CTO Sunday Thoughts #1

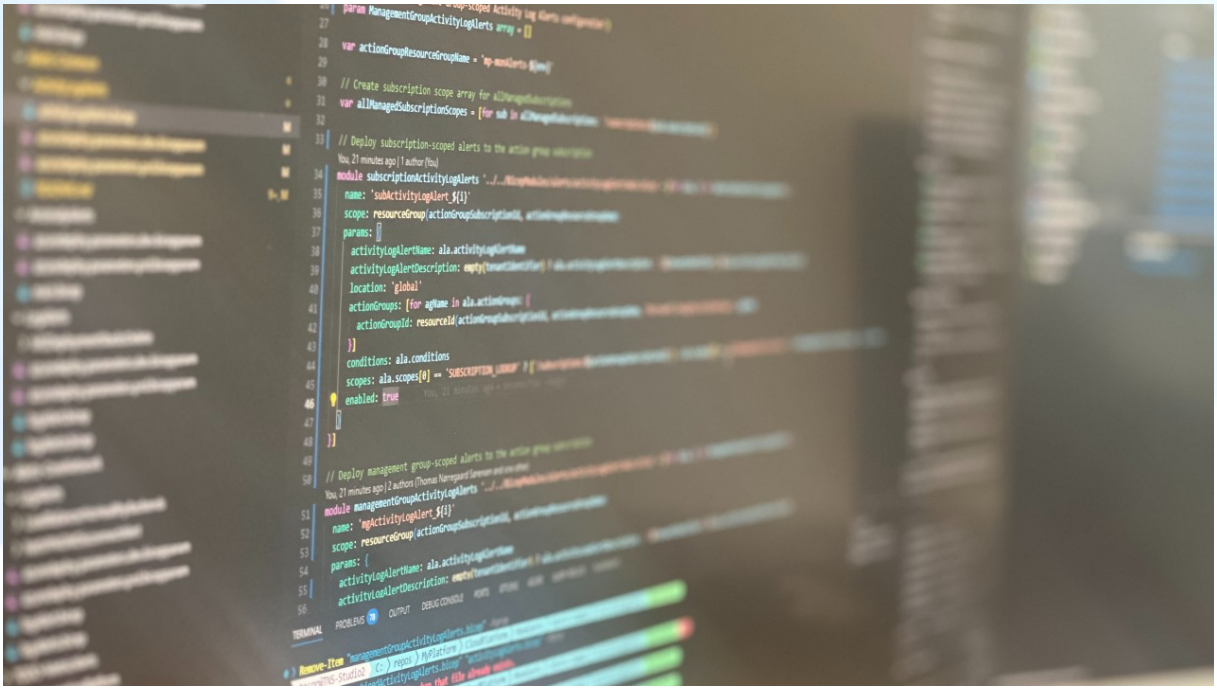
Recurring performance, Security, and operational challenges in Azure deployments

Analysis for IT leaders and Cloud Architects

This report presents key observations from a CTO perspective regarding recurring issues in Azure environments as they evolve over time. It focuses on common challenges that impact performance, security, and overall reliability, including silent governance drift, identity and access vulnerabilities, and the operational backlog, which IT leaders and cloud architects must address for sustainable cloud success.

As a CTO, I've guided many organizations on their journey to the Azure cloud. The initial phase is almost always filled with optimism. The promise of agility, scalability, and innovation is compelling. Teams deploy their first workloads, the business sees new capabilities, and the move feels like an unqualified success. But the most profound risks in the cloud don't appear on day one. They creep in over months and years through a process I call "silent drift", a slow, insidious decay of governance, security, and financial control.

Many of our partners and new customers come to us not because their initial Azure deployment failed, but because their ongoing *operation* is becoming chaotic, expensive, and insecure. This article outlines the four primary areas where things go wrong over time and provides a framework for maintaining long-term health in your Azure environment.



CTO Sunday Thoughts #1

1. The Erosion of Governance: From guardrails to suggestions

The most common failure point is the gradual erosion of your governance framework. What starts as a well-architected landing zone with clear "guardrails" slowly devolves into a set of mere suggestions.

Policy and Configuration Drift

The journey begins with a solid set of Azure Policies designed to enforce security and tagging standards. However, under the pressure of project deadlines, a team requests a "temporary" exemption for a specific resource group. That exemption is never revoked. Soon after, another developer, facing a blocker, makes a manual change through the Azure portal instead of using the approved Infrastructure-as-Code (IaC) pipeline. This is the classic "ClickOps" problem.

Let this go on for too long, and it becomes impossible to reconcile the deployed state with the intended state in code. Your policies are still assigned, but so many resources are exempt or misconfigured that your compliance dashboard is a sea of red. Your guardrails are no longer enforcing anything; they are just documenting failure.

The Remediation backlog

While some policies can auto-remediate issues, many require manual intervention on the workload itself. The customer is ultimately accountable for fixing non-compliant workloads. Without a dedicated team, these remediation tasks pile up, creating a massive backlog of known security gaps and configuration debt that no one has the time to address.

At **MyPlatform**, we address this by delivering policies as code and, crucially, providing automated remediation tasks for common issues, such as fixing broken VM extensions. While the customer still owns their workload compliance, we handle the automation that prevents the platform itself from contributing to the problem.

2. The identity maze: When access control becomes a mystery

Cloud Success or Silent Sabotage? Navigating the Hidden Hazards of Azure Deployments

Identity is the new security perimeter, and over time, that perimeter can become dangerously porous. The customer is fully responsible for managing every aspect of identity, from Entra ID security to PIM and RBAC assignments. This is where silent drift can be most damaging.

Privilege Creep

An employee starts in a junior role with limited access. They move to a new team and are granted additional permissions. They then lead a special project and get elevated access to more subscriptions. A year later, they have accumulated a wide array of high-level permissions that are no longer required for their day-to-day job. This "privilege creep" creates a massive and unnecessary attack surface.

Neglecting Just-in-Time Access A core tenet of modern security is the principle of least privilege, which includes time-bound access. We deploy PIM-enabled groups from day one, but the customer must manage all approvals and access reviews. We often see organizations fall back into the old habit of assigning permanent administrative roles. This leaves highly privileged accounts "standing" and active 24/7, making them a prime target for attackers.

Stale Identities Without rigorous and regular access reviews, the accounts of former employees, contractors, or old automated service principals can remain active long after they should have been disabled, posing a significant security risk.

A managed platform should provide the *tools* for good identity hygiene, like PIM-enabled groups, but the organization must commit to the operational discipline of managing memberships, approvals, and reviews.

3. The 'Evergreen' challenge: Fighting platform obsolescence

Perhaps the most underestimated long-term problem is platform decay. Azure is not a static environment; it is "evergreen," with a relentless pace of updates, new features, and deprecated services. A platform built today will be obsolete in 18 months if it isn't actively maintained.

As we discussed with Microsoft, a huge portion of an expert team's time can be consumed just "making the plumbing work." This includes:

- Updating IaC templates to use new API versions.
- Rewriting deployment scripts because a CLI command has been deprecated.
- Integrating new Azure services into existing governance policies (e.g., ensuring all new PaaS services have logging enabled).
- Updating security baselines in Microsoft Defender for Cloud as new recommendations are released.

This creates a significant "maintenance tax" that consumes resources that should be focused on innovation. The customer either pays this tax by dedicating internal resources to it or their platform slowly becomes fragile, insecure, and incompatible with modern Azure capabilities. This is why our core model is a managed SaaS offering; we are **responsible and accountable for delivering these evergreen platform updates as part of the service**.

Avoiding silent drift is not about having a perfect deployment; it's about having a sustainable operational model built on a clear **shared responsibility framework**.

Automate the Foundation

The core GRC foundation—the management hierarchy, policies, logging, and security baselines—should be managed as code by a dedicated provider. This ensures it remains evergreen and consistent.

Empower the Customer The customer must own and operate what is unique to their business:

their identities, their data, their costs, and their workloads.

Establish Clear Roles (RACI)

A formal RACI document is not bureaucracy; it is a critical tool for clarity. It ensures that everyone knows who is **Responsible**, who is **Accountable**, and who needs to be **Consulted** or **Informed** for every aspect of the cloud environment.

Your cloud platform should be an accelerator, not an anchor of complexity. By embracing an automated, managed foundation and a clear shared responsibility model, you can mitigate the risks of silent drift and ensure your Azure investment continues to deliver business value for years to come.

Sunday 7th of September 2025

- Homepage – **https://www.myplatform.net**
- Azure Marketplace – **https://www.myplatform.net/Azure/**
- Company LinkedIn – **https://www.linkedin.com/company/myplatformnet**