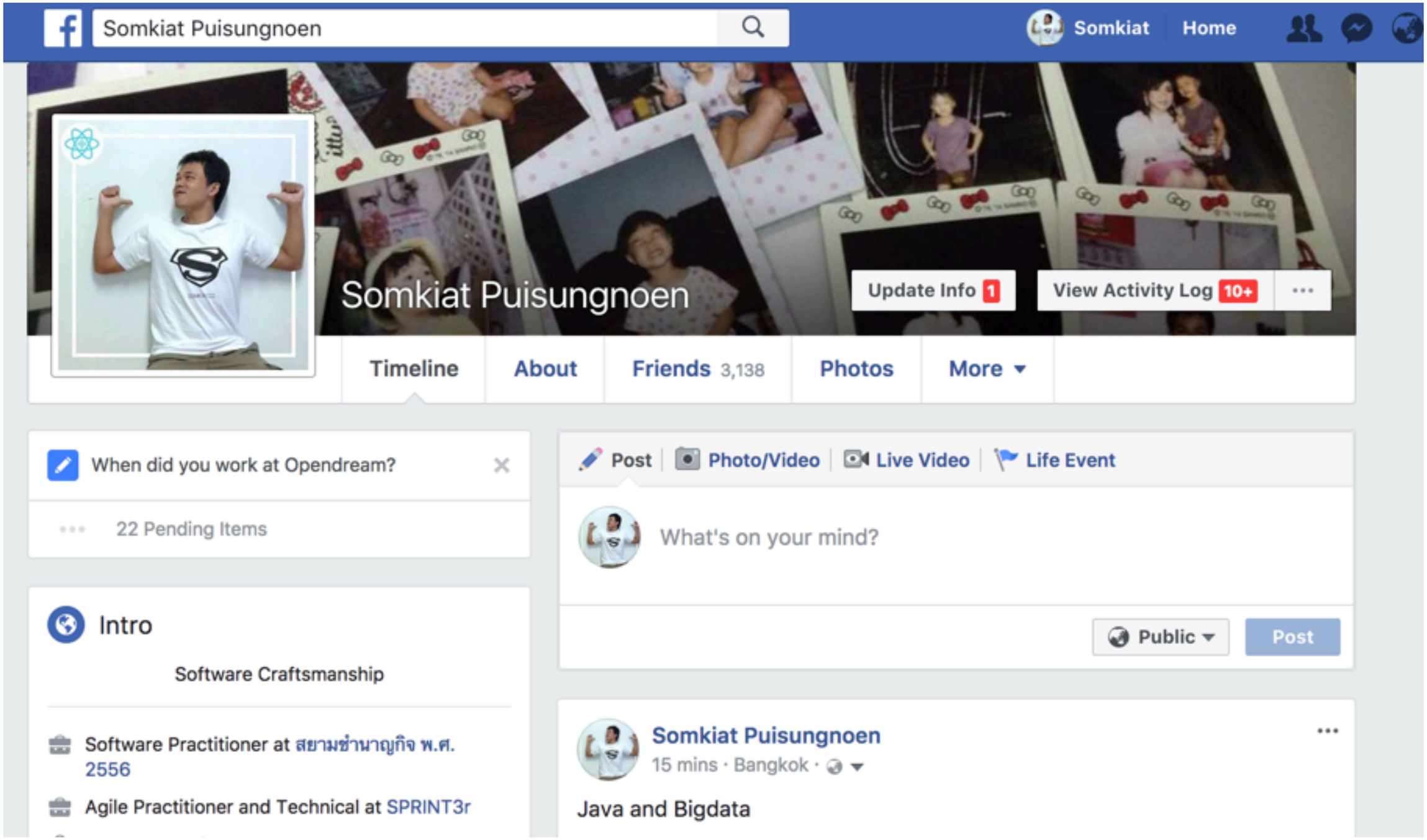
Learning Go

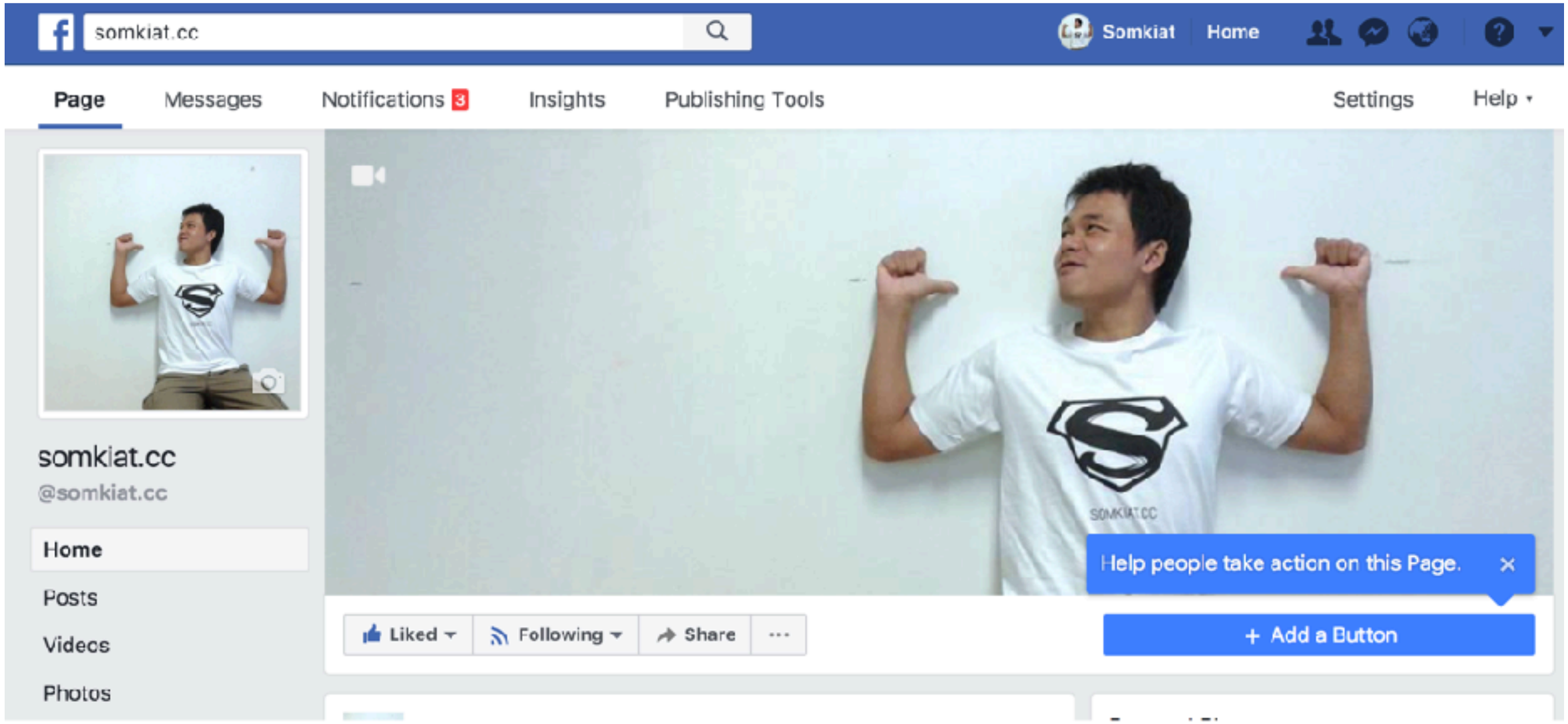
Test Double

สยามชำนาญกิจ
SIAM CHAMNAN KIT

Shu Ha Ri

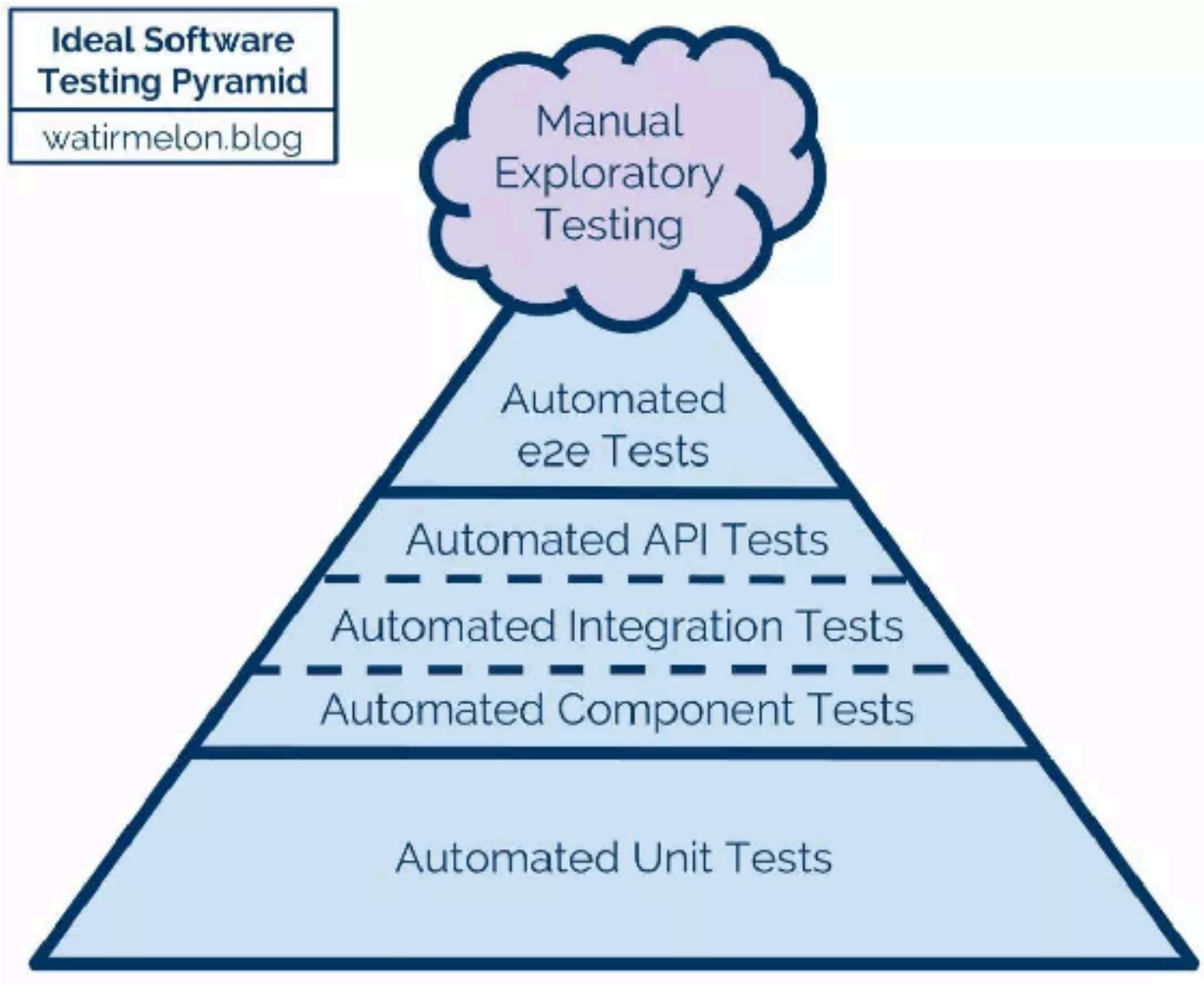
THE SCHOOL OF SOFTWARE DEVELOPMENT





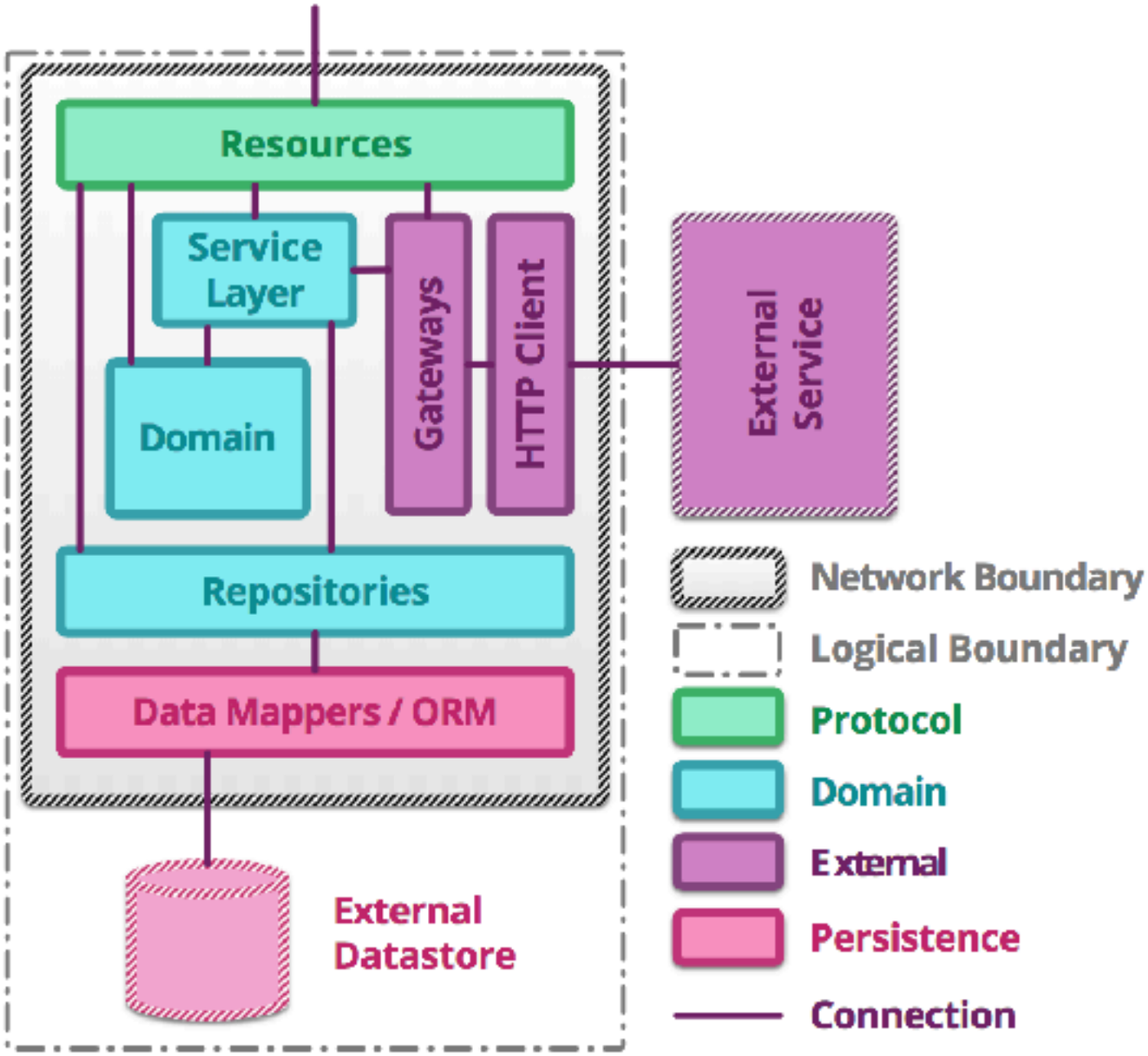
Test Double

Testing Pyramid



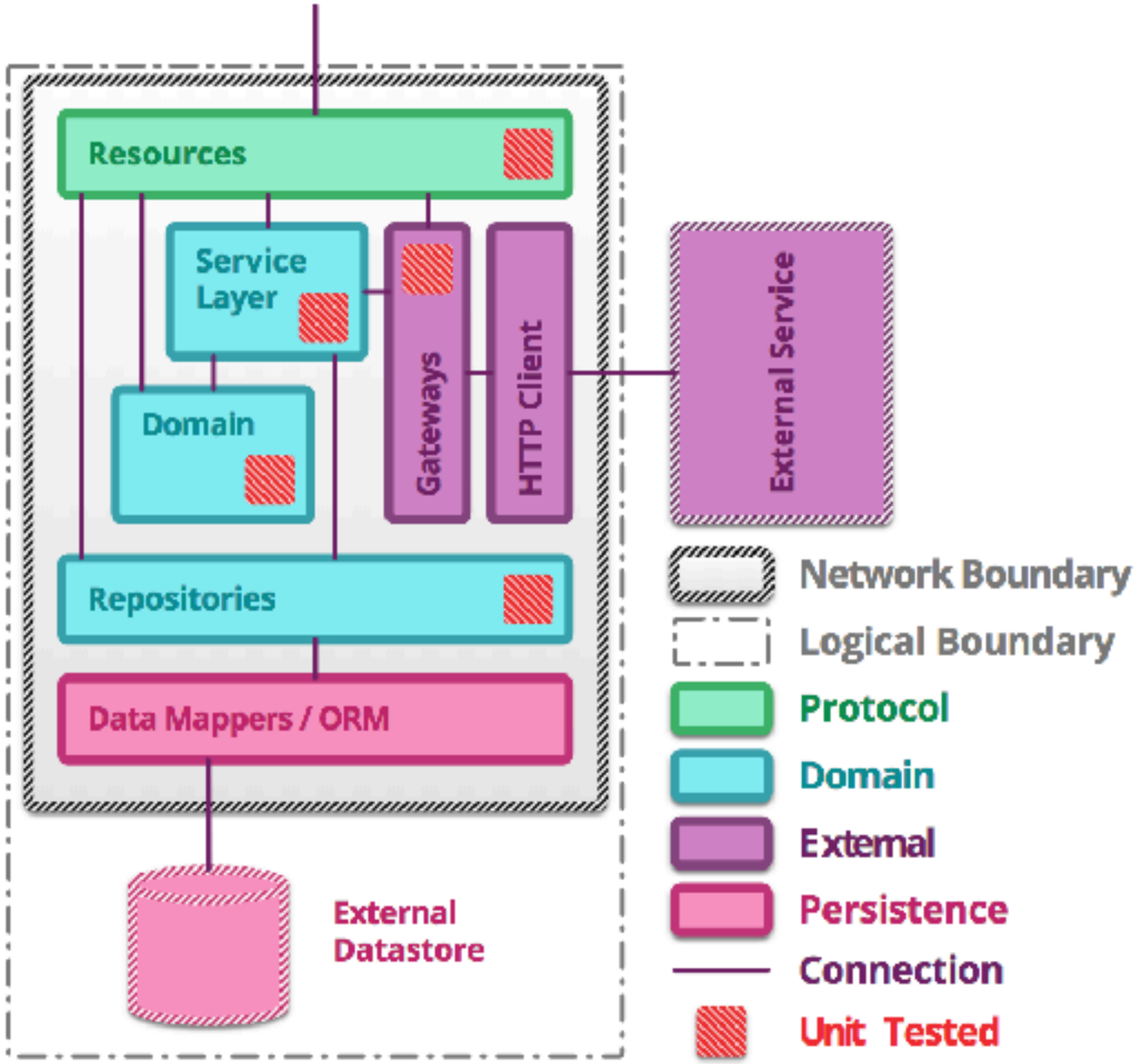
https://watirmelon.blog/testing-pyramids/

Project/Service Structure



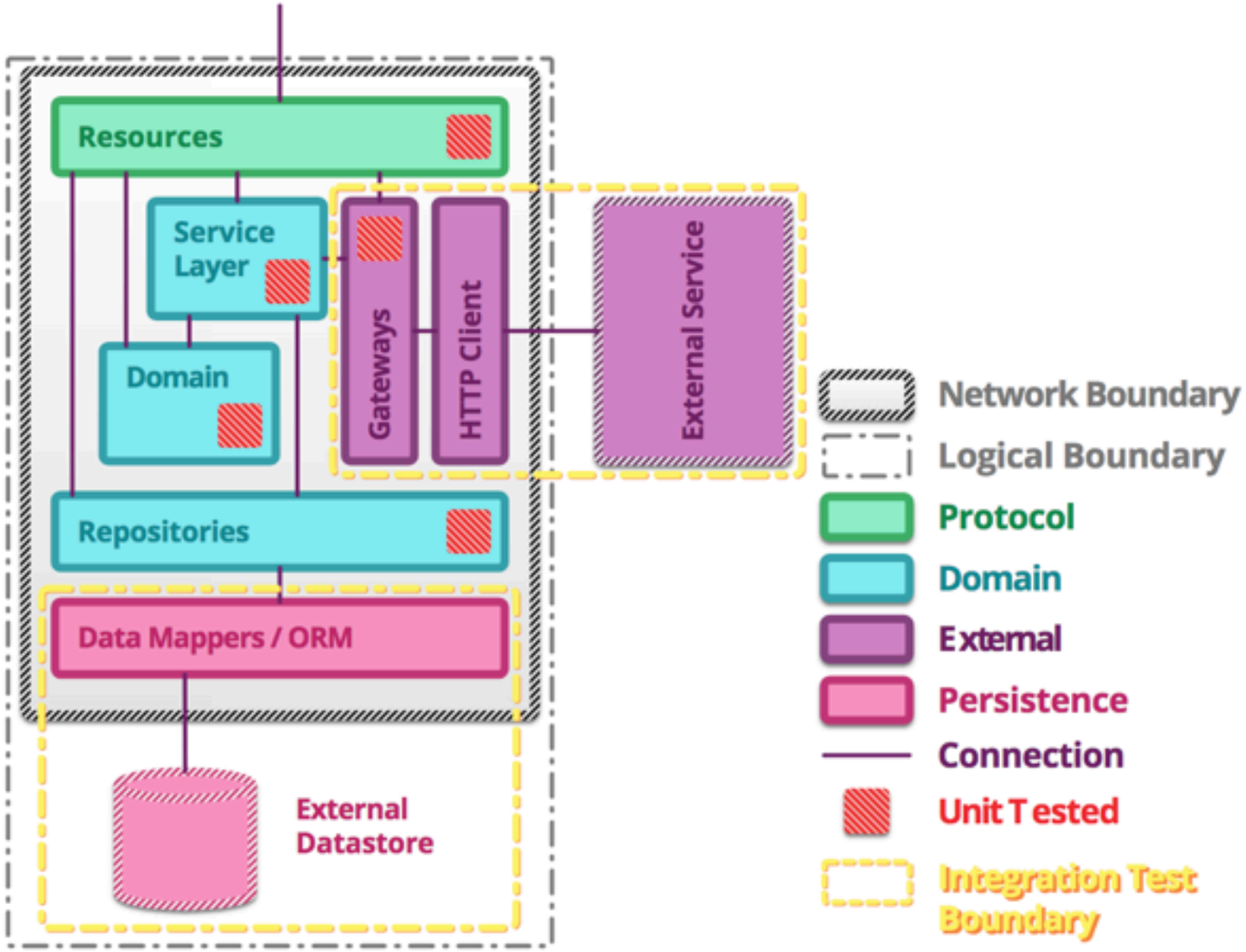
https://martinfowler.com/articles/microservice-testing

Unit Testing



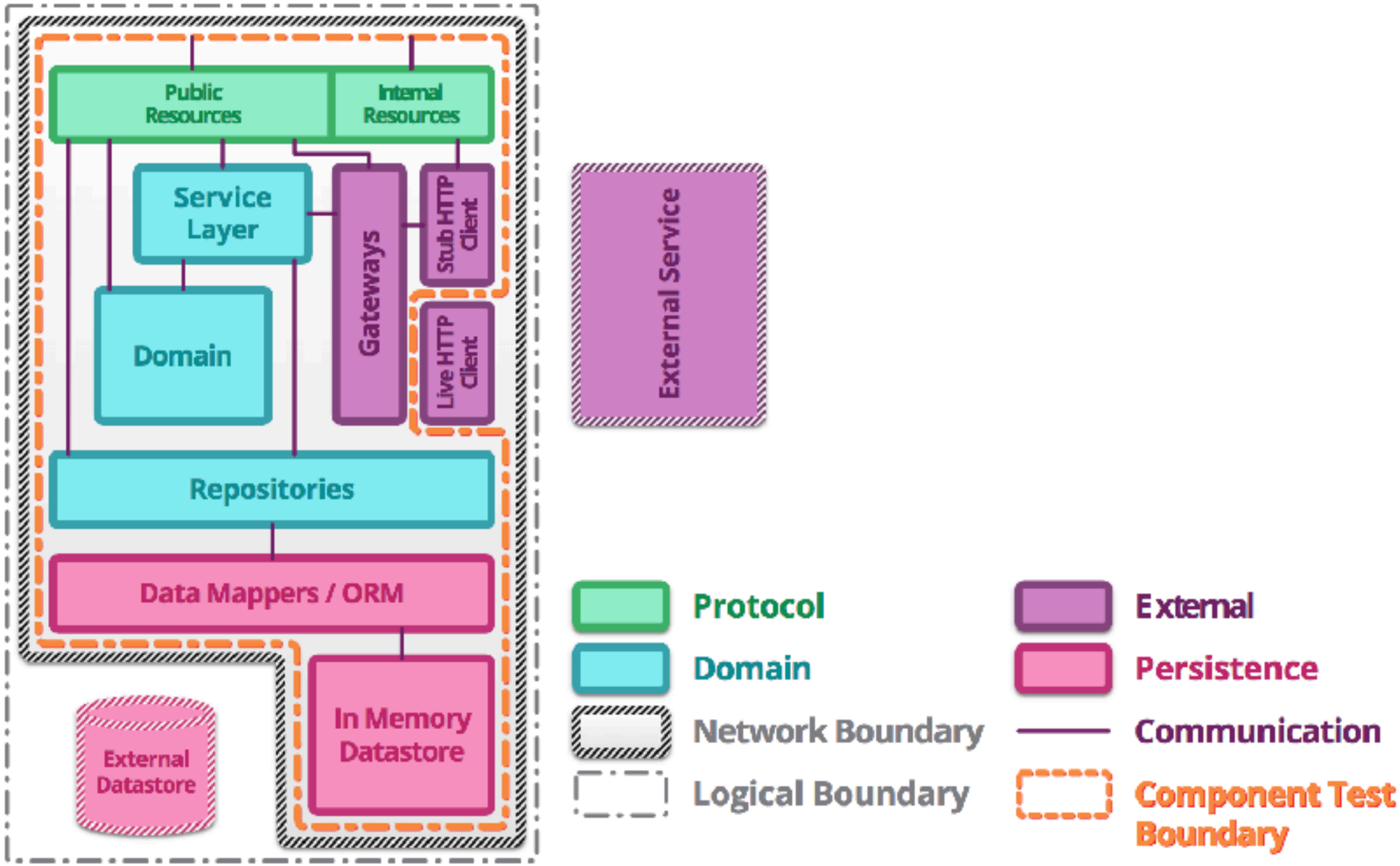
https://martinfowler.com/articles/microservice-testing

Integration Testing



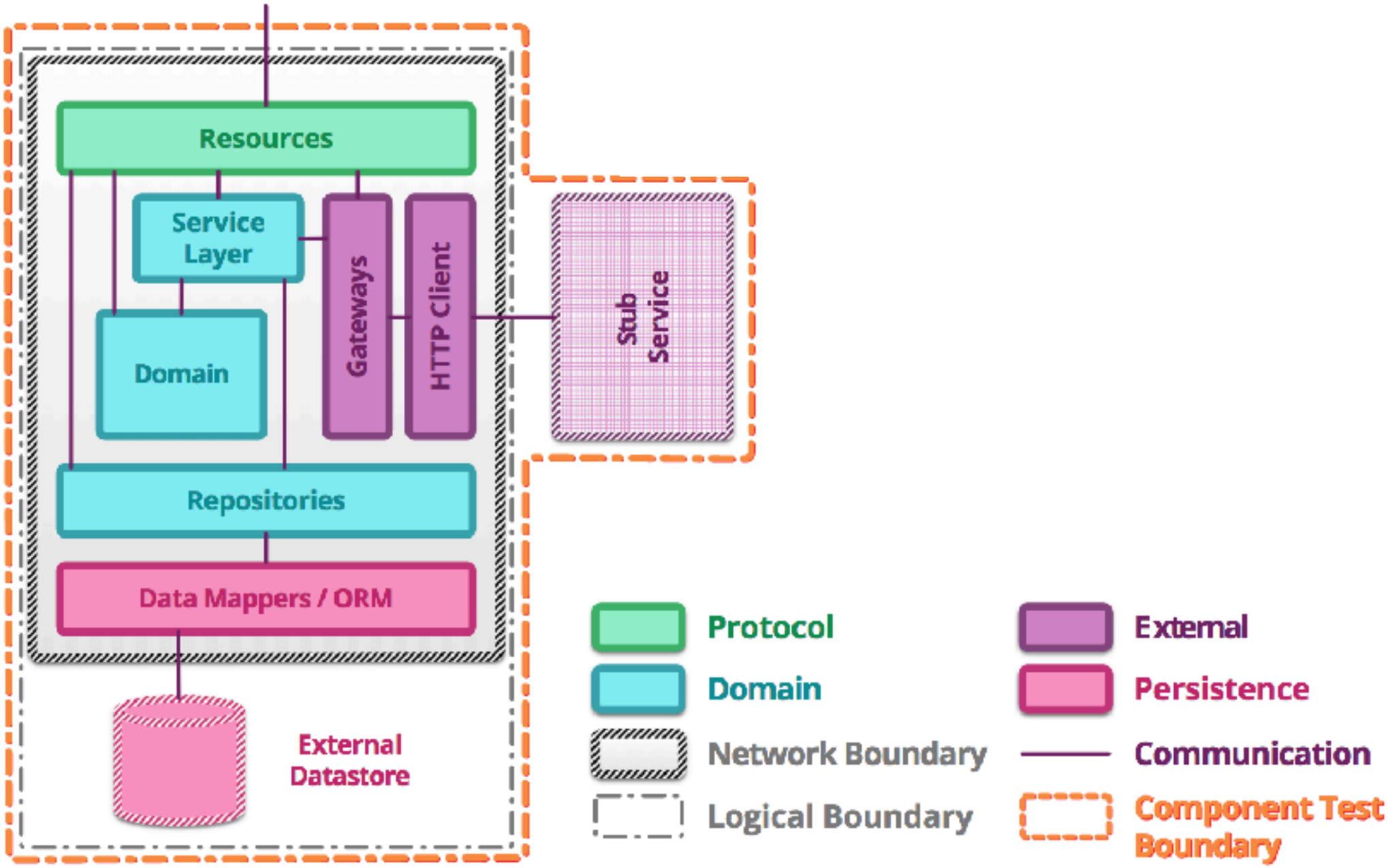
https://martinfowler.com/articles/microservice-testing

Component Testing (1)



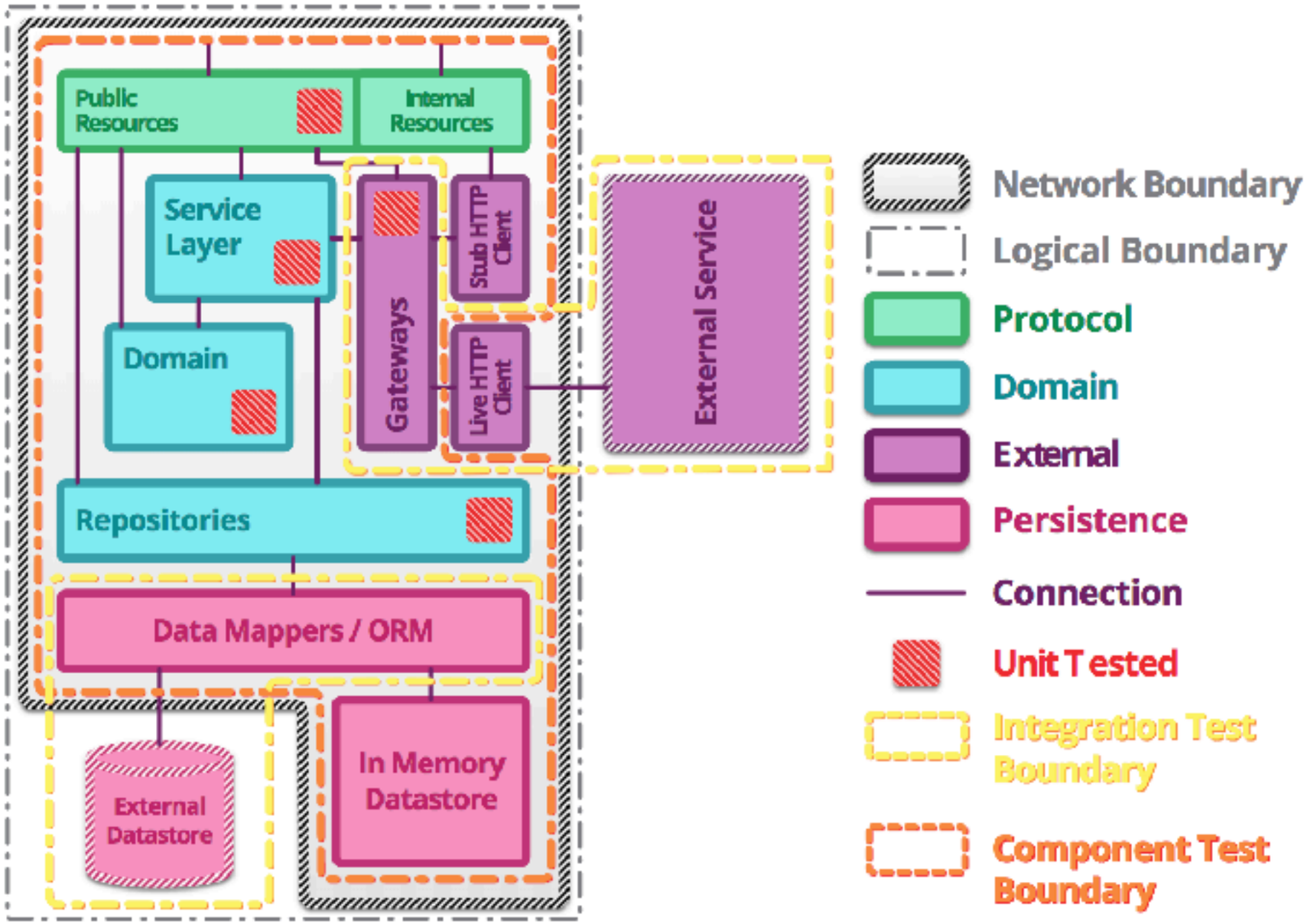
https://martinfowler.com/articles/microservice-testing

Component Testing (2)



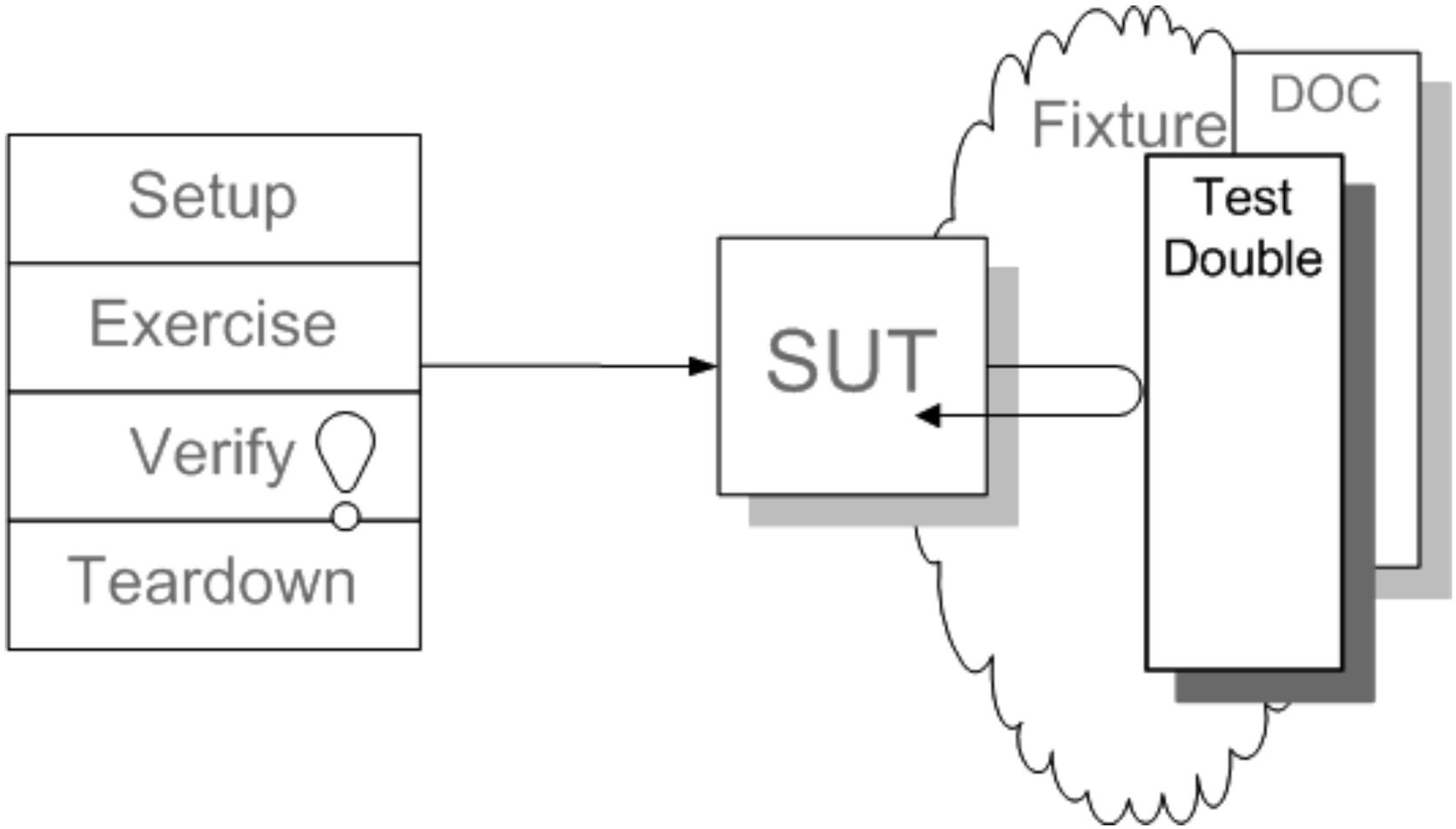
https://martinfowler.com/articles/microservice-testing

More confident



https://martinfowler.com/articles/microservice-testing

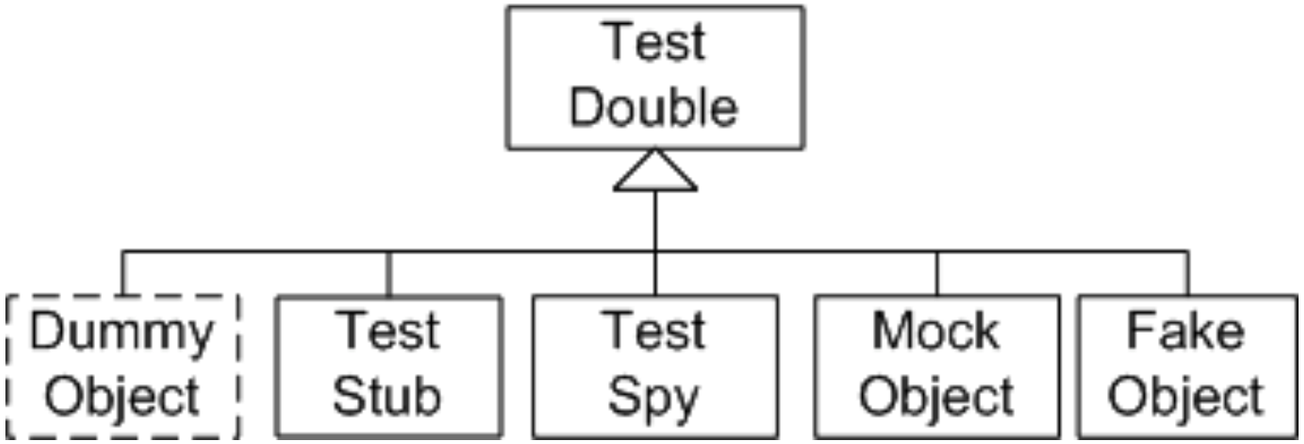
Test Double



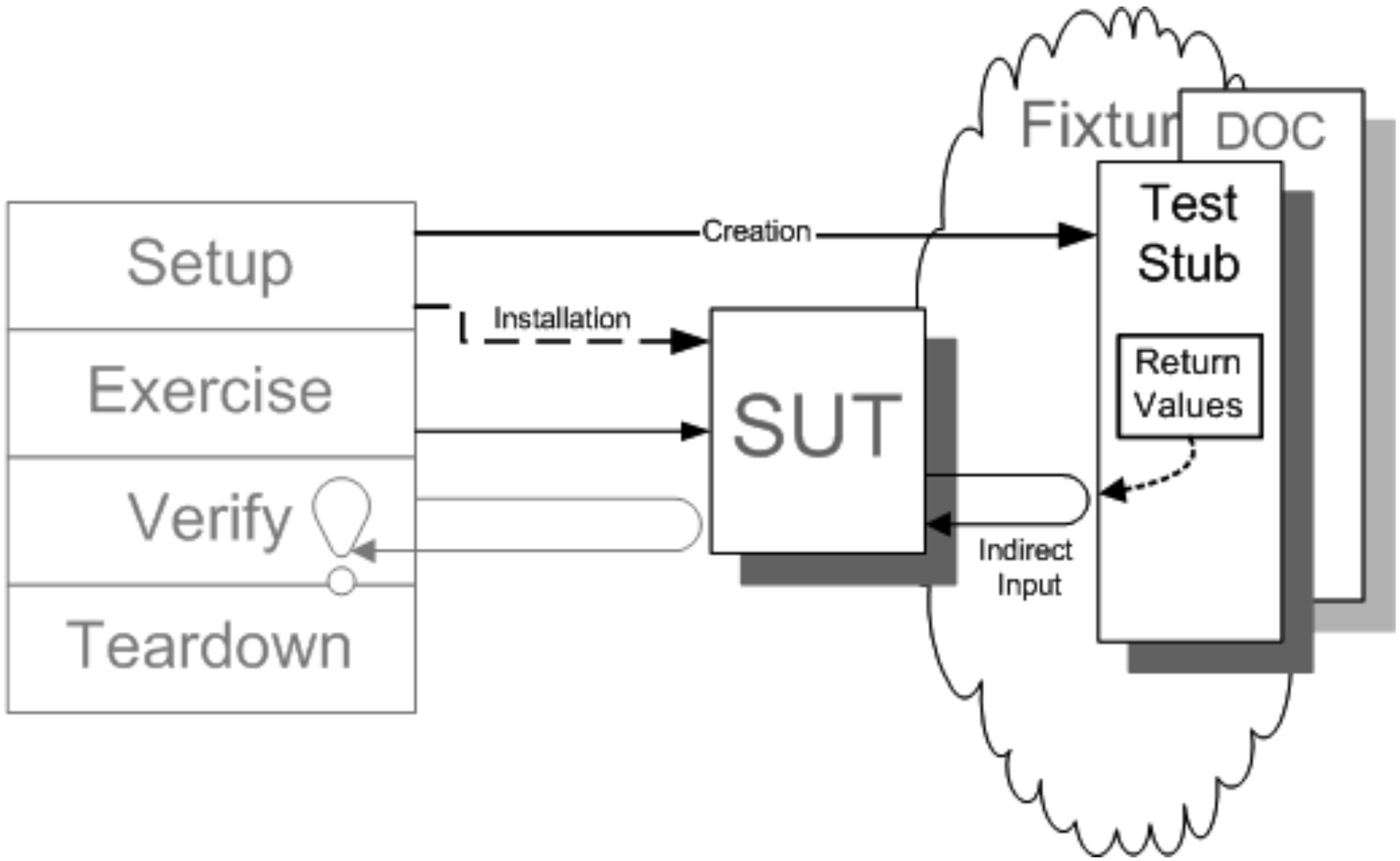
http://xunitpatterns.com/Test%20Double.html

Test Double

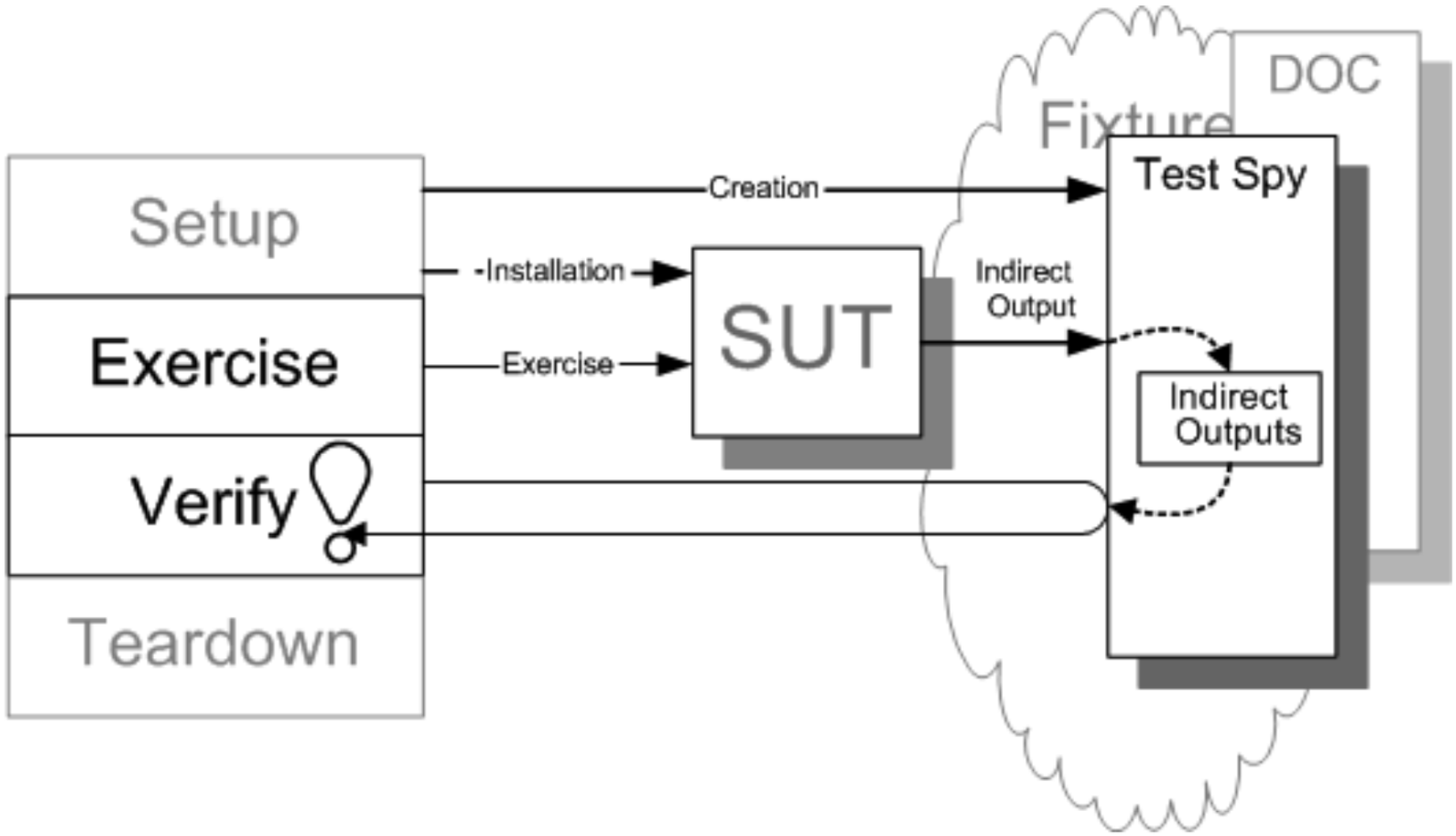
Dummy
Stub
Spy
Mock
Fake



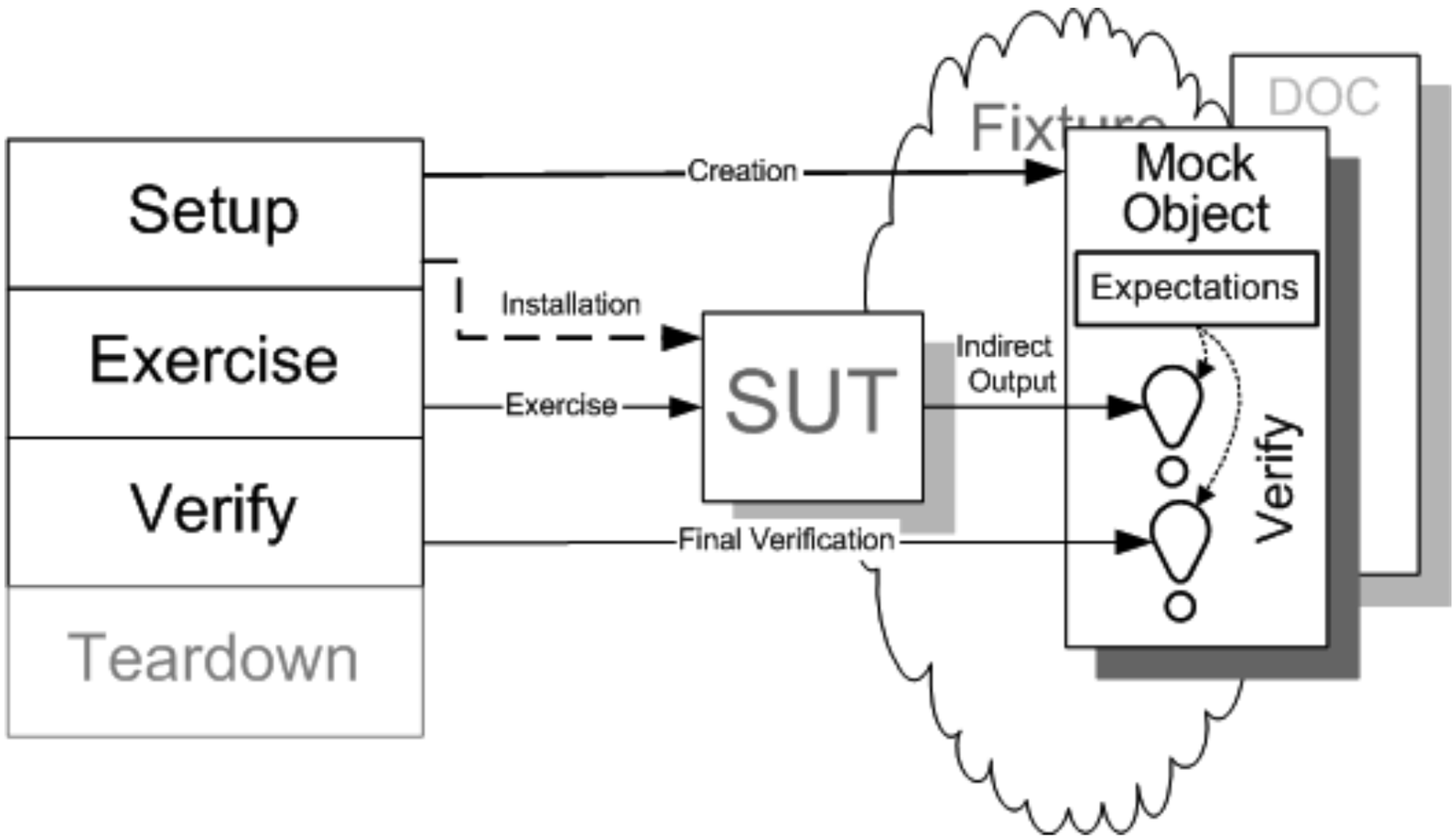
Stub



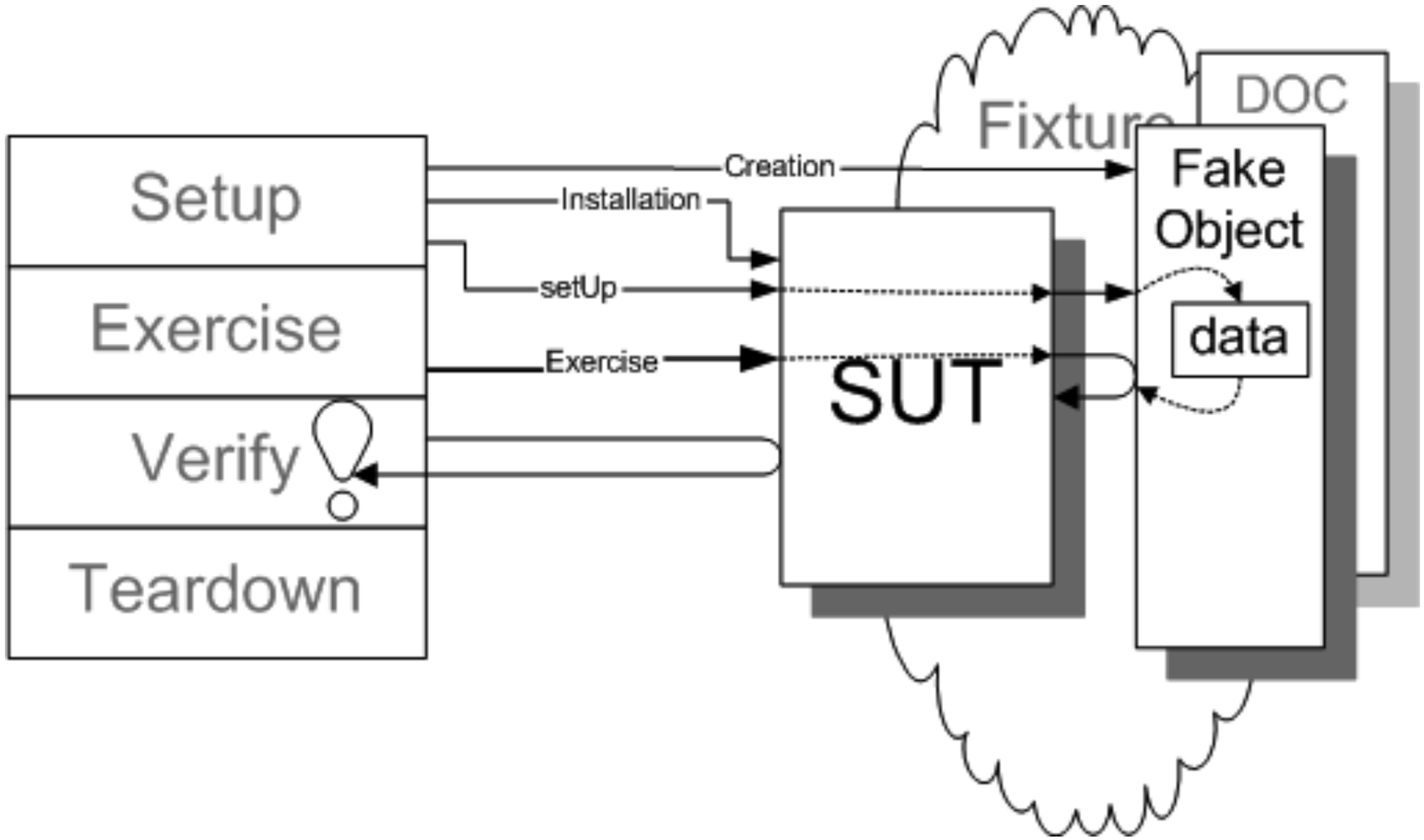
Spy



Mock

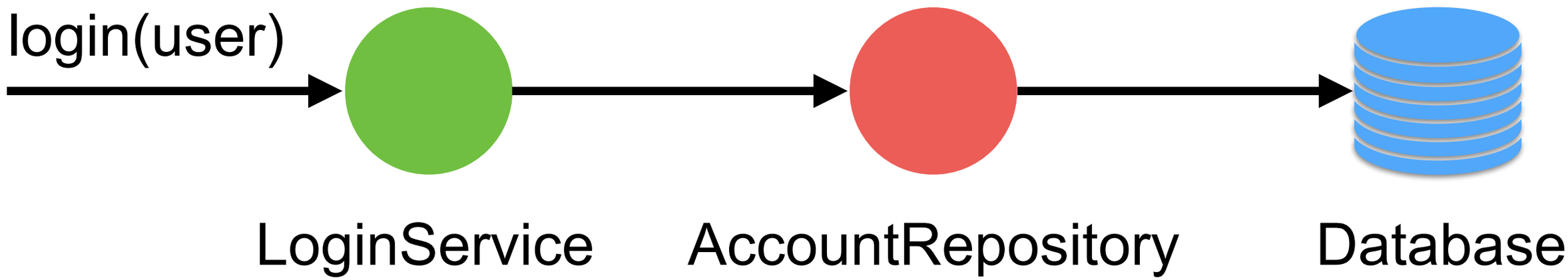


Fake

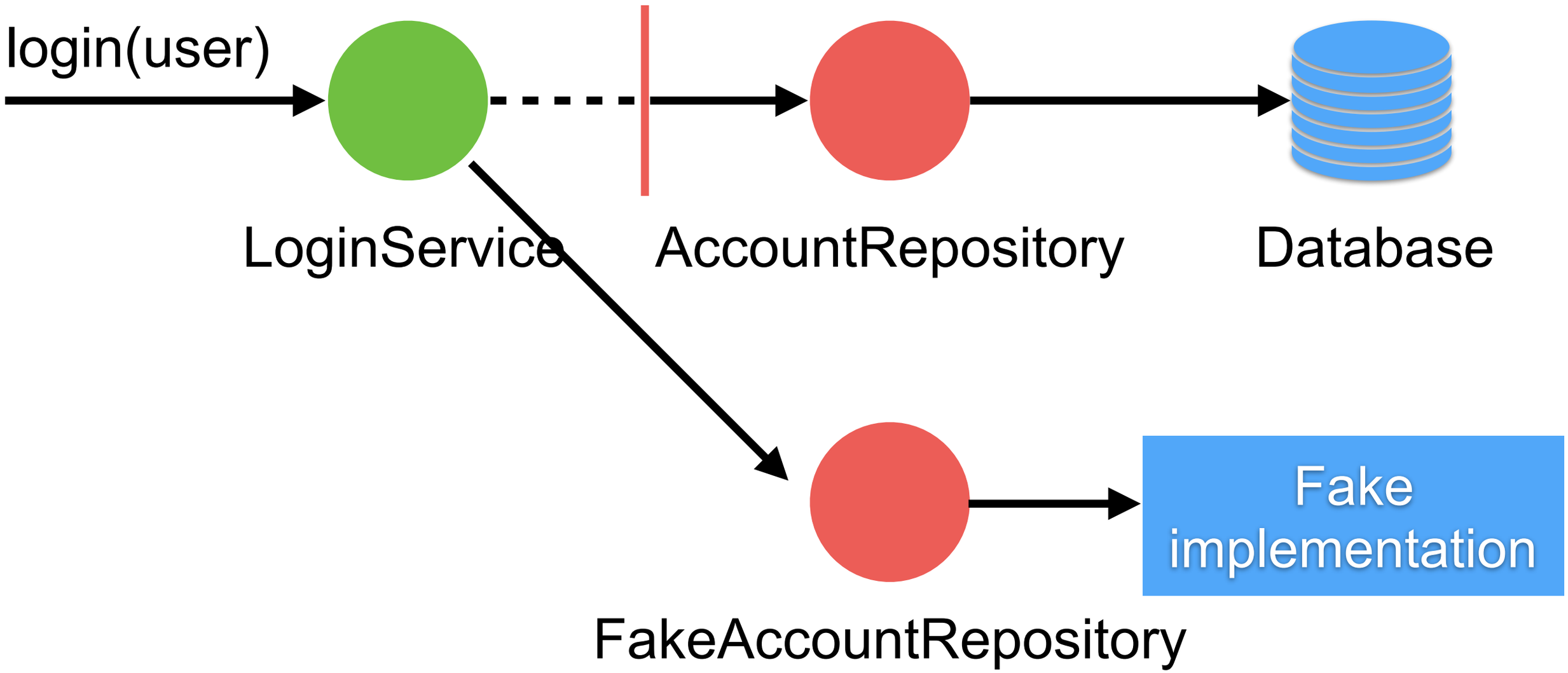


Test Double
in Software Development

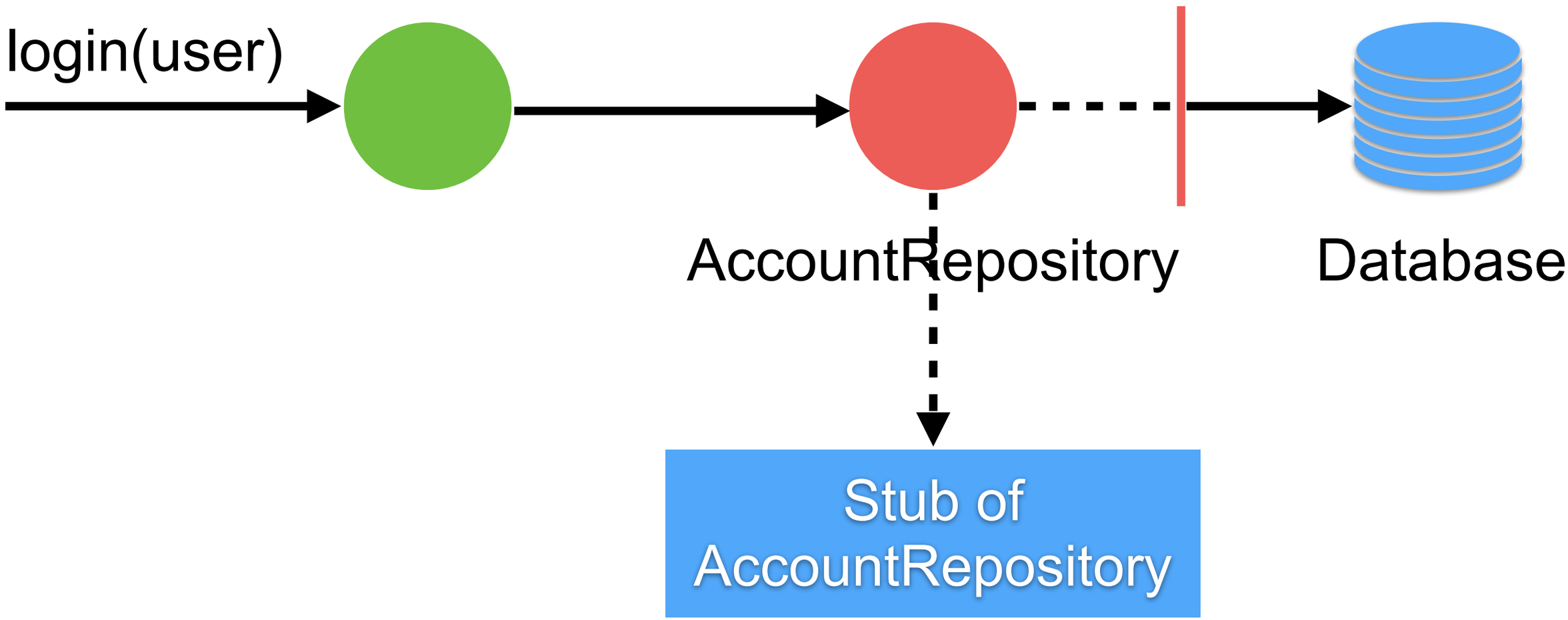
Simple flow



Fake



Stub



Working with Golang

Working with Golang

Working with Interface

Use 3-party libraries (go-sqlmock, testify, mockery)

Use container

Use mock/stub server for REST APIs

Develop

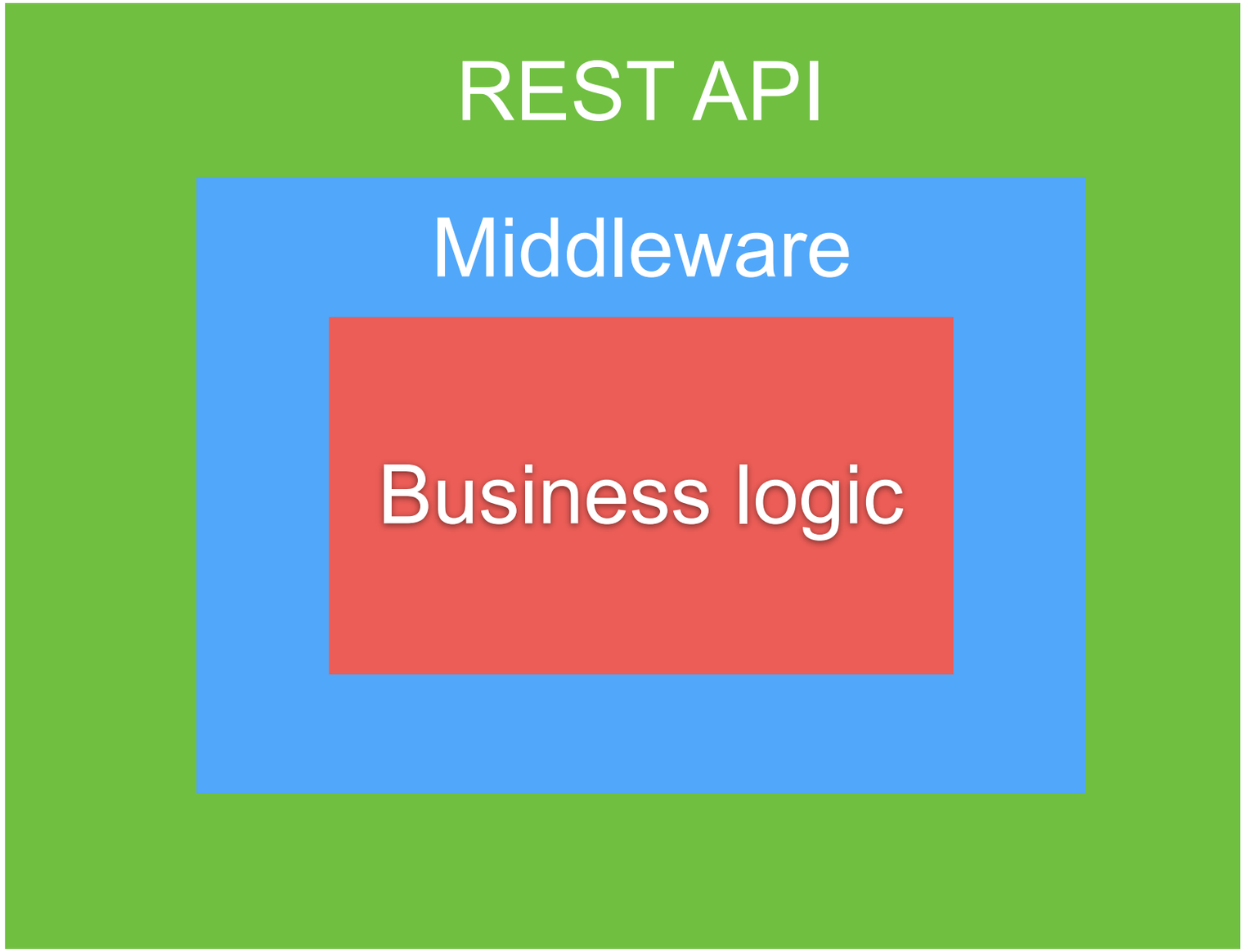
Focus on business logic

Create service via REST

Middleware (endpoint/routing)

Testing services

Develop



Develop :: Business logic

Define **interface** in each service

Create **struct**

Try to **testing**

Interface Semantics

Interface Semantics

```go
package main
import "fmt"

type printer interface {
  print()
}

type user struct {
  name string
}

func (u user) print() {
  fmt.Println("User name: ", u.name)
}
```

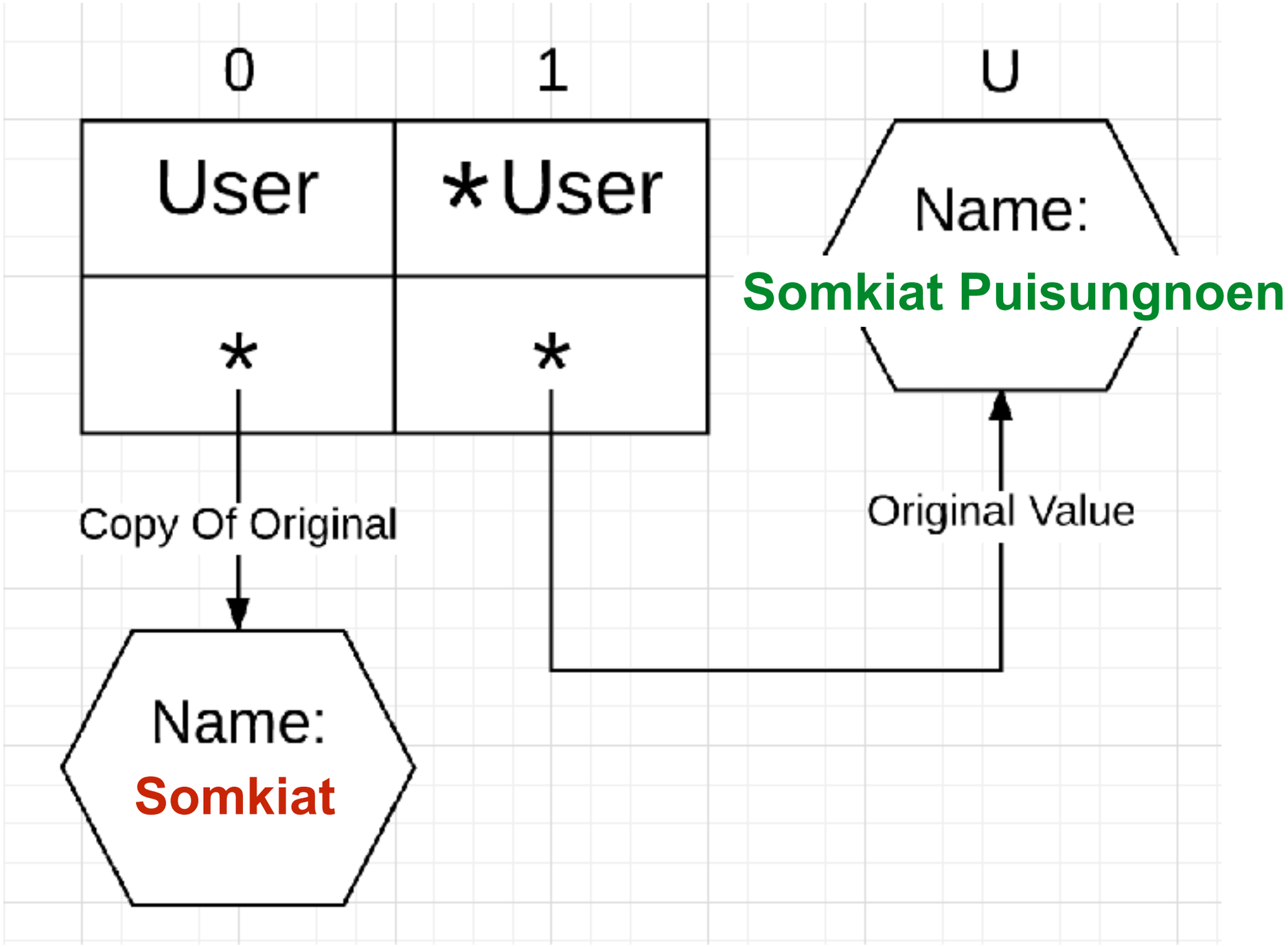
Interface Semantics

```go
func main() {
  u := user{"Somkiat"}
  datas := []printer{
    u,
    &u,
  }
  u.name = "Somkiat Puisungnoen"

  for _, data := range datas {
    data.print()
  }
}
```

https://play.golang.org/p/870xD9DBpvq

Interface Semantics (Value/Pointer)



Interface Semantics

Value semantic (Copy from original)

Pointer semantic (Original value)

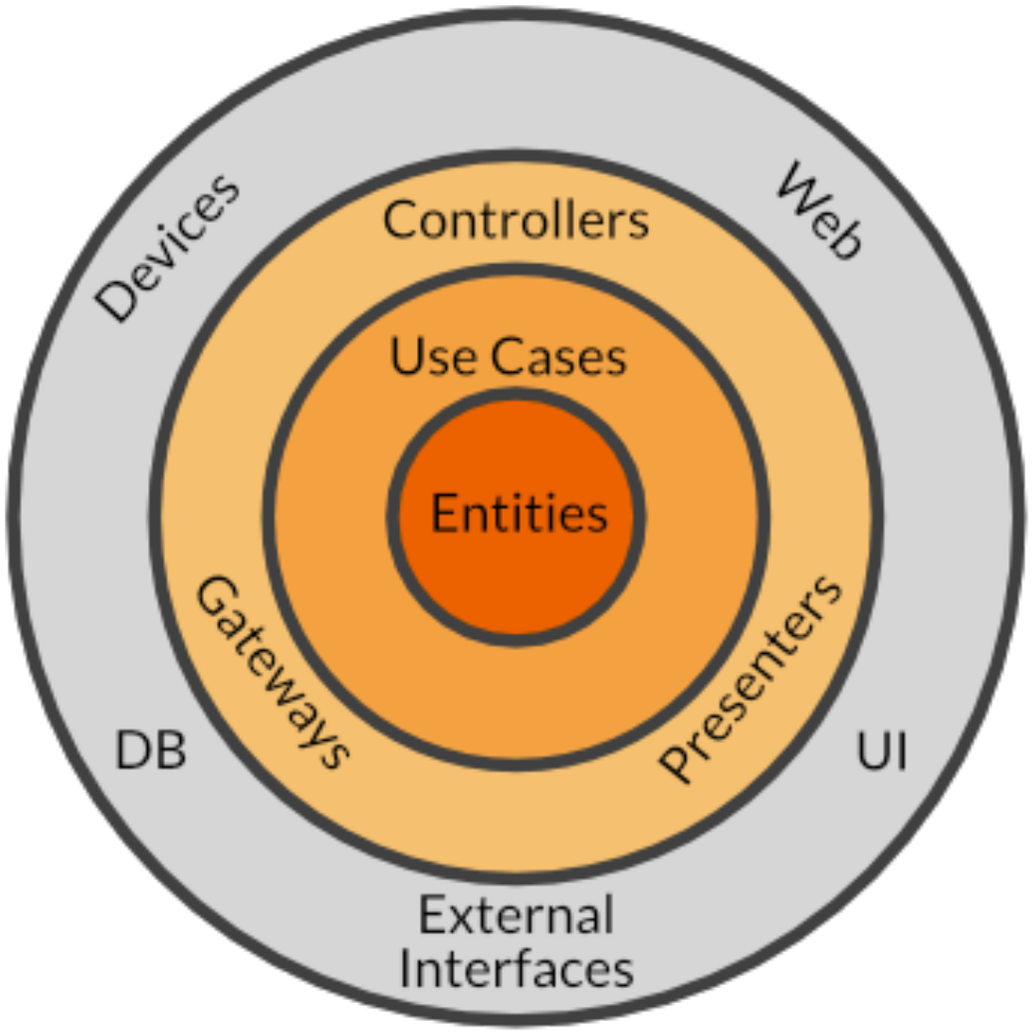
Suggestion

Choosing one semantic (consistency)

Don’t mix

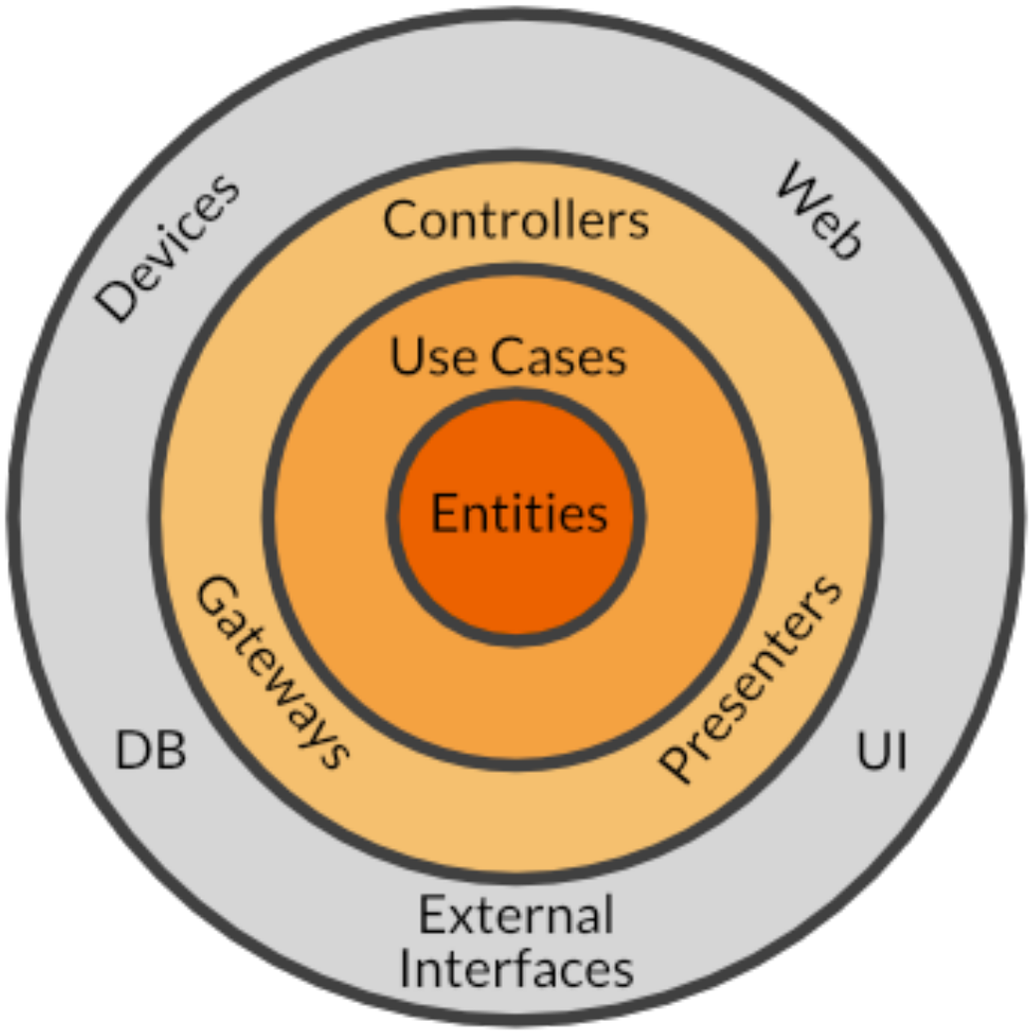
Clean Architecture

Separation of Concern (Abstraction layer)



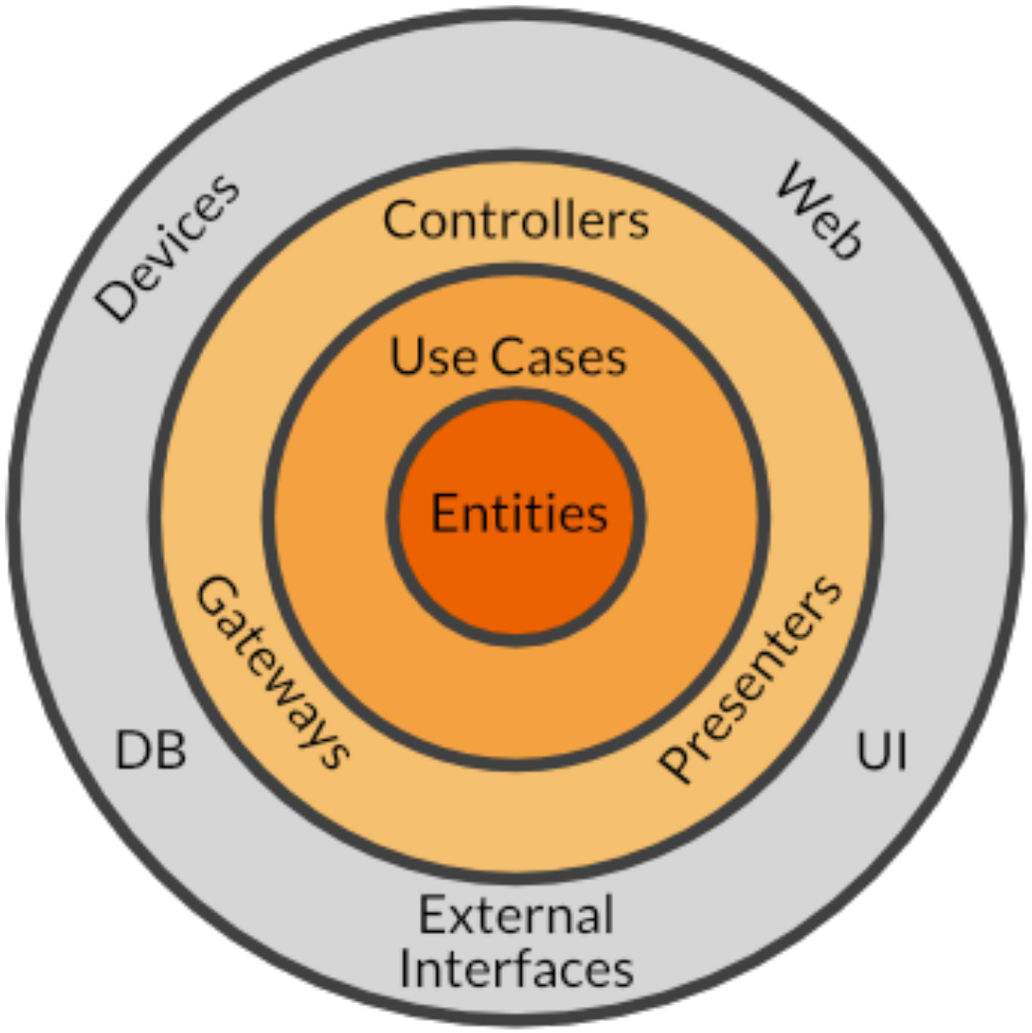
Clean Architecture

Code in each layer never access to other

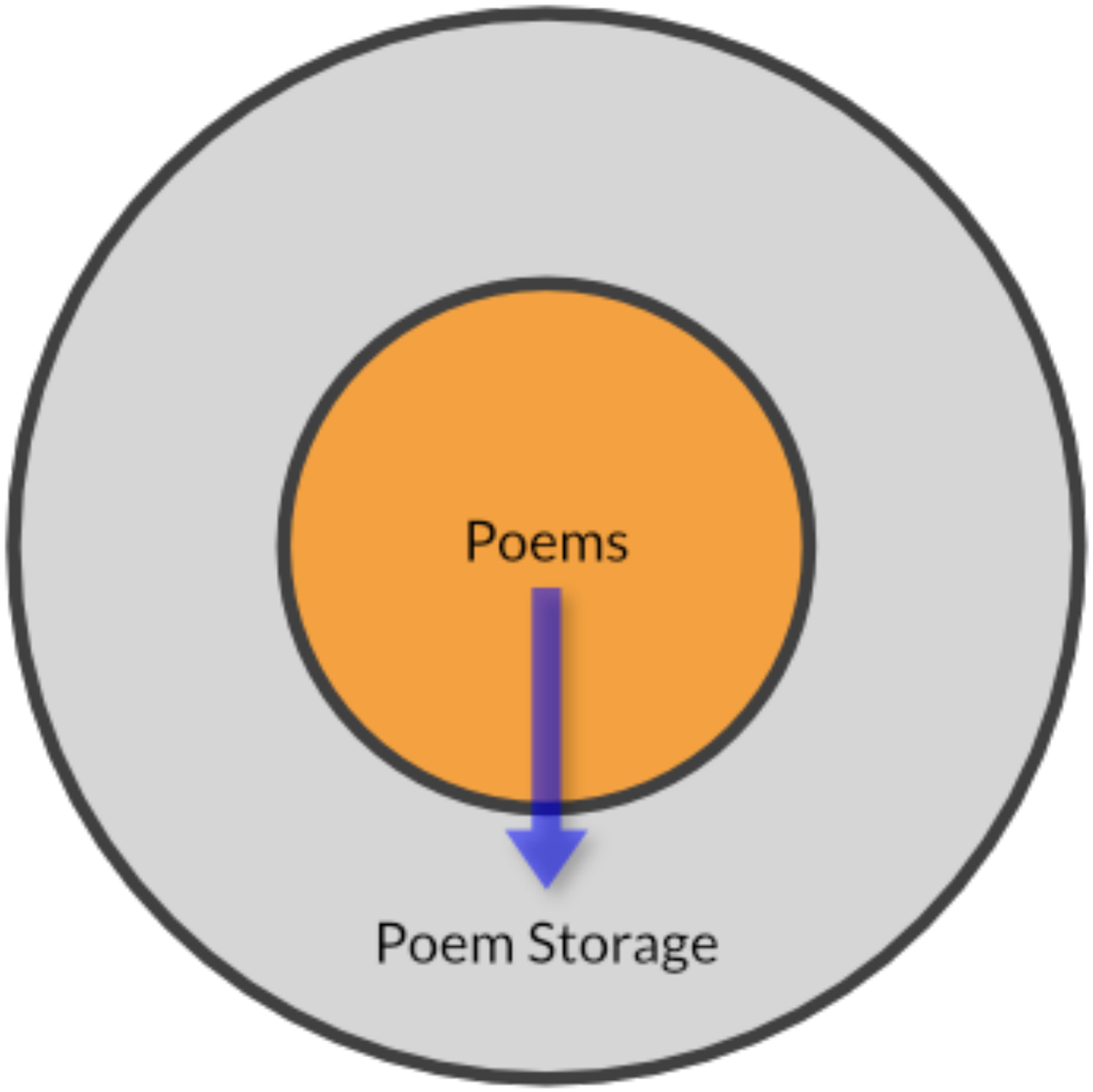


Clean Architecture

Dependency Injection(DI) Rule



Example



Bad example

```go
type PoemStorage struct {
}

type Poem struct {
  content []byte
  storage PoemStorage
}
```

Better example

```go
type PoemStorage interface {
  Load(string) []byte
  Save(string, []byte)
}

type Poem struct {
  content []byte
  storage PoemStorage
}
```

Demo/Workshop