

Using Middleware

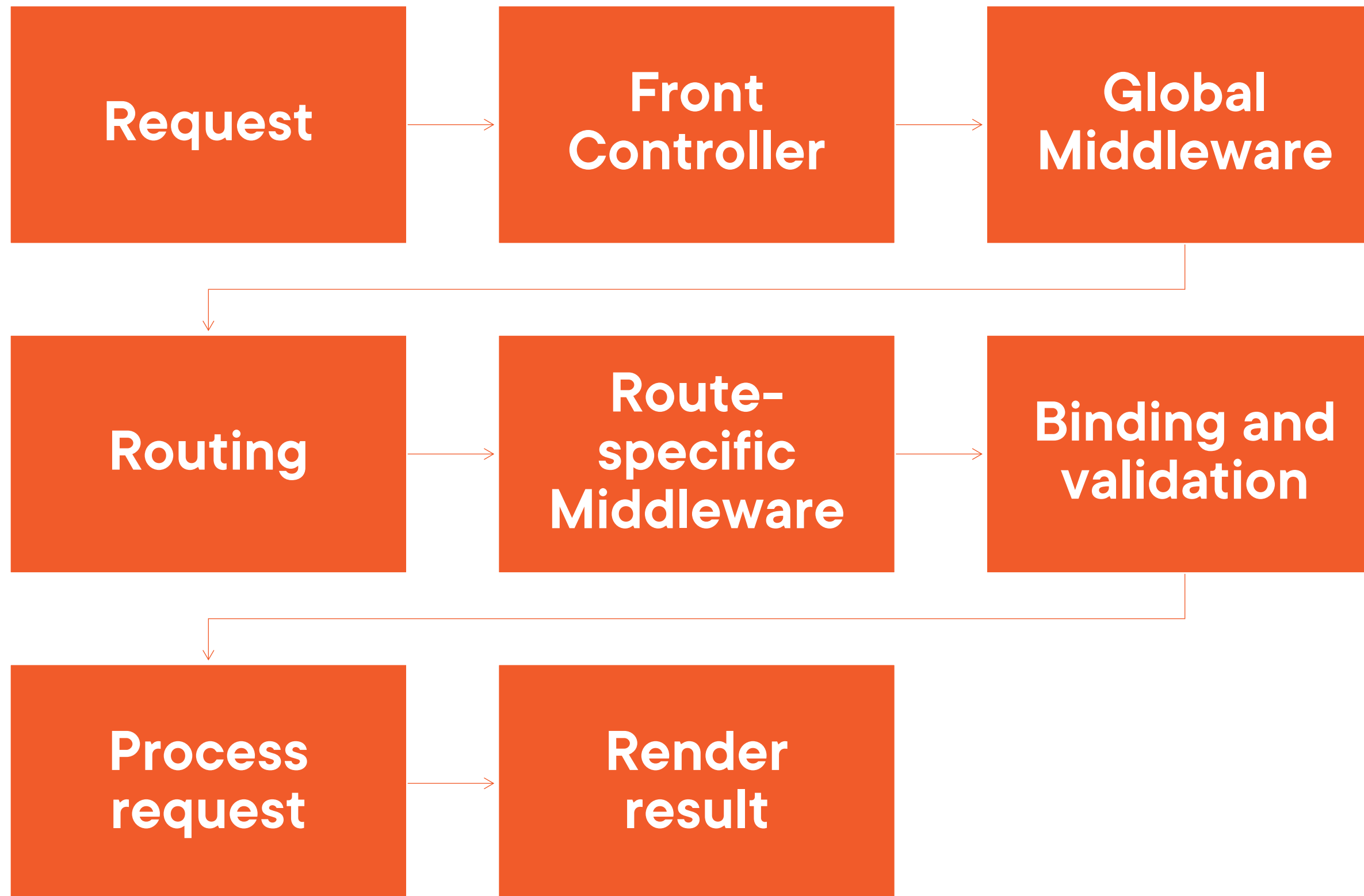


Michael Van Sickle

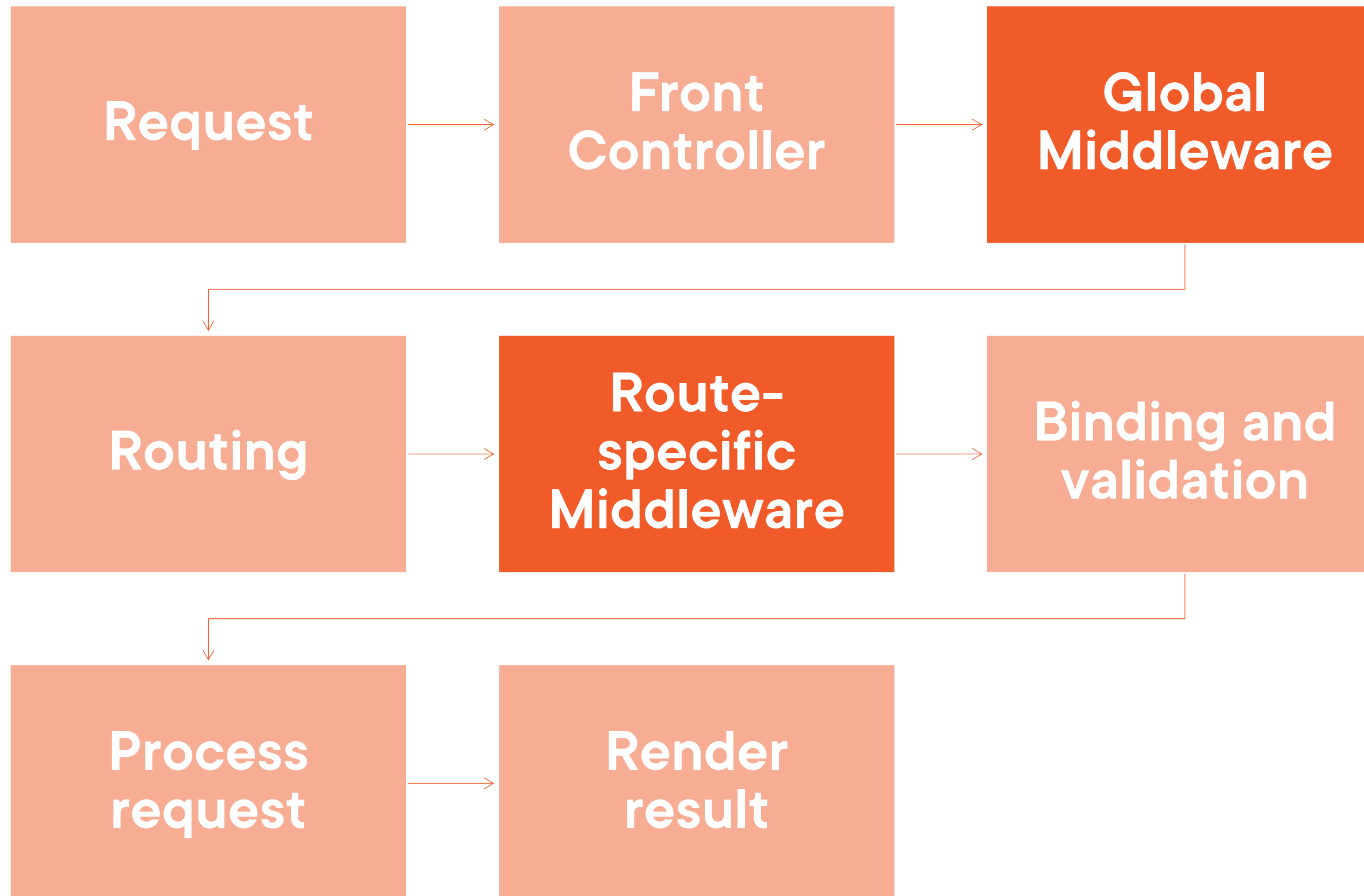
@vansimke



HTTP Request Pipeline



HTTP Request Pipeline



Overview



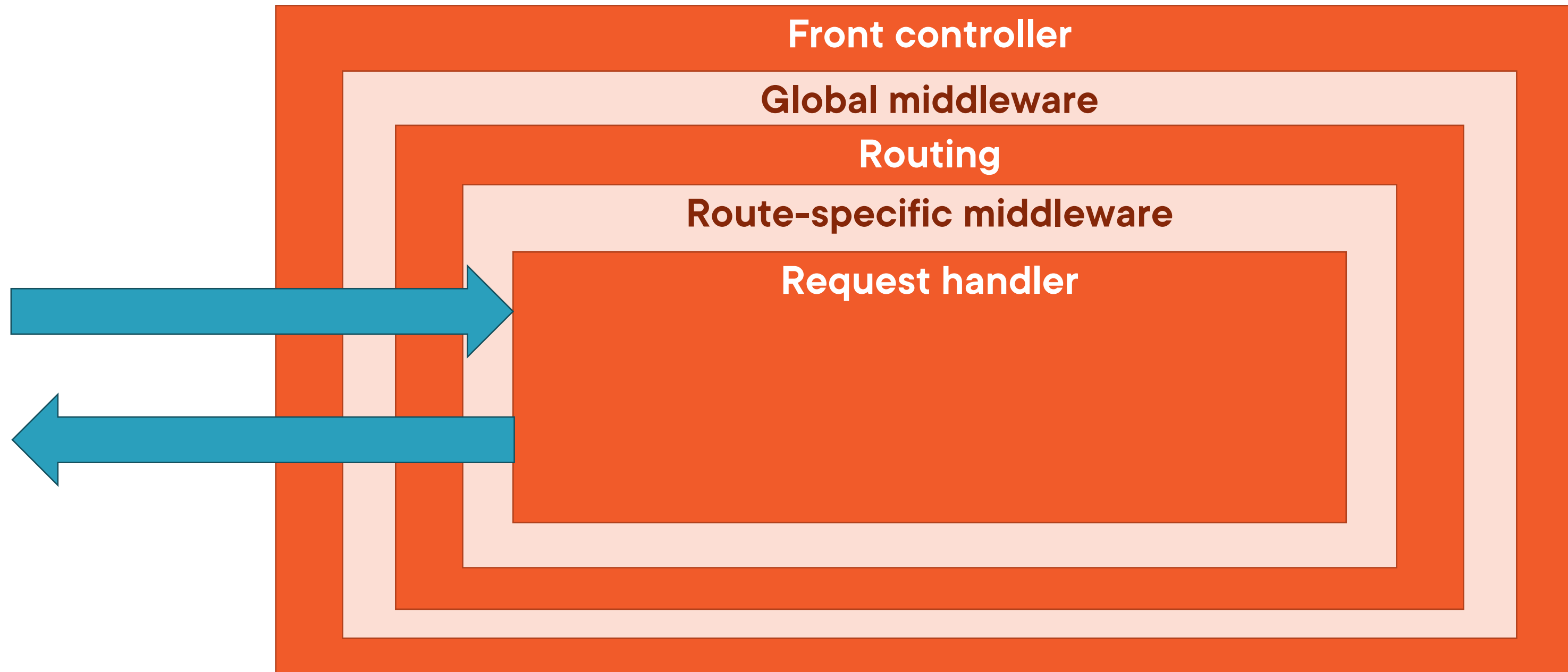
What is middleware?

Using pre-made middleware

Creating custom middleware



What is Middleware?



What is Middleware?

```
type MyMiddleware struct {  
    next http.Handler  
}  
  
func (mm MyMiddleware) ServeHTTP(w http.ResponseWriter, r *http.Request) {  
    // do things before next middleware or handler  
  
    mm.next.ServeHTTP(w, r)  
  
    // do things after next middleware or handler  
}
```



Using Middleware

```
router := gin.Default()

// global middleware
r.Use(middleware.HandlerFunc)

// route-specific middleware
r.GET(path string, handlers.HandlerFunc) // final HandlerFunc is the request
handler, others are middleware
```



Premade Middleware

Built-in middleware

`gin.Logger()`

`gin.Recovery()`

`gin.BasicAuth()`

more at pkg.go.dev/github.com/gin-gonic/gin#HandlerFunc

Other official middleware

`gzip`

`cache`

`cors`

more at github.com/gin-contrib



Custom Middleware

```
func myGinMiddleware(c *gin.Context) {  
    // do things before next middleware or handler  
    c.Next()  
    // do things after next middleware or handler  
}  
  
func main() {  
    router := gin.Default()  
    router.Use(myGinMiddleware)    // register global middleware  
  
    // register route-specific middleware (as many as needed!)  
    router.GET("/interesting/path", myGinMiddleware, func(c *gin.Context) {  
  
    })  
  
    // register middleware for route group  
    group := gin.Group("/secret", myGinMiddleware)  
    group.GET(...)  
}
```



Summary



What is middleware?

Using pre-made middleware

Creating custom middleware

