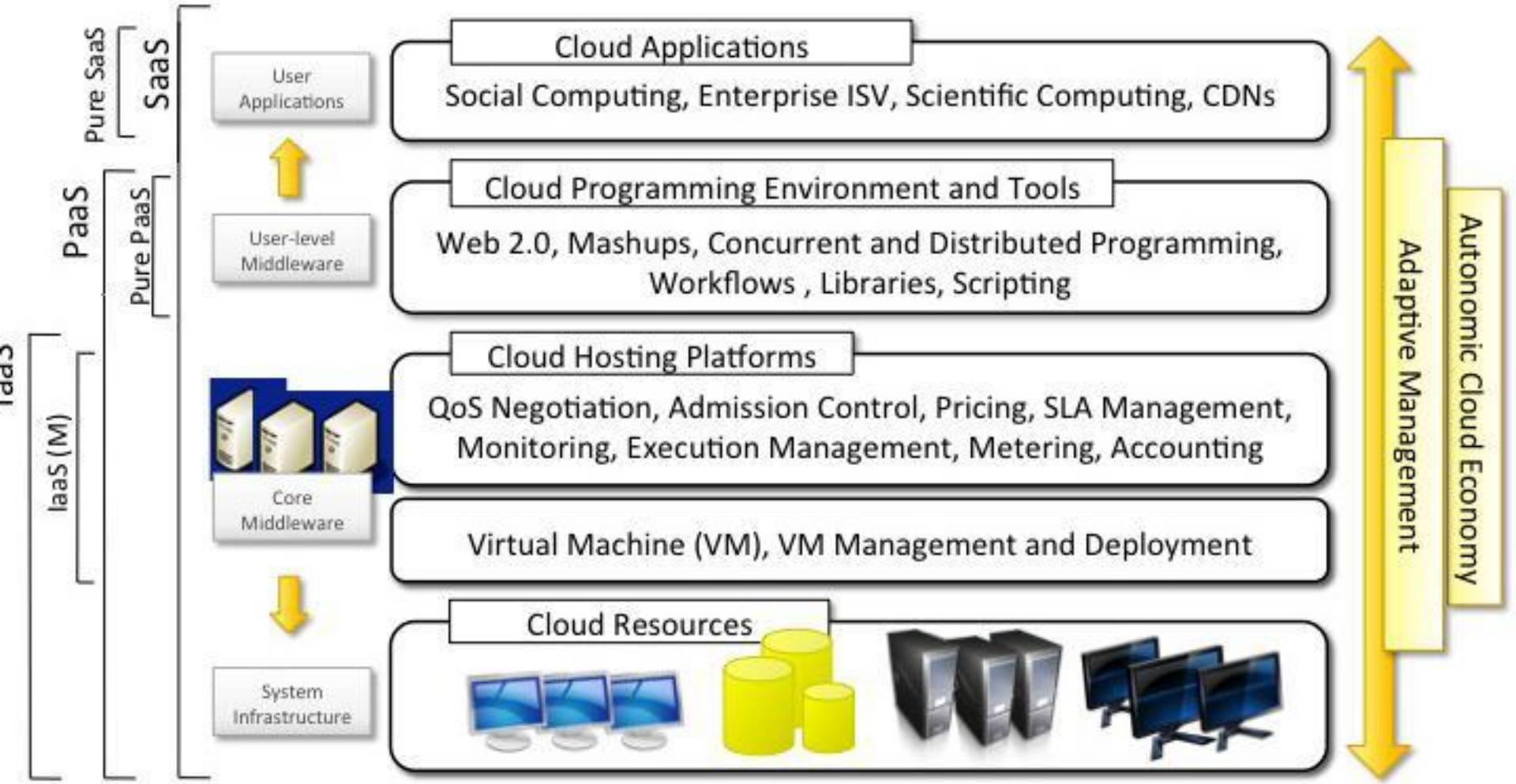
Chapter 4
Cloud Computing Architecture

Mastering Cloud Computing

Coleman Kane
(based on material by Paul Talaga)

UNIVERSITY OF Cincinnati

Layered View



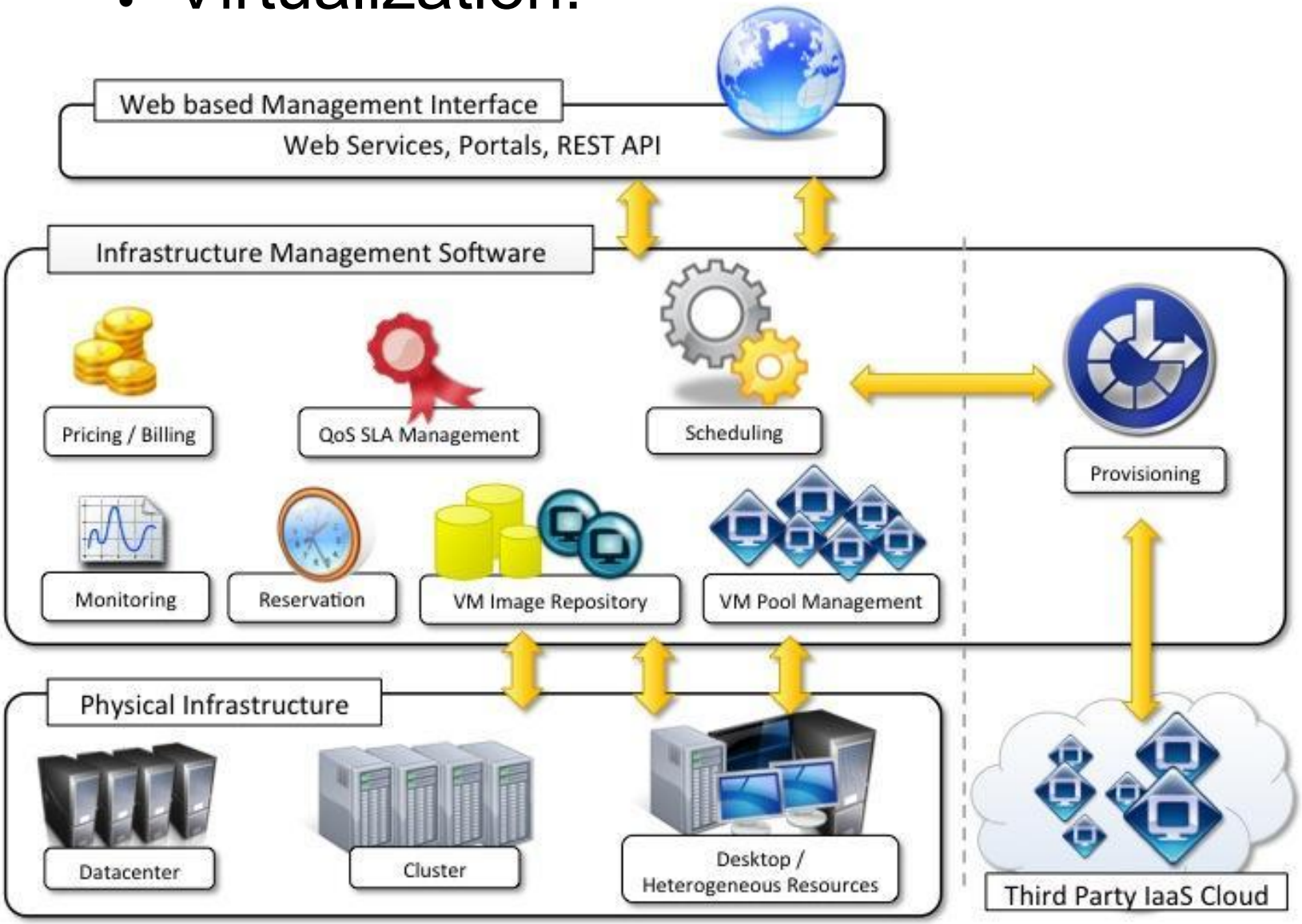
Categories

- IaaS - Infrastructure as a Service (AWS)
  - IaaS(M) - Only management layer
- PaaS - Platform as a Service (AppEngine) Use APIs provided to build
  - Pure PaaS - Only middleware, bring hardware
- SaaS - Software as a Service (Gmail) - Adaptively scaling to demand

UNIVERSITY OF Cincinnati

IaaS (HaaS)

- Virtualization!

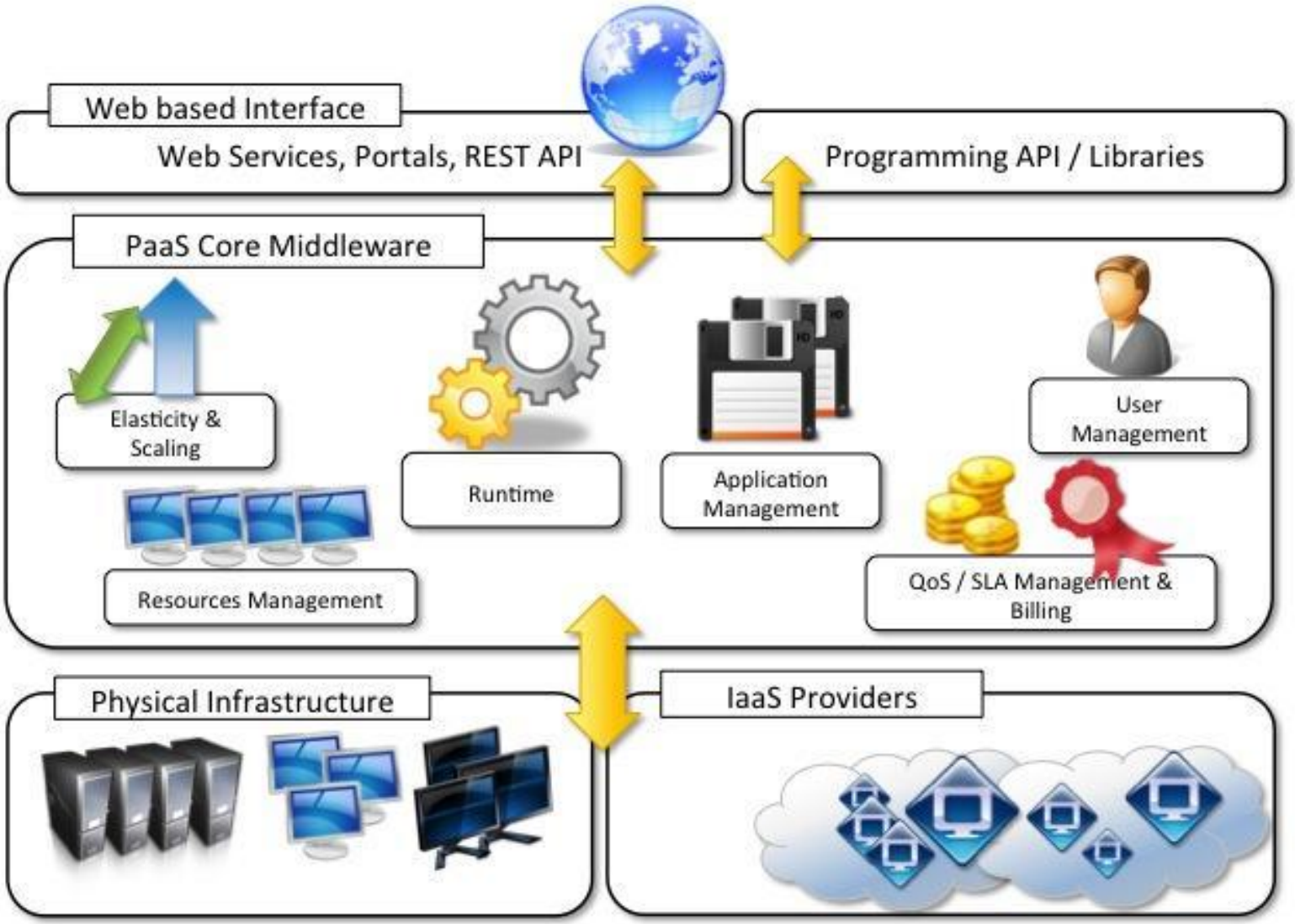


IaaS

- Customers are provided with virtualized hardware and storage on top of which they can build their infrastructure
- Ex: Amazon EC2 & S3
  - DataPipe
  - Microsoft Azure VMs
  - Samsung Joyent, etc…. (many)
- IaaS(M) - manage cloud infrastructure
  Ex: Enomaly
  - Eucalyptus
  - OpenNebula
  - OpenStack
  - CloudStack

UNIVERSITY OF Cincinnati

PaaS



Cincinnati

PaaS

- Application Management!
  - No ability/need to manage infrastructure
- Support Java, .NET, Python, Ruby, etc..
  - Some restrictions for scalability (no write fs)
- Some provide a local dev environment
- Subcategories:
  - PaaS-I : Rapid prototyping - Web 2.0 dev - Ex: Force.com, Longjump
  - PaaS-II : Runtime environment for web - upload code, use their APIs - Ex: Google AppEngine, AppScale, Heroku, Engine Yard

PaaS

- Subcategories (cont)
  - PaaS III - General purpose apps - Ex: MS Windows Azure (hw as well), Aneka, Apprenda SaaSGrid, Cloud IQ
- Characteristics of PaaS:
  - Runtime Framework - software stack
  - Abstraction - manage software, not HW
  - Automation - Auto scale resources
  - Cloud Services - API Ex: Storage, identity management, computation

UNIVERSITY OF Cincinnati

PaaS

- Issues….
  - Vendor lock-in
    - Runtime environment: Force.com
      - AppEngine: standard, Python, PHP, Java…
    - APIs may be proprietary
- Cost benefit
  - Reduces management costs for consumer
  - Vendor does hardware/software updates -> saves *them* money
- Fueling startup culture!
  - Don’t need to be an infrastructure expert

UNIVERSITY OF Cincinnati

SaaS - Software as a Service

- Access to software from web browser
  - No software to install, no upfront costs
- Good for ‘one size fits all’ systems - little customization needed
- Good fits:
  - Customer relationship management (CRM)
  - Enterprise resource planning (ERP)
- Pay-as-you-go (PAYG) pricing

UNIVERSITY OF Cincinnati

SaaS

- Multitenancy - multiple customers - same system
  - Optimized resources
  - Continuous upgrades
  - Distribute costs
  - Developers know how to run the best
  - Significant cost savings - no distribution, little support needed
  - Rapid implementation

UNIVERSITY OF Cincinnati

SaaS

- Very successful delivery model!
- SaaS 2.0 - Integrate reliable services
  - Use SLAs
  - Adds another dimension & options
  - Could promote open market
- Plugins becoming popular - extend
  - Facebook, LinkedIn
- Office automation
  - Google Documents, Zoho Office, etc

UNIVERSITY OF Cincinnati

Types of Clouds (review)

- Public
  - EC2
- Private
  - Customer information protection
  - Infrastructure ensuring SLA
  - Compliance w/ procedures and operations
- Hybrid or heterogeneous
  - Cloudbursting - using public cloud for spikes
- Community
  - Media Industry, Healthcare, Energy, Scientific

UNIVERSITY OF Cincinnati

Cloud Economics

- Why Cloud Computing?
  - Reduce capital IT costs
  - Eliminate depreciation
  - Replace software license with subscription
  - Reduce maintenance & admin costs of IT

UNIVERSITY OF Cincinnati

Open Challenges

- Defining Cloud Computing!!!!
- Cloud Interoperability
- Standards
- Scalability & Fault Tolerance
- Security, trust, privacy
- Changing role of IT in a business
  - What does a ‘new’ IT dept do?
  - Compliance?
  - Loss of control of data?

UNIVERSITY OF Cincinnati

Summary

- Service models: SaaS, PaaS, IaaS
- Deployment models: Public, Private, Hybrid, Community
- Driving force behind cloud --> $$$$
- Issues & Challenges

UNIVERSITY OF Cincinnati