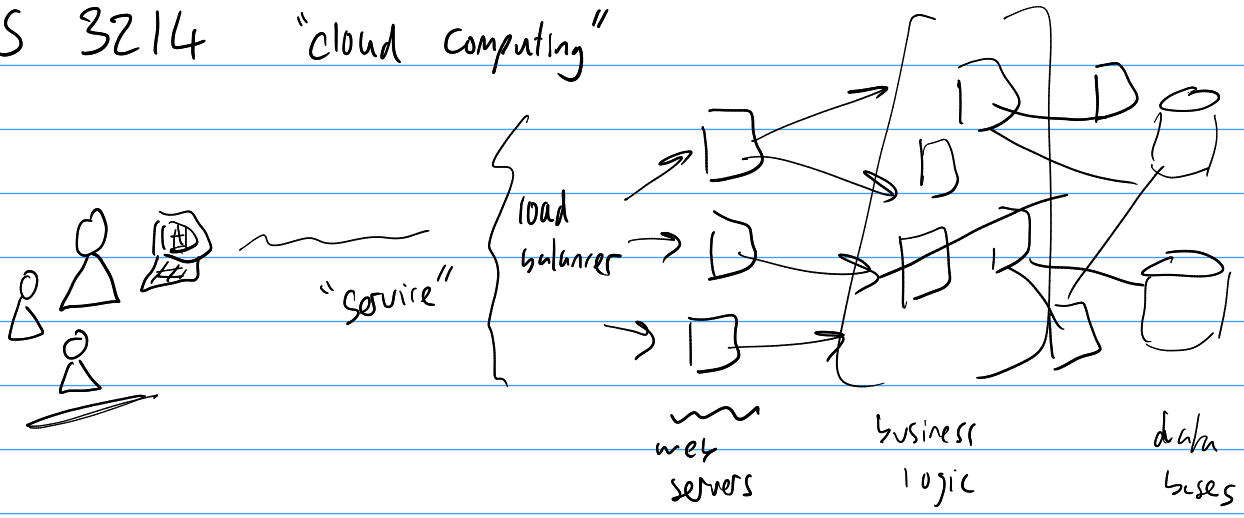
CS 3214 "cloud computing"



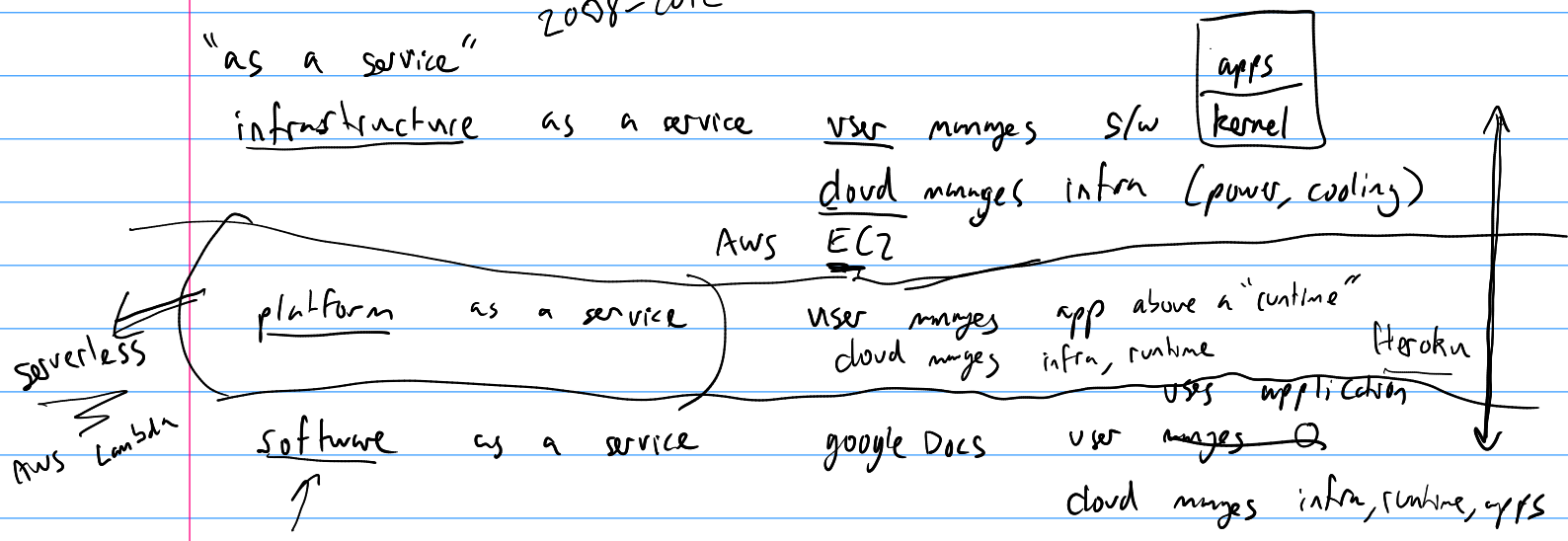
"Slashdot effect"

Elasticity (auto scaling)

2008-2012 NIST

"as a service"

infrastructure as a service — user manages S/W; cloud manages infra (power, cooling) — AWS EC2



3 characteristics

- elasticity: ↑ ↓ "instances" only pay for what you use
  - Scale granularity changed over time
- multitenancy → high utilization (statistical multiplexing)

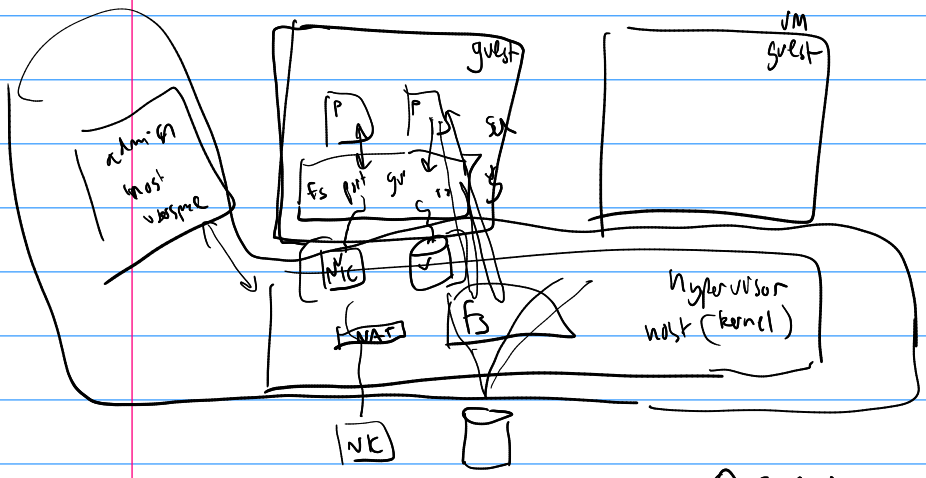
→ need isolation

users want "private view" of system

tenants should not interfere with each other

Process?
+ private address space
app dependencies — libraries?
will they interfere? YES
- ports shared
- fs shared /etc/configs

Not really "application" abstraction
→ "bare metal"
→ Virtual machines
→ Containers



1970s
Popek & Goldberg req's
trap-and-emulate
- equivalent fidelity
- resource control
- efficient (most instructions should run directly on CPU)

QEMU — emulator
↳ KVM acceleration

1999 Vmware binary translation
2002 Xen paravirtualization
2005/2006 H/W support VT-x
Intel VMX
AMD SVM

"Nested virtualization"

A lot of OS stuff needs to be recreated in VM world

guest virtual
↓
guest physical
↓
host physical

Containers: we are already doing it!

use kernel mechanism

FS: chroot "jail"

using this directory for my '/' and all of my children

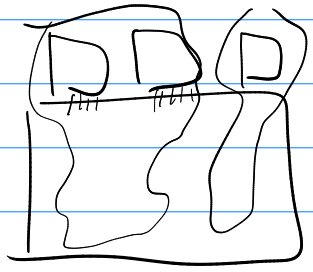
Namespaces

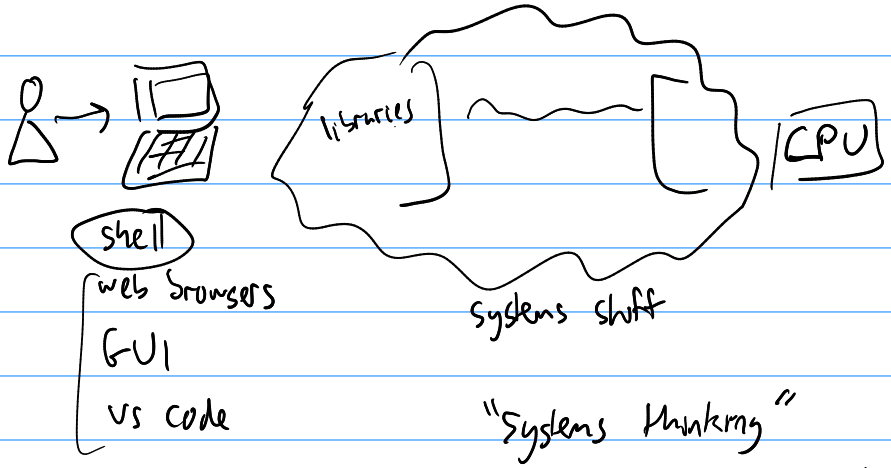
- UTS
- PID
- mount
- user
- IPC
- network

cgroups

limit resource consumption

CPU, mem





"Systems thinking"

experiencing getting deep into how things work

→ you can go deep on ANYTHING

become a super programmer

make systems better for everyone