Engineer

Recovery point objective

the maximum time span of data that may be lost

What it is

A recovery point objective, usually shortened to RPO, is the limit on how far back in time a service can go when restoring data after a disruption. An objective of fifteen minutes means that the organisation can tolerate losing at most the changes from the final fifteen minutes before the event under the agreed failure scenario. If the last verified recoverable state is fifty minutes old, the result misses that objective even if the service starts immediately.

Specify the data and the scenario alongside the number. Losing a recent local cache may matter little when it can be rebuilt from a trustworthy source. Losing accepted orders matters differently, particularly when a customer has already received a confirmation. A deleted row, a full regional outage and a compromise of operating credentials can leave different backups available. A target attached to one data store and one well stated scenario can be tested.

The recovery time objective measures how long the system can be unavailable while it is restored. Recovery point and recovery time therefore lead to different questions. One asks which state survived. The other asks when useful service returned. A daily backup that restores in minutes may meet the time target and still lose a day’s work. A copy that retains recent transactions may meet the point target and still take too long to verify and serve.

Use the concept in a decision

Put “What is a recovery point objective?” to work in a real decision. We can help relate the concept to the system you are building or buying. Backups that are tested rather than assumed, and a written plan for the morning a server does not come back on.

Explore our services Get in touch →

Test the assumptions behind the term

Use the explanation and linked work to frame questions for your project. We can examine the assumptions with you before a tool is chosen. Automated 100 critical data backups using Barman — 100 critical backups automated across the databases

Automated 100 critical data backups using Barman Get in touch →

Why it matters

A backup schedule is an input to a design. The objective is a requirement for what can be recovered. A job advertised as running every five minutes might lag behind the application, fail without a useful alert or hand back a copy that cannot be replayed. A team can test the difference by placing known changes at recorded times and then tracing the last contiguous state it can restore. Measuring the end result reveals problems which an ordinary successful backup log cannot show.

The target should reflect what the organisation is willing and able to reconstruct. Ask what happens to a receipt, an accepted booking or a supplier event produced during the lost window. Some records can be replayed from a durable upstream log. Others exist only in the application that recorded them. Estimate the cost and risk of reconstructing each record, and give the work an owner. A small objective on a small set of important state can be more useful than the same ambitious number across every cache and report.

Frequent copying is only one approach. Depending on the data, a recovery strategy may combine periodic snapshots with a transaction log, a replica, an immutable copy or independent upstream event retention. These options have different properties under different failures. A replica can transfer accidental deletions as quickly as legitimate work. A separate copy helps only if the restoration team can access, interpret and validate it when it is needed.

What it looks like in practice

Engineer ApS describes backups that have actually been restored and the availability and handover terms in its service levels. The portfolio account gives a reader evidence about exercising restoration. The numerical example that follows is a hypothetical way to test a point objective without assuming a target for the published case.

Assume a shop adopts a fifteen-minute RPO for accepted orders in the event of a failed database. It snapshots the database and saves a sequence of transaction records separately. At 14:00 a rehearsal removes access to the primary system.

Write known test orders before and after 13:45, as well as a cancellation and a payment update. Start the recovery with the permissions the duty team will actually have. Restore to a separate environment and choose the latest consistent state whose data can be verified.

If that state includes all expected transactions through 13:50, the observed gap is ten minutes and the point target is met for the rehearsal. A visible order from 13:58 does not alone establish a two-minute gap if earlier records are missing. Compare row identities and totals and check the dependencies an order needs to be actionable. Preserve the restoration evidence and account for what would happen to the orders in the remaining window. Record the time when useful service could return separately.

A second rehearsal can change the circumstance. Remove the ordinary operator’s access to the shared control plane or simulate an accidentally deleted item by choosing a clean point before that edit. Inspect the copies available under that condition. Results for one scenario are useful evidence for that scenario, and a reason to test additional ones whose dependencies differ.

Where it goes wrong

Teams often assume that replica lag is an end-to-end measure of loss. An application may write order details to one store and attach evidence to another. If their time boundaries differ, an apparently recent database may reference files that did not survive. Include linked records and check consistency across those boundaries in the actual recovery exercise.

A recovery run may still rely on unavailable credentials, an undocumented encryption key or a human who remembers how to find the right copy. Give the authorised duty role the actual runbook and exercise access with it. Write down which timing and capabilities are assumptions. Time spent locating copies is also useful evidence for the separate time objective.

Finally, review the target when data becomes harder to reconstruct. The first order application may have started as a copy of a manual log, for which a day’s loss was manageable. After the system becomes the only accepted record, the same target has a different meaning. Revisit the loss limit and repeat the exercise with the affected process owner before treating the old measurement as evidence for the new service.

Further reading

- NIST contingency planning guide for federal information systems
- AWS Well-Architected recovery objectives and disaster recovery
- NIST recovery-point objective terminology

Glossary

Choose for your actual constraints

Bring the deployment path, the operational risks and the people on call. We can scope a change with checks, a recovery route and clear ownership.

See what it costs Get in touch →

Make the terminology practical

Ask how “What is a recovery point objective?” applies to your organisation. A concrete example from your current system is enough to start the discussion.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://engineer.company/glossary/recovery-point-objective/