

CS 162 Spring 2007 Midterm 3 (Final) Review

Thomas Kho
tkho at eecs.berkeley.edu

2007-05-03

Based (heavily) on review by
Karl Chen
Adrian Mettler

Administrivia

- Midterm 3
 - Monday, May 7 at 7pm in 306 Soda
- No class on Monday
- Evaluations!

Metacomments

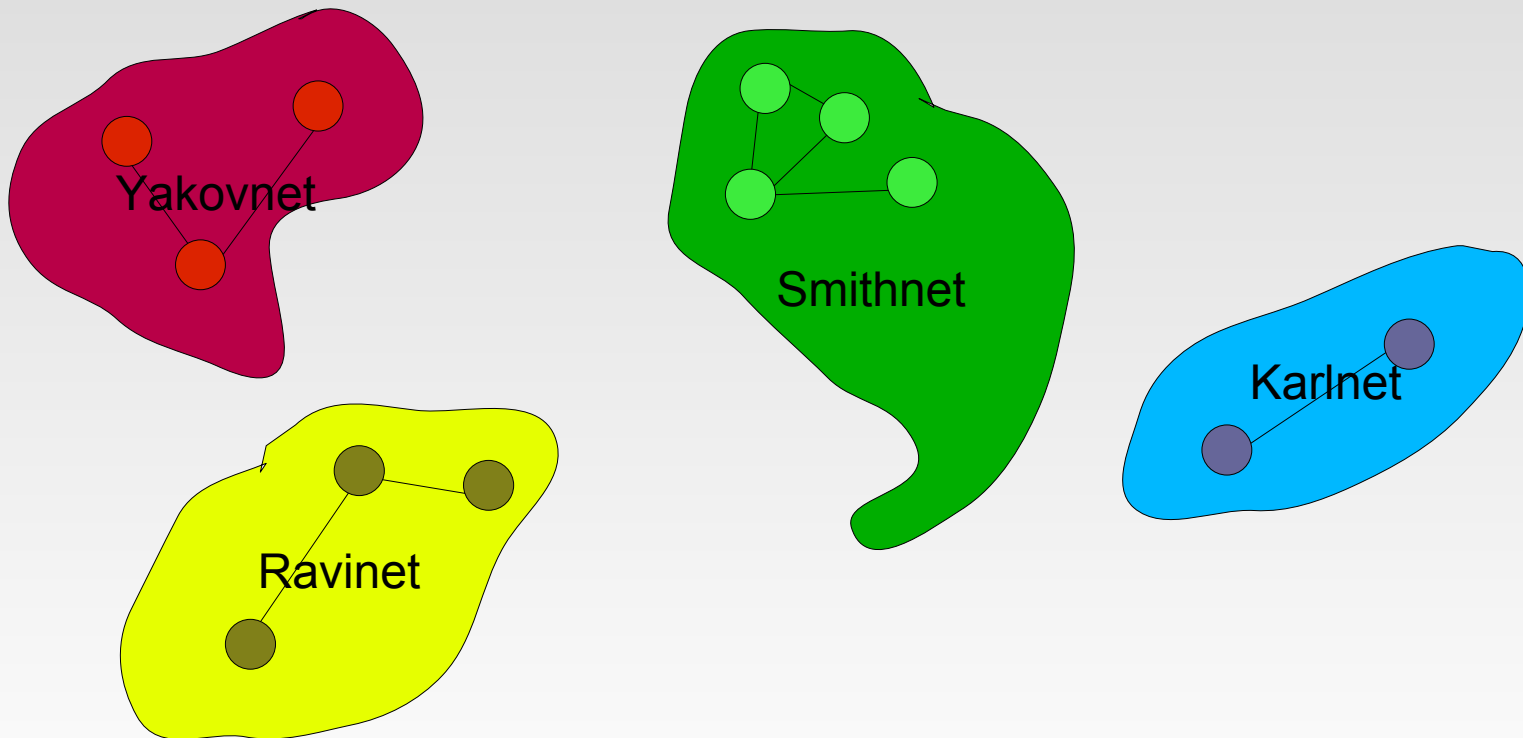
- Entire semester is covered
- Smith's lecture notes are authoritative

Agenda

- Post-midterm 2 material
 - Networking, Protocols
 - Distributed systems
 - Security, Protection
 - Cryptography
 - Virtual machines
- Entire-semester review
 - Sample questions

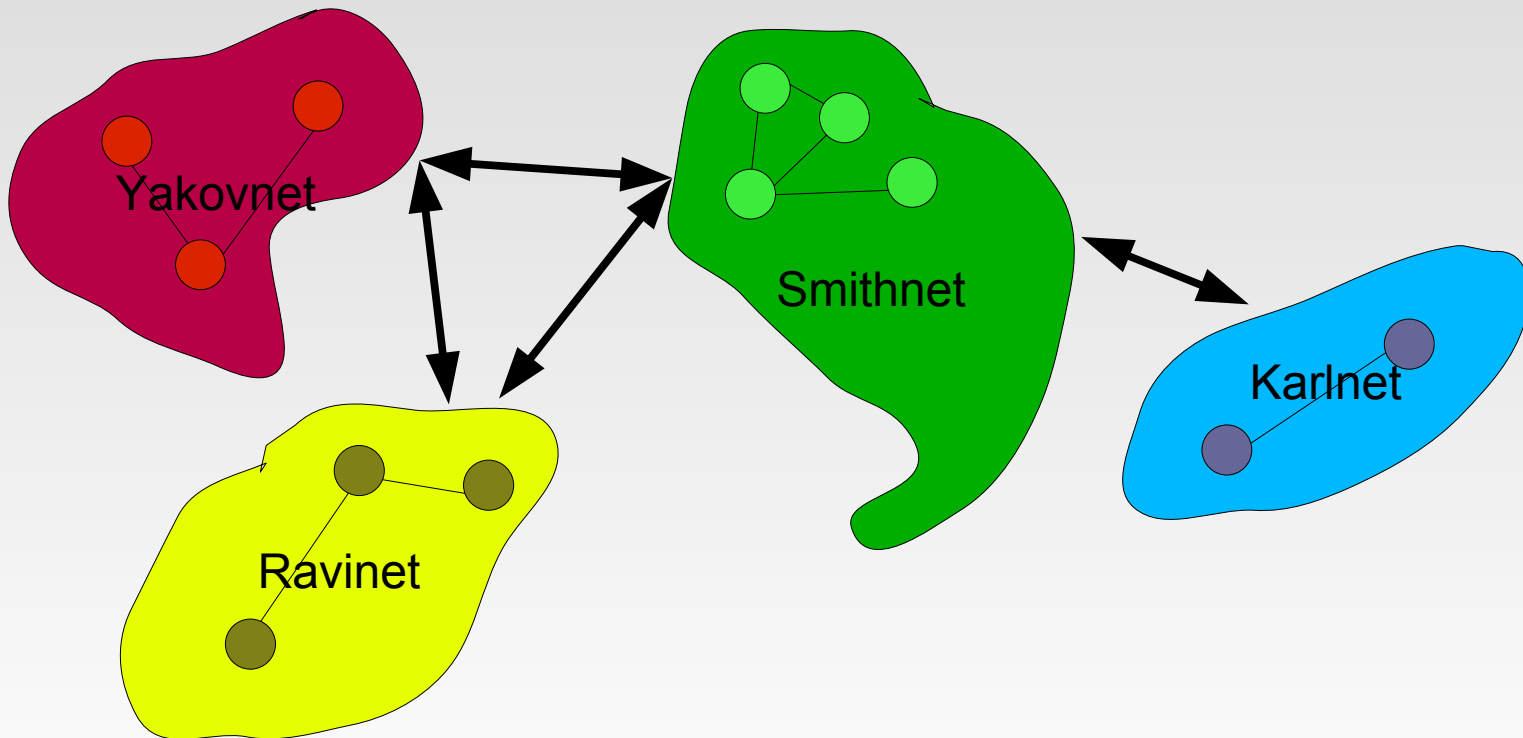
Networking

- Once upon a time: individual LANs



Networking

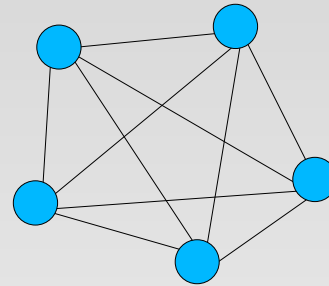
- Why not inter-network them together -> WAN



Networking

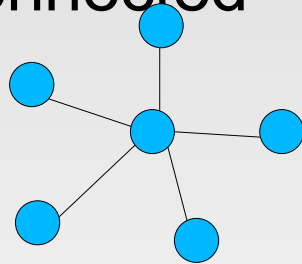
- Topology

- Fully-connected

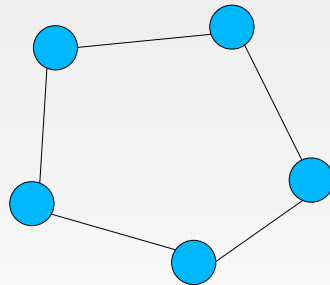


- Partially-connected

- Star



- Ring



- Intermediate nodes (routers) forward messages

Networking

- Broadcasting
 - Radio (Wifi)
 - Ethernet: Hubs (compare switches)
 - Aloha net
 - Can everyone talk at once?
 - Carrier sense
 - Collision detection

Networking

- LANs
 - Ethernet
 - Token-ring
- WANs
 - Circuit switching
 - Packet switching

Networking

- Performance parameters
 - Latency
 - Bandwidth
- Example: DSL
 - Latency = 20 ms
 - Bandwidth = 80 Kbyte/s

Networking

- OSI layers
 - Physical
 - Data link
 - Network
 - Transport
 - Session
 - Presentation
 - Application
- Where do these go:
 - IP, TCP, UDP, HTTP, SMTP

Networking

- Networks are sparsely connected
- Network layer (e.g. IPv4)
 - Addressing
 - Route packets to destination

Networking

- Network links are unreliable
 - Dropped packets
 - Congestion
- Transport layer (e.g. TCP)
 - Reliability
 - Flow control
 - Congestion control
 - Ports

Distributed Systems

- So far, loose coupling:
 - Each machine autonomous
 - Separate accounting, file system, etc.
 - Can send messages, transfer files
 - Can login, execute command remotely
 - Users have to deal with load balancing, process/file migration, etc.

Distributed Systems

- Design goals
 - Unified, transparent file system
 - Unified, transparent computation

Distributed Systems

- Distributed file systems
 - NFS, AFS, SMB, CIFS
 - Use as regular file system – run programs, etc.
 - NFS: mamba.cs.berkeley.edu
 - SMB/CIFS: Windows shares
 - Issues:
 - Reliability
 - Performance
 - Consistency

Security

- Authentication:
 - Who are you?
- Authorization:
 - What are you allowed to do?

Security

- Authentication
 - Passwords
 - Public-keys
 - Photo badge, secret handshake

Security

- Authorization
 - solutions.txt access matrix:

	Read	Write	Execute
cs162-ta	y	y	n
cs162-tb	y	y	n
cs162-ra	y	n	n
cs162-rb	y	n	n
cs162-rc	y	n	n
cs162-aa	n	n	n
cs162-ab	n	n	n
cs162-ac	n	n	n

Security

- Authorization

- solutions.txt access control list:

	Read	Write	Execute
cs162-ta	y	y	n
cs162-tb	y	y	n
cs162-ra	y	n	n
cs162-rb	y	n	n
cs162-rc	y	n	n
cs162-aa	n	n	n
cs162-ab	n	n	n
cs162-ac	n	n	n

- read: cs162-`{ta,tb,ra,rb,rc}`
- write: cs162-`{ta,tb}`
- execute: (none)

Security

- Authorization
 - Capability list:

	Read	Write	Execute
cs162-ta	y	y	n
cs162-tb	y	y	n
cs162-ra	y	n	n
cs162-rb	y	n	n
cs162-rc	y	n	n
cs162-aa	n	n	n
cs162-ab	n	n	n
cs162-ac	n	n	n

- cs162-ta: read solutions.txt, write solutions.txt
- cs162-ra: read solutions.txt
- cs162-aa: (none)

Security

- Confinement problem
 - Don't allow Java applet to delete My Documents
 - Don't allow Java applet to send My Documents, applet author to third-party, or even another program on my computer
- Covert channels
 - E.g. Morse code of coughs and throat-clears to cheat on exam
 - → confinement problem is very difficult

Security

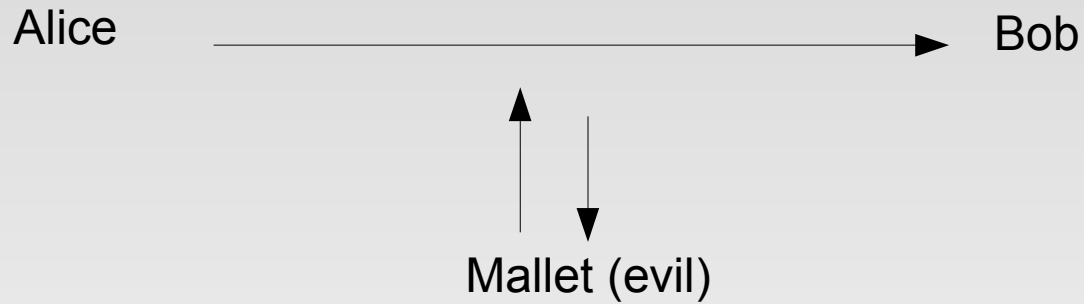
- Design principles
 - Economy of mechanism – KISS
 - Fail safe defaults: access decisions must be based on permission, not exclusion
 - Complete mediation: every access to every object must be checked
 - Open design: obscurity is NOT security
 - Separation of privileges – distribute your trust

Security

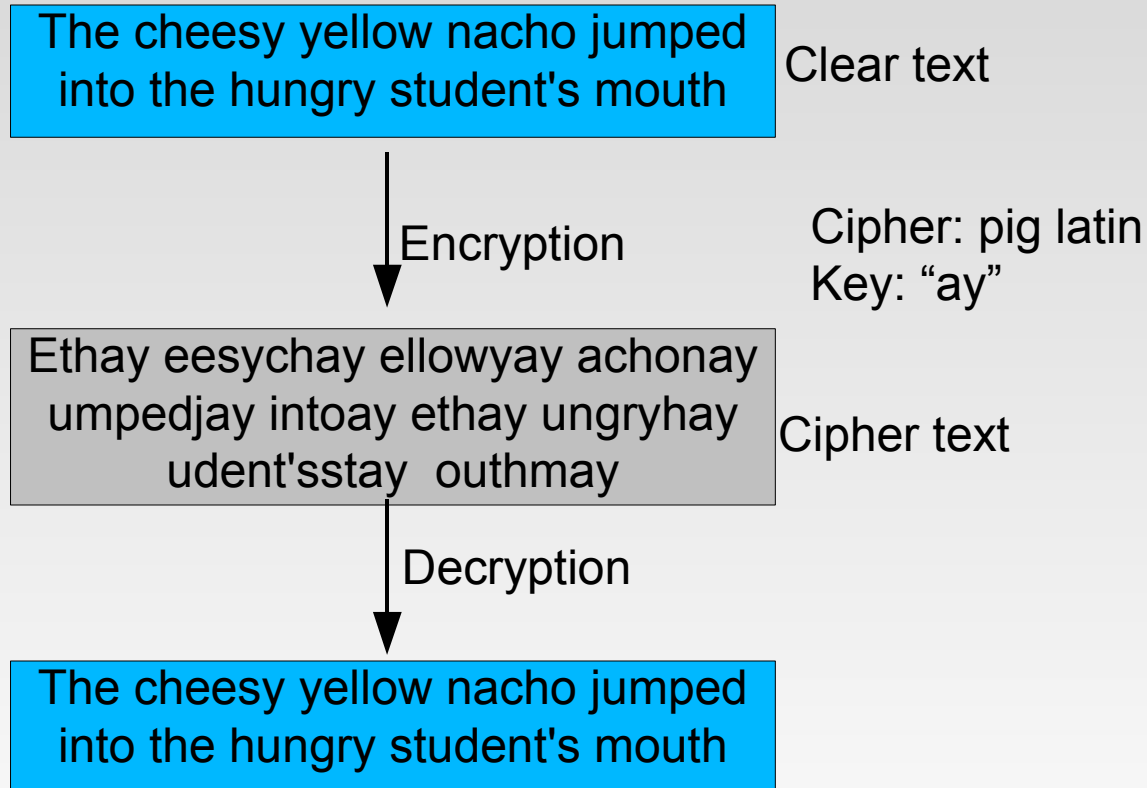
- Design principles (continued)
 - Least privilege – give no more than required access
 - Least common mechanism – the system is only as secure as the weakest link
 - Psychological acceptability – no protection if no one wants to use the mechanism

Security

- Adversaries



Cryptography

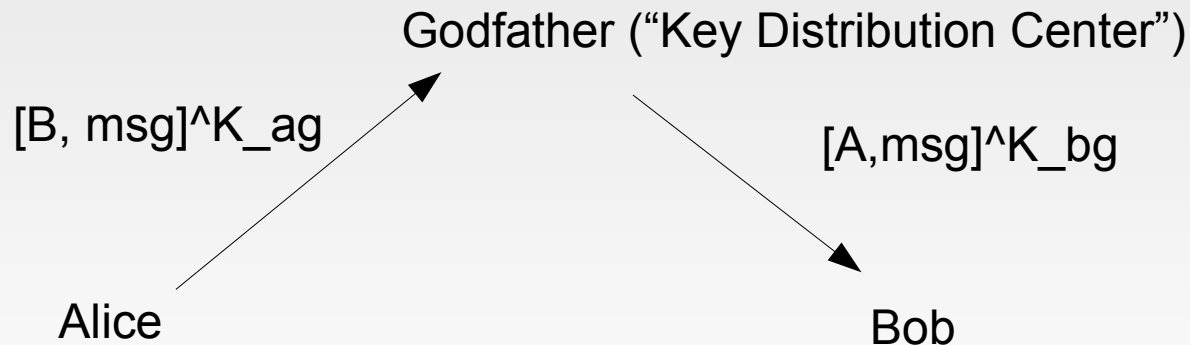


Cryptography

- Old-school ciphers:
 - Caesar cipher
 - Substitution
 - Transposition
 - Polyalphabetic
 - Running key
 - Codes
- Modern symmetric ciphers:
 - 3DES, Blowfish, AES

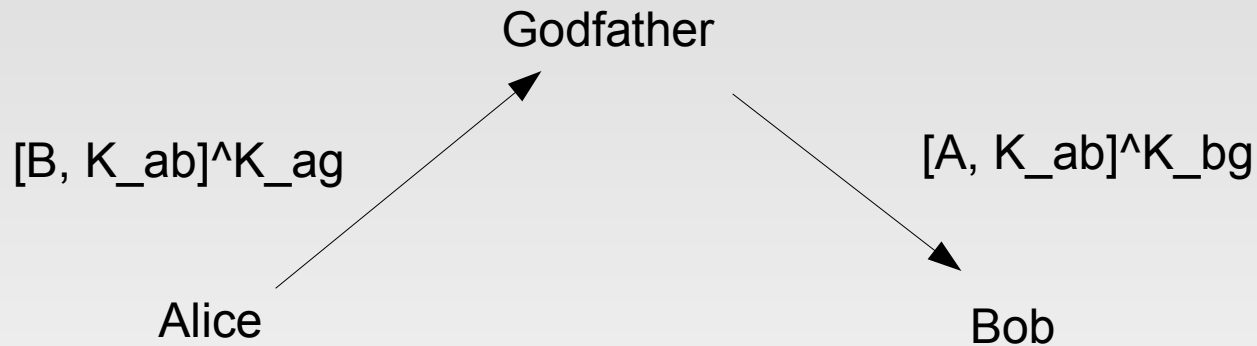
Cryptography

- Key distribution for symmetric keys (Kerberos)
 - Alice to Godfather: {tell Bob “I'm taking final on Monday”} (encrypted w/ “AliceRulez”)
 - Godfather to Bob: {Alice says “I'm taking final on Monday”} (encrypted w/ “BobRulez”)

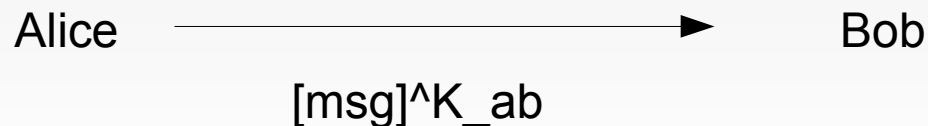


Cryptography

- Key distribution (continued)
 - Alice to Godfather: {tell Bob “temp password is NachozRulez”} (encrypted w/ “AliceRulez”)

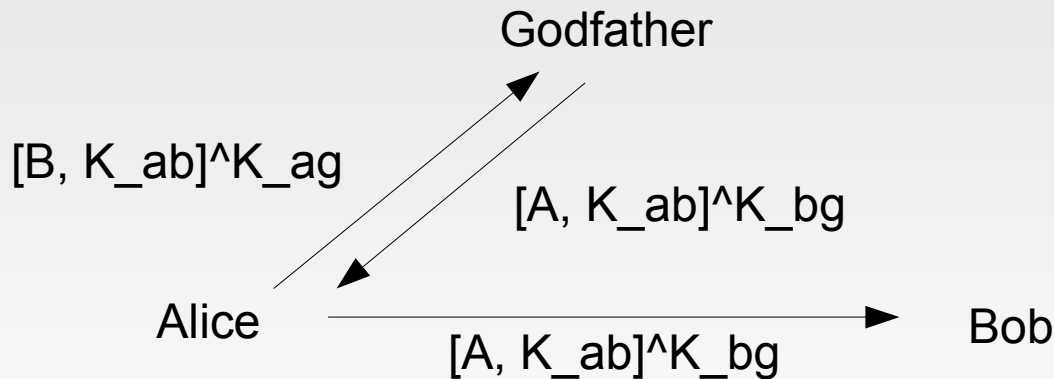


- Alice to Bob: {I'm taking final Monday} (encrypted w/ “NachozRulez”)



Cryptography

- Key distribution (continued)
 - Alice to Godfather: {tell Bob “new password is NachozRulez”} (encrypted w/ “AliceRulez”)
 - Godfather to Alice: {tell Bob “new pw...”} (encrypted w/ “BobRulez”)



Cryptography

- Asymmetric (public-key) encryption
 - encryption key \neq decryption key
 - Encryptor and decryptor don't need to share secret
 - $[\text{clear text}]^{\text{pubkey}} = \text{cipher text}$
 - $[\text{cipher text}]^{\text{privkey}} = \text{clear text}$
- Signatures
 - $[\text{clear text}]^{\text{privkey}} = \text{signature}$
 - if $([\text{signature}]^{\text{pubkey}} == \text{clear text})$: unmodified

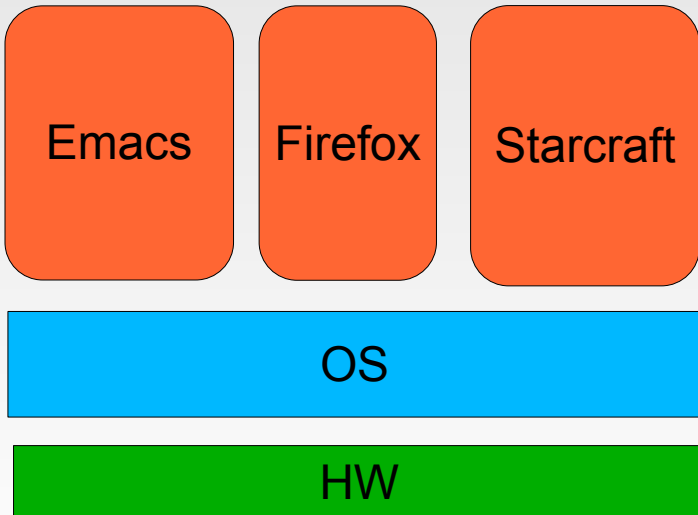
Cryptography

- Cryptographic hash functions
 - `sha1sum("I love nachos") = f3428bd551822152524f956ce7d4a7a766a42b38`
 - No reverse function
- Passwords:
 - Store `hash(password)` instead of password
 - Salt
- Signatures:
 - Sign `hash(message)` instead of message

Virtual Machines

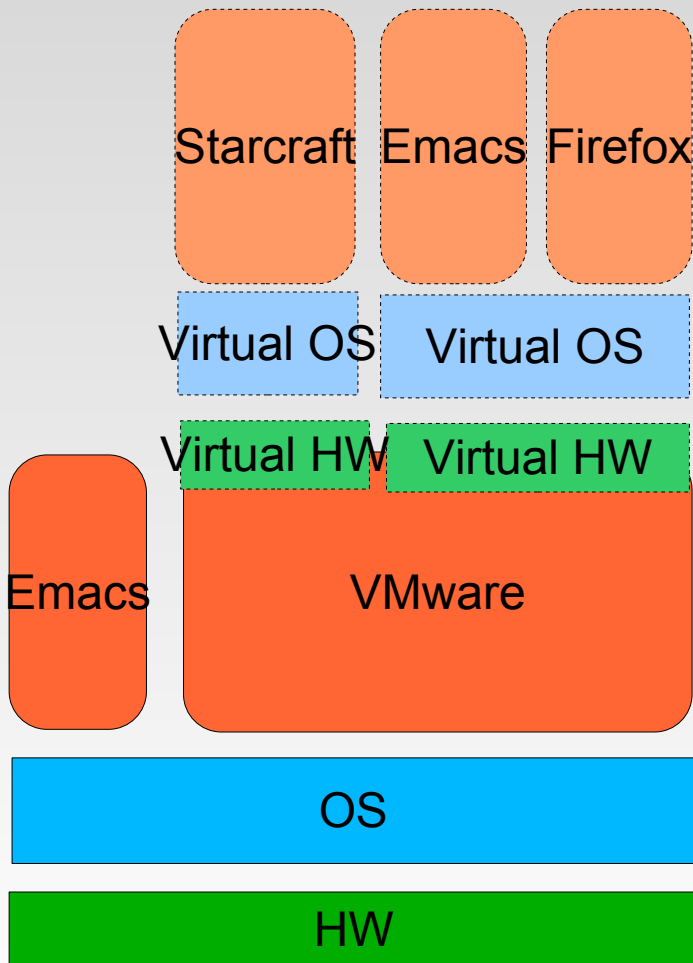
- Regular OS model

- Run OS on HW
- Run user programs on OS



Virtual Machines

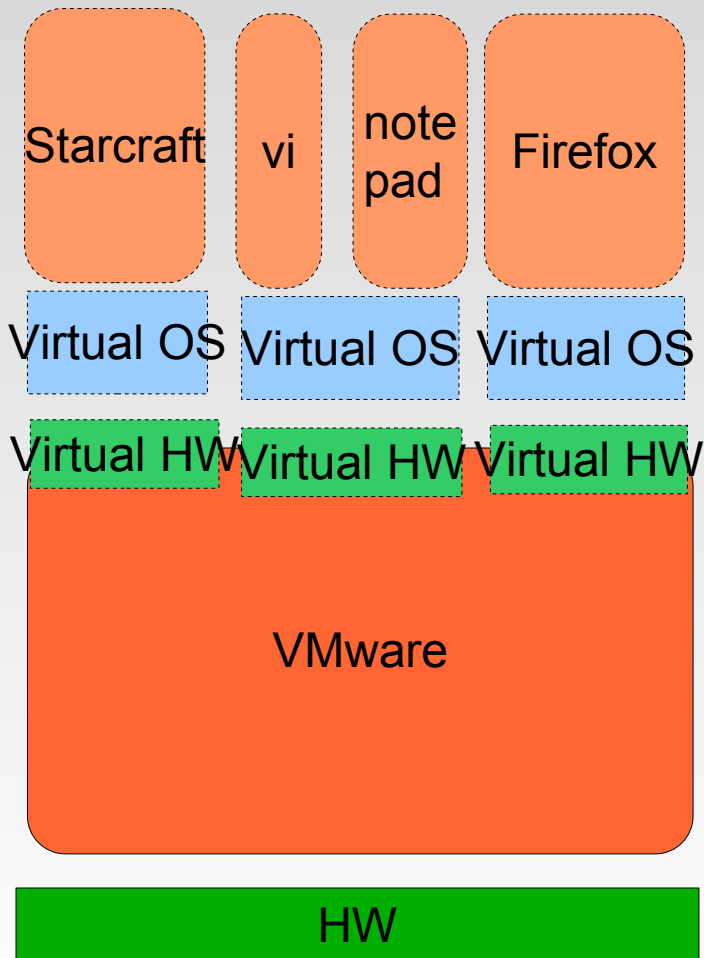
- Virtual machine model (1)



- Run user programs on OS
- Run virtual machine monitor as user program
- Run virtual OS on virtual HW
- Run virtual user programs on virtual OS

Virtual Machines

- Virtual machine model (2)



- Run virtual machine monitor on HW
- Run virtual OS on virtual HW
- Run virtual user programs on virtual OS

Virtual Machines

- Page tables:
 - Two mappings:
 - Virtual memory -> physical memory
 - Virtual virtual memory -> Virtual physical memory (virtual memory)
 - Compose them:
 - Virtual virtual memory -> physical memory