



Using OSGi technology in Eclipse

Jeff McAffer

Equinox Project Lead

24 October 2003



Eclipse Overview

- *Original Mission: Create a platform independent, language/domain neutral platform for tooling*

***Provide “middleware” for
application development tool builders***

- *Main features:*
 - *Plug-in model: similar to bundles*
 - *Extensions & Extension-points: declaratively specify connections*
 - *SWT: platform independent, native-based widget toolkit*
 - *UI framework: infrastructure for GUIs*
- *One of the largest open source projects in existence*



Eclipse as a Rich Client Platform

- *Eclipse community is pushing into non-tooling domains*
 - *Productivity applications, enterprise applications, ...*
- *Work in progress for Eclipse 3.0*
- *Remove assumptions driven by tooling domain*
- *Address end-user requirements*
 - *Dynamic function*
 - *Security*
 - *UI Scalability*
 - *Constrained execution environments*

A good match for OSGi technology



eclipse.org History

- *Started in 1999 as an internal IBM tooling platform effort*
- *Fully open source Eclipse 1.0 November 2001*
- *Eclipse 2.0 June 2002*
- *Eclipse 2.1 March 2003*
- *eclipse.org hosts wide range of projects;*
 - *Java, C and Cobol tooling*
 - *graphical editors*
 - *collaborative work environments*
 - *Aspect-oriented programming, courseware, ...*

Runaway success!



Who's on Board?

eclipse.org consortium includes most major tooling vendors



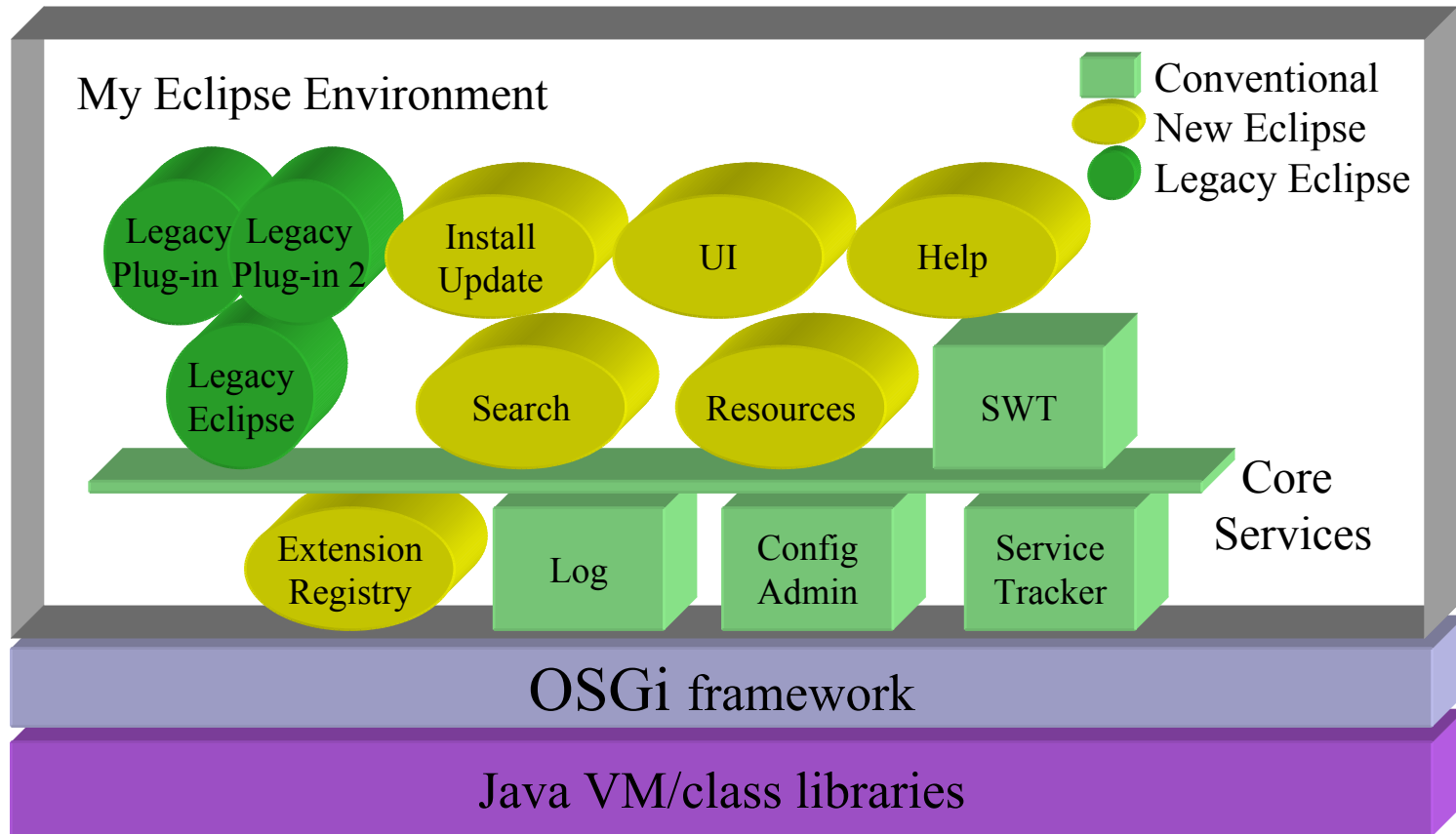


How OSGi technology is being used by Eclipse

- *OSGi Service Platform is a well-defined runtime specification*
- *Replace the Eclipse runtime with an OSGi framework implementation*
- *Eclipse plug-ins map onto OSGi bundles, plus:*
 - *Applications*
 - *Extensions/Extension points*
- *Eclipse backward compatibility layer (99% binary compatible)*
- *Use existing OSGi services (e.g., URL, XML, ...)*
- *Use services to manage dynamic behaviour*



OSGi Technology Based Architecture





Pressures on OSGi Technology

- *Wider range of usage scenarios*
 - *Desktop environment*
 - *5000+ bundles*
 - *User managed installations*
- *Multiple bundle/package versions a requirement*
 - *Multiple independently developed bundle sets concurrently installed*
 - *Each demands different versions*
- *Laziness is next to scalability*
 - *Declarative services*
 - *On-demand bundle activation*



Eclipse and OSGi Service Platform Specification

- *Working through the RFP/RFC process (see RFP 44) to address some of the challenges for the next release of the OSGi Service Platform specification*
- *Bundles as modules*
- *Bundle fragments*
- *Multiple concurrent versions of bundles/packages*
- *Demand loading of services*
- *Transactional bundle install/update/uninstall*
- *Bundle suites*



eclipse.org Timeline/Schedule

- *Currently working on Eclipse 3.0*
- *Target release 2Q04*
- *OSGi technology based runtime available now*
 - *Equinox project*
<http://eclipse.org/equinox>
- *Initial steps are modest: Use but not exploit*
- *Future uses will exploit/expose more OSGi technology*

Visit eclipse.org



OSGi[™]
Alliance
2003 World Congress



Questions?