

Python vs Enterprise

Jakub Czapliński



Uwaga

To, co tu usłyszycie, jest moją opinią.

Możecie się nie zgodzić:)

Skąd ten temat?

Kontekst - co widziałem:

- Serwerownie Gratka.pl
- API AutoCAD
- Silnik StarCAD
- Chaos projektów w Unity3D
- System do sprzedaży polis
- System do likwidacji szkód
- System sprzedaży polis w UK
- Oprogramowanie z branży cyfrowego kina (Python! :D)

Częste problemy:

- Błędy po wydaniu nowej wersji
- Bardzo długi proces testowania
- Wydajność
- Skalowalność

Co zauważyłem:

- Jest zbiór problemów, które są wspólne
- Każda z branż ma unikalne dla niej problemy

Czy zawsze było tak źle?

Czego się nauczyłem:

- Pisać testy jednostkowe (takie prawdziwe)
- Aplikacje są jak ogry - mają warstwy (a przynajmniej mogą mieć)!
- Dekompozycja kodu potrafi czynić cuda!

A poza tym można uczyć się na
błędach innych! :)

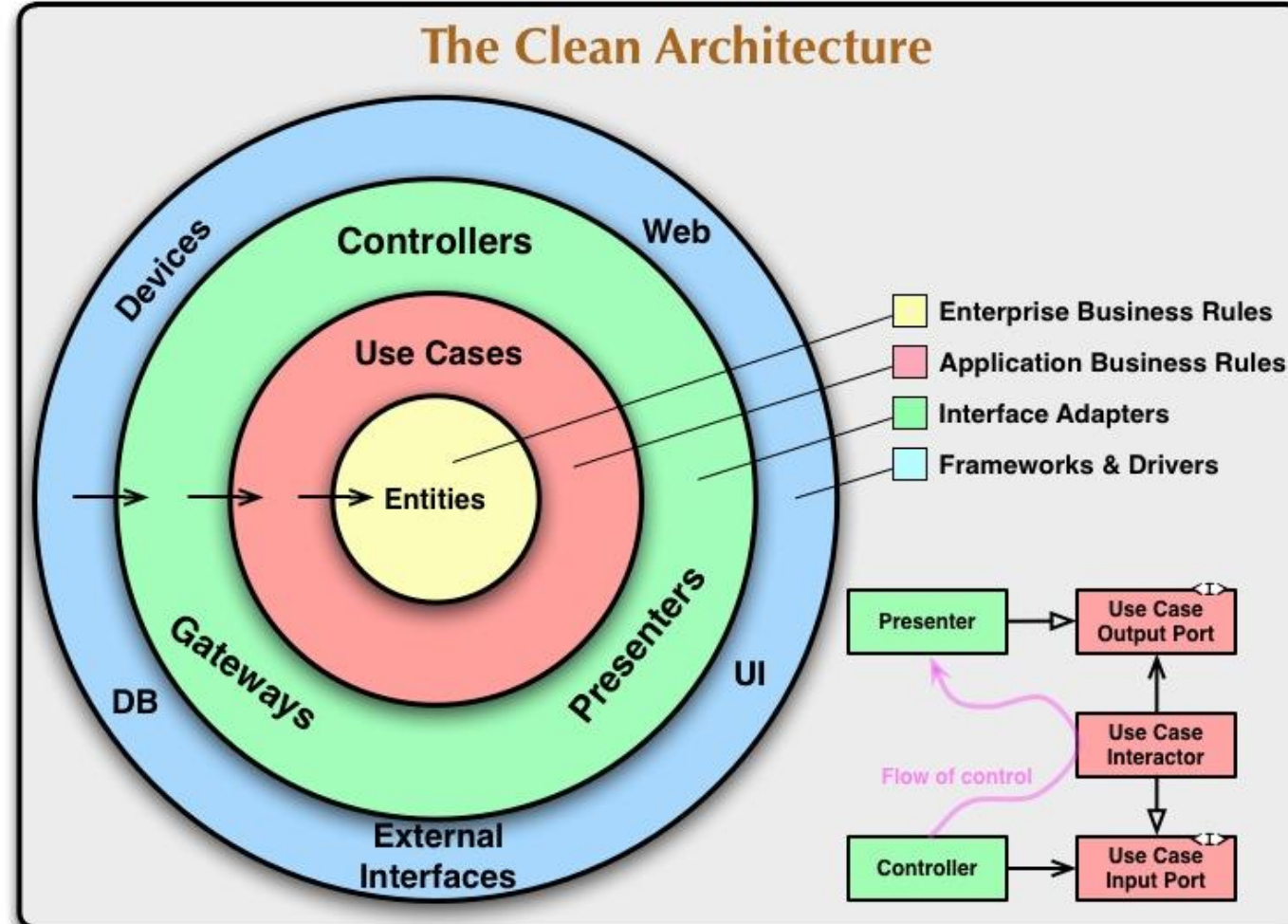
Kluczowe cechy aplikacji "enterprise"

- Rozbudowana logika biznesowa
- Długi cykl życia produktu

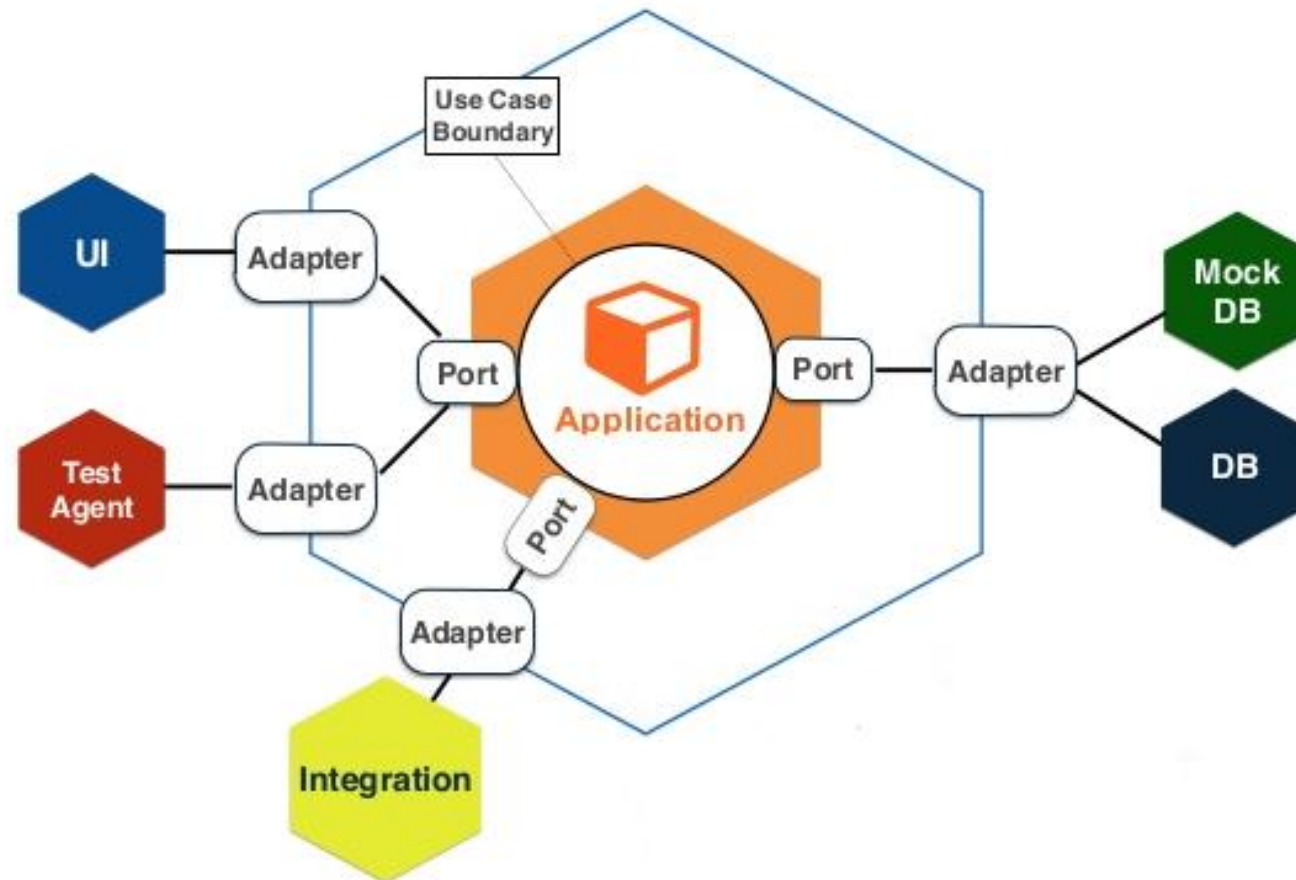
Co jest istotne w aplikacji z łątką "enterprise":

- Łatwość utrzymania aplikacji
- Możliwość łatwego wprowadzania zmian
- Dodawanie nowych funkcjonalności
- Integrowanie z nowymi usługami
- Testowalność

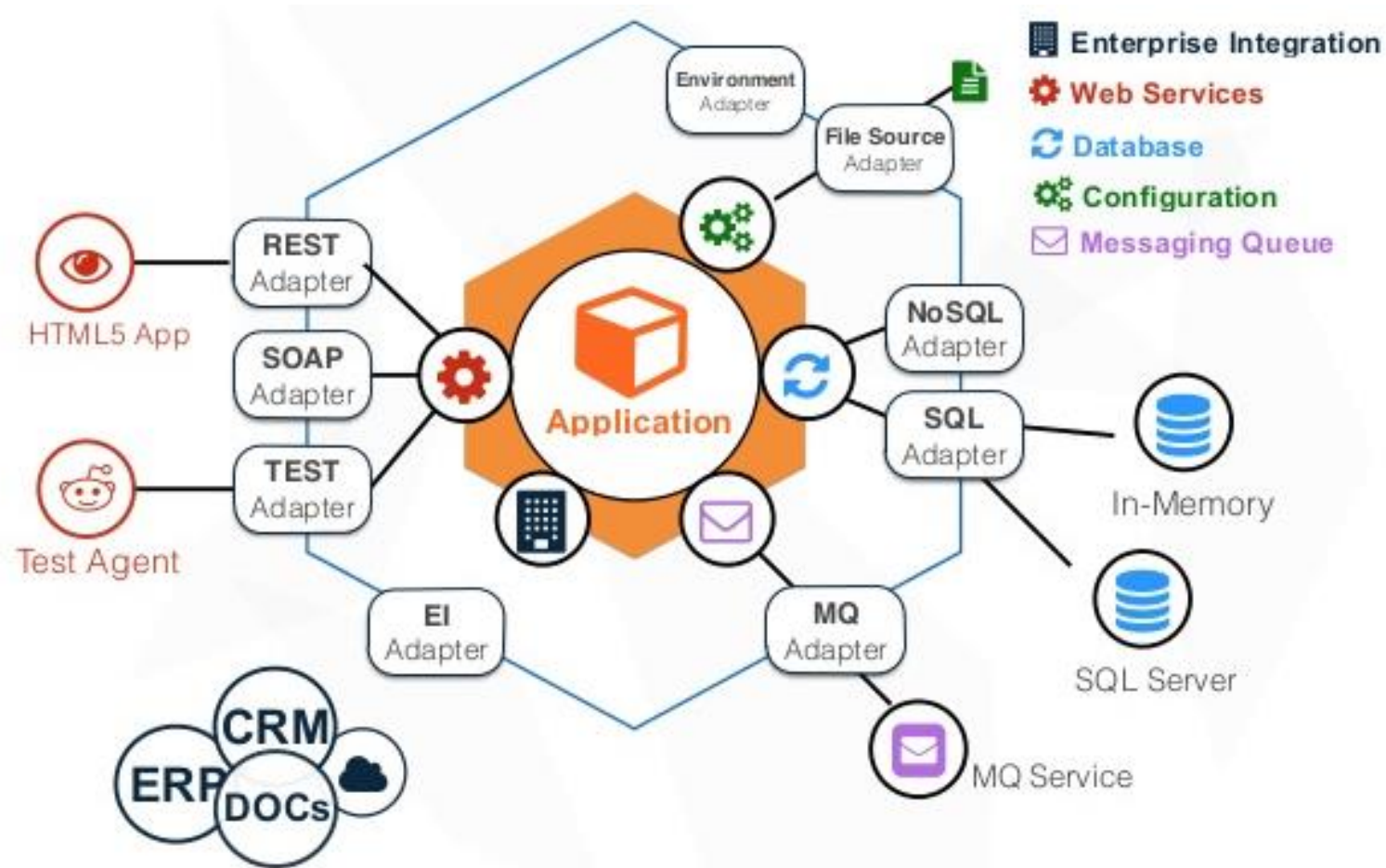
Clean Architecture



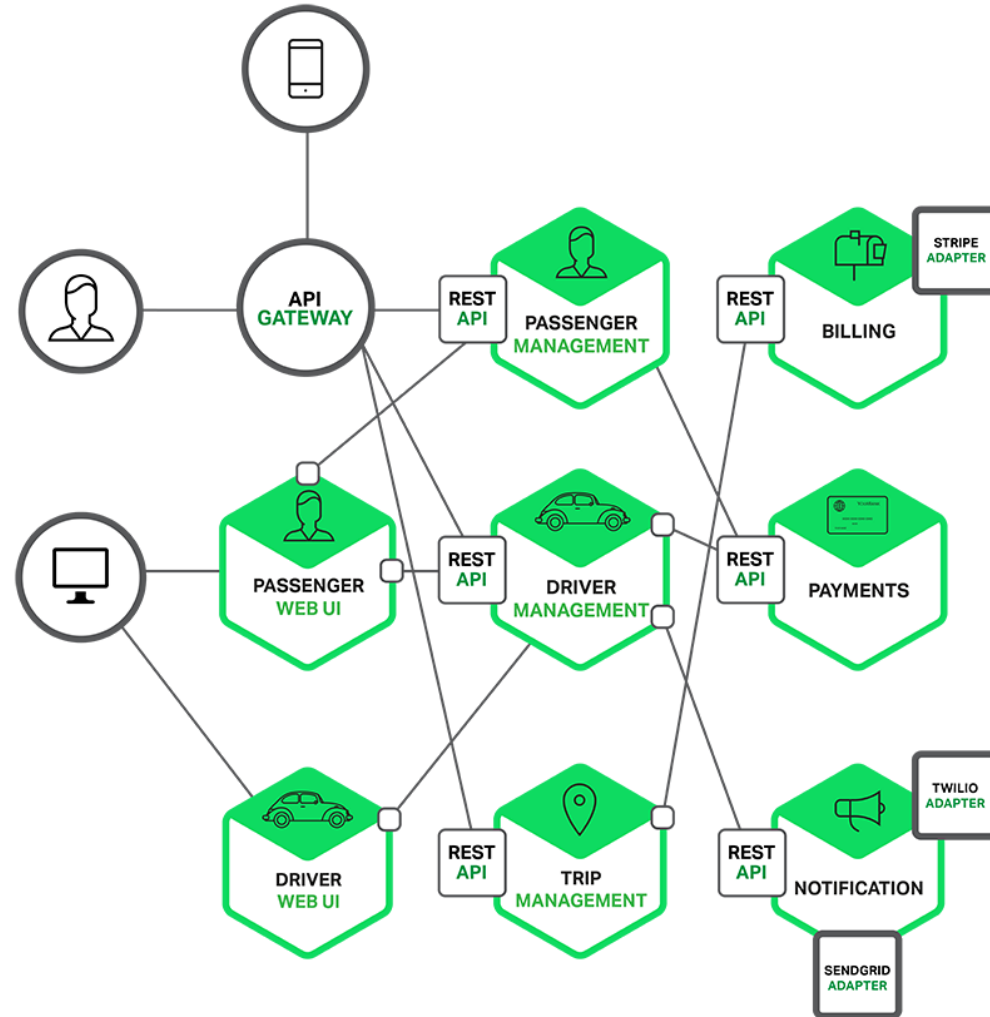
Hexagonal Architecture



Hexagonal Architecture



Microservice Architecture



Wspólny mianownik?

Dekompozycja

Dependency injection

Patch!?

Często pomijane

Wzorce projektowe

SOLID

DI/IOC (Angular JS)

To co świetnie sprawdza się w
małych projektach, niekoniecznie
działa w dużych

Pythona ma fantastyczny arsenał narzędzi:

- context manager
- generator
- dekorator
- duck typing

I to czasem powoduje że stajemy się leniwi.

Jeśli trzymasz w ręku młotek, to
wszystkie twoje problemy
zaczynają wyglądać jak gwoździe

Dzięki!



Pytania?



Linki

Warstwy

<https://martinfowler.com/bliki/PresentationDomainDataLayering.html>

Hexagonal architecture

<http://alistair.cockburn.us/Hexagonal+architecture>

<https://www.slideshare.net/fabricioepa/hexagonal-architecture-for-java-applications>

Clean Architecture

<https://youtu.be/Nsjsiz2A9mg>

<https://youtu.be/DJtef410XaM>

Linki

Design patterns:

<https://youtu.be/PBQN62oUnN8>

https://youtu.be/Er5K_nR5IDQ

Unit testing:

<https://youtu.be/wEhu57pih5w>

IoC / DI

<https://www.martinfowler.com/articles/injection.html>



PGS Software S.A.

Tel.: +48 71 79 82 692

Fax: +48 71 79 82 690

E-mail: pgs-soft@pgs-soft.com

www.pgs-soft.com

