



*Pulsar*

*Mobile Architecture Proposal*

*v1.1*

*May 6, 2010*

*Eclipse Mobile Industry Working Group*

[Mailing list: mobile-iwg@eclipse.org](mailto:mobile-iwg@eclipse.org)

# *Disclaimer*



- The Architecture proposal for the Eclipse Pulsar product is intended to be used as a reference for mobile focused efforts within Eclipse
- The document is a living document and will be updated/changed as needed to meet the evolving technology requirements for mobile

# Version



Version	Date	Comments
1.0	24 Apr 2010	Initial proposal
1.1	6 May 2010	Rking – Updates to new template and disclaimer

# *Motivation*



- Developer Challenges
  - Fragmentation of devices and technologies
  - Necessity of installing multiple vendor tools and SDKs
  - Difficulties integrating vendor tools with build systems
- Realities
  - Fragmentation will not go away
  - ...but it may be mitigated through tooling

# *Motivation*



- The ideal
  - Mobile developers maintain a single set of sources...
  - ...to produce multiple products...
  - ...with a single suite of tools
  - A mobile development suite so powerful and comprehensive that developers will rally around it
- If we can alleviate developer challenges...
  - An enthusiastic embrace of the tooling
  - A strong Eclipse mobile developer community

# *Motivation*



- Benefit to Vendors
  - Shared tooling development
  - High quality of tooling
  - Access to a large and motivated mobile developer community
  - Visibility and recognition
  - Vendors compete on device features and less on tooling

# *Requirements*



- It should be...
  - Well designed
  - Generic
  - Easy to use
  - Completely integrated with Eclipse
  - Vendor extensible\*

\* Ideally, an architecture could be designed such that it encourages vendors to extend it rather than build proprietary applications on top of it.

# *Requirements*



- It should have...
  - A shared core of common features for all mobile tooling
  - No duplication/conflict with functionality supplied by Eclipse core projects
  - Specialization for specific mobile technologies (projects)
  - A common user interface where possible
  - A robust architecture



# Architecture



- **Overview**

- Specialized projects (Web, Java, native) are built on top of a common layer. Specializations within a project (eg. J2ME vs. Android) are supported by the project.

Widgets	Web	J2ME	CLDC	CDC	Android	Linux	Other
Eclipse Mobile for Web		Eclipse Mobile for Java				Eclipse Mobile for Native	
Eclipse Mobile Common							
Eclipse Projects							

# Architecture



- **Eclipse Mobile Common**

- Supplies common components required by all mobile projects. Offers only functionality unique to mobile that cannot be found in the Eclipse projects.
- Includes SDK installation/management, device management, permissions, pre-processing, signing, etc.

Device Management	Preferences	Permissions
Localization	Pre-Processing	Obfuscation
Building	Packaging	Signing
Debugging	Unit Testing	Deployment

# Architecture



- **Eclipse Mobile Common**

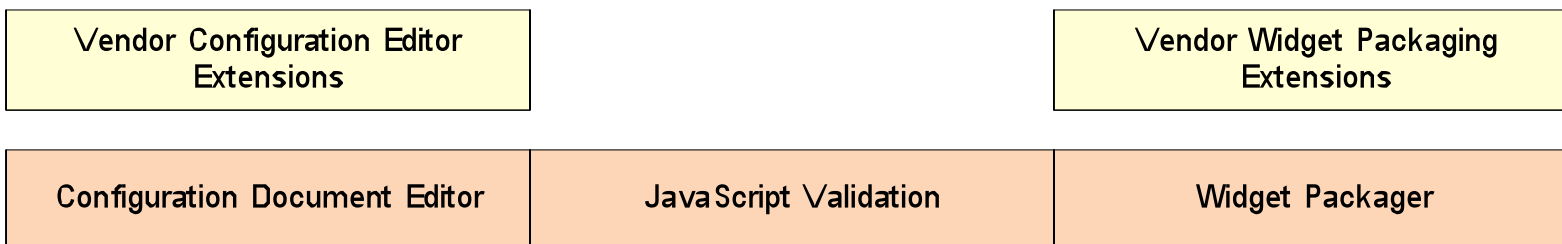
- Include a unified framework for Device Discovery and Target Management
- Targeting for OSS simulation (QEMU) and vendor emulators
- Mobile project creation ( project nature), Template wizard model
- Integrated SDK help framework

Device Management	Preferences	Permissions
Localization	Pre-Processing	Obfuscation
Building	Packaging	Signing
Debugging	Unit Testing	Deployment

# Architecture



- **Eclipse Mobile for Web**
  - Project that supports Web-like development for mobile
  - Allows developers to create browser-based applications and Widgets
  - Edits HTML, JavaScript, CSS, configuration documents, and other Web artifacts
  - Packages Web-based mobile applications



# Architecture



- **Eclipse Mobile for Java**
  - Project that supports all flavors of mobile Java development
  - It would essentially be MTJ with common items extracted and additional Java support added

Manifest Editor	Preverification	Java Localization
Aligning (Android)	Java ME Unit Testing	

# Architecture



- **Eclipse Mobile for Native**

- Project that supports multiple C/C++ based platforms
  - Symbian/Qt
  - Linux/MeeGo
- Flexible build/post build packaging support for common builders
  - Make, Qmake, SBSv2
  - Debian Packaging, Symbian SIS
- Fast indexing support for large SDK environments

C++ Builder (CDT)	Flexible C++ Project Model	Indexing (CDT)
C++ Unit Testing	Dynamic/Static Analysis	Application Packaging (SIS/Debian)