

Containing the blast radius in the Microsoft 365 configuration layer

How a leading European nuclear energy provider brought continuous configuration change under control, after a four-month build attempt failed

EXECUTIVE SUMMARY

The configuration layer moves every day, in small increments.

A leading European nuclear energy provider spent four months and approximately \$1.7 million building configuration control for its Microsoft 365 and Entra ID estate using the open-source PowerShell DSC platform. **It never completed a single full tenant restore, even after the scope was narrowed to Conditional Access alone.**

That didn't mean the requirement behind that effort had gone away. A tenant of this size is not a static object: Conditional Access policies are tuned, mailbox rules are added, groups are nested, app registrations are consented to, sharing settings are relaxed for one project and never tightened again. The configuration layer moves every day, in small increments, and while most of that change is legitimate, some is not. A share of the legitimate change can turn out later to have unexpected consequences. **That is the exposure in the configuration layer, and it is why the organization needed a control that could see it, gate it, and undo it when required.**

This paper details how they used CoreView Tenant Resilience to define a hardening model, monitor and detect configuration drift daily, enforce change control, and validate restore capability at twice their production object count.

KEY METRICS AT A GLANCE

60,000+

configurations restored in a single run

<24h

to complete the full tenant configuration restore

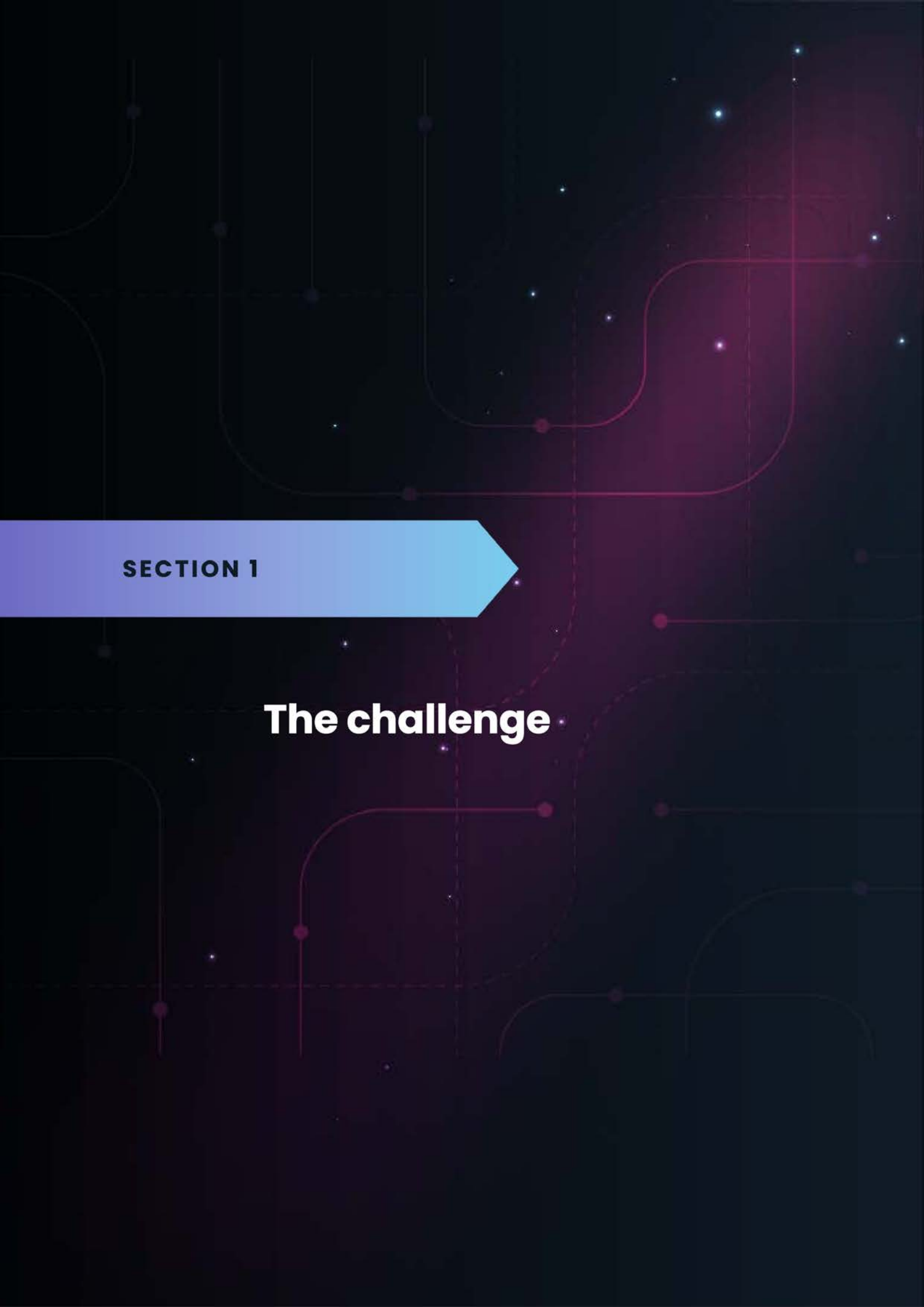
2×

every production object count doubled for proof-of-value load testing

CONTENTS

What's inside this paper

01	Executive summary →	2
02	The challenge →	4
03	Proof of value →	8
04	How Tenant Resilience works →	11
05	Conclusion →	18

The background is a dark, deep blue or black gradient. It is decorated with a network of thin, glowing lines in shades of light blue and magenta. These lines are mostly horizontal and vertical, but many have rounded ends, giving them a futuristic or digital feel. Small, bright dots of the same colors are scattered throughout the background, some appearing to be at the intersections of the lines. A horizontal bar with a blue-to-white gradient and a right-pointing arrow shape is positioned in the upper left quadrant.

SECTION 1

The challenge

THE STARTING POINT

Four months, \$1.7M, and no working restore

The organization operates nuclear power plants across Europe, under a regulatory environment that demands **configuration state is auditable evidence rather than an internal implementation detail**. As part of its cloud adoption program the energy provider set out to bring its Microsoft 365 and Entra ID configuration layer under formal control, creating a defined baseline, visibility of every deviation, gated change, and the ability to return to a known-good state.

The first attempt used the open-source PowerShell DSC platform. Over roughly four months and approximately \$1.7 million of engineering investment, two structural problems emerged.

4

months of build effort on
PowerShell DSC

~\$1.7M

engineering investment

0

full tenant restores
completed

PROBLEM 1

The model did not scale

A single-threaded execution model became prohibitively slow at this object count.

PROBLEM 2

Restore was never achieved

Full tenant configuration restore could not be completed, even after the team created a custom variant of the DSC codebase, narrowed the scope to Conditional Access alone, and raised the problem with Microsoft's product team.

THE CONCLUSION

Continued investment would not meet the organization's requirements.

THE DECISION

Open a formal proof of value with CoreView, and make purchase conditional on validating every capability in their own environment.

THE PROBLEM TO SOLVE

Continuous change

A Microsoft 365 tenant of this size does not stand still. Conditional Access policies are tuned, mailbox rules are added, groups are nested, app registrations are consented to, sharing settings are relaxed for one project and never tightened again. At the enterprise level, that adds up to a steady stream of configuration change every working day. Individually, they may not be dramatic changes, and almost none would warrant a ticket. **Yet collectively, this is the process by which a hardened tenant stops being hardened.**

Security teams rarely have anything in place to help them fully track these changes, and three assumptions are usually made to bridge that gap. Unfortunately, none of them hold.

ASSUMPTION 1

"Audit logs cover it"

WHY IT DOESN'T HOLD

Microsoft's audit trail records that a change happened and who made it. It does not record whether the change should have happened. There is no notion of an approved state to compare against, no severity model, and nothing in the log can reverse anything.

ASSUMPTION 2

"Backup covers it"

WHY IT DOESN'T HOLD

Data backup restores content. Restoring a mailbox does not restore the Conditional Access exclusion that lets an attacker reach it, and no data-protection product carries a versioned definition of what the tenant policy set is supposed to look like.

ASSUMPTION 3

"Change management covers it"

WHY IT DOESN'T HOLD

It covers the changes that go through change management. At this scale, a meaningful share of daily change comes from automation, delegated admins, app consent, and Microsoft itself.

WHAT WAS MISSING

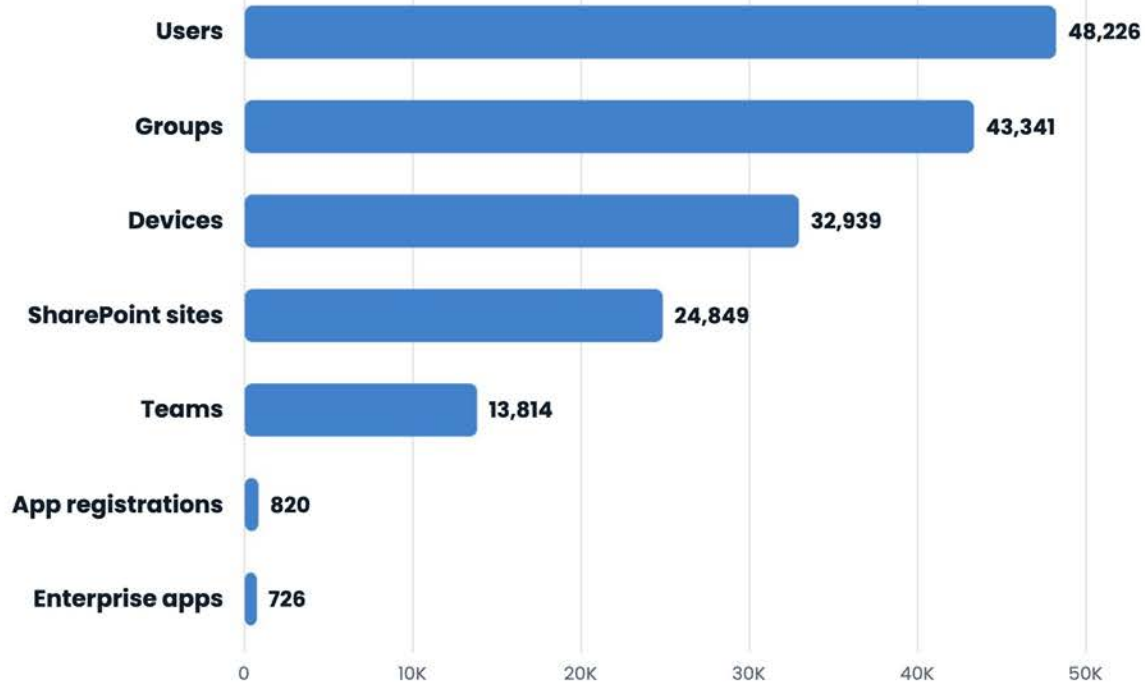
A control layer that continuously compares live configuration against an approved definition, alerts teams the moment the two diverge, and can pull the tenant back to a safe state when required.

THE SCALE OF THE CHALLENGE

Too large for manual review

The scale of the organization's configuration surface is what made manual review untenable. This is a list of objects under continuous monitoring in this environment.

Object counts under continuous monitoring

**164,000+**

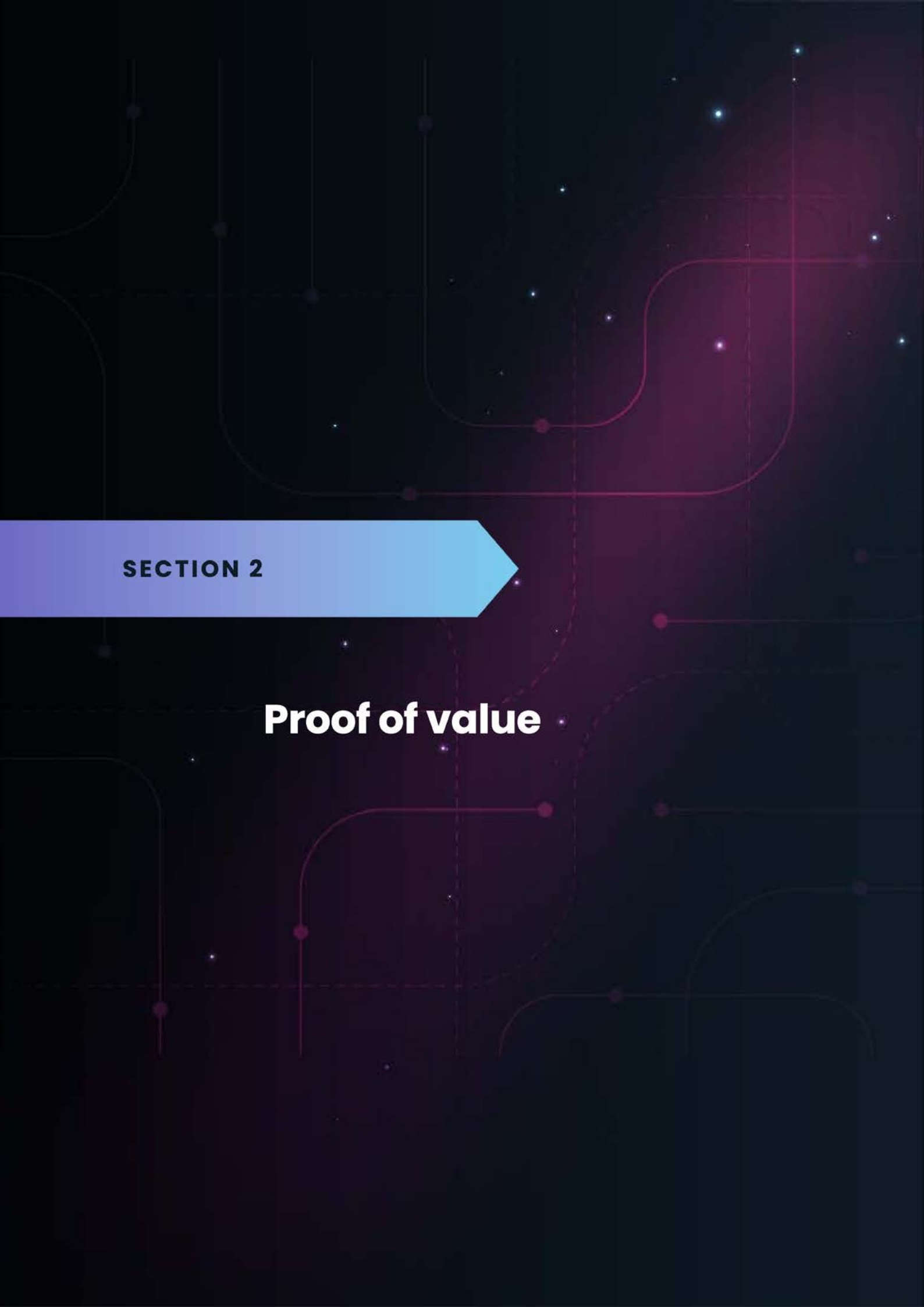
objects across seven types
under continuous
monitoring

48,226

Teams members in total:
37,074 internal and 11,152
external

1,546

app registrations and
enterprise apps combined

The background is a dark, deep blue or black gradient. It is decorated with a network of thin, glowing lines in shades of light blue and magenta. These lines are mostly vertical and horizontal, but many have rounded, curved ends, giving the impression of a digital or neural network. Small, bright dots of light are scattered throughout the background, some appearing as single points and others as small clusters. A horizontal bar with a blue-to-white gradient and a right-pointing arrow shape is positioned in the upper left quadrant.

SECTION 2

Proof of value

PROOF OF VALUE

Proving what DSC had failed to do, under load

Given the organization's prior experience, the PoV was scoped to focus on the specific things DSC had failed to do, and to prove them **under load rather than in a demonstration tenant**. Four elements were highlighted.

01 · SCALE

Every production object count was doubled for the test. The restore was therefore validated against roughly **96,000 users, 87,000 groups, and 50,000 SharePoint sites**, as well as proportionally scaled devices, Teams, app registrations, and enterprise apps.

02 · BREADTH

Users, groups, enterprise apps, devices, app registrations, SharePoint sites, and Teams, together with Conditional Access, mailbox policies, and SharePoint tenant settings, **rather than a single narrowed resource type**.

03 · COMPLETENESS

A full tenant configuration restore **run to completion**, not a partial or best-effort pass.

04 · PRE-PURCHASE

Every capability was **validated before any commercial commitment** was made.

THE HEADLINE RESULT

60,000+

 configurations restored in a single run

Complete in under 24 hours, at twice production scale. The comparison is what gives that number its weight: **the same organization with the same tenant had spent four months failing to complete a restore at half those object counts.**

SIDE BY SIDE

PowerShell DSC vs. CoreView Tenant Resilience

Same organization, same tenant, same requirement.

TESTED	OPEN-SOURCE POWERSHELL DSC	COREVIEW TENANT RESILIENCE
Full tenant configuration restore	Never achieved, including after scope was narrowed to Conditional Access only	60,000+ configurations restored in a single run
Restore time	Jobs routinely timed out without completing	Completed in under 24 hours
Execution model	Single-threaded; degraded sharply with object count	Multi-threaded; sustained throughput at load
Scale under test	Struggled at production counts	A system that could take 2× the load of the production scale
Elapsed effort	4 months and \$1.7M with no working restore	Validated end to end during a formal PoV, before purchase

BEFORE

0

full restores in four months, at production object counts

WITH COREVIEW

60,000+

configurations restored in one run, at twice production object counts

The background is a dark, deep blue or black gradient. It is decorated with a network of thin, glowing lines in shades of light blue and magenta. These lines are mostly vertical and horizontal, but many have rounded, curved ends, giving the impression of a circuit board or a digital data network. Small, bright dots of light are scattered throughout, some appearing as single points and others as small clusters. The overall effect is futuristic and high-tech.

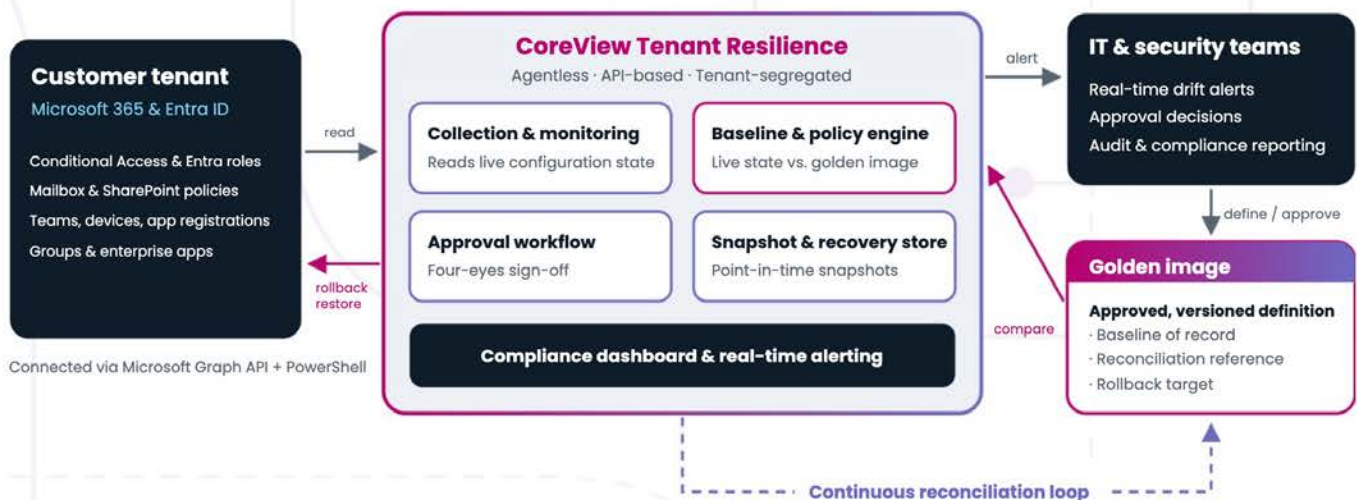
SECTION 3

How Tenant Resilience works

HOW COREVIEW TENANT RESILIENCE WORKS

Architecture and data flow

CoreView Tenant Resilience connects to the customer's Microsoft 365 and Entra ID tenant across a broad range of Microsoft API endpoints, including Microsoft Graph and PowerShell, to continuously read the tenant's configuration state. **The PowerShell path is important because it extends coverage to resource types where Microsoft has not yet shipped Graph support.** Live state is evaluated against defined baselines and, on an approved change, rollback or restore actions are written back into the tenant. All monitoring, approval, and recovery data is held in CoreView's own platform, isolated from the customer's Microsoft environment.



Drift detected against the golden image is alerted in real time, gated through four-eyes approval, and reverted to the approved baseline.

Figure 1. Collection, baseline reconciliation, approval, and write-back.

Collection and monitoring engine

Reads live configuration state across Conditional Access, mailbox policies, SharePoint, Teams, and Entra objects.

Baseline and policy engine

Compares live state against the golden image and classifies drift by severity.

Approval workflow engine

Enforces multi-level, four-eyes sign-off before any change reaches production.

Snapshot and recovery store

Holds point-in-time configuration snapshots used for targeted rollback and full-tenant restore.

Compliance dashboard and alerting

Real-time notification plus an auditor-facing view of compliance status.

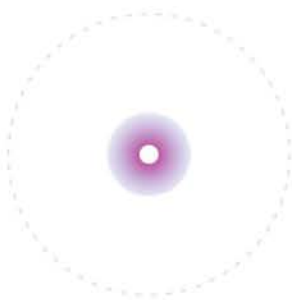
TENANT HARDENING

Baselines and the golden image

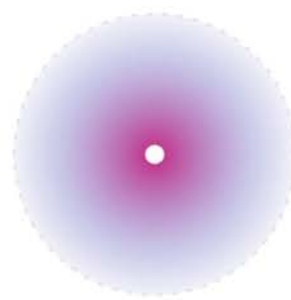
An approved, versioned definition of what the tenant's configuration is supposed to be.

The golden image is the standard the live tenant is continuously measured against, and the target that any unwanted change is pulled back toward.

Its value is blast radius arithmetic. **The smaller the gap between live state and the golden image at any given moment, the less ground a single bad change can cover before something catches it.**

**RECONCILED DAILY**

Blast radius measured in **hours of drift**.

**DISCOVERED DURING AN INCIDENT**

Blast radius measured in **however long it has been since anyone last looked**.

GOLDEN IMAGE COVERAGE

- Conditional Access policies
- Mailbox policies
- SharePoint tenant and site-level settings
- Teams configuration
- Entra objects including roles and directory settings

OWNERSHIP IS SPLIT DELIBERATELY

The customer defines and maintains the baselines, because only they can say what correct looks like for their regulatory position.

CoreView owns collector coverage as Microsoft ships new resource types, so the golden image does not silently develop the blind spots that ended the DSC attempt.

DETECTION

Drift monitoring in practice

Detection is where daily changes stop being a statistic and become an operational workflow. Tenant Resilience compares live settings against the golden image and raises drift whether or not the change was authorized, because **authorized-but-undocumented change is a compliance finding in its own right** for an operator in this sector.

HOW THE ORGANIZATION RUNS DETECTION

- **Near real-time alerting on high-impact change**

The organization defined granular alerts for the changes carrying the most risk, Conditional Access modifications and mailbox forwarding rules among them, routed in real time to both the Microsoft 365 team and the security team.

- **Daily summary for the long tail**

Lower-priority drift is batched into a daily report rather than triggering an immediate alert, which is what keeps the alerting channel credible.

- **Severity classification**

Drift is classified rather than listed, so volume is triaged before it reaches a human.

- **Compliance dashboard**

A centralized view of compliance status that surfaces critical deviations for auditors and IT security without a data pull.

- **Documented response**

CoreView professional services worked with the team to build a runbook covering automated rollback procedures via Tenant Resilience, so detection has a defined next step attached to it.

STEP 01**Detect**

Live state vs. golden image

STEP 02**Classify**

Severity, not volume

STEP 03**Route**

Real-time alert or daily summary

STEP 04**Respond**

Runbook and rollback

WHAT WE LEARNED FROM SIMILAR SIZED TENANTS

Triage by severity, not by volume

In large enterprise Microsoft 365 tenants of comparable size, on the order of 90,000 managed objects, we consistently observe close to 100 configuration changes a day. Roughly one in twenty-five of those touches a security-relevant control: some thirty items a week that a human has to look and rule on.

~90K

managed objects

~100configuration
changes a day**1 in 25**touch a security-
relevant control**~30**items a week for a
human to rule on

Where change happens vs. where risk sits

Approximate shares, as observed

All configuration change

App registrations ≈ half

All other change

Security-relevant change

Privileged roles + Conditional Access > ¾

Other

Within the security-relevant slice, over three-quarters concentrates in two areas alone: privileged role assignment and Conditional Access. The single largest source of raw volume is app registrations, which account for close to half of all change and carry almost none of the security weight.

That distribution is the argument for classification over listing. Triage by volume and the attention goes to app registrations. Triage by severity, and it goes to the few percent that can actually move the tenant's security posture, with the remainder arriving in the next morning's summary.

CONTROL

Four-eyes approval and audit trail

Detection tells you a change happened. Change control decides whether it happens at all.

FOUR-EYES GATES

No single administrator can push a configuration change to production. Every modification routes through a defined approval chain inside the CoreView UI and requires at least two independent approvers.

AUDIT LOGGING

Every request, every approval and every resulting configuration change is recorded, giving internal governance and external auditors a single continuous record.

ENFORCED REVERSION

Settings that are unauthorized or misconfigured are returned to the last known good state rather than negotiated after the fact.

For a nuclear operator, segregation of duties is not a preference expressed in a policy document. **Enforcing it in the product rather than in process is what makes it demonstrable during an audit.**

CONTAINING THE BLAST RADIUS

Roll back and restore

Roll back operates at two scopes, and they are different operations with different numbers attached.

TARGETED ROLLBACK

Minutes

A single policy, setting, or object is returned to its last approved state. This is the routine case and it is what contains blast radius day to day: MFA re-enabled, a forwarding rule removed, a Conditional Access exclusion reversed, all done minutes after detection rather than weeks after exploitation.

FULL-TENANT RESTORE

60,000+ <24h

The multi-threaded model restored 60,000+ configurations in a single run, in under 24 hours, against a tenant loaded to twice production object counts. That is the capability four months of DSC engineering could not deliver at half the scale, validated in the customer's own environment before purchase.

Between the two sits the point of the whole system: because drift is detected in near real time, the overwhelming majority of reversions are targeted rollbacks of a handful of objects. **The full-tenant capability exists so that in rare cases a genuine attack is survivable; the daily discipline is what keeps it rare.**

PLATFORM SECURITY AND DATA PROTECTION

Reviewed on its own terms

Granting a third-party platform broad read and write access to Conditional Access, mailbox policy and Entra objects is itself a security decision. For this organization it was reviewed on its own terms before any capability discussion, under the following key areas.

ACCESS MODEL

The CoreView admin UI authenticates through the customer's own Microsoft identity provider, so any MFA or SSO policy already enforced for tenant users applies automatically to CoreView access. There is no separate authentication path to secure.

DATA PROTECTION

Customer data is encrypted at rest with AES 256-bit encryption and in transit with TLS 1.2 or above, logically segregated by tenant so it never blends with other customers' data.

SEGREGATION OF DUTIES

Enforced in-product through the four-eyes approval workflow. No single administrator can push a change to production unilaterally.

AUDIT TRAIL

Every request, approval and configuration change is logged, supporting internal governance and regulatory audit requirements.

CERTIFICATIONS

ISO 27001

ISO 27018

ISO 9001

SOC 2

SOC 3

SCOPE, ASSUMPTIONS AND PREREQUISITES

● Licensing prerequisites

Microsoft 365 E1, E3, E5, E7, F1 or F3, or Entra ID Premium P1 or P2 (or the equivalent G-series licenses for government environments). Entra ID P1 is required for Conditional Access lock-down.

● Object types validated in this engagement

Users, groups, enterprise apps, devices, app registrations, SharePoint sites, and Teams. Coverage for other object types should be scoped separately.

● Operational ownership

The customer owns baseline definition and maintenance, the approver list, and the decision on every pending approval. CoreView owns the platform and collector coverage as Microsoft ships new resource types.

● Results

Figures vary by tenant size, object complexity, and existing configuration hygiene. The numbers in this paper reflect this customer's environment and proof-of-value conditions.

CONCLUSION

From unmanaged exposure to a controlled operating condition

The configuration layer of a large Microsoft 365 tenant changes hundreds of times a day. An organization that can define what correct looks like, see every deviation from it within minutes, gate change behind independent approval, and reverse anything unwanted, at any scale, has converted that volume from an unmanaged exposure into a controlled operating condition.

This organization tried to build that control and could not. What it bought instead was validated against twice its own production estate before a contract was signed and now runs as daily practice:

Fewer conditions under which something goes wrong

A smaller blast radius when it does

A proven way back

To discuss the configuration layer of your own tenant, or the practices behind this deployment, contact your CoreView account team.

Talk to CoreView →

ABOUT COREVIEW

The Microsoft 365 resilience platform

CoreView is the Microsoft 365 resilience platform, helping IT and security teams govern, automate and protect their tenant environment as a single auditable discipline.

Within the platform, CoreView Tenant Resilience is built for organizations that cannot afford unmanaged change in their configuration layer. Where native tooling leaves gaps in baseline definition, drift visibility, change control and rollback, Tenant Resilience continuously monitors configuration state across Conditional Access, Entra ID, SharePoint and Teams, flagging every deviation from an approved baseline, in near real time.

When something does go wrong, a misconfiguration, a failed change, or a compromise, CoreView reverses it, whether it is a single setting or restoring the entire configuration state, at scale.

See CoreView Tenant Resilience

Baselines, drift detection, four-eyes change control and rollback for your Microsoft 365 and Entra ID configuration layer.

coreview.com →

Talk to CoreView

Walk through the configuration layer of your own tenant and the practices behind this deployment with your account team.

[Book a demo](#) →