Architecture Design Document
VirtualDesignMaster - Season 2

Challenge 1

Prepared by: Daemon Behr

Date: 2014–07-14

Revision History

Date	Rev	Author	Comments	Reviewers
2014-07-14	R8	Daemon Behr		

Design Subject Matter Experts

The following people provided key input into this design.

Name	Email Address	Role/Comments
Daemon Behr	daemonbehr@gmail.com	Infrastructure Architect

Contents

1. Purpose and Overview

1.1 Executive Summary

In last season we were in the aftermath of a pandemic zombie virus outbreak, which had destroyed most of the worlds civilization. The virus spread was in remission, thus allowing us to start rebuilding core infrastructure as hardware became available. Internet connectivity was restored and multi-site topologies were established.

Since the completion of last season, hardware has been obtained by restoring key manufacturing facilities, commandeering all known component warehouses. Resources from depots have been stockpiled and transport and distribution systems restored. Scalability of core infrastructure components is fully possible with no perceivable constraints on availability or lead-time. Highly available and redundant networks have been deployed between key areas that still have civilization. This provides an extremely high-speed private network as opposed to an Internet. Wired last-mile connectivity is pervasive and reliable in all core facilities.

Some groups have come together to restore a semblance of a global network, but without many resources to maintain it. Reliability and coverage is spotty, but good enough for basic communications. Internet last-mile connectivity is achieved by wireless mesh networks with long haul trunks using repurposed terrestrial microwave with transverted HSMM COTTS access points.

The viral outbreak has returned in a mutated strain that does not respond to previous vaccine. The pandemic has over-run all major cities and medical research facilities, leaving the remaining population defenseless.

The billionaire that provided us with resources in season one has planned for this. His space tourism company has built a large base on the moon that is designed to support what is left of the human race, until the first colony on Mars is ready.

We must work quickly to evacuate what is left of the human race. Currently, there is only one space ship depot on earth, located in Cape Canaveral, Florida in the United States. Three more are being built as fast as possible, with additional to follow.

Each depot is capable of producing a launch ready ship in 72 hours. Larger ships are currently being designed. ANY malfunction of the manufacturing systems is catastrophic, and will cost precious human lives.

A completely orchestrated infrastructure that is highly reliable and easily deployable is required. We must ensure these sites are up and running as soon as possible.

We are expecting a design that includes the complete solution stack, from the infrastructure layer to the application layer.

The application that controls the depot has the following requirements:

· Client facing web layer
· Message queuing middle tier
· Database backend

Explanation is required on why to use the orchestration framework of choice.

1.2 Summary Analysis

The purpose of this design is to support the production of the only launch ready space ship in the world within a 72hr period. There are many different systems and procedures required for a 98% success rate under ideal conditions and failure to meet timelines will result in massive loss of human life.

1.3 Design Interpretation

The number of systems used for modeling will be increased to meet the demands of the strict timelines. These systems do not require the same level of reliability as the mission critical manufacturing systems because of the clustering nature of the compute grid. They are not real-time systems but increased compute power in the grid will decrease over-all modeling time. The design of these systems is not in the scope of this document, but is required for a successful launch.

This design document is specific to the support of the infrastructure manufacturing systems for space ship operational readiness in Cape Canaveral. The manufacturing system is a multi-tiered application environment that requires the highest level of reliability possible.

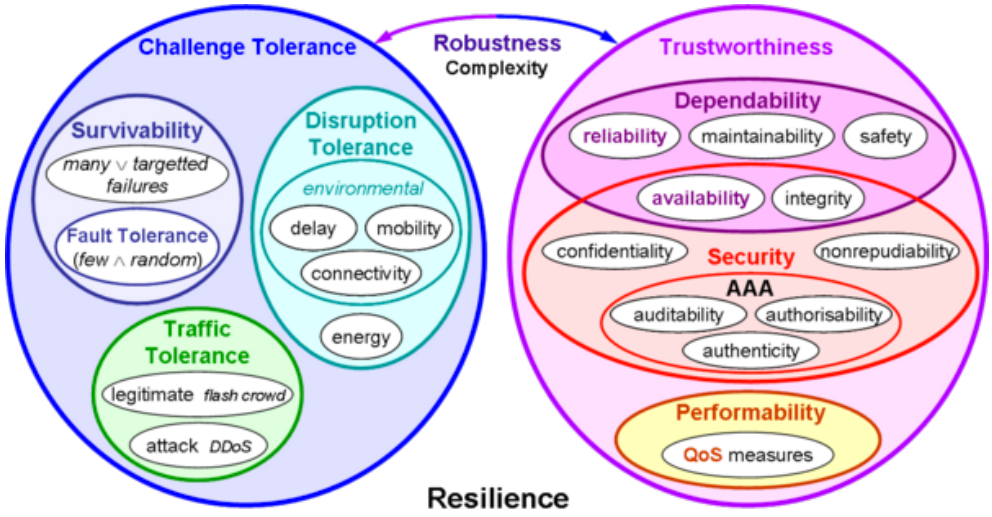
1.4 Intended Audience

This document is meant for the key stakeholders in the project as well as technical staff leads required for a successful deployment.

1.5 Requirements

Below are the requirements as defined by the scenario document as well as additional communication with judges and creators in clarification emails.

The requirements defined will be mapped to the trustworthiness discipline of a resilient system. Since the main driver of the project is dependability during the 72hr period, we will focus on that for the requirement definitions.



1.5.1 Availability

Availability can be defined as “the proportion of the operating time in which an entity meets its in-service functional and performance requirements in its intended environment”. The criticality of the environment requires 100% availability as a service level objective (SLO).

Availability can be understood by understanding the relationship of Maintainability and Reliability. The chance a system will fail is based on Reliability. How quickly it can be fixed is due to it’s Maintainability. The combination of those two provide us with:

MTBF – Mean Time Between Failures (Reliability)

MTTR – Mean Time To Repair (Maintainability)

Availability is equal to MTTR/MTBF over the period evaluated.

R001	**Production systems require a 100% availability SLO**

1.5.2 Maintainability

Maintainability is defined as “the ability of an entity to facilitate its diagnosis and repair”. This is a key factor in availability.

R002	**The infrastructure must be quickly diagnosed and easily repaired**

1.5.3 Integrity

System integrity is defined as “when an information system performs it’s function in an unimpaired manner, free from deliberate or inadvertent manipulation of the system”. In this context, it means that adds / moves / changes are not done on production systems.

R003	**An environment will be required for all development operations**
R004	**A QA environment will be required for pre-production testing**

1.5.4 Reliability

Reliability can be defined by having an absence of errors. Errors in a code base do occur, but there must be a method to ensure that they are tested, identified and resolved before they are put in production. This prevents the errors from affecting the overall application infrastructure.

In addition, infrastructure component configuration errors can cause reliability issues and faults. A method must be in place to ensure that all component configurations are correct.

R005	**A system must be in place to identify errors in the application code base**
R006	**Infrastructure component configurations must be audited for errors before being put in production**

1.5.5 Safety

Safety is defined as “the probability that a system does not fail in a manner that causes catastrophic damage during a specified period of time”.

The current state of the global viral pandemic makes this a very real concern. If a certain facility that houses all key information system becomes over-run by a zombie hoard, then the mission has failed and live on Earth as we know it will cease to exist.

R007	**The system must fortified to protect to IS infrastructure**

1.5.6 Scalability

Although the scope of the scalability has not been defined, the ability for it to occur with minimal additional design required.

R008	**The system must be scalable**

1.5.7 Automation

In order accommodate the needs of all the other requirements, automation and orchestration is required.

R009	**The system must make use of automation wherever possible without conflicting or impairing any other requirements**

1.6 Constraints

C001	**The infrastructure cannot tolerate any malfunction in the production system**
	This will cost time delays, which in turn will cost human lives and possibly cause the entire mission to fail.
C002	**Required technical staff must be available during the 72hr launch readiness period.**
	Without the required staff, the mission will fail.
C003	**Communications are limited to between core facilities.**
	Internet connectivity is limited and unreliable.
C004	**Automation must be used.**
	In order accommodate the needs of all the other requirements, automation and orchestration is required. This is a non-functional requirement.

1.7 Risks

R001	**The virus spread may occur more quickly than anticipated.**
	Time to prep would be reduced and facility security would need to be increased.
R002	**System malfunctions may delay timelines**
	Delays will cost human lives.
R003	**Adequate power may not be available**
	This would impact the ability to scale
R004	**Staffing levels may not be adequate**
	This would put more strain on available staff and increase possibility of human errors when tired.
R005	**Weather changes may delay launch**
	This may delay launch and cost human lives.

1.8 Assumptions

A001	**The required staff will be available and have been trained to support the environment and work as a team.**
	The minimum required is indicated in D006
A002	**All required hardware for scalability is available**
	This should be in a facility nearby that is accessible by the technical staff and secure from zombies. All tools required for deployment are also available.
A003	**Adequate power is available**
	This includes clean power, multiple UPS battery banks and redundant generators equivalent to a tier3 data center.
A004	**All staff in Primary roles have adequate supporting supplies**
	This consists of a large stockpile of high-quality coffee. Lack of this may cost human lives.
A005	**All staff in Secondary roles have adequate protective equipment**
	This should consist of standard equipment: 12 gauge shotgun with 24 shell bandolier, flamethrower (2 per 8 person shift), concussion grenades (8), high power rifle (7.62mm round) with 5 x 30 round magazines. An adequate number of chainsaws and machetes should also be made available.
A006	**All equipment in this design is new and has been burn-in tested**
	A period of 1 week was used and all tests were passed without issue.
A007	**All component firmware is at the identical revision**
	This is for each component type.
A008	**Software development will not be done by infrastructure technical team**
	Technical team scope is very specific and thus supports the infrastructure for the application, not the application itself. The Dev technical team will provision infrastructure components for the software developers, but never work with the configuration of said components or touch any application code.

2. Architecture Design

2.1 Design Decisions

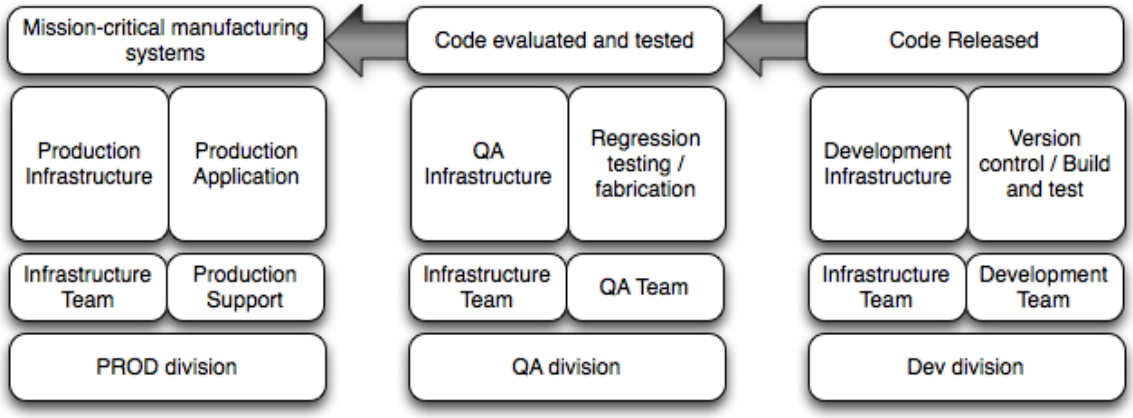
The architecture is described by a logical design, which is independent of hardware-specific details.

The following design decisions are presented with their justification.

D001	**The system will have the fewest number of logical components possible**
	Complexity needs to be reduced to allow for easier maintainability and quicker diagnosis of issues.
D002	**Automation will be done by scripting as opposed to using an automation engine**
	This relates back to D001. The fewest number of components possible.
D003	**There will be Production, Development and QA environment**
	This is to meet the requirements of R003, R004, and R006
D004	**The facility will be fortified**
	There were two possible methods to meet R007. Fortify the facility, or distribute the infrastructure between several sites. Fortification was the decided on method because it reduces complexity and requires less staff and time.
D005	**Continuous Integration and Continuous Delivery will be implemented**
	Iteration of application code builds and infrastructure designs will occur hourly. This is not to say that changes will occur every hour, but it does mean that adds / moves / changes will be reviewed at that time. This allows for fail-fast of code builds and infrastructure changes.
D006	**Technical teams will be made up of 9 groups of 8 people. 24 people in Prod, 24 in QA and 24 in Dev.**
	This is based on having a dedicated technical team for Production, QA and Dev. Each group will work in 3 shift groups, A, B and C. Shift A will be on primary operations for 12 hrs, then moved to secondary operations for 4 hrs, then sleep for 8 and resume. Shift B will overlap with Shift A at the 8hr mark. Shift C will overlap with Shift B. Below is the shift schedule for each one of the technical teams. **Hour | Role and associated shifts** 0-8 - Primary A, Secondary C, Rest B 8-12 - Primary A,B Secondary C, Rest (none) 12-16 - Primary B, Secondary A,C, Rest (none) 16-20 - Primary B, Secondary A, Rest C 20-24 - Primary B,C, Secondary A, B, Rest (none) 24-28 - Primary C, Secondary B, Rest A 28-32 - Primary C,A, Secondary B, Rest (none) 32-36 - Primary A, Secondary C, Rest B 36-40 - Primary A, Secondary C, Rest B

	40-44 - Primary A,B Secondary C, Rest (none) 44-48 - Primary B, Secondary A,C, Rest (none) 48-52 - Primary B, Secondary A, Rest C 52-56 - Primary B,C, Secondary A, B, Rest (none) 56-60 - Primary C, Secondary B, Rest A 60-64 - Primary C,A, Secondary B, Rest (none) 64-68 - Primary A, Secondary C, Rest B 68-72 - Primary A,B Secondary C, Rest (none) Primary Operations is the critical support role for each infrastructure. Secondary Operations is securing the facility from the advance of the zombie hoard. Rest is done as scheduled, but can also be done by the overlapped shift in Secondary Operations, zombies permitting.
D002	**Automation will be done by scripting as opposed to using an automation engine**
	This relates back to D001. The fewest number of components possible.
D003	**There will be Production, Development and QA environment**
	This is to meet the requirements of R003, R004, and R006
D004	**The facility will be fortified**
	There were two possible methods to meet R007. Fortify the facility, or distribute the infrastructure between several sites. Fortification was the decided on method because it reduces complexity and requires less staff and time.
D005	**Continuous Integration and Continuous Delivery will be implemented**
	Iteration of application code builds and infrastructure designs will occur hourly. This is not to say that changes will occur every hour, but it does mean that adds / moves / changes will occur at that time.

2.2 Conceptual Design



The divisions are divided into 3 areas Prod, QA and Dev. Workflow will go from developers to code base release, which is then validated and tested by the QA team. After regression testing has been performed, then the code will be validated and tested on a limited environment. It will be moved to the production systems.

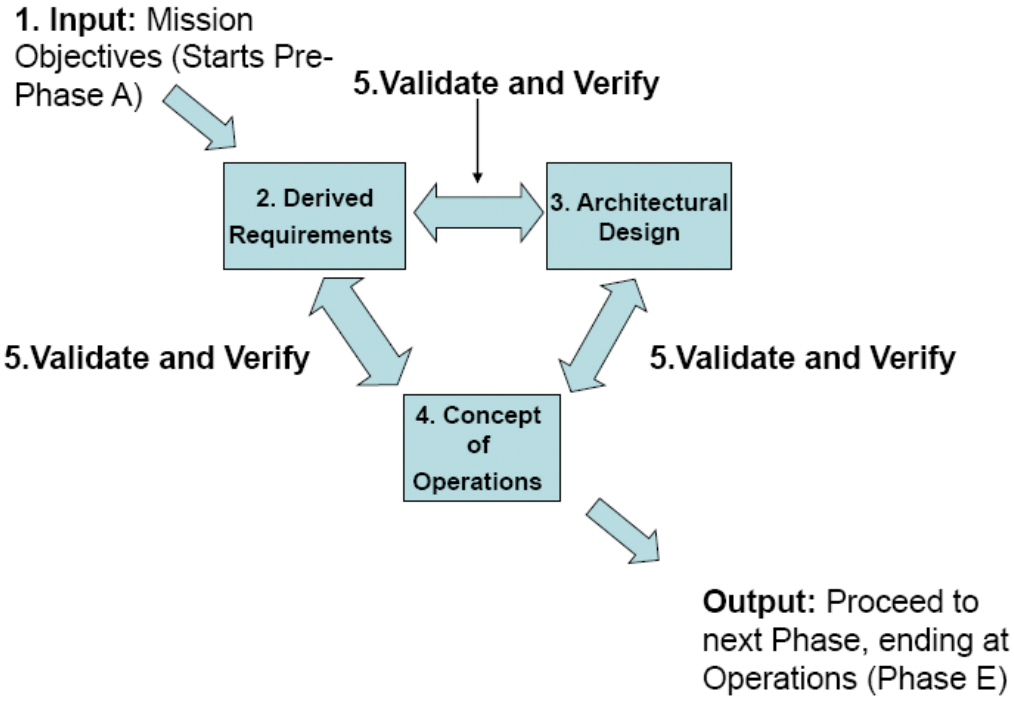
The roles of the infrastructure teams are different as they support different systems. The NASA phases of system engineering are shown below. The divisions will map to phases B (Dev), C (QA), and D (Prod).

Phase	**Phase Title**	**Purpose**	**End of Phase Review**
Pre-Phase A	Concept Studies	Produce a broad spectrum of ideas and concepts, establish mission objectives	Mission Concept Review (MCR)
Phase A	Concept and Technology Development	From multiple approaches develop a single system concept for the mission, with system requirements and architecture. Perform trade studies and identify needed technologies.	System Definition Review (SDR)
Phase B	Preliminary Design and Technology Completion	Establish a preliminary design, with subsystem requirements, interfaces, and with technology issues resolved.	Preliminary Design Review (PDR)
Phase C	Final Design and Fabrication	Complete the design and drawings, purchase or manufacture parts and components, code software.	Critical Design Review (CDR)
Phase D	System Assembly, Integration, Test and Launch	Assemble subsystems, integrate subsystems to create systems, test to verify and validate performance, deploy the system.	Readiness Review (RR)
Phase E/F	Operation and Sustainment/Closout	Operate System, decommissioning, disposal	Decommissioning Review

NASA Phases of the Systems Engineering Life-Cycle

The infrastructure team for the Dev division will deploy the components for developers. This will consist of such things as the hardware platform, OS environment, application stacks and end user applications. The applications used will be for collaboration, code revision, automated testing, and modeling. This division will not require the same level of scalability as the QA or Production environment.

The infrastructure team for the QA division will deploy the components for Quality Assurance and Fabrication. In the diagram below, the “Validate and Verify” period between “Architectural Design” and “Concept of Operations” is where the QA team resides. Modeling systems and code analysis tools are used to audit code and design. Control of fabrication equipment is done through SCADA / HMI systems.



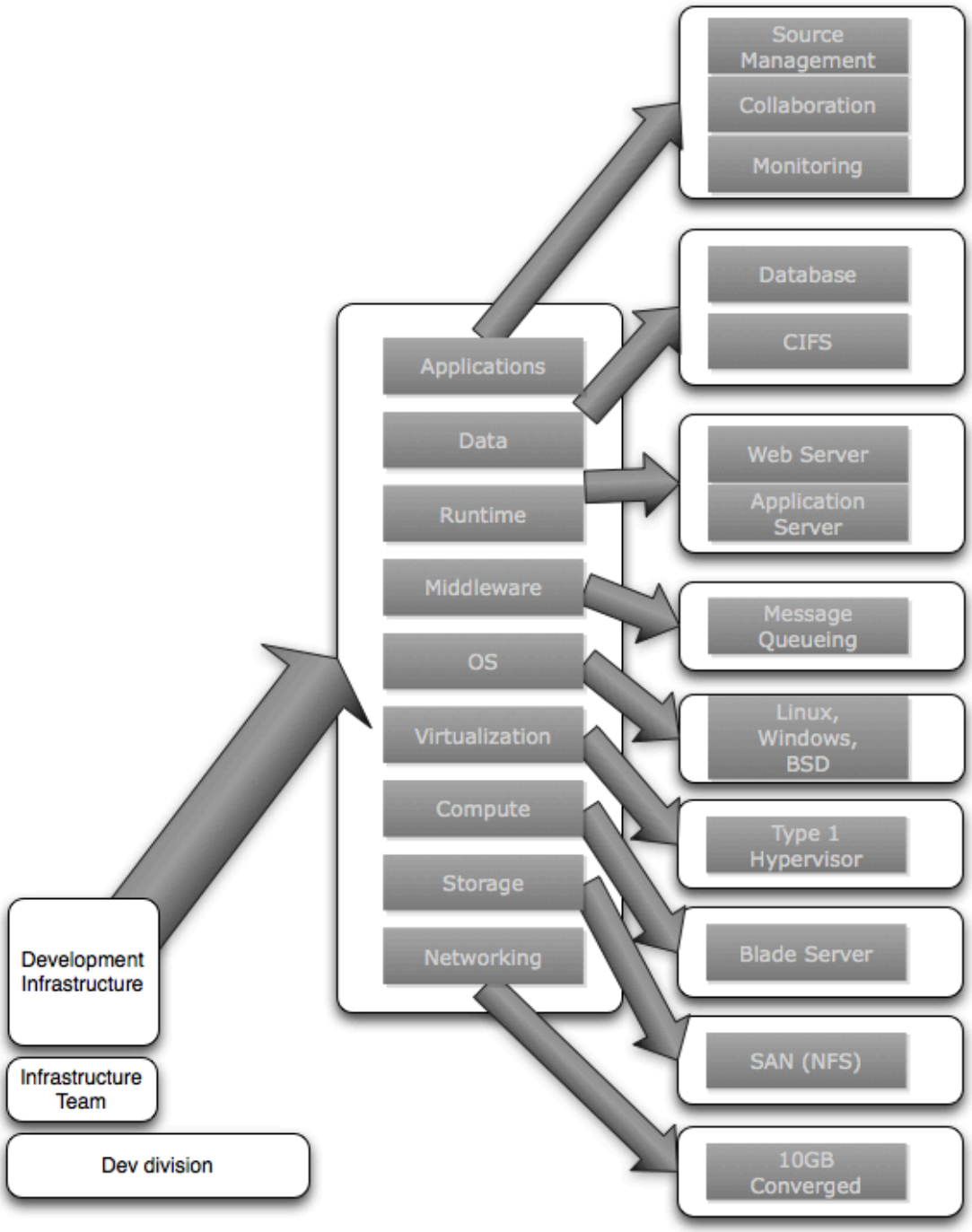
Prod is the most critical division, as it has to ensure that it is the most reliable system. It is the System Assembly, Integration, Test and Launch (SAITL) environment. The infrastructure team is responsible for deploying SCADA / HMI systems, real-time monitoring and response systems, FAIL-FAST systems that communicate directly with the Dev and QA divisions so that failures can be fixed in the hourly iteration of code / design.

2.3 Logical Design

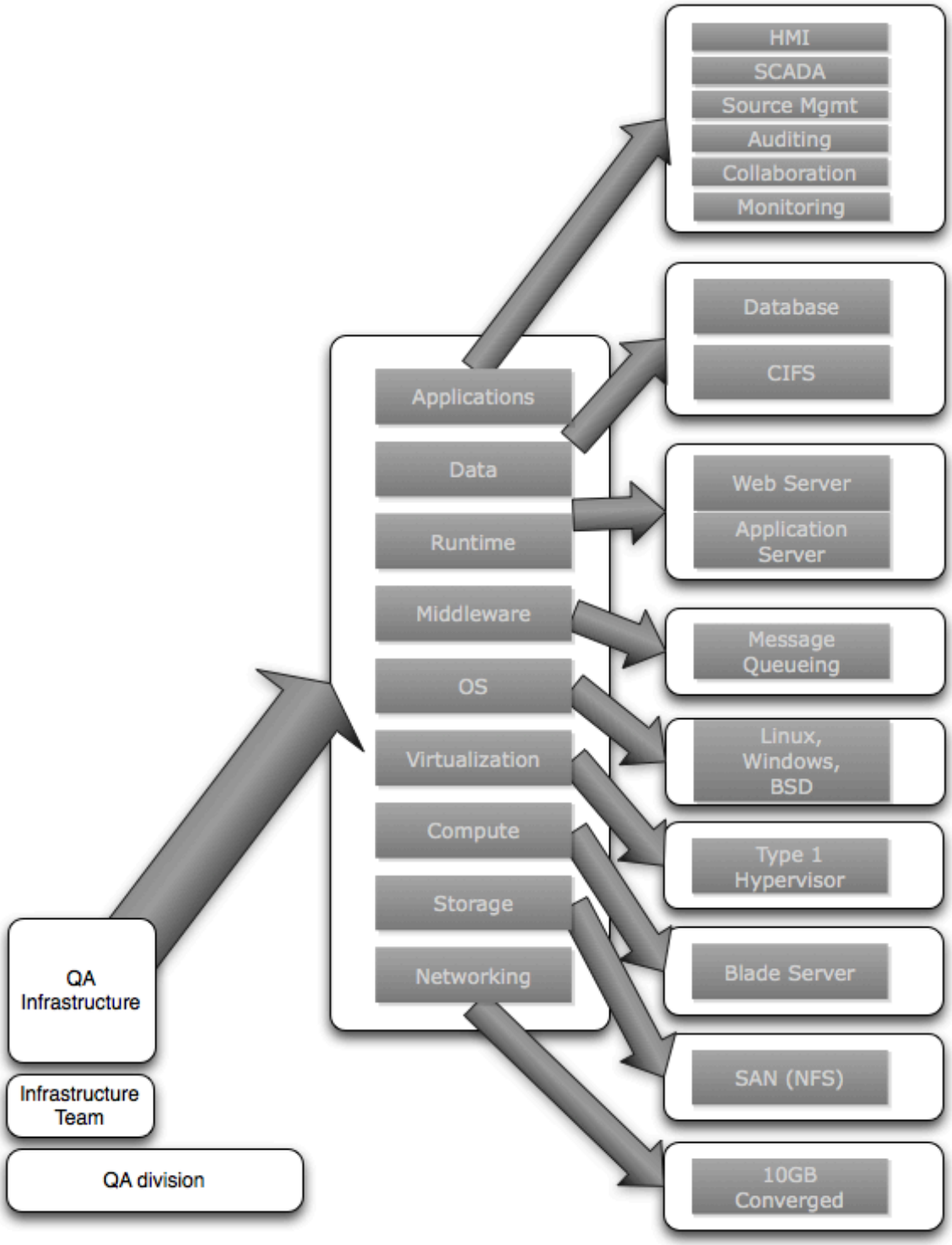
The Logical Design provides a more detailed view of the Conceptual Design components to meet the requirements. The architecture building blocks are defined without the mapping of specific technologies.

Below is the logical design for each division. It is expanded in a hierarchical manner indicating the component areas, but not the quantity, as it will change over time. Adds, moves and changes will affect the physical design, but not the logical unless a major infrastructure change occurs.

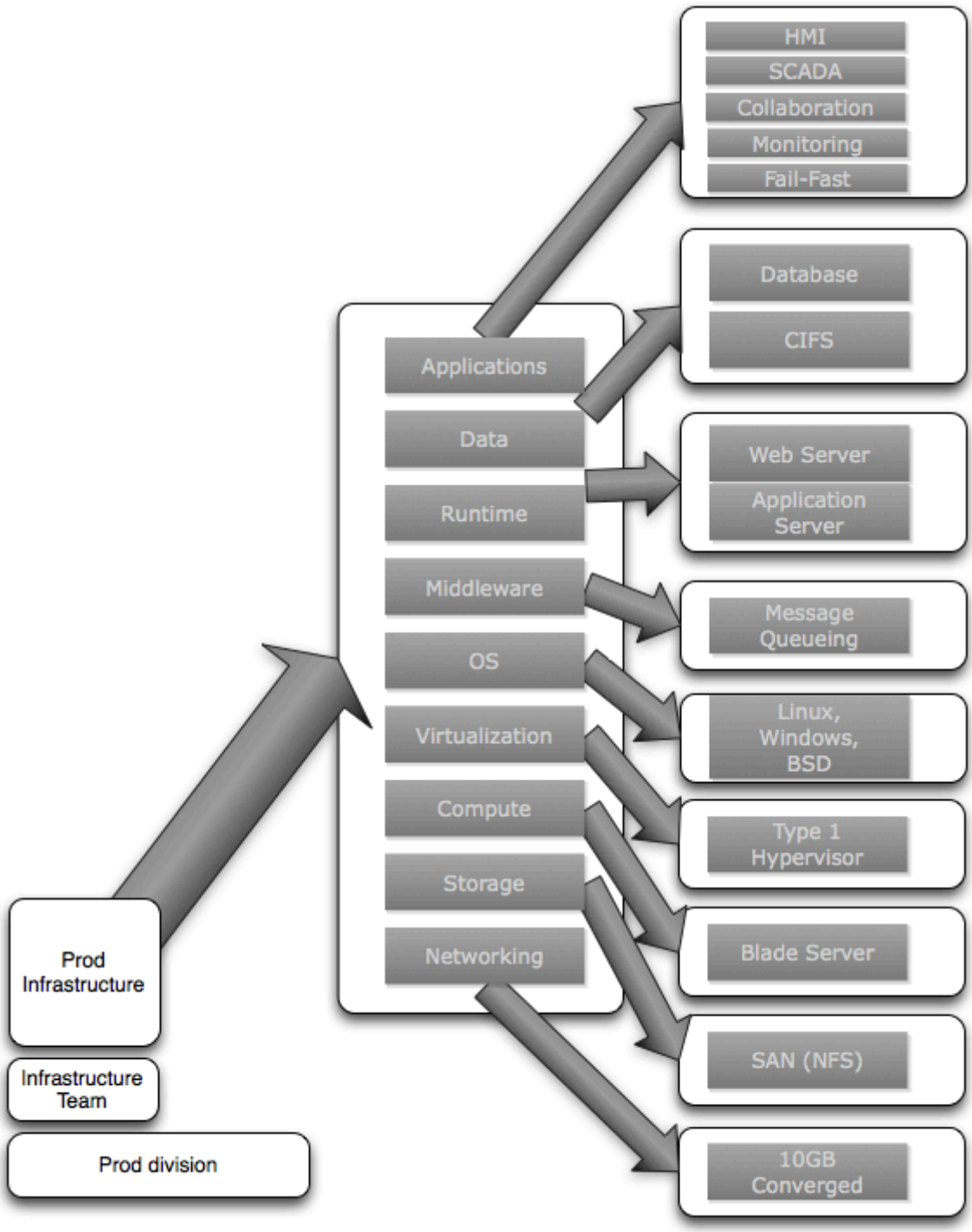
2.3.1 Dev Division Logical Design



2.3.2 QA Division Logical Design



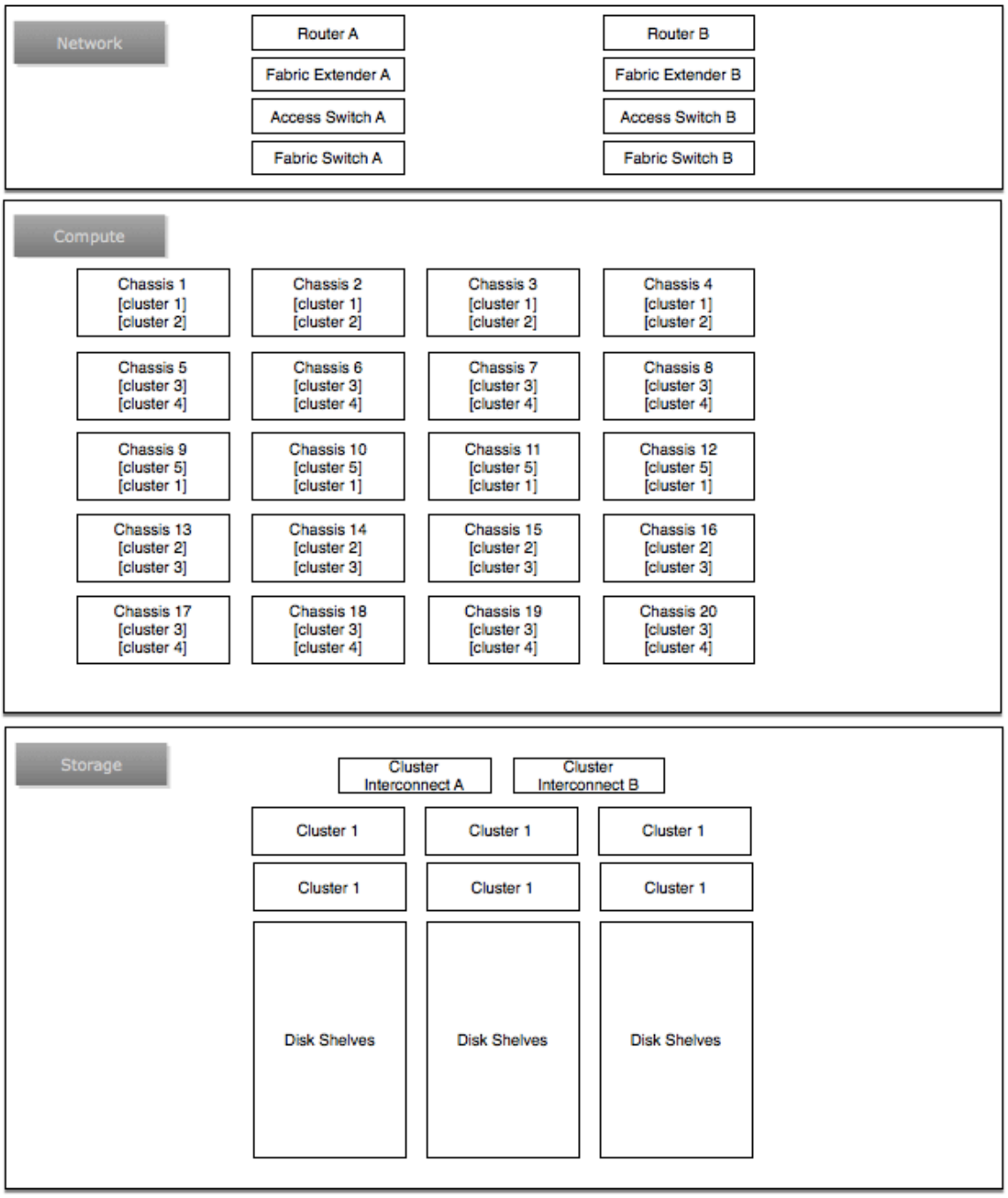
2.3.3 Prod Division Logical Design



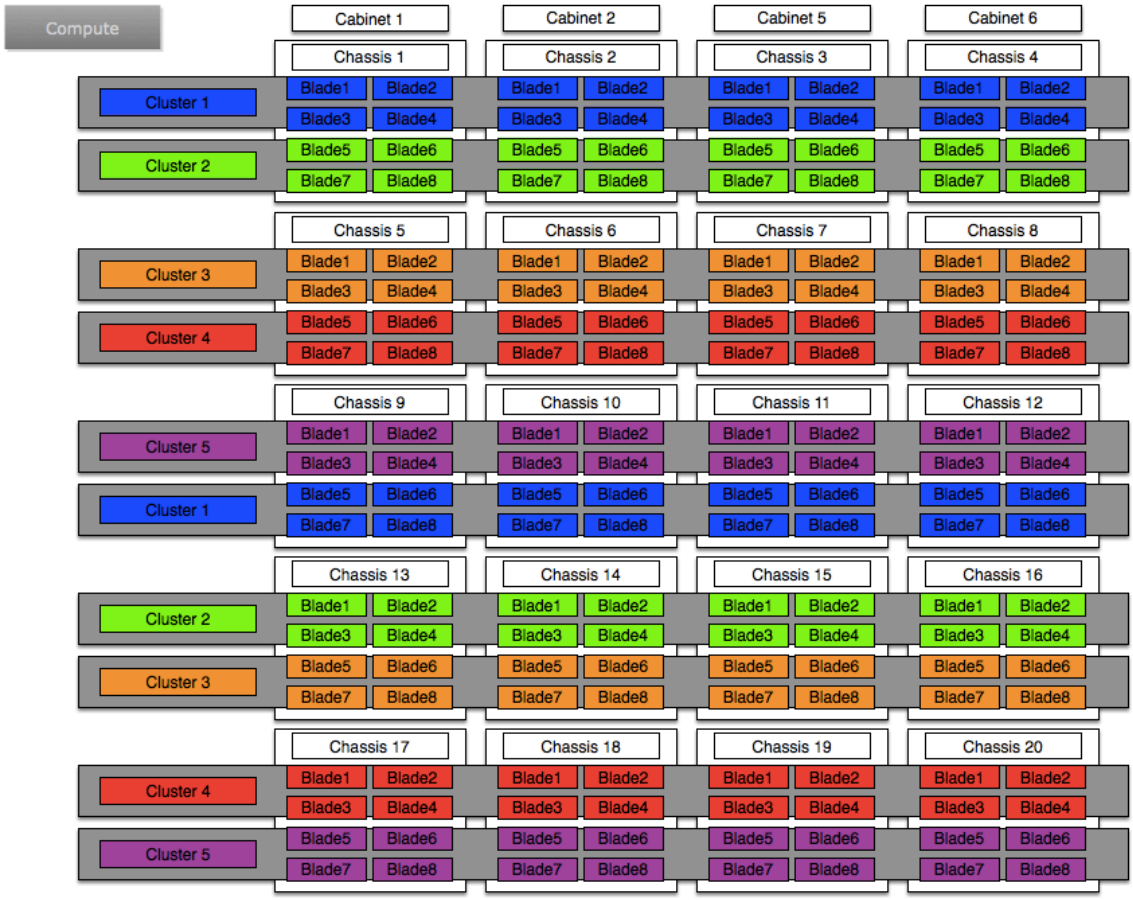
2.3.4 Divisional Pod Logical Design

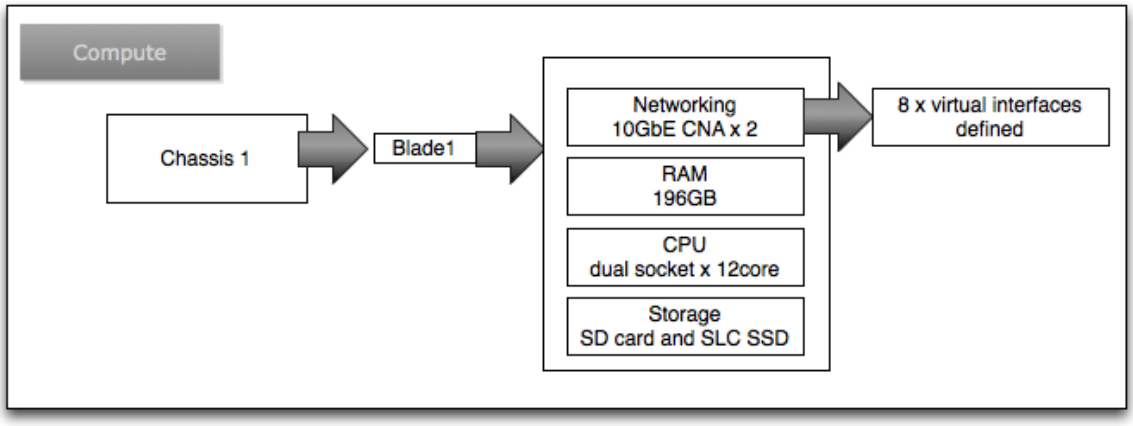
The 3 divisions will have identical infrastructure components, except for the scale and the applications that run on them and the systems that they interface with. This allows for simplified management and scalability.

Below is the logical design for the compute, storage and networking of a division pod.



Each division will be provisioned with a full pod of blade chassis and storage. Each cabinet will contain 5 chassis. Each chassis will have 8 blade servers. In vSphere there will be 5 x 32 host clusters. The clusters will be spanned across chassis and cabinets. As seen in the diagram below, this allows for no more than 12.5% cluster resource failure from a chassis malfunction, or 25% on a full cabinet failure.

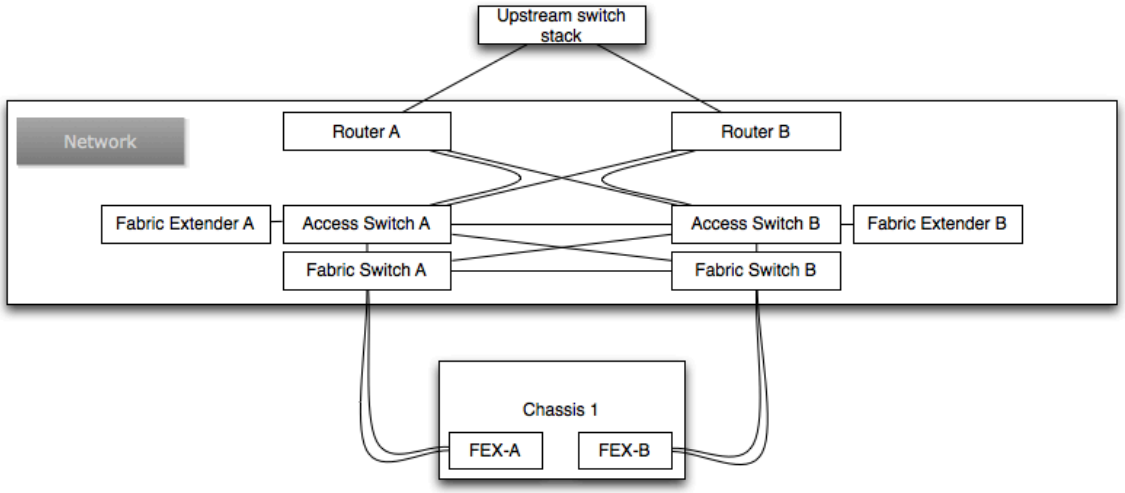




The chassis will uplink via fabric extenders to the fabric switches. The fabric switches will work in HA and can support a maximum of 20 chassis per pod (5 clusters). Each chassis

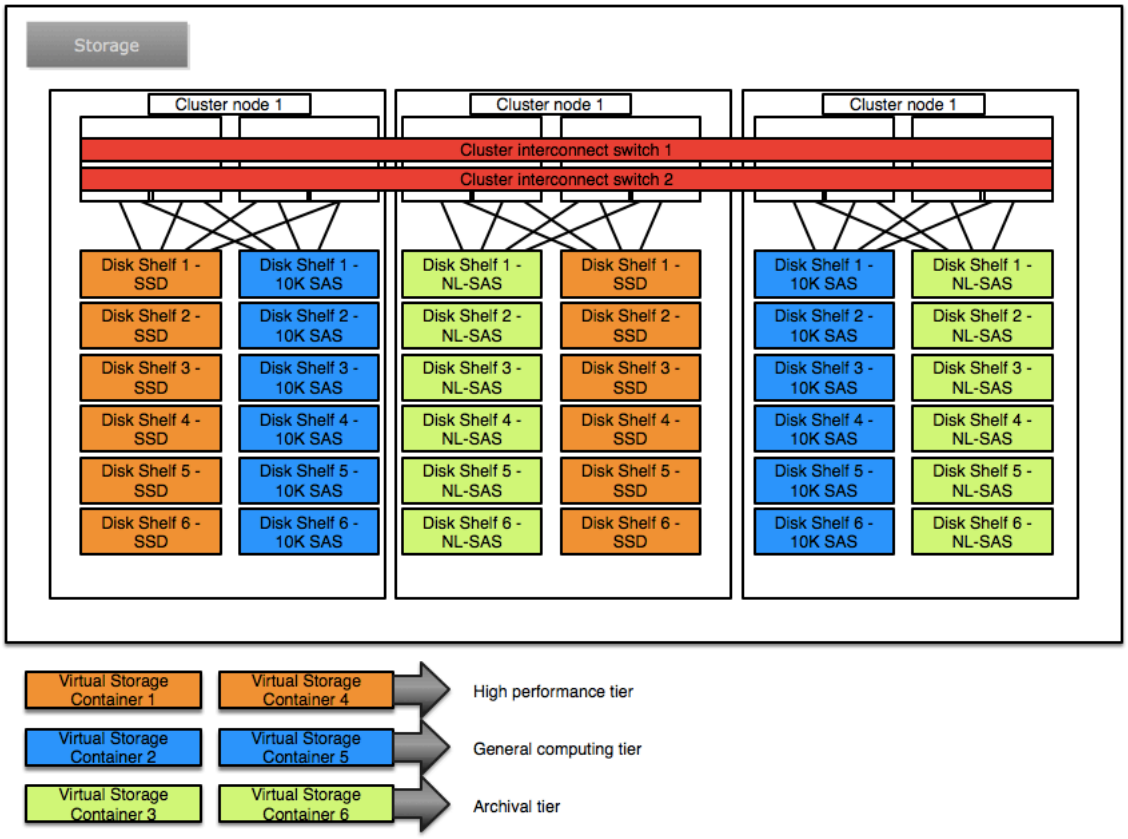
will 2 fabric extenders, each having 2 x 10GbE uplinks to it's associated switch. The blades will use an onboard 10GbE adapter as well as a 10GbE Mezzanine card. The aggregate bandwidth of each chassis is 20Gbps per fabric. The aggregate bandwidth of the blade servers within the chassis is 80GbE per fabric. Therefore the over-subscription ratio is 4/1.

The network is converged, carrying both storage and data traffic within the same physical devices. Connectivity from the chassis FEXs will be aggregated in a port channel with the fabric switches. The fabric switches will aggregate bandwidth via a virtual port channel across both access switches. The access switches have a LAG group between then to accommodate interswitch traffic. The fabric extenders connected directly to the access switches provide 1Gbs connectivity for any users that need direct connectivity. Otherwise user access will be routed through the upstream switch stack.



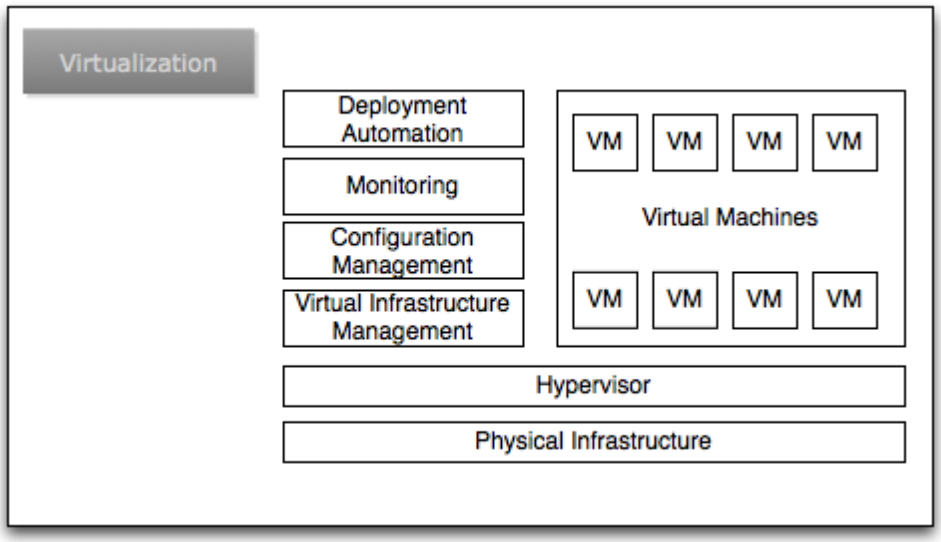
Storage will be comprised of clustered nodes in a scale out model. Each node will have active / active controllers. Nodes will be responsible for their own disks, but they can take over the disks of another node in the event of a failure.

Virtual storage containers are spanned across the aggregates of "like" performance tiers in each node cluster. This allows for expandability and performance. Volumes will be defined within the containers.



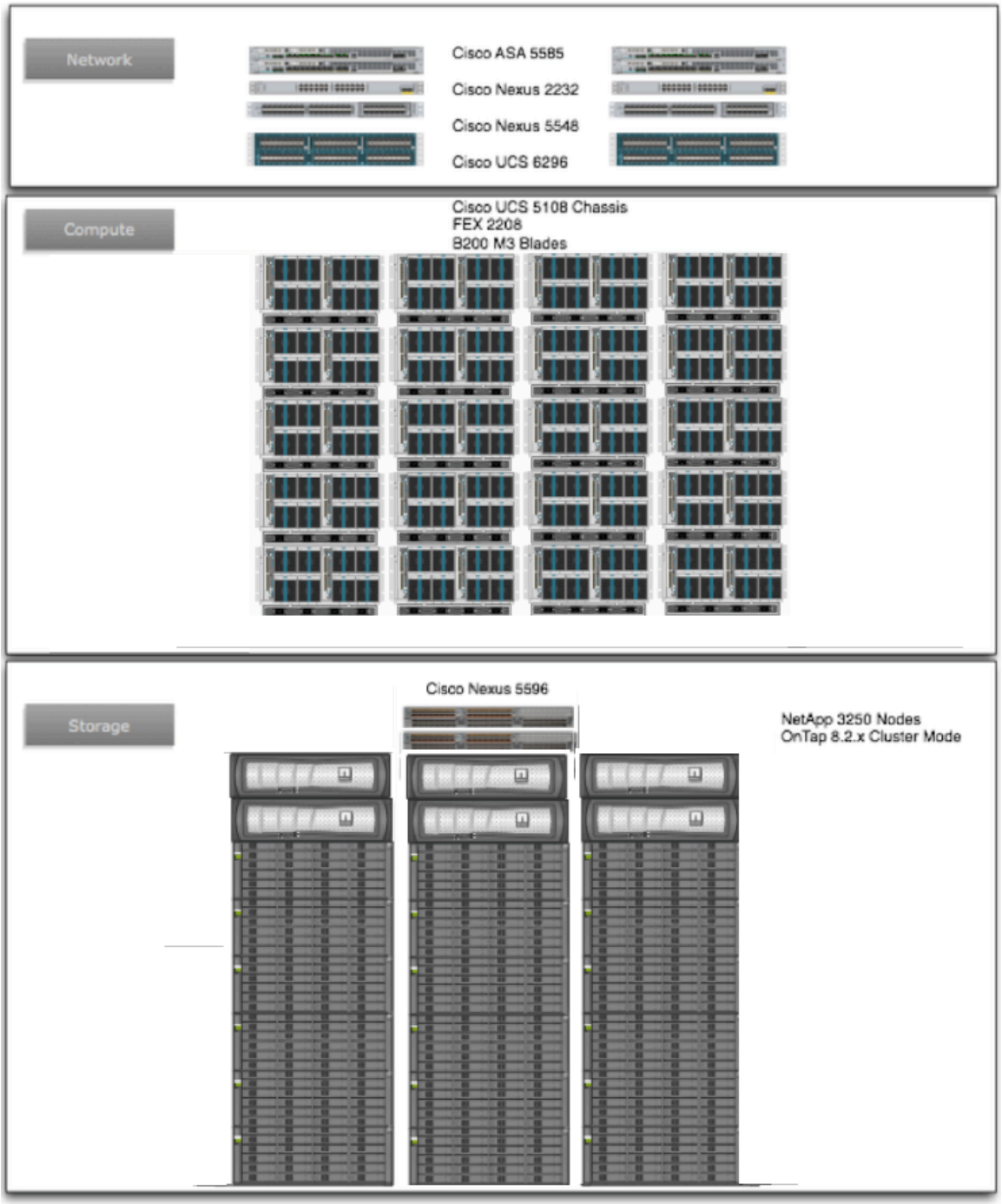
An additional storage tier exists outside of the clustered SAN environment. That is server side caching by making use of internal SLC SSDs within the blades.

The virtualization layer consists of the following components; a hypervisor, an infrastructure management system, configuration management, deployment automation and monitoring.



2.4 Physical Design

The physical design will be examined from first a datacenter row, then cabinet, then the technology area. The logical design for the divisional pod will be mapped to a row.



2.4.1 Physical rack layout of divisional pod

Cabinet row D

Above is the rack diagram for row D, for the Dev division. The other rows are P and Q, respectively.

The Cisco UCS infrastructure was chosen for the compute and network environment because of the ease of management, deployment and scalability.

NetApp was chosen for the shared storage because of the resiliency and scalability when using Clustered OnTap.

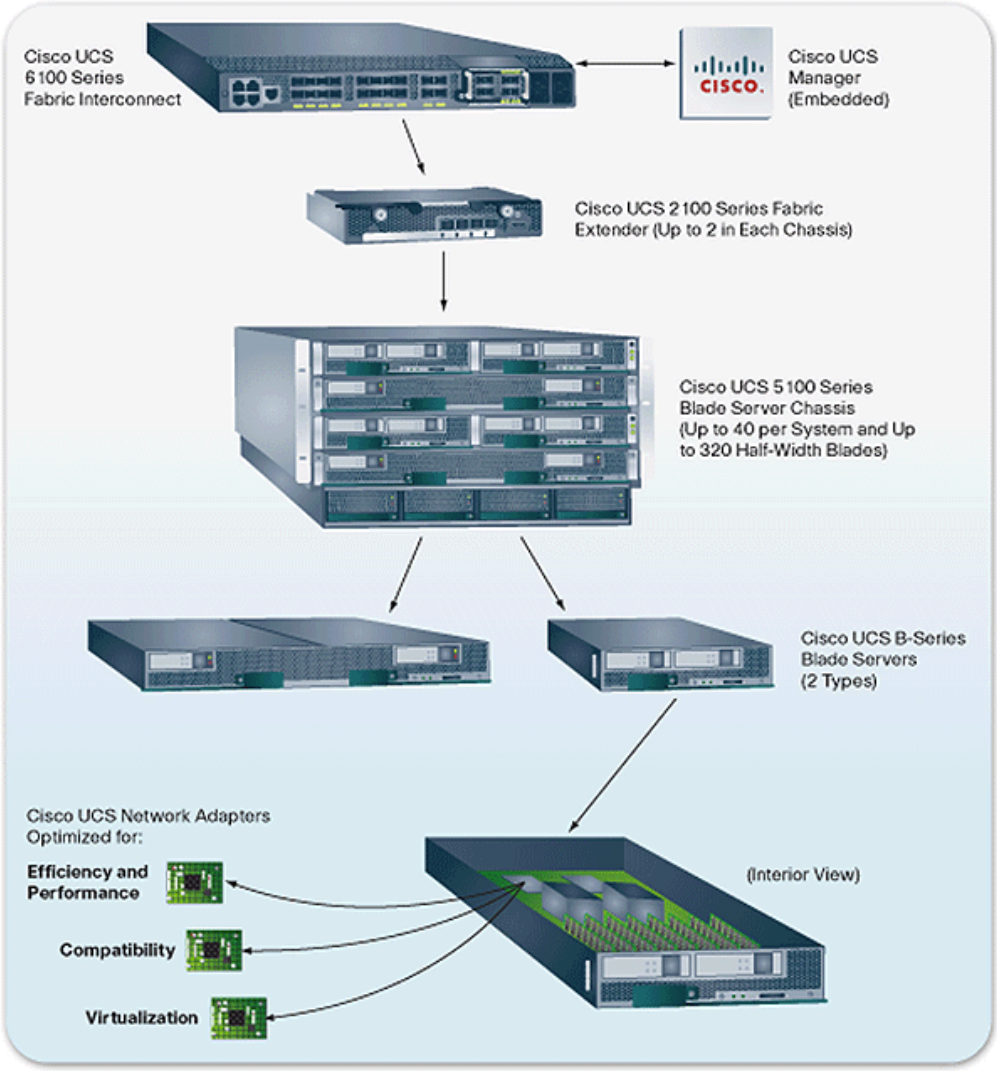
The server side caching will be done by PernixData FVP, making use of the SLC SSDs in the blades.

The virtualization layer and application development will be handled by the VMware vSphere Suite and VMware vFabric Suite.

Atlassian Confluence and the OpenFire XMPP server will handle collaboration.

SCADA and HMI systems will use proprietary applications have been built by the development team.

2.4.2 Cisco UCS Architecture

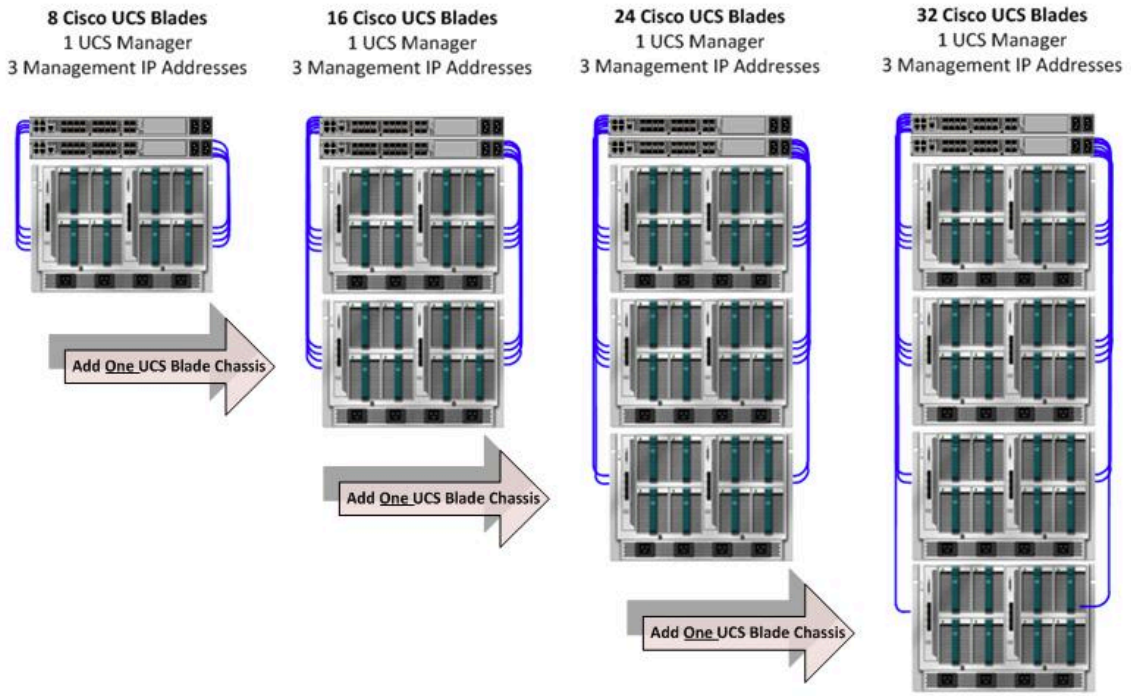


The Cisco Unified Computing System is a data center server platform composed of computing hardware, virtualization support, switching fabric, and management software.

The Cisco 6200 Series switch (called a "Fabric Interconnect") provides network connectivity for the chassis, blade servers and rack servers connected to it through 10 Gigabit converged network adapter.

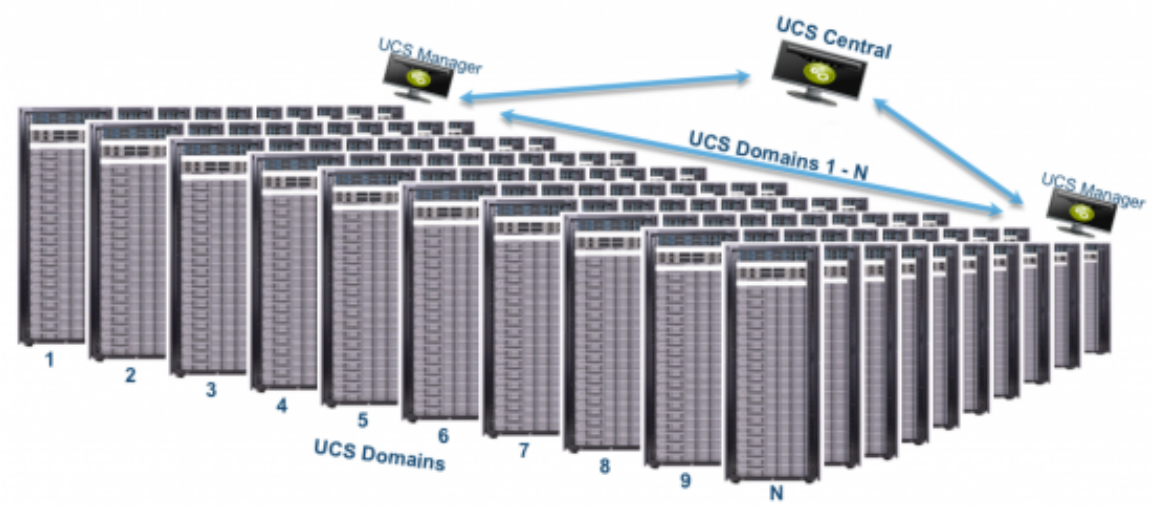
The Cisco UCS Manager software is embedded in the 6200 series Fabric Interconnect handles. The administrator accesses the interface via a web browser.

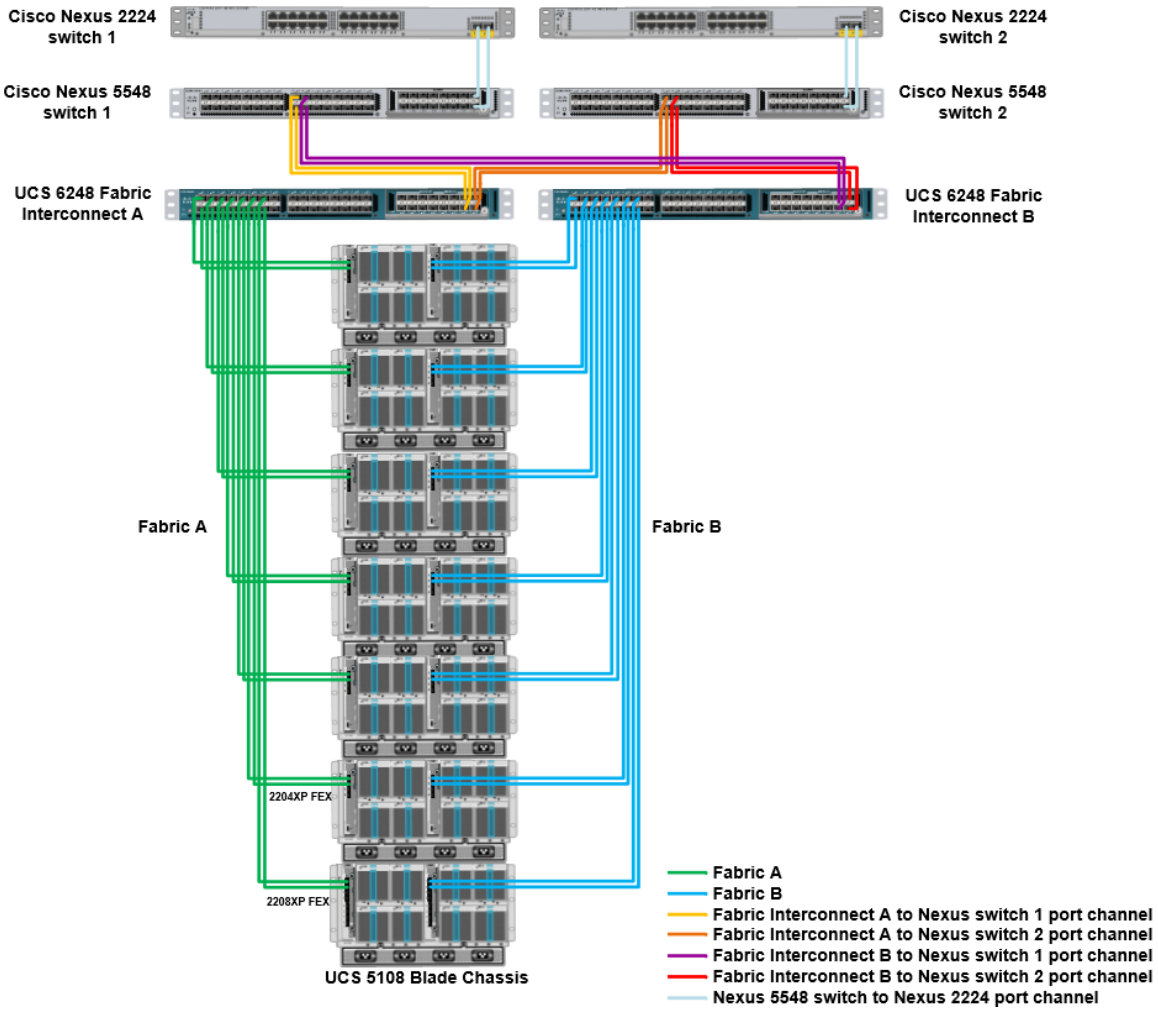
A key benefit is the concept of Stateless Computing, where each compute node has no set configuration. MAC addresses, UUIDs, firmware and BIOS settings for example, are all configured on the UCS manager in a Service Profile and applied to the servers.



In the diagram above you are able to see how to scale chassis in a UCS domain.

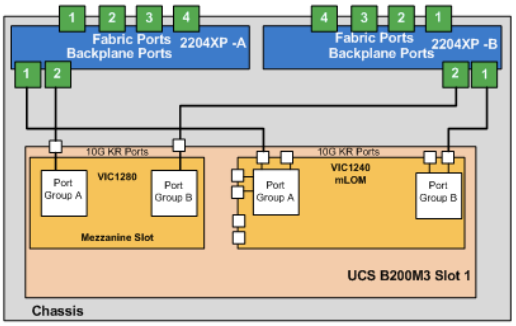
There is a maximum of 20 chassis per UCS domain. To scale beyond that, UCS central will be used to manage (n) domains.





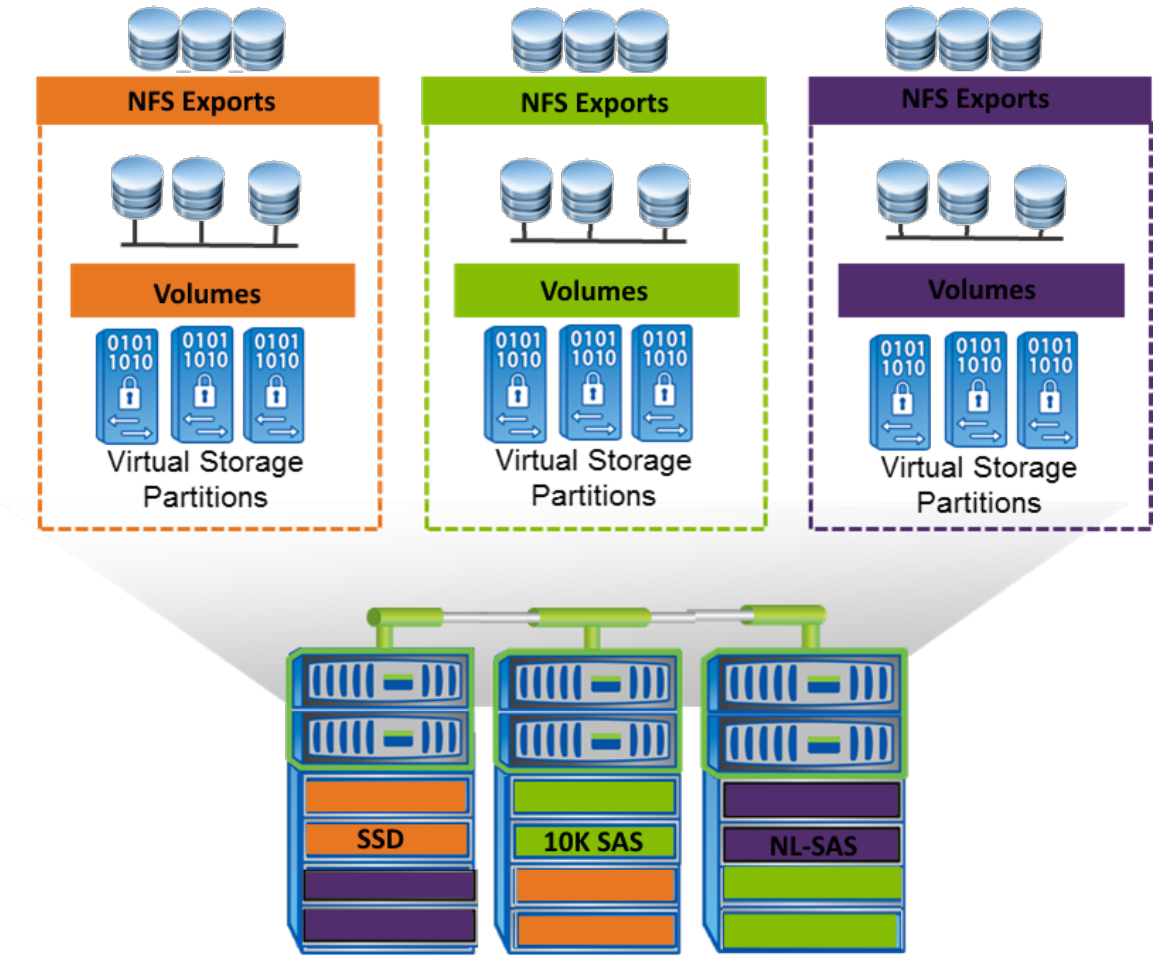
The above cabling diagram shows an example connectivity map with 7 chassis connecting to UCS 6248 Fabric Interconnects. The FIs then have a virtual port channel with the Nexus 5548 switches and then LAG connections with the fabric extenders.

The networking within the blade server is as below:



Mez	IOM	BandWidth
VIC 1280	2204XP	20Gb Per Fabric

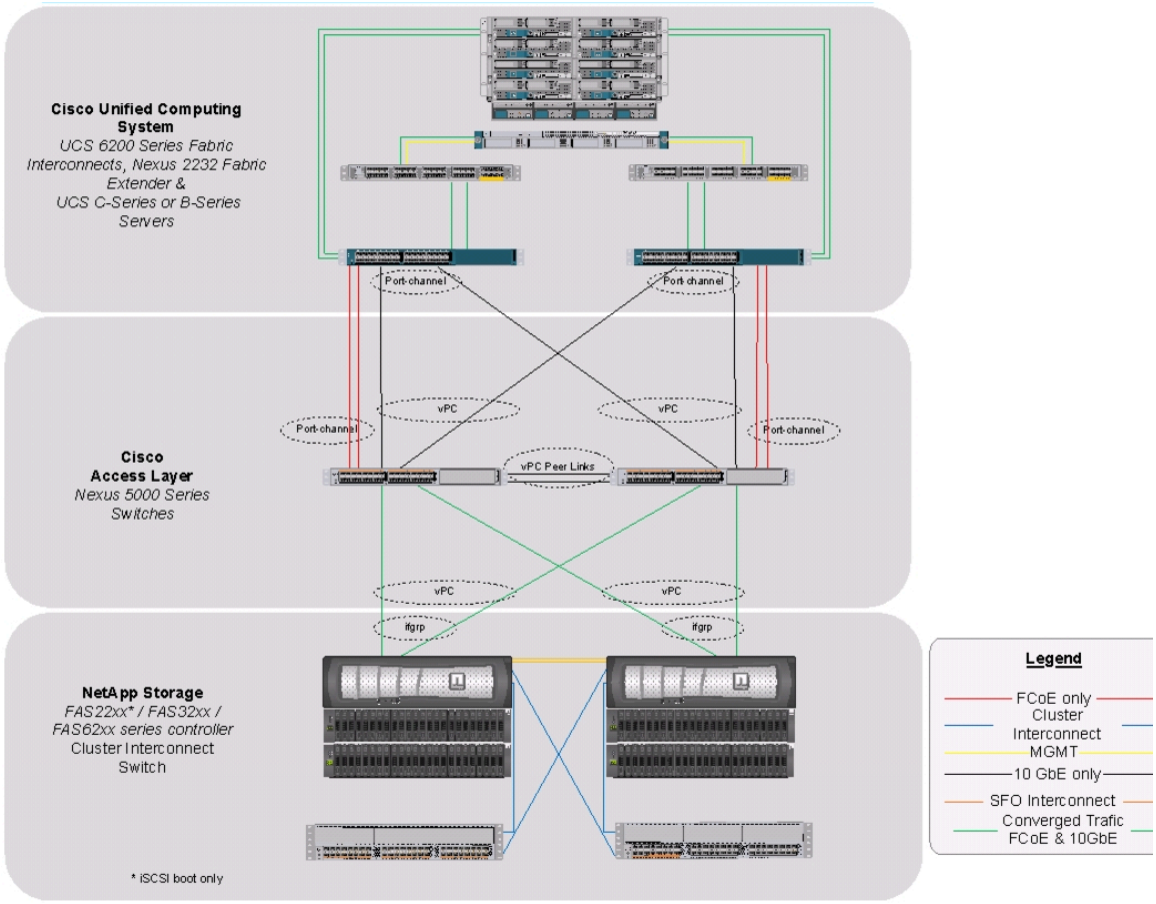
2.4.3 NetApp Architecture



The NetApp OnTap Cluster-Mode OS a horizontal scale out architecture. In the diagram above, you will see several cluster nodes. These nodes have the ability to span the filesystem across multiple aggregates and HA-Pairs by use of an SVM (Storage Virtual Machine). The SVM acts as a storage hypervisor and turns all the disks into resource pools that can be expanded and contracted as required.

By doing this, the system resilience and scalability has been increased. There is a risk that if not managed correctly that the storage system can start sprawling and troubleshooting performance issues is more time consuming due to the added complexity.

An example of the architecture is seen below.



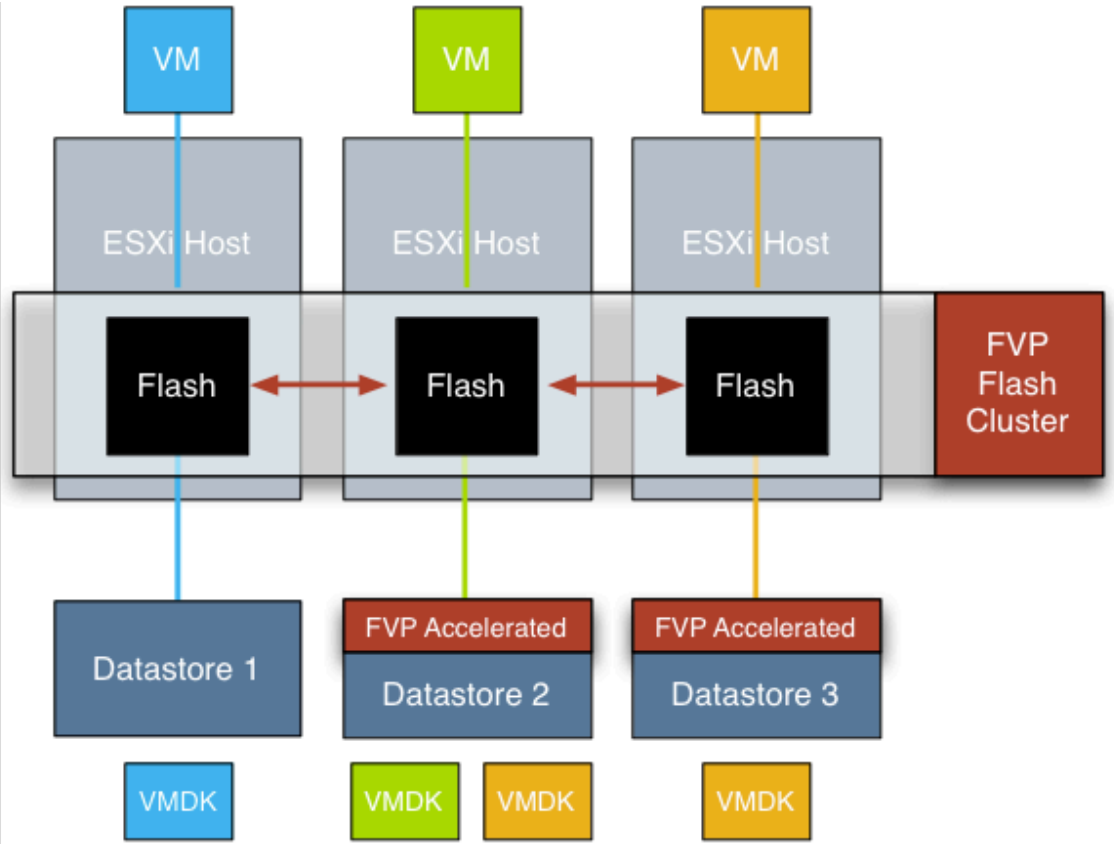
Detailed configuration guides for Cisco UCS, Cluster-Mode NetApp and Nexus switching can be reviewed in the Cisco Validated Designs (UCS_CVDs).

http://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/UCS_CVDs/esxi51_ucsm2_Clusterdeploy.html

2.4.4 PernixData Architecture

PernixData FVP virtualizes server-side caching that enables IT administrators to scale storage performance independent of capacity. Each server in the FVP Flash Cluster will make use of local SLC SSDs to cache read and write I/O of the storage system.

The VMs that require the extra IOPs will be placed on the FVP accelerated datastores.



2.5 Virtualization Network Layer

2.5.1 High Level Network Design Network Segmentation and VLANs

Within the Cisco UCS blades there are 2 VICs (virtual interface cards), an mLOM and a Mezzanine card. One is embedded in the server, the other is an add on card. These VICs then virtualize the physical NICs in the blade server so that any host OS running on the baremetal will see it as configured. In this design, we have presented 8 NICs to the host. These will be divided into the following VLANs and port groups.

VLAN 1011 – Management
VLAN 20 – Storage Communication (data channel)
VLAN 30 – vMotion
VLAN 1001-1999 – VM networks
Out of band (OOB) communication will be done on the Management network

The VIC to VNIC mapping is as follows:

VNIC0 – mLOM – Fabric A
VNIC1 - mLOM – Fabric B
VNIC2 - mLOM – Fabric A
VNIC3 - mLOM – Fabric B
VNIC4 - Mezz – Fabric A
VNIC5 - Mezz – Fabric B
VNIC6- Mezz – Fabric A
VNIC7- Mezz – Fabric B

2.5.2 Virtual Switches & Virtual Distributed Switches

There will be 2 vSwitches; a standard vSwitch for host management and a distributed vSwitch for all other communication. The uplinks will be as follows:

vSwitch0 is a VSS. Uplinks are:

VNIC0
VNIC5
There will be one management port group named VSS-1011-MGMT and one VMkernel, both on VLAN 1011

This is for interface redundancy. One is on the mLOM and the other is on the Mezzanine adapter.

vSwitch 1 is a VDS. Uplinks are

VNIC1
VNIC2
VNIC3
VNIC4
VNIC6
VNIC7

The VDS will have uplinks defined per port group.

The following port groups will be defined:

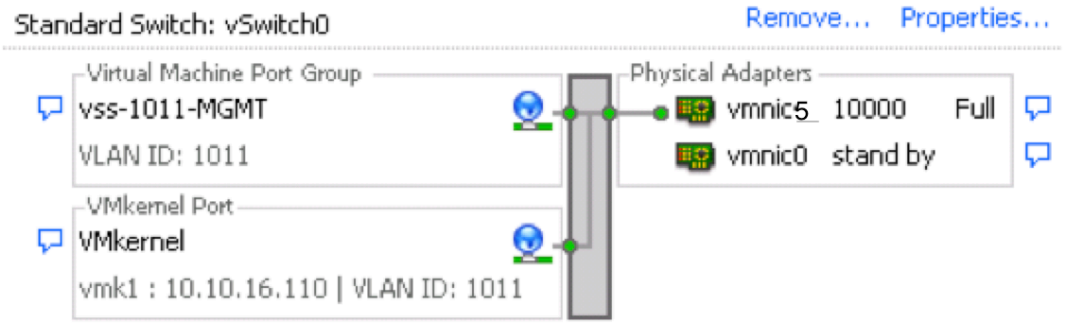
Name	VLAN	Purpose	Uplinks
VDS-1020-Storage	1020	Storage Connectivity	1,4
VDS-1030-vMotion	1030	vMotion	2,6
VDS-2001-VM	2001	VM Network	3,7

VM Network VLANs will span from VLAN 2001 to 2999 as required.

There is a VMKernel on VLAN 1020 for connectivity to the datastores and another on VLAN 1030 for vMotion.

2.5.3 NIC Teaming

The uplinks will be in active / standby configuration



2.5.4 Network I/O Control

Network IO control will not be applied on the VDS as there is already segmentation with the separate virtual interfaces in UCS. QoS will be applied at the UCS level by the Class of Service. Management will be Bronze, VM Data is Best Effort, and NFS is Gold. This is applied to the specific interfaces within UCS.

2.5.5 Physical Switches

The physical switches will be:

- Cisco UCS 6296 for the fabric interconnects
- Nexus 5548 for the upstream switches
- Nexus 2232 Fabric Extenders for any devices that require 1GB connectivity to the environment
- Nexus 5596 for the NetApp cluster Interconnects

2.5.6 DNS and Naming Conventions

Domain will be zombee.local
Hostnames will be constructed by location, division, role, numerical ID.

Example:
Location: Kennedy Space Center, Launch site 39
Division: Production
Role: ESXi server
Numeric ID: 001

FQDN: KSC-LC39-PROD-ESXi-001.zombee.local

2.6 ESXi Host Design

2.6.1 ESXi Host Hardware Requirements

Each blade server will have:

2 sockets with intel E52600 series CPUs (8 core).
512GB RAM
2 x 600GB SLC SSD drives
Cisco VIC1280 Mezzanine Card

The first blade in every chassis will have dual SD cards to boot ESXi from. This for the eventuality that a total power loss occurs and the vCenter server does not start. vCenter will have DRS affinity rules to those hosts.

2.6.2 Virtual Data Center Design

Each division will have it's own vCenter instance, named as per the naming conventions, ie: KSC-LC39-PROD-VC-001

There will be one datacenter defined by the division name.

2.6.3 vSphere Single Sign On

Single Sign-on will be used and authenticated to Active Directory

2.6.4 vCenter Server and Database Systems (include vCenter Update Manager)

The vCenter Server Appliance (VCSA) will be used, as the total number of hosts is within the maximums. Update Manager will be deployed on it's own VM.

2.6.5 vCenter Server Database Design

The embedded database will be used for the vCenter server. A separate Windows server 2008 R2 server with SQL 2008 standard will be used for the shared database for other components, such as VUM.

2.6.6 vCenter AutoDeploy

vCenter AutoDeploy will be used to deploy all hosts, except for the 20 (1st blade in a chassis) hosts that are booting from SD.

2.6.7 Clusters and Resource Pools

There will be 5 clusters spanned across the 4 UCS cabinets and 20 chassis. 32 hosts will be in a cluster.

a. Enhanced vMotion Compatibility

2.6.8 Fault Tolerance (FT)

FT will not be used.

2.7 DRS Clusters

HA and DRS will be enabled and set to aggressive.

vCenter will have affinity rules to the first server in the chassis for the cluster it is in.
Any clustered application servers or databases will have anti-host affinity or if required, anti-chassis affinity rules.

HA admission control will be set to 25% tolerance.

2.7.1 Multiple vSphere HA and DRS Clusters

Each cluster will have the same roles, so the rules will stay the same. The only exception is where vCenter is located, which is cluster-01

2.7.2 Resource Pools

Resource pools will not be used unless required.

2.8 Management Layer Logical Design

2.8.1 vCenter Server Logical Design

The vCenter server will be on the VCSA with 32GB of RAM allocated to support all the hosts. Active directory will be installed on a VM named KSC-LC39-PROD-DC-001. A shared SQL server for VUM will be located on a VM named KSC-LC39-PROD-SQL-001. vCenter Update manager will be named KSC-LC39-PROD-VUM-001.

2.8.2 Management and Monitoring

vCenter Operations Manager will be used to monitor the environment in detail.

Log insight will be used to provide real-time analysis and speedy root cause analysis.

2.9 Virtual Machine Design

2.9.1 Virtual Machine Design Considerations

Operating systems will be comprised of a system volume and one or more data volumes. System volumes will not be larger than 50GB in size and will be thin provisioned by default.

Swap files for all VMs will be located on a swap datastore.
No RDMs will be used.

Virtual Hardware Version 10 will be used by default.
All operating systems that support VMXNET3 will use it as the network adapter of choice.

2.9.2 Guest Operating System Considerations

All VMs will be provisioned from templates and configuration policies applied afterwards. No configuration changes will be applied manually.

2.9.3 General Management Design Guidelines

All management of the environment will be done by authorized personnel only, from a dedicated VM named KSC-LC39-PROD-MGMT-001. All actions will be audited and reviewed.

2.9.4 Host Management Considerations

The UCS servers have a built in IP KVM in their management framework. The UCS plugin will be added to vCenter so that baremetal host management can occur from there.

2.9.5 vCenter Server Users and Groups

Active Directory will be used as the user management system. Roles based on access levels and job function will be applied to the appropriate groups.

2.9.6 Management Cluster

Cluster 1 will be the management cluster. All VMs for management purposes, whether virtual infrastructure, storage or otherwise, will run on cluster 1.

2.9.7 Management Server Redundancy

vCenter server is protected by HA across the cluster. In a worst case scenario, the blades with SD cards will still be able to boot and start the vCenter server if the storage is accessible.

2.9.8 Templates

All virtual machines will be spawned from a master template. Using VAAI, rapid cloning of virtual machines can be done.

2.9.9 Updating Hosts, Virtual Machines, and Virtual Appliances

vCenter update manager will be used to update hosts, VMs and appliances.

2.9.10 Time Synchronization

A GPS synced time server will be used for accurate time measurement. All hosts will connect to that NTP server. Active Directory VMs will be set to obtain the time of their hosts, then provide it to network computers.

2.9.11 Snapshot Management

VM snapshots will stay no longer than 24 hrs. SAN snapshots will use a 10% volume reserve and occur at 15min, 30min, 1hr, 4 hr, 8 hr, Retention will be hourly, daily and weekly.

2.10.1 Performance Monitoring

Performance monitoring will be done by vCOPs for the infrastructure and vFabric Hyperic for application analysis.

2.10.2 Alarms

All default vCenter alarms are enabled and the recipient is set to an internal email user.

2.10.3 Logging Design Considerations

Log insight will be deployed as well as the dump collector. Logs will be configured to go to a shared datastore.

2.11 Infrastructure Backup and Restore

2.11.1 Compute (ESXi) Host Backup and Restore

Host configuration will be maintain in a single host profile and backed up with a PowerCli script.

2.11.2 vSphere Replication

vSphere replication will not be used, as it's a single site. It may be used for inter-division migrations down the road.

2.11.3 vSphere Distributed Switch Backup and Restore

The VDS config will be backed up from the vSphere client

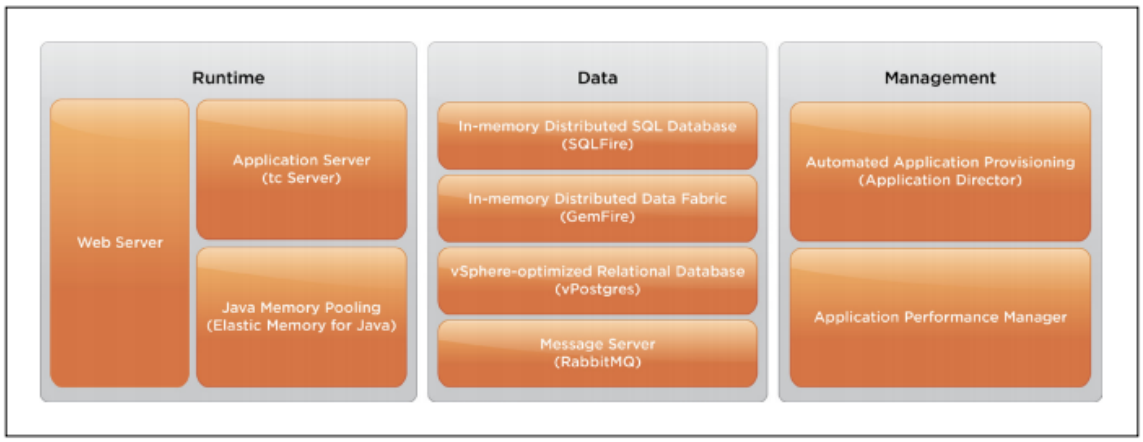
2.11.4 vCenter Databases

vCenter uses the internal VCSA database. The other components that require external databases are using KSC-LC39-PROD-MGMT-001

2.12. Application provisioning automation

All VMs will be provisioned from templates then puppet will apply the appropriate configuration. For custom-built applications, the vFabric Suite will be used.

2.12.1 vFabric Overview



vFabric is a development and automation framework for creating multi-tier application and managing code release cycles. With the combination of vFabric Hyperic, (for testing application performance) and vFabric Application Director (for scaling and deployment automation), complex applications can be created, tested and deployed using a Continuous Integration and Continuous Deployment model.