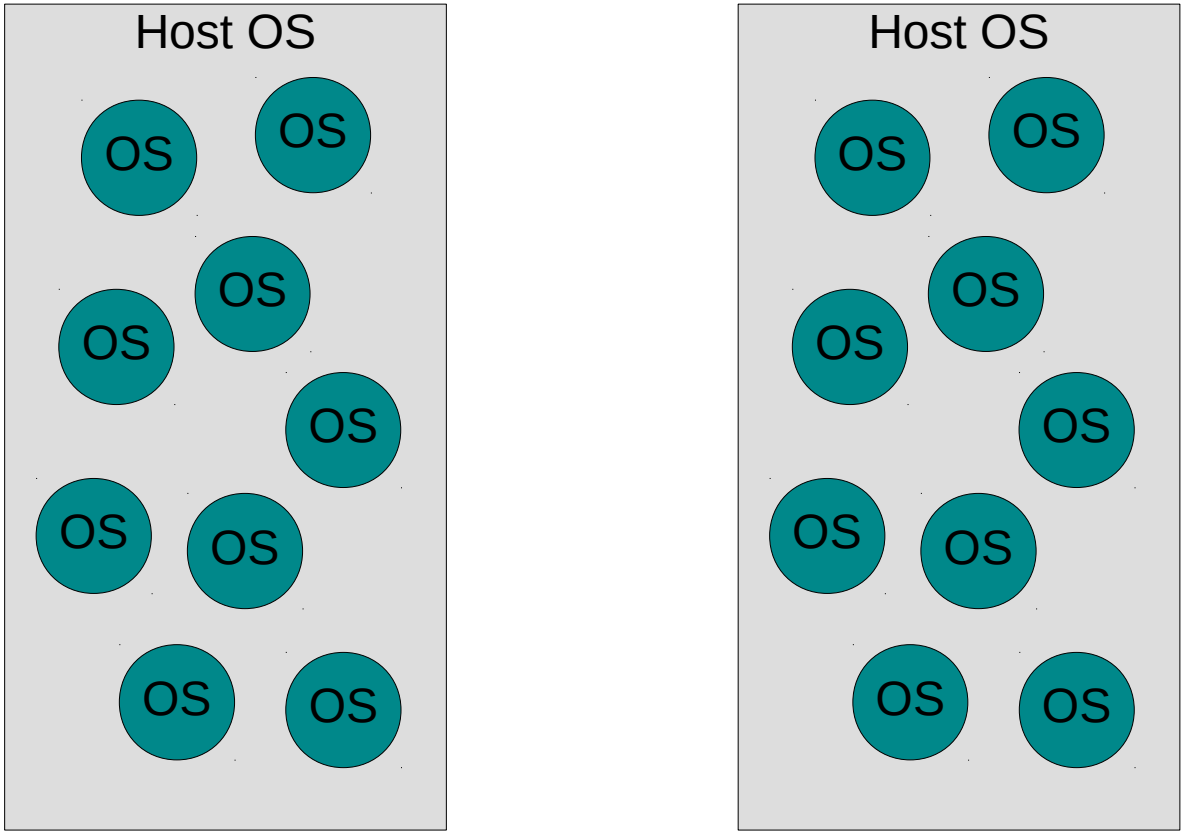
The ideal versus the real: a brief history of secure isolation in virtual machines and containers

Allison Randal
University of Cambridge

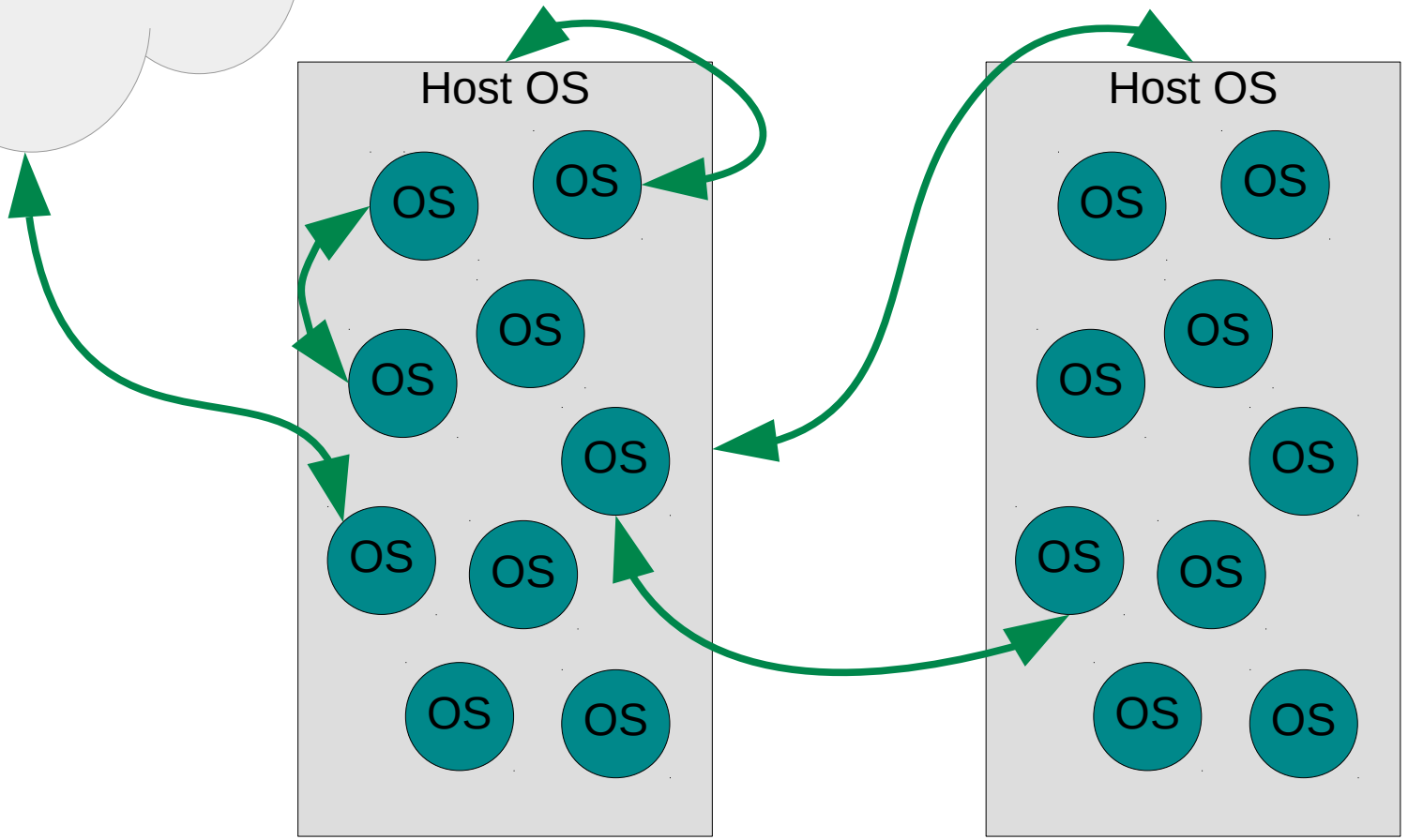
Between the idea
And the reality
Between the motion
And the act
Falls the Shadow

–T.S. Eliot, “The Hollow Men”

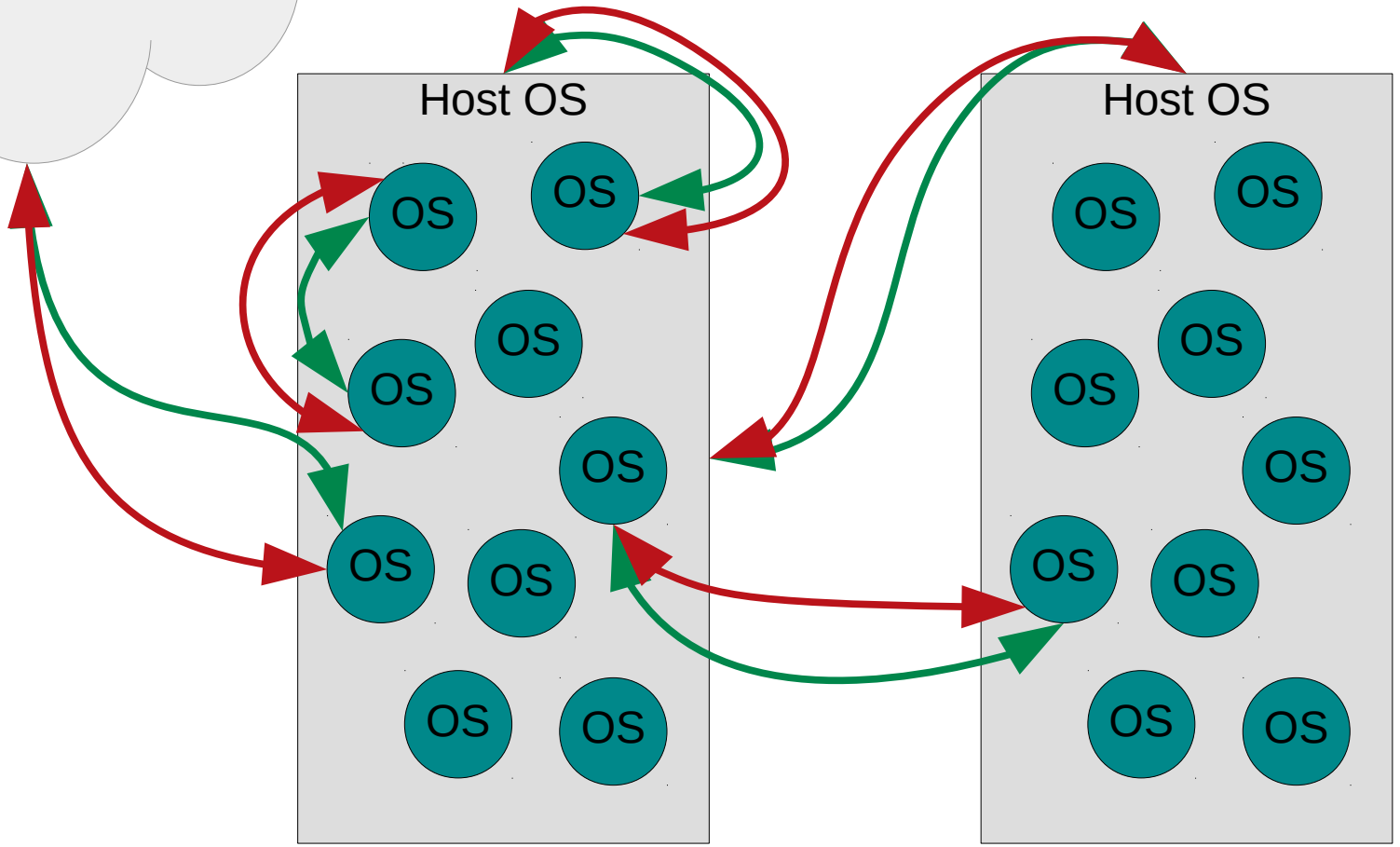
Secure Isolation



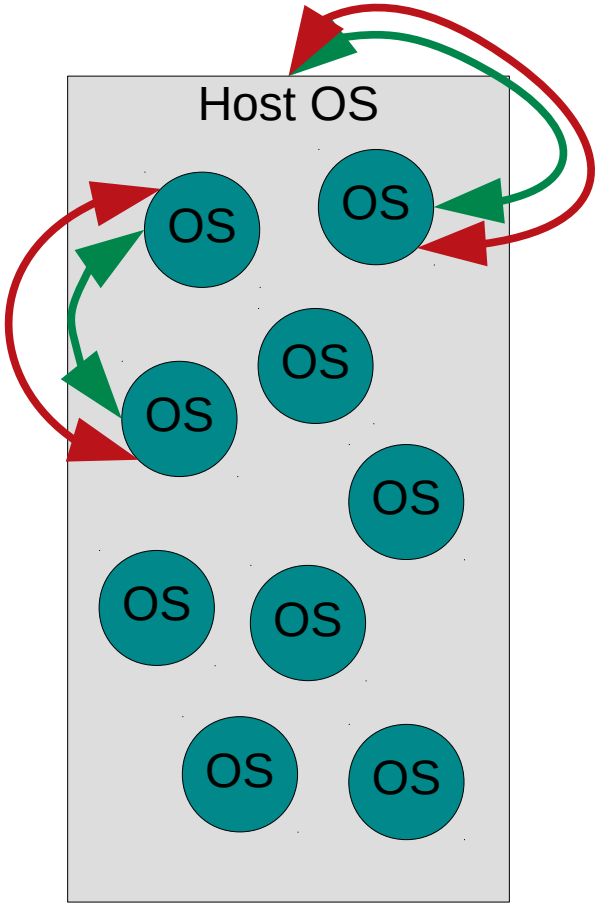
Secure Isolation



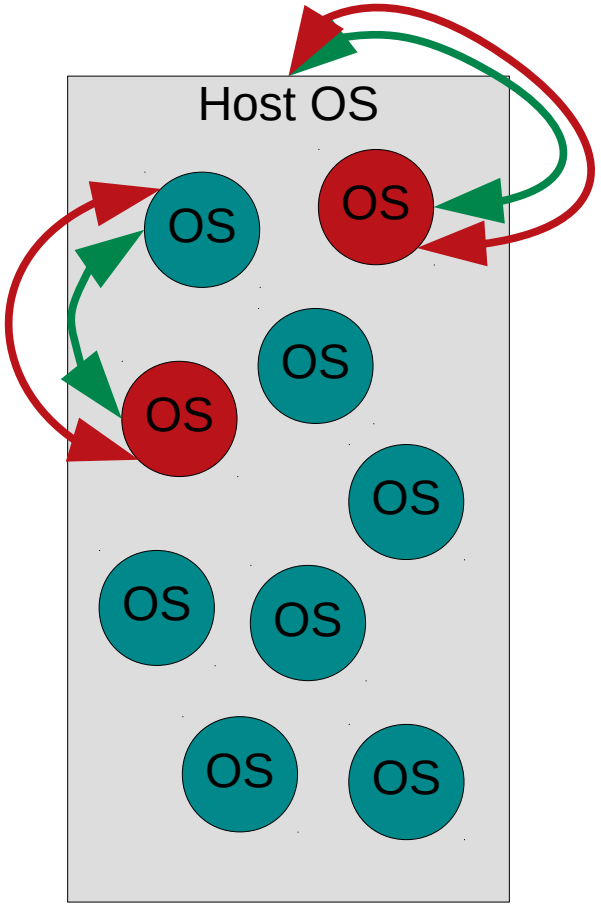
Secure Isolation



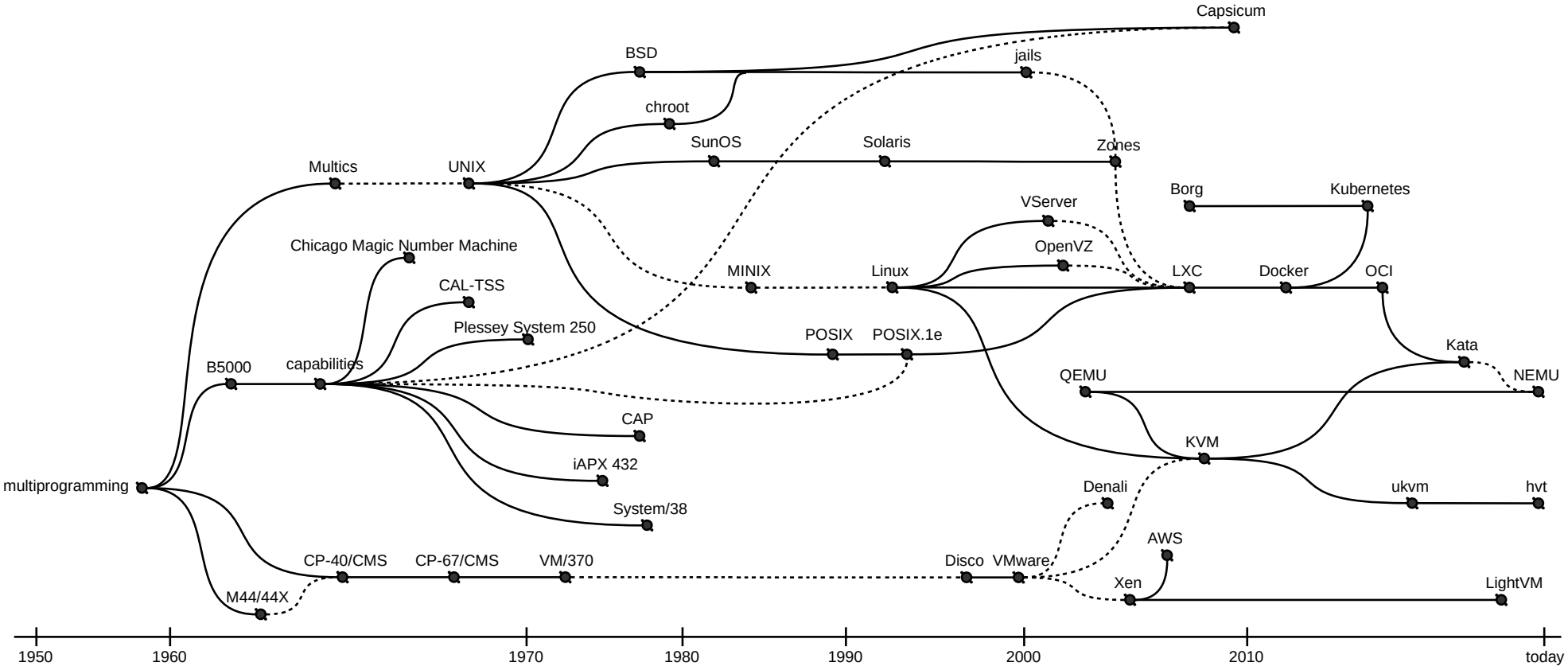
Secure Isolation

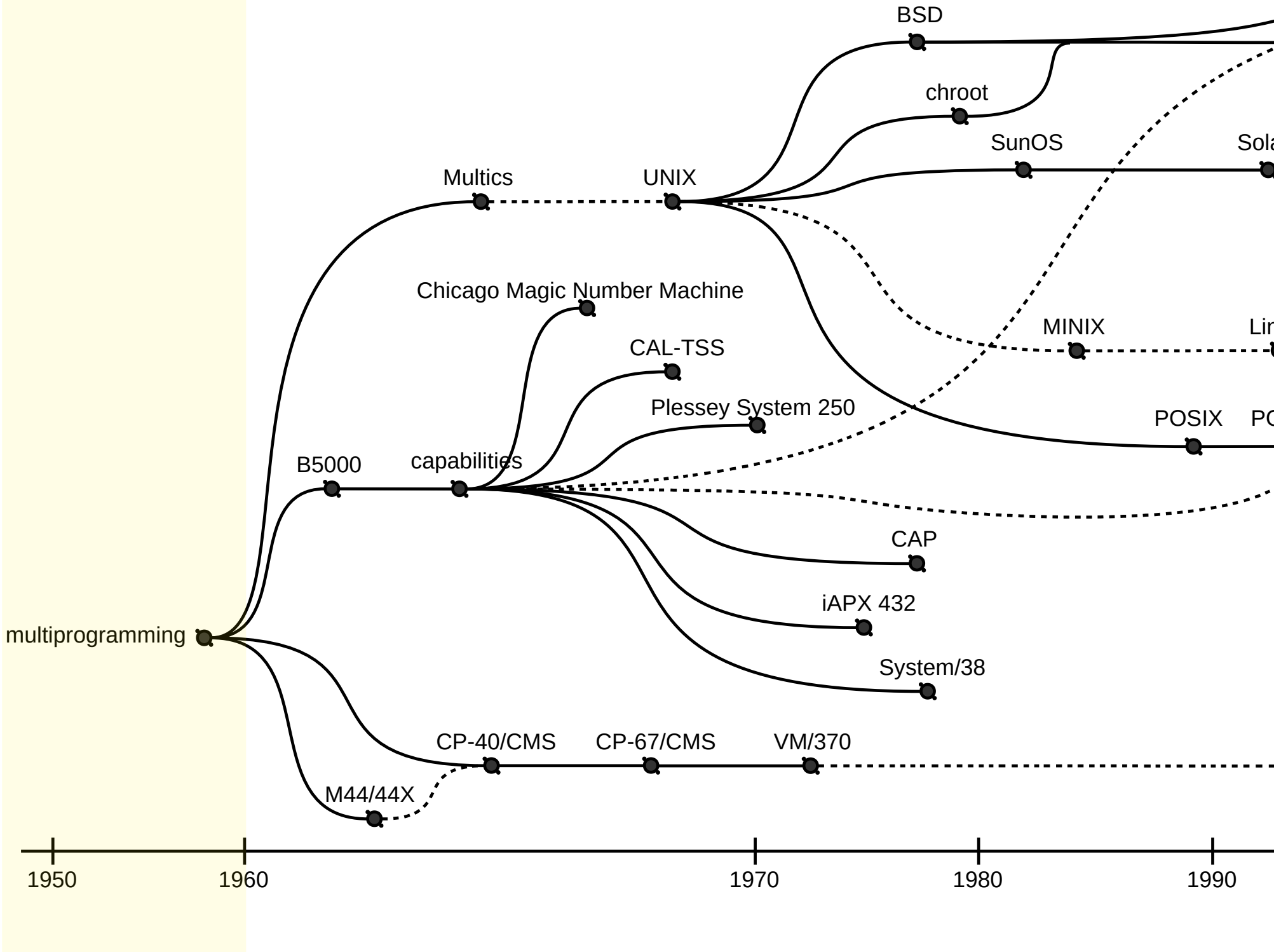


Secure Isolation



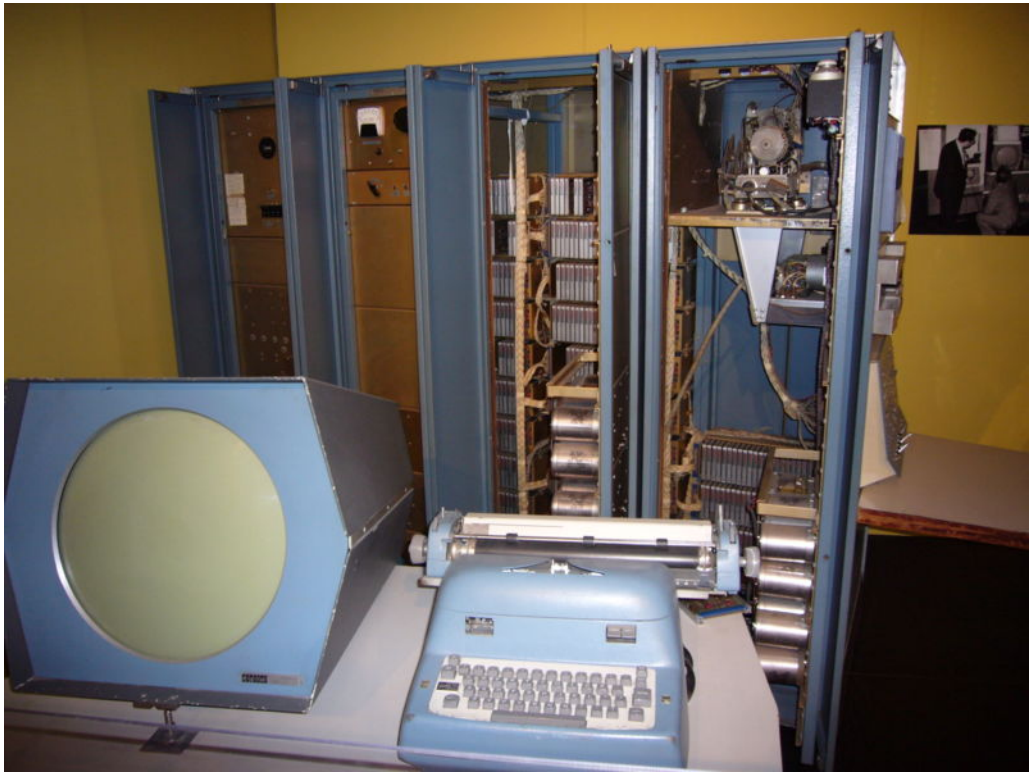
a securely isolated process,
running on a kernel,
containing an OS image


Capsicum
BSD
jails
chroot
SunOS
Solaris
Zones
Multics
UNIX
Borg
Kubernetes
VServer
Chicago Magic Number Machine
OpenVZ
MINIX
Linux
LXC
Docker
OCI
CAL-TSS
Plessey System 250
POSIX
POSIX.1e
Kata
B5000
capabilities
QEMU
NEMU
CAP
KVM
iAPX 432
multiprogramming
Denali
ukvm
hvt
System/38
AWS
CP-40/CMS
CP-67/CMS
VM/370
Disco
VMware
Xen
LightVM
M44/44X
1950
1960
1970
1980
1990
2000
2010
today


BSD
chroot
SunOS
Sola
Multics
UNIX
Chicago Magic Number Machine
MINIX
Lin
CAL-TSS
Plessey System 250
POSIX
PO
B5000
capabilities
CAP
iAPX 432
multiprogramming
System/38
CP-40/CMS
CP-67/CMS
VM/370
M44/44X
1950
1960
1970
1980
1990

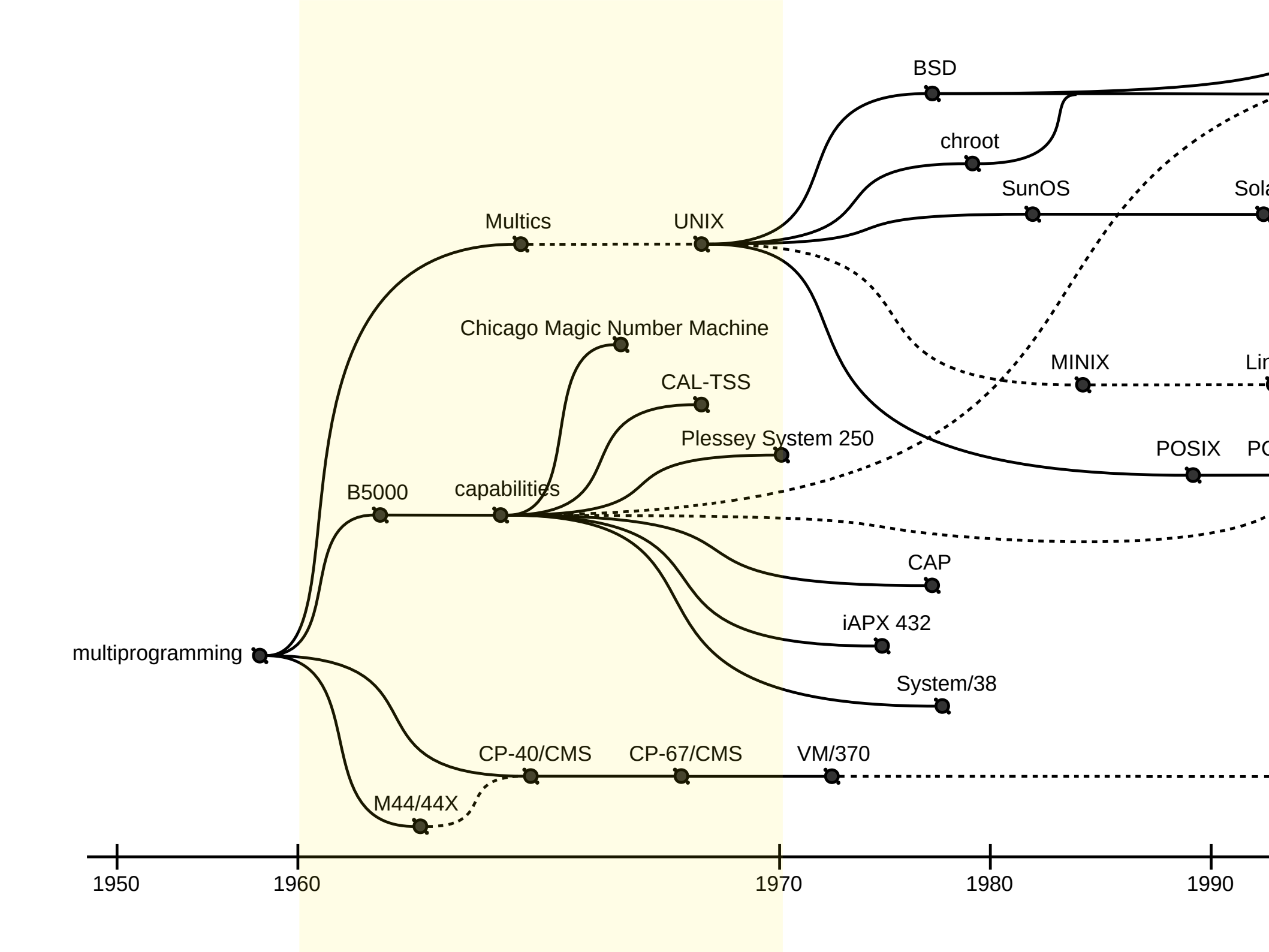
1950s

- Multiprogramming[1] [2]
  - multitasking
  - multiprocessing: I/O processors and multiple CPUs
  - time-sharing
  - increase utilization
  - risk of disruption
  - complex to program
- kernel isolation[3] [2]



PDP-1, (C) 2006, Matthew Hutchinson, CC BY 2.0

[1]E. F. Codd, E. S. Lowry, E. McDonough, and C. A. Scalzi. Multiprogramming STRETCH: Feasibility Considerations. *Communications of the ACM*, 2(11):13–17, Nov. 1959.
[2]A. Opler and N. Baird. Multiprogramming: The Programmer's View. *In Proceedings of the 14th National Meeting of the Association for Computing Machinery*, 1–4, 1959.
[3]J. P. Buzen and U. O. Gagliardi. The Evolution of Virtual Machine Architecture. *In Proceedings of the National Computer Conference and Exposition*, AFIPS '73, 291–299, 1973.


BSD
chroot
SunOS
Sola
Multics
UNIX
Chicago Magic Number Machine
MINIX
Lin
CAL-TSS
Plessey System 250
POSIX
PO
B5000
capabilities
CAP
iAPX 432
multiprogramming
System/38
CP-40/CMS
CP-67/CMS
VM/370
M44/44X
1950
1960
1970
1980
1990

1960s

- Capabilities
  - B5000[1] descriptors
  - theoretical[2] protected memory, ownership, subsets
  - MIT implementation on (modified) PDP-1[3]
  - Chicago Magic Number Machine[4]
  - CAL-TSS[4]
  - Provably Secure Operating System[5 6]

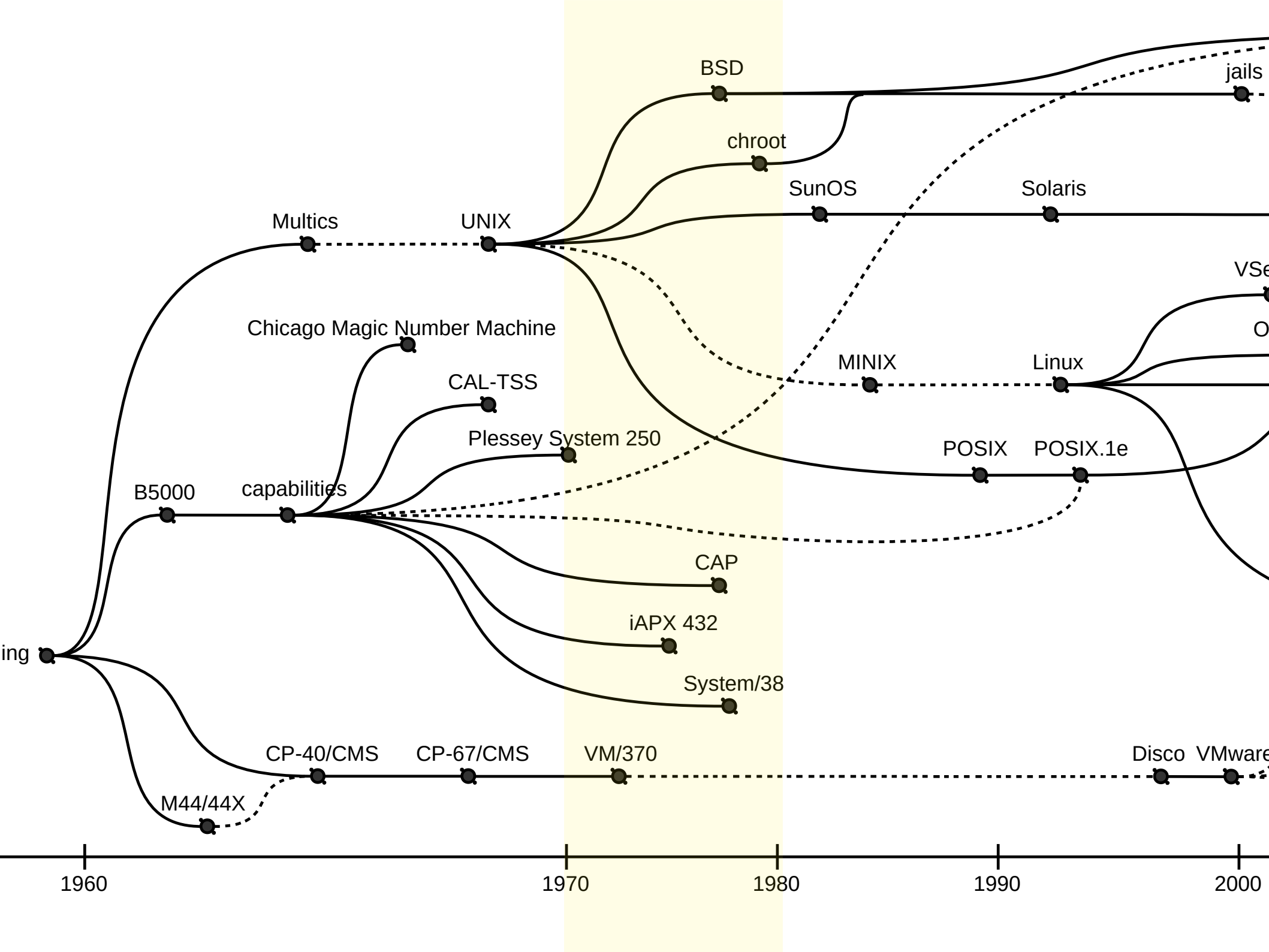
Burroughs B5000, origin unknown
http://www.retrocomputingtasmania.com/home/projects/burroughs-b5500/b5000_b5500_gallery

[1]A. J. W. Mayer. The Architecture of the Burroughs B5000: 20 Years Later and Still Ahead of the Times? *SIGARCH Comput. Archit. News*, 10(4):3–10, June 1982.
[2]J. B. Dennis and E. C. Van Horn. Programming Semantics for Multiprogrammed Computations. *Communications of the ACM*, 9(3):143–155, Mar. 1966.
[3]W. B. Ackerman and W. W. Plummer. An Implementation of a Multiprocessing Computer System. *In Proceedings of the First ACM Symposium on Operating System Principles* (SOSP '67), 5.1–5.10, 1967.
[4]H. M. Levy. *Capability-Based Computer Systems*. Digital Press, 1984.
[5]P. G. Neumann. A Provably Secure Operating System: The system, its applications, and proofs. *Technical report, Computer Science Laboratory, SRI International*, 1980.
[6]P. G. Neumann and R. J. Feiertag. PSOS revisited. *In Proceedings of the 19th Annual Computer Security Applications Conference*, 208–216, Dec. 2003.

1960s

- VMs
  - M44/44X[1] virtual memory
  - CP-40/CMS[2], CP-67/CMS[3] for IBM System/360 interrupt separation, paged guest memory, simulated devices, efficient utilization
- OS
  - Multics[4]
  - Unix[5]

[1]R. A. Nelson. *Mapping Devices and the M44 Data Processing System*. Research Report RC-1303, IBM Thomas J. Watson Research Center. 1964.
[2]R. J. Adair, R. U. Bayles, L. W. Comeau, and R. J. Creasy. *A Virtual Machine System for the 360/40*. Technical Report 36.010, IBM Cambridge Scientific Center, May 1966.
[3]*Control Program-67 Cambridge Monitor System*. IBM Type III Release No. 360D-05.2.005. IBM Corporation, Oct. 1971.
[4]J. B. Dennis. Segmentation and the Design of Multiprogrammed Computer Systems. *Journal of the ACM*, 12(4):589–602, Oct. 1965.
[5]D. Ritchie. The Evolution of the Unix Time-Sharing System. *In Proceedings of a Symposium on Language Design and Programming Methodology*, 25–36, 1980. Springer-Verlag.


BSD
jails
chroot
SunOS
Solaris
Multics
UNIX
VSe
Chicago Magic Number Machine
O
MINIX
Linux
CAL-TSS
Plessey System 250
POSIX
POSIX.1e
B5000
capabilities
CAP
iAPX 432
ing
System/38
CP-40/CMS
CP-67/CMS
VM/370
Disco
VMware
M44/44X
1960
1970
1980
1990
2000

1970s

- Capabilities
  - Plessey System 250[1] telephone-switch controller
  - CAP[2] hardware and OS
  - Intel iAPX 432[3] poor performance[4]
  - IBM System/38[5]

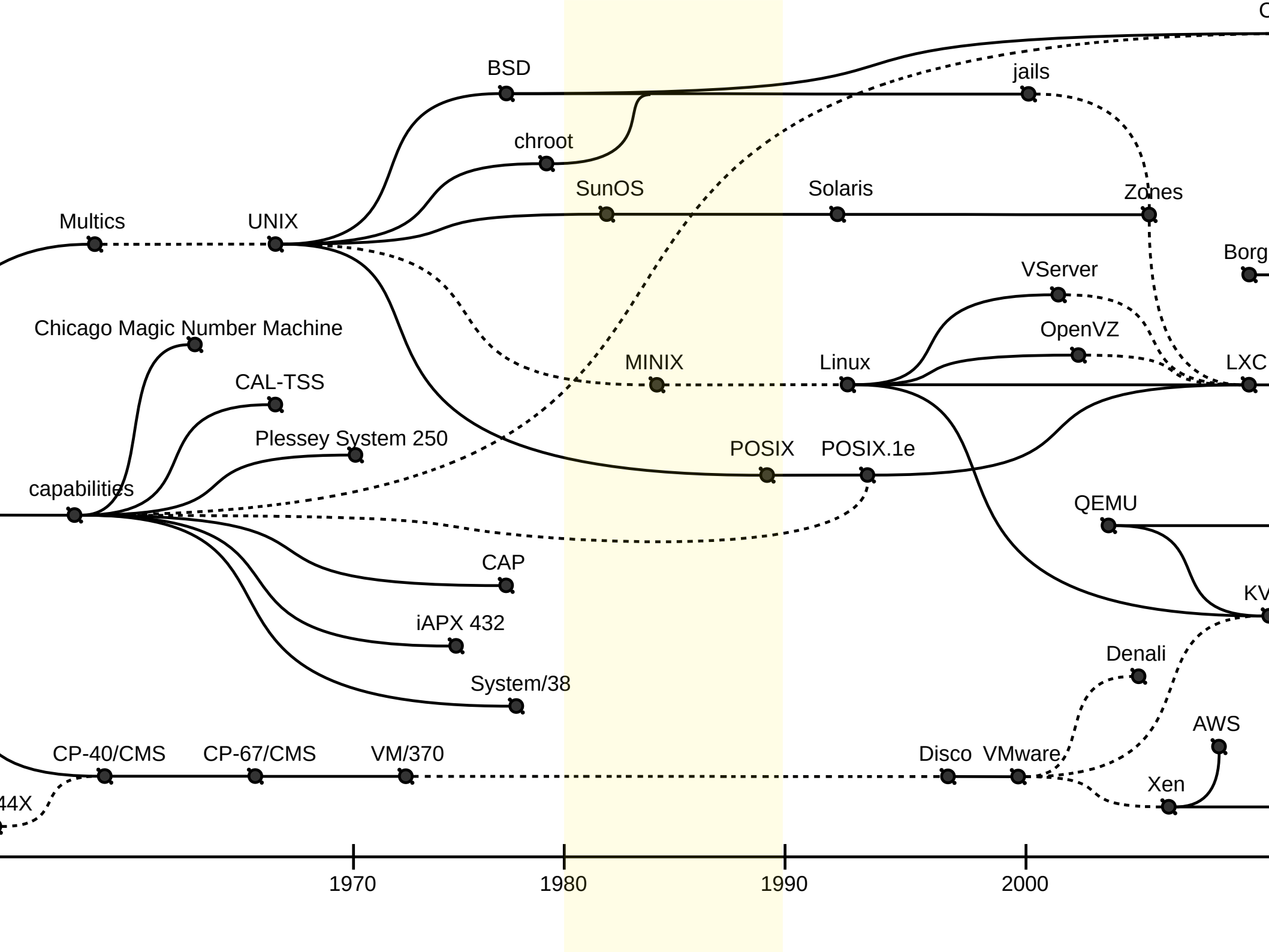
CAP, (C) 2004, Daderot, CC BY-SA 3.0

[1]D. M. England. Capability Concept Mechanism and Structure in System 250. *In Proceedings of the International Workshop on Protection in Operating Systems*, 63–82, Aug. 1974. IRIA.
[2]R. M. Needham and R. D. H. Walker. The Cambridge CAP Computer and its protection system. *In Proceedings of the Sixth ACM Symposium on Operating Systems Principles*, 1–10, Nov. 1977. ACM.
[3]*iAPX 432 General Data Processor Architecture Reference Manual*. Intel Corporation, 1981.
[4]P. M. Hansen, M. A. Linton, R. N. Mayo, M. Murphy, and D. A. Patterson. A Performance Evaluation of the Intel iAPX 432. *SIGARCH Comput. Archit. News*, 10(4):17–26, June 1982.
[5]M. E. Houdek, F. G. Soltis, and R. L. Hoffman. IBM System/38 Support for Capability-based Addressing. *In Proceedings of the 8th Annual Symposium on Computer Architecture*, 341–348, 1981. IEEE.

1970s

- VMs
  - VM/370[1] for IBM System/370 virtual memory hardware
  - “Since a privileged software nucleus has, in principle, no way of determining whether it is running on a virtual or a real machine, it has no way of spying on or altering any other virtual machine that may be coexisting with it in the same system. […] In practice no virtual machine is completely equivalent to its real machine counterpart.”[2]
- OS
  - BSD[3]
  - chroot[4] filesystem namespaces

[1]R. J. Creasy. The Origin of the VM/370 Time-Sharing System. *IBM Journal of Research and Development*, 25(5):483–490, Sept. 1981.
[2]J. P. Buzen and U. O. Gagliardi. The Evolution of Virtual Machine Architecture. *In Proceedings of the National Computer Conference and Exposition*, AFIPS ’73, 291–299, 1973.
[3]M. K. McKusick, M. J. Karels, K. Sklower, K. Fall, M. Teitelbaum, and K. Bostic. Current Research by The Computer Systems Research Group of Berkeley. *In Proceedings of the European UNIX Users Group*, Apr. 1989.
[4]B. Kernighan and M. McIlroy. *UNIX Time-sharing System: UNIX Programmer’s Manual, volume 1, Seventh Edition*. Bell Telephone Laboratories, 1979.

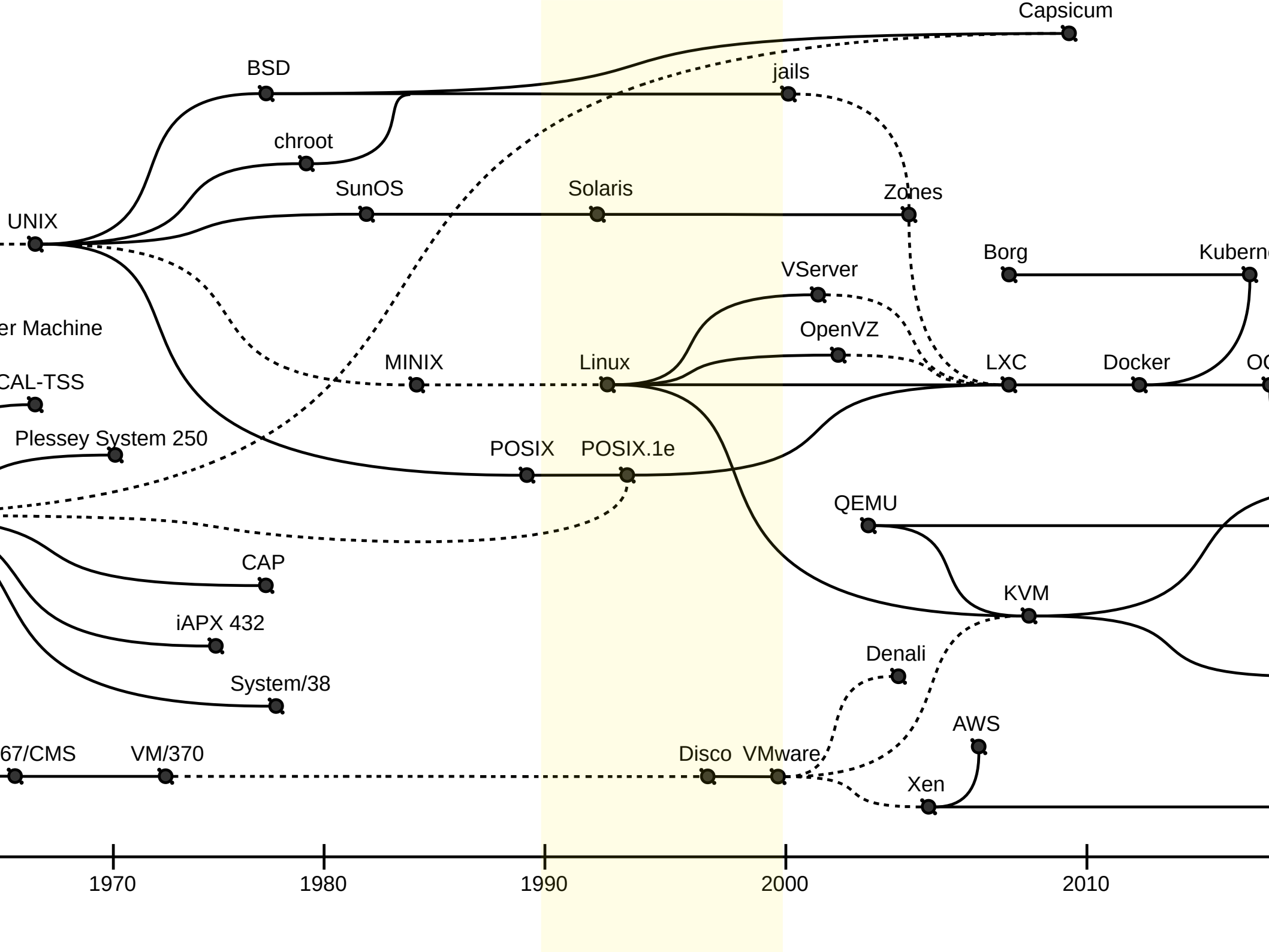

BSD
jails
chroot
SunOS
Solaris
Zones
Multics
UNIX
Borg
VServer
Chicago Magic Number Machine
OpenVZ
MINIX
Linux
LXC
CAL-TSS
Plessey System 250
POSIX
POSIX.1e
capabilities
QEMU
CAP
KV
iAPX 432
Denali
System/38
AWS
CP-40/CMS
CP-67/CMS
VM/370
Disco
VMware
Xen
1970
1980
1990
2000

1980s

- personal computing[1] & monolithic servers
- hardware without virtualization support[2]
- general purpose OS
- Intel x86[3] "a crash program…to save Intel's market share"[4]
- RISC[5] vs CISC

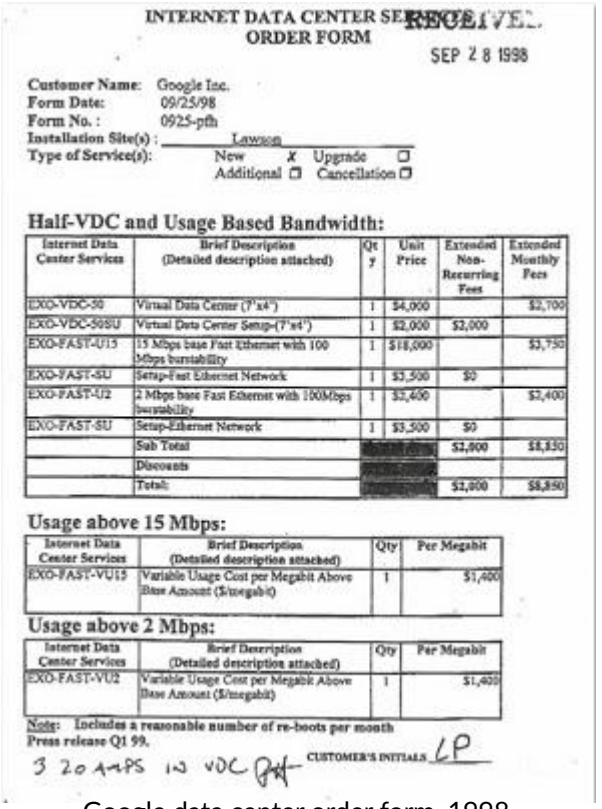
IMSAI 8080 from "WarGames", (C) 1983, MGM/UA

[1]R. J. Creasy. The Origin of the VM/370 Time-Sharing System. *IBM Journal of Research and Development*, 25(5):483–490, Sept. 1981.
[2]L. I. Dickman. Small Virtual Machines: A Survey. *In Proceedings of the Workshop on Virtual Computer Systems*, 191–202, 1973. ACM.
[3]S. P. Morse, B. W. Raveiel, S. Mazor and W. B. Pohiman. Intel Microprocessors–8008 to 8086. *IEEE Computer*, 13(10): 42-60, Oct. 1980.
[4]S. Mazor. Intel's 8086. *IEEE Annals of the History of Computing*, 32(1):75–79, Jan. 2010.
[5]D. A. Patterson and C. H. Sequin. RISC I: A Reduced Instruction Set VLSI Computer. *In Proceedings of the 8th Annual Symposium on Computer Architecture*, 443–457, 1981. IEEE.


Capsicum
BSD
jails
chroot
SunOS
Solaris
Zones
UNIX
Borg
VServer
OpenVZ
MINIX
Linux
LXC
Docker
Plessey System 250
POSIX
POSIX.1e
QEMU
CAP
KVM
iAPX 432
Denali
System/38
AWS
VM/370
Disco
VMware
Xen
1970
1980
1990
2000
2010

1990s

- Containers
  - POSIX.1e capabilities[1]
  - Linux Kernel capabilities[2]
  - Plan 9 namespaces[3] filesystem, process, network, memory
- VMs
  - Disco[4] binary translation
  - VMware[5]
- Google scale?



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	[redacted]		$2,000	$8,850
	Discounts	[redacted]			
	Total:	[redacted]		$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

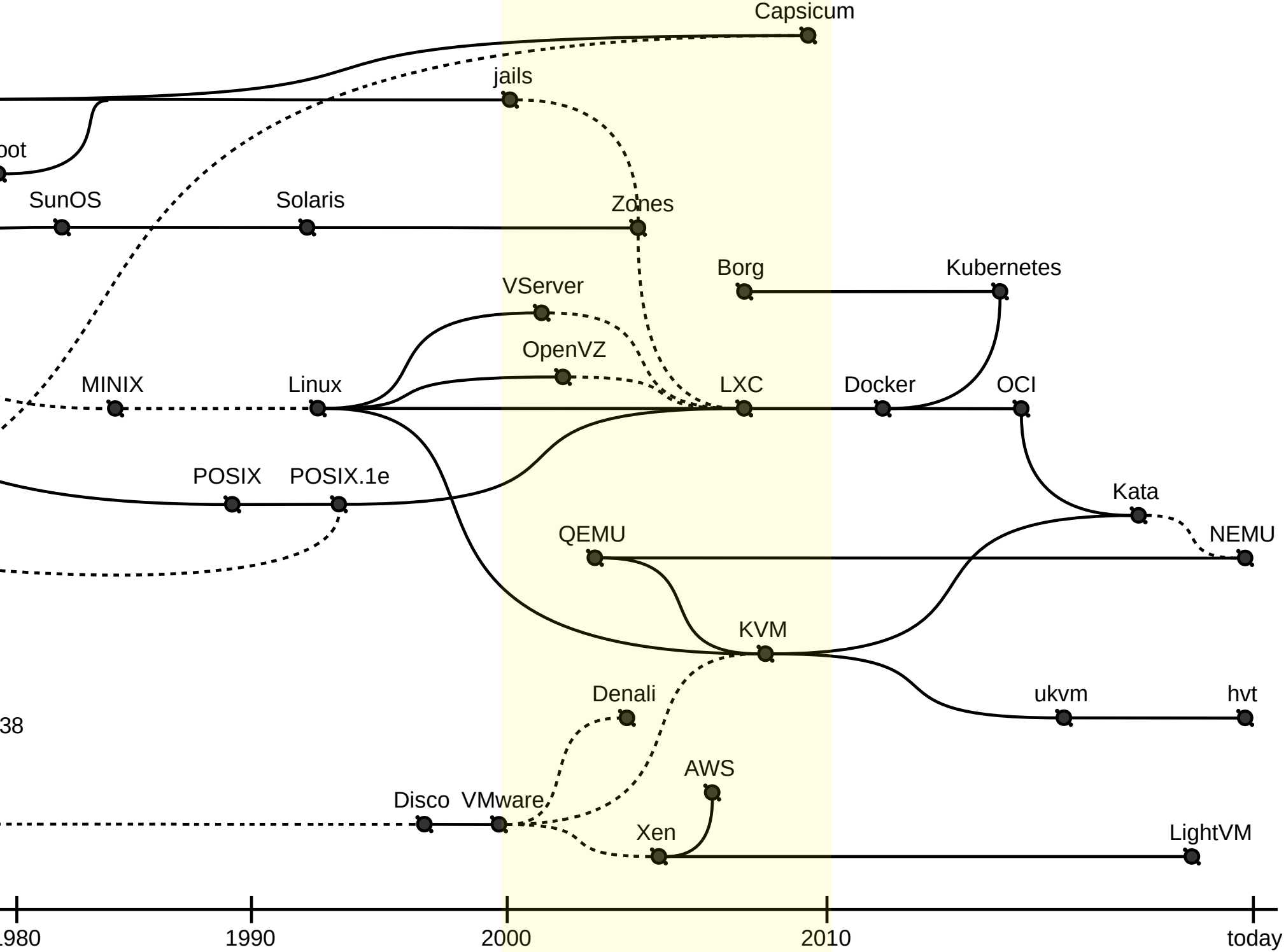
Usage above 2 Mbps:

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 mbps in VDC CUSTOMER'S INITIALS LP

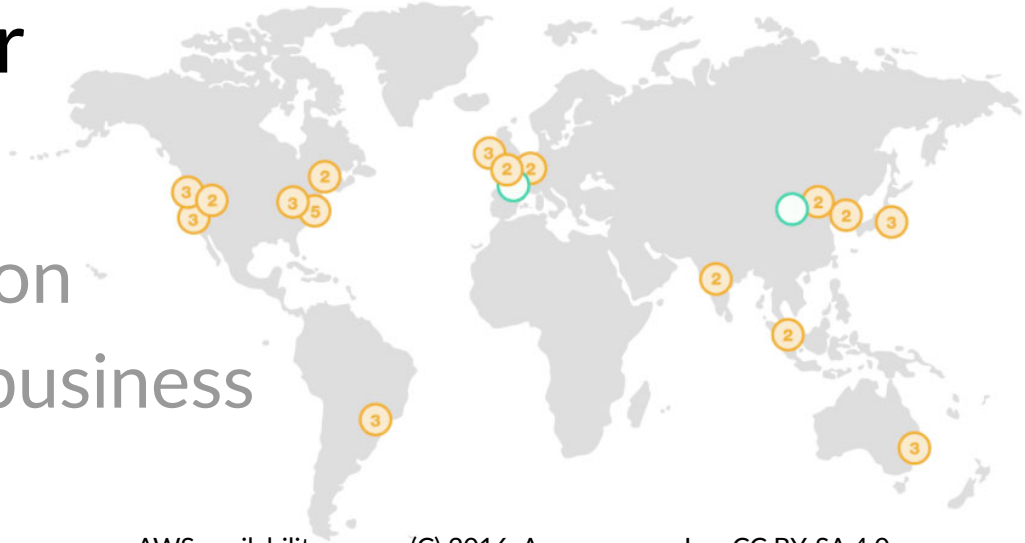
Google data center order form, 1998
https://plus.google.com/+UrsH%C3%B6lzle/posts/UseinB6wvmh

[1] *Protection, Audit and Control Interfaces.* Draft POSIX Standard 1003.1e, IEEE, Oct. 1997.
[2] *capabilities(7) man page*, http://man7.org/linux/man-pages/man7/capabilities.7.html.
[3] R. Pike, D. Presotto, K. Thompson, H. Trickey, and P. Winterbottom. The Use of Name Spaces in Plan 9. SIGOPS Oper. Syst. Rev., 27(2):72–76, Apr. 1993.
[4] E. Bugnion, S. Devine, K. Govil, and M. Rosenblum. Disco: Running Commodity Operating Systems on Scalable Multiprocessors. *ACM Trans. Comput. Syst.*, 15(4):412–447, Nov. 1997.
[5] E. Bugnion, S. Devine, M. Rosenblum, J. Sugerman, and E. Y. Wang. Bringing Virtualization to the x86 Architecture with the Original VMware Workstation. *ACM Trans. Comput. Syst.*, 30(4):12:1–12:51, Nov. 2012.


Capsicum
jails
oot
SunOS
Solaris
Zones
Borg
Kubernetes
VServer
OpenVZ
MINIX
Linux
LXC
Docker
OCI
POSIX
POSIX.1e
Kata
QEMU
NEMU
KVM
Denali
ukvm
hvt
38
AWS
Disco
VMware
Xen
LightVM
1980
1990
2000
2010
today

2000s

- Web 2.0, smaller/lighter
- VMs
  - Denali[1] [2] paravirtualization
  - Xen[3] multitenancy as a business
  - Amazon Web Services[4] cloud, VM orchestration
  - x86 hardware virtualization[5]
  - KVM[6] (with QEMU)



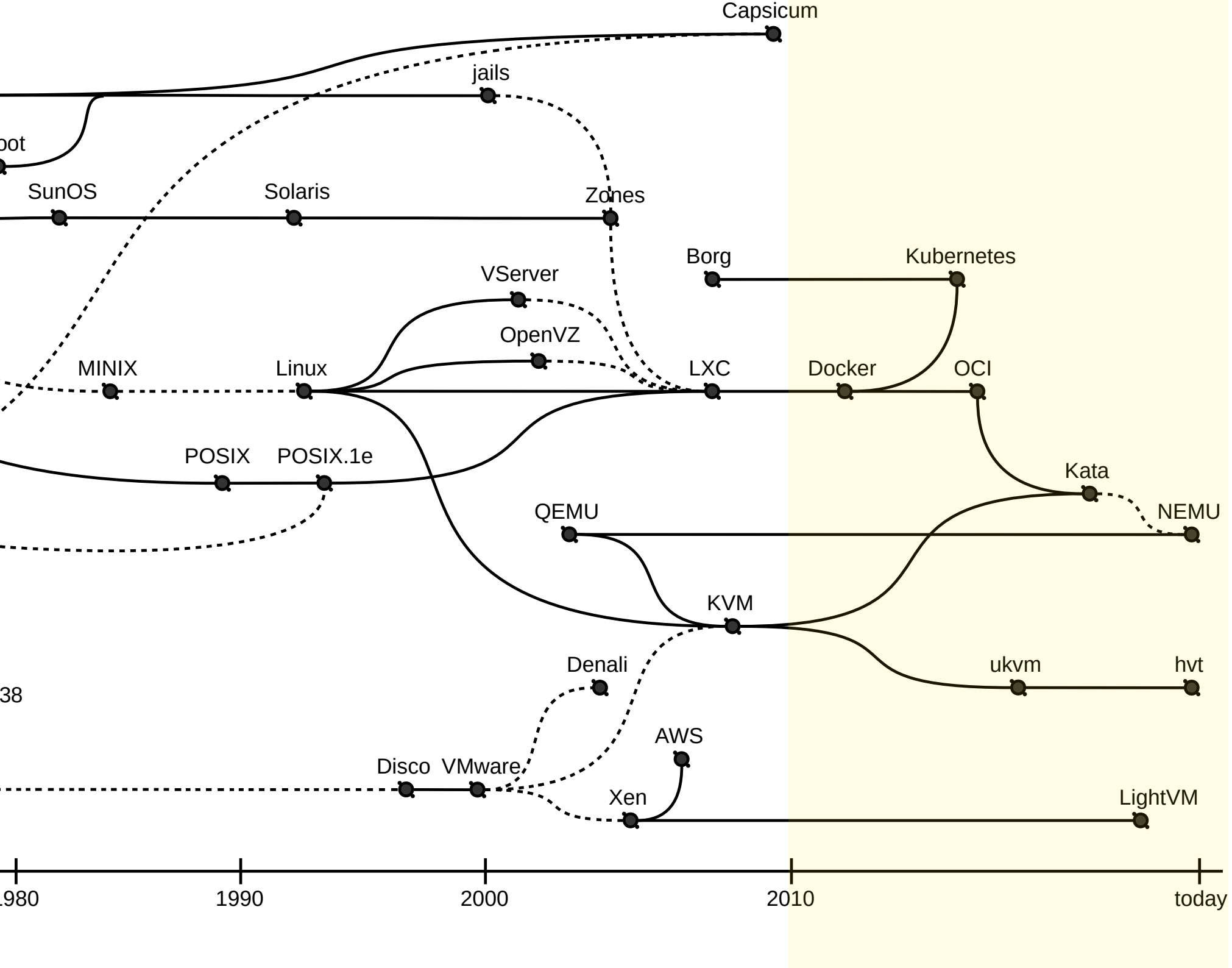
AWS availability zones, (C) 2016, Amazon.com, Inc. CC BY-SA 4.0

[1]A. Whitaker, M. Shaw, and S. Gribble. *Denali: Lightweight Virtual Machines for Distributed and Networked Applications*. Technical report, University of Washington, 2002.
[2]A. Whitaker, M. Shaw, and S. D. Gribble. Denali: A Scalable Isolation Kernel. *In Proceedings of the 10th Workshop on ACM SIGOPS European Workshop*, 10–15, 2002.
[3]P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the Art of Virtualization. *In Proceedings of the 19th ACM Symposium on Operating Systems Principles* (SOSP '03), 164–177, 2003.
[4]J. Barr. *Amazon EC2 Beta*. https://aws.amazon.com/blogs/aws/amazon_ec2_beta. 2006.
[5]J. S. Robin and C. E. Irvine. Analysis of the Intel Pentium's Ability to Support a Secure Virtual Machine Monitor. *In Proceedings of the 9th USENIX Security Symposium*, 129–144, 2000.
[6]A. Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguori. KVM: the Linux Virtual Machine Monitor. *In Proceedings of the 2007 Ottawa Linux Symposium*, 2007.

2000s

- Containers
  - FreeBSD Jails[1] & Solaris Zones[2] filesystem, process, network, resource limits
  - Linux VServer[3] and OpenVZ[4]
  - Linux namespaces[5] filesystem, process, IPC, network
  - Linux cgroups[6] resource/process control
  - LXC[7] cgroups, namespaces, capabilities
- Borg[8] workload orchestration

[1]P.-H. Kamp and R. N. M. Watson. Jails: Confining the omnipotent root. *In Proceedings of the 2nd International SANE Conference*, 2000.
[2]D. Price and A. Tucker. Solaris Zones: Operating System Support for Consolidating Commercial Workloads. *In Proceedings of the 18th USENIX Conference on System Administration (LISA '04)*, 241–254, 2004.
[3]S. Soltesz, H. Pötzl, M. E. Fiuczynski, A. Bavier, and L. Peterson. Container-based Operating System Virtualization: A Scalable, High-performance Alternative to Hypervisors. *In Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems*, 275–287, 2007.
[4]J. N. Matthews, W. Hu, M. Hapuarachchi, T. Deshane, D. Dimatos, G. Hamilton, M. McCabe, and J. Owens. Quantifying the Performance Isolation Properties of Virtualization Systems. *In Proceedings of the 2007 Workshop on Experimental Computer Science*, 2007.
[5]E. W. Biederman. Multiple instances of the global linux namespaces. *In Proceedings of the 2006 Ottowa Linux Symposium*, 1:101–112, 2006.
[6]J. Corbet. Process containers, *LWN*. https://lwn.net/Articles/236038/. 2007.
[7]Á. Kovács. Comparison of different Linux containers. *In 2017 40th International Conference on Telecommunications and Signal Processing*, 47–51, 2017.
[8]A. Verma, L. Pedrosa, M. Korupolu, D. Oppenheimer, E. Tune, and J. Wilkes. Large-scale Cluster Management at Google with Borg. In Proceedings of the Tenth European Conference on Computer Systems (EuroSys '15), 18:1–18:17, 2015.


Capsicum
jails
oot
SunOS
Solaris
Zones
Borg
Kubernetes
VServer
OpenVZ
MINIX
Linux
LXC
Docker
OCI
POSIX
POSIX.1e
Kata
QEMU
NEMU
KVM
Denali
ukvm
hvt
38
AWS
Disco
VMware
Xen
LightVM
1980
1990
2000
2010
today

2010s

- Containers
  - Docker[1] mass adoption
  - Linux user namespaces[2]
  - Kubernetes[3] workload orchestration

[1]Á. Kovács. Comparison of different Linux containers. *In 2017 40th International Conference on Telecommunications and Signal Processing*, 47–51, 2017.
[2]E. W. Biederman. Multiple instances of the global linux namespaces. *In Proceedings of the 2006 Ottowa Linux Symposium*, 1:101–112, 2006.
[3]E. A. Brewer. Kubernetes and the Path to Cloud Native. *In Proceedings of the 6th ACM Symposium on Cloud Computing*, 167–167, 2015.

Myths: VM performance

- ukvm[1] renamed to hvt
- LightVM[2] faster Xen
- NEMU[3] minimal QEMU

[1]D. Williams and R. Koller. Unikernel Monitors: Extending Minimalism Outside of the Box. *In 8th USENIX Workshop on Hot Topics in Cloud Computing* (HotCloud 16), 6, 2016.
[2]F. Manco, C. Lupu, F. Schmidt, J. Mendes, S. Kuenzer, S. Sati, K. Yasukata, C. Raiciu, and F. Huici. My VM is Lighter (and Safer) Than Your Container. *In Proceedings of the 26th Symposium on Operating Systems Principles* (SOSP '17), 218–233, 2017.
[3]https://github.com/intel/nemu

Myths: container security

- Kata Containers[1] (was Intel Clear Containers[2])
  - QEMU+KVM
- gVisor[3]
  - kernel
  - devices
  - syscall filtering
- Depends on kernel security[4] [5] and “self-protection”[6]

[1]https://katacontainers.io/
[2]A. van de Ven. An introduction to Clear Containers. *LWN*. https://lwn.net/Articles/644675/. 2015.
[3]https://github.com/google/gvisor
[4]E. Reshetova, J. Karhunen, T. Nyman, and N. Asokan. Security of OS-Level Virtualization Technologies. *Secure IT Systems*, *Lecture Notes in Computer Science*, 77–93. Springer, 2014.
[5]X. Gao, Z. Gu, M. Kayaalp, D. Pendarakis, and H. Wang. ContainerLeaks: Emerging Security Threats of Information Leakages in Container Clouds. *In 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. 237–248, 2017.
[6]S. Bratus, M. E. Locasto, A. Ramaswamy, and S. W. Smith. VM-based Security Overkill: A Lament for Applied Systems Security Research. *In Proceedings of the 2010 New Security Paradigms Workshop*, 51–60, 2010.

Myths: VM security

- Lines of code only vague potential for security[1] [2]
- Attack vectors[3]
  - *source*: VM guest (Xen 71%, KVM 66%)
  - *target*: Ring -1, Dom0, host (Xen 80%, KVM 76%)
- Instruction emulation, arbitrary, unfiltered[4]
- Depends on kernel security[5] and "self-protection"[6]

[1]M. Pearce, S. Zeadally, and R. Hunt. Virtualization: Issues, security threats, and solutions. *ACM Computing Surveys*, 45(2):1–39, Feb. 2013.
[2]D. Williams, R. Koller, and B. Lum. Say Goodbye to Virtualization for a Safer Cloud. *In 10th USENIX Workshop on Hot Topics in Cloud Computing* (HotCloud 18), 2018.
[3]D. Perez-Botero, J. Szefer, and R. B. Lee. Characterizing Hypervisor Vulnerabilities in Cloud Computing Servers. In Proceedings of the 2013 International Workshop on Security in Cloud Computing, 3–10, 2013.
[4]K. Ishiguro and K. Kono. Hardening Hypervisors Against Vulnerabilities in Instruction Emulators. *In Proceedings of the 11th European Workshop on Systems Security* (EuroSec'18), 7:1–7:6, 2018.
[5]F. Lombardi and R. Di Pietro. Secure virtualization for cloud computing. *Journal of Network and Computer Applications*, 34(4):1113–1122, July 2011.
[6]S. Bratus, M. E. Locasto, A. Ramaswamy, and S. W. Smith. VM-based Security Overkill: A Lament for Applied Systems Security Research. *In Proceedings of the 2010 New Security Paradigms Workshop*, 51–60, 2010.

Myths: VM security

- Separate kernel mitigates some classes of vulnerabilities
- Speculative execution vulnerabilities
  - Spectre, NetSpectre[1] [2]
  - Meltdown[3]
  - Foreshadow, L1TF[4] [5]

Spectre, Meltdown, and Foreshadow icons, (C) 2018, Natascha Eibl, CC0

[1] P. Kocher, D. Genkin, D. Gruss, W. Haas, M. Hamburg, M. Lipp, S. Mangard, T. Prescher, M. Schwarz, and Y. Yarom. Spectre Attacks: Exploiting Speculative Execution. *arXiv:1801.01203* [cs], Jan. 2018.
[2] M. Schwarz, M. Schwarzl, M. Lipp, and D. Gruss. NetSpectre: Read Arbitrary Memory over Network. *arXiv:1807.10535* [cs], July 2018.
[3] M. Lipp, M. Schwarz, D. Gruss, T. Prescher, W. Haas, S. Mangard, P. Kocher, D. Genkin, Y. Yarom, and M. Hamburg. Meltdown. arXiv:1801.01207 [cs], Jan. 2018.
[4] J. Van Bulck, M. Minkin, O. Weisse, D. Genkin, B. Kasikci, F. Piessens, M. Silberstein, T. F. Wenisch, Y. Yarom, and R. Strackx. Foreshadow: Extracting the Keys to the Intel SGX Kingdom with Transient Out-of-Order Execution. *In 27th USENIX Security Symposium*, 991–1008, Baltimore, Aug. 2018.
[5] O. Weisse, J. V. Bulck, M. Minkin, D. Genkin, B. Kasikci, F. Piessens, M. Silberstein, R. Strackx, T. F. Wenisch, and Y. Yarom. *Foreshadow-NG: Breaking the Virtual Memory Abstraction with Transient Out-of-Order Execution*. Technical report, Aug. 2018.

Lasciate ogne speranza,
voi ch'intrate

–Dante Alighieri, “Inferno”

(Common translation: Abandon all hope, ye who enter here)

Positive directions

- Capabilities
  - Capsicum[1]
  - CHERI[2]
  - Fuchsia[3]
- Hardware
  - RISC-V[4]
  - Open Titan[5]
- OS
  - OpenBSD pledge[6], unveil[7]

DE4 prototype tablet computer running CHERI, origin unknown, https://www.cl.cam.ac.uk/research/comparch/opensource/de4tablet/tablet-booting-cheri.jpg

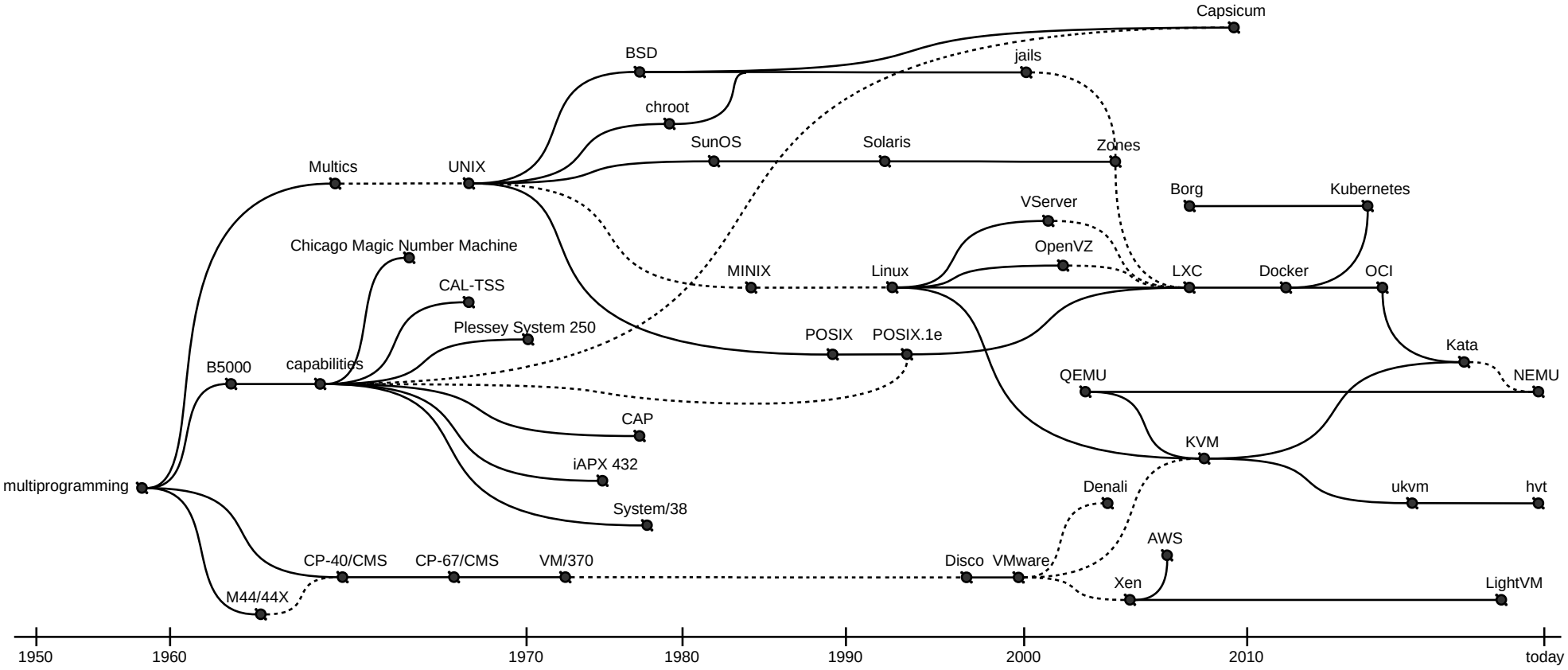
[1]R. Watson, J. Anderson, B. Laurie, and K. Kennaway. Capsicum: Practical Capabilities for UNIX. *In Proceedings of the 19th USENIX Security Symposium*. 2010.
[2]J. Woodruff, R. N. Watson, D. Chisnall, S. W. Moore, J. Anderson, B. Davis, B. Laurie, P. G. Neumann, R. Norton, and M. Roe. The CHERI Capability Model: Revisiting RISC in an Age of Risk. *In Proceedings of the 41st Annual International Symposium on Computer Architecuture*, 457–468, 2014.
[3]Google. *Fuchsia is not Linux: A modular, capability-based operating system*. https://fuchsia.googlesource.com/docs/+/HEAD/the-book/README.md.
[4]K. Asanović and D. A. Patterson. *Instruction Sets Should Be Free: The Case For RISC-V*. Technical Report UCB/EECS-2014-146, University of California, Berkeley, Aug. 2014.
[5]D. Rizzo and P. Ranganathan. Titan: Google's Root-of-Trust Security Silicon. *In Proceedings of the IEEE Hot Chips Symposium*, Aug. 2018.
[6]pledge(2) manpage, https://man.openbsd.org/pledge.2
[7]unveil(2) manpage, https://man.openbsd.org/unveil.2

Future directions

- Reexamine the full stack: hardware, kernel, OS, hypervisor/containers, guest, application workloads
- Synthesis: architecture/systems/security

Questions?

Futuristic data center, origin unknown, https://on.rt.com/lu029w


Capsicum
BSD
jails
chroot
SunOS
Solaris
Zones
Multics
UNIX
Borg
Kubernetes
VServer
Chicago Magic Number Machine
OpenVZ
MINIX
Linux
LXC
Docker
OCI
CAL-TSS
Plessey System 250
POSIX
POSIX.1e
Kata
B5000
capabilities
QEMU
NEMU
CAP
KVM
iAPX 432
multiprogramming
Denali
ukvm
hvt
System/38
AWS
CP-40/CMS
CP-67/CMS
VM/370
Disco
VMware
Xen
LightVM
M44/44X
1950
1960
1970
1980
1990
2000
2010
today