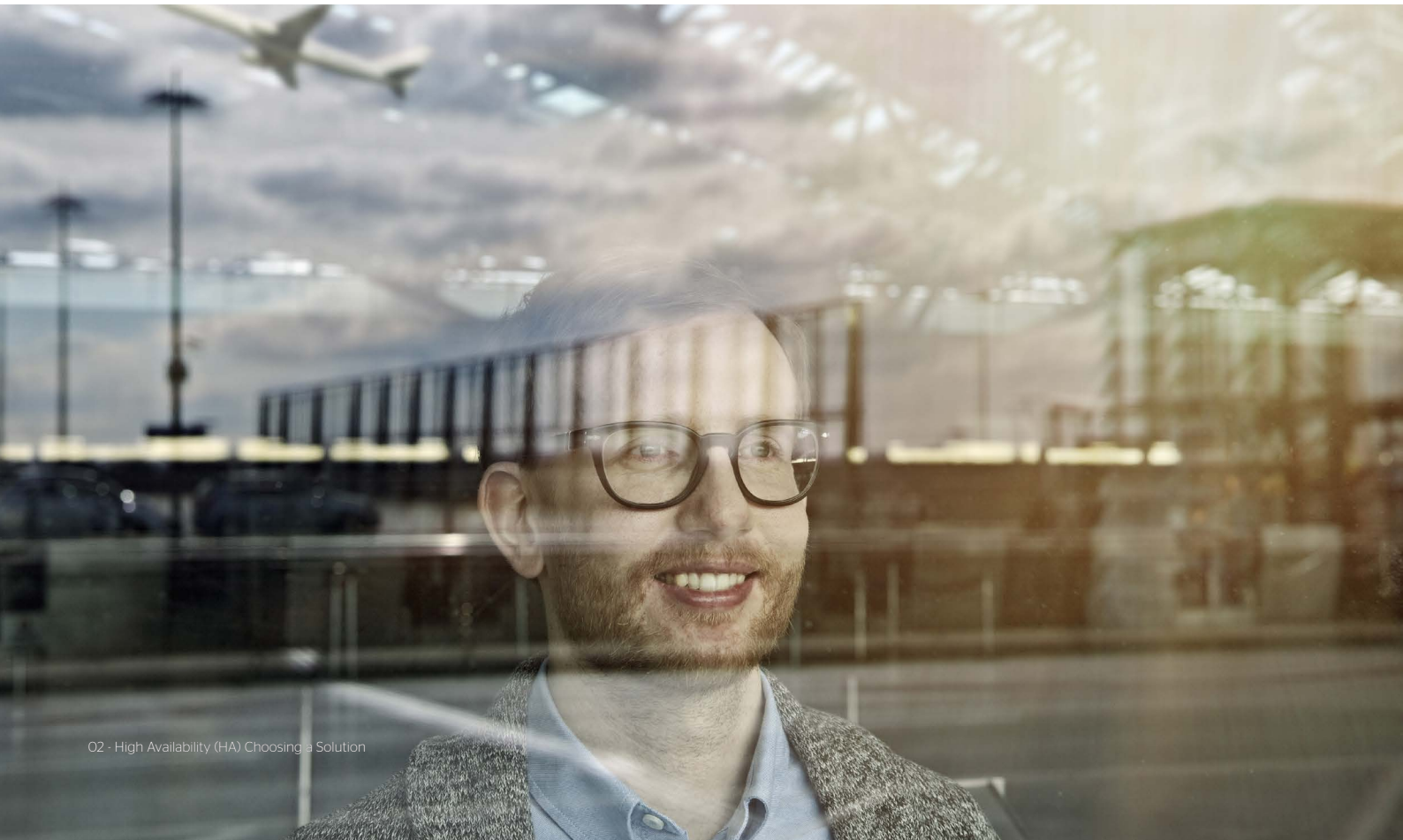
High Availability (HA) Choosing a Solution

Trusted partner for your **Digital Journey**

Summary

What to look for when choosing a high-availability solution?

This white paper is based on a case study: a crisis in an airport.



Case study: a crisis in an airport

Every computer-system-based activity, both big and small, is one day or the other faced with the problem of computer failure. Unfortunately, the day the failure occurs, a small problem may turn into a general crisis due to a succession of errors. This was exactly what happened at an airport, despite the fact that it was equipped with a redundant solution. This story will enable us to highlight the main characteristics of a good-quality high-availability solution.

The crisis occurred at an airport on the traveler-management application. This application displays information on the airport's notice boards, making it possible for travelers to locate their boarding gate or to know where to retrieve their luggage.

High availability based on redundancy

The size of the traveler-management database is not big. But this database is extremely sensitive and must withstand the worst case scenario at an airport: aircraft crash-landing on a computer room.

The selected redundancy solution consisted of two separate servers located in two remote computer rooms. Each server had a local traveler-management database, and a replication tool was used to ensure database redundancy between the main server and the backup server.

The incident

To maintain the database with the new flights for one week, every Sunday the database is updated with the flight schedules for the following week.

Unfortunately, one Sunday, while the flight update operation was in progress on the main server, replication stopped on the backup server. So, at the end of the update operation, the main server contained the flights for the following week, and the backup server the flights for the previous week.

On Monday, the main server supplied the right flight schedules to the airport's notice boards. For the IT department, airport management was working properly, despite the undetected absence of replication on the backup server.

The crisis

In the middle of the week, the main server rebooted automatically as a result of an error. Unfortunately, it remained blocked in its booting process and failed to reboot the traveler-management application. The notice boards at the airport went blank, and the IT department was alerted.

Since the main server was blocked in its booting process, a decision was taken to restart the application on the backup server. The application was then restarted on the backup server without any difficulties and, a few minutes later, all the notice boards at the airport became active again. But they displayed to travelers the flight schedules for the previous week!

The red alert at the IT department soon turned into a black alert. The traveler-management application was immediately stopped on the backup server, and the IT team decided to solve the problem of main server reboot. Meanwhile, the airport's information-management application remained unavailable to all travelers.

The situation gets worse

A few hours later, the administrators succeeded in solving the problem on the main server, and in rebooting this latter. But, unfortunately, in the booting phase, the replication tool on the main server started automatically. It detected that replication was already active on the backup server and resynchronized the main server's local database from the backup server. The database of both servers thus contained the flight schedules for the previous week!

Airport activities were disrupted for a whole day, and the main server could only be correctly resynchronized through the special Sunday operation.

In the end, the airport replaced the replication tool with a complete replication and high-availability solution, based on the criteria described hereinafter.

Where are the vulnerabilities of a high availability solution?

The problem encountered by the airport's IT department was due to the replication solution chosen. Replication was working, but the solution was incomplete in terms of high-availability and recovery in case of failure. Yet in its technical data sheet, this solution is said to have high-availability mechanisms with heartbeat and recovery scripts!

Therefore, it is necessary to properly identify the high-availability needs while choosing a replication solution. The needs expressed by the production team after this crisis at the airport are as follows.

Every administrator must know that replication has stopped on the backup server

The first event that led to the crisis at the airport was the updating of flight schedules on the main server on Sunday while the backup server was out of service.

On Monday, when the entire production team arrives, a simple glance at the replication application's administration console should be enough to detect the absence of replication on the backup server. Therefore, the replication solution must have an administration console which can remotely connect to servers and provide a summary of the status of a high-availability application on two servers. It must equally be possible to send mail and to integrate the product in the administration console used by the customer if replication stops.

It should be very easy for every administrator to restart replication on the backup server

On Monday, upon detection of replication stop on the backup server, any administrator from the IT department should be able to restart the replication:

- either by clicking a button on the replication administration console, or
- through a very simple online command offered by the replication solution on the backup server, or
- by rebooting the backup server.

Restart should never be reserved to a replication solution expert. In fact, the system must be made highly available again even when the expert is not present.

It should be possible to resynchronize the backup server while the application is running on the main server

On Monday, when the IT department resumes work, the passenger-management application at the airport is running on the main server, and replication must be restarted as quickly as possible on the backup server. It should, therefore, be possible to resynchronize the backup-server database once the main server is accessed.

Popular file replication solutions cannot resynchronize a backup server without stopping the application on the main server! These products implement replication solutions but absolutely not high-availability solutions.

If a server's local databases are not updated, the solution must, by default, refuse to run the application on the non-updated server

In the case of the airport, the alert turned into a crisis because this feature did not exist.

The backup server with the flight schedule for the previous week was not up-to-date, still nothing stopped the application from starting on this server.

Later, when the main server was rebooted, data synchronization took place in the wrong direction in that both servers were synchronized with the flights for the previous week, thus resulting directly in the crisis at the airport.

Recovery-control mechanisms must enable the administrator to avoid human error, which a simple replication solution does not offer.

In case of failure, replication should not result in the loss of data about already registered travelers

This important requirement is not met with the asynchronous replication traditionally implemented by replication solutions! Take care and check that the replication solution is synchronous.

In fact, in case of failure with an asynchronous replication solution, you have to locate the passengers registered for a flight prior to the failure, but whose details are not saved in the backup server database due to asynchronous replication. These passengers' reservation is lost after the failure, and their seats are again free in the reservation system!

Data replication on the backup server must be synchronous to ensure high availability

With a synchronous replication solution, data is not lost in case of server failure. In fact, at the check-in counter, a passenger is checked in by reserving a seat for him or her on the aircraft. The reservation request is sent to the main server, and the passenger-management application permanently stores the reservation-related information on the local disk.

In synchronous replication mode, when a piece of information is permanently stored on a server's local database, it is also stored on the remote database.

Thus, when the check-in counter receives the acknowledgement of reservation and the attendant releases the passenger by issuing him or her a boarding pass, the reservation is permanently recorded in both servers' databases (this is not the case with asynchronous replication).

If the main server fails, the passengers currently being checked in are put on hold because the attendants can no longer contact the reservation application. Once the application is restarted on the backup server, the attendants can restart checking in the passengers on standby. Still the reservations for the passengers checked in prior to the failure are not lost since they had been saved in the backup-server database.

It must be possible to combine high availability with disaster recovery

High availability requires the presence of a synchronous replication system between two servers. Both servers must be placed on the same LAN for two reasons. The first reason is that the LAN's bandwidth and latency determines the synchronous replication performances. Secondly, having two servers on the same LAN ensures service IP address switchover when the application is restarted on the backup server.

Disaster recovery and high availability can be ensured simply by extending the same LAN with fiber optic cable in two geographically remote computer rooms. Thus, it is possible to simultaneously ensure application high availability and disaster recovery with two servers in two computer rooms and a real-time data replication solution across the network.

When data must be replicated via a low-speed WAN data replication must be asynchronous, but the asynchronous replication system does not offer high availability: data is lost in case of failure. Asynchronous replication must be compared to a remote backup via the network and not to a high-availability system.

Does the high availability solution resists to "split brain" without corrupting data?

Split brain occurs in situation of network isolation between two servers. Each server becomes primary considering that the other has failed and runs the application. During split brain, some high-availability solutions can corrupt database with two active applications on the same database. It is necessary to avoid the double execution by testing an external network equipment acting as a witness between the two servers.

Thanks to this analysis using the example of the airport, it is now clear that a simple data replication solution is not a high-availability solution. We have seen the pitfalls in case of server failure and we advise the reader to check that a replication solution actually meets the high-availability criteria that we have just explained.

Many products make out that they perform data replication and offer high availability of applications, whereas in reality they only implement data replication and are very incomplete in terms of recovery in case of failure. With this type of product, an IT department insidiously believes it has a redundant high-availability solution. But then it discovers the product's limits the day a problem, such as a server blocking in its booting process, turns into a generalized crisis, with a day of service unavailability for an entire airport!

SafeKit is the ideal solution for high availability of critical applications. It is a comprehensive and simple product. It was chosen after the crisis at the airport and has advantageously replaced the competitor's solution that had led to the critical situation described in this document.

More information on our web site

- High availability architectures and best practices
- Software clustering vs hardware clustering
- Shared nothing vs shared disk cluster
- Virtual machine HA vs application HA
- Byte-level file replication vs block-level disk replication
- Synchronous replication vs asynchronous replication
- Software high availability cluster vs fault-tolerant system
- How to implement redundant servers with a simple software (Windows/Linux)?
- Heartbeat, failover and quorum in a Windows or Linux cluster
- How a virtual IP address works (Windows/Linux)?
- What is the RTO / RPO of a SafeKit high availability cluster?
- Alternative to Microsoft NLB with SafeKit network load balancing

3 partner testimonials

1. The ideal product for a software publisher

Harmonic, TV Broadcasting:

"SafeKit is the ideal application clustering solution for a software publisher looking for a simple and economical high availability software. We currently have more than 80 SafeKit clusters worldwide on Windows with our critical TV broadcasting application through terrestrial, satellite, cable and IP-TV. SafeKit implements the continuous and real-time replication of our database as well as the automatic failover of our application for software and hardware failures. Without modifying our application, it was possible for us to customize the installation of SafeKit. Since then, the time of preparation and implementation has been significantly reduced."

2. The product very easy to deploy for a reseller

NOEMIS, value added distributor of Milestone video surveillance solutions:

"SafeKit by Eviden is a professional solution making easy the redundancy of Milestone Management Server, Event Server, Log Server. The solution is easy to deploy, easy to maintain and can be added on existing installation. We have assisted integrators to deploy the solution on many projects such as city surveillance, datacenters, stadiums and other critical infrastructures. SafeKit is a great product, and Eviden provides great support. Happy to help if you have any questions."

3. The product to gain time for a system integrator

Atos, BU Transport:

"SafeKit is a simple and powerful product for application high availability. We have integrated SafeKit in our critical projects like the supervision of Paris metro lines (CCR, centralized control rooms). Thanks to the simplicity of the product, we gained time for the integration and validation of the solution and we had also quick answers to our questions with a responsive Eviden team."

10 reasons to choose the SafeKit clustering software

1. Software-only high availability solution

Eviden SafeKit is a software-only high availability solution. This solution secures easily and quickly the 24x7 operation of your critical applications. While traditional high availability solutions are focused on the hardware failover of physical servers, SafeKit has chosen to focus on the hardware and software failover of critical applications.

2. High availability which targets all types of failures

The unavailability of an application can be due to 3 types of problems:

- Hardware and environment: including the complete failure of a computer room (20%).
- Software: regression on software update, overloaded service, software bug (40%).
- Human errors: administration error and inability to properly restart a critical service (40%).

SafeKit addresses these issues, which are all essential to ensure the high availability of critical applications.

3. The 3 best use cases of software clustering

After over 20 years of 24x7 experience, SafeKit is the preferred clustering solution on the market in three cases:

- A software publishing company can add SafeKit to its application suite as a software OEM high availability option.
- A distributed enterprise can deploy a high availability solution on standard hardware without the need for or IT skills.
- A data center can provide high availability for multiple applications with a uniform solution on Windows or Linux and with load balancing, real time data replication and failover between two remote sites.

4. Unique on the market: 3 products in 1

Traditionally, three different products are necessary to create an application cluster:

- load balancing network boxes,
- disk bays replicated synchronously on a SAN for data availability,
- high-availability toolkits for application failure recovery.

SafeKit offers these three features within the same software product.
To further reduce implementation costs, SafeKit runs on your existing physical or virtual servers and with the standard editions of OS and databases: Windows, Linux, Microsoft SQL Server, Oracle, Firebird, MariaDB, PostgreSQL or other databases or flat files... and even with Windows editions for PCs!

5. A solution suited for Cloud environments

Application high availability with SafeKit can be deployed in AWS, Azure and Google clouds as well as on premise on physical or virtual machines. Redundancy of Docker applications is also supported.

6. Full virtual machines replication and failover

SafeKit also offers replication and failover of full virtual machines between two active Hyper-V or KVM physical servers. The solution is simple and economical because it requires no shared disk.

7. Plug and play deployment of a software cluster

Once a failover module is configured and tested for an application, deployment requires no specific IT skills. Just install the application, the SafeKit software and the failover module on two standard Windows or Linux servers.

8. Rich choice of application integration inside a software cluster

SafeKit proposes different types of software clusters. Cluster configuration for a given application is rich and is made with one or several application modules. SafeKit proposes mirror modules (primary/secondary with replication and failover), farm modules (network load balancing and failover), and mixed of several modules that can be implemented on the same cluster or on different clusters.
A module is configured with the server IP addresses for heartbeats, the virtual IP address of the cluster, the load balancing rules for a farm module, the file directories to replicate for a mirror module, the hardware and software failure detectors and the service to restart in case of failure.

9. User-friendly administration to avoid human error

SafeKit provides a centralized administration web console. An administrator can remotely monitor status of applications on different clusters and act with simple buttons (start, stop).
You can test SafeKit for free. In less than 1 hour, you can implement your first software cluster on two virtual or physical machines thanks to the administration console.

10. Synchronous replication for transactional applications

SafeKit's synchronous real time replication function strengthens high availability and prevents data loss. With this mechanism, a data committed on a disk by a transactional application is replicated on the secondary machine.
Application servers can be located in geographically remote computer rooms through an extended LAN to withstand the loss of a full room.

About Eviden

Eviden IAM is the Identity and Access Management (IAM) software suite of Eviden.

Eviden IAM is the European leader in identity and access management with a presence which is growing rapidly beyond Europe, particularly in Japan and the US.

More than 5,000,000 users in more than 900 organizations throughout the world connect to their companies every day and manage their access rights with Eviden identity and access management solutions.

For more information: SafeKit web site

CT-210325-JR-HIGH-AVAILABILITY-CHOOSING-SOLUTION-EVIDEN-ENG