



All About Platforms

Lessons Learned from Eclipse

Mike Milinkovich
Executive Director
Eclipse Foundation

April 22, 2006

Four Key Elements of a Platform



1. Great technology
2. Broad Adoption
3. “Architecture of Participation”
4. Hijacked by a fanatical community

Four Key Elements to Platform

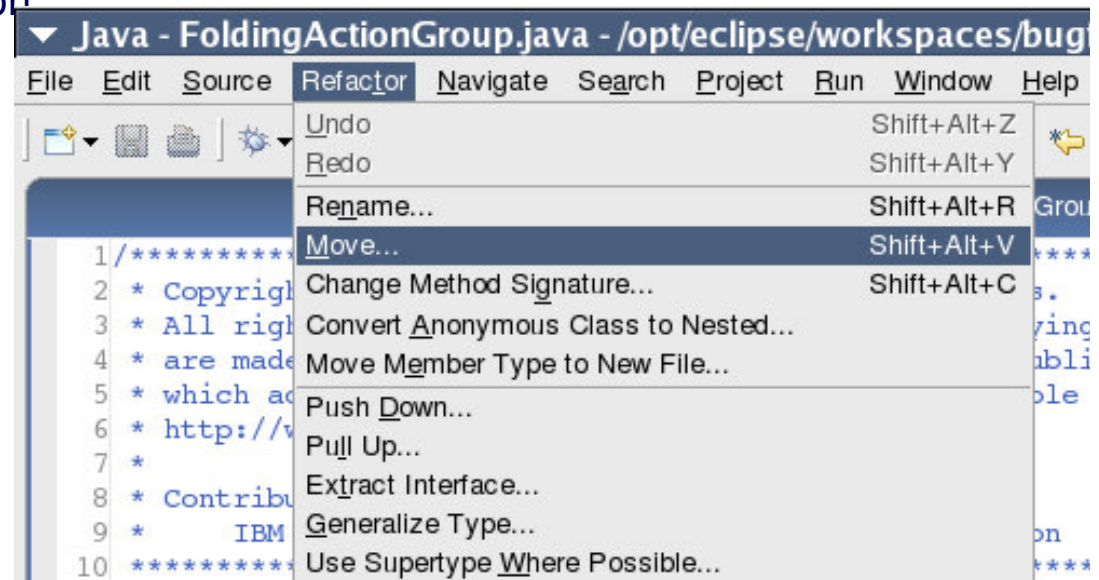


1. Great technology
2. Broad Adoption
3. “Architecture of Participation”
4. Hijacked by a fanatical community

Eclipse is a Java IDE



- Widely regarded as **the** Java development environment
- With all the bells and whistles...
 - Language-aware editors, views, ...
 - Refactoring support
 - Integrated unit testing and debugging
 - Incremental compilation and build
 - Team development support
 - Out of the box support for CVS
 - ...



Eclipse is an IDE Framework

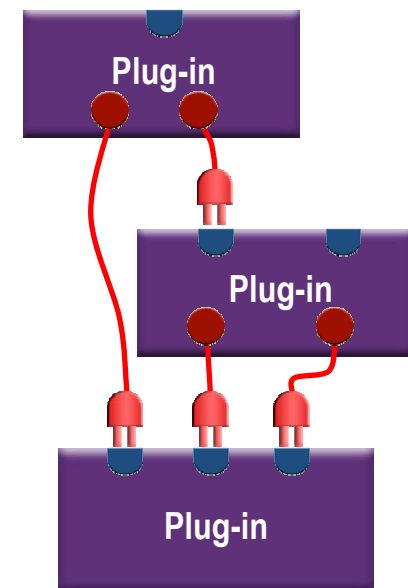


- Eclipse + JDT = Java IDE
 - First class framework for Java
 - Language aware editor
 - Incremental build
 - Integrated debugging
- Eclipse + CDT = C/C++ IDE
 - First class framework for C/C++
 - Language aware editor
 - Refactoring, search
- Eclipse + PHP = PHP IDE
- Eclipse + JDT + CDT + PHP = Java, C/C++, PHP IDE
- ...

Eclipse is a Tools Framework



- Extensibility through OSGi implementation
 - Plug-ins make Eclipse whatever you need it to be
- Focus on developing a universal platform of frameworks and exemplary tools
- Tools extend the Eclipse platform using plug-ins
 - Business Intelligence and Reporting Tools (BIRT)
 - Eclipse Communications Framework (ECF)
 - Web Tools Project (WTP)
 - Eclipse Modelling Framework (EMF)
 - Graphical Editing Framework (GEF)
 - Test and Performance Tooling Project (TPTP)



Eclipse is a Application Framework



- Remove the IDE elements, Java language support, team development support, ... and you're left with a pretty comprehensive general application framework
 - Support for multiple platforms
 - Linux, Windows, Mac OSX, UNIX, embedded
 - Rich widget set, graphics
 - Native-OS integration (drag and drop, OLE/XPCOM integration)
- A platform for rich clients
- Some examples of this later...

Eclipse is an Open Source Enabler



- Nine (9) top level projects
 - The Eclipse Project
 - Tools
 - Web Tools Platform
 - Test & Performance Tools Platform
 - Business Intelligence and Reporting Tools
 - Data Tools Platform
 - Device Software Development Platform
 - SOA Tools Platform
 - Technology (Incubators)
- Dozens of sub projects

Eclipse is an Open Source Community



- All Eclipse projects are available for free download
 - No registration, no tracking, no sales calls, ...
- Contributors number in the hundreds
- Hundreds of plug-ins provided by commercial companies, organizations, and individuals
- Enthusiastic blogger community
 - www.planeteclipse.org
- Numerous Eclipse portals, etc...
 - EclipseZone (<http://eclipsezone.com>)
 - EclipseWiki (<http://eclipsewiki.editme.com>)
 - Eclipse Plug-in Central (<http://eclipseplugincentral.com>)
 - Others (<http://eclipse.org/community>)

Eclipse is an Eco-System



- Eclipse is focused on nurturing the eco-system to complement, and enhance the Eclipse Platform
- 128 members, including major Java, Linux and Embedded vendors
 - BEA, Borland, JBoss, IBM, SAP, RedHat, Novell, Monta Vista, Wind River, Mentor, ENEA, QNX
- Dozens of open source projects
- 50 million download requests to date



Eclipse is a Foundation



- The Eclipse Foundation
 - Was created to manage and direct the ongoing development of the Eclipse open source software project
 - Is responsible for providing IT infrastructure required by development teams
- Independent not-for-profit Foundation formed in 2004



Eclipse is all these things...



- A Java IDE
- An IDE Framework
- A Tools Framework
- An Application Framework
- An Open Source Enabler
- A community
- An eco-system
- A foundation

Four Key Elements to Platform



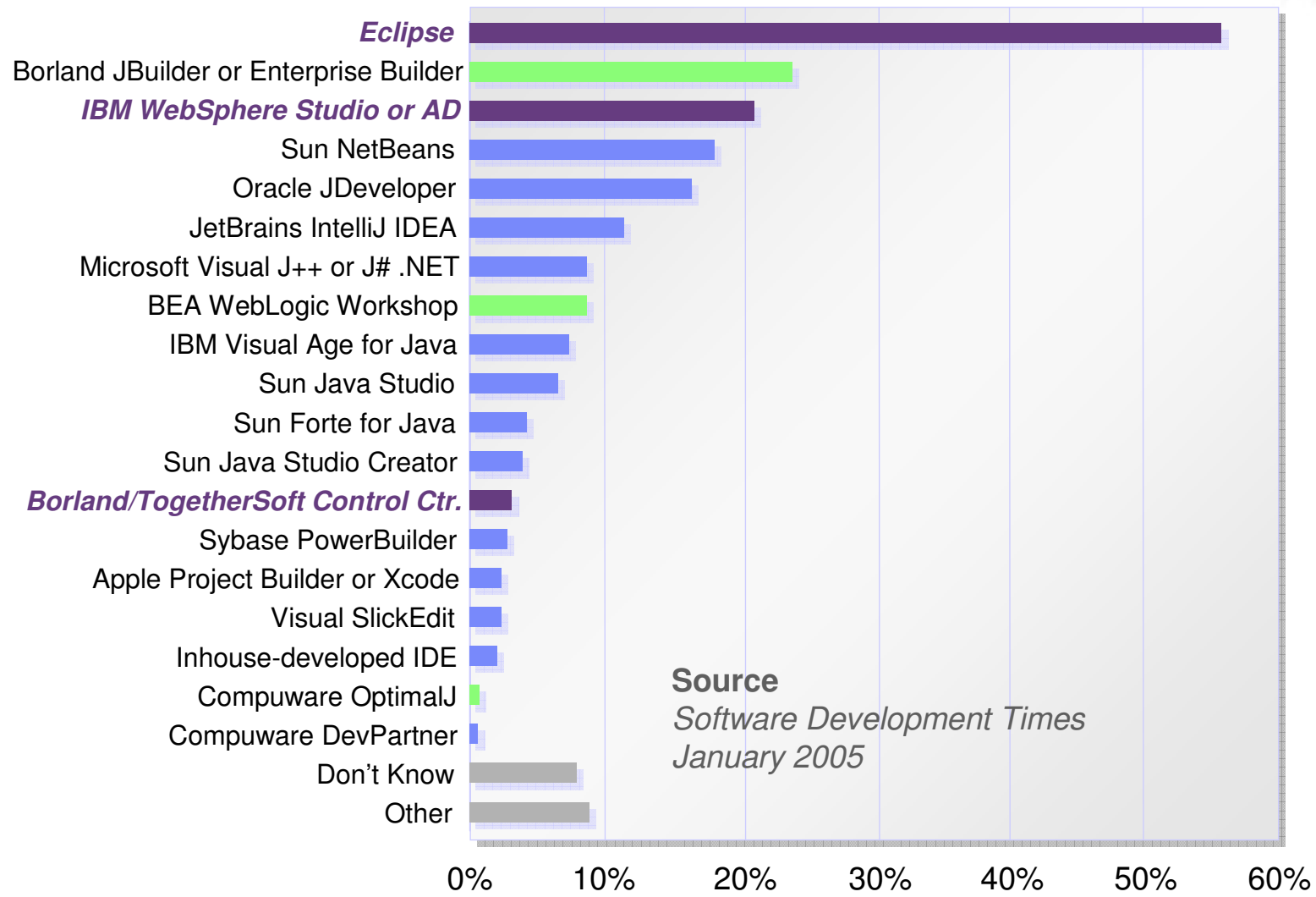
1. Great technology
- 2. Broad Adoption**
3. “Architecture of Participation”
4. Hijacked by a fanatical community

Eclipse Today



- 130+ members, including major Java, Linux and Embedded vendors (Actuate, HP, IBM, Sybase, BEA, Borland, JBoss, SAP, RedHat, Novell, Nokia, Monta Vista, Wind River, Mentor, ENEA, QNX)
 - *Hundreds of shipping products based on Eclipse technology*
- 50+ open source projects
- 55-65% market share of Java IDE
- Rich Client Platform adoption growing like crazy
- Downloads: 50 million++; 1 million in 40 days for last release
- Web site visits:
 - 36,000 unique visitors per day
 - 1,140,000 unique visitors per month (IP addresses)
- Bugs: 126K so far....(typically ~15K-20K per major release)
- Newsgroup postings: 9K/month

Eclipse – Widespread Adoption



Four Key Elements to Platform



1. Great technology
2. Broad Adoption
- 3. “Architecture of Participation”**
4. Hijacked by a fanatical community

Defining the Term



*“what really distinguishes open source is not just source, but an **architecture of participation**” that includes low barriers to entry by newcomers, and some mechanism for isolating the cathedral from the bazaar. This architecture of participation allows for a real free market of ideas, in which anyone can put forward a proposed solution to a problem; it becomes adopted, if at all, by acclamation and the organic spread of its usefulness.”*

Tim O'Reilly

<http://www.oreillynet.com/pub/wlg/3017?wlg=yes>

Key Points

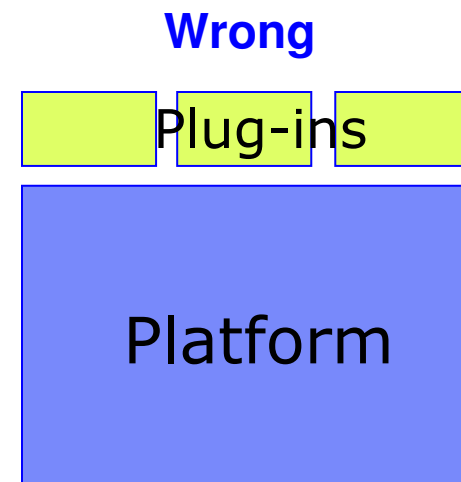
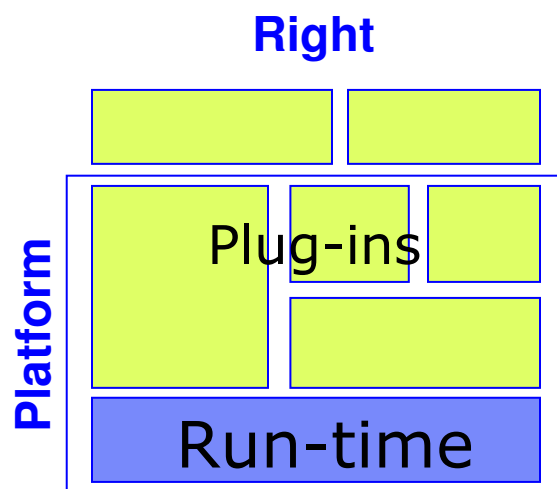


- You need a cathedral to enable the bazaar
 - This is not about joining the project itself *per se*, but about making it easy for individuals to add their own ideas to the ecosystem
- Low barriers to entry
- New add-ons are first class citizens, not second class
- Empowering individuals and small groups is key
 - Innovation comes from committed fanatics in small groups, not corporate committees
- *Laissez faire* --- never ever ever pretend you can pick a winner
- Provide the seeds for a broad ecosystem

The Eclipse Experience



- Eclipse (Rich Client) Platform
 - It has an open, extensible architecture
 - Built out of layers of plug-ins
 - Everything is a plug-in...both from the cathedral and from the bazaar



Distinctively Eclipse



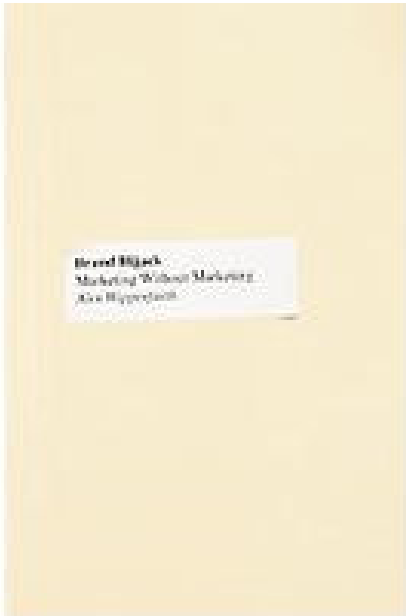
- Predictability spurs adoption
 - June 28, 2004: Eclipse 3.0 ships
 - June 28, 2005: Eclipse 3.1 ships
 - June ??, 2006: Eclipse **Callisto** ships
- Quality matters
 - APIs: If you build it, they will come
 - Platform Quality APIs are formally part of the Eclipse project culture

Four Key Elements to Platform



1. Great technology
2. Broad Adoption
3. “Architecture of Participation”
4. Hijacked by a fanatical community

Definition: Brand Hijack



- Let the community define what “Eclipse” means
- The antithesis of traditional marketing
 - Let the community take over
 - Be spontaneous and fun, not control-oriented
 - Embrace uncertainty
- Embrace community mavens and leaders
- Win by letting go

Brand Hijack: Marketing Without Marketing

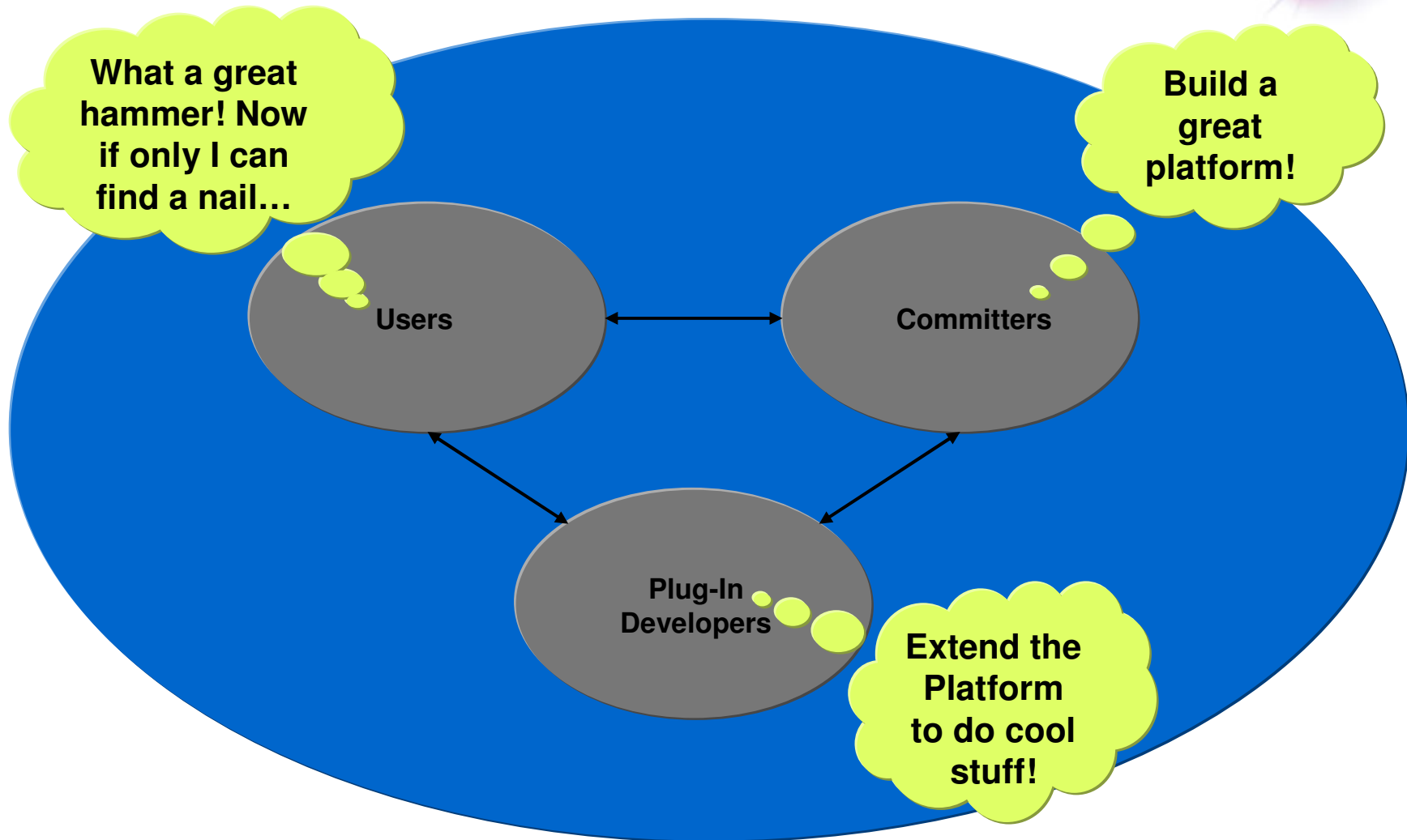
By Alex Wipperfurth

The Anatomy of a Community

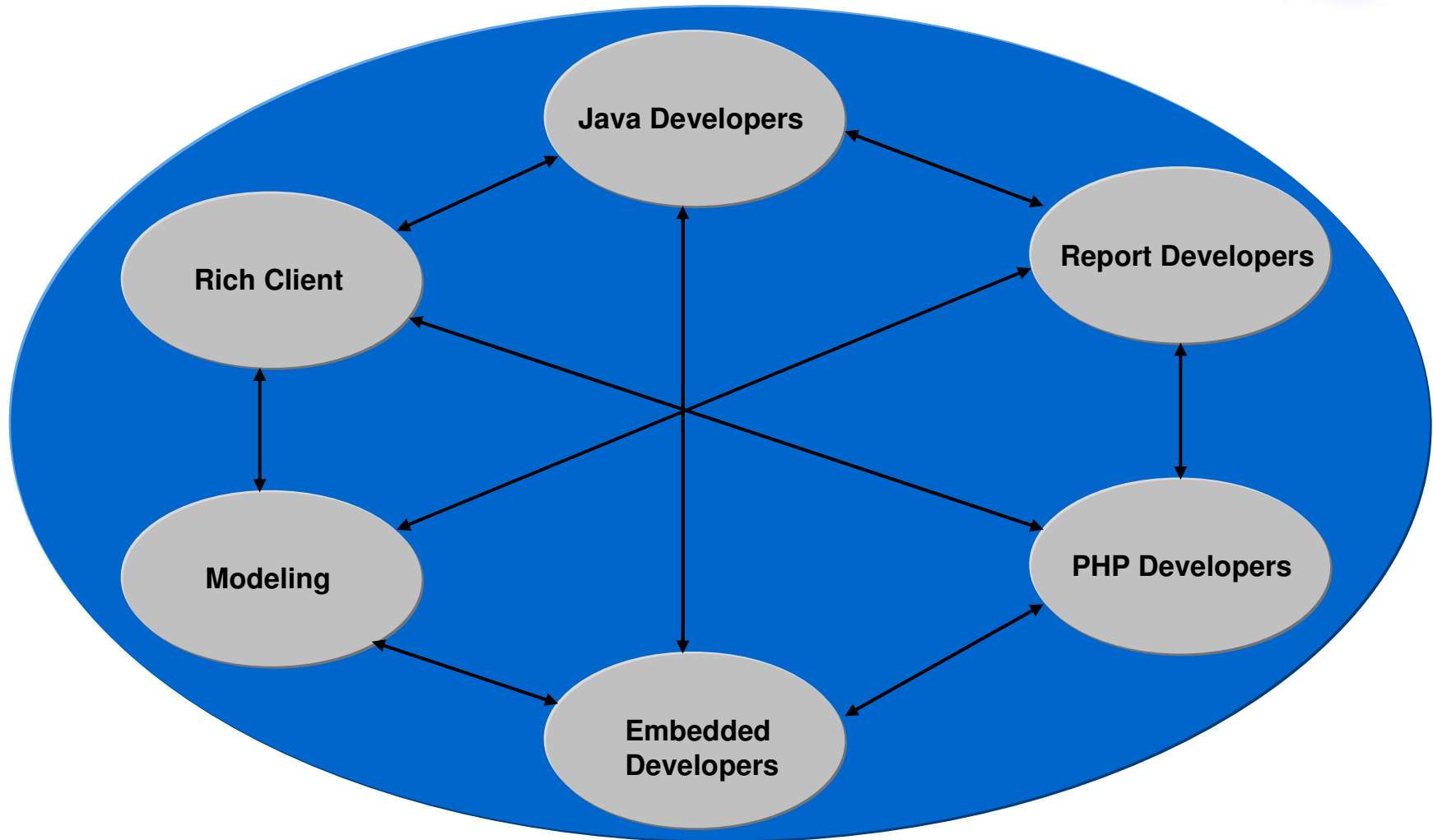


The Eclipse Community

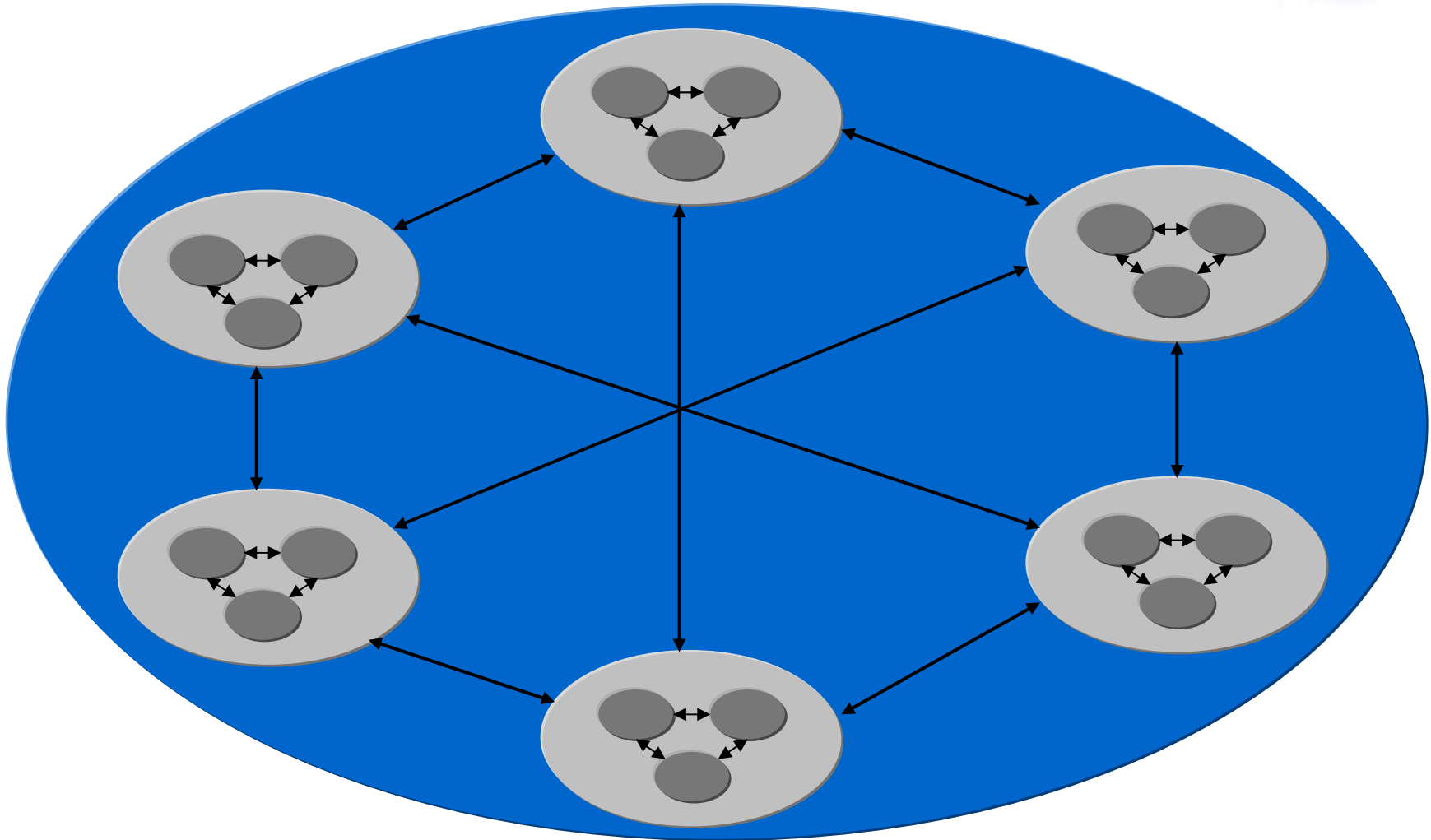
Role-based Communities



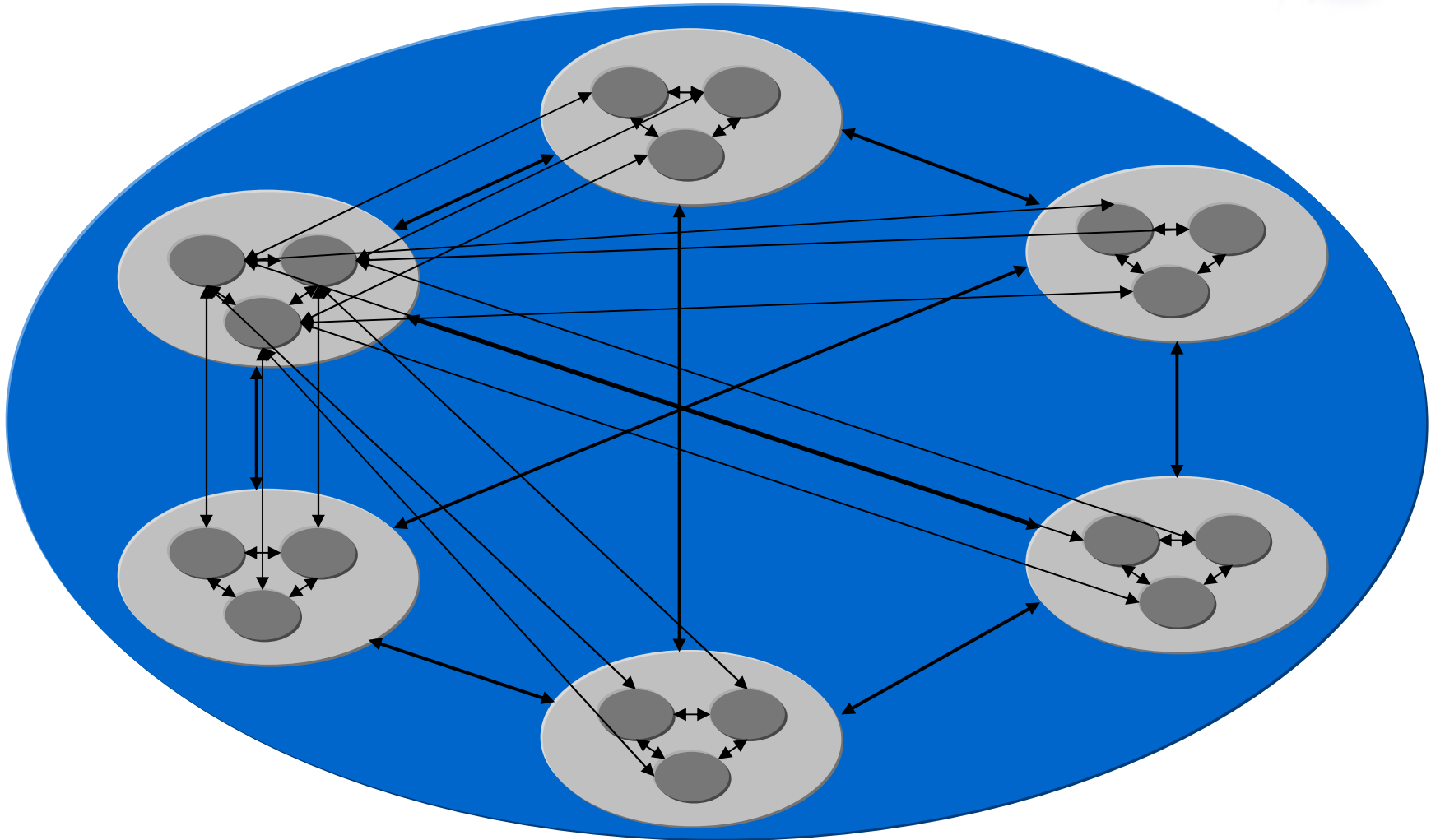
Technology-based Communities



The real picture



Metcalfe's Law Works for Communities as Well



Eclipse Community by the Numbers



- 70 project leaders, PMC members
- 639 committers
- 24,426 registered Bugzilla users
- Estimated 2 million ++ users

But is it fanatical?



- Eclipse 3.1 metrics
 - Number of newsgroup posts: 32,223
 - Number of mailing list posts: 7,792
 - Number of Eclipse-specific blogs: 27
 - Number of eclipse.org page views: 500,000,000
 - Average Bugzilla queries per day: 600,000
 - Bugzilla bug or enhancement reports entered: 21,831
 - Bugzilla reports fixed: 9,871
 - Most comments on a single bug: 202
 - Number of bugs reported entirely in verse: 2

How to get there



- First, get rid of all the marketers
- Second, admit that we're all marketers
- If you love something, set it free
- It's about the human conversation
 - And developers want to talk to developers, not marketing flacks
 - And the conversation is about code
- Embrace others as leaders
- Be transparent
- Say "Thank You"

If you're interested in creating passionate users, or keeping your job, or breathing life into a startup, or getting others to contribute to your open source project, or getting your significant other to agree to the vacation you want to go on... congratulations. You're in marketing. Now go kill yourself.

... and then he said, "Shoot me, Jimmy, shoot me... I can't take the pain" and then I said, "You're gonna be OK Joe..." but I was lying. He was my best friend, and I had to shoot him... they'd transferred him to marketing.



Credit: Creating Passionate Users

WRONG - Bzzt



“My development manager thinks we are being out marketed, so I would like to do a press release....”

Press release do not create a conversation

Developers and Marketers need to share the responsibility of creating the community.

Summary



- Platforms require
 - Great technology
 - Broad Adoption
 - “Architecture of Participation”
 - Hijacked by a fanatical community
- The next generation of broadly adopted platforms won't all be open source, but open source sure helps



Thank You!