

Python i JVM pod jednym dachem

Rekomendacje

Rekomendacje w Allegro

Rekomendacje w Allegro

- Jak nas widać - głównie karuzele ofertowe

Rekomendacje w Allegro

- Jak nas widać - głównie karuzele ofertowe
- Gdzie nas widać:
 - strona oferty

Rekomendacje w Allegro

- Jak nas widać - głównie karuzele ofertowe
- Gdzie nas widać:
 - strona oferty
 - strona główna

Rekomendacje w Allegro

- Jak nas widać - głównie karuzele ofertowe
- Gdzie nas widać:
 - strona oferty
 - strona główna
 - koszyk
 - ...

Rekomendacje w Allegro

- Jak nas widać - głównie karuzele ofertowe
- Gdzie nas widać:
 - strona oferty
 - strona główna
 - koszyk
 - ...
- Nie oferty sponsorowane, nie oferty promowane



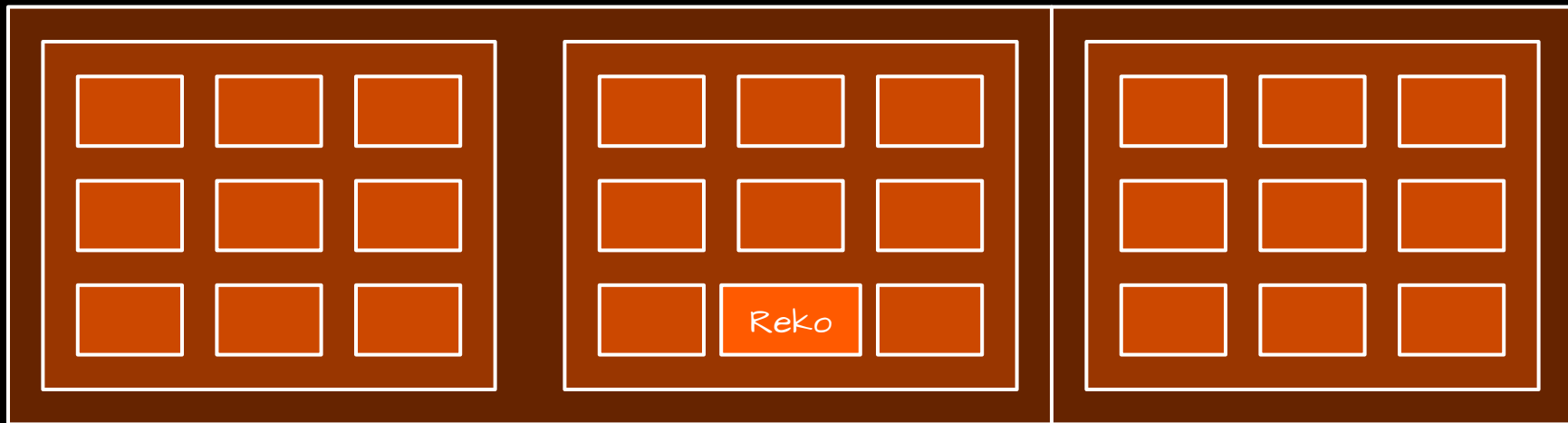
kolodziej.j.info
@unit03



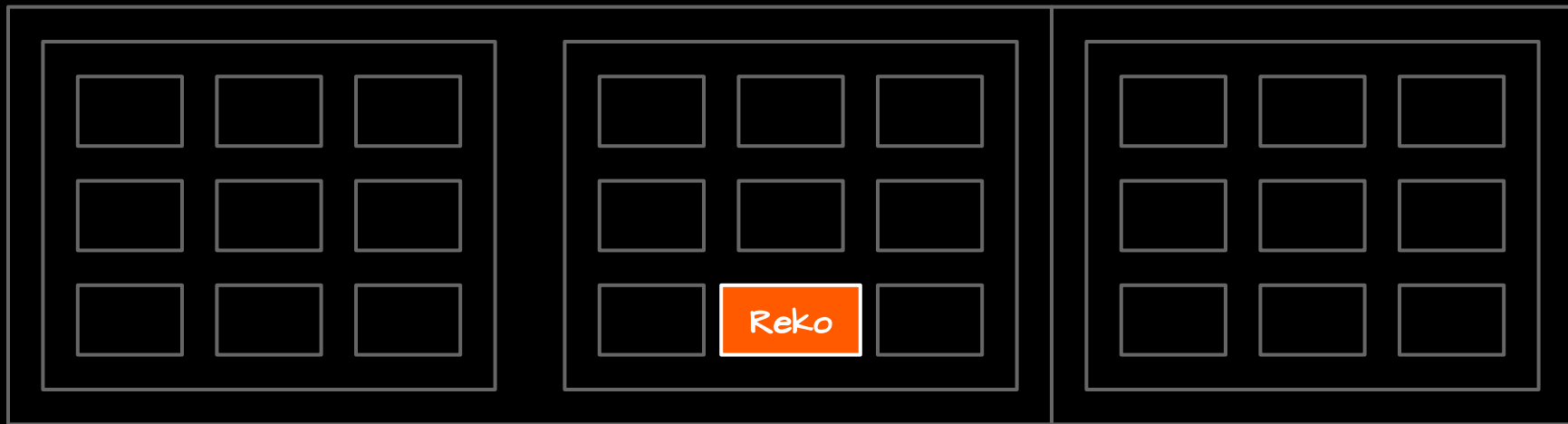
Perspektywa



Perspektywa



Perspektywa



Statystyki*

* dotyczy samych Rekomendacji, nie całego Allegro!

Statystyki*

Aktualnie:

- % Kodu:
 - Java:
 - Scala:
 - Python:

Statystyki*

Aktualnie:

- % Kodu:
 - Java: 17%
 - Scala: 36%
 - Python: 27%

Statystyki*

Aktualnie:

- % Kodu:
 - Java: 17%
 - Scala: 36%
 - Python: 27%
- % repozytoriów**:
 - Java: 43%
 - Scala: 14%
 - Python: 43%

** główny język; pomijając repo konfiguracyjne, testowe etc.

Tech*

Tech*

Webserwisy:

Przetwarzanie danych:

Tech*

Webserwisy:

- Spring, Akka HTTP
- Flask
- Cassandra
- Hermes (Kafka)
- Graphite/Prometheus, Kibana

Przetwarzanie danych:

Tech*

Webserwisy:

- Spring, Akka HTTP
- Flask
- Cassandra
- Hermes (Kafka)
- Graphite/Prometheus, Kibana

Przetwarzanie danych:

- Spark
- BigQuery
- AI Platform (też od Google)
- Airflow

Python i JVM pod jednym dachem

Python i JVM

- Py4J, Jython

Python i JVM

- Py4J, Jython ✨

Python i JVM

● ~~Py4J, Jython~~ ✨

Python i JVM

- ~~Py4J, Jython~~ 
- Priorytety:
 - stabilność

Python i JVM

- ~~Py4J, Jython~~ 
- Priorytety:
 - stabilność
 - prostota

Python i JVM

- ~~Py4J, Jython~~ 
- Priorytety:
 - stabilność
 - prostota
 - niski koszt utrzymania

Python i JVM

- ~~Py4J, Jython~~ 
- Priorytety:
 - stabilność
 - prostota
 - niski koszt utrzymania
- Komponenty (np. webserwisy)

Python i JVM

- ~~Py4J, Jython~~ 
- Priorytety:
 - stabilność
 - prostota
 - niski koszt utrzymania
- Komponenty (np. webserwisy)
 - względnie niezależne
 - łatwe w obsłudze
 - wąska, jasno określona odpowiedzialność

Punkty styku

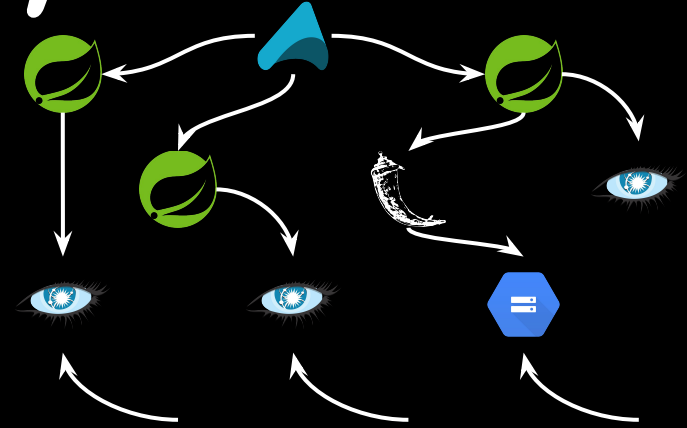
Punkty styku

- Między webserwisami - HTTP



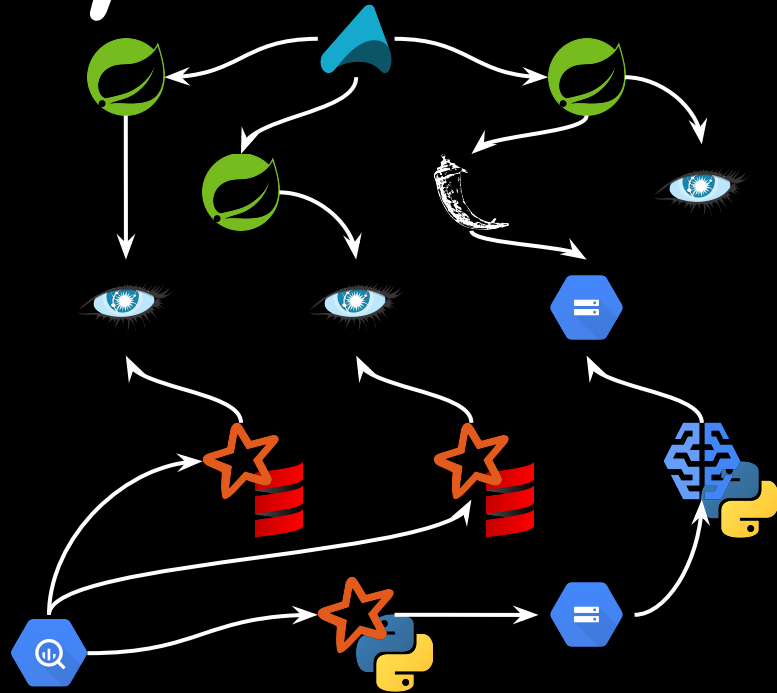
Punkty styku

- Między webserwisami - HTTP
- Dane dla webserwisów - storage/bazy danych



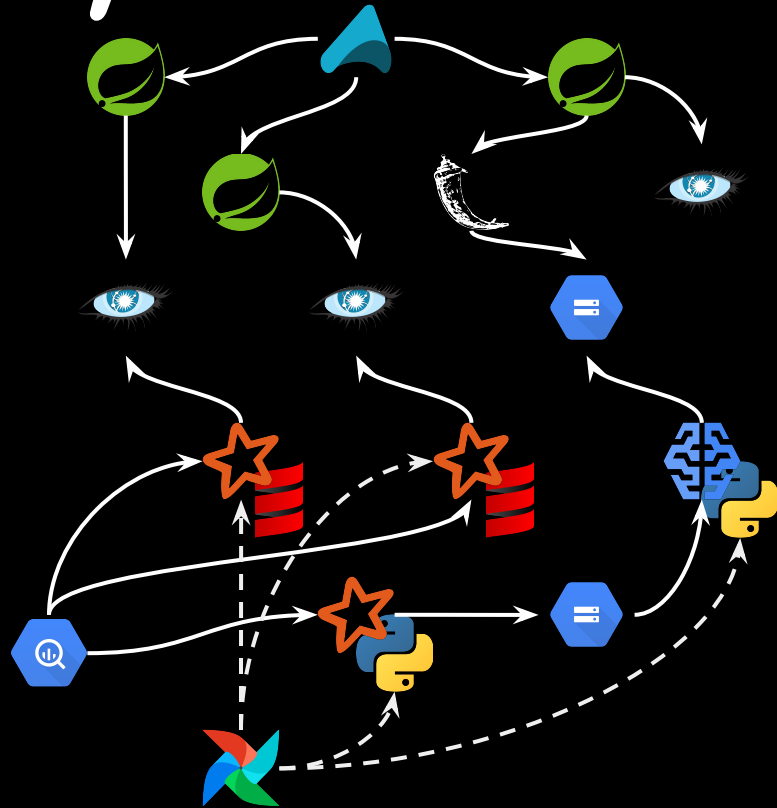
Punkty styku

- Między webserwisami - HTTP
- Dane dla webserwisów - storage/bazy danych
- W ramach przetwarzania danych - j.w.



Punkty styku

- Między webserwisami - HTTP
- Dane dla webserwisów - storage/bazy danych
- W ramach przetwarzania danych - j.w.
- Orkiestracja przetwarzania - DAGi Airflow



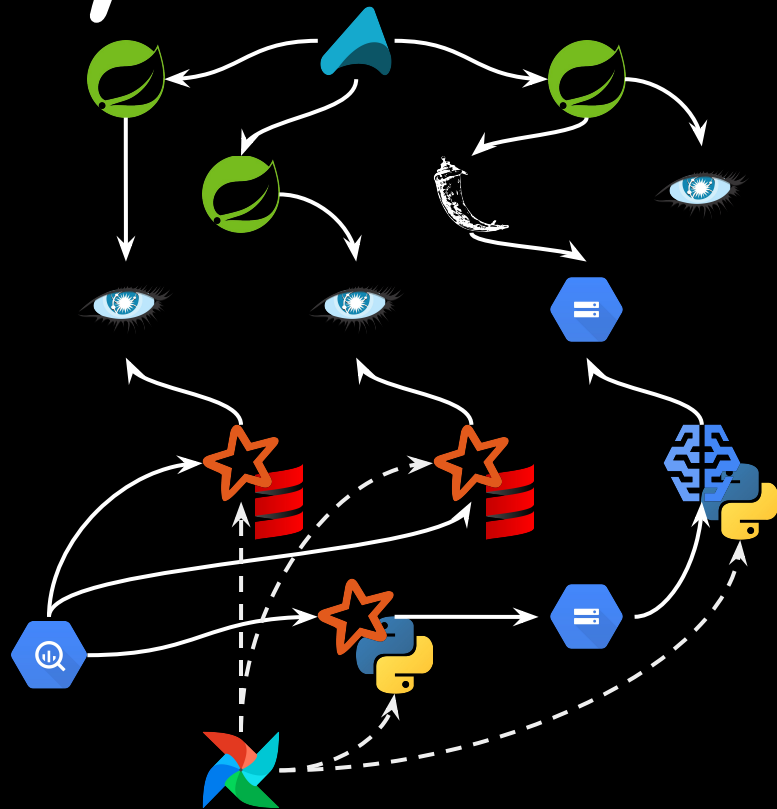
Dekompozycja

Dekompozycja

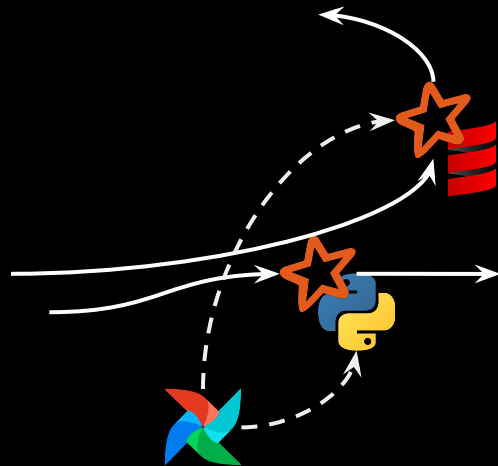
Wąskie, jasno zdefiniowane:

- odpowiedzialność
- interfejsy

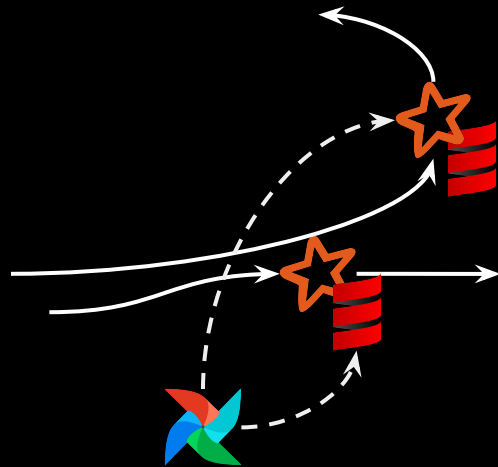
Punkty styku



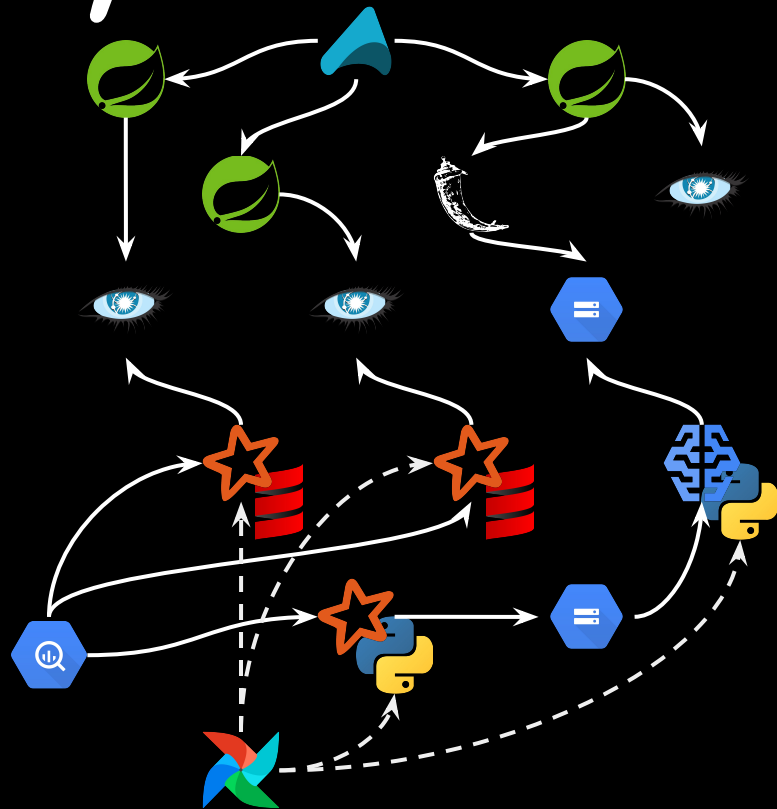
Punkty styku



Punkty styku



Punkty styku



Punkty styku

- Między webserwisami - HTTP
- Dane dla webserwisów - storage/bazy danych
- W ramach przetwarzania danych - j.w.
- Orkiestracja przetwarzań - DAGi Airflow

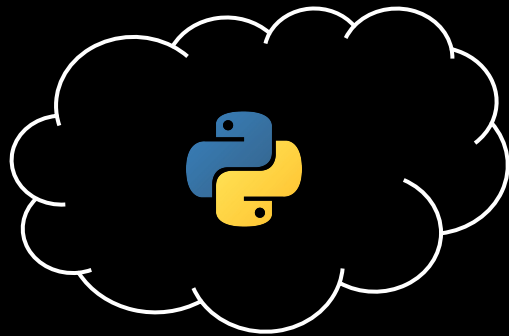
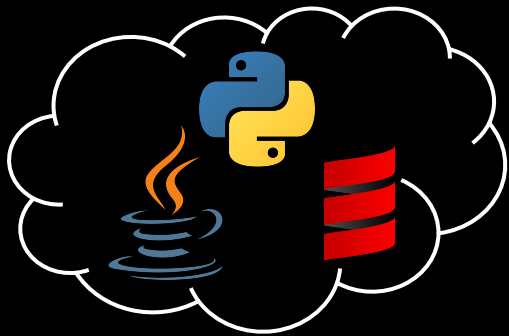
-> Architektura serwisowa

Punkty styku

- Między ludźmi

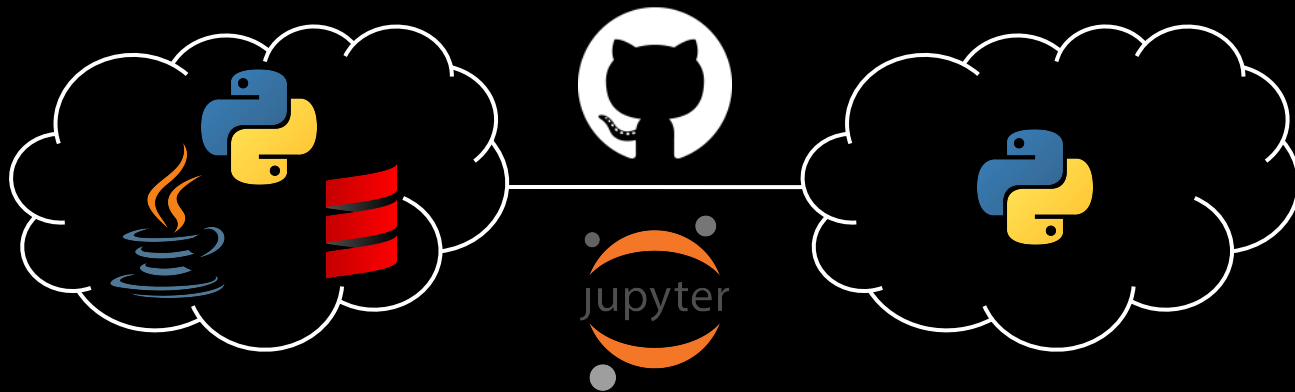
Punkty styku

- Między ludźmi



Punkty styku

- Między ludźmi:
 - Git/GitHub
 - Jupyter Notebooks



Piszemy nowy komponent

Piszemy nowy komponent

- Stan zastany
 - platforma, biblioteki, narzędzia

Piszemy nowy komponent

- Stan zastany
 - platforma, biblioteki, narzędzia
 - istniejący kod, szablony

Piszemy nowy komponent

- Stan zastany
 - platforma, biblioteki, narzędzia
 - istniejący kod, szablony
 - kompetencje

Piszemy nowy komponent

- Stan zastany
 - platforma, biblioteki, narzędzia
 - istniejący kod, szablony
 - kompetencje
- Wymagania funkcjonalne

Piszemy nowy komponent

- Stan zastany
 - platforma, biblioteki, narzędzia
 - istniejący kod, szablony
 - kompetencje
- Wymagania funkcjonalne
- Wymagania нефunkcjonalne

Piszemy nowy komponent

- Stan zastany
 - platforma, biblioteki, narzędzia
 - istniejący kod, szablony
 - kompetencje
- Wymagania funkcjonalne
- Wymagania нефunkcjonalne
 - heurystyki wydajnościowe

Heurystyki wydajnościowe

← Webserwis, głównie I/O?



Heurystyki wydajnościowe

← Webserwis, głównie I/O?



Przetwarzanie danych,
"SQL-like" operacje?



Heurystyki wydajnościowe

← Webserwis, głównie I/O?



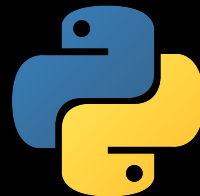
...



...



...



Przetwarzanie danych,
"SQL-like" operacje?



Piszemy nowy komponent

- Stan zastany
 - platforma, biblioteki, narzędzia
 - istniejący kod, szablony
 - kompetencje
- Wymagania funkcjonalne
- Wymagania нефunkcjonalne
 - heurystyki wydajnościowe

Piszemy nowy komponent

- Stan zastany
 - platforma, biblioteki, narzędzia
 - istniejący kod, szablony
 - kompetencje
- Wymagania funkcjonalne
- Wymagania нефunkcjonalne
 - heurystyki wydajnościowe
 - różnice między językami

Piszemy nowy komponent

- Stan zastany
 - platforma, biblioteki, narzędzia
 - istniejący kod, szablony
 - kompetencje
- Wymagania funkcjonalne
- Wymagania нефunkcjonalne
 - heurystyki wydajnościowe
 - różnice między językami
- Stabilność i utrzymanie
vs. usprawnienia, innowacje i rozwój umiejętności

Tzw. engineering

- Oczywistości?

Tzw. engineering

- Oczywistości?
- Emocje i uprzedzenia a metodyczność

Tzw. engineering

- Oczywistości?
- Emocje i uprzedzenia a metodyczność

"Ooh, shiny!"

Tzw. engineering

- Oczywistości?
- Emocje i uprzedzenia a metodyczność

"Ooh, shiny!" → "Rozważmy
użycie"

Tzw. engineering

- Oczywistości?
- Emocje i uprzedzenia a metodyczność
- Narzędzia i procesy

Tzw. engineering

- Oczywistości?
- Emocje i uprzedzenia a metodyczność
- Narzędzia i procesy
 - code review

Tzw. engineering

- Oczywistości?
- Emocje i uprzedzenia a metodyczność
- Narzędzia i procesy
 - code review?
 - retrospekcje

Tzw. engineering

- Oczywistości?
- Emocje i uprzedzenia a metodyczność
- Narzędzia i procesy
 - code review?
 - retrospekcje?
 - design docs

Tzw. engineering

- Oczywistości?
- Emocje i uprzedzenia a metodyczność
- Narzędzia i procesy
 - code review?
 - retrospekcje
 - design docs:
 - potrzeba zmiany

Tzw. engineering

- Oczywistości?
- Emocje i uprzedzenia a metodyczność
- Narzędzia i procesy
 - code review?
 - retrospekcje
 - design docs:
 - potrzeba zmiany
 - dostępne alternatywy

Tzw. engineering

- Oczywistości?
- Emocje i uprzedzenia a metodyczność
- Narzędzia i procesy
 - code review?
 - retrospekcje
 - design docs:
 - potrzeba zmiany
 - dostępne alternatywy
 - za i przeciw

Tzw. engineering

- Oczywistości?
 - Emocje i uprzedzenia a metodyczność
 - Narzędzia i procesy
 - code review?
 - retrospekcje
 - design docs:
 - potrzeba zmiany
 - dostępne alternatywy
 - za i przeciw
-
- różnorodne opinie
 - współpraca

Dzięki! :)
Pytania?

kolodziejj.info/talks/python-jvm