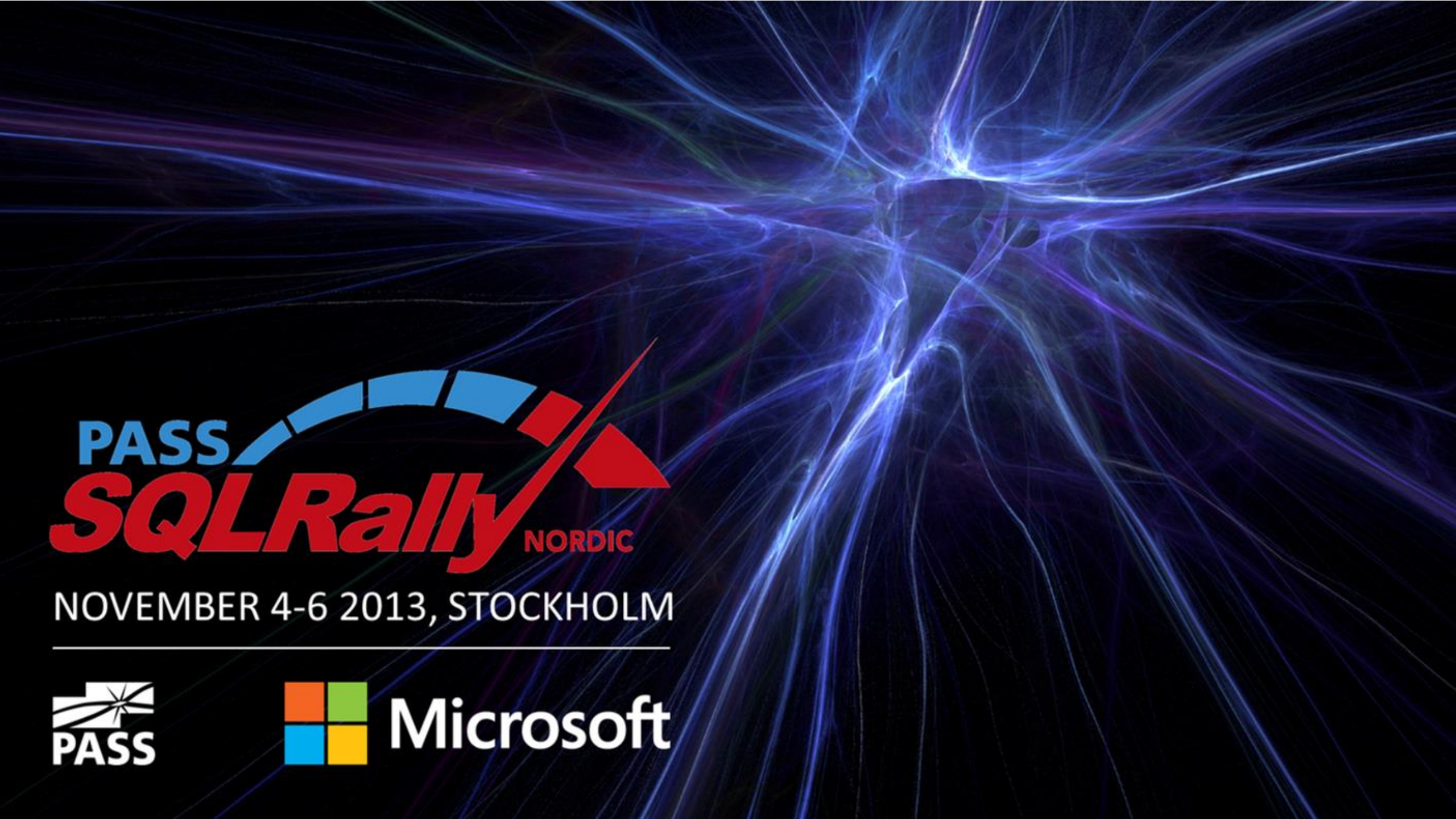

PASS
SQLRally
NORDIC
NOVEMBER 4-6 2013, STOCKHOLM
PASS
Microsoft

SharePoint for the DBA

Régis Baccaro

Introduction

Régis Baccaro

@regbac

regis@baccaro.com

http://Theblobfarm.wordpress.com

http://Thelovefarm.wordpress.com

- Founder and lead organizer of SQL Saturday Denmark
- Works for IBM
- Passionate about the community
- .Net developer, BI guy, SharePoint fellow and accidental DBA

Why SharePoint at a SQL Server event ?

- SharePoint heavily depends on SQL Server
- DBA are from Venus and SharePoint admins from Mars
- You need coordination to make it work!

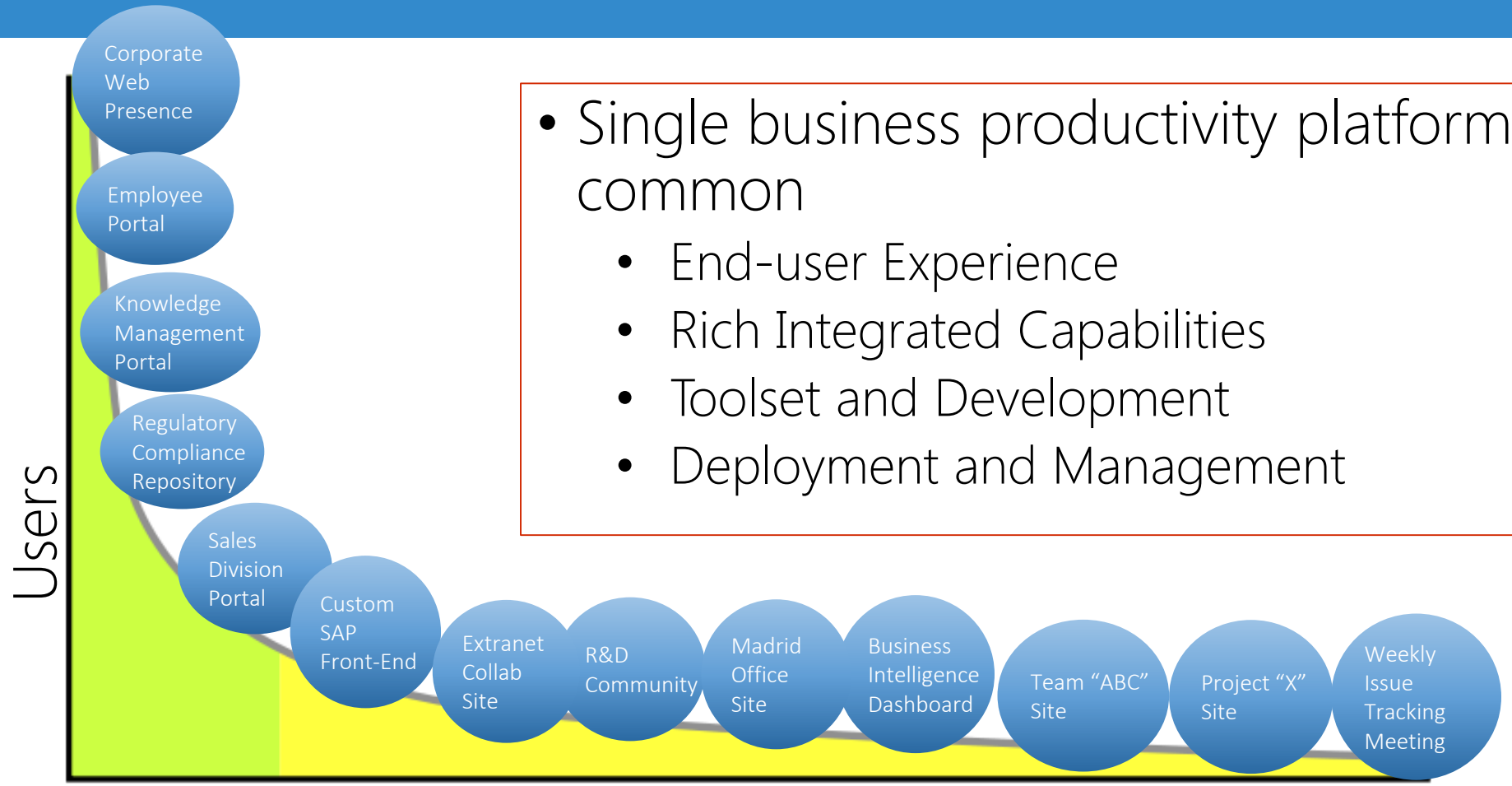
Agenda

- Introduction
- SharePoint Architecture & Design Considerations
- Planning SQL for SharePoint
- Deployment/Configuration/Security
- Maintenance
- High Availability

Introduction

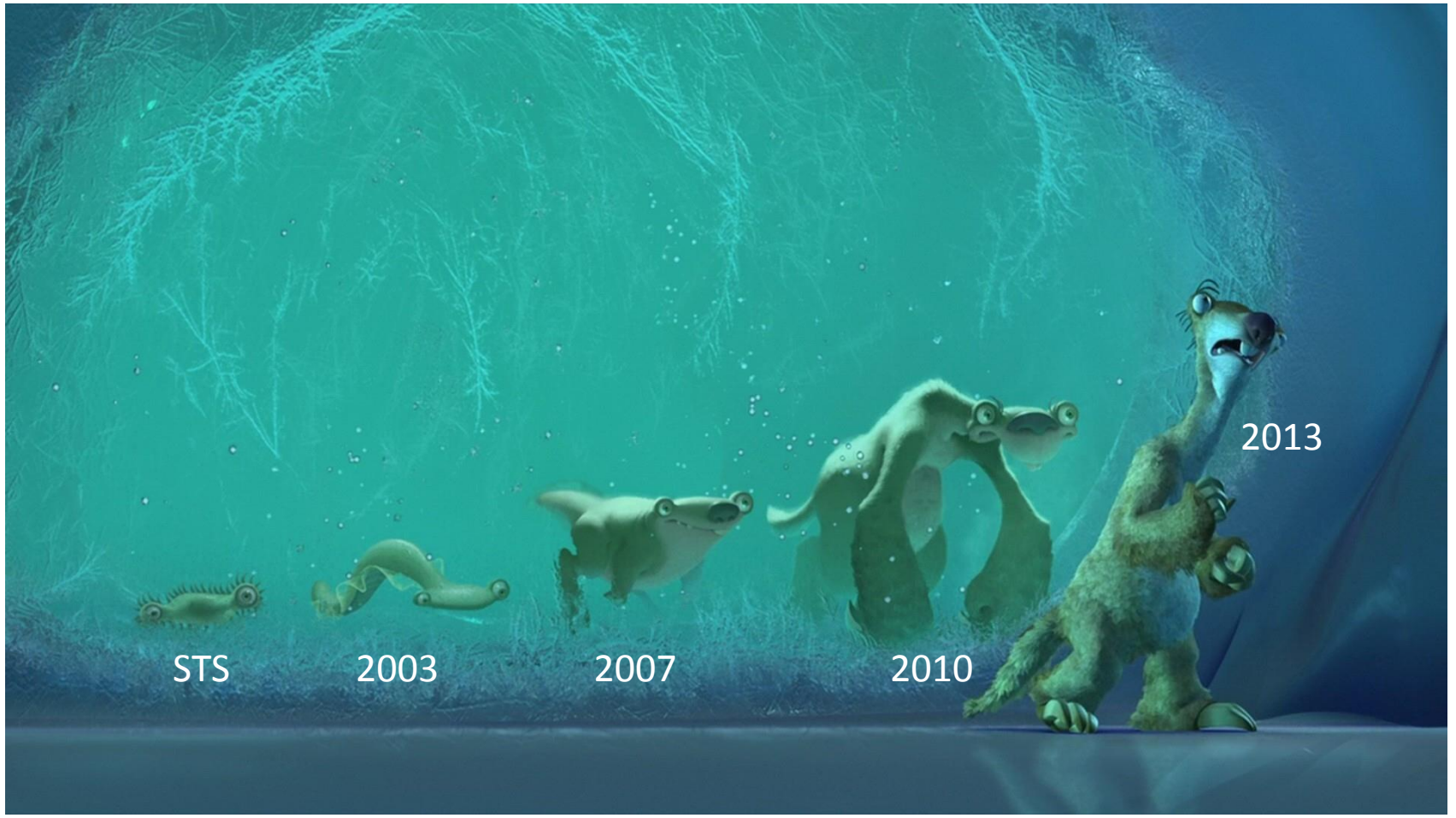
- What is SharePoint?
- Why is SQL Server so important?

Introduction – What is SharePoint ?



- Single business productivity platform leading to common
  - End-user Experience
  - Rich Integrated Capabilities
  - Toolset and Development
  - Deployment and Management

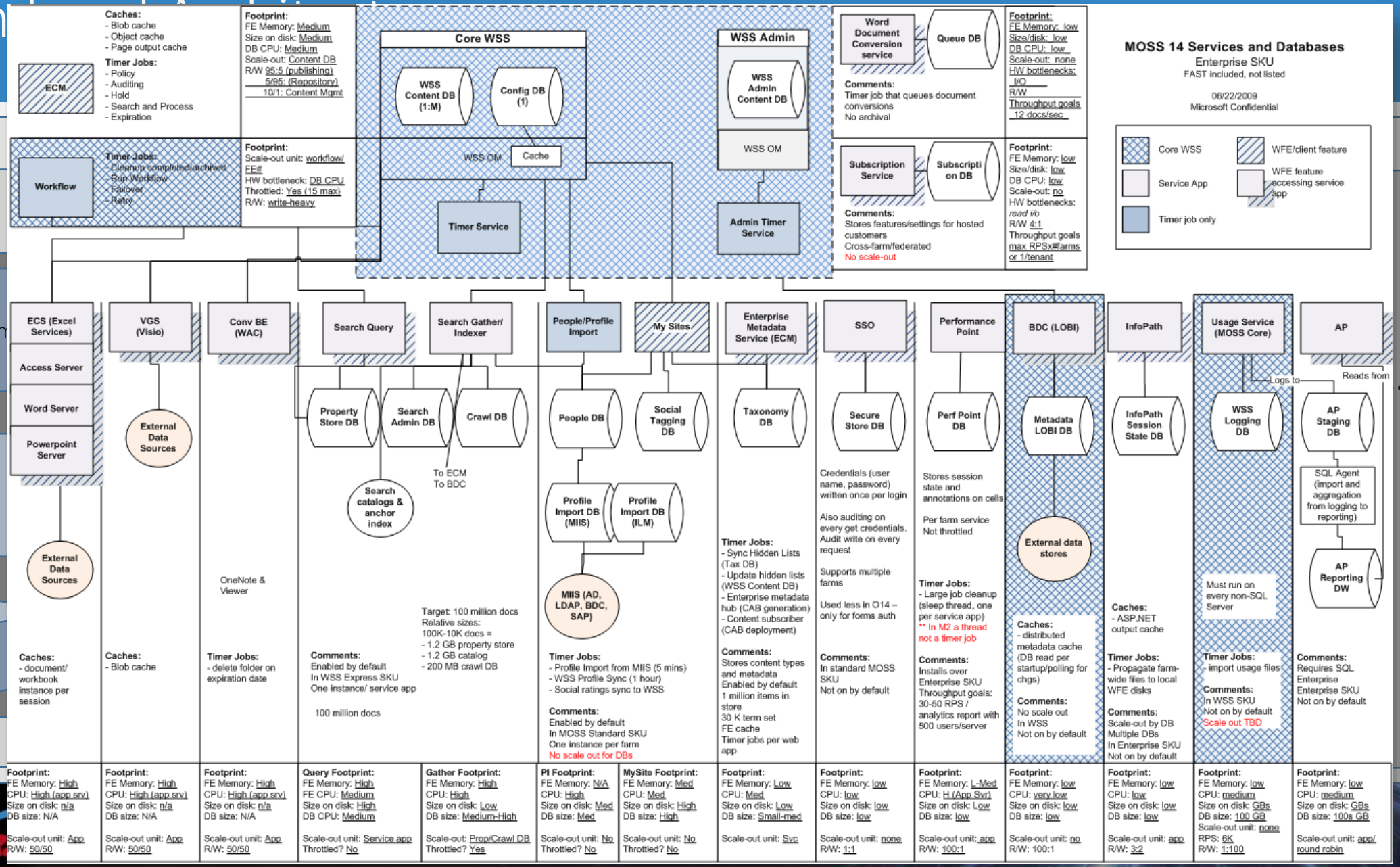
SharePoint through the ages



SharePoint

- Very fast
- Broadly adopted
- Very adaptive

High


MOSS 14 Services and Databases
Enterprise SKU
FAST included, not listed
06/22/2009
Microsoft Confidential
Core WSS
WFE/client feature
Service App
WFE feature accessing service app
Timer job only
Core WSS
WSS Content DB (1:M)
Config DB (1)
WSS OM
Cache
Timer Service
WSS Admin
WSS Admin Content DB
WSS OM
Admin Timer Service
Word Document Conversion service
Queue DB
Comments: Timer job that queues document conversions No archival
Footprint: FE Memory: low Size/disk: low DB CPU: low Scale-out: none HW bottlenecks: I/O R/W Throughput goals 12 docs/sec
Subscription Service
Subscription DB
Comments: Stores features/settings for hosted customers Cross-farm/federated No scale-out
Footprint: FE Memory: low Size/disk: low DB CPU: low Scale-out: no HW bottlenecks: read i/o R/W 4:1 Throughput goals max RPSx#farms or 1/tenant
ECM
Caches: - Blob cache - Object cache - Page output cache
Timer Jobs: - Policy - Auditing - Hold - Search and Process - Expiration
Footprint: FE Memory: Medium Size on disk: Medium DB CPU: Medium Scale-out: Content DB R/W 95:5 (publishing) 5/95: (Repository) 10/1: Content Mgmt
Workflow
Timer Jobs: - Cleanup completed/archived - Run Workflow - Failover - Retry
Footprint: Scale-out unit: workflow/ FE# HW bottleneck: DB CPU Throttled: Yes (15 max) R/W: write-heavy
ECS (Excel Services)
Access Server
Word Server
Powerpoint Server
External Data Sources
Caches: - document/ workbook instance per session
VGS (Visio)
External Data Sources
Caches: - Blob cache
Conv BE (WAC)
OneNote & Viewer
Timer Jobs: - delete folder on expiration date
Search Query
Property Store DB
Search Admin DB
Crawl DB
Search catalogs & anchor index
To ECM
To BDC
Comments: Enabled by default In WSS Express SKU One instance/ service app
100 million docs
Search Gather/ Indexer
Target: 100 million docs
Relative sizes:
100K-10K docs =
- 1.2 GB property store
- 1.2 GB catalog
- 200 MB crawl DB
People/Profile Import
People DB
Profile Import DB (MIIS)
Profile Import DB (ILM)
MIIS (AD, LDAP, BDC, SAP)
Timer Jobs: - Profile Import from MIIS (5 mins) - WSS Profile Sync (1 hour) - Social ratings sync to WSS
Comments: Enabled by default In MOSS Standard SKU One instance per farm No scale out for DBs
My Sites
Social Tagging DB
Enterprise Metadata Service (ECM)
Taxonomy DB
Timer Jobs: - Sync Hidden Lists (Tax DB) - Update hidden lists (WSS Content DB) - Enterprise metadata hub (CAB generation) - Content subscriber (CAB deployment)
Comments: Stores content types and metadata Enabled by default 1 million items in store 30 K term set FE cache Timer jobs per web app
SSO
Secure Store DB
Credentials (user name, password) written once per login
Also auditing on every get credentials. Audit write on every request
Supports multiple farms
Used less in O14 – only for forms auth
Comments: In standard MOSS SKU Not on by default
Performance Point
Perf Point DB
Stores session state and annotations on cells
Per farm service Not throttled
Timer Jobs: - Large job cleanup (sleep thread, one per service app) ** In M2 a thread not a timer job
Comments: Installs over Enterprise SKU Throughput goals: 30-50 RPS / analytics report with 500 users/server
BDC (LOBI)
Metadata LOBI DB
External data stores
Caches: - distributed metadata cache (DB read per startup/polling for chgs)
Comments: No scale out In WSS Not on by default
InfoPath
InfoPath Session State DB
Caches: - ASP.NET output cache
Timer Jobs: - Propagate farm-wide files to local WFE disks
Comments: Scale-out by DB Multiple DBs In Enterprise SKU Not on by default
Usage Service (MOSS Core)
Logs to
WSS Logging DB
Must run on every non-SQL Server
Timer Jobs: - import usage files
Comments: In WSS SKU Not on by default Scale out TBD
AP
Reads from
AP Staging DB
SQL Agent (import and aggregation from logging to reporting)
AP Reporting DW
Comments: Requires SQL Enterprise Enterprise SKU Not on by default
Footprint: FE Memory: High CPU: High (app srv) Size on disk: n/a DB size: N/A Scale-out unit: App R/W: 50/50
Footprint: FE Memory: High CPU: High (app srv) Size on disk: n/a DB size: N/A Scale-out unit: App R/W: 50/50
Footprint: FE Memory: High CPU: High (app srv) Size on disk: n/a DB size: N/A Scale-out unit: App R/W: 50/50
Query Footprint: FE Memory: High FE CPU: Medium Size on disk: High DB CPU: Medium Scale-out unit: Service app Throttled? No
Gather Footprint: FE Memory: High CPU: High Size on disk: Low DB size: Medium-High Scale-out: Prop/Crawl DB Throttled? Yes
PI Footprint: FE Memory: N/A CPU: High Size on disk: Med DB size: Med Scale-out unit: No Throttled? No
MySite Footprint: FE Memory: Med CPU: Med Size on disk: High DB size: High Scale-out unit: No Throttled? No
Footprint: FE Memory: Low CPU: Med Size on disk: Low DB size: Small-med Scale-out unit: Svc
Footprint: FE Memory: low CPU: low Size on disk: low DB size: low Scale-out unit: none R/W: 1:1
Footprint: FE Memory: L-Med CPU: H (App Svr) Size on disk: Low DB size: low Scale-out unit: app R/W: 100:1
Footprint: FE Memory: low CPU: very low Size on disk: low DB size: low Scale-out unit: no R/W: 100:1
Footprint: FE Memory: low CPU: low Size on disk: low DB size: low Scale-out unit: app R/W: 3:2
Footprint: FE Memory: low CPU: medium Size on disk: GBs DB size: 100 GB Scale-out unit: none RPS: 6K R/W: 1:100
Footprint: FE Memory: low CPU: medium Size on disk: GBs DB size: 100s GB Scale-out unit: app/ round robin

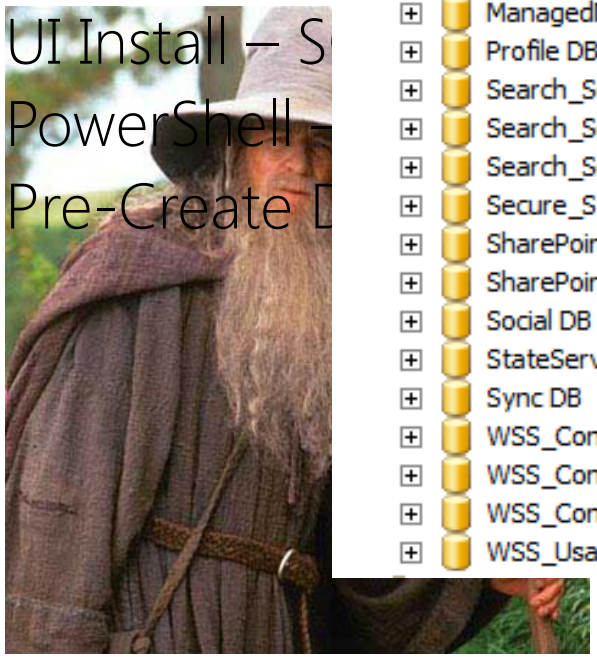
Design paradigm

- Single data platform
  - Web Content Management
    - Mostly READ
    - Structured queries and Search
  - Enterprise Content Management
    - 80/20 READ/WRITE
    - Ad-hoc queries
- Upgrade and patch management
  - Requires consistency and integrity
- Application logic expectations on schema
- Enforced integrity and constraints

Deployment

Different options for installing Share

- Basic Installation – Uses SQL Express
- Wizard Instal[illegible] – SQL Server but no
- UI Install – S
- PowerShell
- Pre-Create D



- BDC_Service_DB_275a1883-d769-44dd-ad2b-5
- ManagedMetaData
- Profile DB
- Search_Service_Application_1_CrawlStoreDB_b
- Search_Service_Application_1_DB_210e1fd479
- Search_Service_Application_1_PropertyStoreDB
- Secure_Store_Service_DB_05c8810a-d3a8-4d7
- SharePoint_AdminContent_89d89dcc-f084-42a3
- SharePoint_Config
- Social DB
- StateService_70cb25da5a34485e9f7dc8c11509
- Sync DB
- WSS_Content
- WSS_Content_8dd957c022ab478d8f832f5764f
- WSS_Content_df7005218a354ac893736689ed9
- WSS_UsageApplication

Services

Select the services you want to run in your farm. The services you select below will run with default settings on all servers in your farm.

- ☑ **Access Services 2010**
  Allows viewing, editing, and interacting with Access Services 2010 databases in a browser.
- ☑ **Access Services**
  Allows viewing, editing, and interacting with Access Services databases in a browser.
- ☑ **App Management Service**
  Allows you to add SharePoint Apps from the SharePoint Store or the App Catalog.
- ☑ **Business Data Connectivity Service**
  Enabling this service provides the SharePoint farm with the ability to upload BDC models that describe the interfaces of your enterprises' line of business systems and thereby access the data within these systems.
- ☑ **Excel Services Application**
  Allows viewing and interactivity with Excel files in a browser.
- ☐ **Lotus Notes Connector**
  Search connector to crawl the data in the Lotus Notes server.
- ☑ **Machine Translation Service**
  Performs automated machine translation.
- ☑ **Managed Metadata Service**
  This service provides access to managed taxonomy hierarchies, keywords and social tagging infrastructure as well as Content Type publishing across site collections.
- ☑ **PerformancePoint Service Application**
  Supports the monitoring and analytic capabilities of PerformancePoint Services such as the storage and publication of dashboards and related content.
- ☑ **PowerPoint Conversion Service Application**
  Enables the conversion of PowerPoint presentations to various formats.
- ☑ **Search Service Application**
  Index content and serve search queries.
- ☑ **Secure Store Service**
  Provides capability to store data (e.g. credential set) securely and associate it to a specific identity or group of identities.

SharePoint Database design

- Types of Database
  - Administration
  - Content
  - Service Applications
- How many databases will be required?
- How large do they need to be?
- How fast will they grow?

Configuring SQL Server 2012 for SharePoint (2013)

- Always install an instance of SQL Server that will be dedicated to SharePoint
- SQL instance for SharePoint should be a non-default instance
- Make sure you use a SQL alias
- Max. Degrees of Parallelism must be set to 1
- For all SharePoint DB's, set the Default Collation setting to: Latin1_General_CI_AS_KS_WS

System Database Modifications that must be performed

- TempDB
  - Increase its initial size. Try starting with 500 MB
  - Set Autogrowth to use MB, not %
  - Autogrowth value should be larger, not smaller. Something like 500 MB should be a good starting value.
  - Put on the fastest available non-system drive. Separate the files – one per core on the server (max 8 files)
- ModelDB
  - Initial size should be set to 500 MB as a good start
  - Set to full recovery mode
  - Set Autogrowth to use MB, not %
  - Autogrowth should be set to something like 250 MB
  - Initial log size should be 25% of MDF
  - Log growth should set to something like 70 MB

Content Databases

- Each web application has at least on site collection
- Each site collection is contained in a single c… tabase
- Each content database can conta… te colle…
- If site collecti… GB
  - M… own con…
  - …x size of … collection = 200GB
- Si… ends on the amount of content
- Configure temp DB to be at least 10% of total Content DB size

Database size = ((D × V) × S) + (10 KB × (L + (V × D)))

Database size = (((200,000 x 2)) x 250) + ((10 KB × (600,000 + (200,000 x 2))) = 110,000,000 KB

Management Databases

Name	Size	Growth	Read/ Write	Scaling	Recovery Model
SharePoint_Config	Small	Slow	90/10	Up	Full
SharePoint_Admin_Content	Small	Slow	90/10	Up	Full

Search Databases

Name	Size	Growth	Read/ Write	Scaling	Recovery Model
Search_Service_Application_DB_<Servername>	10GB	Many factors determine growth	80/20	Mostly Up	Simple
Search_Service_Application_AnalyticsReportingStoreDB_<Servername>	Medium to Large	Many factors determine growth	90/10	Up	Simple
Search_Service_Application_CrawlStoreDB_<Servername>	Medium	Depends on the amount of content	90/10	Out	Simple
Search_Service_Application_LinkStoreDB_<Servername>	Medium to Large	Depends on multiple factors	90/10	Out	Simple

User Profile Databases

Name	Size	Growth	Read/ Write	Scaling	Recovery Model
Application_ProfileDB	1MB per profile	Multiple factors	90/10	Up per Service App	Simple
Application_SyncDB	630KB per user	Multiple factors	50/50	Up per Service App	Simple
Application_SocialDB	0.009MB per tag	Depends on the amount of activity	80/20	Up per Service App	Simple

Other Databases

Name	Size	Growth	Read/ Write	Scaling	Recovery Model
Secure_Store_Service_DB	5MB per 1000 credentials	Slow, depends on auditing	80/20	Up	Full
SessionStateService	1GB	Depends on usage of other services	80/20	Out	Full
Bdc_Service_DB	Small	Very Slow	90/10	Up	Full
PerformancePoint Service	1GB	Very slow	80/20	Up per Service App	Full

Sizing and Architecture

SQL Server Best Practices for SharePoint

Configuration : Typical Deployment Sizes

Metric	Small	Medium	Large
Content db size	< 50GB	50GB	> 50GB
# of Content dbs	< 20	20	> 20
# of concurrent requests to SQL	< 200	200	> 200
User	< 1000	1000	> 1000
# of items in regularly accessed list	< 2000	2000	> 2000
# of columns in regularly accessed list	< 20	20	> 20

Configuration : Recommended (Minimum) Capacities

Resource	Small	Medium	Large
Minimum DB server memory	4 GB	8 GB	16 GB
Processor L2 cache	2 MB	> 2 MB	> 2MB
Bus bandwidth	Medium	High	High
Disks latencies (msec)	< 20	< 10	< 10 (data) < 5 (T-log)
Network	Gigabit	Gigabit	Gigabit
Network latency (msec)	< 1	< 1	< 1

Configuration : Best practices

- Install latest Service Packs & CUs
- Use dedicated SQL Server for medium/large deployments
- Consider scaling-out Content DBs as deployment grows beyond 4-8 TB (depending on usage model)
  - Some SharePoint databases must be scaled up, example: Config DB
- Use connection alias
  - Simplifies redirecting WFEs to a different database instance

Software Boundaries

- "Hard" Limits…..there are none !
- "Soft Limits"
  - 8 WFEs to 1 SQL Server instance
  - 50.000 site collections per content database
  - 100 GB of data per content database
- Latency < 1ms between Web and database servers

Prioritizing Database Volume

- Recommended database placement priority (fastest to slowest drive)
  1. Temp DB data and t-log files
  2. DB t-log files
  3. Search DB data files
  4. Content DB data files
  5. Profile DB data files
- Place temp DB, Content DB and t-logs on separate LUNs
- Use multiple data files for Content and Search DBs
- Place SharePoint Search crawl & query processing tables on separate spindles

SQL Server TempDB data files

- Data files allocated = number core CPUs in SQL Server (up to 8)
- Data file sizes consistent across all data files
- Data files spread across unique LUNs
  - Separated from Content DB, Search DB, etc.
- Log file separated to unique LUN

SQL Server TempDB data files - continued

- Optimal TempDB data file size formula:

  [MAX DB SIZE (KB)] X [.25] / [# CORES] = DATA FILE SIZE (KB)

- Result (starting size) should be roughly equal to 25% largest content or search DB or 10% of total content DB size
- Use RAID 10; separate LUN from other database objects (content, search, etc.)
- "Autogrow" feature set to a fixed amount; if auto grow occurs, permanently increase size

SQL Server memory

- Set 'Max Server Memory'

SQL Max Memory = TotalPhyMem
*- (NumOfSQLThreads * ThreadStackSize)*
*- (1GB * CEILING(NumOfCores/4))*

NumOfSQLThreads = 256 + (NumOfProcessors- 4) * 8*
ThreadStackSize = 1 MB on x86
2 MB on 64-bit (x64)
4 MB on 64-bit (IA64)

- Or the easy way: leave 2-3GB for the OS

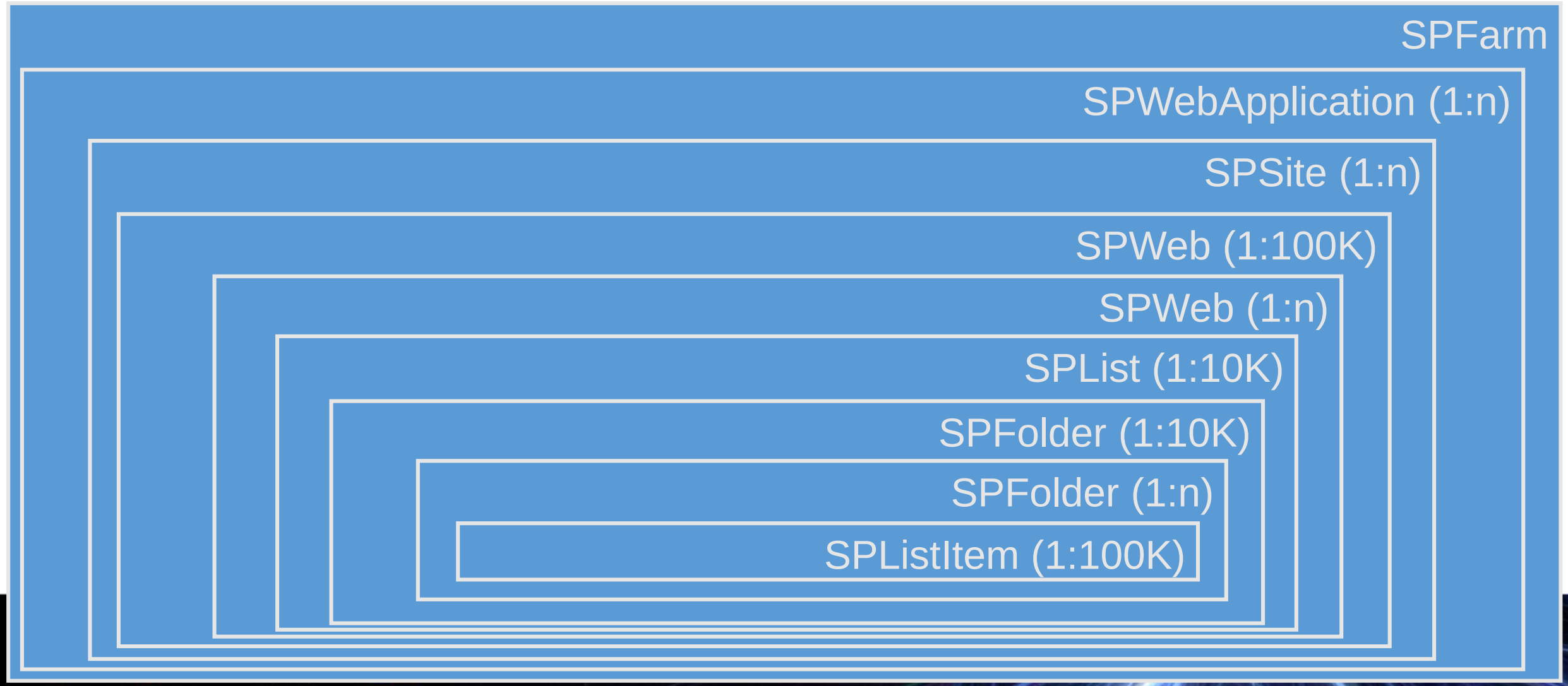
Storage – Recommended I/O Capacities

- Disk/Sec Transfer
  - Data Files > 10 ms
  - Transaction Logs > 5 ms

Type	RAID level	IOPS	SAN Optimization
tempdb	RAID-10	2 IOPS/GB	Write optimized
Transaction Logs	RAID-10	2 IOPS/GB	Write optimized
Search Database	RAID-10	2 IOPS/GB	Read/Write optimized
Content Databases	RAID-10*	0.75 IOPS/GB	Read optimized

* *Raid-5 can be used for static web content*

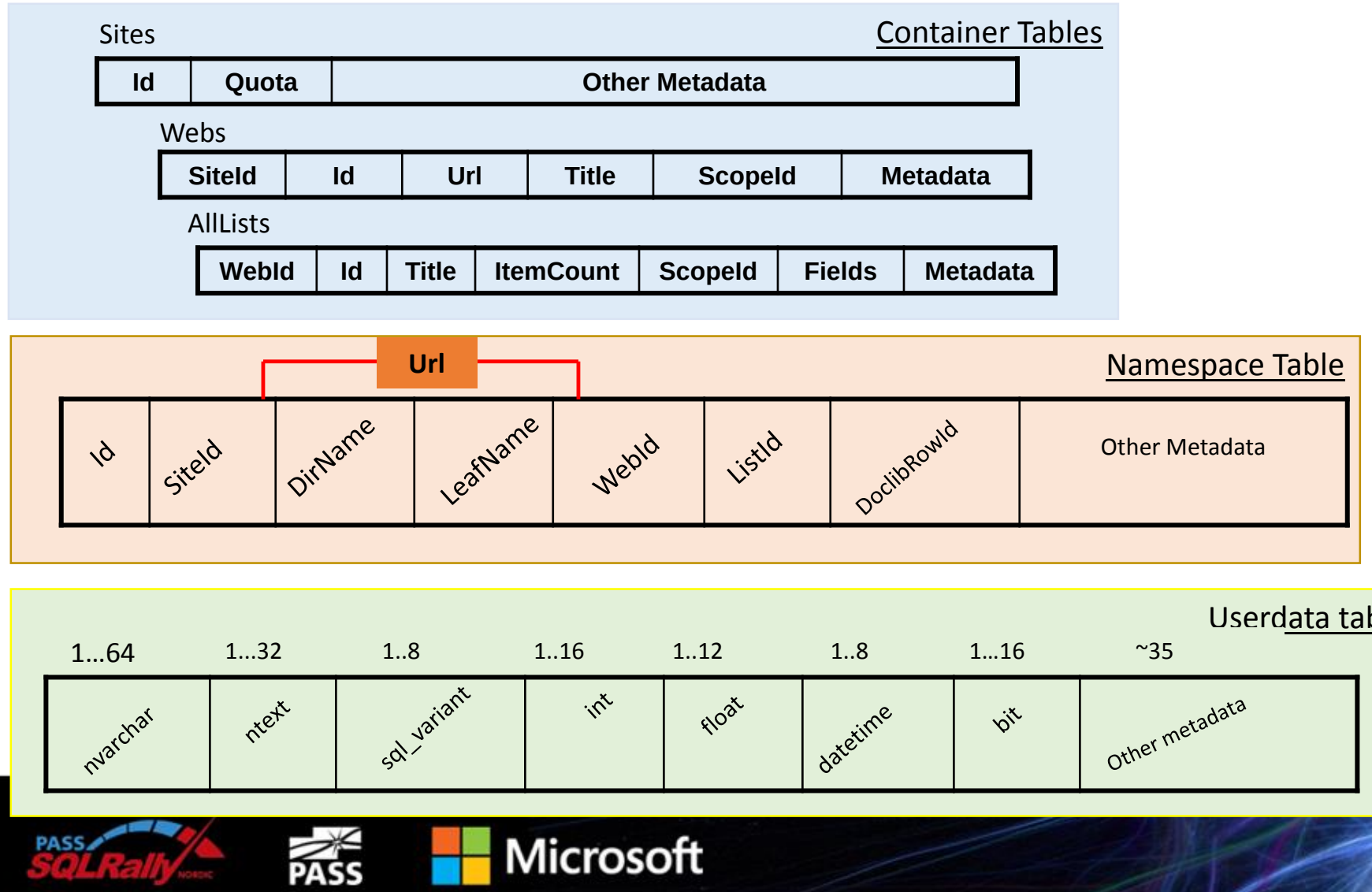
SharePoint container Topology



SharePoint Data Architecture

- Flexible, user extensible types
  - Announcement, Contacts, Document Types
  - Support 10th of a million types in a single DB
  - A few types that may have 100's of properties
- Millions of instances of multiple types in a list
- Efficient display of "all items in a folder"
- End-user queries over multiple lists in multiple sites which is mapped onto a single table

Content DB Architecture



Schema implications (Perce

- SharePoint tables are too wide,
- SharePoint manages its own inc
- SharePoint adds force-order, qu
- Missing indexes for common op
- Excessive use of Dynamic queri
- No SQL Referential Integrity OR
- DBCC with data loss
- Use of @table variables
- Lack of consistency checker
- DB Connect failures
- Missing integration of Back-up/

Column Name	Data Type	Allow Nulls
tp_ID	int	☐
tp_ListId	uniqueidentifier	☐
tp_SiteId	uniqueidentifier	☐
tp_RowOrdinal	int	☐
tp_Version	int	☐
tp_Author	int	☑
tp_Editor	int	☑
tp_Modified	datetime	☑
tp_Created	datetime	☑
tp_Ordering	varchar(512)	☑
tp_ThreadIndex	varbinary(512)	☑
tp_HasAttachment	bit	☐
tp_ModerationStatus	int	☐
tp_IsCurrent	bit	☐
tp_ItemOrder	float	☑
tp_InstanceID	int	☑
tp_GUID	uniqueidentifier	☐
tp_CopySource	nvarchar(260)	☑
tp_HasCopyDestinations	bit	☑
tp_AuditFlags	int	☑
tp_InheritAuditFlags	int	☑
tp_Size	int	☐
tp_WorkflowVersion	int	☑
tp_WorkflowInstanceID	uniqueidentifier	☑
tp_DirName	nvarchar(256)	☐
tp_LeafName	nvarchar(128)	☐
tp_DeleteTransactionId	varbinary(16)	☐
tp_ContentType	nvarchar(255)	☑

Column Name	Data Type	Allow Nulls
tp_ContentTypeId	tContentTypeId:...	☑
nvarchar1	nvarchar(255)	☑
nvarchar2	nvarchar(255)	☑
nvarchar3	nvarchar(255)	☑
nvarchar4	nvarchar(255)	☑
nvarchar5	nvarchar(255)	☑
nvarchar6	nvarchar(255)	☑
nvarchar7	nvarchar(255)	☑
nvarchar8	nvarchar(255)	☑
ntext1	ntext	☑
ntext2	ntext	☑
ntext3	ntext	☑
ntext4	ntext	☑
sql_variant1	sql_variant	☑
nvarchar9	nvarchar(255)	☑
nvarchar10	nvarchar(255)	☑
nvarchar11	nvarchar(255)	☑
nvarchar12	nvarchar(255)	☑
nvarchar13	nvarchar(255)	☑
nvarchar14	nvarchar(255)	☑
nvarchar15	nvarchar(255)	☑
nvarchar16	nvarchar(255)	☑
ntext5	ntext	☑
ntext6	ntext	☑
ntext7	ntext	☑
ntext8	ntext	☑
sql_variant2	sql_variant	☑
nvarchar17	nvarchar(255)	☑

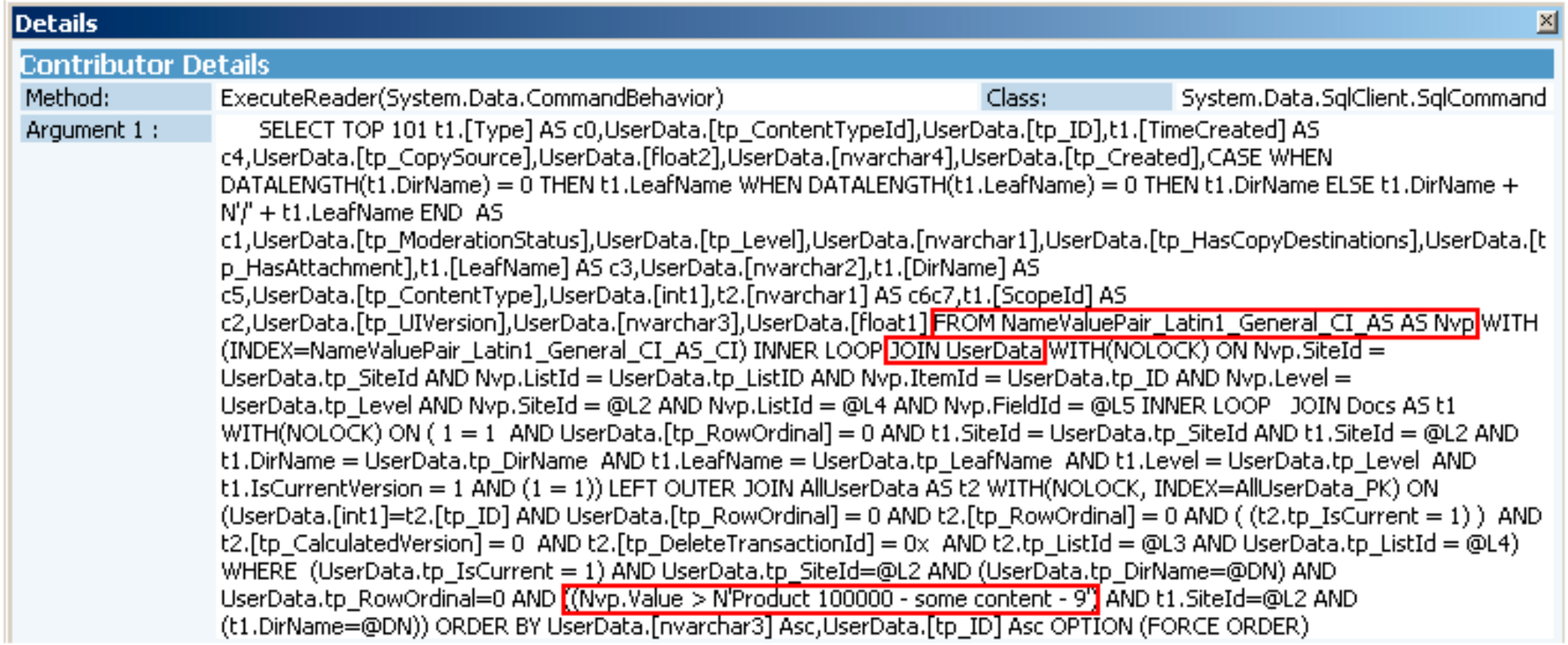
SharePoint Schema

- Demo !

SharePoint maintains its own index

- Multiple types in the same table = untenable SQL Indexing
- Design challenge:
  - How do I put a SQL Index for a given property in all instances of a given type?
  - Do you really suggest 1000+ of index on a table ?
- Solution
  - Maintain Name-Value pairs and index NVP table

SharePoint querying indexed lists



Details

Contributor Details

Method:	ExecuteReader(System.Data.CommandBehavior)	Class:	System.Data.SqlClient.SqlCommand

Argument 1 :

```
SELECT TOP 101 t1.[Type] AS c0,UserData.[tp_ContentTypeId],UserData.[tp_ID],t1.[TimeCreated] AS
c4,UserData.[tp_CopySource],UserData.[float2],UserData.[nvarchar4],UserData.[tp_Created],CASE WHEN
DATALENGTH(t1.DirName) = 0 THEN t1.LeafName WHEN DATALENGTH(t1.LeafName) = 0 THEN t1.DirName ELSE t1.DirName +
N'/' + t1.LeafName END  AS
c1,UserData.[tp_ModerationStatus],UserData.[tp_Level],UserData.[nvarchar1],UserData.[tp_HasCopyDestinations],UserData.[t
p_HasAttachment],t1.[LeafName] AS c3,UserData.[nvarchar2],t1.[DirName] AS
c5,UserData.[tp_ContentType],UserData.[int1],t2.[nvarchar1] AS c6c7,t1.[ScopeId] AS
c2,UserData.[tp_UIVersion],UserData.[nvarchar3],UserData.[float1] FROM NameValuePair_Latin1_General_CI_AS AS Nvp WITH
(INDEX=NameValuePair_Latin1_General_CI_AS_CI) INNER LOOP JOIN UserData WITH(NOLOCK) ON Nvp.SiteId =
UserData.tp_SiteId AND Nvp.ListId = UserData.tp_ListID AND Nvp.ItemId = UserData.tp_ID AND Nvp.Level =
UserData.tp_Level AND Nvp.SiteId = @L2 AND Nvp.ListId = @L4 AND Nvp.FieldId = @L5 INNER LOOP  JOIN Docs AS t1
WITH(NOLOCK) ON ( 1 = 1  AND UserData.[tp_RowOrdinal] = 0 AND t1.SiteId = UserData.tp_SiteId AND t1.SiteId = @L2 AND
t1.DirName = UserData.tp_DirName  AND t1.LeafName = UserData.tp_LeafName  AND t1.Level = UserData.tp_Level  AND
t1.IsCurrentVersion = 1 AND (1 = 1)) LEFT OUTER JOIN AllUserData AS t2 WITH(NOLOCK, INDEX=AllUserData_PK) ON
(UserData.[int1]=t2.[tp_ID] AND UserData.[tp_RowOrdinal] = 0 AND t2.[tp_RowOrdinal] = 0 AND ( (t2.tp_IsCurrent = 1) )  AND
t2.[tp_CalculatedVersion] = 0  AND t2.[tp_DeleteTransactionId] = 0x  AND t2.tp_ListId = @L3 AND UserData.tp_ListId = @L4)
WHERE  (UserData.tp_IsCurrent = 1) AND UserData.tp_SiteId=@L2 AND (UserData.tp_DirName=@DN) AND
UserData.tp_RowOrdinal=0 AND ((Nvp.Value > N'Product 100000 - some content - 9') AND t1.SiteId=@L2 AND
(t1.DirName=@DN)) ORDER BY UserData.[nvarchar3] Asc,UserData.[tp_ID] Asc OPTION (FORCE ORDER)
```

Recap

- We have
  - Wide table with no type/app awareness
  - Row wrapping = multi-row objects
  - Name-Value auxiliary table providing app-level index
- Result
  - SQL Server knows very little about SharePoint App semantics !
  - Can't afford to rely on QO/QP to do it right
    - Query over little list followed by a large library
    - Query that join from NVP index to content table

Maintenance

SQL Server Best Practices for SharePoint

SharePoint DB Health monitoring

- Monitor SQL Server performance regularly
  - Use SQL Server DMVs
  - Use Recommended Perfmon counters
  - Allocate extra disk space for diagnostics information
- Check integrity of the database routinely
- DBCC CHECKDB
  - Can use REPAIR_REBUILD option to fix errors (not always possible)
  - REPAIR_ALLOW_DATA_LOSS not supported
  - Time consuming operation, run during non-peak hours

SharePoint DB maintenance

- Does SharePoint maintain indexes ?
- Use DBCC CheckDB (~~REPAIR_ALLOW_DATA_LOSS~~)
- Set of Rules:
  - Databases used by SharePoint have fragmented indices
  - Search – One or more property databases have fragmented indices.
  - Search - One or more crawl databases may have fragmented indices
- Define a maintenance plan

SharePoint Databases maintenance considerations

- Fragmentation occurs by design on SharePoint ;-)
- Increase space utilization & I/O → degrades performance
- Content and Search dbs most susceptible
- Rebuild / Reorganize indexes to eliminate fragmentation
- Use sys.dm_db_index_physical_stats to measure
  - More accurate than DBCC SHOWCONTIG, often reports higher fragmentation numbers
- Use a framework like Ola Halengren's

SharePoint Databases maintenance considerations

- Do's
  - Auto-defrag only available for content databases
  - Only shrink content databases, not others
  - Only perform if free space > 50% (after content reorg)
  - Do not perform as part of maintenance plan
  - Perform during off-peak hours (resource intensive)
  - Update statistics – don't rely on the timer service
  - Use DBCC SHRINKDATABASE or DBCC SHRINKFILE
  - Have reliable backups for all databases before implementing maintenance operations
  - Check for and repair consistency errors by using DBCC CHECKDB
  - Change the server-wide fill factor setting to 70

SharePoint Databases maintenance considerations

- Don'ts
  - Drop and re-create indexes
  - Rebuild indexes or run consistency checks during business hours
  - Set fill factor for individual tables or indexes
  - Shrink any databases other than content databases
  - Auto-shrink databases
  - Shrink databases at all unless you really need to

Schema modifications constraints

- Adding database triggers
- Adding new indexes or changing existing indexes within tables
- Adding, changing, or deleting any primary or foreign key relationships
- Changing or deleting existing stored procedures
- Calling existing stored procedures directly, except as described in the SharePoint Protocols documentation
- Adding new stored procedures
- Adding, changing, or deleting any data in any table of any of the databases for the products that are listed in the "Applies to" section
- Adding, changing, or deleting any columns in any table of any of the databases for the products that are listed in the "Applies to" section
- Making any modification to the database schema
- Adding tables to any of the databases for the products that are listed in the "Applies to" section
- Changing the database collation
- Running DBCC_CHECKDB WITH REPAIR_ALLOW_DATA_LOSS (However, running DBCC_CHECKDB WITH REPAIR_FAST and REPAIR_REBUILD is supported, as these commands only update the indexes of the associated database.)
- Enabling SQL Server change data capture (CDC)
- Enabling SQL Server transactional replication
- Enabling SQL Server merge replication

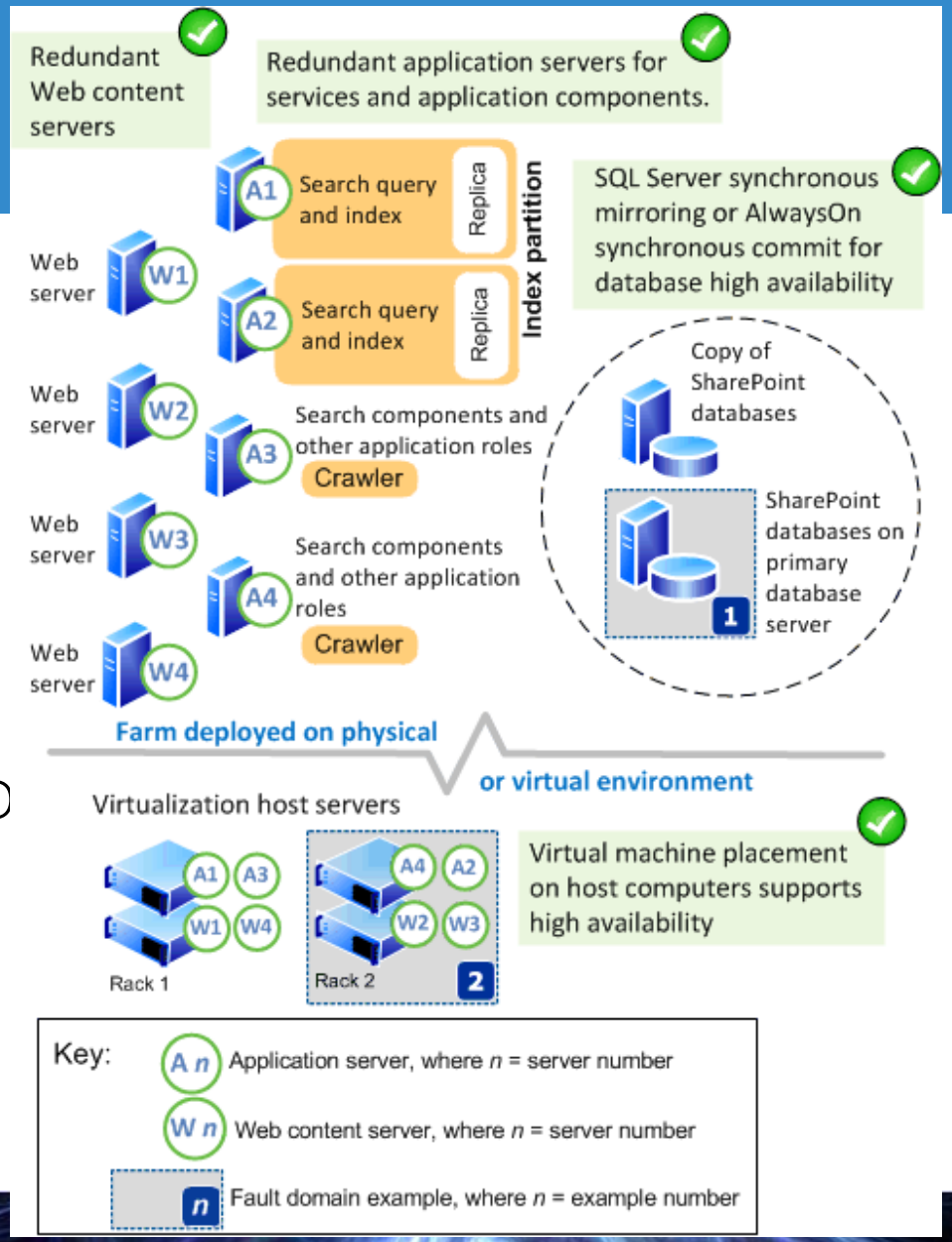
High Availability

SQL Server Best Practices for SharePoint

High Availability

- High availability options supported:
  - Database mirroring
  - Failover Clustering (local & stretch clusters)
  - Log Shipping
  - Always On

- Database Mirroring is by far the most po
  Always On is catching up !
- Virtualization of hosts servers



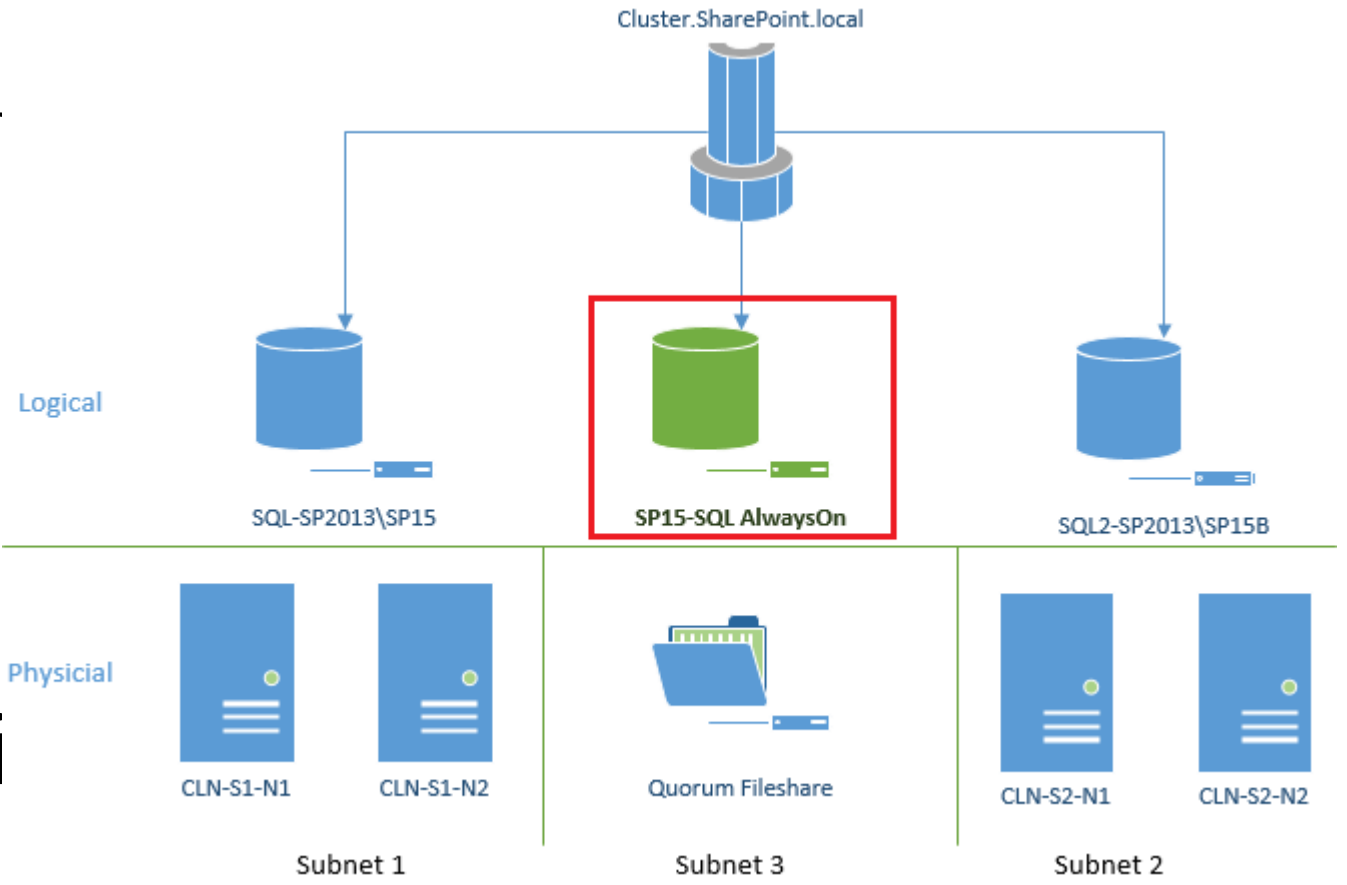
SharePoint 2013 and Always On

Cluster of clusters

- Shared disks rather th
- Instance redundancy
- Resist to up to 3 mac

Cluster with 2 nodes

- Single node failure wi

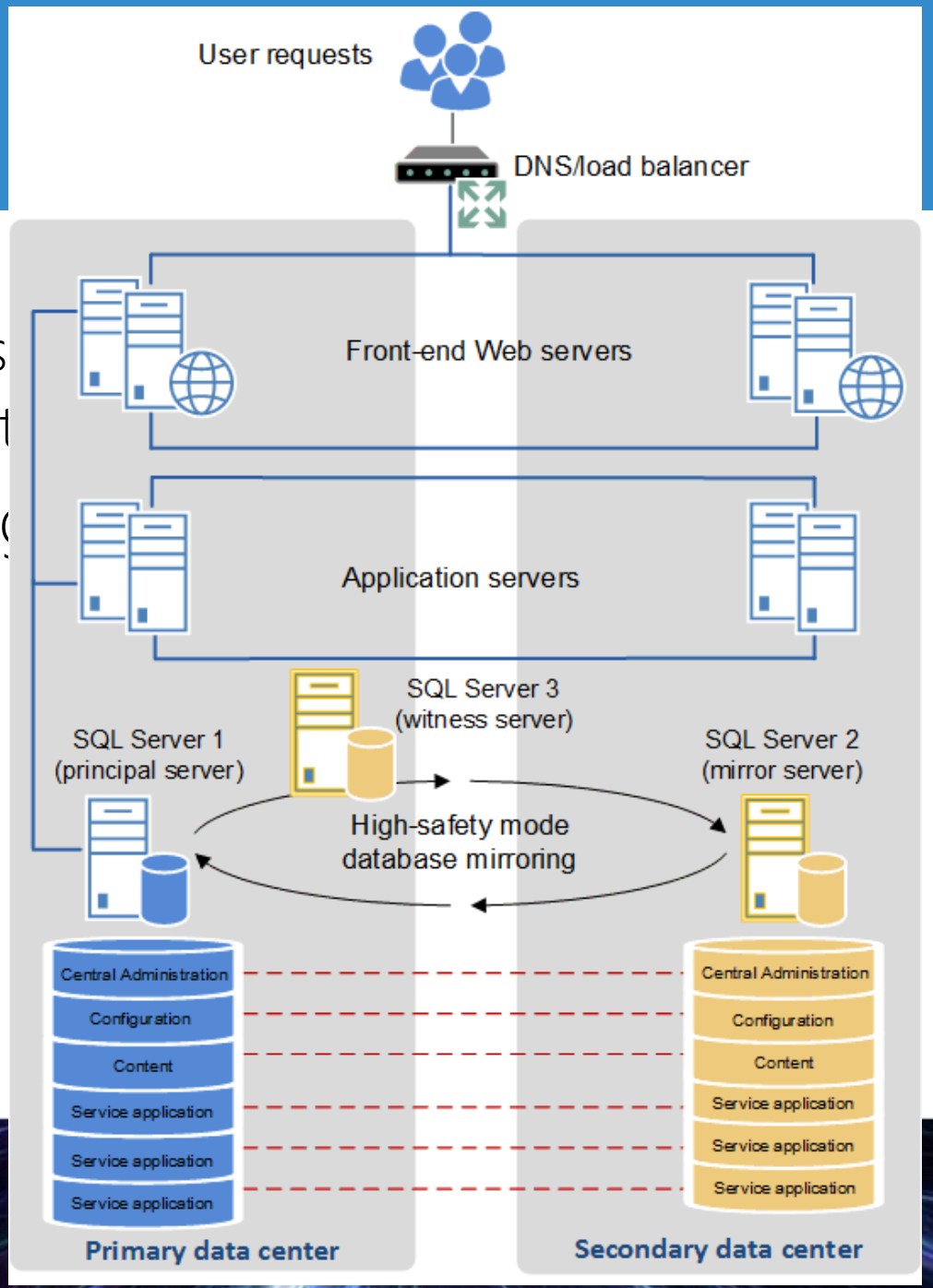


Pros And Cons -RPO and RTO comparison based on database technology

SQL Server solution	Potential data loss (RPO)	Potential recovery time (RTO)	Automatic failover	Readable secondaries (Not supported by SharePoint)
AlwaysOn Availability Group (synchronous-commit)	Zero	Seconds	Yes	0 - 2
AlwaysOn Availability Group (asynchronous-commit)	Seconds	Minutes	No	0 - 4
AlwaysOn Failover Cluster Instance	Does not apply An FCI itself does not provide data protection. The amount of data loss depends on the storage system implementation.	Seconds to minutes	Yes	Does not apply
Database mirroring - High-safety (synchronous mode + witness server)	Zero	Seconds	Yes	Does not apply
Database mirroring - High-performance (asynchronous mode)	Seconds	Minutes	No	Does not apply
Backup, copy, restore	Hours or zero if the tail of the log can be accessed after the failure.	Hours to days	No	Not during a restore

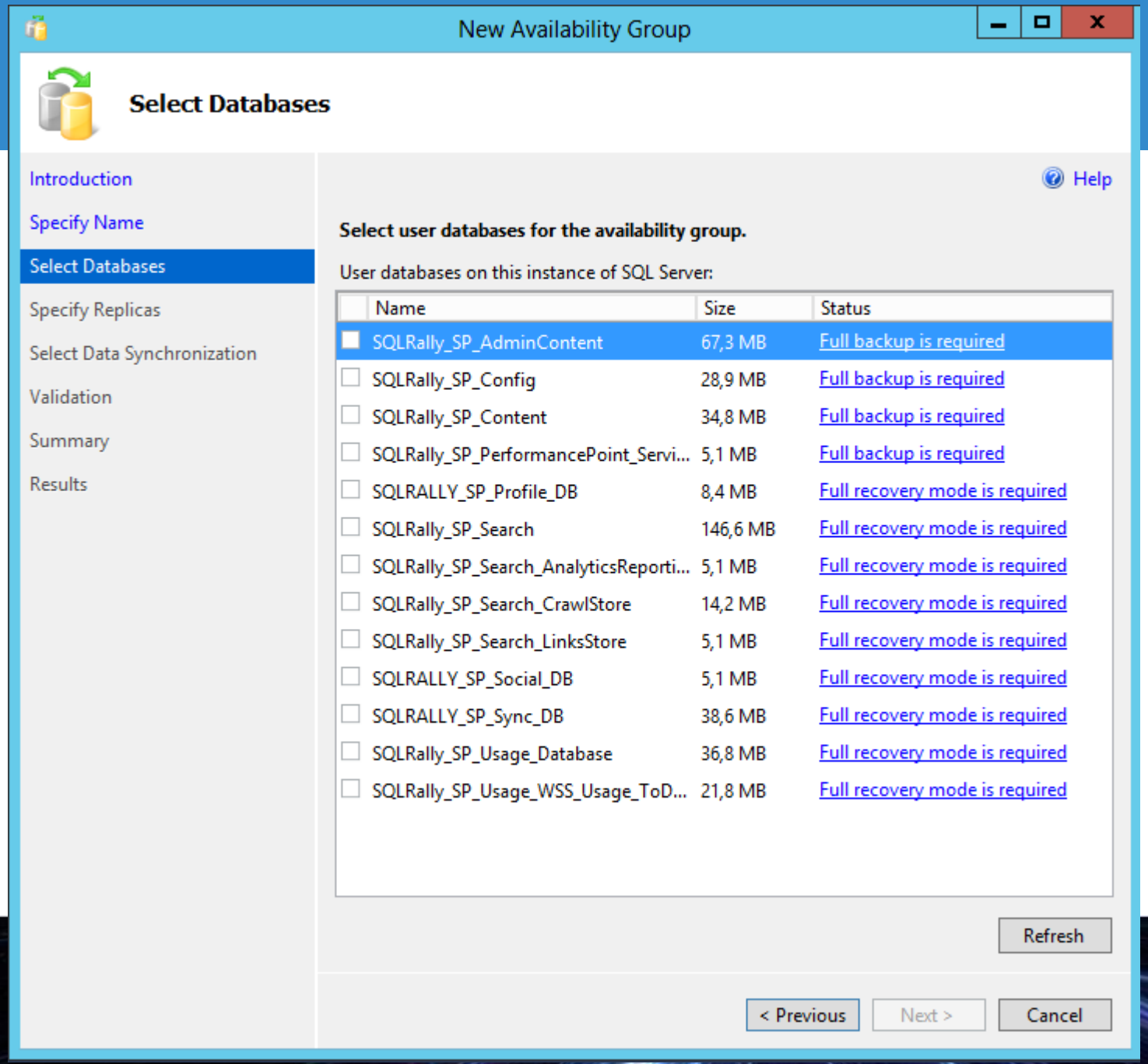
Stretched Farm

- Requires:
  - Highly consistent intra-farm latency of <1ms
  - 99.9% of the time over a period of ten minut
- The bandwidth speed must be at least 1 g



Always On

- Demo !



Tack Så Mycket

Régis Baccaro

@regbac

Th

Régis Baccaro

@regbac

regis@baccaro.com

http://Theblobfarm.wordpress.com

http://Thelovefarm.wordpress.com

- Founder and lead organizer of SQL Saturday Denmark
- Works for IBM
- Passionate about the community
- .Net developer, BI guy, SharePoint fellow and accidental DBA