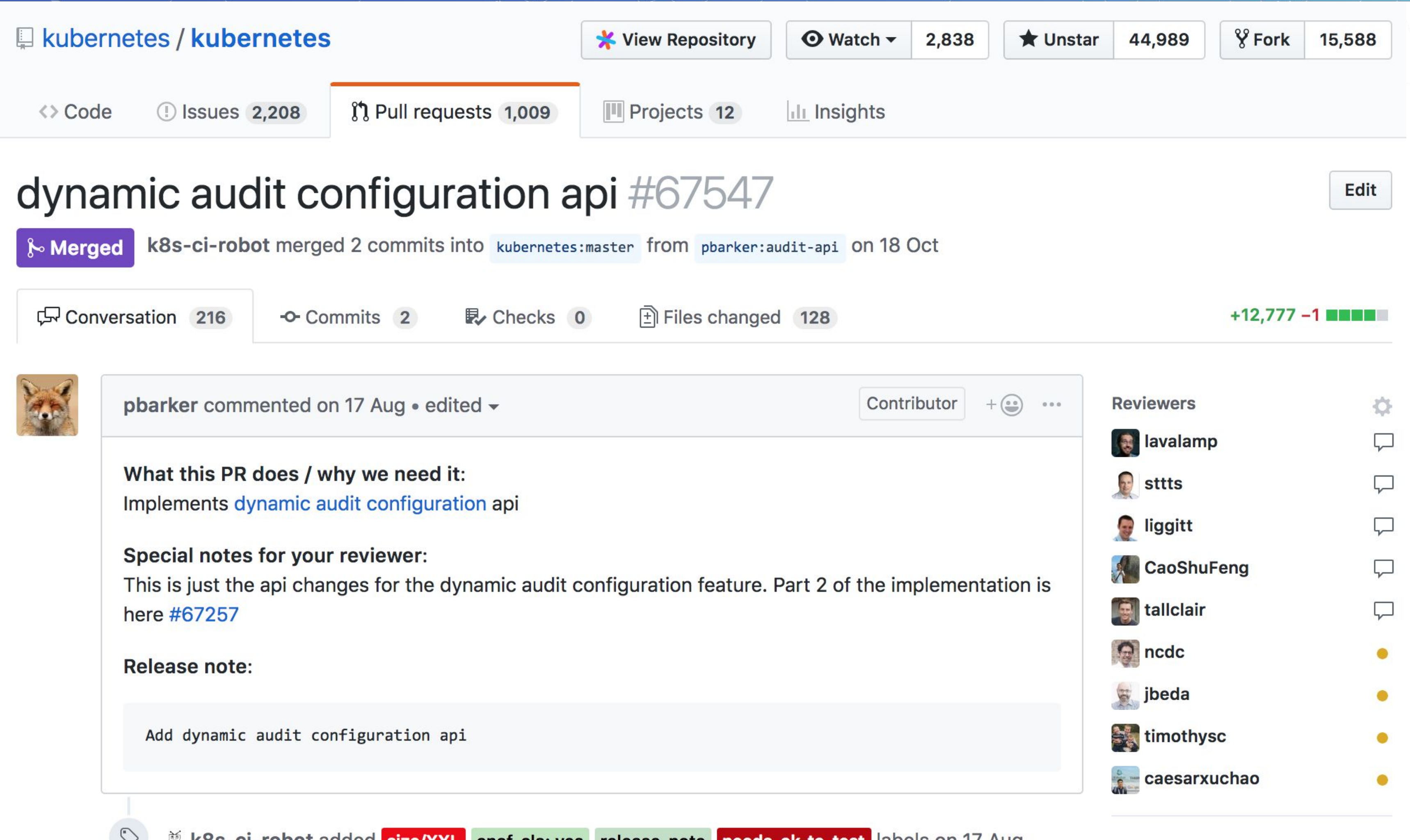
API Codebase Tour

Stefan Schimanski / @the_sttts / Red Hat

Hacking the kube-apiserver

kubernetes

Defining API types



kubernetes / kubernetes

View Repository | Watch 2,838 | Unstar 44,989 | Fork 15,588

Code | Issues 2,208 | Pull requests 1,009 | Projects 12 | Insights

dynamic audit configuration api #67547

Merged k8s-ci-robot merged 2 commits into `kubernetes:master` from `pbarker:audit-api` on 18 Oct

Conversation 216 | Commits 2 | Checks 0 | Files changed 128 | +12,777 −1

pbarker commented on 17 Aug • edited — Contributor

What this PR does / why we need it:
Implements dynamic audit configuration api

Special notes for your reviewer:
This is just the api changes for the dynamic audit configuration feature. Part 2 of the implementation is here #67257

Release note:

```
Add dynamic audit configuration api
```

k8s-ci-robot added size/XXL cncf-cla: yes release-note needs-ok-to-test labels on 17 Aug

Reviewers

- lavalamp
- sttts
- liggitt
- CaoShuFeng
- tallclair
- ncdc
- jbeda
- timothysc
- caesarxuchao

Golang types

v1alpha1 types: staging/src/k8s.io/api/auditregistration/v1alpha1

- types.go – actual Golang types (with JSON and Proto tags)
- register.go – registration code: AddToScheme

internal types: pkg/apis/auditregistration

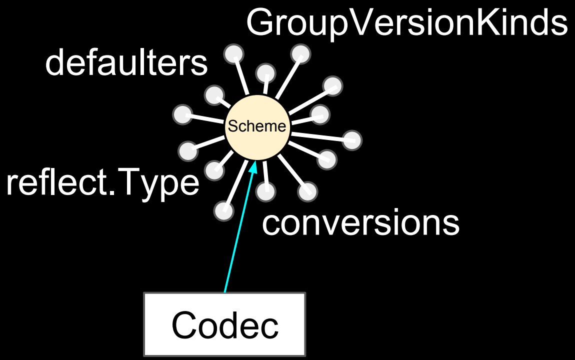
- types.go – internal (hub) Golang types (without JSON/Proto)
- register.go – registration code: AddToScheme

Installer: pkg/apis/auditregistration/install:

```
func Install(scheme *runtime.Scheme)
```

Scheme: register Golang types & Golang funcs w/ GroupVersionKind

`k8s.io/apimachinery/pkg/runtime.Scheme`



```go
type Scheme struct {
    // versionMap allows one to figure out the go type of an object with
    // the given version and name.
    gvkToType map[schema.GroupVersionKind]reflect.Type

    // typeToGroupVersion allows one to find metadata for a given go object.
    // The reflect.Type we index by should *not* be a pointer.
    typeToGVK map[reflect.Type][]schema.GroupVersionKind

    // defaulterFuncs is an array of interfaces to be called with an object to provide defaulting
    // the provided object must be a pointer.
    defaulterFuncs map[reflect.Type]func(interface{})

    // converter stores all registered conversion functions. It also has
    // default coverting behavior.
    converter *conversion.Converter
}
func (s *Scheme) AddKnownTypes(gv schema.GroupVersion, types ...Object) {
```

Golang types

v1alpha1 types: staging/src/k8s.io/api/auditregistration/v1alpha1

- types.go – actual Golang types (with JSON and Proto tags)
- register.go – registration code: AddToScheme

internal types: pkg/apis/auditregistration

- types.go – internal (hub) Golang types (without JSON/Proto)
- register.go – registration code: AddToScheme

Installer: pkg/apis/auditregistration/install:

```
    func Install(scheme *runtime.Scheme)
```

Generated Code

not in k8s.io/api!

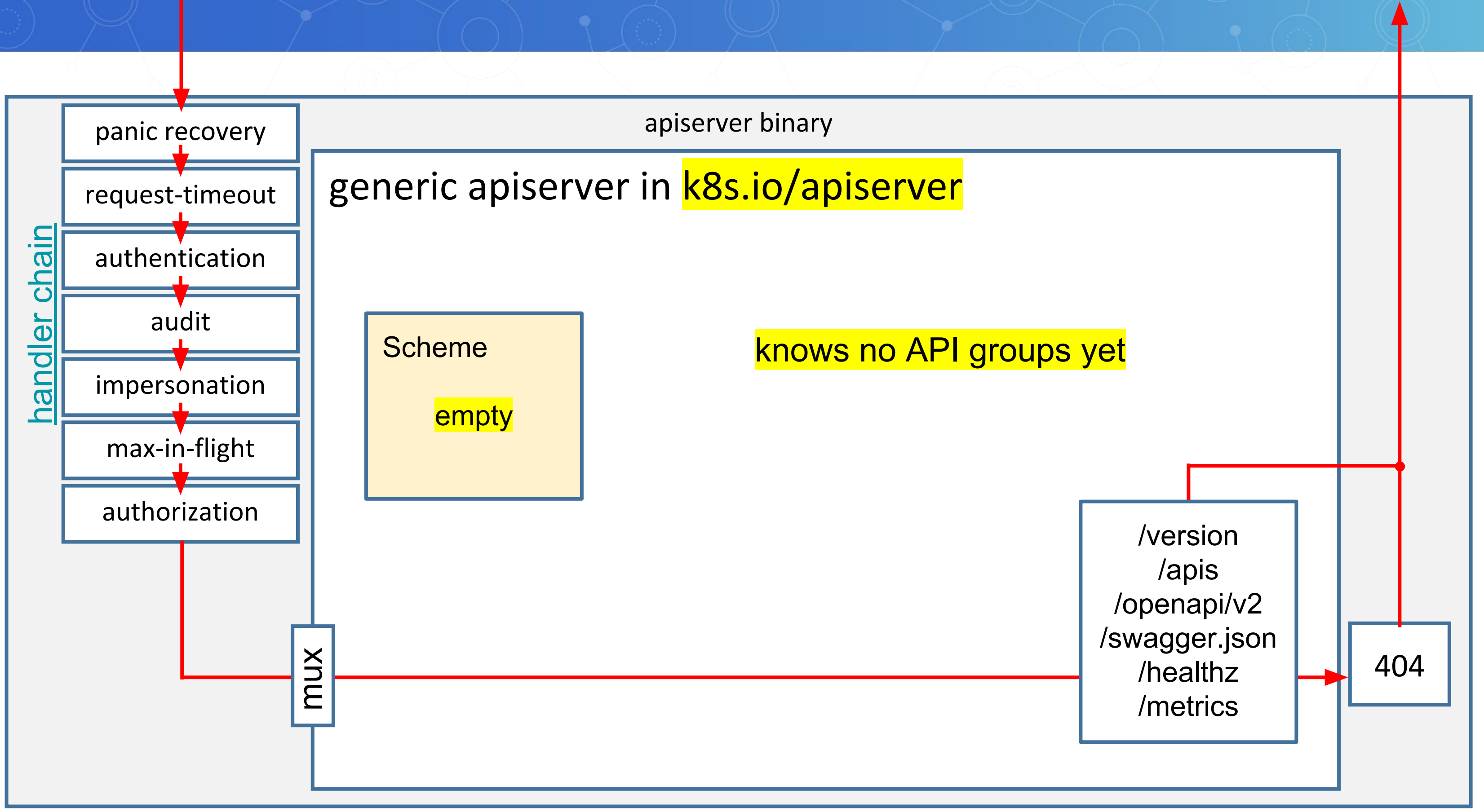
Conversions: pkg/apis/auditregistration/v1alpha1

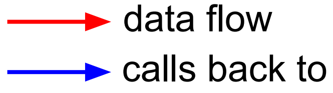
- conversion.go – custom conversions
- zz_generated.conversion.go – generated conversions

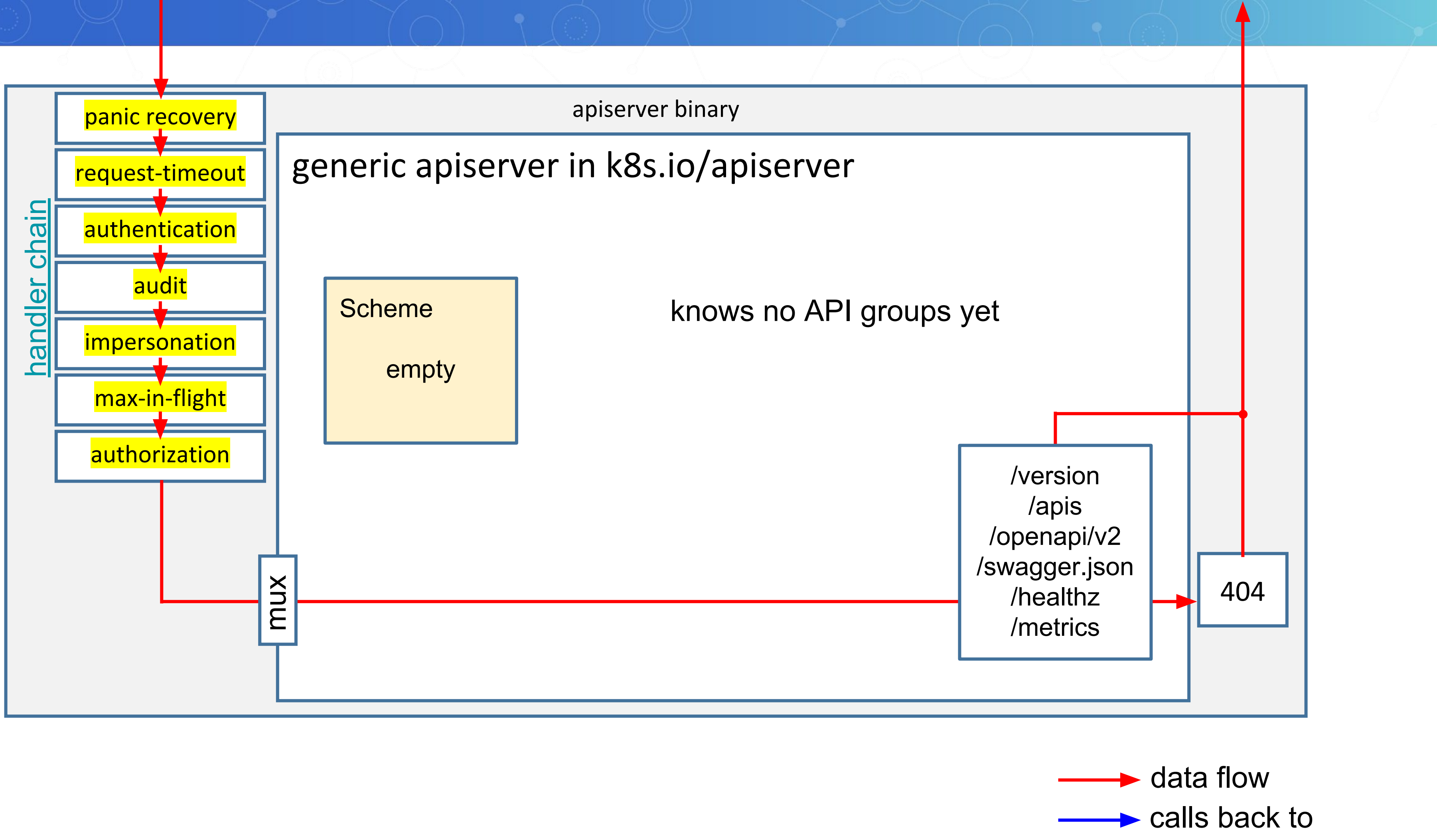
Defaults: zz_generated_defaults.go

DeepCopy: zz_generated_deepcopy.go

Serving the API


apiserver binary
handler chain
panic recovery
request-timeout
authentication
audit
impersonation
max-in-flight
authorization
generic apiserver in k8s.io/apiserver
Scheme
empty
knows no API groups yet
mux
/version
/apis
/openapi/v2
/swagger.json
/healthz
/metrics
404


data flow
calls back to



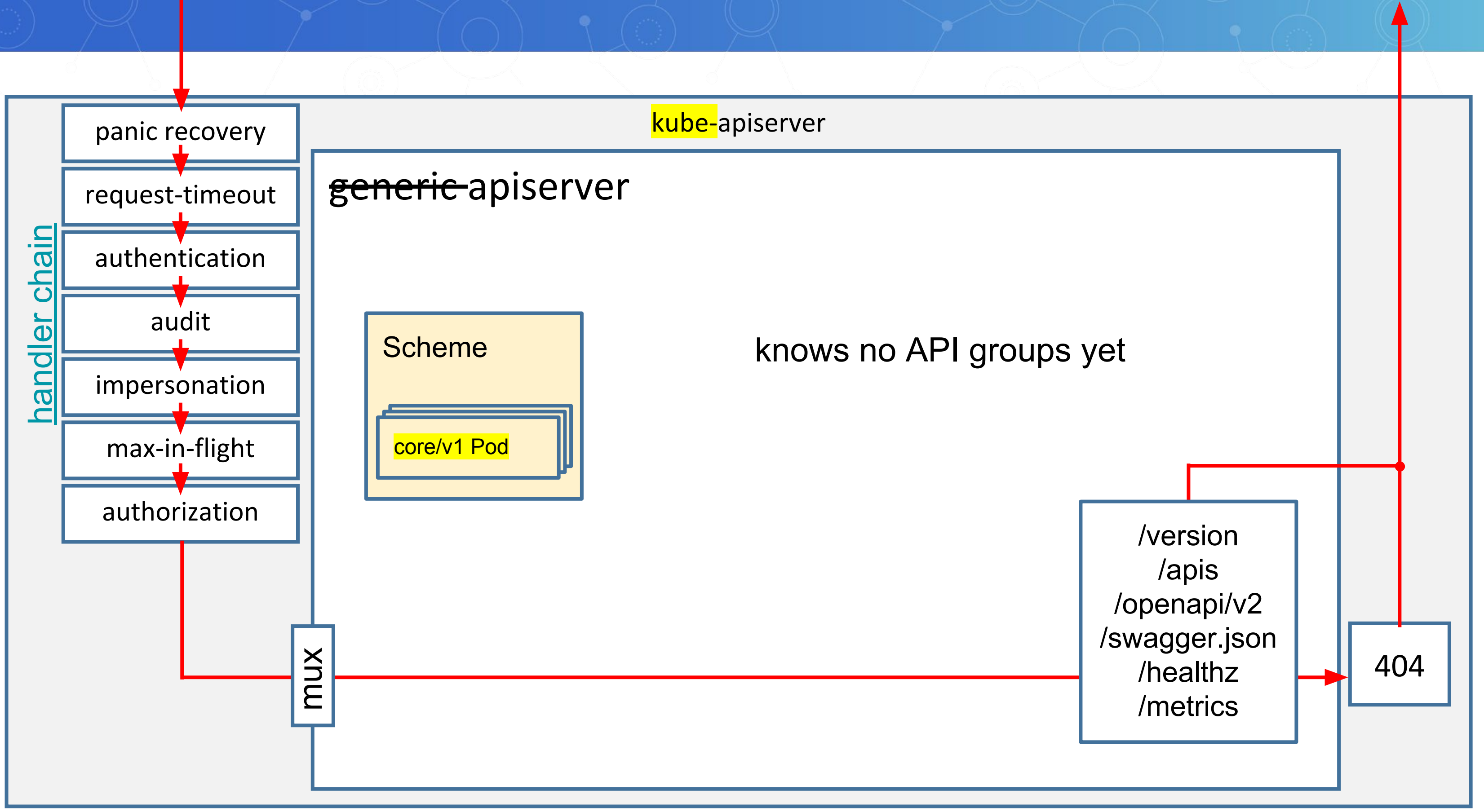
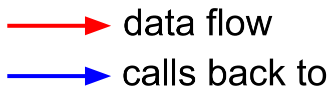
apiserver binary
handler chain
panic recovery
request-timeout
authentication
audit
impersonation
max-in-flight
authorization
generic apiserver in k8s.io/apiserver
Scheme
empty
knows no API groups yet
mux
/version
/apis
/openapi/v2
/swagger.json
/healthz
/metrics
404
data flow
calls back to

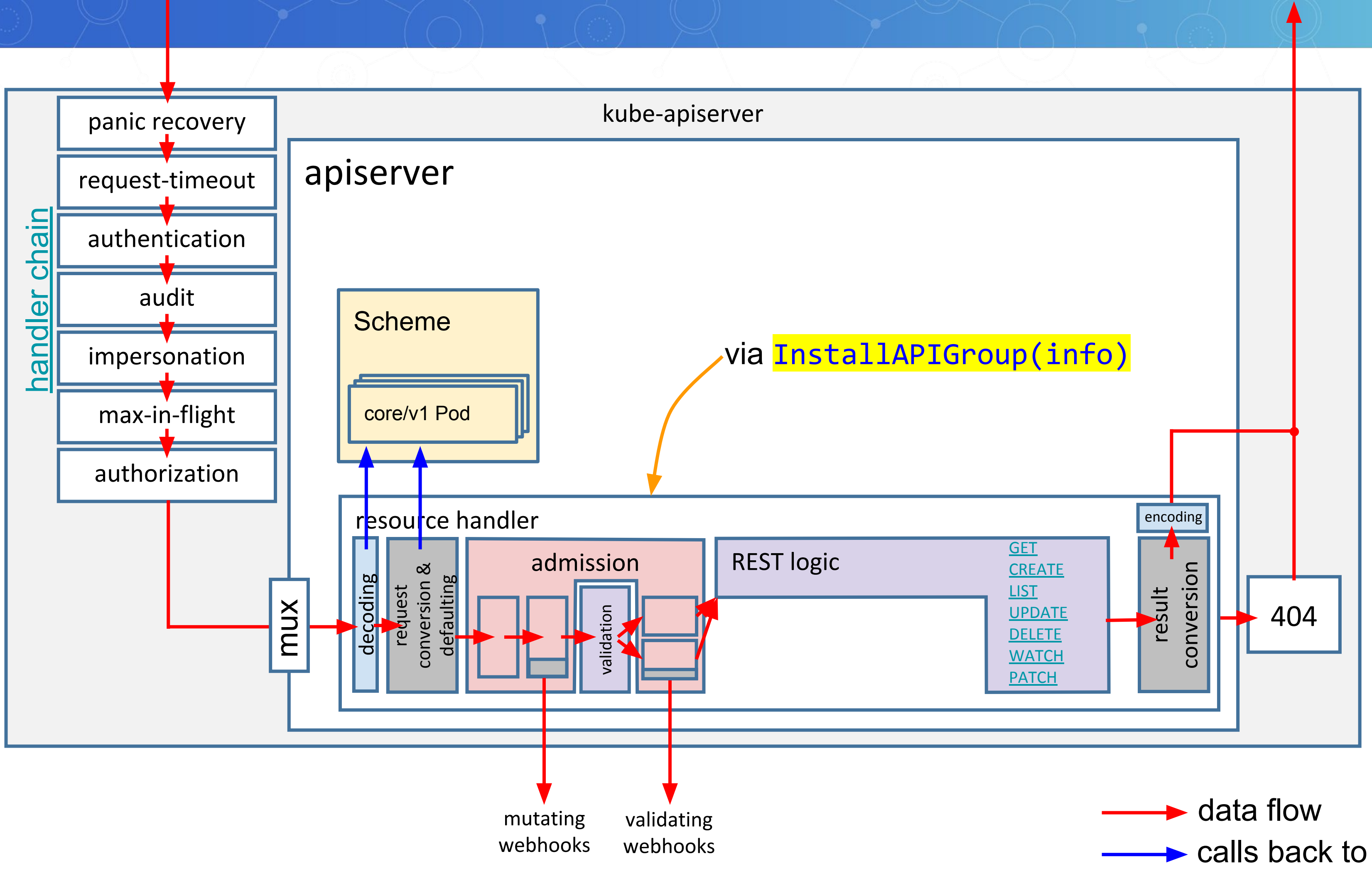
```
func DefaultBuildHandlerChain(apiHandler http.Handler, c *Config) http.Handler {
        handler := genericapifilters.WithAuthorization(apiHandler, ...)
        handler = genericfilters.WithMaxInFlightLimit(handler, ...)
        handler = genericapifilters.WithImpersonation(handler, ...)
        handler = genericapifilters.WithAudit(handler, ...)

        failedHandler := genericapifilters.Unauthorized(...)
        failedHandler = genericapifilters.WithFailedAuthenticationAudit(failedHandler, ...)

        handler = genericapifilters.WithAuthentication(handler, ..., failedHandler, ...)
        handler = genericfilters.WithCORS(handler, ...)
        handler = genericfilters.WithTimeoutForNonLongRunningRequests(handler, ...)
        handler = genericfilters.WithWaitGroup(handler, ...)
        handler = genericapifilters.WithRequestInfo(handler, ...)
        handler = genericfilters.WithPanicRecovery(handler)
        return handler
}
```

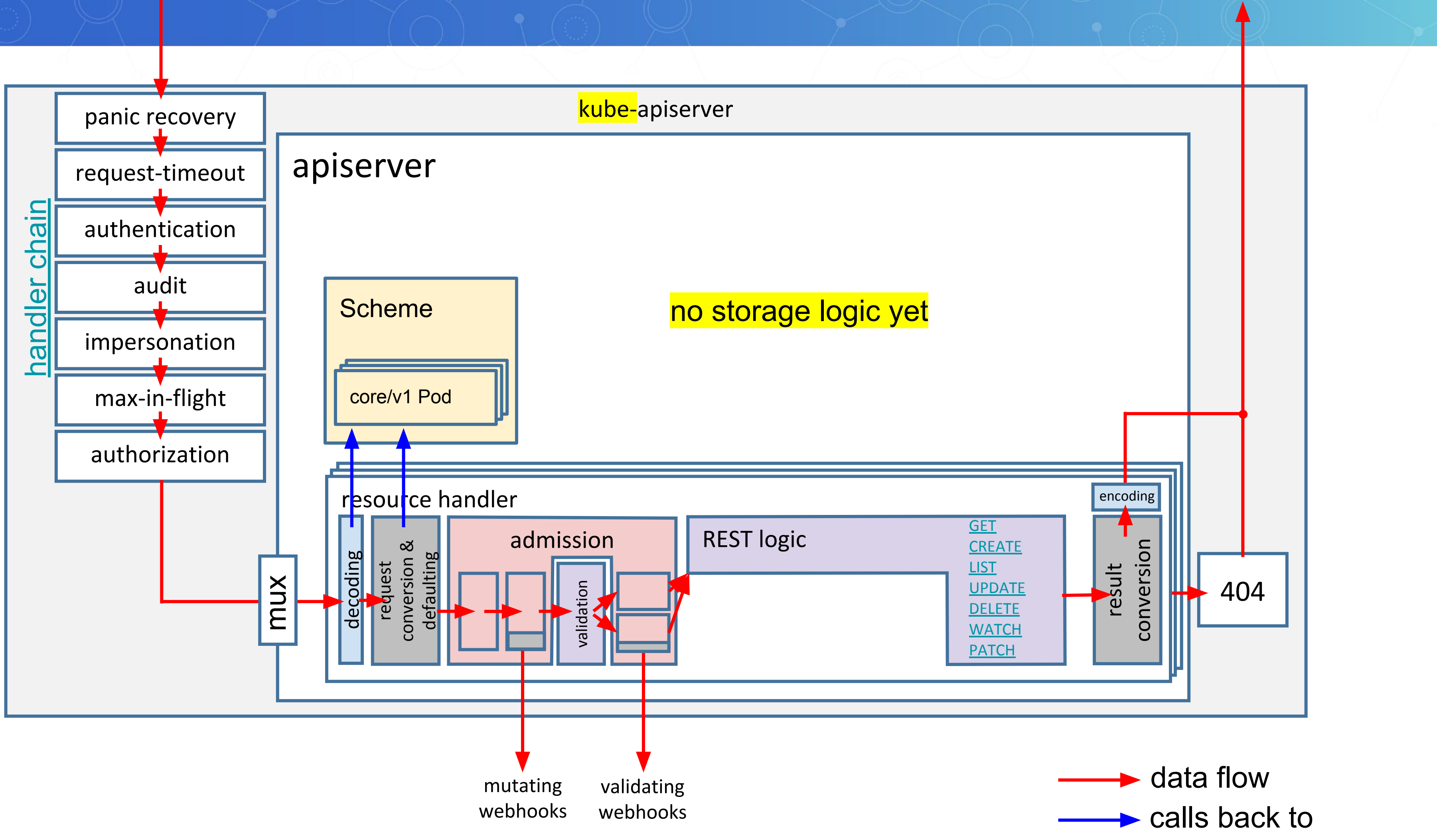
k8s.io/apiserver/pkg/server/config.go

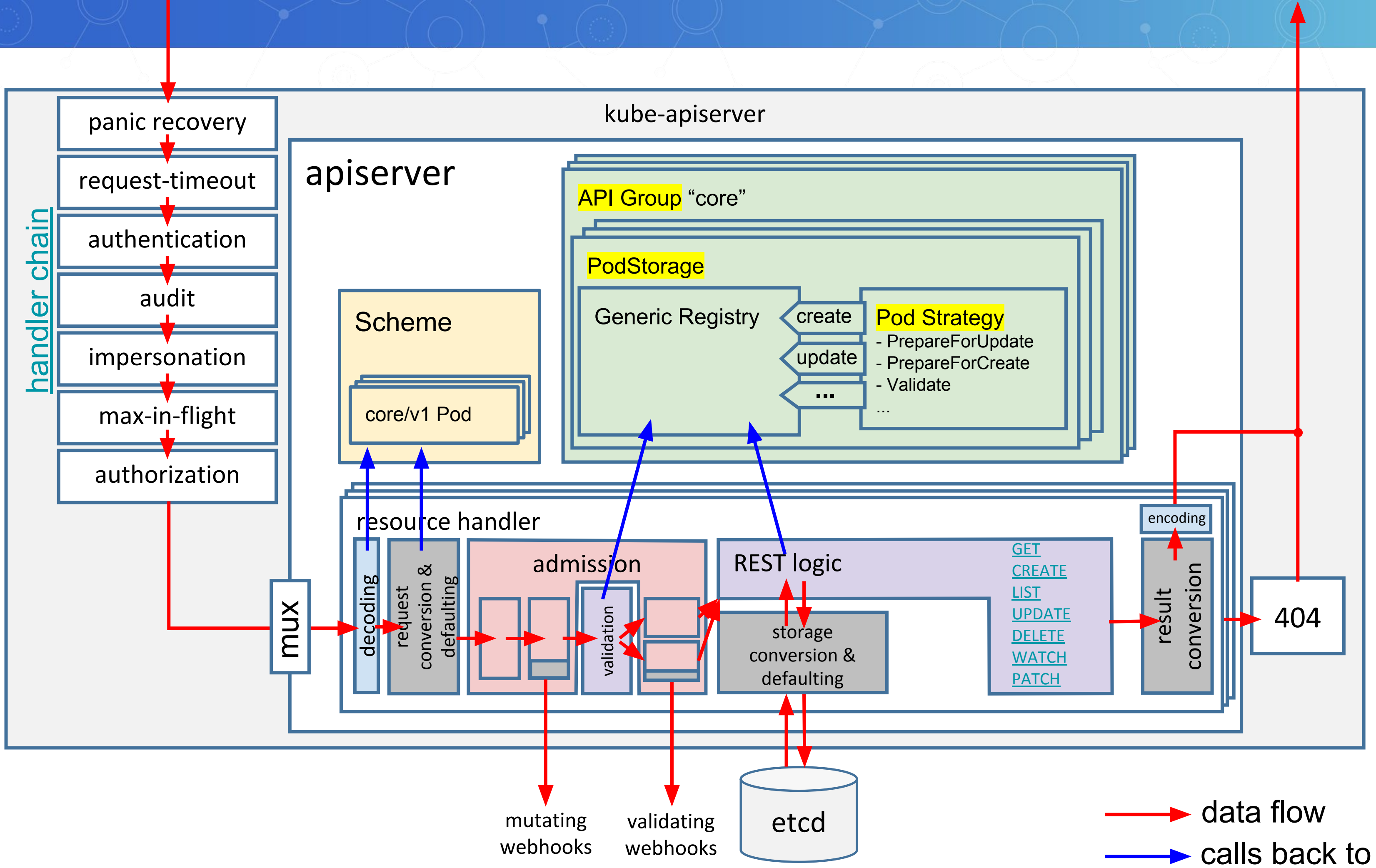

panic recovery
request-timeout
authentication
audit
impersonation
max-in-flight
authorization
handler chain
kube-apiserver
~~generic~~ apiserver
Scheme
core/v1 Pod
knows no API groups yet
mux
/version
/apis
/openapi/v2
/swagger.json
/healthz
/metrics
404

data flow
calls back to



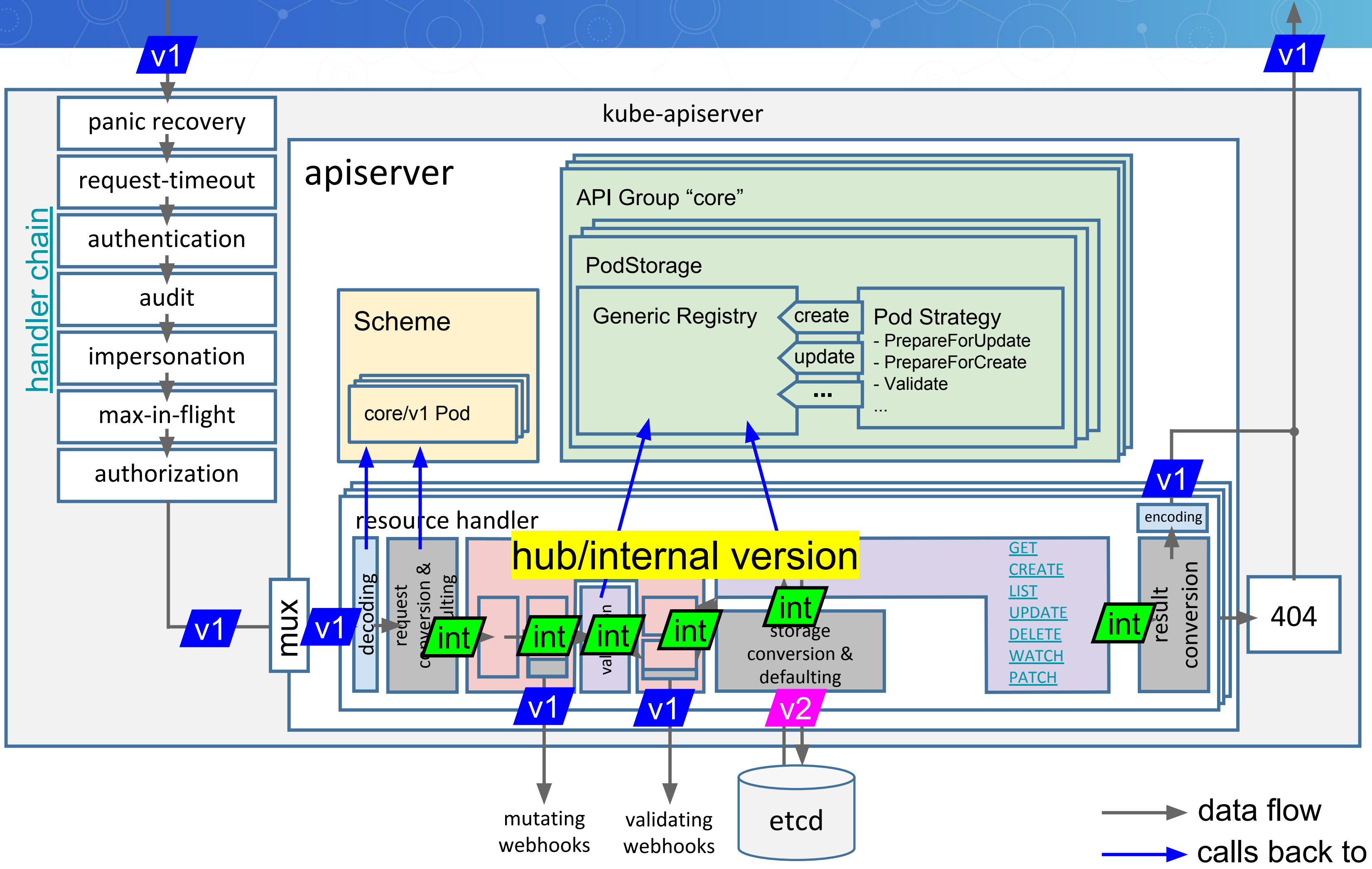
data flow

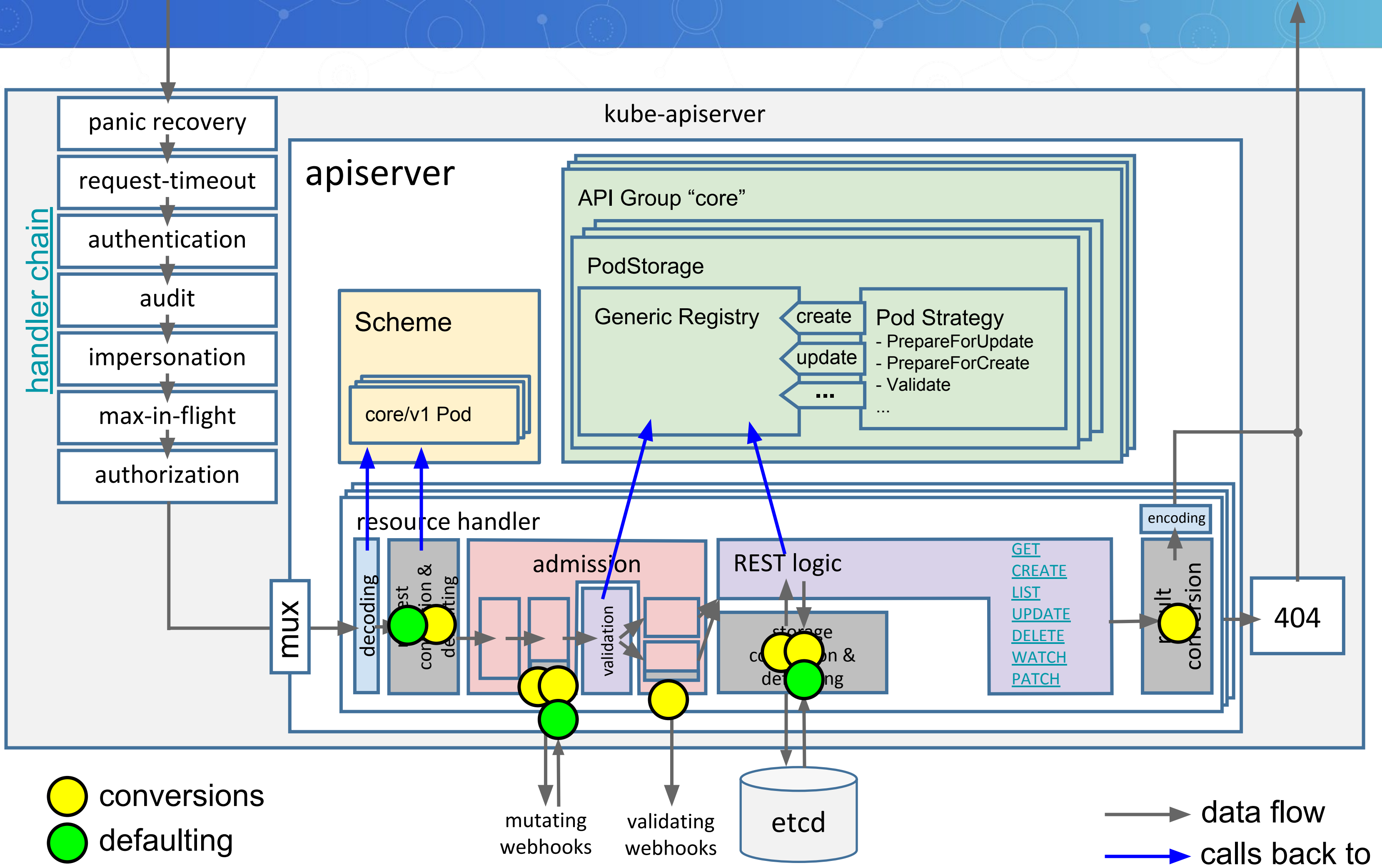
calls back to


kube-apiserver
handler chain
panic recovery
request-timeout
authentication
audit
impersonation
max-in-flight
authorization
apiserver
Scheme
core/v1 Pod
no storage logic yet
mux
resource handler
decoding
request conversion & defaulting
admission
validation
REST logic
GET
CREATE
LIST
UPDATE
DELETE
WATCH
PATCH
encoding
result conversion
404
mutating webhooks
validating webhooks
data flow
calls back to



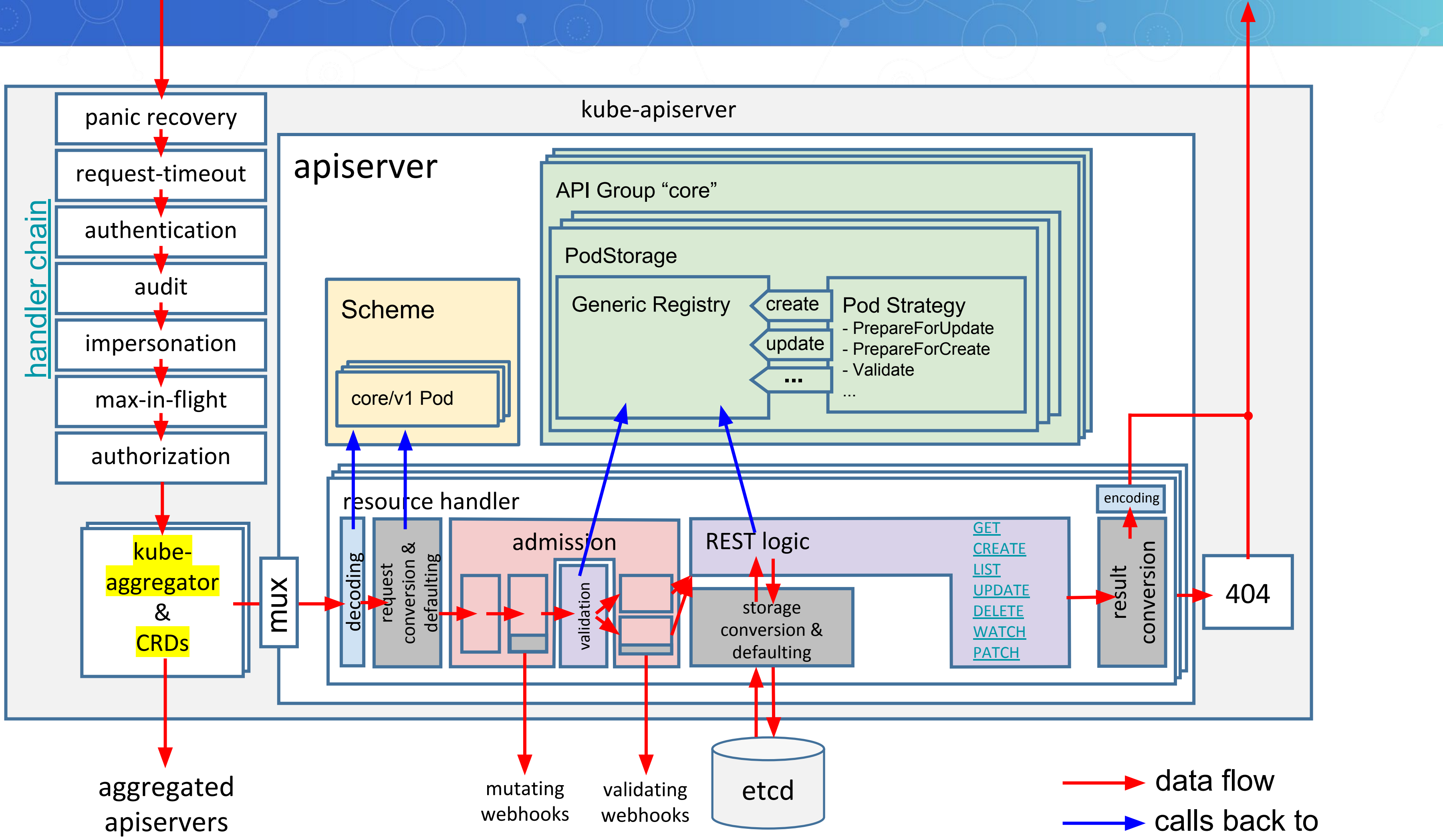
- data flow
- calls back to

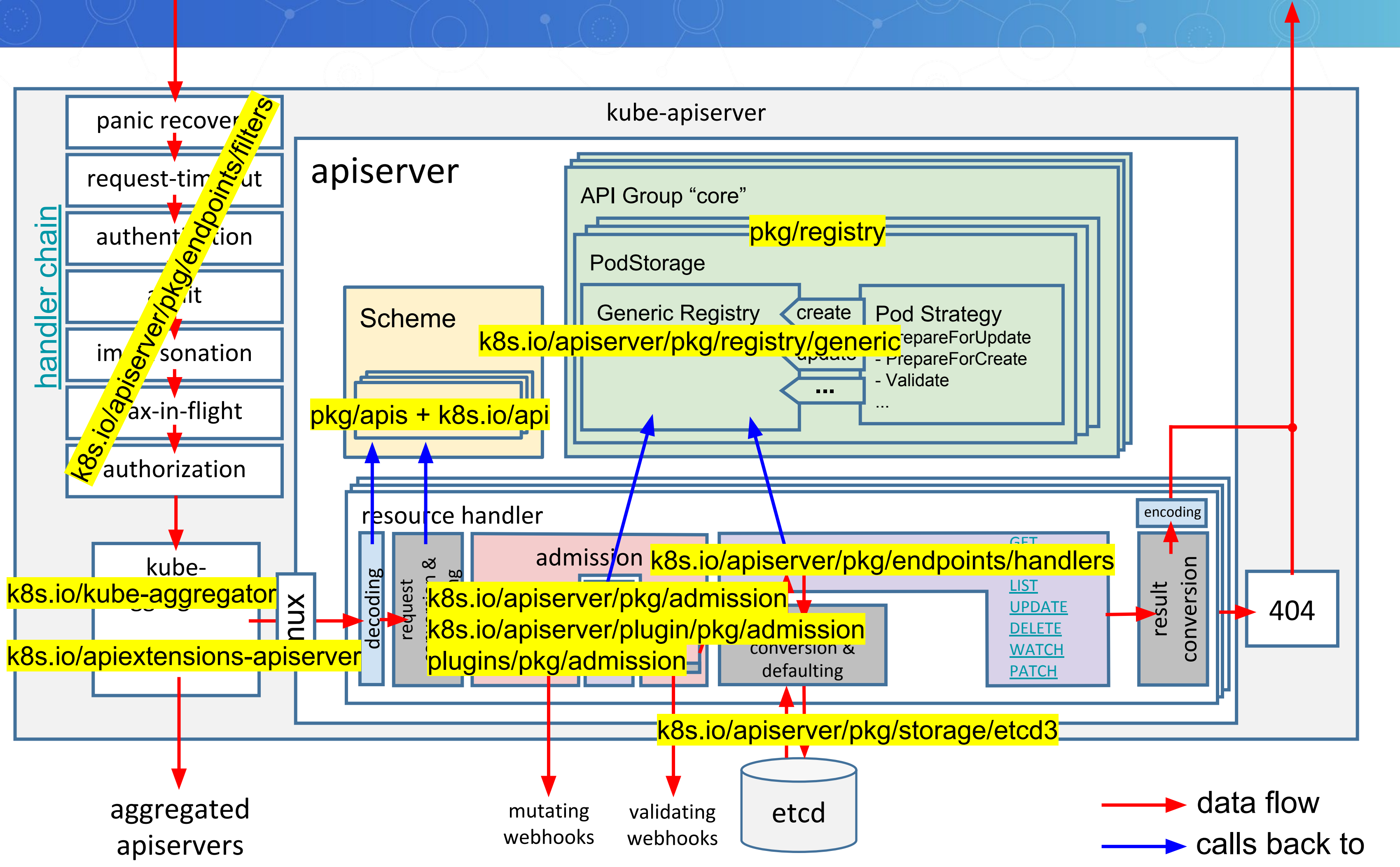

v1
kube-apiserver
panic recovery
request-timeout
authentication
audit
impersonation
max-in-flight
authorization
handler chain
apiserver
API Group "core"
PodStorage
Generic Registry
create
update
...
Pod Strategy
- PrepareForUpdate
- PrepareForCreate
- Validate
...
Scheme
core/v1 Pod
resource handler
hub/internal version
mux
decoding
request conversion & defaulting
int
storage conversion & defaulting
GET
CREATE
LIST
UPDATE
DELETE
WATCH
PATCH
encoding
result conversion
404
v2
mutating webhooks
validating webhooks
etcd
data flow
calls back to



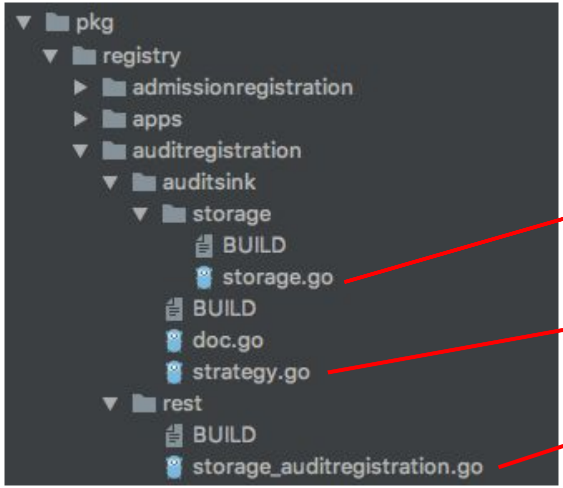
● conversions
● defaulting

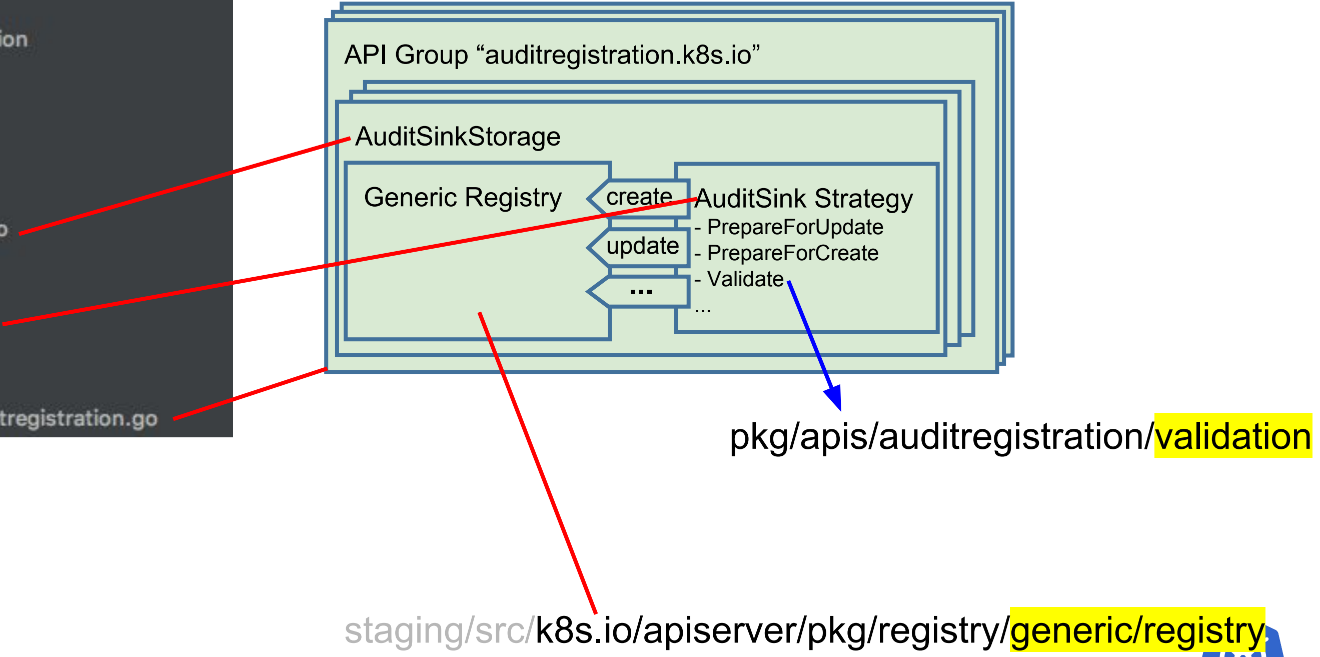
→ data flow
→ calls back to


kube-apiserver
handler chain
panic recovery
request-timeout
authentication
audit
impersonation
max-in-flight
authorization
kube-aggregator & CRDs
mux
apiserver
Scheme
core/v1 Pod
API Group “core”
PodStorage
Generic Registry
create
update
...
Pod Strategy
- PrepareForUpdate
- PrepareForCreate
- Validate
...
resource handler
decoding
request conversion & defaulting
admission
validation
REST logic
storage conversion & defaulting
GET
CREATE
LIST
UPDATE
DELETE
WATCH
PATCH
encoding
result conversion
404
aggregated apiservers
mutating webhooks
validating webhooks
etcd
data flow
calls back to


kube-apiserver
panic recovery
request-timeout
authentication
audit
impersonation
max-in-flight
authorization
handler chain
k8s.io/apiserver/pkg/endpoints/filters
kube-aggregator
k8s.io/kube-aggregator
k8s.io/apiextensions-apiserver
mux
apiserver
Scheme
pkg/apis + k8s.io/api
API Group "core"
pkg/registry
PodStorage
Generic Registry
create
update
...
k8s.io/apiserver/pkg/registry/generic
Pod Strategy
- PrepareForUpdate
- PrepareForCreate
- Validate
...
resource handler
decoding
request conversion & defaulting
admission
k8s.io/apiserver/pkg/endpoints/handlers
k8s.io/apiserver/pkg/admission
k8s.io/apiserver/plugin/pkg/admission
plugins/pkg/admission
conversion & defaulting
GET
LIST
UPDATE
DELETE
WATCH
PATCH
encoding
result conversion
404
k8s.io/apiserver/pkg/storage/etcd3
aggregated apiservers
mutating webhooks
validating webhooks
etcd
data flow
calls back to

“The registry” of a resource


```
▼ pkg
  ▼ registry
    ▶ admissionregistration
    ▶ apps
    ▼ auditregistration
      ▼ auditsink
        ▼ storage
            BUILD
            storage.go
          BUILD
          doc.go
          strategy.go
      ▼ rest
          BUILD
          storage_auditregistration.go
```



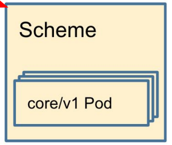
API Group “auditregistration.k8s.io”
AuditSinkStorage
Generic Registry
create
update
...
AuditSink Strategy
- PrepareForUpdate
- PrepareForCreate
- Validate
...
pkg/apis/auditregistration/validation
staging/src/k8s.io/apiserver/pkg/registry/generic/registry

Plumbing into kube-apiserver

pkg/master/import_known_versions.go

```
import (
  _ "k8s.io/kubernetes/pkg/apis/auditregistration/install"
)
```

func init()



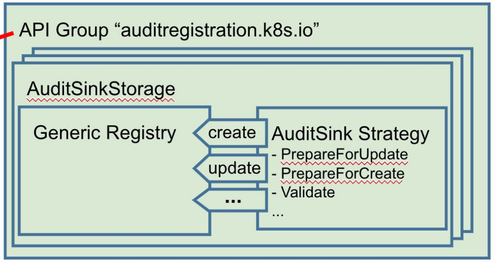
legacyscheme.Scheme

pkg/master/master.go

```
import (
  auditregistrationrest "k8s.io/kubernetes/pkg/registry/auditregistration/rest"
)

restStorageProviders := []RESTStorageProvider{
  auditregistrationrest.RESTStorageProvider{},
  autoscalingrest.RESTStorageProvider{},
  ...
}
apiserver.InstallAPIs(..., restStorageProviders...)
```

installs handlers into the mux



Build system plumbing

- hack/.golint_failures

  ignore lint errors due to generated code

- hack/lib/init.sh

  add to `KUBE_AVAILABLE_GROUP_VERSIONS`,
  used by many `hack/` scripts

- hack/update-generated-protobuf-dockerized.sh

  generate Protobuf code, independent from
  `KUBE_AVAILABLE_GROUP_VERSIONS` for some reason

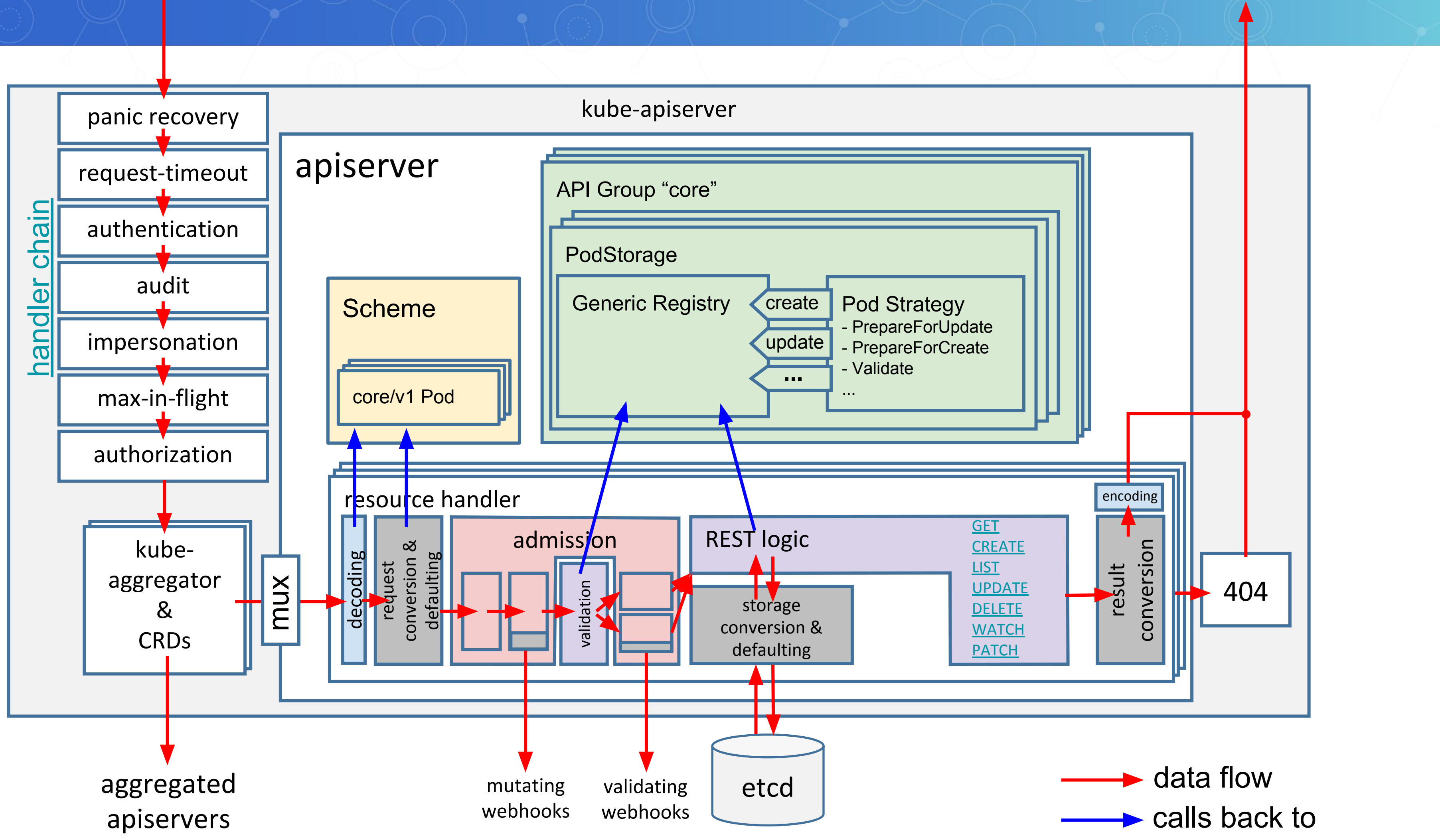
```
$ make WHAT=cmd/hyperkube

$ RUNTIME_CONFIG=auditregistration.k8s.io/v1alpha1=true \
    hack/local-up-cluster.sh

$ kubectl get --raw /apis | grep auditregistration.k8s.io
```

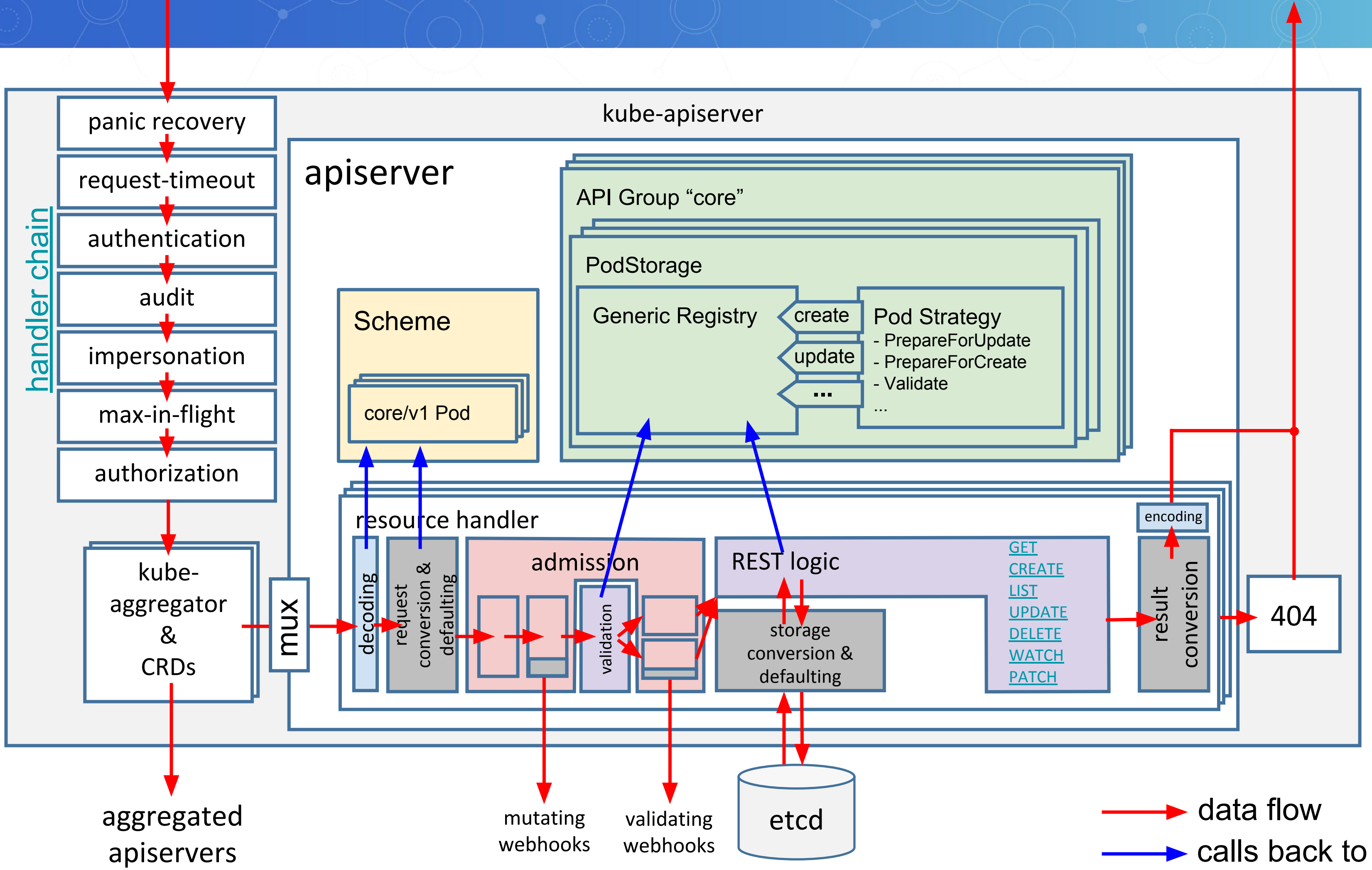
🎉

Live Debugging


kube-apiserver
handler chain
panic recovery
request-timeout
authentication
audit
impersonation
max-in-flight
authorization
kube-aggregator & CRDs
mux
aggregated apiservers
apiserver
Scheme
core/v1 Pod
API Group "core"
PodStorage
Generic Registry
create
update
...
Pod Strategy
- PrepareForUpdate
- PrepareForCreate
- Validate
...
resource handler
decoding
request conversion & defaulting
admission
validation
REST logic
GET
CREATE
LIST
UPDATE
DELETE
WATCH
PATCH
storage conversion & defaulting
encoding
result conversion
404
mutating webhooks
validating webhooks
etcd
data flow
calls back to

Live Debugging*

* perfectly written down in xmudrii's https://xmudrii.com/posts/debugging-kubernetes/



data flow
calls back to

@lavalamp’s “Live API Code Review” after the break