Hexagonal

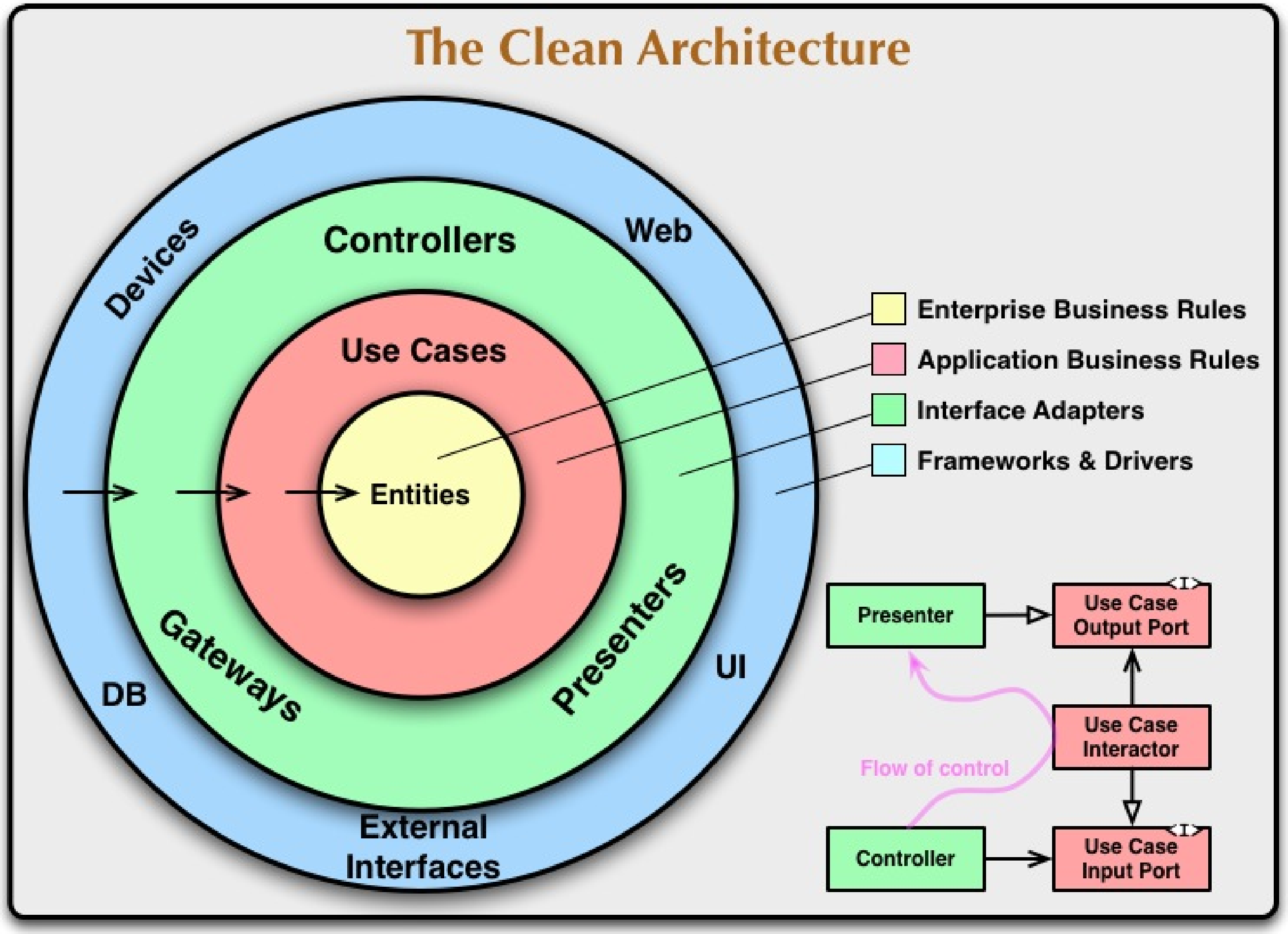
Pallat Anchaleechamaikorn

> Technical Coach
> Infinitas by KrungThai
> Arise by Infinitas

yod.pallat@gmail.com
https://github.com/pallat
https://dev.to/pallat

https://go.dev/tour (Thai)
https://github.com/uber-go/guide (Thai)

Hexagonal



https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html

image created by *Pallat Anchaleechamaikorn* **© 2023**

Why do they need the Clean Architecture?

Independent of Frameworks

Testable

Independent of UI

Independent of Database

Independent of any external agency

```go
func handler(c *gin.Context) {
    db.Exec("SELECT id, title FROM todos")


}
```

Hexagonal

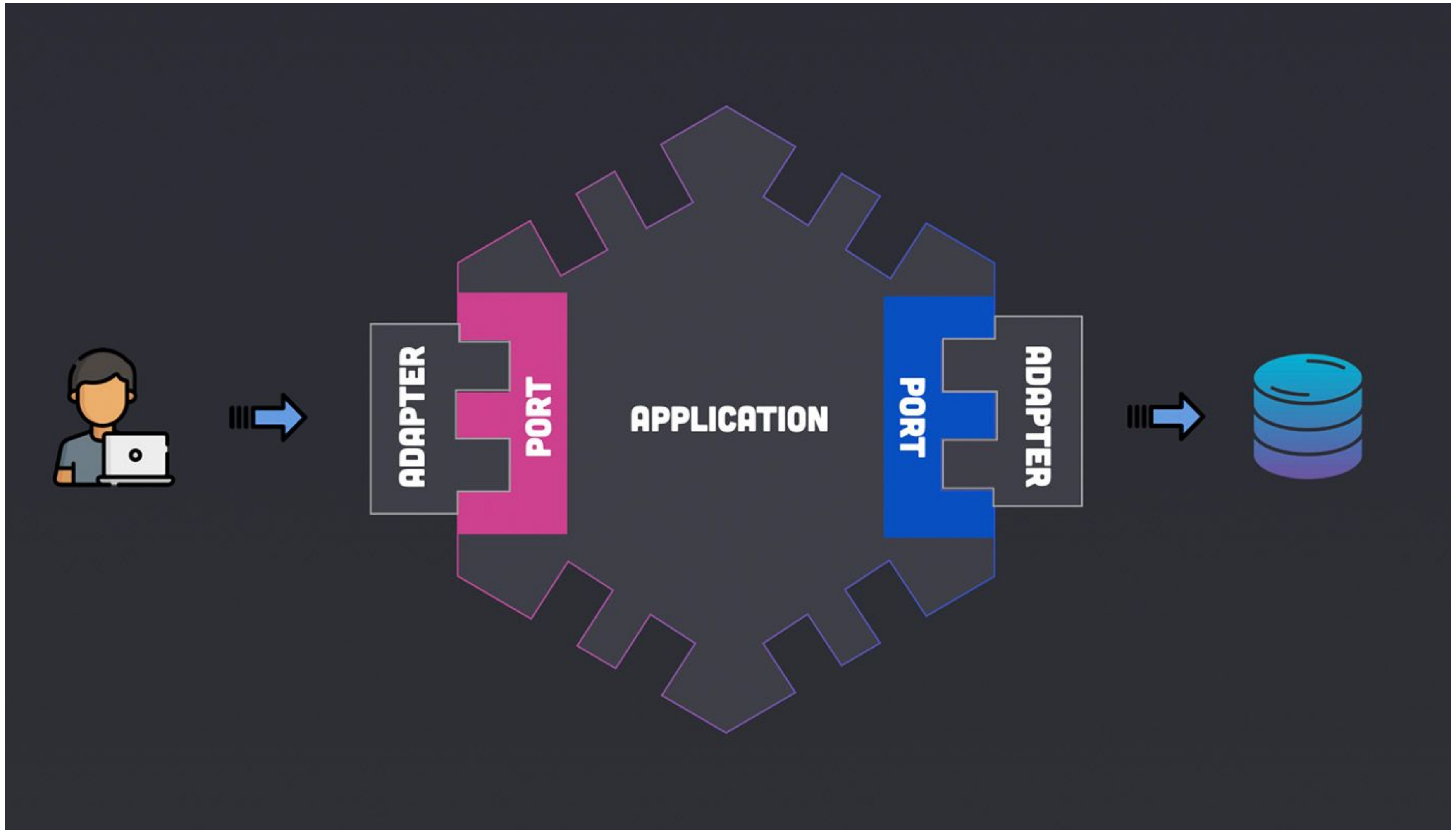


image created by *Pallat Anchaleechamaikorn* **© 2023**

Common Mistakes

- Googling `golang hexagonal architecture`
- layer package naming
- mixed architecture naming

Mixed Architecture Naming

Core

Business Logic

Repository

Model

Service

Database

Interface

Port

Adapter

Presenter

Infrastructure

Entity, etc.

layer architecture

```
go.mod
main.go
    ├─/core
    │   ├─/domain
    │   │    ├─order.go
    │   │    └─payment.go
    │   ├─/ports
    │   │    ├─repositories.go
    │   │    └─services.go
    │   ├─/services
    │   │    └─/ordersrv
    │   │         └─service.go
    ├─/handlers
    └─/repositories
```

layer architecture (2)

```
├─/bin
│   ├─/cmd
│   │    └─/http
│   │        └─main.go
│   ├─/docs
└───└─/internal
         ├─/adapter
         │    ├─/cache
         │    │    └─/redis
         │    ├─/handler
         │    │    └─/http
         │    ├─/repository
         │    │    └─/postgres
         │    │         └─/migrations
         │    ├─/token
         │    │    └─/paseto
         └─/core
              ├─/domain
              ├─/port
              ├─/service
              └─/util
```

image created by *Pallat Anchaleechamaikorn* **© 2023**

Effective Go

https://go.dev/doc/effective_go

```
package-names
interface-names
```

If hexagonal are needed

- Group it by modules
- separate functionality to its own file
- name it by functionality

Step

1. make it works
2. how to write the unit testing
3. what if we need to change some

What unable to test looks like

```go
func handler(c *gin.Context) {
    http.Get()
    time.Now()
    rand.Intn(10)
    db.QueryRow()
}
```

suggestion

```
go.mod
main.go
    ├─/[module ie. order]
    │       ├─order.go
    │       ├─handler.go
    │       ├─entity.go
    │       └─repository.go
    ├─/store
    │         ├─mongodb.go
    │         └─postgres.go
    └─/framework
              └─gin.go
```

Hexagonal

image

image created by *Pallat Anchaleechamaikorn* **© 2023**

Hexagonal

image

image created by *Pallat Anchaleechamaikorn* **© 2023**

Hexagonal

```
├─go.mod
├─main.go
├─/todo
│    ├─todo.go
│    ├─store.go
│    ├─add.go
│    ├─list.go
│    ├─update.go
│    └─delete.go
├─/store
│     └─sqlite.go
└─/framework
      └─gin.go
```

Hexagonal

image created by *Pallat Anchaleechamaikorn* **© 2023**