Leveraging Argo WorkflowTemplates Within a Data Platform

KubeCon | CloudNativeCon
Europe 2023

Who We Are

ArgoCon
EUROPE
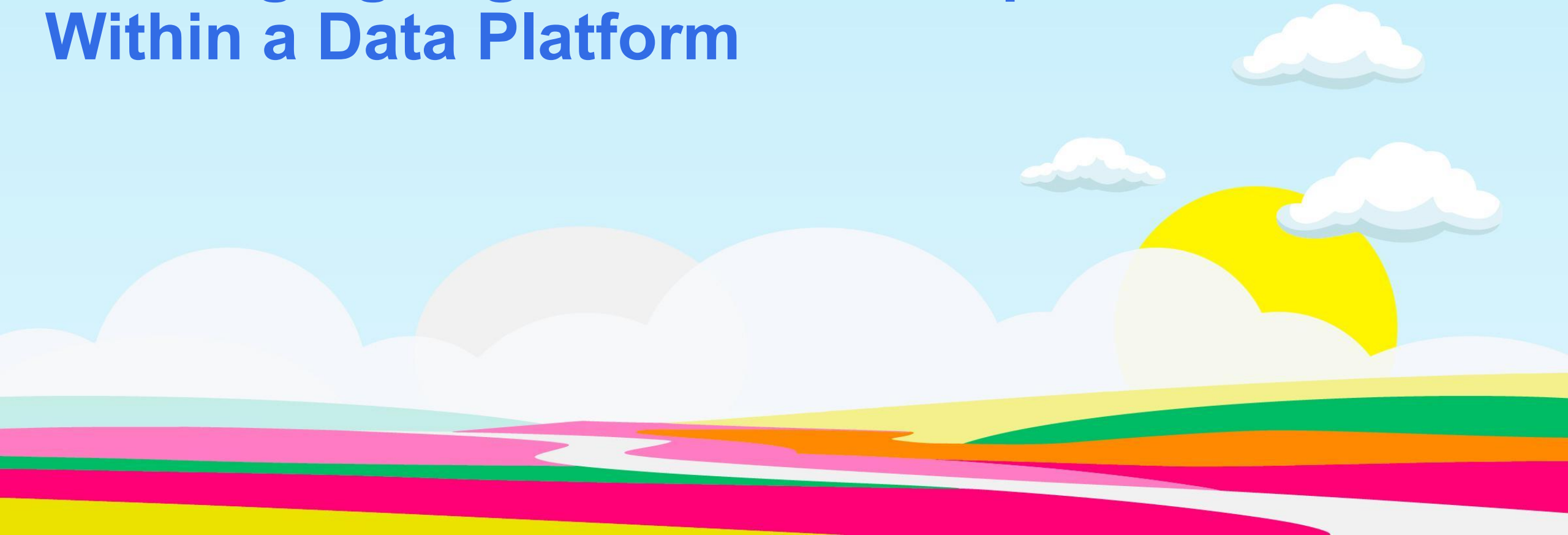
April 18, 2023 | Amsterdam

J.P. Zivalich

CTO & Founder,
Pipekit

Yao Lin

Software Engineer,
Workflow Orchestration
Bloomberg

Prerequisites/Terminology

Prerequisites:
- Basic familiarity with Argo Workflows

Terminology
- Workflow
- WorkflowTemplate
- ClusterWorkflowTemplate

Challenges to Solve

Problem:
- Sharing and managing WorkflowTemplates across clusters and environment is hard

Goal:
- Set up a platform to allow teams to access these resources

Questions:
- How resources are created?
- How resources are persisted?
- How do I track my versions?
- When outdated resources are deleted?
- …

Goals

1. DRY (Don't Repeat Yourself) code
    a. Users not repeat themselves by pulling and using pre-built WorkflowTemplates

2. Logical resource grouping
    a. Group templates within a team/namespace
    b. Add versioning to those templates and enforce naming conventions

3. Clean clusters
    a. Avoid stale and unused resources

4. Security and stability
    a. Promote templates across environments, allowing for approvals

Setup – Bloomberg

1. UI for managing WorkflowTemplates above workflow clusters, for:
   a. Deployment strategies
   b. Consistency

2. Templates managed by an external admin-service cluster, with:
   a. Persist original as source of truth
   b. Syncer agent on each workflow cluster

3. Kyverno policies:
   a. Validate name with semver suffix
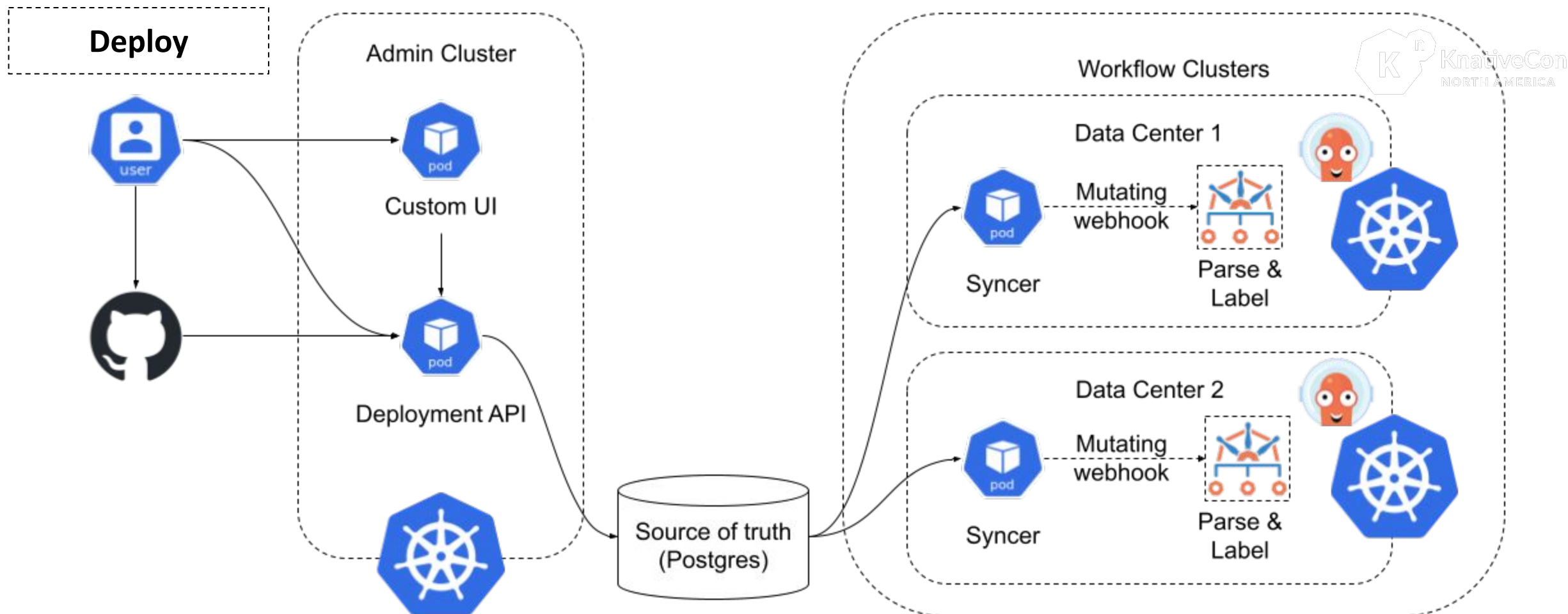   b. Parse version from name and attach labels
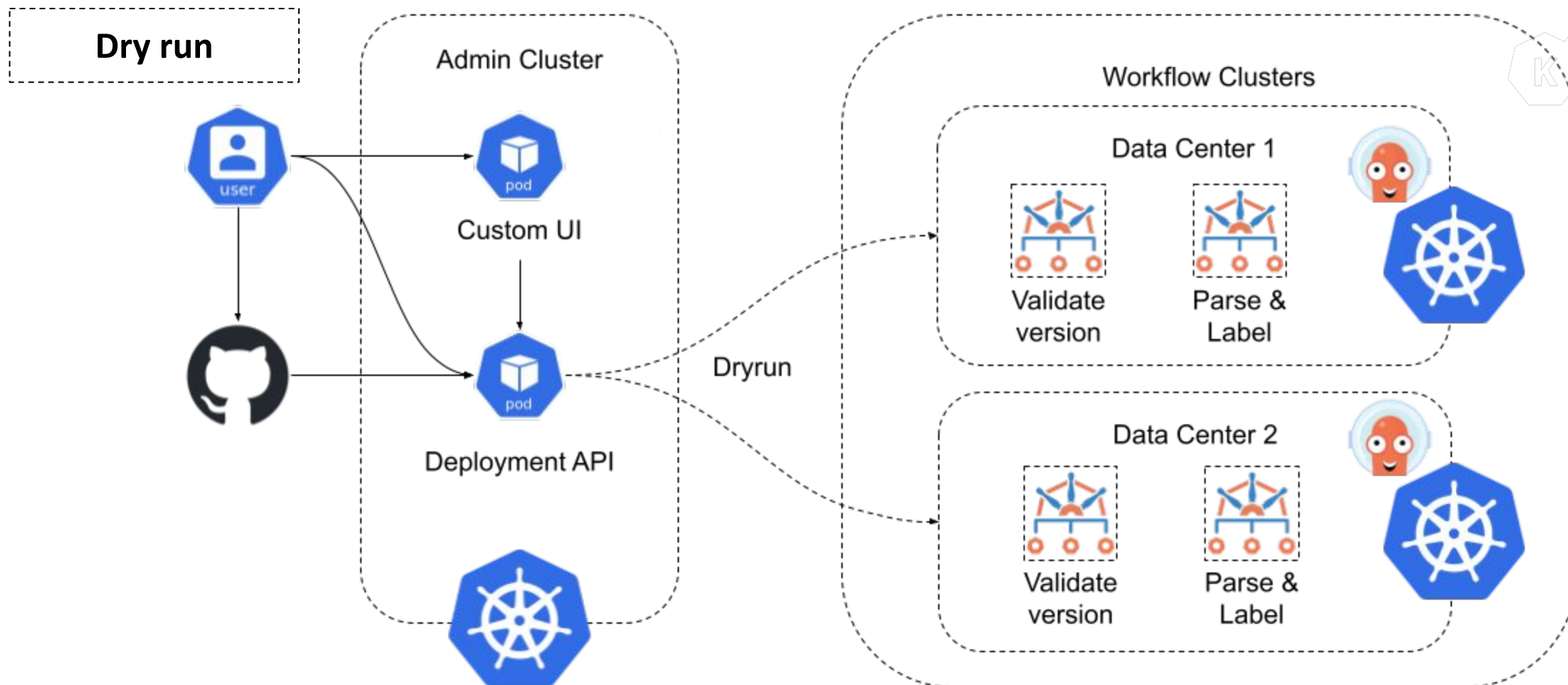
**Bloomberg**
Engineering

Setup – Bloomberg

Setup – Bloomberg

Setup – Pipekit

Git-based approach:
- UI for selecting folders within a GitHub/GitLab repository
  - Clicking "Submit" adds each Template to the Pipekit Control Plane
  - GitHub/GitLab app allows for control over repository permissions

- Cluster admins install pipekit-agent in clusters they manage
  - Pipekit manages Argo Workflows resources on each cluster
  - Templates available on all clusters connected to the Pipekit Control Plane
  - Pipekit-agent pulls Templates from GitHub/GitLab and adds to the cluster at Workflow creation time

- Graceful Template conflict handling
  - Workflows lock in Templates at Workflow creation time
    - Each Workflow stores and flattens Templates as it submits to the Workflow Controller

pipekit.io

pipekit

Setup – Pipekit
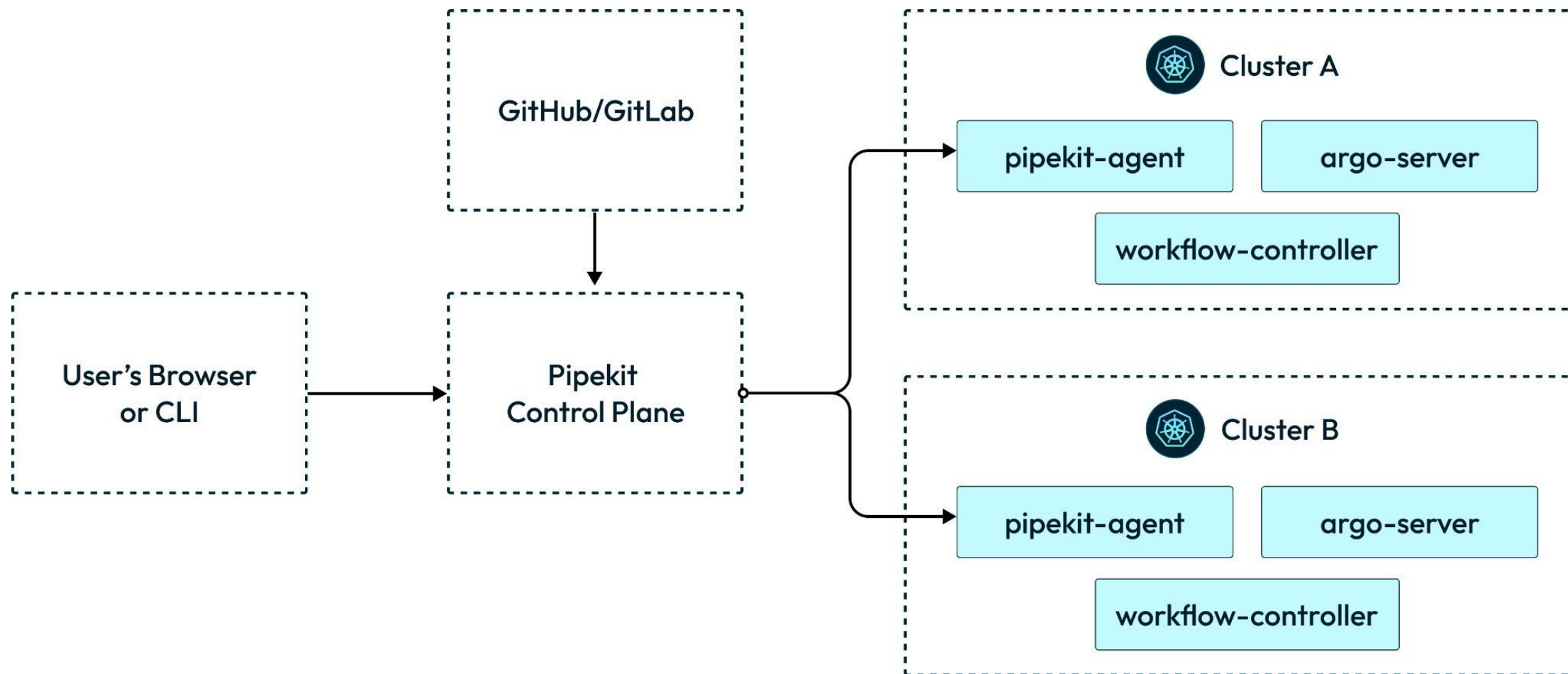
- View version history
  - Main page: group by `<name>-<major>`
  - Details page: all versions history with deployment status

- Approval-required promote
  - Replicate from Development tier to Production
  - Encourage usage of WorkflowTemplates with easier approval
    - "Templated workflows": use purely `templateRef`
   - As opposed to workflows with: custom script, custom container

Usage – Bloomberg

KnativeCon
NORTH AMERICA

Grouped view

Namespace: team-x

Templates

Name	Handle	Major	Deployed clusters	Version counts
awesome-v1	awesome	1	data-center-1 data-center-2	3
sample-v2	sample	2	data-center-1 data-center-2	5

Template details: awesome-v1

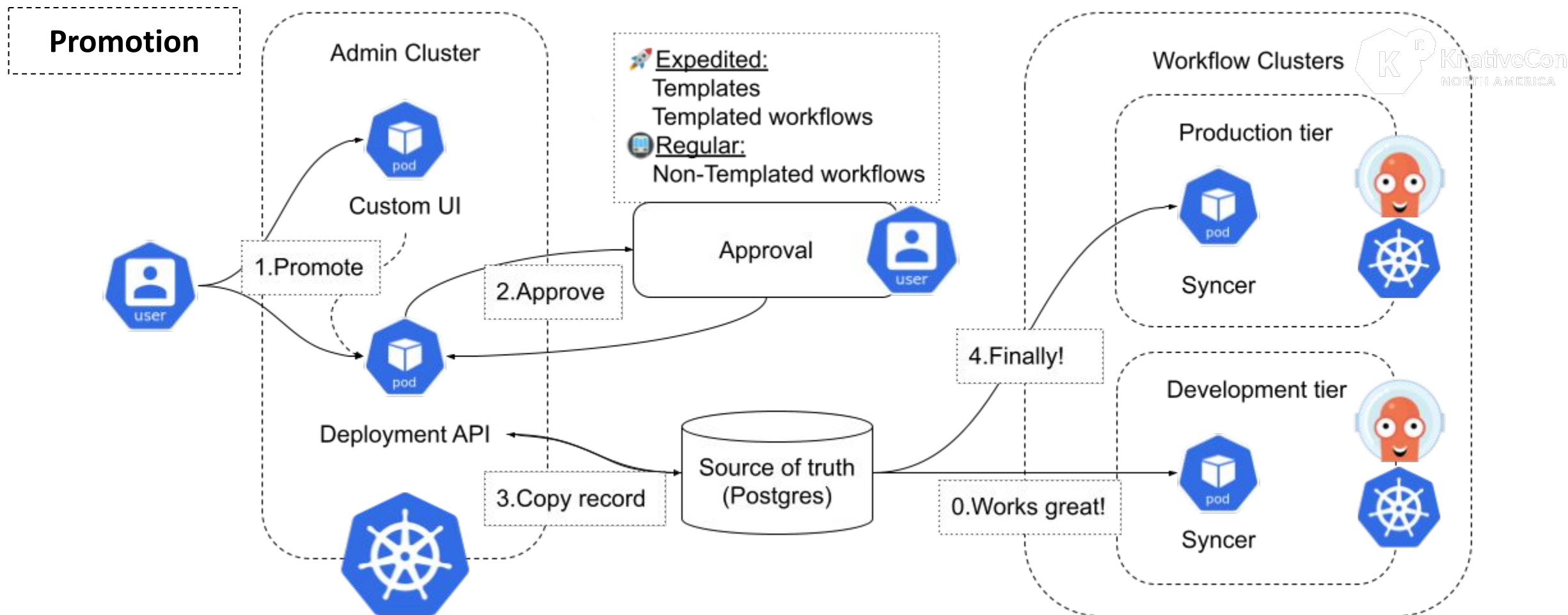| Name	Deployed clusters	Deployment strategy	Creation time
awesome-v1.0.3	data-center-1 data-center-2	Everywhere	December 1, 2022 10:00 AM
awesome-v1.0.2	data-center-1 data-center-2	Everywhere	November 1, 2022 11:00 AM
awesome-v1.0.1	data-center-1	Random Single Cluster	October 1, 2022 12:00 AM
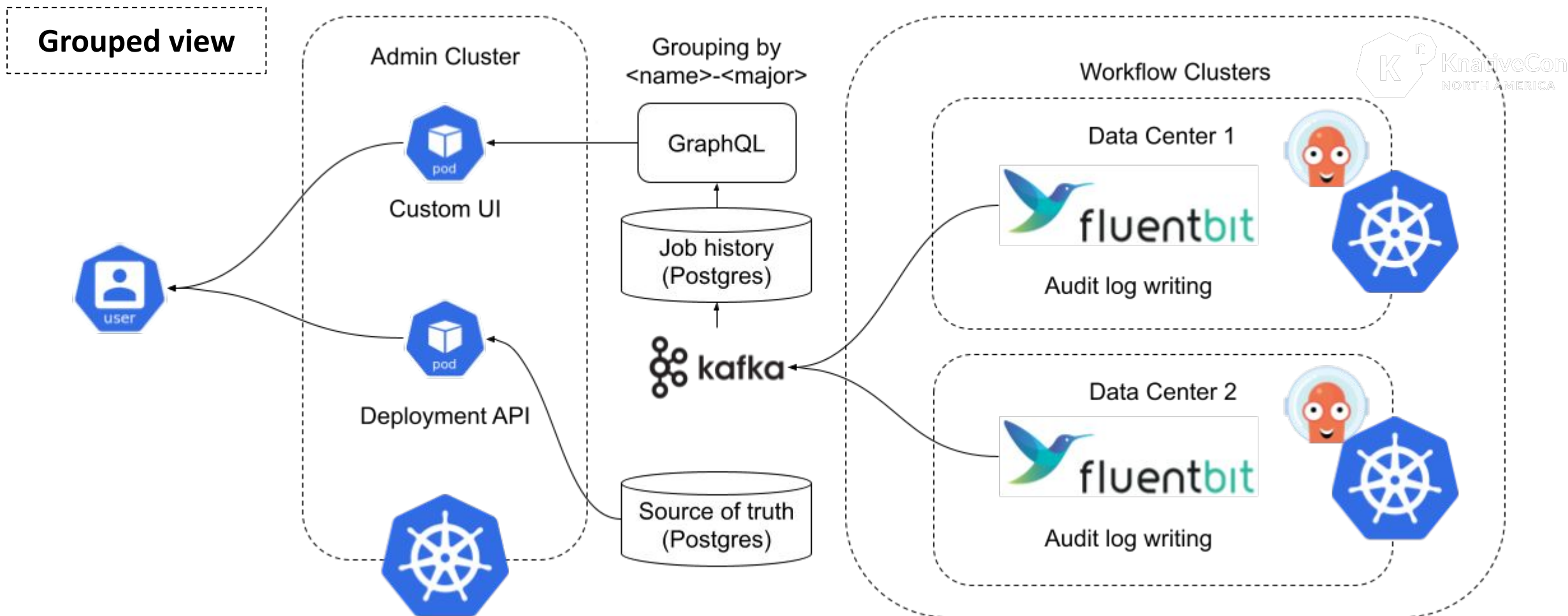
**TechAtBloomberg.com**

Bloomberg

Engineering

Usage – Bloomberg

14

Usage – Pipekit

- Design goals
  - Avoid changing open-source Workflow specification
  - Allow users to select the git tag, commit, branch, etc. for the corresponding WorkflowTemplate definition
- Solution: use metadata!
  - Specify a custom pipekit label when invoking a template
  - Support git tags, commits, branches, etc.

```yaml
apiVersion: argoproj.io/v1alpha1
kind: Workflow
metadata:
  generateName: hello-world-wf-global-arg-
spec:
  entrypoint: whalesay
  arguments:
      parameters:
      - name: global-parameter
      value: hello
  templates:
    - name: whalesay
    metadata:
    labels:
        pipekit.io.templates.gitVersion: v0.14.64
    steps:
    - - name: whalesay-template
        templateRef:
        name: workflow-template-whalesay-template
        template: whalesay-template
        arguments:
        parameters:
        - name: message
            value: "hello world"
```

Cleanup – Bloomberg
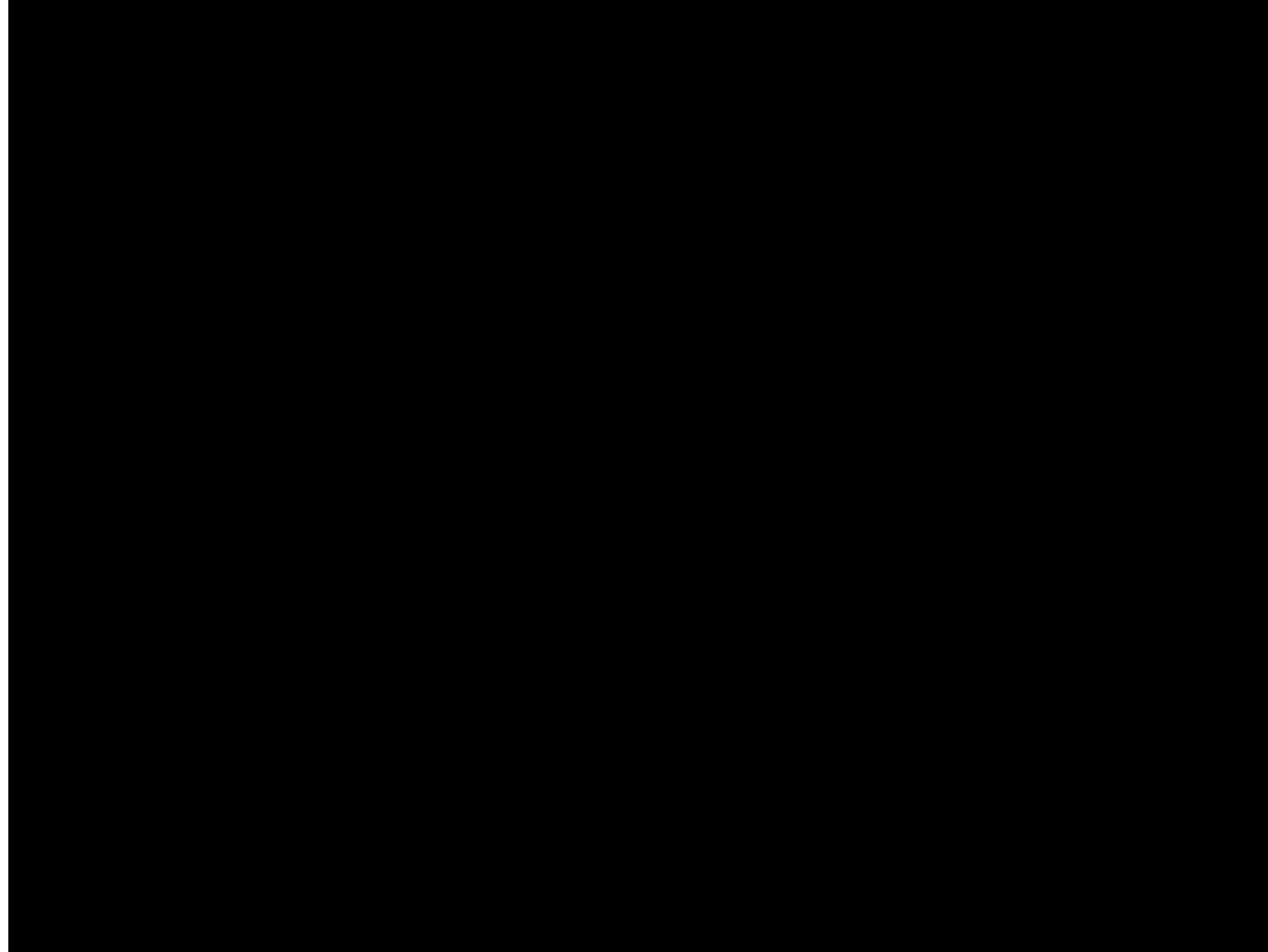
- Async steps
  - Cron workflow
    - Find out stale templates given
      - Templates
      - Active workflows
    - Label templates with expected delete time (now + 7d)
  - Deleter
    - Monitor delete label
    - Execute actual deletion
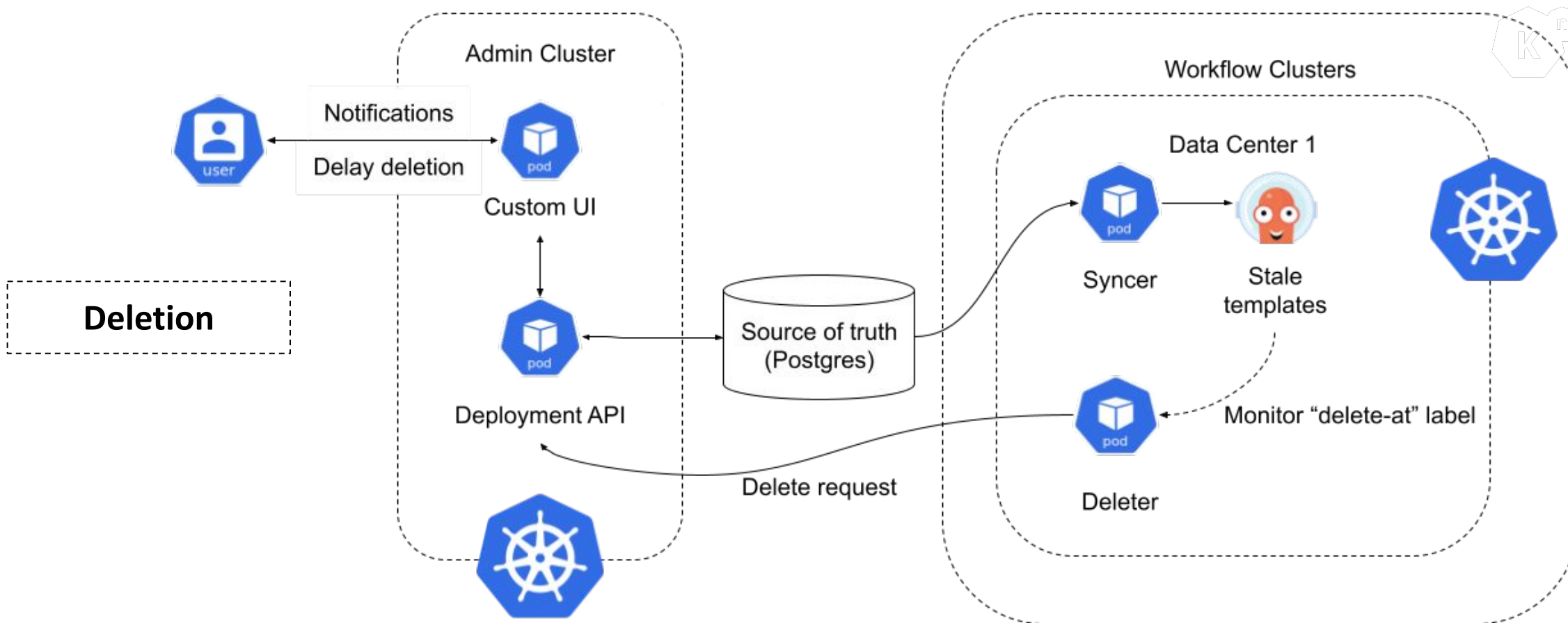 - Allow user to postpone / cancel deletion

Cleanup – Bloomberg

TechAtBloomberg.com

Bloomberg

18
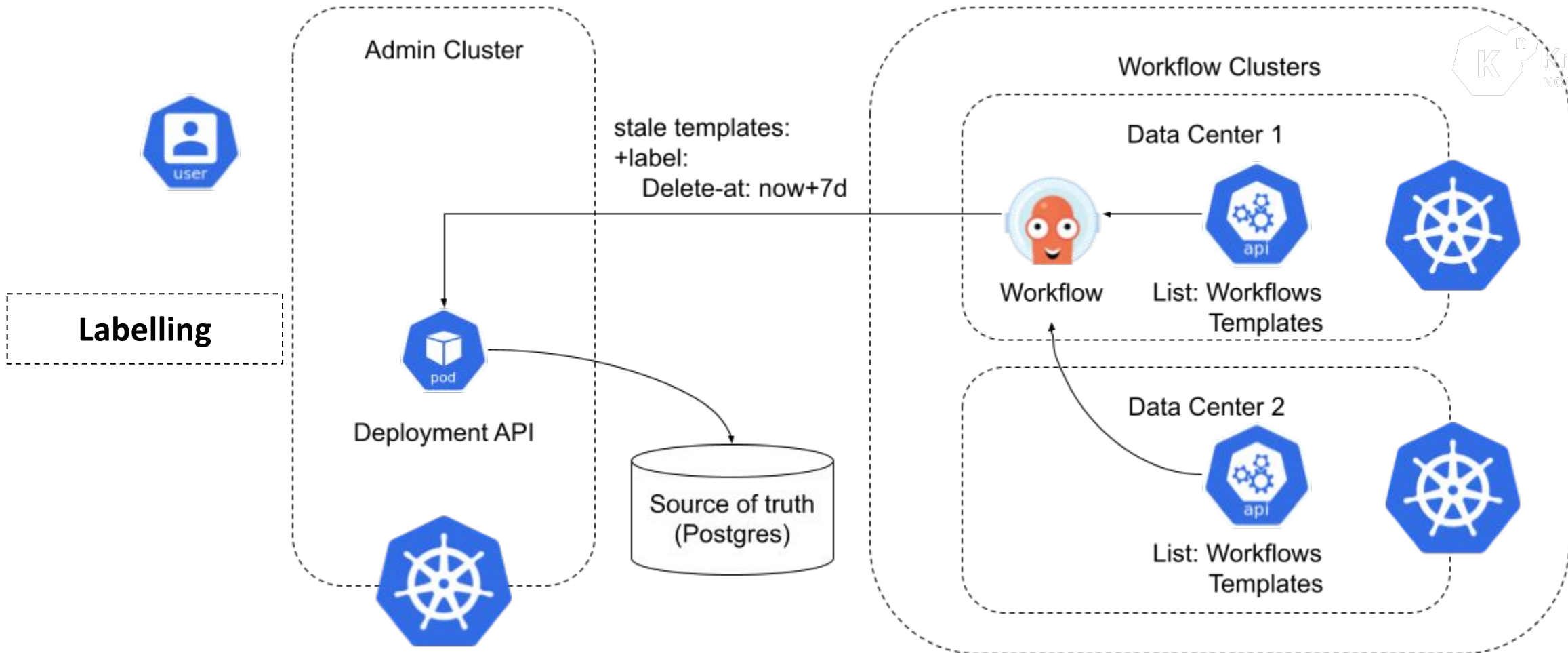
Cleanup – Bloomberg

TechAtBloomberg.com

Cleanup – Pipekit

- pipekit-agent deployment handles garbage collection
  - `last_used_timestamp` label set each time a template is run
  - Go routine checks the templates on the cluster hourly
  - Users specify the TTL with an environment variable
  - Default TTL is 24 hours

```
apiVersion: argoproj.io/v1alpha1
kind: WorkflowTemplate
metadata:
  creationTimestamp: "2023-04-13T08:16:25Z"
  generation: 2
  labels:
    workflows.pipekit.io/last_used_timestamp: "1681374761"
    workflows.pipekit.io/runner:
pipekit-agent-53fe39bc-b163-4d22-8f9e-d9da3dfa0d08
  name: workflow-template-whalesay-template
  namespace: default
  resourceVersion: "211410"
  uid: 8d6637e4-9eed-480c-89c6-d3360e5ecdf2
spec:
  entrypoint: whalesay-template
  podMetadata:
    labels:
    workflows.pipekit.io/pipe_uuid: a60e362d-86ec-4218-badd-fbeec21178fb
    workflows.pipekit.io/run_uuid: f3cedc4b-2594-4658-8df7-36e81ed481d4
    workflows.pipekit.io/runner:
pipekit-agent-53fe39bc-b163-4d22-8f9e-d9da3dfa0d08
```

Cleanup – Pipekit

Comparison between platforms

	Pipekit	**Bloomberg**
User profiles	Generic to support wide variety of use cases	Opinionated towards pre-built WorkflowTemplates Specific SLA
Git handling	Backend provided by GitHub/GitLab	CI/CD agents Info parsed by kyverno policies
Cleanup	Just-in-time delivery of WorkflowTemplates Default cleanup of 24 hours, configurable by users	Always ready on cluster Defaults to a 7 day grace period Canceleable but non-recoverable

Looking forward + Q&A

- Bloomberg
  - Auto version detection & refresh
    - Pin to the latest version
    - Set version range

- Pipekit
  - WorkflowTemplate pull policy
    - Looking to mirror Kubernetes' approach for pulling Docker images