Scalable and Lightweight Cloud-Native Application Sandboxing

Anastassios Nanos, Charalampos Mainas,
Georgios Ntoutsos, and Ilias Lagomatis

Nubis PC

About us

Diverse team of researchers & engineers, working on facilitating and optimizing application execution in Cloud and Edge environments.

Focus areas:

- Low-level systems software
- Hardware acceleration
- Serverless computing

Involved in Research & Innovation Actions and commercial projects

NUBIS

nubificus

About us

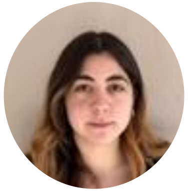
- Hardware acceleration abstractions
- Cloud-native IoT
- Container runtimes

NUBIS

nubificus

Apostolos Giannousas
Software Engineer
agian@nubificus.co.uk

Maria-Rafaella Gkeka
Research Engineer
mgkeka@nubificus.co.uk



Maria Goutha
Software Engineer
mgouth@nubificus.co.uk

Ilias Lagomatis
Systems Software Engineer
ilago@nubificus.co.uk

Charalampos (Babis) Mainas
Research Engineer
cmainas@nubificus.co.uk

Panos Mavrikos
Software Engineer / K8s
pmavrikos@nubificus.co.uk

Anastassios (Tassos) Nanos
Systems Researcher
ananos@nubis-pc.eu

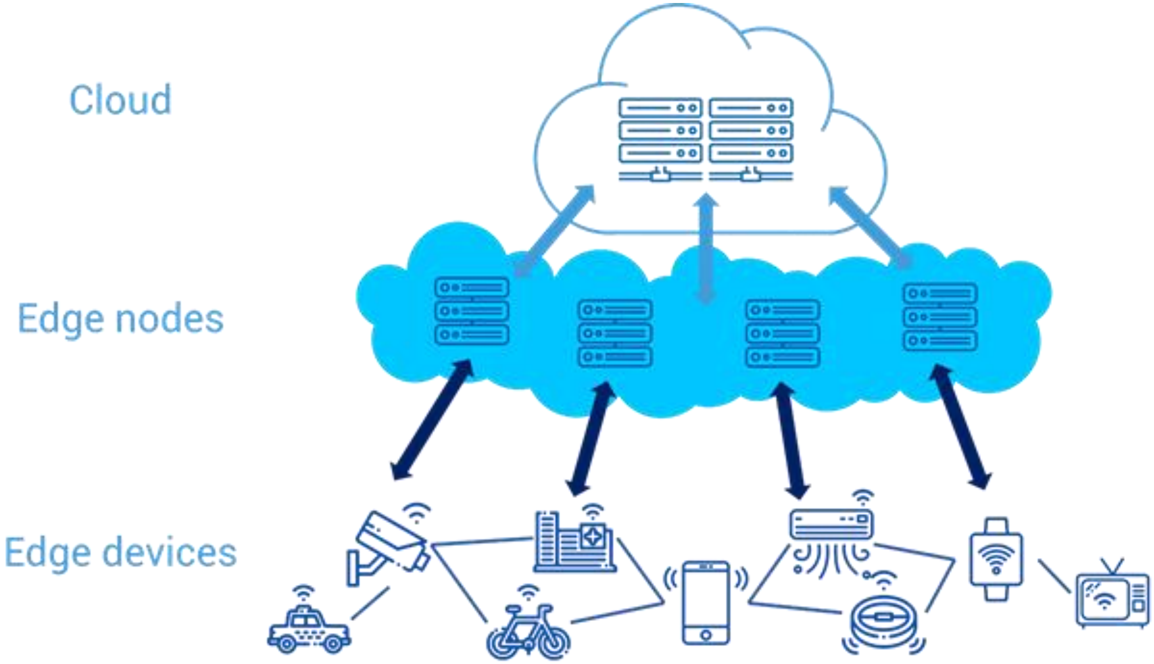
Georgios Ntoutsos
Software & Systems Engineer
gntouts@nubis-pc.eu

Kostis Papazafeiropoulos
Systems Researcher
papazof@nubis-pc.eu

Overview

- Challenges
- Lightweight applications
  - containers
  - unikernels
    - urunc/bunny (bima/pun)
- Hardware acceleration*
  - Cloud-native integration
  - Roadmap & Plan

Cloud-Edge-IoT continuum



Cloud-Edge-IoT continuum

Diverse requirements at each stage of execution:

- Cloud:
  - Vast resources
  - Mostly homogeneous
- Challenges:
  - data security & privacy
  - multi-cloud management
  - interoperability & flexibility

Cloud-Edge-IoT continuum

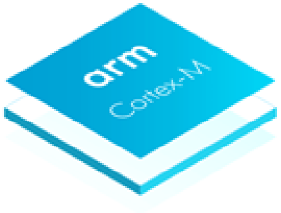
Diverse requirements at each stage of execution:

- Edge:
  - Lots of different devices available for the Edge
    - How to deliver applications ?
    - How to manage multi-tenancy ?
    - How to use & expose hardware accelerators ?

Cloud-Edge-IoT continuum

Diverse requirements at each stage of execution:

- IoT:
  - Even more types of devices available
    - All with their own proprietary SDK
    - No OS – deploy applications OTA (requires manual/user intervention)

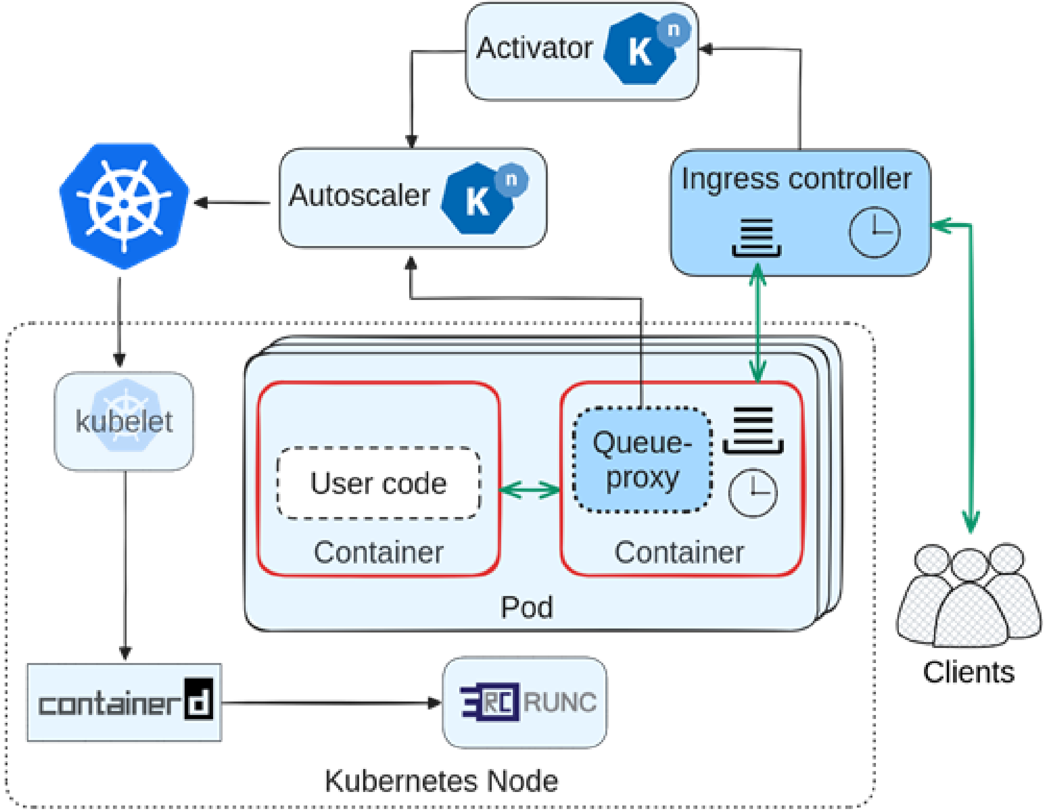


Application Deployment & Execution

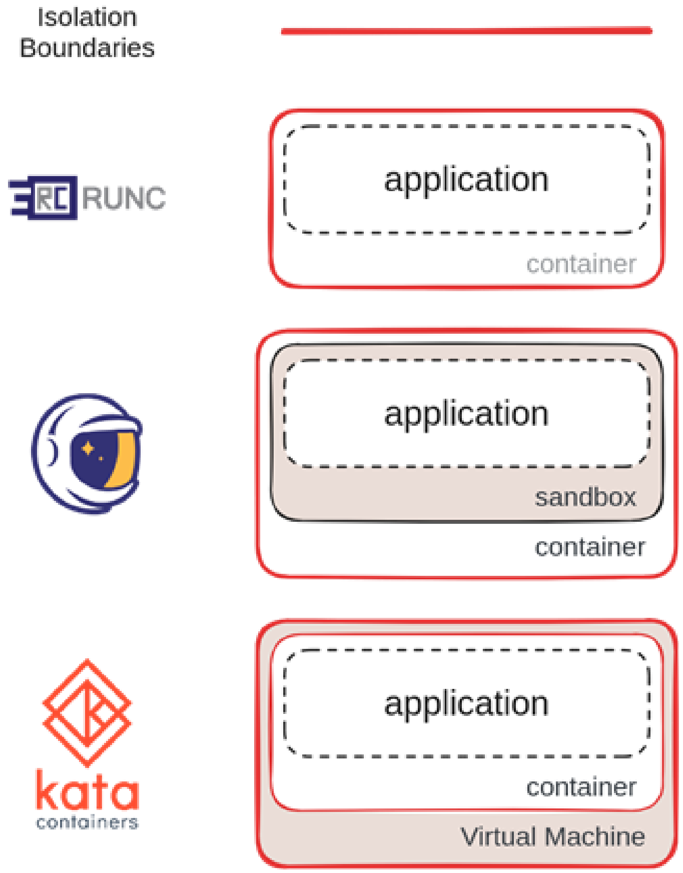
We want to execute a single application in a "containerized" environment

1. How do we sandbox this container so that it doesn’t affect the rest of the infrastructure?
2. Do we need the whole systems stack (OS, all libraries, loaders) ?
   - For instance in a serverless example: Knative

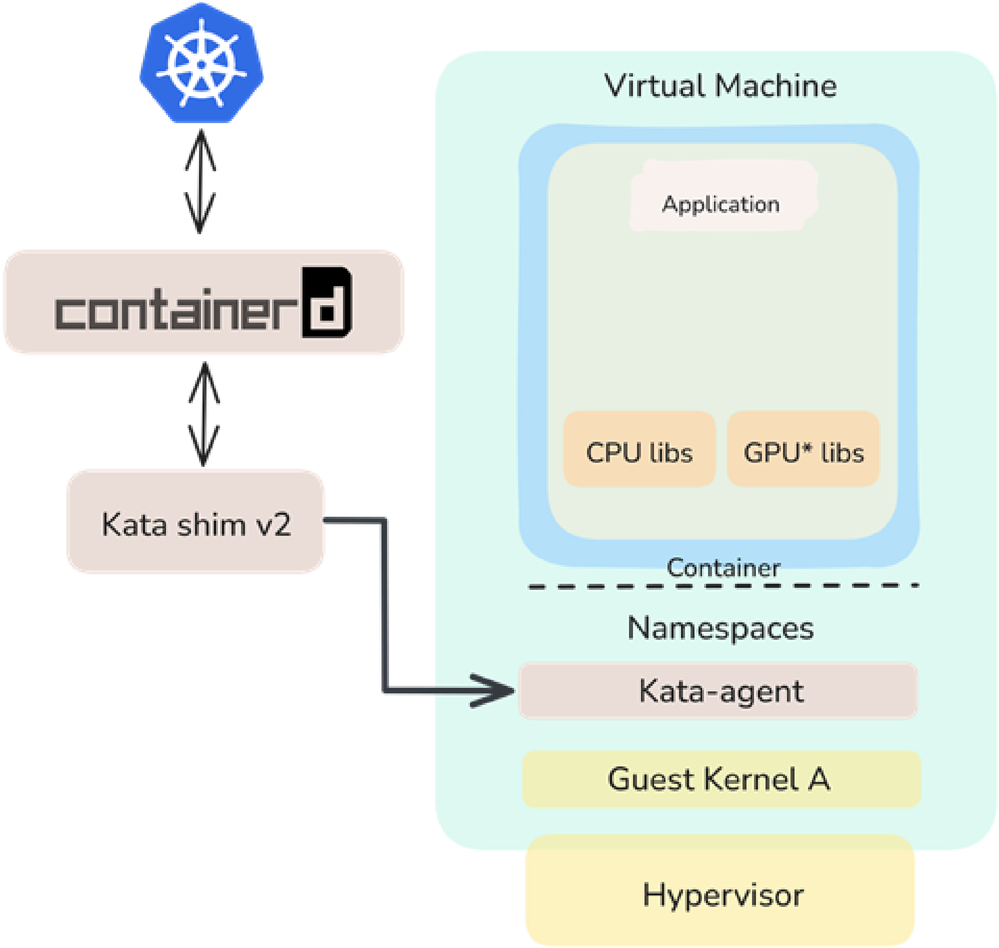
Application Deployment & Execution



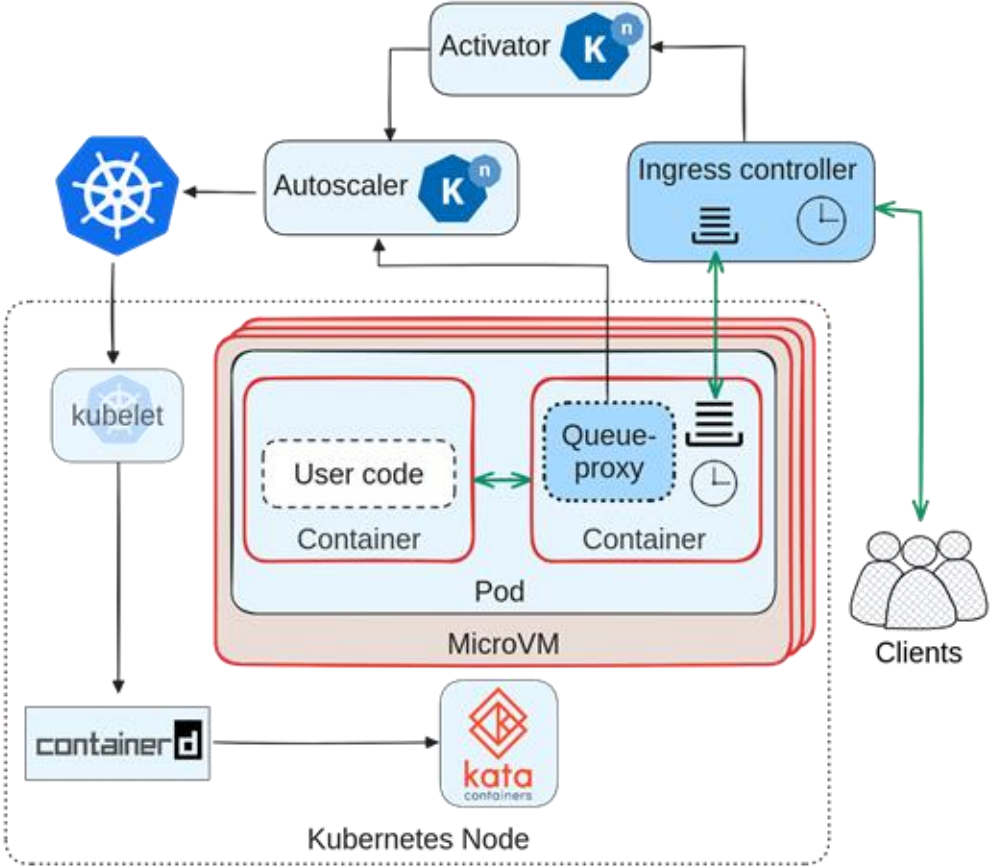
Application Deployment & Execution



Application Deployment & Execution

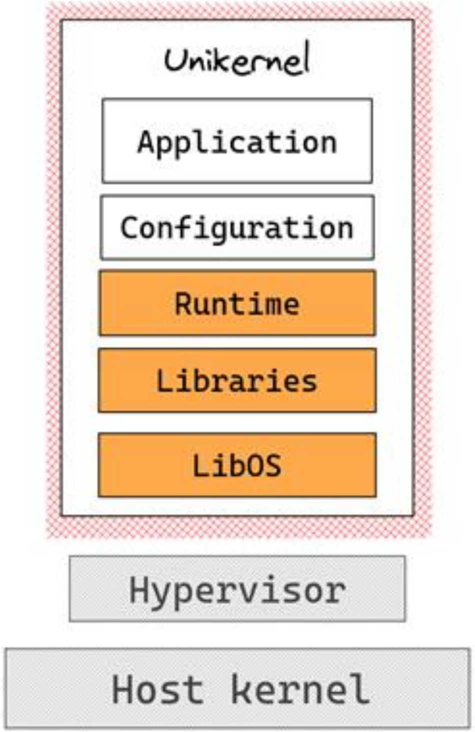


Application Deployment & Execution



Introducing unikernels

- A unikernel is:
  - specialized
  - single address space
  - constructed using a LibOS
- In other words:
  - Tailored for a single application
  - No separation between kernel and user space
  - Contains only what is needed



Containers vs Unikernels

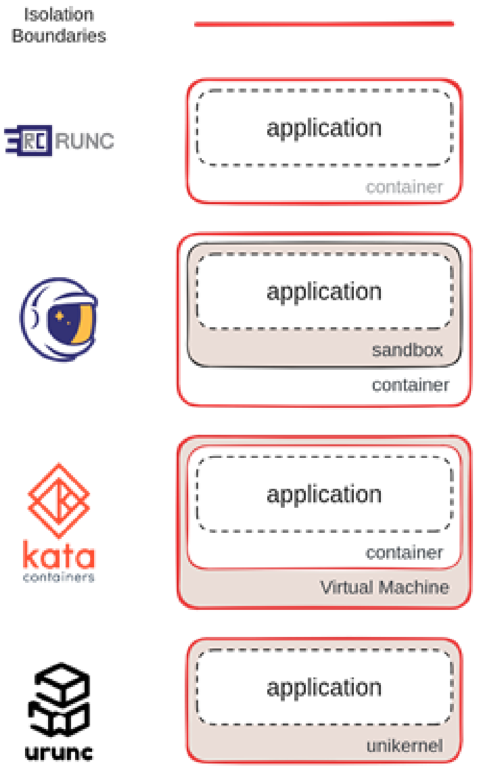
- Lightweight -> More lightweight
- Fast spawn time -> Even faster
- Portable -> Similar
- Scalable -> Similar
- Somewhat isolated -> Truly isolated

poor Cloud-native integration

urunc: unikernel containers

nubificus/urunc

- → CRI-compatible runtime written in Go
- → Treats unikernels as processes -- directly manages applications
- → Spawn unikernel VMs with generic hypervisors
- → Extensible, easy to add support for more unikernel frameworks & hypervisors
- → Hides complexity of unikernel framework-specific hypervisor and command line options

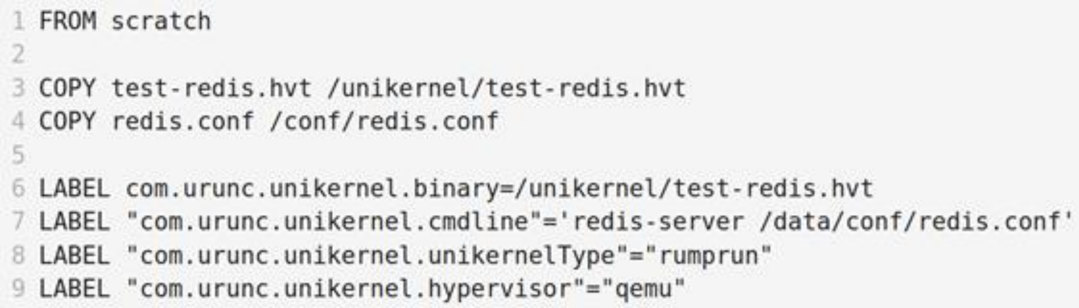


urunc: unikernel OCI images

nubificus/bima

To facilitate packaging, we build a **specialized image builder.**

- **bima** uses a dockerfile-like syntax to create OCI images:

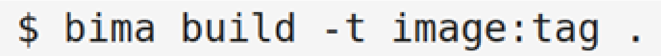


```
FROM scratch

COPY test-redis.hvt /unikernel/test-redis.hvt
COPY redis.conf /conf/redis.conf

LABEL com.urunc.unikernel.binary=/unikernel/test-redis.hvt
LABEL "com.urunc.unikernel.cmdline"='redis-server /data/conf/redis.conf'
LABEL "com.urunc.unikernel.unikernelType"="rumprun"
LABEL "com.urunc.unikernel.hypervisor"="qemu"
```

- Sample **bima** invocation:



```
$ bima build -t image:tag .
```

Lightweight sandboxes for Serverless Functions

deploy a simple HTTP header echo (httpreply) service

Point to:

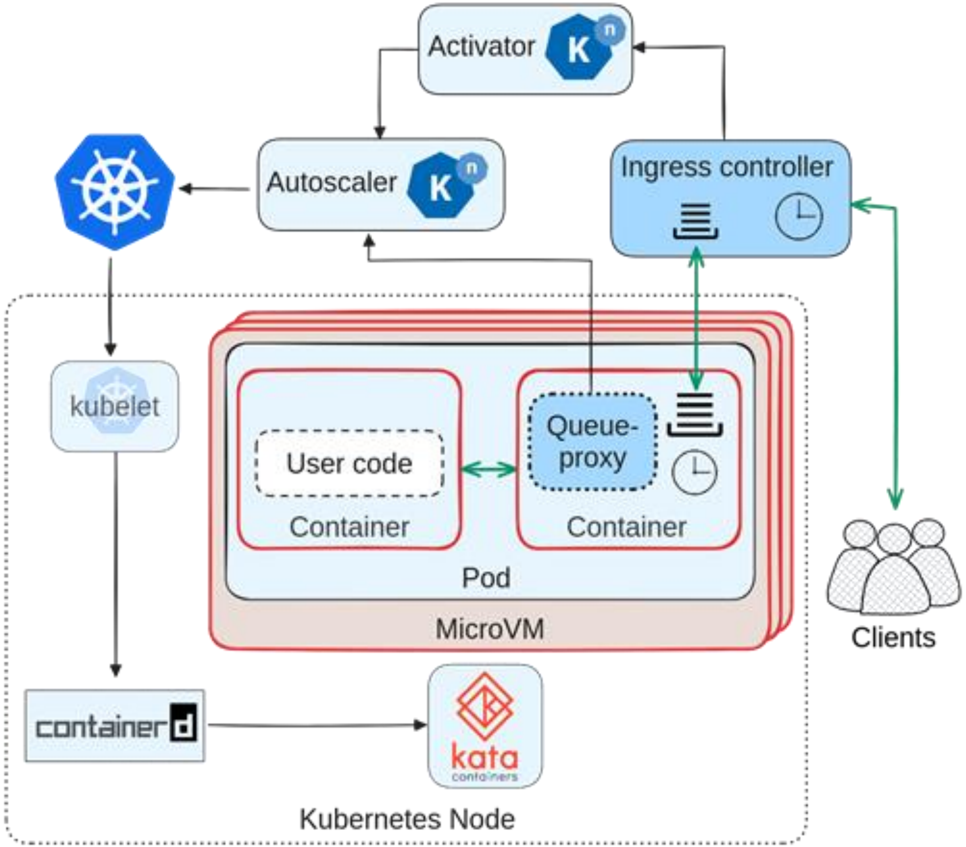
`https://hellocontainer.hipeac.nbfc.io`

`https://hellors.hipeac.nbfc.io/`

`https://helloclh.hipeac.nbfc.io/`

`https://helloqemu.hipeac.nbfc.io/`

`https://hellofc.hipeac.nbfc.io/`



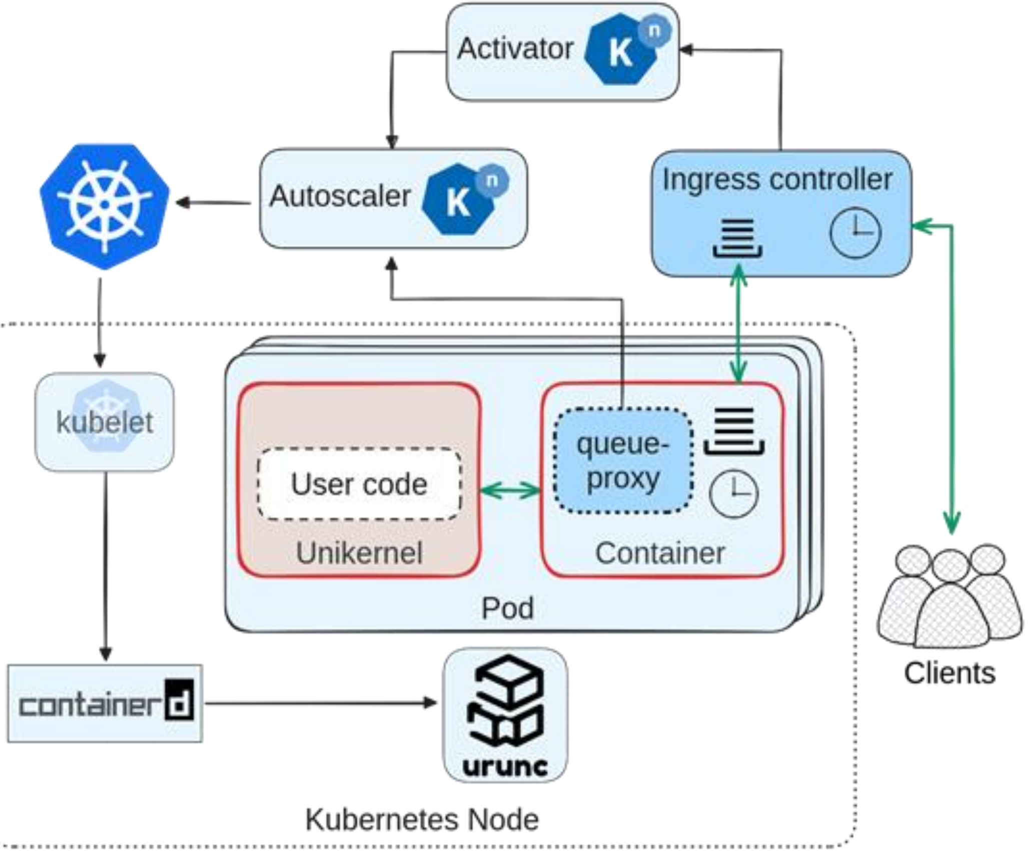
Lightweight sandboxes for Serverless Functions

deploy a simple HTTP header echo (httpreply) service

Point to:

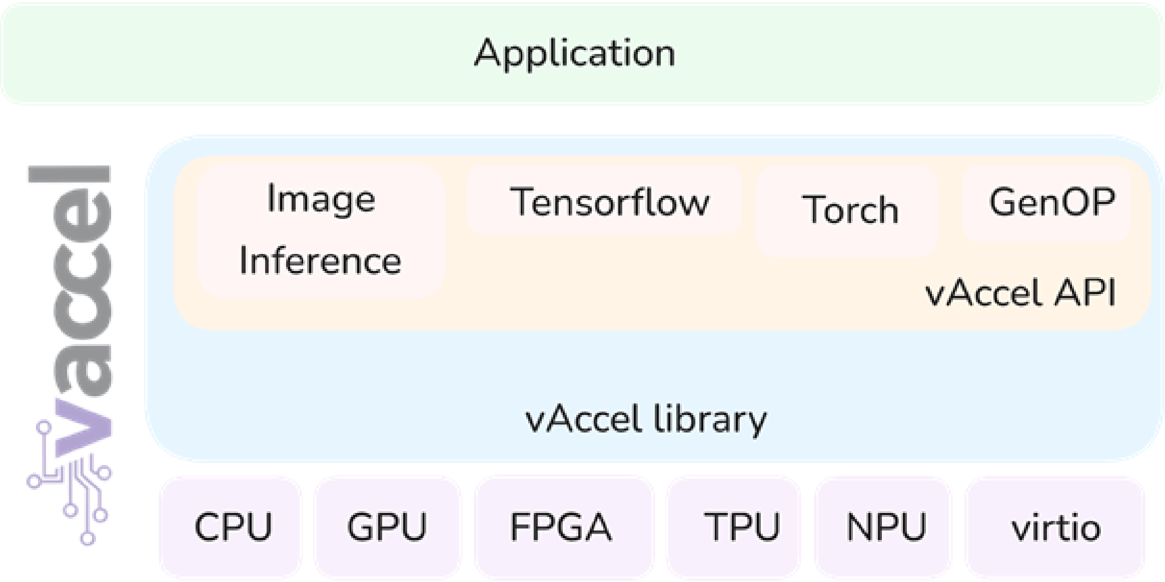
`https://hellouruncfc.hipeac.nbfc.io`

`https://hellouruncqemu.hipeac.nbfc.io`



The vAccel Framework

High-level Architecture

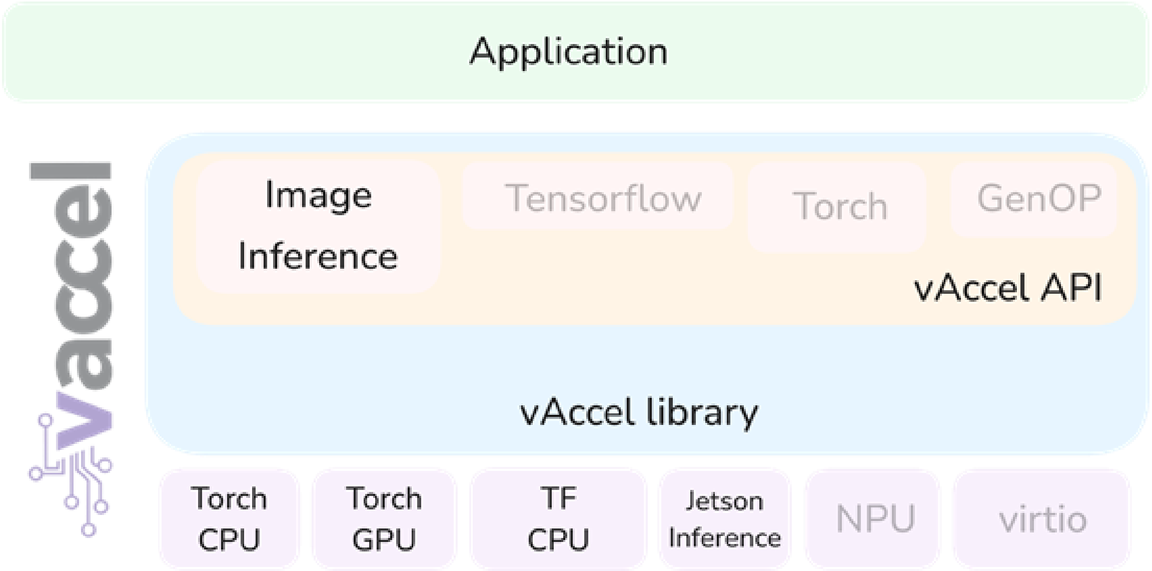


The vAccel Framework

Image Classify Example

- operation: `vaccel_image_classify`
- Multiple plugins
- Chosen based on specific criteria

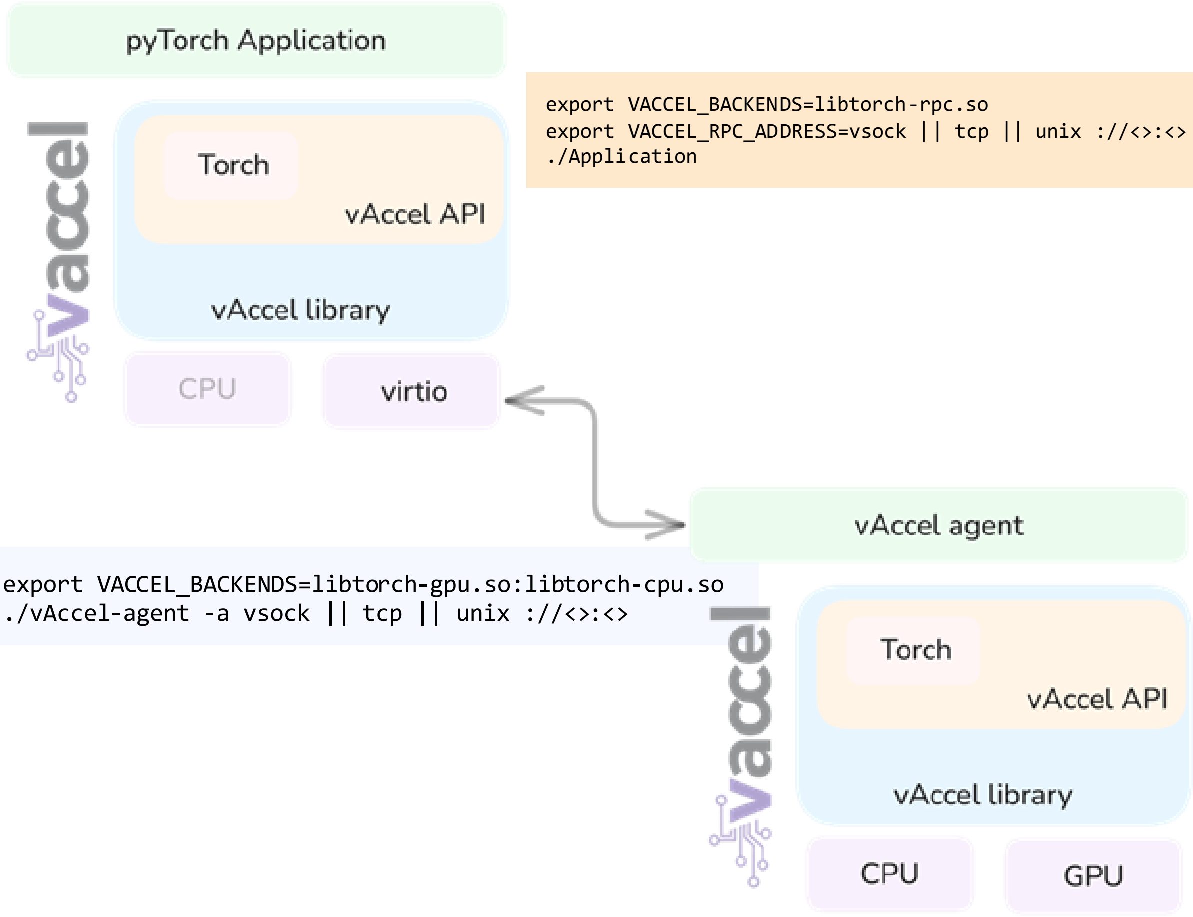
```
export VACCEL_BACKENDS=libtorch-gpu.so:libtorch-cpu.so:libtf-gpu.so:libjetson-
inference.so
./Application
```



The vAccel Framework

Remote Execution:

- Transfer inputs
- Forward operation
- Receive results



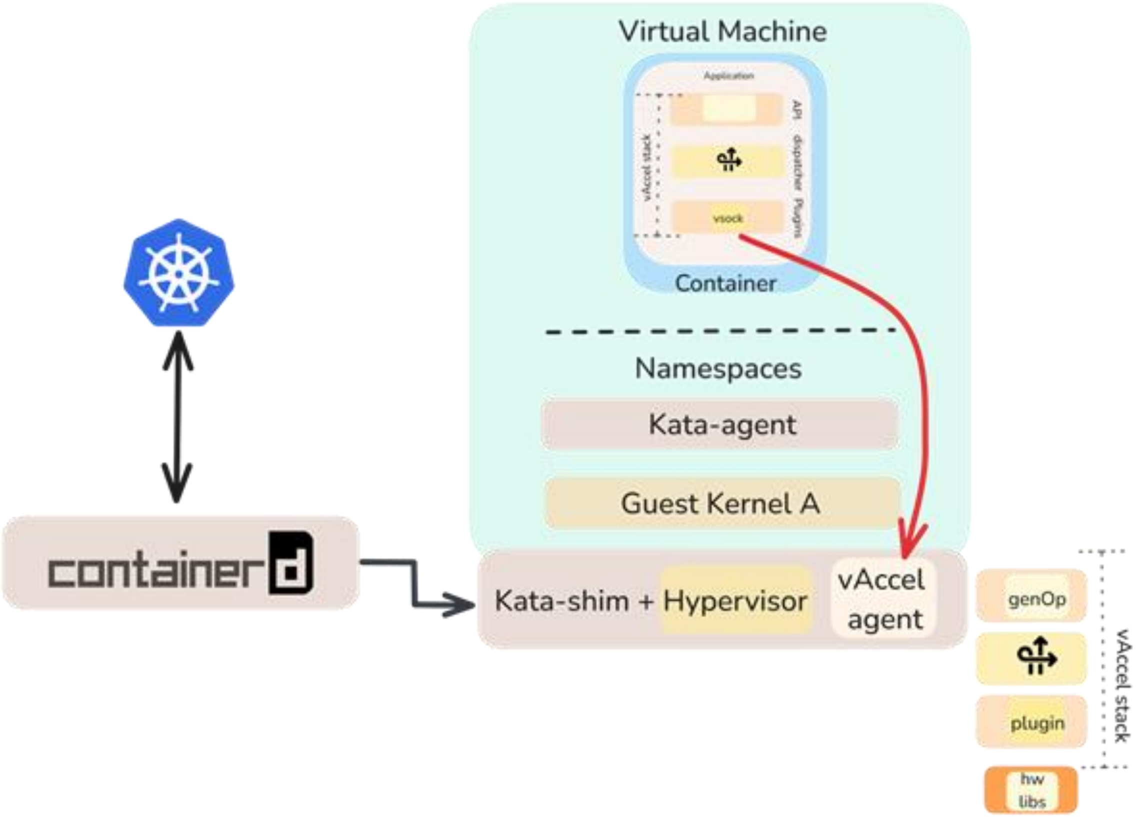
```
export VACCEL_BACKENDS=libtorch-rpc.so
export VACCEL_RPC_ADDRESS=vsock || tcp || unix ://<>:<>
./Application
```

```
export VACCEL_BACKENDS=libtorch-gpu.so:libtorch-cpu.so
./vAccel-agent -a vsock || tcp || unix ://<>:<>
```

The vAccel Framework

Cloud-native integration

- kata-containers++
- **Integrate** agent into the runtime



Roadmap & plan

nubificus/urunc

nubificus/bima

nubificus/bunny

blog.cloudkernels.net

- Enhance unikernel support
- Extend urunc support to lightweight apps (kernel+init)
  - Talk @ FOSDEM2025: Less overhead, strong isolation: Running containers in minimal specialized Linux VMs
- Extend `bima` to `bunny`:
  - Build tool to produce lightweight app packages [WiP
- Introduce WASM sandboxing
  - Talk @ FOSDEM2025: WASM meets unikernels: Secure and Efficient Cloud-Native Deployments
- Breakdown analysis of time spent in each phase of execution
- Explore confidential unikernels*

DESIRE6G

EMPYREAN

MLSysOps

This work is partially funded through Horizon Europe Research and Innovation actions, MLSysOps (GA: 101092912), DESIRE6G (GA: 101096466), and EMPYREAN (GA: 101136024)

Co-funded by the European Union

6GSNS

European Commission

HORIZON EUROPE

Thanks!

Contact info:

- Anastassios Nanos, ananos@nubificus.co.uk
- urunc team, urunc@nubificus.co.uk

Source code:

- GitHub org: https://github.com/nubificus
- urunc: nubificus/urunc
- bima: nubificus/bima
- kata: kata-containers/kata-containers

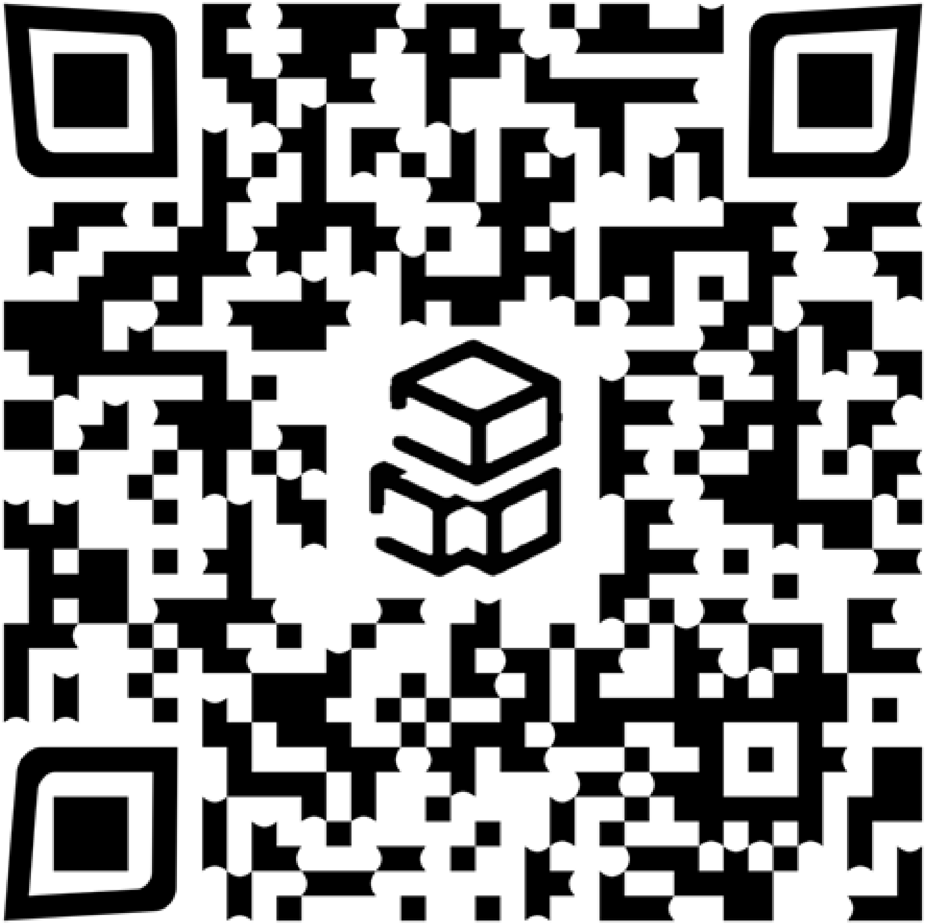
Cloud
Hypervisor

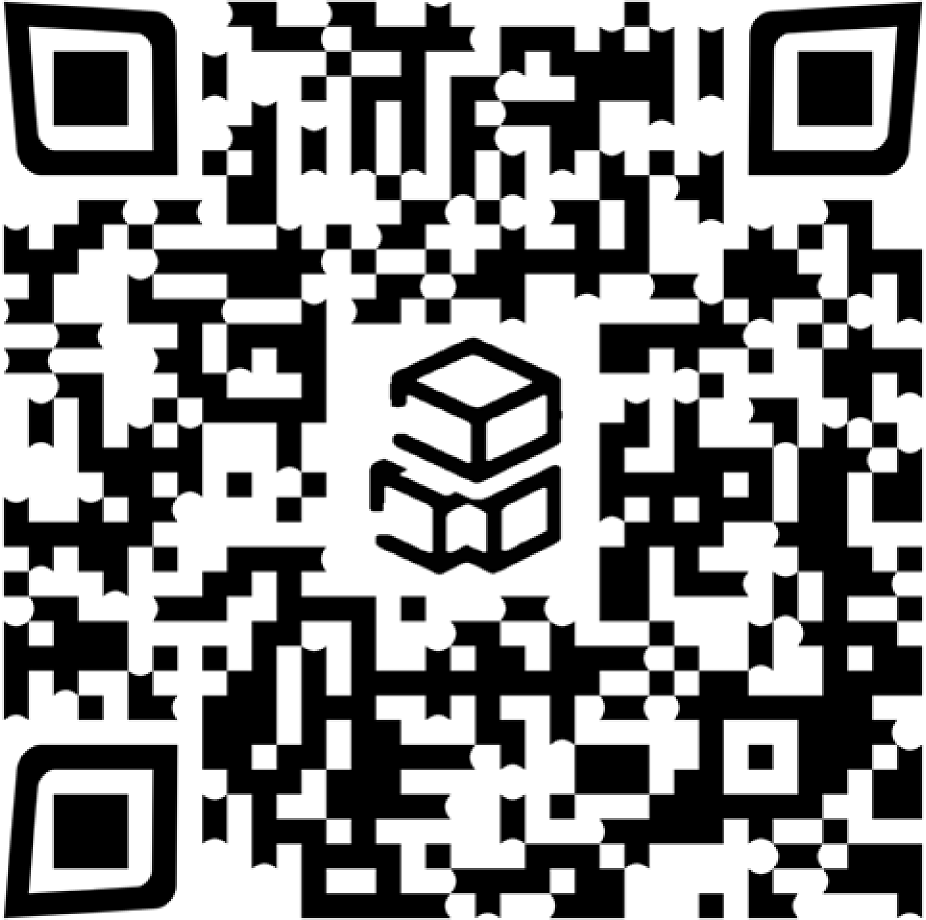
kata
containers

QEMU

kata
containers

RC
RUNC





QEMU