64-bit ARM Unikernels on uKVM

ARM

Wei Chen <Wei.Chen@arm.com>

Beijing / LC3 China 2017
2017-06-20

©ARM 2017

Thanks to

- Dan Williams (IBM), Martin Lucina (Docker), Anil Madhavapeddy (Docker) and other Solo5 contributors who give me lots of help in community.

- All my ARM colleagues who are co-working with me to implement AArch64 uKVM monitor and bring up guest.

©ARM 2017

Agenda

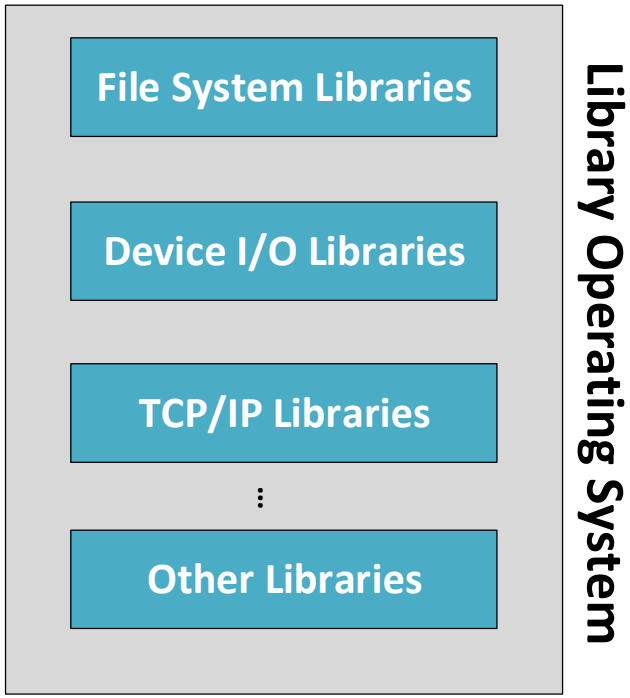
- Unikernel introduction
- Current workload issues and solutions on cloud
- uKVM and ARM work
- Demo
- What's next

©ARM 2017

What are unikernels

The unikernel community, Unikernel.org, defines it as follows:

Unikernels are specialized, single-address-space machine images constructed by using *library operating systems*.

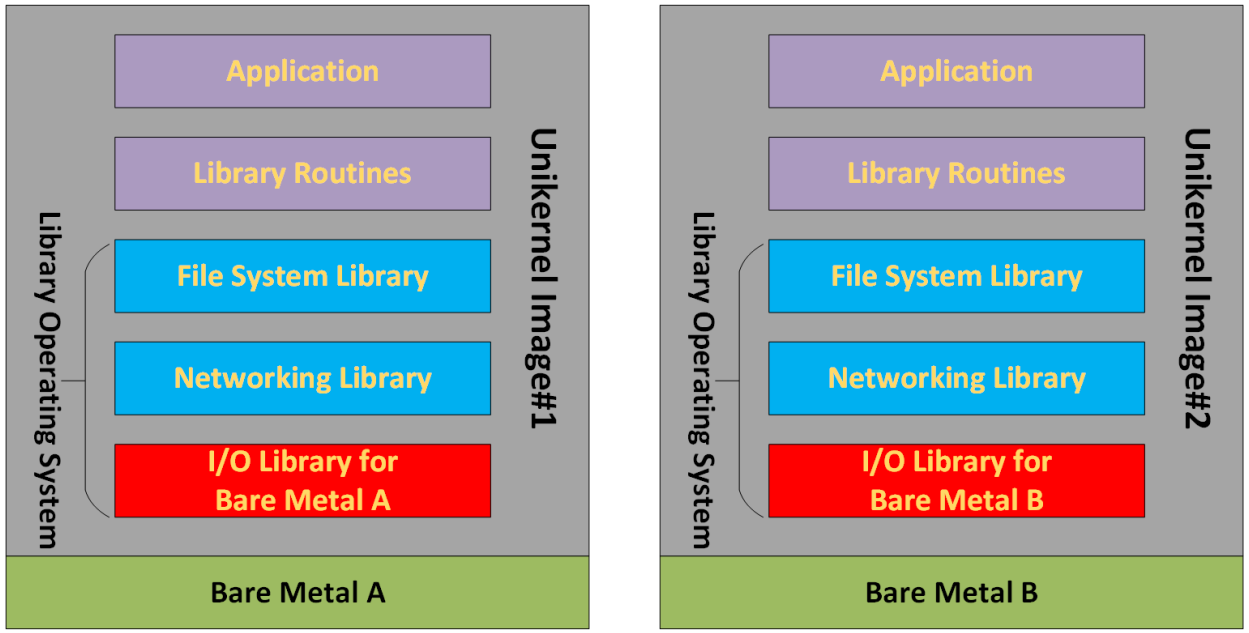


A special collection of libraries that provides needed operating system functions in a compliable format.

©ARM 2017

Unikernels run on bare metal

Unikernels can be designed to run on bare metal directly.



Two big drawbacks:

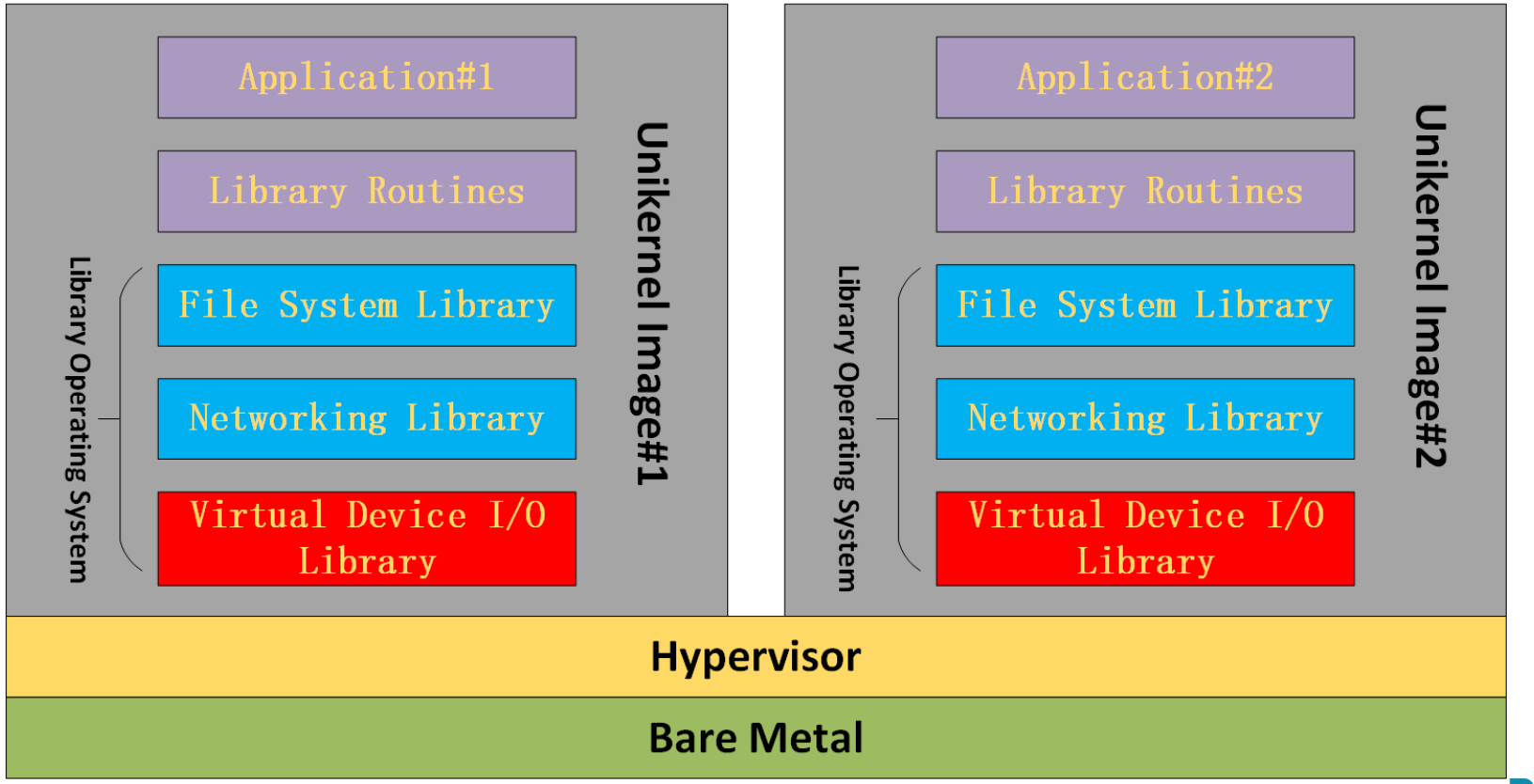
- Resource isolations for multiple unikernels.
- Variety of different devices.

©ARM 2017

Unikernels run on hypervisors

Fortunately, modern hypervisors provide virtual machines with:

- Consistent set of virtual devices.
- Strong context isolation.

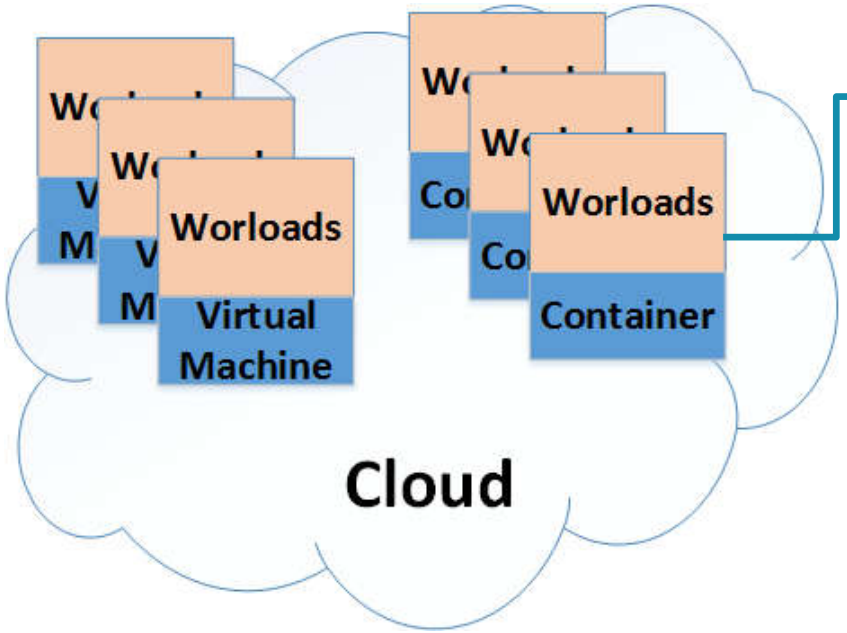


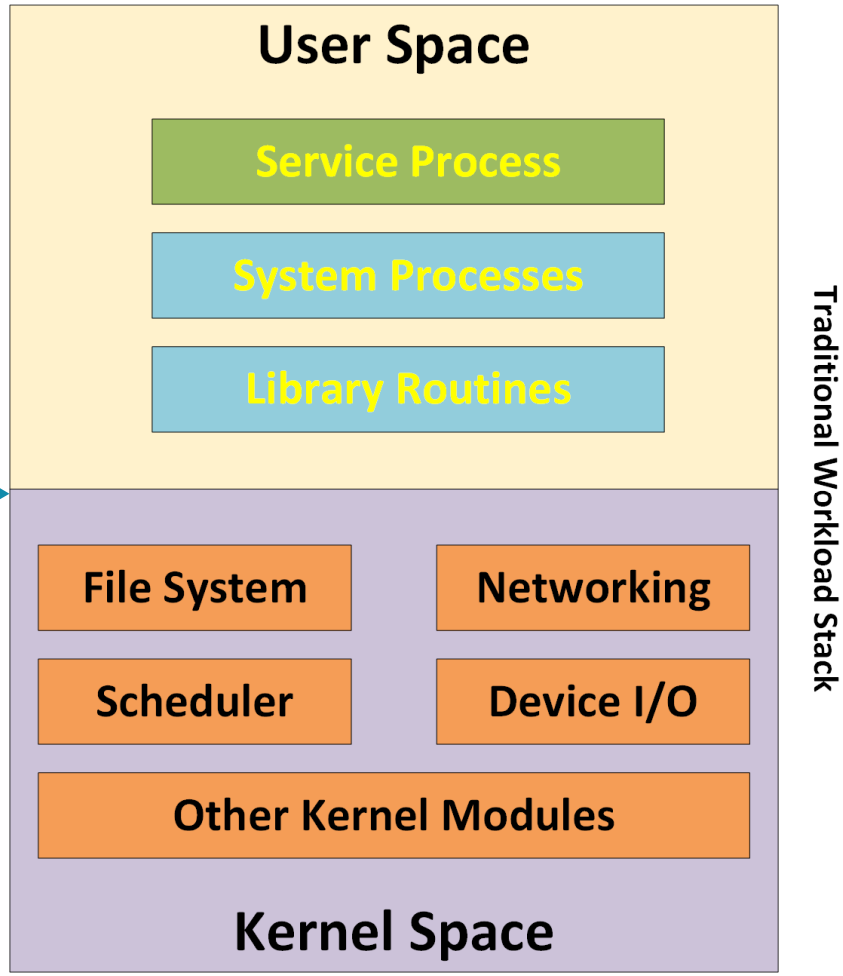
©ARM 2017

Why we need Unikernels?

To address issues of traditional workloads on Cloud

- Slower initialization
- More resources used
- More opportunities to exploit

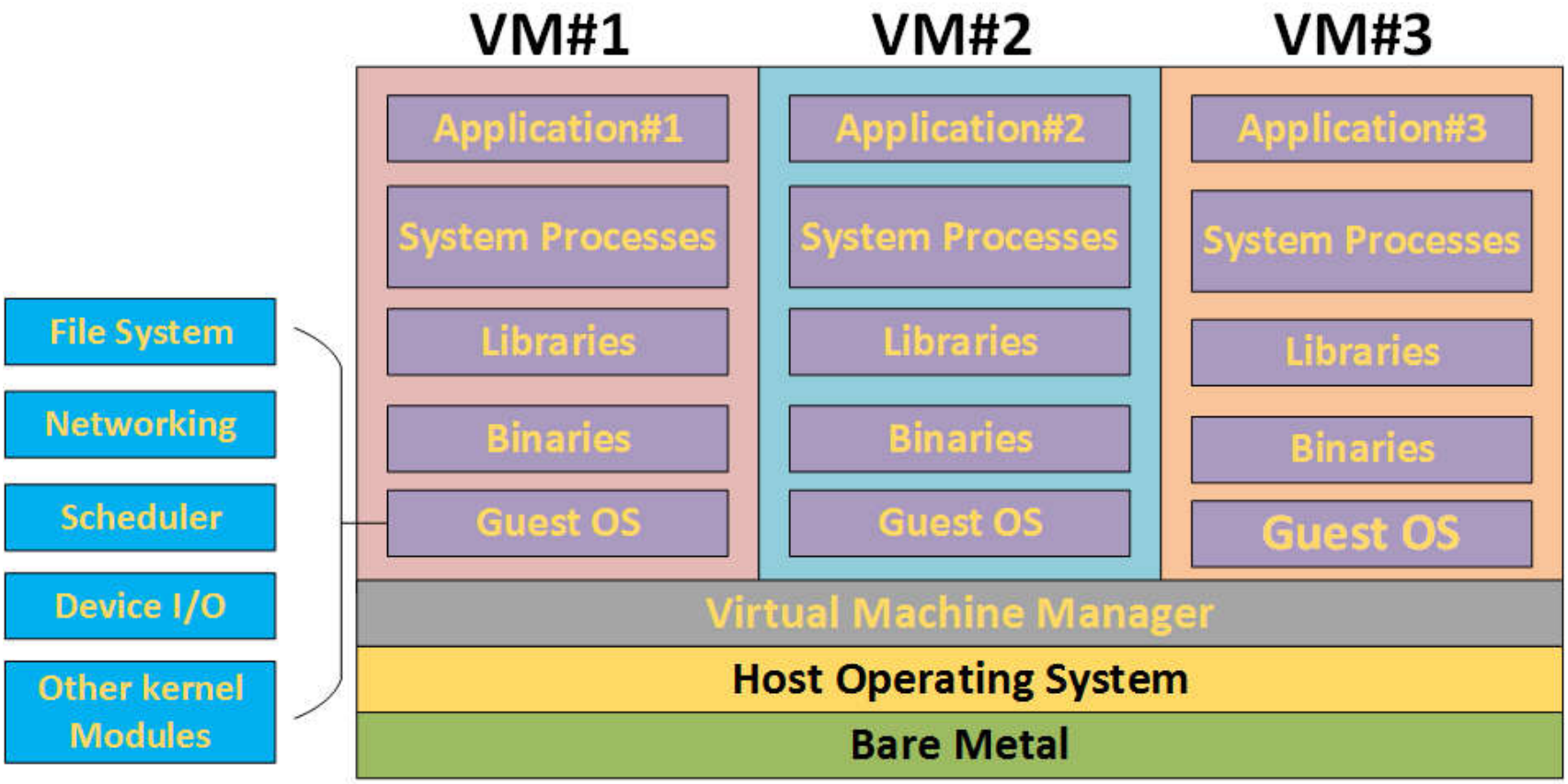




©ARM 2017

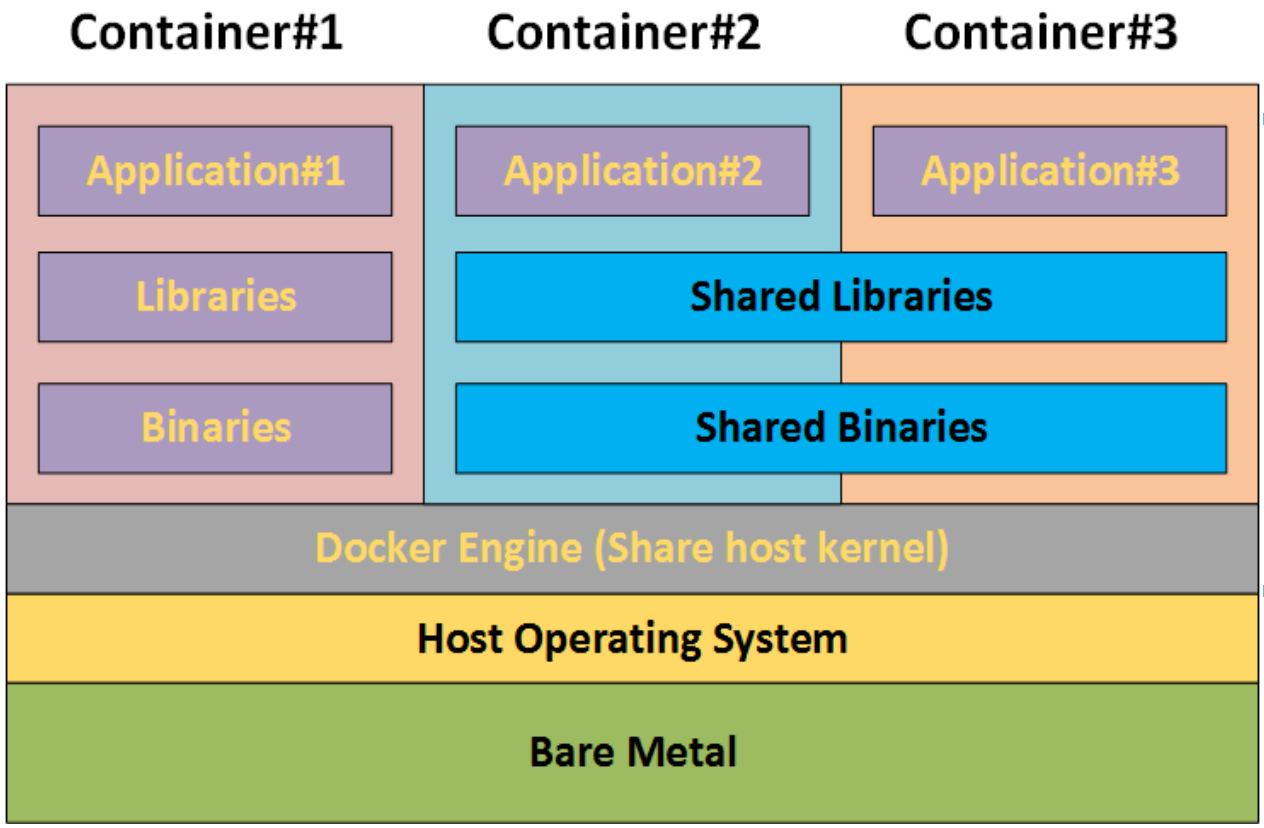
Workloads with Virtual Machine

Move the workloads into the virtual machine:



©ARM 2017

Can Container help?

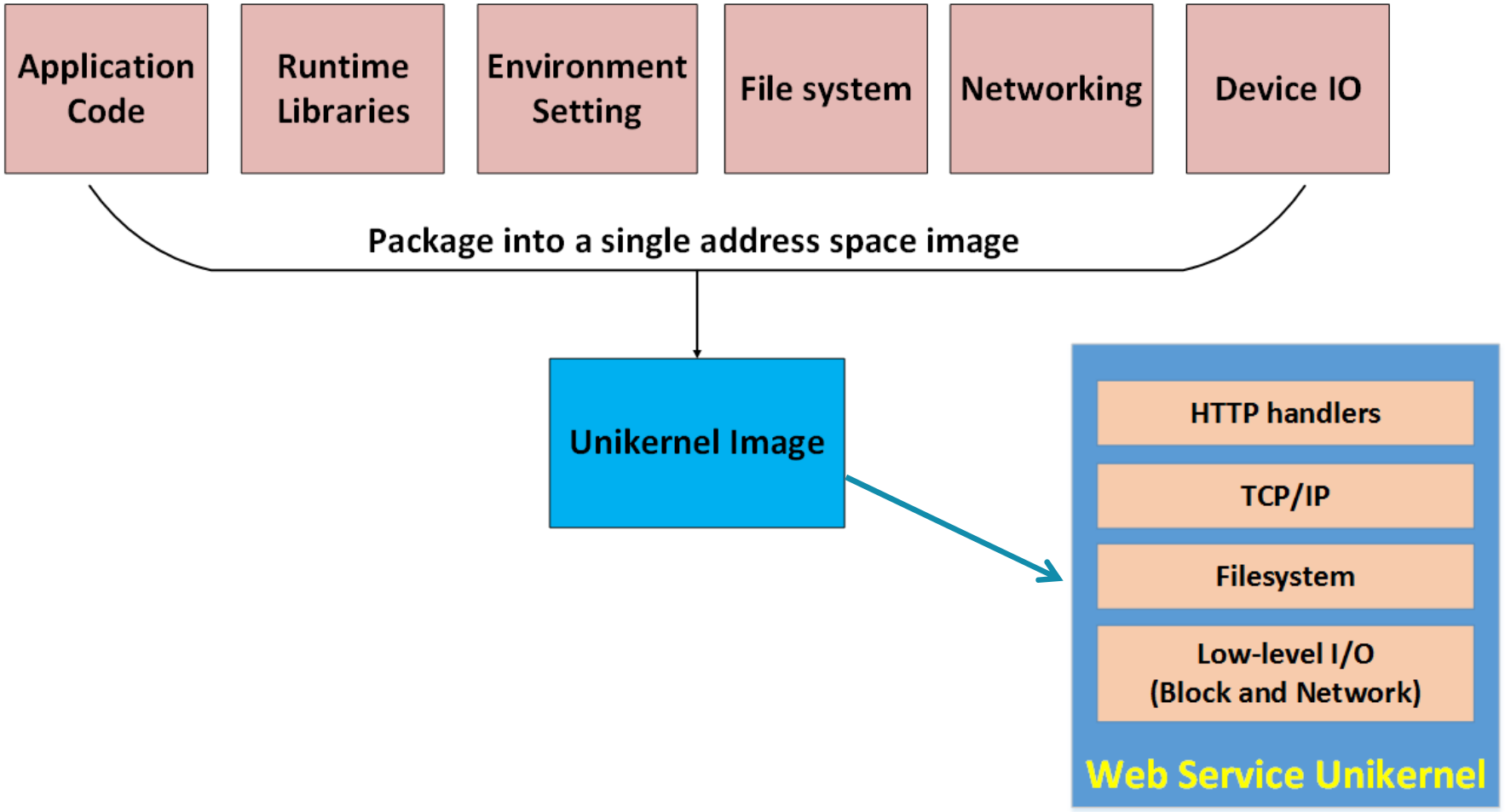


- Pros
  - Lightweight footprint
  - Efficient resource utilization
  - Faster startup
  - High density
  - Fast deployment
- Cons
  - Less secure

©ARM 2017

Are unikernels better solution?

Package only needed modules into an image:

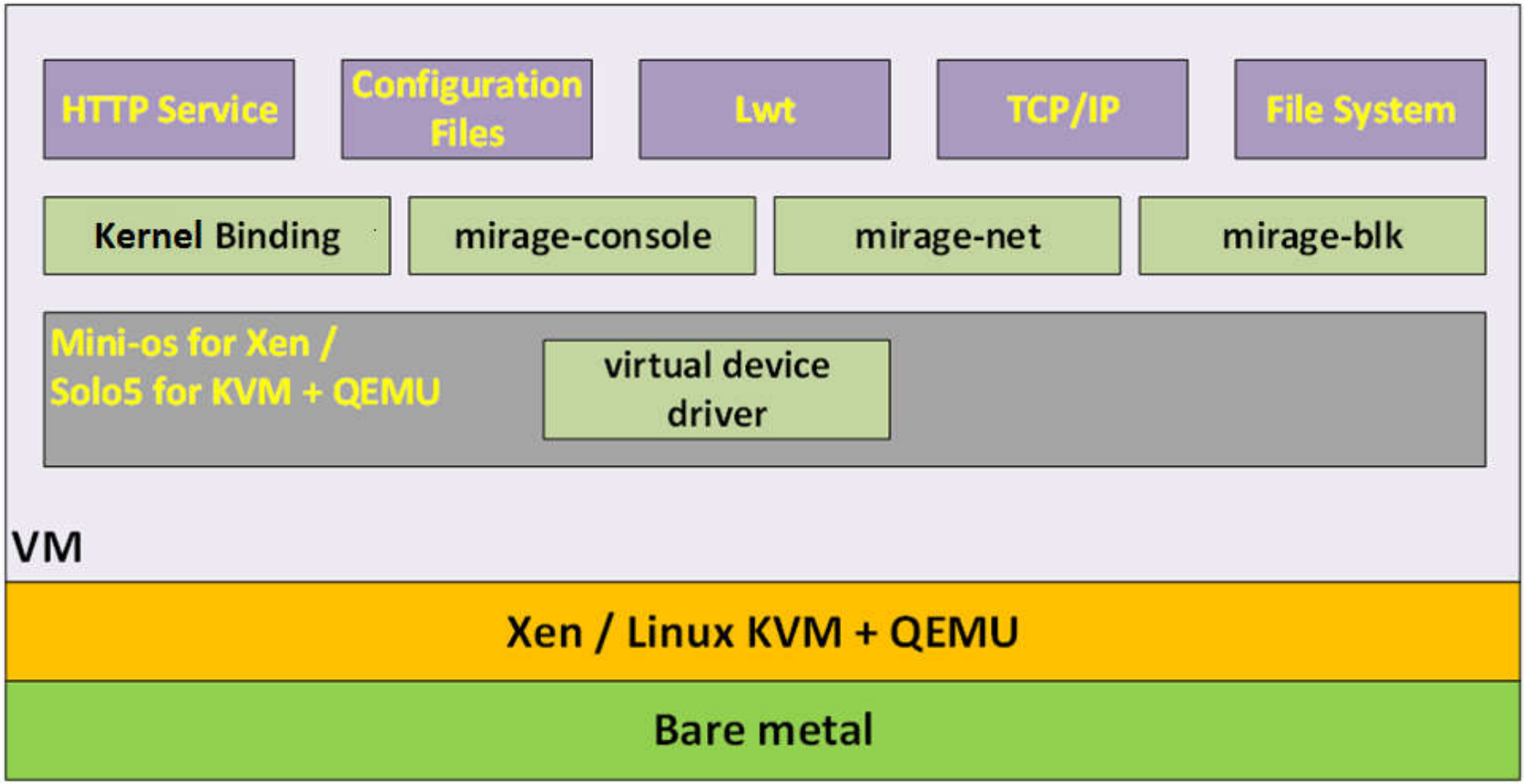


 ©ARM 2017

ARM

MirageOS as an example

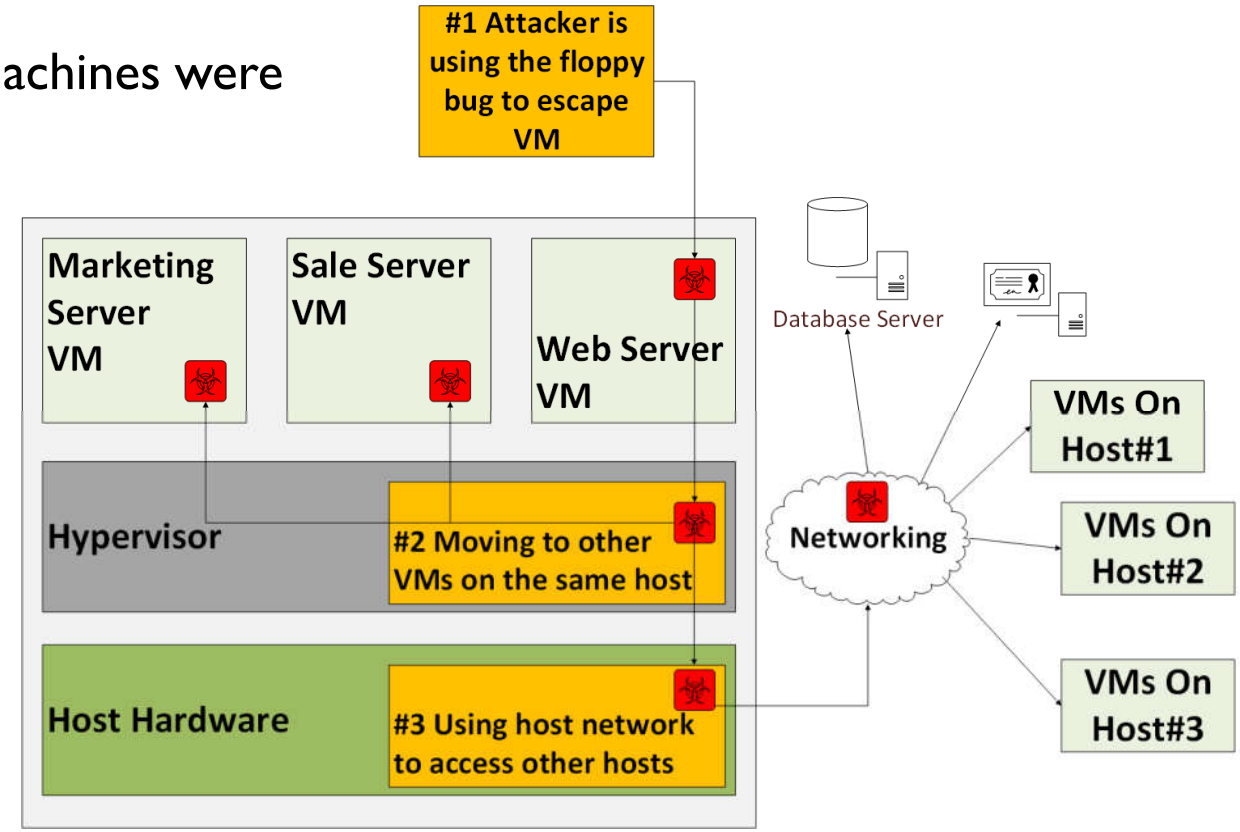
MirageOS unikernel can run on Xen or Linux KVM/QEMU as a guest.



©ARM 2017

VENOM vulnerability

- Origin
  - Bug in virtual floppy emulation.
- Range
  - Millions of virtual machines were potentially at risk.

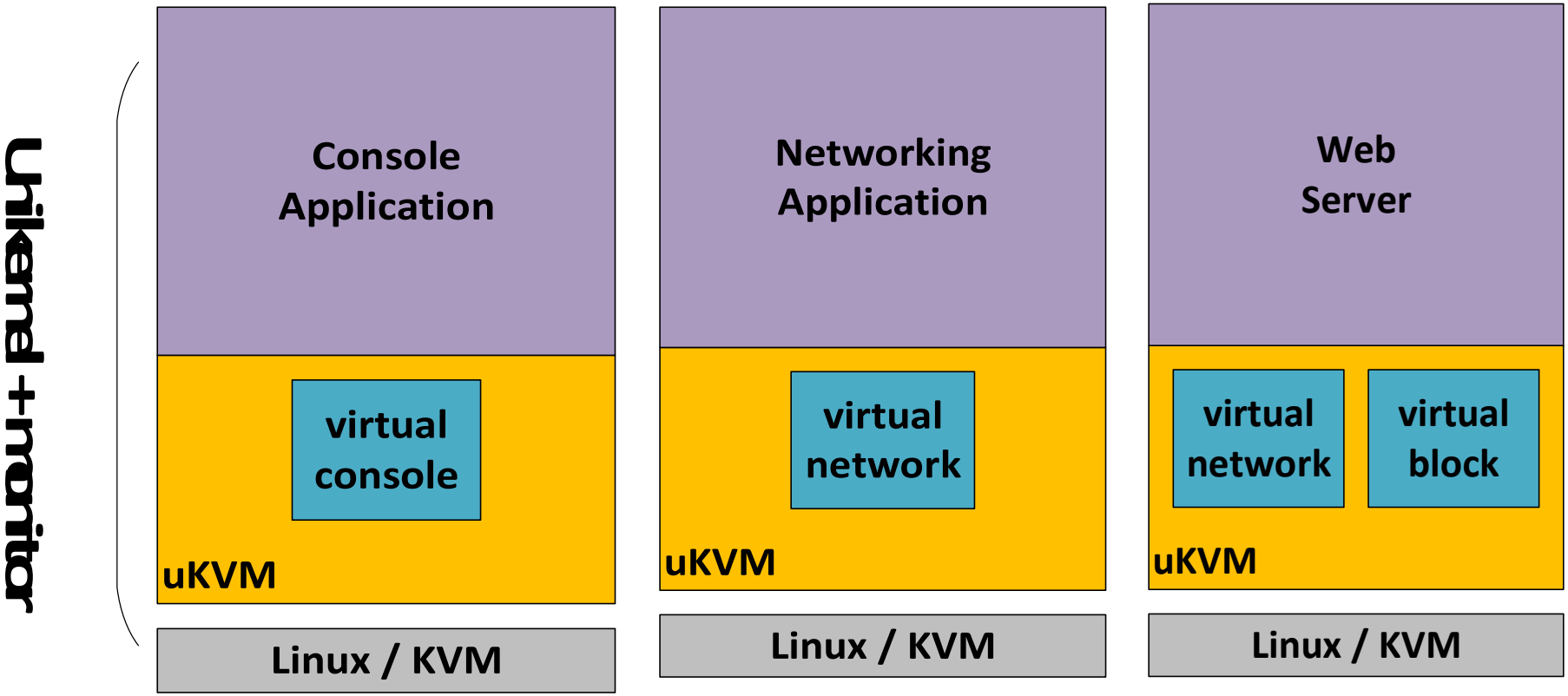


 ©ARM 2017

ARM

uKVM is a specialized unikernel monitor

Package hypervisor interfaces and emulations that only applications required:



 ©ARM 2017

ARM

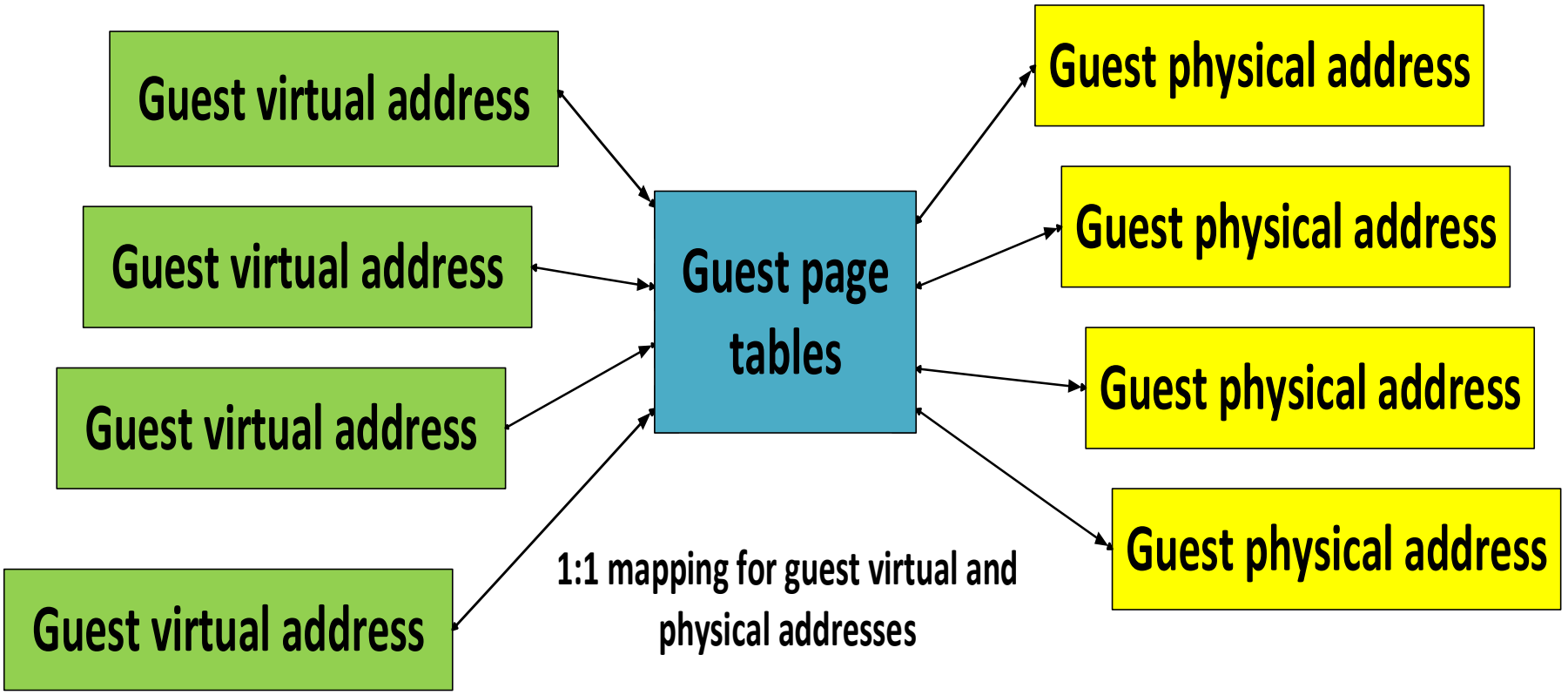
uKVM on AArch64

- We have started to port uKVM on AArch64 at the beginning of this year.
- Currently, we have the following working:
  - Setup guest CPU
  - Setup guest memory
  - Setup guest timer
  - Setup guest MMU
  - https://github.com/Weichen81/ukvm-solo5-arm64
- And we are working with upstream to get support merged at:
  - https://github.com/Solo5/solo5

©ARM 2017

Guest page tables on AArch64

Need to enable MMU for guest to share data with host on AArch64.



 ©ARM 2017

ARM

Demo

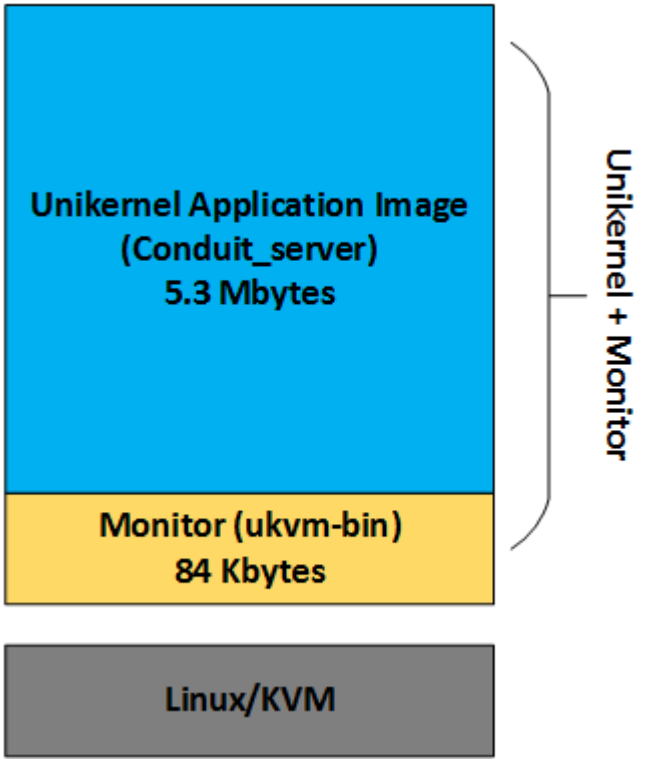
- To demonstrate:
  - Http server binary size, boot time and memory usage.
  - How many http servers can run on this host at the same time.

- Hardware Configuration:
  - 8 Cortex-A53 2Ghz CPU
  - 16 GB memory
  - mirage-solo5-ukvm AArch64 Branch:
    - git checkout –b arm64 https://github.com/Weichen81/ukvm-solo5-arm64
  - Testing tag:
    - demo_for_oss_2017

©ARM 2017

Footprint and boot time

- Footprint:
  - ukvm-bin, 84Kbytes
  - Conduit_server.ukvm, 5.3Mbytes
- Http Server boot time:
  - Launch to uKVM main entry: ~1ms
  - uKVM main entry to conduit_server print "SOLO5": ~50ms




```
          |     __|
  _|  _ \  |  _ \ _ \
\__ \(   | |(   | ) |
____/\___/ _|\___/___/
Solo5: Memory map: 16 MB addressable:
Solo5:     unused @ (0x0 - 0xfffff)
Solo5:       text @ (0x100000 - 0x306fff)
Solo5:     rodata @ (0x307000 - 0x35ffff)
Solo5:       data @ (0x360000 - 0x471fff)
Solo5:       heap >= 0x472000 < stack < 0x1000000
Solo5: new bindings
STUB: getenv() called
STUB: open() called
STUB: getpid() called
STUB: getppid() called
2017-05-16 09:12:03 -00:00: INF [netif] Plugging into 0 with mac a2:1e:82:57:38:65
2017-05-16 09:12:03 -00:00: INF [ethif] Connected Ethernet interface a2:1e:82:57:38:65
2017-05-16 09:12:03 -00:00: INF [arpv4] Connected arpv4 device on a2:1e:82:57:38:65
2017-05-16 09:12:03 -00:00: INF [udp] UDP interface connected on 10.0.0.2
2017-05-16 09:12:03 -00:00: INF [tcpip-stack-direct] stack assembled: mac=a2:1e:82:57:38:65,ip=10.0.0.2
```


©ARM 2017

Multiple instances

256 Conduit Servers:

- CPU usage: 100%

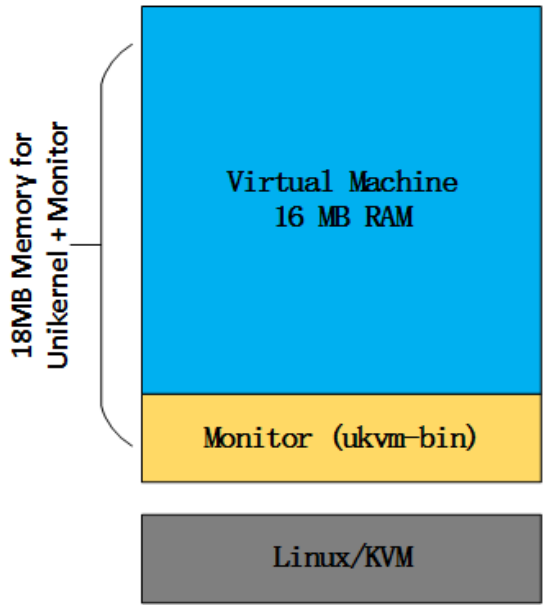
```
top - 08:09:36 up  3:19,  3 users,  load average: 196.78, 74.13, 27.56
Tasks: 458 total, 257 running, 201 sleeping,   0 stopped,   0 zombie
%Cpu0  : 99.0 us,  1.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu1  :100.0 us,  0.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu2  : 99.0 us,  1.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu3  : 99.5 us,  0.5 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu4  : 99.5 us,  0.5 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu5  : 99.5 us,  0.5 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu6  : 99.5 us,  0.5 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu7  :100.0 us,  0.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 16361072 total, 13976696 free,   207636 used,  2176740 buff/cache
KiB Swap: 19730428 total, 19730428 free,        0 used. 13983140 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
27498 weic      20   0   18144   8180   8100 R   3.9  0.0   0:02.38 ukvm-bin
27500 weic      20   0   18144   8112   8028 R   3.9  0.0   0:02.01 ukvm-bin
27502 weic      20   0   18144   7892   7808 R   3.9  0.0   0:02.39 ukvm-bin
27509 weic      20   0   18144   8152   8072 R   3.9  0.0   0:02.36 ukvm-bin
27521 weic      20   0   18144   8156   8072 R   3.9  0.0   0:02.38 ukvm-bin
27534 weic      20   0   18144   8084   8000 R   3.9  0.0   0:02.36 ukvm-bin
27538 weic      20   0   18144   8176   8092 R   3.9  0.0   0:02.35 ukvm-bin
27552 weic      20   0   18144   8048   7964 R   3.9  0.0   0:02.34 ukvm-bin
27554 weic      20   0   18144   8156   8072 R   3.9  0.0   0:02.35 ukvm-bin
27573 weic      20   0   18144   8072   7988 R   3.9  0.0   0:02.36 ukvm-bin
27597 weic      20   0   18144   8112   8028 R   3.9  0.0   0:02.36 ukvm-bin
27606 weic      20   0   18144   8176   8092 R   3.9  0.0   0:02.27 ukvm-bin
27607 weic      20   0   18144   8108   8024 R   3.9  0.0   0:02.11 ukvm-bin
27620 weic      20   0   18144   8180   8096 R   3.9  0.0   0:02.11 ukvm-bin
27625 weic      20   0   18144   8024   7940 R   3.9  0.0   0:02.35 ukvm-bin
27669 weic      20   0   18144   8040   7956 R   3.9  0.0   0:02.27 ukvm-bin
27672 weic      20   0   18144   8048   7964 R   3.9  0.0   0:02.28 ukvm-bin
27684 weic      20   0   18144   8112   8028 R   3.9  0.0   0:02.26 ukvm-bin
27481 weic      20   0   18144   8128   8044 R   3.4  0.0   0:02.13 ukvm-bin
27484 weic      20   0   18144   7776   7692 R   3.4  0.0   0:02.14 ukvm-bin
```

- Memory usage: ~3GB

```
              total        used        free      shared  buff/cache   available
Mem:           15G        202M         13G        1.7G        2.1G         13G
Swap:          18G          0B         18G
```

- Single instance memory layout:



 ©ARM 2017

ARM

Work still needs to be done for AArch64

- Complete the upstream work.
- Add multi-platform supports, currently we only support Linux. If possible, we want to support other platforms like FreeBSD/MacOS.
- Add the VIRTIO support to increase the I/O performance.
- Verify and improve the compatibility of MirageOS libraries on AArch64.

©ARM 2017

Summary

- Unikernels on uKVM is an approach to make workloads to be smaller, faster and have less opportunities to exploit.

- What's next?
  - Running unikernels inside the container.

©ARM 2017

Question?

©ARM 2017

ARM

The trademarks featured in this presentation are registered and/or unregistered trademarks of ARM Limited (or its subsidiaries) in the US and/or elsewhere. All rights reserved. All other marks featured may be trademarks of their respective owners.

Copyright © 2017 ARM Limited

©ARM 2017