



“Service Performance” Aspects for Cloud Service Level Agreements

Dr. Craig A. Lee, Senior Scientist, lee@aero.org
Computer Systems Research Department
The Aerospace Corporation

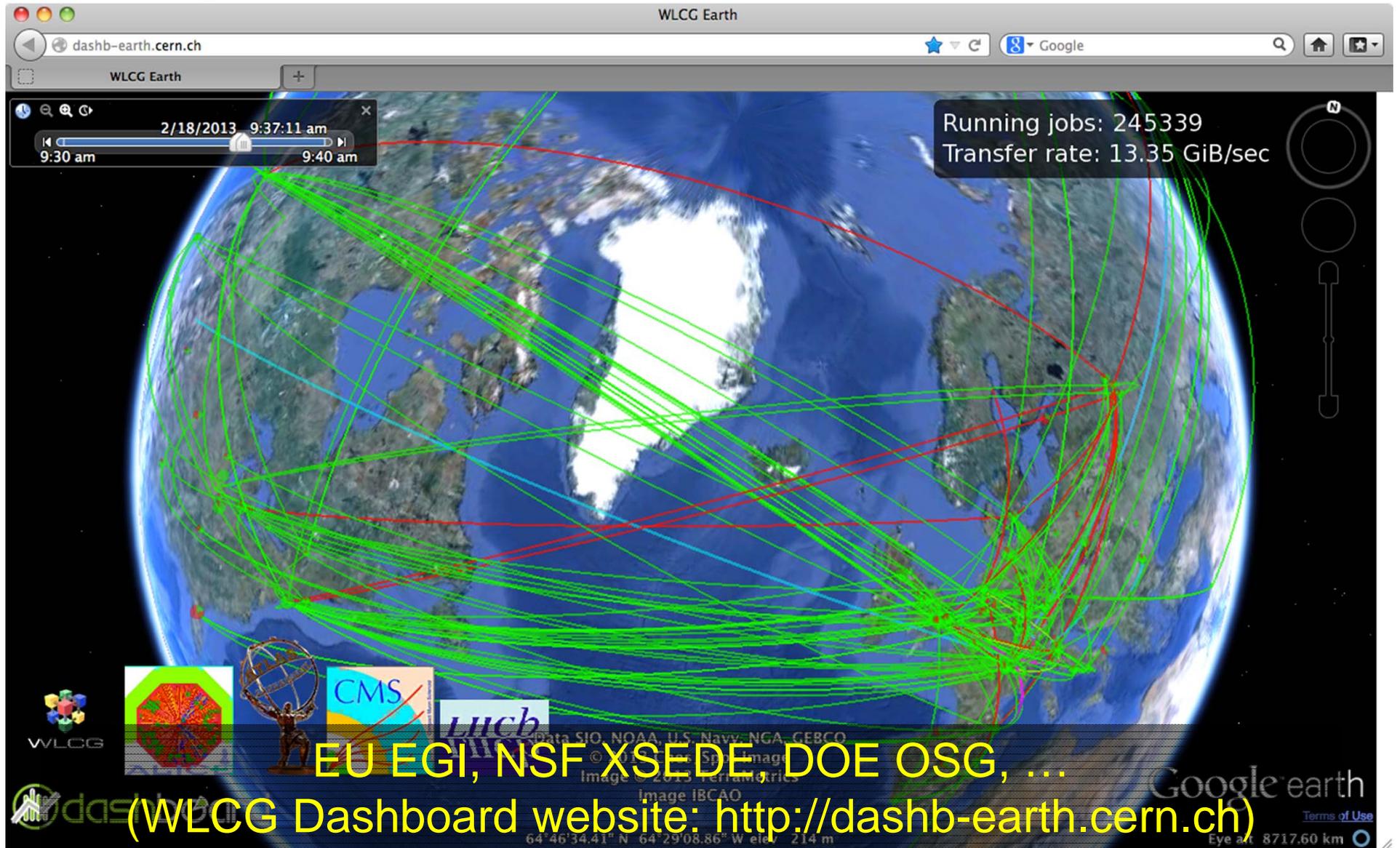
NITRD SLA Workshop
Arlington, VA, August 6, 2014

Why SLAs?

- Some applications will be performance-critical or performance-sensitive
 - *"Best effort" cloud resources may not suffice to meet mission requirements*
- Some applications will have dynamic requirements
 - *Some apps will have varying demands – surge -- at unpredictable times*
- Surge traditionally addressed by over-provisioning with dedicated hardware
 - *Dedicated system was sized for the worst-case, rather than the average case*
 - *Drove acquisition costs and operation costs for the entire life of the system*
 - *Example: Satellite ground systems*
- This is antithetical to cloud computing
 - *Multi-tenant environment where utilization and costs can be better managed*
- *Hence, the goal is to provide the user with a reasonable expectation that performance requirements will be met, through mechanisms that are reasonable for the provider to implement and support for multiple apps*
- These cannot simply be contractual SLAs
 - *These must be capabilities that a provider may provide and a user may use to keep applications "in spec"*
 - ***Dynamic, machine-enforceable SLAs***
- Work in this area has already been done in the Grid community



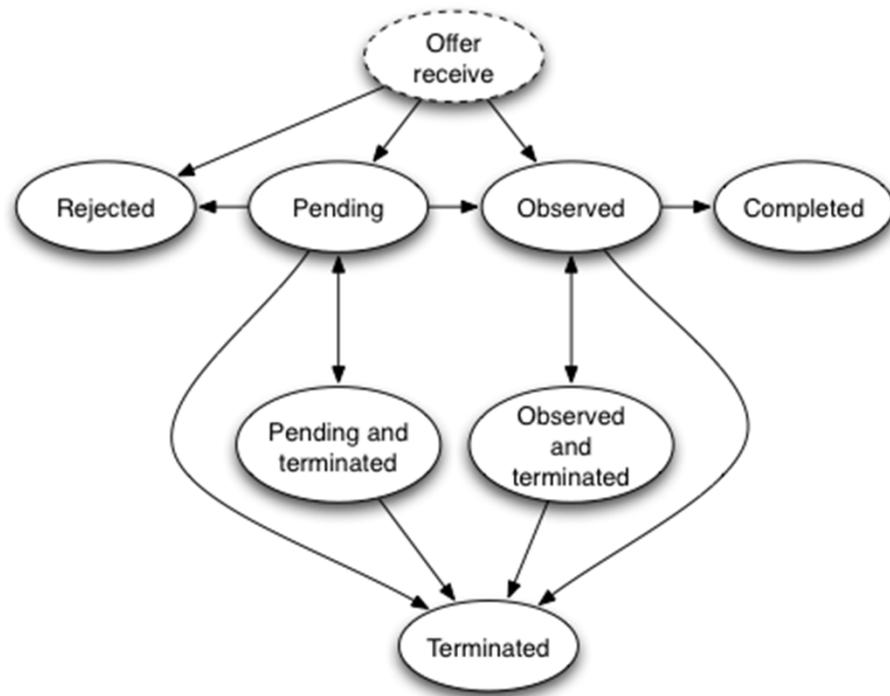
Grid Computing: “Big Science” Collaboration on a Global Scale



WS-Agreement, GFD.192

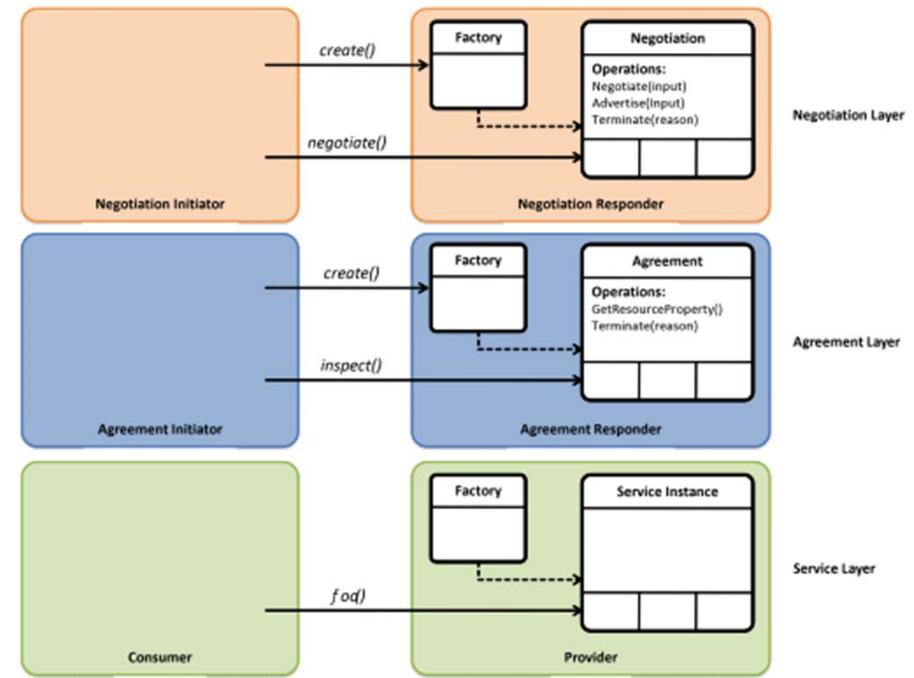


- Defines term language and protocol for advertising service provider capabilities, creating agreements based on offers, monitoring compliance, and penalties/rewards for non-compliance
- RESTful implementations exist



WS-AgreementNegotiation, GFD.193

- Defines an offer/counter-offer model for dynamic exchange of information between a negotiation initiator and responder
- Rounds of negotiation modeled as a rooted tree
 - States: *Advisory, Solicited, Acceptable, Rejected*
- Layered model separates functions and implementations



How to Use this in a Cloud Environment?

- WS-Agreement and WS-AgreementNegotiation are parameter (term language) and cloud agnostic
- Many SLA metrics possible:
 - *Memory/Disk (space: bytes)*
 - *Throughput (rate: x/sec)*
 - *Bandwidth (rate: bytes/sec)*
 - *Latency (time: t)*
 - *Time-to-Solution (time: t)*
 - *Availability (time ratio: percentage)*
- Application-level requirements must be mapped to infrastructure-level requirements
 - *This will be application-specific*
- To have "teeth", an SLA must be monitored *and* enforced
 - *WS-Agreement and WS-AgreementNegotiation are only the front-end of the SLA process*



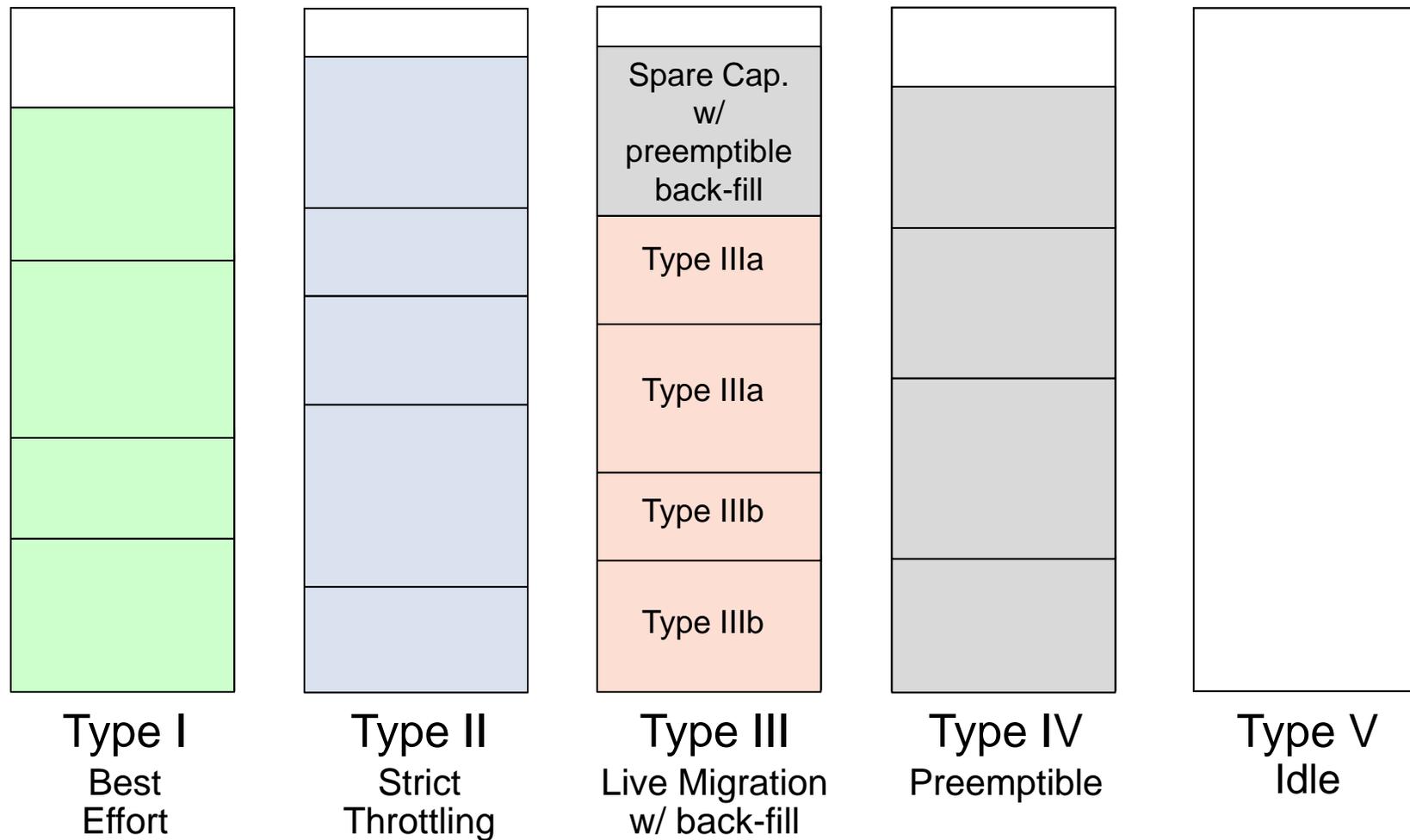
Basic SLA Functions

– *an Autonomic Control Cycle*

- Admission Control
 - *Mapping of app-level requirements to infrastructure-level metrics*
 - *WS-Agreement and WS-Agreement Negotiation*
 - *Term language needed*
- Monitoring - Metrics Collection
 - *Where: host OS/hypervisor, guest OS, application-level*
 - *When: upstream vs. downstream*
- SLA Evaluation
 - *Hysteresis*
 - *Statistical methods, e.g., Median Absolute Deviation, Interquartile Range, Iterative Local Regression*
- SLA Enforcement -- Violation Response
 - *Throttling*
 - *Load migration – process, VM, container migration all possible*
 - *Elasticity -- on-demand resources*
 - *SLA re-negotiation*

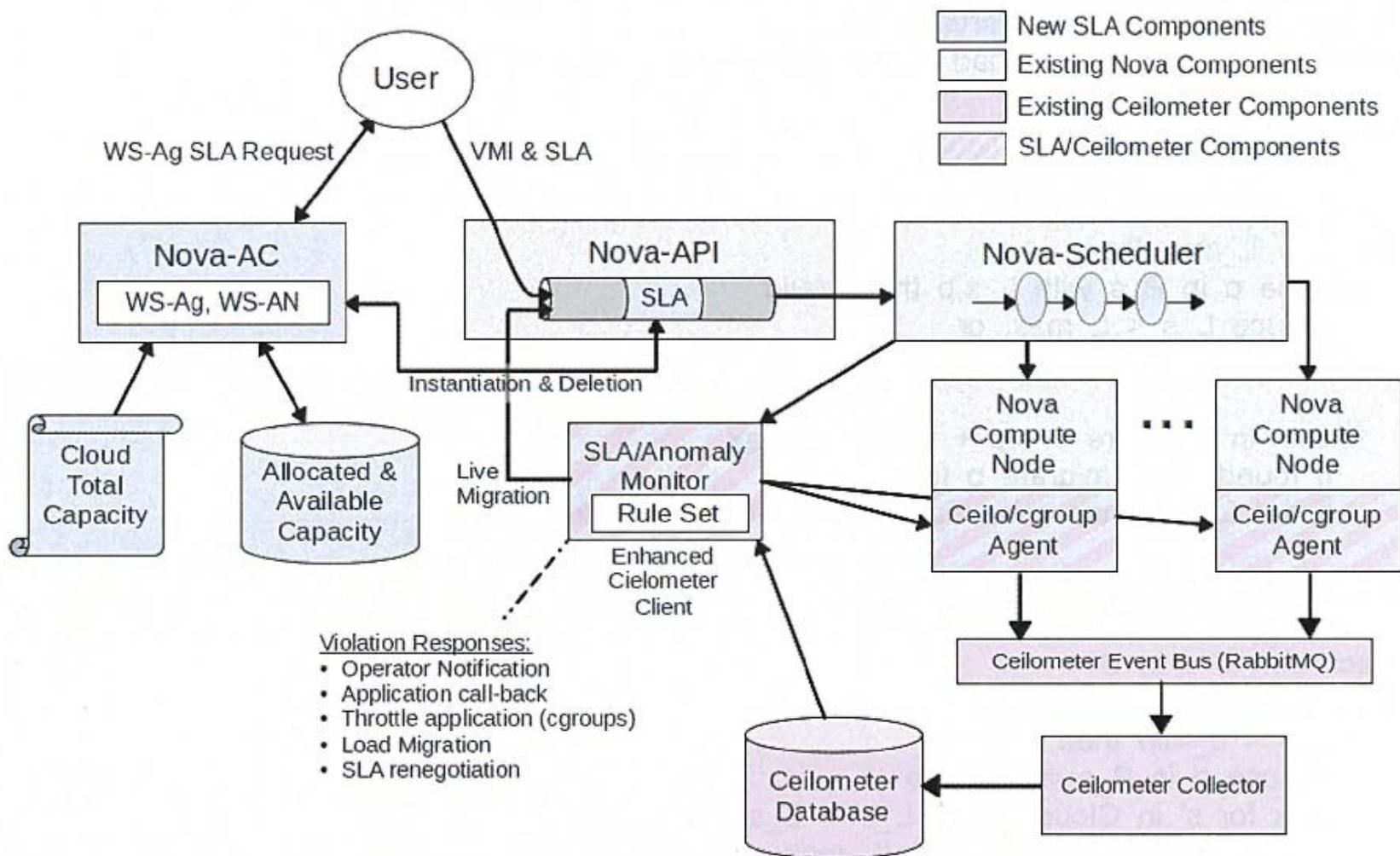


Server Load Types for SLA Management



Lee and Sill, A Design Space for Dynamic Service Level Agreements in OpenStack
Journal of Cloud Computing: Advances, Systems and Applications, to appear, 2014.

An SLA Architecture for OpenStack



Lee and Sill, A Design Space for Dynamic Service Level Agreements in OpenStack
Journal of Cloud Computing: Advances, Systems and Applications, to appear, 2014.

Summary, Findings, Conclusions & Comments

- Lots of Development & Testing needed
 - *What are the simplest SLA mechanisms that "scratch the itch" for the most users?*
 - *Contractual SLAs vs. machine-enforceable SLAs*
 - *Cloud performance SLAs vs. network SLAs (SDNs?)*
 - *Lighter weight alternatives to VM migration*
 - *Process migration; Container-based virtualization -- Docker*
- Capacity Planning & Management
 - *How to estimate query requirements, load demand, time-to-completion*
 - *How to support reasonable load requirements to produce reasonable behavior*
 - *How to manage sets of users such that no one user is disruptive*
- 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*
- Leverage/harmonize existing SLA work
 - *OGF WS-Agreement, WS-Agreement Negotiation*
 - *TeleManagement Forum (TMF)*
 - *Distributed Management Task Force (DMTF), ...*
- *<humor> And don't forget the WS-Disagreement protocol (WS-NO), GFD.199, ;-)*
 - *Most negotiations fail anyway – WS-Disagreement save vast amounts of time and money by immediately going to the "Disagree" state and staying there </humor>*

