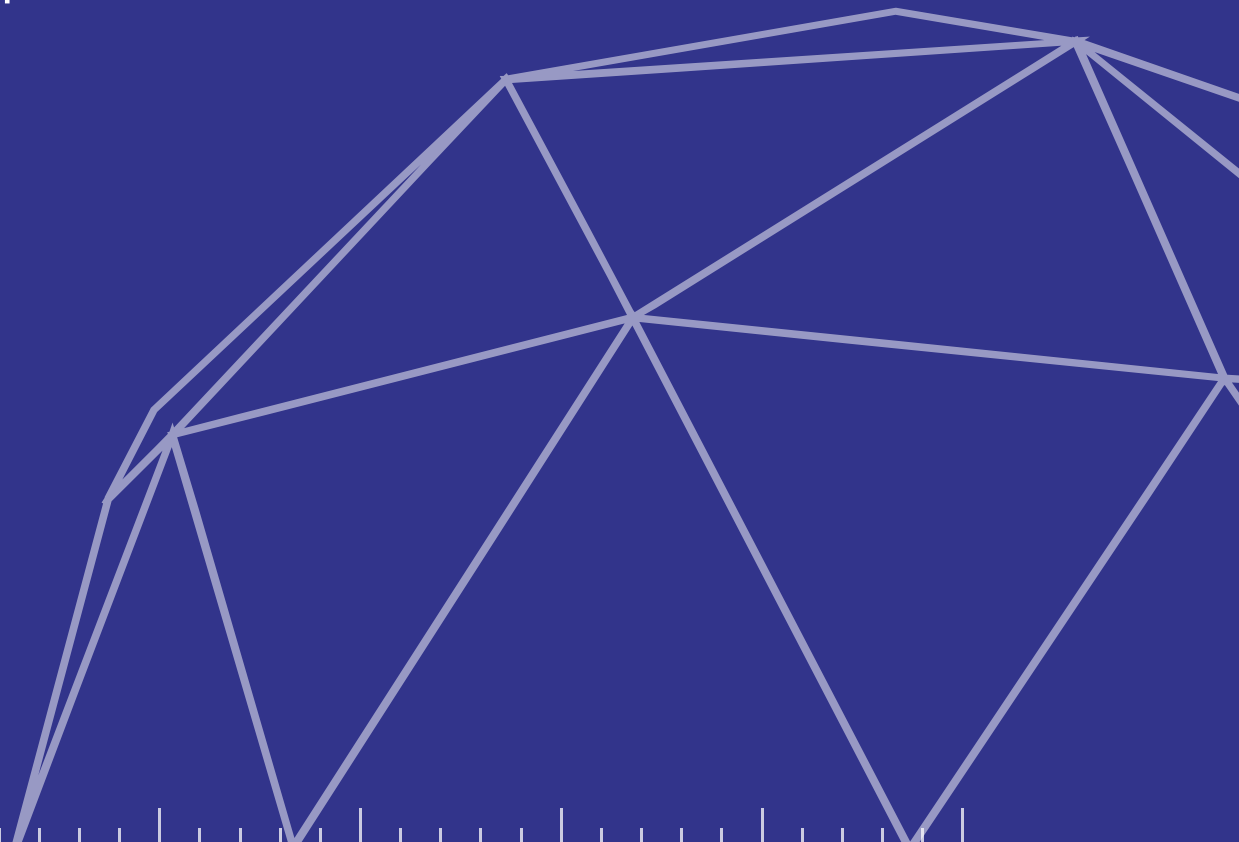
INNA PERSPEKTYWA NA ORGANIZACJĘ NAMESPACE'ÓW W WARSTWIE INFRASTRUKTURY



Zbigniew @Ferror Malcherczyk

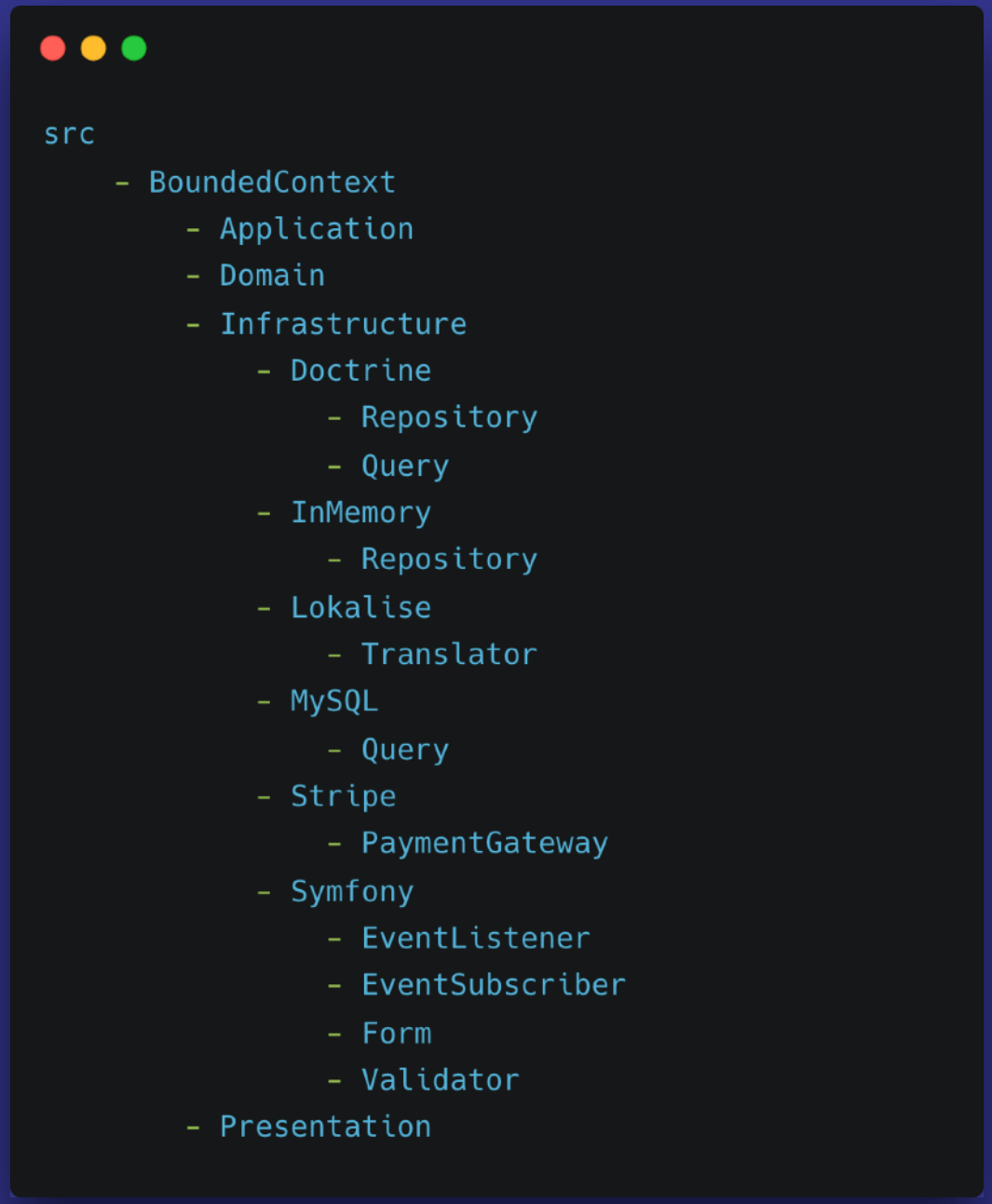
PHPERS SUMMIT 2022

Symfony’s Best Practices directory structure

TYPE - BASED NAMESPACES

Klasy komponentów Symfony organizowane są według Symfony-Best-Practices, które można określić jako przestrzenie nazw zorientowane na typ klasy.

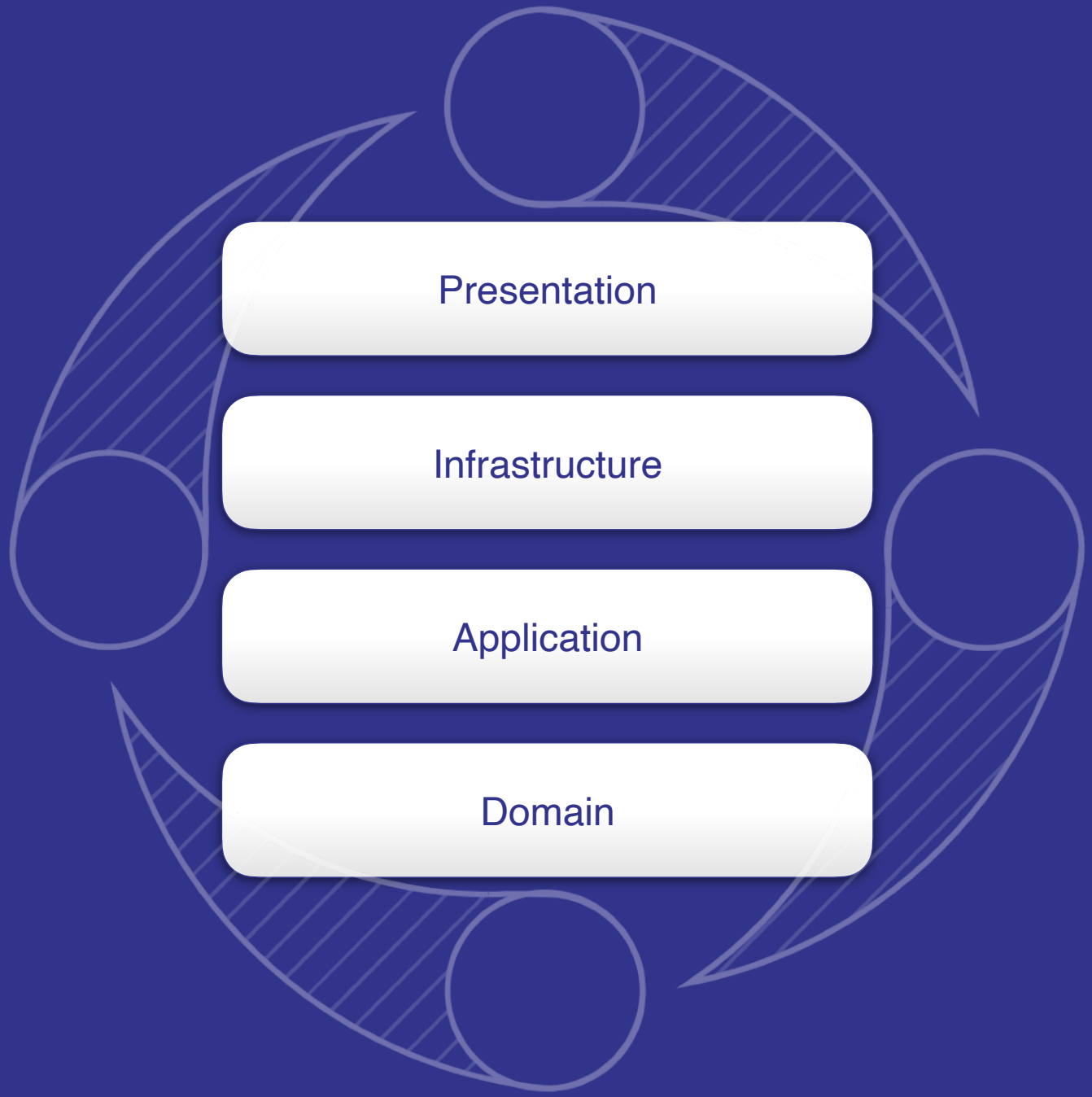
Taka organizacja klas wspiera tworzenie aplikacji o architekturze monolitycznej.



INFRASTRUCTURE NAMESPACE

Alternatywą jest organizacja przestrzeni nazw w architekturze warstwowej, zorientowanej na typ dostawcy infrastruktury.

Wady i zalety trochę później.



LAYERED ARCHITECTURE

Brak zasad w perspektywie czasu sprawia, że projekt dąży do tzw. Big Ball of Mud.

Rozwiązaniem jest architektura warstwowa, która wprowadza zasady.

Każda z warstw może się komunikować sama z sobą i warstwami znajdującymi się poniżej. To znaczy, że warstwa niższa nie wie nic o warstwie wyżej.

phparkitect / arkitect Public

```php
Rule::allClasses()
    ->that(new ResideInOneOfTheseNamespaces('App\Controller'))
    ->should(new HaveNameMatching('*Controller'))
    ->because("it's a symfony naming convention")
```

```yaml
deptrac:
  paths: ["./ModelController1"]
  exclude_files: []
  layers:
    - name: Models
      collectors:
        - type: className
          value: .*MyNamespace\\Models\\.*
    - name: Controller
      collectors:
        - type: className
          value: .*MyNamespace\\.*Controller.*
  ruleset: []
```

TOOLS THAT WILL KEEP YOUR RULES

Posiadanie zdefiniowanych reguł to pierwszy krok do sukcesu. Następnie należy zacząć je respektować.

Narzędzia do analizy reguł przestrzeni nazw, takie jak **PHPAT**, **Deptrac** oraz **phparkitect**.

https://github.com/ferror/ddd-tools-config

INFRASTRUCTURE LAYER ROLE

Zadaniem warstwy infrastruktury jest dostarczanie możliwości technicznych dla innych części aplikacji. […] Sprawdza się ona dobrze wtedy, kiedy jej komponenty zależą od interfejsów zdefiniowanych w pozostałych warstwach.

~Vaughn Vernon

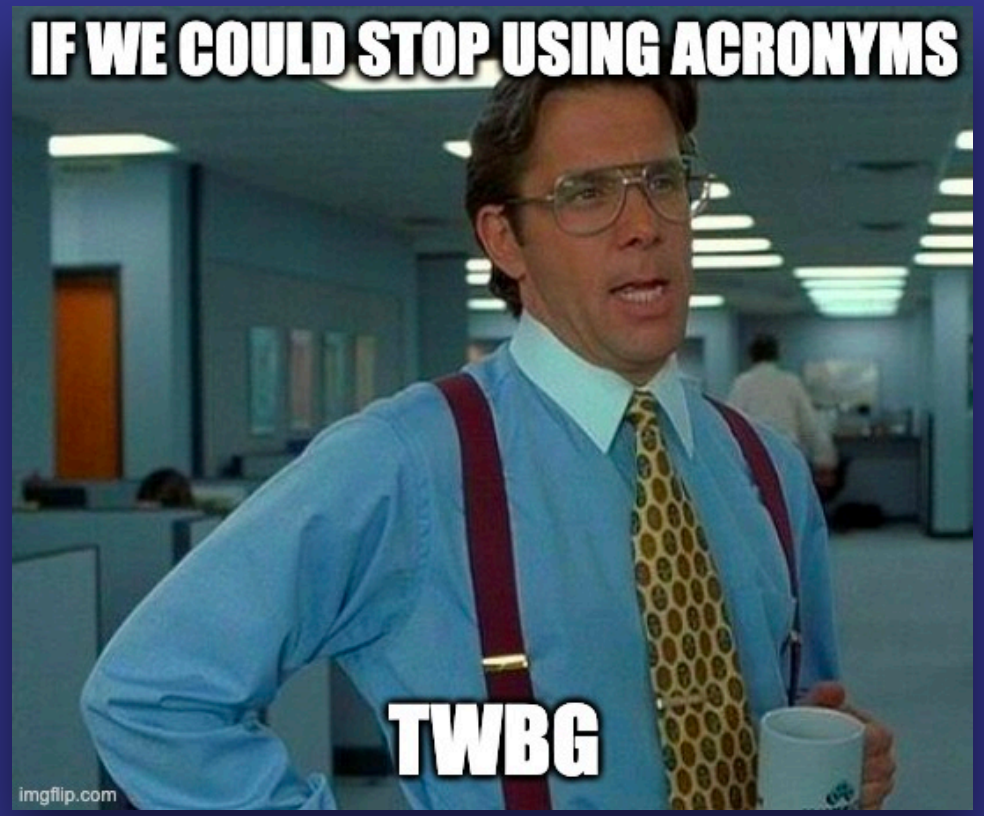
PRESENTATION LAYER ROLE

Vaughn Vernon opisuje również czwartą warstwę. Warstwa interfejsu użytkownika zawiera kod związany jedynie z widokiem prezentowanym użytkownikowi oraz kierowanymi przez niego żądaniami.

USE CASE

```
src
    - Shop
        - Application
            - Product
                - ProductAvailabilityService
                - ProductPricingService
        - Domain
            - Product
            - ProductRepository
            - ProductQuery
        - Presentation
            - Console
                - ProductCommand
            - Web
                - ProductController
```

```
src
    - Shop
        - Infrastructure
            - Doctrine
                - Repository
                    - DoctrineProductRepository
            - InMemory
                - Repository
                    - InMemoryProductRepository
                - Query
                    - InMemoryProductQuery
            - MySQL
                - Query
                    - MySQLProductQuery
```



VO or VO?

Domain-Driven Design wprowadza pojęcie Bounded Context. Oznacza ono grupowanie procesów oraz funkcjonalności w te zależne od siebie lub działające w wspólnym celu.

Akronimy w kodzie mogą działać negatywnie na czytelność i klarowność kodu. Skrót w zależności od kontekstu oraz wiedzy może mieć inne znaczenie.

PROS & CONS

- Separacja dostawców zależności
- Prostsze usunięcie oraz dodanie dostawcy zależności
- Wymienialność zależności
- Widzimy z jakich zależności/usług korzystamy
- Wspiera podejście Framework Agnostic
- Zapobiega powstawaniu Big Ball of Mud

- Początkowa konsternacja
- Brak możliwości łączenia dostawców zależności
- Przestrzeganie reguł

REFERENCES

- https://symfony.com/doc/current/bundles/best_practices.html#directory-structure
- https://github.com/phparkitect/arkitect
- https://github.com/qossmic/deptrac
- https://herbertograca.com/2017/08/03/layered-architecture/
- Implementing Domain-driven Design - Voughn Vernon
- Domain-Driven Design: Tackling Complexity in the Heart of Software - Eric Evans

Zbigniew Malcherczyk
Ferror

Edit profile

21 followers · **41** following

Poland, Silesia
zmalcherczyk@gmail.com
https://malcherczyk.pl

Achievements

x4

Organizations

THE END

Let’s grab a coffee!

Wanna share some feedback?