

# How Smart Database Replication Compares to Netezza Replication Services

Author: Roy Hammett

---



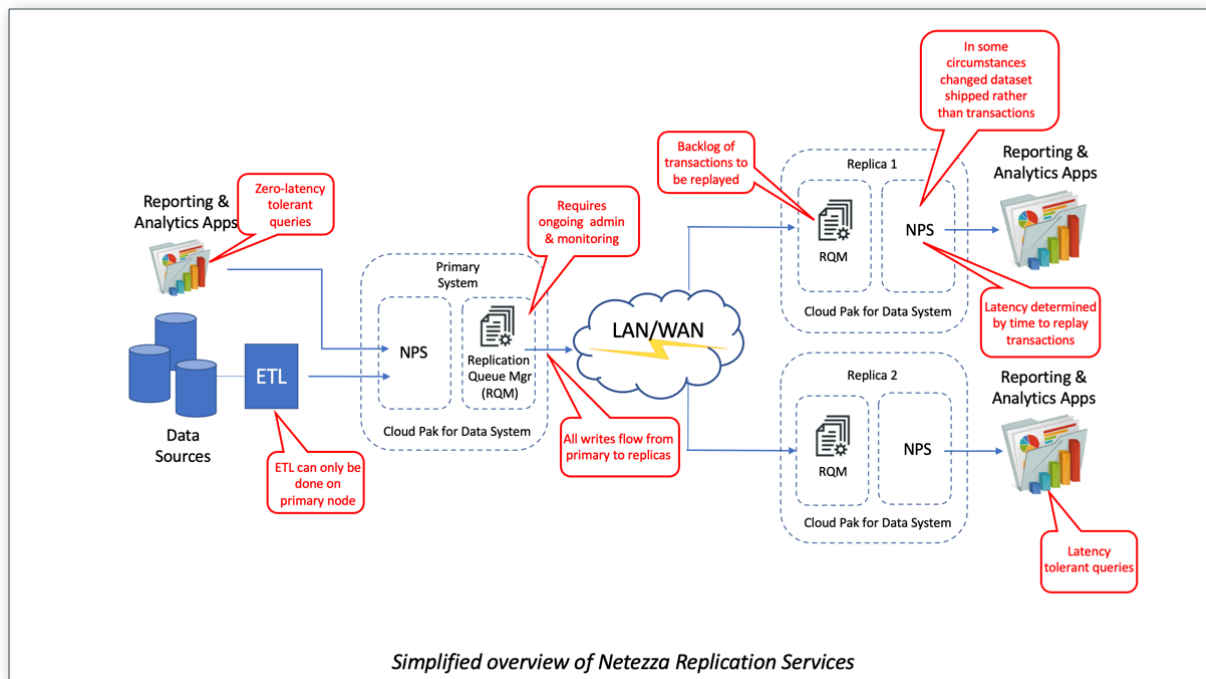
When we heard that IBM was re-engineering Netezza Replication Services for the Cloud Pak for Data (CP4D) platform we were curious as to whether it would include any new functionality, including the points of difference that our Smart Database Replication product boasts.

The good news for Netezza users is that [Netezza Performance Server](#) version 11.2.2 now includes Replication Services, and this means that customers that have two Cloud Pak for Data systems can use one of them as a disaster recovery platform by replicating changes from one to the other using this capability. Although at first glance this sounds very similar to what Smart Database Replication also does, in fact there are many points of differentiation between the two offerings that customers should be aware of before deciding which is the best approach for them. Let's explore some of these.

## What is Netezza Replication Services and how does it work?

IBM Netezza Replication Services protects against data loss by synchronizing data on a primary system with data on one or more secondary nodes.

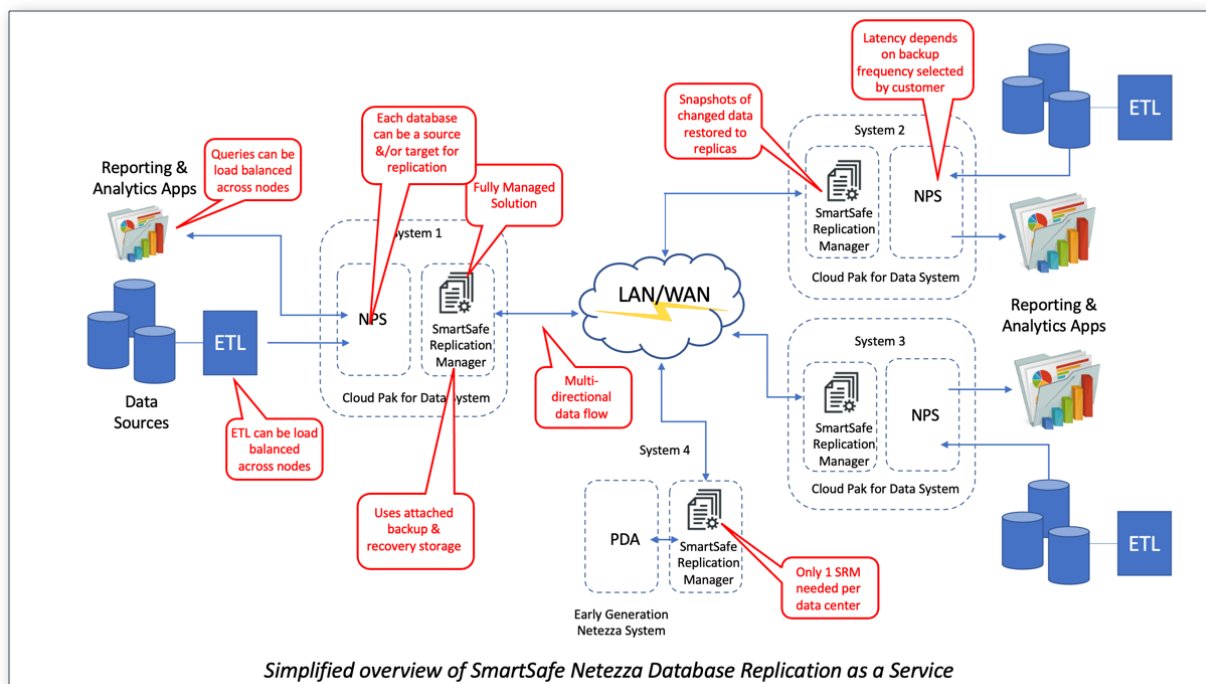
It works by having a Replication Queue Manager (RQM) node – previously a PTS (Persistent Transaction Server) - on the CP4D system which is used as a temporary storage for all transactions that are committed on the primary system. These are then relayed to an RQM node on the replica(s) where they are replayed.



## What is Smart Database Replication and how does it work?

Smart Database Replication also protects against data loss by synchronizing data between primary and secondary nodes but instead of logging individual transactions and replaying them on the replica(s) it takes regular snapshots of changed data and

applies this to the replicas. This occurs outside of the host environment and uses the same network attached storage you already use for backup and recovery purposes, without any additional configuration or software overhead in your host environment.



## How does Smart Database Replication differ from Netezza Replication Services?

We've picked out the main considerations and restrictions from the Netezza Replication Services documentation to help illustrate some of the differences.

1. **Single master.** Netezza Replication Services is omni-directional. You can only have a single primary system in a Replication Services configuration. This means is that you can't divide your ETL workloads and users by having multiple primaries replicating to multiple replicas across multiple Netezza servers in your environment. Furthermore, all your ETL activity will need to occur on the primary node. As far as we are aware, there is no way of shifting some of that workload elsewhere and still have database replication for all users and all data thereafter. You can, however, divide users between primary and replica servers.

*Smart Database Replication* database replication, on the other hand, is bi-directional (BDR) and allows all Netezza nodes to be primary systems. An example of this is where you have two or more systems and each of them can be the primary system for a set of users, one or more applications and one or more source systems, and you can replicate the data between them all, including when you have a mixture of early

generation Netezza and newer CP4D systems. This gives you a natural way of dividing your users and load balancing workloads across your infrastructure whilst still being able to have replication across all those different systems. Moreover, in a multi-master scenario, each master also potentially acts as a disaster recovery site for one or more of the other masters.

2. **Eventually Consistent Design.** Although Netezza is really designed for bulk set operations at-a-time rather than record-at-a time OLTP workloads the reality is that a lot of customers and particularly customer ETL tools will generate single row inserts, updates or deletes. Netezza just handles this but if you are using replication services, all of these transactions have to be captured on the RQM and then replayed on the replica(s).

A transaction must be committed on the primary before it can be shipped to the secondary environment but whereas the primary can perform many transactions in parallel, replicas perform the replicated transactions serially in the order which they were committed on the primary. So, the time it takes to complete a set of concurrent replicated transactions on the primary node is generally the time that is taken by the longest transaction. However, the time it takes to complete these same transactions on a replica is the sum of the times to execute each transaction.

*Smart Database Replication* does not accumulate transactions for replaying on the replica(s). By taking snapshots of the changed data, all the transactions that were processed during the selected interval are shipped as a bulk set operation, which is how Netezza prefers to process data and so *Smart Database Replication* avoids the latency that is an unavoidable feature of Replication Services

3. **Latency.** A consequence of eventually consistent design is that there will be times when the primary node has newer database content than the secondaries. This has two main implications. Firstly, if you are dividing your user groups between your primary and replica servers, you need to ensure that users running queries that must run with zero latency (i.e. real-time) are working off the primary and that those using replicas are aware of the latency. Secondly, if there is a system failure that necessitates failing over to a replica, there may be a considerable time lag before the data that was committed on the primary is replicated to the replica. Furthermore, given that it may be rolling back a significant number of uncommitted transactions it may take some time before you can know exactly how much data was lost during the fail-over event.

With *Smart Database Replication* you decide how much latency is tolerable for each replication set, and how frequently you want to replicate deltas between primary and secondary servers, so users can potentially run all their queries on any node. Furthermore, the fail-over or Mean Time to Recovery is as long as it takes to re-point your DNS servers to the DR system and the Recovery Point Objective is also measured in the minutes since the last delta was shipped to the replica(s).

4.     **Synchronization methods.** Replication Services uses two approaches to maintaining synchronization: by-SQL replication and by-value replication. The method that is used is based on whether the SQL transaction contains a deterministic value. That is to say, does the SQL statement return the same results regardless of where it is executed (by-SQL), or can the query return different results when run on different hosts, for example due to network crossing time and time zone differences (by-value). The impact here is that in large Netezza implementations that span time-zones or where there is high latency, SQL programmers must understand the difference between the two when writing queries to ensure that the results can be fully synchronized between primary and replica nodes.

Because Netezza stores temporal data types (e.g., DATE, TIME and TIMESTAMP) as time plus offset, underlying data will match regardless of which time zone the replica resides. Therefore, *Smart Database Replication's* method of backing up and restoring snapshots of changed data means that programmers do not need to modify their SQL to accommodate the considerations and restrictions of replication services

5.     **Managing Replication Services.** By design, Replication Services never loses data, but it does require quite a lot of administration and monitoring because if the RQM server fills up, you are no longer able to make changes to the primary database. This means that customers should manage their RQM environments just as carefully as they manage their Netezza production environments, and they need to make sure there is sufficient space available for the types of workloads that are performed on the primary server. They should set up Netezza event alerts to notify them if there are any problems with the replication process so that they are investigated and resolved as a priority because if they are ignored it can stop the entire primary system from being able to be used.

Bear in mind that there are RQMs not just on the primary environment but also on the secondary environment(s). If you make changes to the RQM node on the primary but don't mirror the change on the secondary node(s) the replication could fail. So, for example, if you increased the log space on the primary to accommodate longer running transactions without making the same change on the secondary

environment(s) the whole replication process would stop if the secondary logs were to be filled even though there were still room on the primary.

*Smart Database Replication* is a fully managed solution. Customers don't have to manage anything to do with database replication themselves - the service provided by Smart Associates includes installation; configuration; initial synchronization; ongoing monitoring; detection and resolution of replication issues; etc.

6. **Cross Database Dependencies.** Replication sets are per database. If you use a view with cross database references in a statement that modifies replicated data, i.e. INSERT or SELECT from a view, if the reference databases are not replicated this statement triggers by-value replication. So, when you execute CREATE VIEW in a replicated database you cannot reference non-replicated objects with these commands. All object references must resolve to an object in a database that is a member of a replication set or a system table.

*Smart Database Replication* can automatically resolve cross-database dependencies where primary and secondary may be dependent on each other and ordering of replication could otherwise potentially otherwise prevent an object from being created.

## Conclusion

Netezza Replication Services certainly provides a disaster recovery capability that would suit some Netezza clients, although customers should not underestimate the effort and complexity required to install, configure, monitor, and maintain the replication infrastructure.

Many of our clients, however, have found that Replication Services does not meet their needs. This includes companies that:

- Want to achieve Mean Time to Recovery and Recovery Point Objectives measured in minutes rather than, at best, the duration of the longest transaction running on the Netezza primary system
- Want to reduce their dependency on skilled Netezza resources with a fully managed solution
- Are running older and newer versions of Netezza and want to replicate between the two
- Are intending to upgrade to CP4D and would like to avoid a drawn-out migration with system downtime

- Want to replicate Netezza for disaster recovery without having to provide a dedicated Netezza system purely for Disaster Recovery
- Want to load balance queries and users across multiple Netezza systems without having to worry about latency
- Want not to have to modify their queries to accommodate the idiosyncrasies of the replication software design

There are more use case scenarios in our [Smart Database Replication FAQ's](#), and if you would like to discuss further with us, feel free to contact us [here](#).

In this blog we have only described a sub-set of the features of the Smart Database Replication product. To find out more visit the Smart Database Replication page on our [website](#) or [contact us](#) for more information.



Smart Associates (Aotearoa) Ltd	Smart Associates Ltd	Smart Associates ApS
203/11 Vinegar Lane Grey Lynne Auckland 1021 New Zealand	Valley View, The Old Quarry Haslemere Surrey GU27 3SS United Kingdom	% 360 Law Firm Lautrupsgade 7, 3. tv DK-2100 København Ø Copenhagen, Denmark
T: +64 (9) 415-8120	T: +44 (208) 133-6008	T: +45 36 98 71 11