



Architektura i działanie systemów OSGi



Przykład OSGi runtime - Eclipse



- Demo
 - Eclipse (podprojekt Equinox)
 - eclipse -console
 - help, ss, bundle 123



Nowy OSGi bundle (paczka) w Eclipse



- Wymagania
 - Eclipse IDE w wersji JEE (co najmniej Plugin Development Environment)
- Nowy projekt typu plugin
 - target platform: an OSGi framework
 - alternatywa np Eclipse 3.3
- Hello Bundle
- Wyeksportowanie bundle
 - deployable plug-ins



Uruchomienie nowego bundle

- Konsola OSGi
 - install, start, stop
 - np. install
file:///c:/workspaces/osgi/tutorial/tutorialbundle.jar
- Albo
 - wgranie jara do katalogu ../Eclipse/plugins



Standard OSGi

- "The OSGi™ specifications define a standardized, component oriented, computing environment for networked services. Adding an OSGi Service Platform to a networked device (embedded as well as servers), adds the capability to manage the life cycle of the software components in the device from anywhere in the network. Software components can be installed, updated, or removed on the fly without having to disrupt the operation of the device. Software components are libraries or applications that can dynamically discover and use other components. Software components can be bought off the shelf or are developed in house. The OSGi Alliance has developed many standard component interfaces that are available from common functions like HTTP servers, configuration, logging, security, user administration, XML, and many more. Plug compatible implementations of these components can be obtained from different vendors with different optimizations."



OSGi



- Implementacje
 - np. Apache Felix, Knoplerfish, Eclipse (podprojekt Equinox)
- Zakres platformy
 - Środowisko wykonania dla komponentów i aplikacji
 - przeznaczone dla producentów, dostawców usług oraz twórców oprogramowania
 - Umożliwia zarządzanie komponentami i ich konfiguracją w czasie działania środowiska
 - startowanie, stopowanie, instalowanie, uaktualnianie



Zakres OSGi cd

- Orginalnie zaprojektowane dla urządzeń połączonych w sieć
 - obecnie używane w różnych zastosowaniach (od systemów wbudowanych po IDE)
- Jest wąską warstwą, która umożliwia komponentom Javowym na współpracę w ramach jednej JVM
- Udostępnia model współpracy, w którym aplikacje mogą dynamicznie udostępniać i wykrywać serwisy
 - serwis w OSGi - obiekt, który może być wykorzystywany przez inne bundle



Zakres OSGi cd

- Udostępnia możliwość zdalnego zarządzania komponentami
 - nie zdefiniowano jednego standardu zarządzania (wiele różnych protokołów SNMP, CMISE, CIM, OMA DM)
- Przewiduje wiele standardowych serwisów takich jak logowanie, HTTP, device discovery...

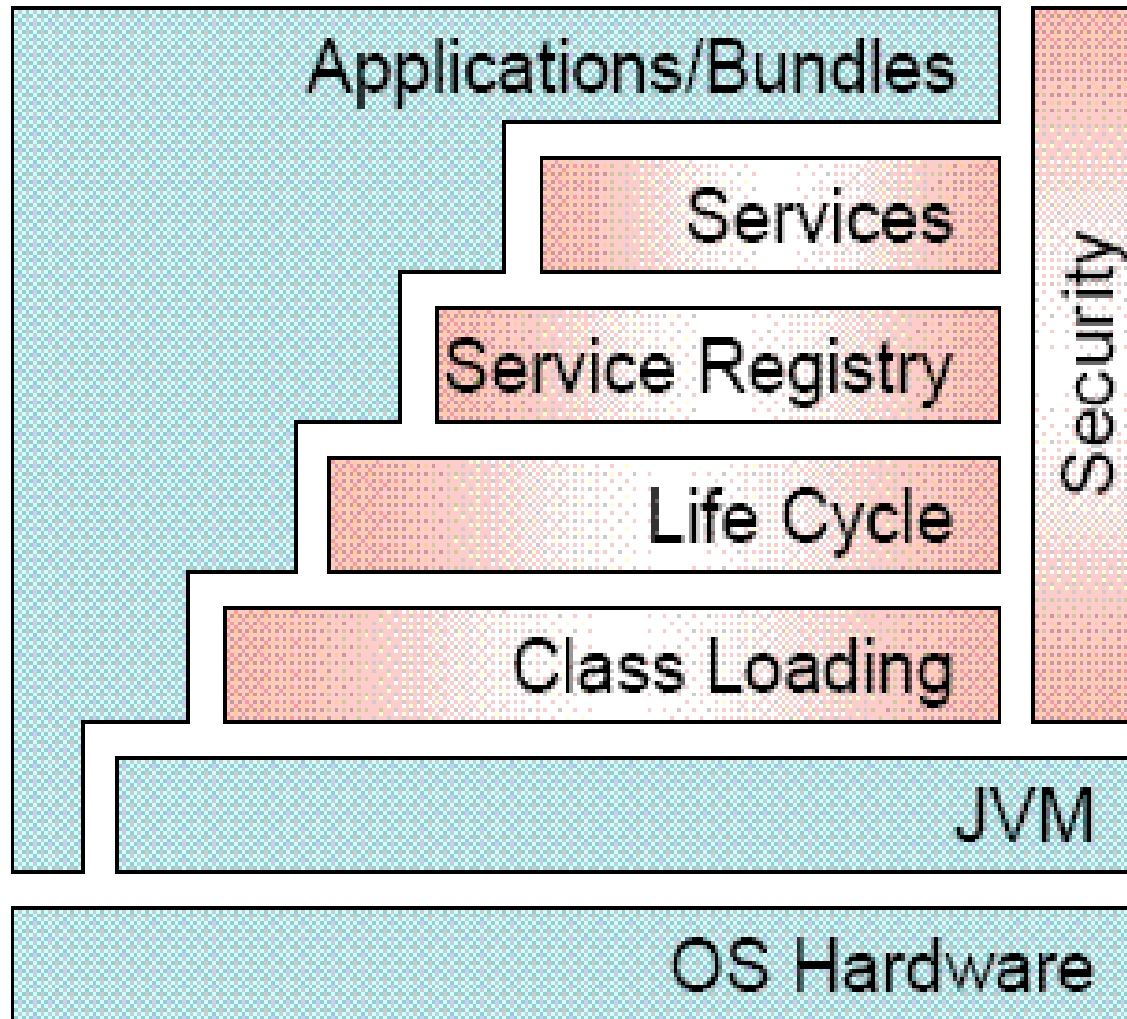


Przykłady zastosowań

- Telefony komórkowe
- Automatyka domowa
 - np. Shell HomeGene zawiera połączenie kamer, czujników, włączników itd.
- Eclipse IDE
- Przemysł samochodowy
 - np. BMW 5 series



Architektura OSGi





Architektura OSGi cd

- Środowisko wykonania: JVM
- Class Loading
 - bazuje na Java class loading, export, import packages
- Demo
 - exported packages, imported packages, required plugins



Architektura OSGi cd

- Life Cycle Management
 - start, stop bundle, obiekt BundleContext - API frameworka
 - BundleActivator - metody start, stop
- Service Registry
 - Bundle mogą dynamicznie startować / zatrzymywać się
 - Wykorzystanie listenerów na zdarzenia byłoby zbyt obciążające
 - Service Registry: rejestracja obiektów, wyszukanie, otrzymywanie powiadomień o konkretnych obiektach
 - Przechowywany w pamięci
- Services, Security, Applications/Bundles



Demo - Utworzenie nowego serwisu

- Utworzenie service
- Kroki rejestracji
- Wykorzystanie serwisu w innym bundle
 - Imported packages: org.osgi.util.tracker, servicebundle
 - Exported packages: servicebundle
- Wyeksportowanie bundles i zainstalowanie ich w runtime



Eclipse (OSGi) z Eclipse (IDE)

- Uruchomienie jako nowej instancji Eclipse
- Run dialog > Eclipse instance
 - Wykorzystywane pluginy, środowisko działania
 - opcja -console
 - Uruchomienie samego plugina - nie działa, serwis http nie jest wystartowany
 - odszukanie bundla z http - status
 - Uruchomienie ręczne bundle, który udostępnia serwis - zacznie działać
- Zmiana treści zwracanej przez servlet
 - hot code replace