Cloudinomicon ::

Idempotent Infrastructure, Survivable Systems & Bringing Sexy Back to Information Centricity

v1.1

The Internet Is Over*

So The Cloud's Got that going for it...

* Everybody knows you can't argue with royalty

Hustle & Flow

+ *Key Takeaways*
+ *Fist Pump the Cloud: Jersey Shore Security*
+ *Blame the French*
+ *Shifts In Thinking*
+ ***Idempotent Infrastructure***
+ ***Survivable Systems***
+ ***Information Centricity***
+ *Wrap-Up*

Key Takeaways

- Not All Public Cloud IaaS Offerings Are Created Equal. Differentiation Based Upon Networking, Security, Transparency/Visibility & Forensics
- Public IaaS Clouds Can Most Definitely Be Deployed As Securely Or Even More Securely Than Those In An Enterprise...
- ...However, They Require Profound Architectural, Operational, Technology, Security and Compliance Model Changes
- Time To Get The Bell Bottoms Out Of The Closet: What's Old Is New Again - Survivable Systems & Information Centricity

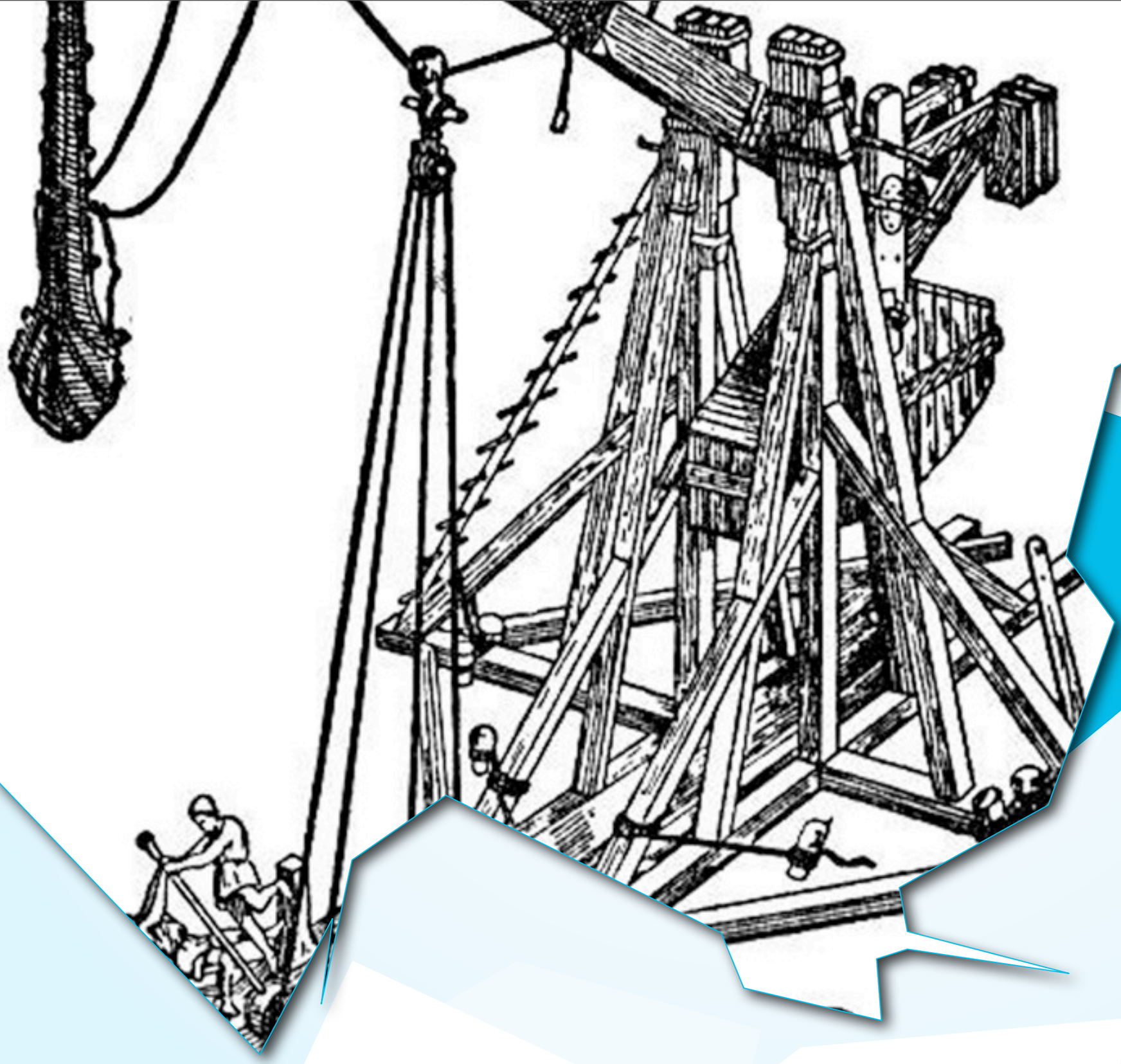
Fist Pump The Cloud

The Car Crash You Just Can't Stop Watching

Four Horsemen Of the Virtualization Security Apocalypse

The Frogs Who Desired a King: A Virtualization & Cloud Computing Fable Set To Interpretive Dance

Cloudifornication: Indiscriminate Information Intercourse Involving Internet Infrastructure

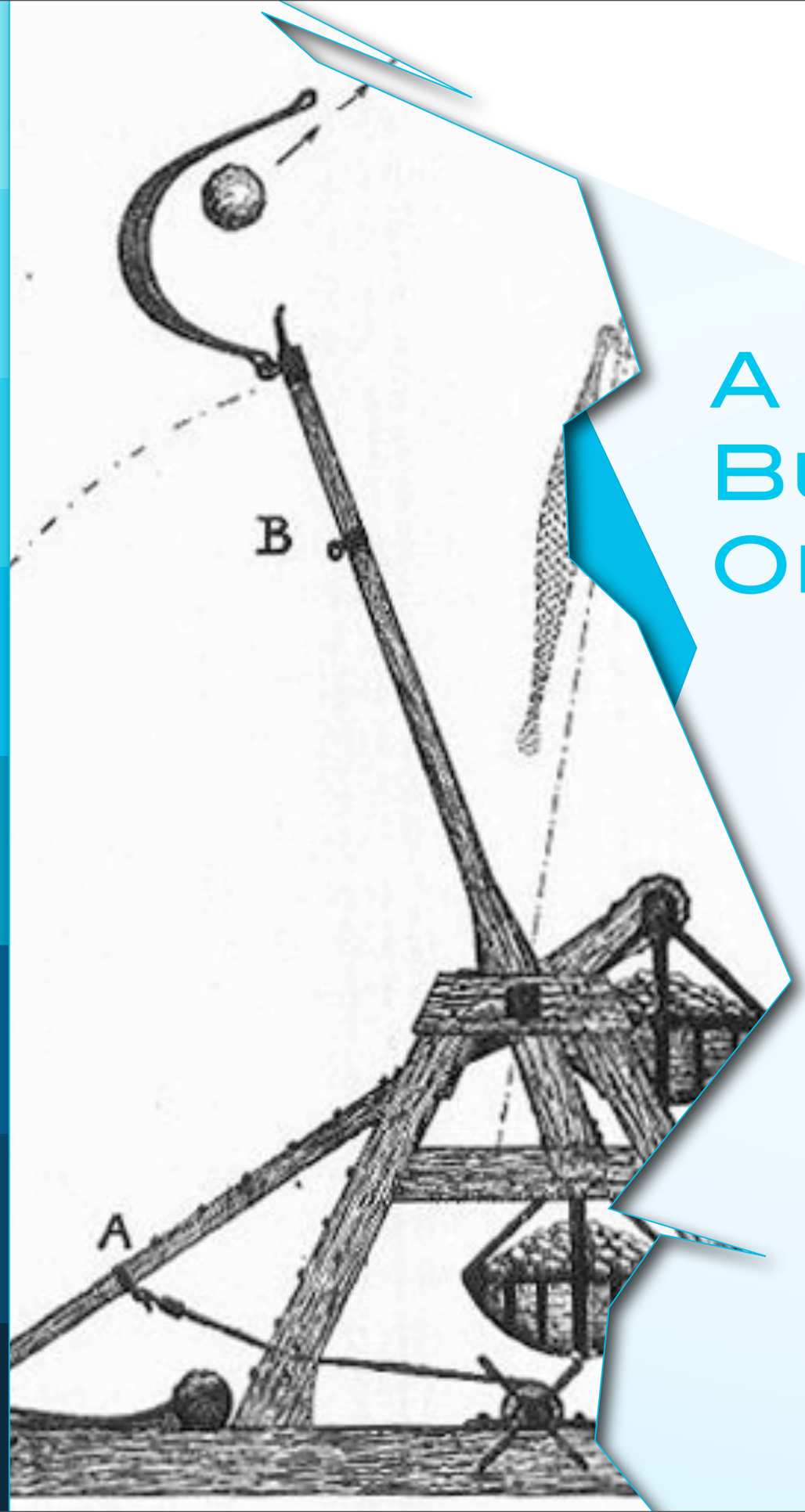


Blame The French::

Siege Warfare & the Trebuchet...

Technically Blame the Greeks & Romans...

- Introduced in ~12th century by the French who bettered the design elements of the catapult & ballista
- The trebuchet utilized a sling to double the power of the engine and throw its projectile twice as far
- Catapults were efficient mechanisms for lobbing loads of 50-60 pounds
- Trebuchets could throw stones of up to 300 pounds and at great distance



A Better Mousetrap But A More Complex Operational Model

+ The sling trebuchet was a marriage of previous catapult design, application of better physics & advanced physical science.

+ It works on a simple principle, but there was nothing simple about making sure a sling trebuchet was built or operated with precision...*

*http://www.medieval-castle-siege-weapons.com/sling-trebuchet.html

WTF Does That Have To Do With Cloud?

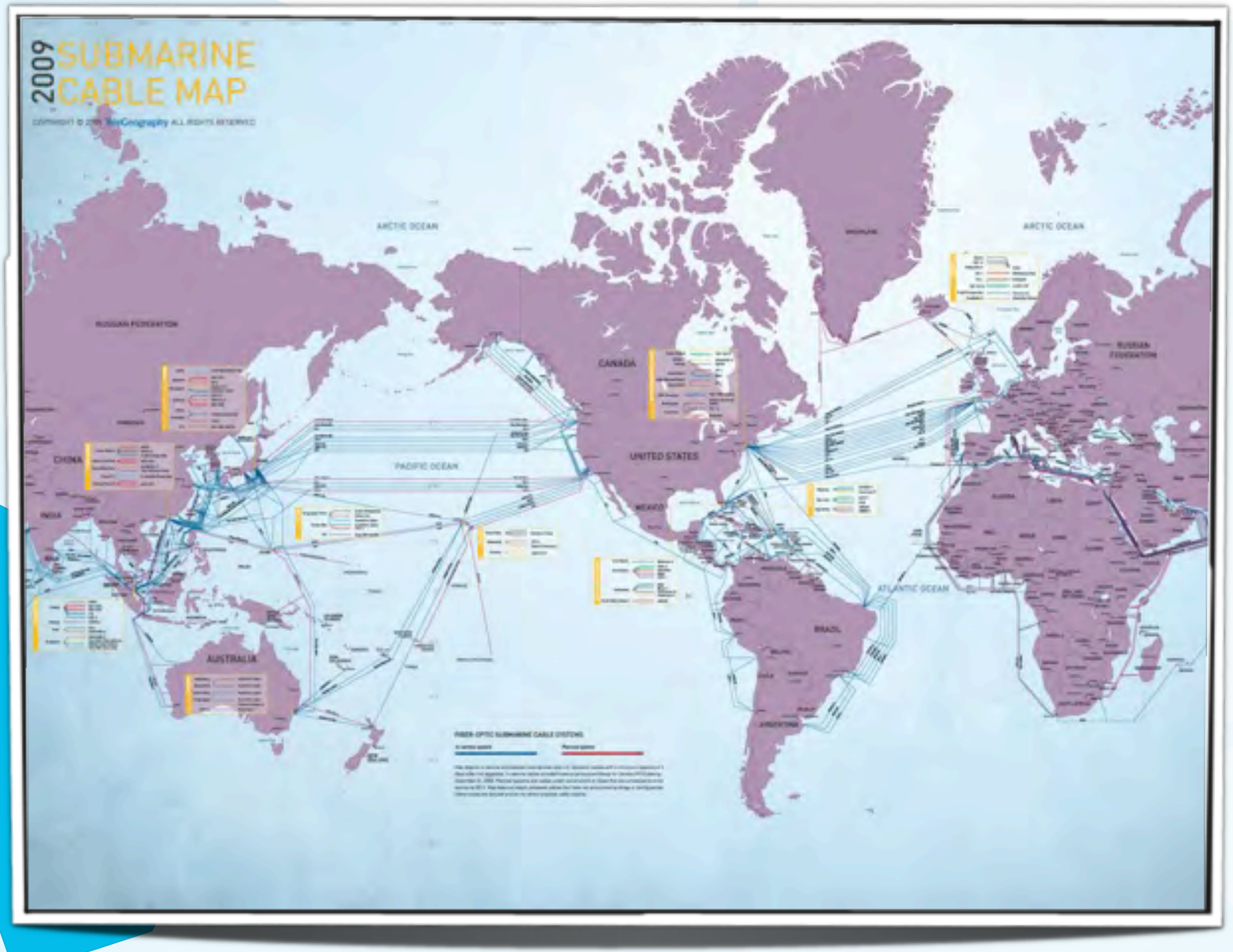
+ Evolutionary application of revolutionary ideas*
+ Caused quite a stir and a wholesale shift in strategy
+ Laid the foundation for even more ass-kicking innovation
+ Automation, FTW!

* Gravity. Who Knew?

Shifts In Thinking*

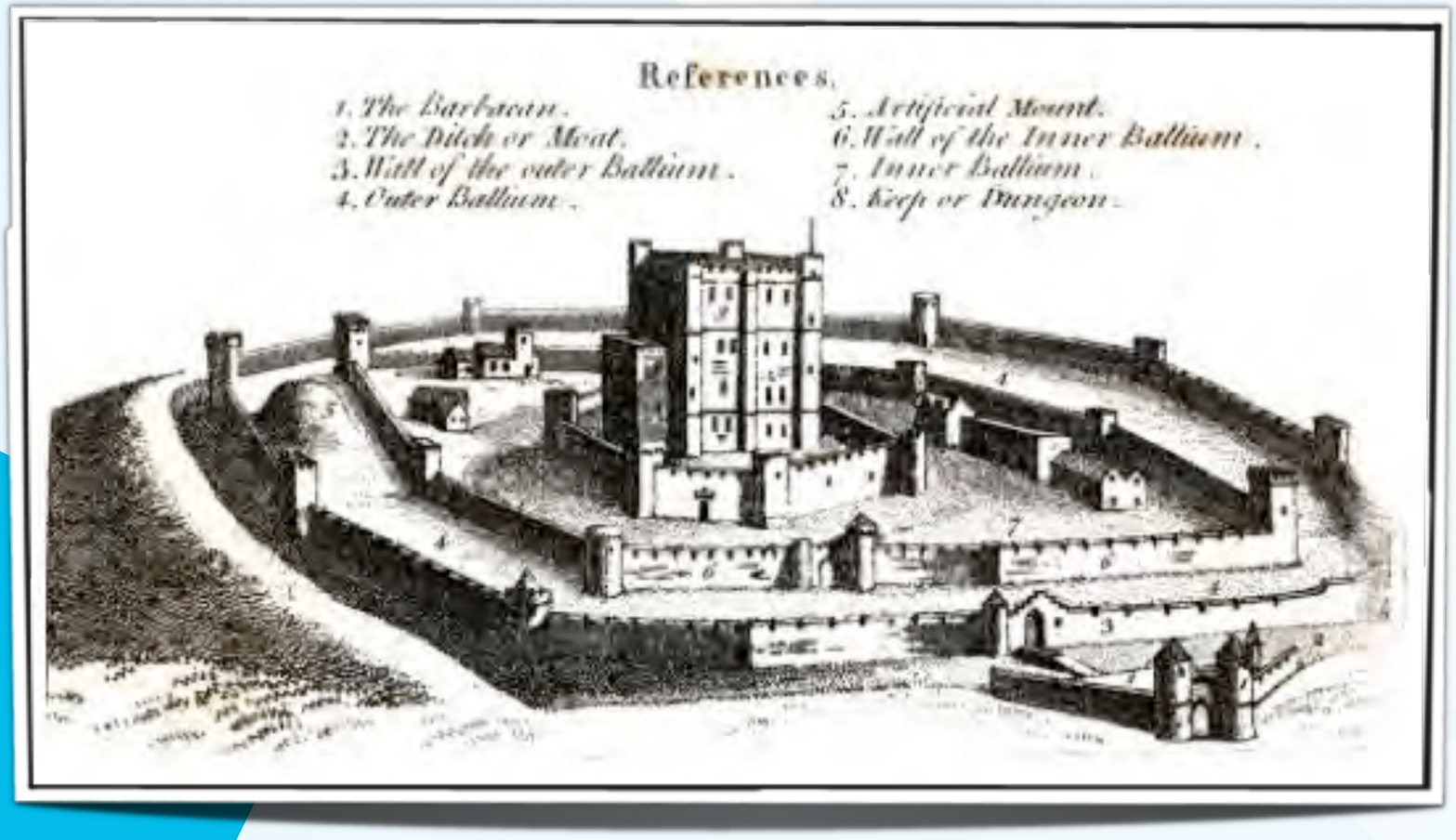
IT Evolves

*2001 - Carnegie Mellon University. Information Survivability: A New Executive Perspective



Centralized To Global

Bounded To Unbounded



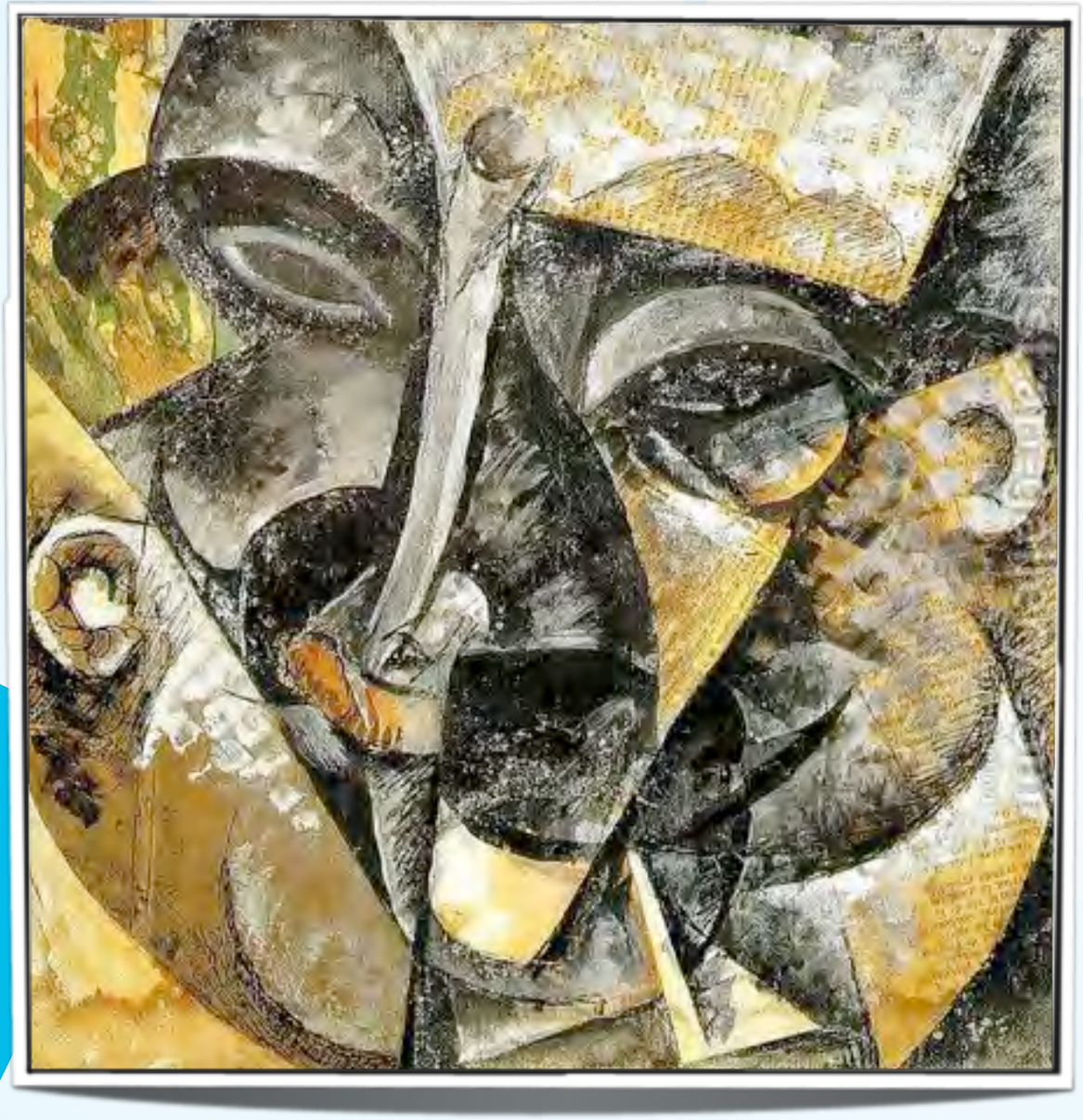
Insular To Networked

Predictable To Asynchronous

Single To Shared Responsibility

Overhead To Essential

Security To Survivability



Static To Dynamic*

*I Added This One

Manual To Automated*

*I Added This One

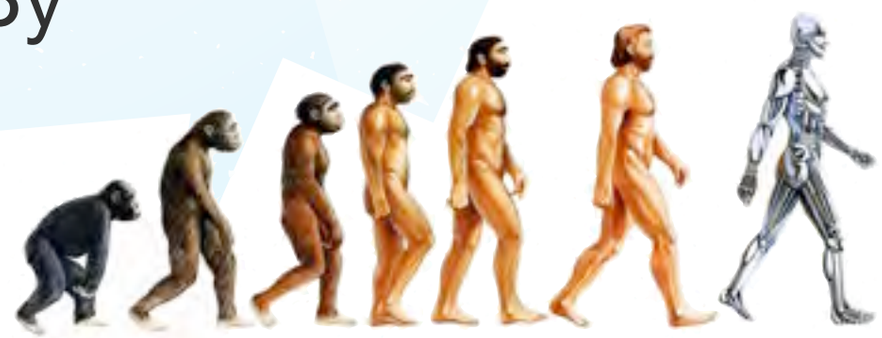
Shifts In Thinking: IT Infrastructure Evolves

+ Consolidating From Servers To Pooled Resources Of Compute

+ Network & Storage Moving From Dedicated Switches & Local Disk To Clusters And Fabrics, Implemented In Both Hardware *And* Software

+ Escaping From Tightly-Coupled Hardware/Software Affinity To Distributed Computing Enabled By Virtualization

+ Transitioning From Infrastructure To Composeable Service Layers



Public Cloud...

+ Not All Public Clouds Are Created Equal Or For The Same Purpose
+ Scale Enabled By Abstracted & Idempotent Infrastructure*
+ Massive Data Centers With Hundreds Of Thousands Of Cores, Huge Storage And Bandwidth
+ Extremely Agile, Heavily Virtualized, Mostly Automated & Hugely Software Driven
+ Management Via API

"Enterprise" IaaS Quandry: To Boldly Go...

Private: Leverage Virtualization To Yield Higher Efficiency In Service Delivery, Agility And Meet Existing Security And Compliance. Infrastructure Exposed.

General Preservation Of Existing Architectural Blueprints But With Virtualized/Converged Infrastructure. Primarily Hardware Infrastructure Enabled & Enterprise-Class Virtualization Layers

Public: Fundamental Re-Architecture Of Application, Operations & Service Delivery Leveraging Virtualization & Automation. Massive Abstraction.

Focuses On Scale, Lower Costs And Homogeneity At Infrastructure Layers. Primarily Software Enabled

Public Cloud:

All About Gracefully Giving Up Direct Operational Control Over Infrastructure

Across The Great Divide...

Therein lies the problem...

+ Huge monocultures of custom hardware and software layers abstracted for your pleasure

+ It's the functional equivalent of Siebel: don't fit the software to the business, change the business to fit the software.

+ ...not necessarily a bad thing, but cultural, operational, security, and compliance issues are daunting.

Shifts In Thinking: The Evolution Of Security

	Past	Present	Tomorrow (?)
Attack Origin/ Motivation	External/ Notoriety	Internal & External/ Leverage	Ubiquitous/Economic & "CyberTerror"
Defensive Strategy	Perimeters & Firewalls	Amorphous Perimeters & Firewalls /VPNs	Re-Perimeterized & Self-Asserting Defense
Approach	Infrastructure-Centric	Application-Centric	Information-Centric
(Id)entity Management	IP Address	Discrete User Credentials	Federated Trust
Application Deployment	Monolithic Apps	Mash-Ups	Distributed Functions
HW/SW Resources	Application/Resource Affinity	Virtualization 1.0	Virtualization-Enabled Cloud
Data Location/ Type	Islands of Data/ Structured	Data Warehouses/Structured & Un-Structured	Re-Distributed Data Marts & Metadata

What Cloud Means To Security

- Focus on sustaining the business/mission in the face of an ongoing attack; requires a **holistic perspective** (not siloed)
- Depends on the ability of networked systems to **provide continuity of essential services, albeit degraded**, in the presence of attacks, failures, or accidents
- Requires that only the **critical assets need the highest level of protection**
- **Complements current risk management** approaches that are part of an organization's business practices
- Includes (but is broader than) traditional information security

We Need Risk Ritalin

There Be Monsters Here...

+ Suffering From Security Attention Deficit Disorder & Lack Of Holistic Approach

+ Threat Model Velocity And Innovation Of Attacker > Defender

+ Security Doesn't Scale (By Design)

+ Defense In Width...

+ Economic Model Does Not Incentivize Solutions That Solve Problems

‘Round & ‘Round We Go…

Achtung! Divergent Models

Mainframes

The Cloud

Web2.0

Client/Server

Web1.0

Display

Compute

Data

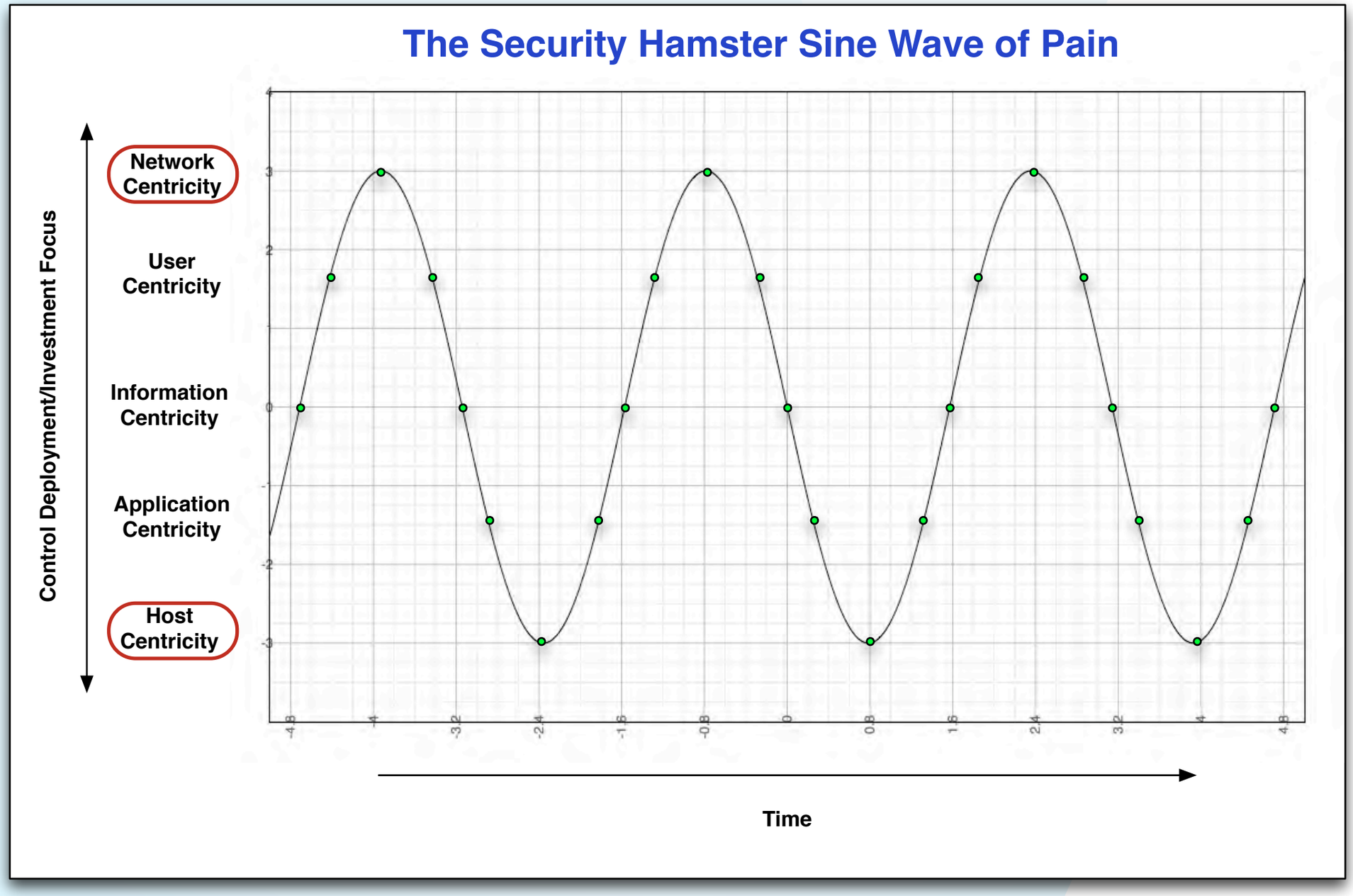
Bandwidth

	Developers	Security
1995	CGI, PERL	Network firewalls, SSL
1997	ASP, JSP	Network firewalls, SSL
1998	EJB, J2EE,	work firewalls, SSL
1999	SOAP, XML	work firewalls, SSL
2001	Rest, SOA	etwork firewalls, SSL
2003	Web 2.0	Network firewalls, SSL

Dis ted

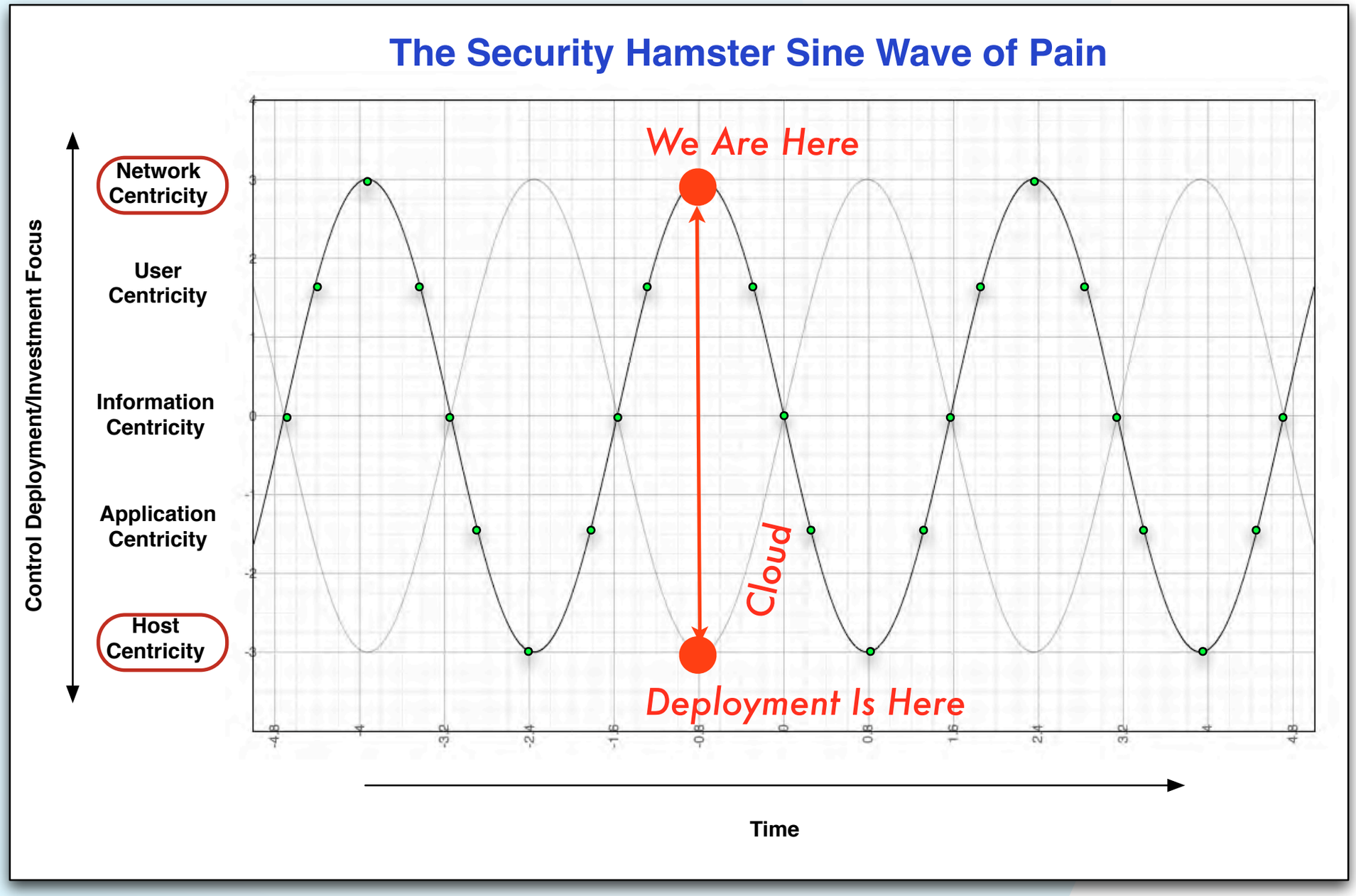
** Credit: Gunnar Peterson*

Revenge Of The Hamsters...



** With Apologies to Andy Jaquith & His Hamster...*

Revenge Of The Hamsters...



** With Apologies to Andy Jaquith & His Hamster...*

Deconstruction

Infostructure

Metastructure

Infrastructure

- **Content & Context** -
  Apps, Data, Metadata, Services
- **Glue & Guts** -
  IPAM, IAM, BGP, DNS, SSL, PKI
- **Sprockets & Moving Parts** -
  Compute, Network, Storage

Idempotent Infrastructure

Idempotent?

idempotent |ˈīdemˌpōtənt| Mathematics

adjective
denoting an element of a set that is unchanged in value when multiplied or otherwise operated on by itself.

noun
an element of this type.
ORIGIN late 19th cent.:
from Latin ***idem 'same'*** + **potent**

*In computer science, the term **idempotent** is used to describe methods or subroutine calls that can **safely be called multiple times**, as invoking the procedure a single time or multiple times **has the same result**;*

Infrastructure

* wikipedia: http://en.wikipedia.org/wiki/Idempotence

Idempotency & Public IaaS Cloud

+ Homogeneity Provides Foundation For Scale [out]
+ Does Not Always Imply Commodity Hardware
+ Maximize Density & Modularity Of Resources
+ Iteratively Extensible
+ Constant Deployment Model (Agile) Of Software Driven "Infrastructure"
+ Code As Infrastructure
+ But Elasticity Isn't Always The Biggest Problem To Solve...

Infrastructure

Attack Of the Stack

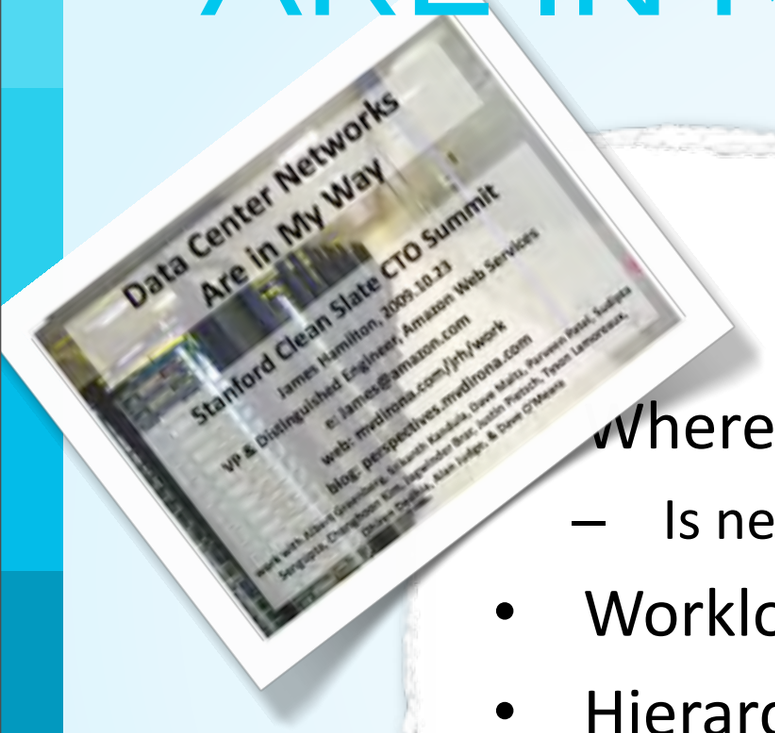
- Some Examples Of The Growing Number Of Available Cloud “Operating Systems”:
  - **OpenStack.org**
  - Cloud.com CloudStack
  - Citrix XenCloud
  - VMware vCloud
  - Enomaly ECP
  - RedHat Cloud Foundations
  - Nimbula Director
  - Eucalyptus Enterprise Edition
- The Stuff That Makes It Tick Underneath Is Interesting & Important to Discuss

Compute Architecture - Cores & Memory

- Compute Fabrics:
  - Commodity, “Engineered” or Proprietary/Specialized
  - Lots of CPUs vs Fewer CPUs With Lots Of Cores
  - CPU vs GPU (or otherwise)
- Low Power, Low BTU, Highly Dense
- Dedicated vs Huge Shared Memory
- Management via RESTful HTTP API

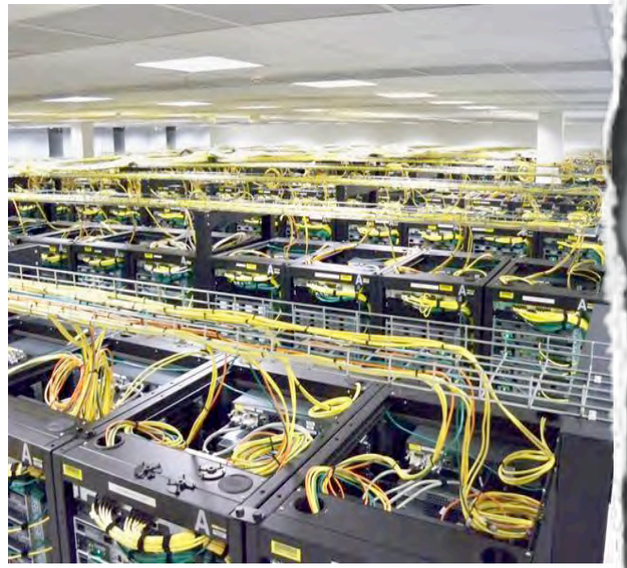
Infrastructure

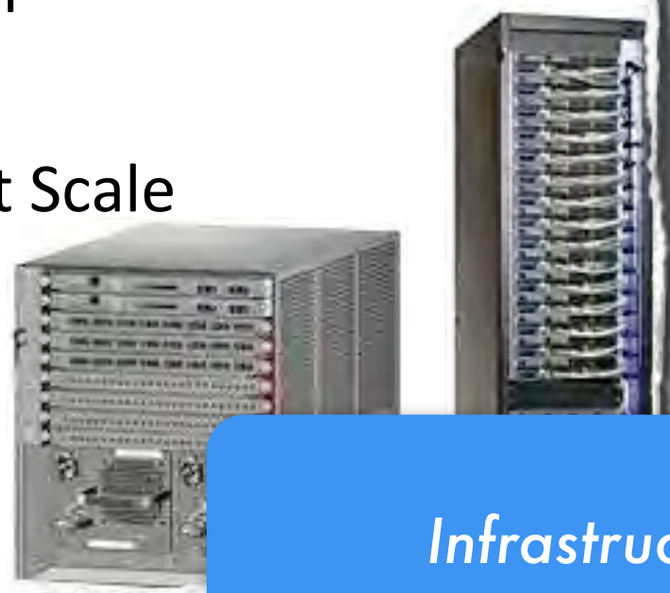
"Datacenter Networks Are In My Way*"



Agenda

- Where Does the Money Go?
  - Is net gear really the problem?
- Workload Placement Restrictions
- Hierarchical & Over-Subscribed
- Net Gear: SUV of the Data Center
- Mainframe Business Model
- Manually Configured & Fragile at Scale
- Problems on the Border
- Summary





Infrastructure

* James Hamilton, AWS

Network Architecture

- **Where's the Network? Software vs Hardware: VMM-Integrated, Nexus 1000v, Open vSwitch, OpenFlow, Nicira, Or Hybrid Models...**
- **Hugely Abstracted Networks Create Challenges With:**
  - **Topology** - L2/L3 Design, Multicast/Broadcast, STP, etc.
  - **Security** - Presentation Of Hooks To Interconnect/Segment, Visibility, Management
  - **Mobility** - Dynamism Stresses Resource: Naming, Location, Addressing, etc.
  - **Performance** - I/O, Packet Ping Pong, QoS
- **Revenge Of The Meshed Overlay VPN**
  - Need for PKI, Link (Physical & Logical) Link Encryption, Authentication, Tagging, Crypto

Infrastructure

Network Architecture (Cont.)

- **Big, "Dumb," Flat, Fast L2 Networks Vs. Next Generation Of Classical Core|Distribution| Access**
  - Heavily Virtualized High Density, Low Latency, Non-Blocking, Line Rate, 10+Gb/s
  - Full Bisection Bandwidth vs. Statistical Multiplexed & Over-subscribed
  - Segmentation, Multi-tenancy & Scale: Abstracted Into VMM or (p)VLANs/VRF or by separating data/control/flow planes
  - Programmable & Open vs. Fixed & Proprietary
  - Data-Only Versus Converged Data & Storage
- **RESTful HTTP APIs and Exposed Interfaces for Automation/Provisioning/Orchestration/ Instrumentation**

Infrastructure

VMs : The New DMZ?

- Practically, The VM Boundary **Is** The Emerging Atomic Unit And Therefore Is The Logical Perimeter. For Now.
- Ultimately We'll See A New Measure As VM's & The Servers They Replaced Give Way To PaaS/Language Abstractions & "Workload" Gets More Defined*
- The Cloud OS Platform Networking Can Clearly Be A Fantastic Differentiator Or A Huge Limiter For Delineating Policy Boundaries:
  - Determines Whether You Get "Good Enough" Or Extensible Security Capabilities
  - Drives Variability In If, How and Where One Might Deploy Compensating Controls:
    - Physical/Virtual (Network|Appliances),
    - VMM, VMM APIs,
    - Guest-Based (Software)

Infrastructure

*E.g. Heroku uses "Slugs" & "Dynos"

Storage

- Local disk vs NAS/SAN/Object-Based
- Storage Can Be Exposed via RESTful/SOAP APIs Or Volumes
- Persistent vs Non-Persistent
- Volume/Bucket/File Size Limits
- Converged (FCoE) Storage & Networking
- I/O and Performance
- Impacts On Application Architecture
- Storage “Mobility”
- BCP/DR/Resilience/Recovery
- Isolation/Security/Forensics in Multi-Tenant Environments

Protocols: That Pesky TCP/IP

- Van Jacobson Summed It Up Well (and I Paraphrase)*:
  - The Cloud Today Is Like The ARPAnet Of Yesterday: "...At The Outset The New Network Looked Like An Inefficient Way To Use The Old Network."
  - Ubiquitous "Any-To-Any" Communication Is Not What TCP/IP Was Created For; When Created, Computers Were Huge Things That Lived In Glass-Walled Machine Rooms & Had 1000 People Using The Same Machines...
  - **Things that people are doing with cloud and what the cloud does inside are different; use cases are the side effect.**
  - Communications Today Are Disseminatory Versus Conversational; Mobility, Naming/Location and Trust Are Big Issues
  - Today's Networking [Protocol] Problems Don't Reflect An Architectural Failing, But Rather TCP/IP's Success In Creating A World Rich In Information & Communication; **TCP/IP Is A "Success Disaster"**

Metastructure

** Van Jacobson: A New Way To Look At Networking | 2006 | http://bit.ly/5jGte*

Application Architecture

Infostructure

+ Architecturally, The Classical n-Tier/DMZ Segment & Asset Grouping Methodology May No Longer Apply

+ Applications Are Likely Not Composable In This Manner As They Are More Topologically & Infrastructure-Insensitive Than Ever (*See Flat L2 designs*)

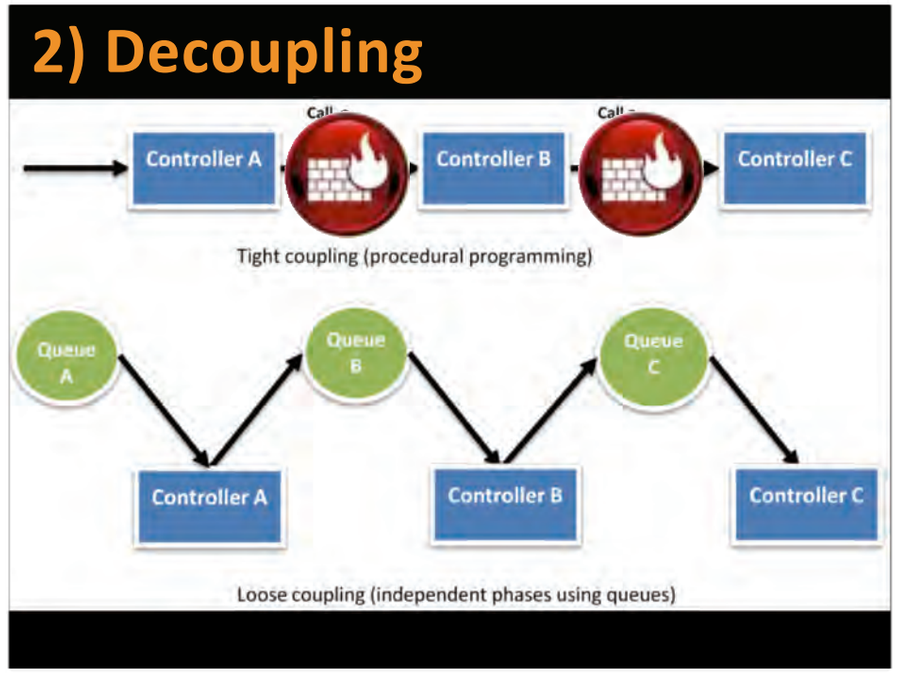
+ In Many Cases, Applications Must Be Completely Rewritten To Leverage Public Cloud & The Security Models Must Be Adjusted

+ Dumb Networks Equal Dumber Security Or At Least Less Options/Capabilities; Workload Sensitivity vs. Network Capability Is Critical

+ People Still Write Crappy Code, No Matter How Good, Abstracted Or Elastic The Infrastructure Is

+ Continuous Deployment Models (Agile & DevOps) Are Taking Root

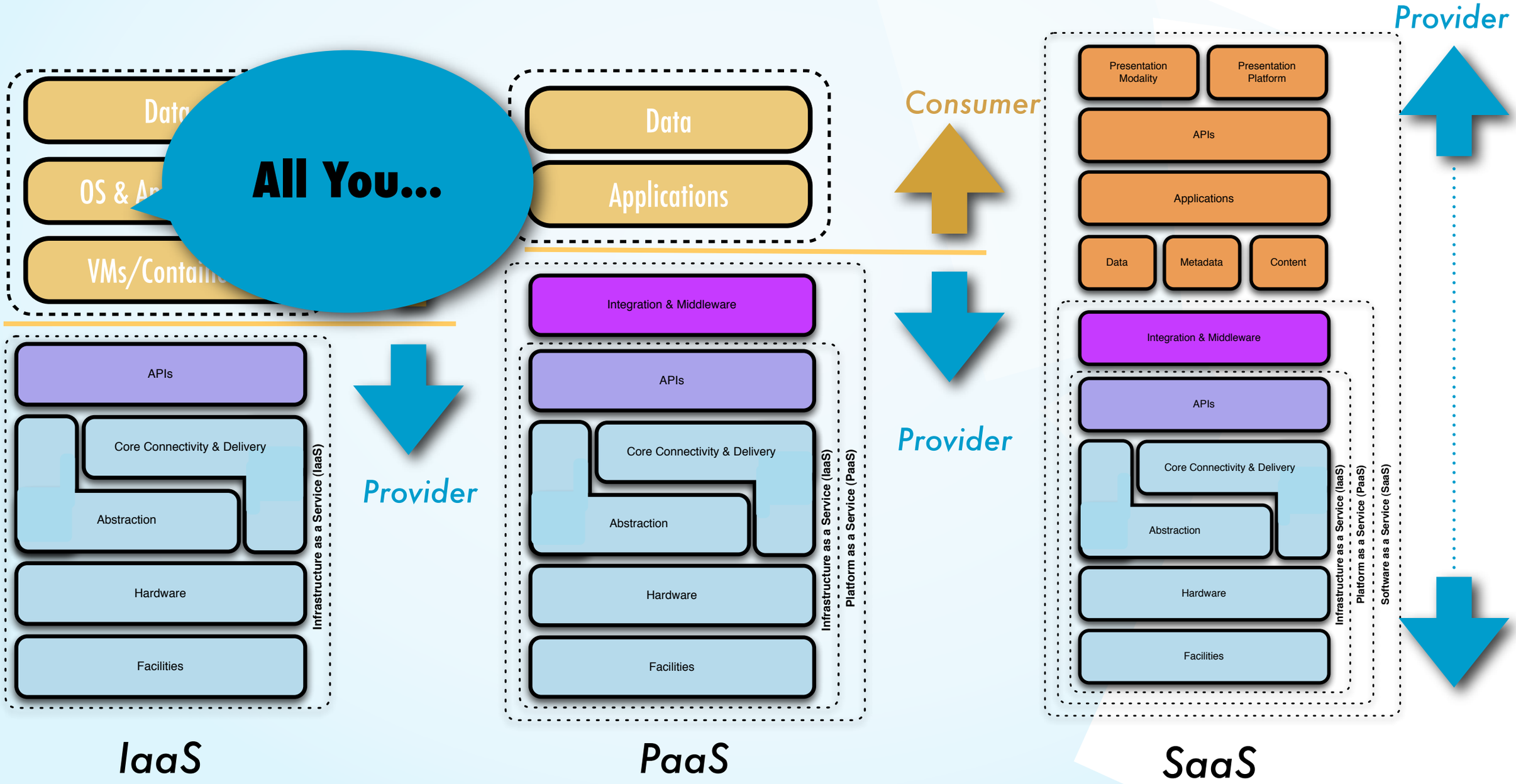
+ Application Security Is Even More Important Than Ever



Simone Brunozzi - http://bit.ly/aws_architect-cloud

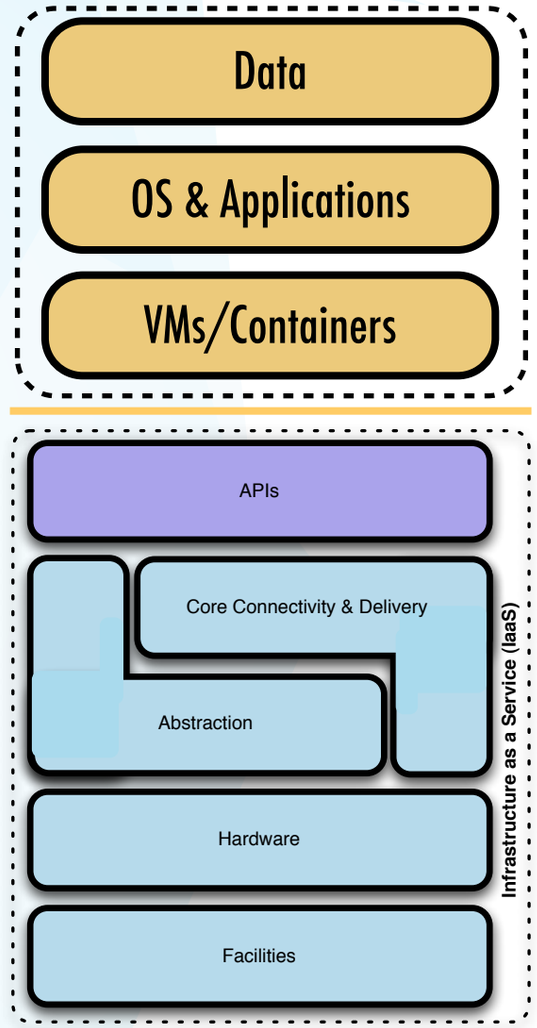
The Solutions Are Not Flawed, The Problems Have Changed

Public IaaS Cloud Security Model



Abstraction Has Become a Distraction

Not Much You Can Do Here...



Your Focus Is Here:

- Building Survivable Systems
- Building Secure Apps
- Securing Data

Survivable Systems

http://ingesidee.de/resourceviewer.php?pgid=63&lang=en&subpage=1&module=content


Those Who
Ignore The Past
Are Doomed To
Build Upon It

Architecting for the Cloud

1) Design for Failure

2) Decoupling

3) Elasticity

4) Security

5) Break Costraints

6) Think Parallel

7) Different Storage Options

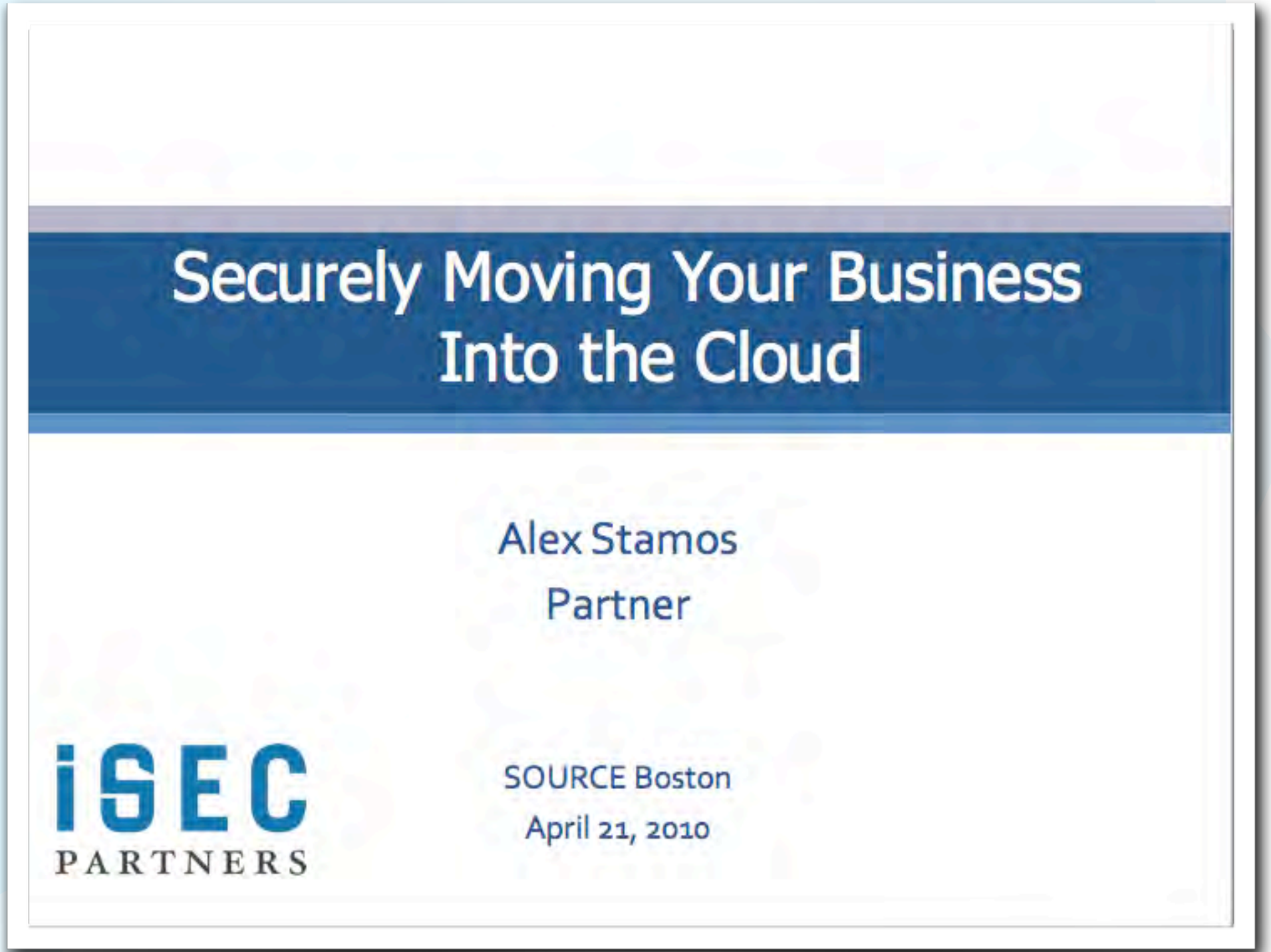
1) Design for Failure

- ackup/Restore Strategy
- Be impe
- Move in tore
- Use Ava C2
- Use Elas
- Use Relational Database Service (RDS)
- Use Elastic IP

In This Case You Are Leveraging Provider-Specific Attributes To Enable Resilience. What If One Needs Portability Or Interoperability And Another Provider Doesn't Offer These Functions?

Simone Brunozzi - http://bit.ly/aws_architect-cloud

Or In Alex Stamos' Words...



Or In Alex Stamos' Words...

Takeaways

- Current conventional wisdom on cloud computing is missing the point
- You cannot securely move into the cloud without re-writing your software
- Secure cloud applications "collapse the perimeter"
- Properly going through this process should leave you more secure than before

iSEC PARTNERS

Or In Alex Stamos' Words...

What is the alternative?

- Go Flat
- Collapse the Perimeter
- Use cloud glue services
- Enforce access control via cryptography

iSEC PARTNERS

16

Sounds Easy Enough, But...

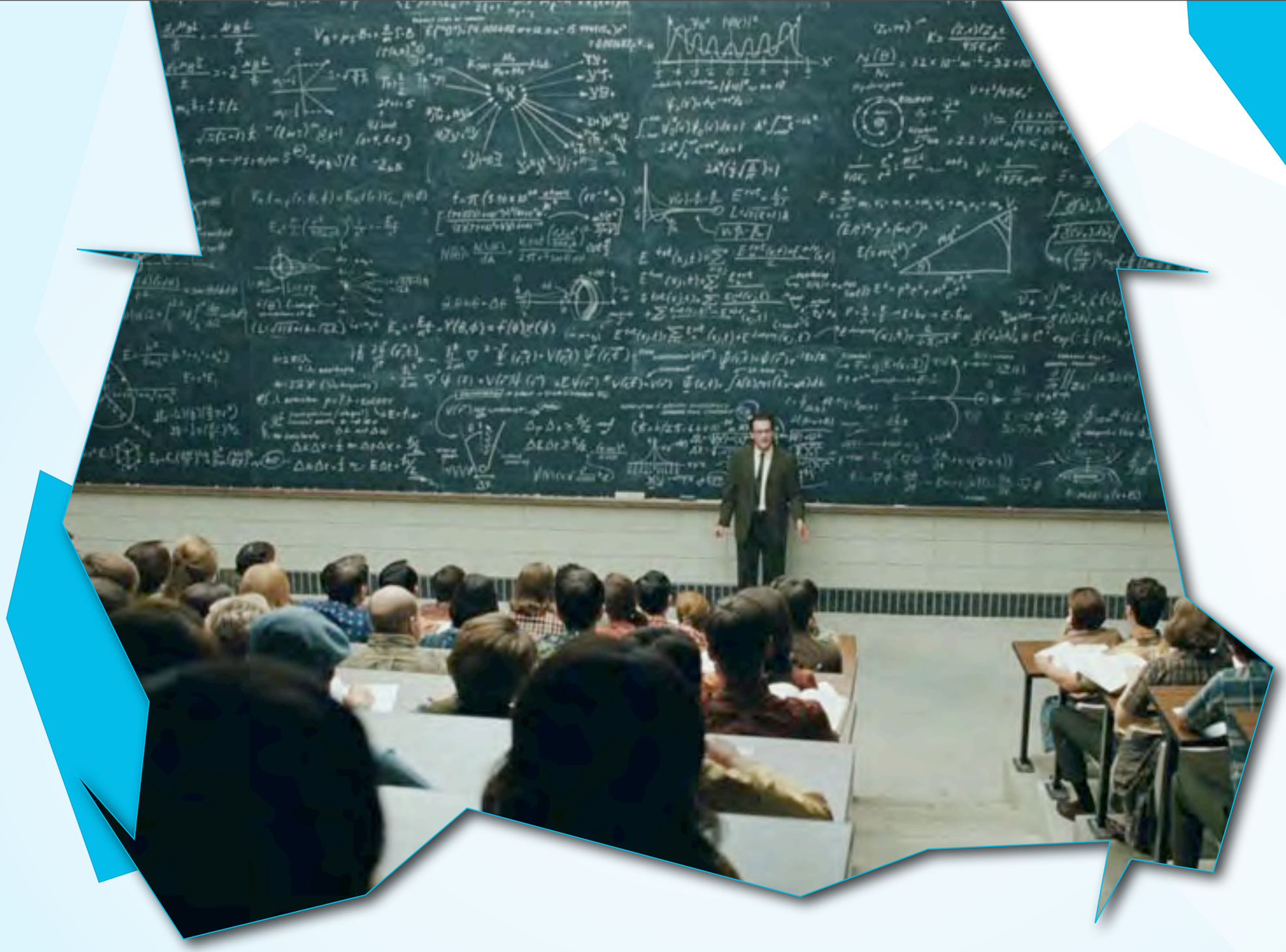
+ There's A Big Difference Between Greenfield and Existing Applications
+ The Software Architecture Changes Dramatically, But The Security Models & Solutions Are Limiting
+ As We Move Greenfield Applications To Public Cloud, We're Forced To Build
  More Survivable Systems Because We Can't Depend On The Provider
+ Focus On The Things That Matter Most: Securing The VM's, The Apps That Run In Them And The Data That Those Apps Trade In

That Means...

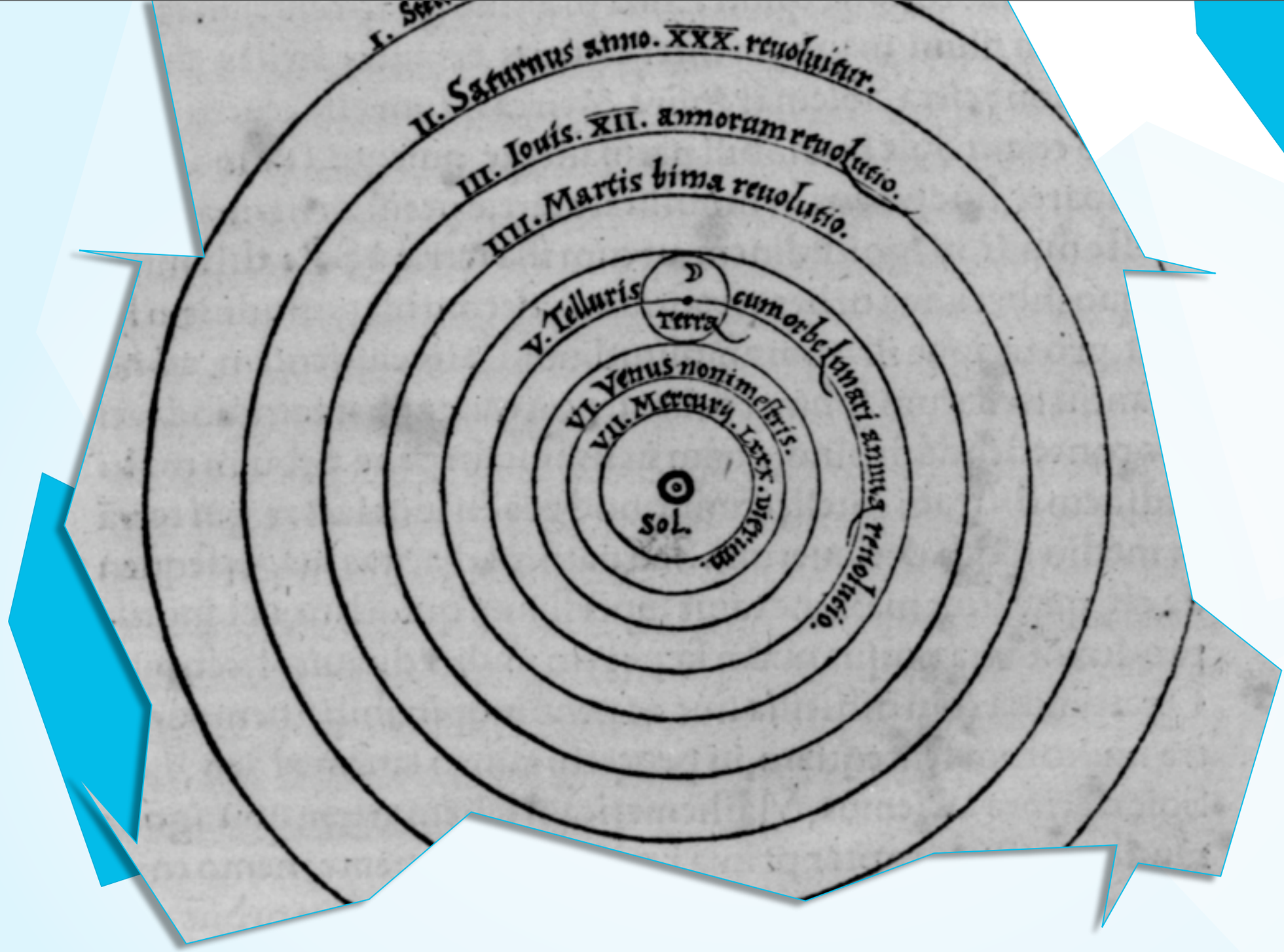
+ Assume All Environments Are Hostile And Isolation in Multi-Tenancy Will Fail
+ Focus on JEOS (Just Enough O.S.)
+ Invest Heavily In {Id}Entity Management + Crypto
+ Ensure Robust Instrumentation & Telemetry At Infra-, Meta- & Info-structure layers
+ Choose Open Protocols and Stacks
+ Leverage Introspection
+ Select Platforms That Enable Robust Monitoring And Forensics
+ Utilize the Three R's and a **SYSTEM-FOCUSED** Design Philosophy

The 3 R's: Resistance, Recognition & Recovery

- **Survivability**: Delivery Of Essential Services and Preservation Of Essential Assets Capable Of Fulfilling Mission Objectives:
  - **Resistance**: Capability Of System To Repel Attack
  - **Recognition**: Capability Of System To Detect Attack and Evaluate Extent Of Damage/ Compromise
  - **Recovery**: Capability To Maintain Essential Services and Assets During Attack, Limit Extent Of Damage and Restore Services
- **We Still Don't Think In Terms Of Systems...But Cloud ~~Helps~~ Forces Us To Get There By Focusing On Being Information Centric**



Information Centricity



Copernican Cloud Security

Jericho Forum Commandments

The Jericho Forum commandments define both the areas and the principles that must be observed when planning for a de-perimeterized future.

Whilst building on "good security", the commandments specifically address those areas of security that are necessary to deliver a de-perimeterized vision.

The commandments serve as a benchmark by which concepts, solutions, standards and systems can be assessed and measured.

Fundamentals

1. The scope and level of protection should be specific & appropriate to the asset at risk

- Business demands that security enables business agility and is cost effective
- Whereas boundary firewalls may continue to provide basic network protection, individual systems and data will need to be capable of protecting themselves
- In general, it's easier to protect an asset the closer protection is provided

2. Security mechanisms must be pervasive, simple, scalable & easy to manage

- Unnecessary complexity is a threat to good security
- Coherent security principles are required which span all tiers of the architecture
- Security mechanisms must scale; from small objects to large objects
- To be both simple and scalable, interoperable security "building blocks" need to be capable of being combined to provide the required security mechanisms

3. Assume context at your peril

- Security solutions designed for one environment may not be transferable to work in another. Thus it is important to understand the limitations of any security solution
- Problems, limitations and issues can come from a variety of sources, including geographic, legal, technical, acceptability of risk, etc.
- Surviving in a hostile world

Surviving in a Hostile World

4. Devices and applications must communicate using open, secure protocols

- Security through obscurity is a flawed assumption - secure protocols demand open peer review to provide robust assessment and thus wide acceptance and use
- The security requirements of confidentiality, integrity and availability (reliability) should be assessed and built in to protocols as appropriate, not added-on
- Encrypted encapsulation should only be used when appropriate and does not solve everything

5. All devices must be capable of maintaining their security policy on an untrusted network

- A "security policy" defines the rules with regard to the protection of the asset
- Rules must be complete with respect to an arbitrary context
- Any implementation must be capable of surviving on the raw Internet, e.g., will not break on any input

The need for trust

6. All people, processes, technology must have declared and transparent levels of trust for any transaction to take place

- Trust in this context is establishing understanding between contracting parties to conduct a transaction and the obligations this assigns on each party involved
- Trust models must encompass people/organisations and devices/infrastructure
- Trust level may vary by location, transaction type, user role and transactional risk

7. Mutual trust assurance levels must be determinable

- Devices and users must be capable of appropriate levels of (mutual) authentication for accessing systems and data
- Authentication and authorisation frameworks must support the trust model

Identity, Management and Federation

8. Authentication, authorisation and accountability must interoperate / exchange outside of your locus / area of control

- People/systems must be able to manage permissions of resources and rights of users they don't control
- There must be capability of trusting an organisation, which can authenticate individuals or groups, thus eliminating the need to create separate identities
- In principle, only one instance of person / system / identity may exist, but privacy necessitates the support for multiple instances, or once instance with multiple facets
- Systems must be able to pass on security credentials /assertions
- Multiple loci (areas) of control must be supported

Access to data

9. Access to data should be controlled by security attributes of the data itself

- Attributes can be held within the data (DRM/Metadata) or could be a separate system
- Access / security could be implemented by encryption
- Some data may have "public, non-confidential" attributes
- Access and access rights have a temporal component

10. Data privacy (and security of any asset of sufficiently high value) requires a segregation of duties/privileges

- Permissions, keys, privileges etc. must ultimately fall under independent control, or there will always be a weakest link at the top of the chain of trust
- Administrator access must also be subject to these controls

11. By default, data must be appropriately secured when stored, in transit and in use

- Removing the default must be a conscious act
- High security should not be enforced for everything; "appropriate" implies varying levels with potentially some data not secured at all

Conclusion

De-perimeterization has happened, is happening and is inevitable; central protection is decreasing in effectiveness

- It will happen in your corporate lifetime
- Therefore you need to plan for it and should have a roadmap of how to get there
- The Jericho Forum has generic roadmap to assist in the planning

Distilled Principles of Information-Centric Security

+ Information (audio/video/ data) must be self describing and defending

+ ...Structured Or Unstructured

+ Policies and controls must account for business context.

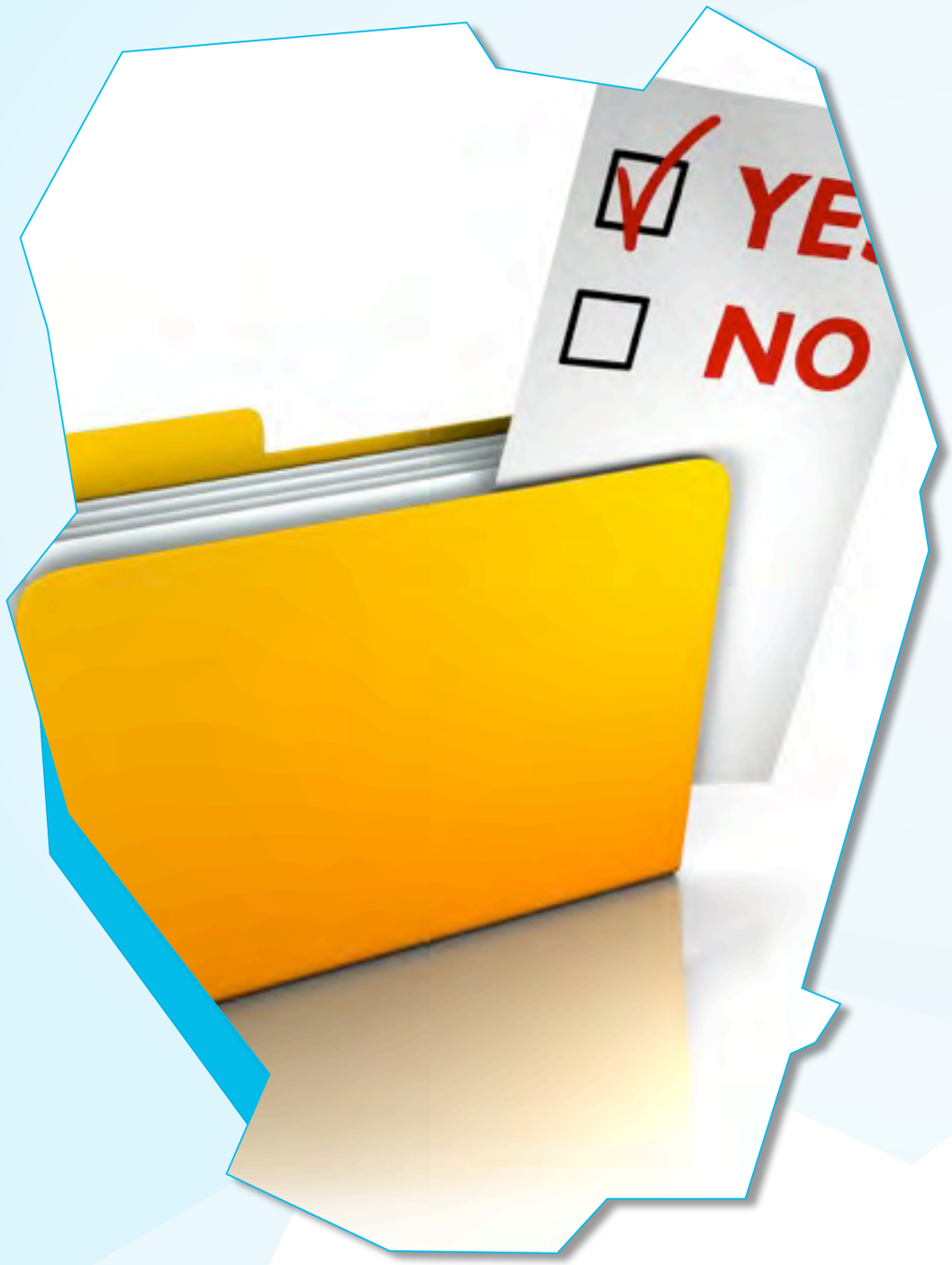
+ Information must be protected as it moves between silos, between locations, and changing business contexts.

+ Policies must work consistently through the different defensive layers and technologies we implement.

Risk Assessment & Threat Modeling

- Risk Assessment and Threat Modeling are NOT Information Alchemy, But <u>Do</u> Require Lots Of Work
- Some Frameworks To Choose From:
  - Microsoft **STRIDE/DREAD**
  - Carnegie-Mellon **OCTAVE**
  - RMI **FAIR**
- *"How Can You Have Your Pudding If You Don't Eat Your Meat?"*

The 6 Key Questions



1. How would we be harmed if the asset became public and widely distributed?
2. How would we be harmed if an employee of our cloud provider accessed the asset?
3. How would we be harmed if the process or function was manipulated by an outsider?
4. How would we be harmed if the process or function failed to provide expected results?
5. How would we be harmed if the information/data was unexpectedly changed?
6. How would we be harmed if the asset was unavailable for a period of time?

Managing Information Across Its Lifecycle*

Create

{Discover}
Classify
Assign Rights

Store

Access Controls
Encryption
Rights Management
Content Discovery

Use

Activity Monitoring
and Enforcement
Rights Management
Logical Controls
Application Security

Share

CMP (DLP)
Encryption
Logical Controls
Application Security

Archive

Encryption
Asset Management

Destroy

Crypto-Shredding
Secure Deletion
Content Discovery

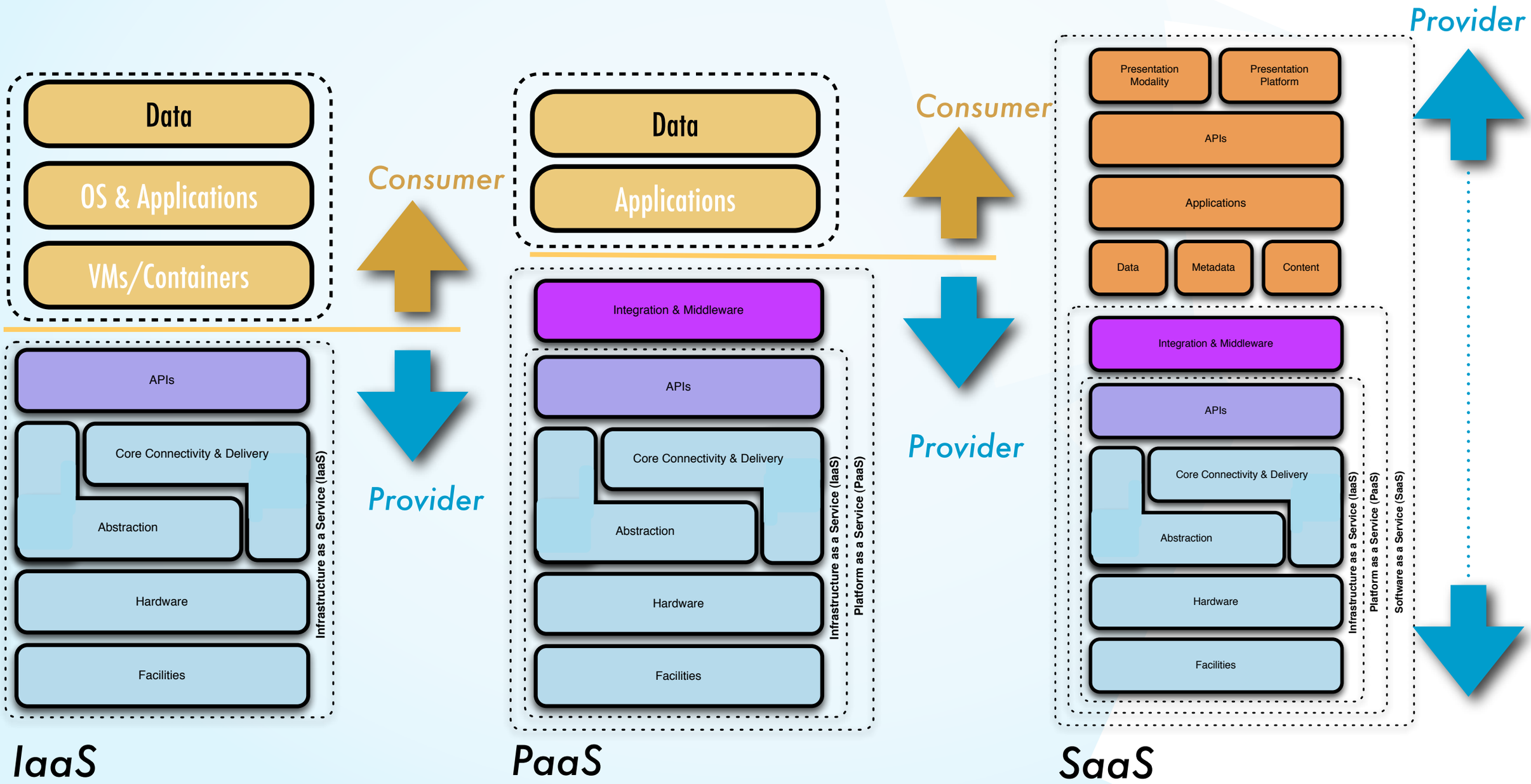
**Rich Mogull - Securosis*

Information Centric Solutions For Cloud

Labels

**Rich Mogull - Securosis*

Delivery Model Mapping Revisited



Where This Take Us

- Content Analysis Fully Integrated Into Both Productivity And Transaction Applications As Well As Datastores
- Rights (And Thus Enforcement) Applied At The Point Of Creation (Or Discovery,) At The Data-Element Level
- Choke Points Between On-premise, Off-premise, And Between Cloud Services Enforce Policies At The Data Level, Enforced By Encryption/DRM
- Rights Transfer And Enforcement Are Maintained Between State Changes
- Don’t Expect Your IaaS Provider To Do Any Of This For You; They Are Blind (By Design) To Most Of Your Data

Are We There Yet?

Conclusion[?]

It's What You Do About It That Counts

Securing the "Centers Of Data" of the Future

+ The Utility Of Public Cloud Services Are Compelling But Change Is Hard; **Requires Fundamental Changes In Infrastructure, Programmatic, Management & Security Architectures**

+ Consumers & Data/Application Owners Must **Make Centers Of Data "Survivable" & Focus On Information-Centric Security** Practices That Can Leverage Existing And Emerging Vendor Solutions

+ The **Security Policies** That Govern Information And Assets **Need To Travel With Them**; We Need Consistency In Metadata

+ Infrastructure Must **Gain Intelligence & Context** Through Consistent, Standards-Based Telemetry, Correlation & Disposition With Shared Execution

Key Takeaways

+ Not All Public Cloud IaaS Offerings Are Created Equal. Differentiation Based Upon Networking, Security, Transparency/Visibility & Forensics

+ Public IaaS Clouds Can Most Definitely Be Deployed As Securely Or Even More Securely Than Those In An Enterprise...

+ ...However, They Require Profound Architectural, Operational, Technology, Security and Compliance Model Changes

+ Time To Get The Bell Bottoms Out Of The Closet: What's Old Is New Again - Survivable Systems & Information Centricity

It Takes A Village...
Potemkin Need Not Apply

Get Involved:

cloud security alliance℠

http://www.CloudSecurityAlliance.org

Demand Transparency & Visibility

http://www.CloudAudit.org

Holler!

+ Christofer Hoff
+ www.rationalsurvivability.com/blog
+ choff@packetfilter.com [not work]
+ hoffc@cisco.com [work]
+ +1.978.631.0302
+ @beaker [The Twitters]

Image Attribution

- Skull Cloud: http://www.clippingimages.com/blog/tutorial-making-a-skull-shaped-cloud/
- Prince: http://www.virginmedia.com/images/prince-lovesexy-gal.jpg
- Star Trek Tribbles: http://i84.photobucket.com/albums/k3/NonStopPop/TheTroubleWithTribbles.jpg
- Jersey Shore: http://images.huffingtonpost.com/2010-03-01-shore.jpg
- Trebuchet:
- Trebuchet, Catapult, ballista: http://bit.ly/bCQTL2
- Trebuchet:
- Chinese Army: http://www.life.com/image/75878629
- Potemkin Village: http://www.flickr.com/photos/wili/274828493/
- Copernican: http://justanapprentice.files.wordpress.com/2009/11/copernicus-universe.jpg
- Floppy Disk: *http://www.flickr.com/photos/yaal/162100723/*
- Switchboard Operator: http://afeatheradrift.files.wordpress.com/2009/06/lily-tomlin-telephone-operator.jpg?w=300
- USB Hub: http://technabob.com/blog/wp-content/uploads/2009/05/usb_lego_hub.jpg
- Hall Of Mirrors: http://historyofeconomics.files.wordpress.com/2008/08/hall-of-mirrors.jpg
- Ritalin: http://commons.wikimedia.org/wiki/File:Ritalin-SR-20mg-1000x1000.png
- Obama Vulcan:http://www.flickr.com/photos/alexisalex/3519670022/ - Shan Carter
- Apple 1984: http://upload.wikimedia.org/wikipedia/en/2/22/Ad_apple_1984_2.png
- Gangsta Chimp: http://www.mattcioffi.com/samples/gangstaChimp24.jpg - Matt Cioffi
- Nostradamus: http://omarab.files.wordpress.com/2009/12/nostradamus-1503-15663.jpg
- Castle: http://www.jokerjitsu.com/images/medieval-castle.jpg
- Apathy: http://www.flickr.com/photos/comiccharacters/3792479746/sizes/l/
- W.O.P.R. http://www.hexkey.co.uk/lee/log/media/2009/08/wargames-wopr-med.jpeg