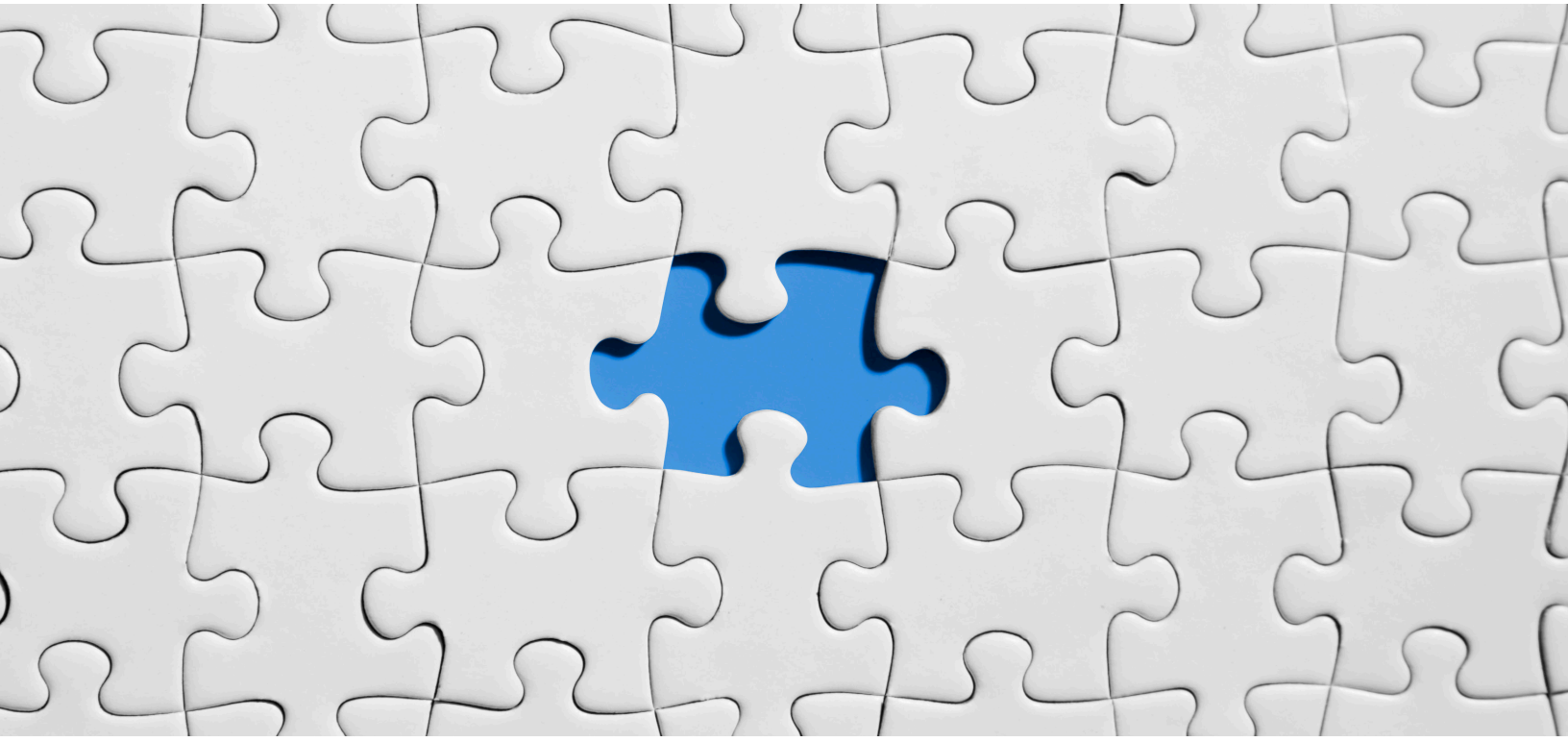
DELL Technologies

TOWARDS STANDARDIZATION OF BACKUP METHODS



Umit Dericioglu

Backup Operations Delivery Team Lead

Dell EMC

Umit.dericioglu@emc.com

Knowledge Sharing Article

DELL Technologies Proven Professional

Table of Contents

Overview

This Knowledge Sharing article provides a quick review of backup methods and definitions, proposals which might pave way to a possible standardization, and suggestions and new ideas to improve backup services.

Why Backup and Some Basic Definitions

Backup is simply the 2nd copy of data. Since backup is taken at certain time intervals, it is a 2nd point-in-time copy. Backup is used to restore data if it is lost or corrupted.

If data is to be restored from the last backup, changes on that data since the last backup to the time of restore would be lost. For example, if a backup is taken daily, maximum loss of changes would be up to a day. If a business requires that the loss of changes not exceed 3 hours, backup should be taken every 3 hours, which is known as **recovery point objective (RPO)**.

Suppose data is lost and we'd like to restore it from backup. How long would restore take? 10 minutes? 10 hours? The answer depends on the amount of data to be restored, backup media and infrastructure used during the restore process between the backup media and the destination where data is to be restored. If a business cannot afford to wait longer than a certain amount of time until data is restored, say 1 hour, that is **recovery time objective (RTO).** The better RPO and RTO required, the more sophisticated and expensive the backup solution and infrastructure.

What if the backup, in other words, the 2nd copy is lost for some reason? Against this possibility, backup administrators may choose to create a copy of the backup (a 3rd copy, or backup of the backup). The 3rd copy is called **clone**.

For disaster recovery purposes, backup medium, as it is being written, can be replicated to a remote backup medium. This is called **replication**. Although there is a cloning process taking advantage of replication capabilities of backup media – called clone-controlled replication (CCR) – clone and replication are not necessarily the same thing. The backup system is aware of the clone. So, if it fails to access the 2nd copy for some reason (the tape is damaged, for example), it can resort to accessing the clone copy (an offsite tape, for example) to do the restore, whereas replication between two media takes place without the backup system knowing about it. If backup data is deleted from the source backup device, it is deleted from the replication destination immediately as long as the replication source and destination are in sync. Therefore, if a backup piece of data is deleted on the backup device for some reason, it cannot be restored from the remote backup device. Replication destination devices at the remote site are used in case of a disaster rendering the local site out of service.

As the 2nd copy of data is created on backup media during a backup, the information about it – i.e. file name, folder name, database table, disk name, etc. and their location in backup media and time/date of backup – are entered in a data base maintained by the backup software. This data base is called **backup catalog**. Backup Catalog is used to retrieve the 2nd copy during a restore job. Once a particular backup expires, its record is deleted from the catalog.

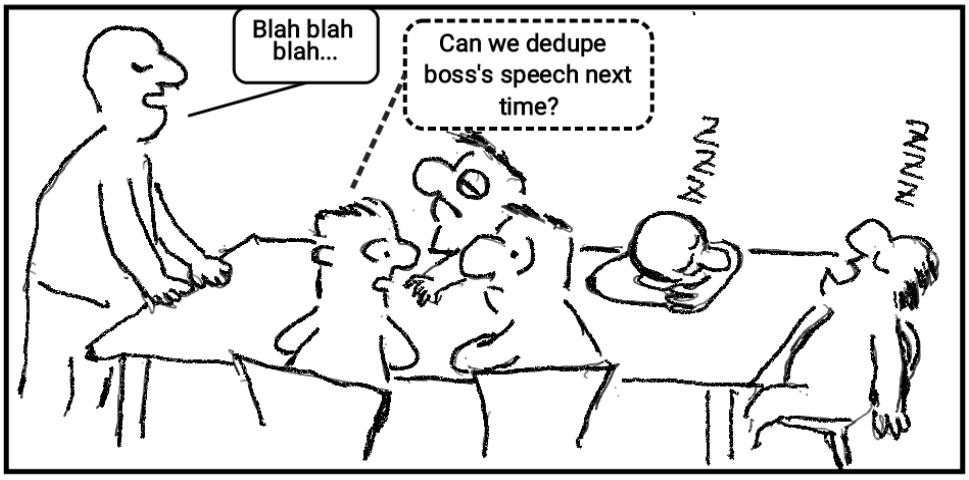
Data piece is a certain amount of data backed up as a unit to be cataloged, be it a file or folder or disk, etc. In some backup solutions data piece is called saveset or dataset.

Data owner is the person or group of people who have the authority to restore data piece(s).

Client is the system or host whose data is backed up.

Backup jobs are scheduled and triggered by the **backup server**. Backup server maintains some databases such as backup catalog, resource data base, etc. It monitors the backup or restore jobs, keeps logs, and reports errors. **Resource data base** has the information about backup definitions such as client info, **retention policy** (how long a backup copy is kept in the backup media before it is deleted and space is reclaimed), scheduled times, backup levels, backup types, etc.

Deduplication is a technique to store a single instance of redundant data regardless of its source. Before deduplication, backed up data used to occupy many times its size on the backup storage depending on the retention policy. Now, with deduplication, it is the other way around; it is not unusual for backed up data to take up 10 to 40 times less space.



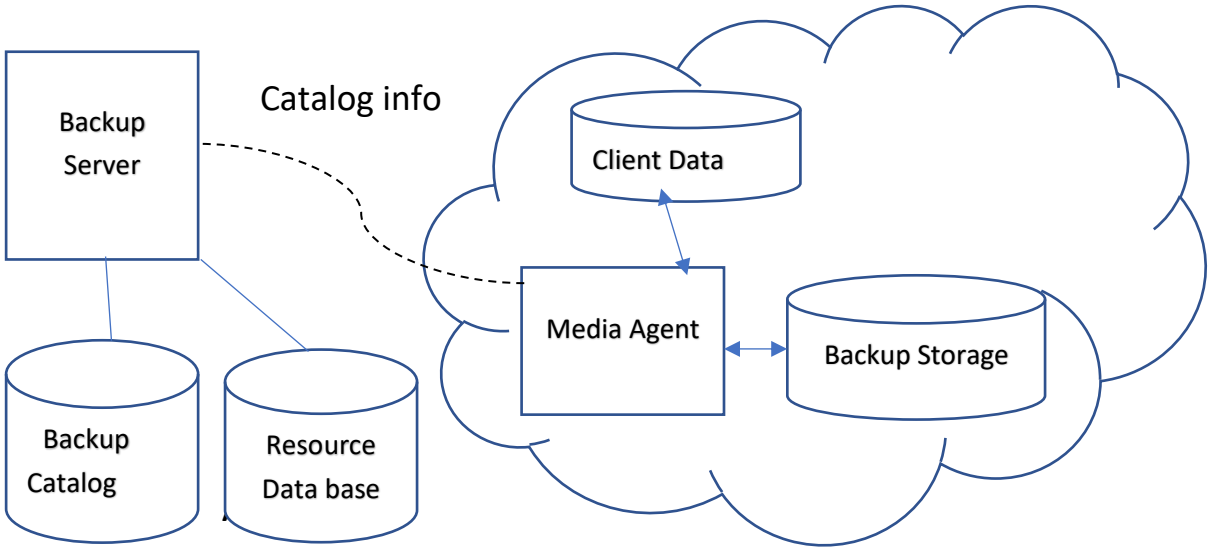
Why Backup Standardization?

There are many standards and protocols in networking, database, web presentation and applications. Even though there have been many backup solutions in the market for last few decades, standards and protocols are missing in the backup arena.

Benefits of standards according to IEEE:

Standards form the fundamental building blocks for product development by establishing consistent protocols that can be universally understood and adopted. This helps fuel compatibility and interoperability and simplifies product development, and speeds time-to-market. Standards also make it easier to understand and compare competing products.[1]

In this article, we propose some ideas towards standardization of backup/restore processes.



A typical backup environment

Each backup solution has backup catalog(s) or database(s) which keep information about data pieces backed up. These catalogs are different and incompatible with each other, yet they have more or less the same information about data backed up. If a company decides to switch from one backup system to another, it is a gigantic and time-consuming task to do the transition, especially when data is backed up with long retention.

Backup catalog can be easily standardized, adhering to certain requirements and format in an SQL data base, for example:

Client	Data piece	Data Piece type	Data Piece owner	Backup location	Backup date & time	Expiry date	Size	….	Optional fields…

[1] https://beyondstandards.ieee.org/general-news/what-are-standards-why-are-they-important/

Standard required fields and their format can be worked out by a standards committee or a task force or by a company which would like to take the initiative.

Optional fields could be left to the backup software developer to add more features.

An Open Standard Resource Database

Similarly, a resource database can be standard as well:

Domain (North Africa, Asia,..)	Category (Critical, non-critical..)	Resource name	Resource type (Group, device)	Resource dependent info (such as clients in a group, backup schedule..)	If resource type is client, its backup catalog name	….	Optional fields..

Again, the details of the standardization can be worked out by the entity who takes up the task.

A Proposed Input/Output Identifier Format

In every backup scenario and solution there is one thing in common: Data is read from the first copy and written to a backup storage as a second copy by some media (I/O) agent and an entry is created in the backup catalog.

We could use a standard identifier for the origin of the data and the location of its second copy on the backup storage.

Since every storage device can now be associated with an IP address, an identifier like this can be proposed:

IP address:port?method:params?data location

where

IP address is the IP address of the media agent. Media agent is usually the same as the client. If not, the IP address of the system to read from or to write to could be specified in params

Port is the IP port where the media agent listens at

Method is either READ or WRITE

Params is where detailed specification of read or write operation and the catalog to be used (when the method is WRITE to backup storage)

Data location or DB name

For, suppose a backup of /xyz file system of a client is to be taken. The identifier could be example something like this:

10.10.11.22:3456?READ:<what device or IP to use for read, etc>?/xyz

This identifier simply means to read folder belonging to a file sytem using a fiber channel device. It tells, in this case, the media agent where to read what on host or 10.10.11.22. Media agent runs on that host and listens at port 3456.

Instead of reading a file system, method could dictate to read from an Oracle DB:

10.10.11.22:3456?READ:rman params?DB name or table name..

Examples so far have been about identifiers for input, i.e. from where to read the first copy of data. Here is an example of identifier to use for output (backup device):

10.10.11.44:6543?WRITE:IP:10.10.11.45,dedupe,<backup catalog info>?/mtree/backup/<client-id>

This identifier instructs the media agent on IP 10.10.11.44, listening at port 6543 to deduplicate data and write it to /mtree/backup/<client-id> on the backup media whose IP address is 10.10.11.45. The client ID is a unique identifier generated by the backup system for the client.

Input identifier to provide data for the above output identifier could be something like this:

10.10.11.55:4444?READ:snap,10.10.11.99,Disk1?/SN/.snap1

READ_SNAPSHOT method takes a snapshot of Disk1 on IP 10.10.11.99 and puts the snap on /SN/.snap1 on the same volume (unless otherwise is specified)

All the methods above may consist of an agent listening at a port on the media agent, and some libraries (for dedupe, snapshot, ...). Or, very simply, a method could be just a command to retrieve or commit a bunch of settings, such as

10.10.33.33:7890?READ:SETTINGS:TAG

10.10.33.35:8890?WRITE:SETTINGS,<catalog specs>:TAG+SETTINGS (restoring to a smart device)

An input and output identifier pair would define a backup or restore job. Certainly, a lot of work is required to standardize the identifiers and methods. Programs running on a client, proxy or even storage with the capacity of media agent would have to adhere to the standard.

Master Backup Record

A master backup record (MBR) is the root or beginning point of a backup server. It could be a simple small file containing information like:

- Location of the resource database
- Location of default backup catalog for clients for which no backup catalog is specified
- IP address of DR backup server
- Location of media agent repository

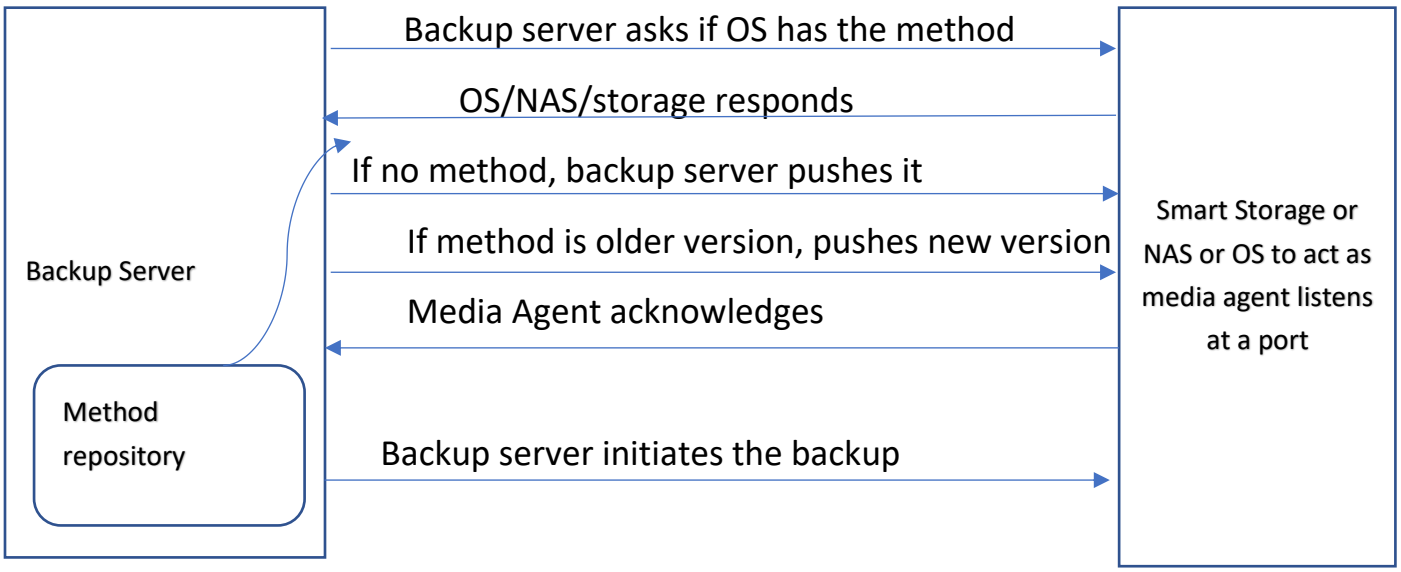
Suppose a backup solution is to be replaced with one from another vendor. The only steps required would be to remove the existing software, install the new one, and point the new backup server to the standard master backup record. The new backup server would come up recognizing all backups, devices, catalogs without any migration effort, thanks to the new standard.

Suppose We Have the Standard, Then What?

As said earlier, developing an open standard for resource database, catalog and input/output identifiers and their methods is no small task. However, once we have them, we'll have the advantages quoted from IEEE earlier:

- A backup product would be easily replaced with another by simply pointing the new product to the backup databases. This would save a huge amount of time and effort and resources for a company unhappy with the backup solution it has and wants to replace it.
- Backup solution providers would be compelled to come up with more robust and competitive products.
- Instead of providing a non-standard, complex software, backup solution providers would compete to provide a more user-friendly, efficient software and better technical support.
- When operating systems, smart storage, and DB systems adhere to the standards, software and libraries for a variety of methods would come with them, rendering the installation of backup agents on clients unnecessary.

In fact, the last point above may lead to a new protocol: OS's, smart storage and DB systems would not have to come with all those methods imbedded. Instead, a protocol might be developed to listen at a certain port through which those methods can be pushed into, say, a smart storage or client OS by the backup server. This way, new and better versions of libraries used for backup methods (specified in the identifier) as they are made available by the backup solution provider can be implemented on the compliant systems:



A Client push feature already exists in backup solutions. However, an agent of the backup software should be installed on the client first. Then, the backup administrator should manually initiate a client push from the backup server to the client and remotely install a new version of its agent.

What is proposed here instead should be a simple protocol where the client (where the media agent would run) would have the protocol embedded in the OS. This way the backup server can automatically check to see if that system has a media agent and would be able to remotely push the media agent and its libraries or upgrade to newer or the latest version, or push the media agent of another backup solution provider.

Each time a backup job starts or periodically, the backup server would check the client to see if an upgrade or replacement is due.

NDMP[2] is used to back up network attached devices (NAS) where a backup agent or media agent cannot be installed on the appliance.

The proposal here for a network backup agent (or media agent) push protocol is an improvement of NDMP. Instead of coming with NDMP capability, a NAS device or a smart storage in particular, and an OS in general can come with the new protocol, eliminating the need for NDMP or agent installation.

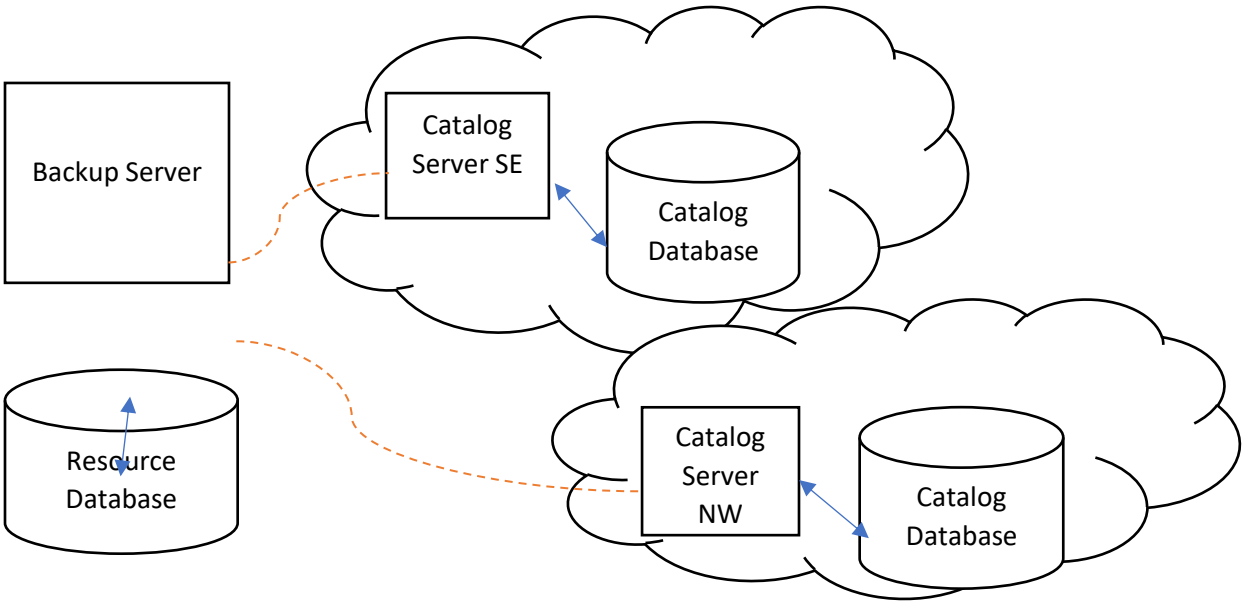
If an OS does not come with the push protocol embedded, backup administrators can build a small binary from a program which would be available in the backup solution with the necessary authentication embedded, which can be installed on the OS. Once installed, it would start listening at a certain port and accept contacts from the backup server. Once the trust is established between the client and the backup server, the agent would be installed. This would be the only instance of the agent being installed on the OS. There would no longer be a need to install backup agents in the future, regardless of the brand or version of the backup solution.

Backup With Multiple Catalogs

Backup catalog resides in the backup server. As time goes by and the number of systems and amount of data backed up grows, the catalog become huge. This makes it difficult for the backup server to handle it and manage the backups at the same time. Therefore, large enterprises typically have multiple backup servers, each one backing up a subset of the systems. This, in turn, increases the complexity, the cost, and reduces the flexibility as it creates “backup islands” which are unaware of each other.

It might be a good idea to have catalog servers reporting to a single backup server instead of having multiple backup servers. Each catalog server would record the backup information of a subset of the entire backup client pool in the enterprise.

[2] **NDMP**, or **Network Data Management Protocol**, is a protocol meant to transport data between network attached storage (NAS) devices and backup devices. This removes the need for transporting the data through the backup server itself, thus enhancing speed and removing load from the backup server. See https://en.wikipedia.org/wiki/NDMP



Catalog servers serving for the backups of clients in 2 geographic locations through the company intranet

This scheme would take a huge burden off the backup server which could now concentrate on managing backups, handling the resource DB and generating logs and reports.

Another benefit would be the ability to restore a client's data on a different client whereas this would not be possible as a backup server would not recognize a client of another in a multi-backup server environment.

Replication-Aware, not Clone-Aware

As mentioned earlier, clone is the backup of a backup. A clone (a 3rd copy) is created in case the backup (2nd copy) is lost for some reason. Clone has been a useful measure particularly for backups on tape. If a backup tape becomes unreadable due to wear and tear, an off-site clone tape would be used to restore data.

With the advent of disk backup storage with RAID, and technologies such as Data Domain Data Invulnerability Architecture[3], tape as a backup storage is fast becoming a thing of the past. Besides, tape storage cannot be used for deduplication. Therefore, cloning is no longer needed in a world of disk storage as backup medium.

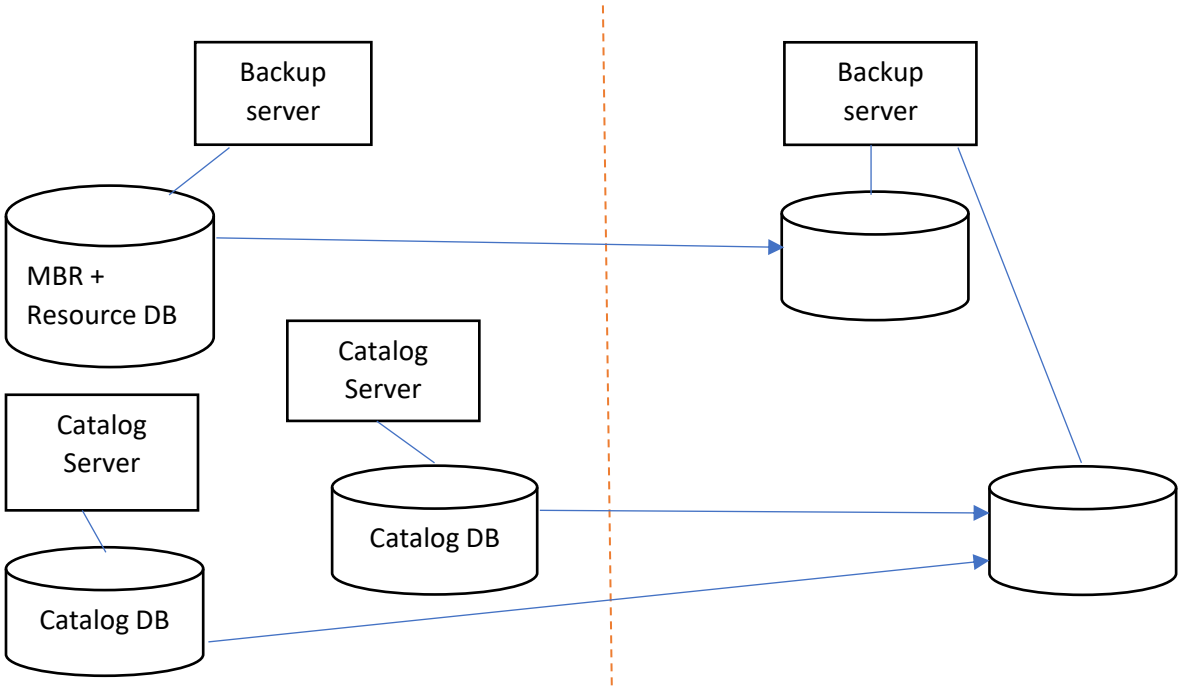
[3] Four key elements of the architecture:

1. End to end verification
2. Fault avoidance and containment
3. Continuous fault detection and healing
4. File system recoverability
   For more info see: Domain Data Invulnerability Architecture: Enhancing Data Integrity and Recoverability, Dell EMC White Paper 2017

The idea of backup without clone may seem a bit extreme or radical, however, backup administrators should ask themselves how many times they needed to restore from a clone copy since they started using disk backup storage.

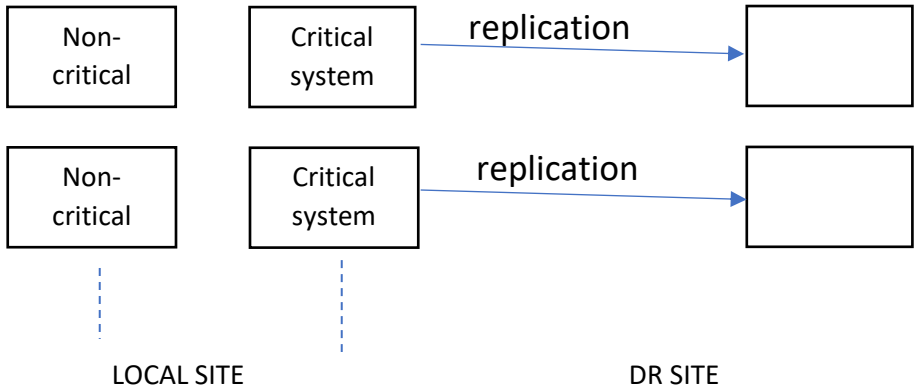
Clone should not be part of a proposed backup standard. Backup solutions without clone capability would be less complicated and less burdened. Instead, a backup server should be replication aware.

As mentioned earlier, backup servers generally are not aware of replication (except a method called Clone Controlled Replication - CCR[4]) and replication is a protection against disasters like complete loss of backup storage in the local site or the loss of an entire local site rather than loss of some backup data. From a backup perspective, a remote DR site should have a backup solution that continues taking backups of critical systems and performs restores as/when necessary. Critical systems in DR site are the systems required for business continuity after losing the local site:



[4] CCR is not a true solution of DR for backup, but a workaround using existing old cloning functionality:

- After backup job(s) are completed clone job(s) are triggered. This is an additional burden and complication on the backup server. Replication should be left entirely to the backup storage.
- Data can be restored on the DR site as the DR site backup server cannot access the 1st copy after the local site is lost, but additional steps should be taken to continue backup operations on the DR site.
- Once the local site is brought back up, it is not easy to incorporate backups done on the DR site, into the local site.



Here, the only critical systems are available on the DR site. For simplicity, all catalog DB's are replicated to a single catalog DB on the DR site backup server. If the local site is gone, backup/restore operations will seamlessly continue on the DR site for the critical systems.

As a part of a new backup standard, DR site backup server's IP address or host name can be a part of the master backup record as well as that of the local site. Similarly, all critical clients would have a DR site IP address or host name in the resource database. Suppose the local site is lost; simply bring up the backup server in DR site and point it to the replicated master backup record. The backup server would first check the IP address or host name in the master backup record to see if it matches the IP address of the DR server. If it does, the backup server realizes that it is running on the DR site and it should back up any (possibly critical) client that has a DR site IP specified in the resource DB which is also replicated.

Resource DB could have a resource such as "Replication" where IP addresses of resource identifiers could be associated with the IP addresses (and maybe ports, too) of replication destination on the DR site:

Local IP	Local port	Remote IP	Remote port
10.10.11.44	6543	10.50.4.4	3456
10.10.11.45		10.50.4.4	

When the backup server is started on the DR site after the local site is lost, for all subsequent backup and restore operations, input/output identifiers would be modified according to the Replication resource in the resource DB because the backup server is now aware that it is running in DR mode. An output identifier such as:

10.10.11.44:6543?WRITE:IP:10.10.11.45,dedupe,<more params>?/mtree/backup/<client-id>

would be converted to:

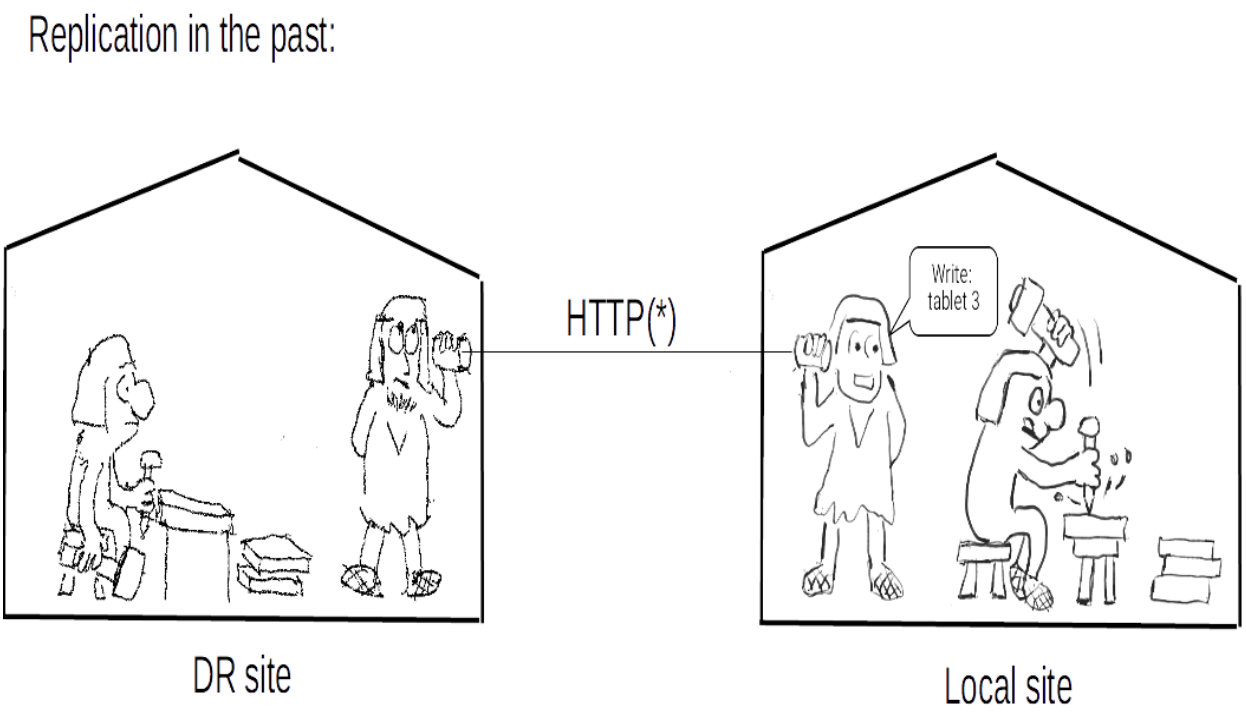
10.50.4.4:3456?WRITE:IP:10.50.4.4,dedupe,<more params>?/mtree/backup/<client-id>

DR site backup server also knows the location of the replicated copies of the catalogs. This means that the moment that the backup server is brought up in the DR site, it is ready to continue to back up the

critical clients in the DR site and ready to restore any data backed up by the backup server at the local site before the disaster.

Since the local and DR site IP addresses or host names are known to the backup server, backup catalogs and DB's can be synced back to the local site once the local site is back up. This process can be done manually on the backup storage or by the backup server automatically if it has the capability to control the replication on the backup storage.

The backup system should be fully replication-aware, not clone-aware and transition of backup operations to the DR site should be seamless. There could be some better ideas to implement this than those proposed here. The objective is to make the backup solution fully DR/replication-aware and eliminate clone functionality completely for simplicity and efficiency.



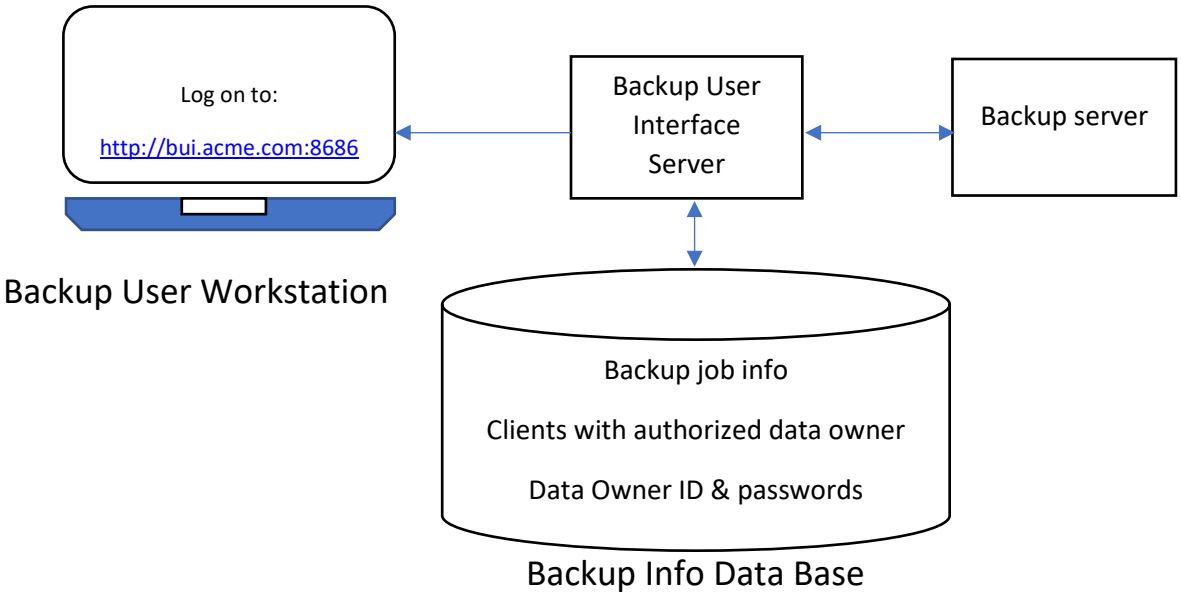
A More Data Owner-Oriented Backup Solution

The way a data owner expects to see the info about backup/restore could be very different from that of backup administrators who have a more technical view of the backup environment to ensure the expected backup/restore performance and troubleshoot problems. Implementation of new backups by the admins and testing are also of a very technical nature.

Data owners, however, have a more administrative view. They would like to know if their data is protected. They would like to be able to have some reports about backups as their management or audit department requests. Sometimes a team is replaced with a new one and because of a poor or hasty handover, the new team may even wonder if their systems are backed up.

For example, in terms of generating reports, data owners may need a report covering a year while backup administrators can only generate reports up to 6 months because of retention policy. In big enterprises, many departments may have different expectations.

That is why a separate server providing a data owner or backup user view is proposed here:

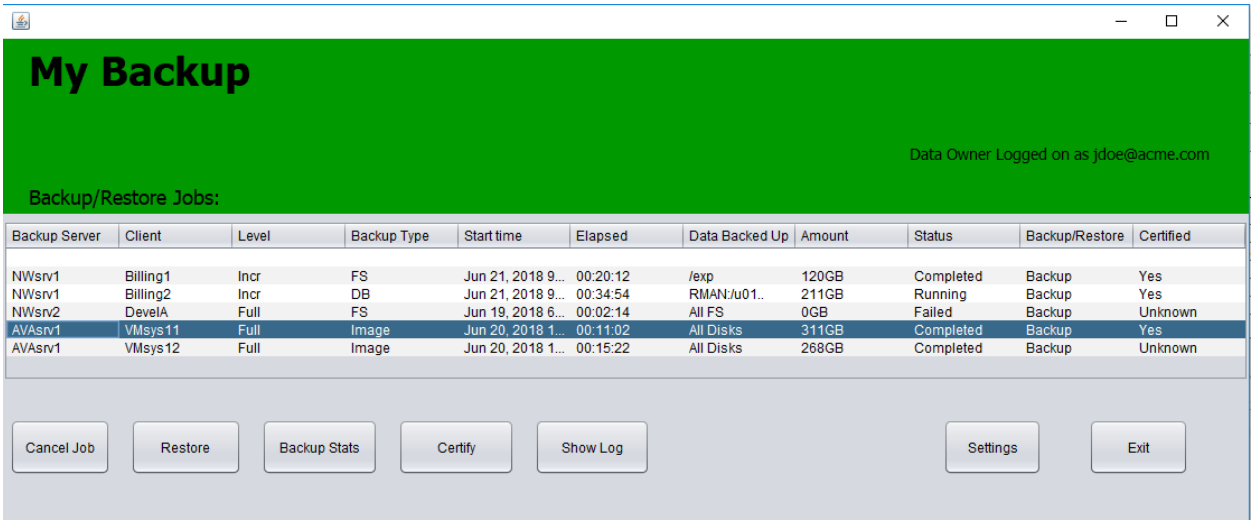


Backup user (data owner) enters http://bui.acme.com:<port> on his workstation. S/he is prompted with logon ID and password. If they match with a data owner ID and password in the backup info DB, user is logged on. And BUI server checks the DB and finds out which clients this data owner is authorized for. Then, the real time backup/restore info is pulled from the backup server and the past information is retrieved from the backup job information DB.

This server could talk to the backup server with backup administrator authority, yet it would present to the data owner the backups s/he is authorized to see and would keep backup information in a DB according to the settings of the data owner. The data owner can log on from his workstation to the BUI server and see the progress of backup jobs, can extract statistical information, initiate restores and do some settings.

Being able to see the progress of a backup job, for example, is very useful for the data owner. Data owners or system owners sometimes ask for an on-demand full backup before patching up their system or maintenance down time. By watching the progress of backup, they can proceed with their plan as soon as the backup completes.

Data owners, especially in big enterprises, query the backup administrators about backups. Since a user interface would provide all that information, backup administrators would not have to provide the information or reports but rather, concentrate on their work.



In this backup user interface[5], for example, data owner can:

- select a backup job running and cancel it
- select a client and click the “Restore” button which would start a wizard where s/he could choose backup date and restore destination, etc.
- request backup stats for a client or for all systems backed up, such as backup success rate, and total amount of data backed for a period of time
- certify a backup. (This will be explained below)
- show log (for a failed backup): This would be very useful to fix failed backups. Since the majority of failures are caused by problems outside the backup solution, data owner could check the error messages and fix most of them quickly, before being contacted by a backup administrator
- do some settings, i.e. whether data owner should be notified only when backup fails or always or not at all, retention period for backup statistics, etc.

Backup Certification

What is the point of backup if data cannot be restored satisfactorily at the time of need? The data owner is the one who should decide whether a restore operation is satisfactory. The only way to find that out is to test restore. A restore test:

- shows that data is restorable from the backup. Yes, theoretically it is possible to restore, however no one knows if this is the case in practice until it is tested.
- uncovers potential problems during restore. Since backup/restore process touches almost everything in a data center infrastructure, many unexpected problems may occur. Restore test provides an excellent opportunity to iron out those issues before they show up during an urgent restore of lost production data.

[5] In BUI example above, there are multiple backup servers. In fact, there should be one server if backup solutions are standardized. In that case backup server column can be removed. Nevertheless, BUI is a very good idea even if there is no backup standard and there are multiple backup servers.

- reveals the restore performance (remember RTO?). Unless restore is tested, it is not possible to know how long it would really take.

With the advent of virtualization, restoring production data to a test system without touching the production system has become much easier. A test VM with similar performance characteristics to the production system can be quickly and automatically provided for the data owner when s/he decides to test restore.

In this context, it is said "**A backup is certified**" once the data owner is satisfied with data restore. Therefore, a backup can be:

- **Certified**: After a successful restore test, if data owner is satisfied with the results, clicks "Yes" for certified
- **Not Certified**: Data owner is not satisfied with restore test and clicks "No"
- **Unknown**: Data owner did not test restore or did not click yes or no for certification

If certification status is "unknown", backup system may send periodic messages to the data owner, reminding that backup of his data may not be reliable unless it is tested and certified.

If certification status is "Not certified" (data owner is not satisfied with restore test), backup system may send a message to the data owner, asking to contact backup team or backup team may contact data owner to solve the problems or discuss options for alternative backup solutions.

Depending on the criticality of data backed up, certification status can periodically be reset to "unknown", urging the data owner to test restore again.

Certification status and its properties such as warning period, etc. can be kept in the backup info DB of Backup User Interface server.

Group Restore

Although many backups can be run in a group, restore has been an individual concept of traditional backup scheme. A restore job is run to restore some data from backup to a single client's disk.

A protocol can be developed to do multiple restores to multiple clients at the same time as a group. This protocol could be very useful as the Internet of Things (IoT) becomes ubiquitous. Let's explain by an example:

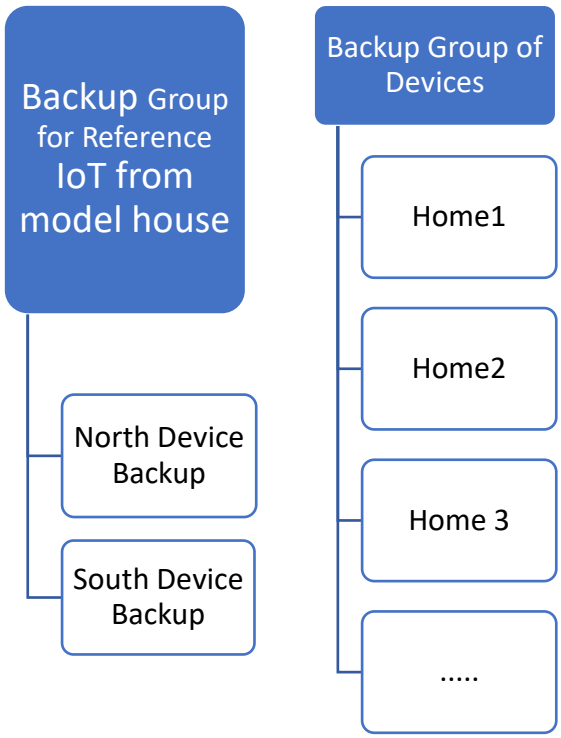
Suppose a builder makes smart homes controlled by things, i.e. smart devices on the Internet. Let's say there are 2 types of homes: North and South, each with IoT with different settings. The builder can test, optimize and set them on a model house and back them up in a group. Now there are 2 backups for settings in the group: devices for South homes and devices for North homes.

The builder can start a group restore job for all newly built homes by checking their IP addresses. After authorization check takes place, the "things" would respond to checks with a "tag" containing information about themselves. For example:

Acme Corp. climate control model: xyz User info: South

Backup system compares the tags with the tags backed up along with settings earlier. If there is a match, settings are restored to that device from that particular backup.

Thus, devices, i.e. IoT of all smart houses built, would be initiated with proper settings:



A restore job can be initiated with this command:

Restore from "Backup Group for Reference IoT from model House" to "Backup Group of Devices"

- Backup server sends an identifier to the IP addresses of each home (home1, home2, home3, …. requesting their tags:
  10.10.11.44:6543?READ:SETTINGS,<more params>?TAG
- home devices send their tag information
- Backup server compares tags to the tags of North and South device backup
- If there is a match, that backup is restored to that home's device:
  10.10.11.44:6543?WRITE:SETTINGS,<more params>?TAG+SETTINGS
- If no match, go to the next home and compare its tag until the list of homes in the "Backup Group of Devices" are exhausted.

Similarly, a company having smart offices worldwide with all sorts of settings backed up, according to the geography, culture and climate, etc. may do a group restore to do the settings when opening a new office or replace the settings with modified ones in a location. Or, if settings are lost due to a disaster, once the facilities are back up, settings can easily be restored with a group restore job.

Final Words

The suggestions for a backup standard and protocols and proposal of some new ideas in this article are based on the experiences of a backup admin. Obviously a backup admin's experience is only a small subset of the experiences in the world of backup solutions. Therefore, some ideas presented here may seem far-fetched, unrealistic or impractical. The main point of this article is to emphasize the importance of developing a backup standard with some protocols and start a discussion towards that goal.

Certainly, many backup admins and backup users could come up with alternate ideas which would be better suited. In my opinion, backup users and admins around the world should come together and work towards a standard. This work can be initiated and sponsored by an international standards body or a company.

A company taking this initiative will be the first to produce a backup solution adhering to the new and modern standard and will be a step ahead of its competitors.

Most important, customers will benefit from much better, data owner-friendly, efficient backup solutions which are more tuned to the 21st century.