



## Proven SQL Server Architectures for High Availability and Disaster Recovery

SQL Server Technical Article

**Writer:** Paul S. Randal (SQLskills.com)

Technical Reviewers: Darmadi Komo, Sanjay Mishra, Prem Mehra, Vineet Rao, Gopal Ashok, Kimberly L. Tripp (SQLskills.com)

**Published:** <Month/Year>

**Applies to:** SQL Server 2005, SQL Server 2008, SQL Server 2008 R2

### **Summary:**

This whitepaper describes five commonly-deployed architectures using SQL Server 2005 and SQL Server 2008 that are designed to meet the high-availability and disaster recovery requirements of enterprise applications. The whitepaper will describe the architectures and also present case studies that illustrate how real-life customers have deployed these architectures to meet their business requirements.

This whitepaper is targeted at architects, IT Pros, and senior database administrators tasked with architecting a high-availability and disaster-recovery strategy for their mission-critical applications. It assumes the reader has a good understanding of Windows and SQL Server technologies and has sufficient knowledge of transaction processing. These basic features and topics are not covered.

## Copyright

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.


Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2010 Microsoft Corporation. All rights reserved.

Microsoft, <plus, in alphabetical order, all Microsoft trademarks used in your white paper> are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

## Contents

	1
Introduction and Overview .....	3
Failover Clustering for High Availability with Database Mirroring for Disaster Recovery .....	4
Deployment Example: CareGroup Healthcare System .....	6
Database Mirroring for High Availability and Disaster Recovery .....	7
Deployment Example: bwin Corporation .....	9
Geo-Clustering for High Availability and Disaster Recovery .....	11
Deployment Example: QR Limited .....	11
Failover Clustering for High Availability Combined with SAN-Based Replication for Disaster Recovery .....	13
Deployment Example: Progressive Insurance .....	14
Peer-to-Peer Replication for High Availability and Disaster Recovery .....	15
Deployment Example: An International Travel Industry Company .....	16
Conclusion .....	17

### Introduction and Overview

SQL Server 2005 and SQL Server 2008 include many technologies that can be used to minimize downtime and maximize data protection so that database administrators can ensure smooth operation, continuous access to business critical data, and meet availability levels according to various service level agreements.

Sometimes high-availability and disaster-recovery architectures are unfortunately designed without considering the necessary business requirements—possibly there is already an incumbent technology, or the designers are familiar with a certain technology and choose it as the basis for a new architecture. This choice, when coupled with a lack of understanding of the capabilities of the various high-availability and disaster-recovery technologies, can lead to an architecture that fails to meet the business needs.

It is imperative that the high-availability and disaster-recovery requirements of the business are the drivers when evaluating which technologies are suitable as part of the architecture. The two major business needs to consider are:

- The duration of acceptable application downtime, whether from an unplanned outage or from scheduled maintenance/upgrades (i.e. the defined Recovery Time Objective—RTO).
- The ability to accept potential data loss from an outage (i.e. the defined Recovery Point Objective—RPO).

There is an existing whitepaper, “High-Availability with SQL Server 2008” (available at <http://msdn.microsoft.com/en-us/library/ee523927.aspx>), that contains information about each of the high-availability technologies in SQL Server 2008, as well as further links to other whitepapers and technical resources. It also describes how to evaluate business requirements and technical/non-technical limitations to help choose appropriate technologies.

However, there is a lack of information regarding proven architectures and real-life customer deployments, where the high-availability and disaster-recovery architecture was chosen after careful requirements analysis and technology evaluation.

This whitepaper provides a consolidated description of five proven and commonly deployed high-availability and disaster-recovery architectures, in terms of the technologies used and the business requirements they are able to meet.

Furthermore, before committing to the implementation of any technology strategy, many companies would like some level of reassurance that what they are attempting has been successfully accomplished previously. To meet this need, Microsoft regularly publishes case studies showing how their technologies have been used. This whitepaper also includes references to relevant case studies of real-life customer deployments for each of the architectures described.

Together these two whitepapers will provide the information necessary to allow the design of an appropriate and successful high-availability and disaster-recovery architecture.

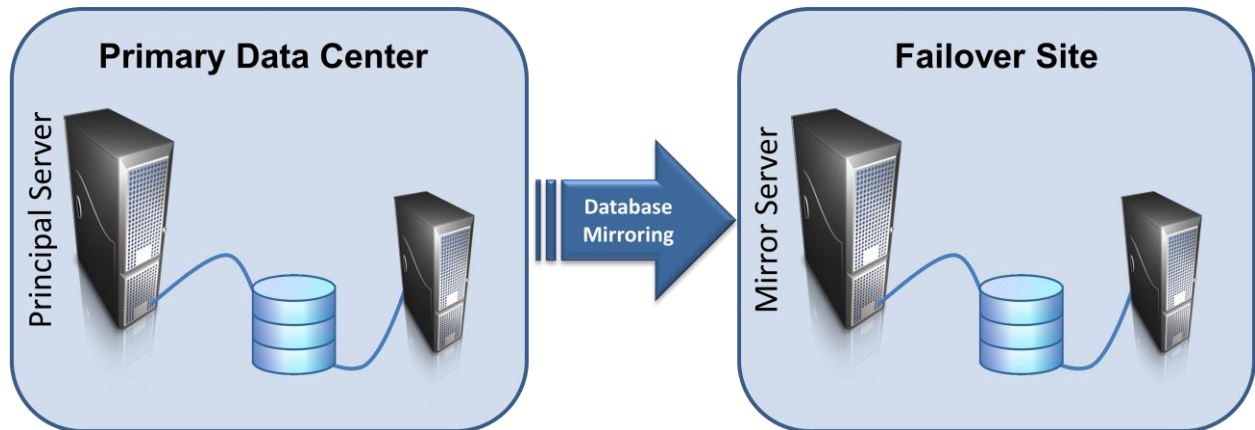
### **Failover Clustering for High Availability with Database Mirroring for Disaster Recovery**

In this architecture, failover clustering provides the local high availability and database mirroring provides the disaster recovery capability. A failover cluster on its own protects against physical server, Windows Server, and SQL Server failures but does not maintain a redundant copy of the data and so does not protect against a major outage like an I/O subsystem failure, power failure, or failure of the network link to the primary data center.

Database mirroring is one way to provide a redundant copy of a single database on a separate physical server, where the server can be in the same data center or geographically separated.

This architecture is widely adopted by customers who are familiar and comfortable with the installation, configuration, and maintenance of failover clusters.

A typical implementation of this architecture involves a failover cluster in the primary data center with database mirroring to a secondary data center or disaster-recovery site, as shown in Figure 1 below.



**Figure 1: Failover clustering combined with database mirroring.**

There are a number of variations and configuration options for this architecture depending on the business requirements, including the following:

1. Each data center has a failover cluster with database mirroring between them. If the business requirements state that the workload performance should not be impacted after a failover to the secondary data center, the mirror server needs to have the same hardware configuration (and hence workload servicing capability) as the failover cluster in the primary data center. The alternative, of course, is to have a less capable stand-alone server as the mirror server—however, this is not a recommended best practice.
2. Synchronous vs. asynchronous database mirroring. Synchronous database mirroring can allow a zero data-loss requirement to be met, potentially with some workload performance impact depending on the type of workload and the network bandwidth between the two data centers. Asynchronous database mirroring does not guarantee zero data loss in the case of a disaster, but has no impact on workload performance.
3. Automatic failover to the secondary data center. When synchronous database mirroring is configured with a third (optional) witness server, the database mirroring system can detect a failure and perform an automatic failover to the secondary data center. If this behavior is desirable, care must be taken to configure the database mirroring *partner timeout* such the local failover cluster fails over before database mirroring performs a failover to the secondary data center.
4. Automatic client connection to the secondary data center. If explicit client redirection is used, the client specifies the `FAILOVER_PARTNER` in the connection string. After a database mirroring failover has occurred, the client simply has to reconnect and the connection will automatically be made to the secondary data center. Alternatively, some form of external routing can be used (some installations have used DNS routing, for instance).

## Deployment Example: CareGroup Healthcare System

CareGroup manages a number of hospitals in the Boston area and has 390 databases underpinning 146 mission-critical clinical applications, totaling 2 terabytes of data.

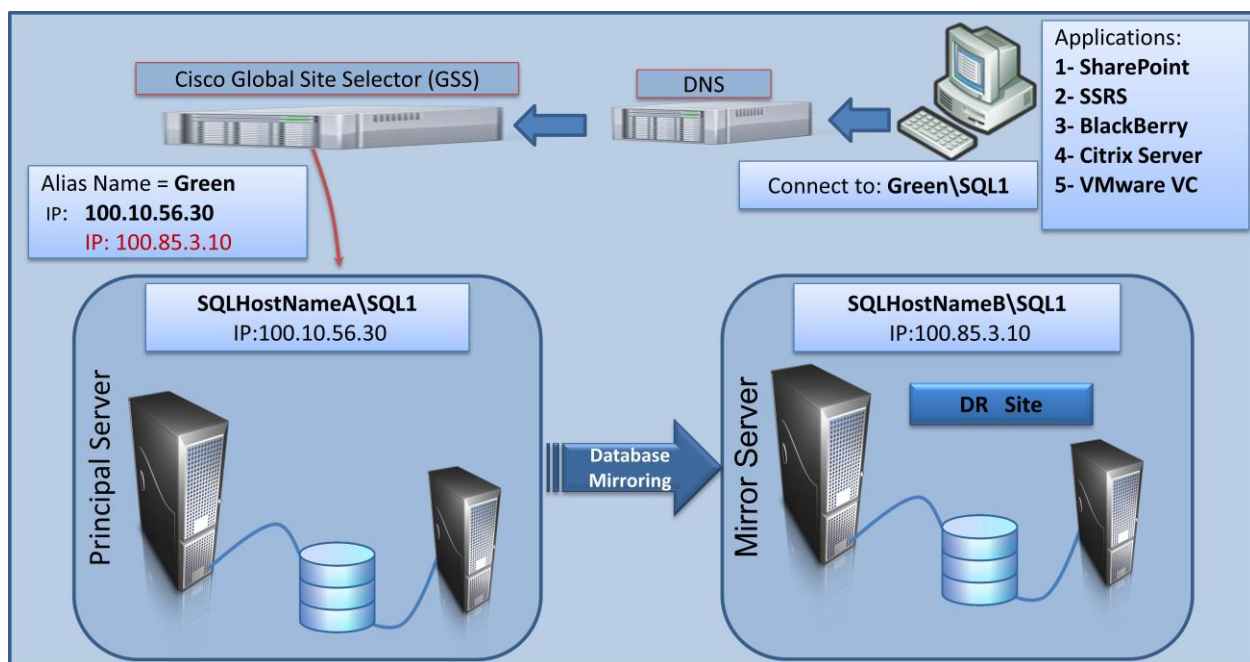
The RPO and RTO requirements for their databases depend on the importance of the data contained within the database. CareGroup defined three tiers to classify this:

- 'AAA': zero downtime and zero data loss
- 'AA': up to one hour of downtime and data loss
- 'A': up to 1 day of downtime and data loss

CareGroup also wanted to remove the need to hard-code the database mirroring partner server names in the application connection string to redirect client connections during a disaster recovery failover.

Using these requirements, they were able to determine that a combination of SQL Server failover clusters in two data centers with database mirroring between the data centers was the appropriate solution. For the 'AAA' databases, database mirroring is configured synchronously to avoid data loss, and for the lower-classed databases it is configured asynchronously. In the event of a failure, DNS routing is used to redirect traffic to the secondary data center.

The architecture that CareGroup deployed is illustrated in Figure 2 below.



**Figure 2: High-availability and disaster-recovery architecture deployed by CareGroup.**

The Global Site Selector (GSS) enables the various applications at CareGroup to seamlessly connect to the appropriate database mirroring principal server, without having to specify partner server names in the connection string for the client redirection. This is necessary as some of the

applications that CareGroup uses are from 3<sup>rd</sup>-party vendors that do not permit (or require too much work for) the client connection string to be altered to use explicit client redirection.

Instead, the applications specify one SQL Server instance name in the connection string, of the form "Green\SQL1". In this connection string where the server name "Green" is a DNS alias that resolves to the GSS device, which in turn translates the alias "Green" into the appropriate IP address of the current database mirroring principal server.

Using this architecture, CareGroup was able to meet their availability requirements, including performing an upgrade to SQL Server 2008 using database mirroring that only involved a few minutes of downtime.

As an aside, by upgrading to SQL Server 2008, CareGroup can also take advantage of some of the other features in the product:

- Transparent Data Encryption to allow all data in CareGroup's databases to be encrypted without requiring costly changes to existing applications, which satisfied their desire to increase the security of patient records.
- Advanced Server Auditing to allow CareGroup to monitor all activity in databases across their enterprise and ensure compliance with HIPAA and other sets of regulations.
- Policy-based Management and Performance Data Collection gives CareGroup enhanced configuration policy enforcement, and easy performance diagnostic information generation, storage, and analysis, respectively.
- Resource Governor allows CareGroup to guarantee critical workload performance and prevent unexpected workloads from affecting application availability.
- Reporting Services will allow CareGroup to create a single, consolidated reporting solution.

More information on this solution can be found at:

- [http://www.microsoft.com/casestudies/Case\\_Study\\_Detail.aspx?casestudyid=4000001003](http://www.microsoft.com/casestudies/Case_Study_Detail.aspx?casestudyid=4000001003)

Another example of this architecture is described in the case study of the deployment by ServiceU Corporation, available at:

- <http://sqlcat.com/whitepapers/archive/2009/08/04/high-availability-and-disaster-recovery-at-serviceu-a-sql-server-2008-technical-case-study.aspx>

### **Database Mirroring for High Availability and Disaster Recovery**

In this architecture, synchronous database mirroring can be used to maintain an up-to-date, redundant copy of a single database by continually sending transaction log records from the principal database on the principal server to the mirror database on the mirror server.

If a failure occurs, the mirror database can be brought online as the new principal database and client connections can be failed over. As long as the mirror database remains synchronized with the principal database, zero data loss results when a failover is necessary.

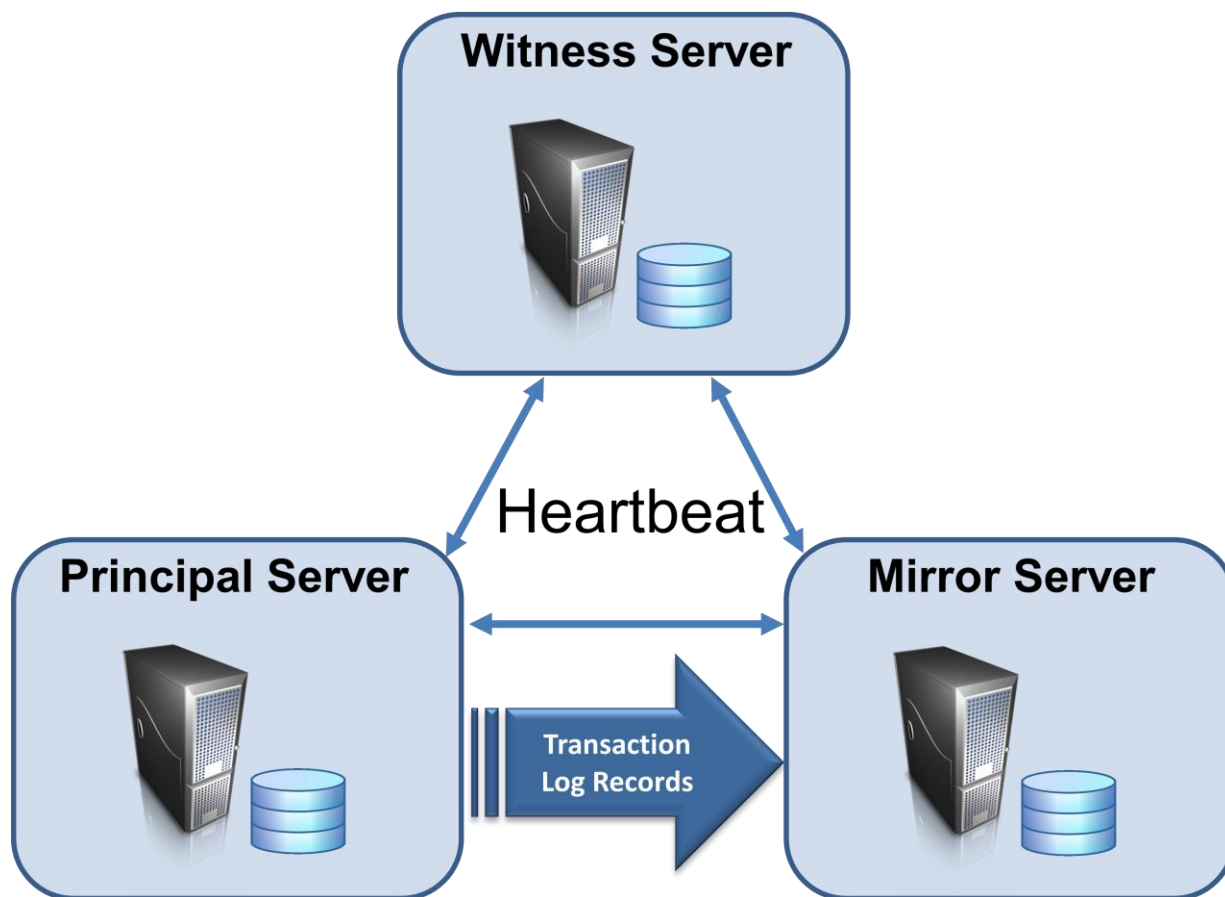
There are a number of variations and configuration options for this architecture depending on the business requirements, including the following:

1. Configuring a third server, the witness. When a witness server is included as part of a synchronous database mirroring architecture, a failover can be performed automatically when a failure is detected, providing the highest availability of the data. If database mirroring is used between two data centers, it is recommended to place the witness in a third data center, for the highest availability.
2. Configuring asynchronous database mirroring. When the network link between the principal and mirror servers is not sufficient to synchronously send the transaction log records without leading to workload performance degradation, database mirroring can be configured to send the transaction log records asynchronously. While this removes the performance degradation, it also removes the assurance of zero data-loss if a failover is necessary. This may be perfectly acceptable depending on the desired RPO.
3. Configuring database mirroring and log shipping. Database mirroring allows a single mirror of the principal database, so for added redundancy, one or more log shipping secondary servers can also be configured as warm-standby databases.

This architecture is typically lower cost than one involving failover clustering, as the principal and mirror servers can be standalone servers with direct-attached storage, rather than each part of a multi-server failover cluster with SAN storage. It is most commonly used when the business requirements call for databases to be protected for disaster recovery purposes and for some businesses, when there is some technical or operational reason for not using failover clustering.

A typical implementation of this architecture involves a principal server in the primary data center with a mirror server in a secondary data center or disaster-recovery site. There is often a third server, the witness, included in the architecture as shown in Figure 3 below.





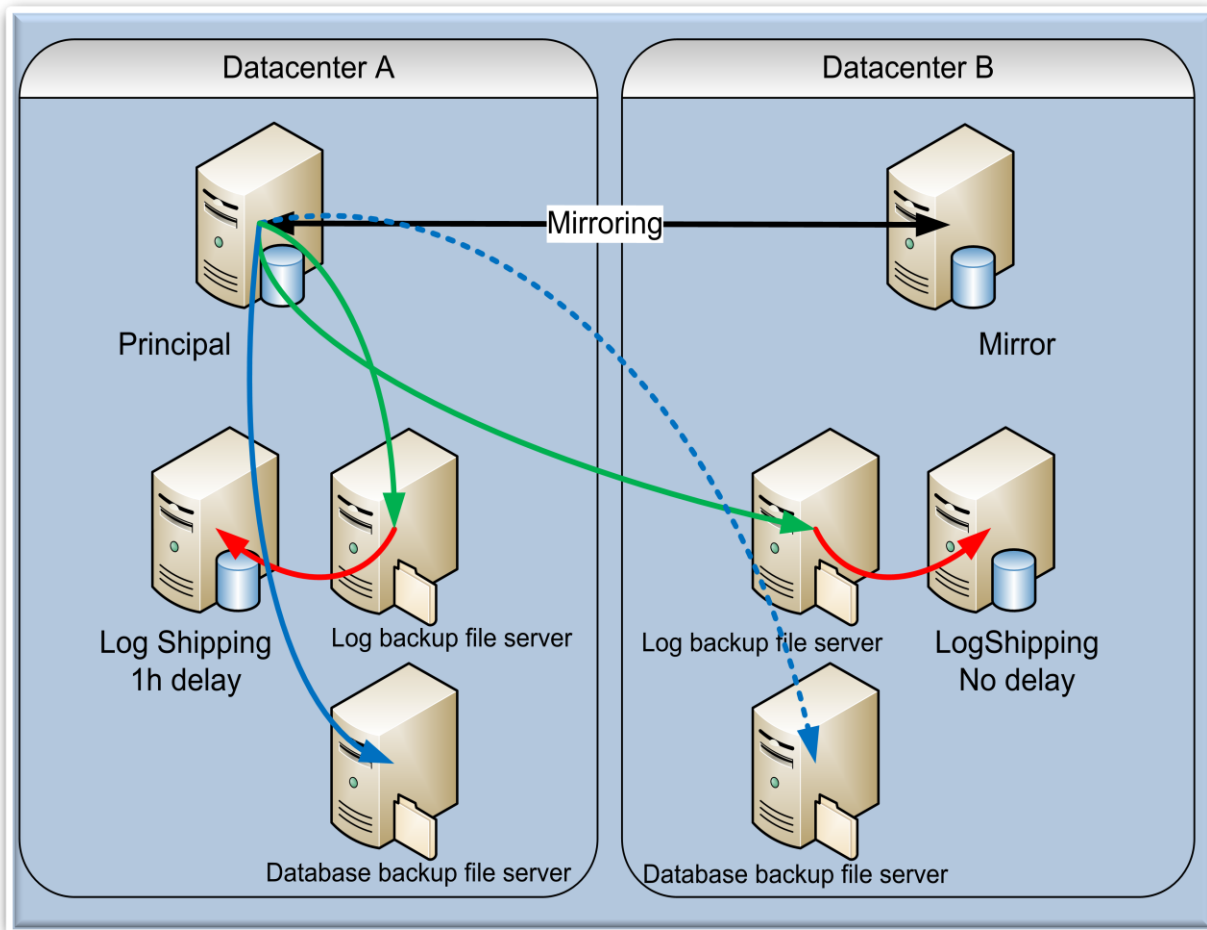
**Figure 3: Database mirroring for high availability and disaster recovery.**

### Deployment Example: bwin Corporation

bwin is an online gaming company that provides a wide variety of games and sports betting, with up to 1 million bets per day placed on more than 90 sports. They have more than 100 terabytes of data spread over 850 databases on more than 100 instances of SQL Server, with the largest single database being more than 4 terabytes. At peak times their system can support more than 450 thousand Transact-SQL statements per second.

They wanted to be able to cope with complete loss of their primary data center, and their budget allowed them to implement a solution which meets their business requirements. They also want zero data-loss and 99.99% availability 24x7. The solution they chose involved synchronous database mirroring over dark-fiber between two data centers that are 11 kilometers apart. They also maintain two log shipping secondaries—one in each data center. The log shipping secondary in the main data center is configured with 1-hour restore delay to allow recovery from accidental user errors (such as delete or update).

The architecture that bwin deployed is illustrated in Figure 4 below.



**Figure 4: High-availability and disaster-recovery architecture deployed by bwin.**

This architecture was deployed on SQL Server 2005 and enabled bwin to meet all their business requirements around high availability and disaster recovery, while also being able to service their peak workload. Bwin plans to upgrade this architecture in future to add a database mirroring witness server to allow automatic failovers.

After moving to SQL Server 2008, bwin is planning to take advantage of some of the new features in the product:

- Database mirroring log stream compression will result in improved throughput.
- Backup compression will reduce the size of some backups by over 80%. This will allow bwin to extend the life of its systems as it experiences rapid growth.
- Enhanced Auditing to allow bwin to comply with the myriad regulations in the countries around the world in which it operates.

More information on bwin's testing an migration to SQL Server 2008 can be found at:

- [http://www.microsoft.com/casestudies/Case\\_Study\\_Detail.aspx?casestudyid=4000001470](http://www.microsoft.com/casestudies/Case_Study_Detail.aspx?casestudyid=4000001470)

## Geo-Clustering for High Availability and Disaster Recovery

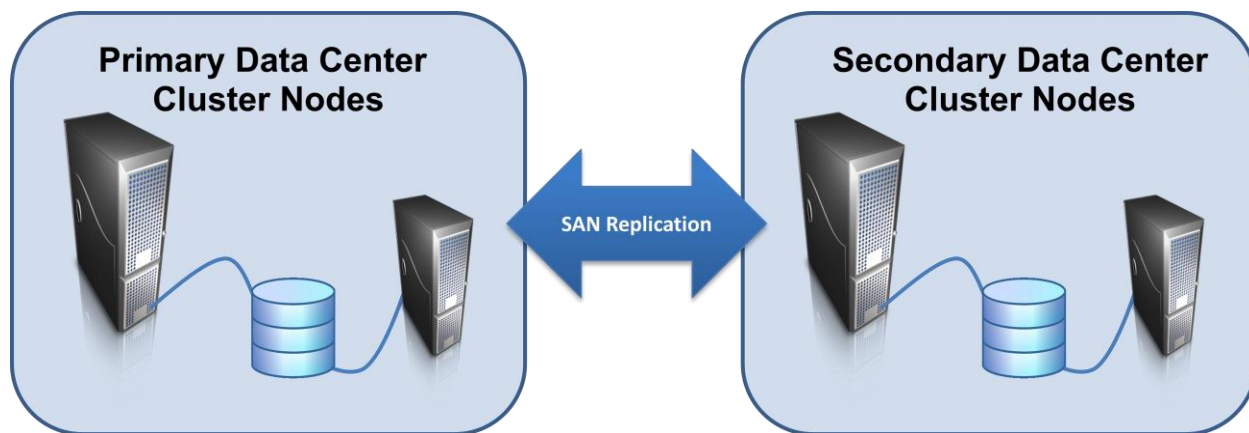
In this architecture, a geographically-dispersed cluster (geo-cluster) is implemented, which behaves like a regular failover cluster but the constituent servers are in geographically separate sites. The failover cluster quorum is maintained between the sites, and the data disks are synchronously or asynchronously mirrored.

If the servers fail in the main data center, the SQL Server instances are started in the secondary data center in a manner similar to when the servers are collocated and the clients reconnect in the same way as for a failover of a regular failover cluster (and vice-versa). To achieve this it is often necessary to use a very fast network link (like dark fiber) and a network configuration that abstracts the physical location of the cluster nodes from the clients.

The cluster nodes themselves are unaware that they are part of a geo-cluster so all replication must be handled at the storage level. If the data disks are synchronously mirrored between sites, then zero data-loss will occur if a failover is necessary, but requires sufficient network bandwidth.

This architecture is deployed when seamless failover of an entire SQL Server instance is required between multiple data centers, avoiding the potential downtime of having to perform a disaster recovery operation.

A typical implementation of this architecture involves the main failover cluster nodes in the primary data center with the other failover cluster nodes in the secondary data center or disaster-recovery site, as shown in Figure 5 below.



**Figure 5: Geo-Clustering for high availability and disaster recovery.**

### Deployment Example: QR Limited

QR Limited is Australia's leader in rail transportation and logistics, operating more than 1,000 train services a day, including carrying more than 170,000 passengers and more than 683,000 tons of freight.

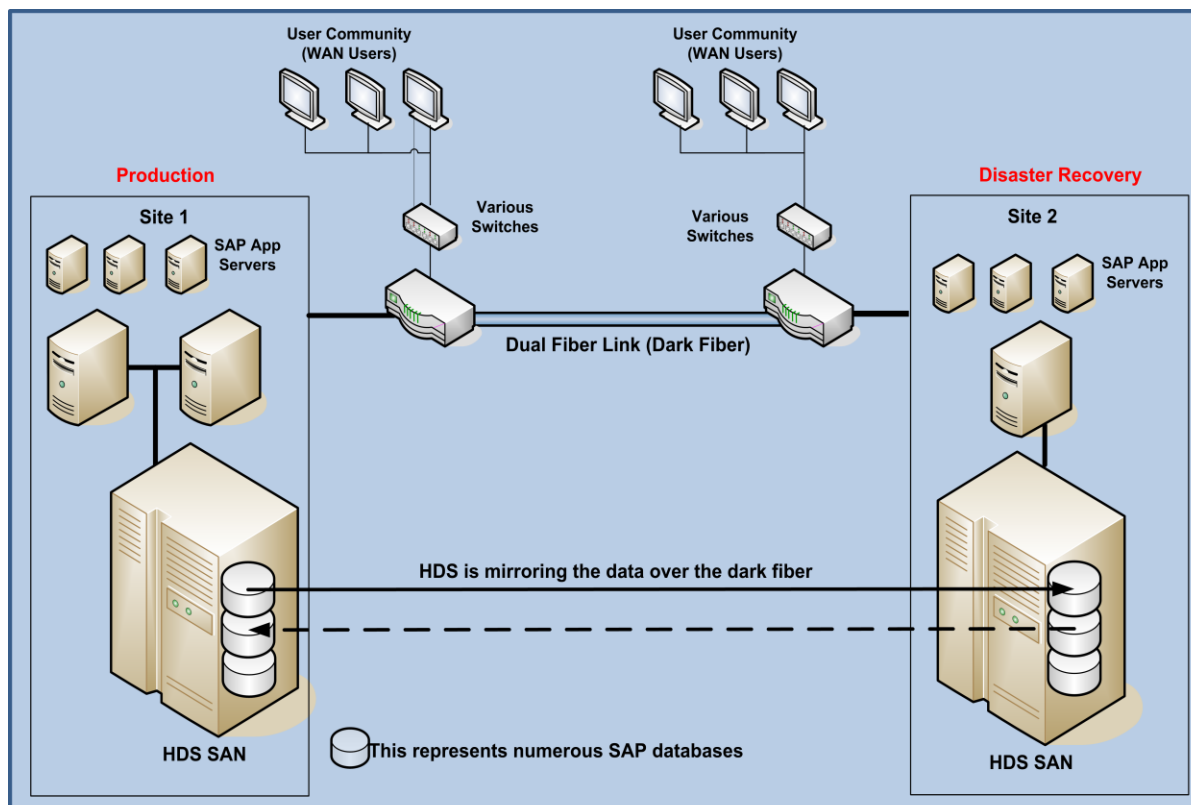
QR Limited migrated their SAP databases from a legacy mainframe onto a SQL Server 2005 and wanted to provide high availability and disaster recovery capabilities for the various SAP

## Proven SQL Server Architectures for High Availability and Disaster Recovery

databases and the one terabyte ERP database, but with the ability to seamlessly protect against loss of a data center without having to perform protracted disaster recovery.

They chose to implement a geo-cluster between two data centers 5 kilometers apart, with a fiber link between them to accommodate the SAN replication network traffic and all client communications to the active cluster nodes. The data disks are synchronously from the production data center to the disaster recovery data center.

The architecture that QR Limited deployed is illustrated in Figure 6 below.



**Figure 6: High-availability and disaster-recovery architecture deployed by QR Limited.**

By switching from mainframe-based DB2 to SQL Server 2005, they realized the following additional benefits to their enhanced high availability and disaster recovery:

- \$100,000 savings per month mainframe cost savings.
- SAP ERP transactional response times that are 20% to 40% faster.
- An 8-to-1 reduction in batch processing time.

More information on this solution can be found at:

- [http://www.microsoft.com/casestudies/Case\\_Study\\_Detail.aspx?casestudyid=400003421](http://www.microsoft.com/casestudies/Case_Study_Detail.aspx?casestudyid=400003421)

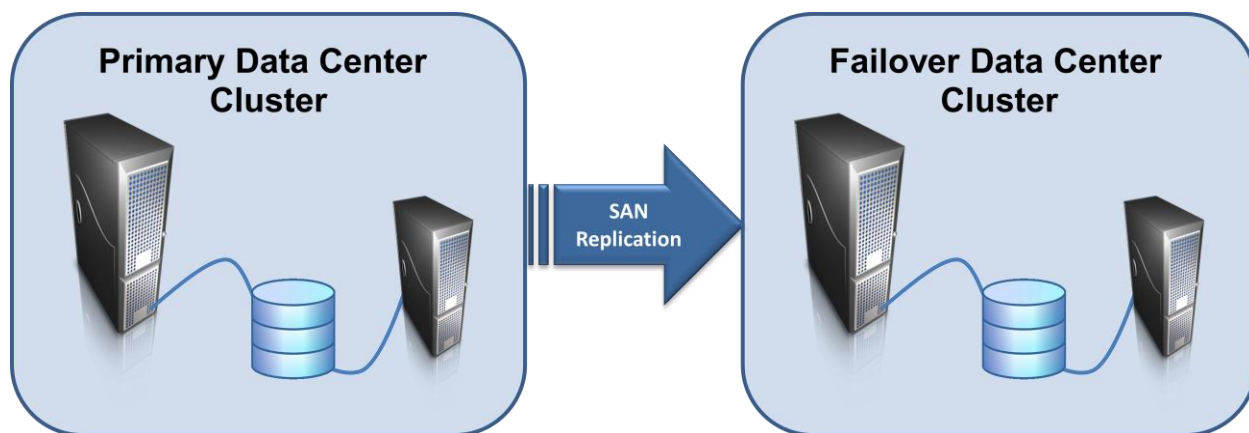
## Failover Clustering for High Availability Combined with SAN-Based Replication for Disaster Recovery

This architecture uses failover clustering to provide local high availability and SAN-based replication to provide disaster recovery. The database volumes on the SAN in the main data center are mirrored to another SAN in a secondary data center, which does not necessarily need to be attached to another failover cluster.

If the main data center is lost, there is no automatic failover of a SQL Server instance to the server in the secondary data center, but there is a redundant copy of the databases that can be mounted and attached to Windows and to a SQL Server instance.

This architecture is often used when a business requires that databases from different vendors, used by related but distinct applications, be logically consistent to maintain data integrity in the case of a disaster.

A typical implementation of this architecture involves a failover cluster in the primary data center with SAN-based replication of the storage used by the various SQL Server instances to a SAN in the secondary data center or disaster-recovery site, as shown in Figure 7 below.



**Figure 7: Failover clustering combined with SAN-based replication.**

There are a number of variations and configuration options for this architecture depending on the business requirements, including the following:

- Synchronous vs. asynchronous replication. With synchronous replication, there is zero data loss when a failure occurs, but more network bandwidth may be required to prevent workload performance degradation. With asynchronous replication, no such assurance is available, but there is no performance degradation.
- Server configuration in the secondary data center. Sometimes there is a standalone server in the secondary data center instead of a failover cluster. This architecture is used when requirements allow local availability to be lower after the loss of a data center, or when budgetary limitations do not allow for a failover cluster in the secondary data center.

- Bi-directional replication. If there is an active server (of failover cluster) in each data center, SAN-based replication can be used to provide data redundancy between the data centers for the data from both failover clusters.

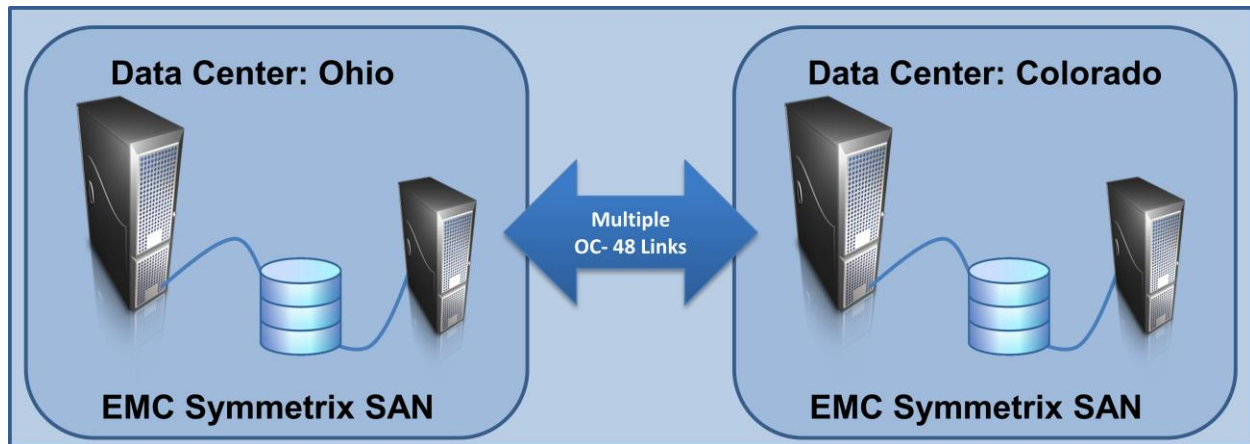
### Deployment Example: Progressive Insurance

Progressive Insurance is one of the largest auto insurance companies in the United States with revenues of more than \$14 billion. They were replacing a 30-year-old mainframe-based policy management application that served millions of customers through a network of 30000 independent insurance agencies. When fully deployed, the total data size will be 10 terabytes and the largest table will have almost 2 billion rows.

As well as replacing the legacy application, Progressive required no more than 1 hour of data loss and a maximum allowable downtime of 24 hours.

Progressive chose to use failover clusters in two active data centers for local high availability, with asynchronous SAN replication between them to provide data redundancy in the event of a disaster.

The SQL Server 2005 architecture that Progressive deployed is illustrated in Figure 8 below.



**Figure 8: High-availability and disaster-recovery architecture deployed by Progressive.**

The OC-48 links provide 2.5 gigabits per second and are shared with other Windows servers and mainframe to provide asynchronous replication between the EMC Symmetrix DMX 3 and 4 series SANs.

**Note to reviewers: this section can be removed – totally up to you.**

Progressive is also making use of the following SQL Server 2005 features to enhance availability:

- Table partitioning to allow easier management of 1-terabyte sized tables, especially using the 'sliding-window' mechanism to allow fast range-deletes without long-running, blocking operations.



- Online index operations to allow critical index maintenance to be performed without requiring scheduled downtime.
- Dynamic Management Views to allow much easier insight into system conditions that could affect performance and data availability.

More information on this solution can be found at:

- [http://www.microsoft.com/casestudies/Case\\_Study\\_Detail.aspx?casestudyid=4000002133](http://www.microsoft.com/casestudies/Case_Study_Detail.aspx?casestudyid=4000002133)

### **Peer-to-Peer Replication for High Availability and Disaster Recovery**

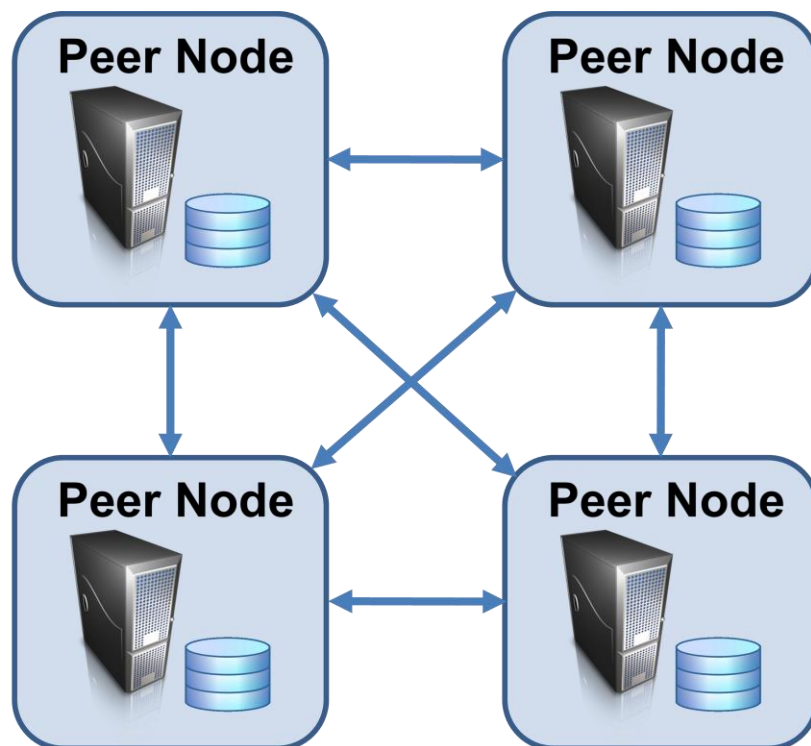
This architecture uses peer-to-peer replication to provide both high availability and disaster recovery. Peer-to-peer replication uses a bi-directional transactional replication stream, with all nodes in the replication topology receiving updates from all other nodes.

Peer-to-peer replication involves some latency between a transaction committing on one node and the change being replayed on all other nodes in the replication topology, so it is not suitable for satisfying zero data-loss requirements. It also does not provide automatic detection of failures or automatic failover. It does, however, allow multiple copies of the protected data to be made, and furthermore, those copies are available for read and (with a lot of planning and care) write activity.

Peer-to-peer replication essentially makes a database both a publication and a subscription database, and so local insert, update, and delete activity is permitted in the same database and tables that are receiving updates from other nodes. For this reason, table schemas and application logic must be carefully developed to avoid conflicts (even with SQL Server 2008, which helps with automatic conflict detection and resolution).

This architecture is used when the secondary data copy is required to be available for reading or writing, and/or when multiple copies of the data must be maintained.

A typical implementation of this architecture involves a peer-to-peer node in each data center, with updates occurring and being received by all other nodes in the other data centers, as shown in Figure 9 below.



**Figure 9: Peer-to-peer replication for high availability and disaster recovery.**

### Deployment Example: An International Travel Industry Company

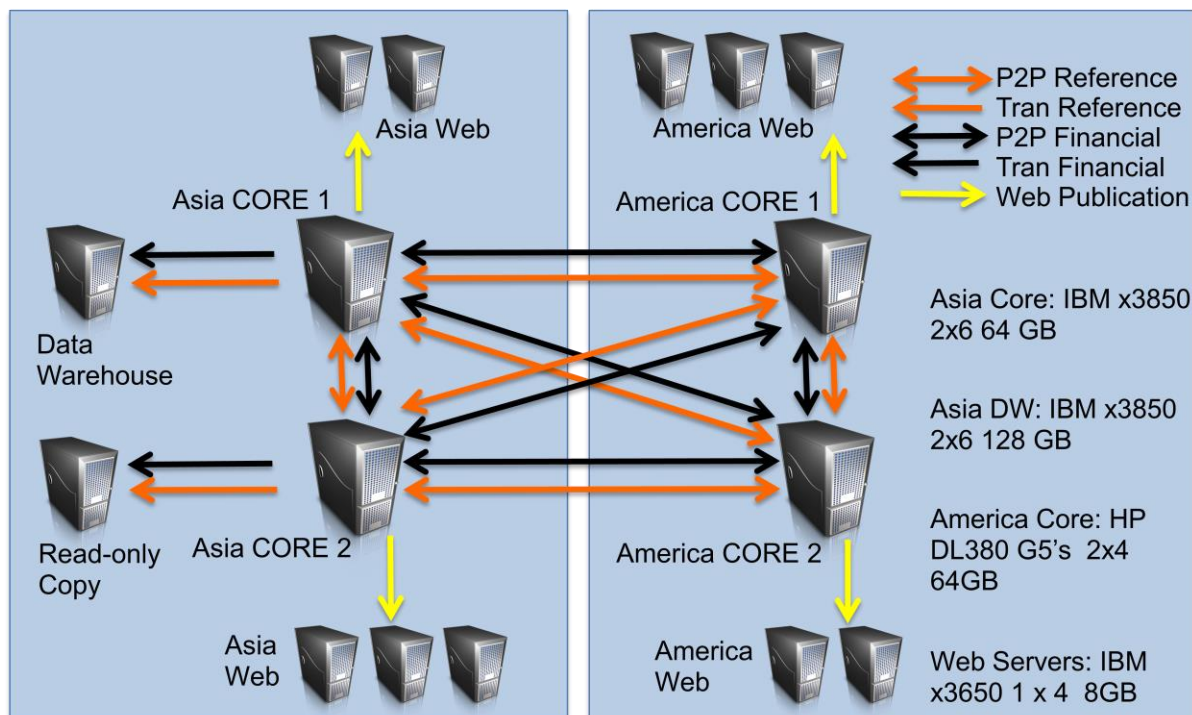
This company is one of Asia's leading and fastest growing provider of online hotel reservations, with data centers in Asia and the United States. In their previous architecture, all write activity was handled in the Asia data center, whereas only reads could be serviced from the US data center.

They wanted to remove the single point of failure—the data center in Asia—by having all data available at both data centers, and either data center able to handle write requests. They chose to implement a combination of peer-to-peer replication as well as traditional transactional replication to use the disaster-recovery hardware to process the read-only workload.

Database mirroring and log shipping were not options as both data centers had to be able to handle write requests—which neither technology permits. Failover clustering was similarly discounted, and also because of a desire to limit the capital expenditure on hardware.

The architecture that the travel company deployed is illustrated in Figure 10 below.





**Figure 10: High-availability and disaster-recovery architecture deployed by the travel company.**

The SQL Server Customer Advisory team worked closely with this customer to produce a very detailed whitepaper describing the requirements analysis, technology analysis, replication solution design, and testing strategy. It is available at <http://sqlcat.com/whitepapers/archive/2009/09/23/using-replication-for-high-availability-and-disaster-recovery.aspx>.

## Conclusion

This whitepaper has highlighted five commonly deployed high-availability and disaster-recovery architectures using SQL Server technologies, along with examples of real-life customer deployments of these architectures.

The high-availability and disaster-recovery technologies provided in SQL Server 2005 have been further enhanced in SQL Server 2008. It is very important to select architectures after carefully considered business requirements, and then deploy the technology to meet those requirements. It can be tempting to select a new and interesting (or possibly incumbent) technology, regardless of the business requirements, but that can be counterproductive in the long run.

It can be very useful to review published reference implementations from SQL Server customers, both to see what technology choices worked for the customers' requirements, and also to potentially learn from their experiences.

## Proven SQL Server Architectures for High Availability and Disaster Recovery

Finally, while SQL Server 2005 provides all the technologies needed to implement a successful high-availability and disaster-recovery architecture, SQL Server 2008 has many enhancements to these technologies, and includes many others that can aid with security, maintainability, and performance

The information presented in this whitepaper, and in those to which it links, should provide a basis for anyone tasked with evaluating and choosing SQL Server 2008 technologies, with the goal of protecting and increasing the availability of critical business data.

For more information:

- <http://www.microsoft.com/sqlserver/2008/en/us/high-availability.aspx>
  - This whitepaper contains links to other whitepapers specific to the high availability and disaster recovery technologies, including how to use them in various combinations.
- <http://sqlcat.com/tags/Availability/default.aspx>
- <http://blogs.msdn.com/psssql/>
- <http://blogs.technet.com/dataplatforminsider/default.aspx>
- <http://www.sqlskills.com/blogs/paul>
- <http://www.sqlskills.com/blogs/kimberly/>
- <http://www.sqlha.com/blog/default.aspx>

Did this paper help you? Please give us your feedback. Tell us on a scale of 1 (poor) to 5 (excellent), how would you rate this paper and why have you given it this rating? For example:

Are you rating it high due to having good examples, excellent screen shots, clear writing, or another reason?

Are you rating it low due to poor examples, fuzzy screen shots, or unclear writing?

This feedback will help us improve the quality of white papers we release.

[Send feedback.](#)