Domino API Password Rotation with HashiCorp Vault

Version 2025-09-15 20:08 UTC

# 1. Purpose

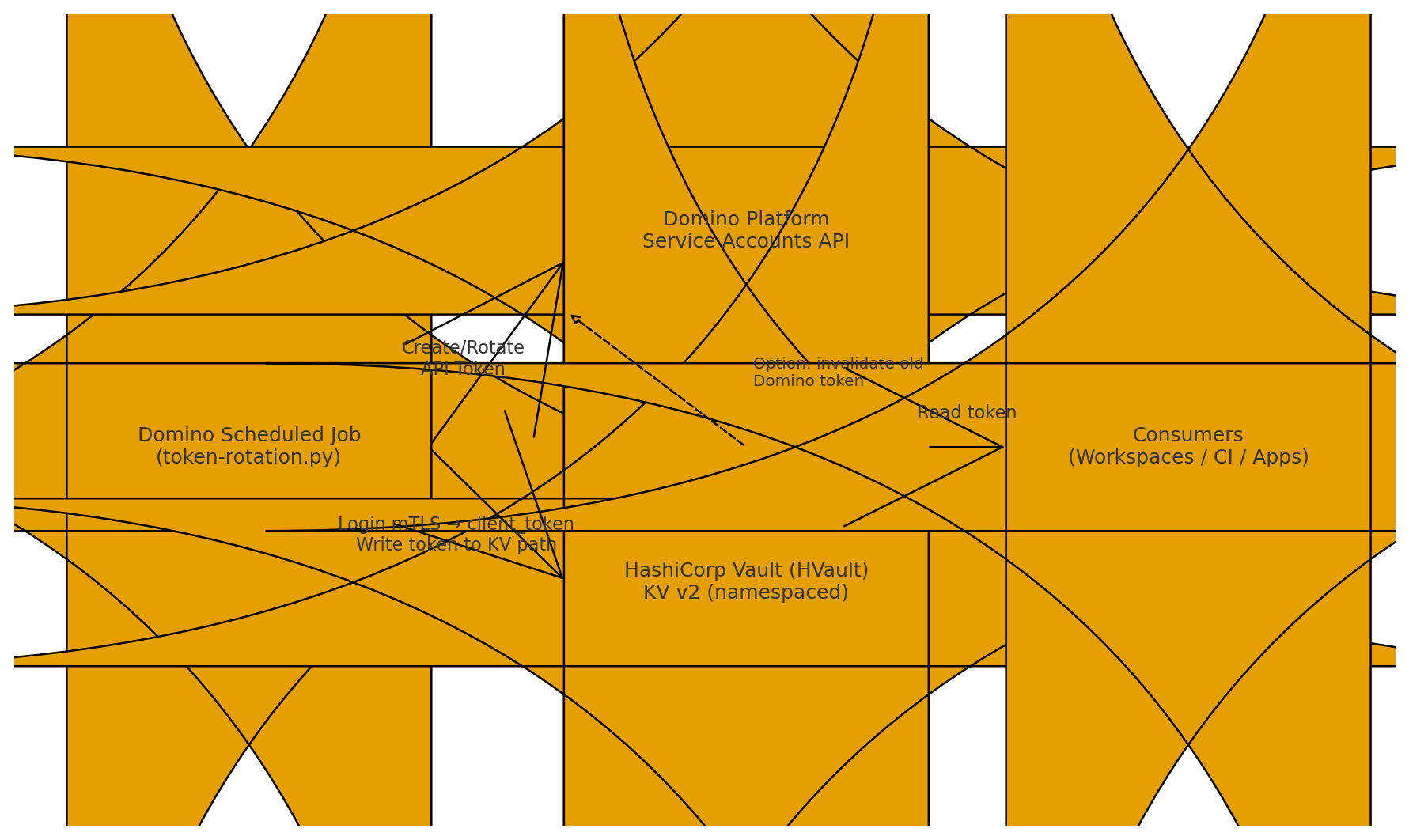
This document describes the automated rotation of Domino API access credentials using a Domino Scheduled Job and HashiCorp Vault (HVault). The solution generates a new Domino Service Account API token, stores it in Vault under a controlled path, and enables consumers to securely retrieve the token.

# 2. Scope

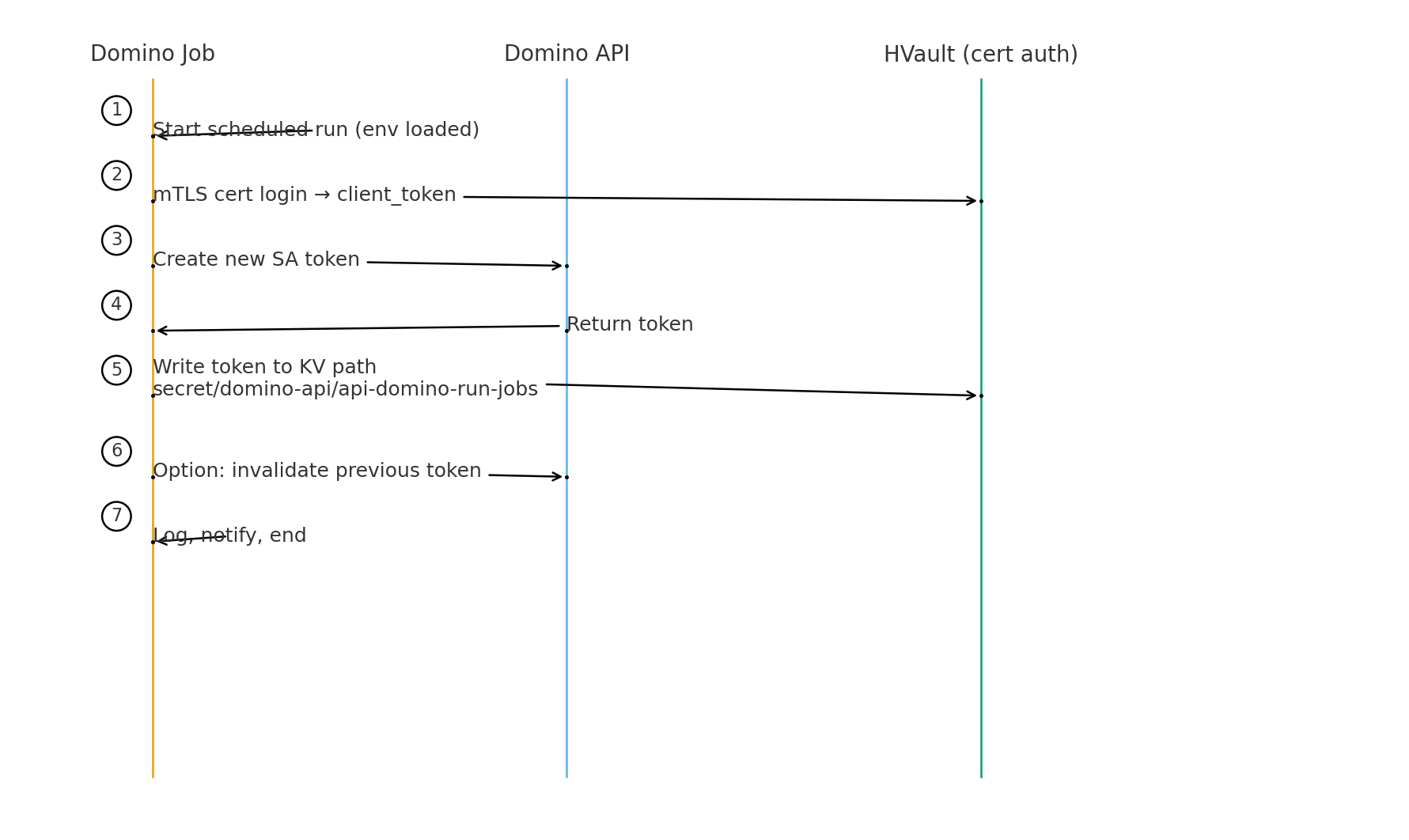
Covers: Domino Service Accounts, token generation, Vault certificate-based authentication, KV storage, Domino Scheduled Jobs configuration, and operational checks. Out of scope: CI/CD consumers' integration details.

# 3. Architecture Overview

High-level components:  
• Domino Scheduled Job runs a Python script.  
• Domino API issues a new Service Account token (≈ 3 months TTL).  
• HVault authenticates via client certificate (mTLS) to obtain a client token.  
• The script writes the Domino token into the Vault KV path used by downstream consumers.  
• (Optional) The script can invalidate the previous Domino token.



# 4. Rotation Flow



# 5. Prerequisites

- Domino user/API key with permission to manage Service Accounts.  
- HashiCorp Vault Enterprise (namespaces) or OSS, with the 'cert' auth method enabled.  
- A client certificate/key pair mapped to a Vault role with appropriate policies.  
- A KV secret engine mounted (e.g., 'secret') – preferably KV v2.  
- Network access from the Domino compute to HVault and Domino API.

# 6. Configuration

Environment variables expected by the rotation script:

DOMINO\_URL=https://<domino-host>/v4  
DOMINO\_API\_KEY=<domino-admin-or-ops-token>  
SA\_NAME=<service\_account\_name>  
SA\_EMAIL=<service\_account\_email>  
  
VAULT\_ADDR=https://hvault.<env>.domain:8200  
VAULT\_NAMESPACE=<namespace/if/enterprise>  
VAULT\_CLIENT\_CERT=/path/to/client.crt  
VAULT\_CLIENT\_KEY=/path/to/client.key  
VAULT\_CACERT=/path/to/ca\_chain.crt  
VAULT\_KV\_MOUNT=secret  
VAULT\_SECRET\_PATH=domino-api/api-domino-run-jobs  
VERIFY\_SSL=true

# 7. Script Overview

The script performs the following steps:  
1) Locate or create the Service Account.  
2) Create a new Domino API token for the Service Account.  
3) Authenticate to Vault via the cert auth method to obtain a client token.  
4) Write (upsert) the key 'api\_domino\_run\_jobs' to 'secret/domino-api/api-domino-run-jobs' (KV v2), or fallback to KV v1 if configured.  
5) (Optional) Invalidate the previous Domino token.  
6) Log outcome and exit non‑zero on failure.

# 8. Vault Policy (example for KV v2)

path "secret/data/domino-api/\*" {  
 capabilities = ["create", "update", "read", "list"]  
}  
path "secret/metadata/domino-api/\*" {  
 capabilities = ["read", "list"]  
}

# 9. Domino Scheduled Job

Create a Scheduled Job in the Domino project that points to the rotation script:  
- Command: `python3 domino\_sa\_rotate.py` (or your wrapper).  
- Hardware tier: small/standard (script is lightweight).  
- Environment variables: provide the variables from Section 6 via project or job-level settings.  
- Schedule: e.g., every 80 days at 02:15 UTC, with retries on failure.  
- Notifications: enable email/Slack on failure.

# 10. Testing & Validation

Run the script once with `--dry-run` to validate Domino API access. Then test Vault connectivity with the provided verification helper which:  
• performs cert login,  
• checks capabilities via `sys/capabilities-self`,  
• reads the KV value to confirm the path and namespace.

# 11. Troubleshooting

• HTTP 403 from Vault: check `X-Vault-Namespace`, cert role policy, and that you are writing to KV v2 `/data/...`.  
• HTTP 404 from KV v2: the mount might be KV v1; the script falls back automatically.  
• SSL warnings: provide a valid CA bundle via `VAULT\_CACERT`; avoid `verify=False`.  
• Domino 409 on Service Account create: the account already exists; the script will reuse it.

# 12. Security Considerations

• Tokens are short‑lived (≈90 days) and rotated automatically.  
• Access to read the KV path should be restricted to only the consumers that need it.  
• Prefer Vault AppRole/namespace scoping for consumers rather than distributing the token broadly.  
• Audit log both Domino token creation and Vault writes.

# 13. API Endpoints (reference)

Domino:  
 POST {DOMINO\_URL}/v4/serviceAccounts/{id}/tokens  
Vault:  
 POST {VAULT\_ADDR}/v1/auth/cert/login  
 POST {VAULT\_ADDR}/v1/secret/data/domino-api/api-domino-run-jobs (KV v2)

— End of document —