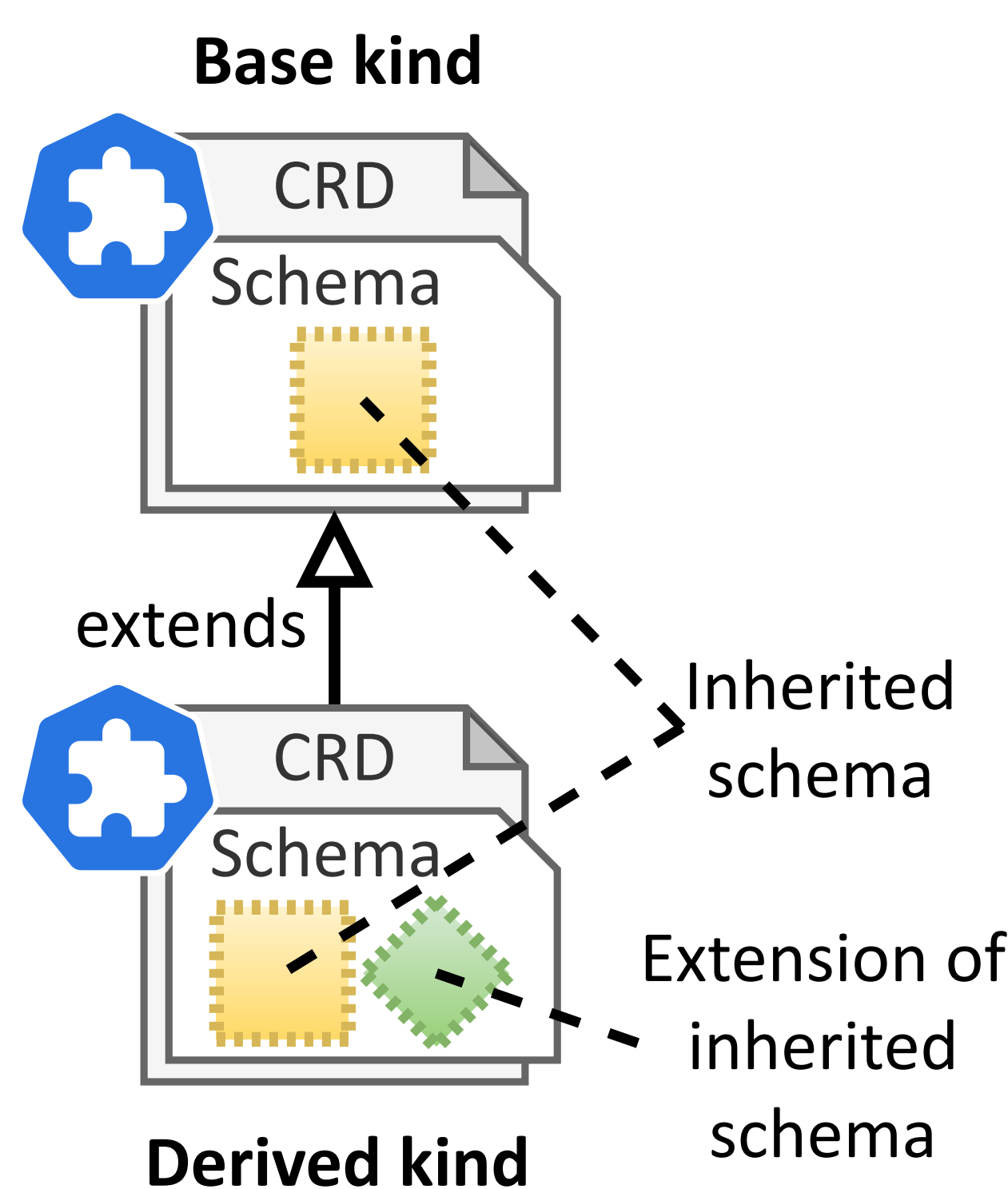


Modularity is Crucial For Adaptable Cloud Native Platforms and Applications

Platform Adaptability $\xrightarrow{\text{Enabled by}}$ Inheritance

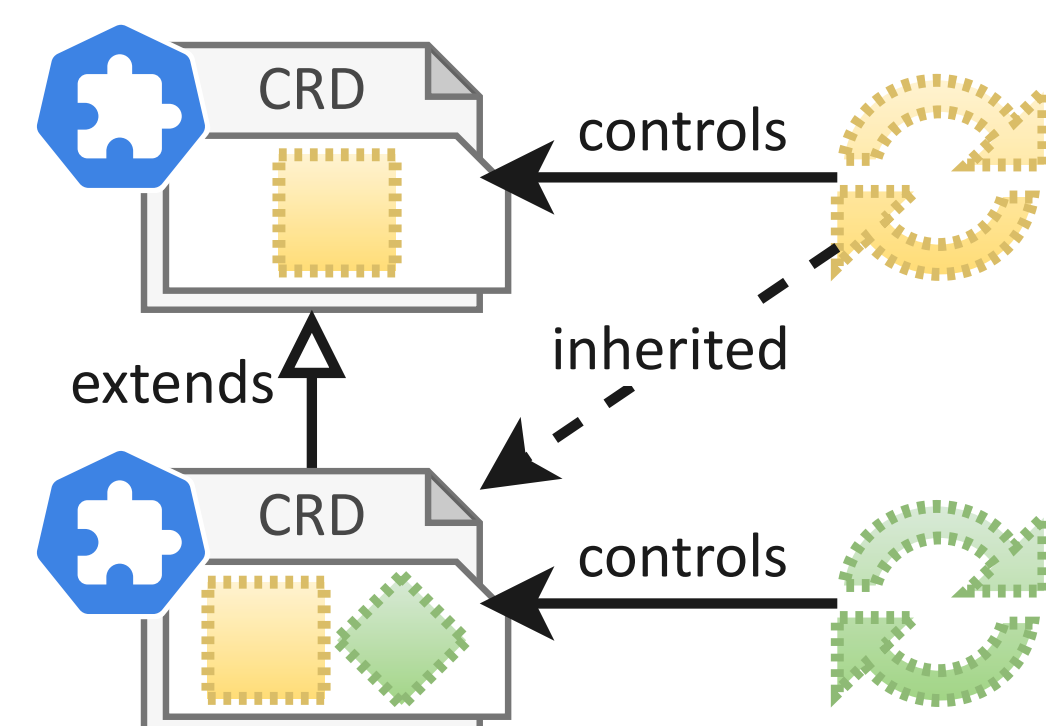
Interface Inheritance: Custom Resource Polymorphism By Reusing API Schemas

- Interface inheritance provides a mechanism for **type-safe substitution** of resource types.
- Subtype polymorphism** allows safely using a **subtype** where the **supertype** is expected.

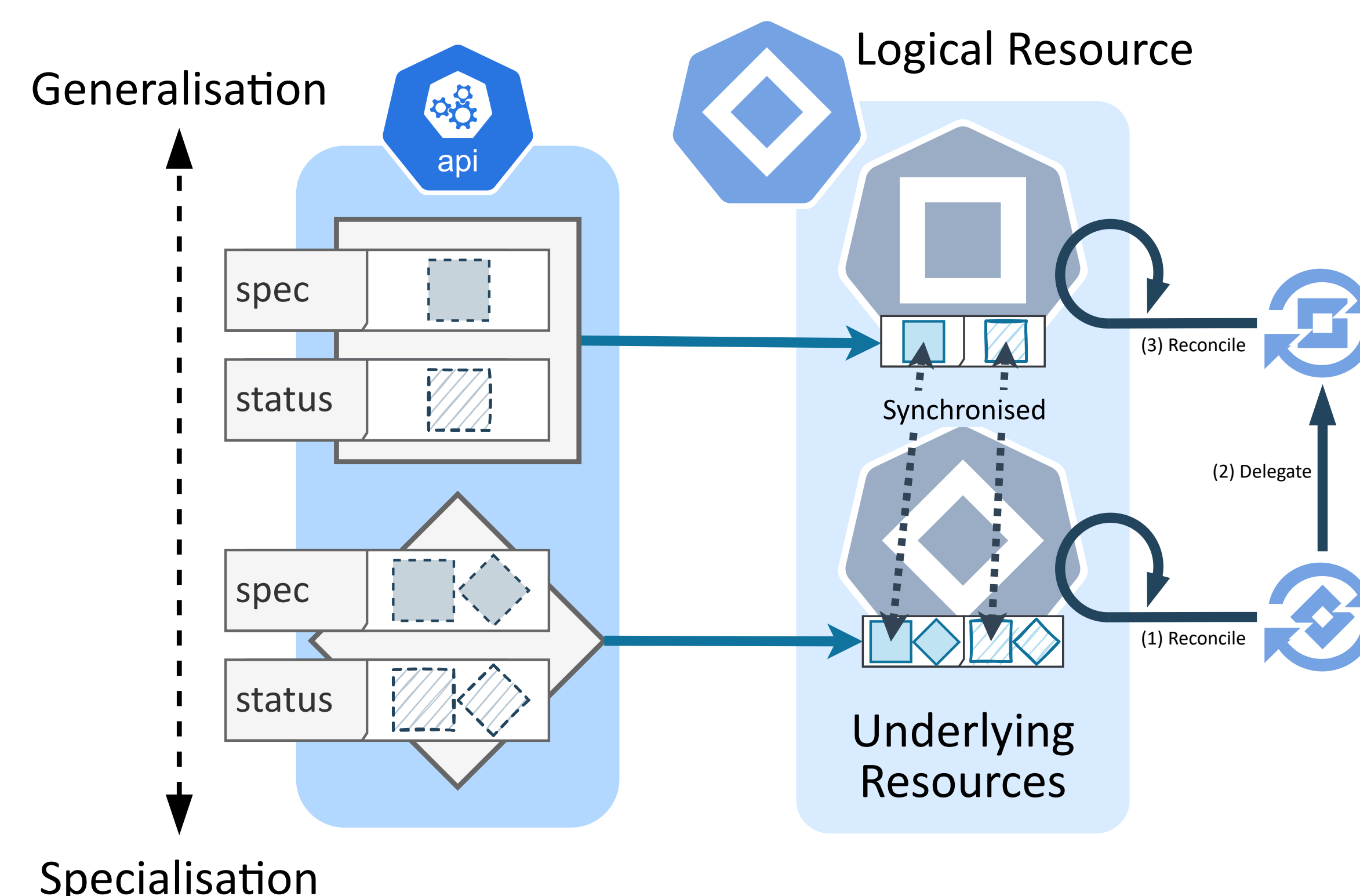


Implementation inheritance: Build Upon Existing Work By Reusing Controller Logic

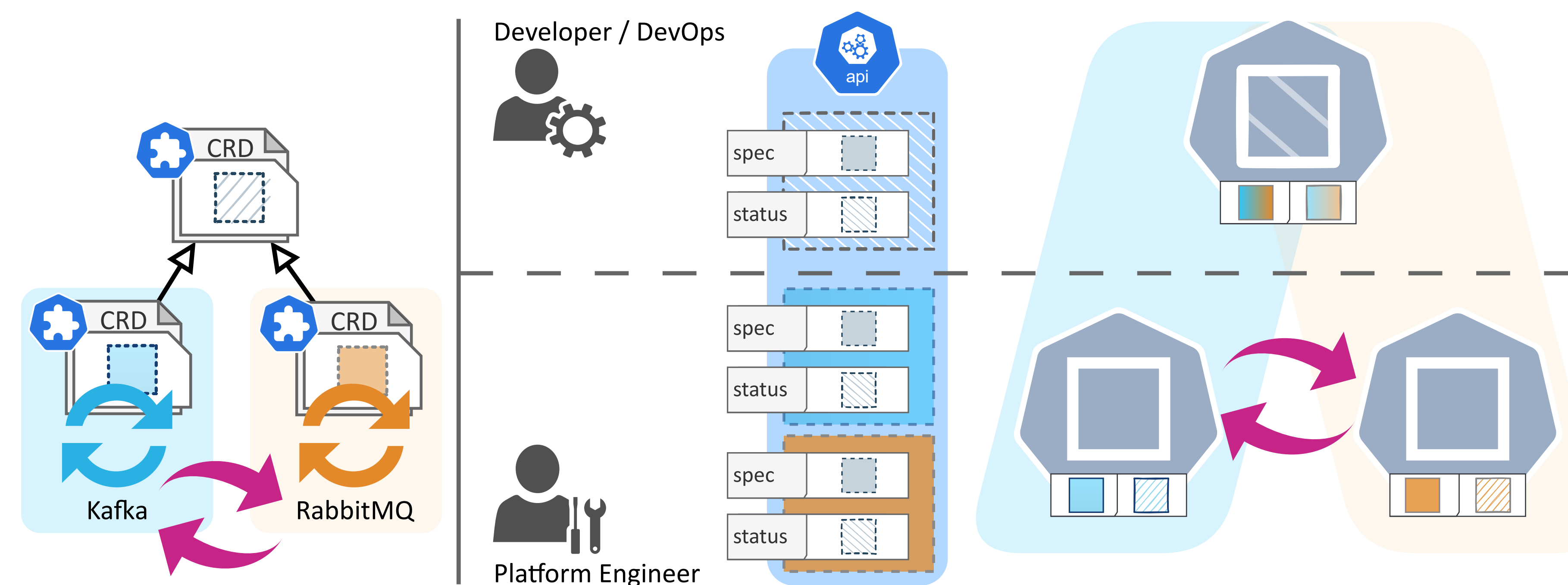
- Any controller(s) of the base kind are inherited **implicitly**.
- Turn-based coordination of controllers can be achieved via a **delegation** mechanism.



Custom Resource Interface and Implementation Inheritance



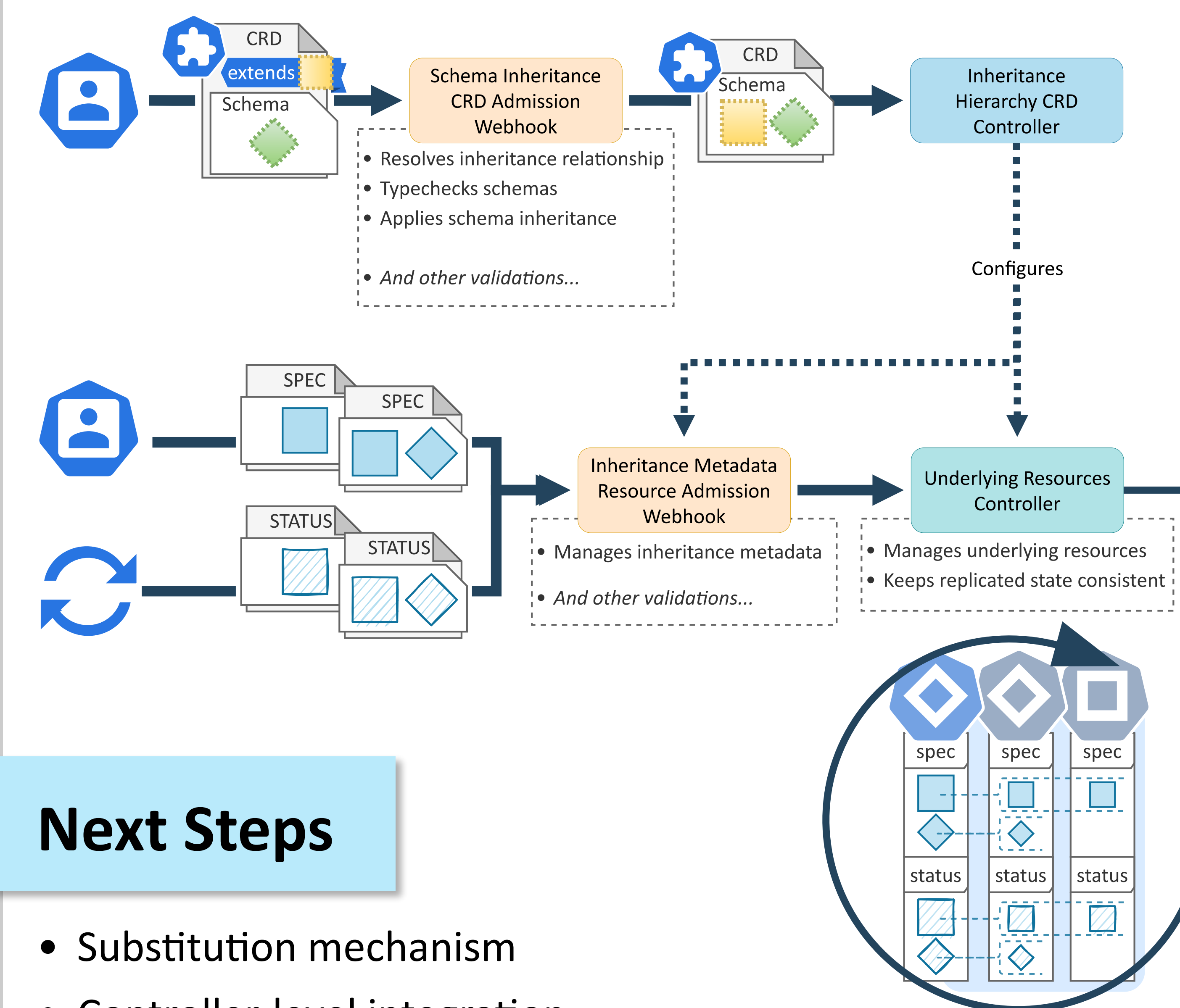
Adapt Underlying Platform Components By Reusing API Schemas To Decouple Controllers From Resource Types



- Substitution is transparent to clients, controllers, and other components that depend on the base API resource.
- Reduce vendor lock-in** by generalizing over cloud and/or technology vendors.

Implementation on Core Kubernetes

- Implementing CRD inheritance without modifying Kubernetes is feasible.
 - With annotations, admission webhooks, and controllers.



Next Steps

- Substitution mechanism
- Controller-level integration

Check it out on GitLab!

