

# **Windows 8 : Storage Provisioning and Management**

**Shiv Rajpal**  
**Principal Development Lead**  
**Microsoft**

## □ Agenda

### □ Storage Provisioning

- Storage Pools
- Storage Spaces

### □ Storage Management

- SMP Architecture
- Tools – UI, Power Shell, WMI
- Providers – SMP, SMI-S

# Cost-Effective Business Critical Storage

## Storage Spaces

- Powerful new platform abstractions
  - Storage Pools: ***are units of administration, capacity aggregation, delegated administration and isolation in a multi-tenant deployment.***
  - Storage Spaces: ***Spaces are virtual disks delivering optimized (just-in-time) allocation, resiliency to component failures, high-performance, and scale.***
- Target design point
  - Industry standard interconnects: SATA or (Shared) SAS
  - Industry standard storage: JBODs

*Storage Spaces enable partners to deliver a new category of highly capable storage solutions to all Windows customer segments at dramatically lower price-points & maximized operational simplicity*

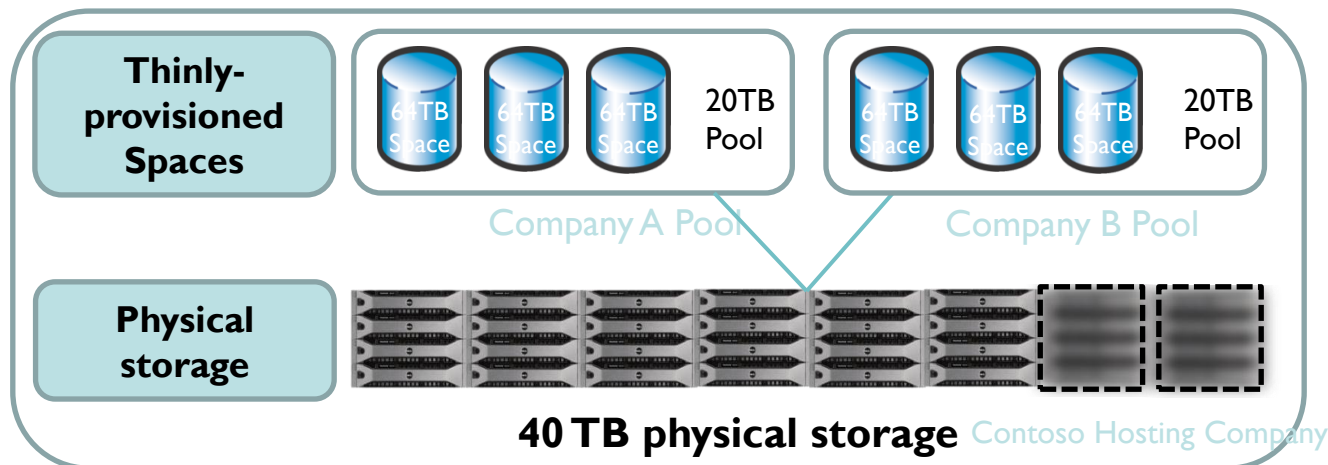
# Storage Spaces Capabilities

- ❑ Optimized storage utilization
  - ❑ Thin (and Fixed) Provisioning Support
  - ❑ Storage optimizer assists in reclaiming slabs
- ❑ Resiliency & application-driven error correction
  - ❑ Mirror and Parity Spaces
  - ❑ Storage optimizer driven error correction
- ❑ High-availability & scale-out with Failover Clustering & Cluster Shared Volumes

# Storage Spaces Capabilities

- ❑ Operational simplicity
  - ❑ Delegated administration & isolation for multi-tenant deployments
  - ❑ Management through new Windows 8 management architecture
- ❑ Workload optimized performance via heterogeneous media support

- On-demand provisioning with trim support
  - For use by NTFS, Hyper-V, and other applications e.g. SQL
- Elastic capacity expansion
  - Simply add more physical drives



- ❑ Redundant storage
  - ❑ Mirrored Spaces and Parity Spaces with integrated journaling
- ❑ Rapid Recovery - per-pool hot-spare support
- ❑ Application driven intelligent error correction



**Resilient Spaces  
contained within  
a Pool**



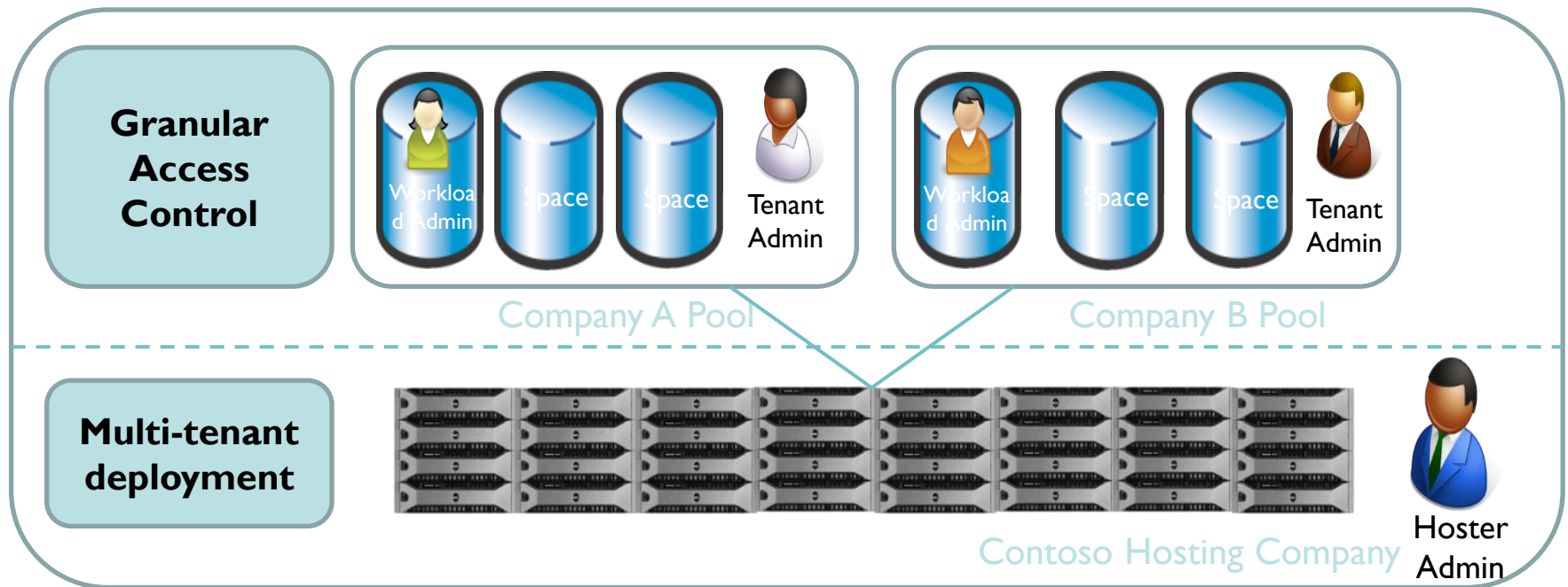
**Industry standard  
physical storage**



# Storage Spaces

## Operational simplicity

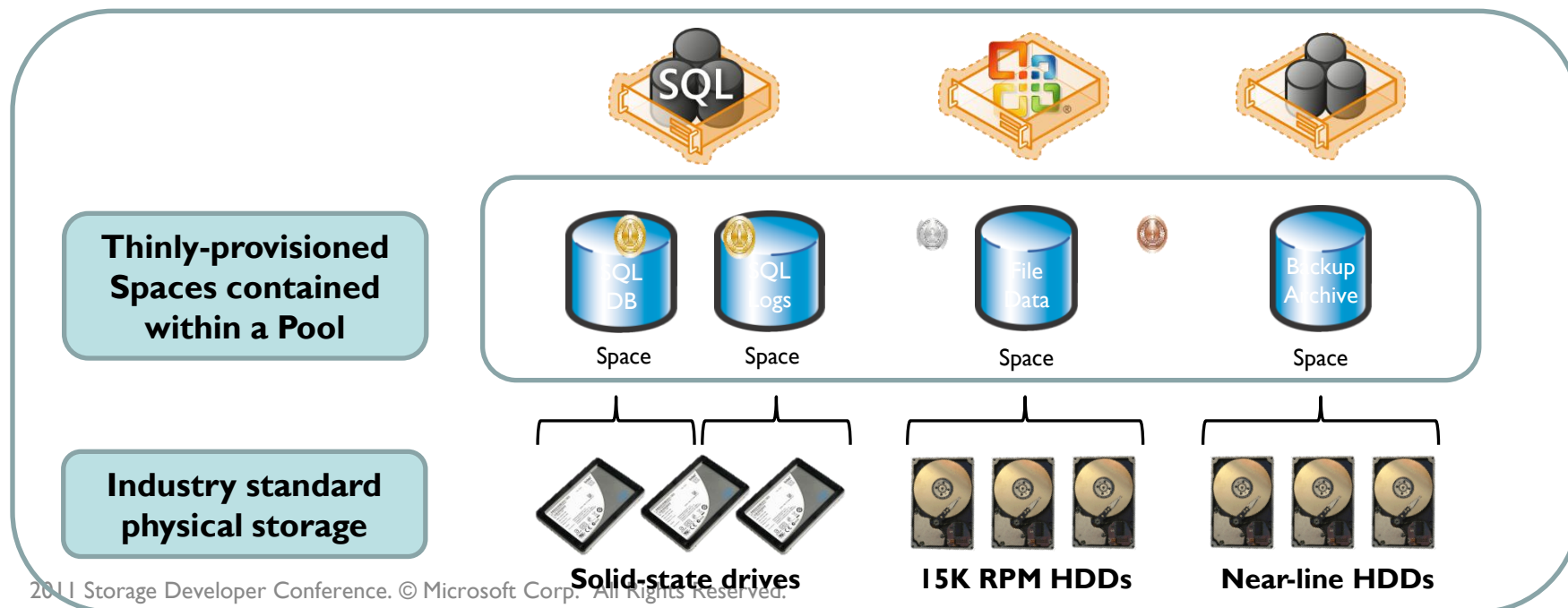
- Managed via Windows Storage Management API
- Flexible & granular administrative control with self-service agility & tenant isolation



# Storage Spaces

## Effectively utilize hardware investment

- Supports deployment requirements
  - Utilize both SSD and HDD media types
  - Control placement at Space granularity



## Physical deployment topologies

*Direct-Attached*



SATA or SAS



OR

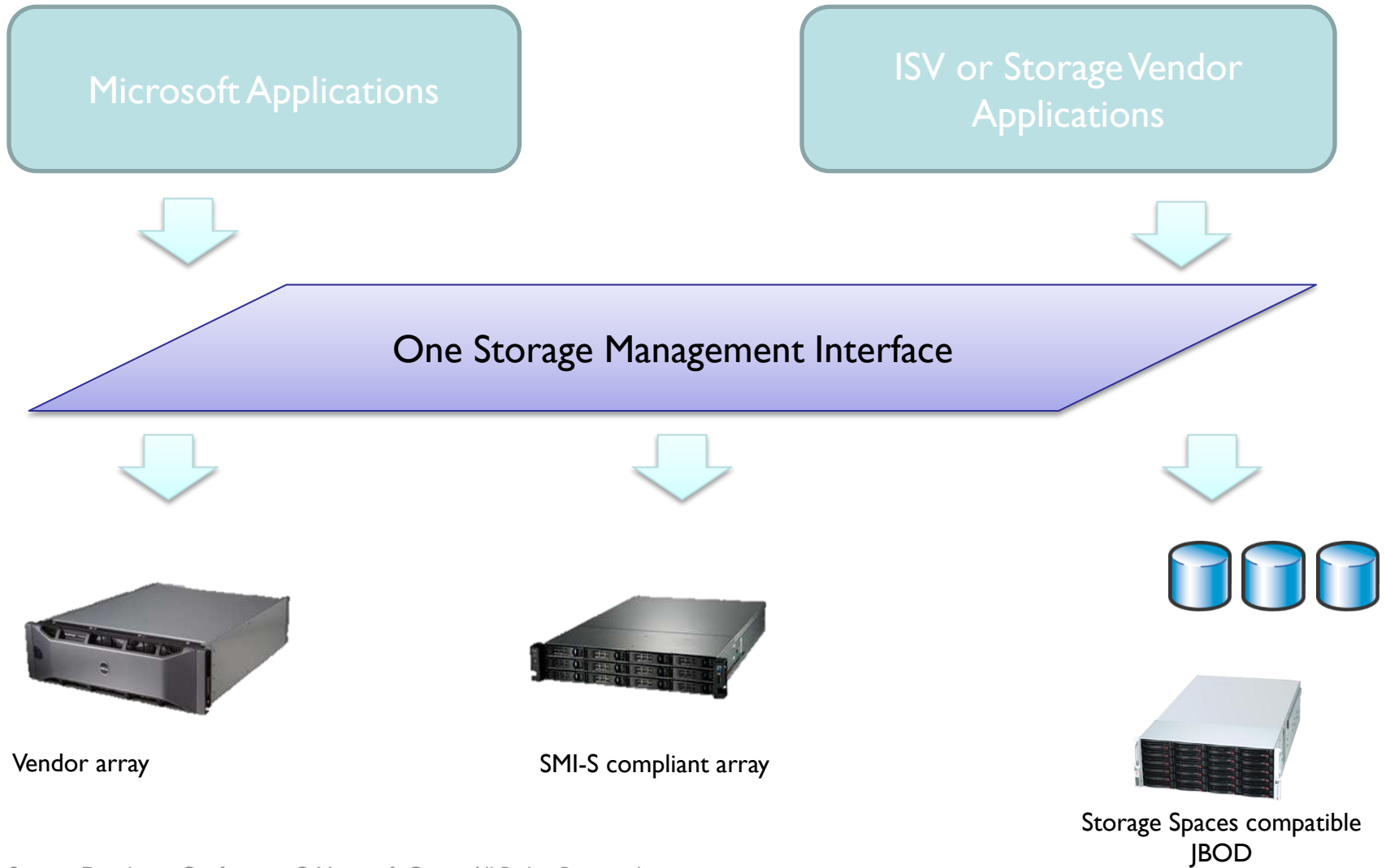
*Shared – Availability & Scale*



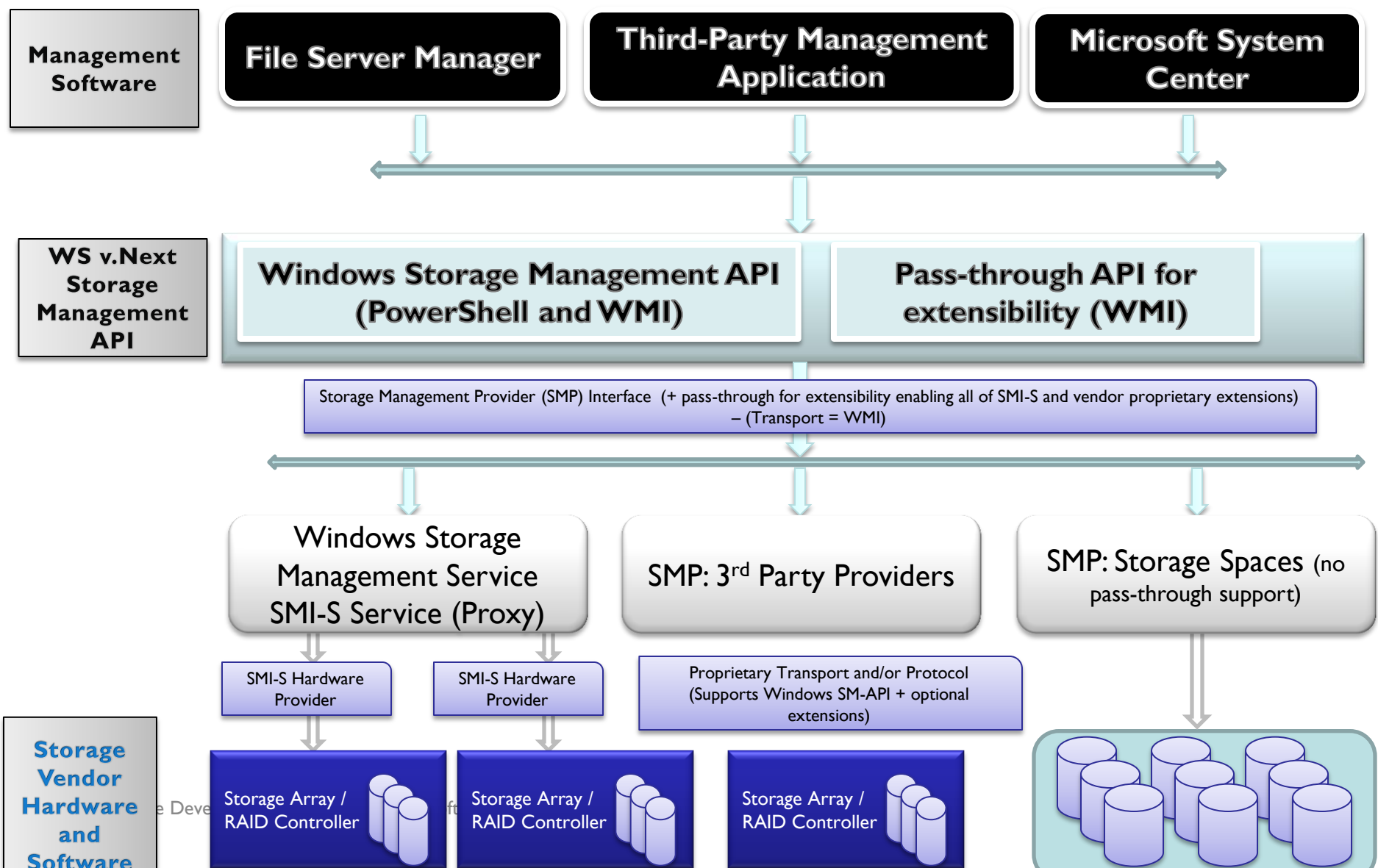
Shared SAS



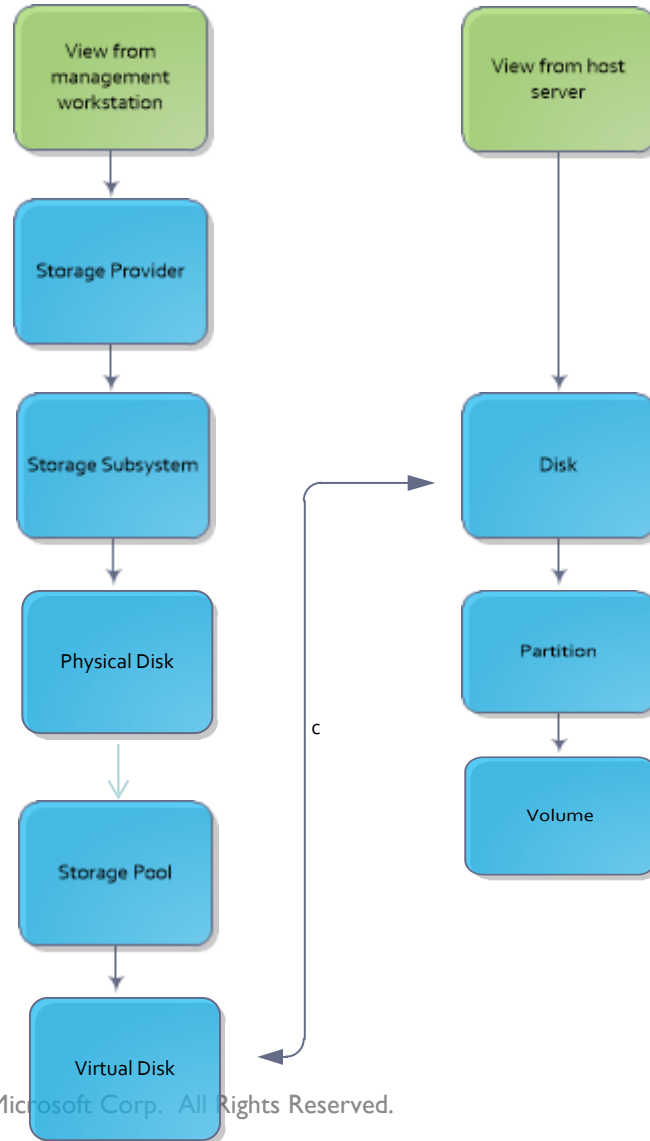
# Operational Simplicity Comprehensive Storage Management



# Storage Management Architecture



# Storage Management



- ❑ Discovery
  - ❑ Supporting different Levels of discovery
- ❑ Discovering Sub System Capabilities in advance
  - ❑ Pooling Capability
  - ❑ Replication/Cloning
  - ❑ Offload Data Transfer (ODX)
  - ❑ Masking Sets

- ❑ Pool Management
  - ❑ Creation, Deletion
  - ❑ Setting properties on Pool (Example - Pool usage type, Thin Provision thresholds)
  - ❑ Creating VirtualDisk
  - ❑ Multiple Resiliency Scheme
  - ❑ Multiple Provisioning Capability
  - ❑ Capacity utilization

- ❑ VirtualDisk Management
  - ❑ Creation, Deletion
  - ❑ Setting properties
  - ❑ Health and Operational Status
  - ❑ Space Management:
    - ❑ Attach/Detach
    - ❑ Automatic / Manual allocation selection
    - ❑ Clustered Spaces

- ❑ Target/Initiator Ports
  - ❑ Masking Sets
- ❑ OS Disk Management
  - ❑ Partitioning
  - ❑ Formatting
  - ❑ Disk refresh/Bus Rescan
  - ❑ Repair Volume

- ❑ Indications
  - ❑ Thin Provisioning Threshold crossed
  - ❑ Pool/VirtualDisk health change notifications
- ❑ Associations. (Examples)
  - ❑ VirtualDisk<->OS disk
  - ❑ Disks<->InitiatorPorts
  - ❑ VirtualDisk<->VirtualDisk
  - ❑ VirtualDisk<->PhysicalDisk

- Support for Pass Through
  - Contract between Application and Provider
  - Microsoft Applications will publish classes which will be used through Pass through

- ❑ File Server role UI
- ❑ Comprehensive set of Power Shell cmdlets
  - ❑ PowerShell provides a uniform, scriptable management experience covering array provisioning, deployment of storage, and host management and provisioning
- ❑ Comprehensive set of WMI interfaces

# Storage Management – File Server UI

Server Manager

SERVER MANAGER ▸ FILE SERVICES ▸ POOLS

Manage View Help

SERVERS

**POOLS**

VOLUMES

SHARES

iSCSI VIRTUAL DISKS

**POOLS**  
Storage Pools | 2 total

Filter  Queries Save

Health	Name	Server	Type	Capacity	Free Space	%Used
Storage Spaces on DeepSpaceNine (2)						
	DeepSpacePool	DeepSpaceNine	Storage Pool	3.64 TB	3.35 TB	<div style="width: 92%;"></div>
	Primordial	DeepSpaceNine	Available Disks	0.00 B	0.00 B	<div style="width: 0%;"></div>

**VIRTUAL DISKS**  
DeepSpacePool on DeepSpaceNine

Filter  Queries Save

Health	Name	Status	Redundancy	Provisioning	Type	Volume
DeepSpaceNine (1)						
	FileShares		Mirror	Thin		F:

[Go to Volumes >>](#)

**PHYSICAL DISKS**  
DeepSpacePool on DeepSpaceNine

Filter  Queries Save

Health	Name	Status	Capacity	Spindle Speed	Bus Type	Slot Number
	PhysicalDisk2		466 GB	1	SAS	M3
	PhysicalDisk3		466 GB	1	SAS	M3
	PhysicalDisk4		466 GB	1	SAS	M3
	PhysicalDisk5		466 GB	1	SAS	M3
	PhysicalDisk6		466 GB	1	SAS	M3

```
#Get All Physical Disks.
```

```
$DisksToUse = Get-PhysicalDisk | Where-  
Object {$_ .IsPooled -eq $False}
```

```
#Locate the Storage Subsystem object for  
Windows Storage spaces to use as an input  
for Storage Pool creation.
```

```
$SubSystem = Get-StorageSubsystem |  
Where-Object {$_ .Manufacturer -eq  
"Microsoft Corporation"}
```

#Create the Storage Pool by utilizing the *\$DiskstoUse* and *\$Subsystem* variables created previously.

```
New-StoragePool -StorageSubSystemId  
$SubSystem.ID -FriendlyName DemoPool  
-PhysicalDisks $DisksToUse
```

```
$SpacesDemoPool = Get-StoragePool -  
FriendlyName DemoPool
```

```
# Create Several Storage Spaces
```

```
New-VirtualDisk -FriendlyName "SQLLogs"  
-StoragePoolID $SpacesDemoPool.ID -  
StorageAttributesName Simple -  
ProvisioningScheme Sparse -Size 42TB
```

```
#Get the newly created Virtual Disk  
objects
```

```
$SQLLogsV = Get-VirtualDisk -  
FriendlyName "SQLLogs"
```

#Get the Disk objects that correspond to the Virtual Disks and then Initialize them.

```
$SQLLogDisk = Get-VirtualDisk -  
FriendlyName $SQLLogsV.FriendlyName |  
Get-Disk -ErrorAction SilentlyContinue
```

```
Initialize-Disk -InputObject $SQLLogDisk
```

```
#Create Partitions
```

```
New-Partition -InputObject $SQLLogDisk
```

#Format the volume

```
$SQLLogPartition = Get-Partition -  
DiskId $SQLLogDisk.Id | Where-Object  
{$_ .Type -ne "Reserved"}($SQLLogPartition  
| Format-Volume -NewFileSystemLabel  
"SQLLogs" -Confirm:$False -Erroraction  
SilentlyContinue)
```

# Storage Management Providers

- ❑ You no longer need to develop two providers
- ❑ There is **NO** functional difference between the two options
- ❑ SMP:
  - ❑ Lower development overhead
  - ❑ Easier learning curve
  - ❑ Microsoft platform only
- ❑ SMI-S:
  - ❑ Industry standard
  - ❑ Works with many platforms

# For More Information

[sfsquestions@microsoft.com](mailto:sfsquestions@microsoft.com)