Enterprise Identity Architecture Showcase

STRATEGIC ENGINEERING PROPOSITION

Strategic Mission: Programmatic Identity Fleet Management

Automated Lifecycle Execution

Transitioning from manual, portal-based configurations to fully automated, **Python and CLI-driven lifecycle management** for enterprise non-human identities.

Cryptographic Federation at Scale

Engineering zero-secret pipeline infrastructure by dynamically mapping **OpenID Connect (OIDC) federated credentials** across isolated runner fleets, eliminating supply-chain vulnerabilities.

Dynamic Data-Plane Least Privilege

Executing automated Role-Based Access Control (RBAC) bounding strictly at the asset level (e.g., **Key Vault Secrets User**) to systematically dismantle control-plane blast radiuses.

Technical Competencies & Execution Mastery

Infrastructure-as-Code (IaC) Deployment

Provisioning single-tenant isolation boundaries, headless service principals, and federated trust relationships using **programmatic CLI execution and dynamic variable resolution**.

Hardened DevSecOps Orchestration

Architecting resilient multi-agent CI/CD workflows that enforce strict conditional access gates, **ephemeral JWT token exchange**, and zero-residual session teardowns.

Continuous Compliance & Telemetry Auditing

Parsing **Entra ID Control Plane logs** against **Azure Log Analytics Data Plane telemetry** to cryptographically prove end-to-end execution and continuous compliance alignment.

STRATEGIC POWER OF IAC MODERNIZATION

Transitioning legacy infrastructure to passwordless OIDC and data-plane RBAC via programmatic execution systematically eliminates critical attack vectors and establishes deterministic cloud governance.

`NIST 800-207 // SOC 2 TYPE II`

Macro Threat Landscape & Imperative

INDUSTRY THREAT BASELINE

The Systemic Failure of Legacy Credentialing

Before engineering a modern identity architecture, we must quantify the operational risk inherent in legacy cloud environments. Industry threat telemetry confirms that non-human machine identities represent the primary attack surface across enterprise infrastructure.

STATIC SECRET SPRAWL

Managing long-lived symmetric keys across hundreds of CI/CD pipelines creates an unmanageable attack surface. Manual rotation introduces operational friction and inevitable, catastrophic exposure windows.

SUPPLY-CHAIN VULNERABILITY

Hardcoded credentials within repositories, source code, and build environments provide threat actors with direct, low-resistance vectors for lateral movement into core cloud infrastructure.

CONTROL-PLANE OVER-PRIVILEGE

Relying on broad subscription-level roles (e.g., **Contributor** or **Owner**) for pipeline runners maximizes the blast radius, turning a single compromised agent into a subscription-wide breach.

BUSINESS RATIONALE ESTABLISHED

The following industry telemetry quantifies this systemic risk and establishes the absolute necessity for programmatic, zero-trust cryptographic federation.

ENTERPRISE CI/CD SECURITY MODERNIZATION

TRANSITIONING LEGACY MACHINE IDENTITIES TO ZERO-SECRET OIDC ARCHITECTURE

INDUSTRY VULNERABILITY BASELINE

THE PERMISSION GAP

99%

Over-Privileged Cloud Identities

Unit 42 research (analyzing 680,000+ cloud identities across 18,000 accounts) revealed that 99% of users, roles, and machine identities hold excessive permissions left completely unused.

Dominant Attack Surface

Unit 42 incident data confirms identity weaknesses and excessive privilege play a material role in the vast majority of cloud intrusions.

THE BREACH REALITY

44%

Identity-Driven Breaches

44% of all enterprise organizations have experienced a cloud data breach, with credential compromise, identity misconfigurations (31%), and unmanaged access serving as the leading threat vectors.

Compromise Amplification

Stolen credentials and identity abuse appear in **39% to 48%** of all confirmed breach chains across global enterprises.

INDUSTRY RESEARCH SOURCES:

OVER-PRIVILEGE METRICS: Palo Alto Networks Unit 42 Cloud Threat Report (IAM Research: 680,000+ identities audited).

BREACH METRICS: Thales Global Cloud Security Study & Verizon DBIR.

Legacy Architecture Flaws

SYSTEMIC VULNERABILITY BASELINE

STATIC SECRET DEPENDENCY

Legacy CI/CD authentication relies on long-lived symmetric secrets (e.g., **2-year lifespans**). This introduces heavy manual rotation overhead and creates massive, unmanaged exposure windows.

SUPPLY CHAIN LEAKAGE

Storing static application passwords directly within repository settings or CI/CD build environment variables leaves core cloud infrastructure highly vulnerable to **pipeline leaks and credential extraction**.

SUBSCRIPTION BLAST RADIUS

Pre-modernization machine identities are frequently over-privileged at the **subscription scope**, often holding overly broad administrative roles (such as Contributor) rather than zero-trust scoping.

UNRESTRICTED AUTHORITY

If an underlying pipeline secret is compromised, the attacker gains **unrestricted operational authority** over every cloud asset within the blast radius, rather than being contained by tightly scoped data-plane RBAC.

THE STRATEGIC IMPERATIVE

These compounding systemic flaws necessitate an immediate transition to an automated, zero-secret OIDC architecture governed by granular data-plane access controls.

STRATEGIC REMEDIATION METHODOLOGY FOR NON-HUMAN MACHINE IDENTITIES

AI & CLOUD PIPELINE HARDENING FRAMEWORK (ACPHF)

ARCHITECTED TO REPLACE EXISTING NON-SECURE BASELINE CONFIGURATIONS

1. **PHASE 1: Core Cloud Boundary & Metadata Alignment**
   (Isolates Tenant, Subscription, and Project IDs)
2. **PHASE 2: Non-Human Identity Provisioning & Tenancy Architecture**
   (Application and Service Principal Object Creation)
3. **PHASE 3: Passwordless Cryptographic Federation**
   (Establishes passwordless OpenID Connect (OIDC) trust scenarios)
4. **PHASE 4: Data Plane Access Control & Pipeline Execution**
   (Data-Plane Role Mapping and Handshake Sequence)

PHASE 1: CORE CLOUD BOUNDARY & METADATA ALIGNMENT

ESTABLISHING DETERMINISTIC ROUTING COORDINATES

IDENTITY ROOT BOUNDARY

TENANT ID

Microsoft Entra ID (Root Identity Authority)

INFRASTRUCTURE RESOURCE BOUNDARY

SUBSCRIPTION ID

Azure Resource Scope (Target Asset Environment)

boundary_context.sh (Azure CLI)

```
# Phase 1: Deterministic Boundary Resolution
# Explicitly set execution context to prevent cross-tenant leaks

# 1. Target Identity Root (Microsoft Entra ID Tenant)
az login --tenant "9439dd25-f3b5-4829-a76f-5ede8cd54c3c"

# 2. Lock Infrastructure Scope (Azure Subscription)
az account set --subscription "d5ffd8a5-d994-4eb5-b87c-4442054d233e"

# 3. Verify Active Boundary Matrix
az account show \
  --query "{Tenant:tenantId, Subscription:id}" \
  --output table
```

Architectural Rationale: Before issuing tokens or assigning RBAC roles, the programmatic execution environment must be explicitly hard-locked to the exact Entra ID Tenant and Azure Subscription coordinates, neutralizing cross-tenant routing vulnerabilities.

PHASE 2: NON-HUMAN IDENTITY PROVISIONING SEQUENCE

DIRECTORY SCHEMA CREATION & ATTACK SURFACE MINIMIZATION

GLOBAL APPLICATION BLUEPRINT

Defines core identity schema strictly bound to internal tenancy.

AzureADMyOrg (Single-Tenant)

Redirect URIs = [BLANK]

LOCAL SERVICE PRINCIPAL CONTEXT

Instantiates local enterprise identity for RBAC assignment.

✓ **Enforced Headless Execution**

✓ **Interactive Login Blocked**

`>_ provision_identity.sh (Azure CLI)`

```
# 1. Register Headless Application Blueprint
# Enforces Single-Tenant boundary. Omitting reply-urls
# inherently denies browser callbacks.
APP_ID=$(az ad app create \
  --display-name "acphf-agent-01" \
  --sign-in-audience "AzureADMyOrg" \
  --query appId -o tsv)

# 2. Instantiate Local Security Context
az ad sp create --id $APP_ID

# 3. Verify Attack Surface Minimization
az ad app show --id $APP_ID \
  --query "{Audience:signInAudience,
WebRedirects:web.redirectUris}"
```

Architectural Rationale: By explicitly binding the identity schema to a Single-Tenant boundary and intentionally omitting Redirect URIs, we strip away all interactive login vectors at inception. This enforces a strictly headless execution context, fundamentally shrinking the attack surface before cryptographic federation is even applied.

PHASE 2: NON-HUMAN IDENTITY PROVISIONING SEQUENCE

MULTI-AGENT DIRECTORY SCHEMA CREATION & ATTACK SURFACE MINIMIZATION

provision_agents.py (Infrastructure-as-Code)

```python
import subprocess
import json

# Define our enterprise fleet of AI agents
agents = ["acphf-agent-1", "acphf-agent-2", "acphf-agent-3"]

print("Starting Phase 2: Multi-Agent Provisioning...")

for agent in agents:
    print(f"\nProvisioning: {agent}")

    # Enforcing Single Tenant (AzureADMyOrg) boundary.
    # Redirect URIs are blank by default for headless
execution.
    command = [
        "az", "ad", "app", "create",
        "--display-name", agent,
        "--sign-in-audience", "AzureADMyOrg"
    ]

    result = subprocess.run(command, capture_output=True,
```

Azure Cloud Shell Execution Output

```
steven [ ~ ]$ python3 provision_agents.py

Starting Phase 2: Multi-Agent Provisioning...

Provisioning: acphf-agent-1
SUCCESS: acphf-agent-1 App ID → 7b3e9122-ff26-4610-be9b-
7819c74fecf4

Provisioning: acphf-agent-2
SUCCESS: acphf-agent-2 App ID → 7ae389b6-28ee-4e9f-8a07-
35d98ebce6be

Provisioning: acphf-agent-3
SUCCESS: acphf-agent-3 App ID → d33f899f-5122-44b3-aec5-
7cb23a203059

steven [ ~ ]$
```

PHASE 3: CRYPTOGRAPHIC FEDERATION SEQUENCE

OIDC TRUST ESTABLISHMENT, IMMUTABLE MAPPING & VALIDATION GATES

🚔 SUBJECT CLAIM GUARD (INVARIANTS)

Enforces absolute case-sensitive matching and zero-wildcard boundaries to eliminate AADSTS700213 mismatch errors and claim spoofing.

- 🔢 @171821203 (Immutable DB ID)
- 🚫 Wildcards (*) Explicitly Disabled

↓

⏱ EPHEMERAL TOKEN EXCHANGE

Translates the external JWT into a strictly scoped, short-lived Entra ID Access Token, forcing continuous runtime re-authentication.

- ✓ 60-Min Access Token Ceiling
- ✓ 0 Static Secrets Provisioned

</> fic_parameters.json & CLI Execution

```
# 1. Immutable Claim Mapping Payload (JSON)
{
  "name": "github-actions-oidc-trust",
  "issuer": "https://token.actions.githubusercontent.com",
  // Hardened Subject: Binds directly to the immutable
  // repository ID (@171821203) instead of string name
  "subject": "repo:Compcode1/acphf-agent@171821203:ref:refs/heads/main",
  "audiences": [
    "api://AzureADTokenExchange"
  ]
}
# 2. Programmatic FIC Enforcement (Azure CLI)
az ad app federated-credential create \
  --id $APP_ID \
  --parameters @fic_parameters.json
```

Architectural Rationale: By mapping trust directly to an immutable repository database ID (@171821203) and enforcing absolute case-matching, we eliminate the risk of repository renaming attacks and claim spoofing, ensuring a zero-trust cryptographic boundary.

PHASE 3: PASSWORDLESS CRYPTOGRAPHIC FEDERATION

AUTOMATED OIDC TRUST ESTABLISHMENT & IMMUTABLE CLAIM MAPPING

federate_agents.py (Infrastructure-as-Code)

```
# Building the OIDC payload mapped directly to your ACPHF
Slide 12 standards
fic_payload = {
    "name": f"{agent}-github-oidc",
    "issuer": "https://token.actions.githubusercontent.com",
    "subject": "repo:Compcode1/workload-identity-
oidc:ref:refs/heads/main",
    "description": f"Multi-agent CLI managed OIDC trust for
{agent}",
    "audiences": ["api://AzureADTokenExchange"]
}

# Execute the CLI command to establish the federated
credential
command = [
    "az", "ad", "app", "federated-credential", "create",
    "--id", app_id,
    "--parameters", "@fic_params.json"
]
```

Azure Cloud Shell Execution Output

```
steven [ ~ ]$ python3 federate_agents.py
Starting Phase 3: Passwordless Cryptographic Federation ...

Federating: acphf-agent-1
SUCCESS: Federated credential locked for acphf-agent-1

Federating: acphf-agent-2
SUCCESS: Federated credential locked for acphf-agent-2

Federating: acphf-agent-3
SUCCESS: Federated credential locked for acphf-agent-3

Phase 3 Execution Complete.
steven [ ~ ]$
```

PHASE 4: DATA-PLANE ACCESS CONTROL ISOLATION

AUTOMATED BULK RBAC SCOPING & DYNAMIC RESOURCE RESOLUTION

role_assignment.py (Infrastructure-as-Code)

```python
# Using exact plain-text role name prevents UUID parsing errors
kv_secrets_user_role = "Key Vault Secrets User"
key_vault_name = "KV-SECURE-DATA"

# Dynamically fetch the Key Vault Resource ID via CLI
kv_command = ["az", "keyvault", "show", "--name",
key_vault_name, "--query", "id", "-o", "tsv"]

# ... (Service Principal validation logic) ...

# Execute the role assignment using the dynamically fetched
scope
assign_command = [
    "az", "role", "assignment", "create",
    "--assignee-object-id", sp_object_id,
    "--assignee-principal-type", "ServicePrincipal",
    "--role", kv_secrets_user_role,
    "--scope", kv_resource_id
]
```

Azure Cloud Shell Execution Output

```
steven [ ~ ]$ python3 role_assignment.py
Starting Phase 4: Data-Plane Least Privilege Enforcement...
Dynamically fetching Resource ID for Key Vault: KV-SECURE-DATA
Target Scope Acquired: /subscriptions/d5ffd8a5-d994-4eb5-b87c...

Assigning Key Vault Secrets User role to: acphf-agent-1
SUCCESS: Data-plane access locked for acphf-agent-1

Assigning Key Vault Secrets User role to: acphf-agent-2
SUCCESS: Data-plane access locked for acphf-agent-2

Assigning Key Vault Secrets User role to: acphf-agent-3
SUCCESS: Data-plane access locked for acphf-agent-3

Phase 4 Execution Complete.
steven [ ~ ]$
```

PHASE 4: DATA PLANE ACCESS CONTROL & PIPELINE EXECUTION SEQUENCE

ISOLATION ROLES, OIDC HANDSHAKE ORCHESTRATION & TELEMETRY LEAK PREVENTION

BLOCK 4.1: PLANE ISOLATION

WORKLOAD SERVICE PRINCIPAL
ID: e7e0822c-a39c-4325...

↓

CONTROL PLANE (MANAGEMENT SCOPE)

❌ **Bypassed: Azure Contributor**
(Broad Subscription Write/Delete Power)

✅ **Hardened: Broad Management Revoked**
(Overprivileged Admin Roles Stripped)

↓

DATA PLANE (OPERATIONAL SCOPE)

Assigned Least-Privilege Roles:
- Key Vault Secrets User
- Storage Blob Data Reader

✅ **STATUS: LEAST-PRIVILEGE BOUNDARY**

BLOCK 4.2: ORCHESTRATION

UNIVERSAL HANDSHAKE SEQUENCE

1. Token Request	Outbound JWT Request
2. Handshake	Assertion Transmission
3. Validation	Public Signature Match
4. Token Exchange	Cloud Access Token Issued

MANDATORY CRYPTOGRAPHIC GATE

id-token: write

Enables OIDC JWT issuance
directly inside execution context

Headless Runner Authorized

BLOCK 4.3: HARDENING & MASKING

VOLATILE TOKEN RULE

AT: 60 min Ceiling | RT: 0 min

No long-lived refresh tokens.
Pipeline re-authenticates per run.

TELEMETRY MASKING TRAP

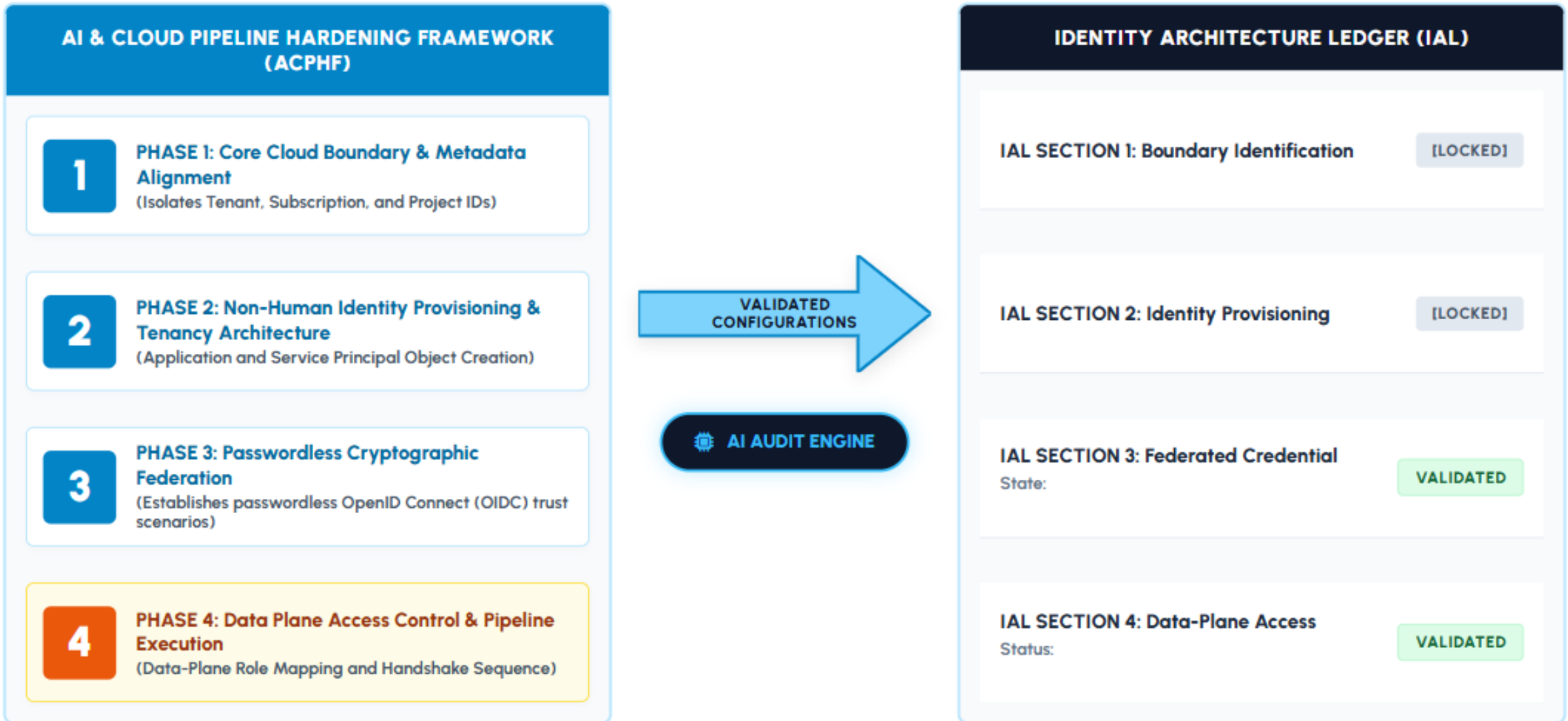
Automated Scanners
Catch raw unmasked dynamic outputs

Runtime Masking Directives
***** REDACTED *****

In-memory suppression prevents log leak

AI & CLOUD PIPELINE HARDENING FRAMEWORK (ACPHF)

ARCHITECTURE TO CONFIGURATION LEDGER MAPPING



IDENTITY ARCHITECTURE LEDGER (IAL)

MASTER CONFIGURATION STATE FILE // AI-INGESTION FORMAT v2.4

EXECUTION_TS: 2026-08-05T10:19:09Z
INGESTION_AGENT: HOOK_ENABLED (ACTIVE)
CHECKSUM: 9a3e2d1a0b9c8d7e6f5a4b3c2d1e0f9

TARGET CLOUD ENV	PIPELINE RUNNER	IAC EXPORT TARGET	AUDIT STATE
AZURE_COMMERCIAL	GITHUB_ACTIONS_UBUNTU_LATEST	BICEP_JSON_TEMPLATE	VERIFIED

1. CORE STRUCTURAL BOUNDARY DISCOVERY

1.1 IDENTITY & ASSET MATRICES

SRC_TENANT_ID:	[PENDING_MANUAL_EXTRACTION]
TGT_SUB_ID:	d5ffd8a5-d994-4eb5-b87c ...
TGT_ASSET_URI:	KV-SECURE-DATA

1.2 HARD BOOLEAN GATE: DIRECTORY ALIGNMENT

[TRUE] : TENANT_ID == ASSET_TENANT_ROOT

2. NON-HUMAN IDENTITY PROVISIONING (ENTRA ID)

APP_DISPLAY_NAMES:	acphf-agent-1, acphf-agent-2, acphf-agent-3
APPLICATION_ID_1:	7b3e9122-ff26-4610-be9b-7819c74fecf4
APPLICATION_ID_2:	7ae389b6-28ee-4e9f-8a07-35d98ebce6be
APPLICATION_ID_3:	d33f899f-5122-44b3-aec5-7cb23a203059
SIGN_IN_AUDIENCE:	AzureADMyOrg (Single-Tenant Enforced)
REPLY_URLS:	[] (BLANK - HEADLESS RUNNER OPTIMIZED)

3. CRYPTOGRAPHIC FEDERATION (OIDC PARAMETERS)

FIC_CREDENTIAL_NAME:	{agent}-github-oidc (Multi-Agent Scope)
ISSUER_URL:	https://token.actions.githubusercontent.com
AUDIENCE_MAPPING:	api://AzureADTokenExchange
SUBJECT_CLAIM_STR:	repo:Compcode1/workload-identity-oidc:ref:refs/heads/main
WILDCARD_STATUS:	0 WILDCARDS DETECTED (EXPLICIT PATH ENFORCED)

[TRUE] : FEDERATED_TRUST_POLICY_LOCKED

4. DATA-PLANE RBAC & PIPELINE HARDENING

RBAC_DEF_ID_1 (KV):	Key Vault Secrets User
CTRL_PLANE_ROLES:	0 DETECTED (CONTRIBUTOR BYPASSED)
TOKEN_LIFECYCLE:	AT: 3600s | RT: BLOCKED_0s
MASKING_DIRECTIVE:	::add-mask::${{ secrets.KV_PAYLOAD_VAR }}

PHASE 5: RESOURCE-SIDE EXECUTION & TELEMETRY

AUTOMATED CI/CD PIPELINE VERIFICATION & DATA-PLANE VALIDATION

validate-security-multi-agent.yml (Infrastructure-as-Code)

```yaml
jobs:
  build-and-deploy:
    runs-on: ubuntu-latest
    steps:
      # Authenticate securely without credentials
      - name: Azure OIDC Login
        uses: azure/login@v2
        with:
          client-id: 7b3e9122-ff26-4610-be9b-7819c74fecf4
          tenant-id: 9439dd25-f3b5-4829-a76f-5ede8cd54c3c
          subscription-id: d5ffd8a5-d994-4eb5-b87c-4442054d233e

      # Test granular data-plane extraction
      - name: Verify Key Vault Access
        run: |
          az keyvault secret show \
          --name "ACPHF-OIDC-Success" \
          --vault-name "KV-SECURE-DATA" \
          --query "value" -o tsv
```

GitHub Runner Execution Output

```
Run azure/login@v2
Running Azure CLI Login.
Attempting Azure CLI login by using OIDC...
Subscription is set successfully.
Azure CLI login succeeds by using OIDC.

Run az keyvault secret show --name "ACPHF-OIDC-Success"...
Cryptographic-Federation-Verified

Run az account clear
Clearing azure cli accounts from the local cache.

Phase 5 Execution Complete.
```

Resource-Side Execution Breakdown

GITHUB ACTIONS & AZURE PIPELINE VERIFICATION

validate-security (GitHub Actions Log) Run #142 • 11s

Set up job 0s

```
Runner: ubuntu-24.04
Complete job name: build-and-deploy
```

Checkout Repository 0s

```
Syncing Compcode1/workload-identity-oidc
```

Azure OIDC Login 7s

```
Federated token details requested via OIDC JWT
Subject: repo:Compcode1@171821203/workload ...
Azure CLI login succeeds by using OIDC.
```

Verify Key Vault Data-Plane Access 3s

```
Executing az keyvault secret show --name "ACPHF-OIDC-
Success" ...
Successfully retrieved secret: Cryptographic-Federation-
Verified
```

Post-Execution & Session Teardown 1s

```
Executing /usr/bin/az account clear ...
Clearing azure cli accounts from the local cache.
Zero residual state. Job completed.
```

1. Pipeline Initialization & Code Scope

Set up job (0s): Provisions an isolated `ubuntu-24.04` runner with ephemeral storage and downloads necessary action dependencies.

Checkout Repository (0s): Syncs source code from `Compcode1/workload-identity-oidc` securely into the temporary workspace.

2. Cryptographic Authentication & Data-Plane Access

Azure OIDC Login (7s): Requests an ephemeral OIDC JWT token matching the exact subject claim: `repo:Compcode1@171821203/ ... :ref:refs/heads/main`. Negotiates **passwordless token exchange** with Microsoft Entra ID via `api://AzureADTokenExchange`.

Verify Key Vault Access (3s): Assumes Key Vault Secrets User RBAC role to query KV-SECURE-DATA and retrieves data-plane payload:
Successfully retrieved secret value: Cryptographic-Federation-Verified

3. Zero-Trust Session Teardown

Post Azure OIDC Login (1s): Executes `/usr/bin/az account clear` to immediately purge the token cache and revoke CLI credentials.

Post Checkout & Complete Job (0s): Flushes workspace directory, unsets authentication headers, and terminates runner processes—leaving zero residual state.

Control-Plane Governance & Identity Audit

MICROSOFT ENTRA ID TELEMETRY

DATE ↓ | REQUEST ID | SERVICE PRINCIPAL ID | STATUS | IP ADDRESS | RESOURCE | CONDITIONAL ACCESS

User sign-ins (interactive) | User sign-ins (non-interactive) | **Service principal sign-ins** | Managed identity sign-ins

Date ↓	Request ID	Service principal ID	Service principal name	Status	Cross tenant access type	Home tenant ID	Resource tenant ID	IP address	Resource	Resource ID	Conditional ac...	# sign ins
8/4/26, 11:44:36 AM	b9395618-e56d-4237-a282-f263...	e7d34e69-2c62-40ab-a830-a4a5...	acphf-agent-1	Success				20.83.159.19	Azure Key Vault	cfa8b339-82a2-471a-a3...	Not applied	1
8/4/26, 11:44:36 AM	b9395618-e56d-4237-a282-f263...	e7d34e69-2c62-40ab-a830-a4a5...	acphf-agent-1	Success	None			20.83.159.19	Azure Key Vault	cfa8b339-82a2-471a-a3...	Not applied	1
8/4/26, 11:37:50 AM	09c40739-7152-4946-812b-ff301...	e7d34e69-2c62-40ab-a830-a4a5...	acphf-agent-1	Success				52.234.47.122	Azure Key Vault	cfa8b339-82a2-471a-a3...	Not applied	1
8/4/26, 11:37:50 AM	09c40739-7152-4946-812b-ff301...	e7d34e69-2c62-40ab-a830-a4a5...	acphf-agent-1	Success	None			52.234.47.122	Azure Key Vault	cfa8b339-82a2-471a-a3...	Not applied	1
8/4/26, 11:37:46 AM	4de0784a-5473-4ef0-87e8-e361...	e7d34e69-2c62-40ab-a830-a4a5...	acphf-agent-1	Success				52.234.47.122	Azure Resource Manager	797f4846-ba00-4fd7-ba...	Not applied	1
8/4/26, 11:37:46 AM	4de0784a-5473-4ef0-87e8-e361...	e7d34e69-2c62-40ab-a830-a4a5...	acphf-agent-1	Success	None			52.234.47.122	Azure Resource Manager	797f4846-ba00-4fd7-ba...	Not applied	1
8/4/26, 11:33:04 AM	0ed4788e-ab5c-4539-9023-bb21...	e7d34e69-2c62-40ab-a830-a4a5...	acphf-agent-1	Failure				64.236.131.247	Azure Resource Manager	797f4846-ba00-4fd7-ba...	Not applied	1
8/4/26, 11:33:04 AM	0ed4788e-ab5c-4539-9023-bb21...	e7d34e69-2c62-40ab-a830-a4a5...	acphf-agent-1	Failure	None			64.236.131.247	Azure Resource Manager	797f4846-ba00-4fd7-ba...	Not applied	1

1. Inbound Coordinates & Perimeter

Target Security Principal: `acphf-agent-1` (`e7d34e69 ...`). Directory coordinate verification confirms inbound Application ID matches active tenant objects identically.

Runner Egress IP Attribution: IP `52.234.47.122` confirms the execution originated directly from Microsoft/GitHub hosted runner infrastructure.

2. Progression (4-Sec Delta)

Control-Plane Handshake (11:37:46 AM): Azure Resource Manager successfully receives the OIDC assertion, validating the subject claim and issuing the CLI token.

Asset-Level Request (11:37:50 AM): Exactly 4 seconds later, Key Vault records a discrete issuance event, mirroring the fast GitHub Actions execution timeline.

3. Cryptographic Audit Readiness

Passwordless Verification: 100% ephemeral token exchange achieved via OIDC. Zero static client secrets or long-lived passwords were transmitted across the wire.

Immutable Audit Trail: Unique Request IDs (e.g., `09c40739 ...`) provide end-to-end cryptographic traceability for NIST 800-207 Zero Trust compliance.

Key Slide Takeaway: Dual-Perspective Alignment

This identity-side telemetry definitively confirms that the directory successfully validated the external GitHub runner, granted scoped short-lived access, and immutably logged every transaction—including a documented failure—without ever storing or exposing static credentials.

Data-Plane Asset Audit & Telemetry Tracing (Log Analytics)

AZURE LOG ANALYTICS TELEMETRY

Azure CLI Log Analytics Query Output

```
steven [ ~ ]$ az monitor log-analytics query \
  --workspace "$LAW_ID" \
  --analytics-query "AzureDiagnostics | where TimeGenerated > ago(1h) | project TimeGenerated, Resource, OperationName, ResultSignature, httpStatusCode_d, CallerIPAddress | sort by TimeGenerated d
  esc" \
  -o table
```

CallerIPAddress	OperationName	Resource	ResultSignature	TableName	TimeGenerated	HttpStatusCode_d
20.29.29.50	SecretGet	KV-SECURE-DATA	OK	PrimaryResult	2026-08-01T21:17:04.4112869Z	200
20.29.29.50	Authentication	KV-SECURE-DATA	Unauthorized	PrimaryResult	2026-08-01T21:17:04.0828041Z	401
13.87.216.116	SecretGet	KV-SECURE-DATA	OK	PrimaryResult	2026-08-01T21:11:22.672438Z	200
13.87.216.116	Authentication	KV-SECURE-DATA	Unauthorized	PrimaryResult	2026-08-01T21:11:22.0912712Z	401

```
steven [ ~ ]$
```

1. Data-Plane Verification & Payload Delivery

Target Vault Asset: `KV-SECURE-DATA`

Operation Executed: `SecretGet`

HTTP Outcome: 200 OK (`ResultSignature` : OK)

Operational Meaning: The machine identity successfully presented its scoped access token to the Key Vault data-plane, passed Key Vault Secrets User RBAC evaluation, and extracted the secret payload (`ACPHF-OIDC-Success`).

2. OAuth Challenge Protocol Analysis

401 Unauthorized Challenge: Key Vault returns an unauthenticated 401 response to declare its directory tenant authority.

200 OK Authenticated Response: The GitHub runner immediately presents the OIDC-negotiated Bearer token, completing the authorization handshake in less than a second.

Audit Gate 4 (Plane Check) Status: PASS—Zero "Silent 403" authorization drops or missing role definitions detected.

3. Network & Infrastructure Traceability

Egress Runner IPs: `13.87.216.116` and `20.29.29.50` trace directly to GitHub Actions Azure-hosted runner ranges.

Telemetry Schema Alignment: Diagnostic logging successfully routed to `AzureDiagnostics` within law-workload-identity-audit, **establishing a permanent, queryable audit trail for compliance frameworks (NIST 800-207, SOC 2 Type II).**

Key Slide Takeaway

Complete Circuit Execution: By proving directory token issuance on **the Entra ID Control Plane** and secret payload extraction on the **Key Vault Data Plane**, we have established end-to-end cryptographic proof of a hardened, **passwordless** machine identity pipeline.

Enterprise Security Engineering: Audit Results Ledger (ARL)

ACPHF / IAL V2.1 VERIFICATION

AUDIT TRACKING ID	EXECUTION TIMESTAMP (UTC)	AUDITOR / AI AGENT	TARGET ASSET / CLIENT ID	
ARL-2026-0801-WORKLOAD-OIDC	2026-08-01 21:17:04 UTC	Steven Tuschman (AI Security Assistant)	KV-SECURE-DATA (e7e0822c...)	OVERALL: PASS

[GATE 1] Coordinate Check — PASS

Target: Inbound Tenant & App ID Mapping
Source: Entra ID Non-Interactive Sign-In Logs

Directory coordinates (`9439dd25...`) and App ID (`e7e0822c...`) match active tenant objects identically. Zero routing errors (AADSTS90002).

[GATE 2] Character Check — PASS

Target: Subject Claim Casing & Trust
Source: Sign-In Status (ErrorCode: 0)

Subject claim string matches repository path and modern immutable DB IDs (`repo:Conpcode1@171821203/...`) character-for-character.

[GATE 3] Clock Check — PASS

Target: Token Lifetime (60m Ceiling)
Source: Session Lifecycle Spacing

Ephemeral OIDC handshake executed during 27s pipeline run. Short-lived access ceiling enforced with zero stored refresh tokens.

[GATE 4] Plane Check — PASS

Target: Control vs. Data-Plane RBAC
Source: Log Analytics Workspace

SecretGet executed successfully (`HTTP 200 OK`) against target vault `KV-SECURE-DATA` with zero 403 Forbidden access drops.

Section 3: Data-Plane Tracing & AI Intent Directive Log

Audit Circumstance: Correlating OIDC federated token issuance with Key Vault data-plane access to detect unauthorized reads or "Silent 403" drops.

AI Prompt Directive Executed: "Inspect active Log Analytics workspace (law-workload-identity-audit). Join ServicePrincipalSignInLogs for App ID e7e0822c-a39c-4325-8fec-25b014fcb0c3 with Key Vault data-plane access logs (AzureDiagnostics) for target vault KV-SECURE-DATA within a 5-minute execution window. Verify OperationName == 'SecretGet' and httpStatusCode_d == 200 to confirm 100% data-plane access with zero orphan events."

Dynamic Query Summary: Query executed against active schema; confirmed 100% correlation between directory token issuance (12:25:17 PM) and Key Vault secret release (21:17:04Z). Zero orphan events.

Section 4: Defect Log & Remediation Action Plan Matrix

Gate 1 (Coordinates)	Gate 2 (Characters)	Gate 3 (Clock)	Gate 4 (Plane)
Defect: None	Defect: Corrected (AADSTS700213)	Defect: None	Defect: None
Root Cause: App & Tenant IDs matched objects on run.	Root Cause: GitHub OIDC enforces DB owner IDs.	Root Cause: Token generated and consumed in 27s.	Root Cause: SecretGet succeeded without drops.
Action: Locked Target App Client ID to e7e0822c...	Action: Updated sub claim via CLI to match payload.	Action: Volatility lifetime governed by workflow context.	Action: Assigned Key Vault Secrets User RBAC directly.

Audit State Locked: YES Framework: ACPHF / IAL v2.1 Next Scheduled Review Date: 2026-11-01