

Microsoft SQL Database Enterprise Design Standards

Version 1.0

Author: Henry Shen

Released: Dec 15 2016

Contents

[Document review and approval 3](#_Toc473121115)

[Document Contributors 3](#_Toc473121116)

[1.0 | Overview 4](#_Toc473121117)

[2.0 | Objectives 4](#_Toc473121118)

[3.0 | Analysis 4](#_Toc473121119)

[4.0 | Requirements 5](#_Toc473121120)

[5.0 | Solution Summary 5](#_Toc473121121)

[5.1 | Design Decision.01 - Shared Model 6](#_Toc473121122)

[5.2 | Design Decision.02 – Tiered Database Service Offering 8](#_Toc473121123)

[6.0 | Standards 10](#_Toc473121124)

[7.0 | Design Specifications 10](#_Toc473121125)

[8.0 | SQL Scale out File Server Witness Share 13](#_Toc473121126)

[9.0 | Constraints, Assumptions, Risks 15](#_Toc473121127)

[10.0 | Assignment and Validation 16](#_Toc473121128)

[11.0 | Implementation 17](#_Toc473121129)

## Document review and approval

This document has been reviewed and approved by:

|  |  |  |
| --- | --- | --- |
| Name | Role | Date |
| Paras Sharma | Provincial Manager, Architecture Services |  |
| Paul Hundal | Provincial manager, Database Services |  |

## Document Contributors

The following individuals have been involved in design and planning.

|  |  |
| --- | --- |
| Name | Role |
| Paul Hundal | Provincial manager, Database Services |
| Alex Suprun | Database Administrator, Database Services |
| Han Wei Chan | Solution Architect, Architecture Services |
| Cliff Sandford | Solution Architect, Architecture Services |

SQL Database Design Standards

# | Overview

Current database environments among the various health authorities have become fragmented lacking consistent standards. As a result, recent database migrations have exposed these risks that must be addressed holistically. These inefficiencies not exclusive to database version control are compounding database management. The continuing deployments of database implementation pose a significant cost for BCCSS and warrant the need for a comprehensive strategy in database management.

# | Objectives

As a centralized source promoting shared services, BCCSS seeks to align and establish strategic direction and guidance for the database environment. Moving towards a common shared and centralized model across health authorities will also help standardize user experience and a more sustainable management approach.

Instituting a shared design standard promotes:

* Efficiencies (performance, procurement, operations, cost, resource allocation, licensing)
* Familiarity (user experience, support)
* Manageability (resource allocation, administration)
* Stabilization (consistent and up-to-date product version control)

# | Analysis

Database implementations among the Health Authorities continue to be deployed in an isolated application oriented configuration. The tendency is for databases to be implemented as stand-alone servers coupled with the application whereby they eventually become unmanaged silos resulting in the following characteristics:

* over-subscribed resource allocation
* Orphaned systems
* Version relapse
* Application coupling
* Overlapping database server redundancies
* Excessive database software licensing

# | Requirements

* Back level versions of SQL
* High availability
* Legacy application preservation
* Custom configuration and security isolation
* Large workloads
* Cost conscious options
* Consolidated licensing

# | Solution Summary

To address existing challenges in resource allocation, cost reduction and database management, greater control must be exercised by establishing clear direction and workable standards.

BCCSS architecture and database infrastructure services seek to institute a new enterprise design model that incorporates the following elements:

1. Re-useable standard model
2. Software defined data centre

Using this approach, these overarching tenets provide the outline in driving design decisions to shape how the future state of MS SQL databases will be envisioned.

The following tables have been used to capture the design decision process in how the solution was derived.

# | Design Decision.01 - Shared Model

|  |  |
| --- | --- |
| ID Number | DD.01 |
| Subject Area | MS SQL Enterprise database design |
| Issue or Problem | Decentralized database sprawl and version relapse |
| Architectural Decision | Standardize all database deployments on a shared model basis with high availability |
| Status | Decided. This has not yet been approved |
| Motivation | Isolated servers hosting only a single database are inefficient and costly |
| Assumptions | Reserved VM cluster and hosts are available for deployment |
| Positions | * Win2003 Migration (main driver are costly CSAs with vendor) * DB team in favour of standardization * Architecture team in favour of standardization * Endeavour to move forward with this initiative * Do nothing and maintain status quo |
| Justification | Properly spec’ed DB servers are more than capable of hosting multiple databases on a single common server. This not only saves on licensing costs but also make more effective use of compute, memory, storage and data centre real estate. Moreover centralizing DBs on DB servers will facilitate DB administration from a tracking, identification and administration perspective. |
| Implications | Cost savings and sustainable efficiencies. Database oversight under one umbrella. |
| Derived Requirements | See section 4.0 |
| Related Decisions | * Efficient resource allocation * Data centre density * Licensing economies * Centralized DB management * Standardization * Controlled database oversight * Referenceable design standards * Convergence * Virtualization |
| Notes |  |

**Table 1: DD.01 Design Decision**



# | Design Decision.02 – Tiered Database Service Offering

|  |  |
| --- | --- |
| ID Number | DD.02 |
| Subject Area | MS SQL Enterprise database design |
| Issue or Problem | Resource over allocation and unstructured growth |
| Architectural Decision | Tiered database service offering |
| Status | Decided. This has not yet been approved. |
| Motivation | Tiering facilitates standardization by accommodating various levels of requirements |
| Assumptions | Existing workloads will be able to function within the proposed tiers |
| Positions | * Win2003 Migration (main driver are costly CSAs with vendor) * DB team in favour of standardization * Architecture team in favour of standardization * Endeavour to move forward with this initiative * Do nothing and maintain status quo |
| Justification | DB deployments are based on requirements. These requirements vary by scale and are more appropriately addressed by classification at established levels. |
| Implications | Organized approach at DB deployment, leveraged economies, better insight to growth and requirement propensities. |
| Derived Requirements | See section 4.0 |
| Related Decisions | * Vertical scaling vs. horizontal scaling * Structured growth * Software based service offering * Unique OS and/or DB security requirements * Legacy preservation * Functional categorization * Variable levels of high availability * Project scale and DB requirements * Re-useable model * Cloud portability |
| Notes |  |

Table 2: DD.02 Design Decision



# | Standards

|  |  |
| --- | --- |
| BCCSS MS SQL Standards | |
| Hypervisor | VMware |
| Database | MS SQL 2016 Enterprise Ed. |
| Deployment Default | Shared |
| High Availability Default | Always On |
| Preferred Datacentre | KDC |
| Witness listener | Dedicated VM. One per datacentre |

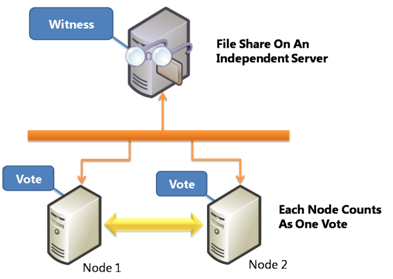
|  |  |
| --- | --- |
| Database Tiers | |
| Tier | DB Characteristics |
| 1 | Shared VM | 1. Large DBs 2. Virtualized 3. SQL 2016 4. Highest availability |
| 2 |Dedicated Physical | 1. Large DB 2. Physical 3. SQL 2016 4. High availability 5. Cost sensitive |
| 3 | Dedicated VM | 1. Application and DB co-tenancy requirement 2. Legacy DB and/or OS version 3. Sensitive security requirements 4. Lower high availability requirements |

# | Design Specifications

|  |  |  |  |
| --- | --- | --- | --- |
| Shared | | | Dedicated |
| Tier | 1 | 2 | 3 |
| HA type | Always On Availability Group  (OS and DB level) | Failover Cluster  (OS and DB level) | Hypervisor options (OS level) |
| State | Virtual | Physical  (No virtualization. VM team will not support fail over clustering on a hypervisor) | Virtual |
| Allocation type | Shared  Dedicated VM SQL DB Cluster | Shared  Dedicated physical chassis slot. Build out as needed | Dedicated  Dedicated VM SQL DB Cluster |
| OS version | Windows *current* R2 DataCentre  (2012) | Windows *current* R2 Standard  (2012) | 1. Windows *current* R2 datacentre (2012) 2. Windows *current-1* R2 Standard (2008) 3. Windows *custom* Standard Ed. (Dependent) |
| SQL version | SQL *current* Enterprise Ed.  (2016) | SQL *current* Enterprise Ed. (2016) | 1. SQL *current* Standard Ed.2016 2. SQL *current-1* Standard Ed.2014 3. SQL *custom* Standard Ed. dependent |
| Components | Database engine | * Database engine * Analysis services | * Database Engine * Analysis Services   + Analysis Services in Power Pivot * Reporting Services   + Reporting Services SharePoint Mode * Integration Services * R Services |
| Standard Tools | 1. Anti-Virus 2. Monitoring  * MS Ops Manager (system center) * IDERA (optional agent that project would have to fund – per DB instance) | 1. Anti-Virus 2. Monitoring  * MS Ops Manager (system center) * IDERA (optional agent that project would have to fund – per DB instance) | 1. Anti-Virus 2. Monitoring  * MS Ops Manager (system center) * IDERA (optional agent that project would have to fund – per DB instance) |
| Infrastructure Requirements | vCenter  SAN storage   * Need to find out available disk technology (Ben Haley)   Licensing | Rack space  Power  HVAC  OS  DB Enterprise Licensing | * Dedicated vCenter VMware cluster for stand alone DB * SAN Storage |
| Benefit | Highest availability  Load balancing  Efficient utilization  Consolidated licensing | Lower storage cost | Legacy preservation |
| Disadvantage | Higher storage cost | Slow to deploy  Higher OPEX costs  Higher facility consumption  Higher licensing cost | Very high vendor support costs  Highest total cost  No BI features |
| Expected adoption | 85% | 5% | 10% |
| Cores / VM | 8 (subject to change. Talk to Shane to determine what a feasible amount to initially allocate would be) | 4 core(Depends as this is not Virtual) | 2 (subject to change) talk to shane |
| Memory / VM | 64 | 32 | 16 (subject to change) |
| Storage  C: (OS)  H: (DB)  I: (Logs) J: (TempDB) K: (Backup) | 60GB Any  *25*% of DB *25*% of DB *150*% of DB | 60GB  Any  *25*% of DB  *25*% of DB  150% of DB | 60GB  <500GB  *25*% of DB  *25*% of DB  150% of DB |
| # of VM clusters | 1 /environment | 1/environment | 1/environment |
| Hosts/VM cluster | 2 (to support vMotion and anti-affinity) with ability to scale out as needed | 2/cluster. New cluster growth added once capacity approaches limit. | 2 (to support vMotion and anti-affinity) with ability to scale out as needed. |
| Max nodes / availability group or fail over cluster | 2 to start. 4 max | 2 | n/a |
| AG witness listener | Single dedicated VM hosted by DB VM cluster.   * Single core * 2 GB RAM * 60GB OS   Configured shares as needed | | |
| AG / tier | Type 1: Shared server and DB instance  Type 2: Shared server, separate DB instance  Type 3: Shared server and BI instance  Type 4: Dedicated server and DB instance  1 with ability to scale out (contingent on capacity and security requirements) | n/a | n/a |
| Cluster / tier | n/a | 1 with ability to scale out (contingent on capacity and security requirements) | n/a |
| Licensing | SQL Enterprise per core | SQL Enterprise per core | SQL Standard per core |
| Environments | Production   * Non-Clinical (standard) * Clinical * DMZ   Non-Production | Production   * Non-Clinical (standard) * Clinical * DMZ   Non-Production | Production   * Non-Clinical (standard) * Clinical * DMZ   Non-Production |
| Instance creation | Multiple. As needed. | Single | Single |
| DB | Multiple. Shared | Single | Single |
| BI | Shared. Physical implementation. SSRS, SSAS, SSIS is not supported under Always On. HA can only be achieved through clustering. |  |  |

# | SQL Scale out File Server Witness Share

In order to implement SQL AG groups with 2 nodes, a windows share witness disk is required to achieve quorum incase a SQL node goes down.



Given the critically of this server, HA and a target of 100% uptime is required. The proposed solution is to build a windows scale out file server to enable the HA of the SMB share, supported by SAN backend storage.

SAN storage is a requirement of a scale out file server as the storage has to be available to all nodes of the HA FS farm.



By building a HA file share witness cluster, we can connect multiple SQL AG clusters and eliminate the need for individual file share witness servers that are currently being built on a per cluster basis.

Given the fileshare is hosted on a windows machine there are a number of factors that could cause it to go offline, hence the need for HA.

1. Windows Updates
2. Scheduled backups
3. Failure of an underlying ESXi host.

In order to enable a scale out file server design on a VM hosted by ESXi the following design is proposed:



# | Constraints, Assumptions, Risks

Assumptions

* Each tier will be fully implemented and tested before any database is deployed
* All tiers will be actively monitored for capacity
* Additional server hosts will be added as required before capacity limitations
* All health authorities will adopt and attempt to enforce these standards with new onboarding projects. Vendors will be made aware of BCCSS standards and requirement for conformity, compatibility
* All new requirements will be validated by solution architects based on performance metrics and DB specific technical requirements (not combined with OS)
* Always-on availability groups will have full fail over capabilities with zero or minimal interruption
* Dedicated VM clusters per tier with appropriate number of hosts
* VM hosts have capacity for vMotion fail over as needed
* Backup capacity will be in-line with
* Antivirus scanning exclusions and scheduling
* Appropriate monitoring and alerting will be in place
* Databases tenants will be properly assigned based on requirements
* Common database outage windows will not be relevant as any maintenance on server or DB will be satisfied by redundant nodes in availability group or fail over cluster for shared servers and instances
* Always-on nodes that belong to a common Always-on group will
* All DB deployments will reside in dedicated clusters

Risks

* Undetermined costing model. High upfront VM host requirement costs to infrastructure or to project.
* Length of time in building tiers and delay to urgent projects in need of DB implementations
* Continued DB deployments that are not captured by this new endeavour
* Clinical and non-clinical security zones may dictate and grow excessive SQL tiers (will need to determine if access type (i.e. user based or application based DB connectivity) is more appropriate

Constraints

* Legacy applications that cannot upgrade or function with DB versions

# | Assignment and Validation

The flowchart depicted below will help determine how a database should be assigned under the new database service offering model. To ensure proper designation, tier assignments should be confirmed by an assigned DBA.



# | Implementation

Once a tier assignment has been determined, the database will need to be implemented. This can happen in one of 2 methods:

1. New build
2. Migration

Method 1 is straight forward and will simply require:

* Project budget and approval
* Capacity verification from DB management services
* Approval from architecture services

Method 2 is more complex and while requires all of the prerequisites in method 1, involves a migration method using either a phased or direct approach.

* Phased. Suitable for large scale database implementations involving multiple database servers and large scope user base. Typically application subject matter experts are required.
* Direct. Suitable for smaller scale database implementation where coordinating outages and number of servers is manageable and will not occur over an extended period of time.

Each method must assess the following criteria on criticality, risk and readiness. Finally, migration methods may be more easily determined by identifying the end state of a database in question.

|  |  |
| --- | --- |
| Database End State | |
| State | Procedure |
| New | 1. Confirm requirements and obtain approvals 2. Assign tier |
| Copy | 1. Prepare new DB 2. Move Data    1. using application to populate    2. using backup to restore    3. manually between old and new DB 3. Confirm integrity via testing |
| Decommission | 1. No owner 2. No identifiable business impact 3. Monitor DB (ports), server network activity 4. Service , function retirement or new upgrade planned |
| Legacy Extension | 1. Determine business impact and criticality of application    1. Maintain but outline costs and impact to business    2. Present costs and consequences    3. Obtain departmental cost codes and VP level sign-off for maintaining status quo 2. Identify owner and engage vendor 3. Determine vendor recommended upgrade path    1. Upgrade DB only    2. Upgrade app and DB 4. Find alternate application/vendor that matches BCCSS DB standards 5. Find internal application that can substitute and extend capability to fill this void 6. Virtualize or decommission 7. Upgrade OS but not DB |