COMCAST

Patterns for Service Security in Hybrid Public/Private Cloud Deployments

Gulrukh Ahanger

VP of Product Development

2014.04.23

COMCAST

The cloud is here!
(and it always has been)

Wheel of reincarnation

MULTICS
Mainframes
Client/Server

COMCAST

The cloud is here!
(have we seen this before?)

Cloud Computing: The Mainframe Reincarnated

http://de.sys-con.com/node/723583

The Cloud isn't a mainframe. Seriously.

http://siliconangle.com/blog/2009/10/23/the-cloud-isnt-a-mainframe-seriously/

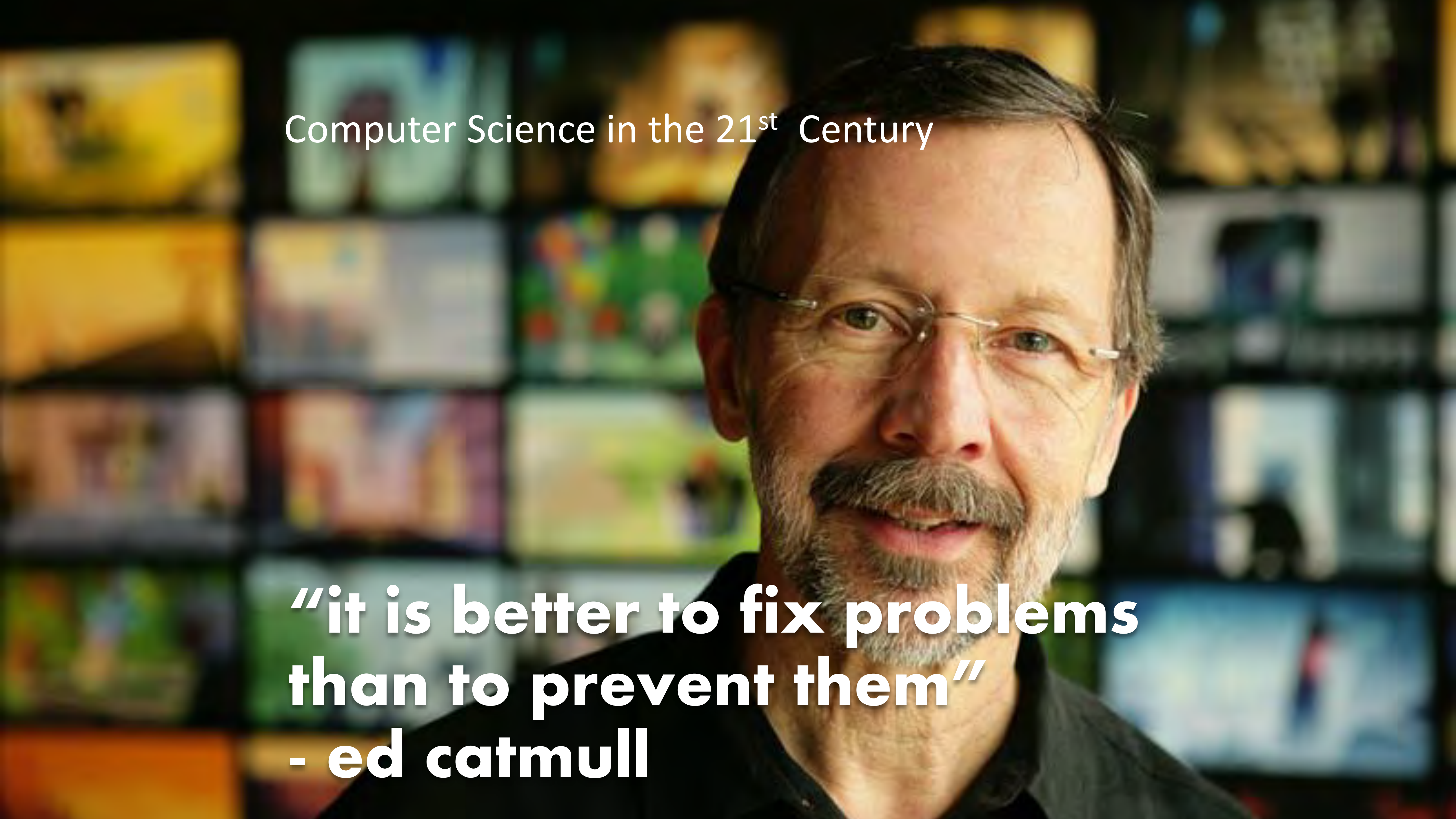
COMCAST

Many similarities, but also a few key differences:

- “n-datacenter”, not multi-datacenter
- Sharded for resiliency, not geographic availability
- “Scarcity” vs “Abundance” as a design philosophy
- Designed with failure as an operating mode

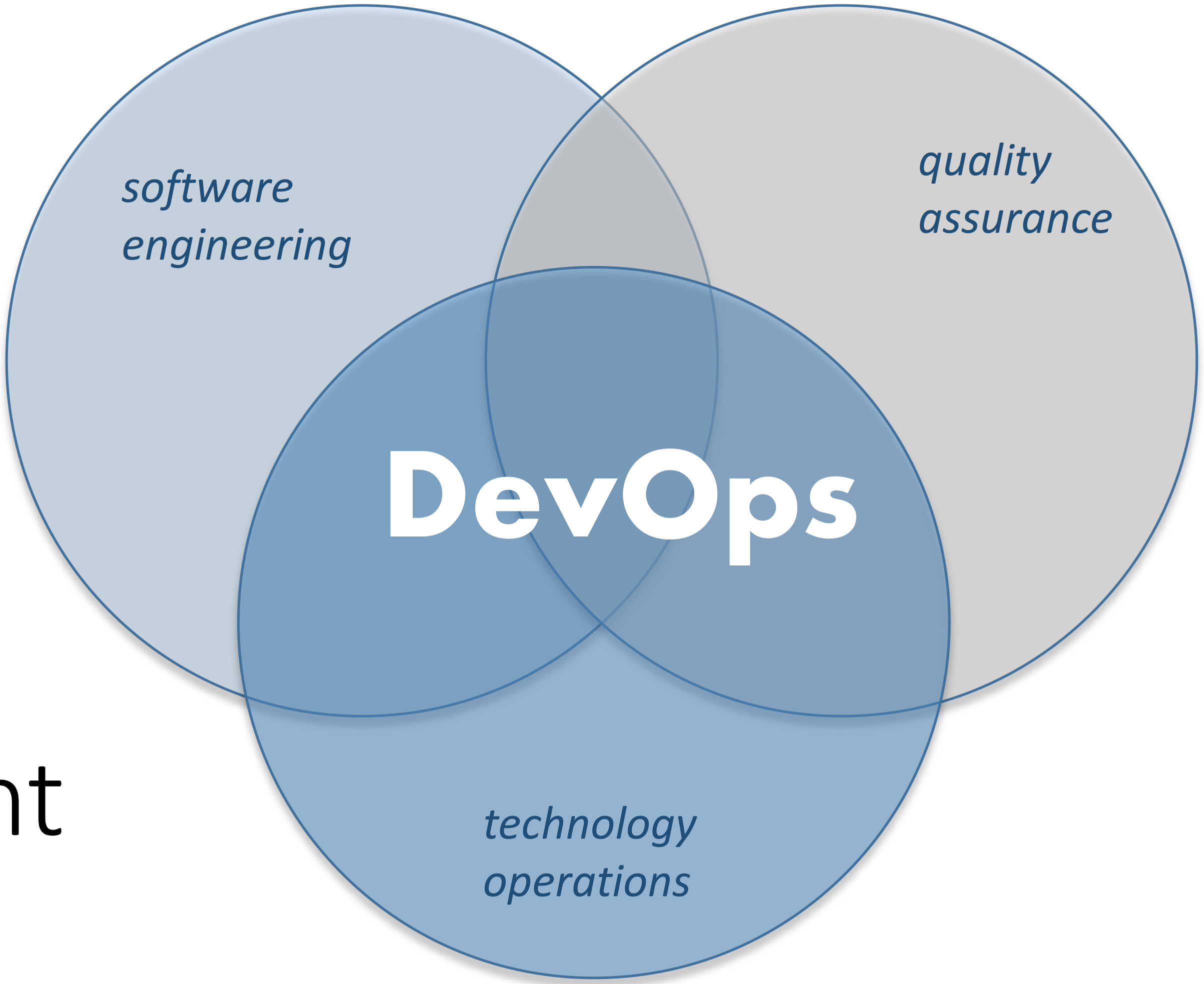
Computer Science in the 20th Century

“measure twice, cut once.”



Computer Science in the 21st Century

"it is better to fix problems than to prevent them"
- ed catmull



Deployment model evolution

AKA the science of software

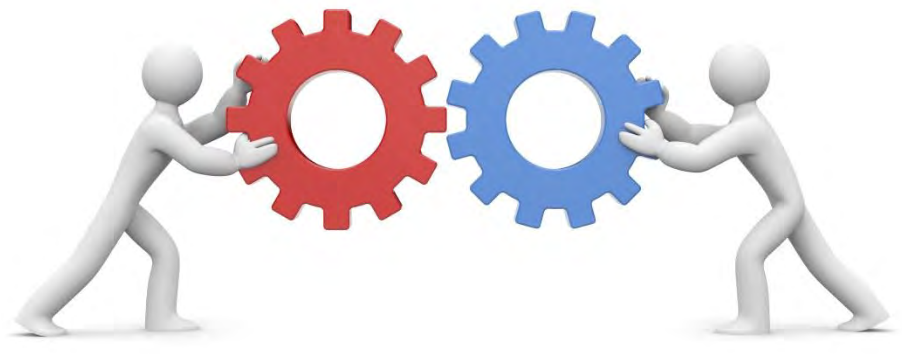
SDLC → Agile → DevOps

Process → Incremental → Autonomous*

Large systems have emergent properties

Implications of DevOps

- tools not tickets, dev=ops
- federation of data and infrastructure , not "source of truth"
- fail fast: ops improvisation is not good
- "traffic shaping" is the new "deploy"



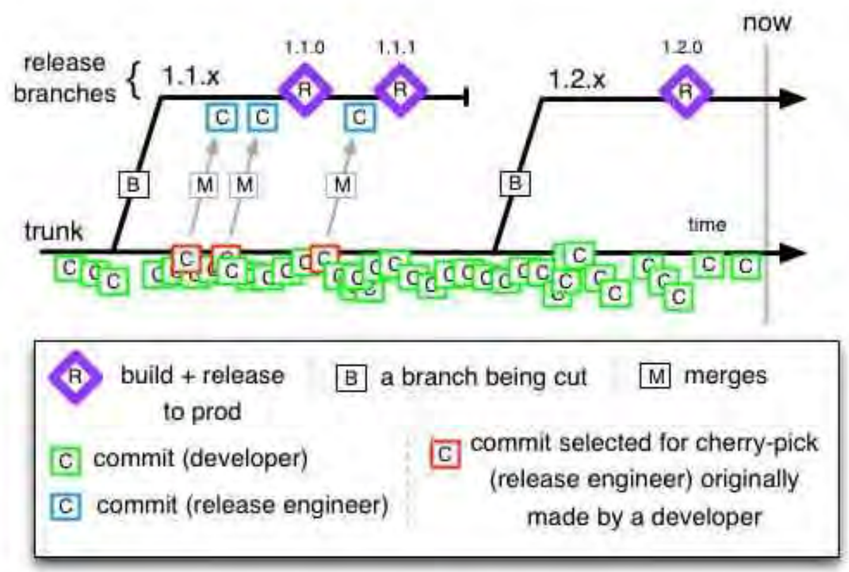
Traffic Shaping is a Core Primitive

- failure is an operating mode
- fail back/fail forward
- multi-instance
- incremental validation
- backwards compatibility
- forwards compatibility
- infrastructure decoupling

COMCAST

Release Management is now *very* different

- trunk based development: branch for release
- continuous Integration
- sprints are when you ship
- shipping happens all the time

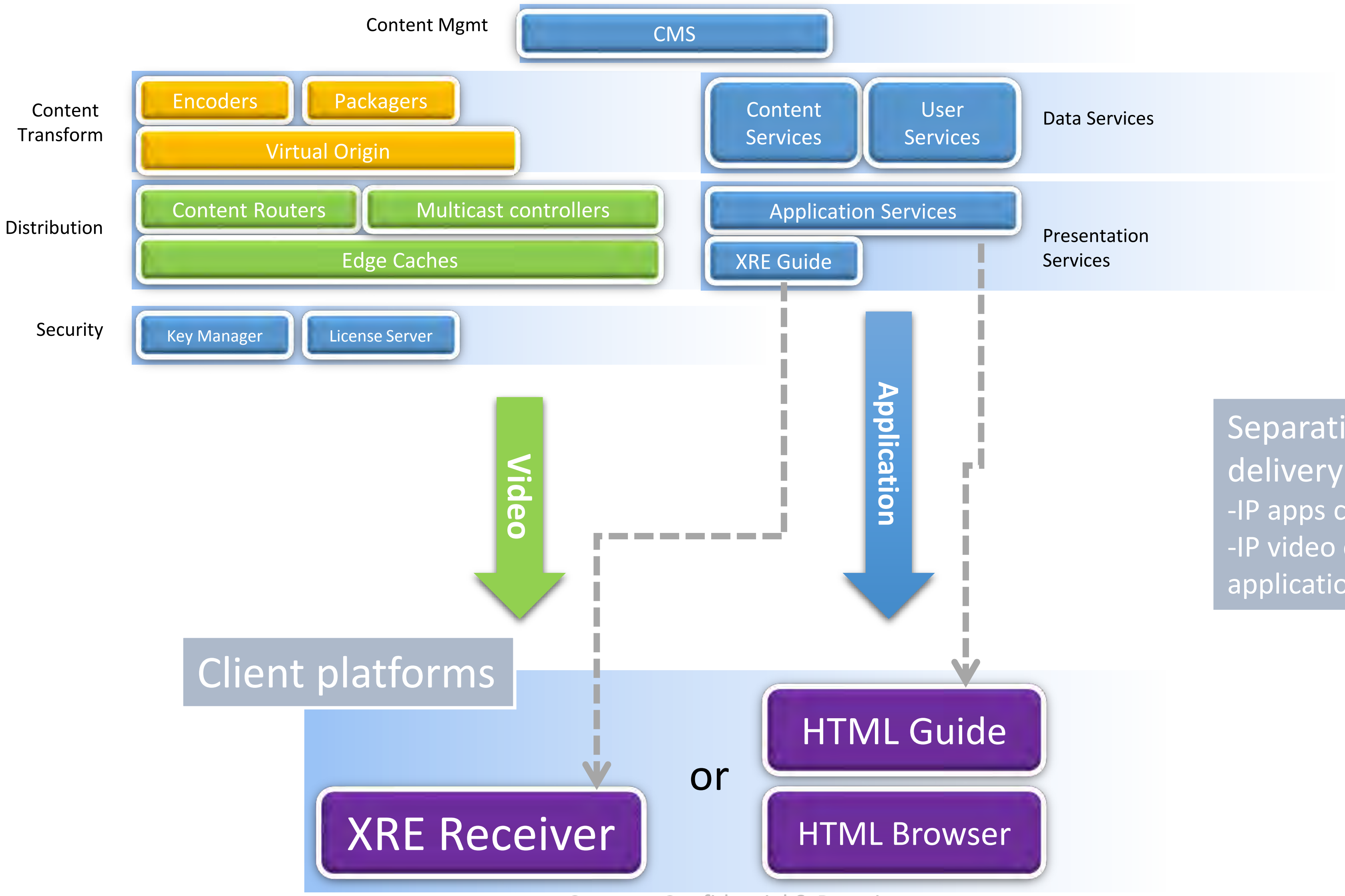


Reuse Equivalency Principle

- the unit of re-use is the unit of release
- effective re-use requires tracking release
- the unit of release should be the unit of fail

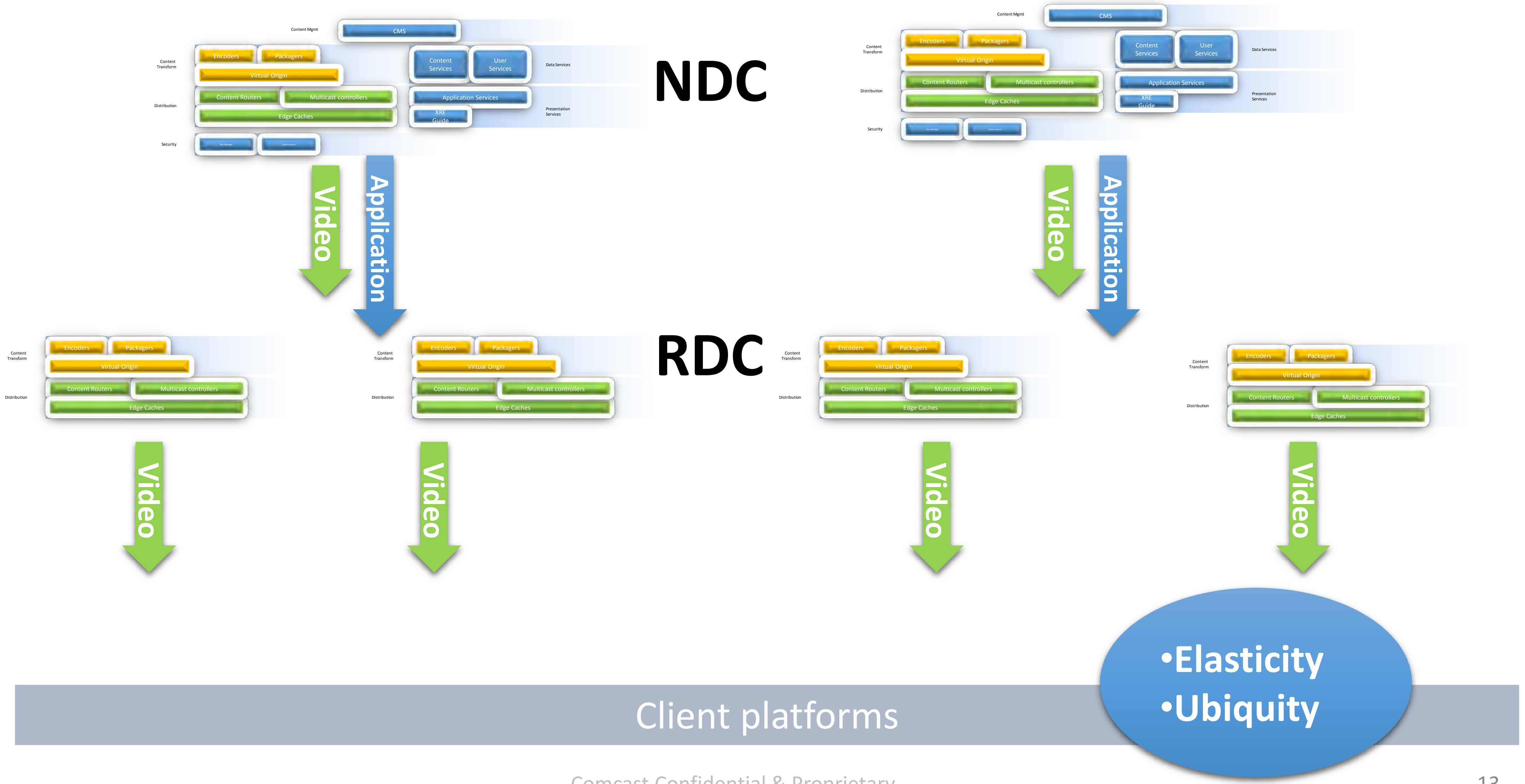
ACID as applied to deployment, not just data

X1 platform: *logical view*

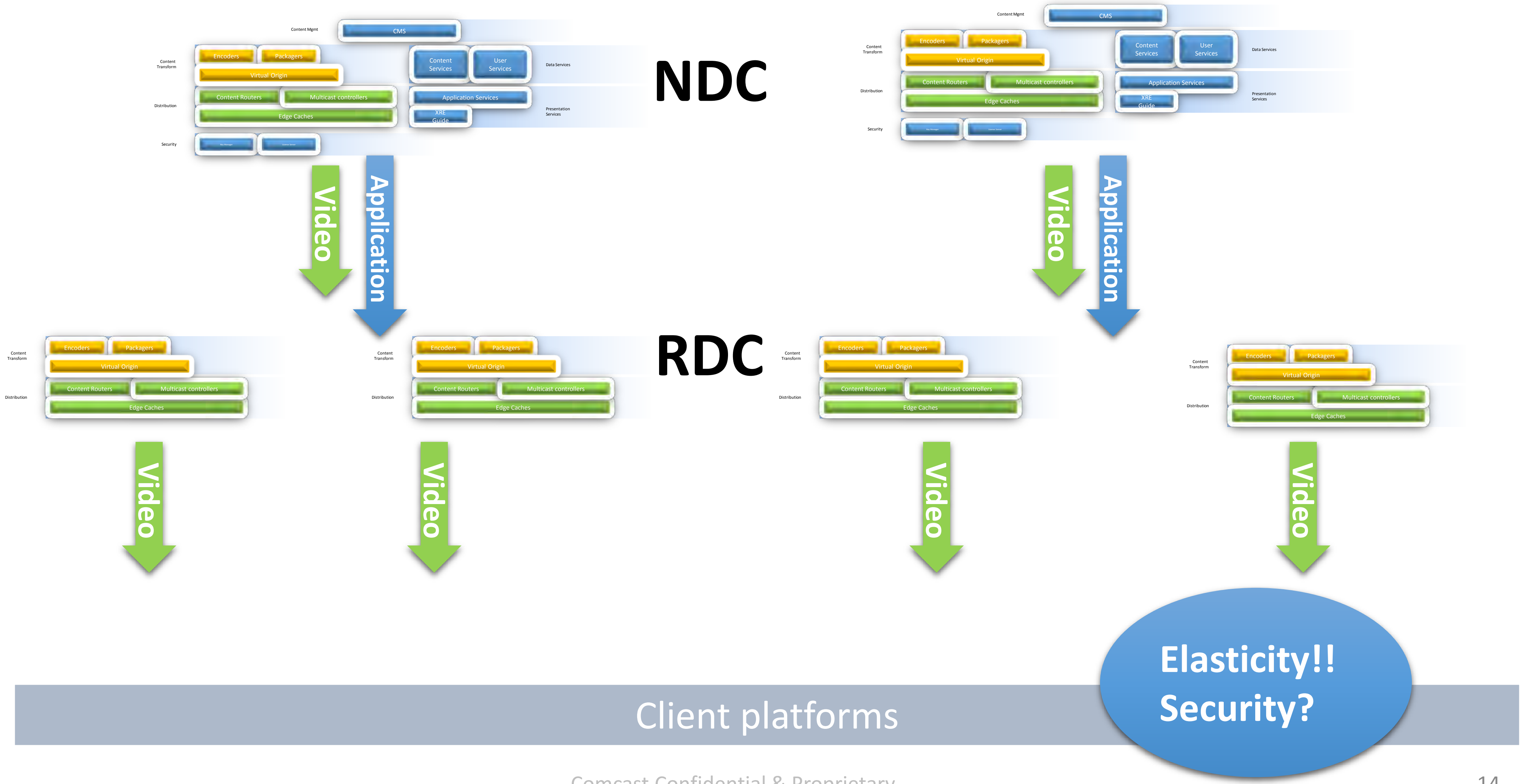


Separating discovery from delivery means:
-IP apps can leverage QAM video
-IP video can be delivered to many applications

X1 platform: ***physical view ::: NO PODS :::***



X1 platform: *replicated to AWS/etc. (public)*



COMCAST

Important similarity to mainframe is of “remote trust”

Traditional enterprise security: Hard shell, soft inside

Perimeter complacency!

COMCAST

Security Considerations

Availability	Privacy
Redundancy	Access controls
Responsiveness	Theft of data

COMCAST

Redundancy

Anti-Pattern	Pattern
Oracle w/Golden Gate	CouchDB
MongoDB	Cassandra
	PostgreSQL

Key attributes:
multi-master replication, Availability/Partition tolerance, Eventual consistency

COMCAST

Responsiveness

Anti-Pattern	Pattern
Microservices (s/c)	Macroservice (s/c)
Geo-stickiness (because zombies)	Persistence (s/s)
	HA
	Realtime metrics

Key attributes:
Limit surface area of attack, connection pooling, abstraction of performance

COMCAST

Access Controls

Anti-Pattern	Pattern
DB access	Service Auth/Auth
	Limited Access
	Network protection
	Advance Analytics

Key attributes:
Access based on User Context, risk-based approach: extra monitoring, locate meaningful signals

COMCAST

Theft of Data

Anti-Pattern	Pattern
Encryption	Encryption
Multiple Standards	SSL
	Public vs Private

Key attributes:
Data theft, Multi-tenancy

COMCAST

Summary

- “Flatter” networks in the cloud mean services have to consider security
- Security has to consider scalability and be built-in, not bolted on
- Data security and service security are not the same thing
- Atomicity of services defines the security surface and the scalability surface: avoid “stacks” (AKA coconuts)