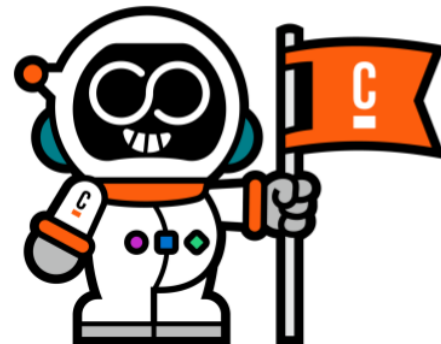


KEEPING SECRETS

MASTERING SECURE PROCESS AUTOMATION AT SCALE WITH VAULT





HEY, I'M **STEFAN** (HE/HIM)



Consultant & People Lead



Consid GmbH, Hamburg



Process Automation

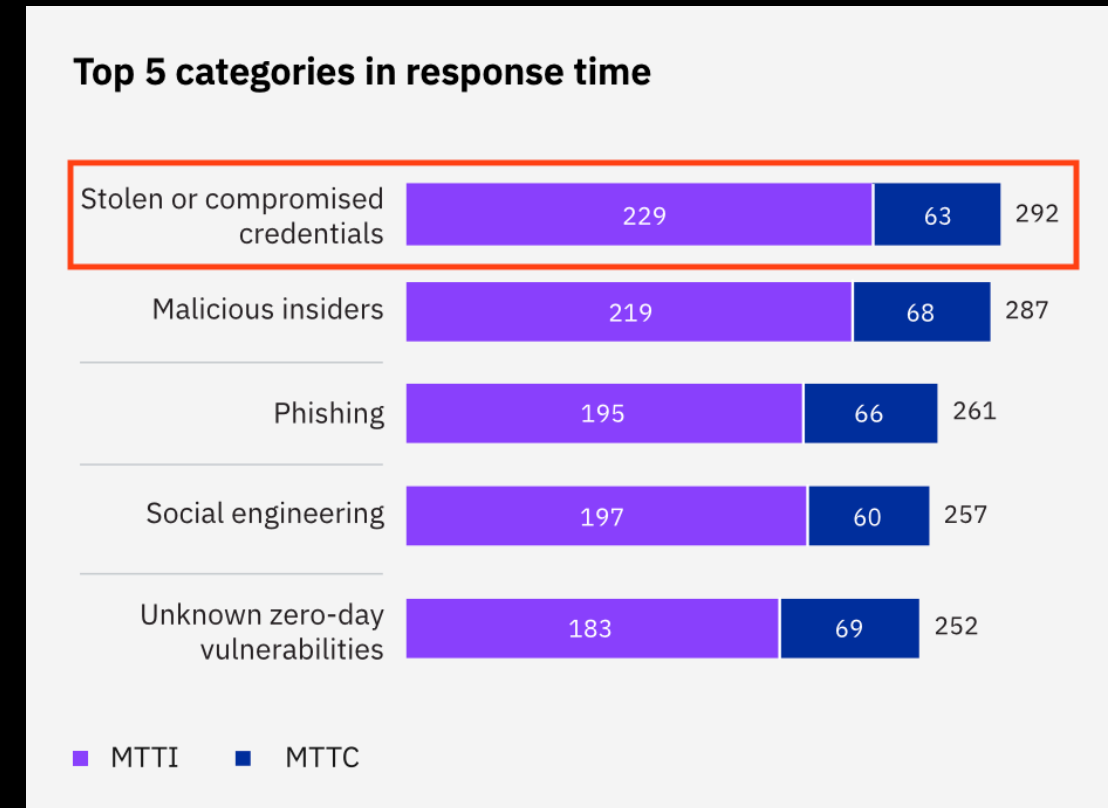
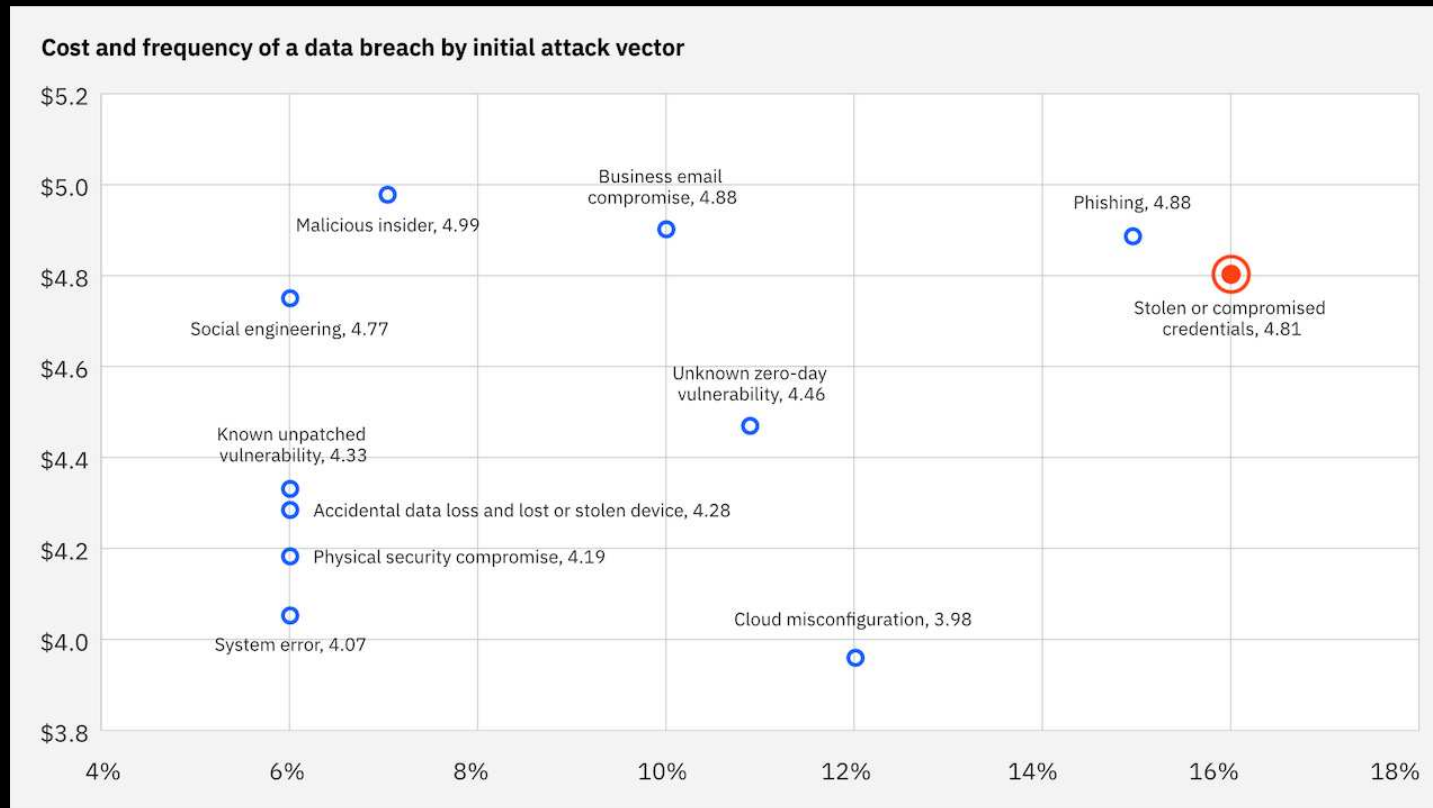


Software Architecture



Camunda Champion

IBM'S COST OF A DATA BREACH REPORT (2024)



👉 expensive & hard to contain

CUSTOMER SCENARIO

FACTS

highly regulated sector

camunda for orchestration

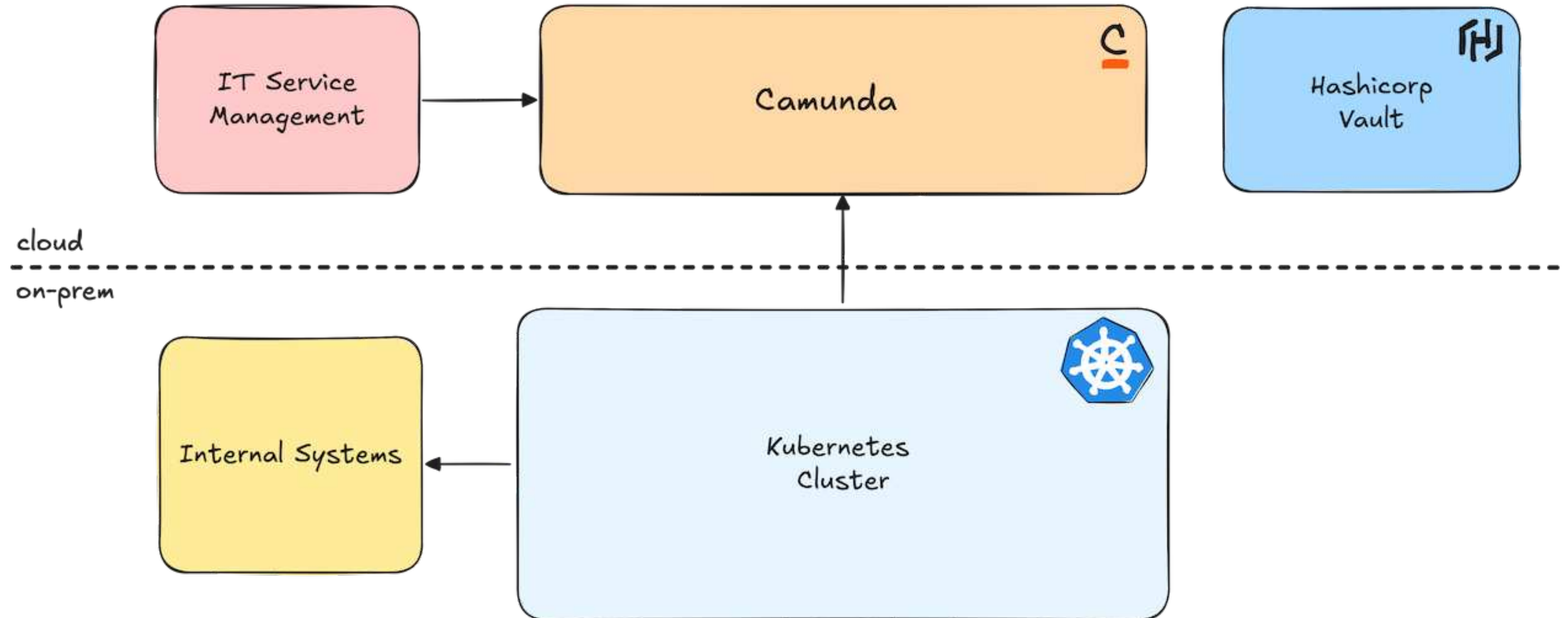
integration of many systems

take security very serious



Energy

INITIAL SYSTEM LANDSCAPE

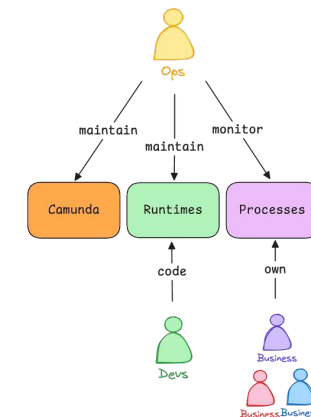


UNADDRESSED ISSUES

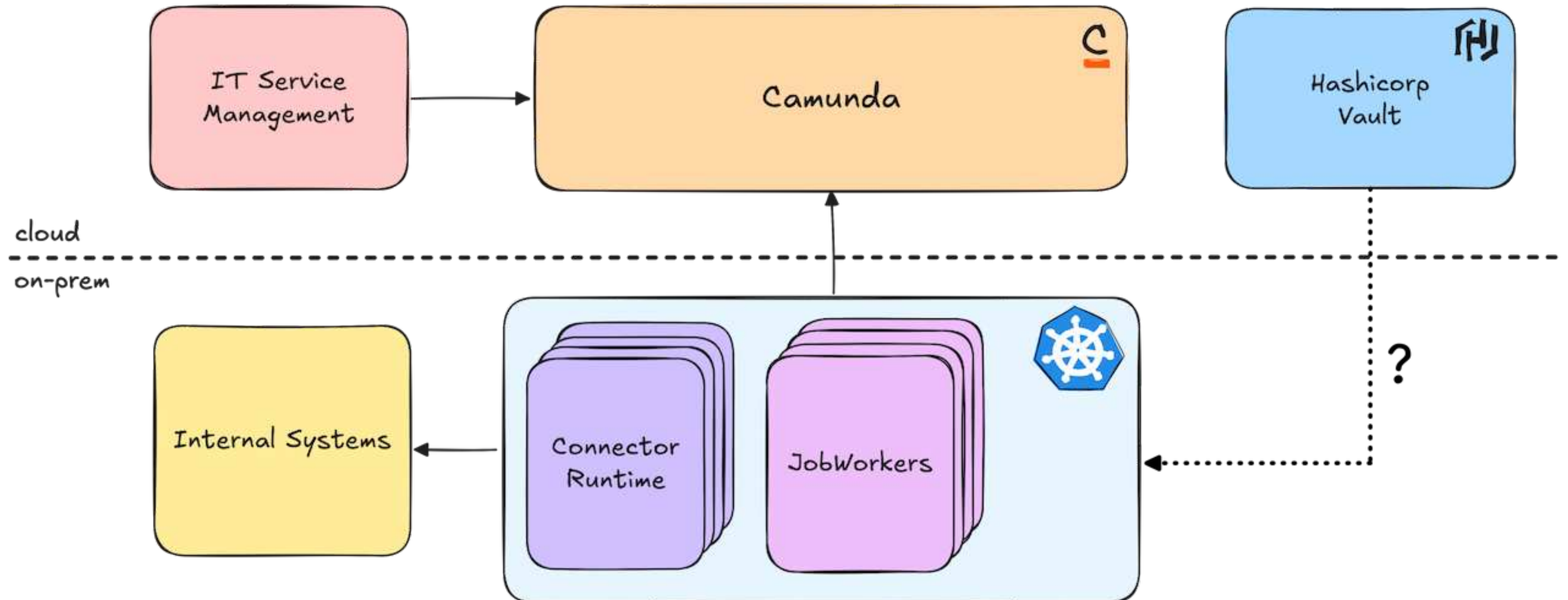
- ✗ number of runtimes
- ✗ getting secrets safely into runtime
 - ✗ support secret rotation

DESIGN ASPECTS

- support of 4-8 different units
- different responsibilities
- shared connectors
- some shared processes
- no shared secrets



SOME PROOF OF CONCEPTS LATER...



HASHICORP VAULT

- central secret management
- sophisticated access control
- audit and access logs
- static & dynamic secrets
- high availability



PHASES OF RETRIEVAL

AT STARTUP

✓ loaded once

✗ restart for rotation

✗ secrets in-memory

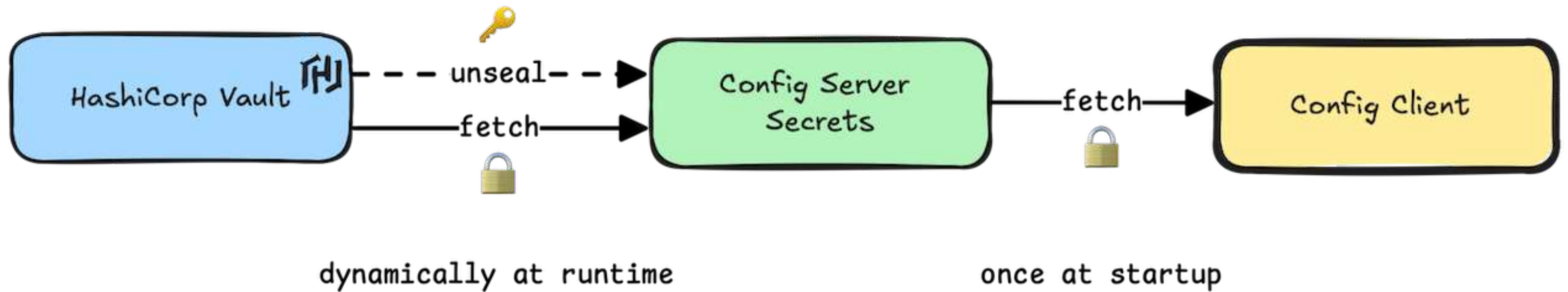
DURING RUNTIME

✓ always current

✗ added latency

✗ availability

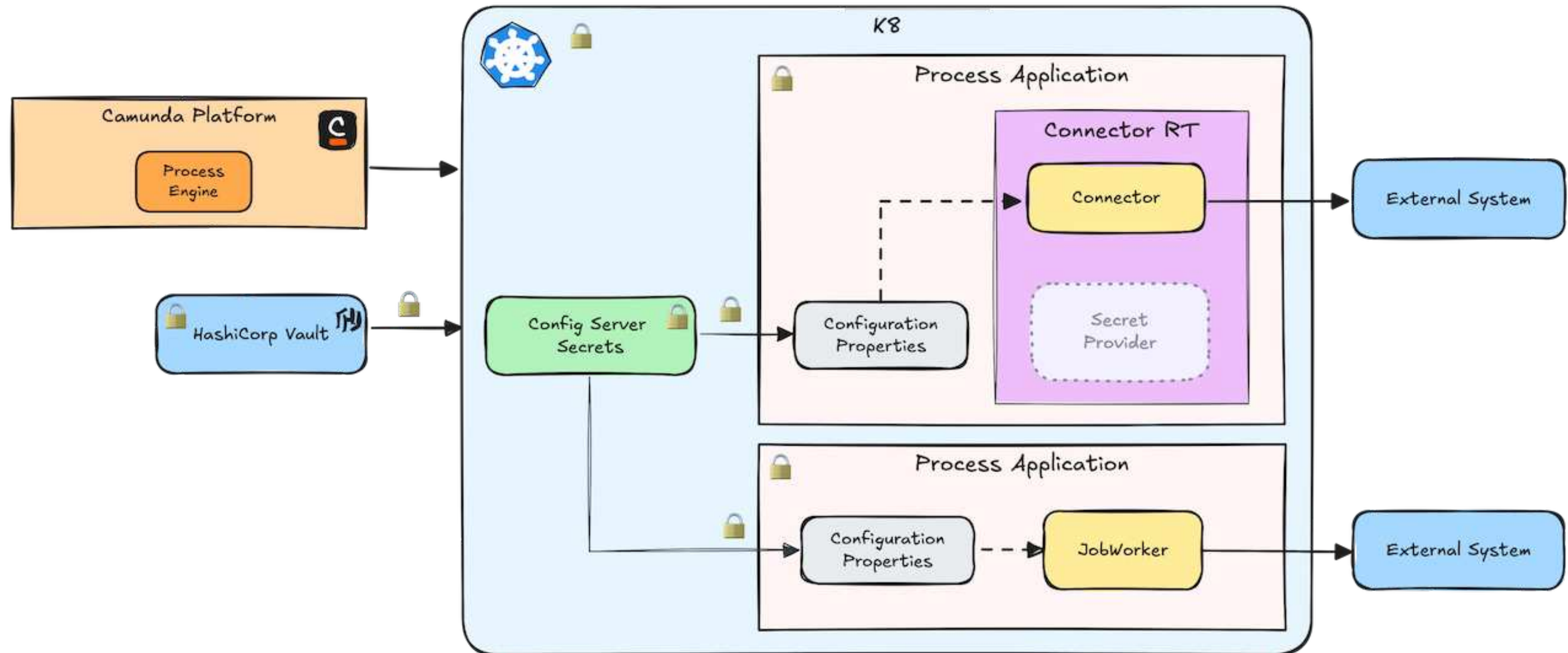
MIXING PHASES



SECURITY ASPECTS

- vault sealed by default
- access via https/tls only
- spring security for endpoints
 - restricted logging
 - k8s cluster security

TECHNICAL DESIGN



UNADDRESSED ISSUES

- ✓ number of runtimes
- ✓ getting secrets safely into runtime
 - ✓ remove secrets from templates
 - ✗ support secret rotation
 - ✗ connector template duplication

SECRET ROTATION WITH BPMN

The screenshot displays the Zeebe BPMN editor interface. On the left, a BPMN diagram shows a start event (thin circle) connected to a task (rounded rectangle) labeled 'do REST call'. The task has a blue border and a blue circle with a plus sign at its bottom-right corner, indicating it is a retryable task. Above the task is a blue circle with an 'R' inside. The task is connected to an end event (thick circle). A toolbar with various icons is positioned above the task. Below the task, a 'Send feedback' button is visible. On the right, the 'Properties' panel is open, showing the configuration for the selected task. The 'Error handling' section is expanded, showing an 'Error expression' field with an equals sign and a link to 'documentation'. The 'Retries' section is also expanded, showing 'Retries' set to 3 and 'Retry backoff' set to PT0S. The 'Execution listeners' section is collapsed. At the bottom right, a status bar indicates 'Check problems against: Zeebe 8.6'.

expression to map the response into process variables

Error handling

Error expression *fx*

=

Expression to handle errors. Details in the [documentation](#).

Retries

Retries *fx*

3

Number of retries

Retry backoff *fx*

PT0S

ISO-8601 duration to wait between retries

Execution listeners

Check problems against: Zeebe 8.6

SECRET ROTATION IN CODE

```
try {
    authenticate();
    // do worker logic here...
} catch (AuthException e) {

    // fail job with custom retry back-off
    client.newFailCommand(job)
        .retries(job.getRetries() - 1)
        .retryBackoff(Duration.ofSeconds(60))
        .errorMessage("Authentication failed!")
        .send()
        .join();
}
```

JobWorkers

```
try {
    authenticate();
    // do connector logic here...
} catch (AuthException e) {

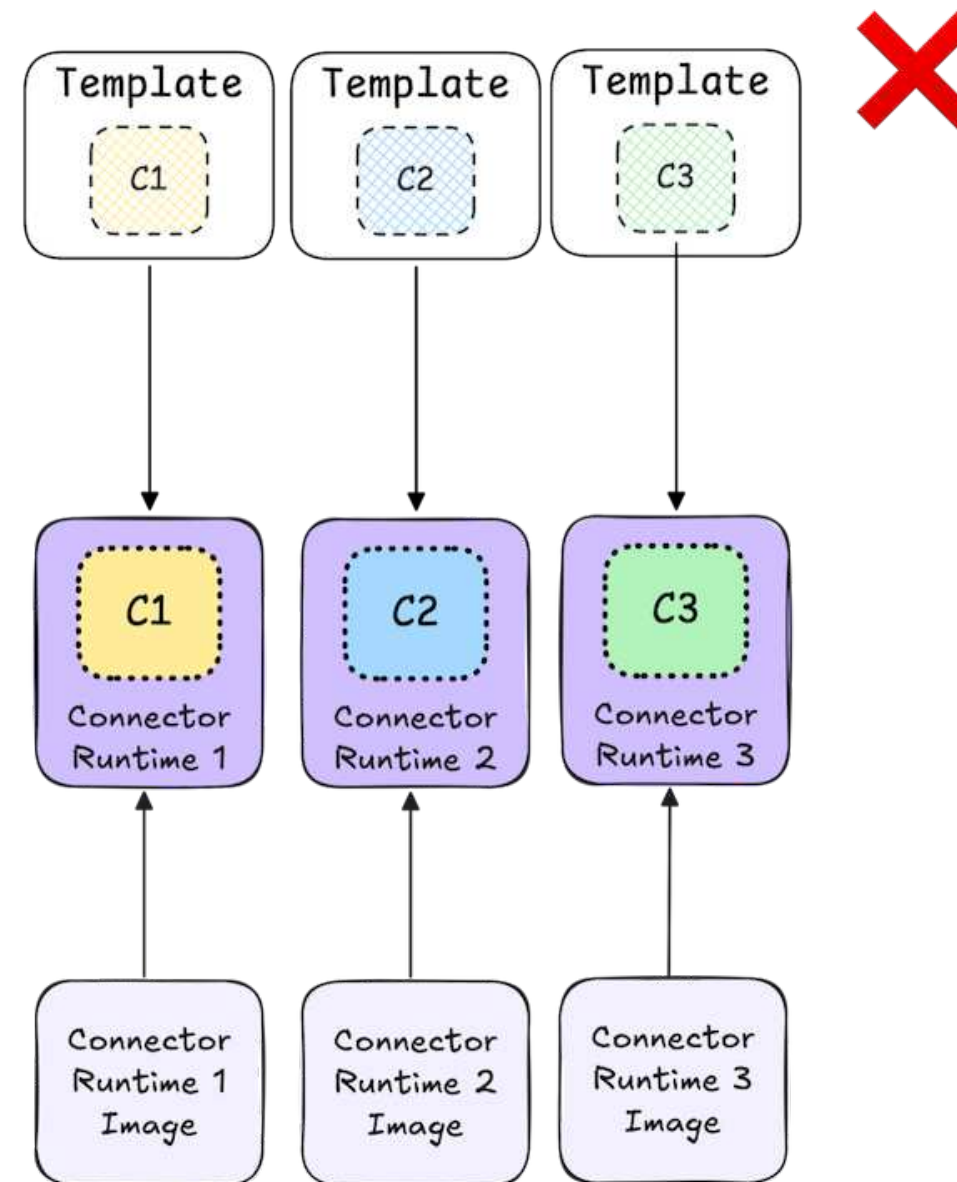
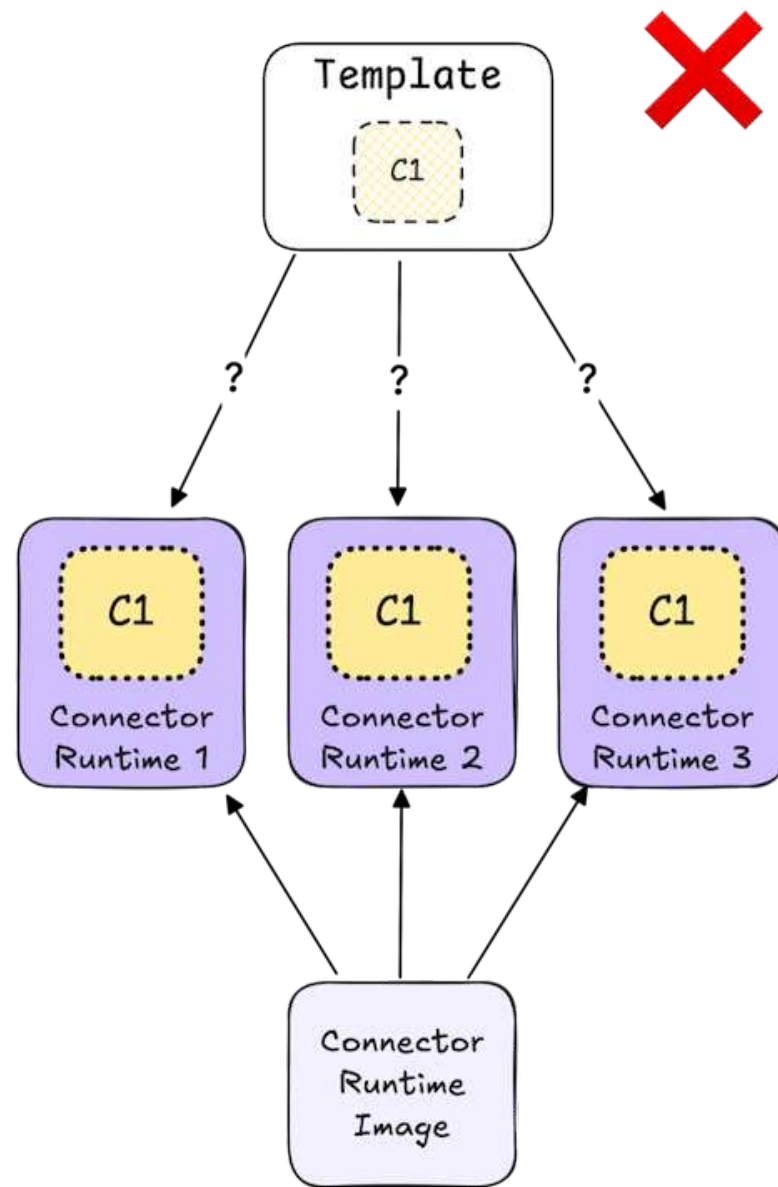
    // fail connector with custom retry back-off
    throw new ConnectorRetryExceptionBuilder()
        .retries(retries - 1)
        .backoffDuration(Duration.ofSeconds(60))
        .errorCode("AUTH_EXCEPTION")
        .message("Authentication failed!")
        .build();
}
```

Connectors

UNADDRESSED ISSUES

- ✓ number of runtimes
- ✓ getting secrets safely into runtime
 - ✓ remove secrets from templates
 - ✓ support secret rotation
 - ✗ connector template duplication

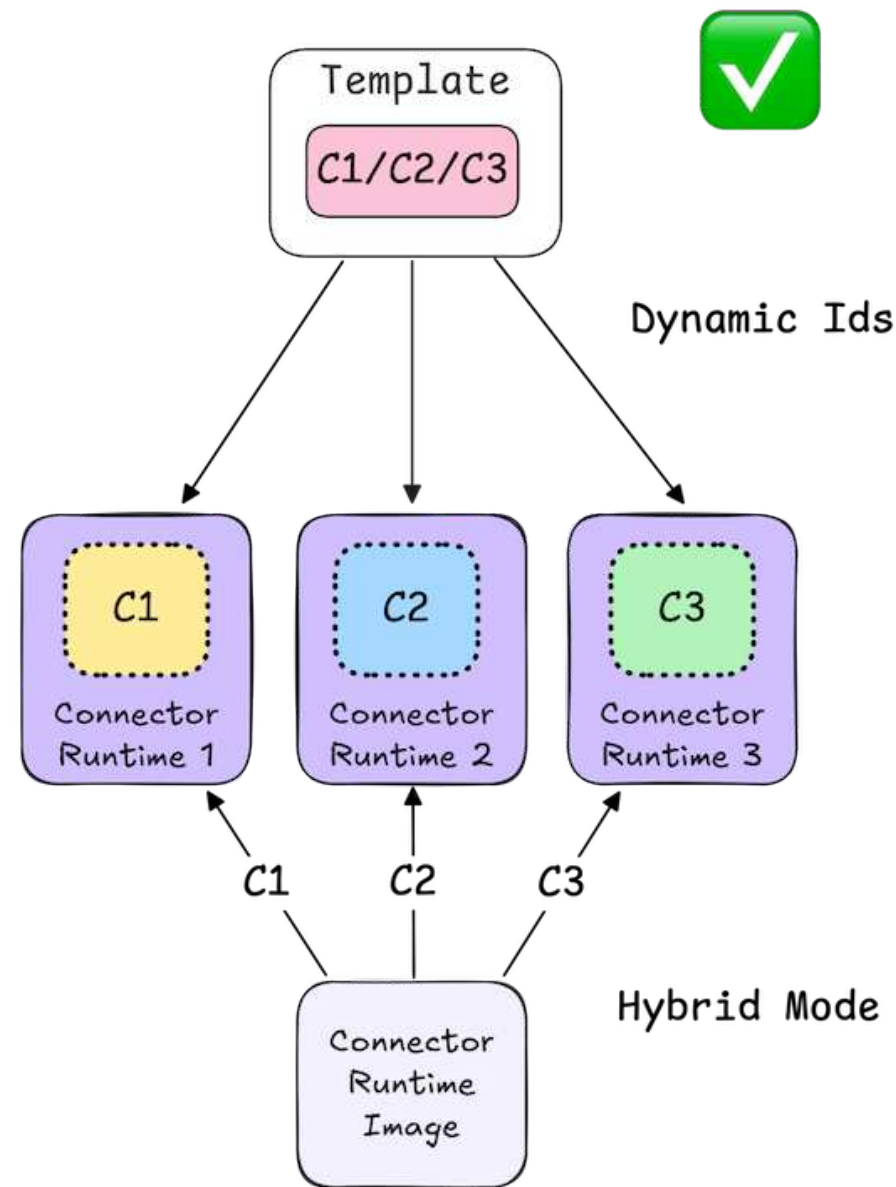
TEMPLATE DUPLICATION



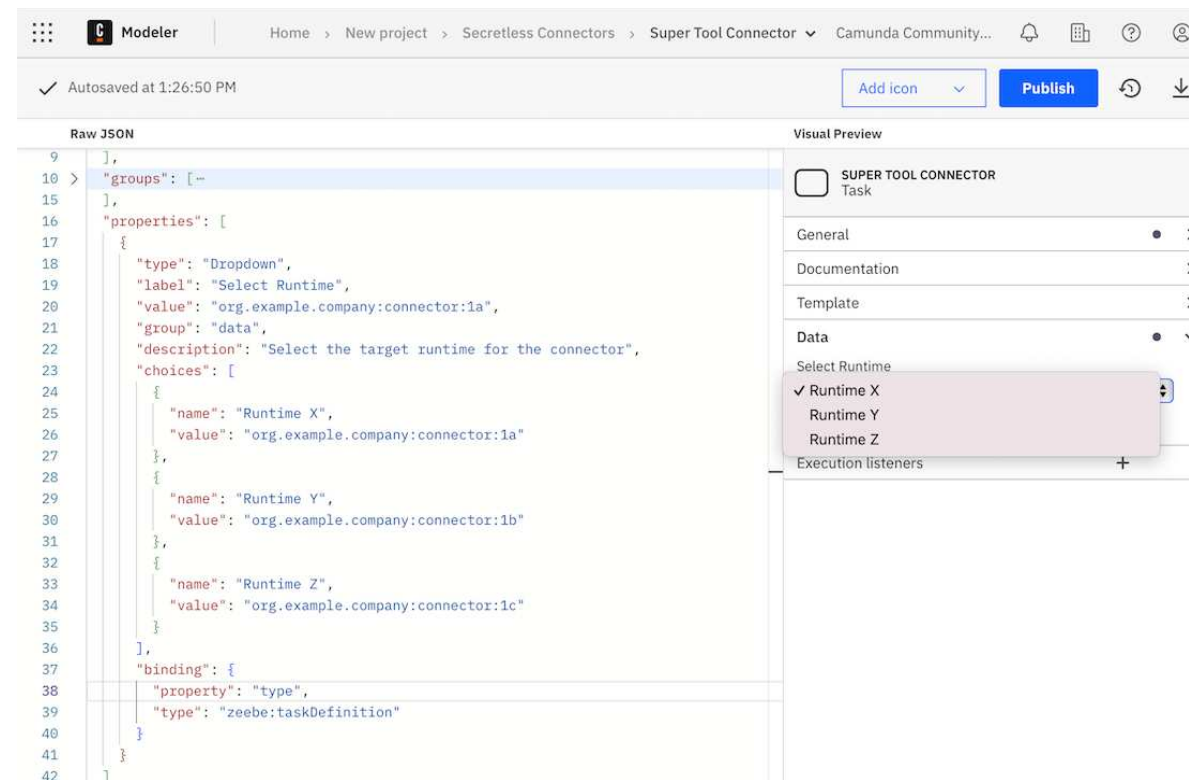
CONNECTOR **HYBRID** MODE

- one template for multiple runtimes
- runtime with overwritten connector ids
 - target runtime via dropdown
- runtime knows allowed process ids

CONNECTOR HYBRID MODE

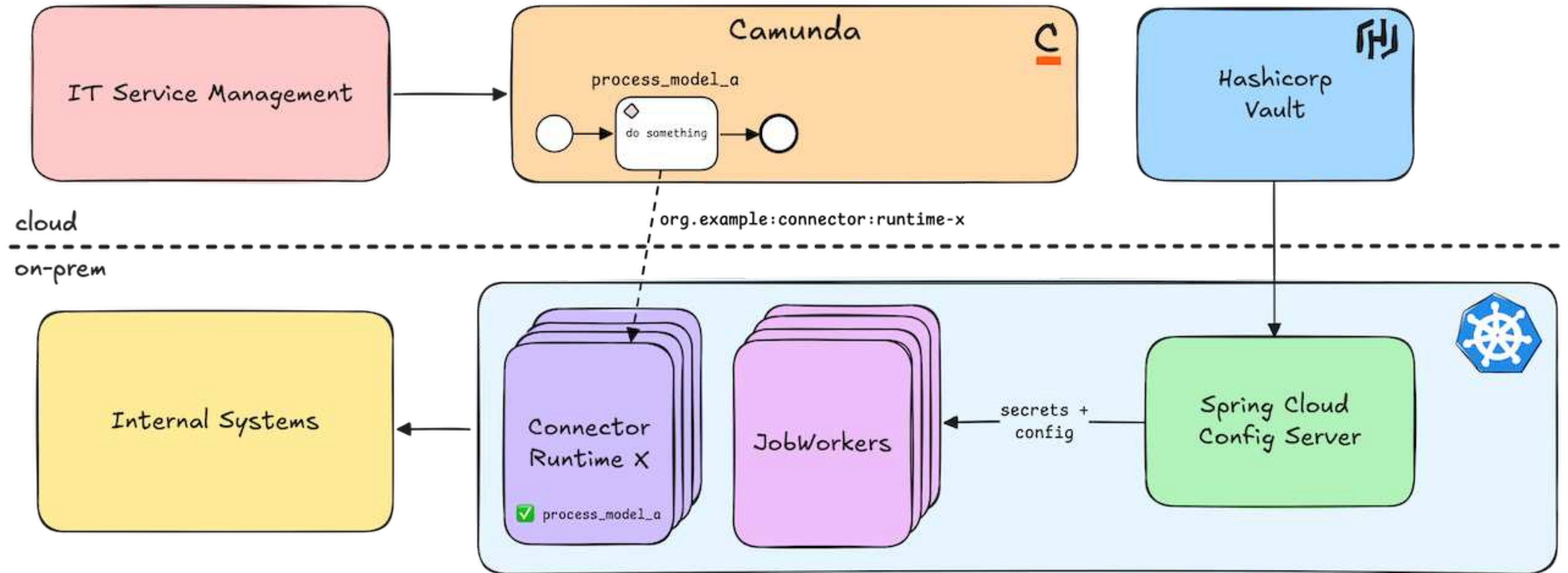


CONNECTOR HYBRID MODE



```
docker run --rm --name=MyConnectorRuntime \
  -e CONNECTOR_MY_CONNECTOR_TYPE='org.example.company:connector:1a'
```

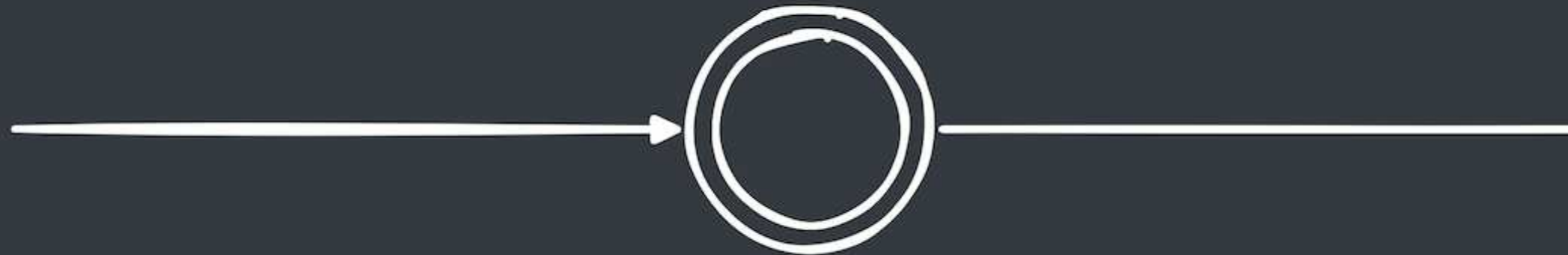
TARGET ARCHITECTURE



WHAT WE'VE **LEARNED**

- actively manage your secrets
- secrets vulnerable outside vault
- hybrid connectors for multiple runtimes
 - lower risk with secret-less templates
 - Camunda supports secret rotation

FOLLOW ME ON **SUBSTACK** FOR MORE



The Intermediate
Catch Event

THANK YOU!

ENJOY CAMUNDA CON

