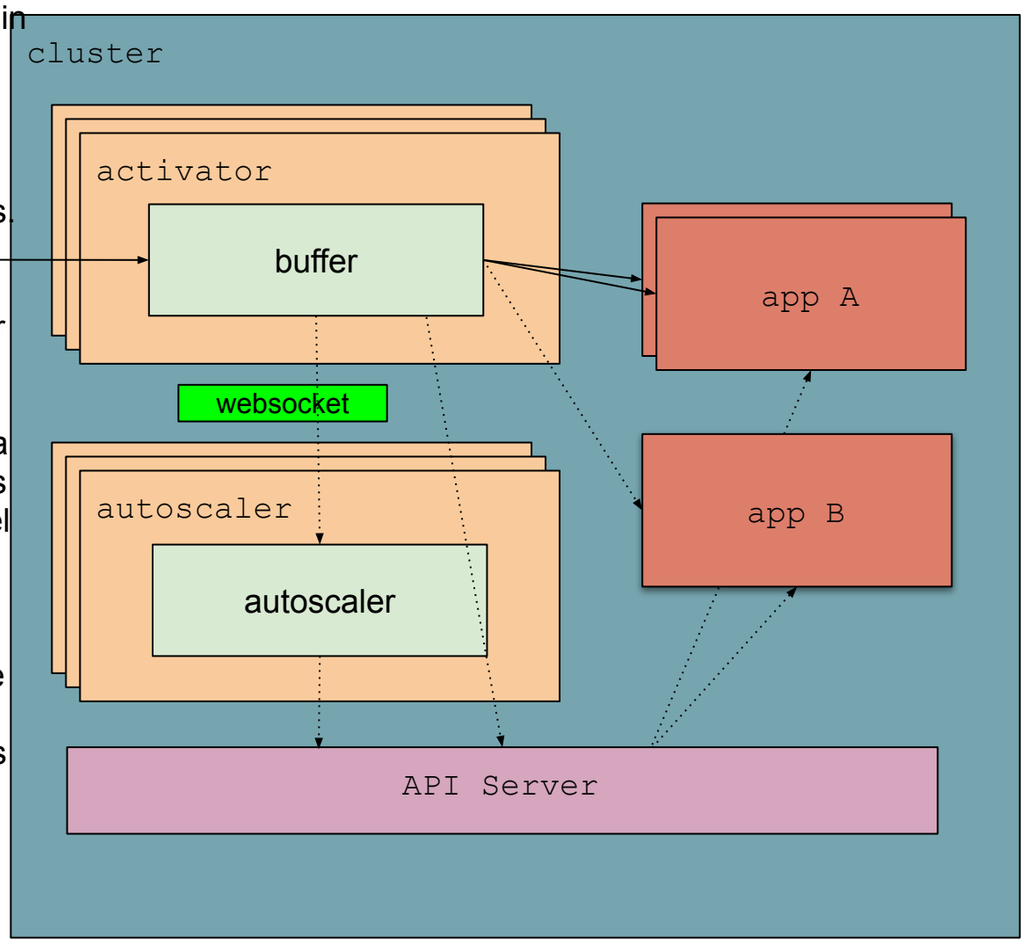
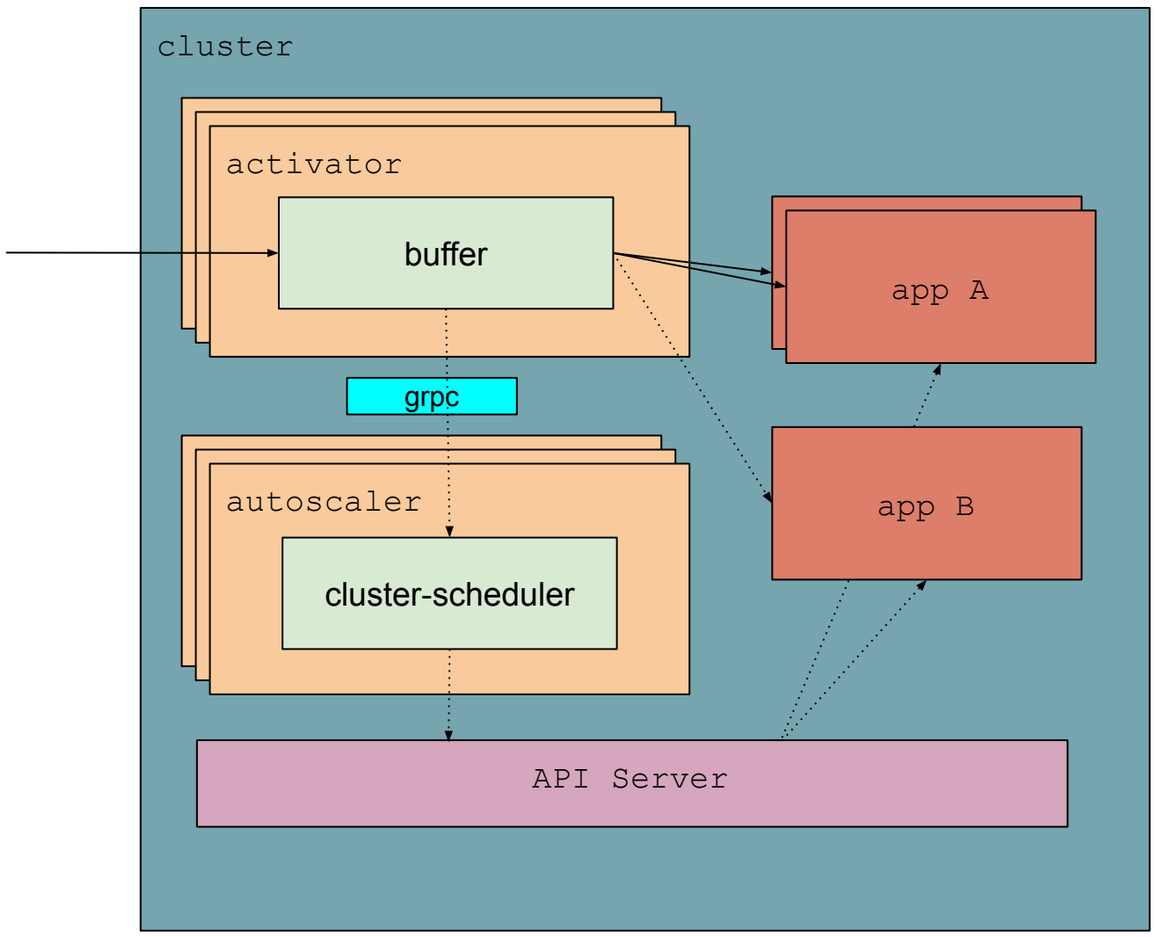
pharetra luctus felis. Proin vel tellus nec in felis volutpat amet molestie cum sociis.

- Donec risus dolor porta venenatis
- Pharetra luctus felis
- Proin vel tellus in felis volutpat
- Molestie nec amet cum sociis

today's Just-in-Time scheduling (cluster)



near-term Just-in-Time scheduling (cluster)



Can we define a GRPC interface that in the near-term can replace our websocket channel to the autoscaler, and in the long-term can become a path to co-scheduling? Perhaps these modes of operating Knative can/should continue to co-exist for some time, or indefinitely?

The activator would become more of a dumb buffer, and would rely on the streaming GRPC connection to tell it where to send requests. In the near term, those requests would be directed across the cluster to pods scheduled through the API Server. In the long-term, those requests would be directed to pods colocated on the same node.

It is likely that we will need to straddle these models, and support cluster-scoped scheduling until the capacity to perform local scheduling decisions is commonplace in K8s services. However, vendors so inclined may choose to vet and support the use of node local scheduling, if it is compatible with their flavor of Kubernetes.

long-term Just-in-Time scheduling (node)