AEROSPACE
Assuring Space Mission Success

The SMC Enterprise Ground Architecture Cloud Studies

Commercial Capabilities Working Group (CCWG)
Joe Bannister, Mel Cutler, Doug Enright, Bob Lindell, Eric Coe, Eltefaat Shokri, Richard Yee, Wayne Wheeler, Craig Lee

The Aerospace Corporation

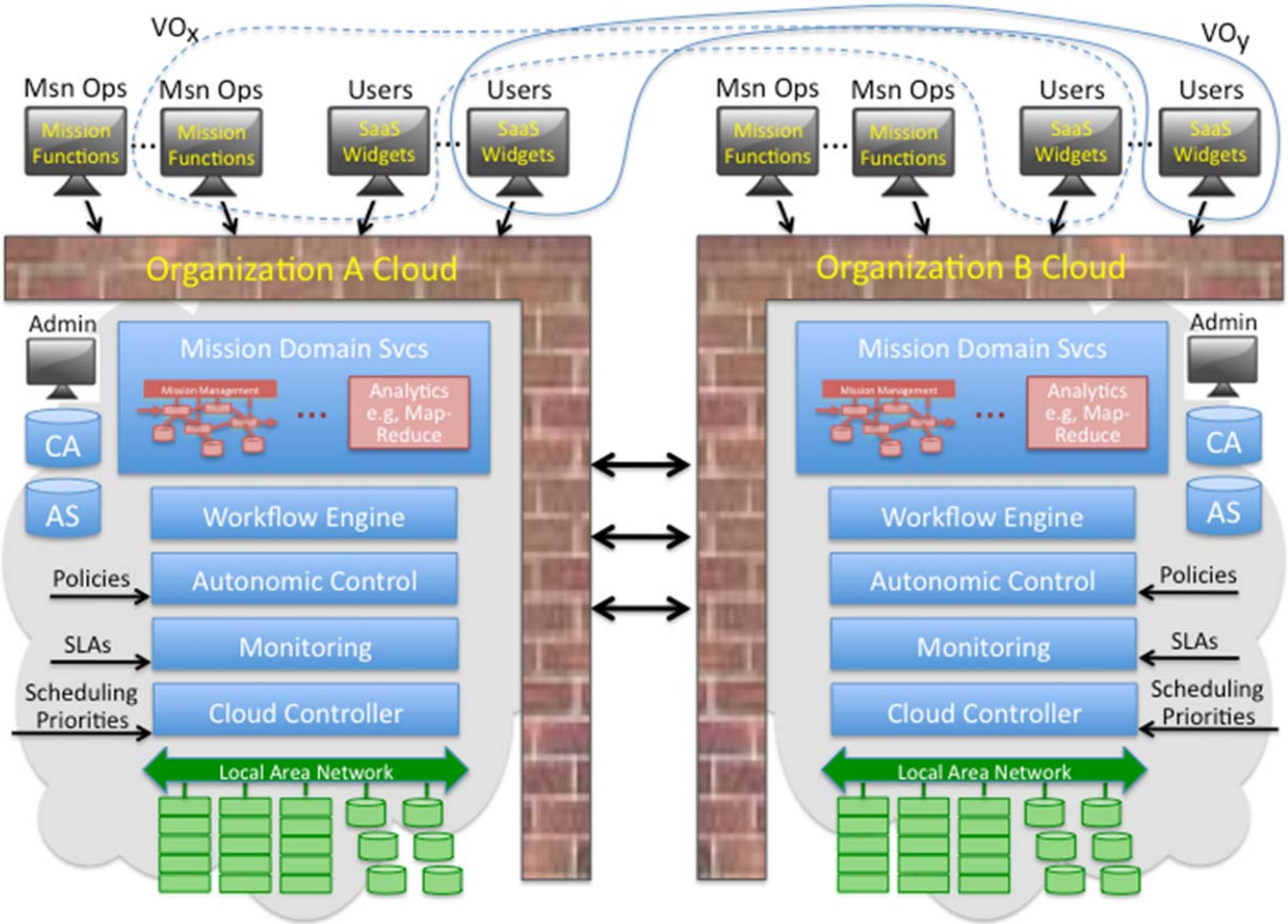
March 4, 2015

© The Aerospace Corporation 2015

CCWG Study Phase II Objectives & Approach

- Primary Objectives
  - *Understand the applicability, benefits, and challenges of using new techniques, tools, and technologies in future (enterprise) ground architectures*
  - *Develop Roadmap that identifies critical capabilities*
    - Characterize and develop approaches that are feasible and improve key qualities (flexibility, resiliency, affordability, performance)
  - *Enhance Aerospace capabilities to support architectures and requirements development for future SMC ground systems*
- Operational Concept
  - *Leverage experiments already done by others, reuse software*
  - *Refine Roadmap through Aerospace-executed experiments in a laboratory environment*
    - Target SMC-specific needs not addressed/covered by others
    - Gain hands-on experience with technologies
    - Build lab using same technologies we are evaluating (e.g., hybrid private/commercial cloud, virtualization, software-defined networking)

Notional Target Architecture



Notional Target Architecture Description

- Multiple cloud data centers, possibly supporting different organizations
- Each site possibly operates their own
  - *Compute, Storage, Network resources*
  - *Monitoring Infrastructure*
    - Monitors for performance requirements (SLAs) and integrity
  - *Autonomic Control Agent(s)*
    - System Policies are enforced based on monitored information
  - *Missions*
    - Complex sets of applications
    - Possibly running in their own *virtual cloud* or *virtual data center*
  - *Identity provisioning*
    - Certificate Authorities (CA) issuing PKI certs
    - Attribute Servers (AS) defining authorization policies
- Some missions may be distributed
  - *Mission components operate and interact across different sites*
  - *Sites can spare for one another -- provide fail-over for critical ops*
- Multiple, distributed sites are *federated*
  - *Supports both local and remote users*
  - *Federated access control is managed through Virtual Organizations and attribute-based policy enforcement*

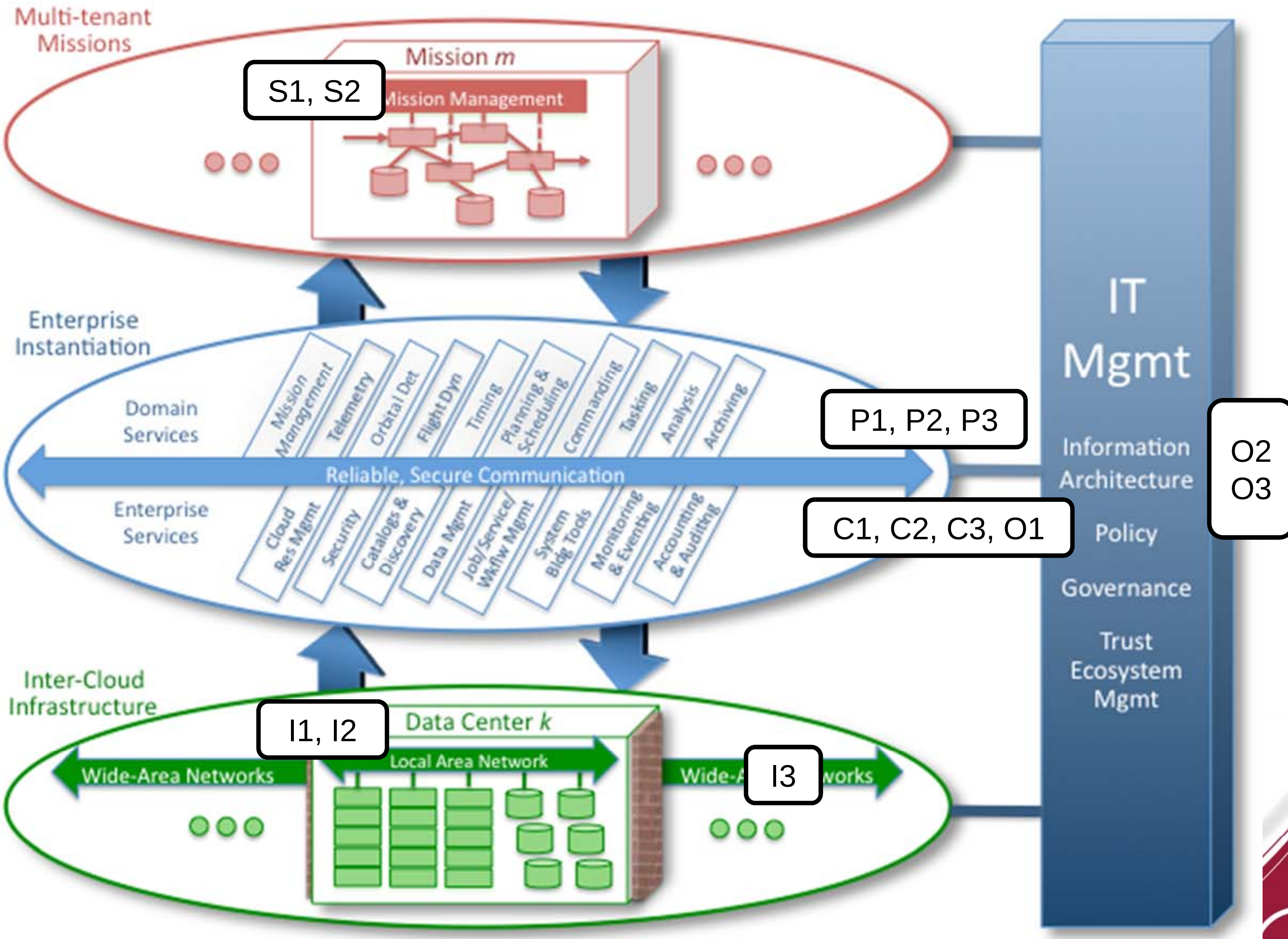
CCWG Phase II Candidate Tasks: Top-Level View

Area	ID	Experiment	F	R	A	P
IaaS	I1	Virtualization Survey	•		•	•
	I2	Cloud-based Failover		•	•	
	I3	Route Hopping	•	•	•	
PaaS	P1	"Cloudy GMSEC"	•			
	P2	Platform Architecture		•	•	
	P3	Proactive SLA Monitoring		•	•	
SaaS	S1	Mission-Level SLAs			•	•
	S2	Location Transparency	•	•	•	
Configuration	C1	Continuous Integration/Build Automation	•		•	
	C2	Automated Provisioning	•	•	•	•
	C3	Workflow Management		•	•	•
Operations	O1	Proactive System Monitoring		•	•	
	O2	FIM and Virtual Organizations	•	•		
	O3	Cyber-Security Techniques		•		
	P3	*(Proactive SLA Monitoring)*		•	•	

F/R/A/P = Flexibility, Resiliency, Affordability, Performance

Note: Some experiments with limited F/R/A/P focus have other benefits (e.g., cost savings)

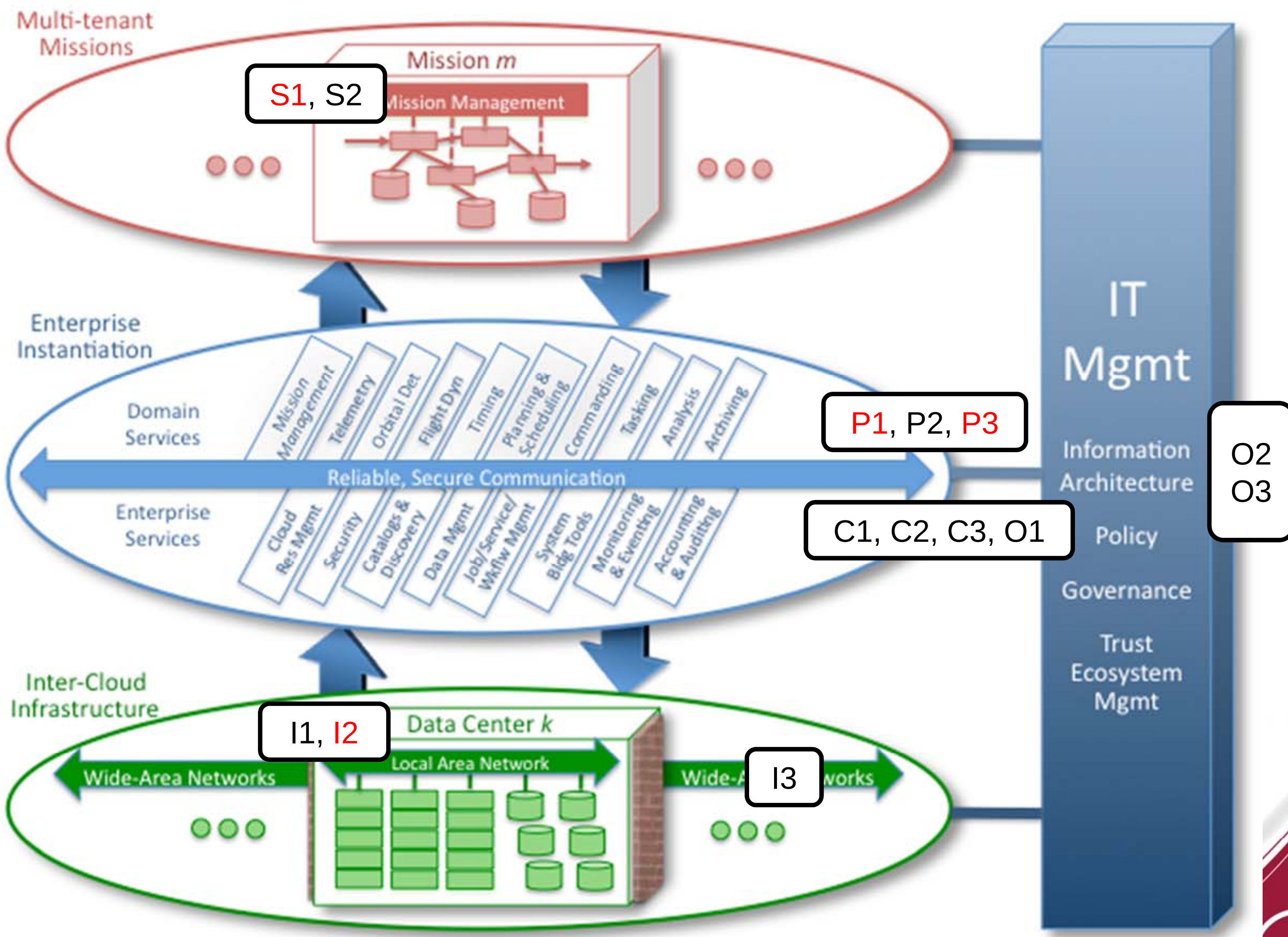
Tasks Mapped to "3-Ovals" Reference Model



Phase II Experiment Strategy

- Initial set of 3 experiments – Phase IIa
  - *Selected based on what can be accomplished in a short time with computing resources on hand or readily obtainable*
  - *May or may not relate directly to the early stages of the roadmap*
- Follow-on experiments – Phase IIb
  - *Build on Phase IIa experiments*
  - *Enable larger, more extensive, experiments by the acquisition of additional resources*
  - *Combined physical and virtual test infrastructure*
    - Owning key hardware enables us to do experiments at any level in the software stack, e.g., router, cloud controller
    - Acquire additional virtual resources on-demand to enable experiments at a scale not otherwise feasible

Initial Experiments Include All Architecture Layers



Phase IIa Experiments and Results

Task I2 – Cloud-based Migration and Failover

Demonstrating how failure of entire application is not noticed by the user

Study Objective

- Understand how off-the-shelf technologies can support application fail-over in a cloud environment
- Explore abilities and role of leading-edge software-defined networking capabilities in supporting fail-over
- Explore utility of IT automation tools to rapidly and automatically configure environments

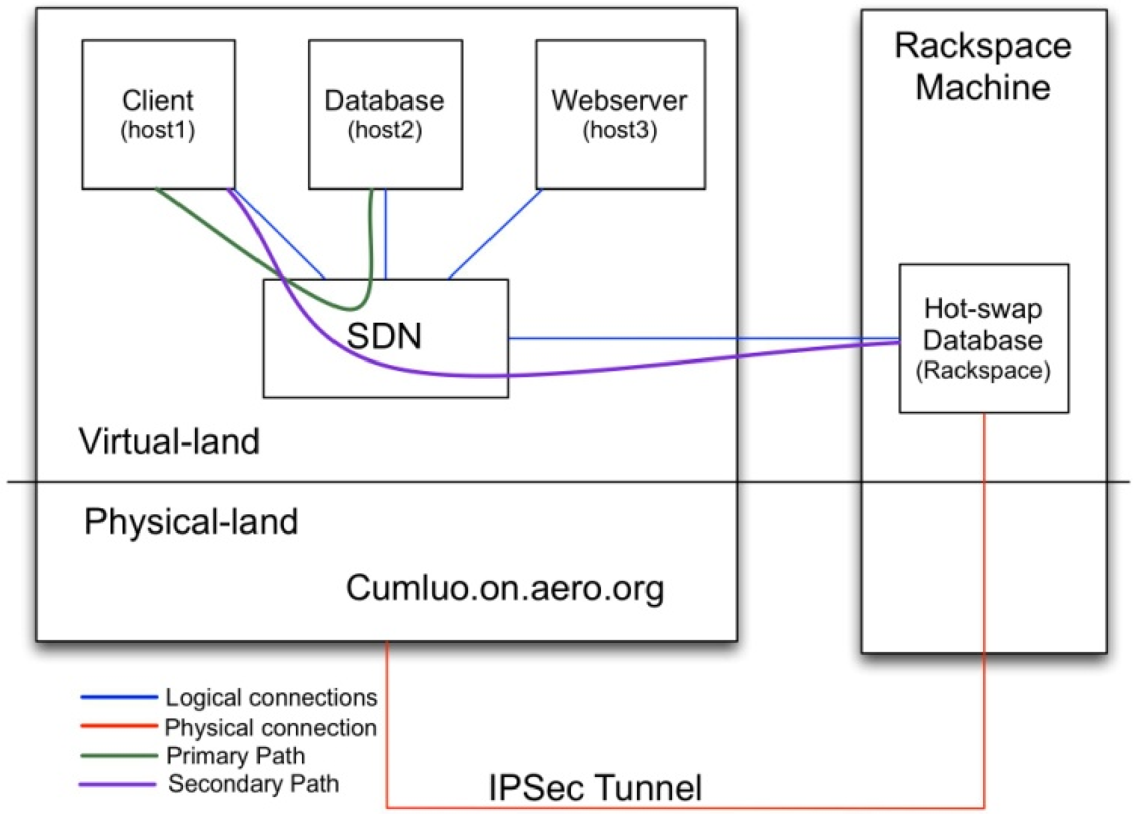
Research Approach

- Set up a “two zone” cloud on different providers
- Use IT automation tools to install and configure mission application (simulant) on both providers (one primary, one backup)
- Trigger fail-over from primary to backup
- Use software-defined networking technology to reconfigure network to point at backup
- Measure key performance parameters (downtime, time to reconfigure elements)

Cloud-based Failover Demonstration

- Demonstrated recovery from a single tier failure in a multi-tier architecture
  - *Database Tier failure results in failover to a hot-swap resource in a geographically dispersed data center (GDDC)*
  - *Failover and failback happen on the order of hundreds of milliseconds*
  - *Used Software Defined Networking to redirect data flows to GDDC*
  - *Transitioned from a privately owned data center resource to a publicly available data center resource*

- Automated system configuration and policy/compliance enforcement using *puppet*
  - *One command turns "plain vanilla" virtual machine into fully-configured server*



Findings and Observations

- SDN is likely to be a game-changer
  - *Failover with less than N-times the hardware*
  - *Completes the "software-managed data center" concept (virtual machines + virtual storage + virtual networks)*
  - *Potential improvement in cyber-resilience (can easily create and manipulate firewalls, enclaves, isolated networks without pulling any cable)*
  - *Existing SDN technology somewhat immature but this is likely to change*
    - Interfaces, APIs not well-integrated with other products…yet
- IT automation should be the rule, not the exception
  - *Writing policies only slightly more painful than doing the configuration manually*
    - Tools can then enforce policies automatically across many machines
    - Easier to tweak policy and re-enforce than to manually re-configure
  - *Even top-tier tools lack maturity in some areas, but this is likely to change*
- IT automation + SDN could be a very powerful combination
  - *But this is not commonly done in the mainstream marketplace (yet)*

Task P3/S1 – Dynamic Resource Management and SLA Enforcement

Demonstrate important role of resource management in future ground systems

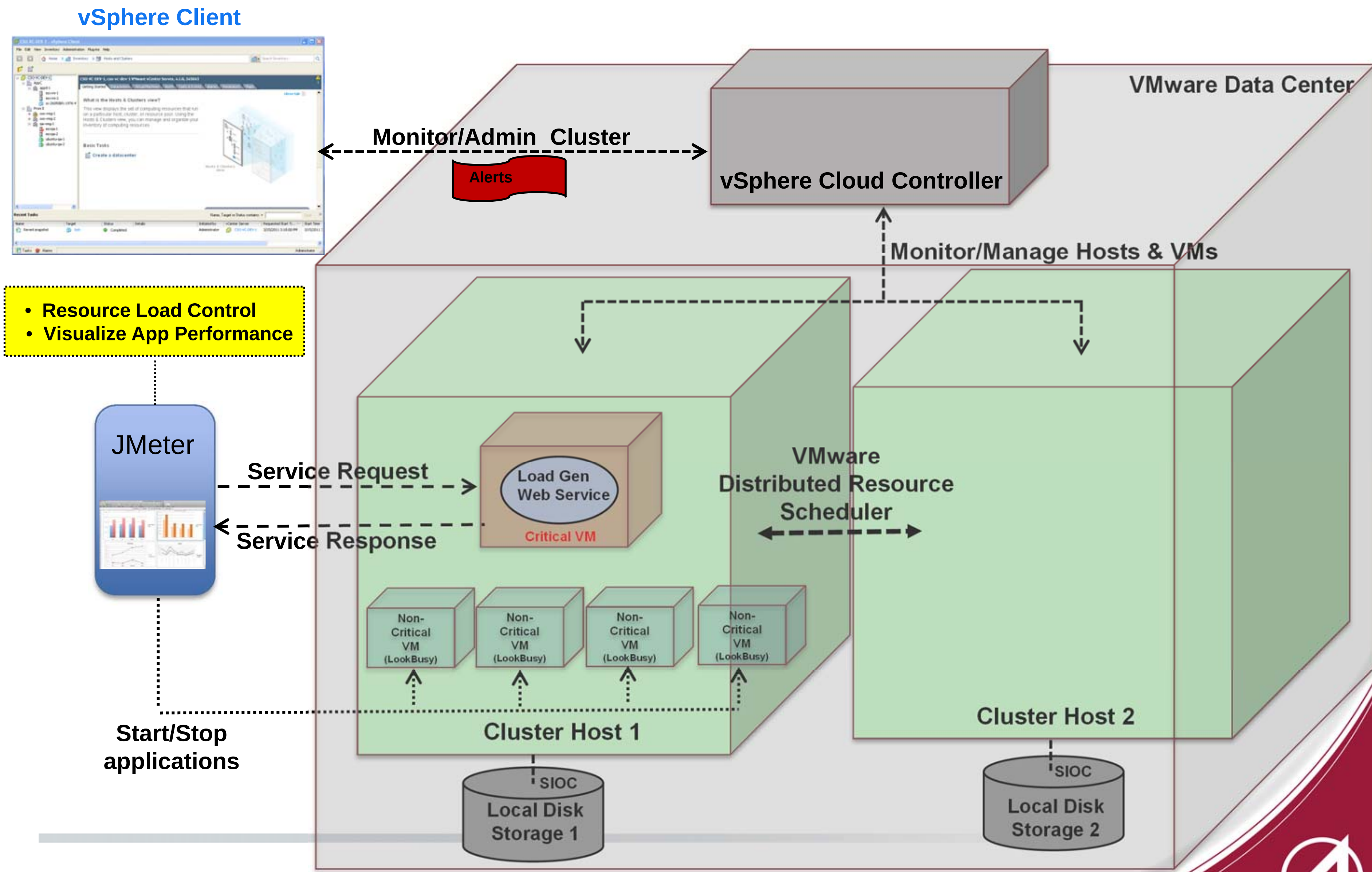
Study Objective

- Partial validation of earlier assessment on availability of commercial products/features for flexible and dynamic resource management of data centers.
- Creation of a foundation for (i) future platform-level SLA management, and (ii) future mission-level SLA management.
- Experiment Scope:
  - *Real-time monitoring of physical hosts and virtual machines with respect to system resources (e.g., CPUs, memory) and resource utilizations.*
  - *Resource utilization anomalies (e.g., excessive resource utilization), detection and alarm/alert notification dissemination.*
  - *Dynamic (and proactive) mitigations for returning resource utilizations to acceptable limits*

Research Approach

- Experiment environment
  - *VMware-based cloud environment*
  - *Use of vSphere client (a visualization tool for data center management)*
  - *Use of a small corporate VMware cluster (with three physical hosts)*
- Experiment and evaluate VMware data center resource/performance monitoring & management capabilities and tools for:
  - *Real-time monitoring of physical hosts and virtual machines*
  - *Resource utilization problem detection and alarm notification via emails or SNMP traps*
  - *Proactive mitigations to return resource utilizations to acceptable boundaries using vCenter*
  - *Apply problem mitigations, e.g., rebooting, suspension, migration, etc.*
  - *Measuring the effectiveness of various mitigation tactics*

Experiment Setup



Findings and Observations

- Mature COTS products for resource monitoring/management are available
  - *Open source products (e.g. OpenStack) may not be ready for prime time with regard to real-time resource management*
- VMware's vSphere is a leading product
  - *Easy to use GUI-based monitoring and management.*
  - *Rich SDK with REST API to integrate with other enterprise management products.*
- VMware vSphere limitations
  - *vSphere (at least an early version we used) resource utilization statistics are confusing and statistics are not updated frequently enough to be of use for real-time monitoring.*
  - *There is limited information (and controllability) on hypervisor behavior, which may lead to inefficient human-in-the-loop resource management.*
- *Although automatic mitigation using Distributed Resource Scheduler might be very effective for many commercial applications, it might be insufficient to enforce SLAs of mission critical applications and systems.*
  - *Due to coarse-grained DRS policy.*
- *Infrastructure resource management alone may not be sufficient for application SLA management and enforcement.*
  - *Platform and application level monitoring are required.*

Tasks P1: Cloudy GMSEC

Leverage existing, proven GMSEC capabilities

Study Objective

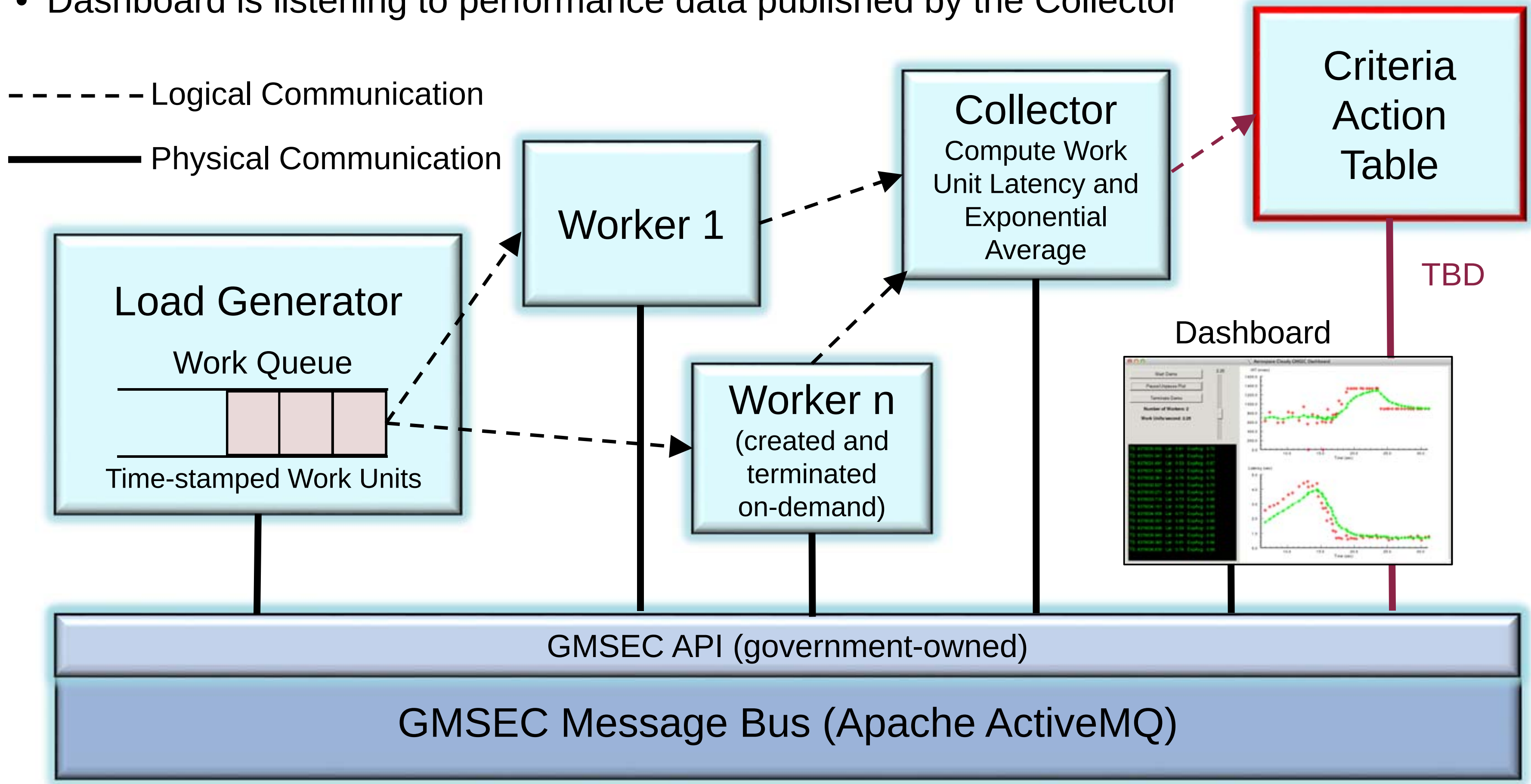
- Many missions will be performance-critical/sensitive
    - *"Best effort" cloud resources may not suffice to meet mission requirements*
- Some missions will have dynamic, unpredictable "surge" requirements
    - *Previously addressed by over-provisioning with dedicated hardware*
- This is antithetical to cloud computing
    - *Multi-tenant environment where utilization and costs can be better managed*
- *There must be a mechanism whereby multiple, multi-tenant missions have a reasonable guarantee that performance requirements will be met*
- Dynamic, machine-enforceable SLAs address this need

Research Approach

- Approach: Use GMSEC as a Monitoring and Control Tool
    - *Message bus approach with government-owned API*
    - *Catalog of GMSEC-compliant components available*
- By provisioning GMSEC service modules, on-demand, in a cloud, we will essentially demonstrate *Ground Systems as a Service*
    - ***GSaaS is a possible architectural approach for a future SMC EGA***

GMSEC Demonstration – Multiple Workers

- Collector monitors work unit latency and worker inter-arrival-time
  - Computes exponential average to smooth out higher frequencies
- If latency gets too high, an additional worker is started
- If workers must wait too long for work, the last worker started gets terminated
- Dashboard is listening to performance data published by the Collector



AEROSPACE

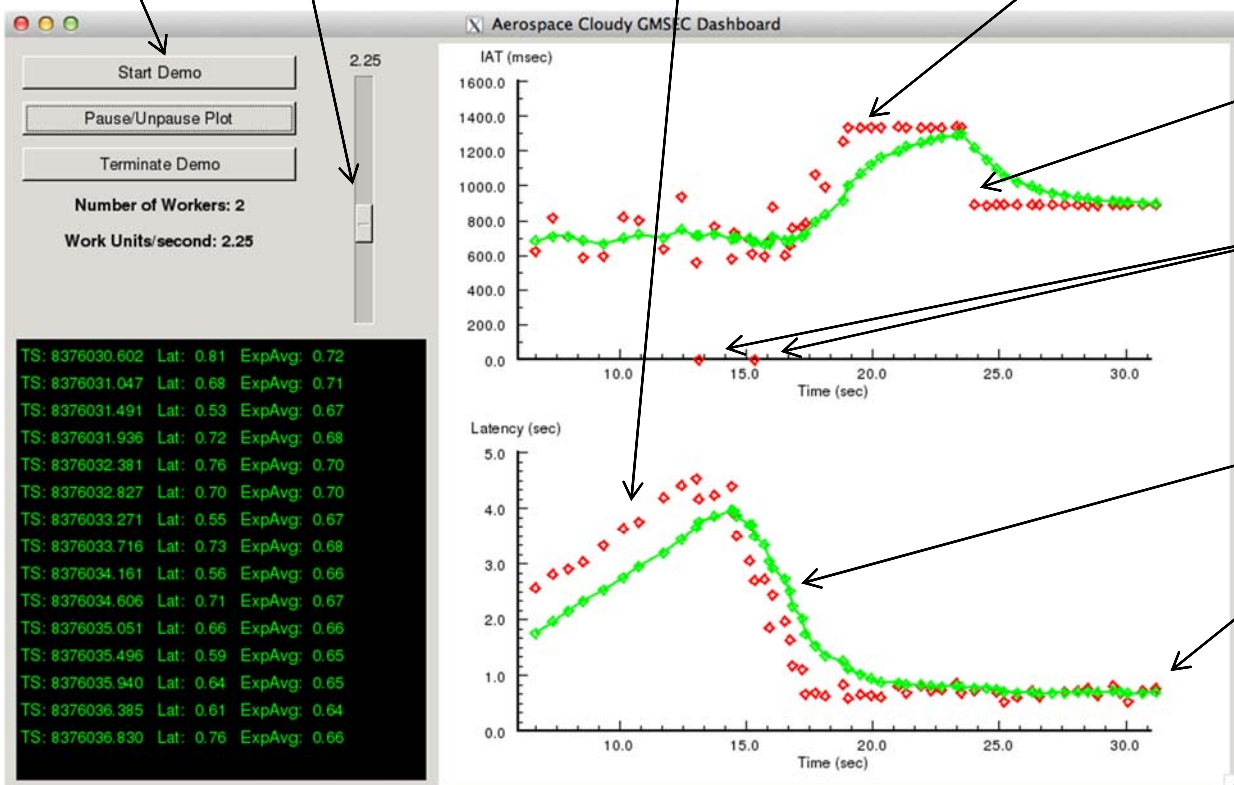
Scenario Performance on Dashboard

3) Load increased or decreased with slider

4) With increased load, Work Unit latency rises since there are not enough Workers to keep up

1) Demo started with “Start Demo” button

7) Inter-arrival time of Work Units at each Worker rises



8) Worker #3 terminated

5) Workers #2 and #3 started

2) “Chatter” starts in text message box

6) Latency falls

9) Latency remains stable with two Workers

AEROSPACE

Findings and Observations

- Development & Test plan needed for SLAs
  - *What are the simplest SLA mechanisms that "scratch the itch" for the most mission requirements?*
- Capacity Planning & Management
  - *How to estimate query requirements, load demand, time-to-completion*
  - *How to support reasonable loads to produce reasonable times-to-completion*
  - *How to manage sets of users such that no one user is disruptive*
  - *How to on-board requirements from other organizations*
- Cyber-security Implications
  - *As clouds become larger and more widely used, there will be more automated tools, i.e., autonomic behaviors*
  - *Autonomic agents become a threat surface -- compromising an agent that controls system behavior would have broad impact*
- More realistic mission processing scenarios with targeted experimental tasks in a scaled-up experimental testbed
  - *Need to demonstrate "Ground System-as-a-Service" possibly using more "realistic" GMSEC test cases that are closer to actual ground systems*
- Many issues to investigate!

Overall Phase IIa Findings and Observations

- It's necessary to "own" key parts of the infrastructure for specific experimental purposes
  - *Software Defined Networks, VM scheduling/migration*
- Autonomic techniques need to be applied
  - *Monitoring, Analysis, Planning and Execution (MAPE) autonomic control loop*
  - *Necessary to demonstrate operational effectiveness in a high availability environment*
- Autonomic Control coupled with IT Automation has broad implications
  - *Performance management, health & status, availability, resilience, intrusion detection*

Next Steps

- Larger test beds are needed
  - *Need to evaluate capabilities at scale*
- Larger experimental test cases are necessary
  - *More realistic mission processing scenarios*
- Many more outstanding issues/challenges at every level in the system software stack
  - *Integration of end-to-end capabilities*