

SYMFONY FLEX

NOWE, EFEKTYWNE PODEJŚCIE DO
BUDOWANIA APLIKACJI W SYMFONY

Krzysztof Wędrowicz



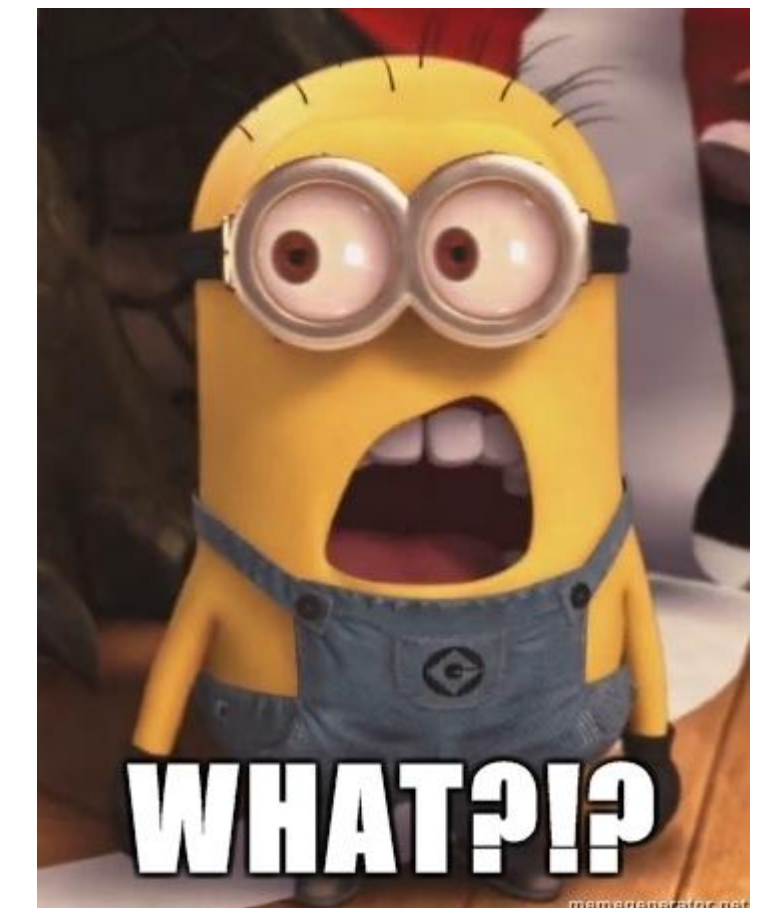
CZY JEST RÓŻNICA?

Stare podejście:

1. composer req xxx
2. AppKernel
3. parameters.yml.dist + parameters.yml
4. config.yml
5. routing

Nowe podejście:

1. composer req xxx



PLAN PREZENTACJI

1. Czym jest Symfony Flex i dlaczego warto z niego korzystać?
2. Repozytoria i receptury
3. Demo

UWAGA, KONKURS!



CZYM JEST SYMFONY FLEX?



COMPOSER PLUGIN

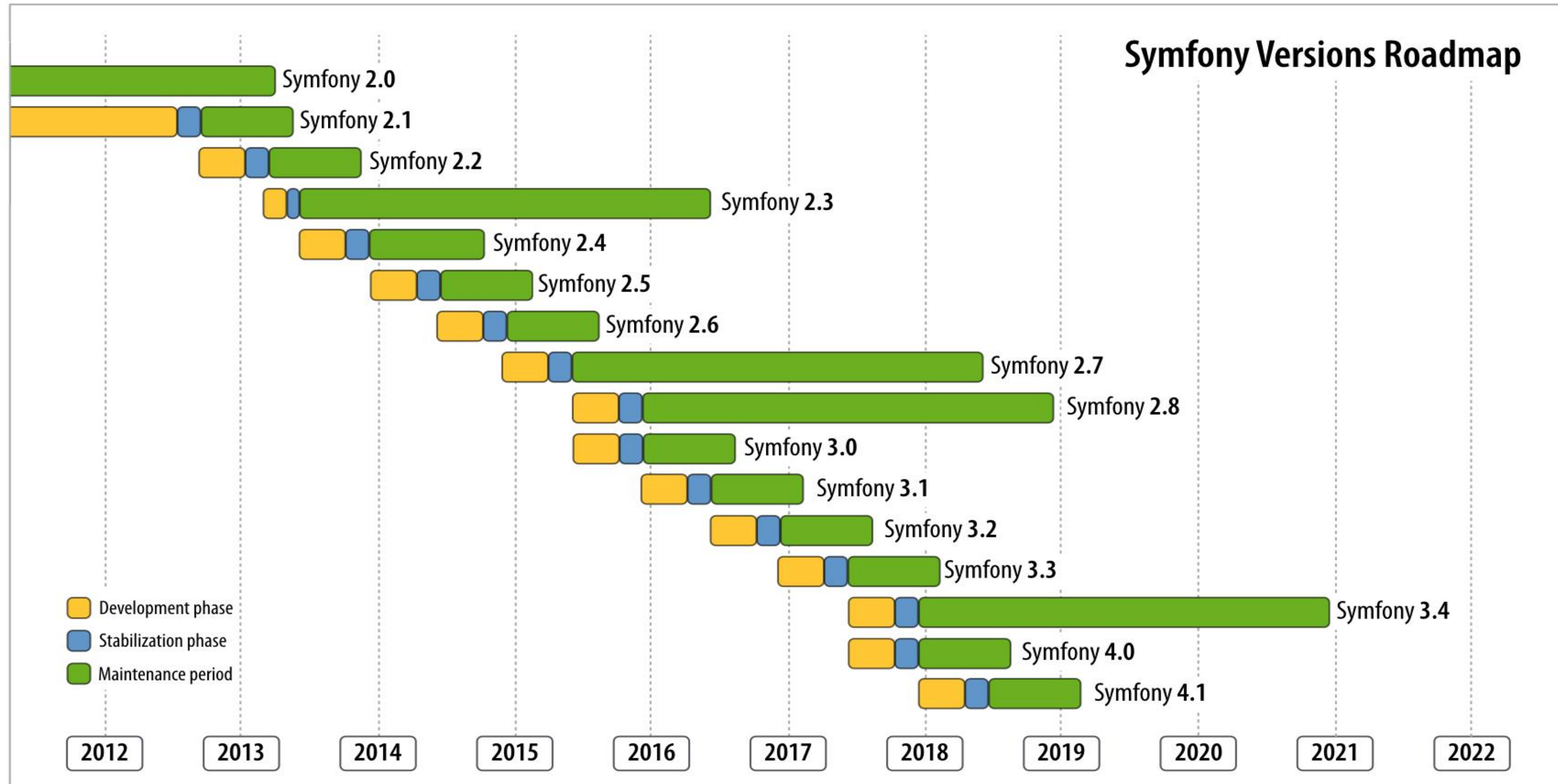


- zwykła paczka do composera wymagająca przynajmniej composer-plugin-api
- zmienia domyślne zachowanie composera
- umożliwia dodatkowe działania w momencie wywołania konkretnych eventów
- narzuca strukturę poprzez interfejsy PluginInterface i Capable

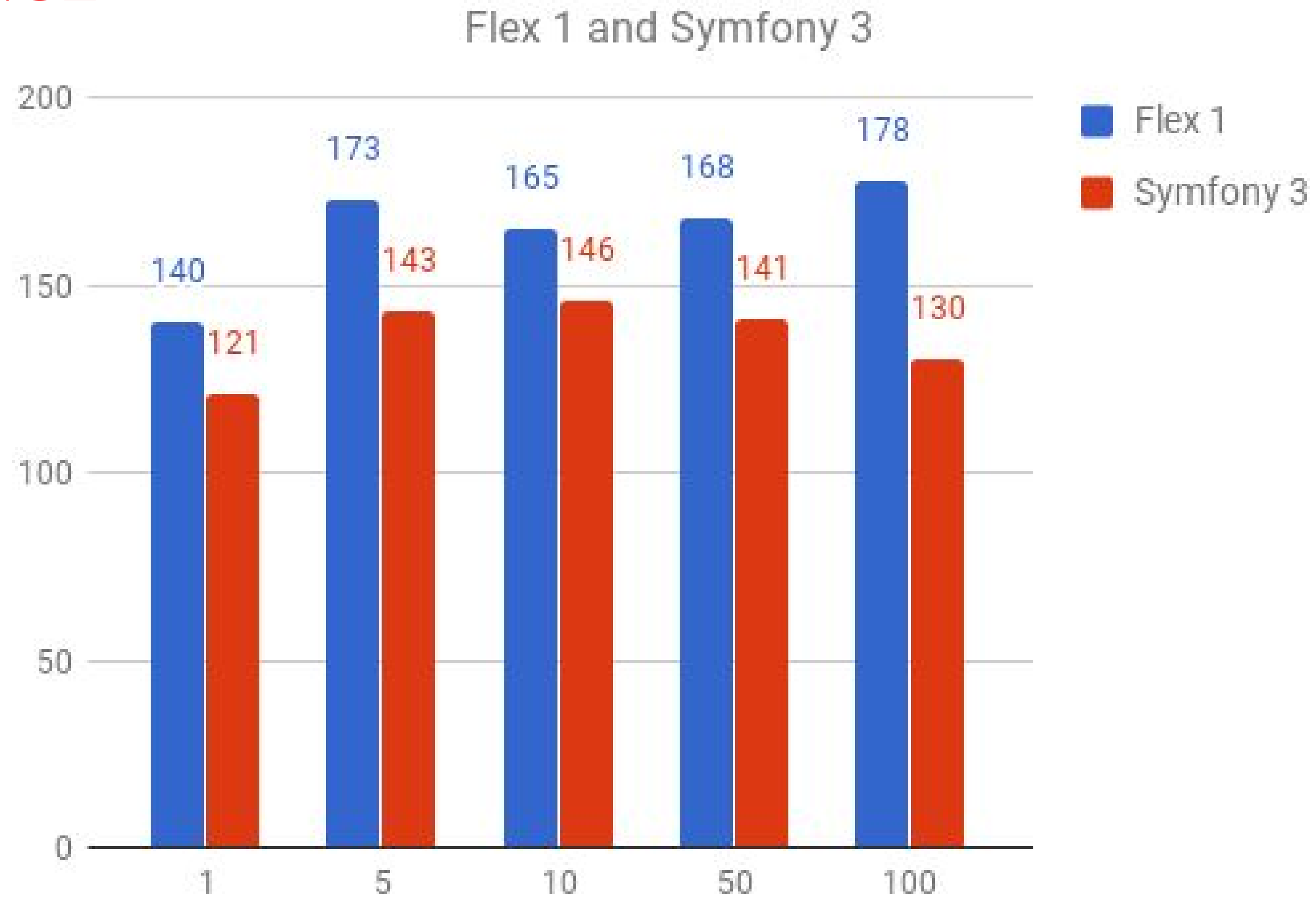
SYMFONY FLEX - PO CO?

- upraszcza instalację i konfigurację zależności
- zwiększając modularność pozwala na wykorzystanie tylko niezbędnych elementów
- większe zaufanie do trafności wyboru paczki

ROADMAPA



PERFORMANCE



<https://symfony.fi/entry/porting-a-symfony-3-application-to-flex>

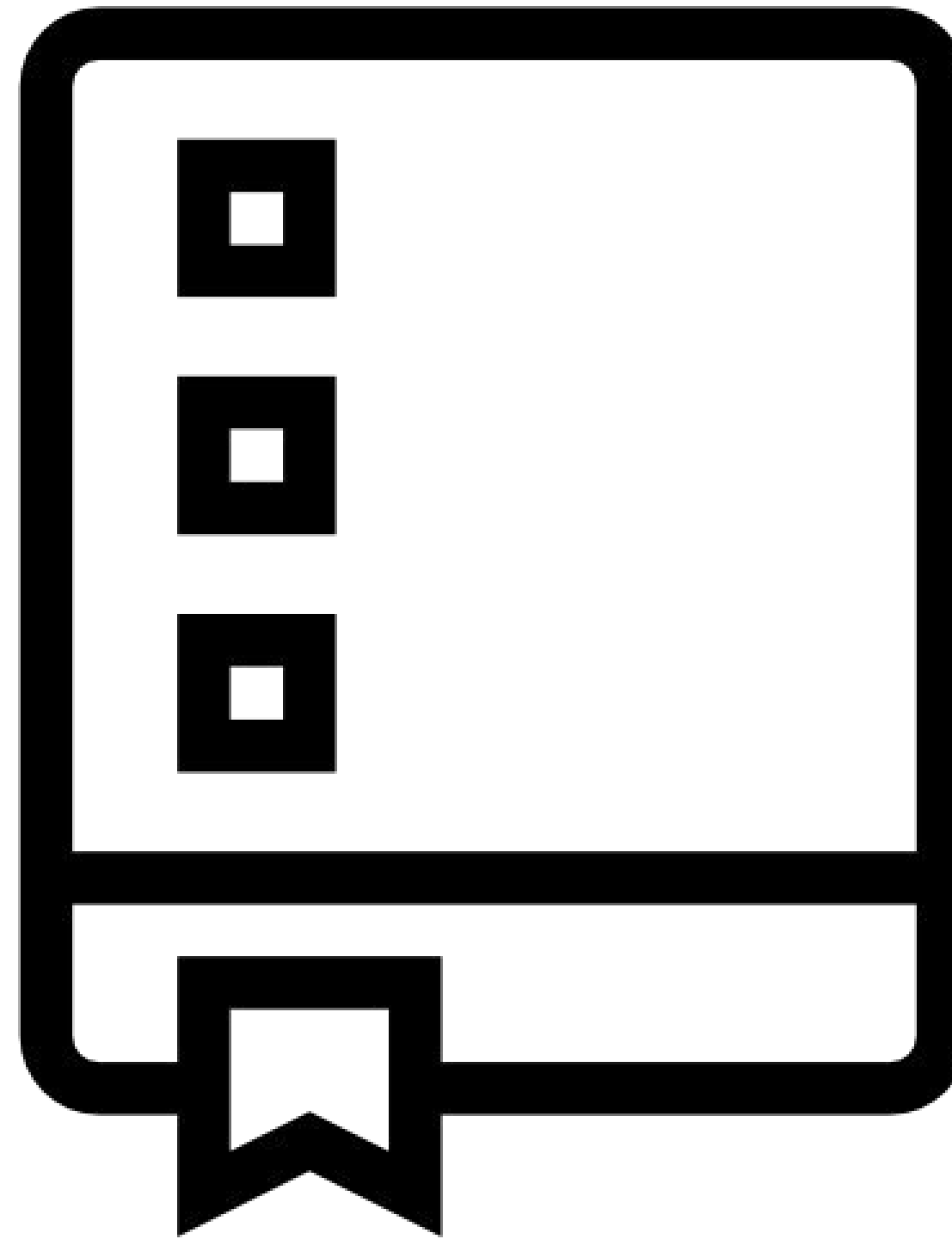


STRUKTURA PROJEKTU

```
1 your-project/  
2 |— Makefile  
3 |— config/  
4 |   |— bundles.php  
5 |   |— packages/  
6 |   |— routes.yaml  
7 |   └─ services.yaml  
8 |— public/  
9 |   └─ index.php  
10 |— src/  
11 |   |— ...  
12 |   └─ Kernel.php  
13 |— templates/  
14 └─ vendor/
```

REPOZYTORIA

CORE



CONTRIB

RECEPTURY

```
1  {
2    "bundles": {
3      "Symfony\\Bundle\\SwiftmailerBundle\\SwiftmailerBundle": ["all"]
4    },
5    "copy-from-recipe": {
6      "config/": "%CONFIG_DIR%/
7    },
8    "env": {
9      "MAILER_URL": "smtp://localhost:25?encryption=&auth_mode="
10   },
11   "aliases": ["mailer", "mail"]
12 }
```

REPOZYTORIA - ILOŚĆ PACZEK

CORE

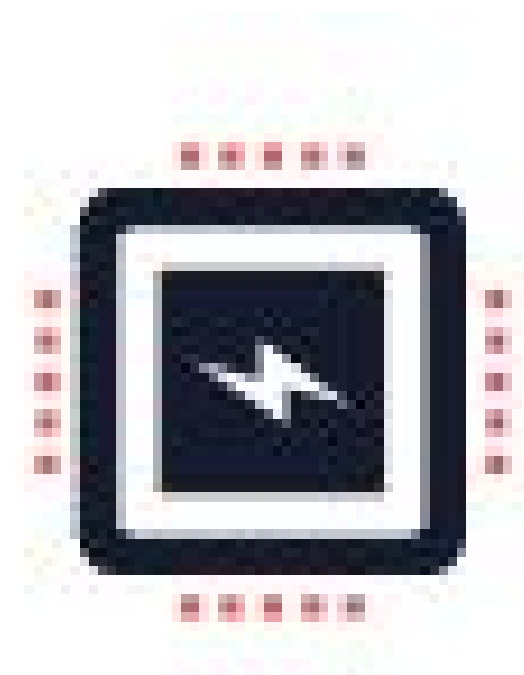
19 modułów Symfony

11 innych paczek

CONTRIB

33 paczki

CO SIĘ DZIEJE POD MASKĄ?



- nadpisanie komend composera (require, update, remove - żeby wstrzyknąć własny PackageResolver)
- wywoływanie dodatkowych metod na eventach

UPGRADE



1. Nowy projekt
2. Skopiowanie zależności do composer.json
3. Przeniesienie brakujących configów
4. Parameters.yml -> services.yaml + .env
5. Przeniesienie kodu i aktualizacja namespace'ów
6. Przeniesienie szablonów
7. Zmiany w index.php

Demo



Konkurs

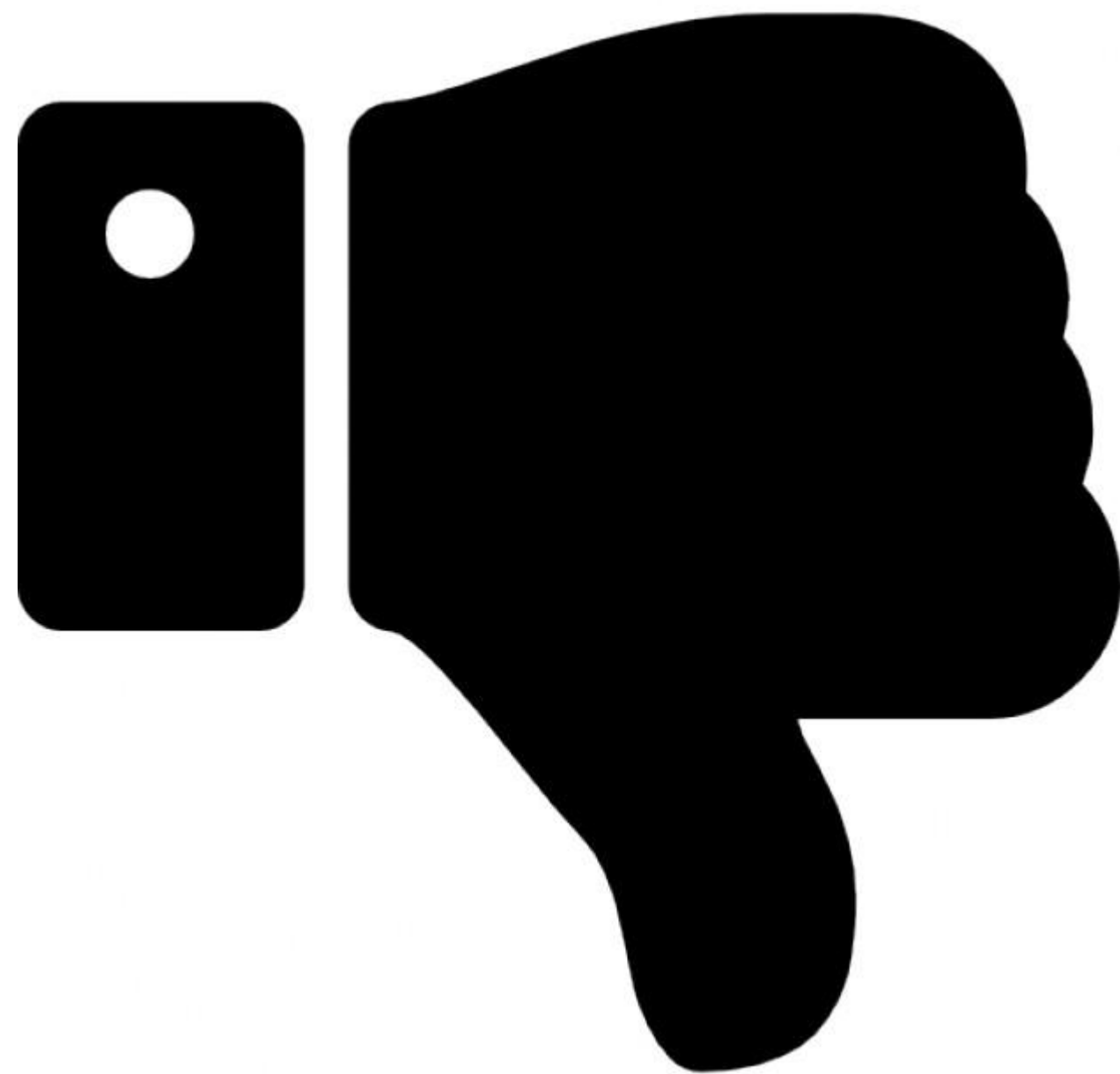


ZALETY



- łatwiejsza instalacja i deinstalacja paczek
- autokonfiguracja
- bardziej przejrzysta dokumentacja
- wybór preferowanych paczek przez ekspertów
- większa modułowość zależności
- znajdowanie paczek po aliasach
- wzrost wydajności

WADY



- nowa struktura projektu
- wymagana ręczna migracja starszych projektów
- na pierwszy rzut oka trochę auto magii
- niestabilna do czasu Symfony 4
- użyteczność zależna od wsparcia community

SYMFONY TRADEMARK



<http://fabien.potencier.org/the-symfony-trademark.html>

ŹRÓDŁA:

- <https://getcomposer.org/>
- https://image.freepik.com/free-icon/thumbs-down-silhouette_318-41911.jpg
- <https://maxcdn.icons8.com/iOS7/PNG/512/Programming/repository-512.png>
- <https://maxcdn.icons8.com/Share/icon/Business//bullish1600.png>
- <https://symfony.com/doc/current/contributing/community/releases.html#version-history>
- <https://symfony.com/doc/current/setup/flex.html>
- https://symfony.com/logos/symfony_black_02.png

DZIĘKUJĘ ZA UWAGĘ!



Poznań // Helsinki // San
Francisco

