





**CLEAN/ONION**

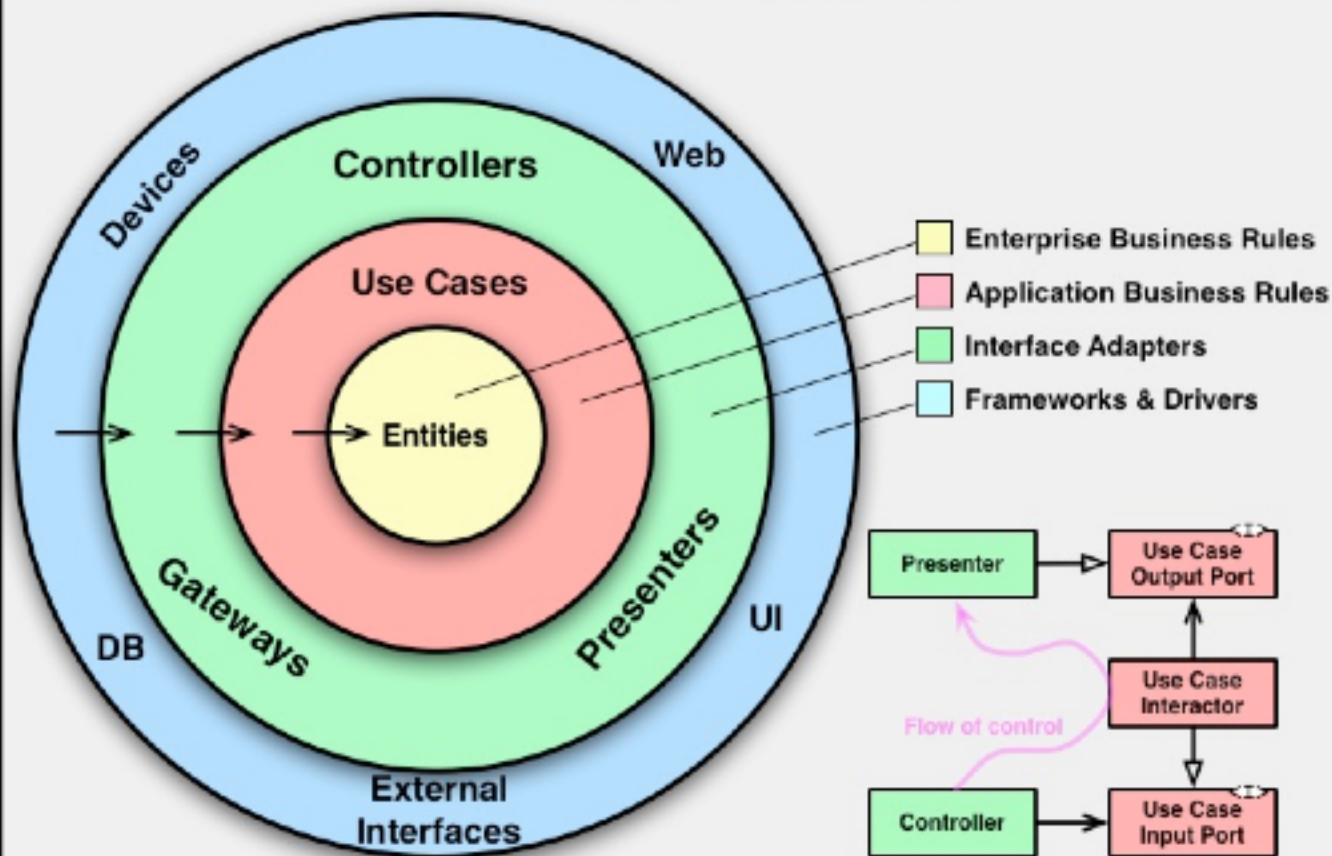


# **Objectifs est d'avoir un logiciel :**

- Indépendant des frameworks**
- Testable**
- Indépendant de l'UI**
- Indépendant de la DB**
- Indépendant des API externes**

MAGNET 2-DUALS - THOUGHTS

# The Clean Architecture



**BREF: DRY + SOLID + EFFET AUX FROGNIÈRES DU PROGRAMME**

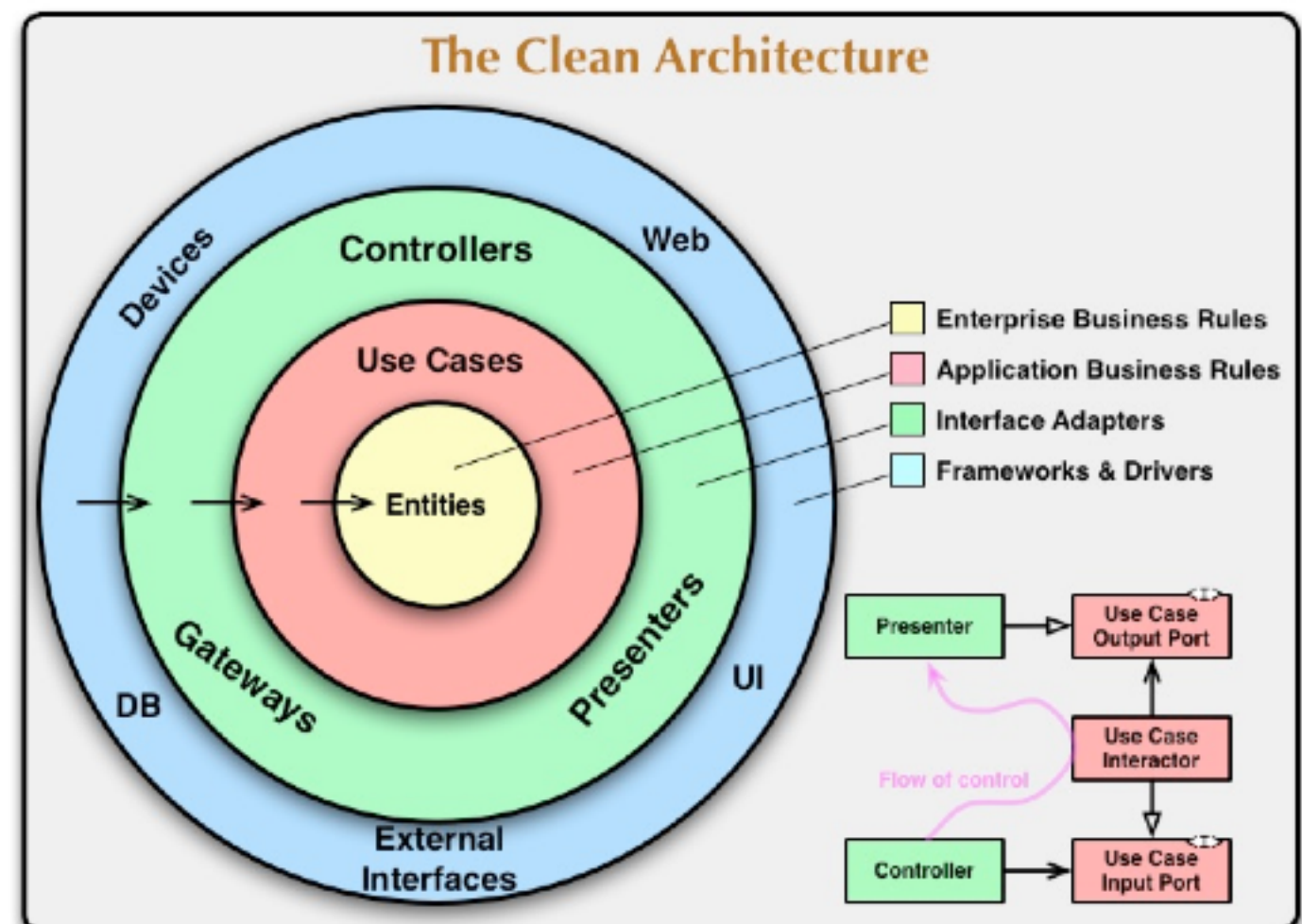


# CLEAN / ONION

UNCLE BOB ARCHITECTURE / JEFFREY PALERMO ARCHITECTURE

Objectifs est d'avoir un logiciel :

- Indépendant des frameworks
- Testable
- Indépendant de l'UI
- Indépendant de la DB
- Indépendant des API externes



BREF : DRY + SOLID + EFFETS AUX FRONTIÈRES DU PROGRAMME

# DDD

## DOMAIN DRIVEN DESIGN - A.K.A ONION MADE BUSINESS FRIENDLY

Une approche top-down :

- Le périmètre d'un logiciel (service) est borné par un domaine business
- Ubiquitous langage (celui du business)
- En appliquant SOLID
- Dans une onion architecture

