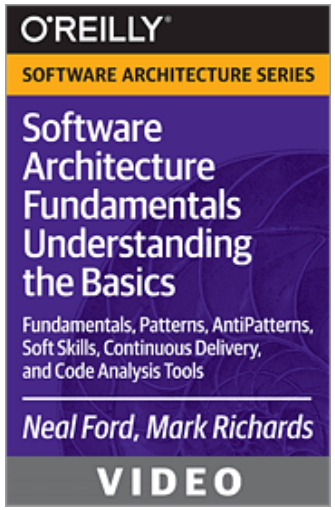
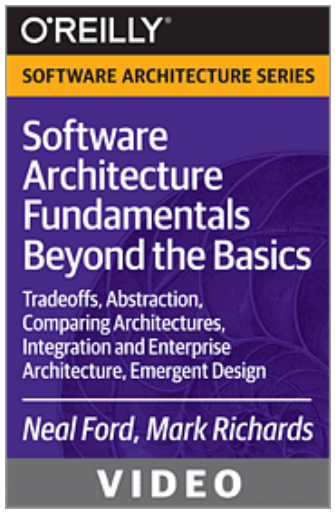
Service-Based Architecture
Microservices Hybrids

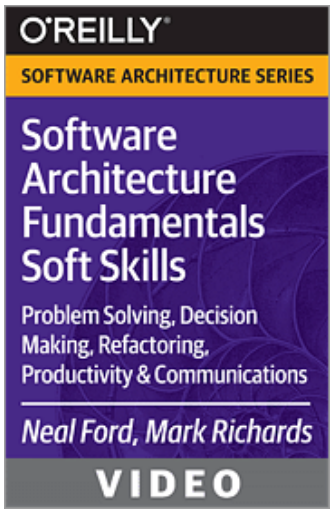
Mark Richards
Independent Consultant
Hands-on Software Architect
Published Author / Conference Speaker
www.wmrichards.com

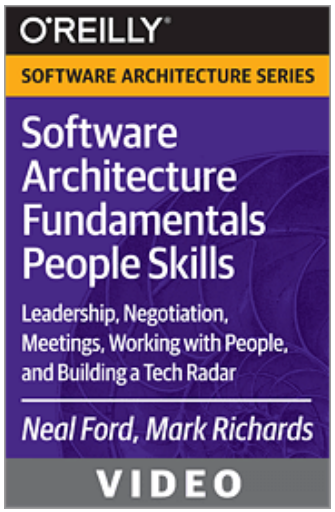
Author of *Software Architecture Fundamentals Video Series* (O'Reilly)
Author of Microservices Pitfalls and AntiPatterns (O'Reilly)
Author of Microservices vs. Service-Oriented Architecture (O'Reilly)
Author of *Enterprise Messaging Video Series* (O'Reilly)
Author of *Java Message Service 2nd Edition* (O'Reilly)

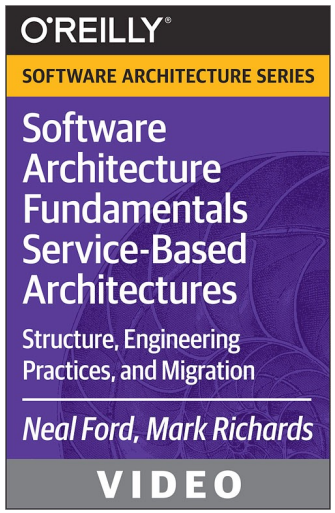
Software Architecture Fundamentals Video Series
Enterprise Messaging Video Series

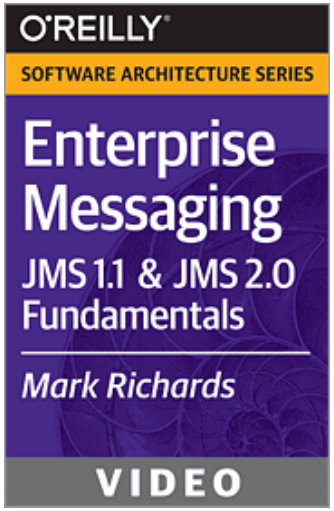


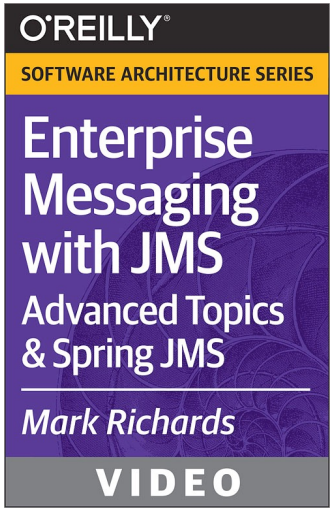




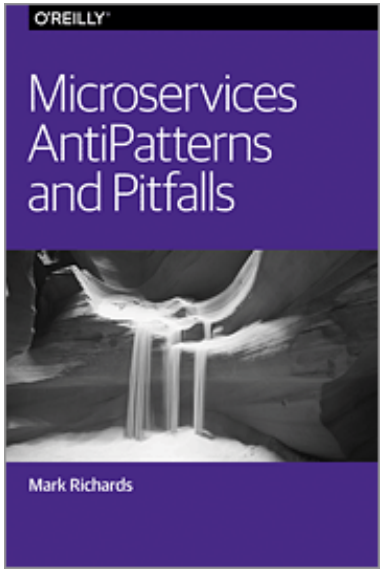


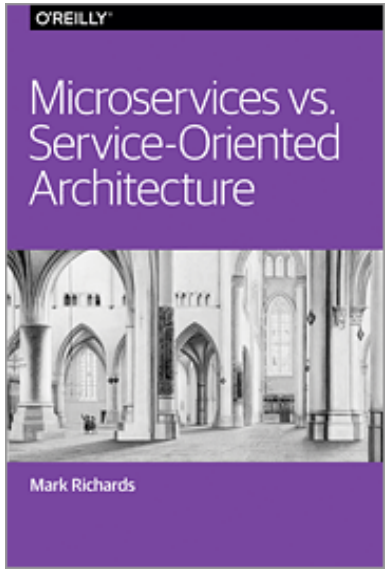






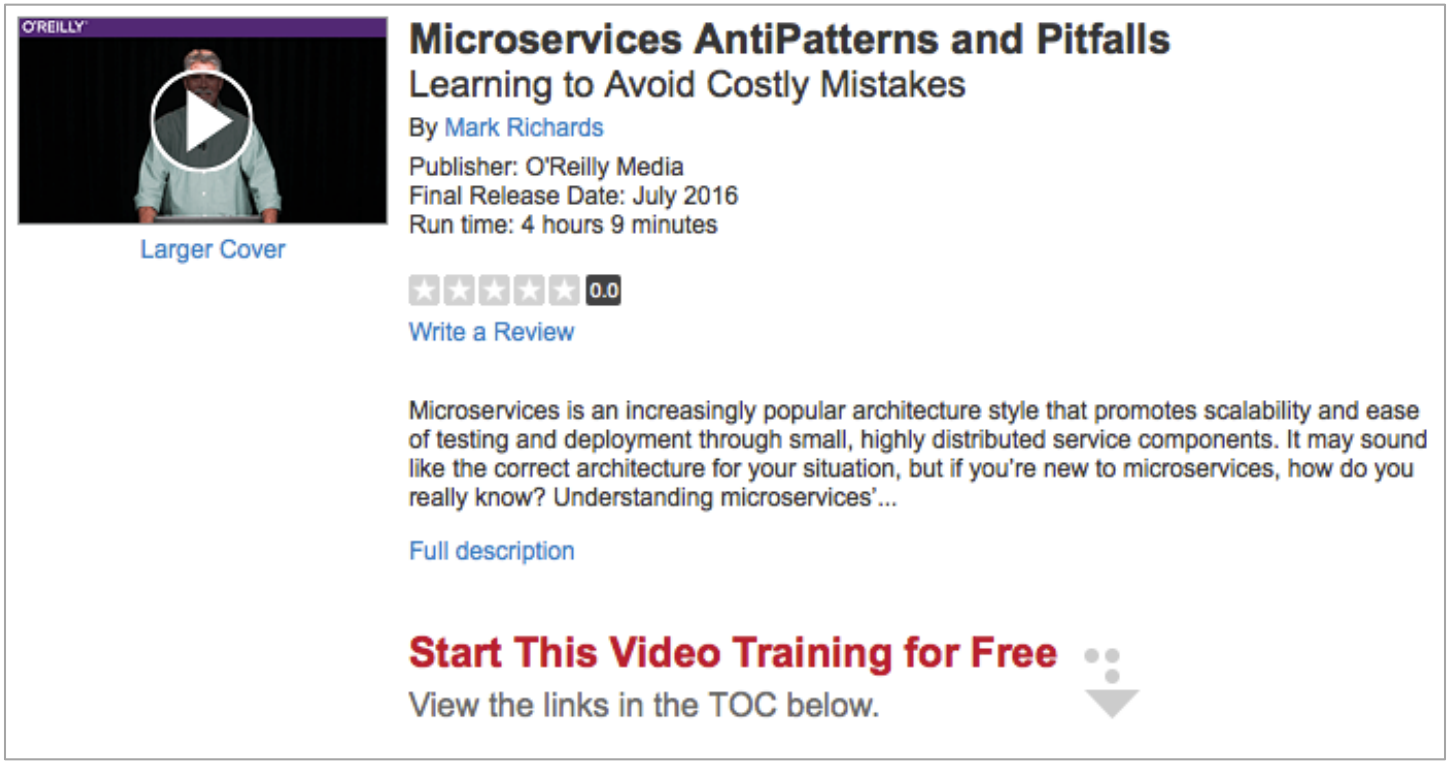
O'Reilly Free Reports





http://www.wmrichards.com/publications

Microservices AntiPatterns and Pitfalls Video



http://shop.oreilly.com/product/0636920052876.do#

Software Architecture Fundamentals Training

https://nofluffjuststuff.com//n/training/schedule

Training Event Schedule

	Architecture Training with Mark Richards	Columbus, OH	September 19 - 21, 2016
	Architecture Training with Mark Richards	Dallas, TX	October 24 - 26, 2016
	Architecture Training with Mark Richards	Boston, MA	November 14 - 16, 2016
	Architecture Training with Mark Richards	Atlanta, GA	December 5 - 7, 2016

agenda

microservices recap

service-based architecture

case studies

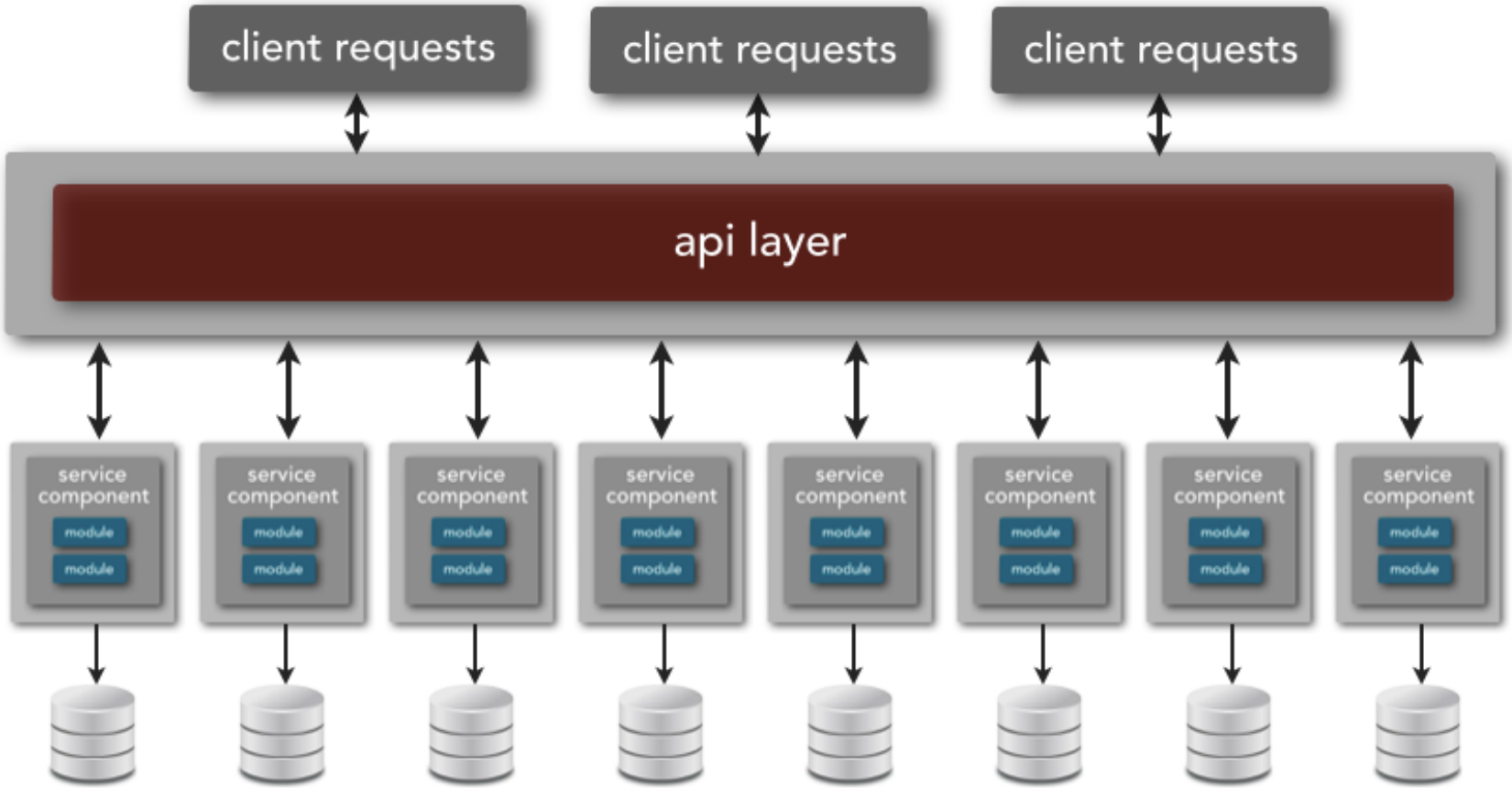
architectural modularity

service design considerations

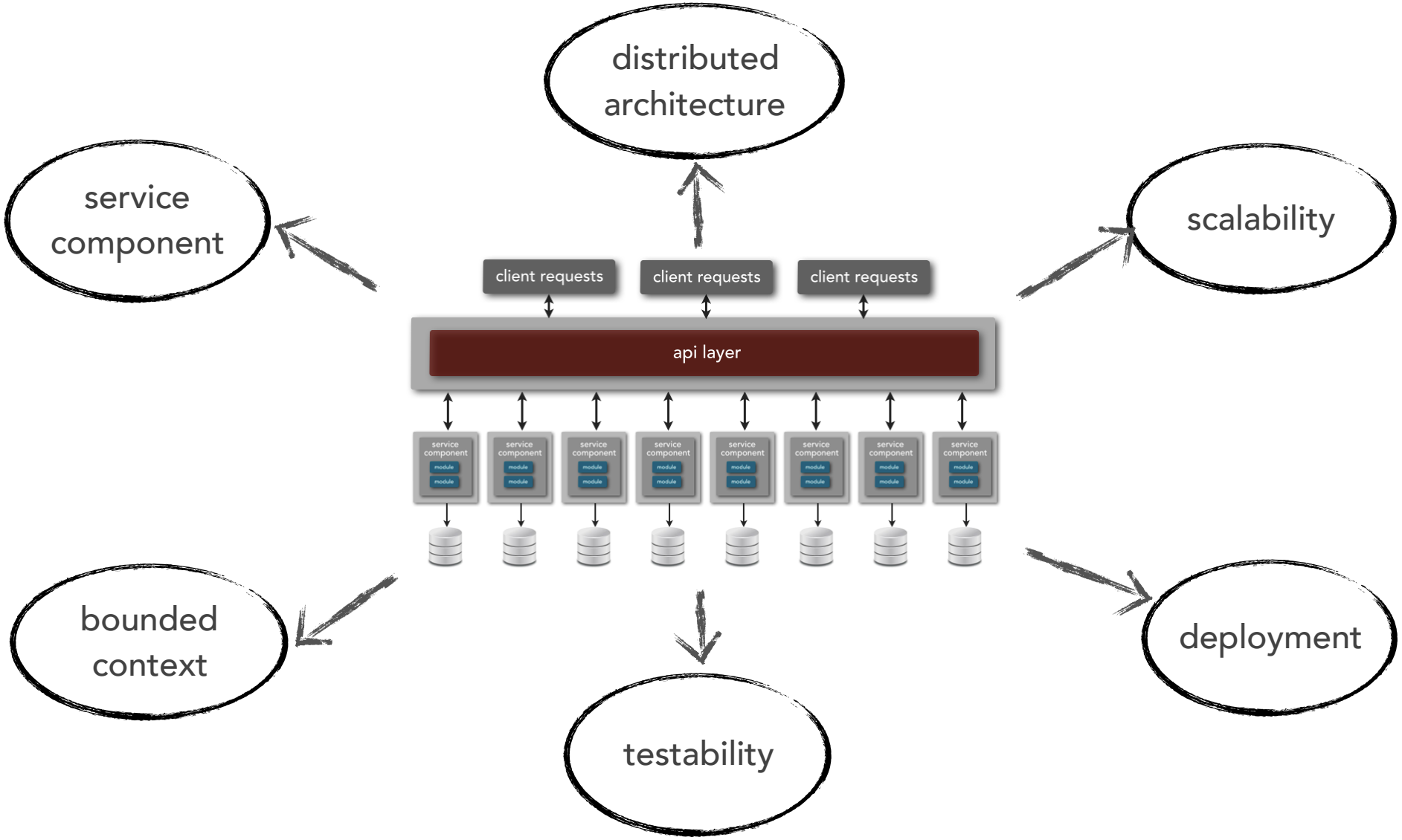
service-based vs. service-oriented architecture

Microservices Overview

microservices architecture

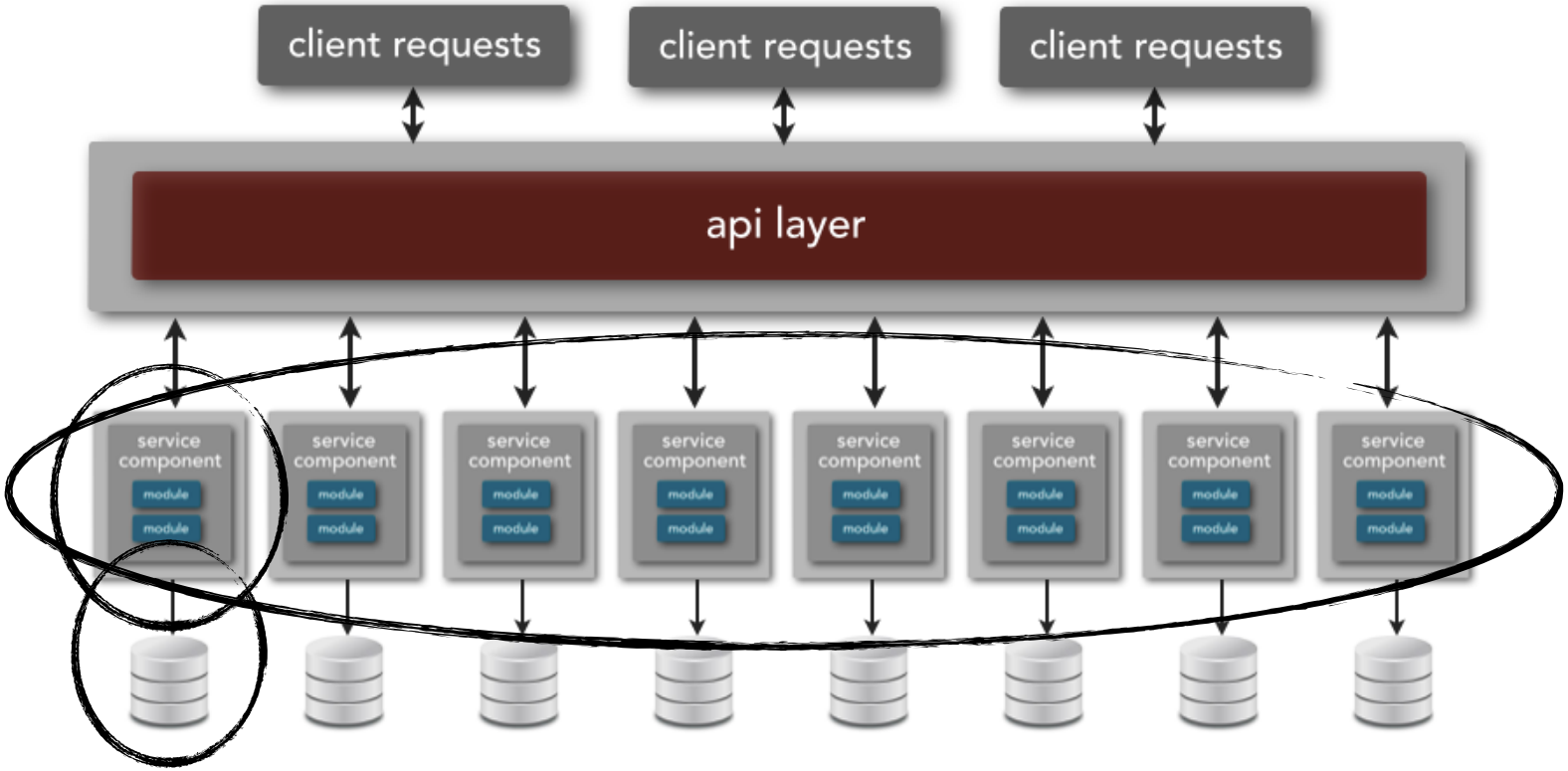


microservices architecture



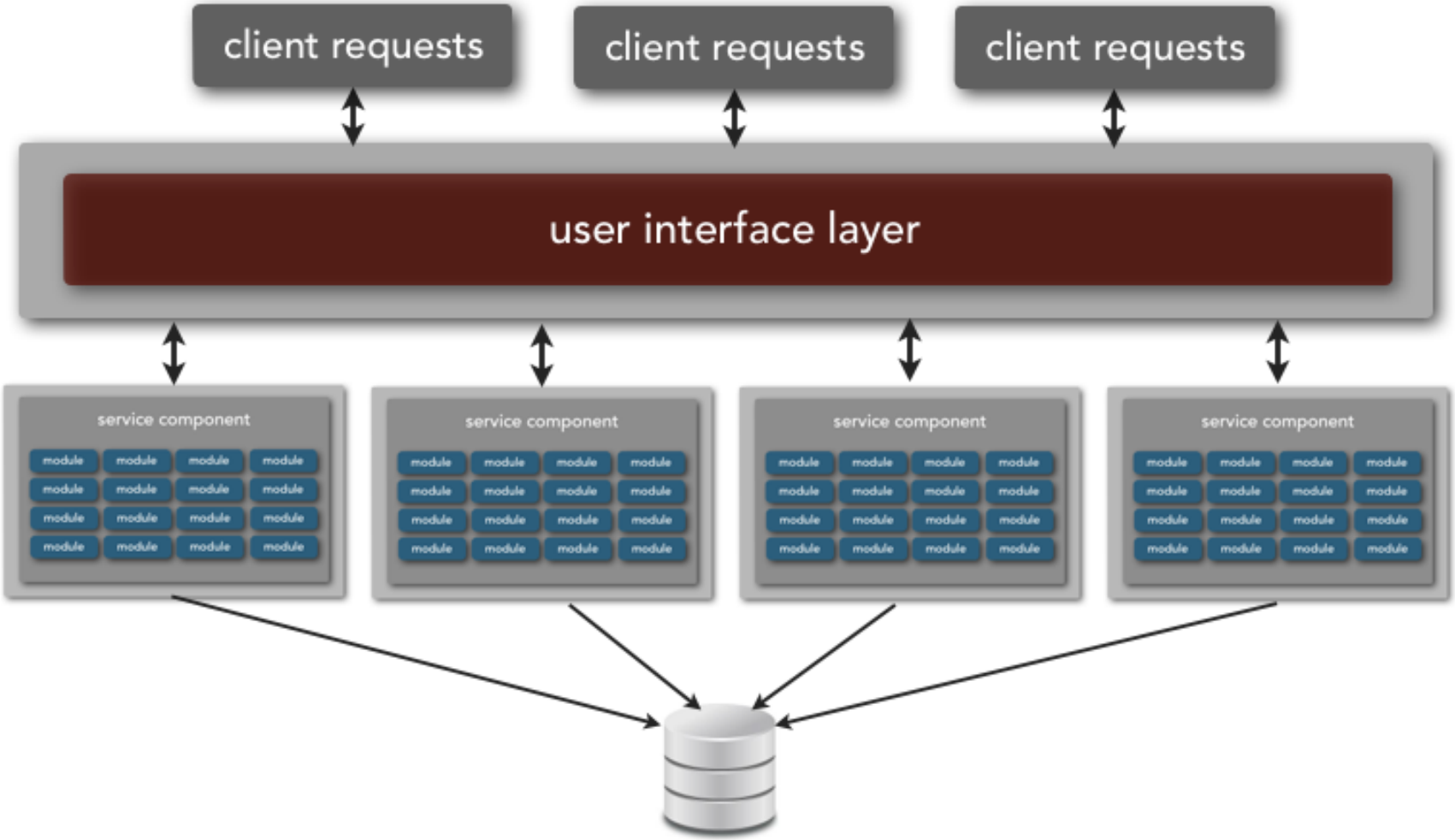
microservices architecture

business applications?

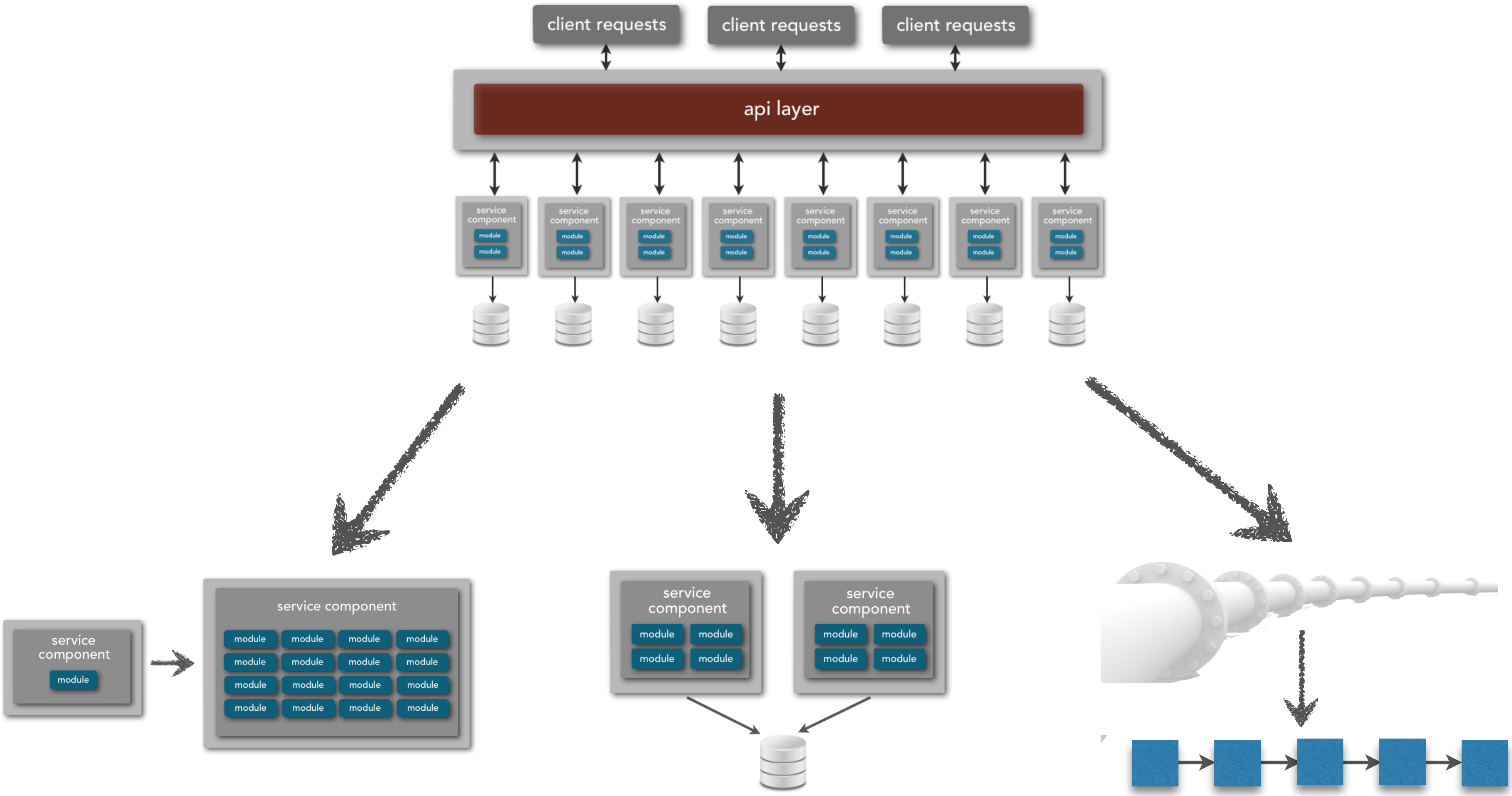


Service-Based Architecture

service-based architecture



service-based architecture



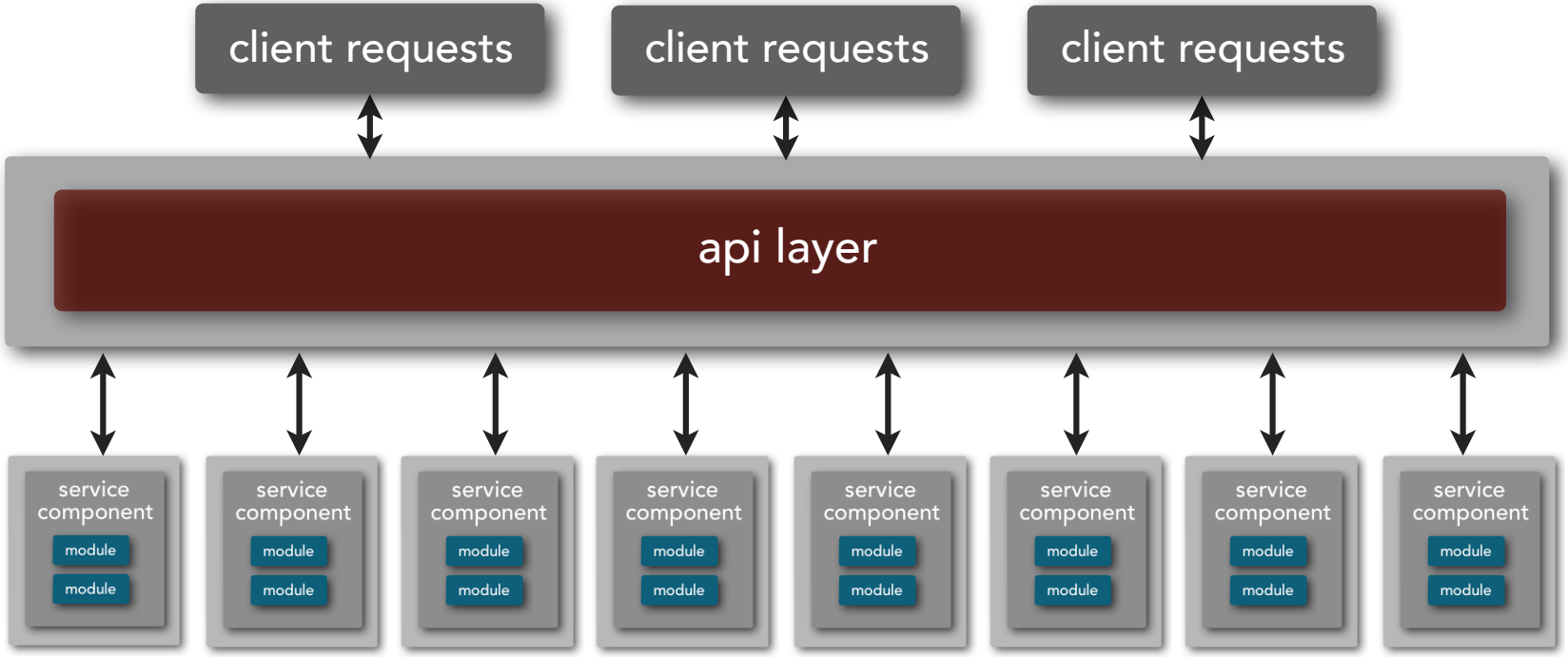
service granularity

database scope

deployment pipeline

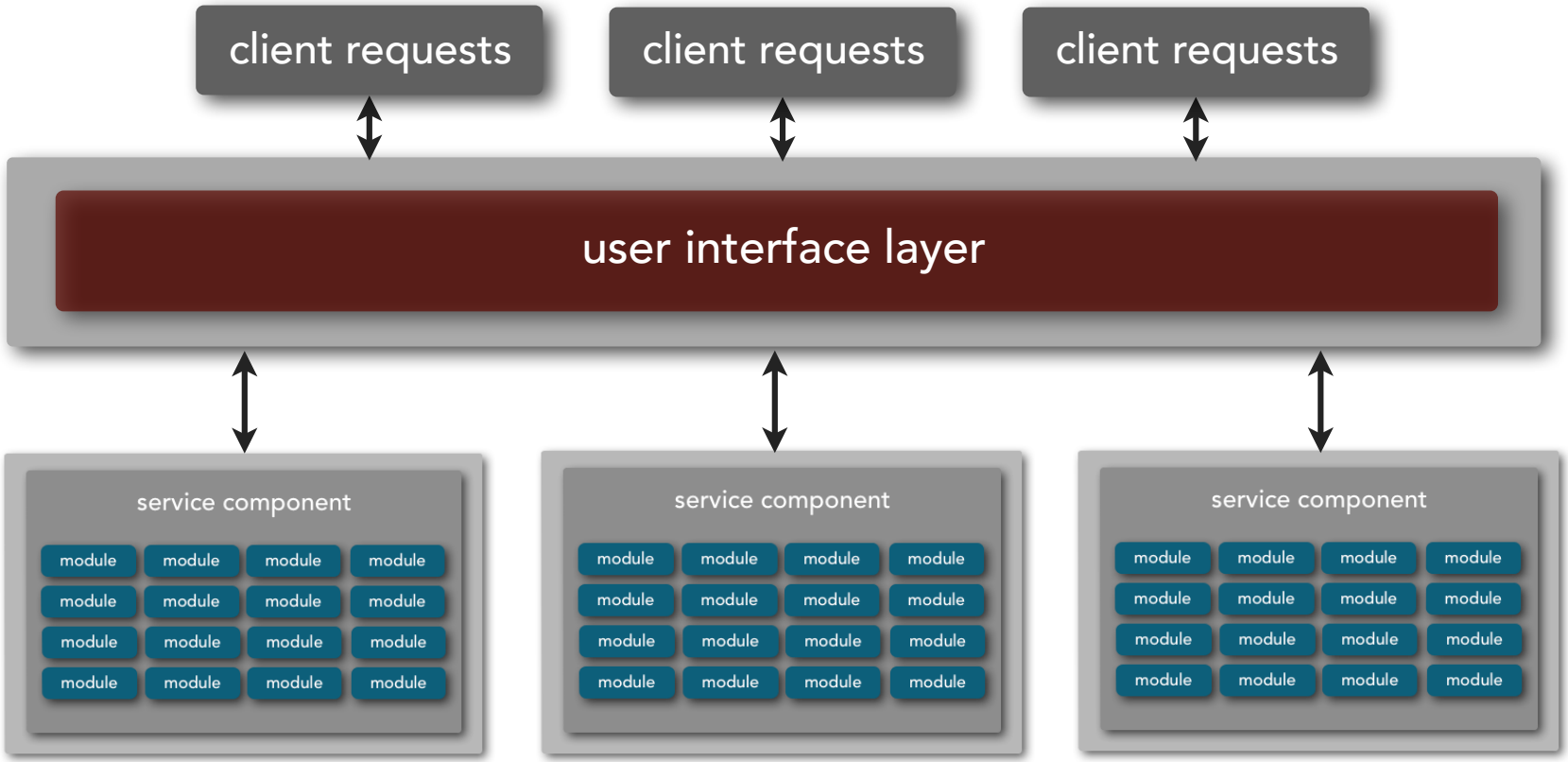
service-based architecture

service granularity



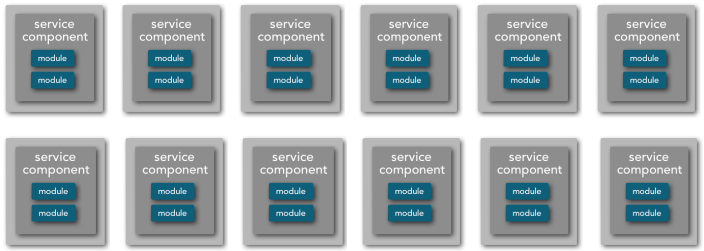
service-based architecture

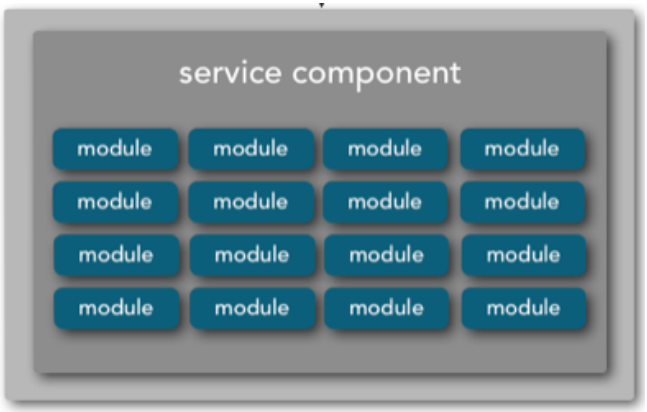
service granularity



service-based architecture

service granularity



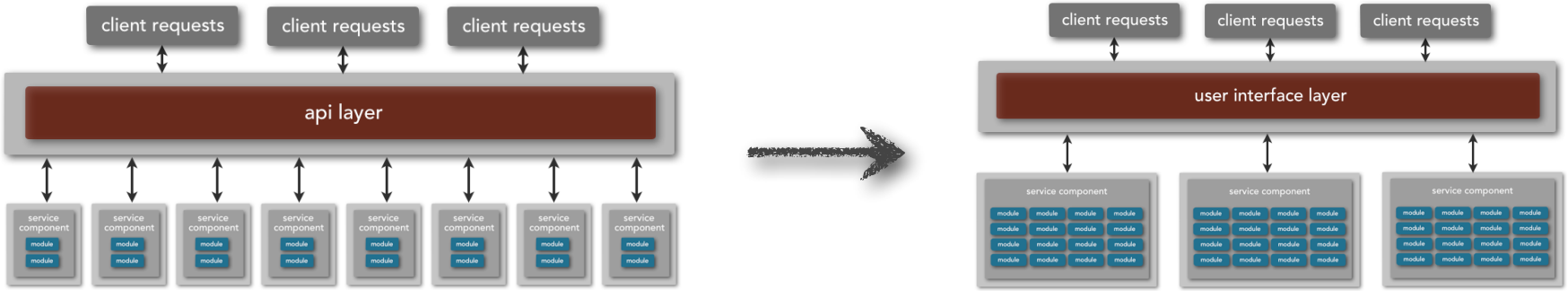


food stamp service
emergency cash service
utility assistance service
child care assist service
health care assist service
nursing facility care service
...

benefit service

service-based architecture

service granularity

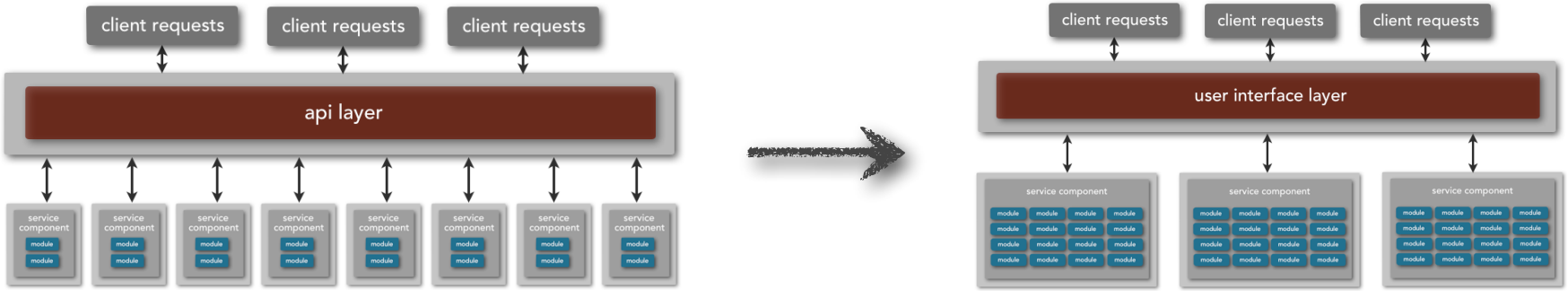


advantages

- unit of work transactional context
- performance and robustness
- domain scope
- shared resources

service-based architecture

service granularity

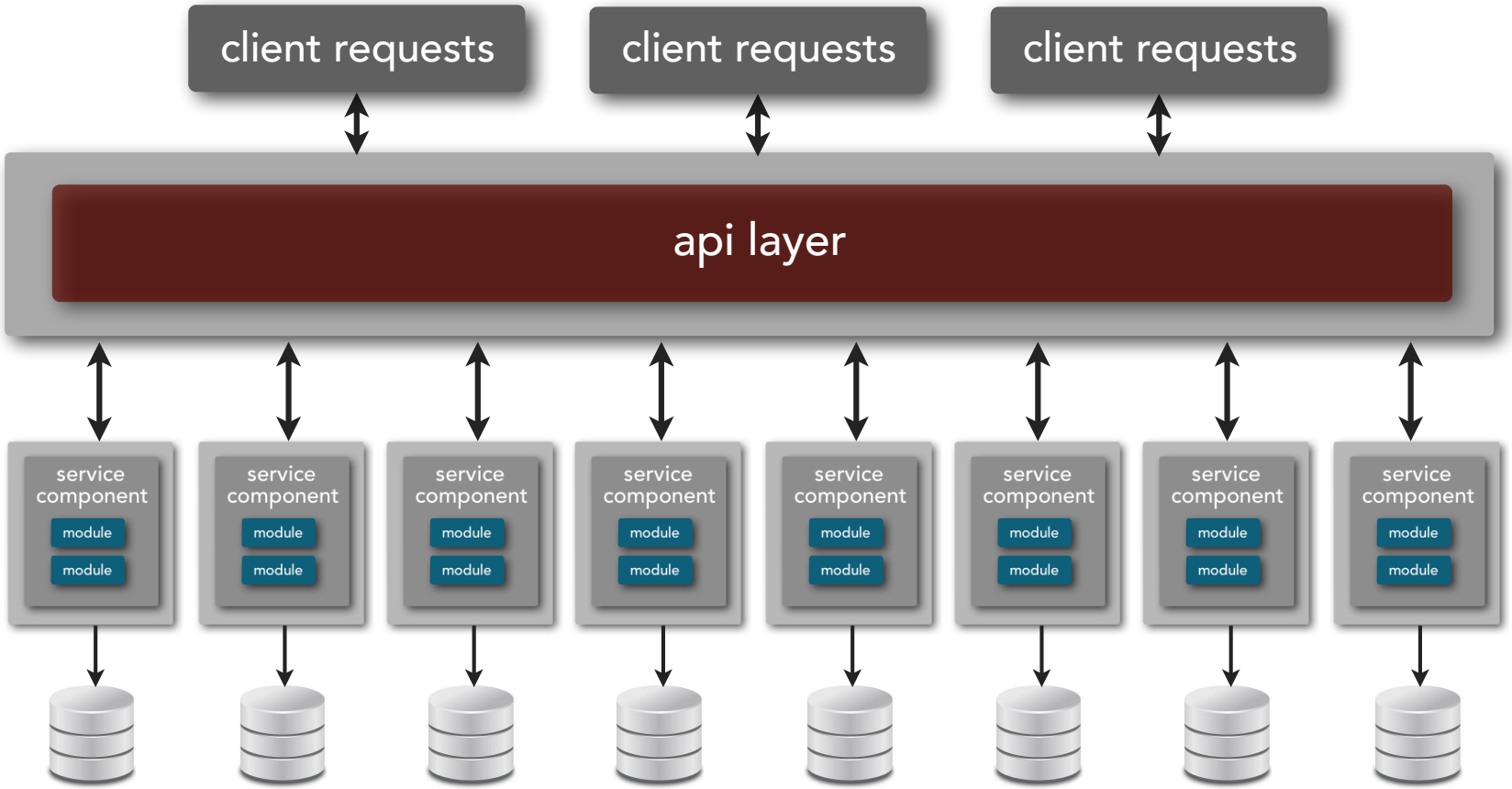


tradeoffs

- services development and testing
- deployment pipeline planning
- change control

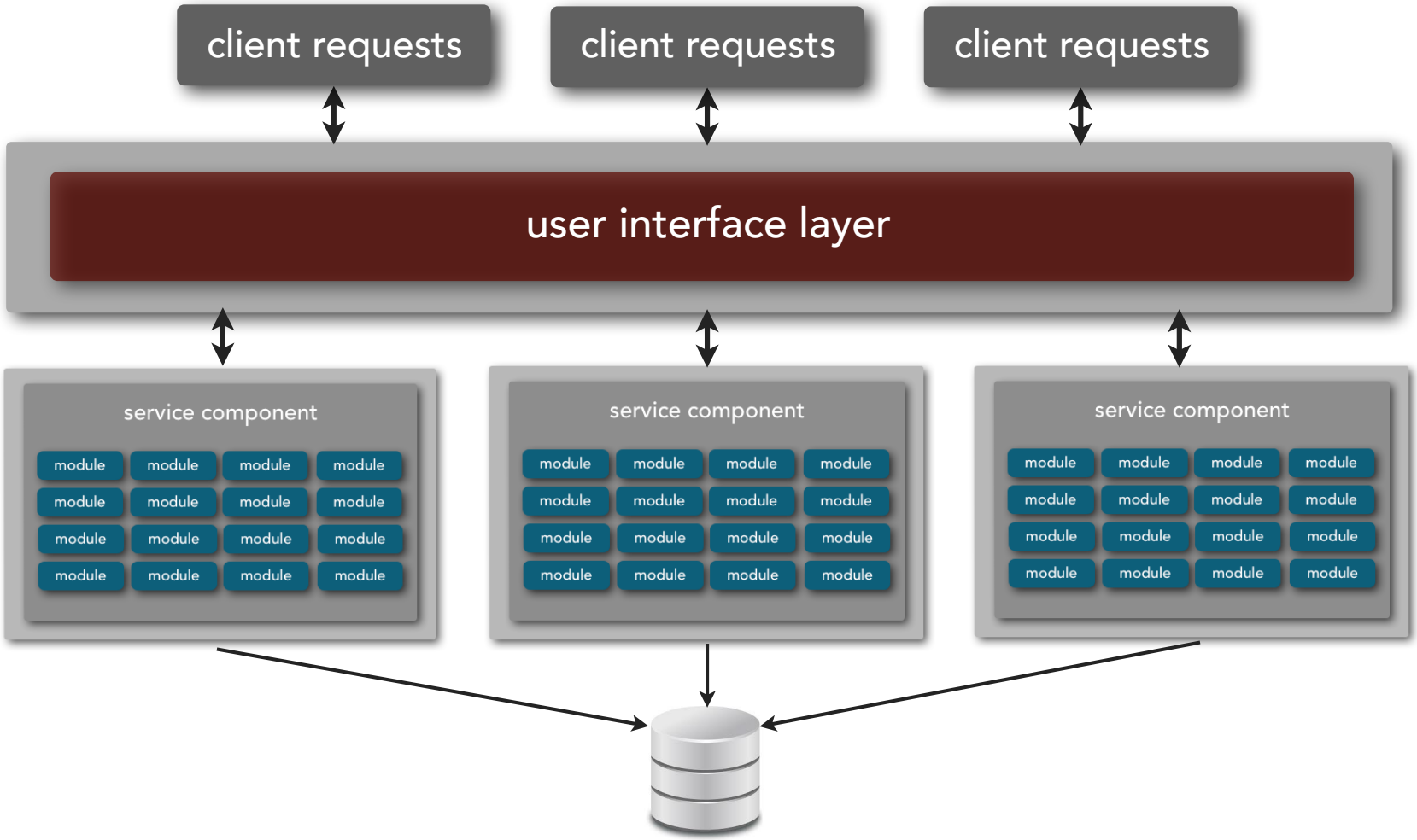
service-based architecture

database scope



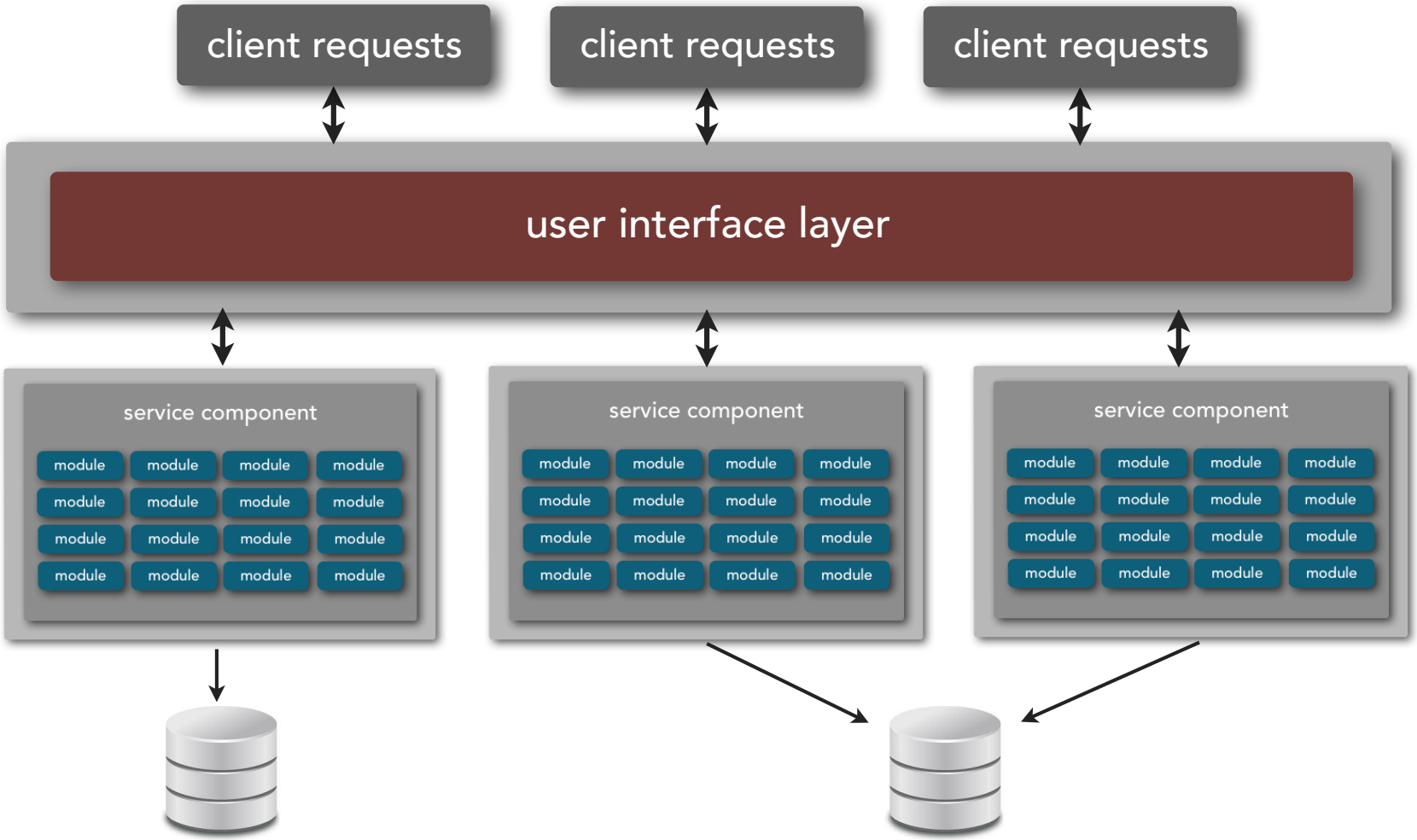
service-based architecture

database scope



service-based architecture

database scope



service-based architecture

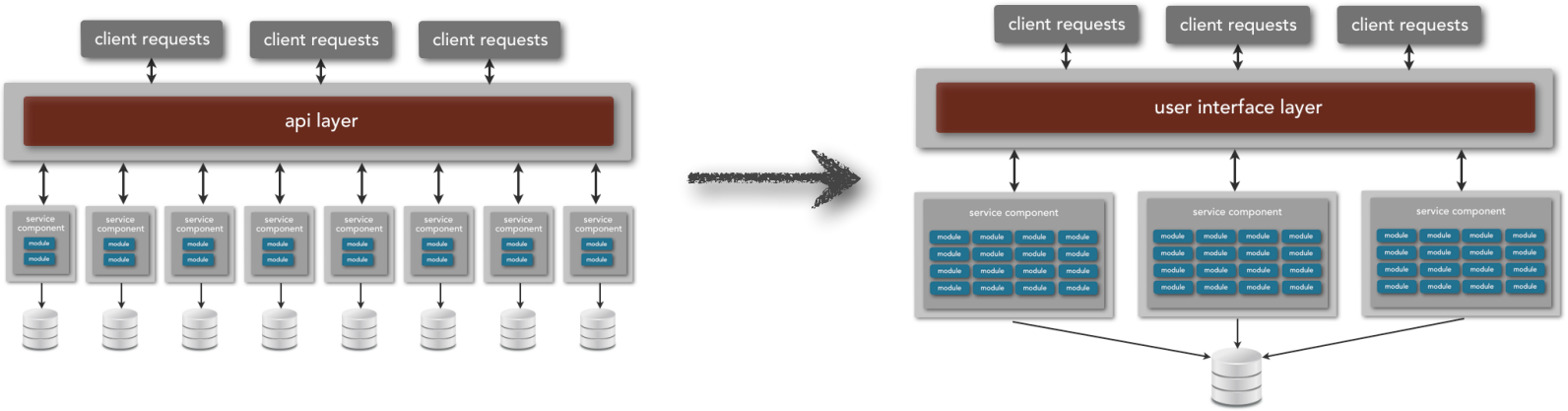
database scope

food stamp db
emergency cash db
utility assistance db
child care assist db
health care assist db
nursing facility care db
...

shared common db

service-based architecture

database scope

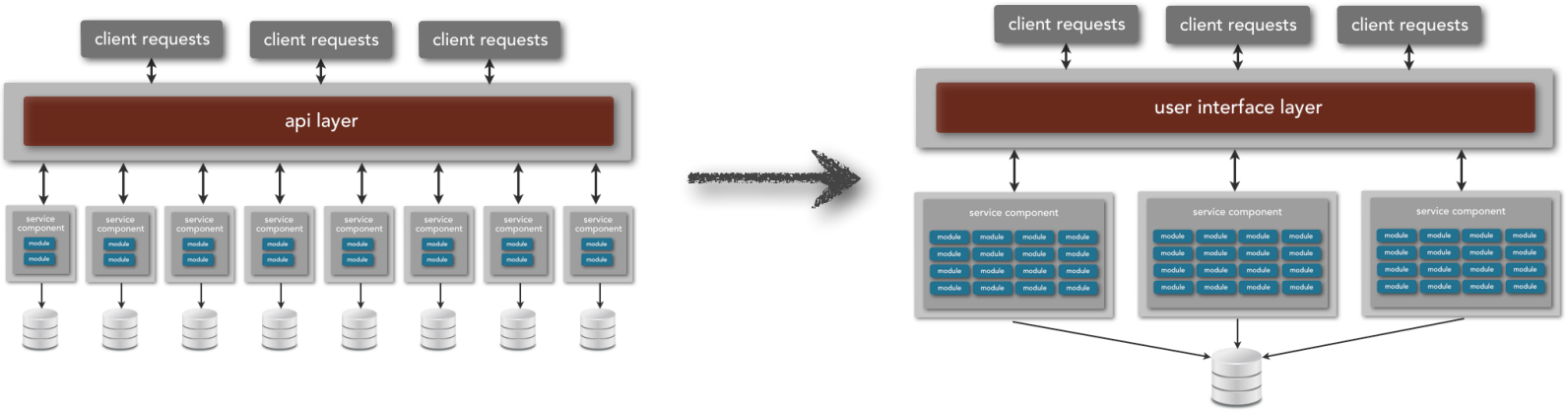


advantages

- performance (joins, orchestration, choreography)
- feasibility

service-based architecture

database scope

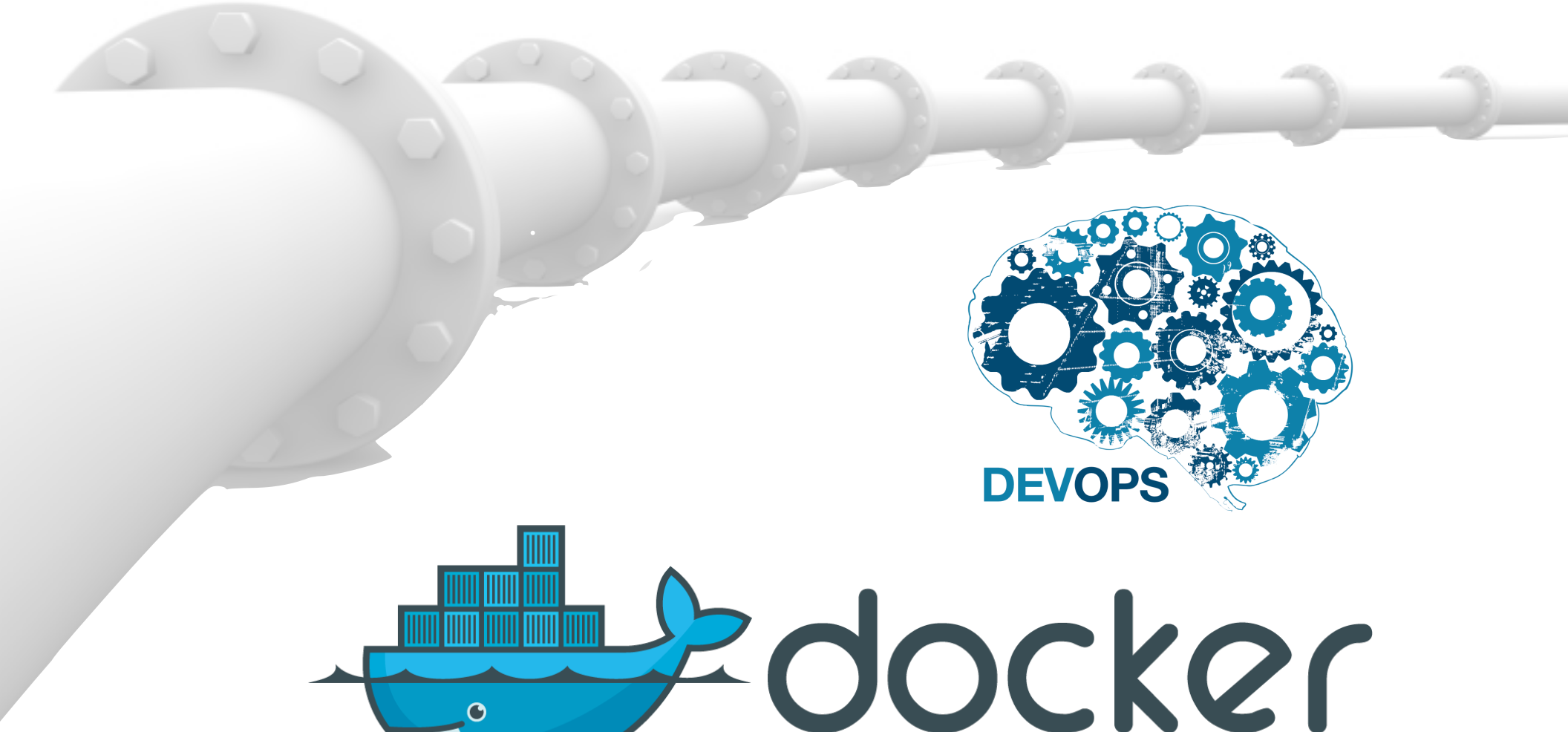


tradeoffs

- bounded context
- service coupling based on schema
- schema changes

service-based architecture

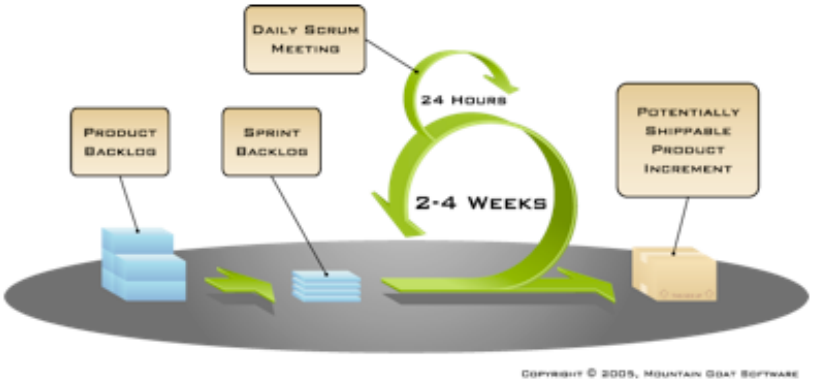
deployment pipeline

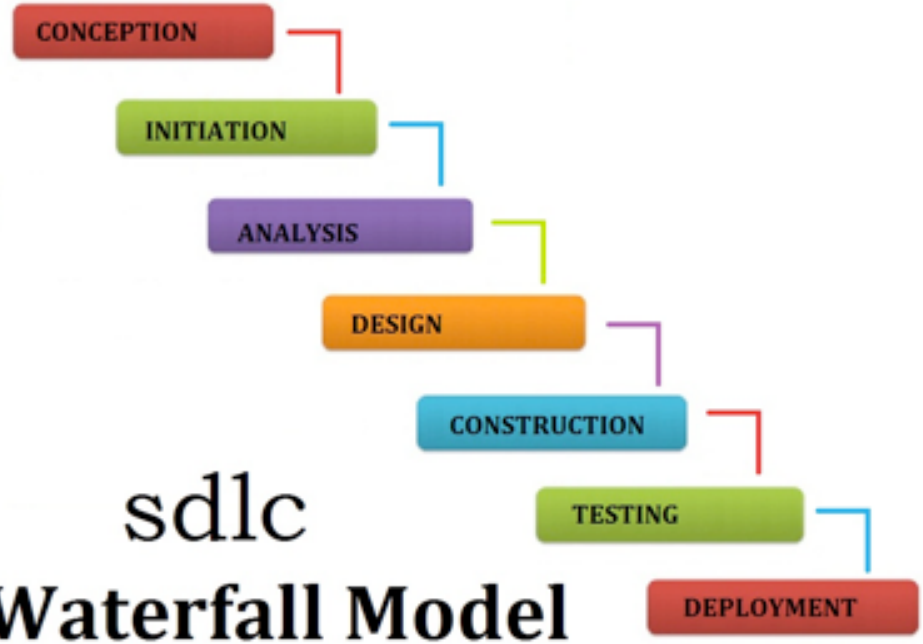


docker

service-based architecture

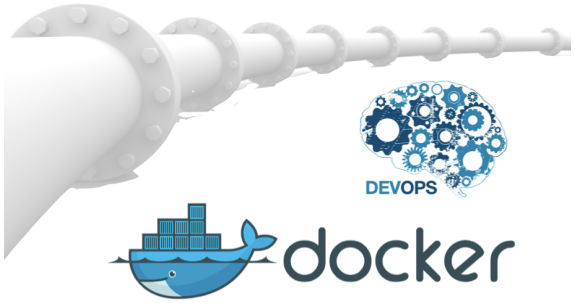
deployment pipeline

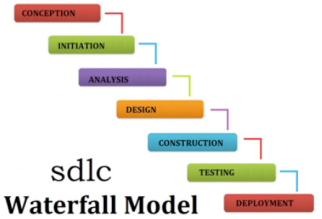




service-based architecture

deployment pipeline



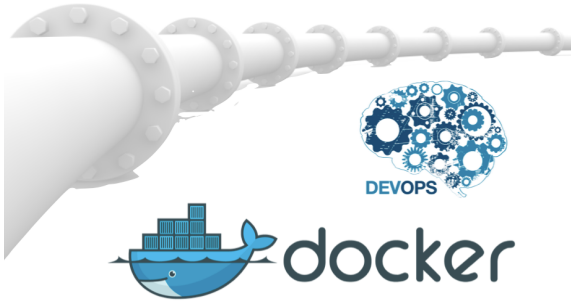


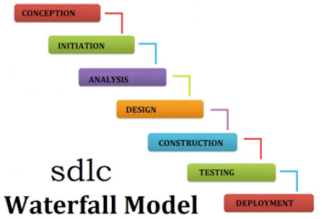
advantages

- no devops complexity
- no organizational change

service-based architecture

deployment pipeline



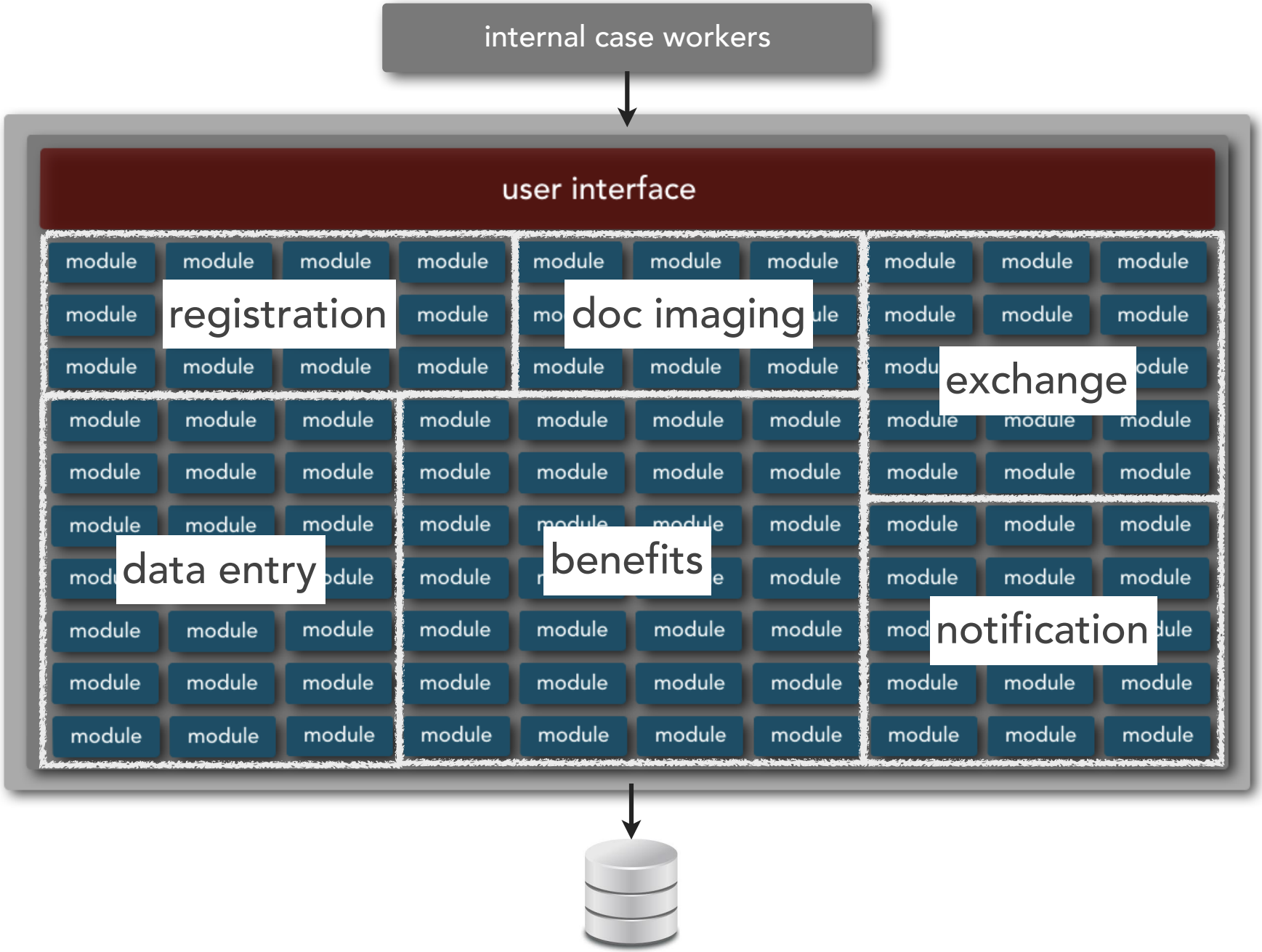


tradeoffs

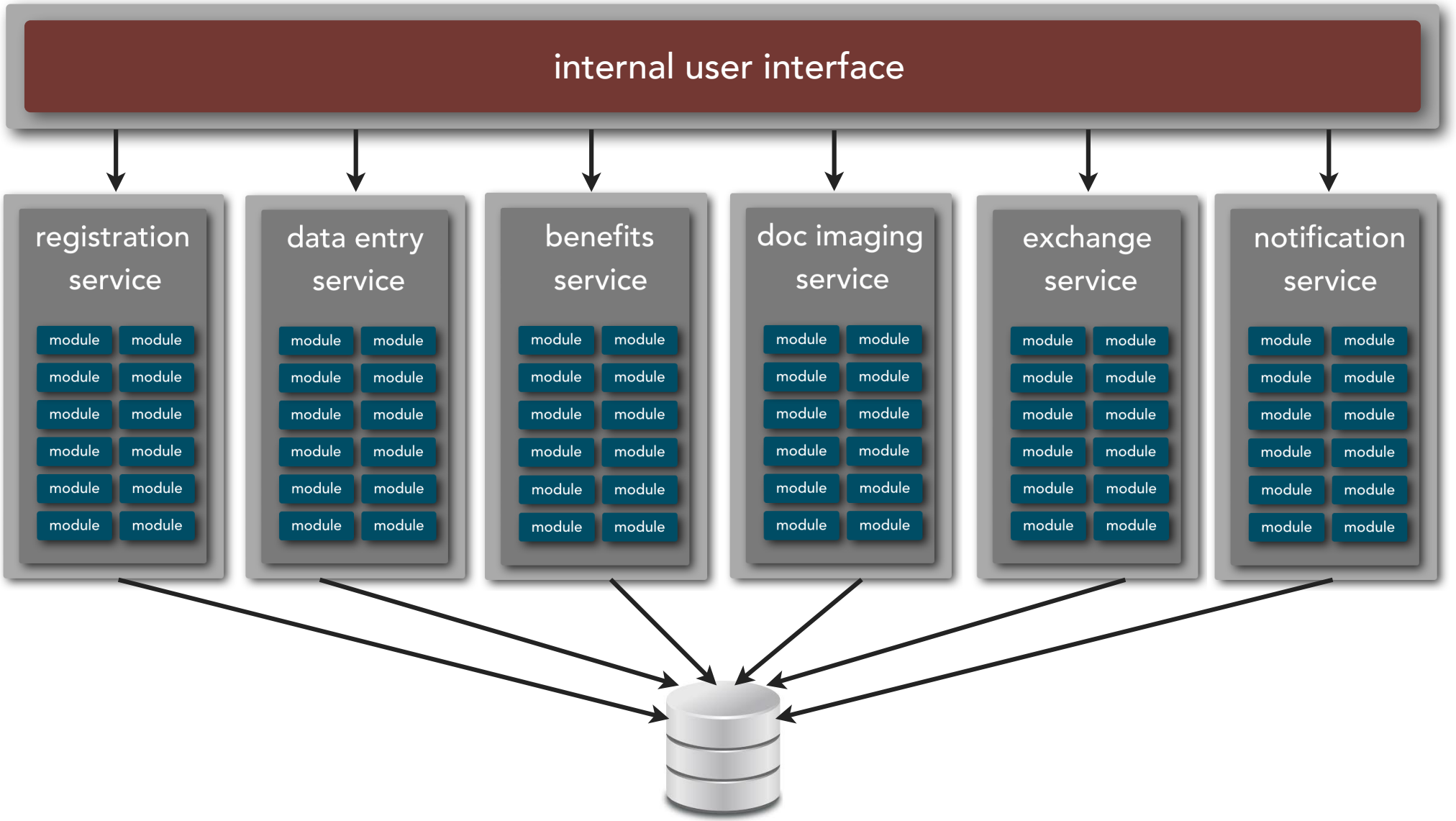
- lack of quick and effective deployments
- additional risk and coordination needed
- poor continuous delivery model

Examples

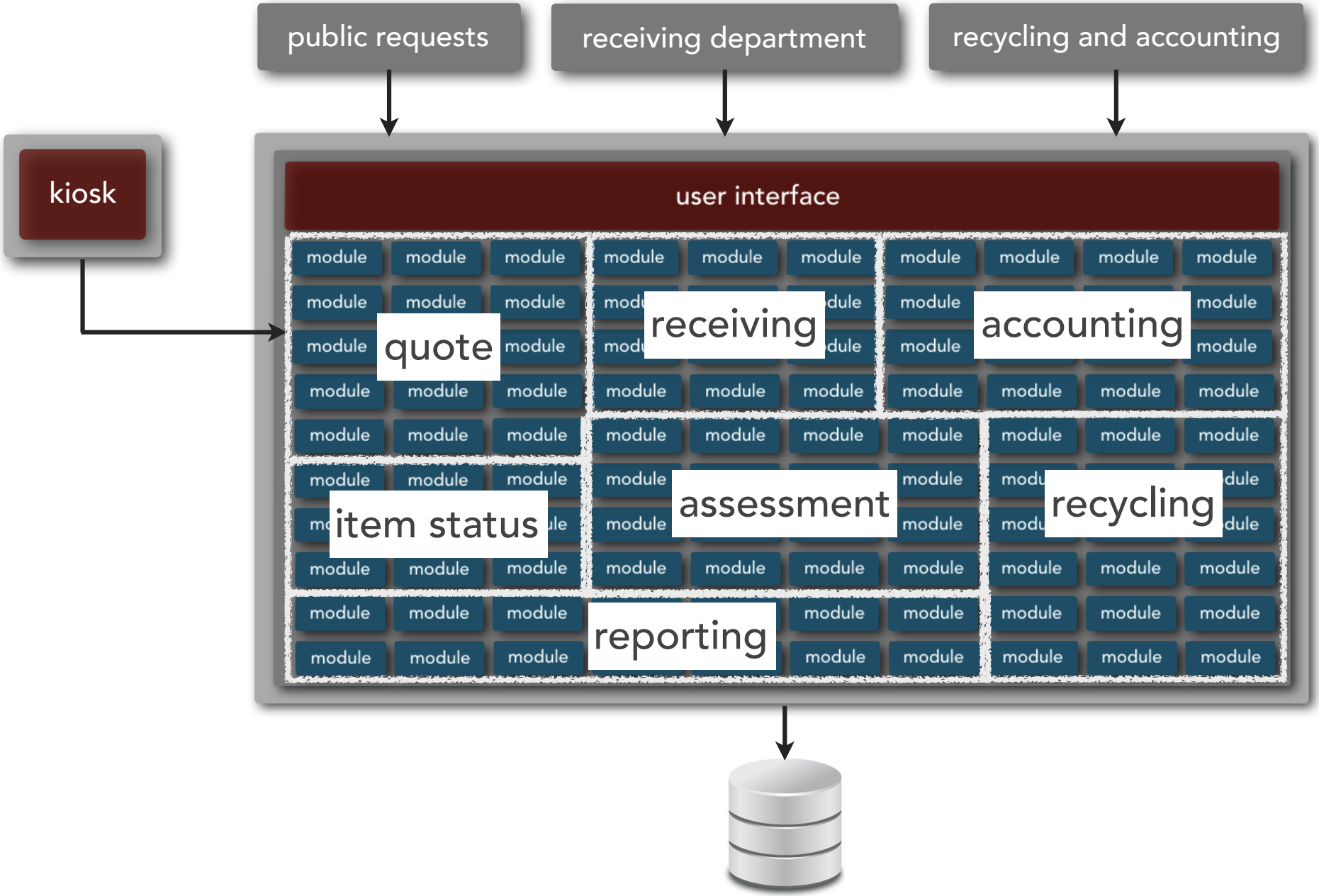
benefits application



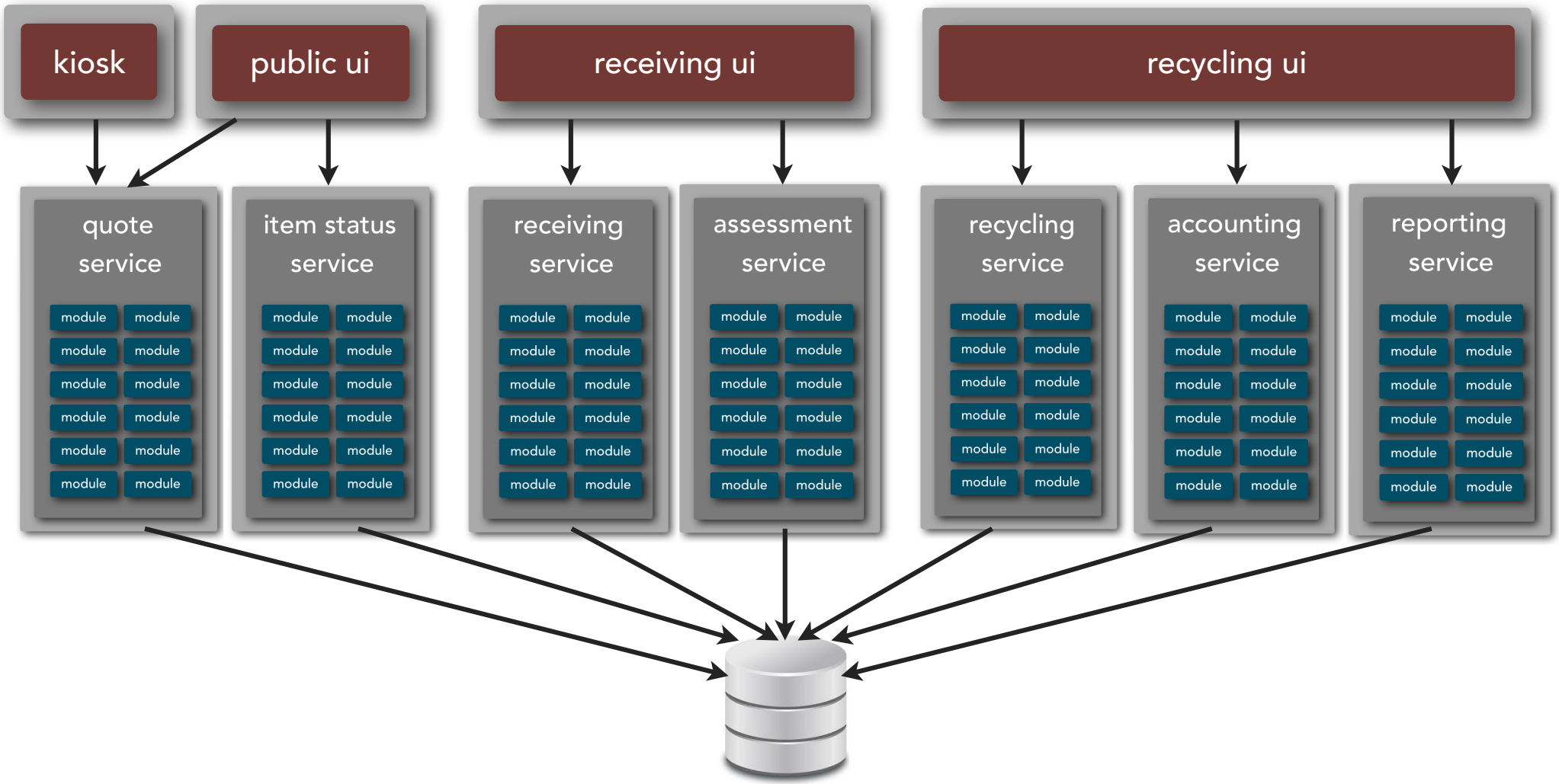
benefits application



electronics recycling application

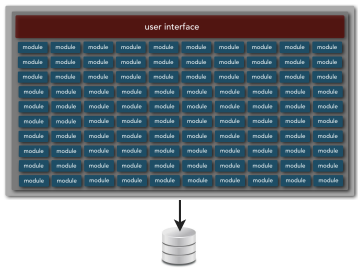


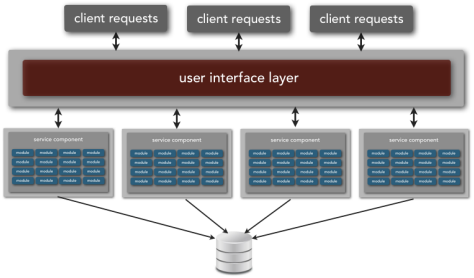
electronics recycling application

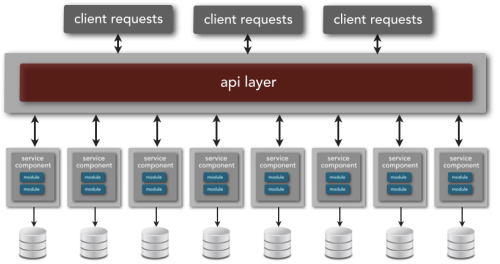


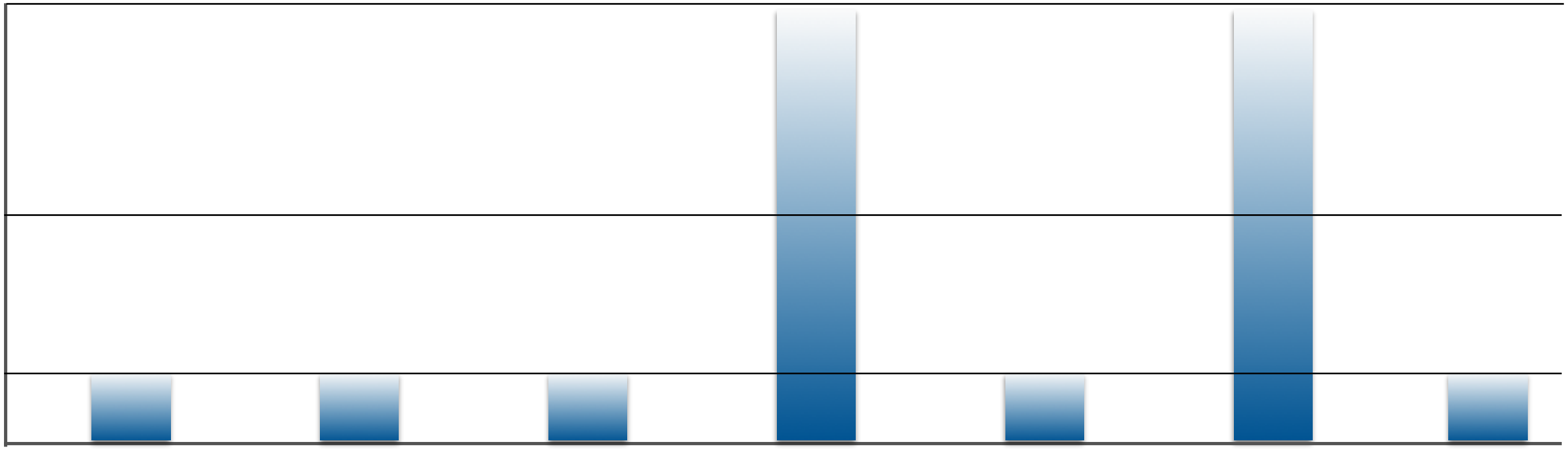
architectural modularity

architectural modularity

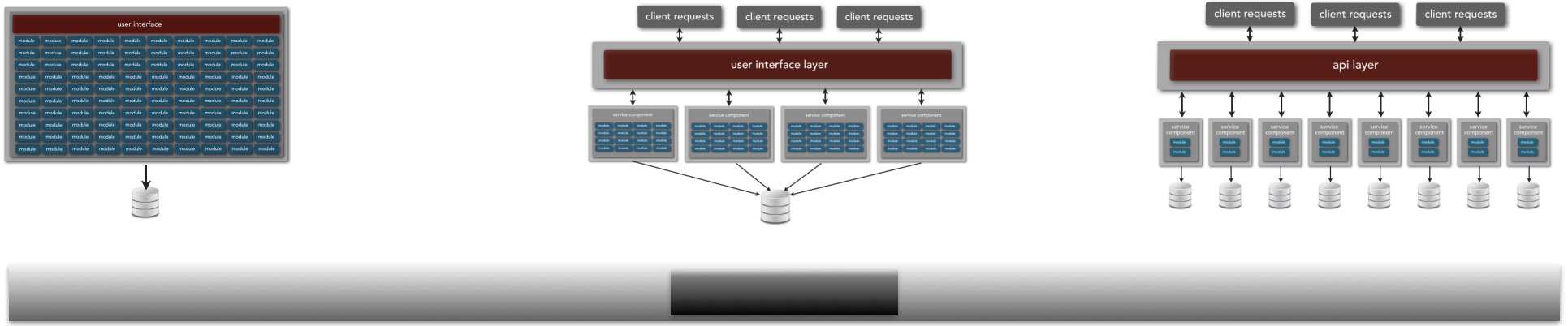


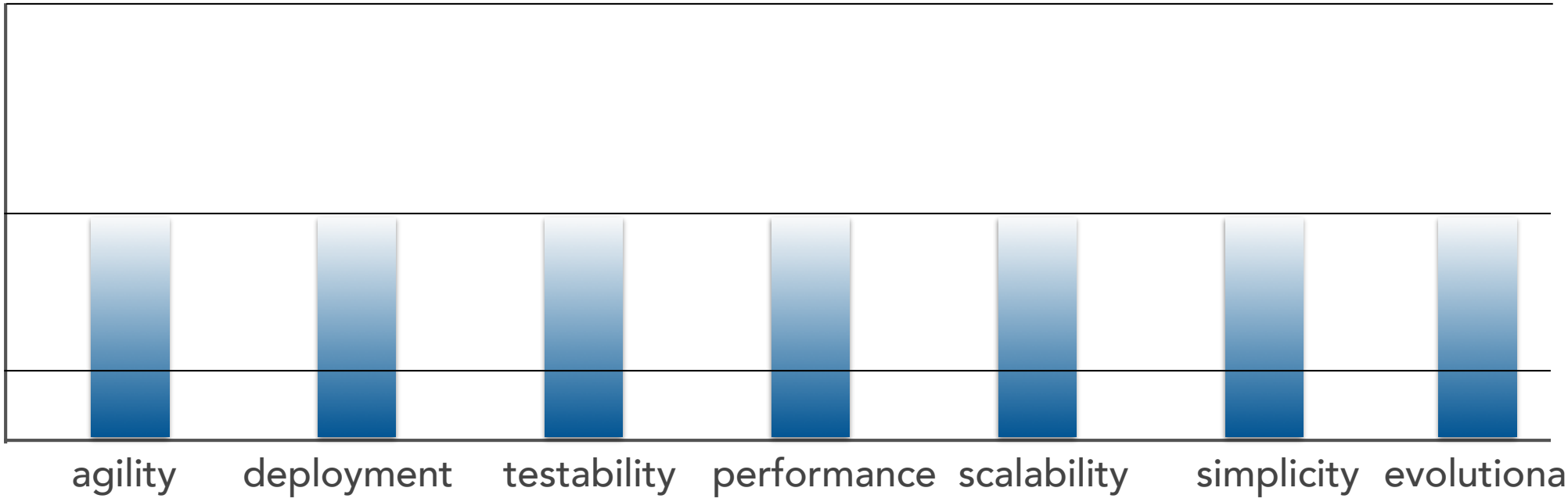




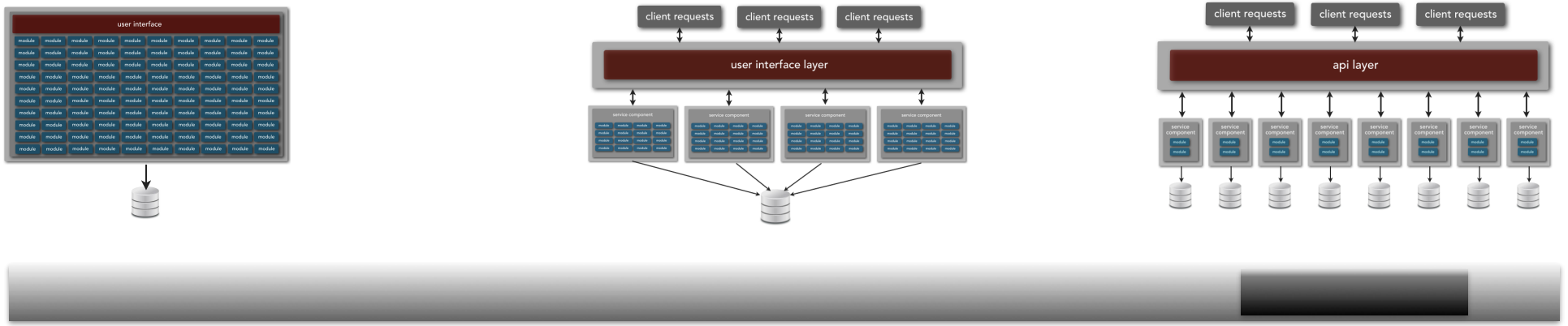


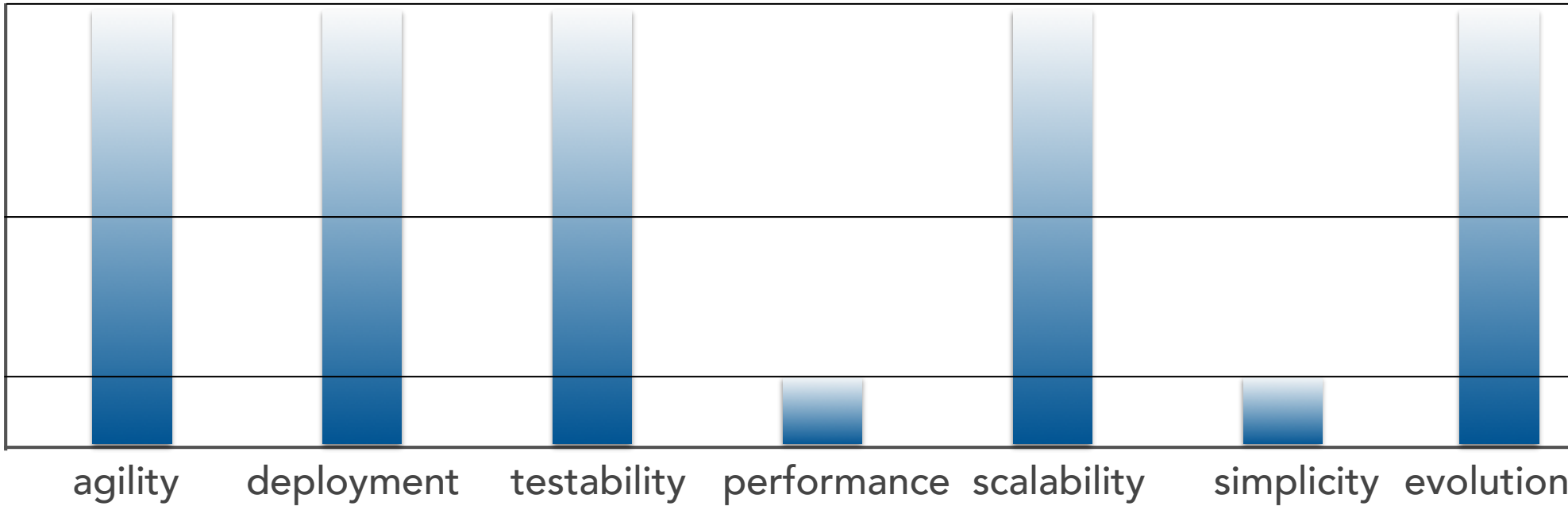
architectural modularity





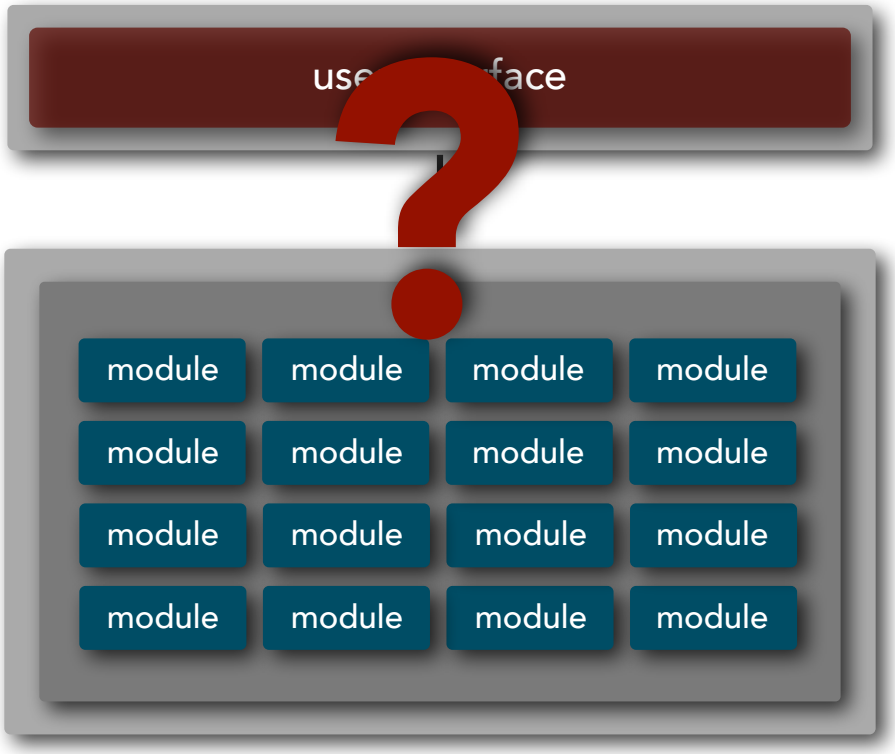
architectural modularity



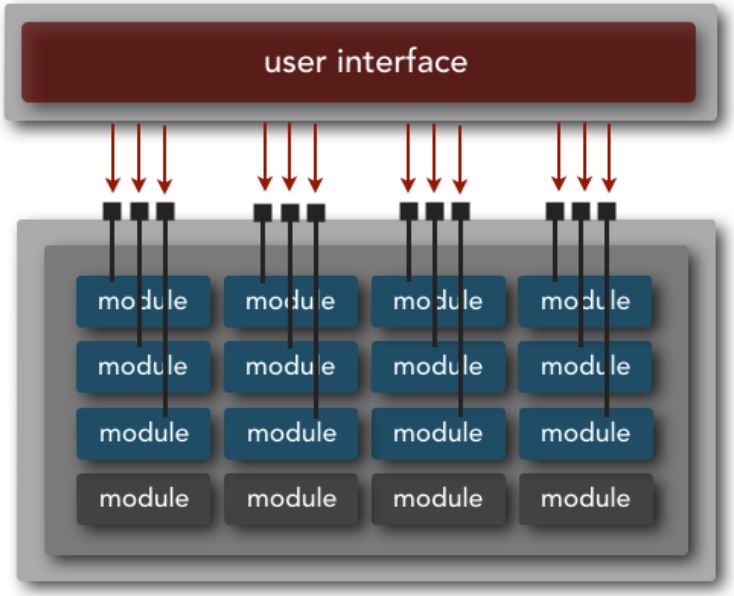


Service Design Considerations

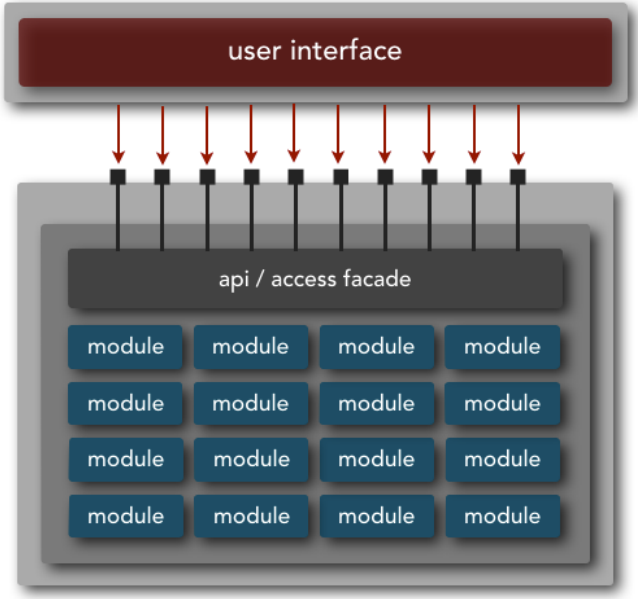
service design considerations



service design considerations



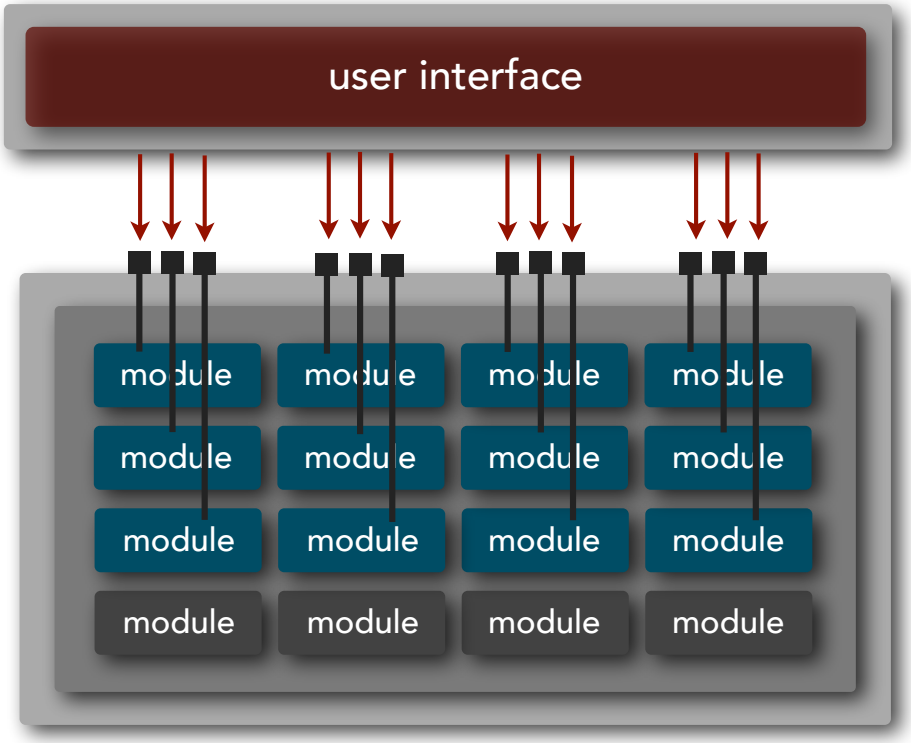
direct access design



api access design

service design considerations

direct access design



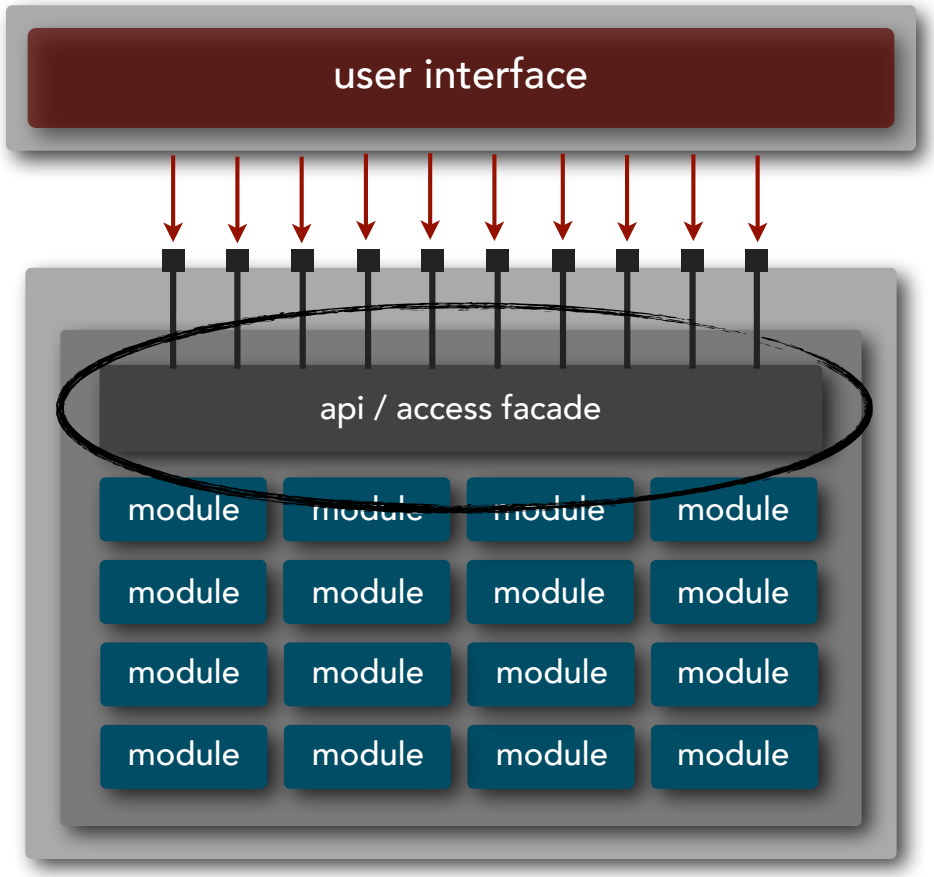
service design considerations

Request Modules

```
@Path("/customer/name")
@Produces(MediaType.APPLICATION_JSON)
public class CustomerNameResource
   @GET
   public String getName() {
      //logic to get name...
   }
}

@Path("/customer/contact")
@Produces(MediaType.APPLICATION_JSON)
public class CustomerContactResource
   @GET
   public String getContactInfo() {
      //logic to get contact info...
   }
}
```

service design considerations

api access design



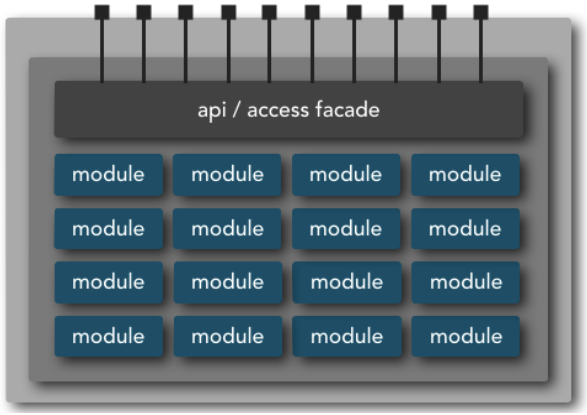
service design considerations

CustomerServiceAPI.java

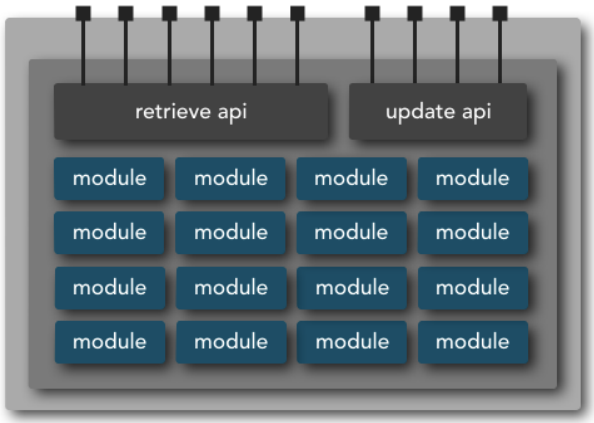
```
@Path("/customer")
@Produces(MediaType.APPLICATION_JSON)
public class CustomerServiceAPI {

   @Path("/name")
   @GET
   public String getNameRequest() {
      //invoke separate class to get name...
   }

   @Path("/contact_info")
   @GET
   public String getContactInfoRequest() {
     //invoke separate class to get contact info...
   }

   ...
```

service design considerations

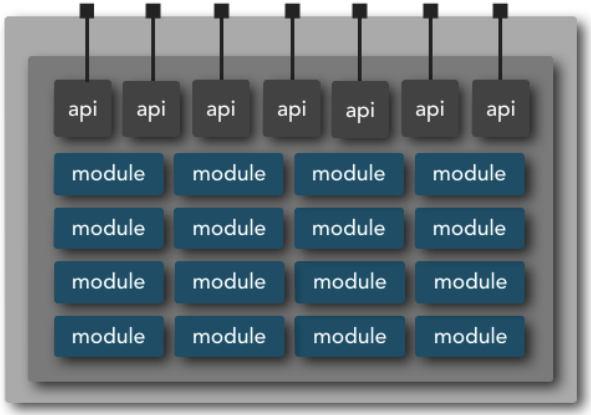
api access design variants



service-based api



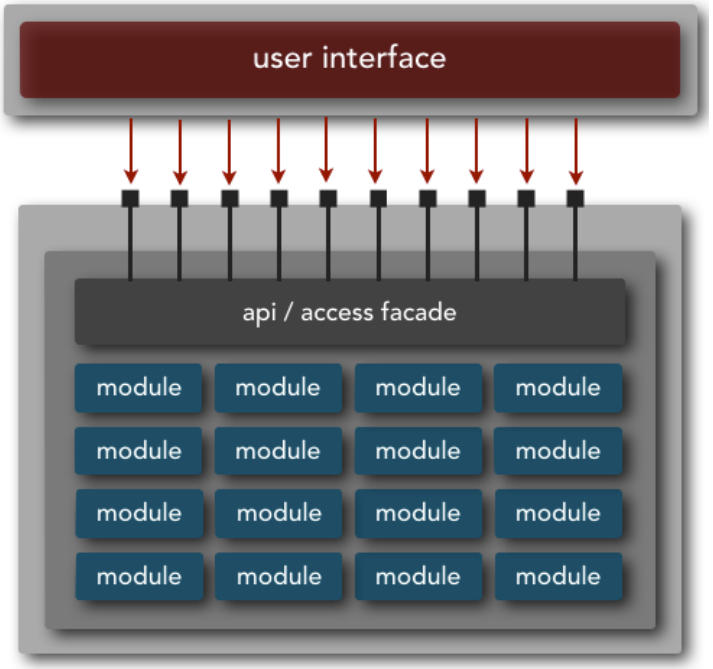
domain-based api



resource-based api

service design considerations

api access design benefits



separation of concerns

protocol-agnostic processing

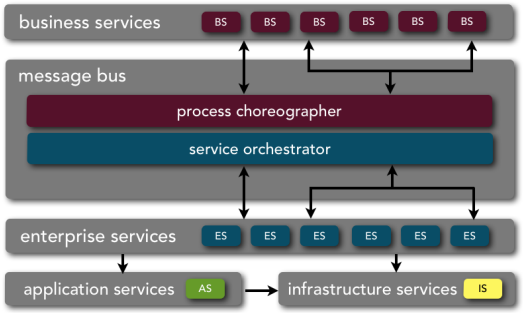
service context / binding

service catalog

Service-Based Architecture
vs.
Service-Oriented Architecture

service-based vs. service-oriented

service-oriented architecture



enterprise scope

heterogeneous integration

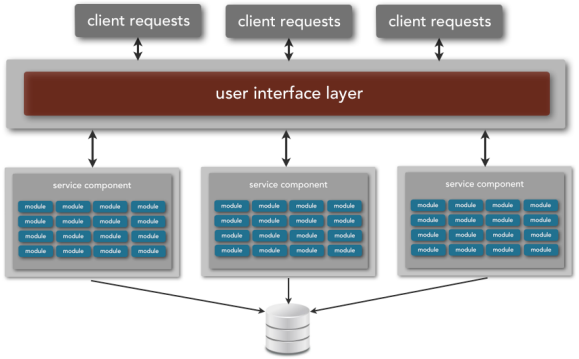
middleware capabilities

resource sharing

multiple databases

strict service taxonomy

service-based architecture



application scope

limited integration

no middleware

limited sharing

single database

no service taxonomy

Service-Based Architecture

Mark Richards
Independent Consultant
Hands-on Software Architect
Published Author / Conference Speaker

http://www.wmrichards.com
https://www.linkedin.com/in/markrichards3
@markrichardssa