Practical GRC Projects & Portfolio Building: Part 3 – The "Read-Only" Architect

1. Introduction: The Sandbox is the Job

In Parts 1 and 2, we focused on using the Cyber Range's "View ALL" visibility to discover assets and simulate a Second-Party Audit. Part 3 pivots to a more advanced persona: the **Cloud Governance Architect**.

In a real-world enterprise, GRC analysts rarely get "Global Admin" rights. They almost always operate with Reader permissions, just like you have in the Cyber Range. This constraint is not a limitation; it is the standard operating environment. The ability to design governance frameworks, draft "Policy-as-Code," and perform gap analyses *without* needing to touch the "Deploy" button is a highly specialized skill set.

This report outlines four advanced activities that leverage your current access to build a portfolio that demonstrates **DevSecOps** competency and high-level strategic risk management. We will utilize the "messy" state of the shared environment to simulate common enterprise challenges: Shadow IT, Compliance Drift, and Supply Chain Risk.

---

2. Project 1: The "Shadow IT" & Data Sovereignty Audit (Clause 5.3 & A.5.10)

2.1 The Theoretical Framework

Data Sovereignty—ensuring data resides within specific legal jurisdictions—is a critical requirement for industries like government, finance, and healthcare.

- **ISO 27001 Clause 5.3 (Organizational roles, responsibilities and authorities):** Requires that the organization defines who is allowed to do what. A "Data Sovereignty Policy" defines *where* they can do it.
- **ISO 27001 Annex A 5.10 (Acceptable use of information and other associated assets):** Mandates rules for the acceptable use of assets.[1] Deploying a database in Brazil when the policy states "Australia Only" is an Acceptable Use violation.

2.2 The Task: Policing the Borders

In a shared environment like the Cyber Range, users often deploy resources in random regions (e.g., East US) because it is the default, ignoring local latency or sovereignty requirements. Your goal is to detect these "Illegal Immigrants."

2.3 The Execution (KQL)

Use Azure Resource Graph to visualize the physical distribution of the infrastructure and then pinpoint the violators.

Query 1: The Global Heatmap

Code snippet

```
Resources

| summarize ResourceCount = count() by location
| render piechart
```

- *Action:* Run this in the Portal. Take a screenshot. It proves you can visualize infrastructure sprawl.[2]

Query 2: The Sovereignty Enforcer
This query identifies every resource not in the approved regions (e.g., Australia East/Southeast).

Code snippet

```
Resources

| where location!in~ ('australiaeast', 'australiasoutheast')
| project
    ResourceName = name,
    Type = type,
    Location = location,
    ResourceGroup = resourceGroup,
    Owner = tags['Owner']
```

| order by Location asc

- *Technical Note:* The !in~ operator is case-insensitive "Not In." This captures AustraliaEast, australiaeast, etc.[3]

2.4 The Artifact: Non-Conformity Report (NCR)

- **Artifact Name:** NCR-2025-03-Sovereignty.pdf
- **Finding:** "Identified 24 resources deployed in unapproved regions (West Europe, East US)."
- **Risk:** "Potential violation of the Australian Privacy Principles (APP) regarding cross-border disclosure of personal information."
- **Portfolio Bullet:** "Developed automated KQL scripts to detect data sovereignty violations across a multi-tenant Azure environment, identifying 20+ resources deployed in non-compliant regions (ISO 27001 Annex A 5.10)."

---

3. Project 2: "Policy-as-Code" Development (Clause 5.1 & A.5.9)

3.1 The Theoretical Framework

Modern GRC is not about writing PDF policies that no one reads; it is about writing code that enforces them.

- **ISO 27001 Annex A 5.9:** Requires an inventory of assets.[5]

- Clause 5.1: Demonstrates leadership commitment.1
  If you cannot enforce the policy (because you are Read-Only), you can still write the Azure Policy definition (JSON) that would enforce it. This shows you understand the implementation of controls, not just the theory.

3.2 The Task: Tagging Enforcement

You will design a policy that denies the creation of any resource group that lacks a "CostCenter" tag.

3.3 The Execution (JSON & KQL)

Action 1: The Policy Definition (JSON)
Save this as policy_cost_center_deny.json in your portfolio repo.

JSON

```
{
  "properties": {
    "displayName": "Deny Resource Groups without CostCenter Tag",
    "policyType": "Custom",
    "mode": "All",
    "parameters": {},
    "policyRule": {
      "if": {
        "allOf":",
          "exists": "false"
        }
      ]
    },
    "then": {
      "effect": "deny"
    }
  }
}
```

- *Key Concept:* The deny effect prevents non-compliant resources from ever being created. This shifts security "left."[6]

Action 2: The Audit Simulation (KQL)
Since you cannot active the deny policy, use KQL to find what would have been blocked.

Code snippet

```
ResourceContainers

| where type == "microsoft.resources/subscriptions/resourcegroups"
| where tags!has "CostCenter"
| project name, tags
```

3.4 The Artifact: Governance Repository

- **Artifact:** A GitHub repository named azure-governance-as-code containing your JSON

definitions and KQL audit scripts.
- **Portfolio Bullet:** "Designed and scripted custom Azure Policy definitions (JSON) to enforce asset tagging governance, validating the logic against live environment data to ensure compliance with ISO 27001 A.5.9."

---

4. Project 3: Backup & Resilience Audit (Clause A.8.14)

4.1 The Theoretical Framework

ISO 27001 Annex A 8.14 (Redundancy of information processing facilities): Requires that information processing facilities (like VMs) have sufficient redundancy to meet availability requirements.8
In a student environment, backups are rarely configured. This is a perfect "Gap Analysis" opportunity.

4.2 The Task: The Protection Gap

You need to compare the list of *all* VMs against the list of *protected* VMs.

4.3 The Execution (KQL)

This advanced query joins the RecoveryServicesResources (the Backup Vaults) with the Resources (the VMs) to find the gap.

Query 3: The Resilience Gap Analysis

Code snippet

```
Resources

| where type =~ 'microsoft.compute/virtualmachines'
| project vmId = tolower(id), vmName = name, resourceGroup
| join kind=leftouter (
    RecoveryServicesResources

| where type ==
"microsoft.recoveryservices/vaults/backupfabrics/protectioncontainers/protecteditems"
| where properties.backupManagementType == "AzureIaasVM"
| project vmId = tolower(tostring(properties.sourceResourceId)), protectionState =
properties.protectionState
) on vmId
```

```
| where isempty(protectionState) // Filter for VMs where the join failed (no backup found)
| project vmName, resourceGroup, Status = "Unprotected - Risk of Data Loss"
```

- *Logic:* We perform a leftouter join. Any VM that does not match a record in the Recovery Services table will have an empty protectionState. These are your at-risk assets.[9]

4.4 The Artifact: Resilience Gap Analysis Report

- **Artifact:** An Excel spreadsheet Resilience_Gap_Analysis.xlsx with a pivot table showing "Protected vs. Unprotected" assets.
- **Portfolio Bullet:** "Conducted a comprehensive Business Continuity audit, utilizing KQL to identify a 40% backup coverage gap in production Virtual Machines, directly addressing ISO 27001 Annex A 8.14 requirements."

---

5. Project 4: Supply Chain Risk Simulation (Clause 5.19)

5.1 The Theoretical Framework

ISO 27001 Annex A 5.19 (Information security in supplier relationships): Requires organizations to agree on information security requirements with suppliers.11
Treat the Cyber Range platform itself ("LognPacific") as a Third-Party Vendor. You are the client auditing them before signing a contract.

5.2 The Task: Vendor Due Diligence

Using your "View ALL" access, assess the security hygiene of the platform. Does it look like a secure place to store sensitive data?

5.3 The Execution (Observation)

Use your "Read-Only" observations to fill out a risk questionnaire.

- **Observation 1 (Isolation):** "I can see the Resource Group names and VM names of other tenants (students)."
  - *Risk:* Inadequate Multi-Tenancy Segregation. Information Leakage.
- **Observation 2 (Network):** "I see hundreds of Public IPs directly attached to VMs (from Project 2)."
  - *Risk:* Poor default network posture. High likelihood of platform IP reputation damage due to compromised neighbors.
- **Observation 3 (Tagging):** "Most resources lack Owner tags (from Project 1)."
  - *Risk:* Poor management oversight; risk of "noisy neighbor" resources consuming platform IOPS without attribution.

5.4 The Artifact: Third-Party Risk Assessment (TPRA)

- **Artifact:** A PDF report Vendor_Risk_Assessment_LognPacific.pdf.
- **Conclusion:** "High Risk. The platform lacks sufficient tenant isolation and network perimeter controls. Not suitable for PII or Confidential data."
- **Portfolio Bullet:** "Performed a Third-Party Risk Assessment (TPRA) on a cloud infrastructure provider, identifying critical gaps in tenant isolation and network security (ISO 27001 A.5.19)."

Summary of Deliverables

GRC Domain	Activity	ISO 27001 Control	Artifact
Sovereignty	KQL Region Audit	**A.5.10 / Clause 5.3**	Non-Conformance Report (NCR)
Governance	Azure Policy (JSON)	**A.5.9 / Clause 5.1**	GitHub Repo: azure-governance-as-code
Resilience	KQL Backup Gap Analysis	**A.8.14**	Gap Analysis Spreadsheet
Vendor Risk	Platform Hygiene Check	**A.5.19**	3rd Party Risk Assessment PDF

Ethical Note for Portfolio

As emphasized in previous parts, **redaction is mandatory**. When you create these artifacts (NCRs, Spreadsheets, Reports), you must replace real data with placeholders (e.g., Student-VM-01 -> ``). This demonstrates your commitment to **Confidentiality**, the first pillar of the CIA triad.