OPTUM Care®

IT Infrastructure Design Document

Workload Scheduling & Automation

Core Service

Version 1.5

Revision Control:

Revision:	Date:	Author:	Change Summary:	Approval:
1.0	6/15/2020	Ken Houghtling	Initial investigation & design	
1.5	7/15/2020	Ken Houghtling	Sending out for design review	
1.6	7/23/2020	Ken Houghtling	Adjust diagram to fix phasing views	

Table of Contents

Overview

Currently the OPTUMCare (Legacy DMG) enterprise domain does not offer a centralized Automated Workload solution. Automation of batch jobs on systems with the domain is conducted on an ad-hoc basis within the scope of individual systems. These solutions are often complexed in terms of establishing and ongoing support. The operations teams are pressed into service to schedule, manage, and support these point solutions ultimately distracting them from primary duties.

Operational and their peer engineering resources are requiring debugging and monitor workload execution via brute-force methods diverting resources from operational stabilization. The lack of a uniform workload scheduler solution requires individual diverse skills that often differ even within the scope of common operating systems. End-users (subject matter experts) who develop jobs that require automated schedule with interdependences on other schedule jobs would like to manage their own scheduled workload. The end-user community understand the execution strategy and interferences want a self-service solution to allow them to manage and debug their own processes.

The IDX System deployed on RedHat Linux OS is the impetus for this effort as it is currently leveraging native “cron” facility to support Automated Workloads. The “cron” facility does not offer functionality and supporting functions required by the IDX System.

The Operational LINUX and Window Server Support Team(s) both require almost indent features and functionality. This document strives to capture the requirements to facilitate the identification of an enterprise solution.

Objectives:

The project objectives include:

- **Capture Business and Functional Requirements**
- **Research and Design a solution that addresses capability requirements**
- **Provide initial tool recommendations**
- **Document proposed design rollout**

Requirements

Business Requirements:

Enterprise Requirements:

- Must provide a Core Service Functional Solution
- Supported by Core Funding Model
- Core Management
- HA / Failover Support offerings
- Event Logging
- Anomaly Detection

Operating System Support:

- Can schedule jobs on Linux (Red Hat EL 7 and Red Hat EL 8, Ubuntu 18+)
- AIX
- Windows Server (2012,2016,2019)
- VM Aware
- Cloud Control Capability

Cyber Security:

- Enterprise Security Alignment
- GPO Alignment
- Encrypted communication traffic between console and endpoint servers
- User authentication with configurable role-based access:
    - Capability of a "read only" role to see status of scheduled jobs
    - Allow users access to only certain groups of jobs or certain groups of servers
- Control groups with capability to support RBAC paradigm

Functional Requirements:

- Provides a centralized operator console
- Ability to schedule jobs from the console
    - Schedule job to run as any authorized user
    - Robust scheduling capability
    - Complex schedules   For Example "last day of the month" or "every third Tuesday of the month"
    - Ability to set maximum runtime on job
- Fault Mitigation
    - Schedule can be resumed cleanly after server reboot or crash
- Robust job failure detection
    - Ability to easily re-run failed jobs

- Workload Serialization - schedule one job to run after another job has completed successfully
- Ability to handle (on the order of) many hundreds of distinct jobs per server
- Ability to see or impute what was running at a point in time in history
- Ability to transmit job output text via email
- Majority of scheduling operations work can be performed by L1 resources (no deep complexity)
- Integrates with ServiceNow for ticket generation
- Ability to engage VAR installation/migration and/or VAR managed services of product

Solution Context

Workload Automation/Scheduling is the practice of leveraging software assets schedule, trigger, run, and manage workloads supporting repetitive business and IT processes. Workloads can be a manual or automated that takes place during course operations of a business. Workload Automation is a close sibling to Job Scheduling, which is the process of scheduling, triggering, and running resource-intensive workloads in during desirable periods oftener referred to as Batch Windows.

The solution facilitates a centralized design where a core system/service maintains the scheduling for ALL jobs throughout multiple domains. While there are some opportunities for proactive mediation of issues and integration activities it does require specialized knowledge or specific cross domain data analysis strategies.

The Workload Scheduler is to be a core service offering within the scope of the Operations & Infrastructure Service Catalog. It will consist of a core scheduler with persistent job(s) administration and execution oversight. Access will be limited to administrative staff and certified end-users authorized to manage workload jobs for their individual or group assigned domains.

As depicted the Scheduler will coordinate job execution, logging and reporting within a scope domain of workload servers.

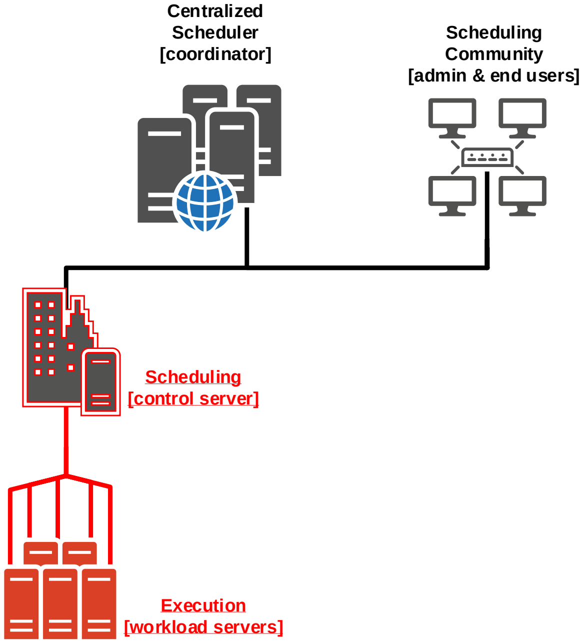

Figure 1: Workload Scheduler Solution Context

The core service will provide the desired integrity and security within access control groups with least amount of privilege afforded to the scheduling end user community.

High Availability, Scalability a Growth:

Essential design requirements for the Workload Scheduling solution require a “High Availability (HA)”, “Salacity”, and “Growth” capabilities. The design will account for these through a Master – Replicant strategy. The Master Scheduling Engine can support the role of any domain replicant. Domain replicant can support peer replicant domains in an outage situation. Master & Replicates facilitate and manage job execution on assigned workload servers based on the schedule assign by the administrative or end user communities.

HA is achieved by multi environment design deployments:

1) The Master Scheduling environment will include HA Clustering
2) The Master can take over any Replicants role based on configuration or manual intervention
3) Replicants can assume localized or cross domain controls based on configuration

The following diagram depicts a high-level overview of operations and possible failover scenarios within the thermotical deployment.

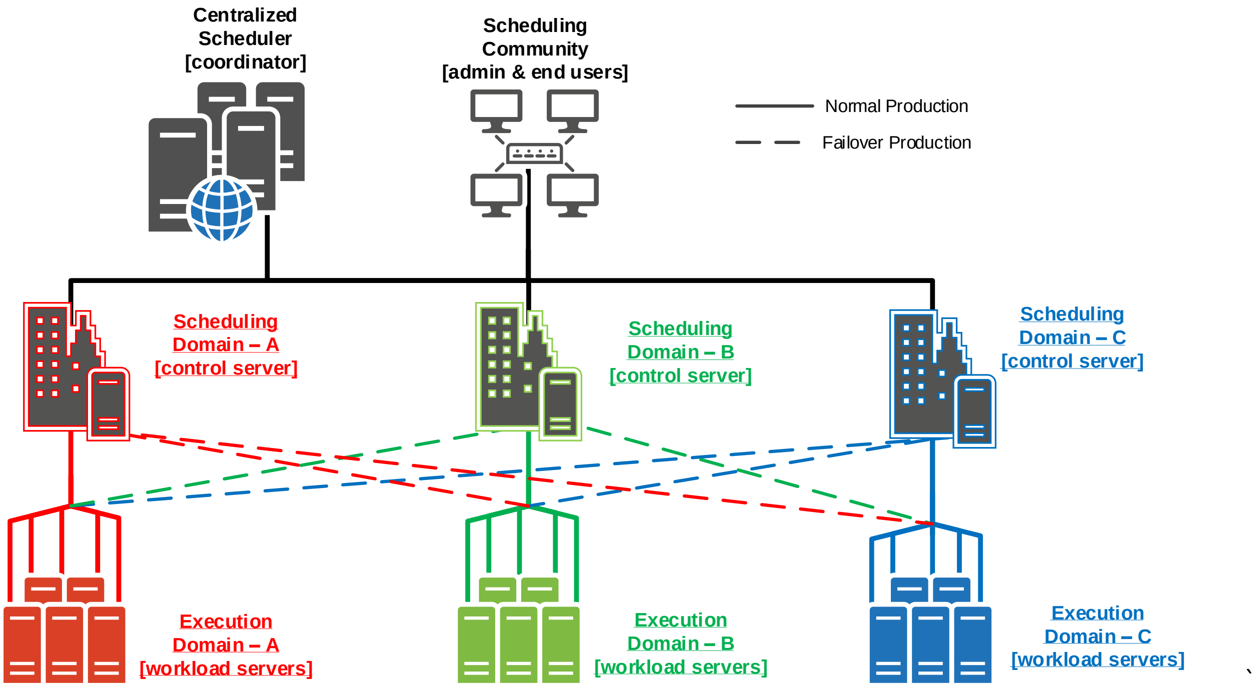
Centralized Scheduler [coordinator]
Scheduling Community [admin & end users]
Normal Production
Failover Production
Scheduling Domain – A [control server]
Scheduling Domain – B [control server]
Scheduling Domain – C [control server]
Execution Domain – A [workload servers]
Execution Domain – B [workload servers]
Execution Domain – C [workload servers]

Figure 2: Enterprise Workload Scheduler Context

Scalability is achieved via adding additional server resources to the Master or Domain Replicants as needed.

Growth is supported by adding additional Domain Replicants. As new tenants are brought into the enterprise environment Domain Replicants allow for nondisruptive growth within the larger enterprise. Have individual domain replicants allow for easily removing a domain should the situation arise.

Access and Authorization will be aligned with current Active Directory and GPO(s) core services. The ACL(s) for the Workload Scheduling solution will be strictly control within the AD facility. This includes domain privileges and end user access controls.

Workload Execution Support:

Workload execution is the Workload Schedule primary function. Workloads are defined as any digital asset that is executed manually or automatically is support of a business process. Currently OPTUMCare is leveraging tools like: Windows Task Scheduler or Cron to facilitate workload processing. These tools lack certain capabilities identified in the Requirements section of the document. Many of these capabilities are curial improvements required for execution stability, workload interdependency and monitoring the workload environment.

Capabilities

Any Workload Scheduler solution must include the fundamental and advance function identified in this sub-section.

Infrastructure

- Cross-platform or cross-infrastructure support (Windows Server (2016-2019), RedHat Linux(6-8.x), AIX, etc.)
- Scalability including horizontal and vertical scaling of resources and domain level control
- High Availability
- Data Center interoperability – OPTUMCare is ramping up to deploy and maintain at least 3 Data Centers incorporating a Hybrid Cloud solution, all these environments need to be supported by solution set
- Network Tenancy Support – OPTUMCare' s growth model is based on acquisition as new business entities are acquired, they are migrated into their own network tenant

Security

- Role Based Security
- AD Integration
- ACL Groups
- Tenant Based Domain support for segregated schedulers management domains
- Operational logging for auditing purposes

Scheduling

- Advanced scheduling capabilities must include:
  - Complex job scheduling time slots
  - Adhoc job controls
  - Operational maintenance support

- Job interdependency support allow for workload execution and completion awareness prior to the execution of subsequent depend jobs
- Conflict notification and Resolution

Reporting

- Workload Success including related metrics
- Workload failure including event logging for failure analysis

Monitoring

- Scheduler control
- Scheduler Health
- Event notification (Including: Success, Informational, Warning, Error, Critical Failure)
- Integration with Event Management system (Splunk)

Operations

- Central management of most (or all) platforms and technologies
- Operational Dash Boarding at all levels (Master to domain level)
- System maintained controls
- Resource level notifications

Solution Specification:

Product Selection:

The IBM Tivoli Workload Scheduler (TWS) product has been selected for several reasons

1) Exists on the UHG / OPTUM approved products list
    a. Require less Risk Analysis and Compliance review time
    b. MSA 7 Business agreements in place to facilitate procurement
2) Product is mature and has a stable future support model
3) Deployment options and support opportunities
    a. Can be hosted on several platforms including (Windows Server & Red Hat Linux)
    b. Supports MS SQL Server as its persistence layer
4) Scalability factors
5) Growth factors
6) High Availability factors

Master Domain Manager Environment:

TWS operates with a domain hierarchy topology. The Master Domain Manager controls all sub-domains and can act as a proxy if required. The is a persistence layer leverage by the Master Domain Manager to store and report on all Workload Scheduler for all domains.

There is a Web Server that acts as the Dynamic Workload Console server. This abstractive interface is the preferred path of Workload Scheduling management. The Master Domain Manager command line capability. Its usage is recommended for qualified-trained administrative staff.

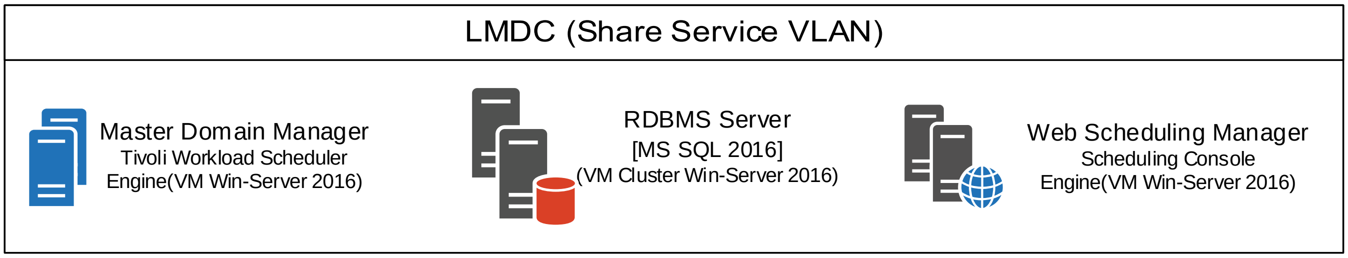
LMDC (Share Service VLAN)
Master Domain Manager
Tivoli Workload Scheduler
Engine(VM Win-Server 2016)
RDBMS Server
[MS SQL 2016]
(VM Cluster Win-Server 2016)
Web Scheduling Manager
Scheduling Console
Engine(VM Win-Server 2016)

Figure 3: Master Workload Scheduling Env.

The prescribe target for the installation of the Master Domain Manager environment is the Lone Mountain Data Center (LMDC). This will act as the CORE service environment.

Hardware Specific Requirements:

Domain	Resource Description	OS	VCPUs	RAM	System Disk	Disk Application	Disk RDBMS
Master Domain	Master Domain Scheduler	Win-Server 2016	8	16 (GB)	100 (GB)	75 (GB)	N/A
	Master Domain Scheduler Alternate	Win-Server 2016	8	16 (GB)	100 (GB)	75 (GB)	N/A
	Master Domain MS SQL SERVER (Cluster)	Win-Server 2016	8	16 (GB)	100 (GB)	75 (GB)	200 (GB)
	Master Domain MS SQL SERVER (Cluster)	Win-Server 2016	8	16 (GB)	100 (GB)	75 (GB)	200 (GB)
	TWS Scheduler WebServer (Cluster)	Win-Server 2016	8	16 (GB)	100 (GB)	75 (GB)	N/A
	TWS Scheduler WebServer (Cluster)	Win-Server 2016	8	16 (GB)	100 (GB)	75 (GB)	N/A

Network Specific Requirements:

Firewall Rules:

When there is a firewall between any of the IWS components, respective Firewall rules must be put in place to allow communication through the specific IWS ports. In addition, Behind the Firewall option in the Workstation definition forces all communication between the Main Domain Manager and the FTA to adhere to the Domain hierarchy.

The following list identifies all IWS ports required to be open between various components.

- The following ports need to be opened from MDM FTA:
  - 31111 net man port of the FTA
  - 3*114 Job Manager port of the dynamic agent where * is the instance number of the agent.
    For example:
    - *31114 for the 1st instance of the agent*
    - *32114 for the 2nd instance of the agent on the same server*
    - *33114 for the 3rd instance of the agent on the same server, etc.*
- The following ports need to be opened from FTA MDM:
  - 31111 net man port of the MDM
  - 31131 Event Processor port on the MDM
  - 31116 Dynamic Workload Broker server port on the MDM
- The following ports need to be opened from DM FTA:
  - 31111 netman port of the FTA
- The following ports need to be opened from FTA DM
  - 31111 netman port of the DM
- The following ports need to be opened from CLI/GUI MDM
  - 9443 HTTPS port on the DWC
  - 31114 HTTPS port on the MDM
  - 31116 composer port on the MDM
  - 31117 Bootstrap port on the DWC

Domain Manager:

This component supports multi-domain network implementation . The Domain Manger manages workloads by assigning it to a predefined workstation normally run a workload statically. In a multi-domain network, all domains below the master domain have fault-tolerant agents configured to be a domain manager to manage the workstations in its domain. A domain manager can manage fault-tolerant, standard, and extended agents. Each domain manager is a fault-tolerant agent in the domain of the next higher level. To define a domain manager, install a fault-tolerant agent on your workstation and then define it as manager in the workstation definition.

Workload Scheduling Agents:

Workload Scheduler controls schedules and runs work on multiple distributed platforms via Tracker-Agents running on every machine under its control. The Tracker-Agent is a program running on Unix, Windows, HP-UX, or any other operating system installed in each workload operations environment. Tracker-Agent communicates with the Controller to feedback information about jobs starting and ending. The Controller uses that information to update its Current Plan, changing statuses and submitting jobs as their predecessor’s finish.

Fault-tolerant agent:

A fault-tolerant agent can resolve local dependencies and launch jobs in the absence of a domain manager. It has a copy of the production control file. This allows fault-tolerant agents to continue processing even if the dynamic domain manager or the network connection is down.

Standard agent:

An agent that launches jobs only under the direction of its domain manager. It is not fault-tolerant. To define a standard agent, install a fault-tolerant agent on your workstation and then define it as a standard agent in the workstation definition.

Extended agent:

Extended agents are logical definitions (hosted by a physical workstation) used to extend job processing to selected applications with integration support.

Recommend Project Approach:

A phased approach is the recommended project path. This approach will allow the project to address the most immediate need (the IDX workload scheduling). Subsequent phases will address the remaining domains in the existing CDO(s) (MWR & FL). The Future Growth Phase provided for the adding in the new Data Centers and acquired CDO(s).

Phase - I:

The initial phase establishes the Master CORE Service and the CA Domain Workload Scheduling environment. This will also allow for some configuration refinement and performance tuning.

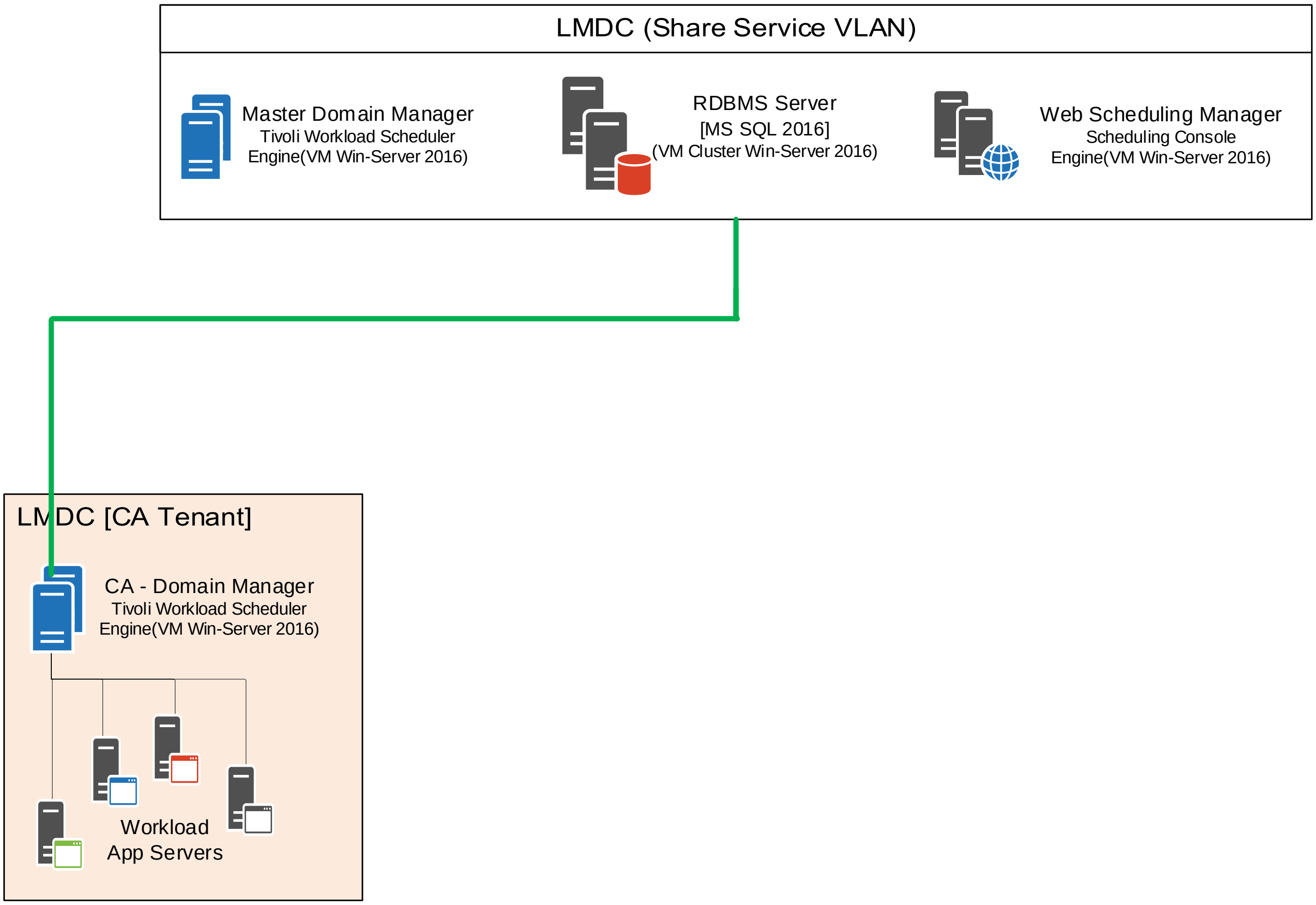
LMDC (Share Service VLAN)
Master Domain Manager
Tivoli Workload Scheduler
Engine(VM Win-Server 2016)
RDBMS Server
[MS SQL 2016]
(VM Cluster Win-Server 2016)
Web Scheduling Manager
Scheduling Console
Engine(VM Win-Server 2016)
LMDC [CA Tenant]
CA - Domain Manager
Tivoli Workload Scheduler
Engine(VM Win-Server 2016)
Workload
App Servers

Figure 4: Phase I - Rollout Diagram

CA - Domain manager

The CA (California) – Domain Manager will facilitate and manage fault-tolerant, standard, and extended agents. Each domain manager is a fault-tolerant agent in the domain of the next higher level.

Hardware Specific Requirements:

Domain	Resource Description	OS	VCPUs	RAM	System Disk	Disk Application	Disk RDBMS
CA - CDO Domain	CA Domain Scheduler	Win-Server 2016	8	16 (GB)	100 (GB)	75 (GB)	N/A
	CA Domain Scheduler Alternate	Win-Server 2016	8	16 (GB)	100 (GB)	75 (GB)	N/A

Network Specific Requirements:

Refer to the network and firewall requirements found in Network Specific Requirements: found on page-number: 12.

Phase - II:

This phase support the establishing the MWR (Mount West Region) and FL (Florida) domains. Since the Master the environment has already be established this phase will be more of an expansion then established TWS core service.

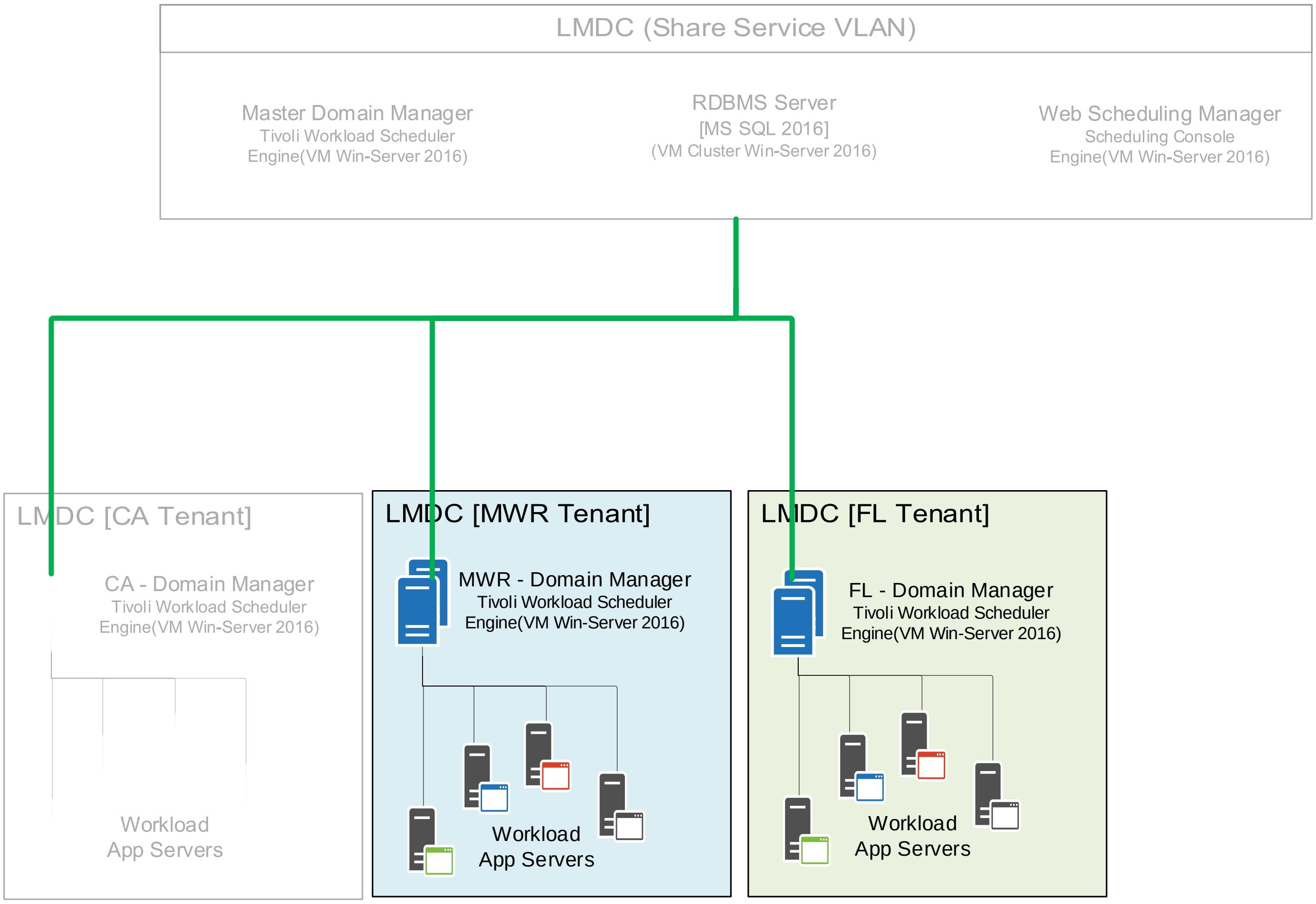
LMDC (Share Service VLAN)
Master Domain Manager
Tivoli Workload Scheduler
Engine(VM Win-Server 2016)
RDBMS Server
[MS SQL 2016]
(VM Cluster Win-Server 2016)
Web Scheduling Manager
Scheduling Console
Engine(VM Win-Server 2016)
LMDC [CA Tenant]
CA - Domain Manager
Tivoli Workload Scheduler
Engine(VM Win-Server 2016)
Workload
App Servers
LMDC [MWR Tenant]
MWR - Domain Manager
Tivoli Workload Scheduler
Engine(VM Win-Server 2016)
Workload
App Servers
LMDC [FL Tenant]
FL - Domain Manager
Tivoli Workload Scheduler
Engine(VM Win-Server 2016)
Workload
App Servers

Figure 5: Phase II Rollout Diagram

Hardware Specific Requirements:

Domain	Resource Description	OS	VCPUs	RAM	System Disk	Disk Application	Disk RDBMS
MWR - CDO Domain	CA Domain Scheduler	Win-Server 2016	8	16 (GB)	100 (GB)	75 (GB)	N/A
	CA Domain Scheduler Alternate	Win-Server 2016	8	16 (GB)	100 (GB)	75 (GB)	N/A
FL - CDO Domain	CA Domain Scheduler	Win-Server 2016	8	16 (GB)	100 (GB)	75 (GB)	N/A
	CA Domain Scheduler Alternate	Win-Server 2016	8	16 (GB)	100 (GB)	75 (GB)	N/A

Network Specific Requirements:

Refer to the network and firewall requirements found in Network Specific Requirements: found on page-number: 12.

Future Growth:

Future growth should follow the same path as Phase II. Infrastructure and networking will require analysis to insure the necessary Domain Managers and agents' requirements are mitigated.

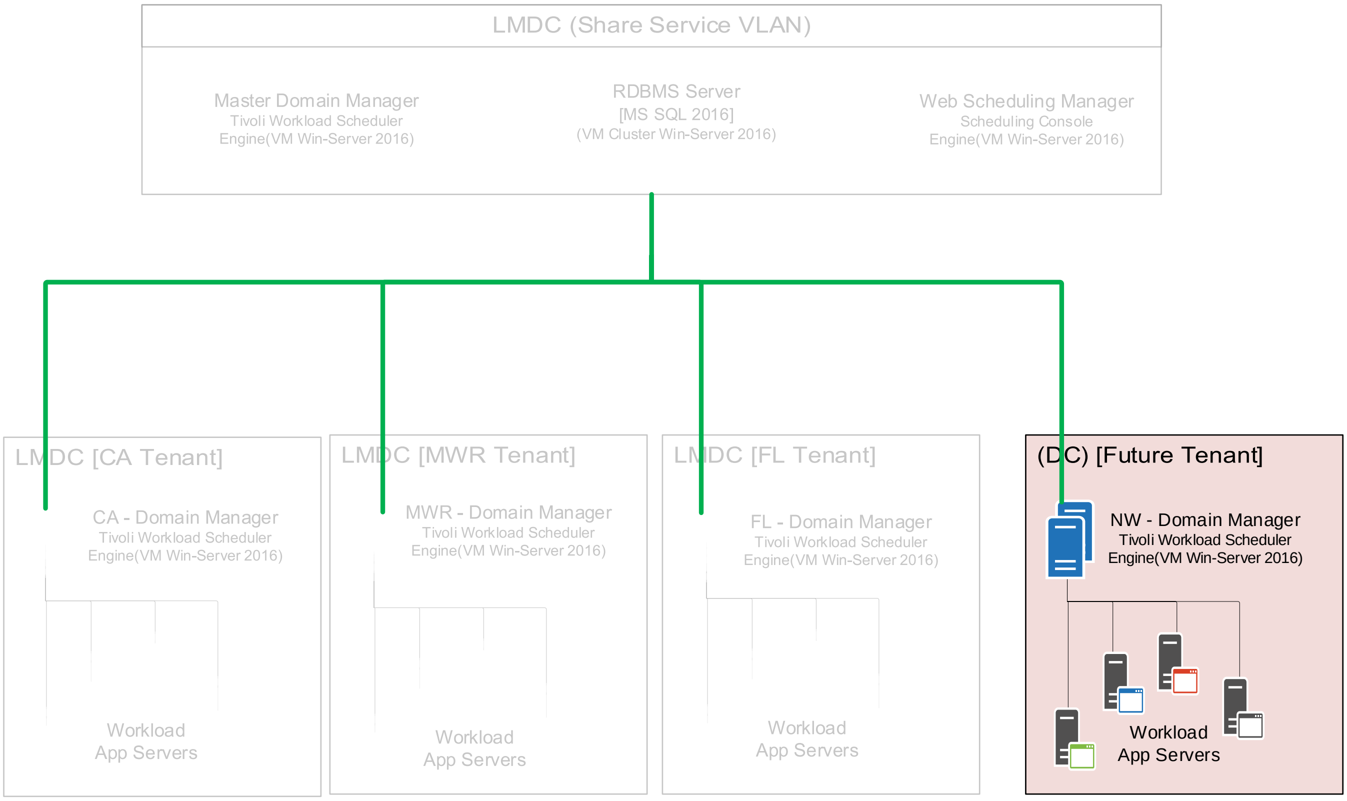
LMDC (Share Service VLAN)
Master Domain Manager
Tivoli Workload Scheduler
Engine(VM Win-Server 2016)
RDBMS Server
[MS SQL 2016]
(VM Cluster Win-Server 2016)
Web Scheduling Manager
Scheduling Console
Engine(VM Win-Server 2016)
LMDC [CA Tenant]
CA - Domain Manager
Tivoli Workload Scheduler
Engine(VM Win-Server 2016)
Workload
App Servers
LMDC [MWR Tenant]
MWR - Domain Manager
Tivoli Workload Scheduler
Engine(VM Win-Server 2016)
Workload
App Servers
LMDC [FL Tenant]
FL - Domain Manager
Tivoli Workload Scheduler
Engine(VM Win-Server 2016)
Workload
App Servers
(DC) [Future Tenant]
NW - Domain Manager
Tivoli Workload Scheduler
Engine(VM Win-Server 2016)
Workload
App Servers

Figure 6: Future Expansion Diagram

Outstanding Issues:

The following issues require attention prior to production rollout for the Workload Scheduling Automation CORE Service:

1) Production Turnover "As-Built" documentation
2) Engineering Assignment to assume responsibility of the Workload Scheduling Automation CORE Service
3) Define and implement Operational Support model for the Workload Scheduling Automation CORE Service
    a. Monitoring and Reporting of the Workload Scheduling Automation CORE Service aligned with Command Center operations.
4) Define and implement (SNOW) Queues to facilitate:
    a. New Domain Manager Quests
    b. New workload server rollout (Agents)
    c. Problem resolution queues
    d. Product support
    e. Domain / Server retirement
    f. End User Workload Scheduler access

Appendix

References

Table of Figures