Revolutionizing Cloud Infrastructure: Unikraft and Unikraft Cloud - Unleashing the Power of Unikernels

Todor Todorov

Staff Engineer, Man Group UK Ltd.

Agenda

- What is a Unikernel?
- Introduction to Unikraft
- Introduction to Unikraft Cloud
- Demo
- Q&A

Agenda

- What is a Unikernel?

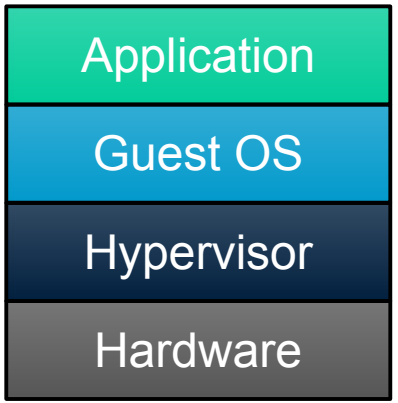
What is a Unikernel?

- Specialized library operating system
- Single process*
- Multi-threading
- Single user
- Single address space

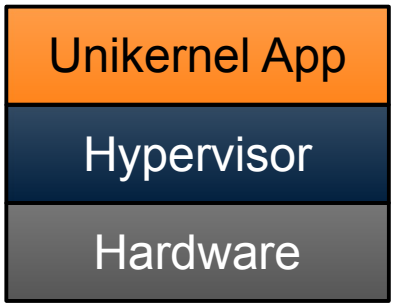
Benefits

- Reduced resource consumption
- Enhanced security
- Faster boot times
- Improved performance
- Simplified deployment

Traditional VM vs Unikernel



Traditional
Virtual Machine



Unikernel

Notable Projects

- http://unikernel.org/projects/
- OSv https://osv.io/ (2012-12-01)
- MirageOS https://mirage.io/ (2013-02-02)
- IncludeOS https://www.includeos.org/ (2014-07-07)
- Hermit OS http://hermit-os.org (2015-05-09)
- Unikraft https://unikraft.org/ (2017-11-29)

Agenda

- Introduction to Unikraft

Introduction to Unikraft

- Single address space
- Fully modular system
- Single protection level
- Static linking
- POSIX support
- Platform abstraction

KraftKit

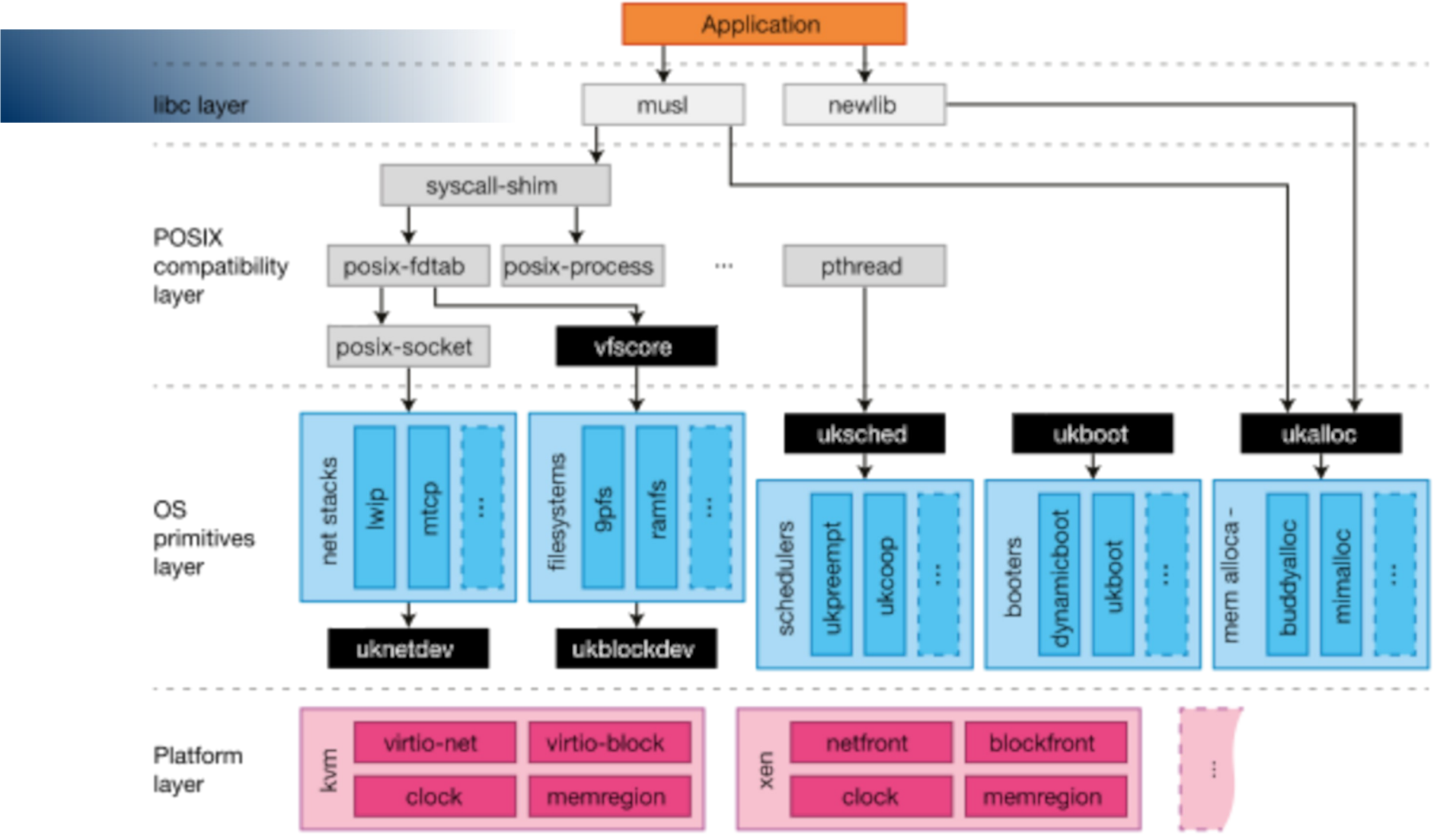
- Building, packaging, running locally and deploying unikernels to Unikraft Cloud
- Resembles the docker command
- Supports compose files
- Manages resources on Unikraft Cloud
- Can read instructions from Kraftfile
- Can set KConfig options

Kraftfile

```
spec: v0.6

name: nginx

rootfs: ./Dockerfile

cmd: ["/usr/bin/nginx"]

template:
  source: https://github.com/unikraft/app-elfloader.git
  version: staging

unikraft:
  source: https://github.com/unikraft/unikraft.git
  version: staging
  kconfig:
    CONFIG_LIBPOSIX_ENVIRON_ENVP0: "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
    CONFIG_LIBPOSIX_ENVIRON_ENVP1: "LD_LIBRARY_PATH=/usr/local/lib:/usr/lib:/lib"
    CONFIG_LIBPOSIX_ENVIRON_ENVP2: "HOME=/"
    CONFIG_LIBPOSIX_ENVIRON: 'y'

libraries:
  lwip:
    source: https://github.com/unikraft/lib-lwip.git
    version: staging
    kconfig:
      CONFIG_LWIP_LOOPIF: 'y'
      CONFIG_LWIP_UKNETDEV: 'y'

targets:
- fc/x86_64
- qemu/x86_64
```

Architecture

- Micro-libraries – implement core Unikraft APIs
- Build system – Kconfig-based menu


Application
libc layer
musl
newlib
syscall-shim
POSIX compatibility layer
posix-fdtab
posix-process
...
pthread
posix-socket
vfscore
OS primitives layer
net stacks
lwip
mtcp
...
filesystems
9pfs
ramfs
...
uksched
ukboot
ukalloc
schedulers
ukpreempt
ukcoop
...
booters
dynamicboot
ukboot
...
mem alloca -
buddyalloc
mimalloc
...
uknetdev
ukblockdev
Platform layer
kvm
virtio-net
virtio-block
clock
memregion
xen
netfront
blockfront
clock
memregion
...

Build System

- Makefile – specifies locations of the Unikraft repository and external libraries; not needed for porting external libraries
- Makefile.uk – specifies sources, include paths, flags, and application-specific targets
- Config.uk – populates Unikraft's menu with application-specific options
- exportsyms.uk – lists symbols to be exported
- extra.ld – optional amendment to the main linker script

Build Process

1. Configuring the Unikraft unikernel
2. Fetching and preparing the source code
3. Compiling the libraries and the core Unikraft code
4. Linking the final unikernel image

Debugging

- Strace-like messages for binary system calls
- Global or per compilation unit debug messages
- Connecting GDB to QEMU’s GDB server

Agenda

- Introduction to Unikraft Cloud

Introduction to Unikraft Cloud

- Based on unikernels
- Milliseconds boot time
- Auto-scale and scale to zero
- No traffic metering
- Based on Equinix
- Leveraging Firecracker as a Virtual Machine Monitor

Concepts and Features

- Instance – running virtual machine
- Service – exposing the apps to the Internet
- All instances have a private IP and DNS by default
- Restart policy – never, always or on-failure
- Instances are created from images pulled from an OCI compatible registry
- Support of volumes, custom domains and rolling updates

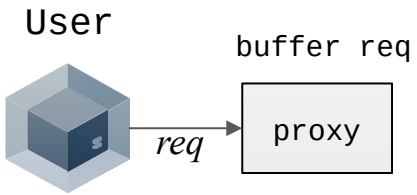
Integrations

- RESTful API
- GitHub Actions for building and deploying services
- Crossplane provider for deploying and managing unikernel instances as Kubernetes resources
- Terraform provider for deploying and managing unikernel instances
- metrics API exporting metrics in JSON or Prometheus format

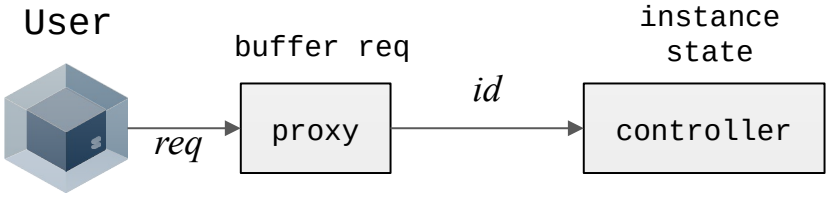
Millisecond Scale to Zero

User

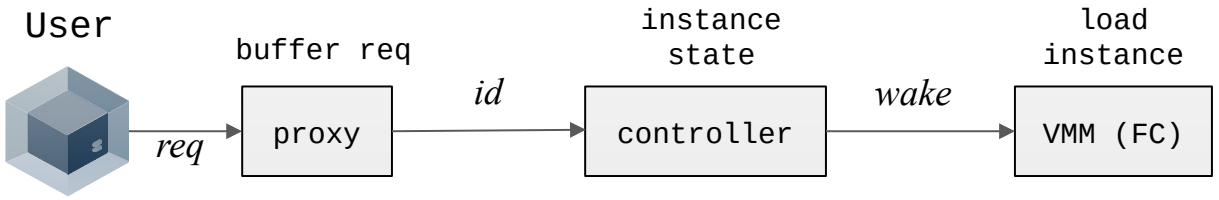
Millisecond Scale to Zero



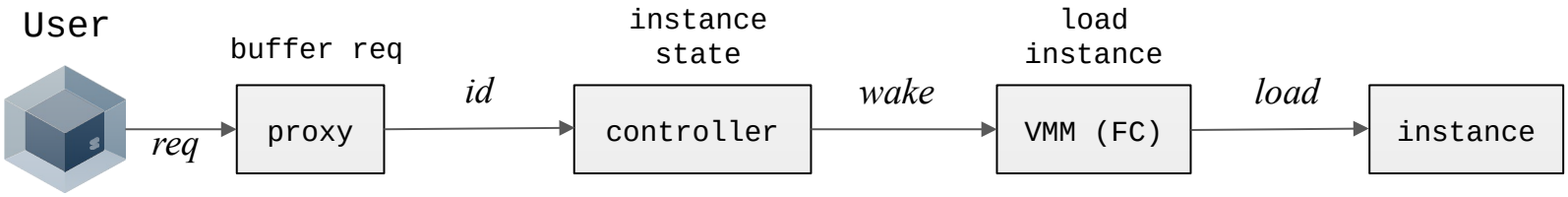
Millisecond Scale to Zero



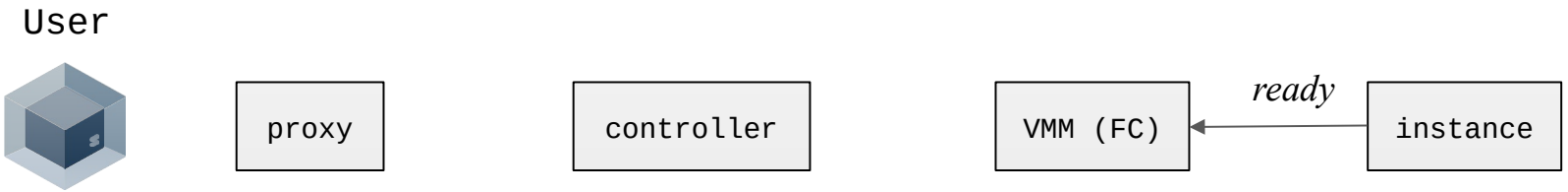
Millisecond Scale to Zero



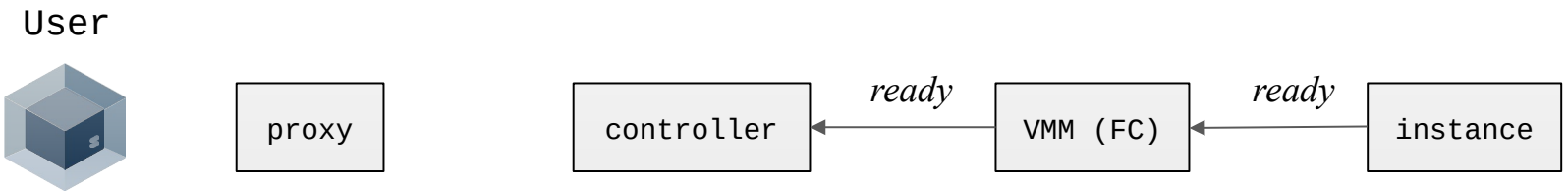
Millisecond Scale to Zero



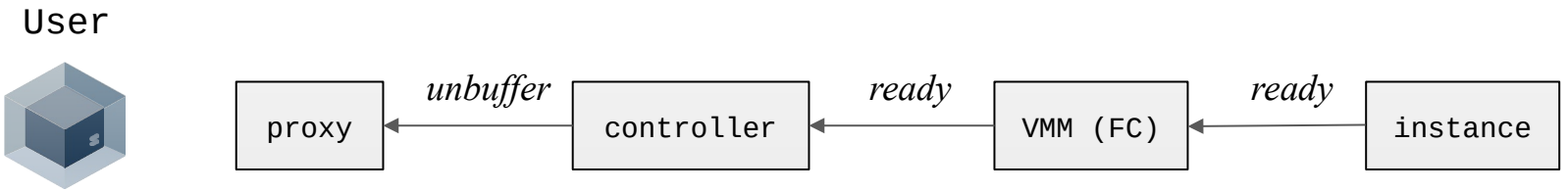
Millisecond Scale to Zero



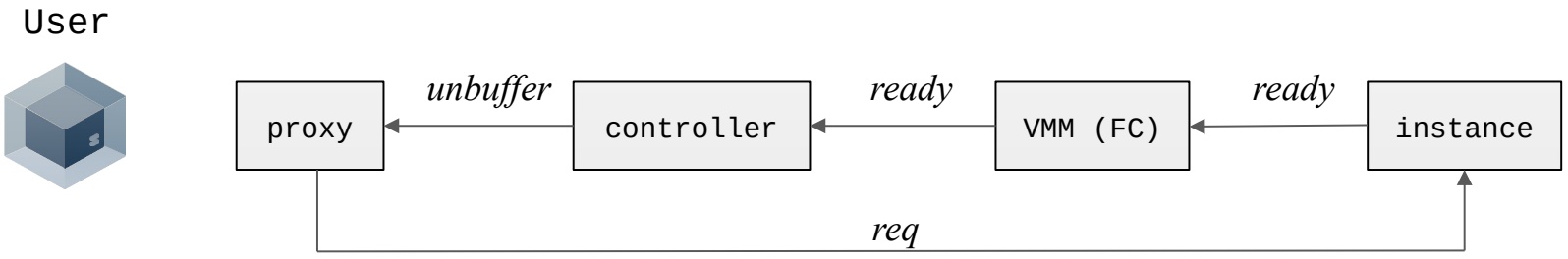
Millisecond Scale to Zero



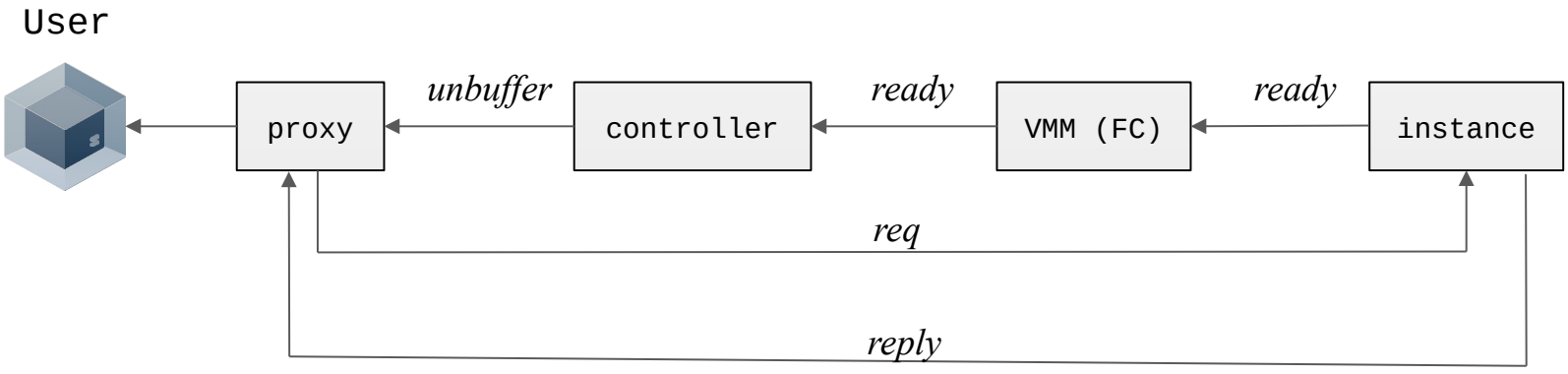
Millisecond Scale to Zero



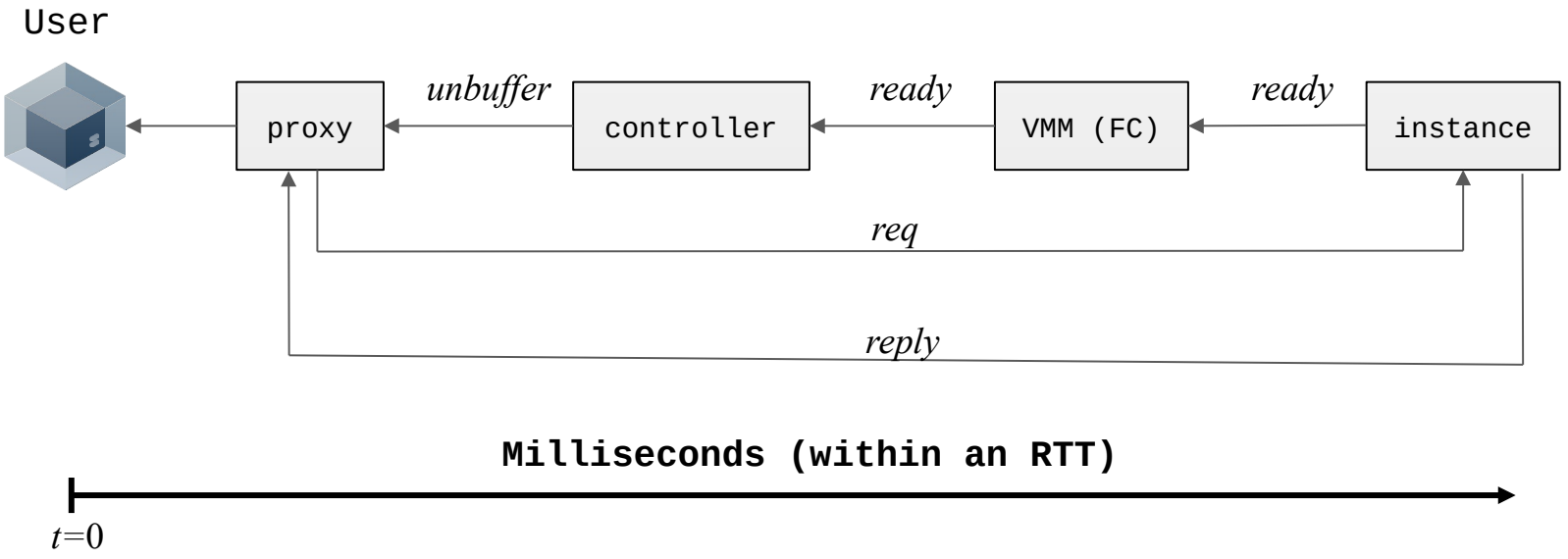
Millisecond Scale to Zero



Millisecond Scale to Zero

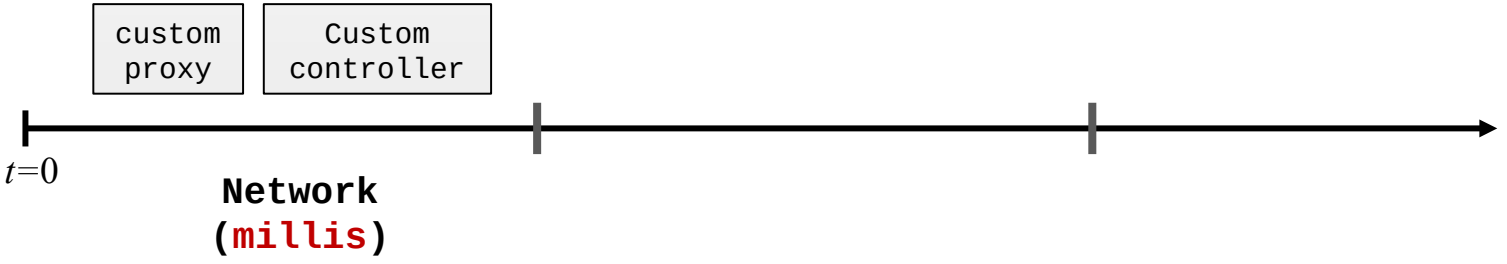


Millisecond Scale to Zero



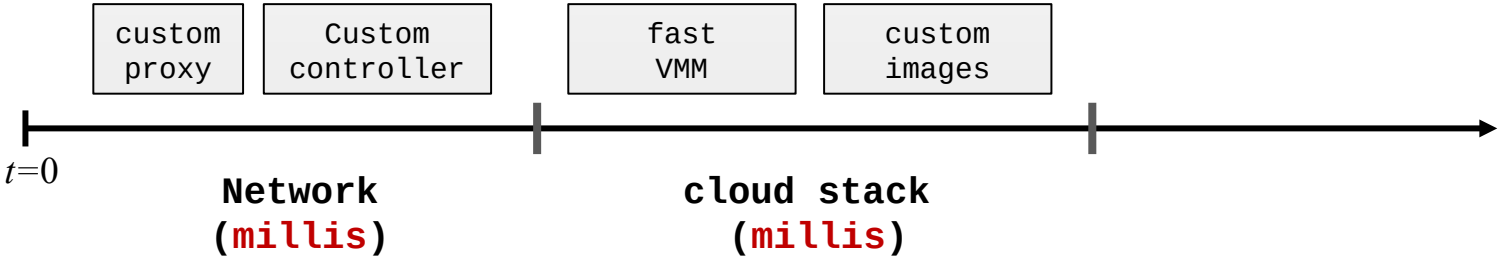
Why it is Fast

- Network: custom load balancer = millis response time



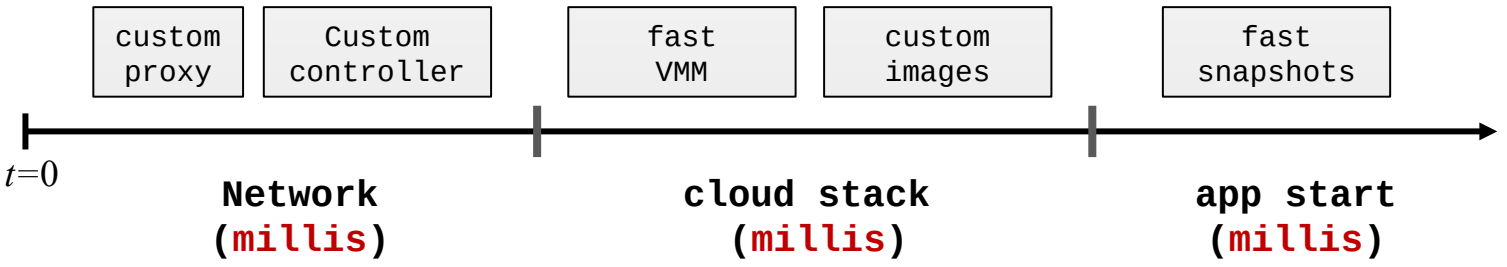
Why it is Fast

- Network: custom load balancer = millis response time
- Cloud stack: customer controller and images = millis response times



Why it is Fast

- Network: custom load balancer = millis response time
- Cloud stack: customer controller and images = millis response times
- Application start time: snapshotting = millis response time



DEMO

Q&A

Thank you