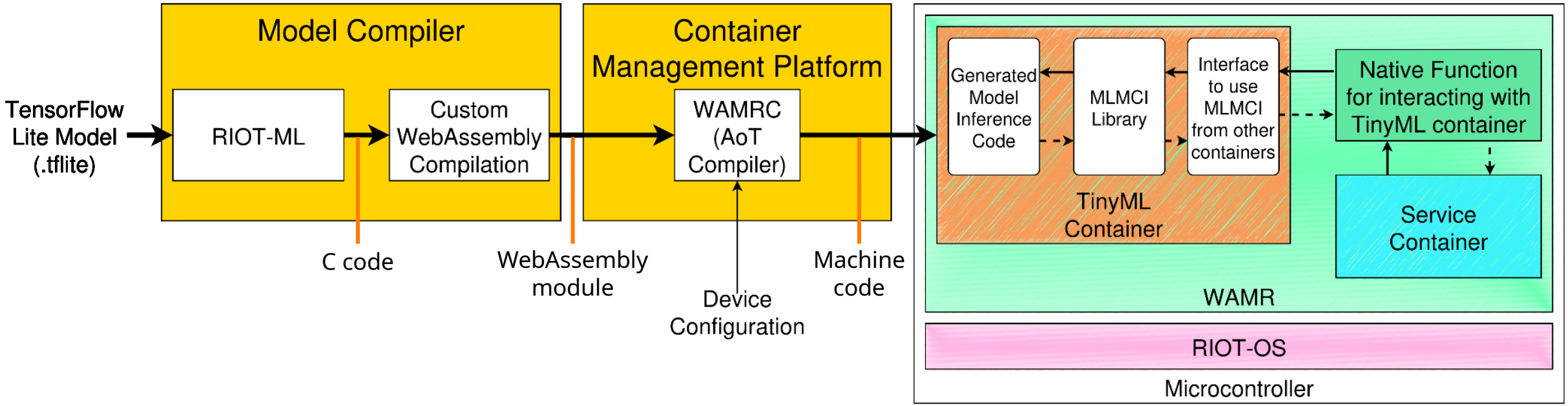
- No AoT requirement
- Use in-host integration for TinyML service

Variant 1: ML in Container

PoC: TinyML running in WebAssembly containers using Ahead of Time compilation

Based on open source tools:

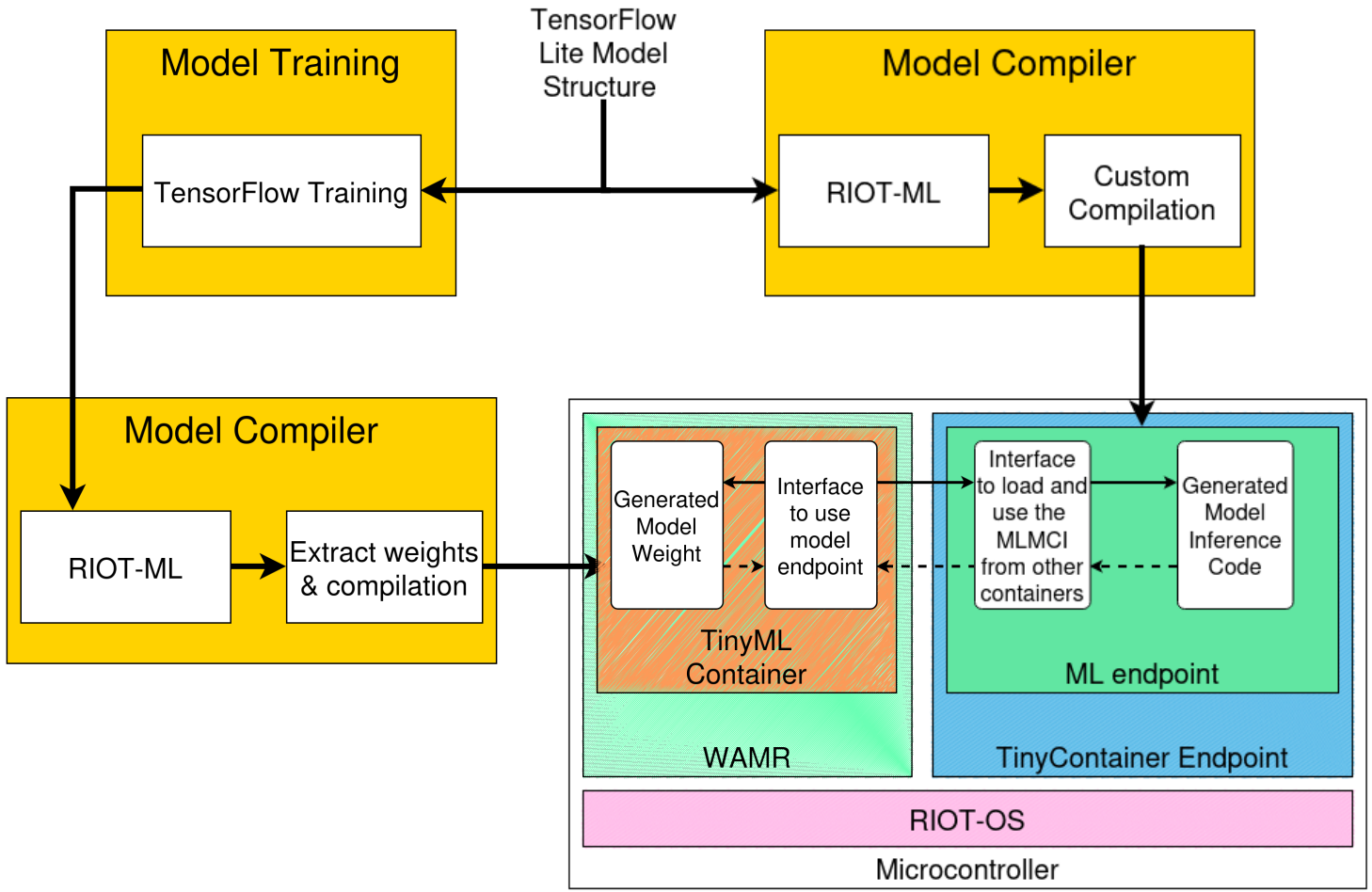
- **RIOT-ML [1]: compile ML models to C code using Apache TVM**
- **WAMRC: Wasm to machine code compiler**
- **RIOT OS**



[1] Huang, Z., Zandberg, K., Schleiser, K., & Baccelli, E. (2024). RIOT-ML: toolkit for over-the-air secure updates and performance evaluation of TinyML models. Annals of Telecommunications, 1-15.

Variant 2: Driver-based ML

Proposition: The host optimizes one type of model and allow container to run their model through an endpoint provided by TinyContainer



Observation on containerized TinyML on multi-tenant microcontrollers

Advantages:

- **Deploying as-a-Service TinyML models on microcontrollers**
- **Good performances with AoT compilation and TinyML**

Limitations:

- **High memory usage (x2 ROM and RAM)**
- **No integration with underlying hardware (such as AI chip)**
- **Interpreter mode give a big performance overhead (execution time x96)**
  - So Ahead-of-Time compilation should be use for resource intensive tasks

Optimization possible with WebAssembly Bulk Memory Operations Proposal

Key takeaways

- **Containerization on microcontrollers opens for new use cases and can add security**
- **Memory overhead should not be overlooked**
- **Software isolation solutions on microcontrollers are moving fast**

Multi-tenant microcontrollers: a vibrant ecosystem domain

WebAssembly Micro Runtime

VeloxVM

Femto-Container

SpaceWasm

WasmTime Pulley

Ocre & Atym

Propeller by Abstract Machines

WaseFire

TreVM

MicroEJ

• • •

Thanks

Open Source Projects available on:

https://github.com/Orange-OpenSource/CS4WAMR

https://github.com/Orange-OpenSource/RIOT/tree/tinycontainer/examples/advanced/tinycontainer

By Bastien BUIL

RIOT Summit 2026

orange™ le cnam

TinyContainer

CS4WAMR