PH7X SYSTEMS ®

0.1.0 · CHAPTER 36

Write the way back before cutover, and know the moment it closes

SharePoint Migration Field Guide · Chapter 36

pH7x Systems

2026-11-10

36. Write the way back before cutover, and know the moment it closes

> The old farm is your way back for exactly as long as nothing true exists only on the new one.

Chapter 33 wrote what stops a wave, Chapter 34 named rollback as one of three exits from a failed criterion, and Chapter 35 takes the go or no-go. All three assume that somebody can still put people back on a system that works. This chapter is that assumption, written down: the plan, the trigger, who may pull it, the moment the clean way back closes, and the procedure for each side of that moment.

> **If the move goes wrong, how do you put people back on a system that works, and until when can you?**

The decision this chapter leads to is one, taken by a named person: roll back or fix forward, against a trigger and a point of no return that were written and rehearsed before cutover.

Which moves this chapter is for, and what must already be true

A move from SharePoint Server 2016 or 2019 to SharePoint Server Subscription Edition by database attach (Chapter 23), treated first and in most depth. A copy into SharePoint Online has its rollback states in Chapter 9; one short section here says what differs.

The chapter takes four things from the chapters before it, and repeats none of them:

- from Chapter 33, the wave calendar: which databases sit behind which address, the stop criterion, and who may invoke it;
- from Chapter 34, the acceptance record: the lines somebody signed, and the test items validators created;
- from Chapter 32, the restore test: the backups restored somewhere else, checked and timed;
- from Chapter 35, four things: the recorded instant at which write authority moved, the previous value of every step of its runbook, the list of who could write since, and a 2016 or 2019 farm nobody has touched.

It hands the trigger list and the expiry date to Chapter 37, which watches them during hypercare, and the same date to Chapter 38, which may not switch off the old farm before it.

What the old farm still is

For an attach, Microsoft's sequence leaves the SharePoint Server 2016 or 2019 farm standing. Its databases are set to read-only before the backup, so that "users can continue to view content. However, they will be unable to add or change content", and the backup "guarantees that you have the data in a safe state if you must enable the original farm again" (copy databases to the new farm, checked on 10 November 2026). That sentence is all that Microsoft's Subscription Edition upgrade pages say about going back. Its page for the upgrade to SharePoint 2013 says one thing more: "if you encounter a problem with upgrade, you can restore the read-only farm to read-write and restore access to users while you rework your plans before you try upgrade again" (determine strategy for upgrade to 2013, checked on 10 November 2026). Both pages say that you can. Microsoft does not document, as a sequence, how. The steps in this chapter are put together from Microsoft's pages on each part, and the order is our recommendation.

Four things were done to the old farm, and a return undoes each:

WHAT THE ATTACH DID TO THE 2016 OR 2019 FARM	WHERE MICROSOFT DOCUMENTS IT	WHAT A RETURN HAS TO UNDO
Content databases set to read-only	Copy databases, above	Each database set back to read-write
Business Data Connectivity, Managed Metadata, Secure Store and User Profile databases set to read-only	Copy databases, and upgrade service applications: "Set the User Profile databases to read-only"	Each set back to read-write
Search service application paused, and its administration database set to read-only, while that database was copied	Upgrade service applications: "you have to pause the Search service application", with `Suspend-SPEnterpriseSearchServiceApplication`	Where it is still paused: database read-write, then the application resumed
The address moved to the new farm at cutover	Not documented as a step. Each web application keeps "the same URL (including name, port, and host header)" (upgrade content databases)	The address, and the service principal name where Kerberos is used, put back to the previous values Chapter 35 recorded

All three pages were checked on 10 November 2026.

The upgrade runs one way

The databases on the new farm are not a second way back. Attaching a content database to Subscription Edition upgrades it: `Mount-SPContentDatabase` "causes an upgrade of the schema of the database" (Mount-SPContentDatabase, checked on 10 November 2026; the reference lists its parameters for SharePoint Server 2010 to 2019, and the Subscription Edition upgrade page is what runs it there).

What Microsoft says about the opposite direction, page by page, all checked on 10 November 2026:

- **No page documents it.** We found no Microsoft page that describes attaching a database upgraded by Subscription Edition to a SharePoint Server 2016 or 2019 farm, and none that says in so many words that it fails. Chapter 9 reached the same result on 1 November 2026.
- **Central Administration refuses a different version.** An existing database attached there "must be the same version as the SharePoint Server 2016 farm or this operation will fail" (attach or detach content databases). The page sends a database of a different version to PowerShell, where the cmdlet upgrades it.
- **Export and import do not cross versions.** "You cannot import a site, list or document library exported from one version of SharePoint Server to another version of SharePoint Server" (import a list or document library). `Export-SPWeb` on the new farm is not a way to carry a week of changes to the old one.
- **SQL Server does not go down a version.** "No SQL Server backup can be restored to an earlier version of SQL Server than the version on which the backup was created" (RESTORE), and "A database created by a more recent version of SQL Server can't be attached in earlier versions" (database detach and attach). This applies where the old farm's SQL Server is an earlier version than the new farm's.

So, as our conclusion and not Microsoft's sentence: treat the upgrade as one way. The way back is the preserved 2016 or 2019 farm, never the upgraded copy, and whatever people wrote on the new farm does not travel with a return unless somebody carries it by hand.

The point of no return

Microsoft's trial upgrade guidance, written for the upgrade to SharePoint 2013, asks the question in these words: "Is there a point of no return? Test your fallback plan and make sure that it works before you begin your real upgrade" (use a trial upgrade, checked on 10 November 2026). The Subscription Edition pages we found do not repeat it. For an attach there are two lines, and the plan names both:

WHERE THE WAVE IS	WHAT A RETURN COSTS	PROCEDURE
Before the cutover instant	Nothing the business wrote. This is a stop under Chapter 35, whose runbook reverses its own steps	Chapter 35's way back; then, if the old farm reopens, the return below without steps 2 to 4
After the cutover instant, before the first business write on the new farm	Nothing the business wrote, if you can prove there was no write	The return below, every step
After the first business write	Every change since the instant is re-entered by hand or accepted as lost	Return after writes, below. The default is to fix forward
After the expiry date in the plan	The new farm is the system of record	No rollback. Chapter 37 fixes forward, Chapter 38 may start

The first business write is the clean line: before it, a return is a return; after it, a return is a small migration in the wrong direction. The expiry date is the last line. A third thing closes the way back without anybody deciding: the old farm stops being intact, because a database was changed, a backup aged out of its retention, or a server was reused.

We found no Microsoft page that gives one report of everything written to a farm since a given moment. We recommend that the plan says how you will know, before anyone needs to: Chapter 8's read of the new farm filtered to items created or modified after the cutover instant, less the test items Chapter 34's record lists. Where that read cannot be made, the plan says that whether a write happened is not established, and the decider signs with that limit, as Chapter 9 requires.

The rollback plan

One page per wave, written with the wave calendar and signed before Chapter 35's decision. Print it; the last column is for a signature.

LINE	WHAT IT HOLDS	OWNER	SIGNED
Scope	The addresses in the wave and every content and service application database behind them	Migration lead	
Trigger	The conditions that open the decision, each with how it is measured and how long a fix may be tried first	Decider	
Authority	Who decides, a deputy reachable for the whole window, and who may call for the decision	Decider	
Clean line	The cutover instant, and how a business write after it is detected	Migration lead	
Expiry	The date after which there is no rollback	Decider	
Procedure	The steps below, with a named person per step and the time each took in rehearsal	Farm administrator	
Backups	Where the pre-upgrade backups are, their retention, the result of Chapter 32's restore test and the time a restore may take	Database administrator	
Address	How it moved at cutover and how it moves back, with the time people's machines take to follow	Network owner	
Losses	What a return after writes loses, written out, accepted by the owner of each site	Site owners	
Messages	The text sent on a return, written in advance	Migration lead	
Rehearsal	The date, what failed, what changed in the plan	Migration lead	

The trigger, and who may pull it

A trigger opens a decision; it does not take it. We recommend writing each one as a measurable condition with a time limit, because during an incident every failure looks fixable in another hour:

- a whole sign-in method fails on the new farm and is not fixed within the time the plan gives;
- a kept customisation or integration that a business process cannot work without fails, with no fix inside that time;

- content is found wrong or missing and the cause lies in the upgrade, not in one site;
- Chapter 34's acceptance cannot finish inside the read-only window and its owners will not extend it;
- anything Chapter 33's stop criterion lists, discovered after the go.

Three roles, three people. Anyone on the migration team or among the validators may call for the decision. The plan's owner takes it, or the deputy; we would make that the decider Chapter 1 named, so that the go and the return sit with one person. The farm administrator, the database administrator and the network owner carry it out. Chapter 35 confirms that all of them are reachable before it says go; a rollback that needs somebody who is on a flight is not available.

After the first business write, the decision has a second question, and only the owners of the affected sites can answer it: is losing or re-entering what was written since cutover better than working on the new farm while the fault is fixed? Write their answer down with their names.

Rehearse it

Chapter 31's rehearsal farm and Chapter 32's restore test are where this procedure runs for the first time. We recommend rehearsing three things, and timing each:

1. On a copy of the old farm, or in an agreed maintenance window on the old farm itself, set one content database read-only and back to read-write, and watch what its site collections and scheduled jobs do.
2. Restore one pre-upgrade backup to a SQL Server of the old farm's version and open a site from it. This is Chapter 32's restore test. Microsoft's own sequence restores one database first, "and then verify that the restoration has worked before you restore the other databases" (copy databases, above).
3. Move the address of a test web application to the rehearsal farm and back, and measure how long a user's machine takes to follow.

The backups already passed one test without anybody planning it: every database the new farm attached was restored from its backup, so each backup is known to be readable. That does not time a restore on the old farm, which is why the second rehearsal exists.

Return before any business write

For a SharePoint Server 2016 or 2019 farm whose databases are still read-only. A no-go or a stop under Chapter 35 changes nothing on the old farm and opens a decision that belongs here: the plan's owner decides whether the farm stays read-only until a new date or reopens to writes. Reopening records that decision as step 1 and continues at step 5. A return after the cutover instant runs every step.

STEP	WHAT YOU DO	OWNER	CHECK BEFORE THE NEXT STEP
1	Record the decision: the trigger, the time, who decided	Decider	The entry exists and names the trigger
2	Lock the new farm's site collections read-only again, as Chapter 35's runbook did before the instant, so that only one farm can accept a write	Farm administrator, new farm	A test write by a named user is refused
3	Prove that no business write happened since the cutover instant, from Chapter 35's list of who could write and the read described above	Migration lead	The read lists only test items. If it lists anything else, stop and go to the next section
4	Put the address, and the service principal name where Kerberos is used, back to the previous values Chapter 35 recorded	Network owner, directory owner	From a user's desk, the address reaches the old farm and sign-in works
5	Confirm that no machine still carries a name-resolution entry for the new farm (Chapter 34)	Farm administrator	No validator's machine reaches the new farm
6	Set the old farm's service application databases to read-write: Business Data Connectivity, Managed Metadata, Secure Store, User Profile, Search administration	Database administrator	`is_read_only` is 0 for each
7	Set the old farm's content databases to read-write	Database administrator	`is_read_only` is 0 for each
8	Check the lock of every site collection	Farm administrator	Each shows the lock it had before the window
9	Resume the Search service application, where it is still paused	Farm administrator	A crawl runs, and a new test item is found
10	A validator creates, edits and deletes a test item in each web application	Validator	The read-only statement is gone, and the validator signs
11	Send the prepared message	Migration lead	Sent, with the time
12	Stop the next wave, as Chapter 33 writes, and keep the new farm as it is until the cause is known	Migration lead	The calendar is reissued

Steps 6 and 7. Microsoft documents the change in SQL Server Management Studio: in the database's properties, "In the Database Read-Only row, click the arrow next to True, select False" (health rule for read-only databases, checked on 10 November 2026). The account is "a member of the db_owner fixed database role in each database". The Transact-SQL form is `ALTER DATABASE` with `SET READ_WRITE`, and "Changing the state of a database or file group to READ_ONLY or READ_WRITE requires exclusive access to the database" (ALTER DATABASE SET options, checked on 10 November 2026).

The example below is illustrative and was not run against a farm, because none is available to us. It uses `Get-SPContentDatabase -WebApplication` (reference), `Invoke-Sqlcmd` with `-ServerInstance`, `-Database` and `-Query` (reference) and the `is_read_only` column of `sys.databases`, which Microsoft's own example on the ALTER DATABASE SET options page reads after changing the state; the reference page for `sys.databases` gives its meaning as "1 = Database is READ_ONLY". All were checked on 10 November 2026.

```
# Illustrative. Not run against a farm.
# Run on the SharePoint Server 2016 or 2019 farm, in a shell that has
# both the SharePoint cmdlets and the SqlServer module.
# It prints what it would do. It changes nothing until $Apply is $true.
$Apply       = $false
$WebAppUrl   = "<URL of the web application>"
$SqlInstance = "<SQL Server instance of the old farm>"

$databases = Get-SPContentDatabase -WebApplication $WebAppUrl
foreach ($db in $databases) {
    $name  = $db.Name
    # Read the current state first: 1 means read-only.
    $state = Invoke-Sqlcmd -ServerInstance $SqlInstance -Database "master" -Query "SELECT name, is_read_only FROM sys.
    "Database $name : is_read_only = $($state.is_read_only)"
    "Would run: ALTER DATABASE [$name] SET READ_WRITE"
    if ($Apply) {
        Invoke-Sqlcmd -ServerInstance $SqlInstance -Database "master" -Query "ALTER DATABASE [$name] SET READ_WRITE"
    }
}
```

The service application databases are not content databases, so the script does not list them: take their names from Chapter 23's record of what was copied.

Step 8. When a content database becomes read-only, its site collections are "automatically set to read-only if the locking status of the site collection was previously None, No Additions, or Read-Only", and one set to No Access stays so (run a farm that uses read-only databases, checked on 10 November 2026). That page does not say what happens to those locks when the database returns to read-write, and we found no page that does. So we would record every site collection's lock before the window, and read each again here, in Central Administration under "Configure quotas and locks" or with the documented command (manage the lock status for site collections, checked on 10 November 2026):

```
# Illustrative. Not run against a farm. One site collection at a time,
# and only one whose recorded lock before the window was "Not locked".
"Would unlock: <SiteCollection>"
Set-SPSite -Identity "<SiteCollection>" -LockState "Unlock"
```

A site that was locked on purpose before the window stays locked.

Step 9. The Search service application "does not function when its databases are set as read-only" (run a farm that uses read-only databases, above), so the database comes first. While the application was paused, the old farm's index "isn't updated" (upgrade service applications, above), so results are as old as the pause until a crawl catches up. `Resume-SPEnterpriseSearchServiceApplication` "Resumes a search service application that was suspended" (reference, checked on 10 November 2026):

```
# Illustrative. Not run against a farm. Run on the 2016 or 2019 farm,
# after the Search administration database is read-write again.
$Apply = $false
$ssa   = Get-SPEnterpriseSearchServiceApplication -Identity "<SearchServiceApplicationName>"
"Would resume the Search service application: <SearchServiceApplicationName>"
if ($Apply) { $ssa | Resume-SPEnterpriseSearchServiceApplication }
```

Expected result. People open the address they always used and reach the 2016 or 2019 farm; the statement at the top of the home page that "the site is read-only" is gone; a validator can write; search finds an item created after the return. The estate is where it was before the backup, and nothing the business wrote is missing, because the business wrote nothing.

Return after writes

The default from here is to fix forward, as Chapter 9 sets out: the fault is repaired on the new farm, under Chapter 37. A return is chosen only when the site owners have answered the second question above in writing. Then:

STEP	WHAT YOU DO	OWNER	CHECK BEFORE THE NEXT STEP
1	Record the decision and the owners' acceptance of what is lost	Decider	Each affected owner's name is on it
2	Make the new farm read-only: its site collections locked as in Chapter 35's runbook, or its content databases set to read-only. People can still read what they wrote	Farm administrator or database administrator, new farm	A test write by a named user is refused
3	Back up the new farm's databases as they are. They are the only copy of what was written since cutover	Database administrator, new farm	The backup is listed and readable
4	List every change since the cutover instant: files, list items, permissions, site settings, terms, profiles, stored credentials	Migration lead	The list is complete, or its gaps are stated
5	For each change, the site's owner decides: carried back by hand, re-entered, or accepted as lost	Site owners	Every line has a decision and a name
6	List what the writes triggered outside SharePoint, and decide a compensation for each, as Chapter 9's table requires	Owner of each other system	Each effect has a decision
7	Reopen the old farm: steps 4 to 10 of the previous table	As there	As there
8	Carry the changes: owners download their files from the read-only new farm and add them to the old one; list items are entered again	Site owners	Each line of the list is marked done
9	Send the prepared message, with what was lost	Migration lead	Sent
10	Keep the new farm read-only, reachable to the team only, until every line is closed	Farm administrator	The date it may be cleared is written

For step 3, Microsoft documents `Backup-SPFarm -Directory <BackupFolder> -BackupMethod {Full | Differential} -Item <ContentDatabaseName>`, and "If you are backing up the content database for the first time, you must use the Full option" (back up content databases, checked on 10 November 2026). SQL Server tools do the same job.

Three limits, stated plainly:

- **The carry is by hand.** Export and import do not cross versions, as above. We found no Microsoft page that documents a tool for moving content from Subscription Edition to SharePoint Server 2016 or 2019. A third-party tool may; we have not tested one, and it is out of scope here.
- **A re-entered change is a new change.** We would expect the old farm to record the person who re-entered it, the time of the re-entry and no earlier versions. We do not know every case, so the rehearsal re-enters one file and one list item and reads what the old farm kept.
- **Service data stays behind.** A term, a profile property or a stored credential changed on the new farm is in databases that do not go back either. Each is on the list in step 4 or it is lost without anybody deciding.

When the old databases themselves are gone

The return above needs the old farm's databases intact. If one was changed, dropped or damaged during the window, the pre-upgrade backup is what "guarantees that you have the data in a safe state". It holds the database as it stood when it became read-only, which is also everything the business ever wrote to the old farm, so nothing is lost by restoring it.

Microsoft's procedure for a content database with SQL Server tools: "If the SharePoint Timer service is running, stop the service and wait for several minutes for any currently running stored procedures to finish. Do not restart the service until after you restore the content databases"; restore with "Overwrite the existing database" selected and no other restore op-

tion; then start the service (restore content databases, checked on 10 November 2026). The same page gives `Restore-SPFarm -Directory <BackupFolder> -RestoreMethod Overwrite -Item <ContentDatabase>` for a backup held in a SharePoint backup folder; the attach's backups were taken in SQL Server Management Studio, so the SQL Server procedure is the one that applies to them. Check the restored database's `is_read_only` afterwards: we do not know whether a backup of a read-only database restores as read-only in every case, and the rehearsal shows it.

This is what Chapter 32's restore test contributes: a restore somebody has done, on a server of the old farm's version, with its duration written in the plan. A backup nobody has restored is a file.

Validate the return

A return is finished when the old farm has been accepted again, not when the last command ran. Three readings, in the order they get more expensive:

1. **The state.** Every database in the plan's scope reads `is_read_only` 0; every site collection has the lock it had before the window; the Search service application is running.
2. **The content.** Chapter 8's read of the old farm, compared with the read taken before the backup. After a return with no business write the two match, and a difference is a finding. After a return with writes, the differences are the lines of the change list and nothing else.
3. **The work.** The lines of Chapter 34's record that matter most, run by the same validators on the old farm: a sign-in for each method, a test write, a new workflow instance, each integration's four results (Chapter 22) and a search. They sign as they signed acceptance.

Each integration that was pointed at the new farm at cutover has its own way back, owned by the other system's owner; the address moving back does not move a connection that used a server name.

What commonly fails

WHAT YOU SEE	LIKELY CAUSE	WHAT TO DO
`ALTER DATABASE` waits or fails	The change "requires exclusive access to the database", and SharePoint holds connections to it	Microsoft's own example "sets the database to SINGLE_USER mode to obtain exclusive access" first (ALTER DATABASE SET options, above). Rehearse it, and time it
A site is still read-only after its database is read-write	The site collection's own lock	Step 8. Microsoft does not document that the lock clears by itself
Search returns nothing new	The application is still paused, or its administration database is still read-only	Step 9, in that order
Editing a profile, a term or a connection returns an error	A service application database is still read-only: tasks that write to it "return errors" (run a farm that uses read-only databases, above)	Step 6
Some people still reach the new farm	Their machines kept the new address, or a name-resolution entry was not removed	Steps 4 and 5; wait the time measured in rehearsal
On a later attempt, the attach succeeds but nobody can open the sites	The first upgraded copy is still attached: "you cannot add the same site collection two times to the farm", and "you will be unable to browse to the site collection" (upgrade content databases, above)	Detach the earlier copy first. `Dismount-SPContentDatabase` "will not delete the content database" (reference, checked on 10 November 2026), and Microsoft's own example previews the detach with `-WhatIf` before it is performed
Work that fell due during the window runs when the farm reopens	Timer jobs, workflows and scheduled integrations paused by the read-only state	Microsoft does not document what a 2016 or 2019 farm does with them, and we do not know. Watch them in rehearsal, and warn the owners of anything that sends messages

After any return, a second attempt starts again from the read-only step of Chapter 23 with new backups. The copies on the new farm are older than the old farm from the first write after reopening, and no delta closes that gap on an attach.

A copy into SharePoint Online: what differs

Chapter 9 holds the five states and the three rules, and they apply unchanged. Four things differ from an attach:

- **The source was never upgraded or copied away.** It is the same SharePoint Server 2016 or 2019 site collection, locked. Reopening it is one documented command per site collection, `Set-SPSite -Identity "<SiteCollection>" -LockState "Unlock"`, which "Unlocks the site collection and makes it available to users" (manage the lock status, above).
- **The destination can be frozen per site.** `Set-SPOSite -Identity <site URL> -LockState ReadOnly` is the "one writer" step: the valid values "are: NoAccess, ReadOnly and Unlock", and with ReadOnly "a message will appear on the site stating that the site is under maintenance and it is read-only" (Set-SPOSite, checked on 10 November 2026).
- **The unit is a site, not an address.** One site can return while the rest of its wave stays, which an attach cannot do (Chapter 33). Every link and redirect that Chapter 35 pointed at the new address is reversed for that site.
- **Microsoft's tool copies one way.** The SharePoint Migration Tool exists "to help you migrate content from on-premises SharePoint sites and file shares to Microsoft 365" (SPMT overview, checked on 10 November 2026). We found no Microsoft tool documented for the opposite direction, so changes written online come back by hand, as above.

Risks and limits

- **A return goes back to an unsupported product.** SharePoint Server 2016 and 2019 reached the end of support on 14 July 2026, after which there are "no new security updates, non-security updates, free or paid assisted support options or online technical content updates" (ending support in 2026, checked on 10 November 2026). A rollback buys time to try again. It is not a place to stay, and the plan gives the second attempt a date.
- **Every day the option stays open costs something.** The old farm stays patched as far as it can be, backed up and staffed until the expiry date. That cost is why the date exists.
- **After the first business write, a return loses something.** At the least, the history of what was re-entered; at the most, whatever the change list missed.
- **This procedure is assembled, not published.** Each step rests on a Microsoft page; the sequence does not. Until you have rehearsed it on your own farm, with your own customisations and jobs, its timings and side effects are not established.

What rollback hands on

To Chapter 37: the trigger list and the expiry date. During hypercare, every incident is read against the triggers first; one that meets a trigger goes to the decider, and one that does not is fixed forward. After the expiry date nothing is.

To Chapter 38: the same date. The old farm is the way back, so it may go only when there is no way back left to keep: after the expiry date, with Chapter 9's evidence gate passed. The pre-upgrade backups outlive the farm on whatever retention Chapter 38 gives them.

To Chapter 33: after a return, the cause, the reissued calendar and the measured time of the return itself, which the next wave's plan uses.

A rollback you have never rehearsed is not a plan, it is a hope.

Out of scope

Chapter 35 owns the go or no-go, the cutover instant and every way back before it; Chapter 37 owns every incident that does not meet a trigger; Chapter 38 owns switching the old farm off. Chapter 9 owns the rollback states of a copy, and Chapter 23 the attach. Backup and recovery design, high availability, disaster recovery between datacentres, third-party tools that move content between versions, and the recovery of the systems SharePoint integrates with are not covered.

Before you continue

- [ ] The rollback plan for each wave was written with the calendar and signed before the cutover decision
- [ ] Each trigger is a measurable condition with a time limit, and the decider, a deputy and the people who carry it out are named and reachable
- [ ] The plan names the clean line, the first business write, with the way a write is detected, and the expiry date
- [ ] The return was rehearsed and timed: a database to read-only and back, one backup restored on the old farm's version, the address moved and moved back
- [ ] Every database the attach set read-only is listed, service application and search databases included, with each site collection's lock before the window
- [ ] A return after writes has its change list, its owners' written acceptance of what is lost, and a backup of the new farm first
- [ ] The return is validated like a wave: state, Chapter 8's read, and the validators' signatures
- [ ] Hypercare and decommissioning hold the same expiry date, and the old farm and its backups are untouched until then