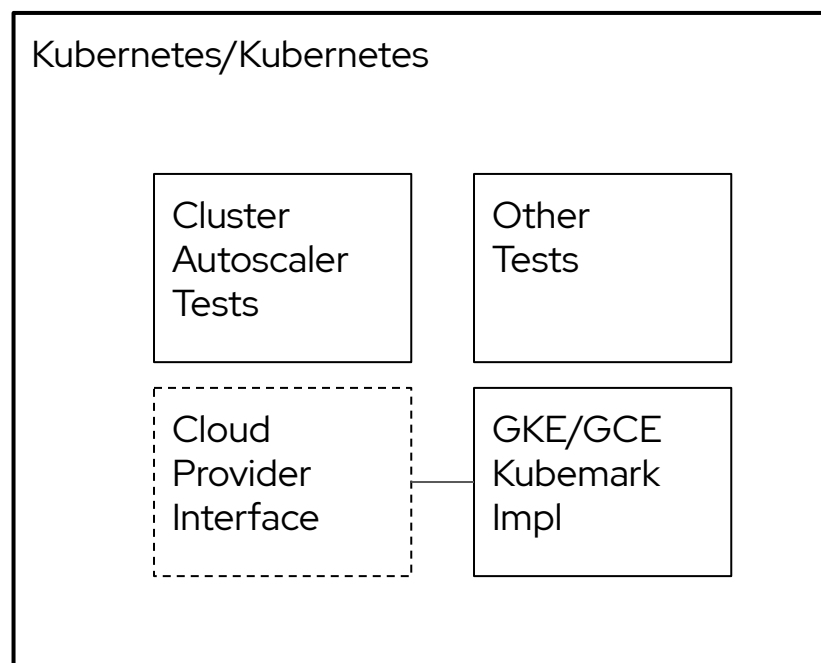# Improving Kubernetes Autoscaler E2E Tests
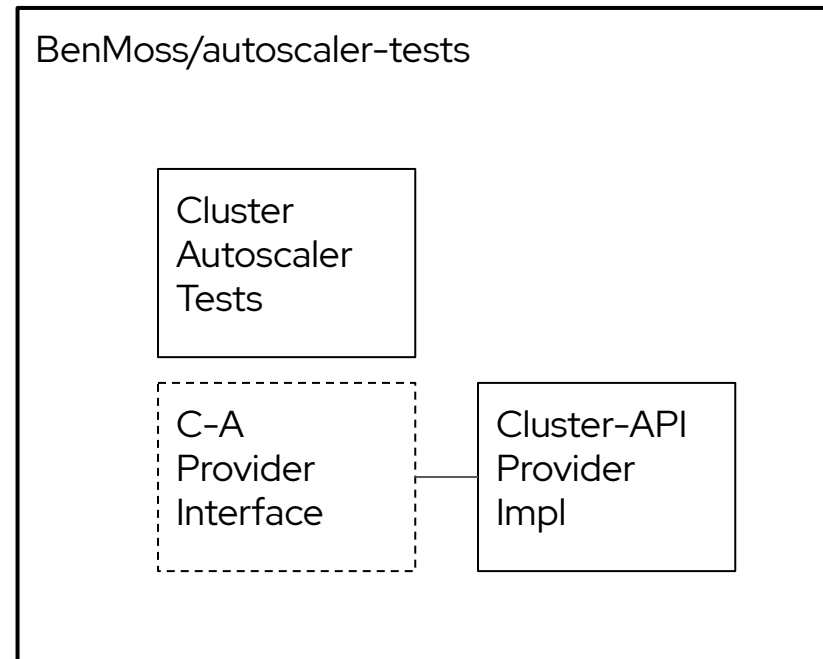
Michael McCune

@elmiko

Red Hat

Goals

- Add a Cluster-API provider to Cluster Autoscaler e2e tests
- Enable more automated tests for Cluster Autoscaler
- Open pathway for more contributors to create Cluster Autoscaler tests for their providers
- Stretch Goal: Demonstrate a possible future for other project communities to increase their testing
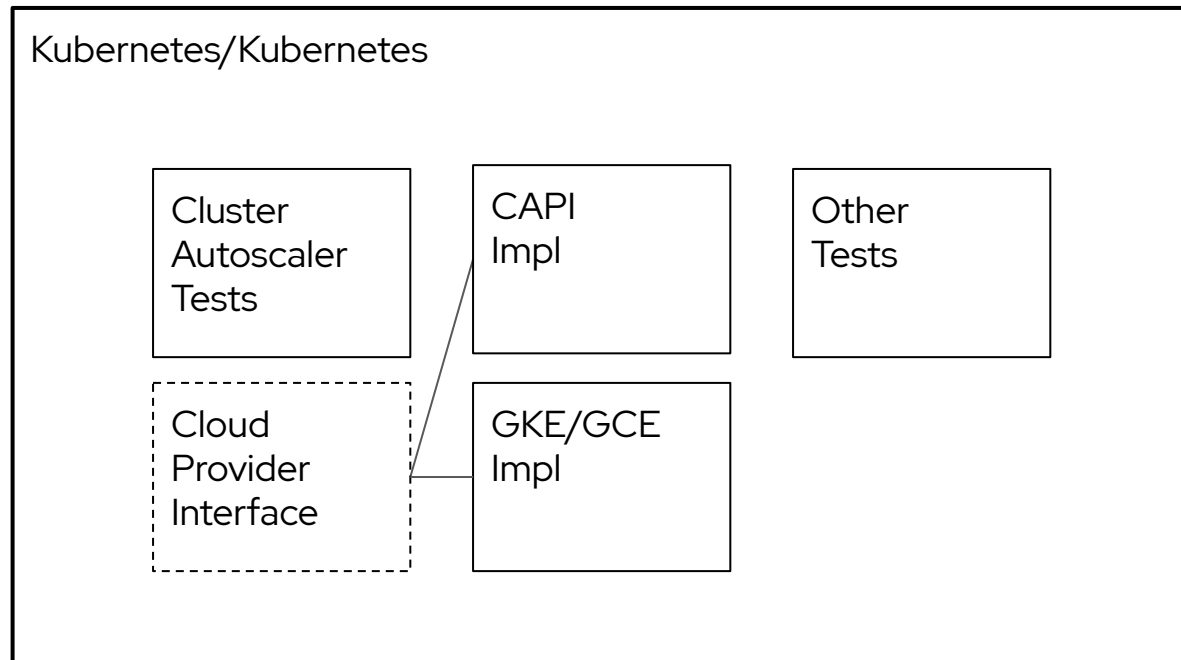
Red Hat

Current state of tests

Kubernetes/Kubernetes

Cluster Autoscaler Tests

Other Tests

Cloud Provider Interface

GKE/GCE Kubemark Impl

Red Hat

Proof of concept experiment

BenMoss/autoscaler-tests

Cluster Autoscaler Tests

C-A Provider Interface

Cluster-API Provider Impl

Example 1: Tests and Providers in Kubernetes

Kubernetes/Kubernetes

Cluster
Autoscaler
Tests

CAPI
Impl

Other
Tests

Cloud
Provider
Interface

GKE/GCE
Impl

Example 2: Tests in Kubernetes, Providers outside

GKE/GCE
C-A
Provider

CAPI
C-A
Provider

Kubernetes/Kubernetes

Cluster
Autoscaler
Tests

Other
Tests

Cluster
Autoscaler
Provider
Interface/Lib
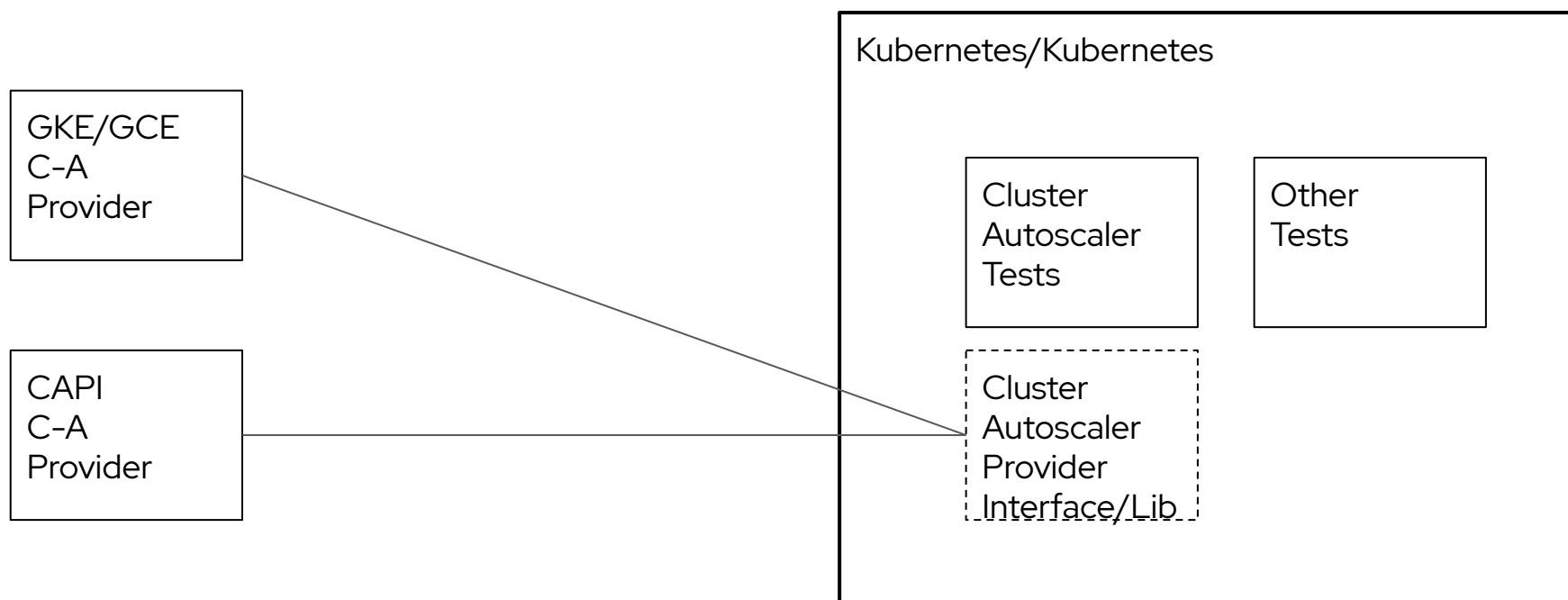
Example 3: Tests and Providers outside of Kubernetes

GKE/GCE
C-A
Provider

CAPI
C-A
Provider

Kubernetes/Autoscaler?

Cluster
Autoscaler
Tests

Cluster
Autoscaler
Provider
Interface/Lib

Kubernetes/Kubernetes

Other
Tests

Cloud
Provider
Interface

Red Hat
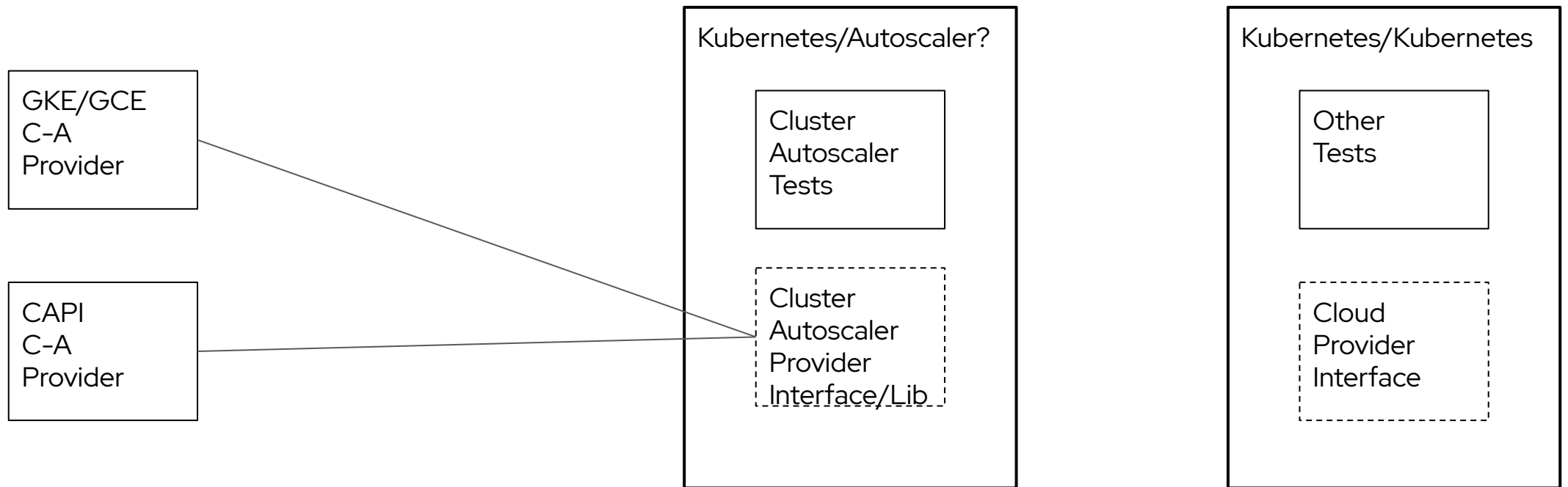
Discussion

- Example 1, seems like this is undesirable (see https://github.com/kubernetes/kubernetes/issues/70194)
- Example 2, Kubernetes/Kubernetes would contain a cluster-autoscaler provder interface/library to ensure cloud providers have the appropriate functionality and setup.
  - How would cloud provider concrete implementation be linked during test deployment?

- Example 3, This seems like easiest approach from a breakage standpoint, but represents a shift in thinking from current testing deployment and workflow.
- Regardless of approach, separating the provider implementation from the tests is desirable for independent development, although this may add maintenance load.
- A *library of building blocks* approach is desirable to ensure consistency for setup and teardown.

Red Hat

References

- https://github.com/kubernetes/kubernetes/issues/70194
  - issue from 2018 where this idea was discussed in more depth
- https://github.com/benmoss/autoscaler-tests
  - Experimental break out of tests
- Improving the state of Kubernetes cluster autoscaler testing
  - Gist with some early thoughts about refactor

Red Hat