The Ubiquitous Edge Platform

Lincoln Bryant
Rob Gardner

US ATLAS Technical Planning Meeting
August 1, 2016

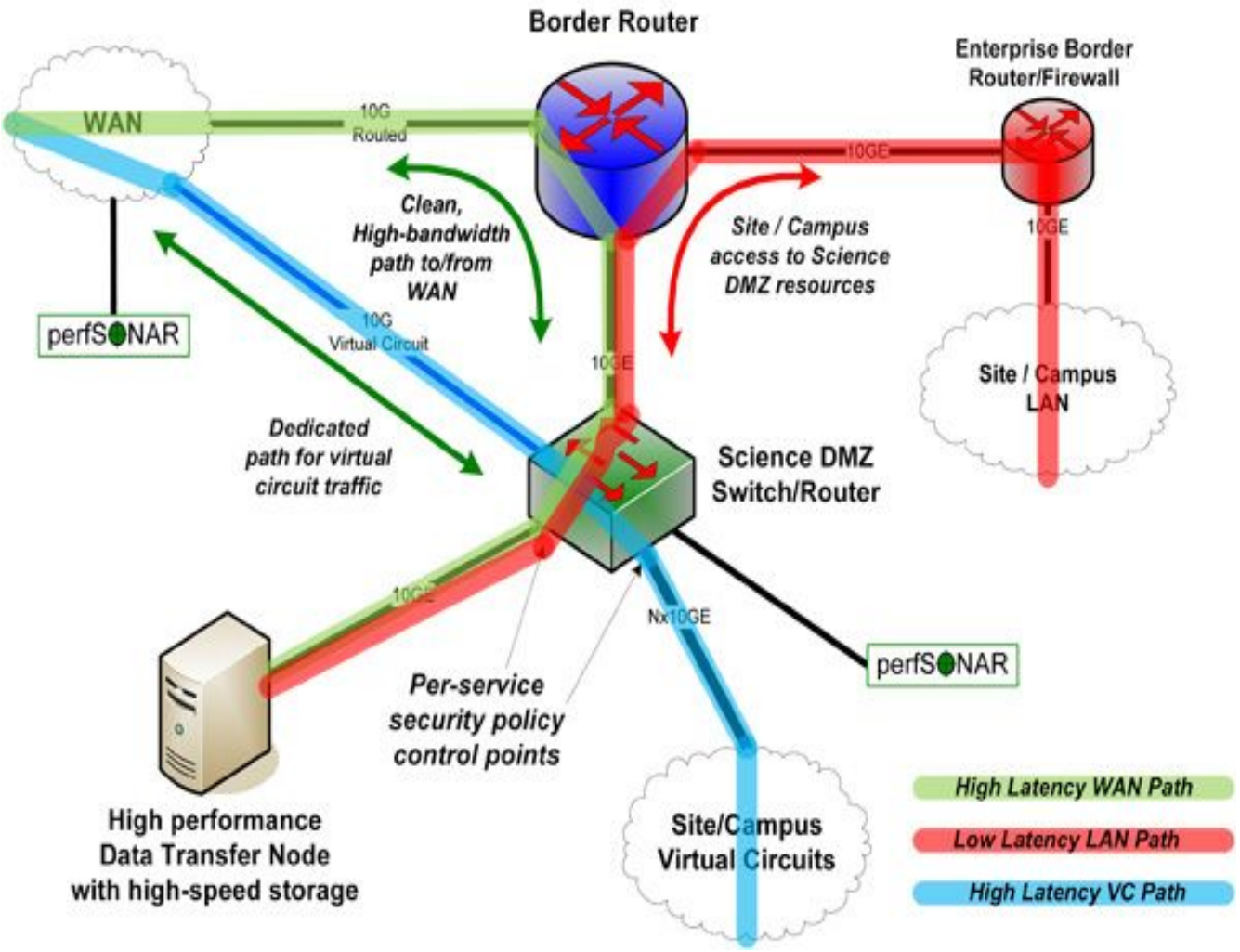
Ubiquitous & Easy “CI Substrate”

- On-premise edge device(s) for deploying scientific computing frameworks and software
  - Batch processing, data caching and transfer, etc
- Centralized web-based administration console and matchmaking service
  - Bringing a VO or science community to sites without significant IT burden
- Designed to scale
  - Services for connecting to local clusters, campus grids
  - Cloud-bursting to AWS or other providers

Distributed Virtualized Data Centers

- Coordination and operation efforts take a considerable amount of people time
  - Configuration, new versions, vulnerability patching, site-specific peculiarities
- Reduce the need for local IT to become experts in the stack
- Standardized, simplified deployment of CEs, SEs, caches, and components yet to be realized.

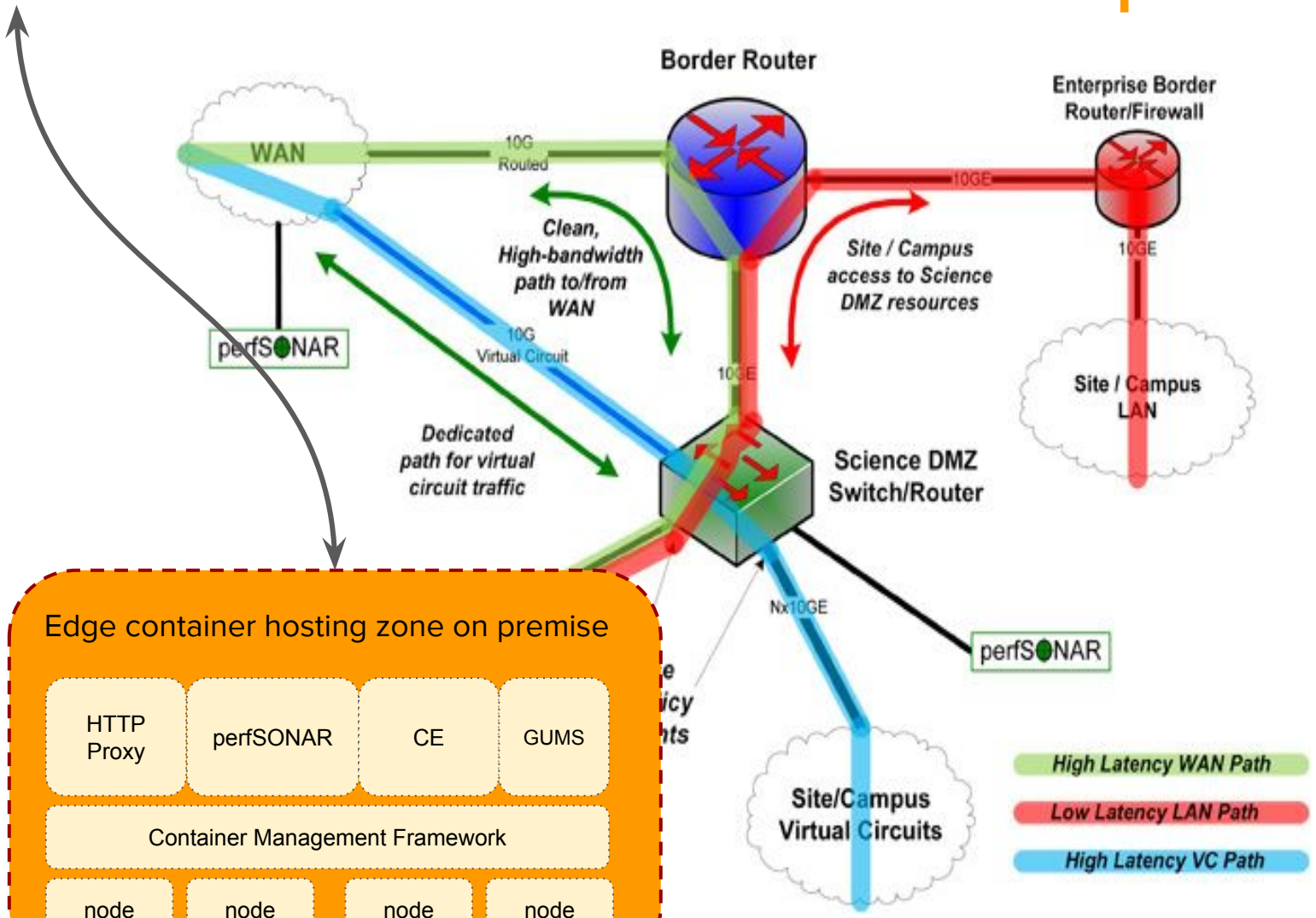
Canonical SciDMZ



CI central ops console:

$ slate install osg-squid.3.3 --sites SiteA SiteB SiteC

SuperDMZ



Edge container hosting zone on premise

HTTP Proxy	perfSONAR	CE	GUMS
Container Management Framework			
node	node	node	node

Standardized hardware

- Produce a reference specification for edge container platform
- Allows support team to focus on software curation rather than hardware troubleshooting and optimization
- Initial focus on low-cost, single node deployment
- Could be paired with a standard network device to ‘seed’ a larger infrastructure

Software underpinning

- Exploring a number of options for container runtime and base operating system
  - Canonical's LXD seems to have the right mix of API, security, and ease of use.
- Trying to find a happy medium between platform features and ease-of-use for local site administrators
- Web interface and RESTful API
  - Requires development work, out-of-the-box options are scarce

Application deployment

- Containerized applications created, vetted, maintained by central operations team.
  - Pushed by operators down to subscribed sites
- Built-in monitoring/analytics
  - Every service should get its own set of applicable collectors
  - Leverage our existing monitoring expertise

Automation efforts

- It should be possible for me to stand up and destroy an OSG/ATLAS site in a completely automated way.
- Many points where human interaction is currently needed.
- Can we separate approvals (requiring human interaction) from configuration?
  - OIM, AGIS, and equivalents
  - Certificate registration
  - etc

Benefits for US ATLAS

- Could potentially deploy CEs, SEs, caching proxies, etc all within “the box”.
  - Best known versions and configurations get automatically pushed to downstream
  - Updates should be atomic, so rollbacks are easy.
- Containerization effort putting more eyes on existing documentation and builds
  - Example: Patches submitted for GUMS to build on EL7
    - https://github.com/opensciencegrid/gums/pull/27

Summary

- Platform for “edge” services on Science DMZs with well-defined reference hardware.
- Container-based applications, maintained by VO operators
- Built-in service discovery, configuration, and monitoring
- Flexible, adaptable to the needs of other projects.

Thank you!
Questions?

Extra slides / Open questions

Security concerns

- Who has root on the machine?
- Can trusted users allocate resources and start containers remotely?
- Is Docker secure enough to be used? Many claims of a busted security model.
  - User namespaces and unprivileged containers seem to be semi-working in new kernels? (Affects OS choice!)
- Ultimately: What is the correct privilege separation between owner and operator?