

Value Creation and Cooperative Resource Exchange in the Eclipse Ecosystem

Patrick Wagstrom
Sonali Shah
James D. Herbsleb

The contents of this presentation
are released under the terms of the
Eclipse Public License 1.0



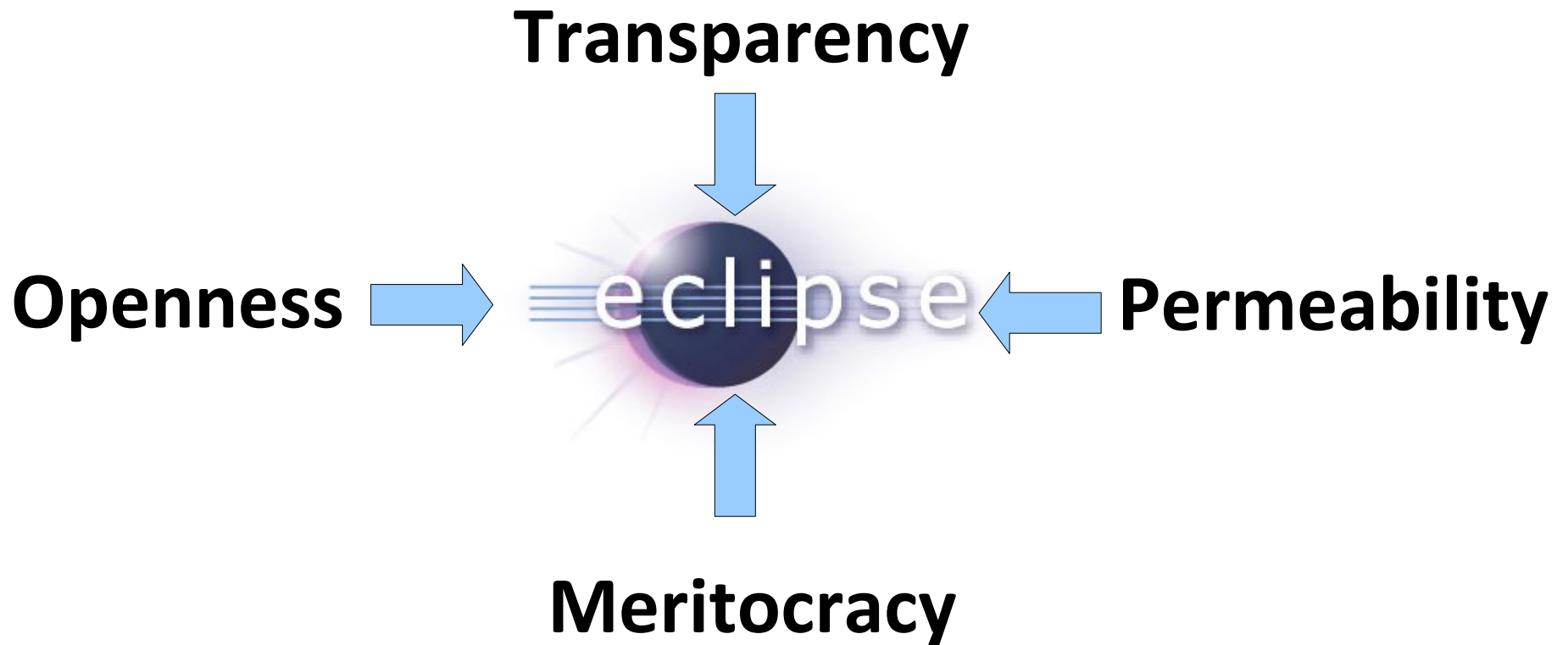
Our Problem in a Nutshell

- Each year sees more OSS communities
- Eclipse is seen as a model for OSS development
- What is it about Eclipse that makes it succeed while other communities fail?

Background and Method

- We're researchers interested in the “bigger” picture
- Interviewed 32 people (still looking for more)
- Read up on almost every Eclipse member company
- Hope to add quantitative elements

Four Principles





The Why and the How

- Eclipse currently has two levels of membership:
 - Strategic Developer
 - Add In Provider
- Membership level does not indicate how or why some firms participate

Why Firms Participate

- Market Consolidation
- Product Sales
- Complimentary Goods
- Consulting Services
- Usage Convenience



How Firms Participate

- Platform Stewardship
- Platform Extension
- Plug In Development
- Platform Building
- Users

Merging the How and Why

	Market Consolidation	Product Sales	Complimentary Goods	Consulting Services	Usage Convenience
Platform Stewardship	X				
Platform Extension		X	X		X
Plugin Development	X	X	X	X	
Platform Building		X	X	X	
Users					X

Platform Stewardship

- Requires large company with many resources
 - IBM
- Four principles are not necessary
 - IBM has experience building proprietary ecosystems
- Expansion of Eclipse ecosystem makes it possible for additional stewards
- As Eclipse expands, is it possible to survive without additional stewards?

Platform Extension

- Heavy modification of Eclipse
- Typically independent
 - Meritocracy is less important
- Transparency, Openness, and Permeability helpful to control project
 - Ensure changes to the trunk don't venture too far away

Plugin Development

- Rely heavily on openness, transparency, and permeability
 - Meritocracy rarely mentioned
- Most firms indicated that their plugin development was all internal
- Not much of a difference from other ecosystems (Visual Studio)

Platform Building

- Not what people normally think about when referring to Eclipse
- Inherit the four principles from Eclipse
- Generally too new to evaluate to what degree the four principles are necessary

Users

- Hardest group to track
- Many highlighted the four principles
 - Few did anything related to the principles
 - Saw the principles as a guarantee of sorts
- Some firms developed internal plugins

Organizational Highlights

- Foundation Structure
- Firm Cooperation
- End Users
- Future Ecosystem Expansion

Foundation Structure

- Structure was almost universally positive
- Organization with no business motive owning code was key factor
 - unambiguous licensing and ownership
 - independence of short term business pressures
 - no reliance on external firms
- The foundation brand is trusted
- Bold step for IBM to release control

Issues with First Movers

- The Eclipse release train is powerful
- Eclipse structure makes it difficult to propose an alternative project
- The first project usually “wins”
- But the first project is not always the best
- Lack of integration leaves little incentive to release competing tools as Open Source

Cooperation in Eclipse

- Clear benefits for IDE developers
 - Starting to see many new IDEs based on Eclipse
- Plugin developers often participate in competing Open Source projects to get a preview of future competition
- Developers of complimentary goods are able to compete on a level playing field

Public Goods Problem

- Has Eclipse solved the public goods problem?
 - No
- Foundation and member companies have different planning horizons
- Common components frequently subject to the “Eclipse bluffing game”
- Solution is not to have Foundation do the work
- Does Eclipse need additional stewards?

End User Success

- Plugins allow firms to customize project for their setup
- Eclipse distributions take this a step further and make it more useful
- Plugin documentation makes it easy for firms to customize

End User Issues

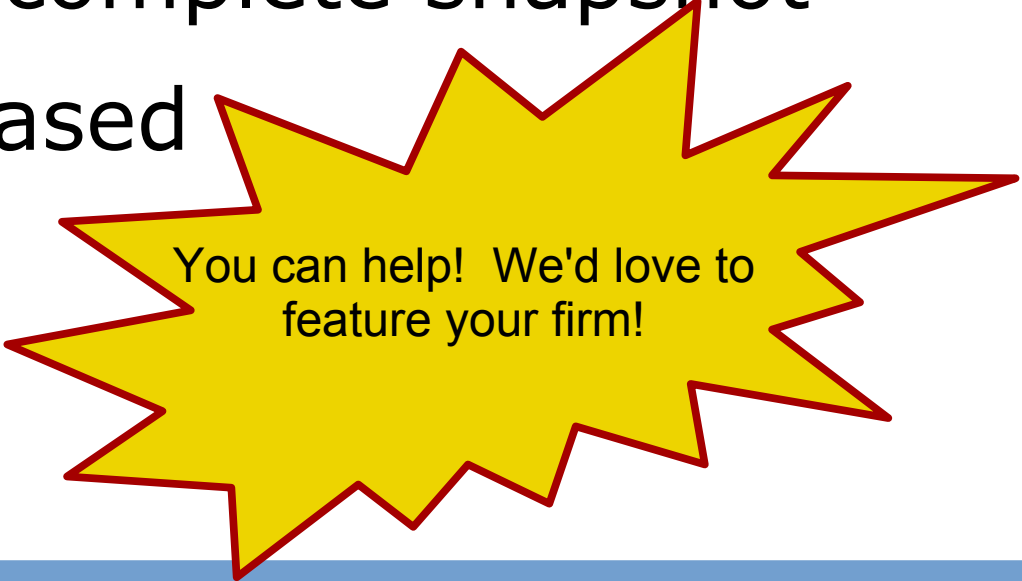
- Distributed development makes it difficult for end users to get a say
- Traditional feedback channels are not present
- Large firms can hire developers to customize
- Smaller firms face more difficult prospects

Future Ecosystem Expansion

- Eclipse is moving beyond the IDE – but most people don't realize it
- Nested Platform Building
 - Eclipse Runtime Project is HUGE
- Brand confusion issues
 - Similar issues face Apache

Future issues

- This is based only on what people say
 - Research has shown people don't always perceive reality
- Dynamic nature of Eclipse makes it very difficult to get a complete snapshot
- Our sample is biased



You can help! We'd love to feature your firm!

Questions/Comments

- patrick@wagstrom.net (email/jabber)

This work was funded by the National Science Foundation, the Office of Naval Research, and the Software Industry Center at Carnegie Mellon University – a Sloan Foundation Industry Studies Center. The views expressed in this work are those of the author(s) and are not necessarily those of their institutions or funding agencies.