

The background of the slide is a photograph of ocean waves. The top portion shows a wave cresting with white foam against a teal-green sea. The middle section is a solid dark blue band containing text. The bottom portion shows the churning, white-foamed wake of a boat in greyish-blue water.

Aaron Schultz

Software Developer

National Center for
Ecological Analysis and Synthesis

Kepler Architecture Solution Eclipse Plugins

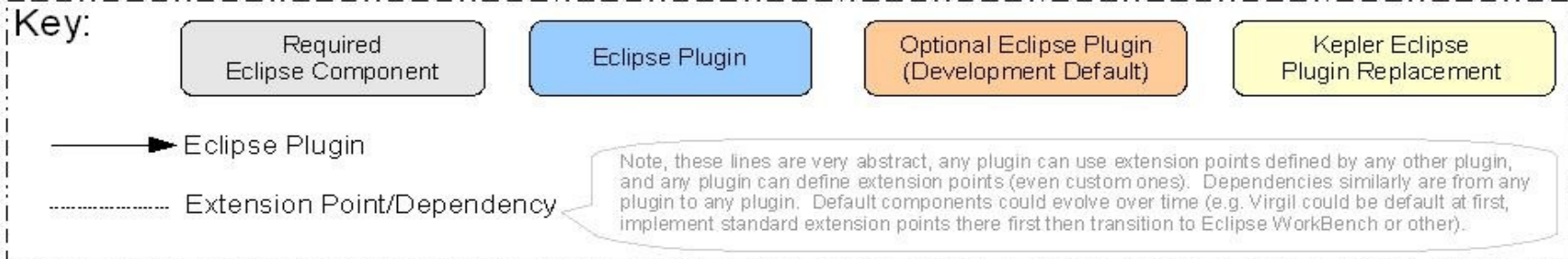
2008-06-13

- Eclipse Background
- Solution Diagram
- Kepler Usage of Eclipse
- Building Kepler
- Benefits
- Risks
- Reference

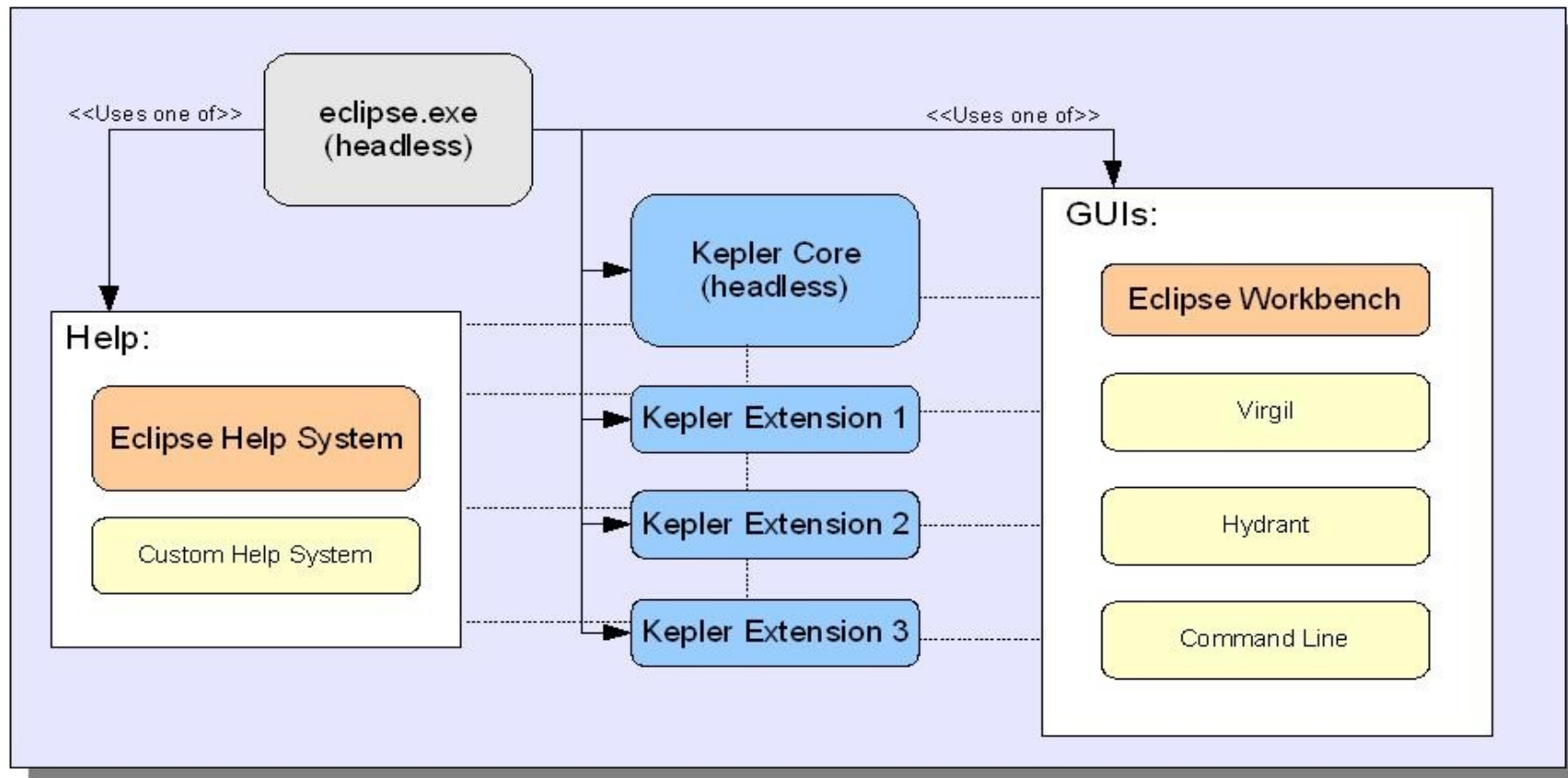
Eclipse Background

- Eclipse is not what you probably think it is, a full blown IDE. The IDE is just an optional Eclipse plugin. Eclipse is really a 56 KB executable that implements a plugin mechanism.
- The Eclipse IDE was developed at IBM for a reported \$40 million and released as open source software in November of 2001
- An industry consortium came together in 2003 to form the Eclipse Foundation
- In June 2004 Eclipse 3.0 was released and had (mostly*) adopted the OSGi standard
- New Eclipse versions are released yearly at the end of June
 - 3.2 Callisto June 30, 2006
 - 3.3 Europa June 29, 2007
 - 3.4 Ganymede June 25, 2008
- Eclipse is licensed under the Eclipse Public License and is not compatible with GPL software for commercial distribution

Solution Diagram



Solution: Kepler as Eclipse plugin



- Eclipse uses a peer delegation classloader model to provide independent namespaces and on-demand class/resource loading for plugins
- Eclipse includes a full suite of Plug-in development tools
- Eclipse plugins can define their own extension points for other eclipse plugins in a standardized way
- Dependency management is handled by editing xml files with the graphical Plug-in development tools
- Product configurations allow for building Rich Client Applications on 4 platforms (win32, macosx, linux, solaris)
- Eclipse fragments allow for adding functionality to a plugin

Kepler usage of Eclipse for plugin management

- Kepler need only use the Eclipse extension points that it needs to operate
- Eclipse Extension Points can be reused in custom code (e.g. Eclipse Workbench extension points can be reused in a Web Interface or custom GUI like Virgil)

- Maven integrates with Eclipse and can be used for the NMI nightly build system

- Already very well developed and documented
- Large existing community to draw on for support
- All the benefits and flexibility of separate namespaces and independent Kepler extensions
- Greater visibility of Kepler to a large developer community

Drawbacks/Risks

- Would need to refactor some code in order to keep major components like Virgil
- Some maintenance cost in upgrading each year for the new Eclipse version
- Some learning curve for developers
- Eclipse would become standard for developing and building Kepler
- Unknown how this would work with NMI build system
- Unknown if overriding pieces of other plugins is allowed (although I've seen hints that it is I'm still not really sure)

References

- IBM makes \$40 million open-source offer CNET news 2001
- Eclipse Wikipedia Page
- Plug-in development 101 Parts 1-2 IBM 02/2008
- Eclipse Development Visualization (just for fun)