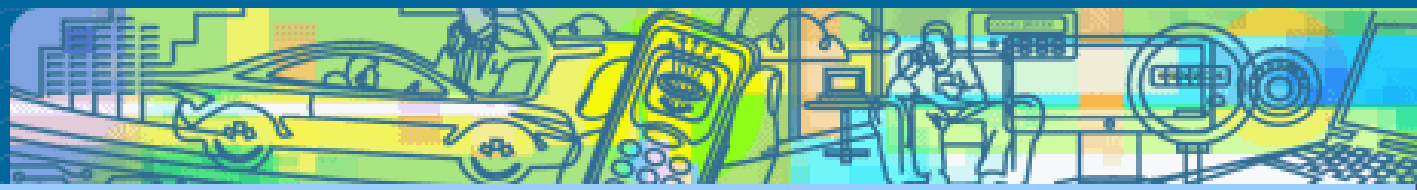
OSGi

Agata Hejmej

4.05.2009



Plan prezentacji

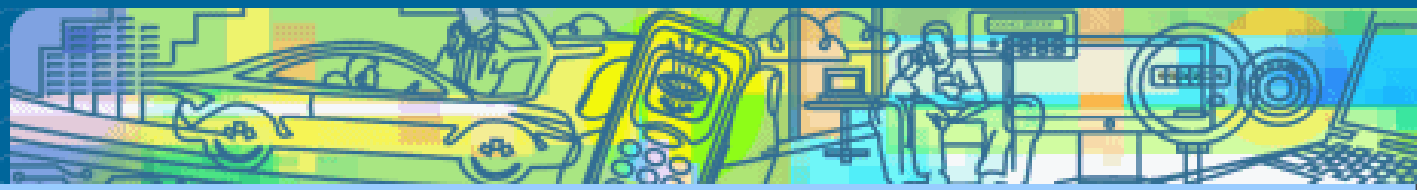
Co to jest OSGi

Jakie problemy rozwiązuje

Opis standardu

Przykładowa aplikacja

Podsumowanie korzyści



Co to jest OSGi?

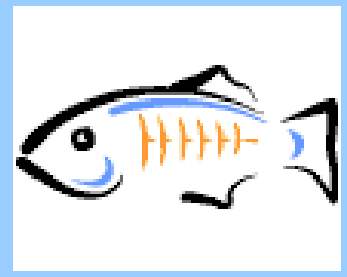
Standard, który pozwala na tworzenie wysoce modularnych aplikacji w Javie

Koncepcja podobna do SOA, tylko ograniczona do jednej JVM

Do OSGi Alliance (organizacji odpowiedzialnej za rozwijanie standardu) należą m. in.:

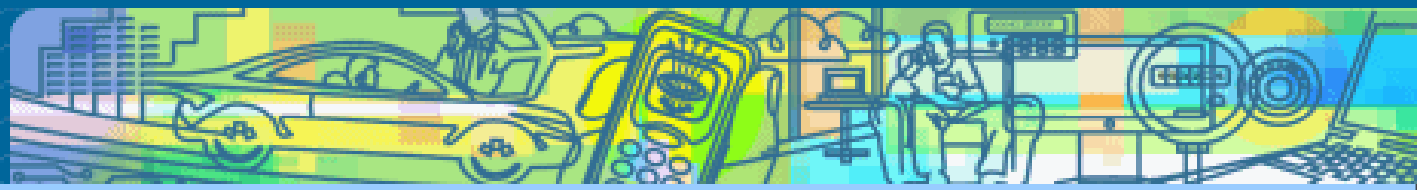
IBM, Oracle, Red Hat, Spring Source,
Ericsson, Deutsche Telekom, Motorola, Nokia, Samsung, Siemens

Aplikacje, które powstały przy użyciu OSGi to m.in.:



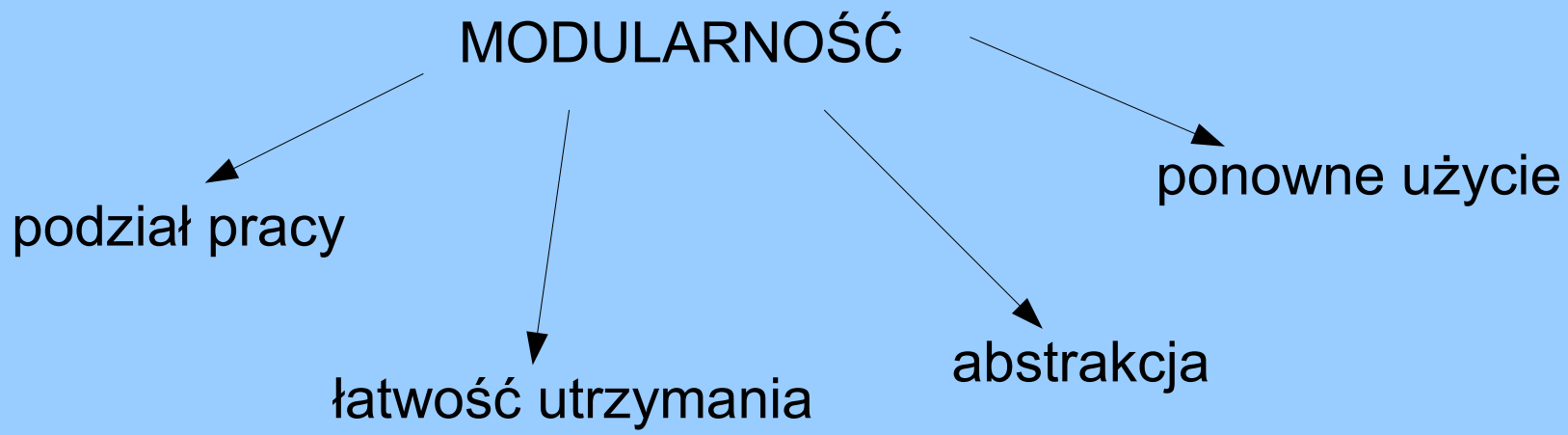
GlassFish

Weblogic



Problem

Najważniejsze wyzwanie przy budowie dużych systemów:
ich **złożoność**



Java to oczywiście umożliwia, ale...
jeśli nie wymusimy modularności, tak naprawdę nie będziemy jej mieć

Jaki powinien być moduł?

stanowiący logiczną całość

wewnętrznie spójny

luźno powiązany z innymi modułami

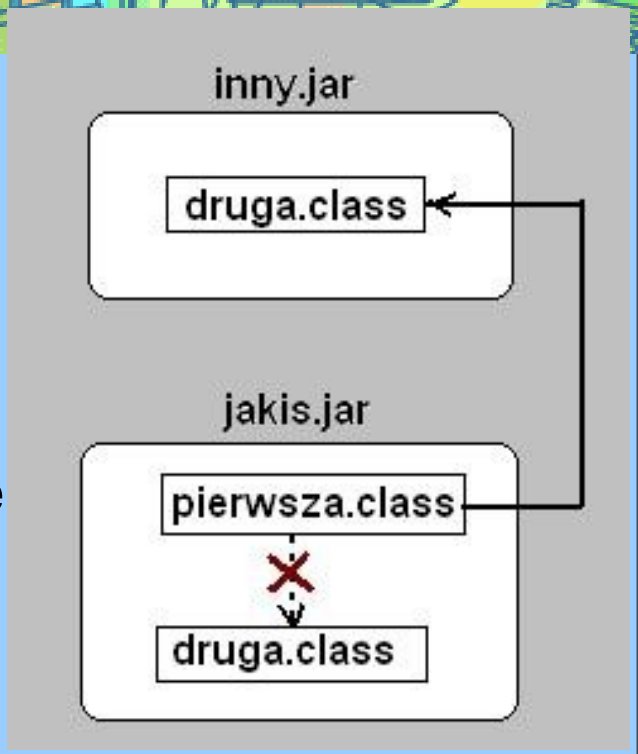
z dobrze zdefiniowanym interfejsem – co dajemy i czego oczekujemy

Problem: „JAR hell”

JARy dostarczają zazwyczaj albo pojedynczą bibliotekę albo część funkcjonalności aplikacji

Tylko małe aplikacje w jednym JARze, duże - rzędu kilkudziesięciu, kilkuset JARów, które trzeba ze sobą złożyć

Problemy z JARami

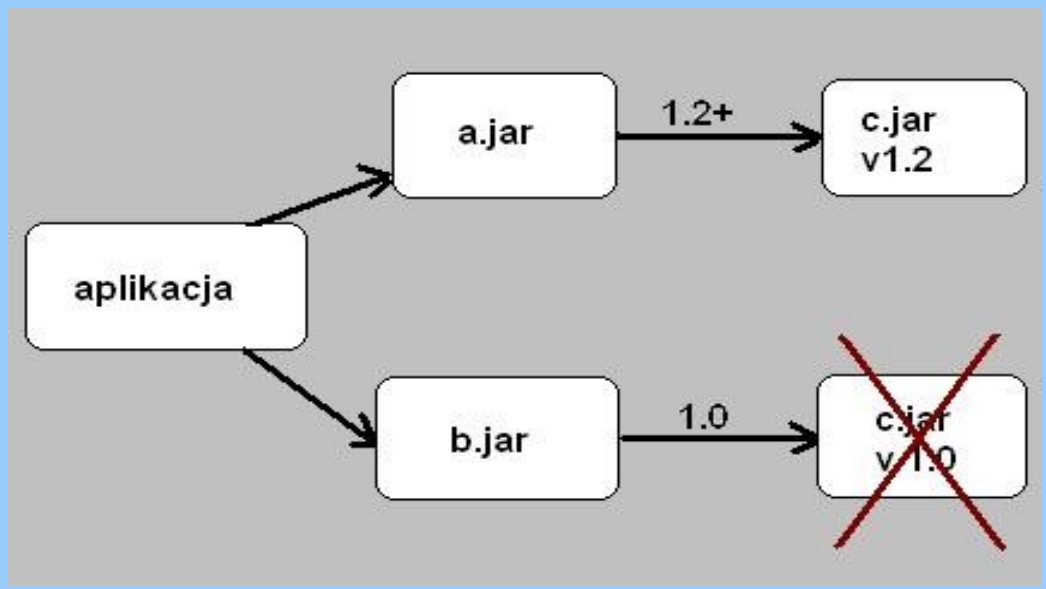


liniowy classpath

wewnętrzne zależności JARów nie są podane explicite (ClassNotFoundException podczas wykonania...)

brak ukrywania informacji pomiędzy JARami – tylko między pakietami (package)

brak informacji o wersjach



JARy nie spełniają oczekiwań co do modułów !

Możliwe rozwiązanie: OSGi

dynamiczny system modułów dla Javy

definiuje sposób tworzenia prawdziwych modułów

definiuje sposób interakcji pomiędzy modułami w czasie wykonania

moduł OSGi = bundle

zamiast jednej globalnej classpath dla aplikacji, każdy moduł posiada swoją

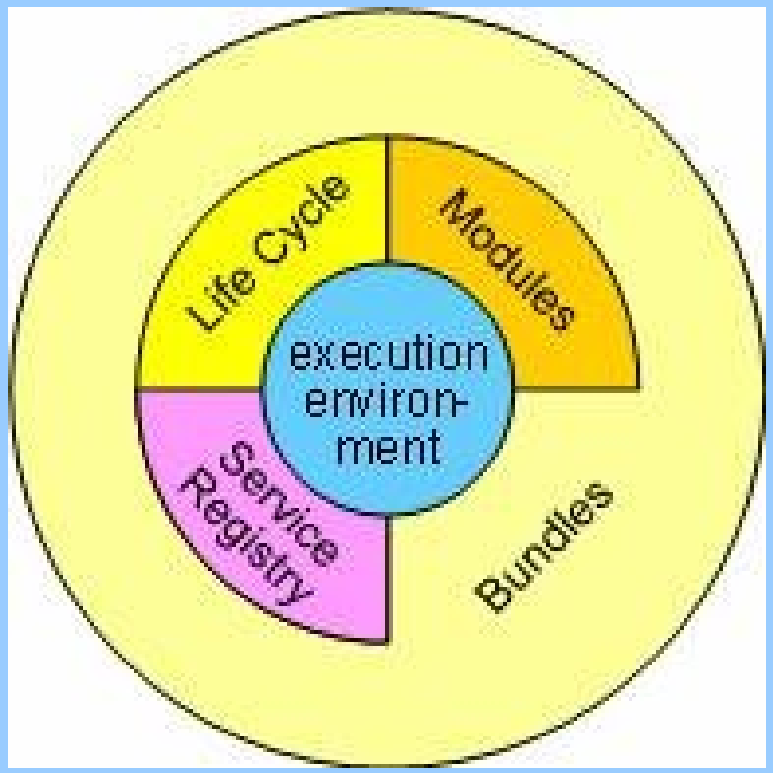
dzielenie klas poprzez importy i eksporty explicite

dynamiczne podłączanie, wymienianie, wyłączanie modułów - „serwisy”

tak naprawdę OSGi bundles to zwykłe JARy (!), z dodatkowymi wpisami w manifeście

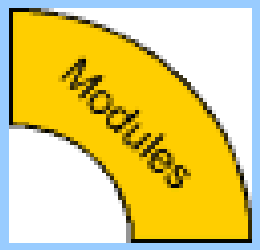
całą organizacją zarządza dodana warstwa middleware –
OSGi framework

OSGi framework

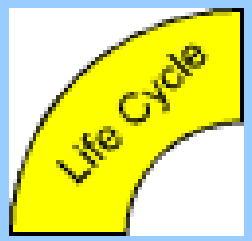


http://www.osgi.org/About/Technology

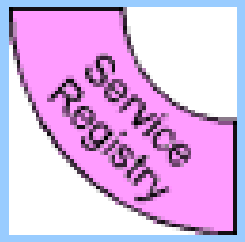
środowisko wykonania – Java (J2SE, J2EE, J2ME, konfiguracje i profile jak MIDP)



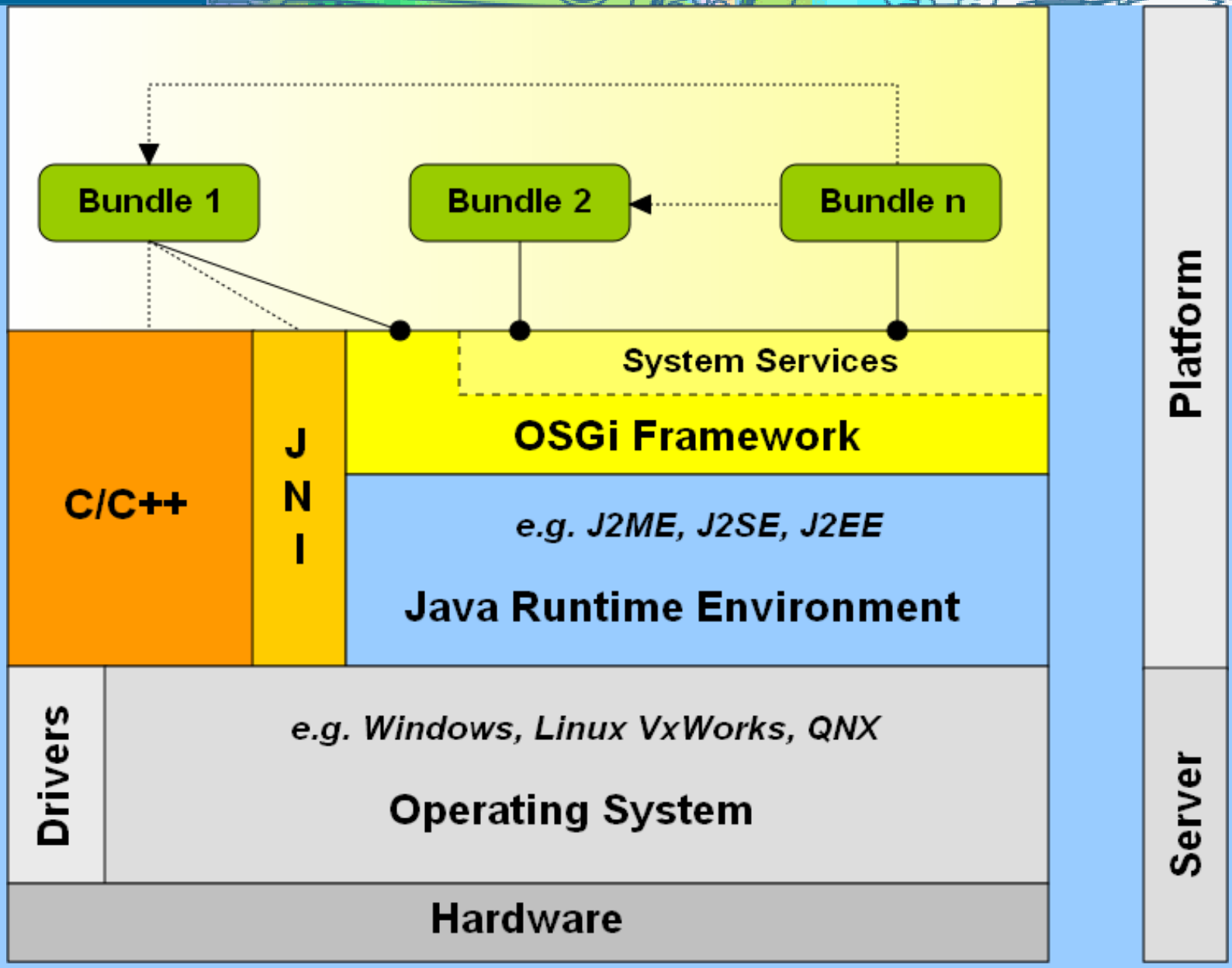
warstwa zarządzająca ładowaniem klas, zgodnie z opisanym podziałem na moduły



warstwa zarządzająca cyklem życia modułów w trakcie wykonania – dynamiczne instalowanie, odinstalowywanie, uruchamianie, wstrzymywanie



moduły mogą kooperować poprzez tradycyjne dzielenie klas, ale to przeszkadza w ich dynamicznym cyklu życia; rozwiązanie: serwisy



http://en.wikipedia.org/wiki/File:Osgi.svg

Dynamiczne serwisy

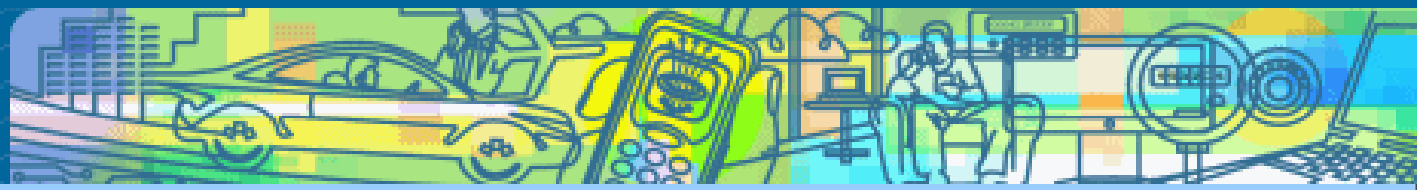
Serwisy to zwykłe obiekty Javowe

Serwisy rejestrują się w OSGi Service Registry pod jednym lub kilkoma interfejsami

Klient, który ich potrzebuje, szuka ich w OSGi Service Registry pod nazwą interfejsu

Co stanie się, gdy uruchomimy bundle z serwisem A, który korzysta z serwisu B, zanim uruchomimy bundle z serwisem B ?

Nic złego.. A po prostu poczeka, aż B stanie się dostępny



Dynamiczne serwisy cd.

Możemy podmieniać pojedyncze komponenty bez potrzeby restartu całego systemu

Oczywiście łatwiej jest używać czegoś, co jest cały czas dostępne, stąd dodatkowy koszt, ale OSGi Service Registry wydatnie ułatwia radzenie sobie z tym problemem

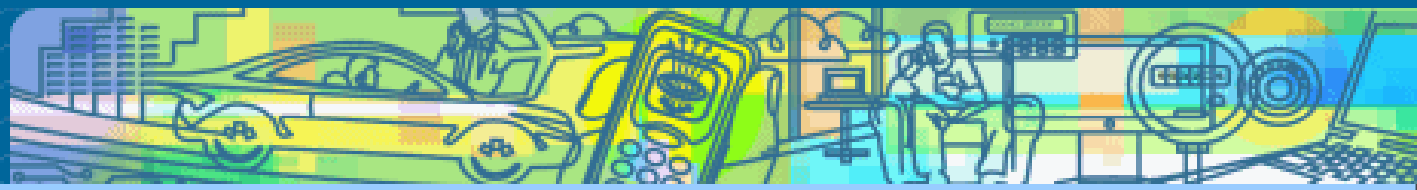
OSGi a SOA

Serwisy w OSGi to w zasadzie implementacja koncepcji SOA

SOA – Service Oriented Architecture (architektura zorientowana na usługi)

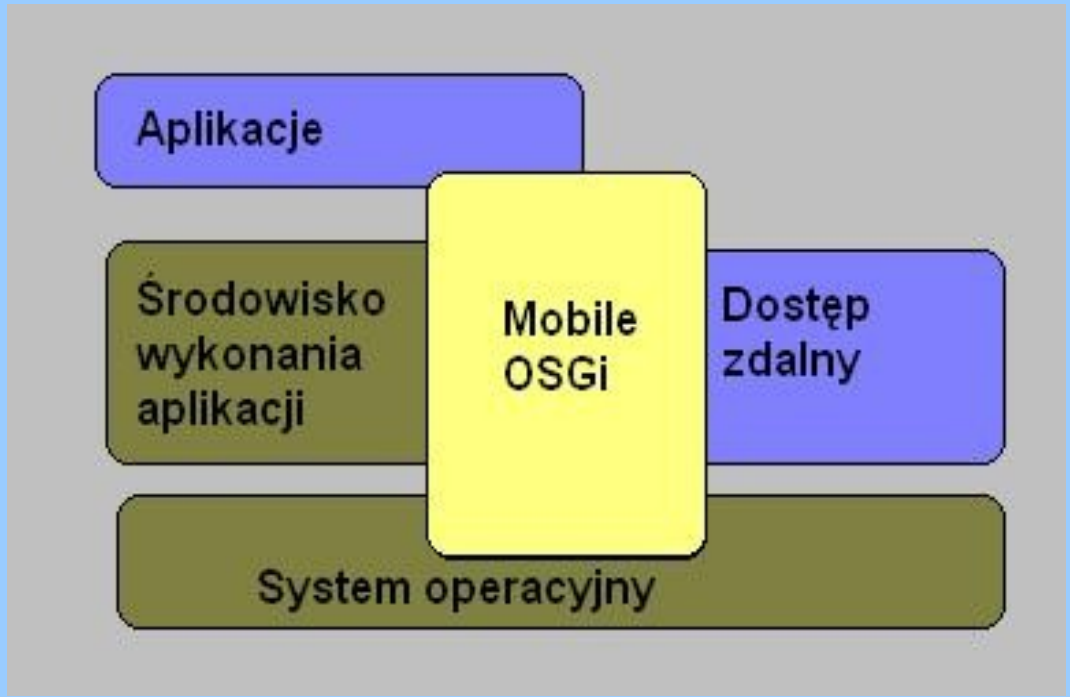
- Aplikacje działają korzystając z serwisów (usług), które mogą działać na innej maszynie
- Usługi komunikują się ze sobą za pomocą magistrali komunikacyjnej i specyficznych komunikatów
- Pozwala to np. na łatwe łączenie aplikacji z części napisanych w różnych językach

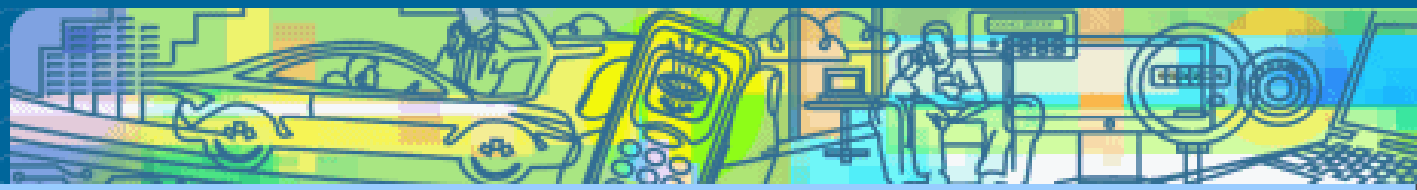
W OSGi serwisy komunikują się w ramach jednej JVM dlatego nie potrzeba magistrali, funkcję tego typu spełnia OSGi Service Registry



Mobile OSGi

Istnieją implementacje OSGi na Win Mobile, Nokia S60, Android





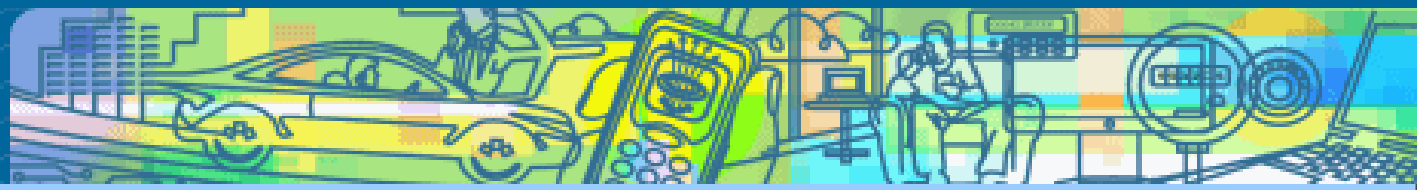
Implementacje standardu OSGi

Istnieje kilka niezależnych implementacji frameworków (zrębów) OSGi, w tym 3 popularne open-source'owe:

- Eclipse Equinox
- Apache Felix
- Knopflerfish

Poza tym istnieje wiele narzędzi wspomagających

Przykładowa aplikacja – część praktyczna

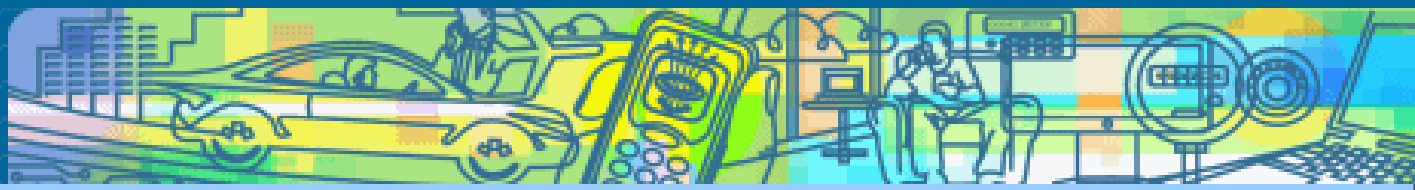


Korzyści – podsumowanie

Pozwala tworzyć oprogramowanie złożone z dynamicznych komponentów; jasno zdefiniowany cykl życia komponentów pozwala na odpowiednią obsługę zdarzeń

Dostarcza bezpieczne środowisko, w którym komponenty mają dostęp tylko do tych zasobów, do których powinny mieć

Niejako wymusza modularność, co poprawia jakość kodu



Literatura

http://www.osgi.org - strona OSGi Alliance

http://neilbartlett.name/blog/osgibook/ - darmowa książka o OSGi

http://java.dzone.com/articles/dozen-osgi-myths-and - ciekawy artykuł o OSGi

Dziękuję za uwagę