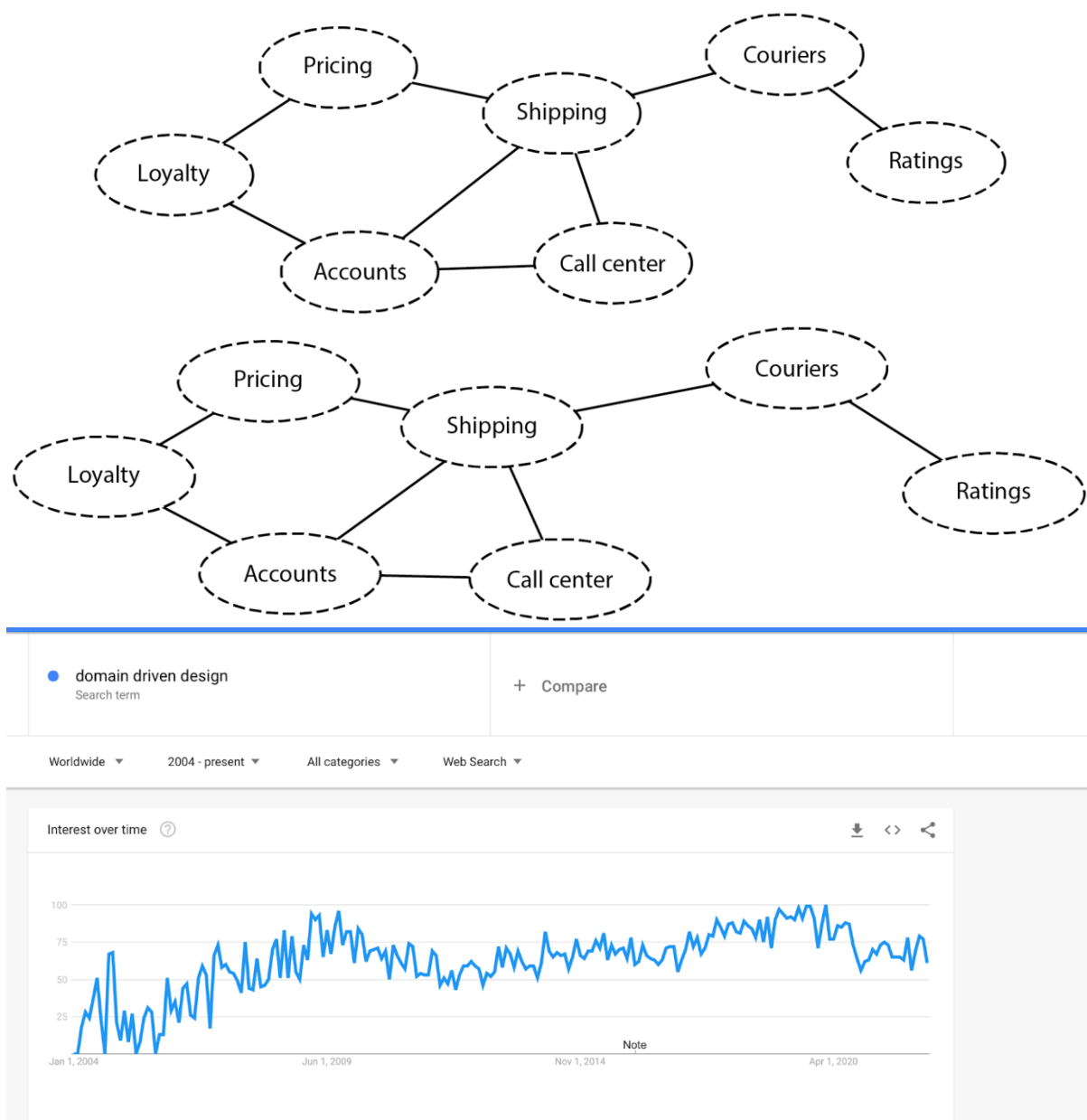
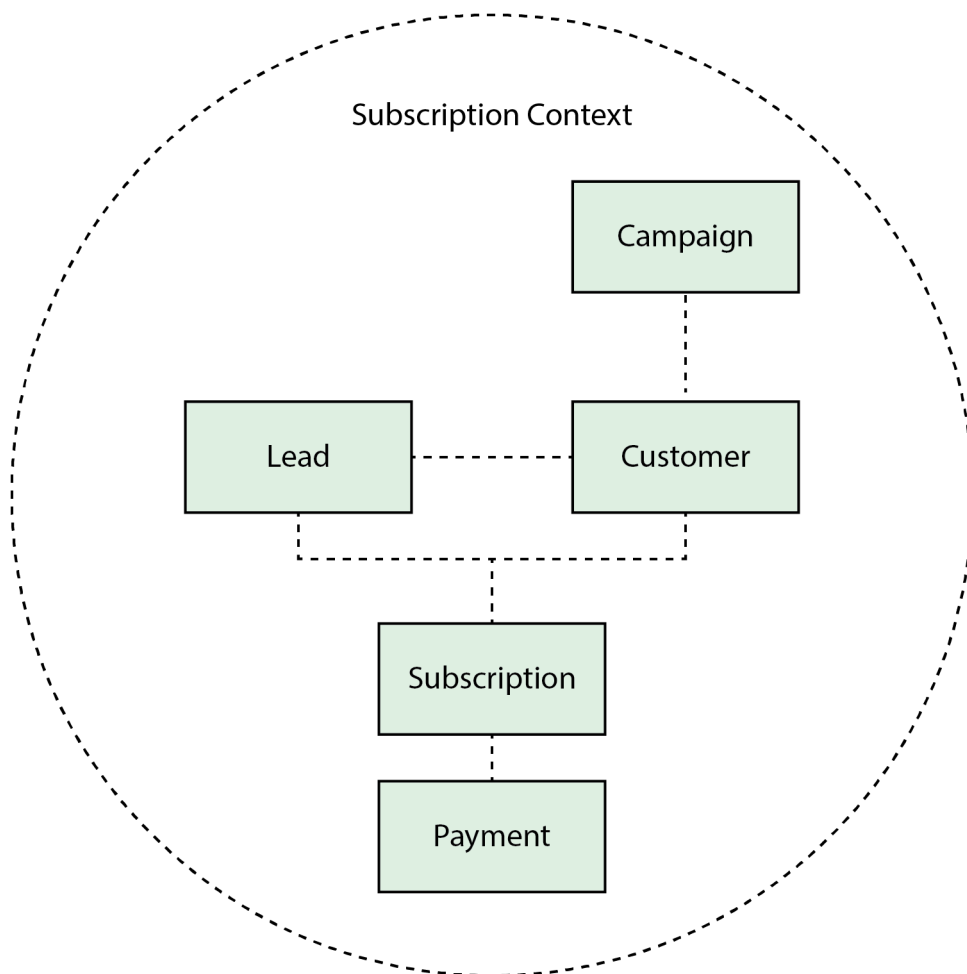
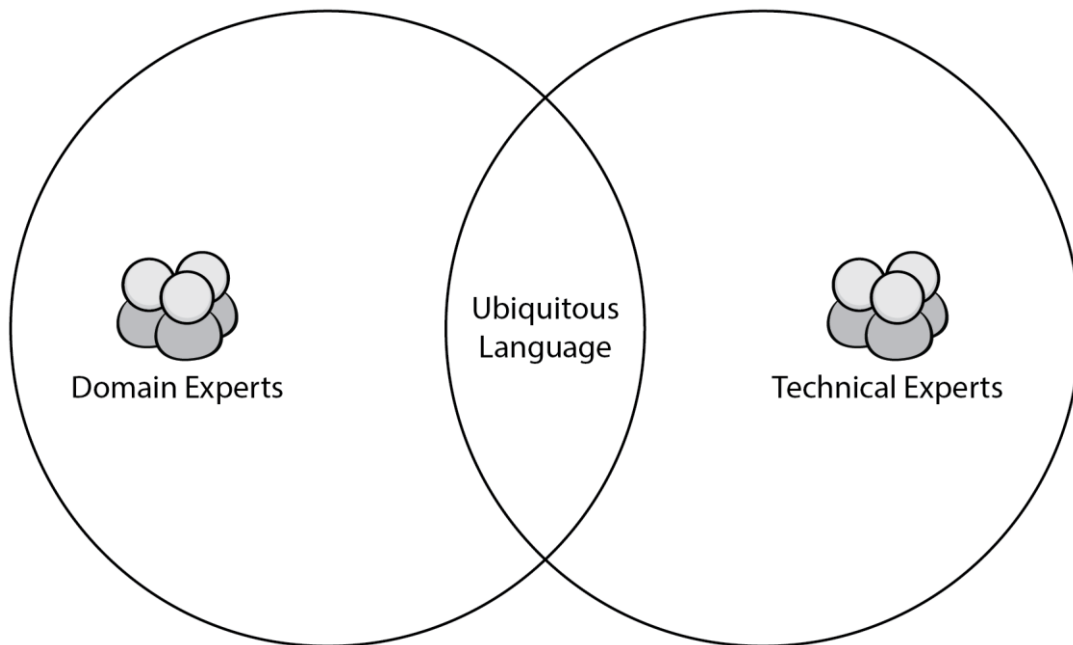
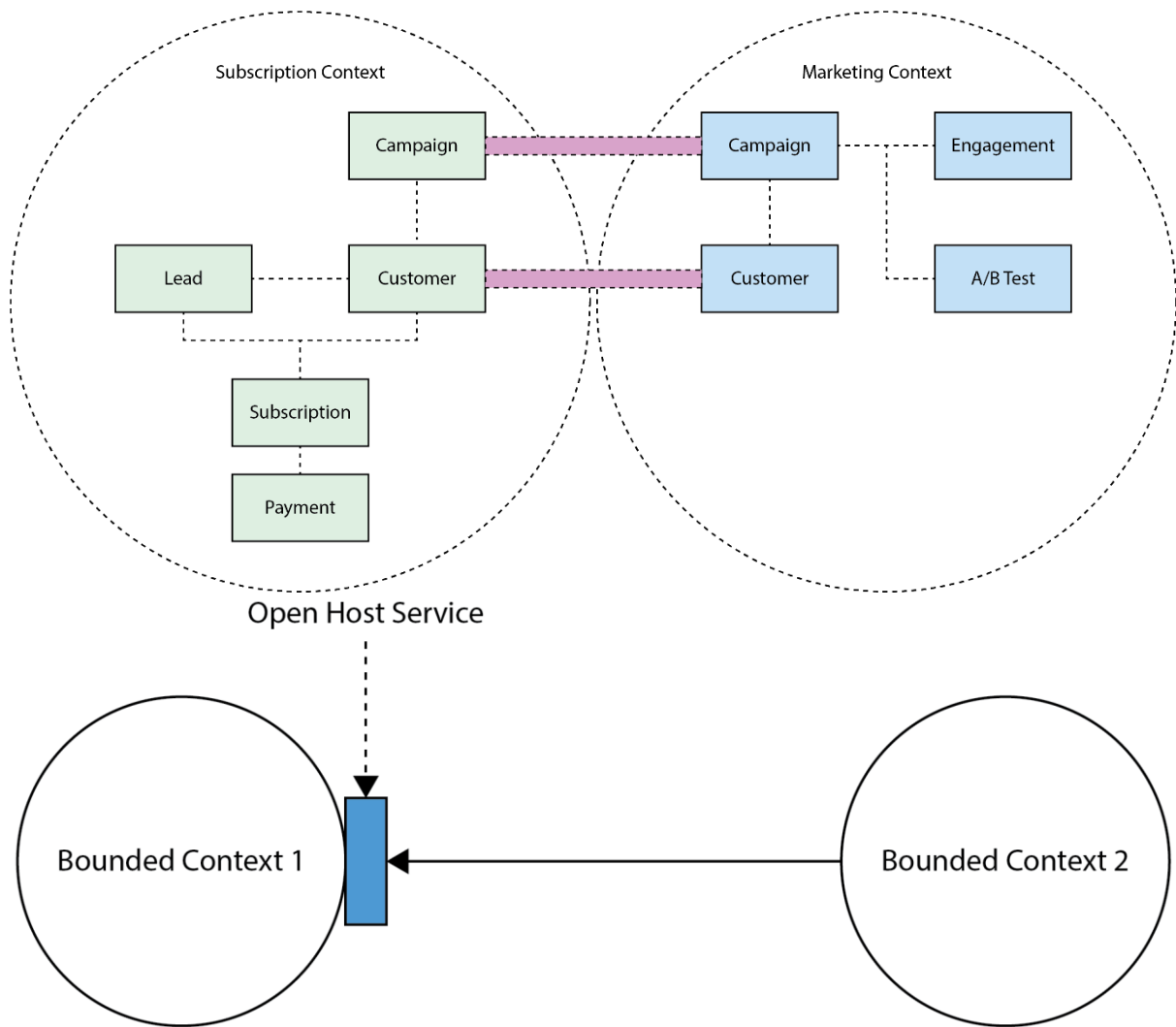


Chapter 1: A brief history of Domain Driven Design



Chapter 2: Understanding Domains, Ubiquitous Language, and Bounded Contexts





Payment & Subscription API 1.0.0

[Base URL: `api.payments.com`]

Public documentation for payment & subscription System

[Contact the developer](#)

Schemes

HTTPS

default

GET

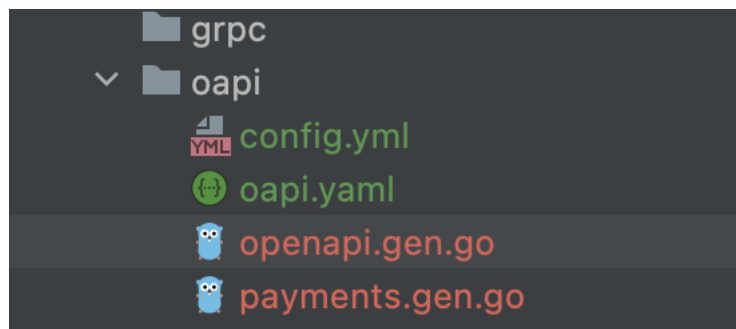
`/{{id}}/user` Find a users payment and subscription information by ID.

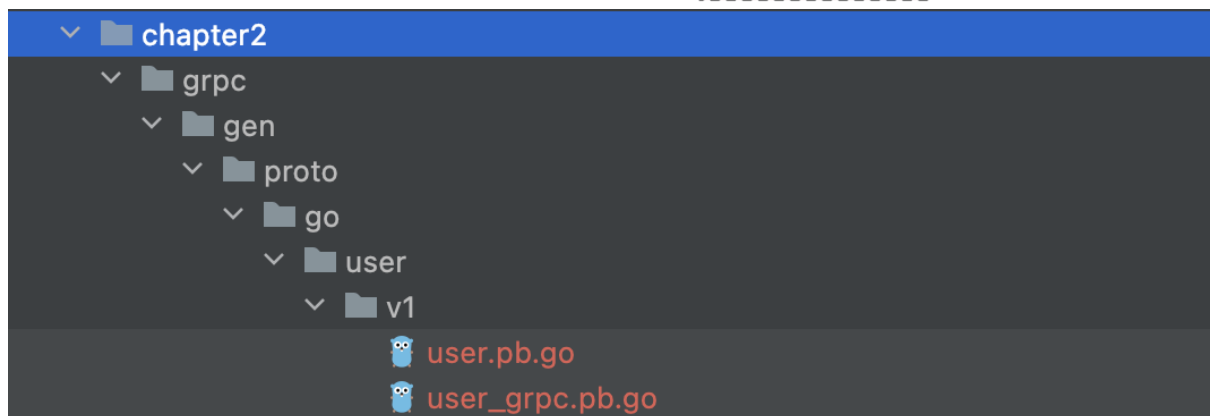
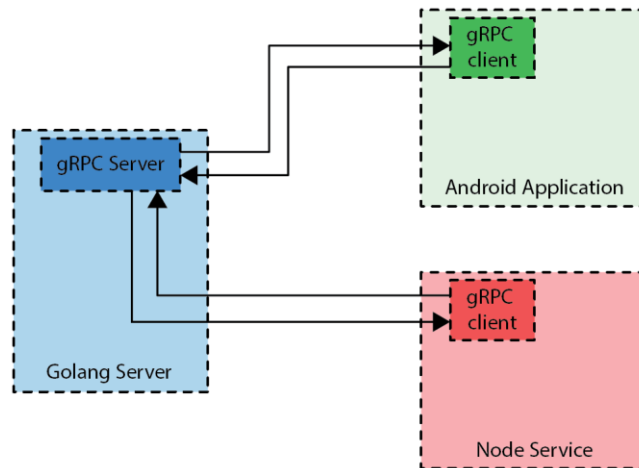
Models

```
User {
  id          integer($int64)
  username    string
  subscriptionStatus boolean
  subscriptionType string
  email       string
}
```

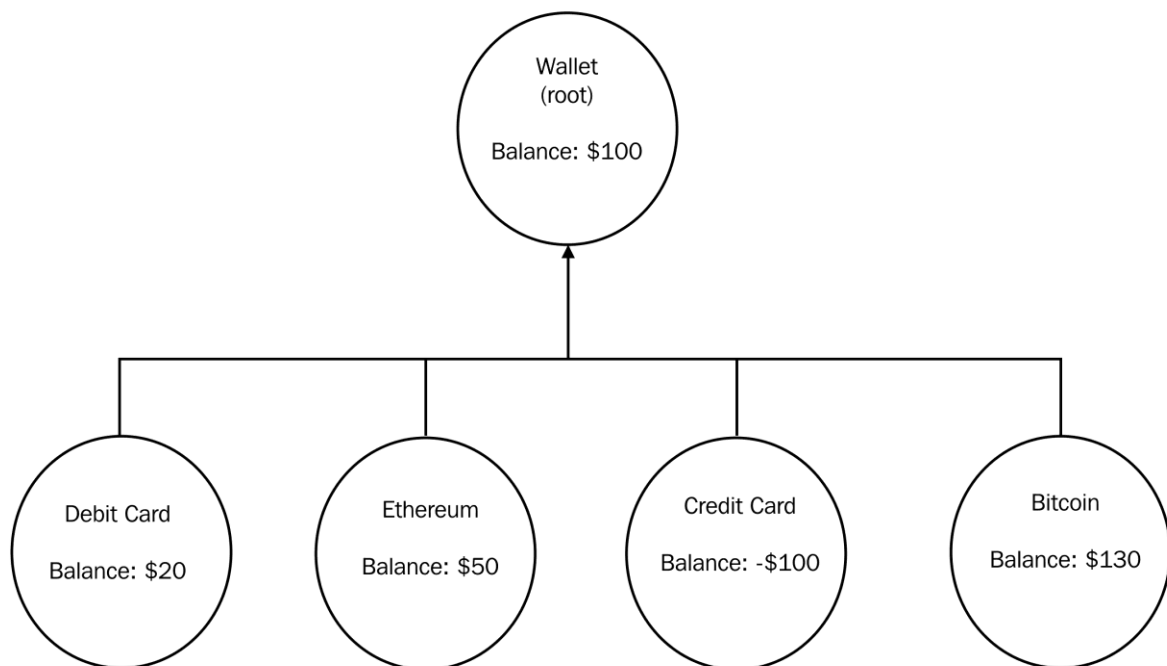
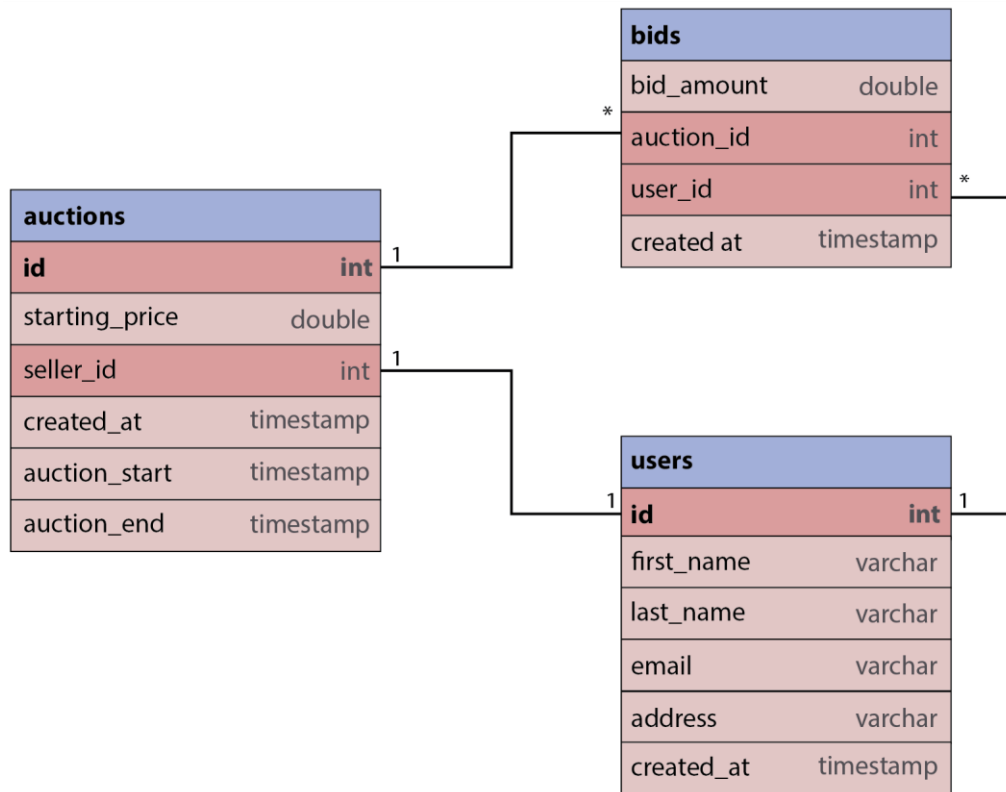
```
ApiResponse {
  code    integer($int32)
  type    string
  message string
}
```

ada-server	jaxrs-cxf-cdi	python-flask
aspnetcore	jaxrs-resteasy	rails5
erlang-server	jaxrs-resteasy-eap	restbed
finch	jaxrs-spec	rust-server
go-server	kotlin-server	scala-lagom-server
haskell	lumen	scalatra
inflector	msf4j	sinatra
java-pkmst	nancyfx	slim
java-play-framework	nodejs-server	spring
java-vertx	php-silex	undertow
jaxrs	php-symfony	ze-ph
jaxrs-cxf	pistache-server	

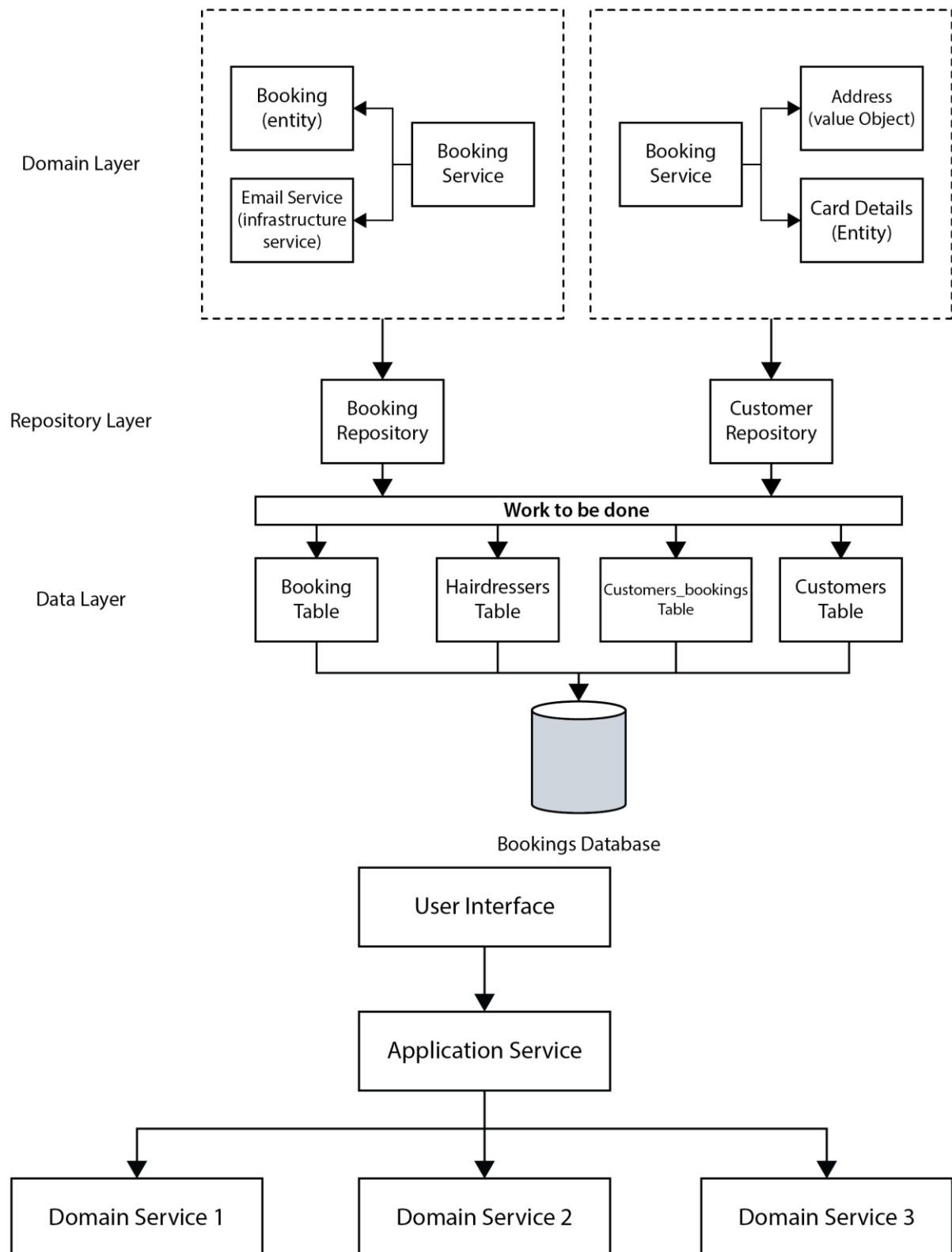




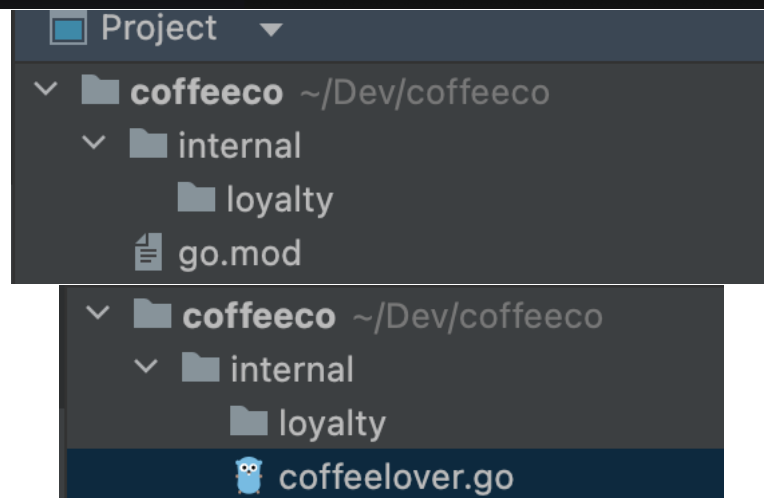
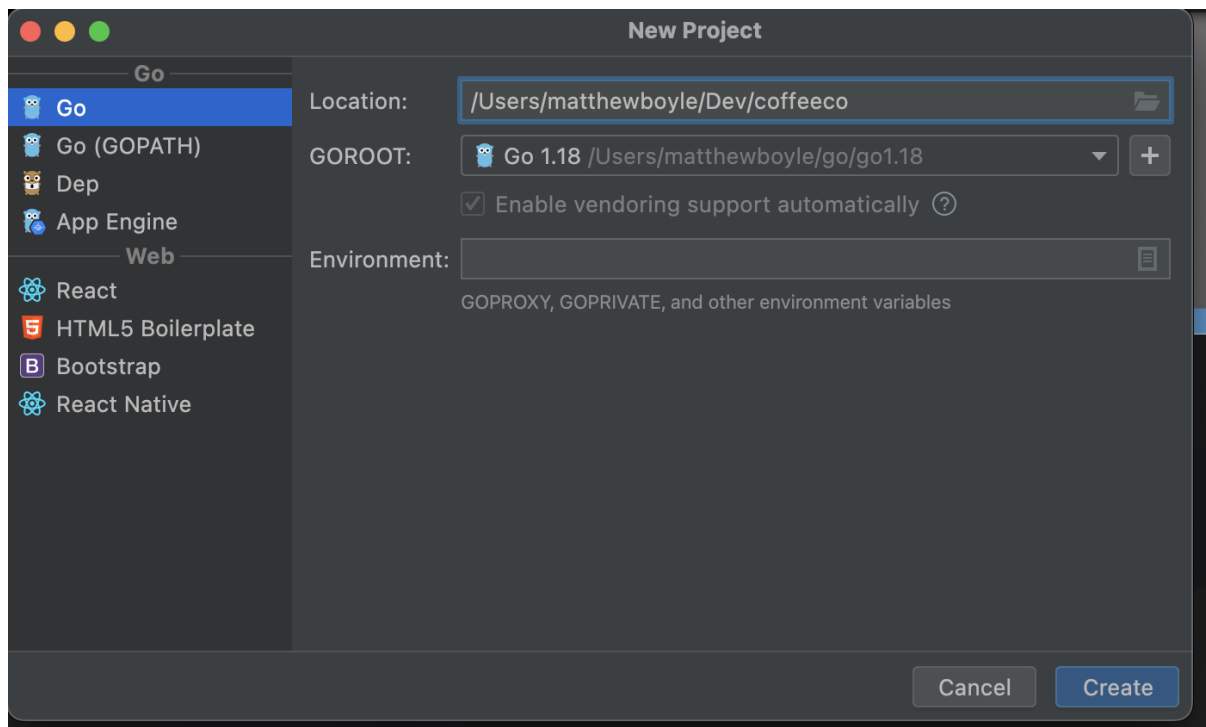
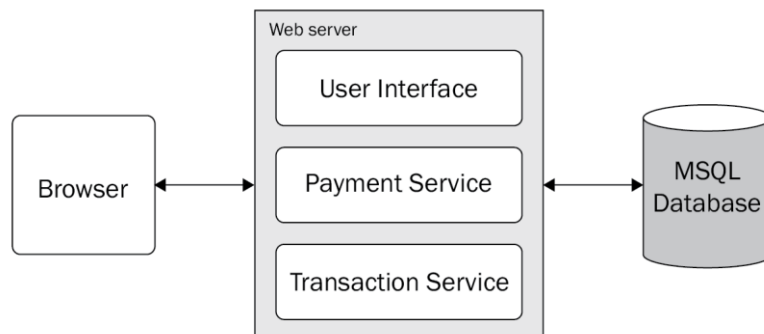
Chapter 3: Aggregates, Entities & Value Objects

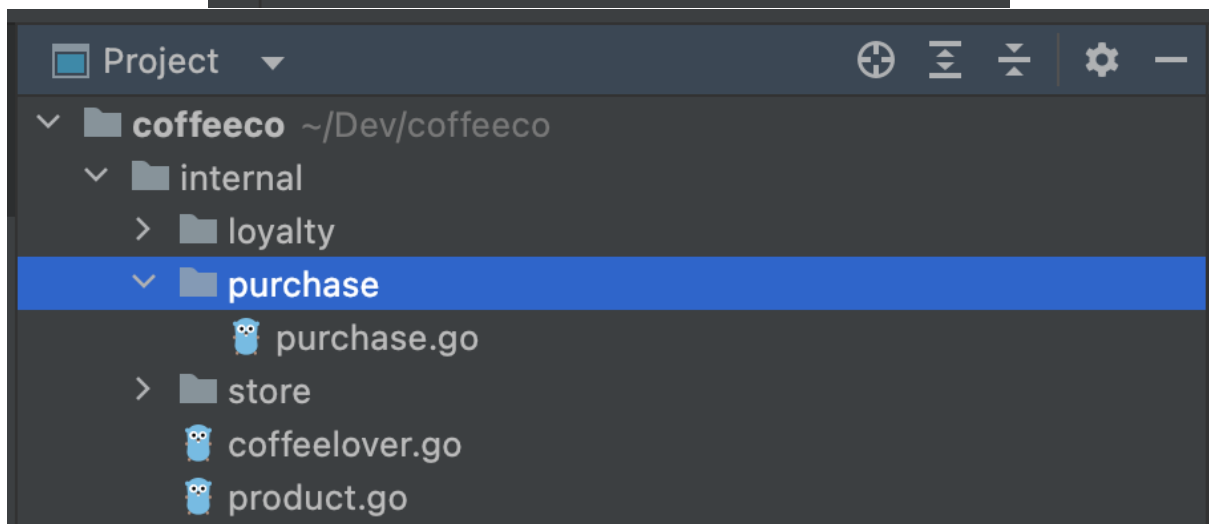
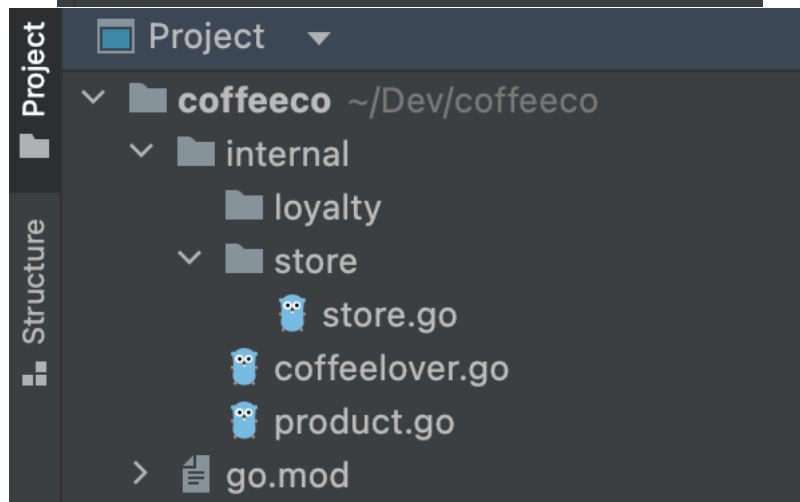
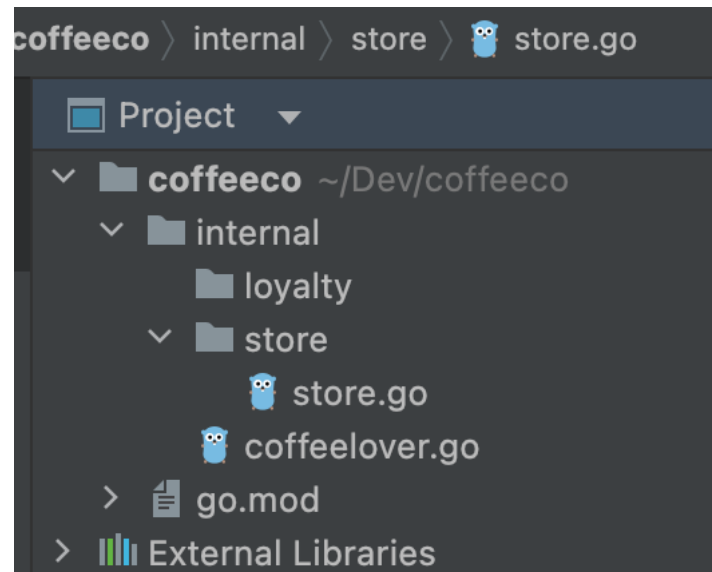


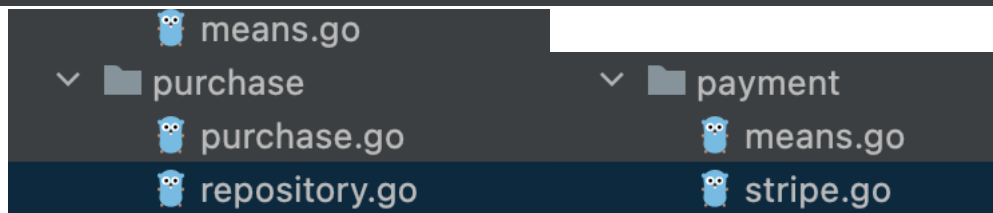
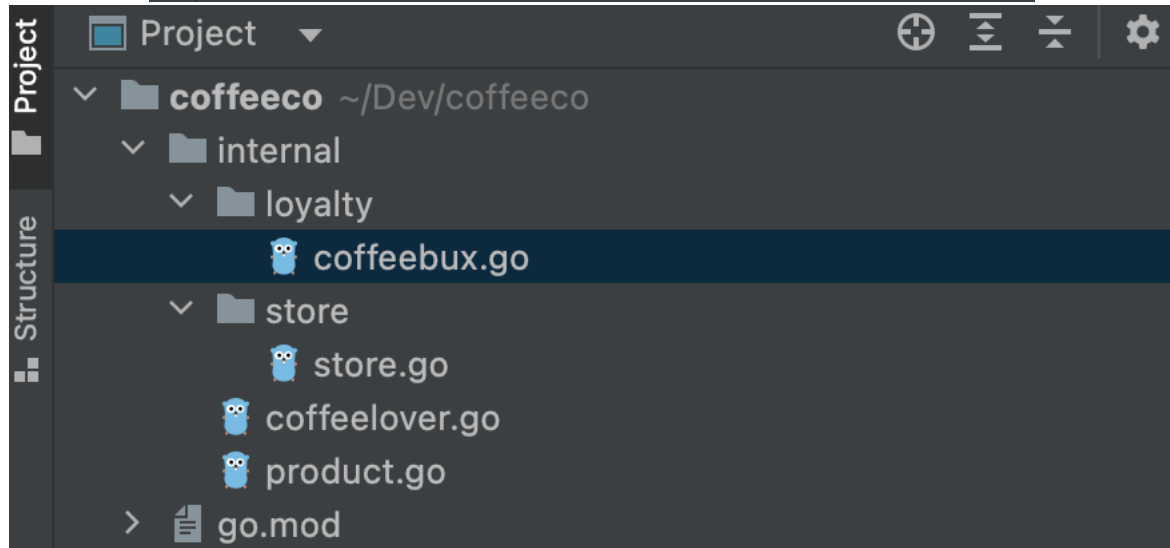
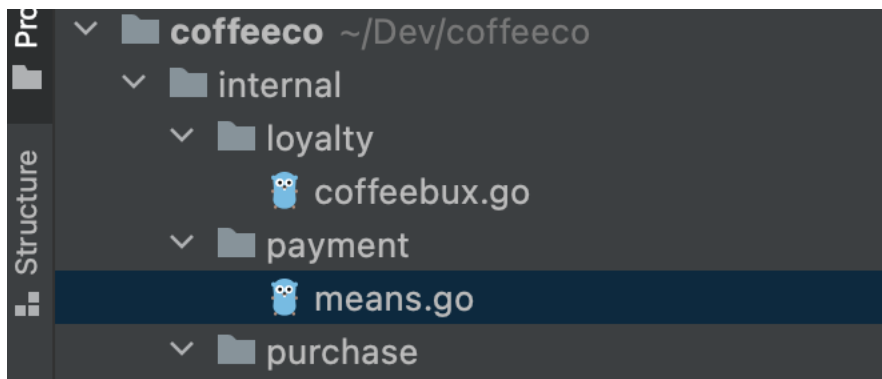
Chapter 4: Factories, Repositories & Services



Chapter 5: Applying Domain Driven Design to a monolithic application





















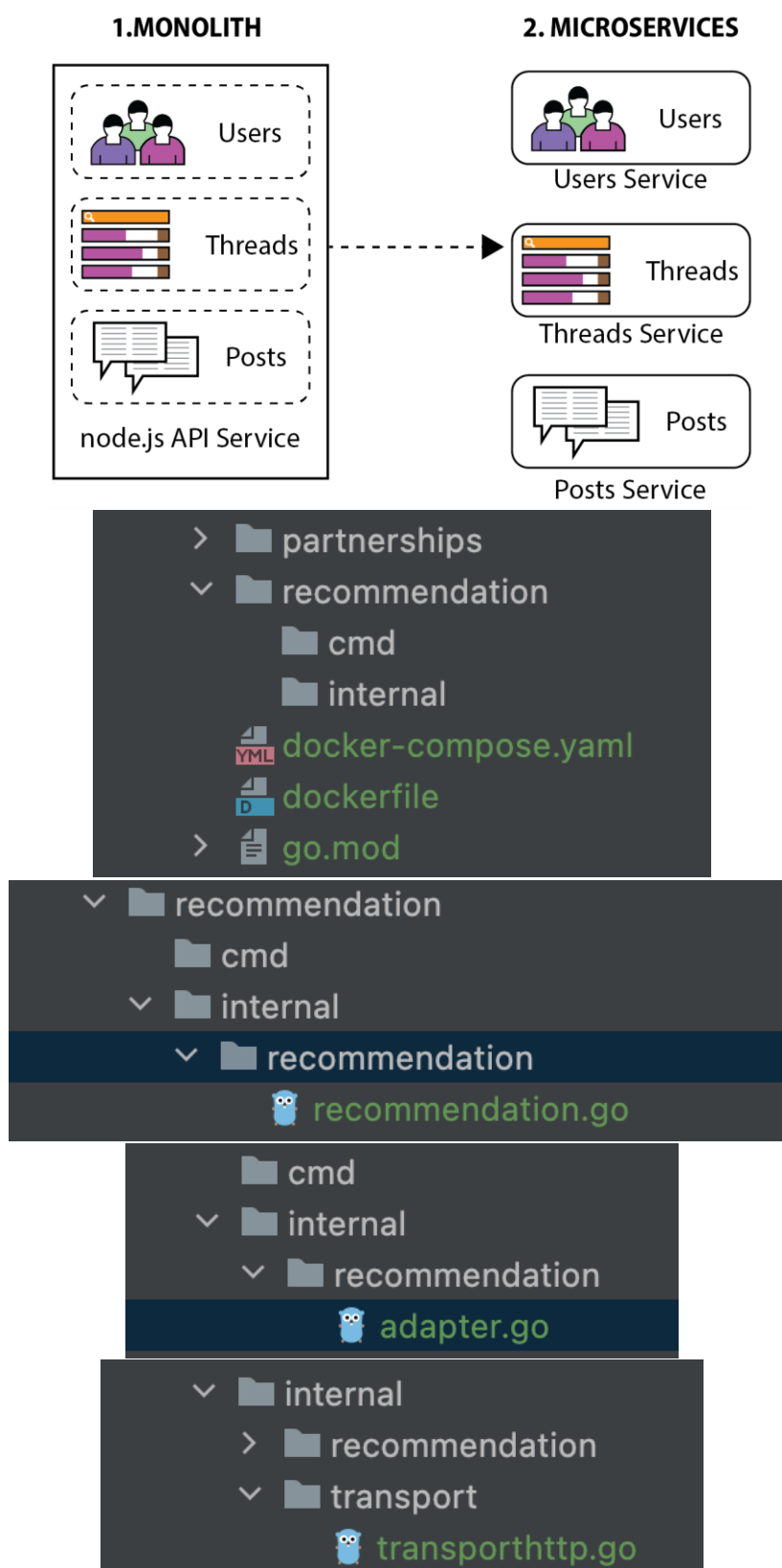
▼  store

-  repository.go
-  store.go

▼  internal


- ▼  loyalty
 -  coffeebux.go
- ▼  payment
 -  means.go
 -  stripe.go
- ▼  purchase
 -  purchase.go
 -  repository.go
- ▼  store
 -  repository.go
 -  store.go
 -  coffeelover.go
 -  product.go

Chapter 6: Building a microservice using domain driven design

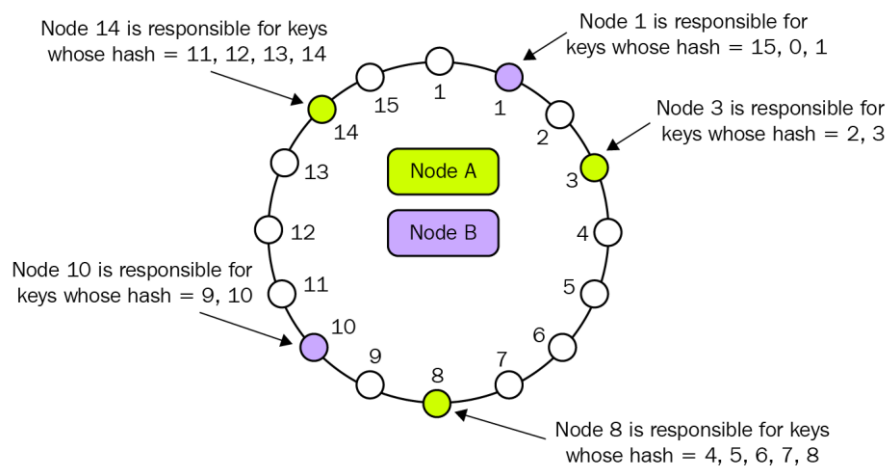
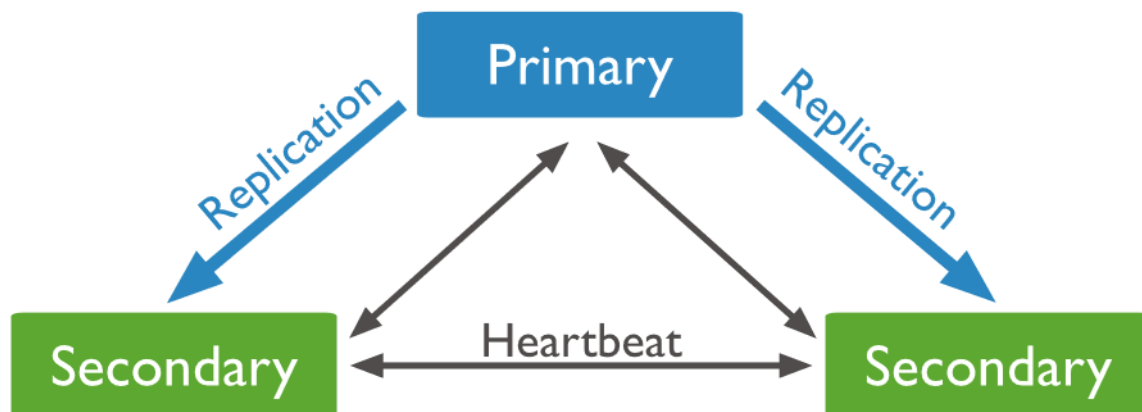
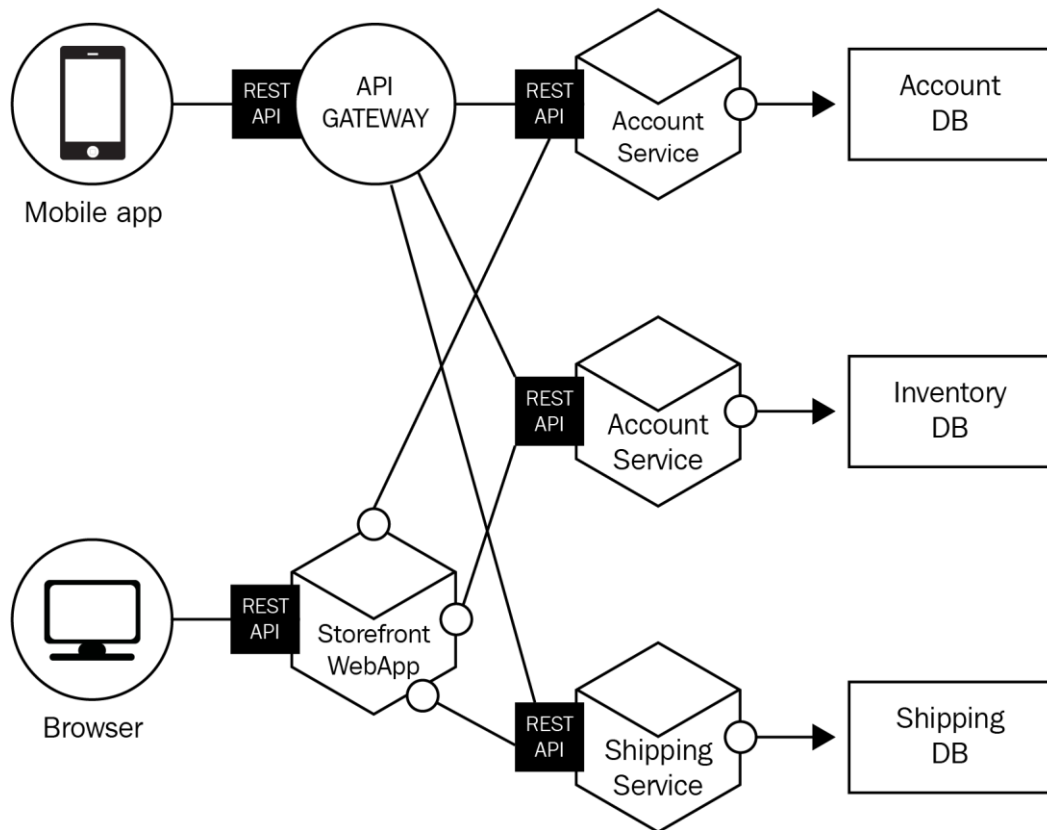


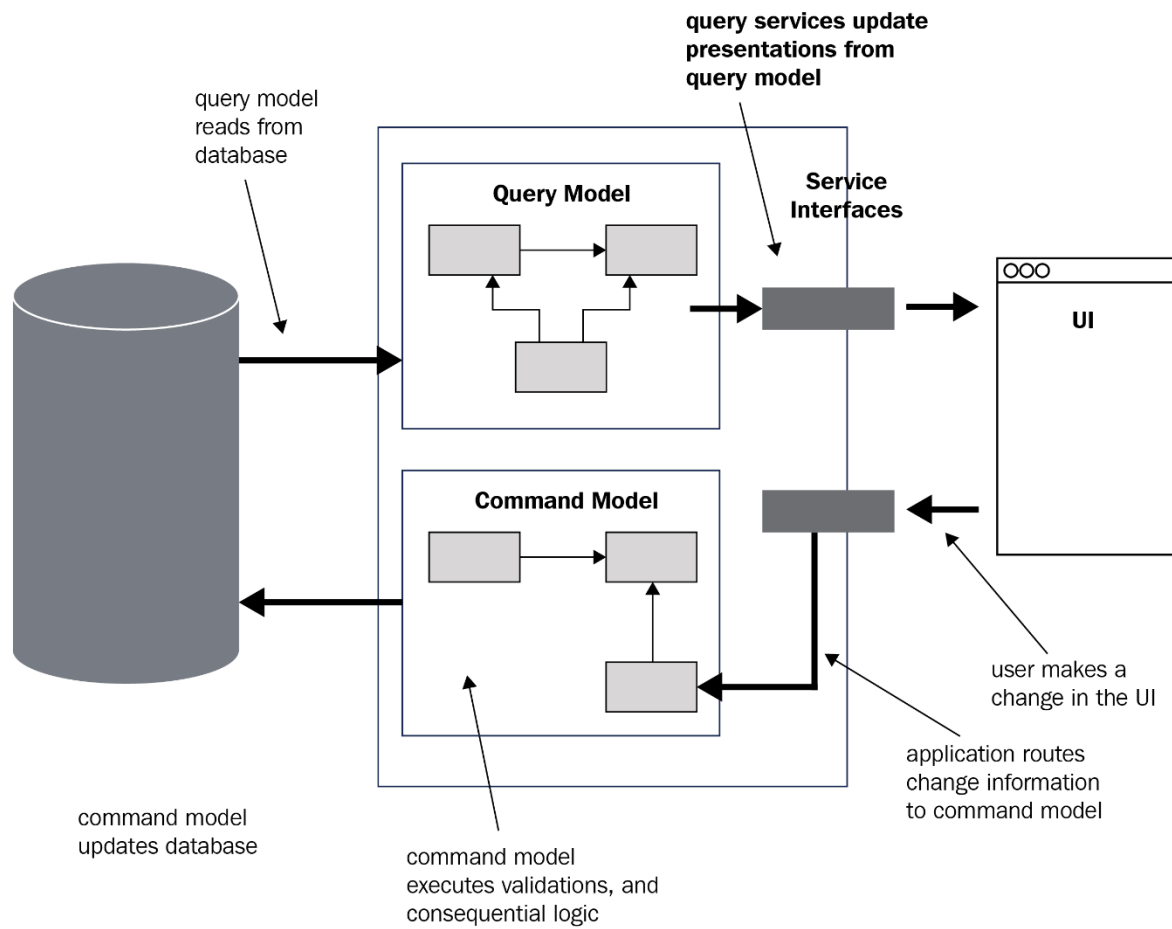
▼  recommendation

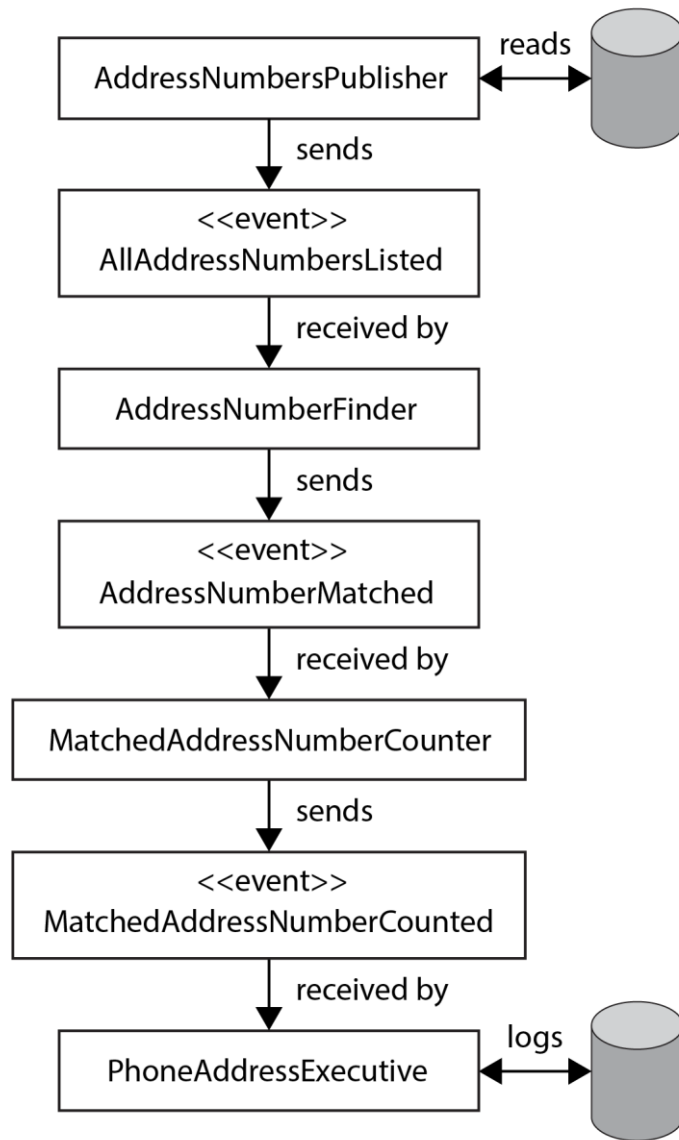
▼  cmd

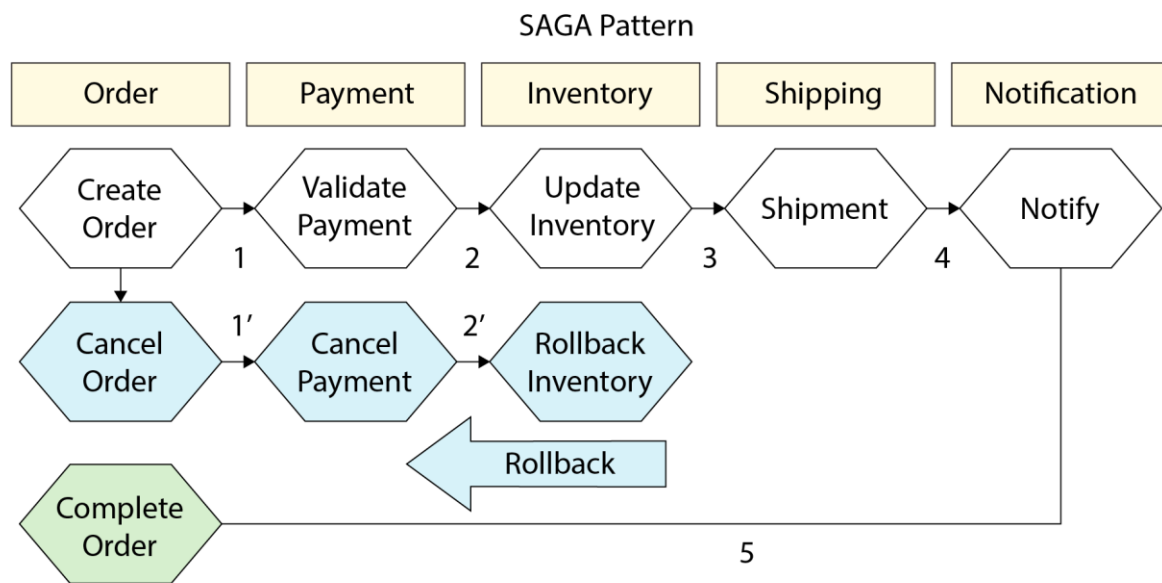
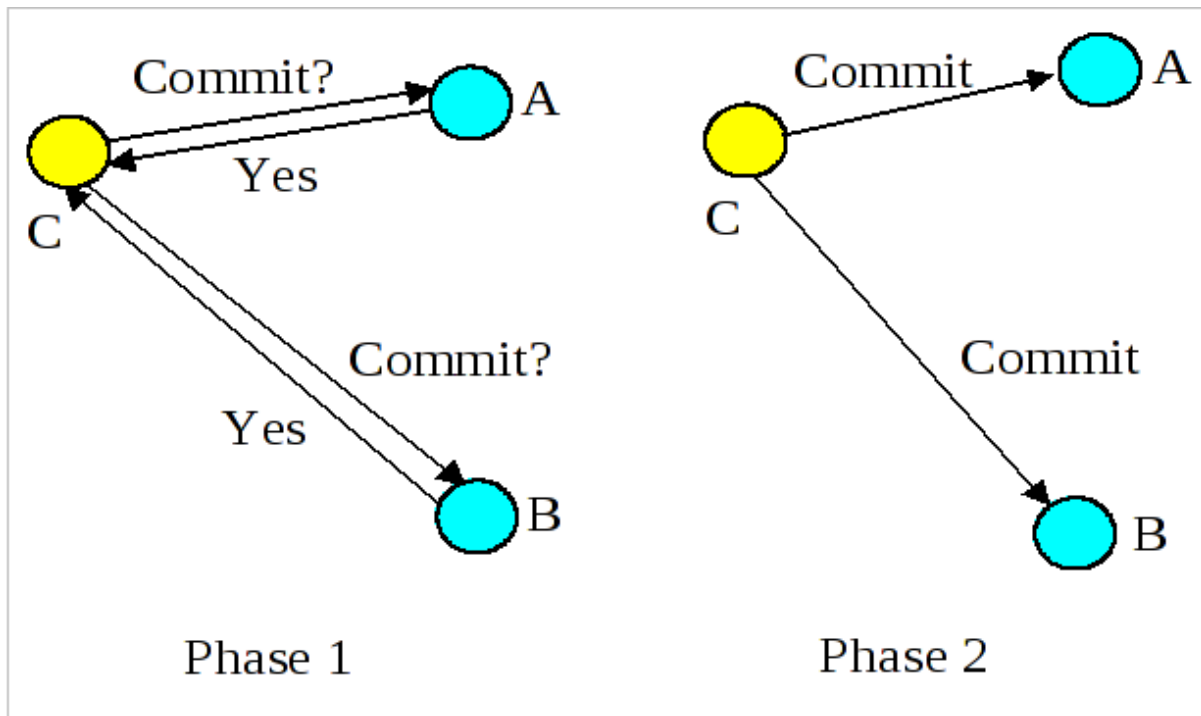
 main.go

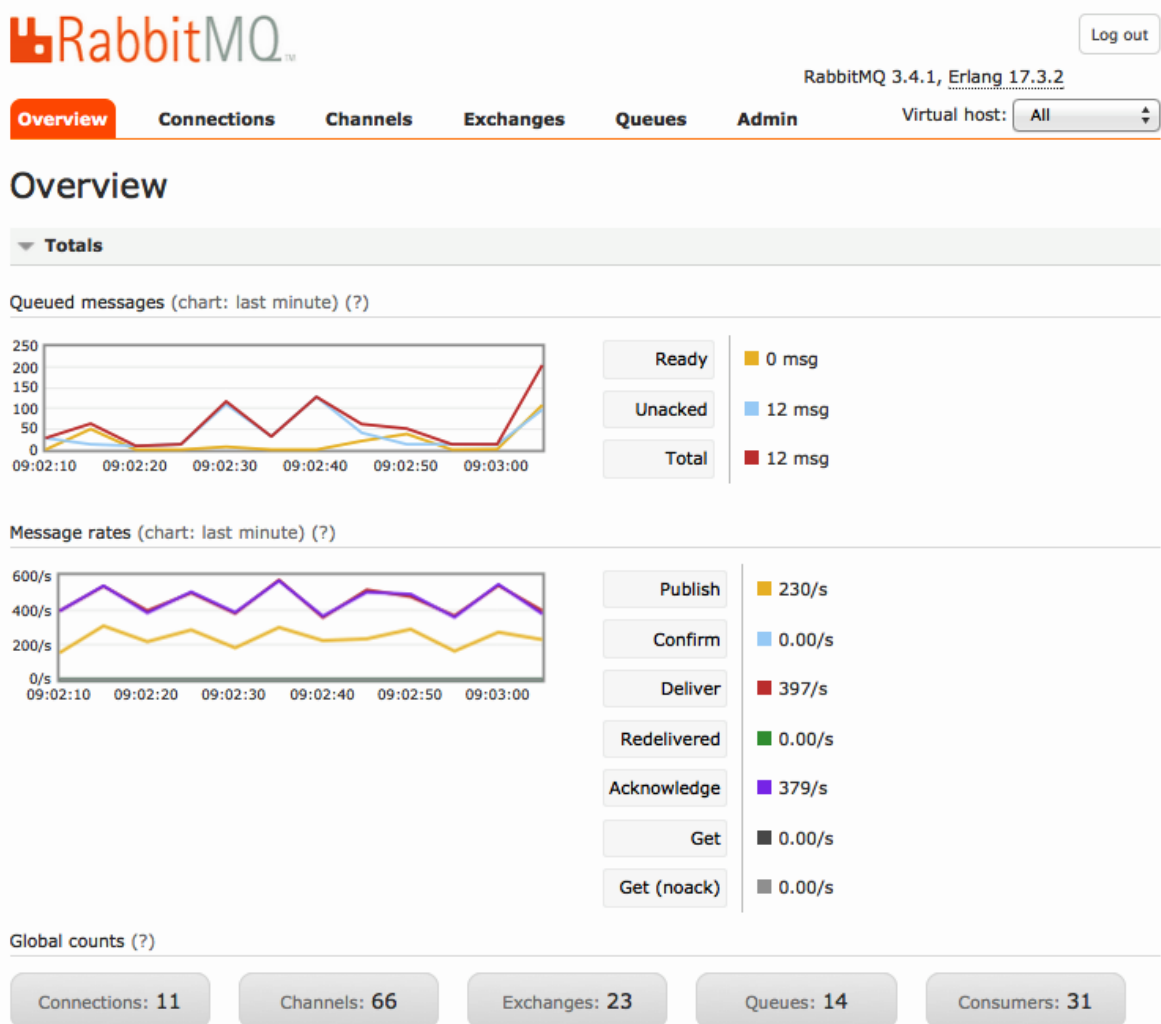
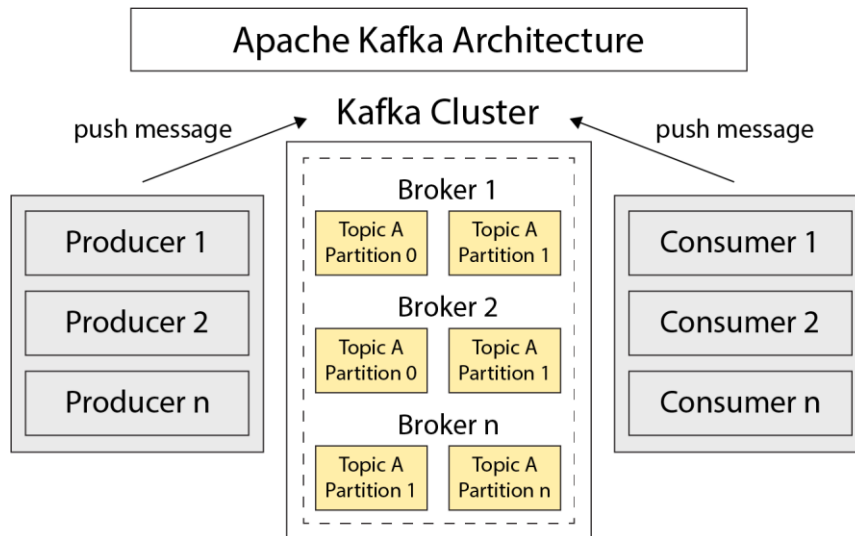
Chapter 7: Domain Driven Design for distributed systems



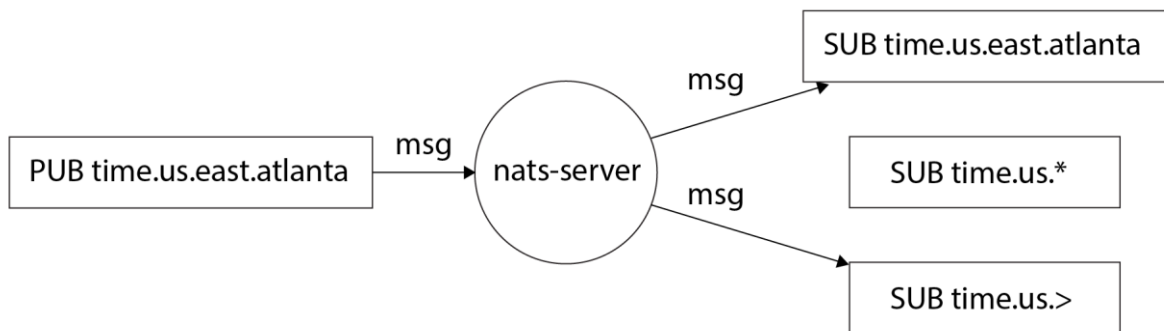
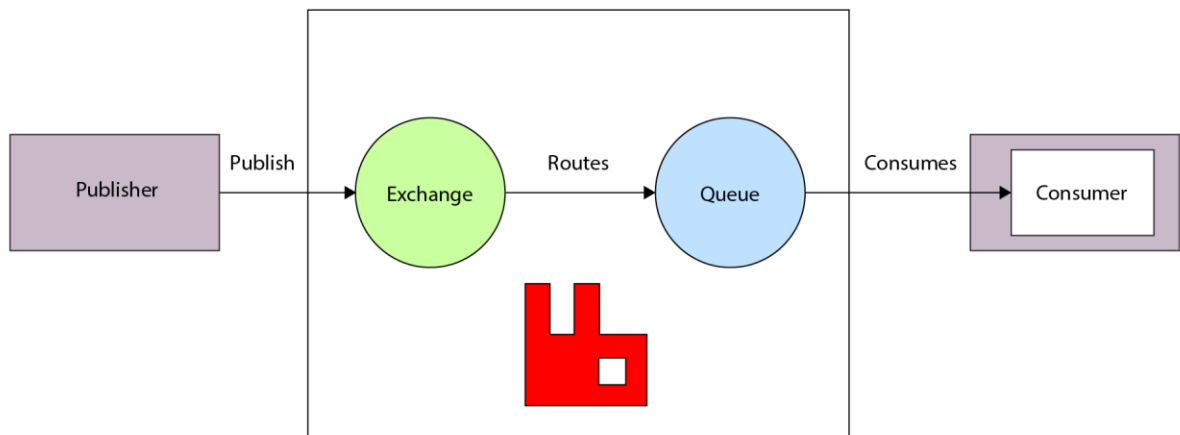




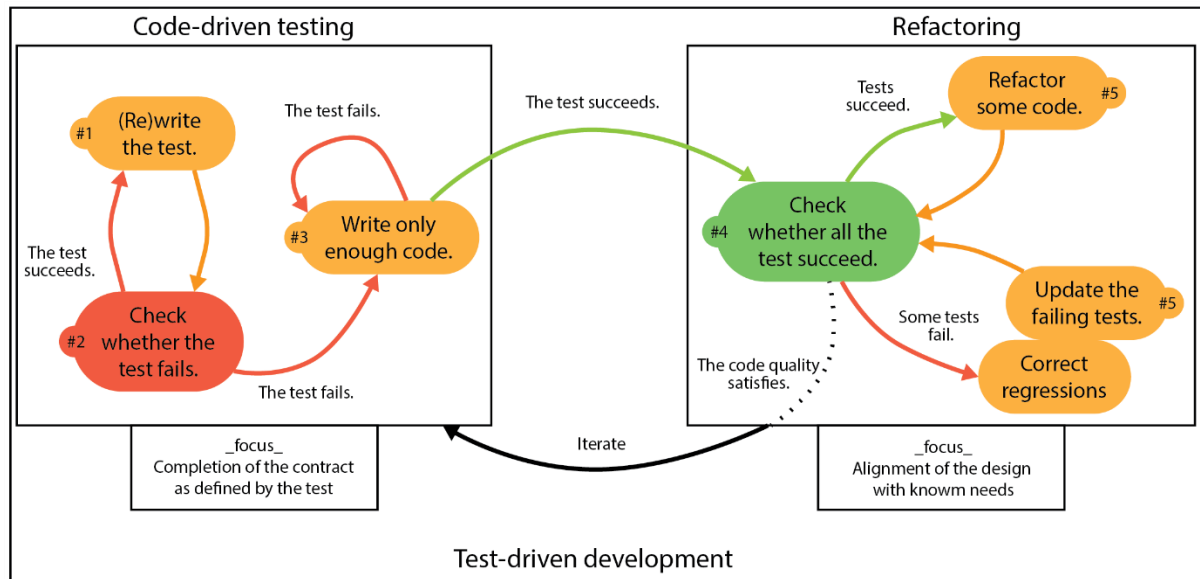




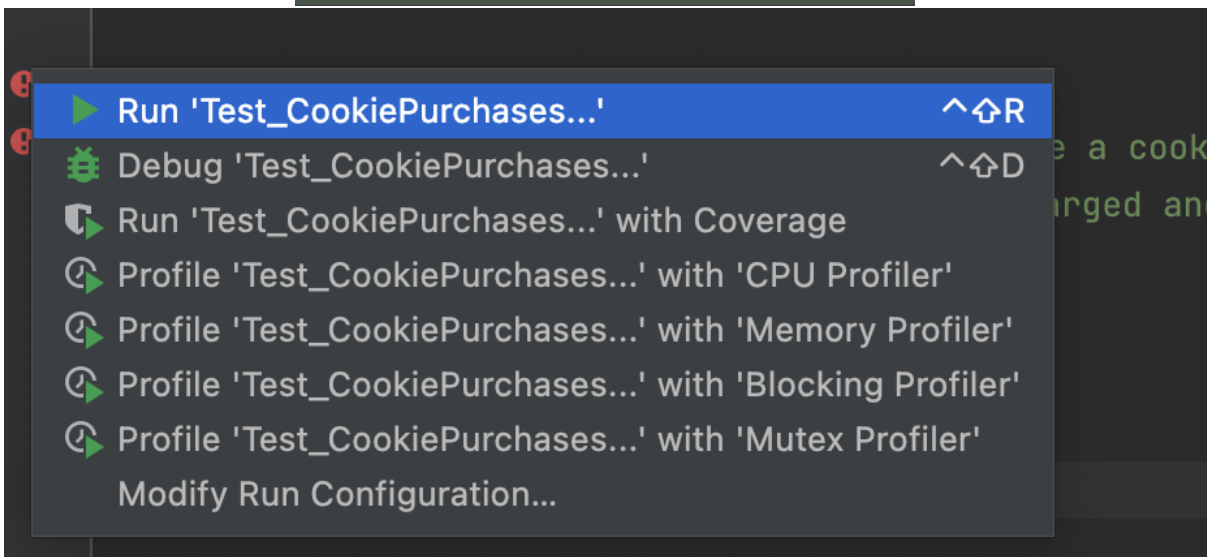
“Hello, world” example routing



Chapter 8: TDD, BDD and DDD

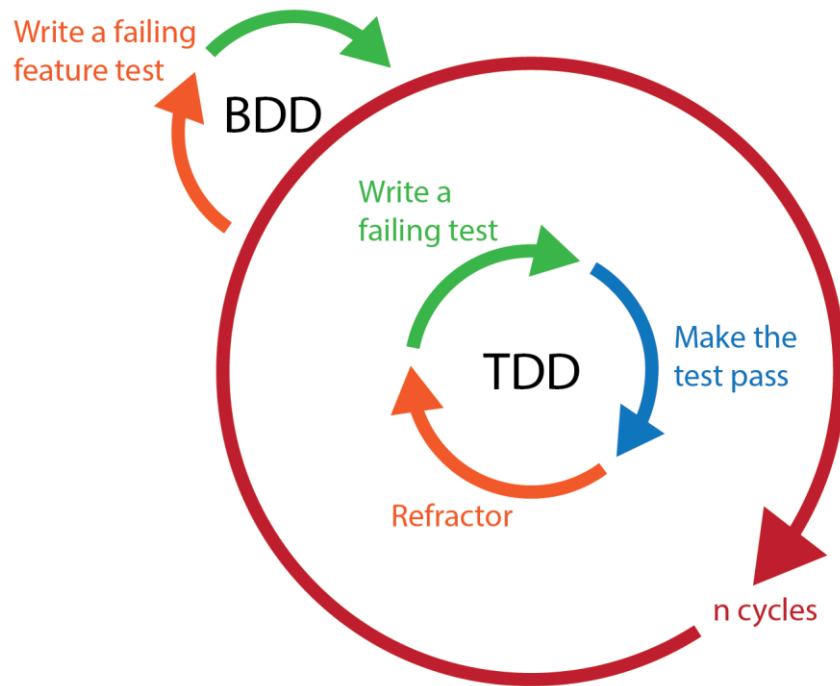


```
> chapter7
v chapter8
  cookies.go
  cookies_test.go
```



```
v chapter8
  v mocks
    cookies.go
```

```
cookies.go 100% statements
cookies.go 100% statements
cookies_test.go
```



▼ chapter8 100% files, 100% statements
features