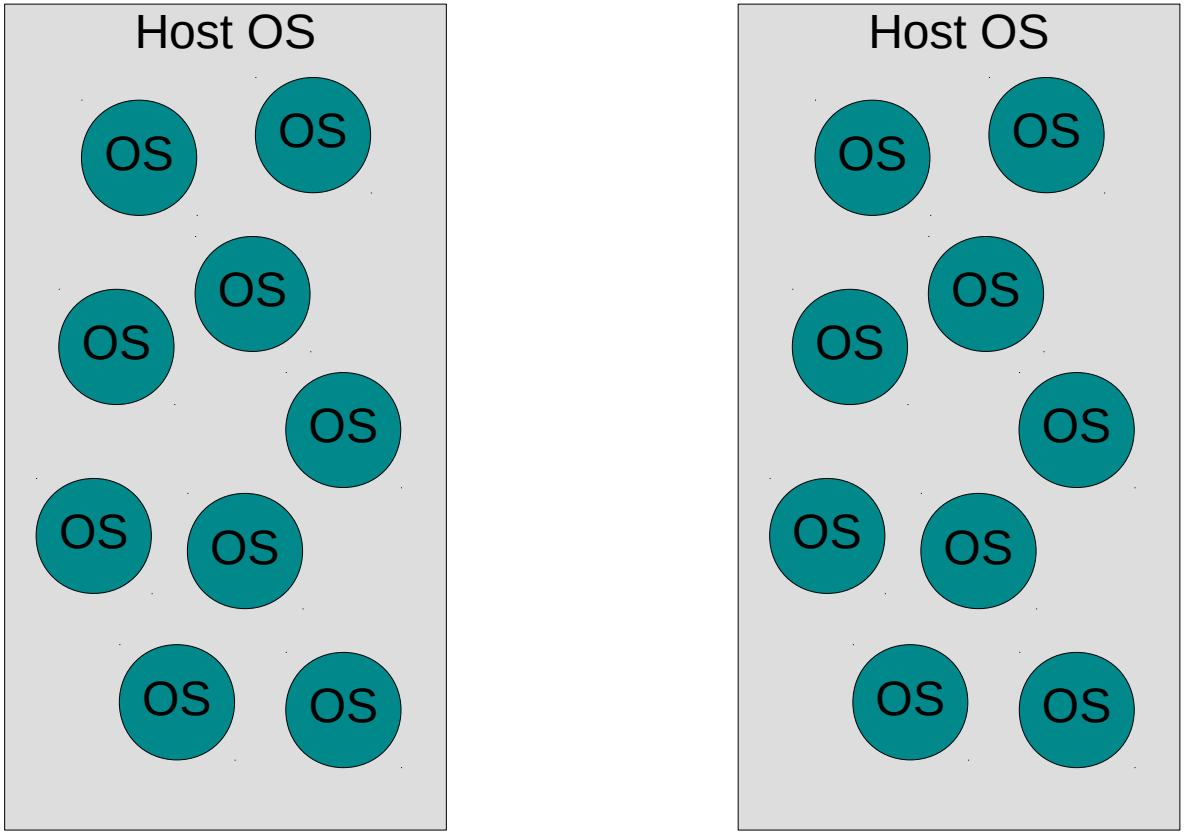
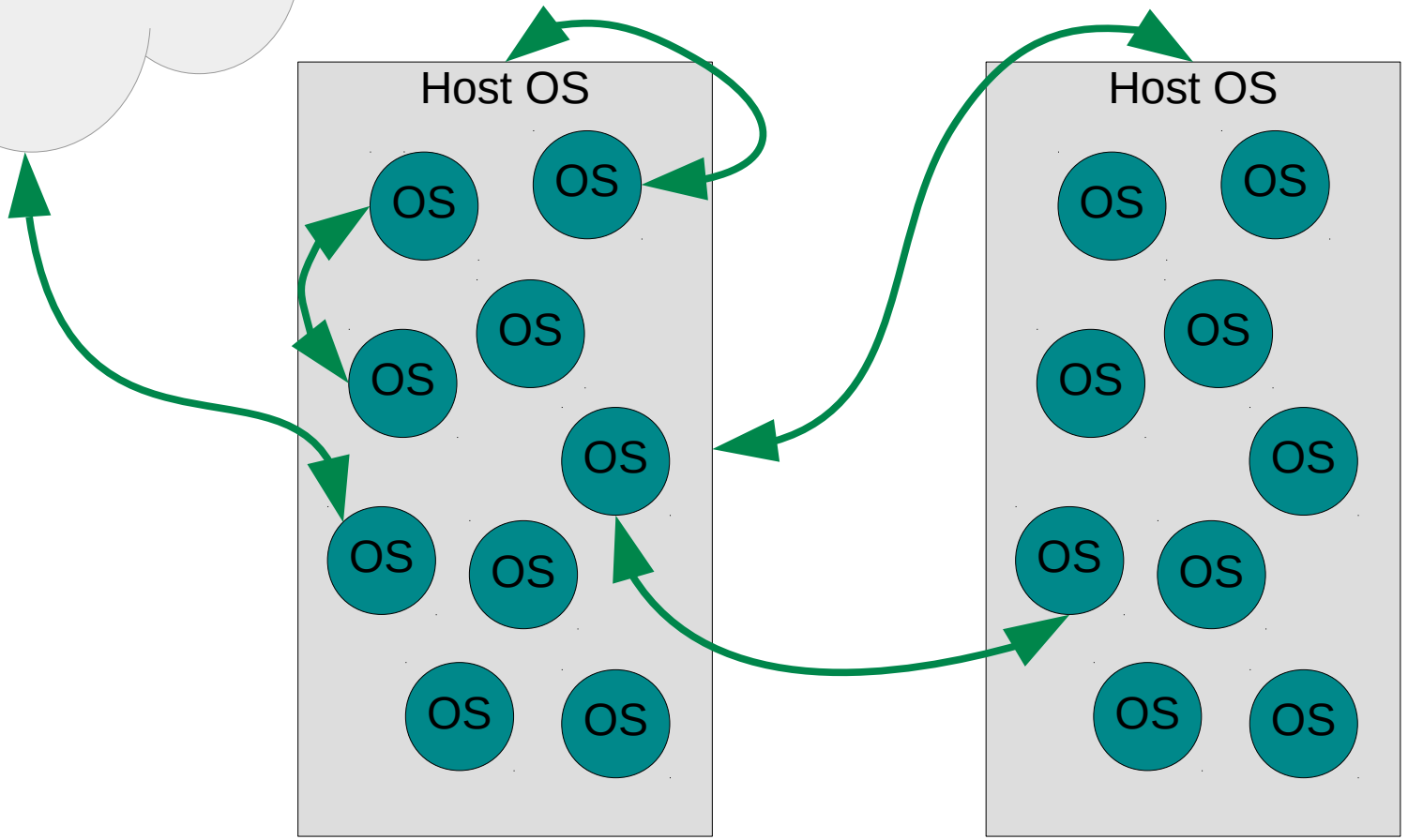
Secure isolation in Rust: virtual machines, containers, and the future of composable infrastructure

Allison Randal
University of Cambridge

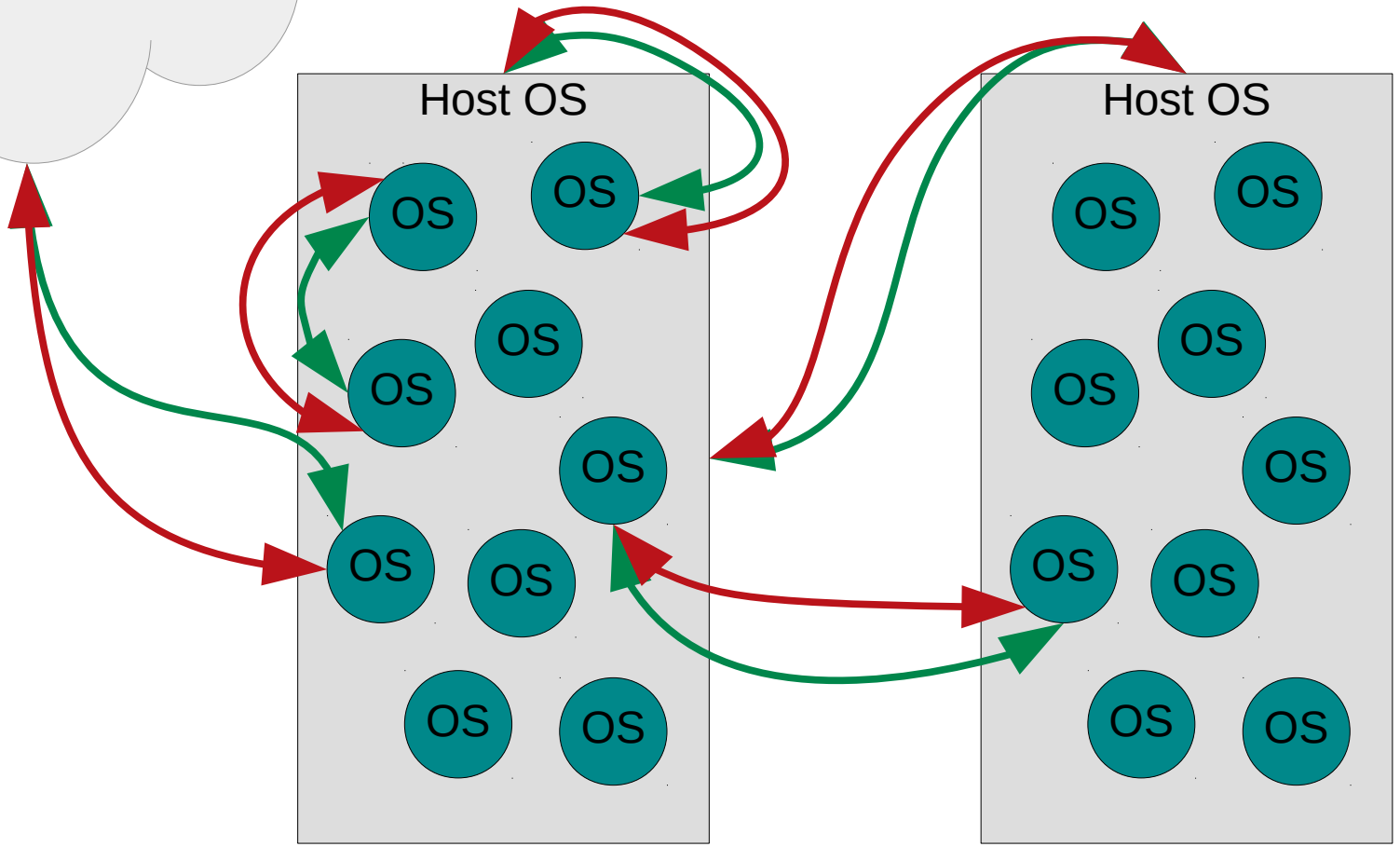
Secure Isolation



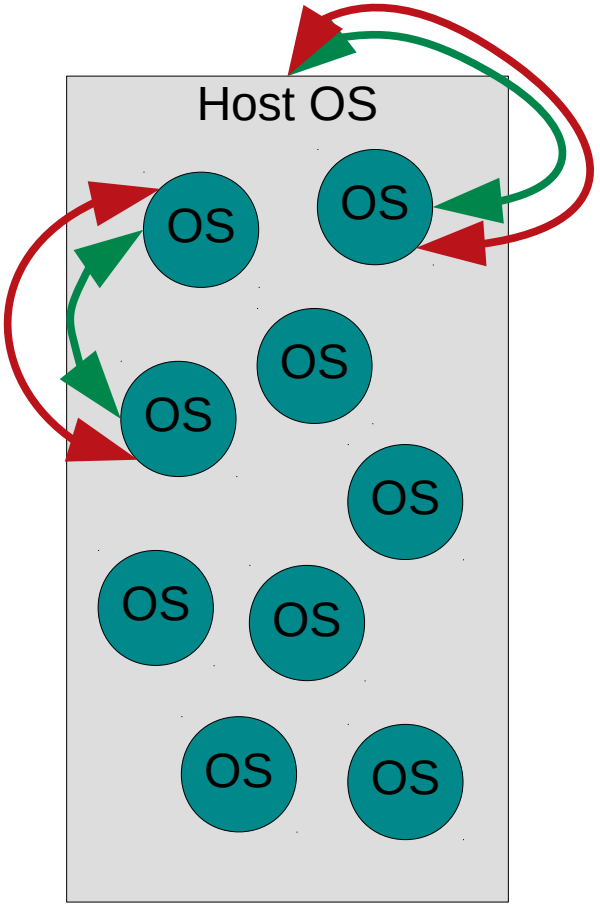
Secure Isolation



Secure Isolation



Secure Isolation



a securely isolated process,
running on a kernel,
containing an OS image

libKVM

- Lightweight hypervisor library
- Improve performance over QEMU
- Rust for memory safety
- Compatibility with C

Memory Safety

- Eliminates a class of security flaws *(see Szekeres, 2013)*
  - Spatial errors: out-of-bounds pointer, buffer overflow/underflow
  - Temporal errors: dangling pointer, null pointer, use-after-free, double free
- Doesn’t eliminate all security flaws or vulnerabilities

Memory Safety in Rust

- Compile-time checks
- Based on linear types *(see Wadler, 1990)*
- Ownership and borrowing (references)
- Mutable (one) or immutable (multiple) borrows
- Useful for rapid prototype and long-term maintenance

Compatibility with C

- Machine type constants or variables in Rust

```
const KVM_CAP_USER_MEMORY: u64 = 3;
```

- Can be passed directly to C

```
let result = libc::ioctl(self.ioctl.as_raw_fd(),
                         KVM_CHECK_EXTENSION,
                         KVM_CAP_USER_MEMORY);
```

Compatibility with C

- Struct defined in Rust

```
#[repr(C)]
struct kvm_userspace_memory_region {
    slot: u32,
    flags: u32,
    guest_phys_addr: u64,
    memory_size: u64,
    userspace_addr: u64,
}
```

- Has the same alignment as C

```
struct kvm_userspace_memory_region {
    __u32 slot;
    __u32 flags;
    __u64 guest_phys_addr;
    __u64 memory_size;
    __u64 userspace_addr;
};
```

Compatibility with C

- Library defined in Rust, with `extern`

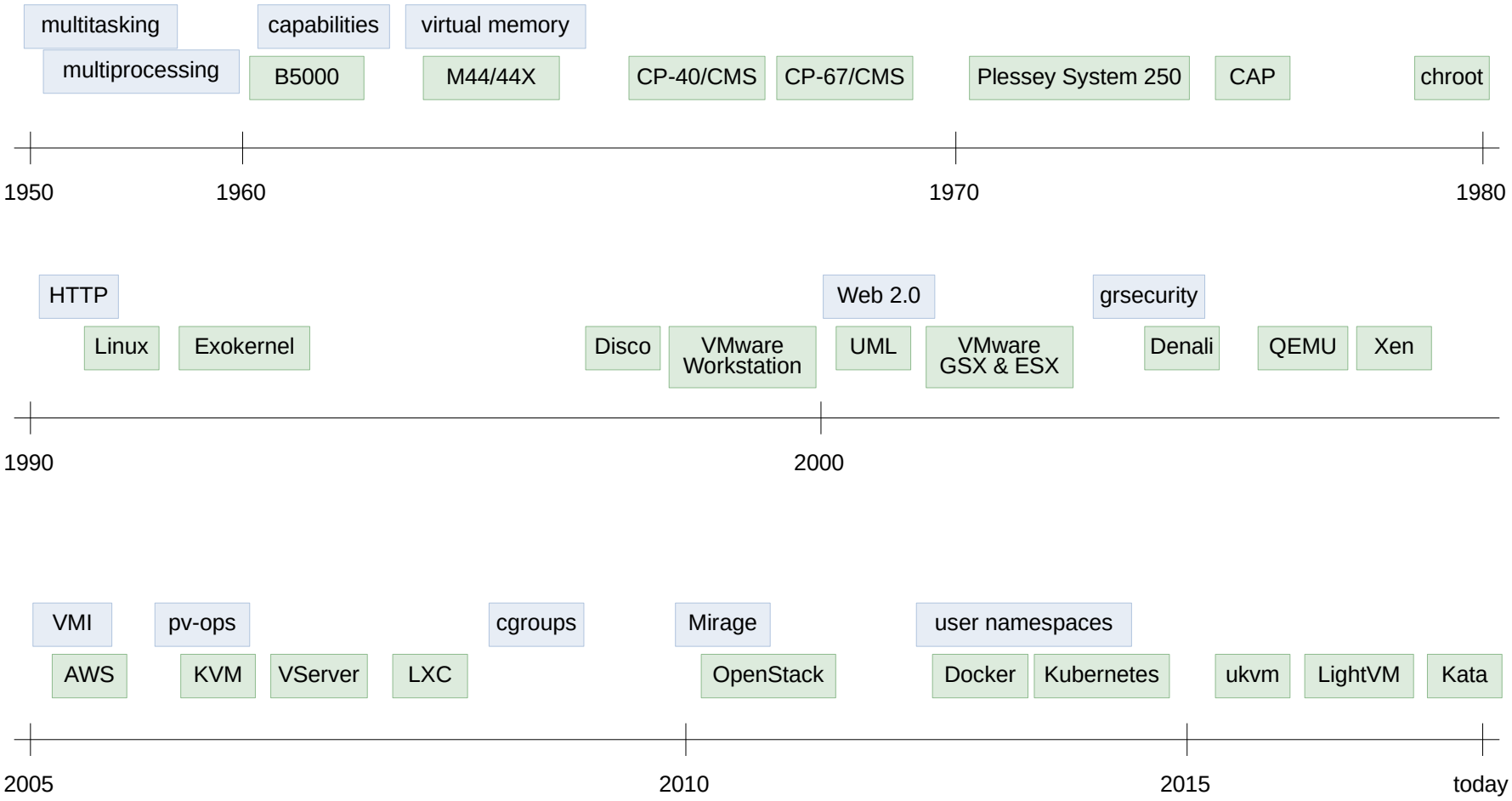
```
#[no_mangle]
pub extern fn foobar() {
    ...
}
```

- Compiled as a dylib
- Produces a `.so` shared object library
- Can be used directly in C (or via FFI)

Compatibility with C

- Advantage of modularity *(see Szekeres, 2013)*
- Build on top of C code with Rust
- Replace memory safety critical sections of C code with Rust

A Brief History



1950s

- “multiprogramming” (multitasking)
- I/O processors and multiple CPUs (multiprocessing)
- multiple processes, multiple users
  - risk of disruption
  - complex to program
- kernel isolation

1960s

- Burroughs B5000, capabilities
- M44/44X experimental machine
- CP-40/CMS and CP-67/CMS for IBM System/360

1970s

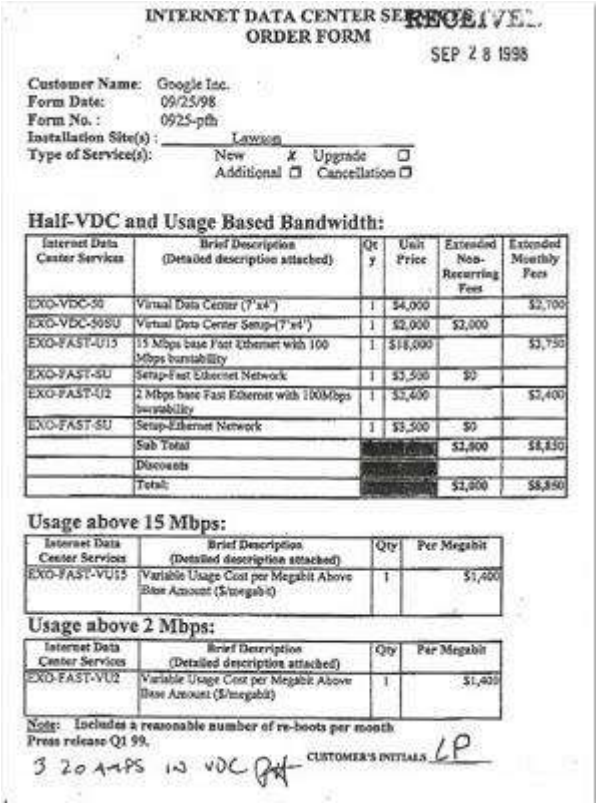
- VM/370 for IBM System/370
- Plessey System 250
- Cambridge CAP computer
- chroot, filesystem namespaces

1980s

- personal computing
- monolithic servers
- Intel iAPX 432 architecture
- IBM System/38
- RISC vs CISC

1990s

- DOS on OS/2, DOS on Windows
- Linux Kernel
- POSIX.1e capabilities
- Disco
- VMware workstation



INTERNET DATA CENTER SERVICE
ORDER FORM

RECEIVED
SEP 2 8 1998

Customer Name: Google Inc.
Form Date: 09/25/98
Form No.: 0925-pth
Installation Site(s): Lawson
Type of Service(s): New X Upgrade ☐
Additional ☐ Cancellation ☐

Half-VDC and Usage Based Bandwidth:

Internet Data Center Services	Brief Description (Detailed description attached)	Qty	Unit Price	Extended Non-Recurring Fees	Extended Monthly Fees
EXO-VDC-50	Virtual Data Center (7'x4')	1	$4,000		$2,700
EXO-VDC-50SU	Virtual Data Center Setup-(7'x4')	1	$2,000	$2,000	
EXO-FAST-U15	15 Mbps base Fast Ethernet with 100 Mbps burstability	1	$18,000		$3,750
EXO-FAST-SU	Setup-Fast Ethernet Network	1	$3,500	$0	
EXO-FAST-U2	2 Mbps base Fast Ethernet with 100Mbps burstability	1	$2,400		$2,400
EXO-FAST-SU	Setup-Ethernet Network	1	$3,500	$0	
	Sub Total			$2,000	$8,850
	Discounts				
	Total:			$2,000	$8,850

Usage above 15 Mbps:

Internet Data Center Services	Brief Description (Detailed description attached)	Qty	Per Megabit
EXO-FAST-VU15	Variable Usage Cost per Megabit Above Base Amount ($/megabit)	1	$1,400

Usage above 2 Mbps:

Internet Data Center Services	Brief Description (Detailed description attached)	Qty	Per Megabit
EXO-FAST-VU2	Variable Usage Cost per Megabit Above Base Amount ($/megabit)	1	$1,400

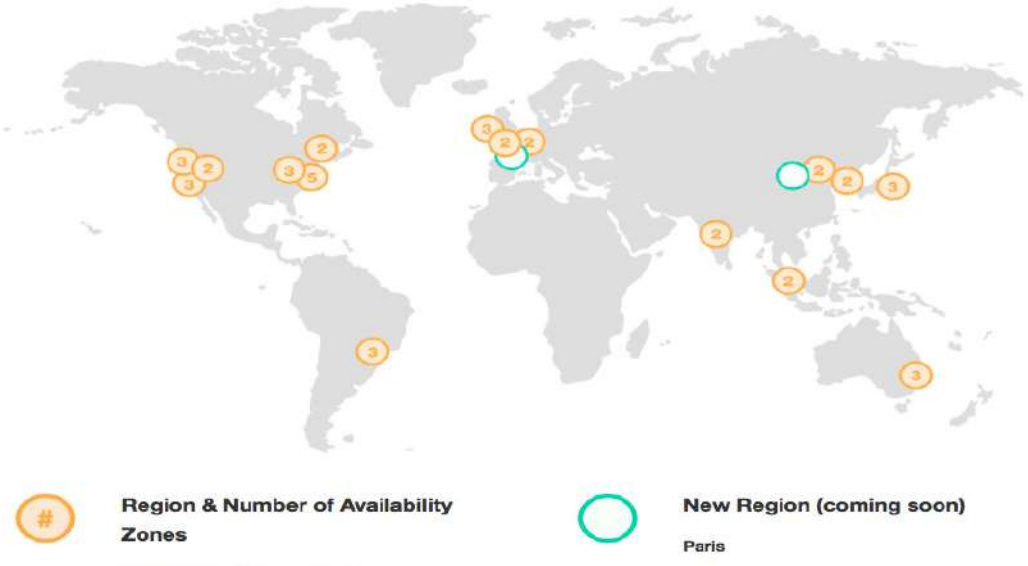
Note: Includes a reasonable number of re-boots per month
Press release Q1 99.
3 20 MPS in VDC
CUSTOMER'S INITIALS LP

2000s

- Web 2.0, smaller/lighter
- FreeBSD Jails
- Linux VServer and OpenVZ
- VMware for servers (ESX & GSX)
- Denali, paravirtualization
- QEMU
- Xen, multitenancy as a business
- Solaris Zones (containers)

2000s

- Amazon Web Services, cloud
- Linux namespaces: process IDs, IPC, network stack
- KVM, hardware virtualization
- Google Borg
- cgroups
- LXC



2010s

- OpenStack
- Docker
- Linux user namespaces
- Kubernetes
- Kata Containers (was Intel Clear Containers)
- ukvm, LightVM

2010s

- FreeBSD capabilities, Capsicum
- CHERI
- RISC-V
- Google Fuchsia, capability-based OS
- Spectre, Meltdown, Foreshadow, L1TF
- OpenBSD pledge, unveil
- Open Titan

What lies ahead?

- Speculative execution vulnerabilities invalidate assumptions about secure isolation in VMs and containers
- Memory safety is like flossing
- Re-examine the full stack from hardware to application workloads
- Party like it’s 1960

Questions?

Further Reading

- A. Balasubramanian, M. S. Baranowski, A. Burtsev, A. Panda, Z. Rakamarić & Leonid Ryzhyk (2017) “System Programming in Rust: Beyond Safety”, *Proceedings of HotOS ‘17*.
- L. Szekeres, M. Payer, T. Wei, and D. Song (2013) “SoK: Eternal War in Memory”, *Proceedings of the 2013 IEEE Symposium on Security and Privacy*.
- J. Blandy & J. Orendorff (2017) *Programming Rust: Fast, Safe Systems Development*, O’Reilly.
- N. D. Matsakis & F. S. Klock, II (2014) “The Rust Language”, *Proceedings of the 2014 ACM SIGAda Annual Conference on High Integrity Language Technology*.

Images

- PDP-1, Copyright 2006, Matthew Hutchinson, CC BY 2.0.
- Burroughs B5000, origin unknown, http://www.retrocomputingtasmania.com/home/projects/burroughs-b5500/b5000_b5500_gallery
- CAP computer, Copyright 2004, Daderot, CC BY-SA 3.0.
- IMSAI 8080 from “WarGames”, Copyright 1983, MGM/UA Entertainment Company.
- Google data center order form, 1998, https://plus.google.com/+UrsH%C3%B6lzle/posts/UseinB6wvmh
- AWS availability zones, Copyright 2016, Amazon.com, Inc. CC BY-SA 4.0.
- Spectre, Meltdown, and Foreshadow icons, Copyright 2018, Natascha Eibl (https://vividfox.me/), CC0.
- Futuristic data center, origin unknown, https://on.rt.com/lu029w