



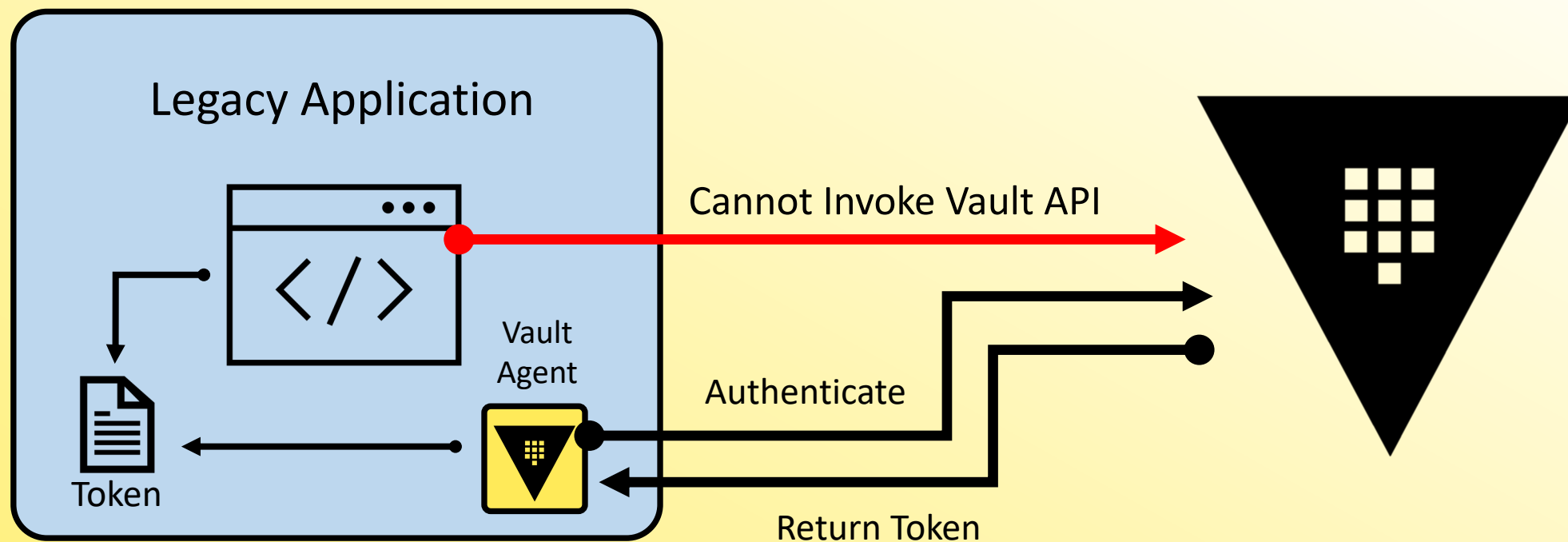
# Vault Agent: Configure Templating



# Legacy Applications – Auto-Auth



How can legacy applications still take advantage of Vault for secrets?



# Consul Template



- A standalone application that renders data from Consul or Vault onto the target file system
  - <https://github.com/hashicorp/consul-template>
  - Despite its name, Consul Template does **not** require a Consul cluster to operate
- Consul Template retrieves secrets from Vault
  - Manages the acquisition and renewal lifecycle
  - **Requires a valid Vault token to operate**



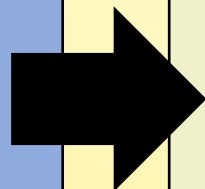
# Consul Template - Workflow



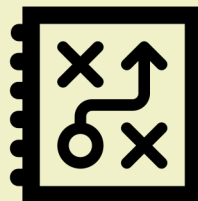
## Step 1



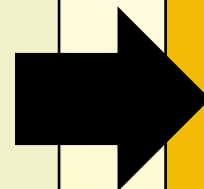
Create a templated file that specifies the path and data you want



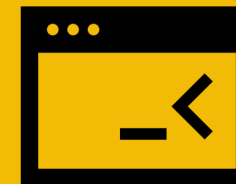
## Step 2



Execute Consul Template  
Consul Template retrieves data and renders to destination file



## Step 3



The application reads the new file at runtime that includes secrets retrieved from Vault



# Consul Template



## Create a templated file

```
production:
  adapter: postgresql
  encoding: unicode
  database: orders
  host: postgres.hcvop.com
  {{ with secret "database/creds/readonly" }}
  username: "{{ .Data.username }}"
  password: "{{ .Data.password }}"
  {{ end }}
```

The path in Vault to read secrets from

The data to be retrieved and consumed from the Vault response



# Consul Template – Destination File (Result)

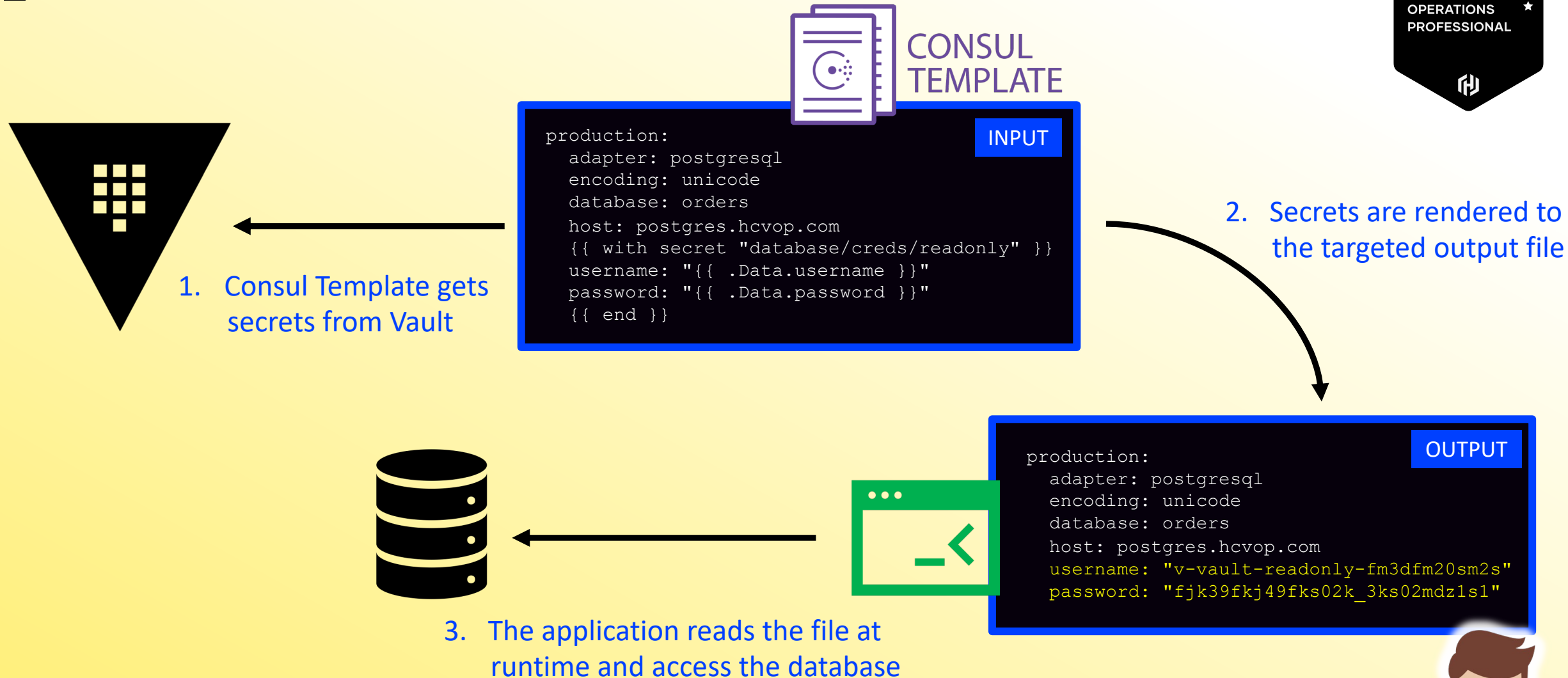


Destination File that application server will read at runtime

```
production:
  adapter: postgresql
  encoding: unicode
  database: orders
  host: postgres.hcvop.com
  username: "v-vault-readonly-fm3dfm20sm2s"
  password: "fjk39fkj49fks02k_3ks02mdz1s1"
```



# Consul Template - Workflow



# Vault Agent Templating

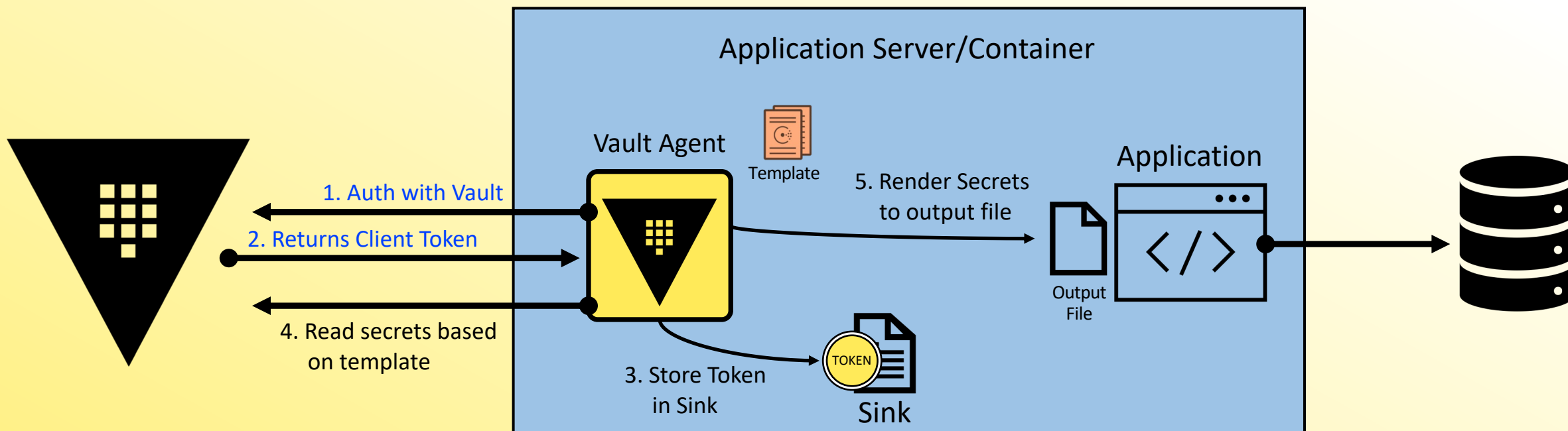


- To further extend the functionality of the Vault Agent, a subset of the [Consul-Template](#) functionality is directly embedded into the Vault Agent
  - No need to install the Consul-Template binary on the application server
- Vault secrets can be rendered to destination file(s) using the Consul-Template markup language
  - Uses the client token acquired by the auto-auth configuration





# Vault Agent Templating



# Template Configuration



```
auto_auth {
  method "approle" {
    mount_path = "auth/approle"
    ...

  sink "file" {
    config = {
      path = "/etc/vault.d/token.txt"
    }
    ...

  template_config {
    exit_on_retry_failure = true
    static_secret_render_interval = "10m"
  }

  template {
    source = "/etc/vault/web.tmpl"
    destination = "/etc/webapp/config.yml"
  }
}
```

} **Auto-Auth  
Configuration  
(AppRole)**

} **Sink  
Configuration**

} **Global Template  
Configurations  
(affects all templates)**

} **Template  
Configuration**





# END OF SECTION

 Vault

CERTIFIED  
OPERATIONS  
PROFESSIONAL

