

Wprowadzenie do modelu aktorów na JVM

Mikołaj Fejzer

Agenda

- Model aktorów
 - historia, założenia
 - związek z programowaniem funkcyjnym
 - wzorce projektowe
- Akka
 - nadzór, sposoby adresacji, skalowalność
 - wdrażanie, integracja
- Źródła, referencje

Historia

Próba odpowiedzi na:

- zbyt niski poziom abstrakcji
- brak kontraktów
- brak naturalnego modelu komunikacji
- pojawienie się koncepcji wielu rdzeni/procesorów

Założenia

- Aktor
 - skrzynka odbiorcza
 - zachowanie (funkcja)
- Odbiór wiadomości:
 - wysłanie skończonej ilości odpowiedzi
 - stworzenie skończonej ilości aktorów
 - zmiana zachowania

Programowanie funkcyjne?

- Future
 - flatMap
 - sequence
 - traverse
 - reduce
- Kompozytowość konfiguracji aktorów
- Modelowanie zachowania
- Niezmiennność wiadomości

Wzorce projektowe

- Strategia/Stan
- Kontekst
- Egzekutor
- Łańcuch zobowiązań (Chain of responsibility)
- Aktywny obiekt (Active object)

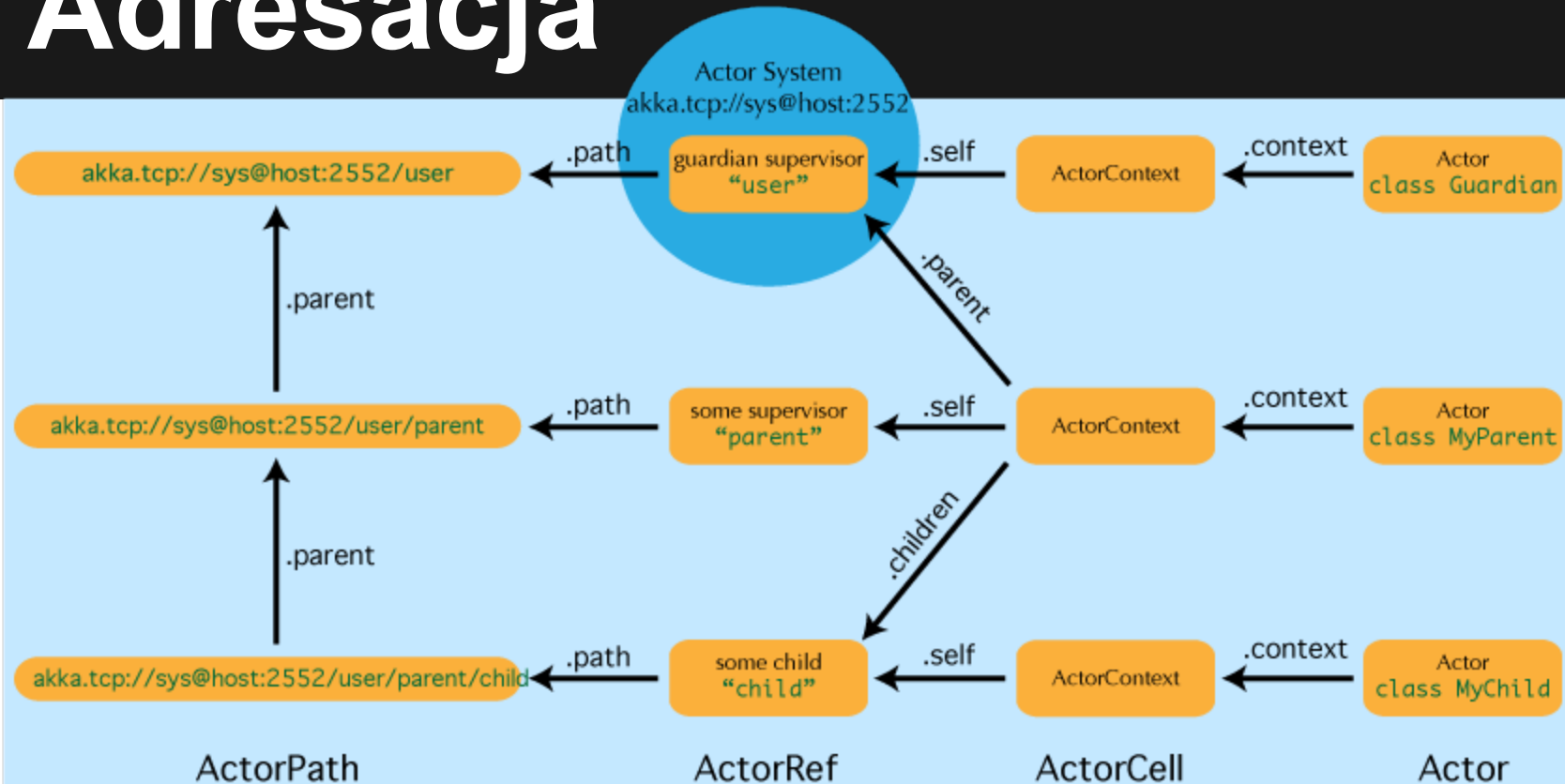
Model aktorów na JVM

- Dlaczego Akka?
- Inne biblioteki:
 - Kilim
 - JActor
 - ActorFoundry
 - Korus
 - GPar

Akka

- Aktorzy - bez typu oraz z
- Cykl życia
- Hierarchia
- Komunikacja zdalna
- Testowalność
- Przechowywanie stanu

Adresacja



Źródła, referencje

- <http://akka.io/>
- http://en.wikipedia.org/wiki/Actor_model
- <https://typesafe.com/activator>
- <http://java.dzone.com/articles/whats-wrong-java-8-part-iv>

Dziękuję za uwagę

Q&A