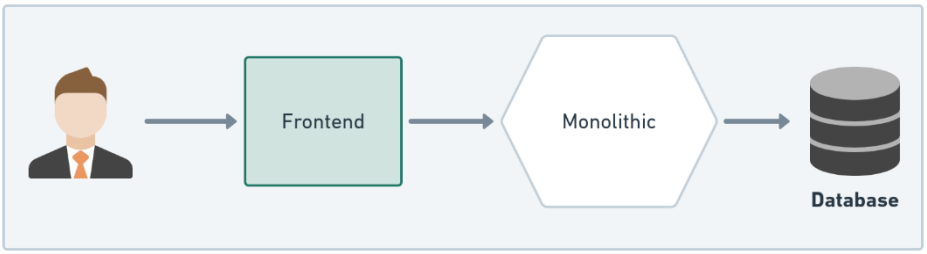
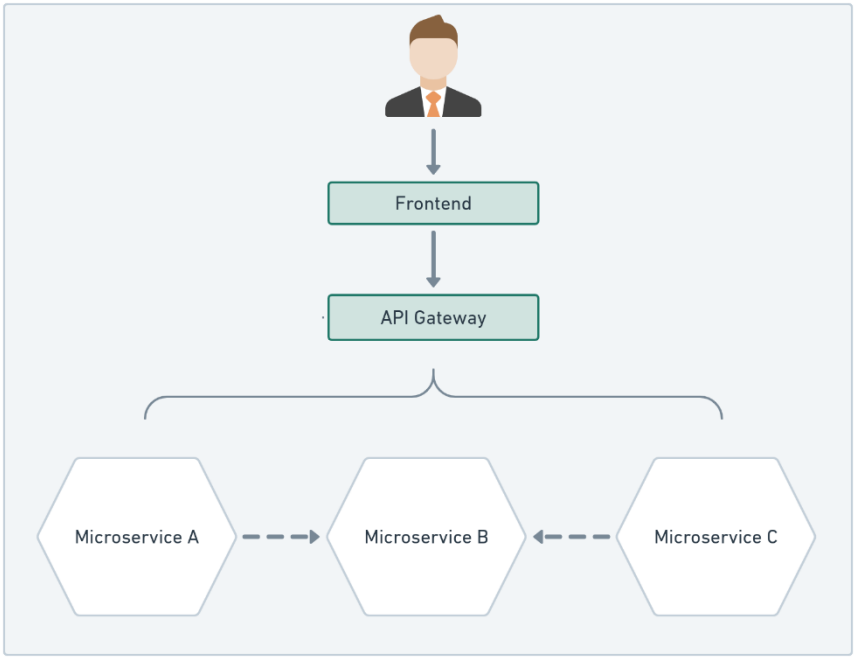
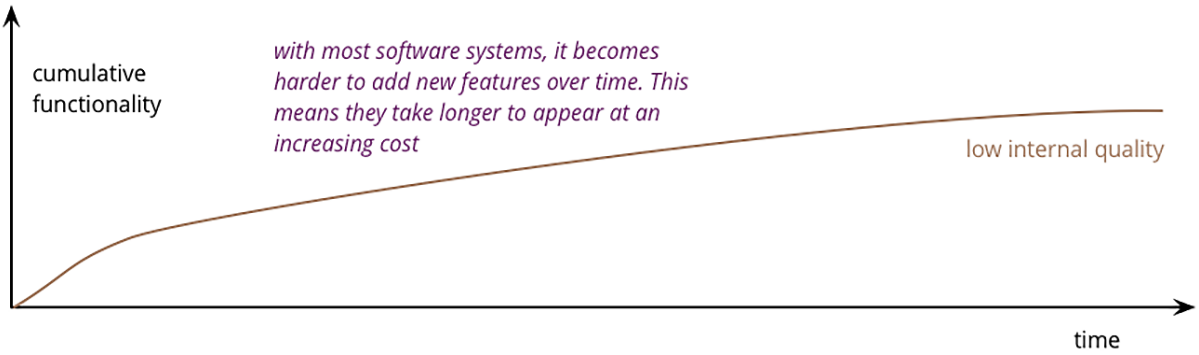
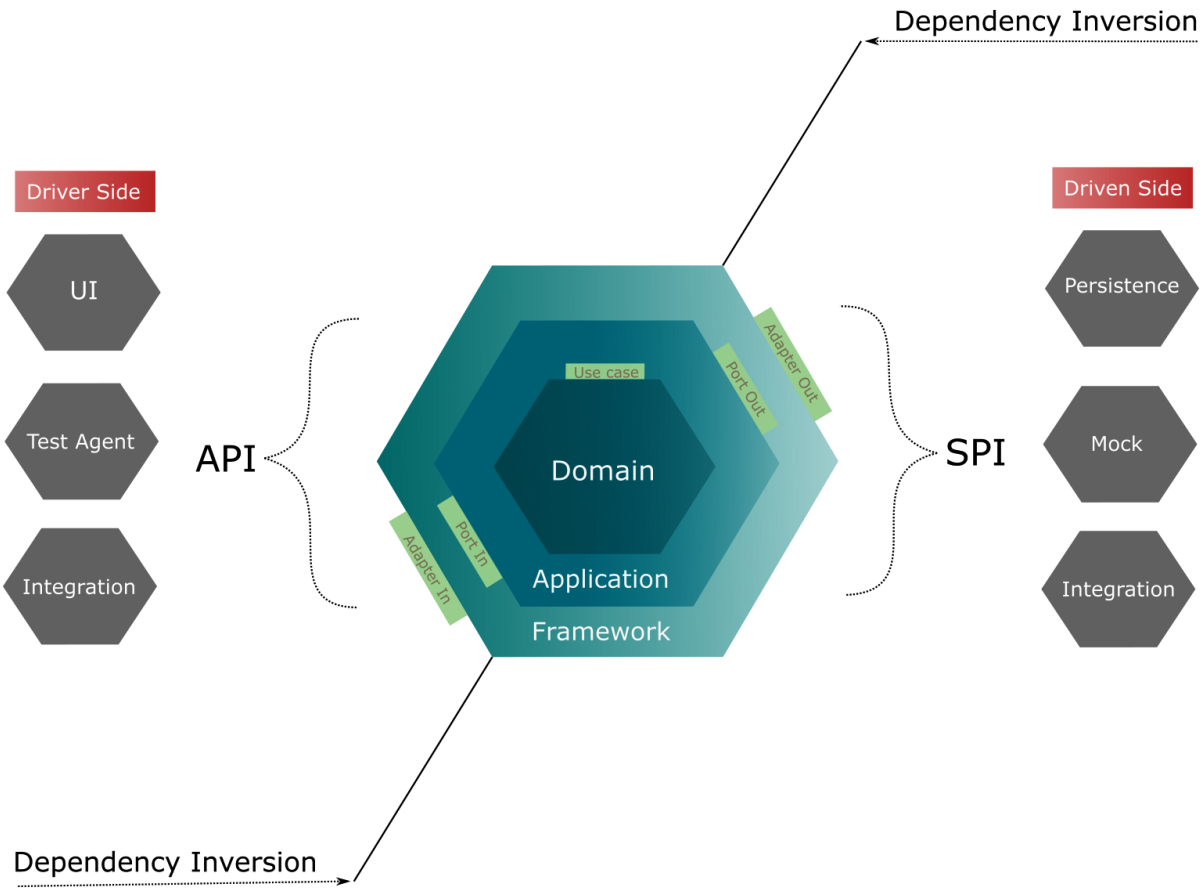
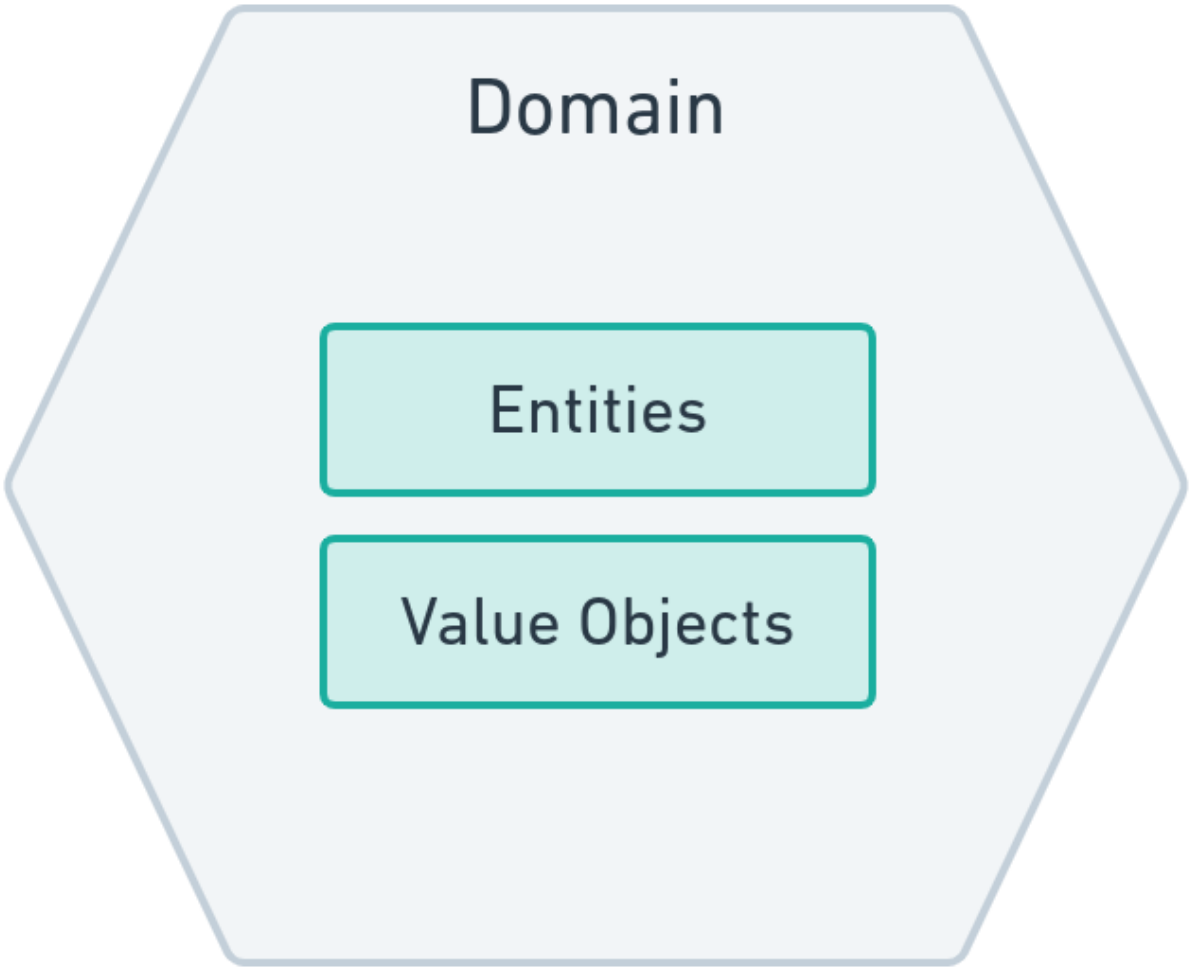
Chapter 1: Why Hexagonal Architecture?

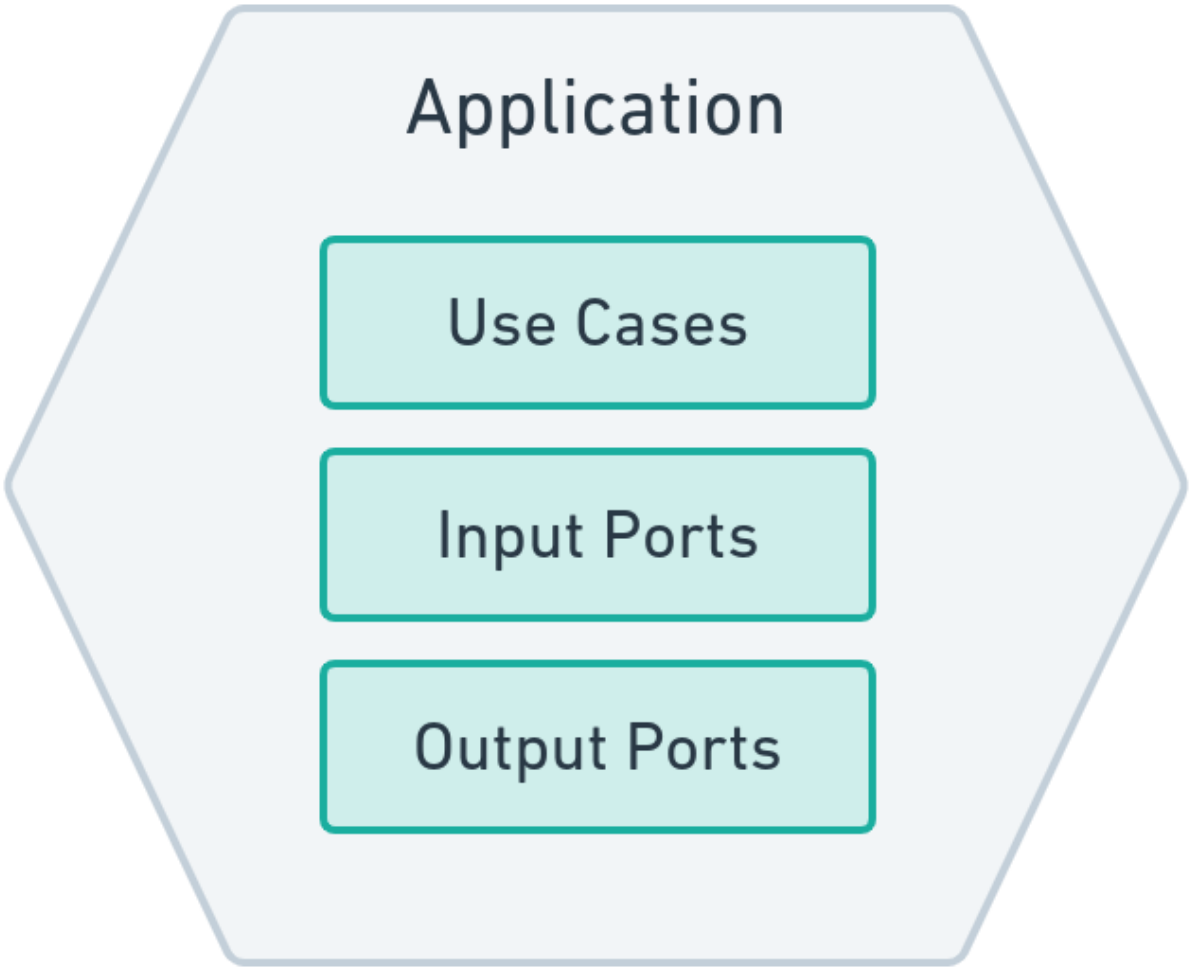


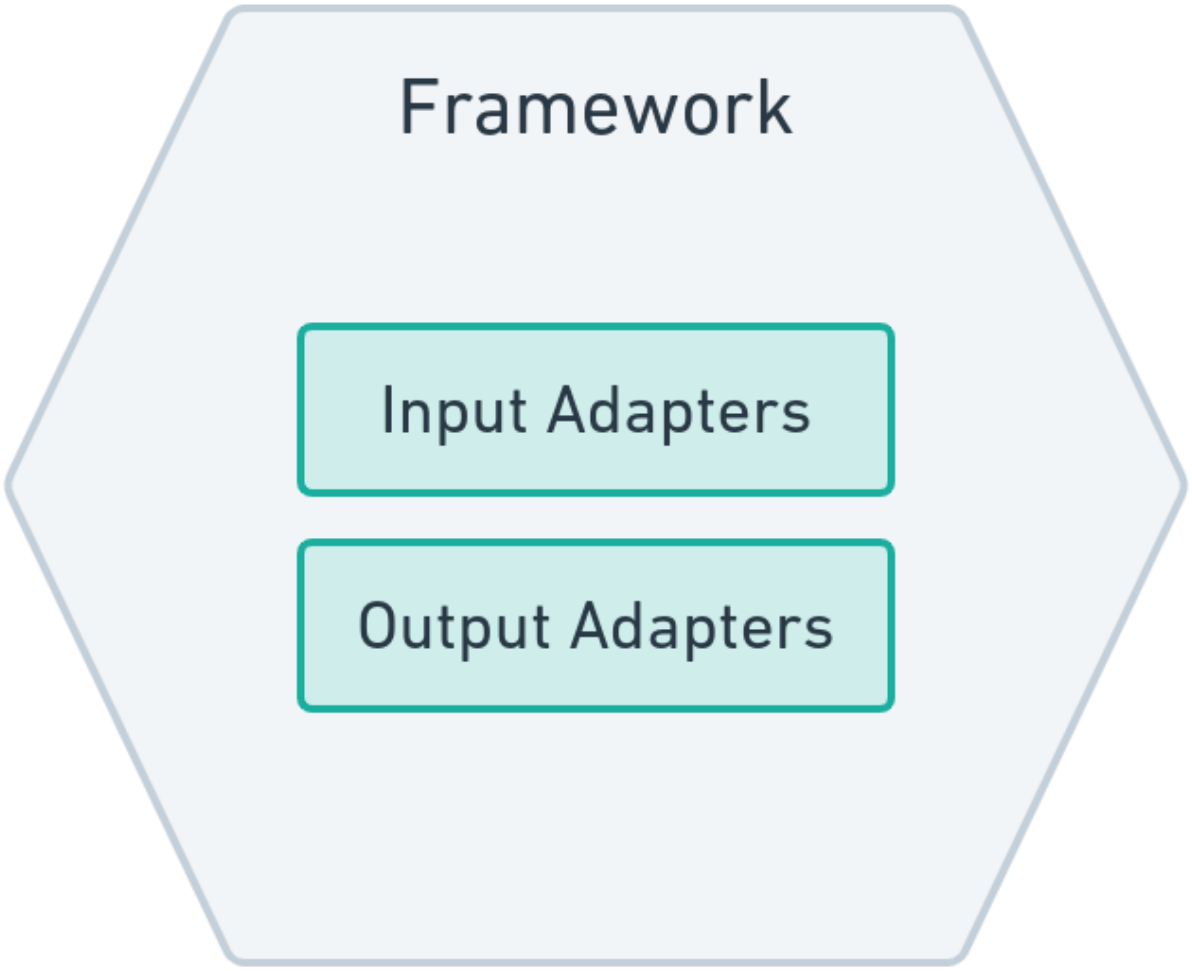


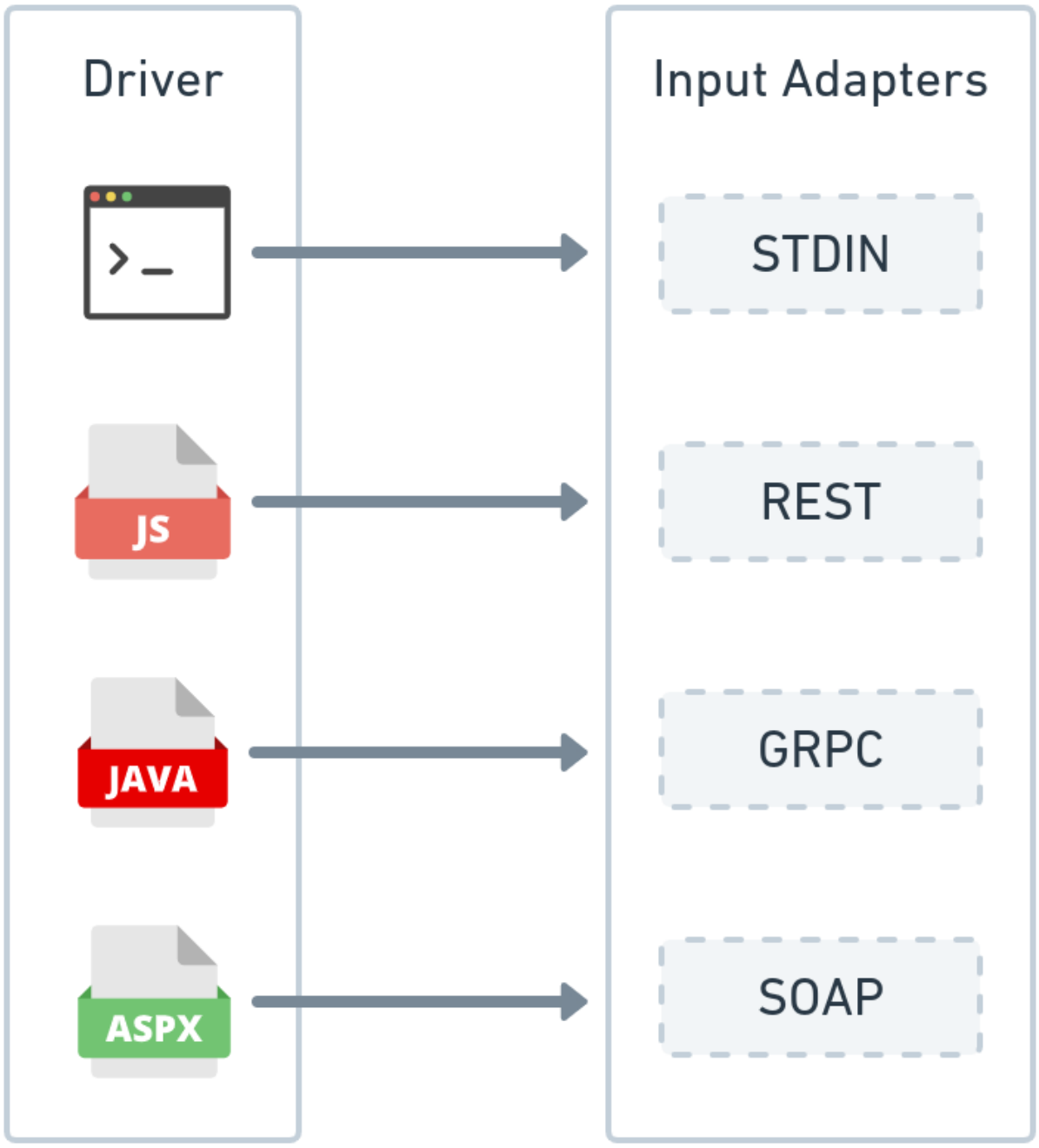


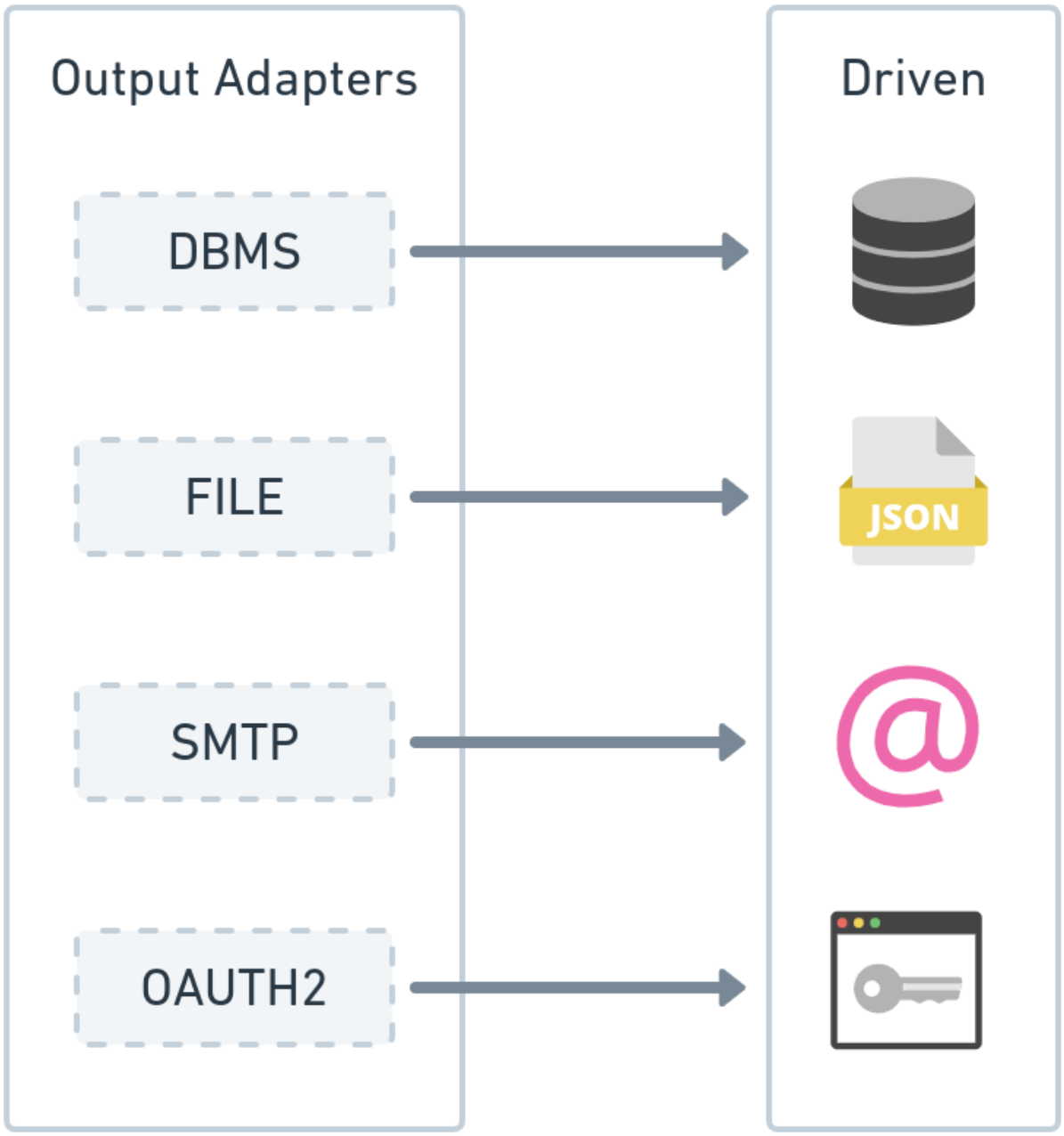

Dependency Inversion
Driver Side
Driven Side
UI
Test Agent
Integration
API
Use case
Domain
Application
Framework
Adapter In
Port In
Port Out
Adapter Out
SPI
Persistence
Mock
Integration
Dependency Inversion


Domain
Entities
Value Objects

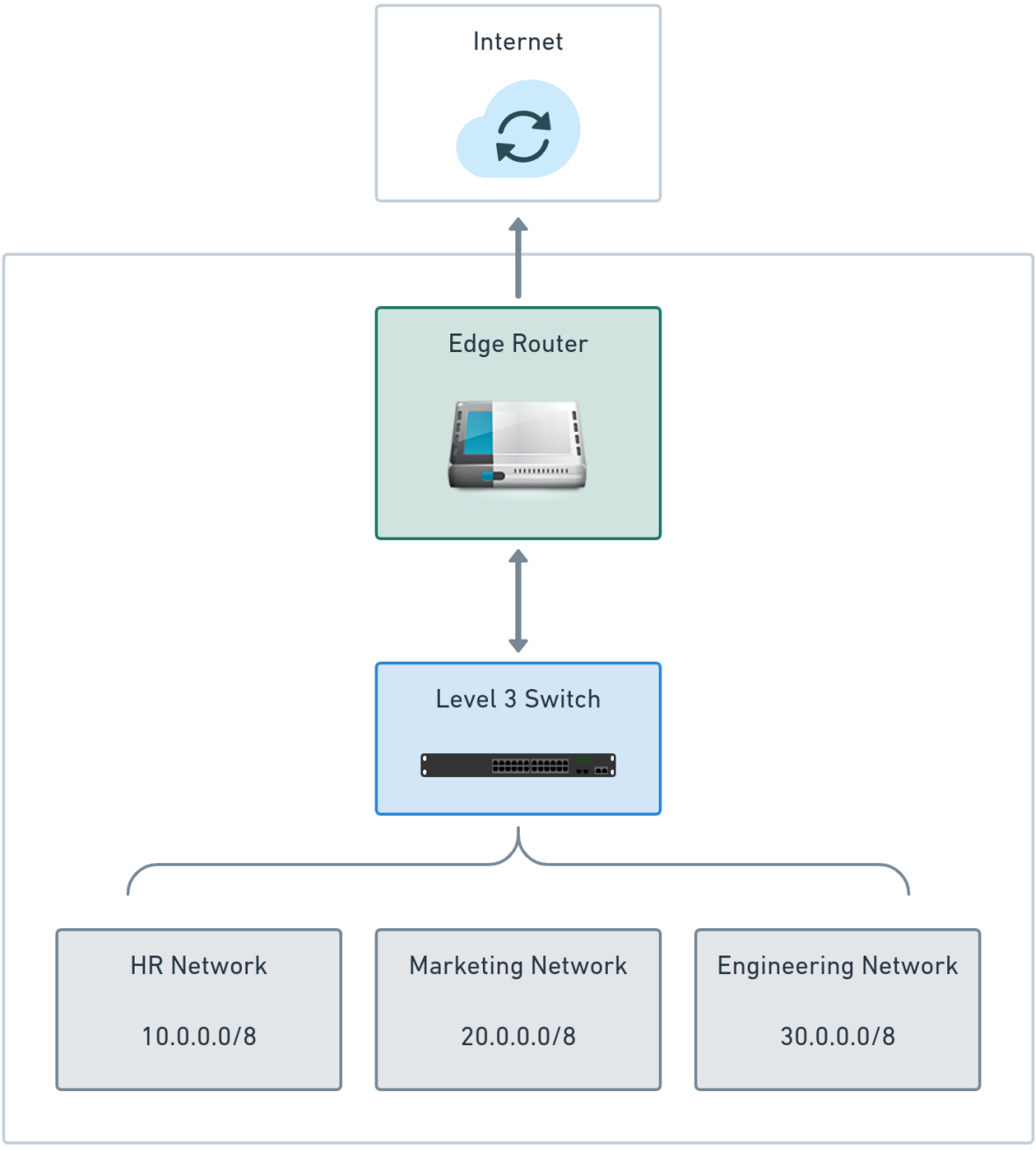

Application
Use Cases
Input Ports
Output Ports

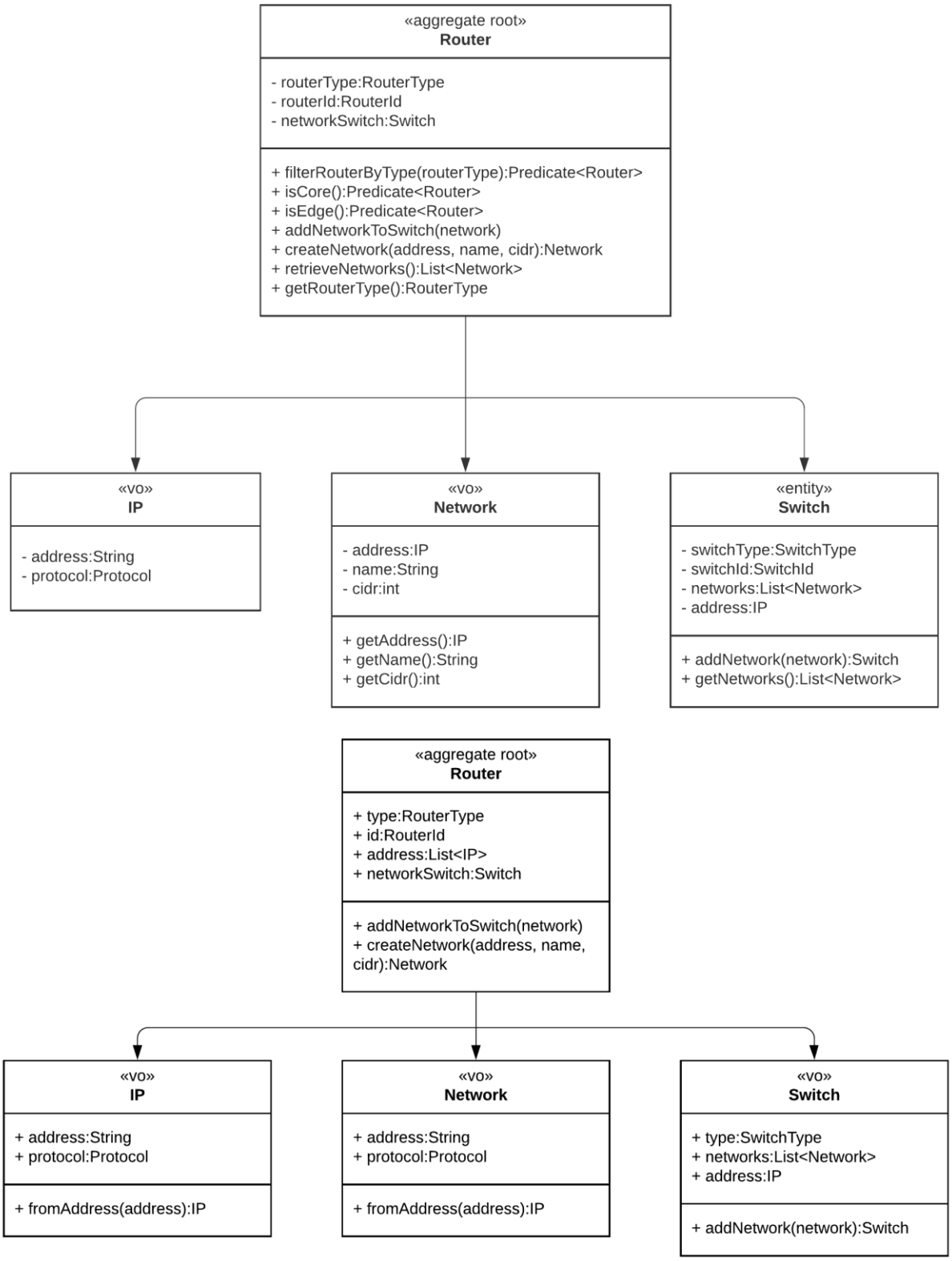

Framework
Input Adapters
Output Adapters


Driver
Input Adapters
STDIN
JS
REST
JAVA
GRPC
ASPX
SOAP

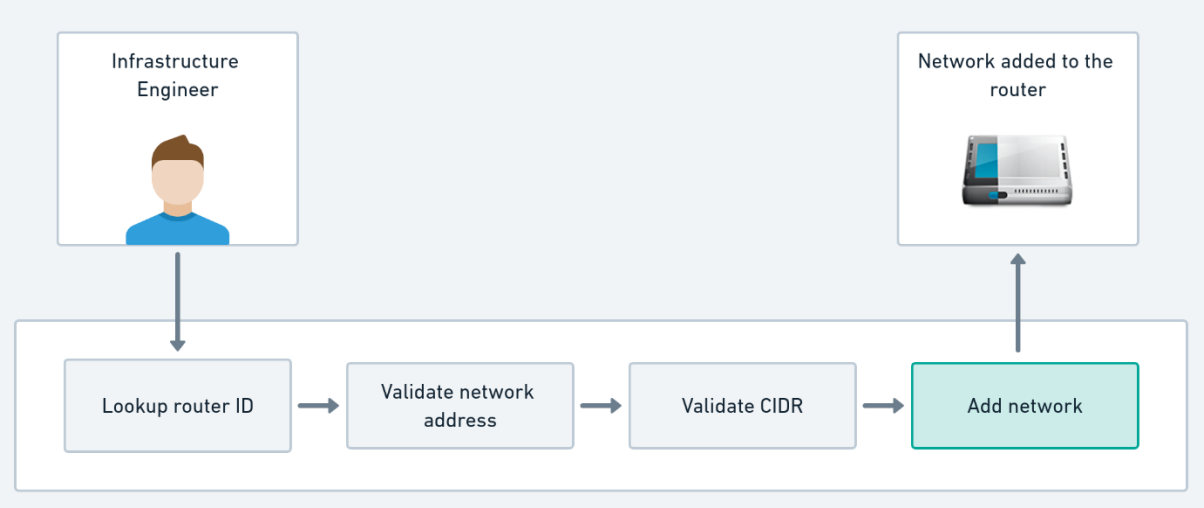

Output Adapters
DBMS
FILE
SMTP
OAUTH2
Driven
JSON

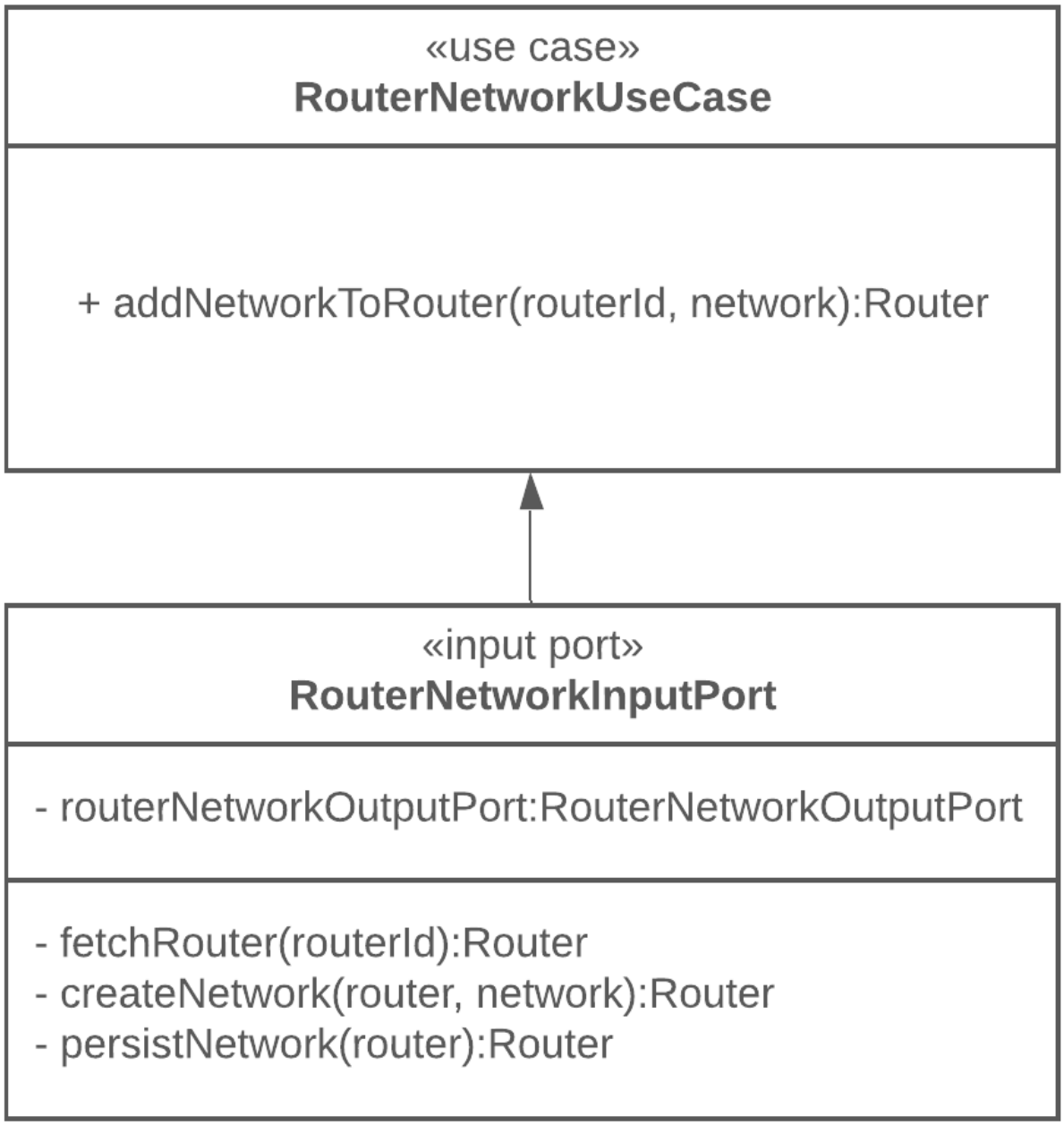
Chapter 2: Wrapping Business Rules inside Domain Hexagon

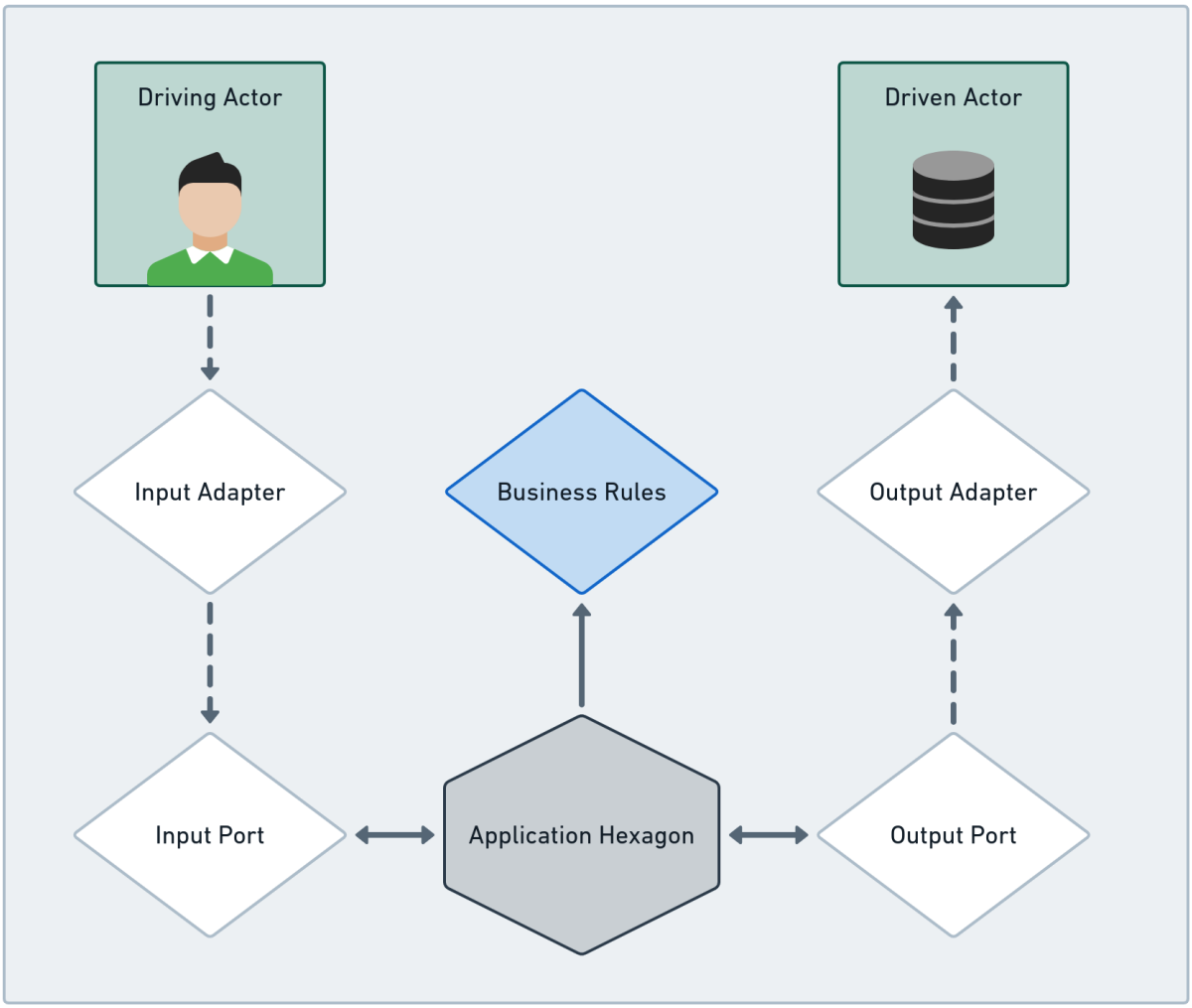


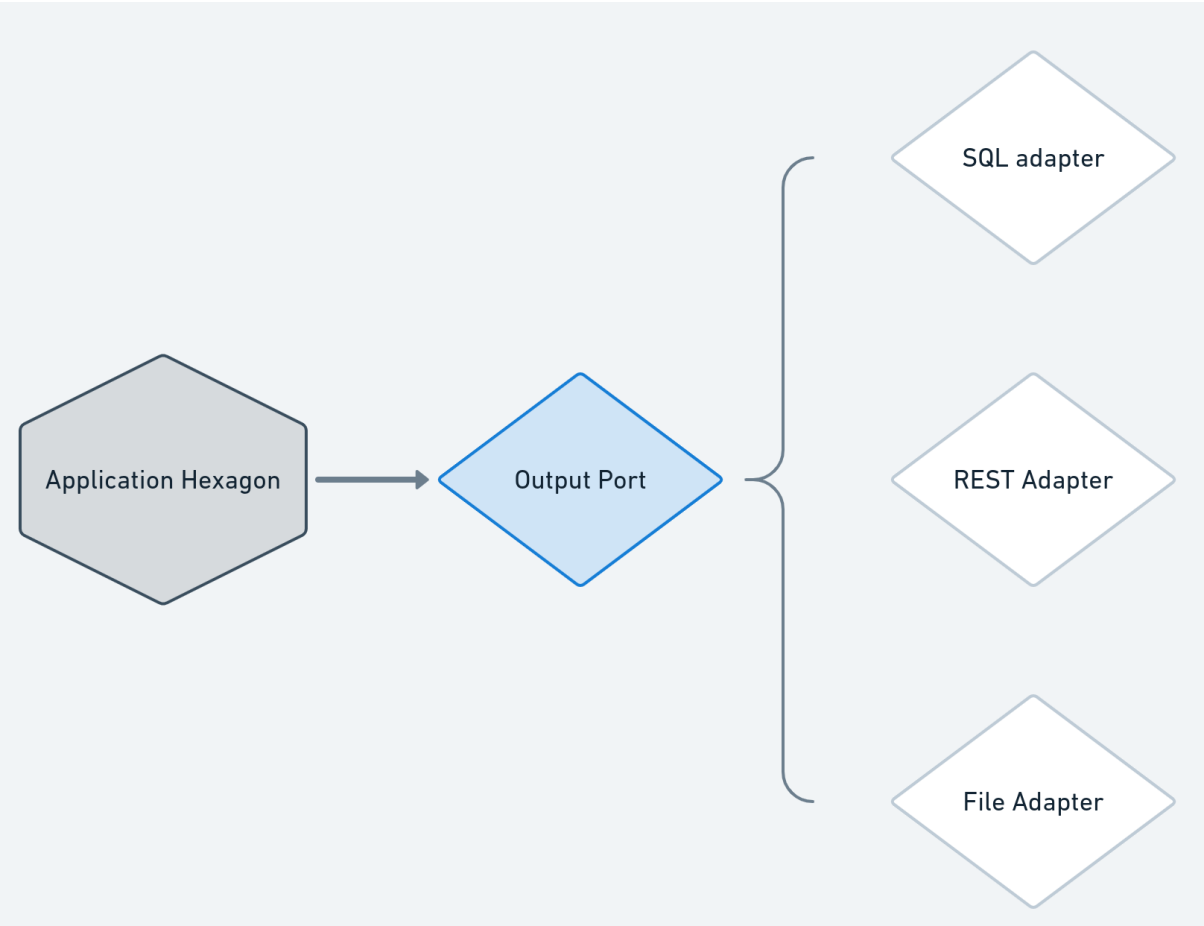

«aggregate root»
Router
- routerType:RouterType
- routerId:RouterId
- networkSwitch:Switch
+ filterRouterByType(routerType):Predicate<Router>
+ isCore():Predicate<Router>
+ isEdge():Predicate<Router>
+ addNetworkToSwitch(network)
+ createNetwork(address, name, cidr):Network
+ retrieveNetworks():List<Network>
+ getRouterType():RouterType
«vo»
IP
- address:String
- protocol:Protocol
«vo»
Network
- address:IP
- name:String
- cidr:int
+ getAddress():IP
+ getName():String
+ getCidr():int
«entity»
Switch
- switchType:SwitchType
- switchId:SwitchId
- networks:List<Network>
- address:IP
+ addNetwork(network):Switch
+ getNetworks():List<Network>
«aggregate root»
Router
+ type:RouterType
+ id:RouterId
+ address:List<IP>
+ networkSwitch:Switch
+ addNetworkToSwitch(network)
+ createNetwork(address, name, cidr):Network
«vo»
IP
+ address:String
+ protocol:Protocol
+ fromAddress(address):IP
«vo»
Network
+ address:String
+ protocol:Protocol
+ fromAddress(address):IP
«vo»
Switch
+ type:SwitchType
+ networks:List<Network>
+ address:IP
+ addNetwork(network):Switch

Chapter 3: Handling Behavior with Ports and Use Cases

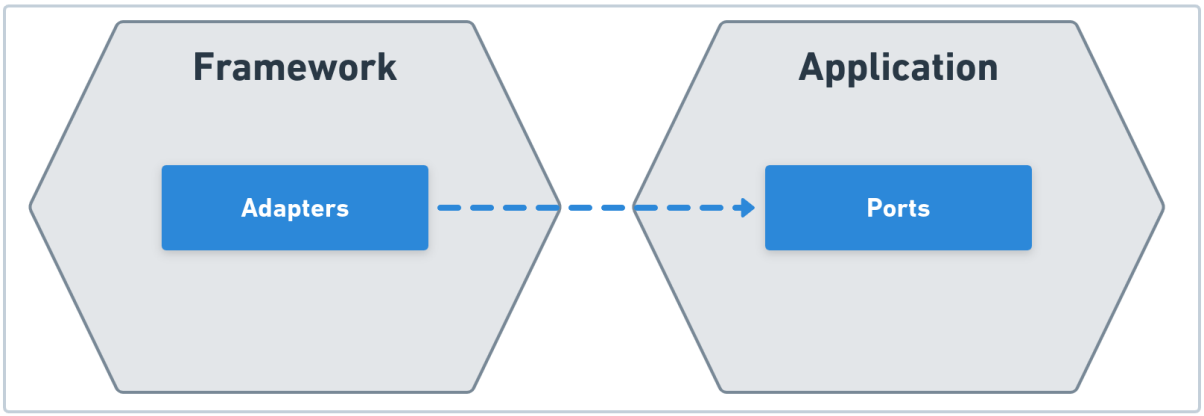


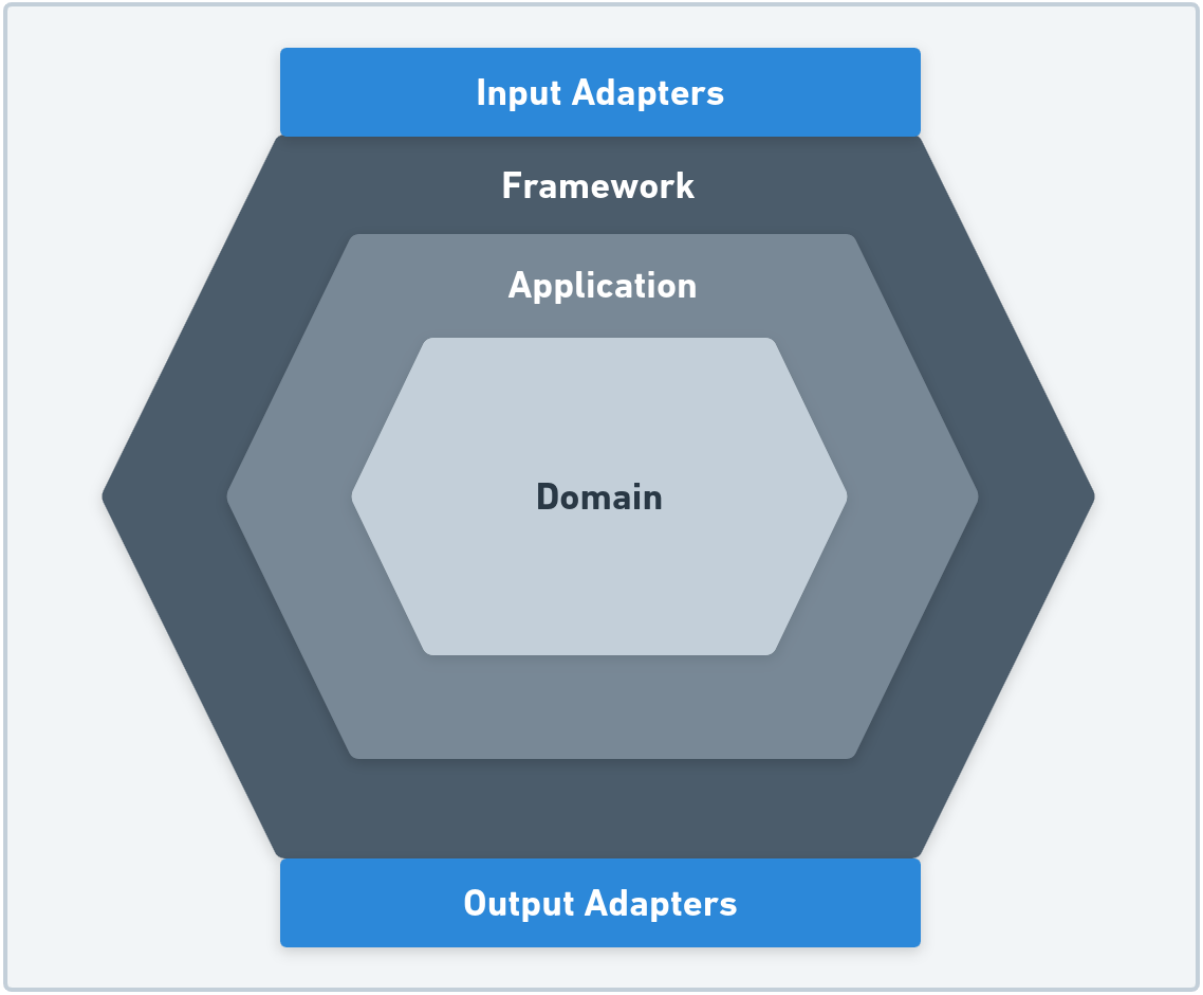

«use case»
RouterNetworkUseCase
+ addNetworkToRouter(routerId, network):Router
«input port»
RouterNetworkInputPort
- routerNetworkOutputPort:RouterNetworkOutputPort
- fetchRouter(routerId):Router
- createNetwork(router, network):Router
- persistNetwork(router):Router

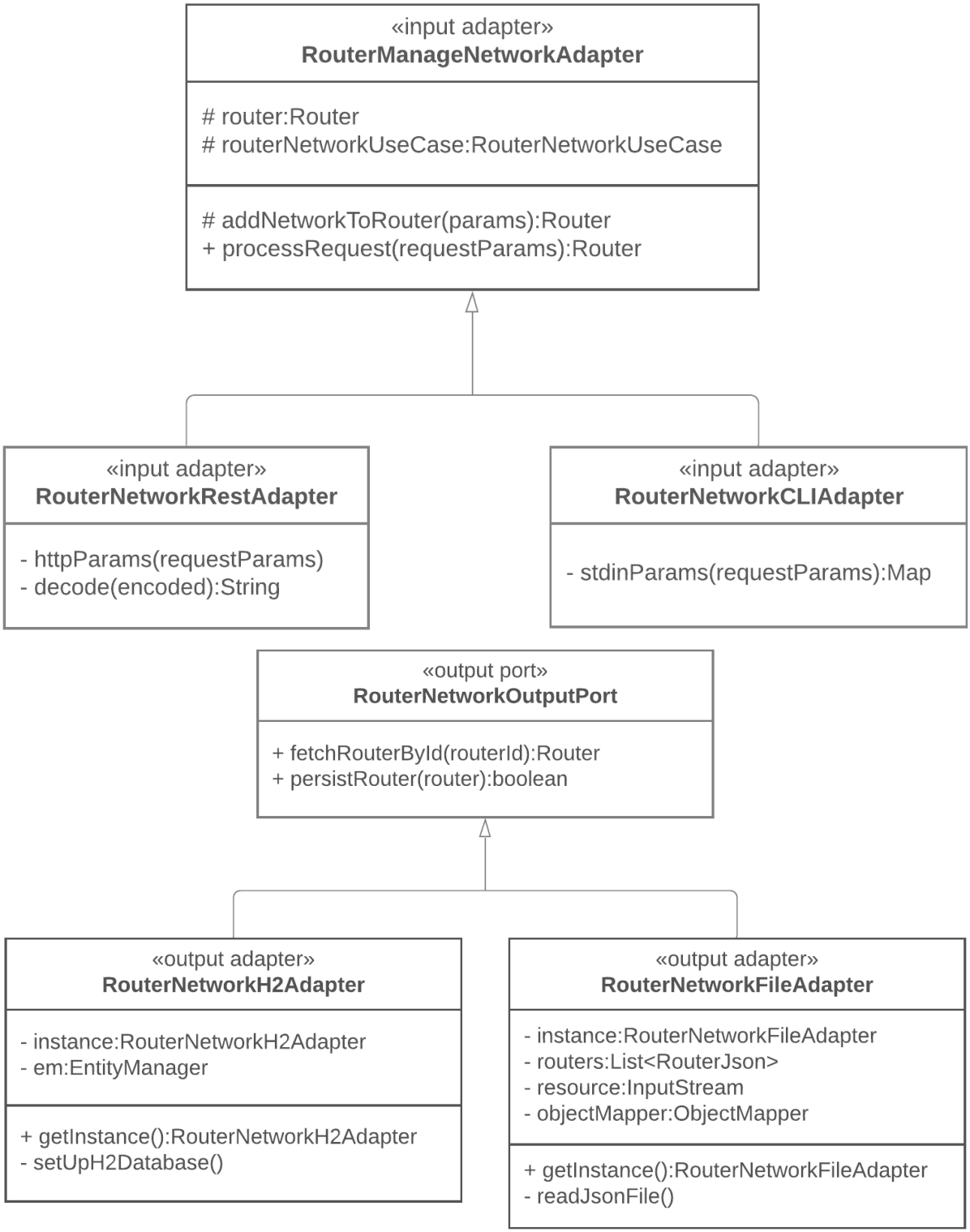

Driving Actor
Driven Actor
Input Adapter
Business Rules
Output Adapter
Input Port
Application Hexagon
Output Port

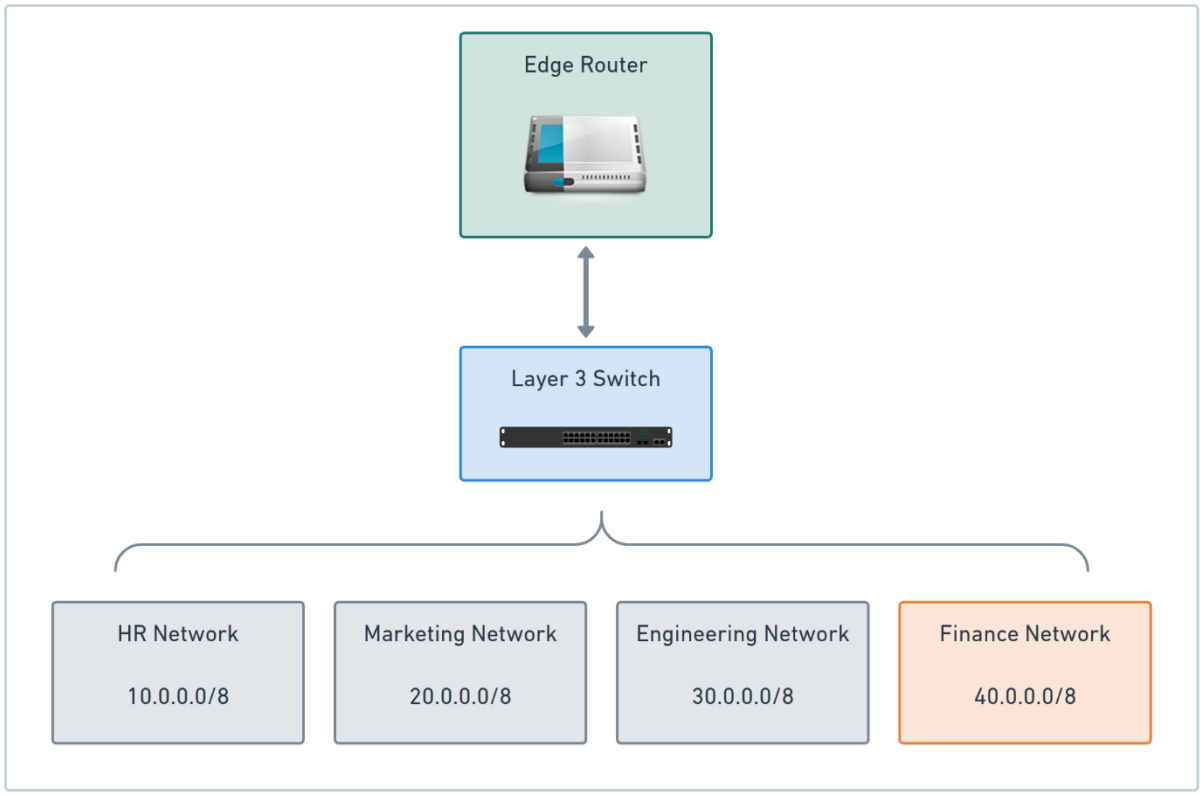

Application Hexagon
Output Port
SQL adapter
REST Adapter
File Adapter

Chapter 4: Creating Adapters to Interact with the Outside World

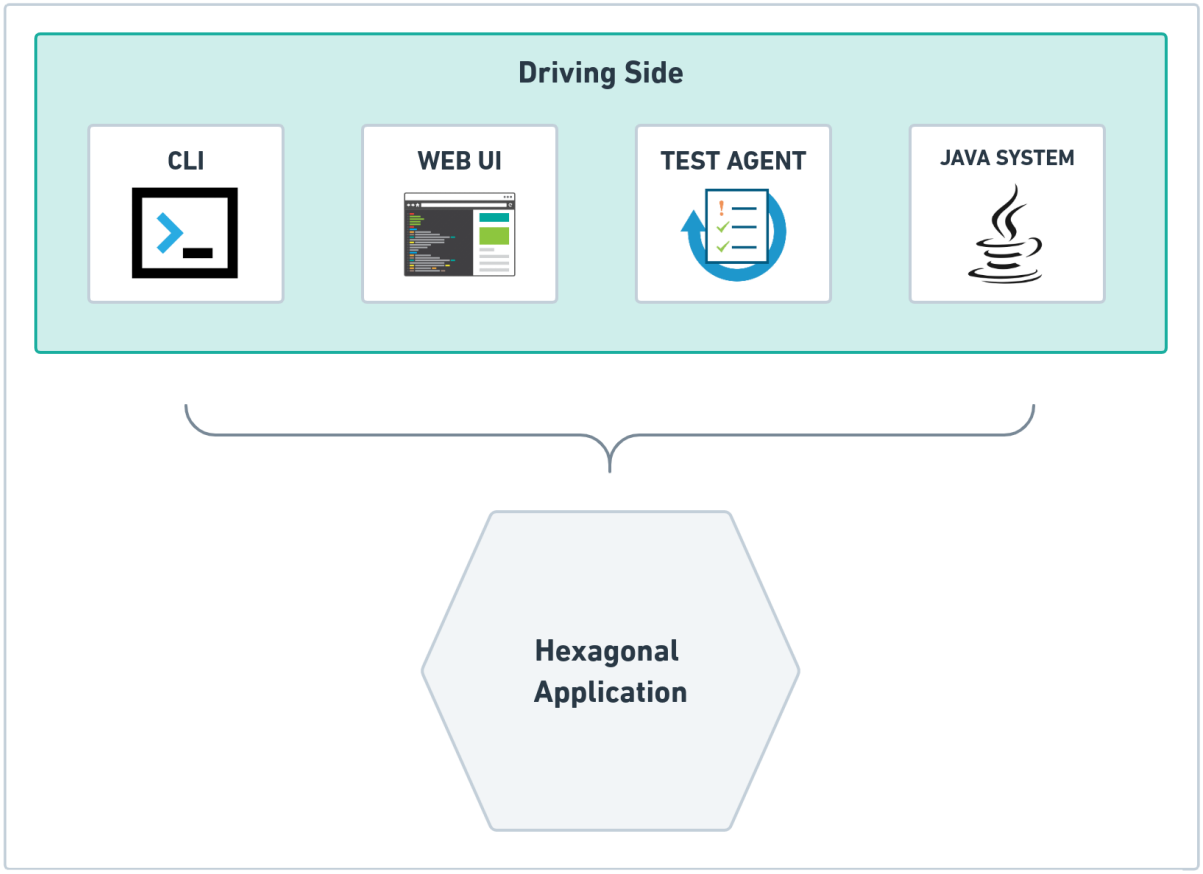


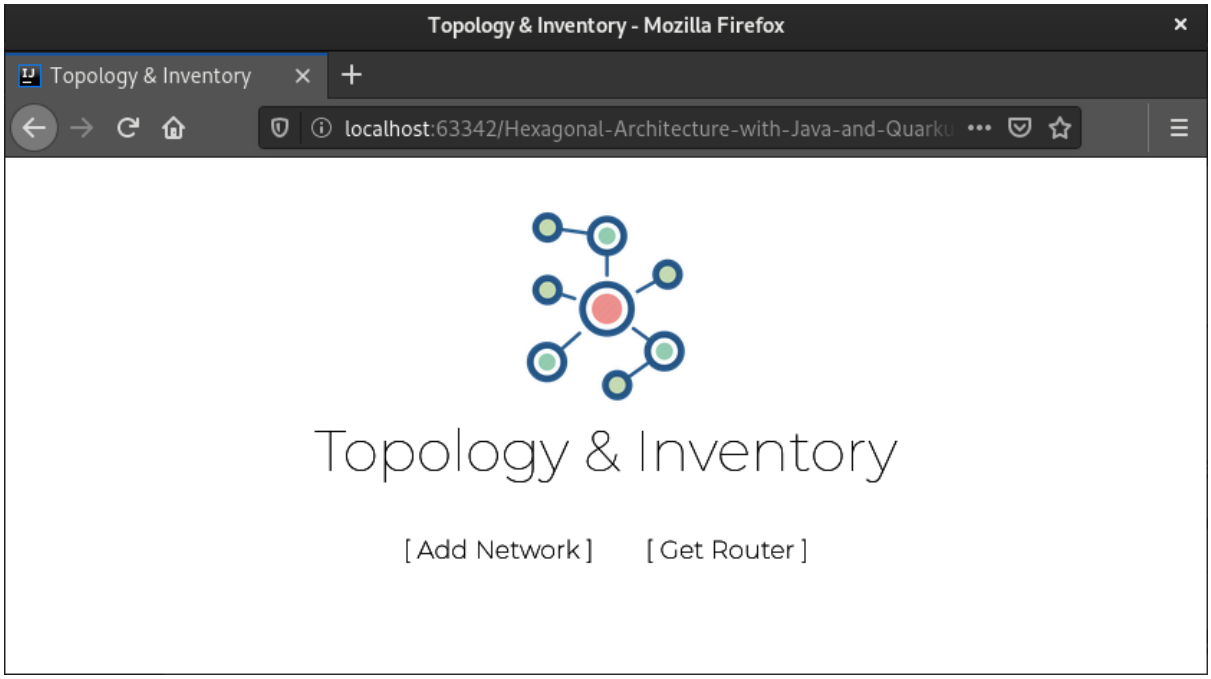


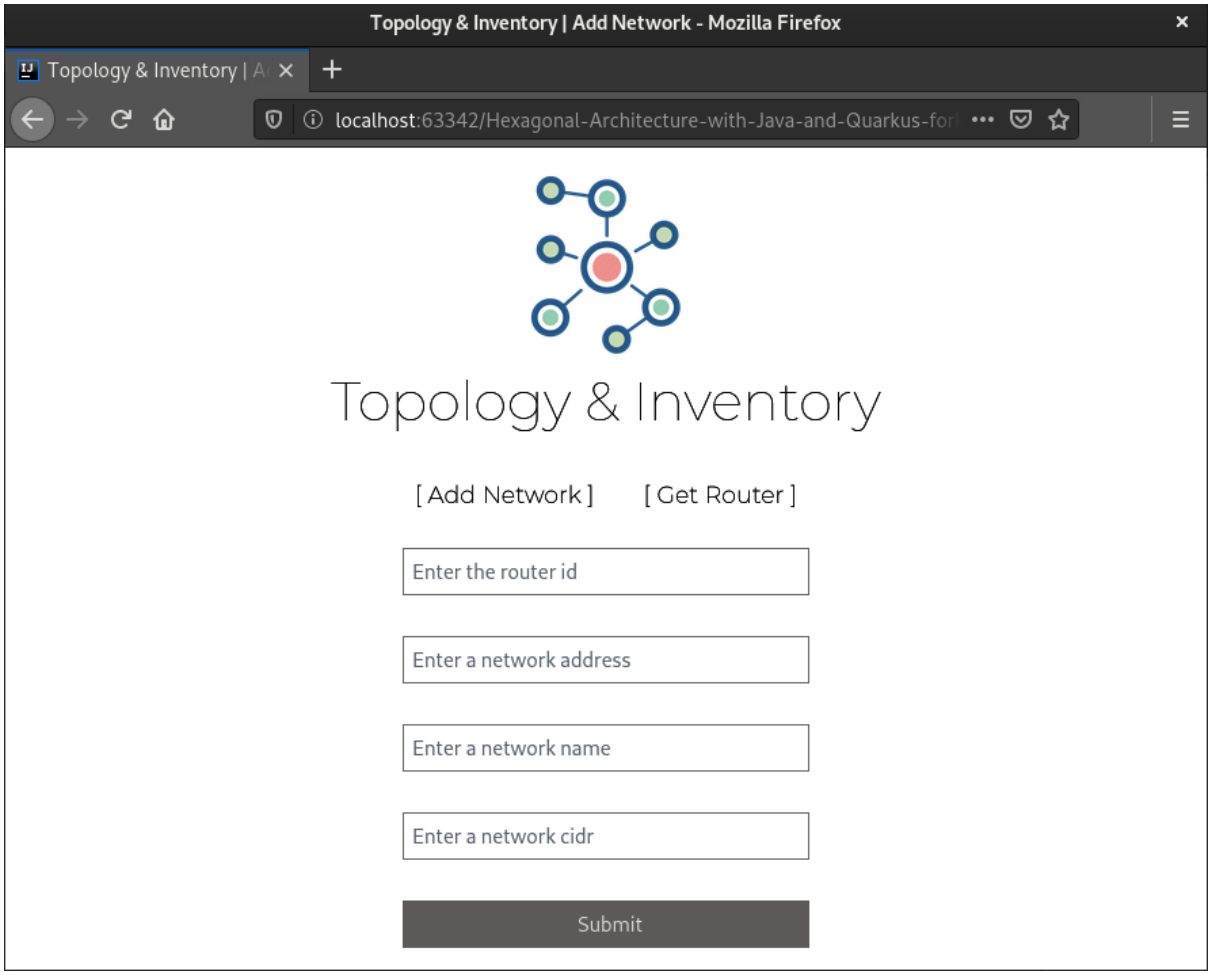

«input adapter»
RouterManageNetworkAdapter
router:Router
routerNetworkUseCase:RouterNetworkUseCase
addNetworkToRouter(params):Router
+ processRequest(requestParams):Router
«input adapter»
RouterNetworkRestAdapter
- httpParams(requestParams)
- decode(encoded):String
«input adapter»
RouterNetworkCLIAdapter
- stdinParams(requestParams):Map
«output port»
RouterNetworkOutputPort
+ fetchRouterById(routerId):Router
+ persistRouter(router):boolean
«output adapter»
RouterNetworkH2Adapter
- instance:RouterNetworkH2Adapter
- em:EntityManager
+ getInstance():RouterNetworkH2Adapter
- setUpH2Database()
«output adapter»
RouterNetworkFileAdapter
- instance:RouterNetworkFileAdapter
- routers:List<RouterJson>
- resource:InputStream
- objectMapper:ObjectMapper
+ getInstance():RouterNetworkFileAdapter
- readJsonFile()

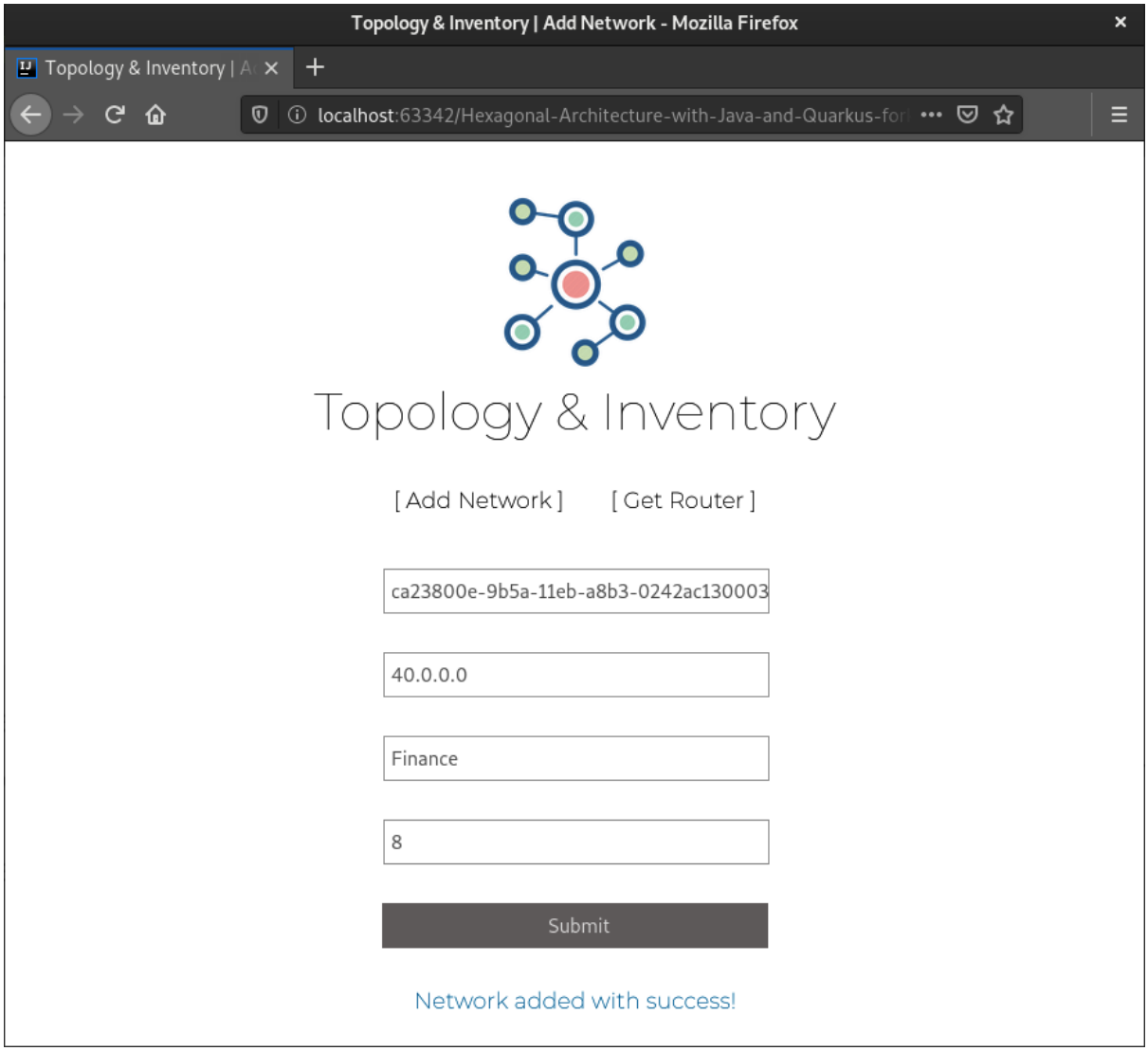

Edge Router
Layer 3 Switch
HR Network
10.0.0.0/8
Marketing Network
20.0.0.0/8
Engineering Network
30.0.0.0/8
Finance Network
40.0.0.0/8

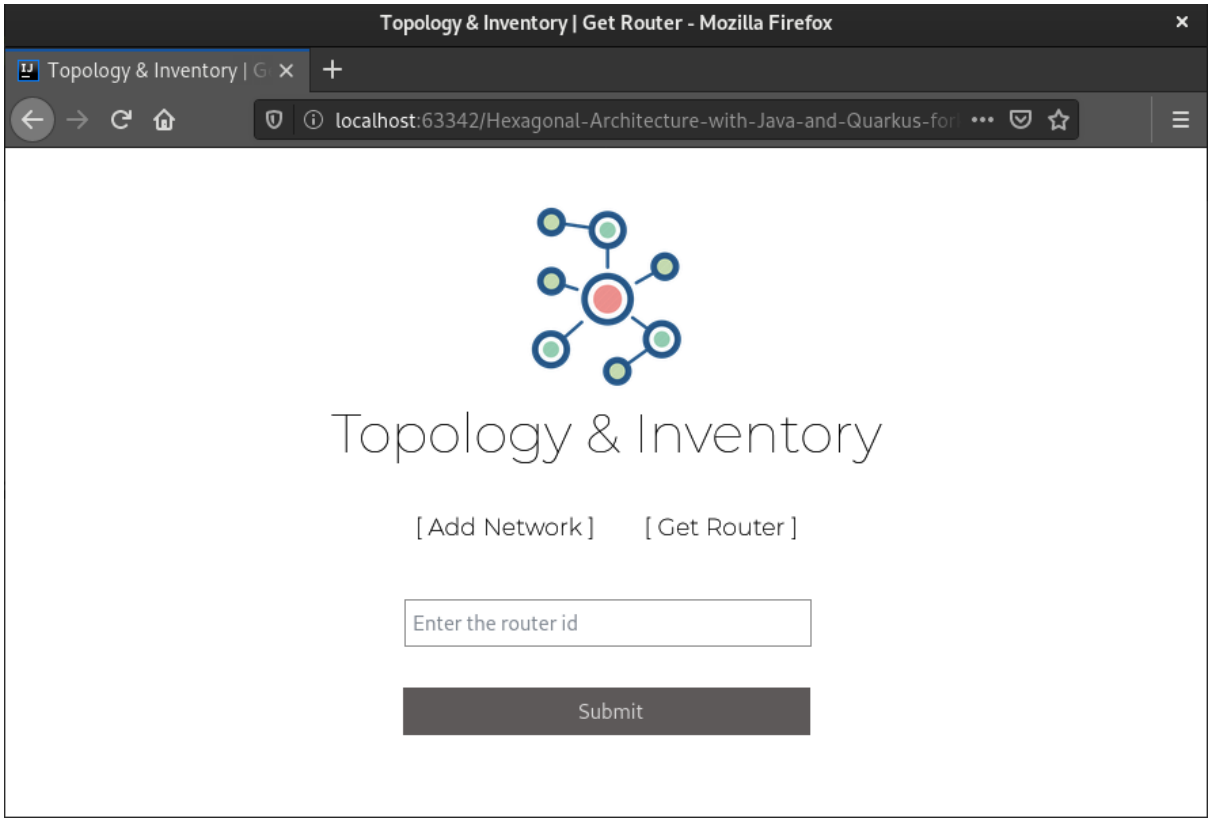
Chapter 5: Exploring the Nature of Driving and Driven Operations

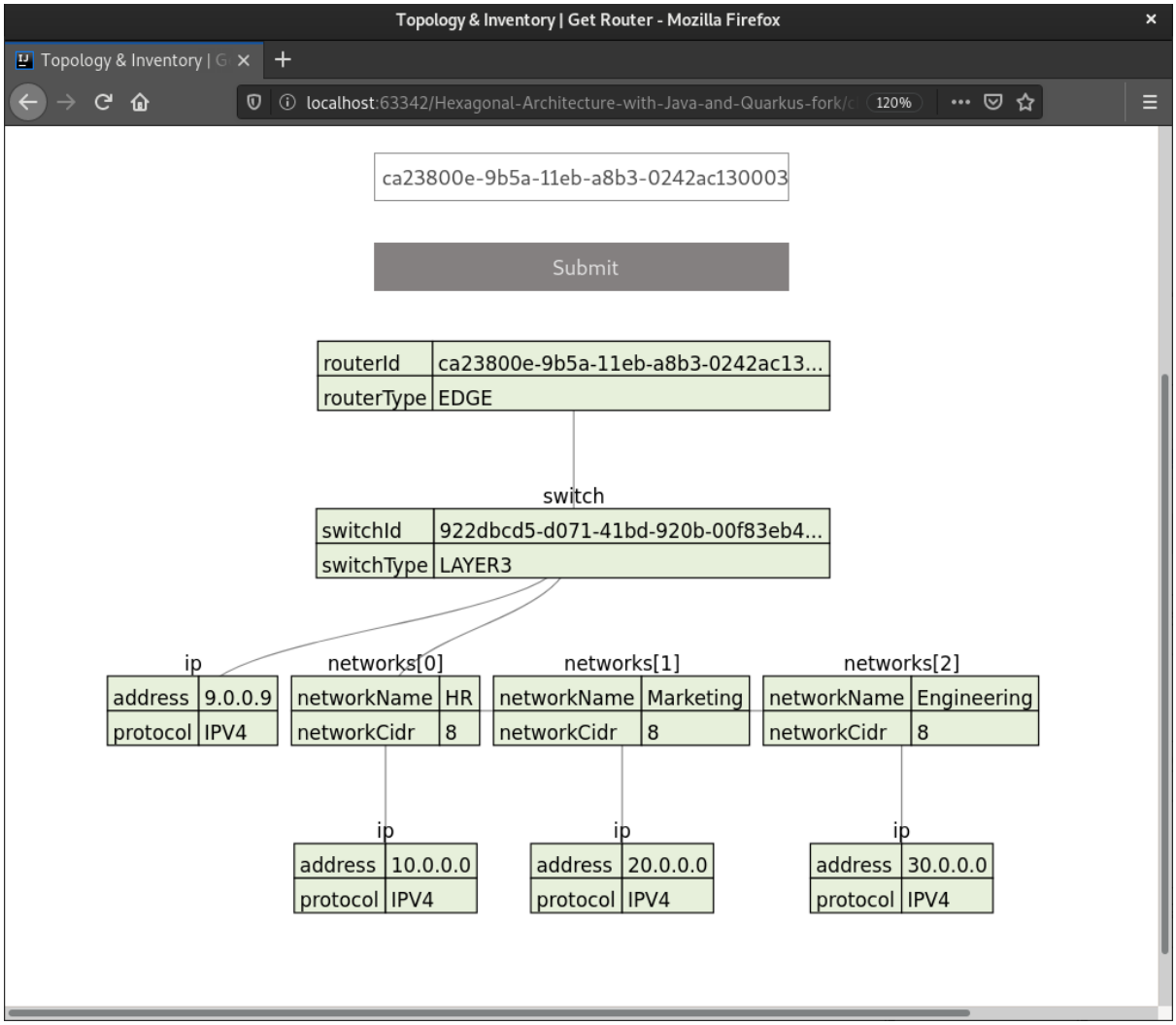


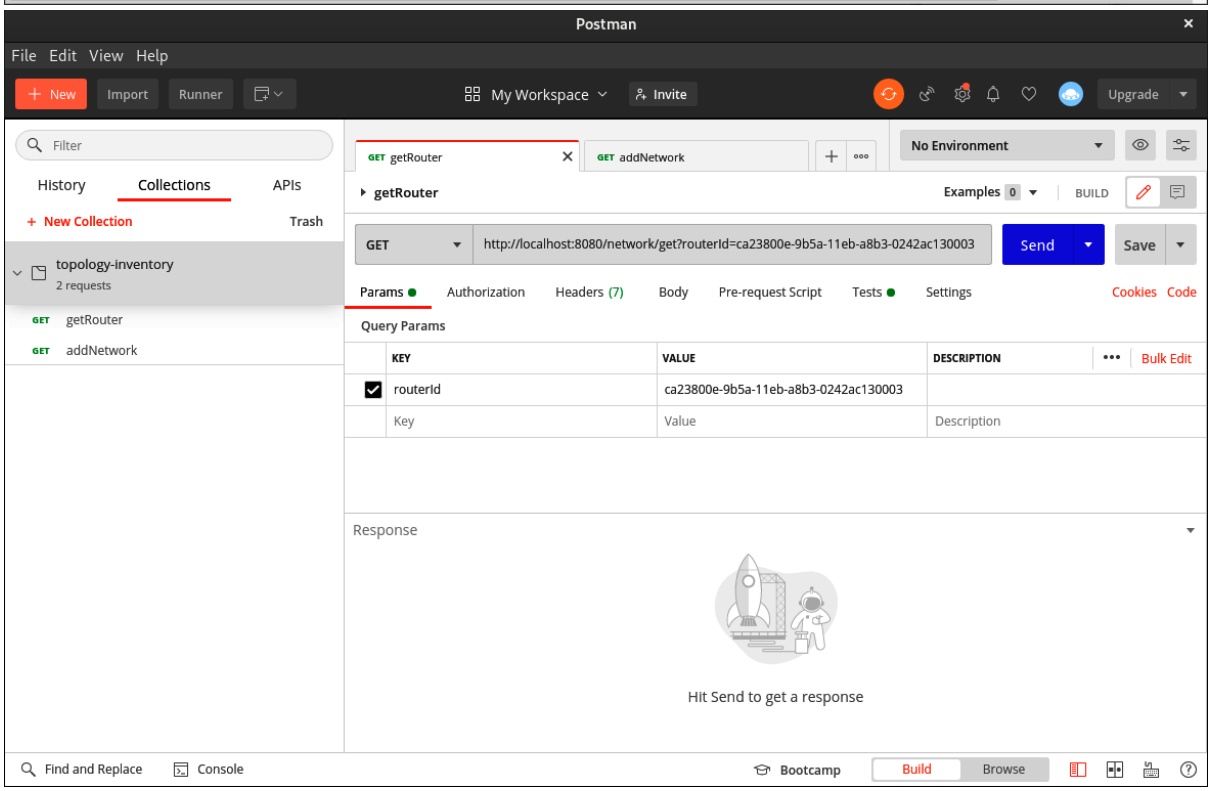



Topology & Inventory | Add Network - Mozilla Firefox
Topology & Inventory | A
localhost:63342/Hexagonal-Architecture-with-Java-and-Quarkus-for
Topology & Inventory
[Add Network]
[Get Router]
Enter the router id
Enter a network address
Enter a network name
Enter a network cidr
Submit


Topology & Inventory | Add Network - Mozilla Firefox
Topology & Inventory | A
localhost:63342/Hexagonal-Architecture-with-Java-and-Quarkus-for
Topology & Inventory
[Add Network]
[Get Router]
ca23800e-9b5a-11eb-a8b3-0242ac130003
40.0.0.0
Finance
8
Submit
Network added with success!


Topology & Inventory | Get Router - Mozilla Firefox
Topology & Inventory | G
localhost:63342/Hexagonal-Architecture-with-Java-and-Quarkus-for
Topology & Inventory
[Add Network]
[Get Router]
Enter the router id
Submit

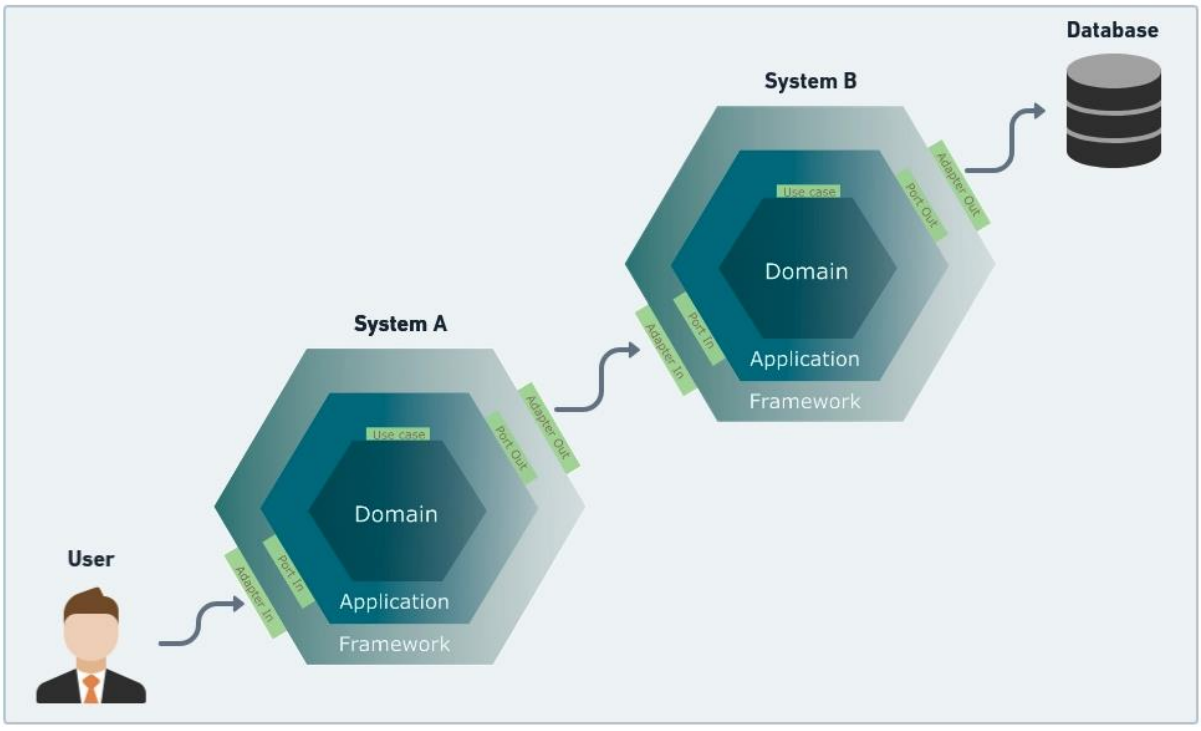

Topology & Inventory | Get Router - Mozilla Firefox
Topology & Inventory | G
localhost:63342/Hexagonal-Architecture-with-Java-and-Quarkus-fork/c
120%
ca23800e-9b5a-11eb-a8b3-0242ac130003
Submit
routerId
ca23800e-9b5a-11eb-a8b3-0242ac13...
routerType
EDGE
switch
switchId
922dbcd5-d071-41bd-920b-00f83eb4...
switchType
LAYER3
ip
address
9.0.0.9
protocol
IPV4
networks[0]
networkName
HR
networkCidr
8
networks[1]
networkName
Marketing
networkCidr
8
networks[2]
networkName
Engineering
networkCidr
8
ip
address
10.0.0.0
protocol
IPV4
ip
address
20.0.0.0
protocol
IPV4
ip
address
30.0.0.0
protocol
IPV4

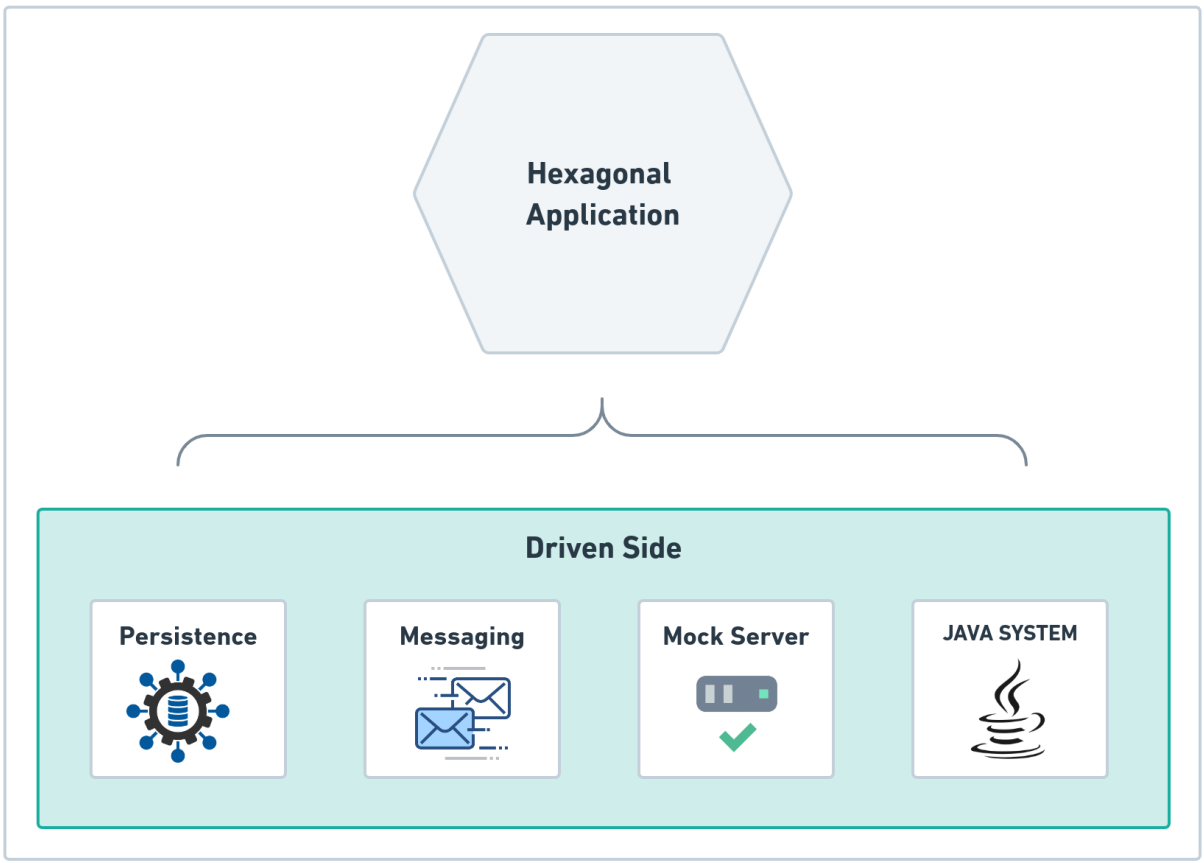

Postman
File Edit View Help
New
Import
Runner
My Workspace
Invite
Upgrade
Filter
History
Collections
APIs
New Collection
Trash
topology-inventory
2 requests
GET getRouter
GET addNetwork
getRouter
No Environment
Examples 0
BUILD
GET
http://localhost:8080/network/get?routerId=ca23800e-9b5a-11eb-a8b3-0242ac130003
Send
Save
Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings
Cookies
Code
Query Params
KEY
VALUE
DESCRIPTION
Bulk Edit
routerId
ca23800e-9b5a-11eb-a8b3-0242ac130003
Key
Value
Description
Response
Hit Send to get a response
Find and Replace
Console
Bootcamp
Build
Browse

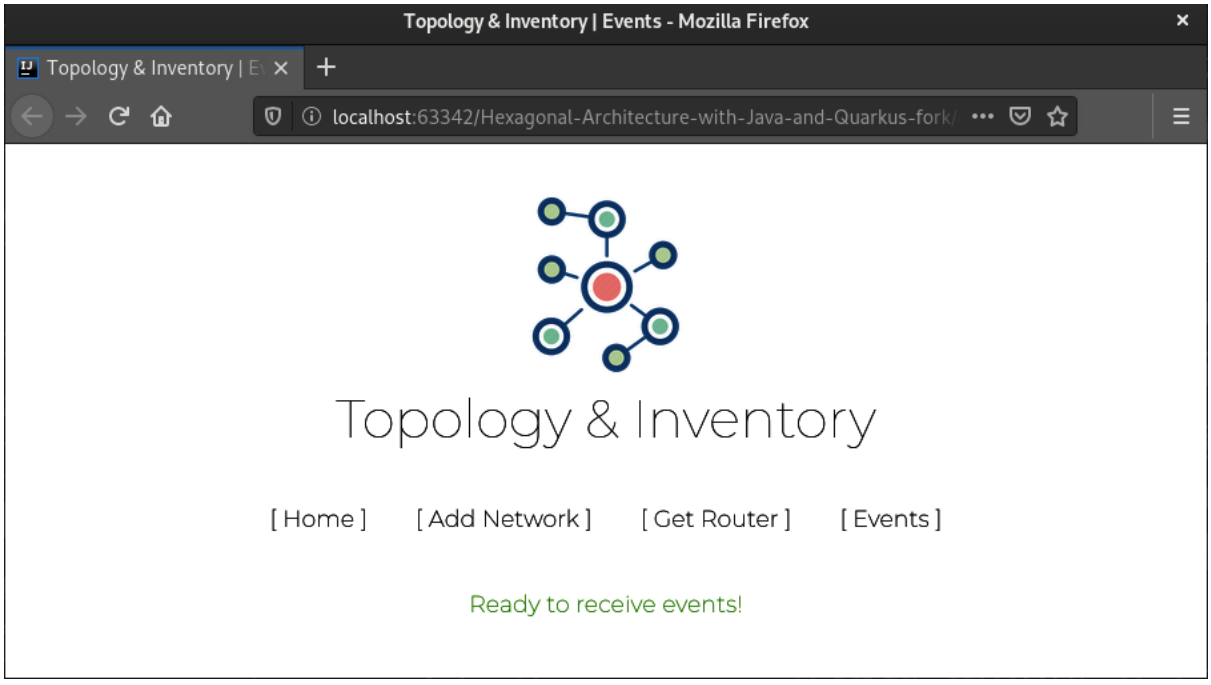
```
m4ndr4ck@casanova:~/IdeaProjects/Hexagonal-Architecture-with-Java-and-Quarkus-fork/chapter5
[m4ndr4ck@casanova chapter5]$ newman run topology-inventory.postman_collection.json
newman

topology-inventory

→ getRouter
  GET http://localhost:8080/network/get?routerId=ca23800e-9b5a-11eb-a8b3-0242ac130003 [200 OK, 574
B, 232ms]
  ✓  Status code is 200
  ✓  The response has all properties

→ addNetwork
  GET http://localhost:8080/network/add?routerId=ca23800e-9b5a-11eb-a8b3-0242ac130003&address=40.0
.0.0&name=Finance&cidr=8 [200 OK, 664B, 10ms]
  ✓  Status code is 200
  ✓  The response has all properties
```

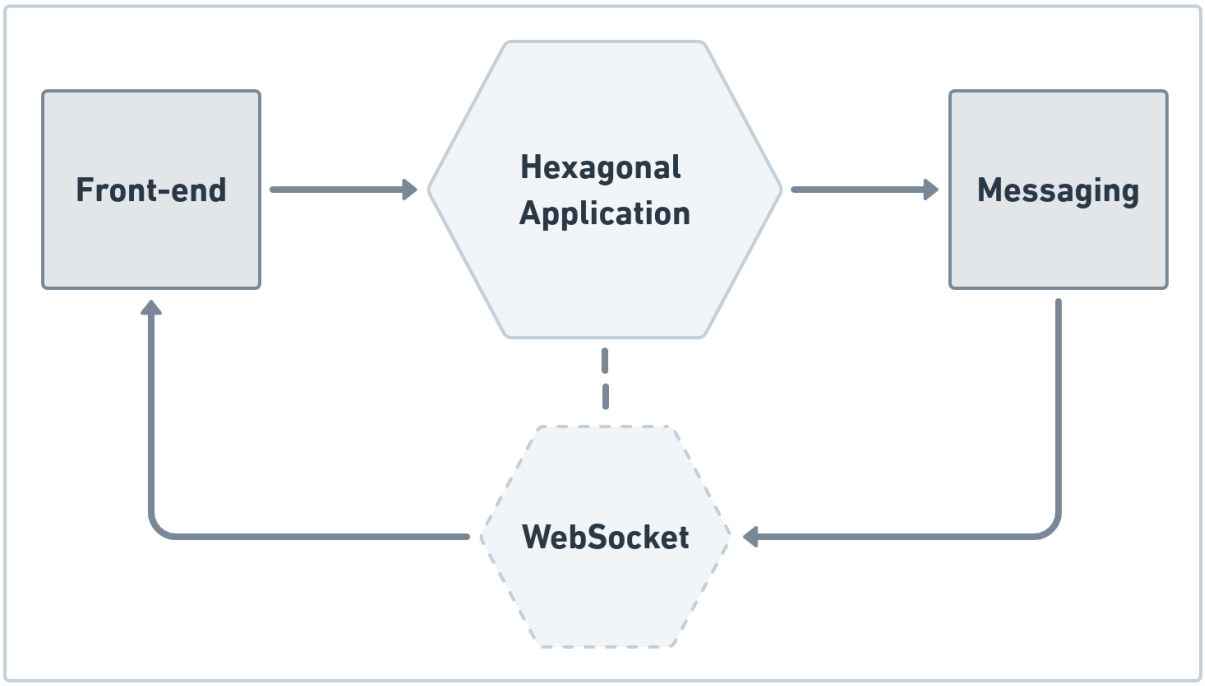
	executed	failed
iterations	1	0
requests	2	0
test-scripts	2	0
prerequest-scripts	0	0
assertions	4	0
total run duration: 296ms		
total data received: 956B (approx)		
average response time: 121ms [min: 10ms, max: 232ms, s.d.: 111ms]		

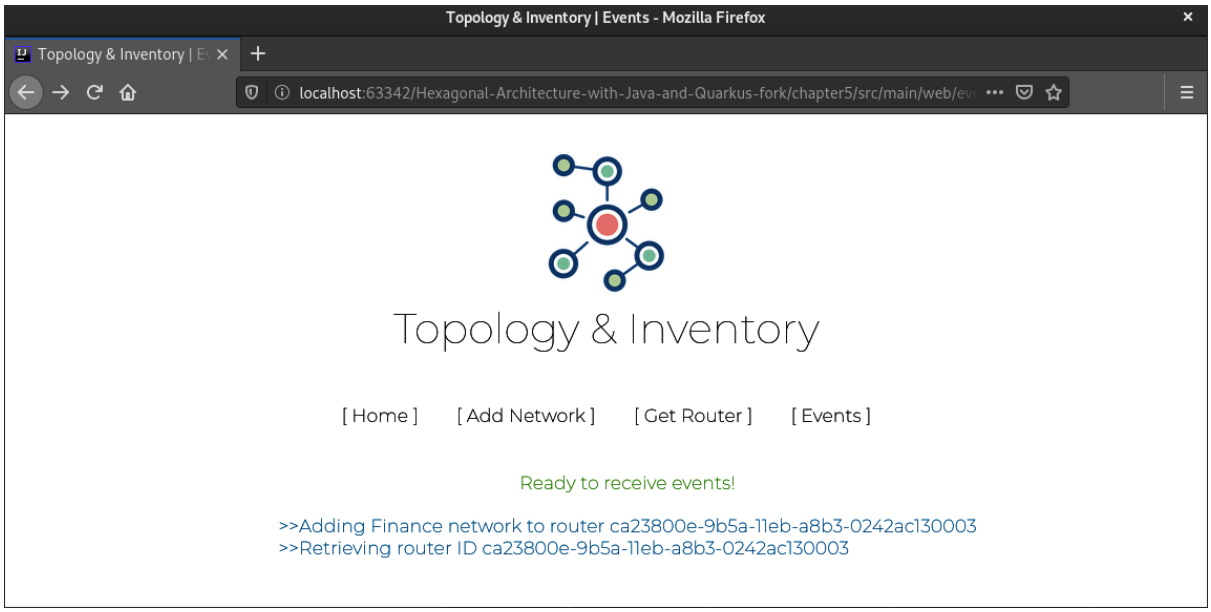
```
[m4ndr4ck@casanova chapter5]$
```



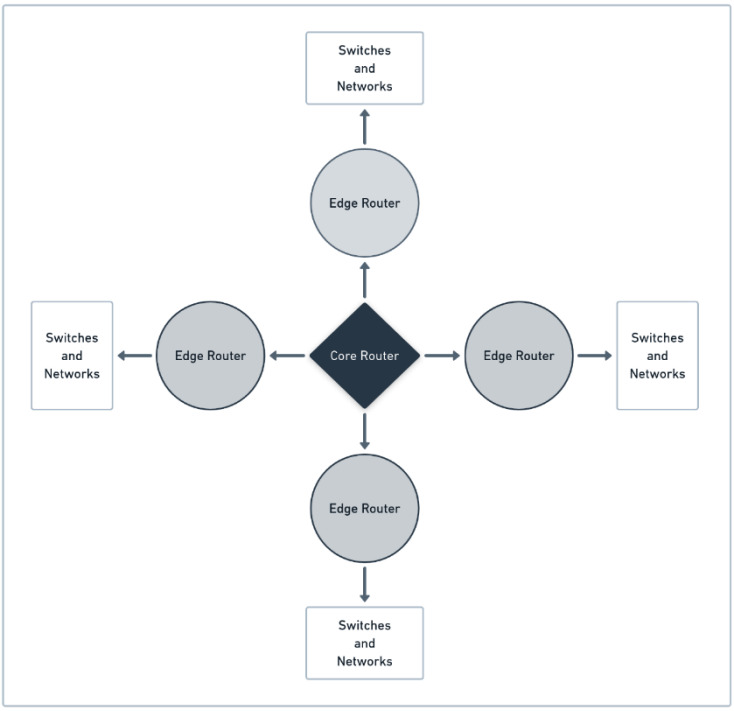

Hexagonal Application
Driven Side
Persistence
Messaging
Mock Server
JAVA SYSTEM

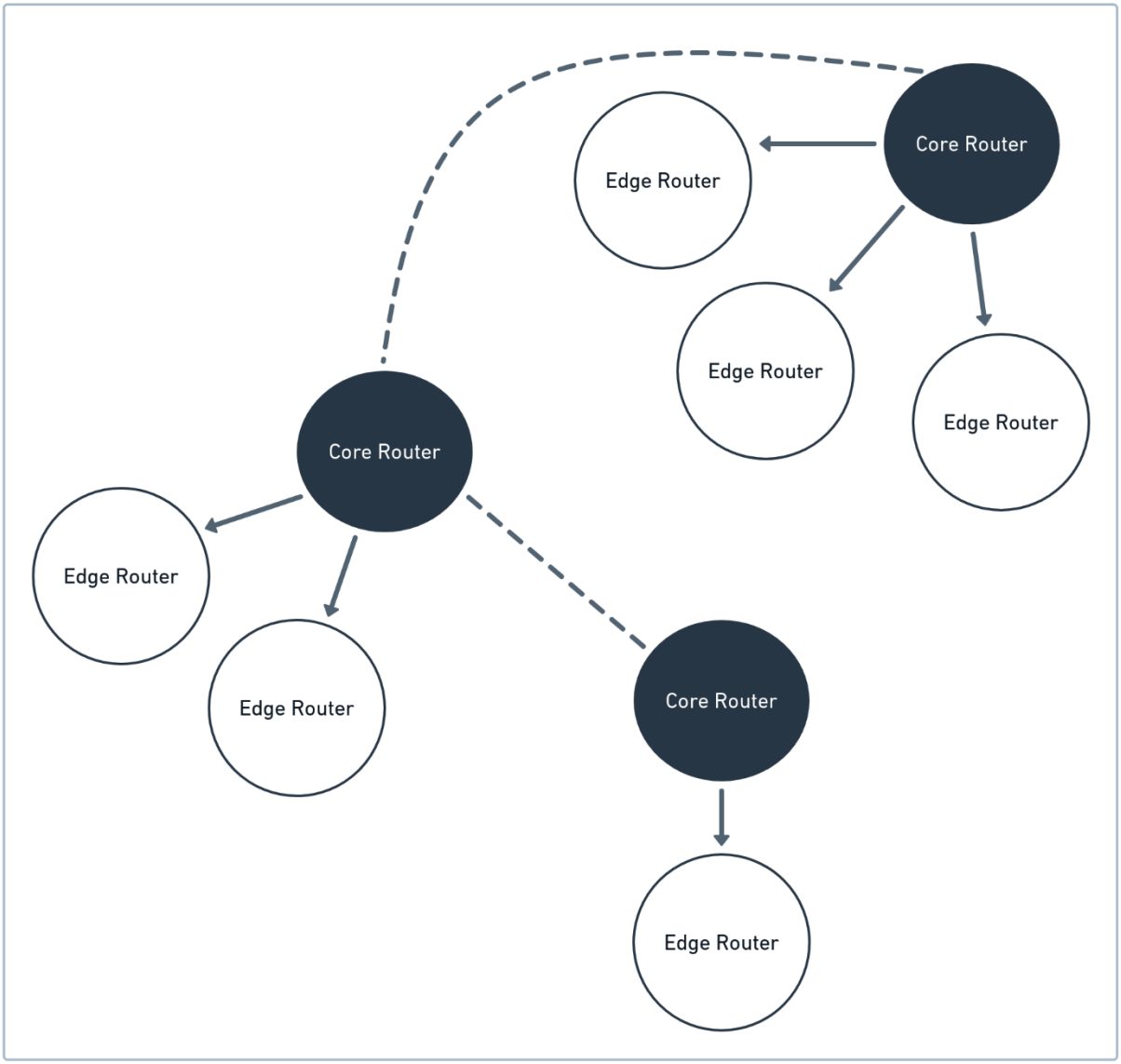

Topology & Inventory | Events - Mozilla Firefox
Topology & Inventory | Ev
localhost:63342/Hexagonal-Architecture-with-Java-and-Quarkus-fork/
Topology & Inventory
[Home] [Add Network] [Get Router] [Events]
Ready to receive events!

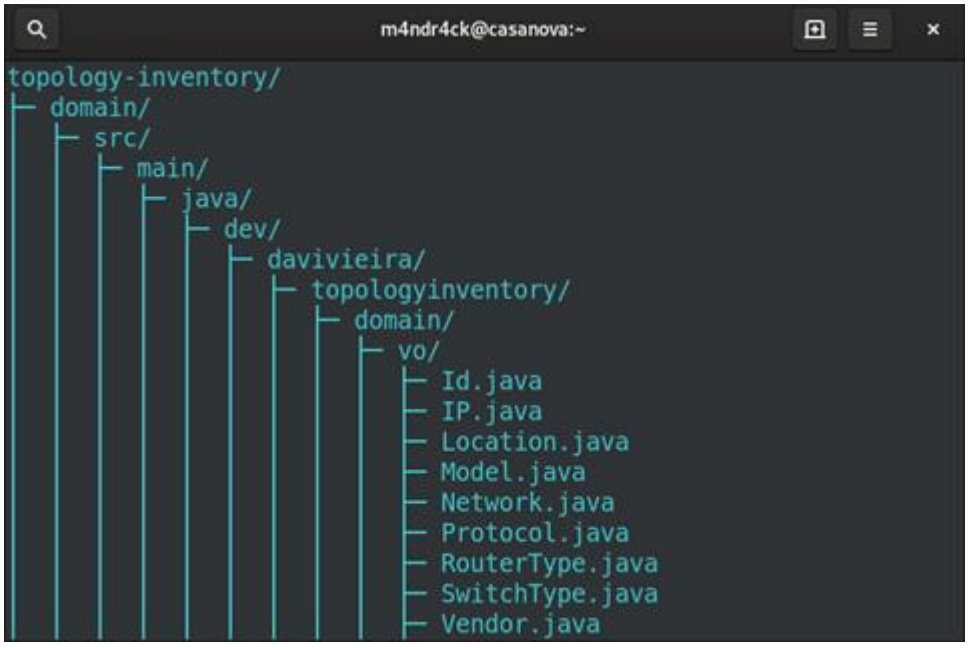

Front-end
Hexagonal Application
Messaging
WebSocket


Topology & Inventory | Events - Mozilla Firefox
Topology & Inventory | Ev
localhost:63342/Hexagonal-Architecture-with-Java-and-Quarkus-fork/chapter5/src/main/web/ev
Topology & Inventory
[Home] [Add Network] [Get Router] [Events]
Ready to receive events!
>>Adding Finance network to router ca23800e-9b5a-11eb-a8b3-0242ac130003
>>Retrieving router ID ca23800e-9b5a-11eb-a8b3-0242ac130003

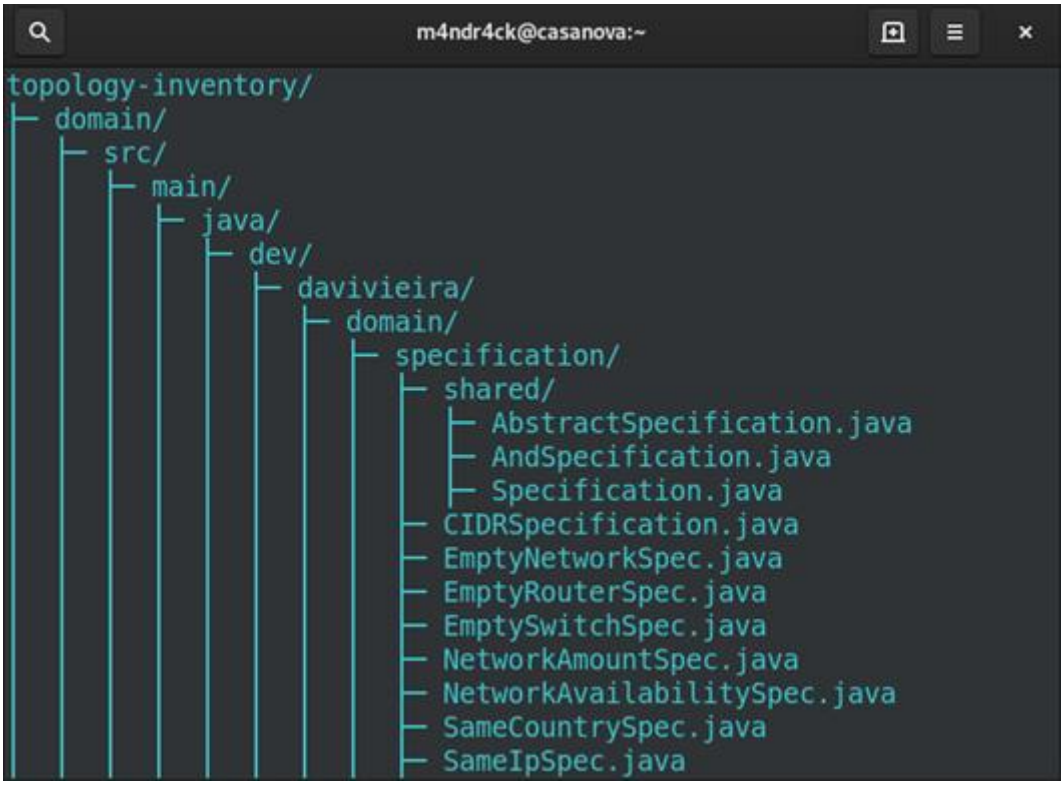
Chapter 6: Building the Domain Hexagon




Core Router
Edge Router
Edge Router
Edge Router
Core Router
Edge Router
Edge Router
Core Router
Edge Router


m4ndr4ck@casanova:~
topology-inventory/
├─ domain/
│ ├─ src/
│ │ ├─ main/
│ │ │ ├─ java/
│ │ │ │ ├─ dev/
│ │ │ │ │ ├─ davivieira/
│ │ │ │ │ │ ├─ topologyinventory/
│ │ │ │ │ │ │ ├─ domain/
│ │ │ │ │ │ │ │ ├─ vo/
│ │ │ │ │ │ │ │ │ ├─ Id.java
│ │ │ │ │ │ │ │ │ ├─ IP.java
│ │ │ │ │ │ │ │ │ ├─ Location.java
│ │ │ │ │ │ │ │ │ ├─ Model.java
│ │ │ │ │ │ │ │ │ ├─ Network.java
│ │ │ │ │ │ │ │ │ ├─ Protocol.java
│ │ │ │ │ │ │ │ │ ├─ RouterType.java
│ │ │ │ │ │ │ │ │ ├─ SwitchType.java
│ │ │ │ │ │ │ │ │ ├─ Vendor.java

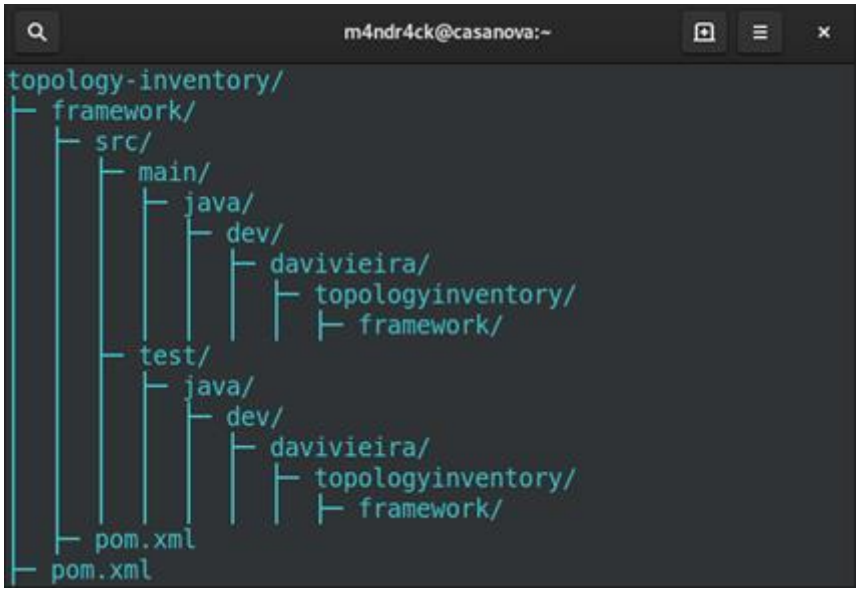

m4ndr4ck@casanova:~
topology-inventory/
domain/
src/
main/
java/
dev/
davivieira/
domain/
entity/
CoreRouter.java
EdgeRouter.java
Equipment.java
Router.java
Switch.java


m4ndr4ck@casanova:~
topology-inventory/
domain/
src/
main/
java/
dev/
davivieira/
domain/
specification/
shared/
AbstractSpecification.java
AndSpecification.java
Specification.java
CIDRSpecification.java
EmptyNetworkSpec.java
EmptyRouterSpec.java
EmptySwitchSpec.java
NetworkAmountSpec.java
NetworkAvailabilitySpec.java
SameCountrySpec.java
SameIpSpec.java

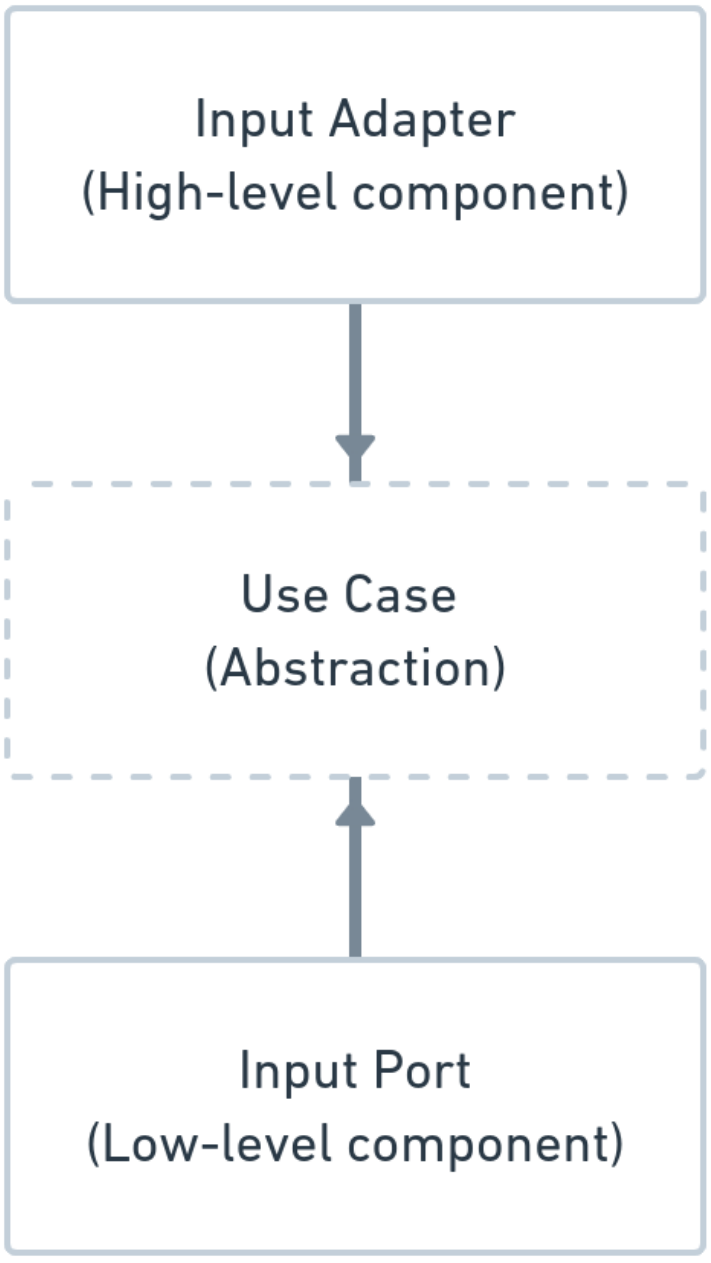

m4ndr4ck@casanova:~
topology-inventory/
domain/
src/
main/
java/
dev/
davivieira/
domain/
service/
NetworkService.java
RouterService.java
SwitchService.java

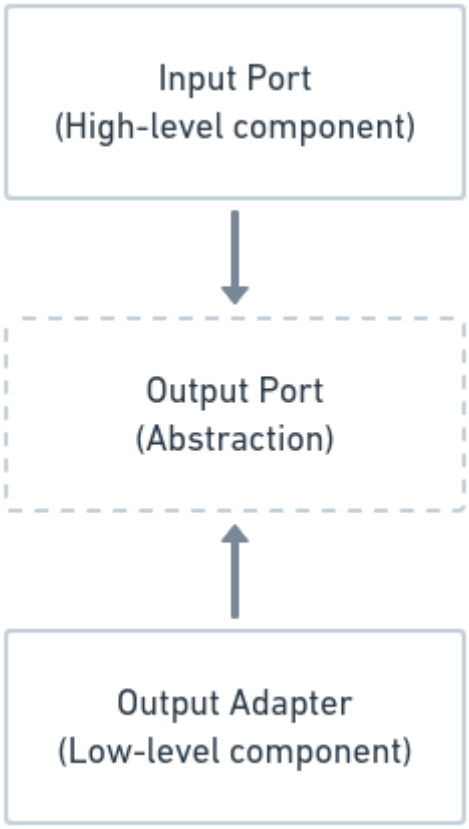
Chapter 7: Building the Application Hexagon

Chapter 8: Building the Framework Hexagon

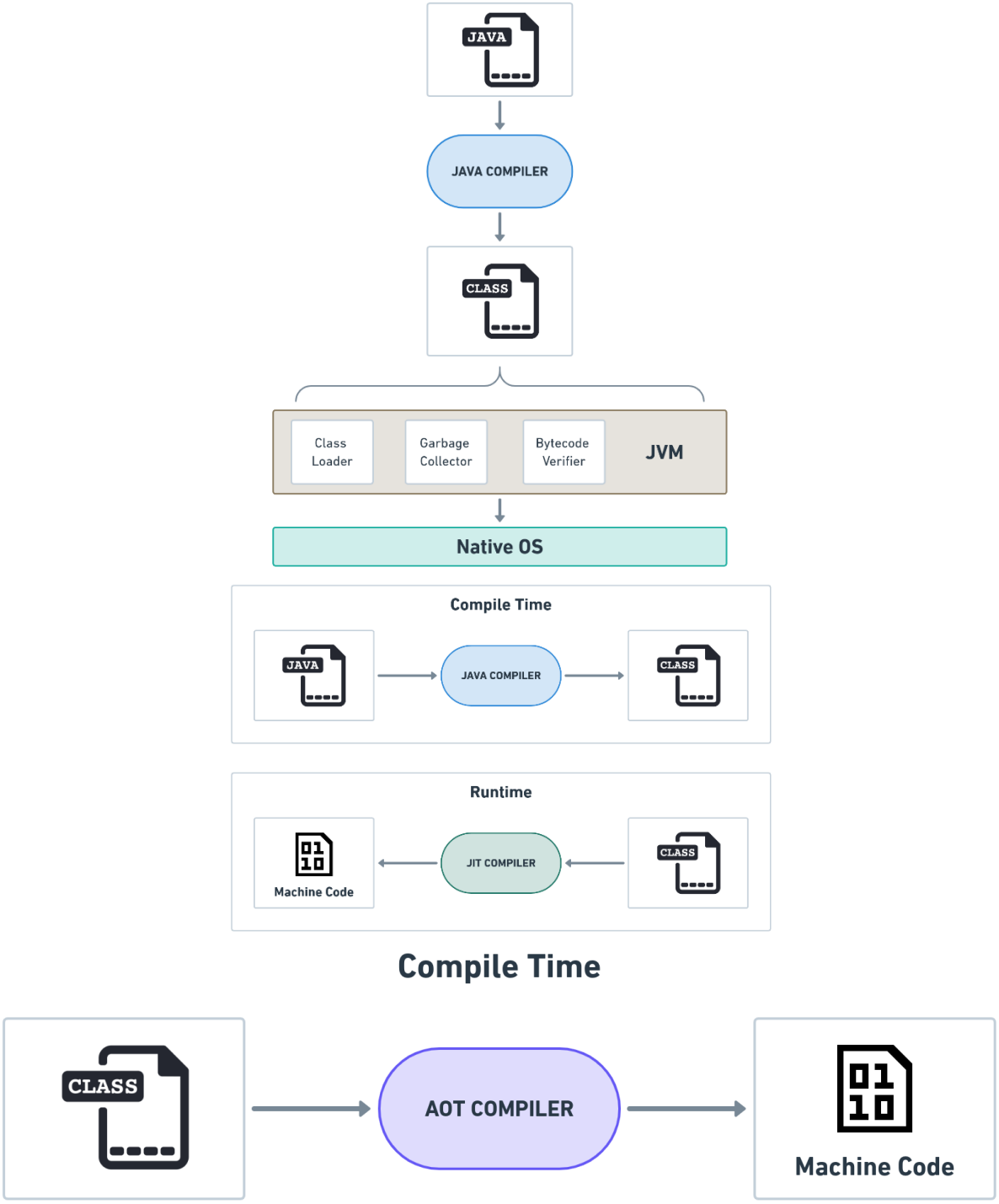


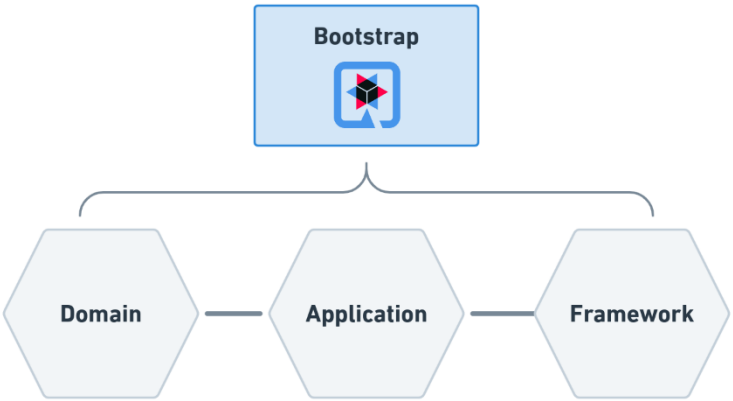
Chapter 9: Applying Dependency Inversion with Java Modules




Input Port
(High-level component)
Output Port
(Abstraction)
Output Adapter
(Low-level component)

Chapter 10: Adding Quarkus to a Modularized Hexagonal Application





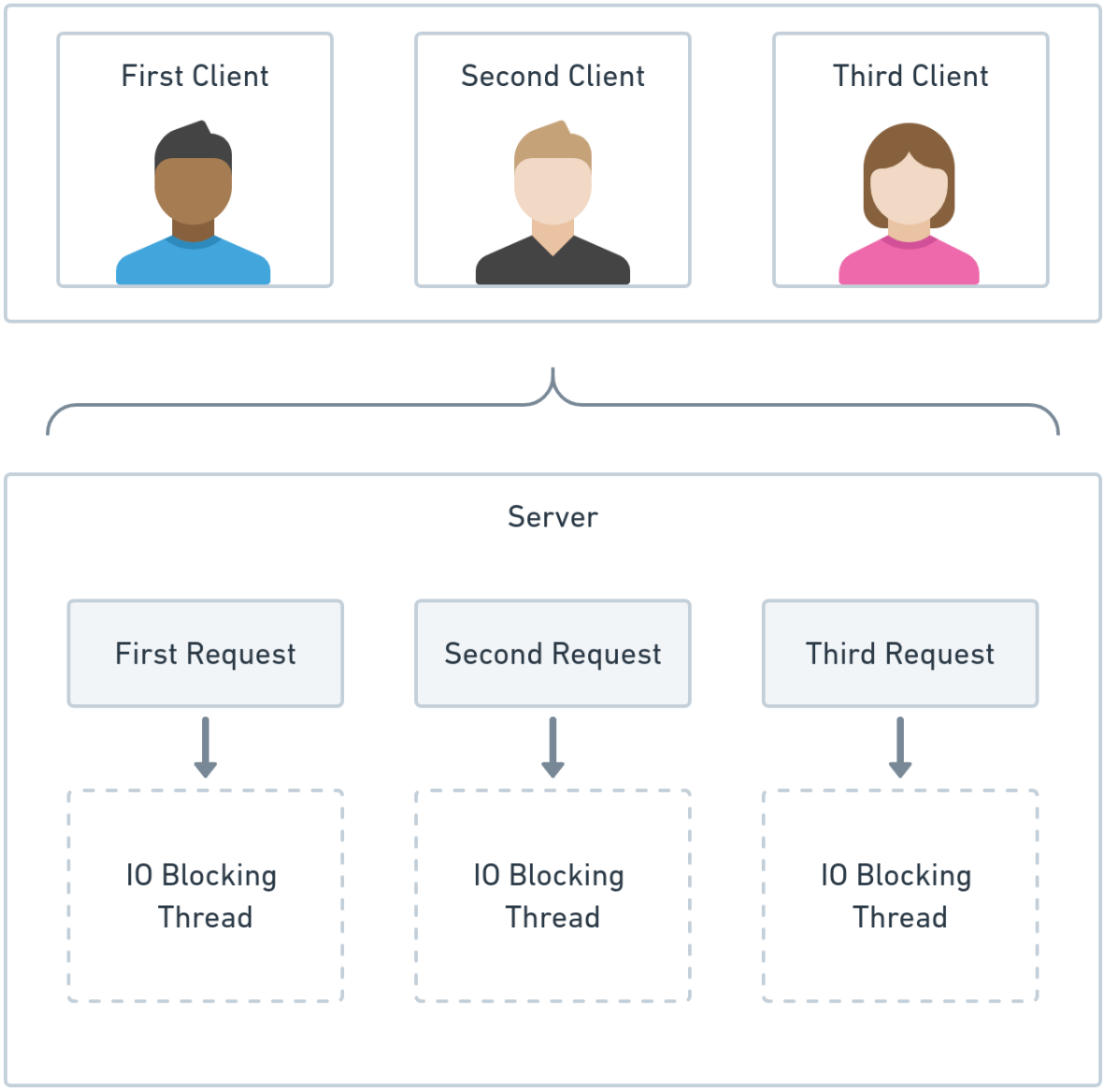

```
m4ndr4ck@casanova:~
topology-inventory/
├─ application/
├─ bootstrap/
├─ domain/
├─ framework/
```

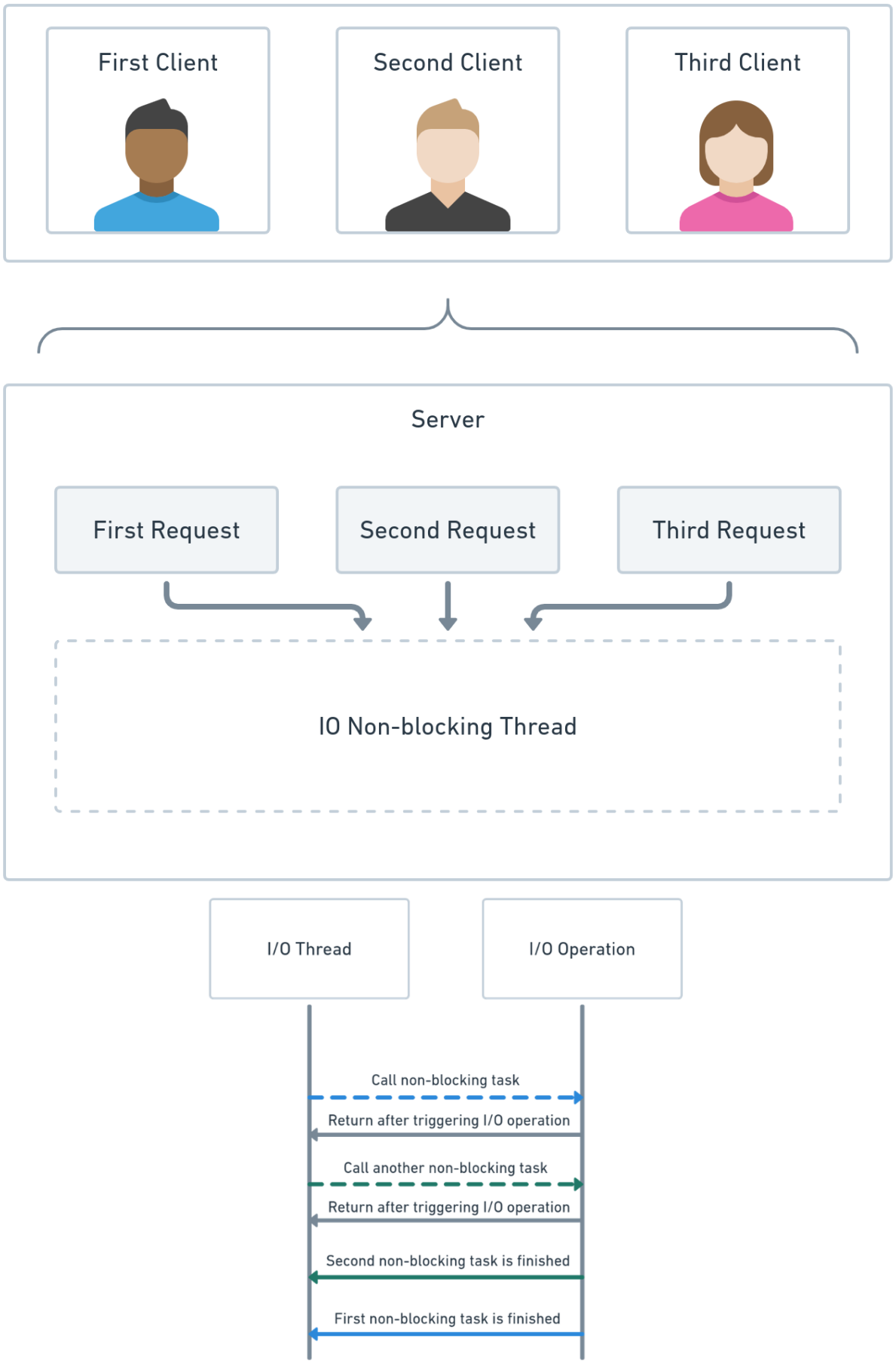
```
m4ndr4ck@casanova:~/IdeaProjects/Hexagonal-Architecture-with-Java-and-Quarkus-fork/chapter10
[m4ndr4ck@casanova chapter10]$ java -jar bootstrap/target/bootstrap-1.0-SNAPSHOT-runner.jar
__  ____  __  _____   ___  __ ____  ______
 --/ __ \/ / / / _ | / _ \/ //_/ / / / __/
 -/ /_/ / /_/ / __ |/ , _/ ,< / /_/ /\ \
--\___\_\____/_/ |_/_/|_/_/|_|\____/___/
2021-08-29 00:00:28,835 INFO  [io.quarkus] (main) bootstrap 1.0-SNAPSHOT on JVM (powered by Quarkus 2.1.3.Final) started in 3.091s. Listening on: http://0.0.0.0:8080
2021-08-29 00:00:28,840 INFO  [io.quarkus] (main) Profile prod activated.
2021-08-29 00:00:28,840 INFO  [io.quarkus] (main) Installed features: [agroal, cdi, hibernate-orm, hibernate-validator, jdbc-h2, narayana-jta, resteasy, resteasy-jackson, smallrye-context-propagation, smallrye-openapi, swagger-ui]
```

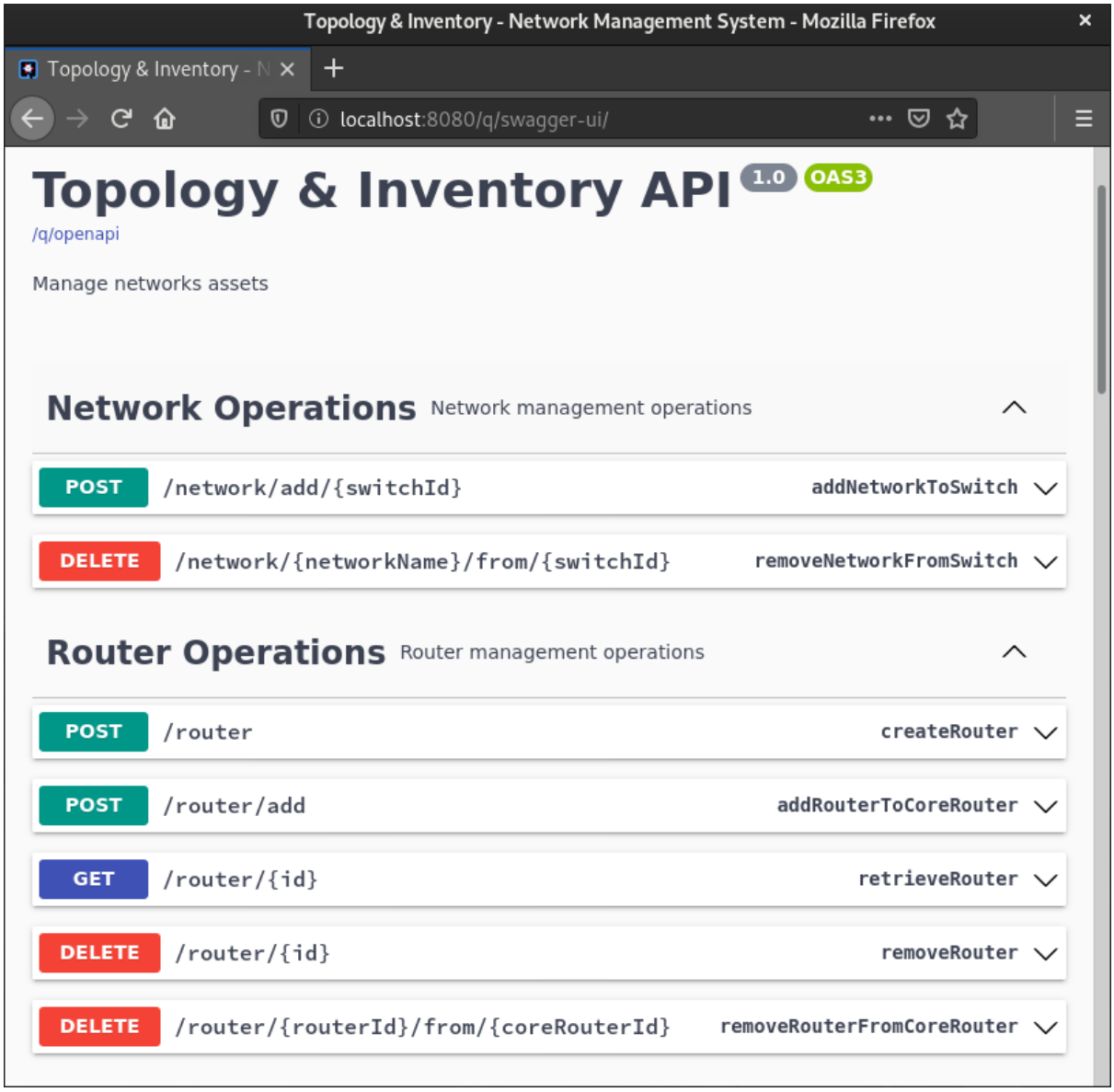
Chapter 11: Leveraging CDI Beans to Manage Ports and Use Cases

No images…

Chapter 12: Using RESTEasy Reactive to Implement Input Adapters



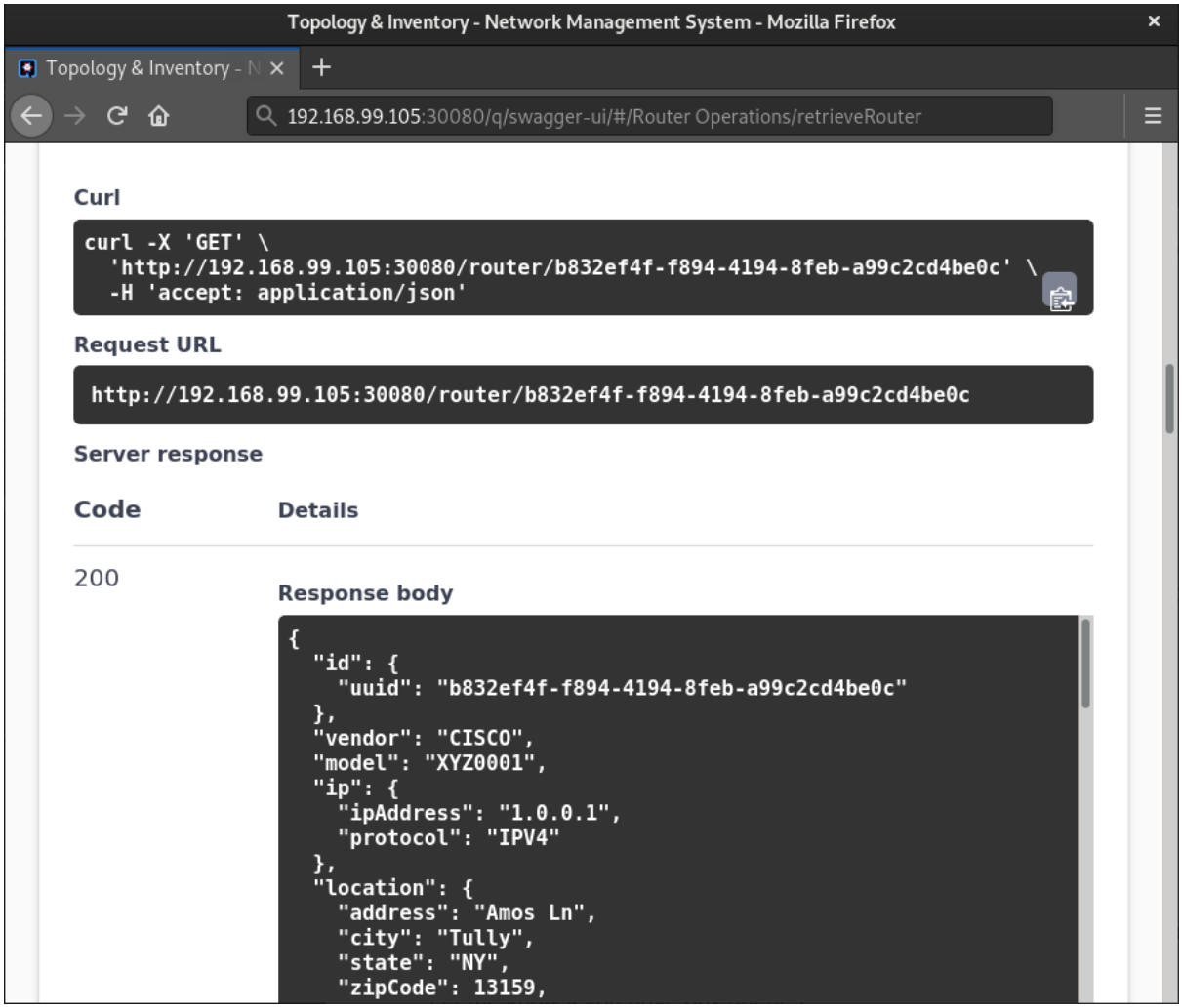

First Client
Second Client
Third Client
Server
First Request
Second Request
Third Request
IO Non-blocking Thread
I/O Thread
I/O Operation
Call non-blocking task
Return after triggering I/O operation
Call another non-blocking task
Return after triggering I/O operation
Second non-blocking task is finished
First non-blocking task is finished


Topology & Inventory - Network Management System - Mozilla Firefox
Topology & Inventory - N
localhost:8080/q/swagger-ui/
Topology & Inventory API
1.0
OAS3
/q/openapi
Manage networks assets
Network Operations
Network management operations
POST /network/add/{switchId} addNetworkToSwitch
DELETE /network/{networkName}/from/{switchId} removeNetworkFromSwitch
Router Operations
Router management operations
POST /router createRouter
POST /router/add addRouterToCoreRouter
GET /router/{id} retrieveRouter
DELETE /router/{id} removeRouter
DELETE /router/{routerId}/from/{coreRouterId} removeRouterFromCoreRouter

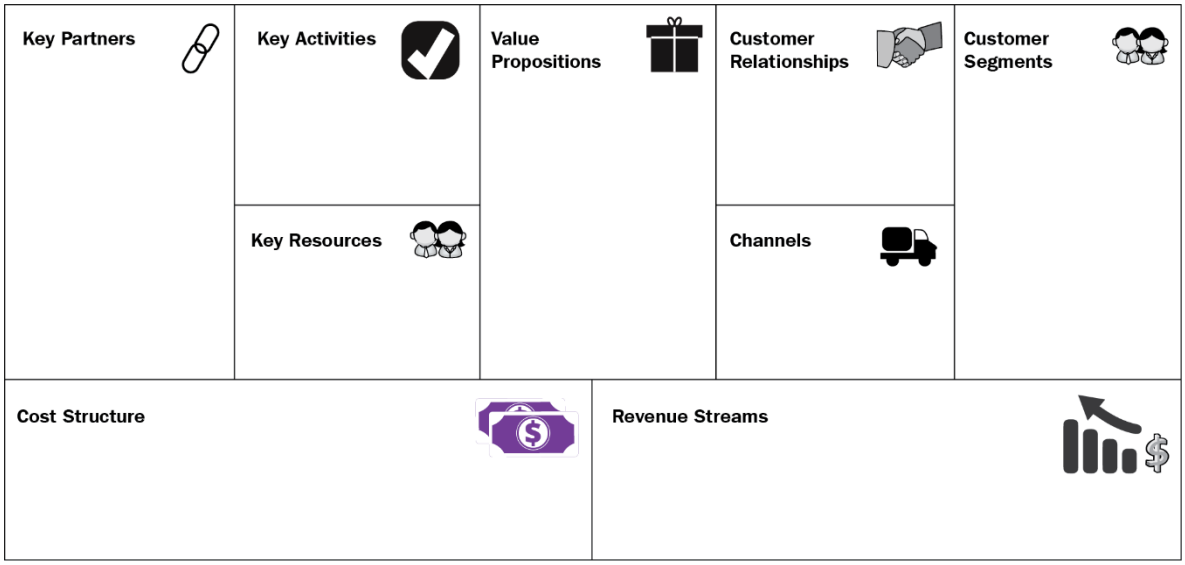
Chapter 13: Persisting Data with Output Adapters and Hibernate Reactive

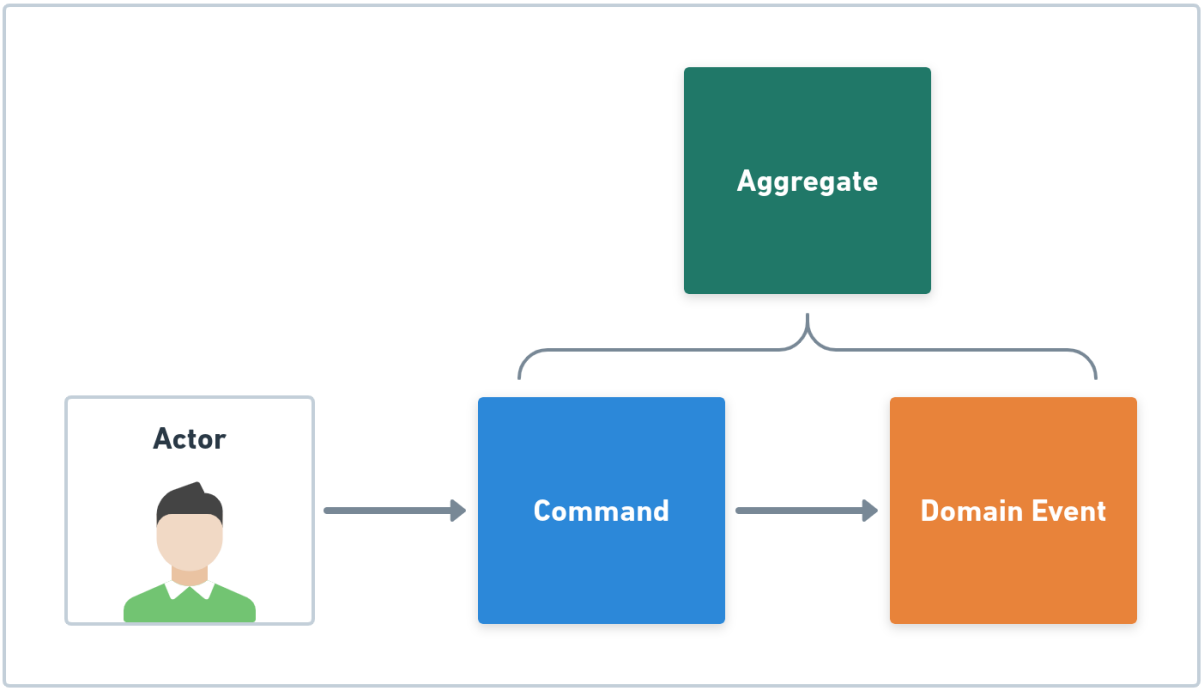
No images…

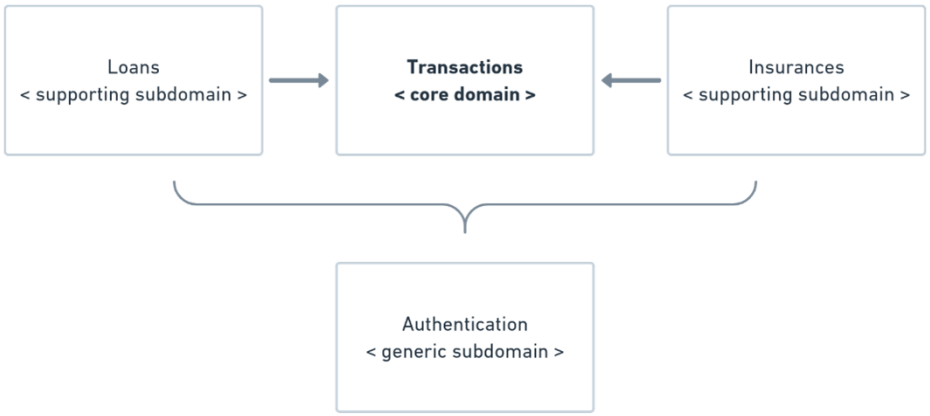
Chapter 14: Setting Up Dockerfile and Kubernetes Objects for Cloud Deployment

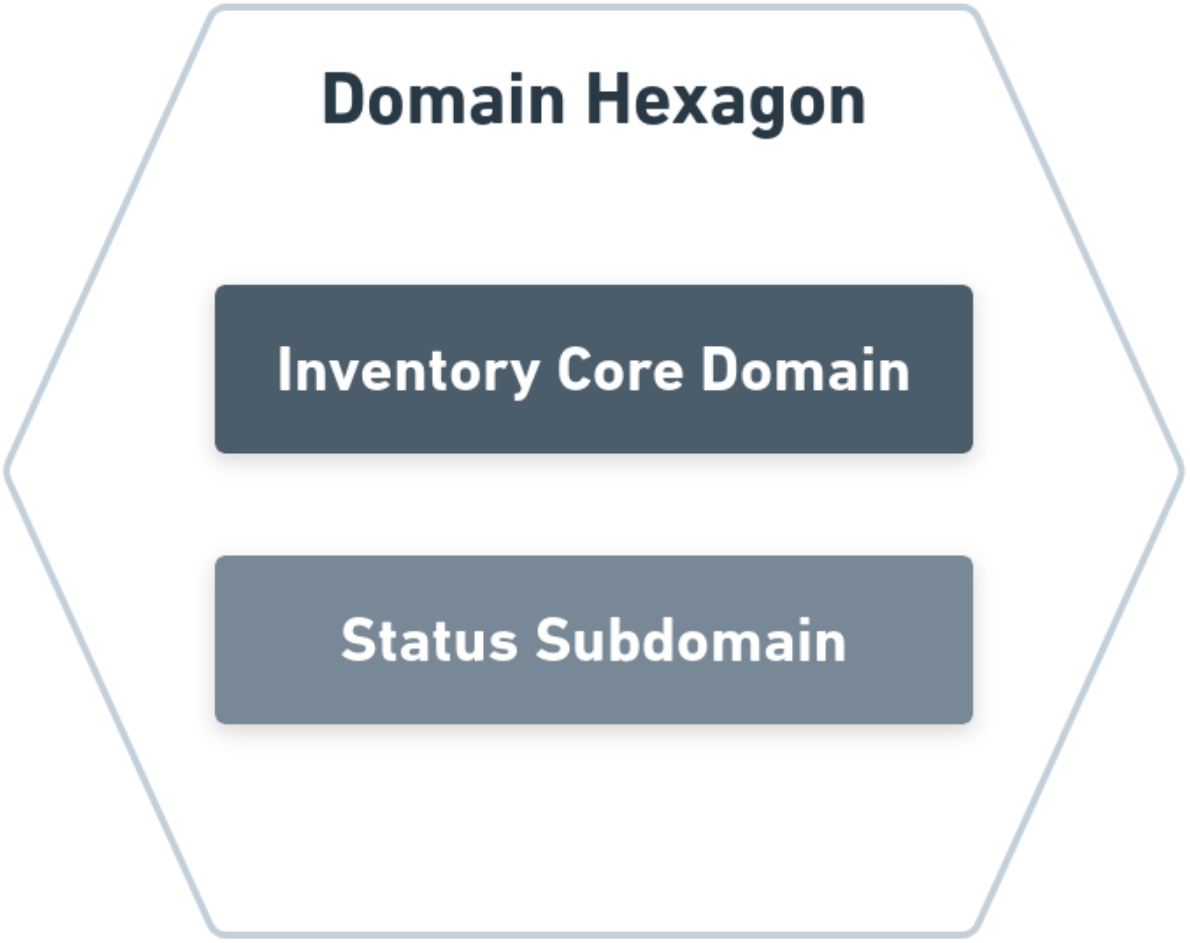


Chapter 15: Good Design Practices for Your Hexagonal Application






Loans
< supporting subdomain >
Transactions
< core domain >
Insurances
< supporting subdomain >
Authentication
< generic subdomain >


Domain Hexagon
Inventory Core Domain
Status Subdomain