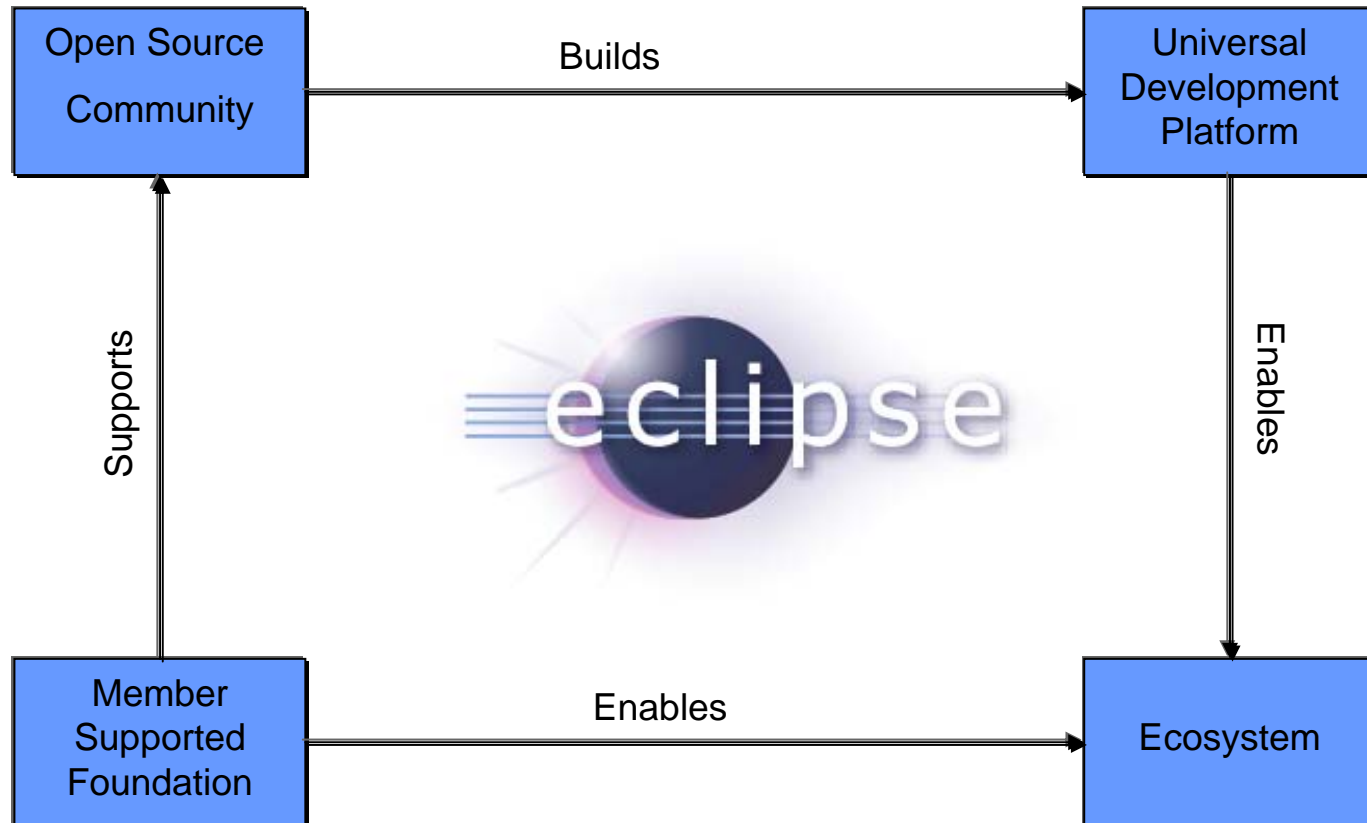




# Eclipse: Accelerating OSGi Adoption

Mike Milinkovich  
Executive Director  
Eclipse Foundation

# What is Eclipse?



# History of the Eclipse Organization



- Initiated by IBM in 2001 and supported by a wide range of vendors.
  - Original Consortium board comprised Borland, IBM, MERANT, QNX Software Systems, Rational Software, Red Hat, SuSE, TogetherSoft and Webgain
- Eclipse Today
  - Independent not-for-profit Foundation formed in 2004
  - 111 members, including major Java, Linux and Embedded vendors (BEA, Borland, JBoss, IBM, SAP, RedHat, Novell, Monta Vista, Wind River, Mentor, ENEA, QNX,...)
  - 50+ open source projects
  - 50 million download request to date
    - 1 million Eclipse 3.1 downloads in 40 days

# The Members of Eclipse



- 16 Strategic Members
- 80 Add-in Providers
- 15 Associate Members (Publishers, Research Institutes, Standards Org., etc.)
- Large community of open source developers (500++)



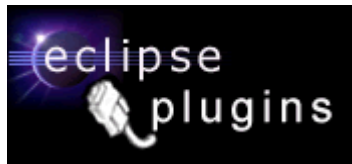
Computer Associates®



# Eclipse Eco-System – Community of Add-in Providers



- 80 Add-in Provider members;
- 950+ available Eclipse add-ins
- 500+ Eclipse based projects on SourceForge



# Example of Eclipse Based Commercial Tools



## Enterprise IT

- Borland Together Edition for Eclipse
- HP OCMP OClet Development Env.
- IBM Rational Application Developer
- Oracle Collaxa BPEL Designer
- SAP NetWeaver Studio

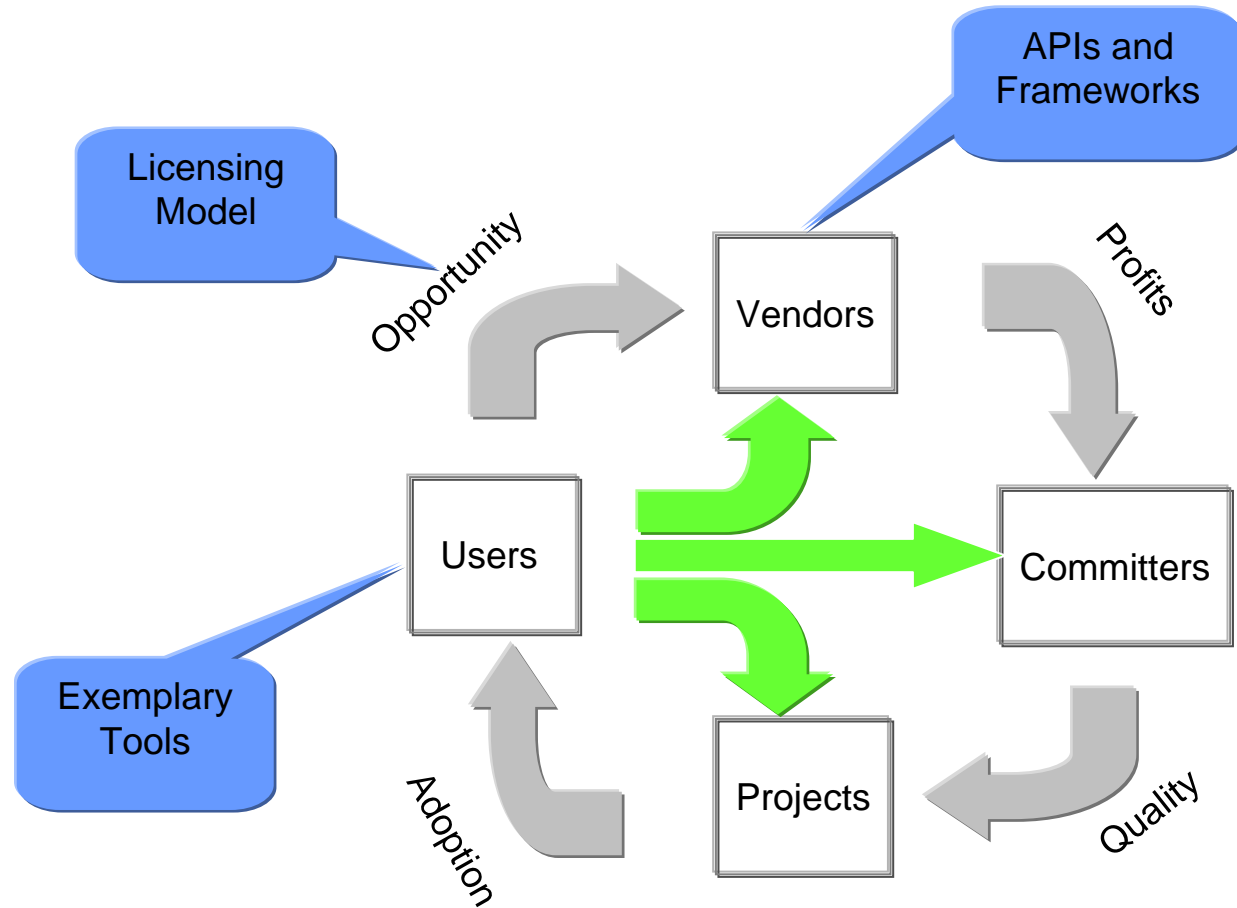
## Embedded

- PalmOS Dev Suite
- Monta Vista DevRocket
- Wind River Workbench
- QNX Momentics
- TimeSys TimeStorm IDE
- Tensilica Xtensa Xplorer IDE
- Mentor Graphics Nucleus Edge

## Linux

- Novell/SuSE Linux SDK
- Red Hat Developer Studio
- Intel Compiler for Linux

# Eclipse Economics 101



# Eclipse is a Level Playing Field



- Eclipse is unique in the transparent way in which software firms cooperate in the creation of new technology
- A new model:
  - The level playing field and open interaction of a standards organization .....with the wonderful attribute that the output actually runs!
  - Innovation happens best in development, not in standards organizations
- This model is a large component of what drives companies to join as Strategic Members
  - Shared cost, shared risk for developing the underlying building blocks for products
  - Often based on open standards
  - Still lots of room for competition
- All of this is predicated by having an open and transparent governance model

# The Eclipse Community is Unique



- Commercial-friendly
  - Most projects led by companies
  - Compelling model for shared industrial development
  - Commercial-friendly licensing model
- Focused on predictability
  - Have hit their dates four releases in a row
  - Open and transparent development and roadmap processes
  - Recognition that predictability spurs adoption
- Focused on quality
  - Explicit Eclipse quality objectives
  - Focus on frameworks and extensible, exemplary tools
- Clearly defined mission
  - *The Eclipse technology is a vendor-neutral, open development platform supplying frameworks and exemplary, extensible tools*

# Technology Matters: Secrets to Success



Lightweight Extensible Core

Eclipse-hosted projects

E.g. the Java  
Development Tool

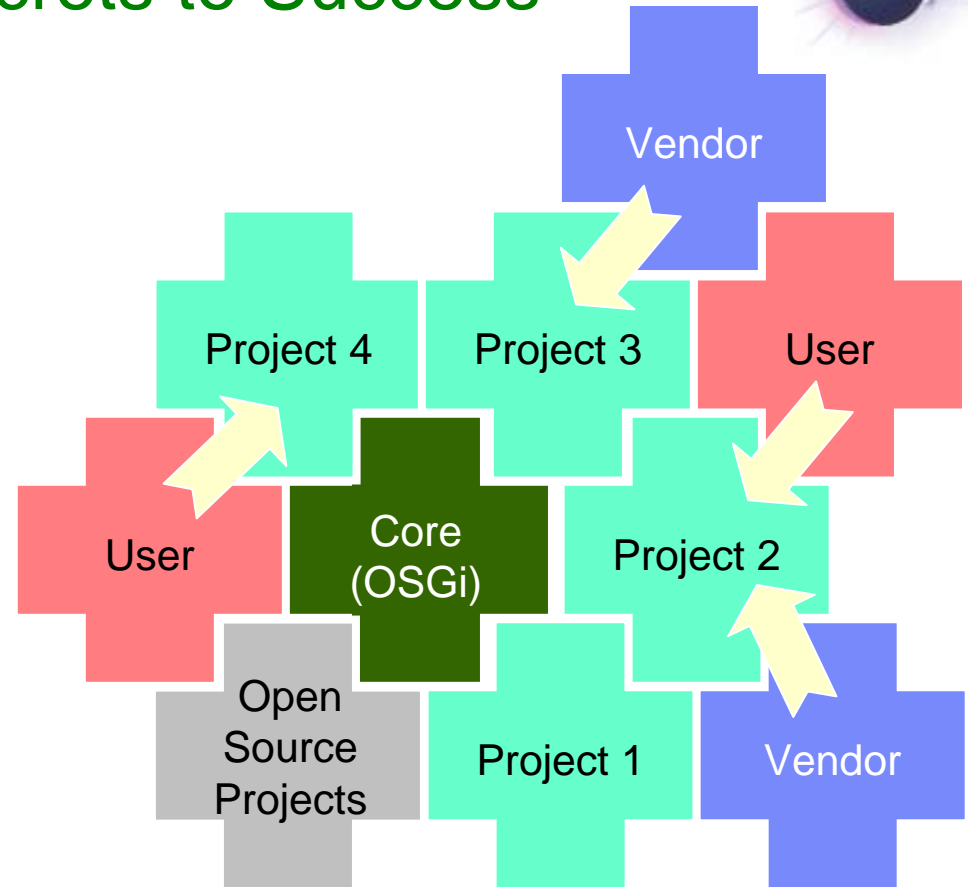
Lots of others

Related Open  
Source Projects

Vendor value-add

User value-add

Vendor and User value-add  
migrates into projects



# Process Matters: Is Open Source Chaotic?

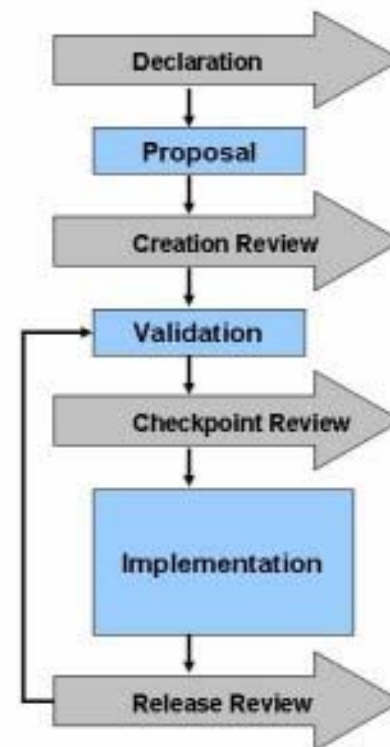


- OS projects are highly structured
  - Explicit rules (more than in most closed source projects)
  - Who may change the source code?
  - Who is responsible for delivering?
  - Who decides about the architecture?
  - ...
- Commit right rules: public "meritocracy"
  - Only a small number of developers can modify the source code: committers
  - Key architecture defined by a small team of lead developers
  - Peer pressure among committers – continuous reviewing
  - Continuous review and feedback by the community
  - Contributions from outside have to be reviewed by committers

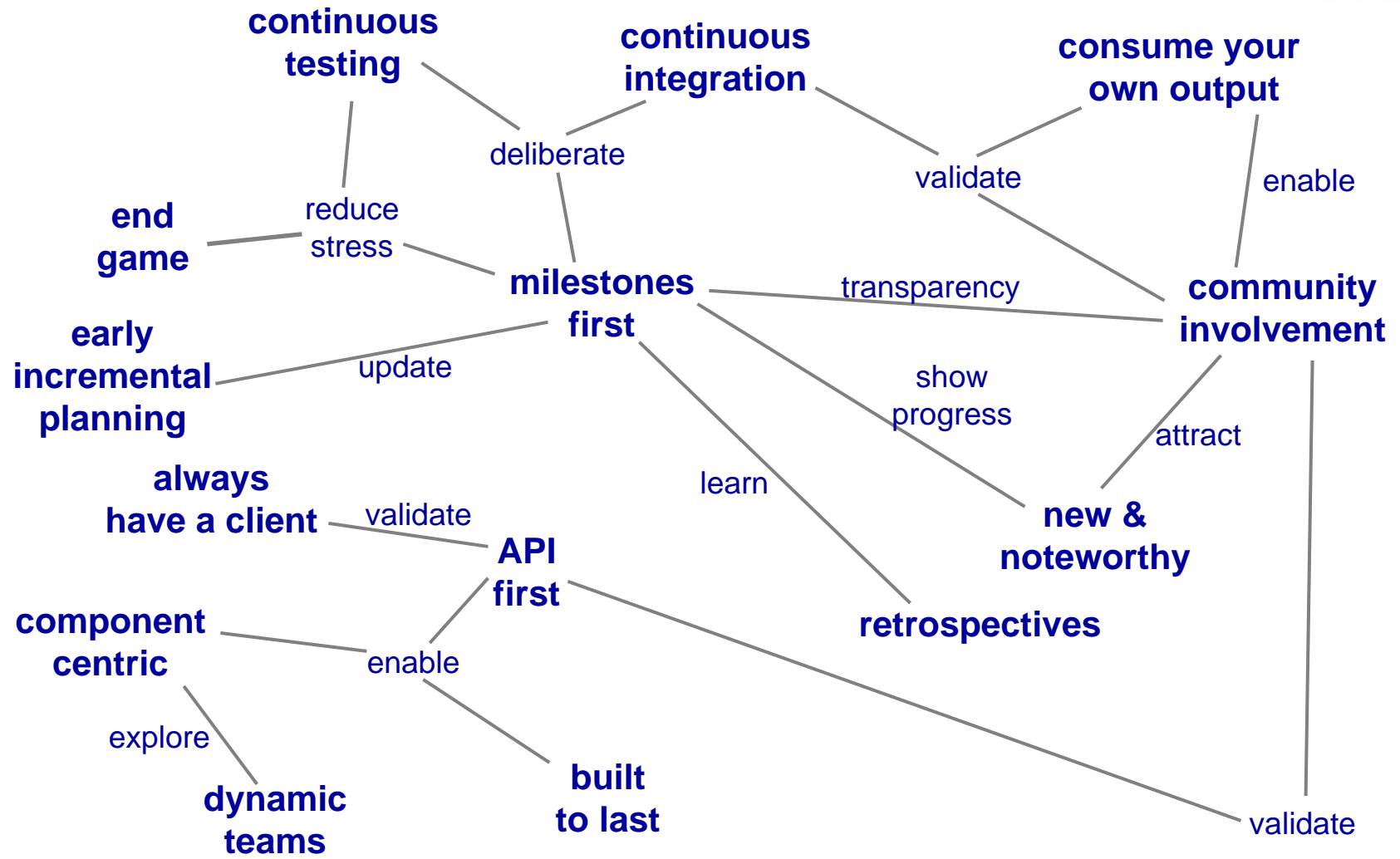
# Eclipse Project Lifecycle



- **Pre-proposal**
- **Proposal**
- **Validation/Incubation** – establish a fully-functioning open-source project
- **Implementation/Mature**
- **Archived** – after reaching the end of their natural lifecycle
- Official Reviews between each phase



# The Eclipse Way



# Community Involvement



- An active community is the major asset of an F/OSS project
- F/OSS project gives and takes:
  - F/OSS developer gives:
    - Listen to feedback and react
    - Demonstrate continuous progress
    - Transparent development
  - F/OSS developer takes:
    - Answer user questions so that developers do not have to do it
    - Report defects and feature requests
    - Validate technology by writing plug-ins
    - Submit patches and enhancements
- Give and take isn't always balanced
  - The community isn't shy and is often demanding



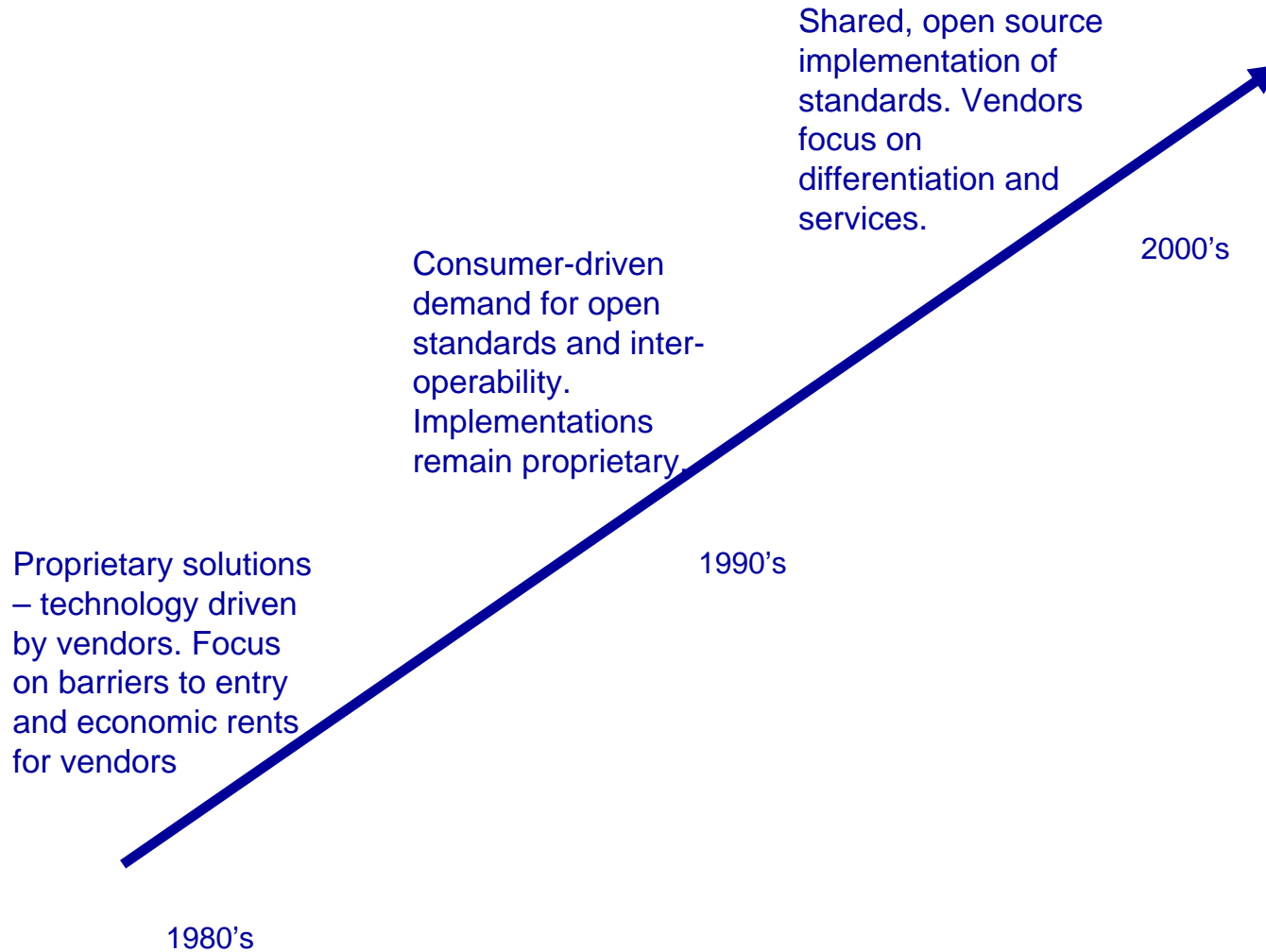
## Open Source and Open Standards

# Why Open Source?



- Open source allows multiple stakeholders to share the cost and benefits of a single shared implementation
  - Vendors, customers and standards bodies can all participate as equals
- Multiple groups can influence and participate in the architecture and in the support of a freely available framework
  - Consumers avoid proprietary lock-in
  - Vendors can invest in a shared platform
- Multiple benefits for adopters:
  - Open source allows for easier understanding of a framework
    - Trace issues through the actual source code
    - Don't have to rely on only incomplete documentation
  - Easy to workaround issues by modify source code and shipping it with your product.
  - Open source project continues to fix defects and enhance the software
  - No royalty for redistribution.

# Open Source: The Next Step





# The Business Model for Open Source and Open Standards

- Shared implementations of standards:
  - Save time to market
  - Increase rate of standards adoption
  - Reduce risk
  - Provide thought leadership and first mover advantages to supporting vendors
  
- Yes, open source means shareholder value!
  
- Vendors who share open source implementations still compete
  - Product differentiating features
  - Service, support
  - Branding, routes to market



## Eclipse and OSGi

# Eclipse Uses OSGi



- Eclipse is based on an implementation of the OSGi framework R4.0 specification
- The concept of an Eclipse plug-in is synonymous with an OSGi bundle
- Eclipse today has the best tooling environment for building and deploying OSGi-based applications
  - GUI tool for manifest editing
  - Manifest file validation

# OSGi Uses Eclipse



- Eclipse 3.1 is an immediately available implementation of the OSGi R4.0 framework specification
- Based on Eclipse 3.0 experience, Eclipse team brought many requirements to the OSGi for consideration during R4.0 development
- Eclipse has placed OSGi implementations onto the machines of millions of Java developers

# Future Steps



- The Eclipse Equinox project has “graduated” to become part of the Eclipse Platform top-level project
  - Bring more visibility to the OSGi community at Eclipse
- The Embedded Rich Client (eRCP) project is aimed at bringing Eclipse runtime technology to mobile and constrained devices
- Continued partnership with the OSGi Alliance for the benefit of both organizations



Thank you!