## Are you testing your Observability?

GoDays 2020, Berlin



## 





What is observability 🦫



Why Prometheus?



Metrics are essential \*\*

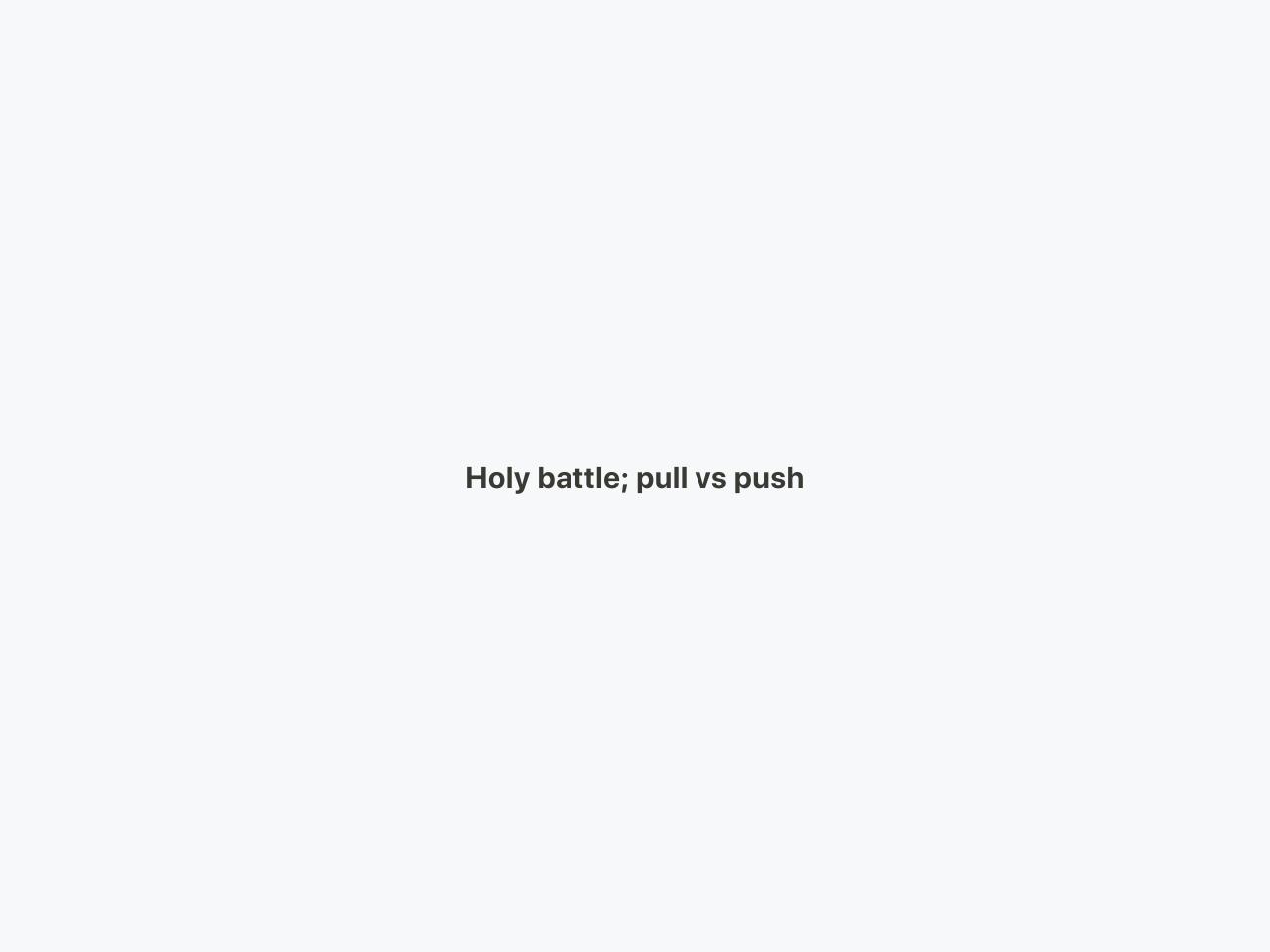
Prometheus - How to expose?

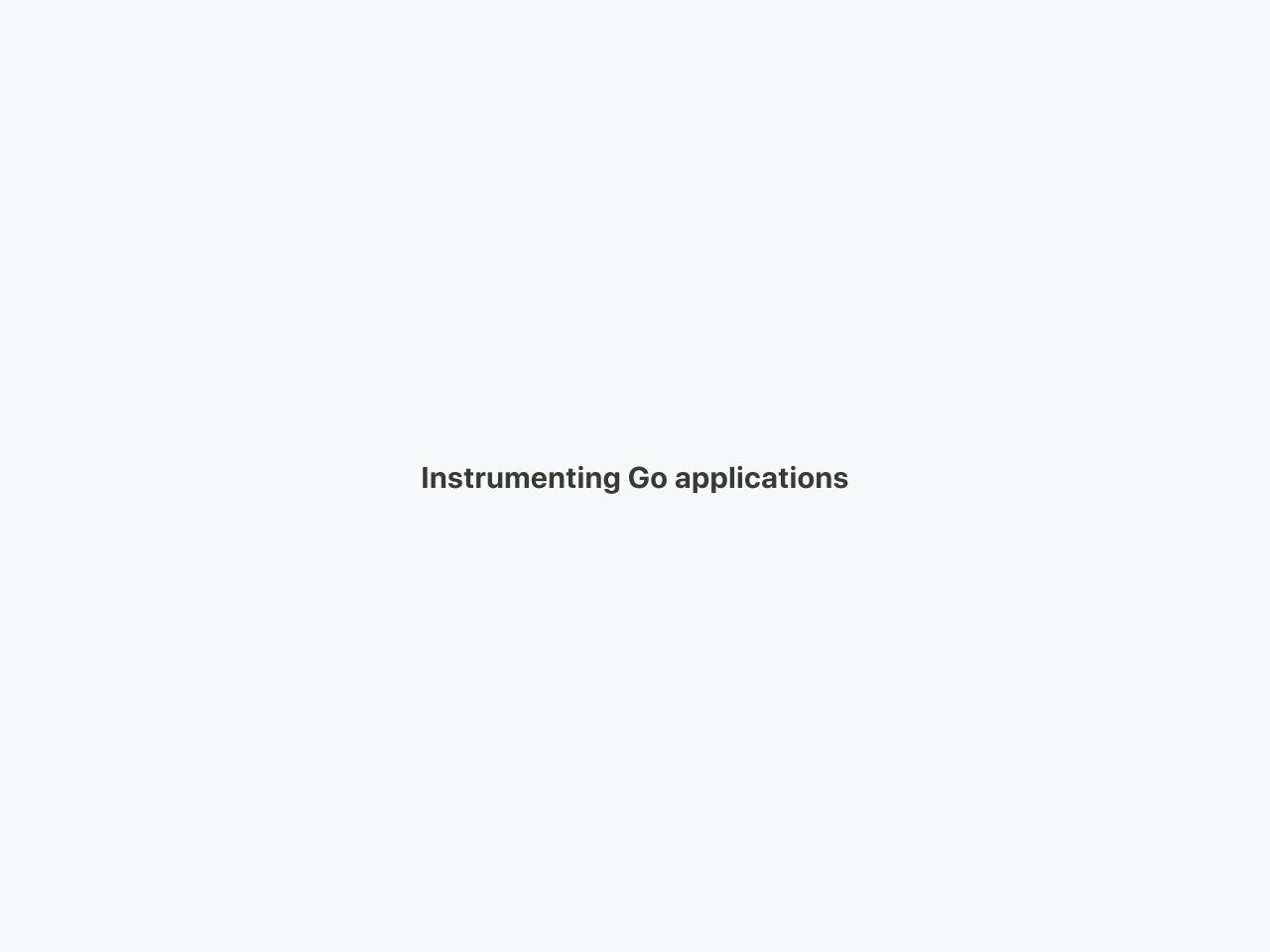
## This is a title

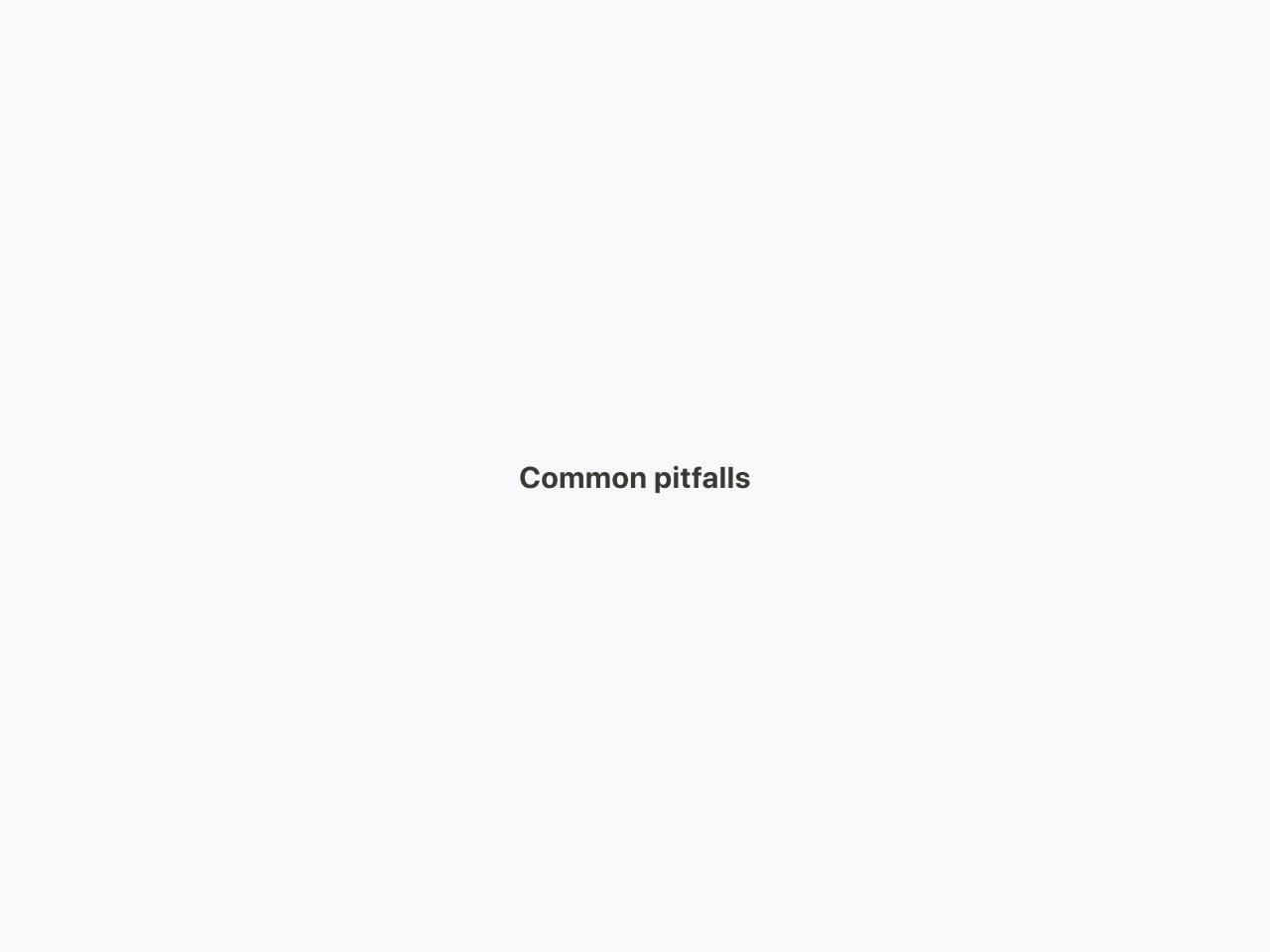
```
// Register standard Go metric collectors, which are by default registered when using global registry.
reg.MustRegister(
   prometheus.NewGoCollector(),
   prometheus.NewProcessCollector(prometheus.ProcessCollectorOpts{}),
   prometheus.NewGaugeFunc(prometheus.GaugeOpts{
        Namespace: namespace,
        Name: "configured_max_objects",
        Help: "Configured maximum number of objects cache can take.",
}, func() float64 { return float64(*maxObjectsCount) }),
```

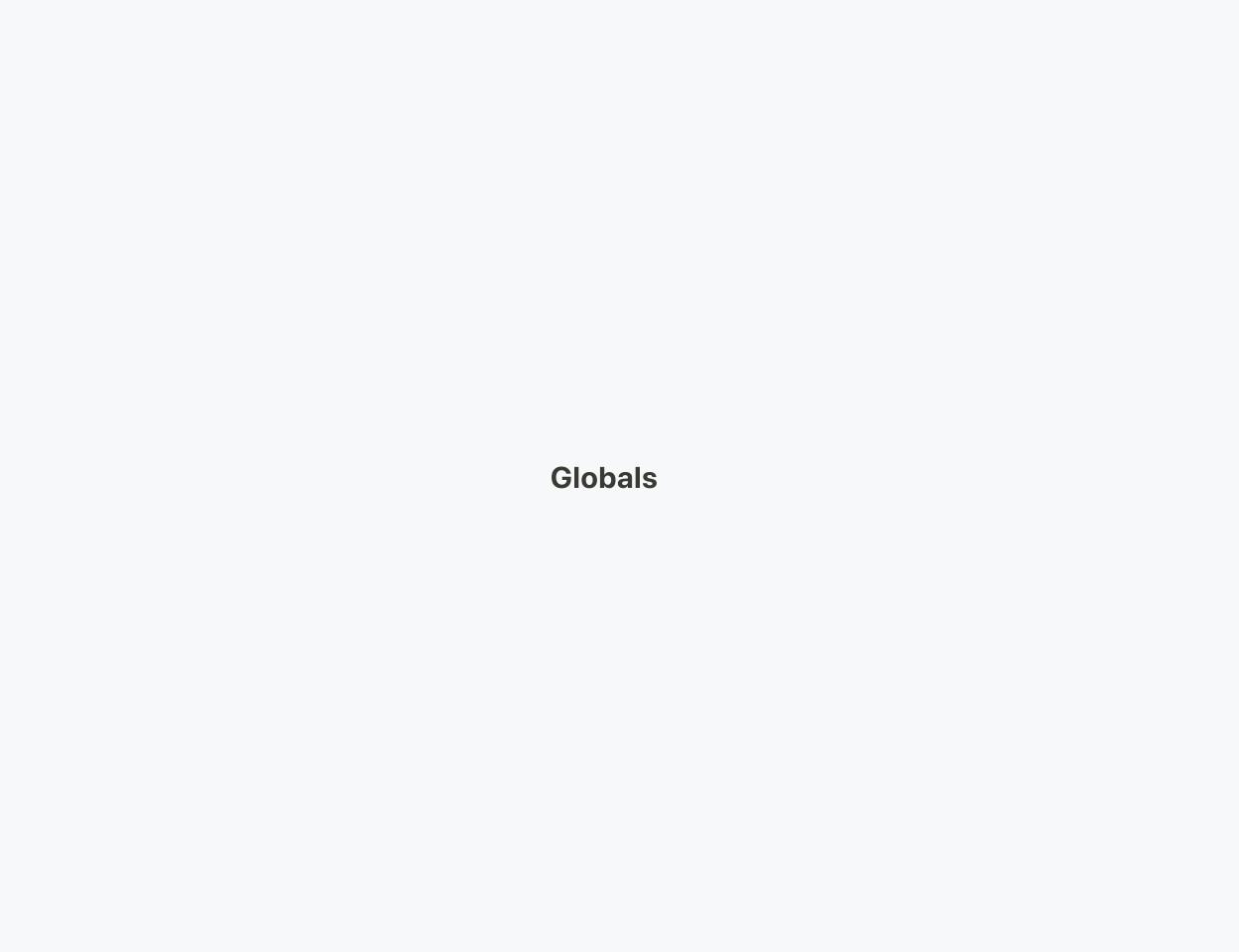
and this a subtitle

```
mux.Handle("/metrics", promhttp.HandlerFor(reg, promhttp.HandlerOpts{}))
mux.Handle("/set", http.HandlerFunc(c.SetHandler))
mux.Handle("/get", http.HandlerFunc(c.GetHandler))
```





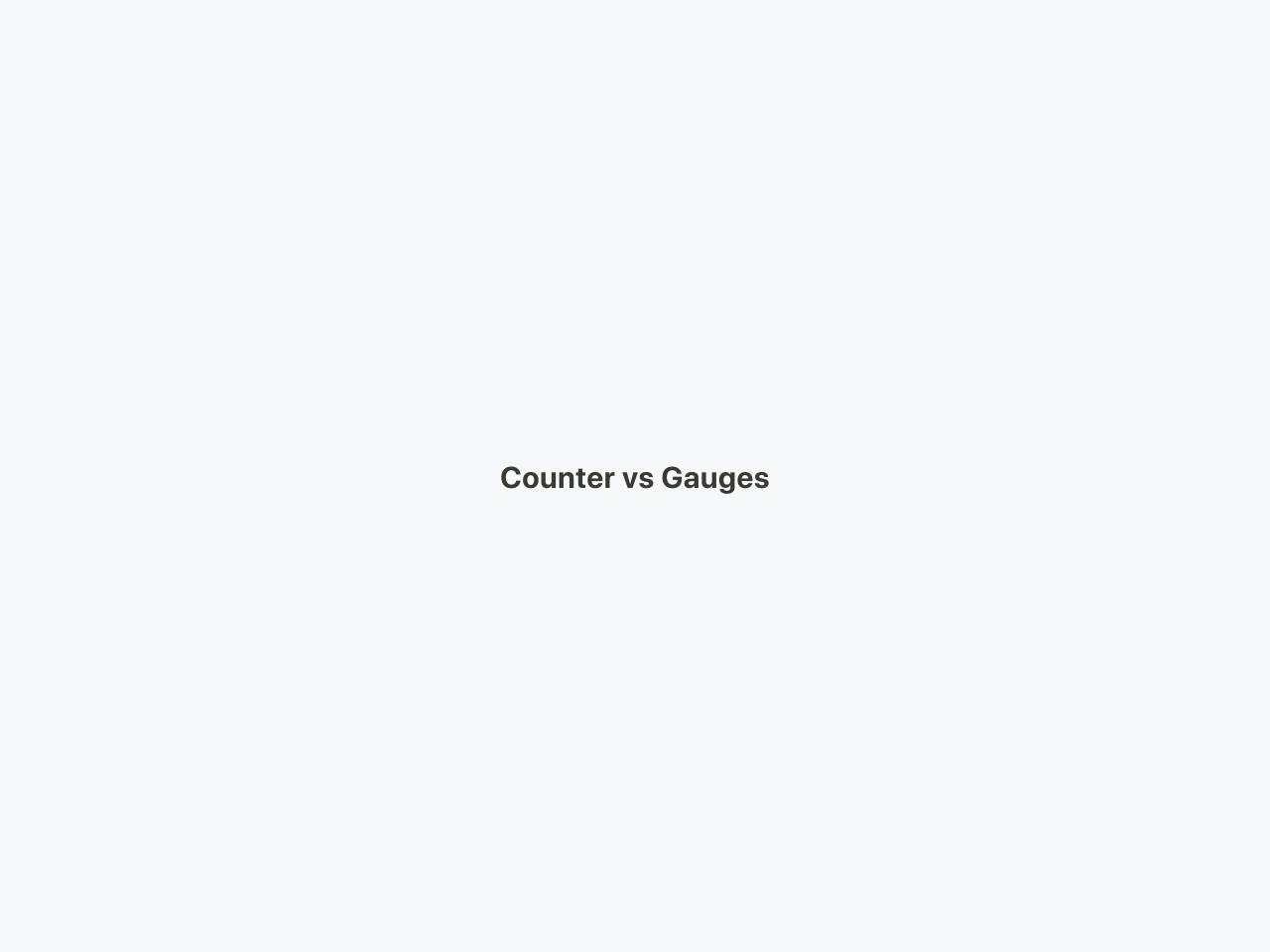




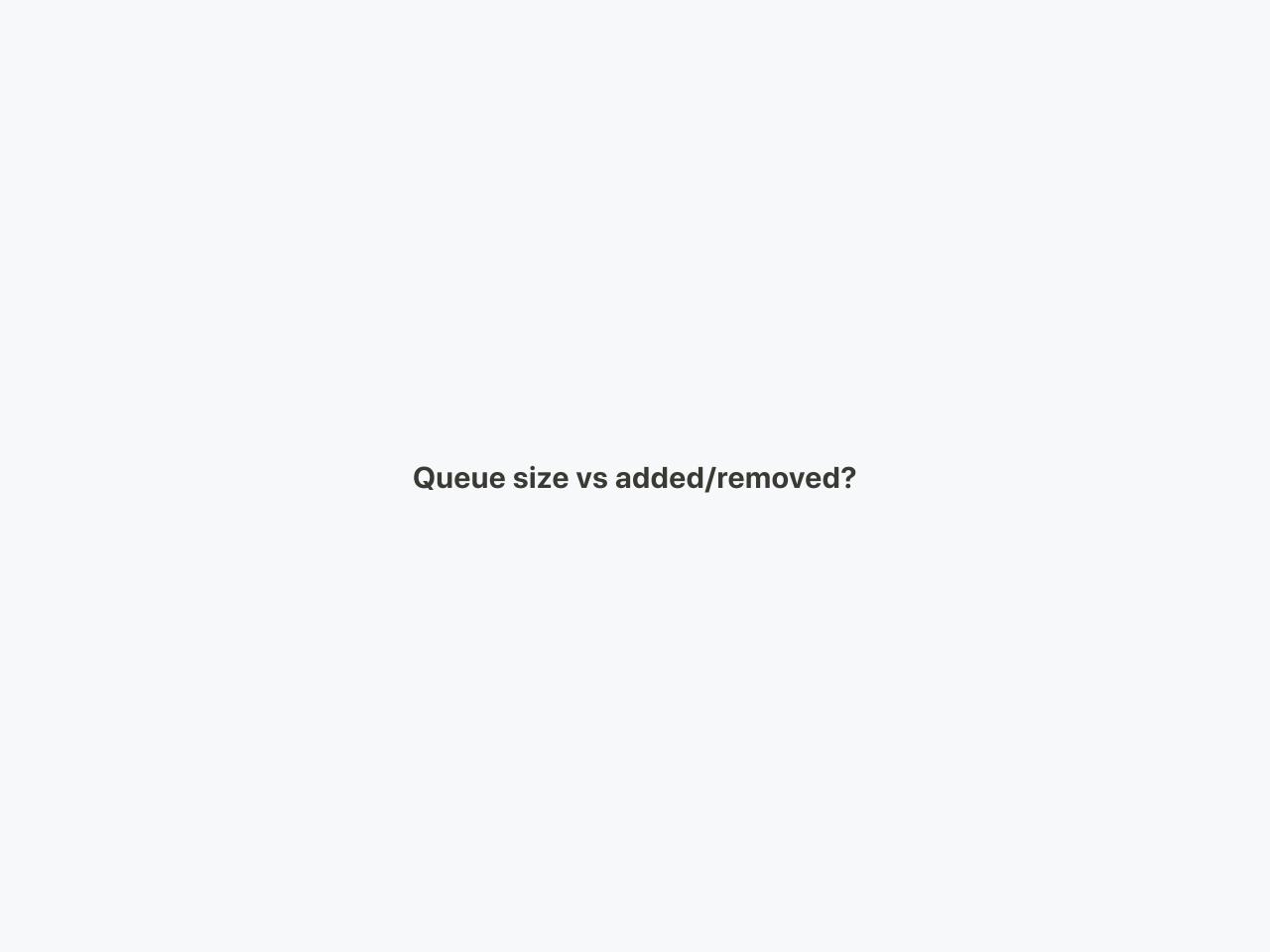
Intro: define metric; register; expose HTTP; usage

**Problems: terrible for libs** 

Solution: custom registered

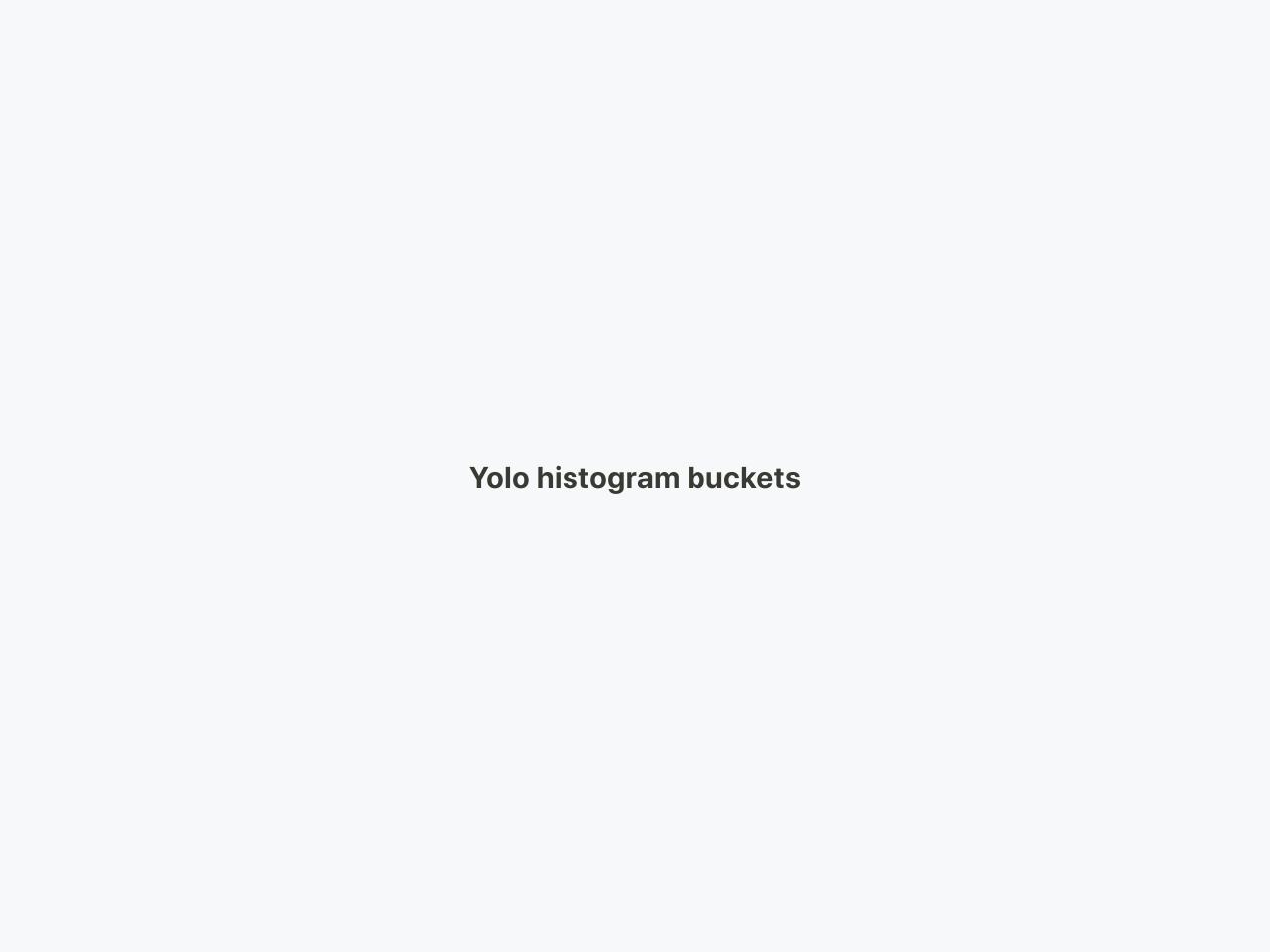


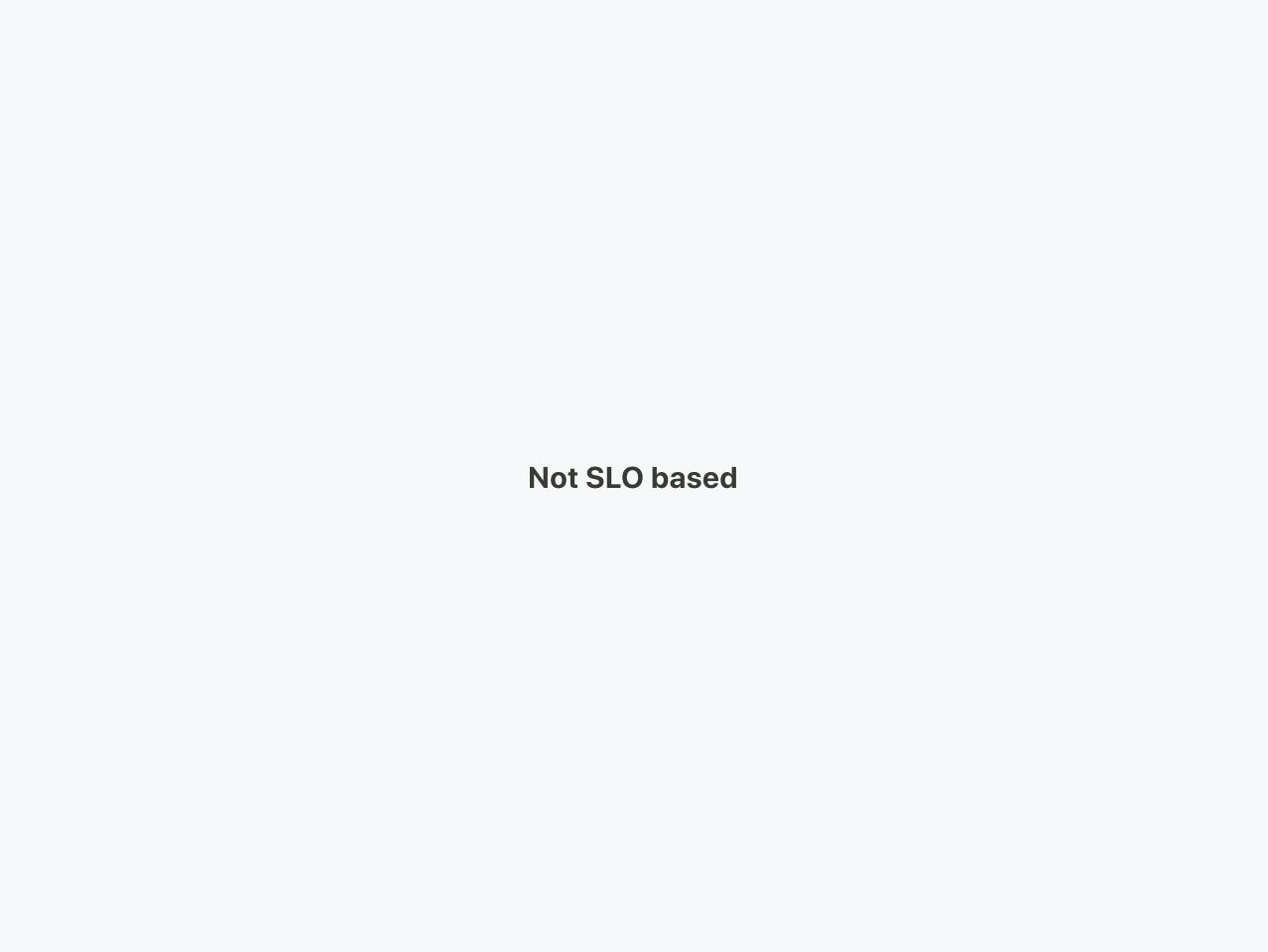


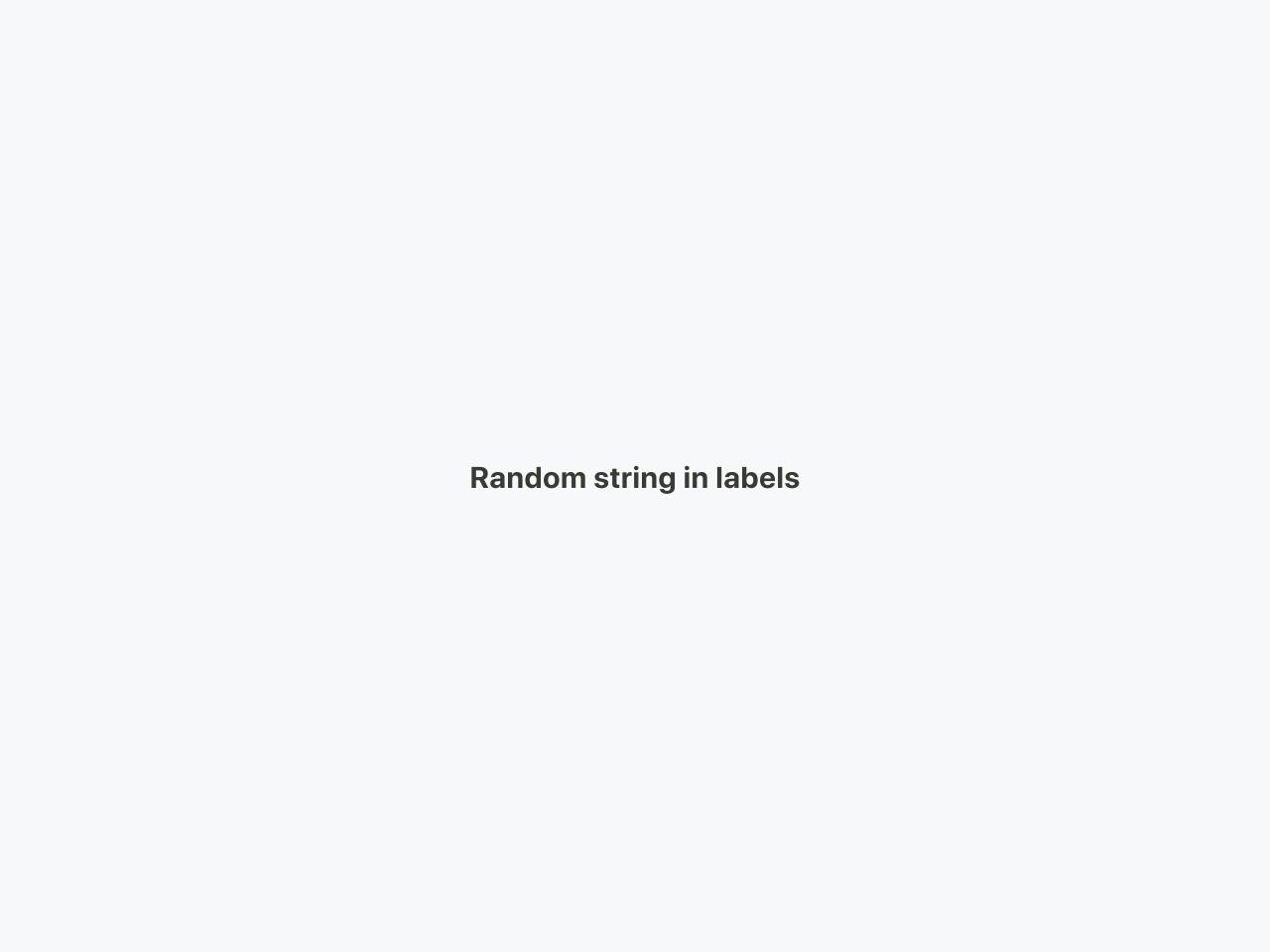


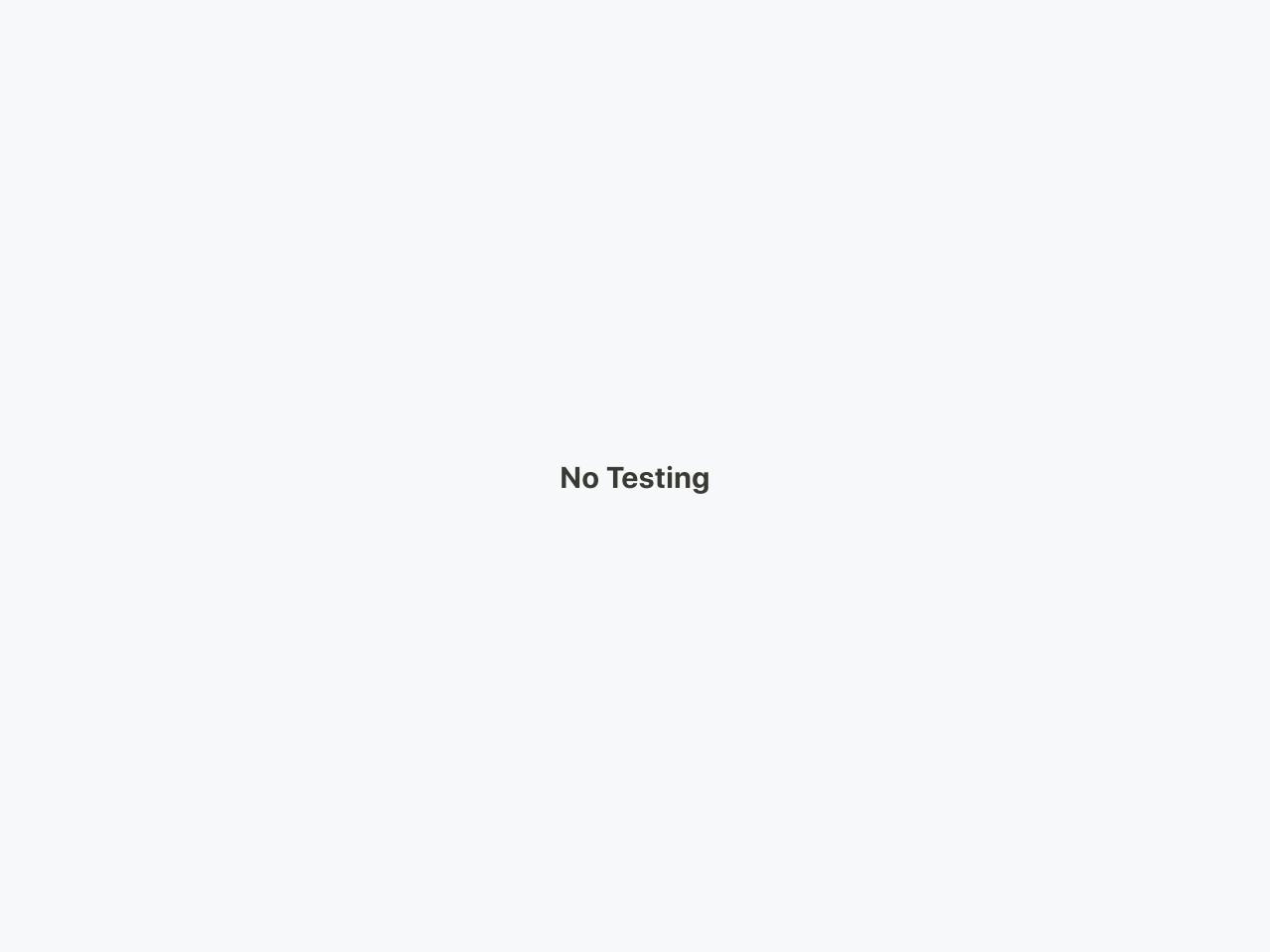
Gauge, reset etc loss of info





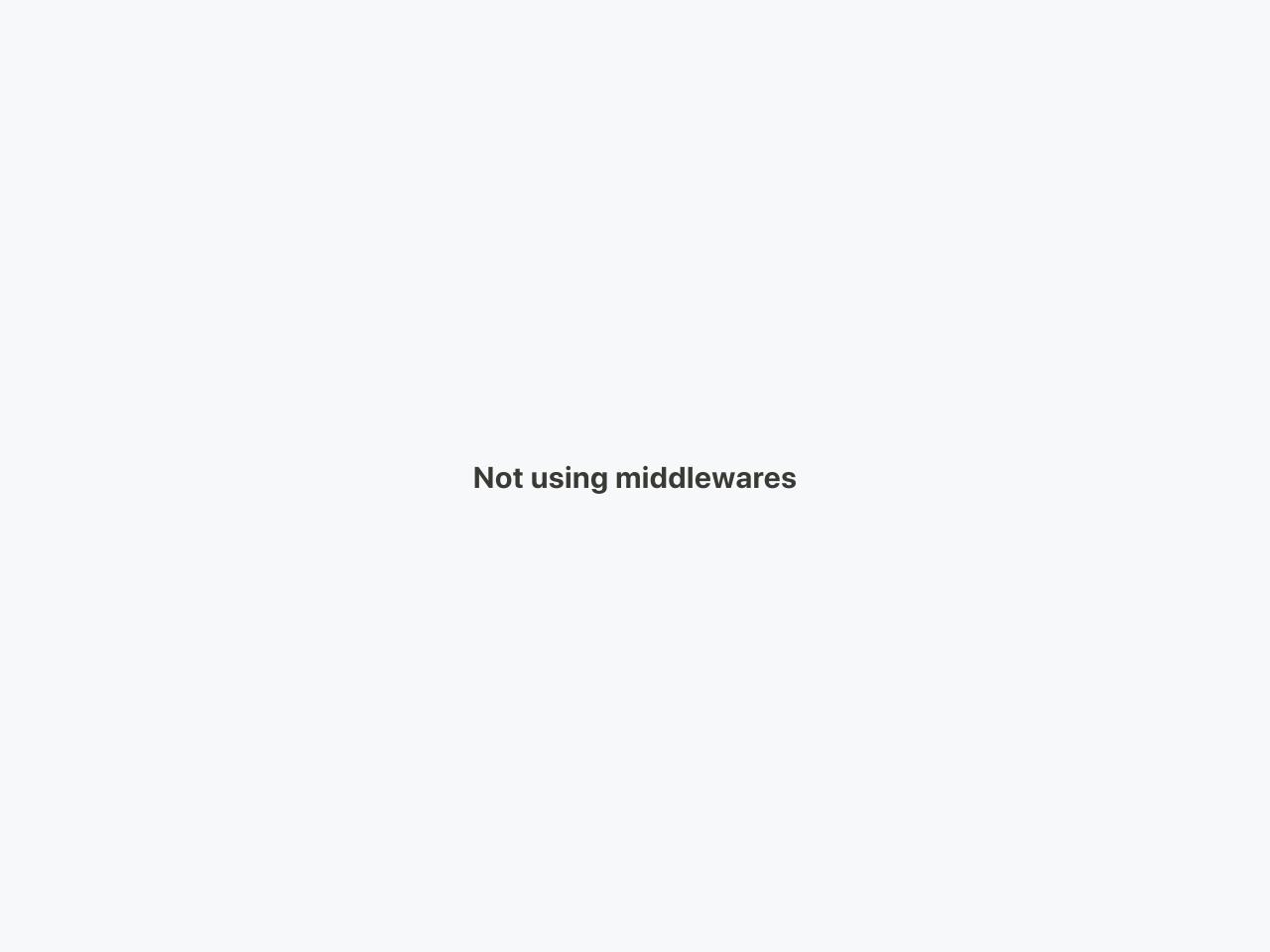


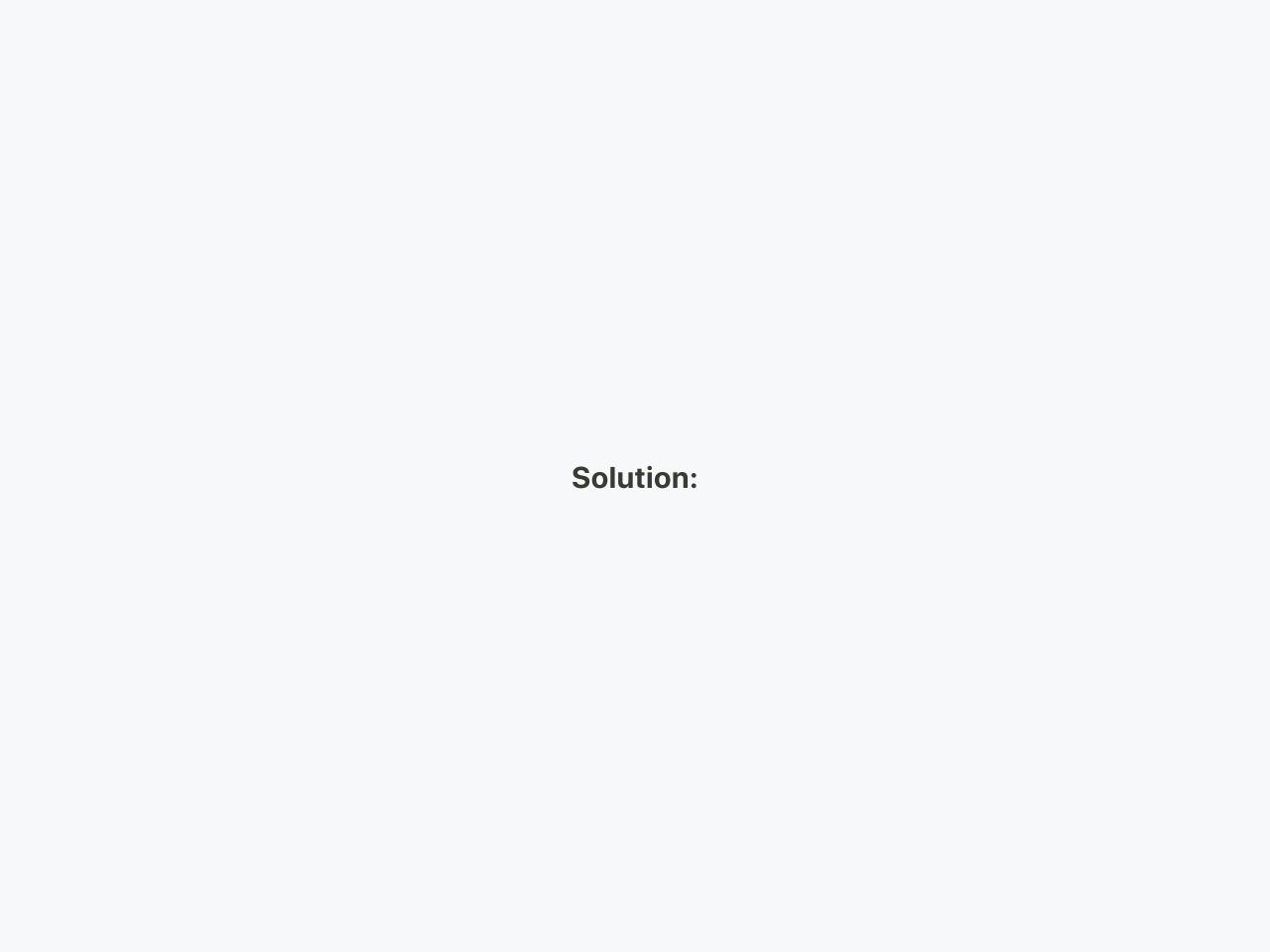


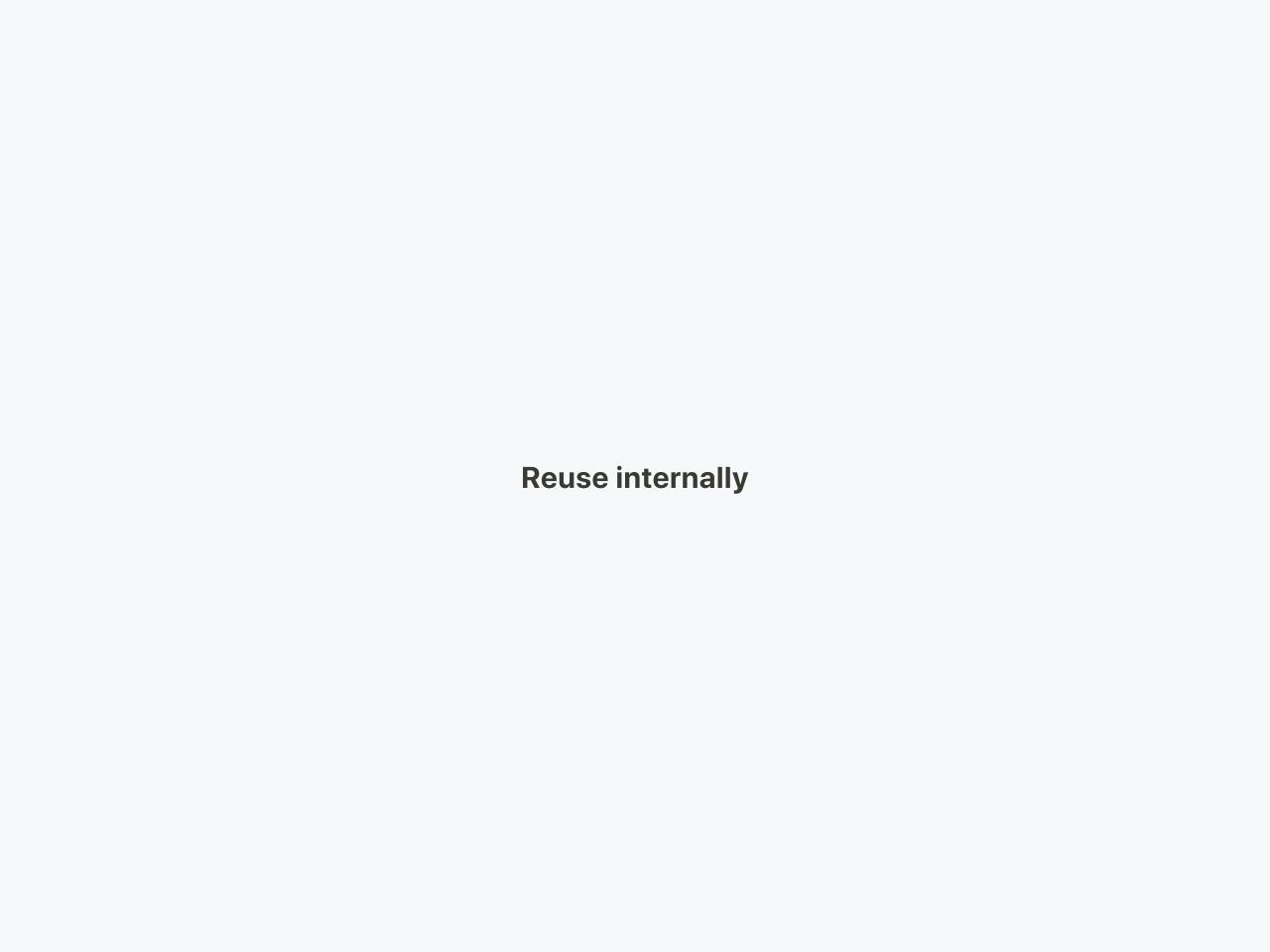


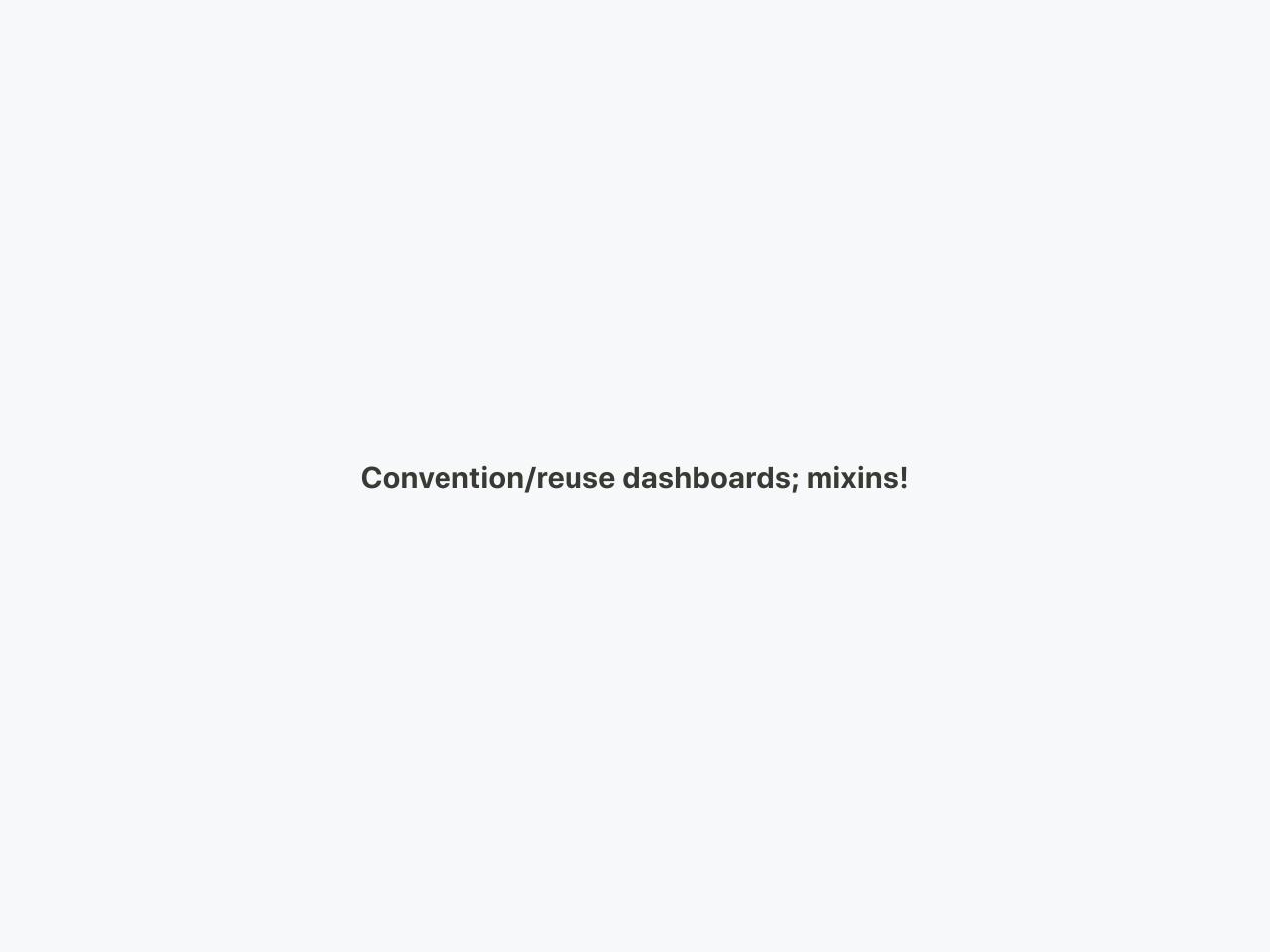
Hardcoding namespace in libraries/package











Q/A

**Discussion** 

Reference:

•