odaseva

6 Concepts to Archive Salesforce Data at Scale

Authors:

Sovan Bin
CEO & Founder, Odaseva,
Salesforce CTA

Melissa Shepard
CEO & Founder, Lizztech
Consulting, Salesforce CTA

Cem Yöndem
VP End to End Digital Customer
Relationship Management,
Schneider Electric

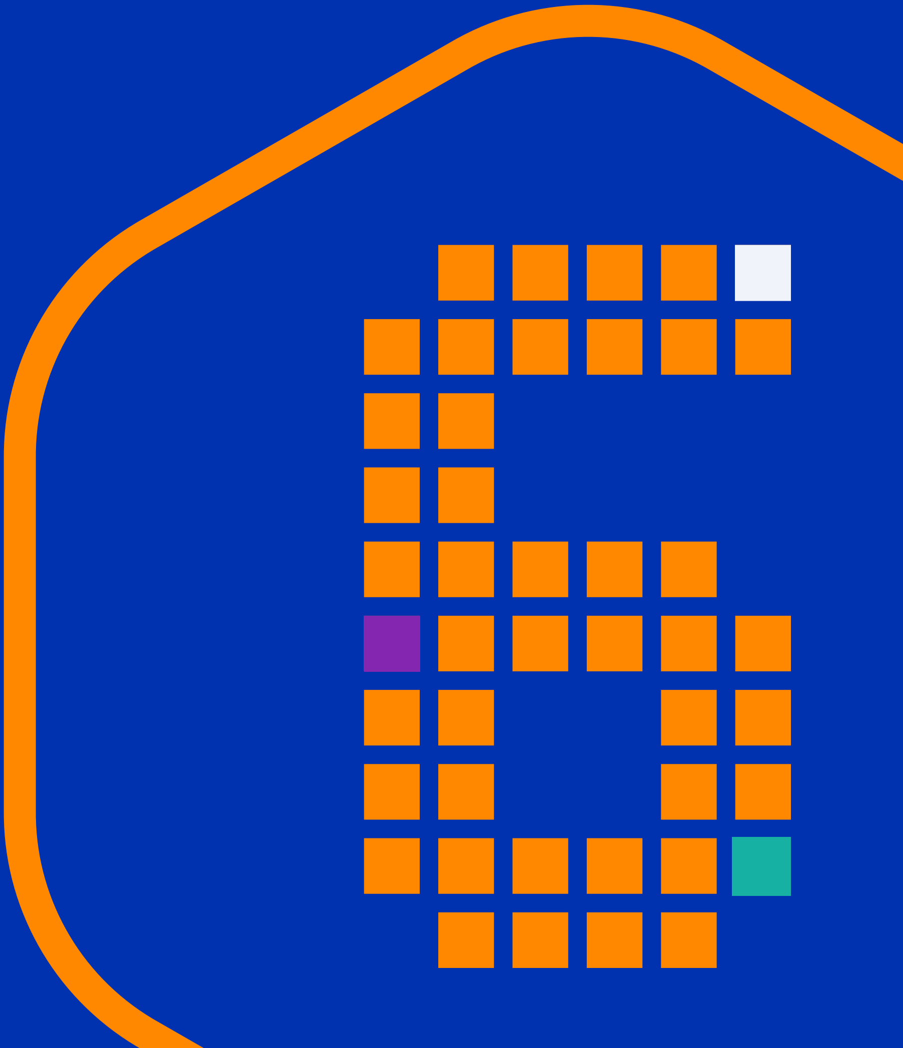


TABLE OF CONTENTS

Executive Summary

For large Salesforce Orgs, implementing an effective archiving strategy is mandatory. Although Salesforce scales with usage, it does not scale indefinitely. As the data model grows in volume and complexity, the only solution is to remove data while preserving the user experience.

The time to implement data archiving is before data scale affects platform efficiency. A proactive approach involves aligning the data archiving strategy to the enterprise's specific needs.

This whitepaper details why large Salesforce Orgs must archive data and the six related architectural concepts:

Data depth

Data dependencies can be many levels deep. A parent object may have multiple children objects which, in turn, may have their own children. During data extraction it is important to retrieve every related record at all levels.

Data volumes

Large Data Volumes (LDVs) are not only about how much data you have, but also how much data you continue creating. Two archiving strategies, steady-state and backlog, solve different LDV challenges.

Data deletion

A delete operation performed while archiving can have unforeseen consequences. This includes cascade delete of related records or leaving orphaned records. The impact caused by data model relationships must be considered before deleting records in the archiving process.

Data drift

The Salesforce metadata underlying a data model, and its UI representation, changes over time. The archiving process must have the capability to self-discover these changes and adapt. Being drift-tolerant avoids errors and gaps when archiving data, and a virtualization layer that adapts to UI changes allows users to work uninterrupted.

Data retention

Your data retention policy may be defined by the classification of data, regulations, and other factors. Archived data can be corrupted over time, and so one of the most important KPIs when archiving LDVs is data durability. How you plan for corruption, and how you rectify it, defines your data durability percentage.

Data encryption

Data encryption provides continuous protection when data resides in Salesforce, in transit, in processing, or at rest in a stored archive. It's a key security control when protecting the confidentiality of your information.

You will also learn about Schneider Electric's use of Odaseva Data Archiving, which incorporates these concepts and supports the most advanced Salesforce data archiving use cases employed by large enterprises.

Why archive Salesforce data?

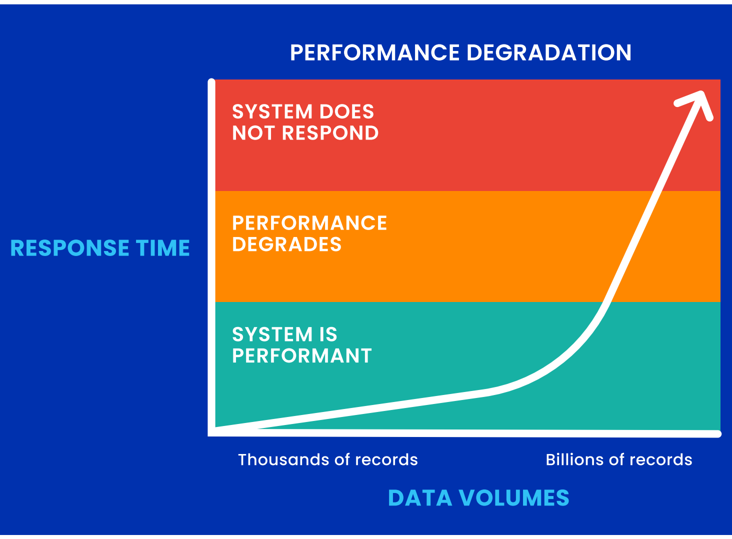
Salesforce data is created from many sources: integrations with other systems, from users, and from customers. Over time, the volume of this stored data increases. If your adoption of Salesforce is successful, a massive volume of data will be created, causing exponential growth in data volumes which must be stored and managed.

The solution is to extract and delete data from Salesforce while preserving the user experience with archiving.

Signs you are experiencing poor system performance include:

- Longer search, report execution, and calculation times
- Exceeding governor limits including storage limits
- Slower processes like integrations, data protection processes, and data cleaning

Business users want all their data, but Salesforce data growth leads to performance degradation



Enterprises archive Salesforce data for:

1. **Compliance with regulations:**
   Regulations may dictate how long you keep data in Salesforce. For example, some industry regulations may require you to keep data for longer than its business relevance.
2. **Improved system performance (while preserving the user experience):**
   Excessive data report load times and slow processes cause system stability issues. It can even render Salesforce unresponsive - grinding business processes to a halt. Increasing the speed of the system by removing data helps improve the user experience, Salesforce adoption, and reduces business interruptions.
3. **Maximized business relevance**
   Retaining all historical data in Salesforce makes it difficult for users to find relevant data, degrading the user experience. Removing this data maximizes the business relevance of data that is in Salesforce.

6

CONCEPTS TO ARCHIVE SALESFORCE DATA AT SCALE

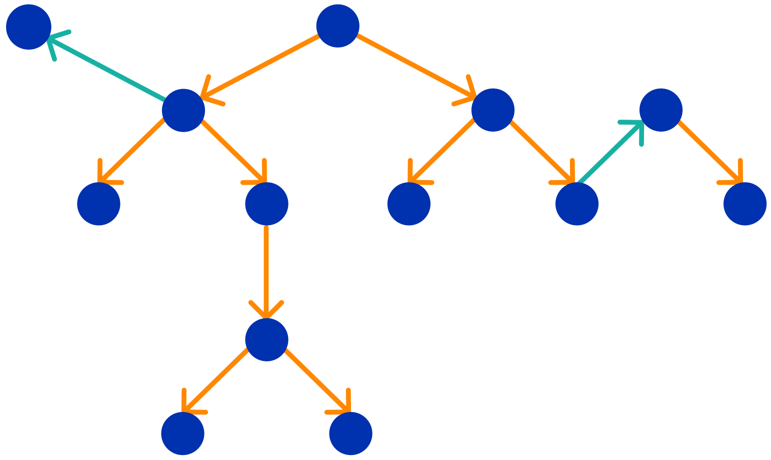
01 DATA DEPTH

Parents, children, and more

The depth of relationships in a data model extends beyond just the parent and children records. The relationships can stretch to 10, 20, or more levels deep. With Schneider Electric for example, the depth of relationships went up to 30 levels.

After creating a complete data archive, the records will be deleted. Deleting data will impact the related parent and children records at all relationship depth levels. It is necessary to retrieve all related records to ensure the complete data integrity of the archive.

Relationships between objects in a data model can either go up or down. A parent-child relationship goes down the object model with the ability to have multiple related records. When viewed from the other direction, a child-parent relationship goes up the object model to the related record. The consequence of this is that relationships in Salesforce can be seen in both directions, which means that if you delete a parent you can create an orphan, and if you delete a child you might miss some information on the parent record.



Data dependencies extend up and down the data model, multiple levels deep

When using SOQL to query relationships, there are limitations. For example, no more than seven levels can be queried in a single query (current object + five levels up-queries + one level subqueries). In order to have a complete archive, these limitations must be addressed.

02 DATA VOLUMES

Steady-state and backlog archiving

When it comes to archiving, LDVs are not only about how much data you have, but also how much data you continue creating.

Two archiving strategies, steady-state and backlog, solve different LDV challenges.

Steady-state archive demands resilience

Steady-state archiving is an ongoing process, where you regularly archive as much data as you create to maintain the same volume of data.

When data is archived constantly, typically on a weekly or monthly basis, the process must be resilient. It should adapt with changes in the source system. It must also provide administrators with monitoring and log management capabilities to detect errors and take corrective action.

Backlog archiving demands performance

If you wait too long to implement an archiving strategy, millions of records can pile up over the years. Backlog archiving is a one-time event that removes this large volume of data.

Backlog archiving involves breaking down large volumes of data into manageable chunks because otherwise you may considerably increase resource consumption like API calls, exceeding governor limits that will slow down or even halt Salesforce performance.

You therefore need a high-performance solution that will archive the backlog quickly and efficiently at scale to ensure that millions of records are archived, while avoiding negative impacts to system performance or accidental data loss.

Archiving Large Data Volumes intensifies challenges

When facing LDVs, one may immediately consider using the Bulk API. However, other limitations could arise:

- ✓ The archive might require both object records and files, whereas querying the actual files from Salesforce is not supported by the Bulk API, requiring the REST API to be used.
- ✓ Bulk Governor Limit might be insufficient and ongoing day-to-day operations should not be impacted by archiving large volumes of data.
- ✓ Data Depth might require to run complex up and down queries that are not supported by the Bulk API.

03 DATA DELETION

After successfully extracting data, an archive process should delete the data from Salesforce. This can have unforeseen consequences if not properly planned. You should have a thorough understanding of how to delete data, and the resulting impact on record relationships.

Deleting data as part of the archiving process can have a significant impact on the integrity of your data model. While the Recycle Bin captures deletes for a short period when archiving Salesforce data, it does not capture everything. To ensure that you can completely restore your data, it's crucial to have a backup as close as possible to the point of deletion to provide a reliable recovery point.

Before deleting any data, you should perform a data model analysis to assess the impact of delete operations on parent-to-child and child-to-parent relationships. This will help you make informed decisions and maintain the integrity of your data.

Salesforce has two types of relationships with different behaviors when deleting data:

- ✓ Master-detail relationships will cascade the delete operation to related child records.
- ✓ Lookup relationships will set the related field to a null value, creating an orphan that no longer relates to the parent. This orphan record is not captured by the Recycle Bin and can be missed in restore operations if not considered.

Handling cascade deletes

‘Cascade delete’ of related records can be your biggest ally or worst enemy. Passing a single record id to Salesforce for deletion can result in dozens, hundreds, or even thousands of related records also being deleted. Leveraging cascade delete provides an efficient way to minimize API calls. However, once you encounter a ‘Delete Operation Too Large’ error, more control is needed.

This is where an architect requires a powerful solution to optimize delete batch size and control impacted related records. Instead of relying on cascade delete, the relationship hierarchy needs to be unwound and it would be recommended to delete records object-by-object from the bottom up.

Handling orphaned records

When a lookup relationship is used, deleting a parent record does not typically delete the child record. Instead, the lookup field in the child record will be emptied when the parent is deleted. This relationship with the parent is then lost and the child record becomes an orphan record. Although rarely enabled, lookups have an option to ‘Cascade delete on custom lookup relationships’ which causes cascade delete behavior.

For example, consider a custom Survey object with a Case__c lookup field to the Case object. When the parent Case record is deleted, the Survey will remain with a null value for the Case__c lookup. The business insight, that a customer was satisfied with the support, has now been lost.

As an architect, you have several options:

- ✓ Delete the entire related record with a separate delete operation
- ✓ Keep the record without the lookup
- ✓ Preserve the logic by updating a text field with related information such as the Case Number
- ✓ Recreate a relationship with the archived parent record by providing a URL that redirects to the virtualized archive parent record

04 DATA DRIFT

Data model drift

A significant challenge in an archiving project is to remove the need for maintenance so that innovation cycles are not stifled. Archiving is a long-term project, as the archived data will be kept for many years. At the same time, the underlying metadata and data model are constantly modified by Salesforce's seasonal periodic releases and the enterprise's customer changes. This means historical archived data has a shape that is different from the current Org definition - a situation that only degrades further over time.

Data model drift also means new objects and fields appear or disappear. The archiving process must have the capability to self-discover the changes and adapt automatically to ensure drift is handled without interference. Without these abilities, you will have to manually adjust the archiving process every time a Salesforce change is made, which could negatively impact your business.

Data virtualization drift

Data is extracted and deleted from Salesforce during archiving whether it's because:

1. The data has reached the end of its useful life, in which case data is infrequently searched and accessed by a few users like system administrators

2. It's necessary to improve system performance, in which case some business users like champions in a Center of Excellence (CoE) will need to find, view, and interact with archived data

An archiving solution must, therefore, include a data virtualization capability that emulates the Salesforce interface: this allows you to archive data while achieving the best system performance - all without impacting the user experience.

The same metadata changes that cause data drift affect the visualization of archived data. The page layout of a record that was archived five years ago will be very different from the page layout of a record archived yesterday.

Salesforce may also change UI elements between seasonal releases. The look and feel of the virtualized data should match that of the Salesforce UI to allow businesses to have a consistent user experience. The data virtualization layer should adapt to UI changes as much as possible so that users can work uninterrupted.

05 DATA RETENTION

Different industries and types of data require different retention periods. The archiving plan must accommodate the retention required for specific business information. Some data classification will require that you set up a data retention policy of 10 years or more. Archived data can be corrupted over this time, this is why you will need to optimize data durability to prevent this corruption from happening. While data durability is a critical concept and is one of the most important KPIs when archiving LDVs, it is often overlooked or misunderstood. Prioritizing durability will help you archive accurate and intact data over a long period of time, without the risk of corruption. Conversely, if you disregard or misunderstand durability, you risk storing corrupted data, eradicating its usefulness over time.

DATA RETENTION

Data retention is a policy of how long you keep information.

Your data retention policy may be defined by the classification of data, regulations, and other factors. For example, your data retention policy for Human Resources data may be to archive data for 10 years, while log management data may only be stored for three months.

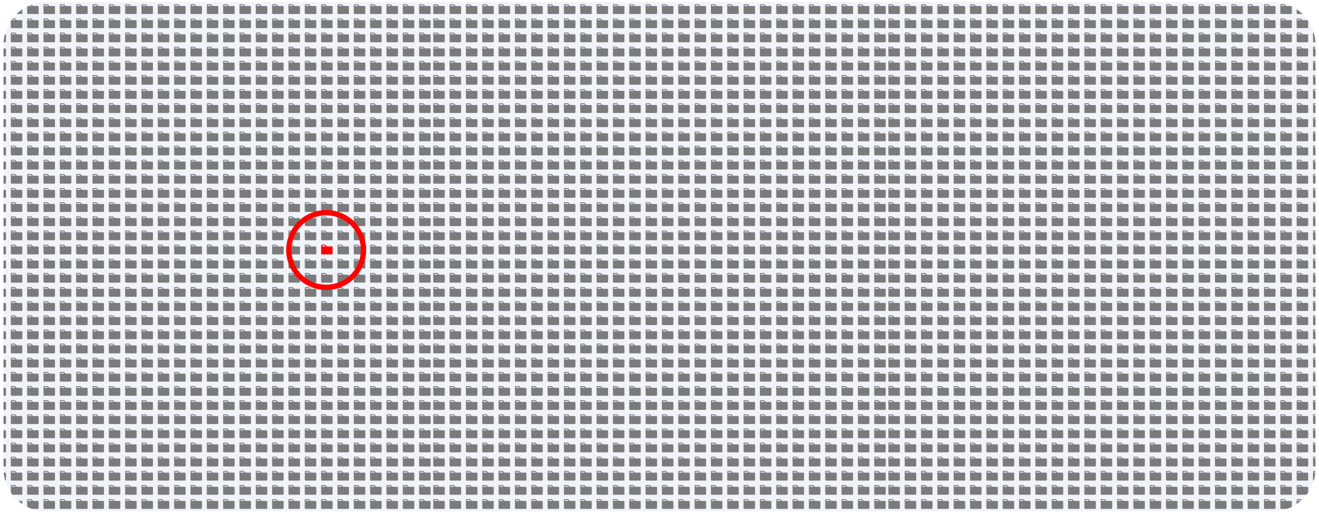
DATA DURABILITY

Data durability is the safeguarding of data against corruption when stored over time.

Archived data inevitably becomes corrupted over time due to a number of reasons, from environmental factors to the degradation of hardware. How you plan for this inevitable corruption, and how you rectify it, defines your data durability percentage.

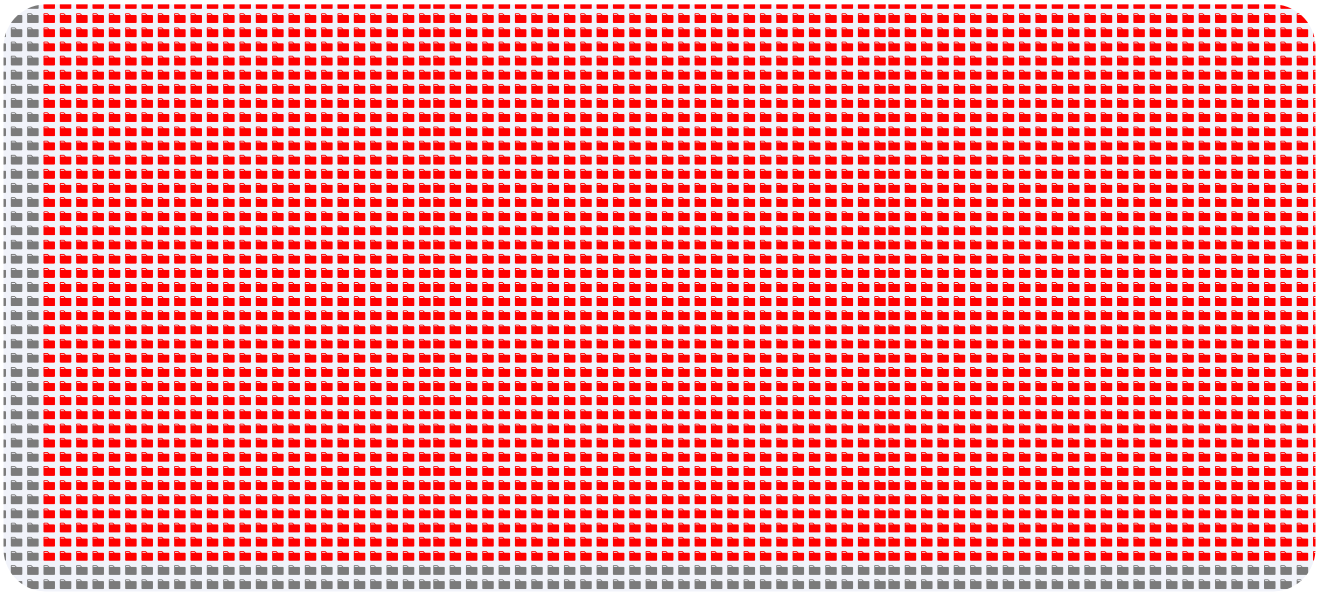
‘Good' data durability vs. 'Bad' data durability

The gold standard in terms of data durability is 99.999999999% (11 nines). It means that after 1,000 years, **only 1 object out of 100,000,000 objects may be corrupted.**



99.999999999% durability: 1 object in 100 million corrupted after 1,000 years (not to scale)

However if you aim for 100 years of data retention but only have 99.8% data durability, you only need to wait two months for one object to be corrupted. And after 99 years, **almost no intact data would survive.** It is easy to understand that retention then becomes irrelevant, why would you keep data for 100 years if it's completely damaged?



99.8% durability: almost all objects corrupted after 100 years

The difference between 99.999999999% durability and 99.8% durability is 1,000 years of perfect data, versus data that's corrupted after just two months. Now what happens to that corrupted data after three months? Three years? 30 years? The corrupted data becomes further and further corrupted.

Causes of bad durability

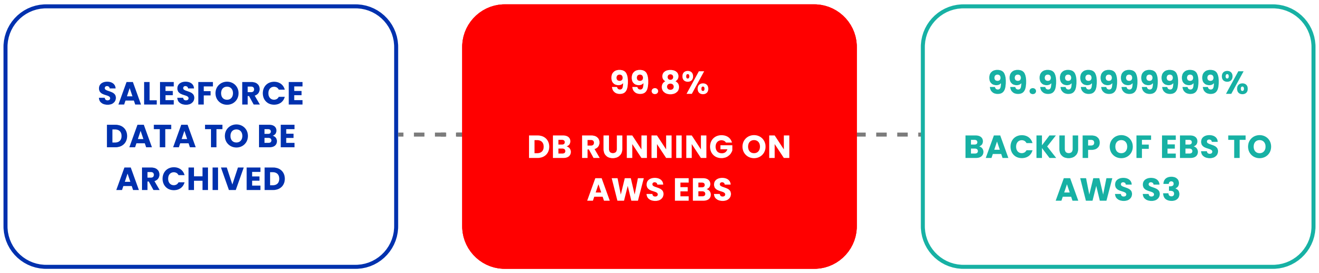
There are a number of factors that impact data durability, and the weakest among them will undermine any stronger durability controls.

Factors that can erode durability include the redundancy and the quality of the materials used to manufacture the physical storage medium, the manufacturing process, environmental conditions where the storage medium exists such as temperature and humidity changes, and physical damage to the storage media.

Any of these factors will erode the overall durability, regardless of the strength of the others. For example if the hard drive that was used to archive data becomes corrupted due to unfavorable environmental conditions, and you take a backup of the data on that hard drive, the backup inherits corrupted data.

Overall durability is only as strong as its weakest point

You must assess the durability of your entire architecture, from end-to-end. The following is an example of an architecture with poor durability:



The durability is poor because the data from Salesforce is archived on a virtual machine running on an AWS EBS disk. On that virtual machine, a database is leveraged to store the archive. Over time, the probability of corruption in that database is high because it offers between 99.8% durability. To mitigate data durability issues, one may think that adding a backup could increase durability by taking a backup of the virtual machine. However, the backup would just propagate the corrupted data.

Because the lowest durability of each individual component defines the overall durability of the architecture, this design illustrated above leads to 99.8% durability, which means that it may start being corrupted after two months and almost no data will survive intact after 99 years.

You must eliminate the weak durability parts of your architecture to avoid corrupt archived data, ideally any aspect of the architecture with durability lower than 99.999999999%.

06 DATA ENCRYPTION

Encryption continuum

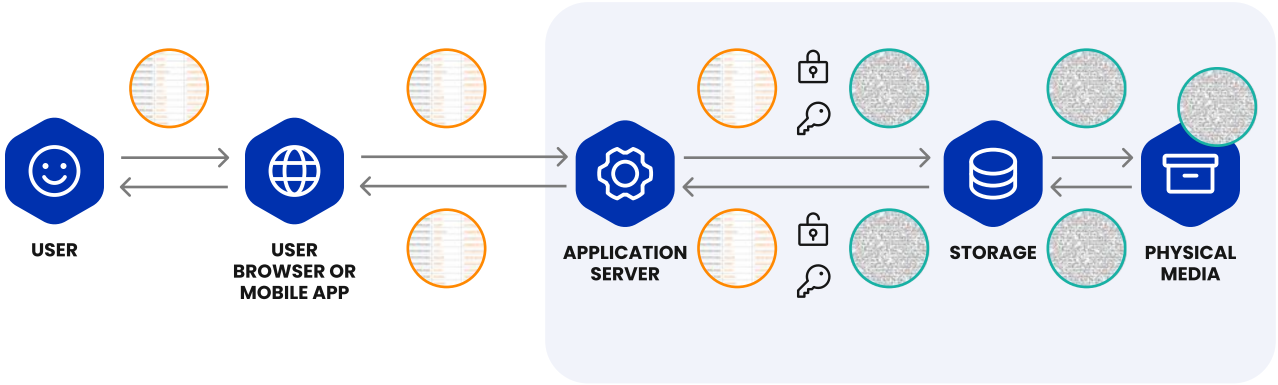
At Salesforce, trust is the #1 value and an Org is provided with numerous security measures. It is critical that any secondary systems be just as secure.

Data in Salesforce can be encrypted at a very granular level, down to the field, using Classic Encryption or Shield Platform Encryption. A similarly strong encryption pattern should be employed on the data archive system.

Encryption at rest prevents access to unencrypted data by ensuring the data is encrypted when on disk. The easiest way to implement it is through container-based encryption, where the database or storage layer transparently performs encryption and decryption operations.

Is this a sufficient level of encryption? Anyone with access to the system, such as database administrators, can see the data in its operational form.

By going one level deeper to granular encryption, data is obfuscated for everyone except the key holders.



Granular Encryption

Case study: Schneider Electric's experience managing archiving with Odaseva

Schneider Electric, the global leader in the digital transformation of energy management and automation, has 128,000+ employees in more than 100 countries. Their Salesforce database has grown substantially due to onboarding new businesses and expanding the capabilities of their Salesforce implementation.

The challenge

Initially, Schneider Electric built a custom archive solution using their internal Salesforce competencies. The bespoke solution was not able to scale to meet their business needs or performance goals.

The solution

Schneider Electric used Odaseva's archiving capabilities to archive 200 million cases, with more than 1 billion related records.

To support its business, the Opportunity object had a custom lookup to the Case object. When the related Case was removed, the field value was updated to null on the Opportunity. Although the opportunities remained in Salesforce, users could no longer see they once referenced a case.

To avoid losing valuable business information, the Odaseva Archive Viewer enables users to see the archived Case data. In addition, a new custom field on the Opportunity points directly to the archived record. With one simple click, users had full access to the previously archived Case and its related lists, using the associated page layout.

The outcome

Since partnering with Odaseva, Schneider Electric has observed significant improvements and increased efficiency in their archiving strategy.

Schneider Electric automatically archives Salesforce data at scale while keeping internal resources focused on delivering business needs.

What to look for in an archiving solution

These six archiving concepts cannot be achieved at enterprise level with point solutions.

Enterprises archiving Salesforce data at scale require a comprehensive, platform solution that enables precise data operations to properly control the safe removal of Salesforce data and archive this data at scale. Odaseva's expertise in data archiving spans more than 10 years. The Odaseva Enterprise Data Protection Platform supports the most advanced Salesforce data archiving use cases among global enterprises.

Odaseva Data Archiving enables global enterprises address the concepts detailed in this whitepaper:

Data depth

- Operate up and down data hierarchies to any depth, manage identifying potential orphan records so they can be handled properly during deletion
- Build the right strategies to identify the scope of data you need to remove
- Work with our Expert Services team to learn the best practices to surpass the limits of Salesforce native queries

Data volumes

- Craft your archiving plan by optimizing and sequencing operations to meet your requirements and performance expectation
- Isolate Tagging Records for Archive, Archiving and Deletion into separate processes to maximize efficiency and performance with LDVs
- Dynamically handle API Limits, ensuring you have head room available before executing archiving jobs to avoid reaching governor limits
- Separate execution of your backlog from your steady-state archiving plan to ensure you optimize effective removal of data without locking records and impacting business users

Data deletion

- Achieve powerful bulk LDV data processing by leveraging the full power of Odaseva
- Securely remove data from Salesforce at any scale
- Queue and balance data loads over days, weeks, or months to maintain the balance of data
- We orchestrate the following data operations to properly control the safe removal of Salesforce data:
    - Cascade Deletes
    - Orphan Records
    - Complex Relationships
    - Query Constraints
    - Uninhibited Restoration

Data drift

- Odaseva Archive Viewer emulates the user experience by leveraging our patent-pending Salesforce emulator for a seamless user experience
- With it, users can:
    - View data directly in Salesforce
    - Maintain relationships to your data
    - Recreate the look and feel of your Org
    - Let users Quickly Search for data
    - Keep data secure with encrypted data storage

Data retention

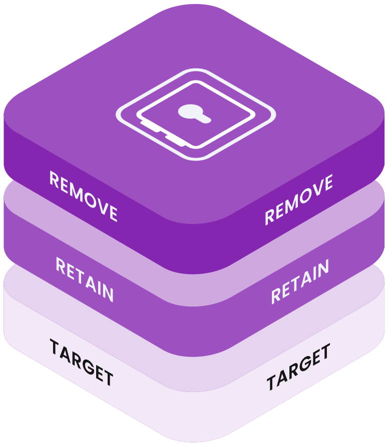
- Control precise retention of data
- Retain your data for long storage in a secure encrypted format that provides resilience and immutability
- Be certain that the data is secured and safe for decades
- Because we don't rely on AWS EBS to store your data, Odaseva is one of the very few data platforms that delivers 99.999999999% data durability

Data encryption

- ✓ Secured with Zero trust data storage
- ✓ Odaseva offers patent-pending five-level encryption and is a “no-view provider” by design, giving you unmatched data security
- ✓ Odaseva is the only Salesforce data platform to offer at-rest column/file/field encryption of your data with Bring Your Own Key (this means that only you can see your data—not Odaseva engineers or data center employees)
- ✓ Our servers are also encrypted at disk-level and OS-level, combining state-of-the-art encryption patterns

Odaseva Data Archiving Features

Odaseva Data Archiving ensures your archiving process meets the requirements of the six concepts outlined in this whitepaper. Odaseva Data Archiving tools include:



DATA ARCHIVER

Remove data in an intelligent way. Sequence the process to target the right data, secure it, then remove it.

Odaseva Data Archiver automates daily archive policies and automates archive backlog with:

- ✓ Real Time API Control
- ✓ Trusted Backup to rollback
- ✓ Separated Backlog control
- ✓ Automated Bulk Batches
- ✓ Tailored Solution to your challenge by our expert team

ARCHIVE VIEWER

Preserve your users' experience. Your organization can still access archived data in real-time.

Odaseva Archive Viewer leverages our patent-pending Salesforce emulator for a seamless user experience with Salesforce data:

- ✓ View data directly in Salesforce
- ✓ Maintain relationships to your data
- ✓ Recreate the look and feel of your org
- ✓ Let users quickly search for data
- ✓ Ultra secure encrypted data storage

Take control of your data strategy with Odaseva

COMPLY	OPTIMIZE	REMOVE	VIEW
Compliance with data policies	Reduce large data services	Retain data securely for admin access	Emulate your salesforce experience for end users
Data Lifecycle	**LDV Data Services**	**Data Archiver**	**Archive Viewer**
Plan the policies you need to remain compliant with your data	Get help to manage how to address your data backlog with our expert services	Retain your data for longer and keep data polices running	Make your archive data available to end users with a Salesforce experience

Key Takeaways

Are Large Data Volumes impacting your organization’s use of Salesforce? If so, it’s time to take a look at your archiving strategy.

When working with LDVs on Salesforce, implementing an effective archiving strategy can mitigate inevitable challenges like non-compliance with regulations, degraded system performance, and poor user experience. The time to solve for this is before data scale affects platform efficiency.

By implementing an archiving strategy that addresses the six concepts outlined in this whitepaper, you can remove data from Salesforce while preserving the user experience.

Odaseva Data Archiving was designed and built to meet the needs of large Salesforce Orgs.

Get in touch for a personalized demo today