eclipseCON™ 2008

Using ECF for Lightweight Distributed Team Collaboration

Scott Lewis
Committer and ECF Project Lead

http://www.eclipse.org/ecf

Team Communication for Open Source Projects

- How are OS Projects Different?
  - Membership
    - Distributed
    - Multiple Groups
    - Diverse Skills and Backgrounds
    - Diverse Cultures
  - Project Organization
    - Flat
    - Self-driven/Voluntary
    - Project Leadership Has Less Control Over Team
  - Community Expectations
    - Open Planning Required
    - Responsiveness to Community Contributions
      - Bug reports, patches, documentation, etc.

ECF: Communication/Collaboration for Open Projects

- Interoperability
  - People use Different Communication Tools
  - Multiple Teams -> Multiple Protocols
- Integrate
  - UI
  - Other Tools – Workbench, Mylyn, Editors, Debuggers, Reporting, RCP apps, other tooling, etc.
- Extensible
  - Team-specific and/or Proprietary Tools
  - Support for specific processes/approaches/conventions

Goal: Lower Barriers to Team and Community Communication

to

Increase Productivity

Increase Trust

Increase Community Participation

with

ECF: Interoperable, Integrated, Extensible

Diversity Trumps Ability for Team Productivity

The Difference: How the Power of Diversity Creates Better Groups, Firms, Schools, and Societies

- Conditions
  - High-performing individuals
  - Difficult problems
- Good Diversity
  - Cognitive Diversity: Think Different(tm)
  - Leads to More/more Varied 'toolboxes'
  - Avoid 'Local Maxima' in Solution Searches
- Bad Diversity
  - 'Fundamental preferences'
  - e.g. 'What are the project's goals'?

Exploit Diversity

- Make it Easy for 'Outsiders' to Communicate/Contribute
  - Client Interoperability
- Communicate Publicly with Community
  - IRC, IM/Conference Calls, ECF Collaboration Groups, etc.
  - Solicit Contributions
    - Bugzilla/BugDay/IRC, etc.
- Expose Unfinished Work/Problems
  - Milestones, Dev Mailing List, Wiki, Bugzilla ALL GOOD
  - Also IM/Chat
  - Opportunities for External Contributions

Increasing Integration

- Real-Time Communication Integrates Well With
  - Workbench (sharing resources)
  - Bug Tracking (Mylyn sharing tasks)
  - Conferencing
    - Shared Editing

Demos

- IM/Chat
  - Multi-Protocol, public services, sharing info as well as direct communication
- Real-Time Collaboration
  - Integrate communication into tooling
  - Provide common UI
- Shared Editing
  - Use Case: Code Review and others

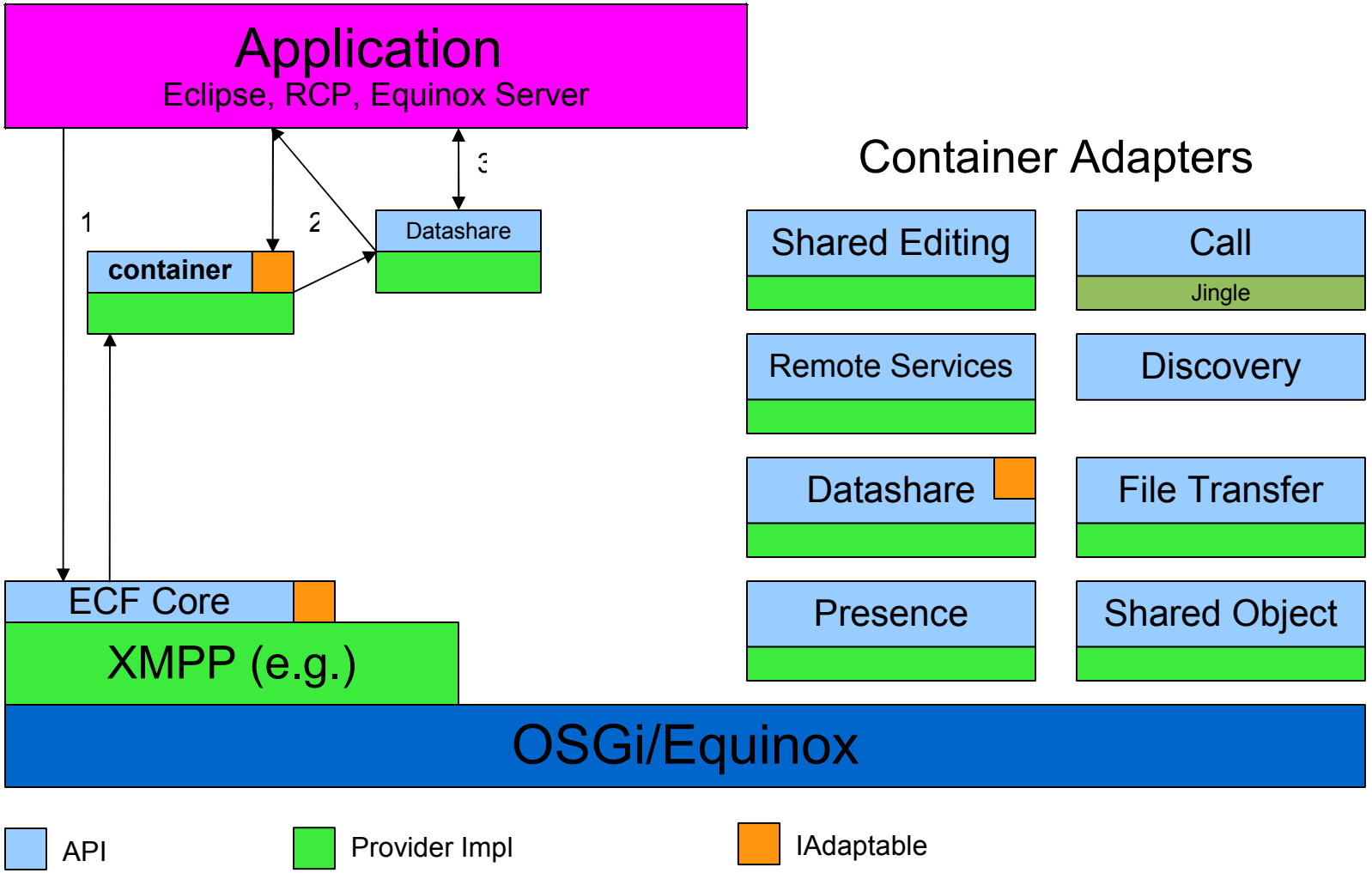
Can Absence Make a Team Grow Stronger?

- Answer: yes
  - Rule 1: Exploit Diversity
  - Rule 2: Use Technology to Simulate Reality
    - Virtual Spaces for Team and Community
  - Rule 3: Hold Team Together: Build Trust
- See Bjorn's Blog Entries

Extensibility

- APIs through Adapters
  - Core: 2 bundles ~120k
  - API plugins/bundles – aka Container Adapters
- Providers Implement: core + 0 or more Adapters
- Extensibility: Exploits Diversity In Community
  - Shallow or Deep Additions
  - Sense of Ownership

ECF Provider Architecture



In Code

- `container = ContainerFactory.getDefault().createContainer();`
- `ds = (IChannelContainerAdapter) container.getAdapter(IChannelContainerAdapter.class);`
- `IChannel channel = ds.createChannel(...)`

API: Dynamic Service Discovery

- org.eclipse.ecf.discovery
  - IDiscoveryContainerAdapter
    - Adapter from IContainer
  - Service Discovery API
    - Properties for existing service types: ITunes, http, etc.
    - Register new types: updatesite, remotesvcs, etc.
  - Two Providers So Far
    - Bonjour
    - SLP (RFC 2608)

Demo

- Dynamic Service Discovery
  - Update site service
  - Other (non-Eclipse services)

API: Asynchronous File Transfer

- org.eclipse.ecf.filetransfer
  - IRetrieveFileTransferContainerAdapter
    - Adapter from IContainer
  - API
    - sendRetrieveRequest
    - Asynchronous notifications to provided listener
      - Start,Data,Done
  - Several Providers
    - URLConnection (JRE), Apache httpclient 3.0.1, SCP/SSH/JCraft, Eclipse File System
    - Uses Jobs API
  - Also have send and browse API
- Being used for p2

Demo

- File transfer
  - Multi-protocol
  - Embeddable into other plugins (e.g. P2)

API: Remote Services

- org.eclipse.ecf.remoteservice
  - IRemoteServiceContainerAdapter
    - Adapter from IContainer
  - API
    - Looks very much like **OSGi Services**
    - **Clients have choice**
      - **Proxy (transparent)**
      - **IRemoteService (explicit)**
        - Asynchronous (Listener) Invocation
        - Futures
  - Providers
    - R-OSGi, JMS ActiveMQ/BEA, ECF generic, JavaGroups, Riena
    - Others (?)

Demo

- Remote Services
  - Transparent and not-transparent for different use cases
  - Multi-protocol
  - Easy to add own service types/discover, access

Summary

- ECF for Open Project Communications
  - Build Diversity, Trust, and Community
  - Through Technology
    - Interoperability
    - Integration
    - Extensibility
  - Project
    - http://www.eclipse.org/ecf
  - Wiki
    - http://wiki.eclipse.org/Eclipse_Communication_Framework_Project
  - IRC
    - irc://irc.freenode.net/eclipse-ecf
  - XMPP server for Eclipse Projects
    - https://bugs.eclipse.org/bugs/show_bug.cgi?id=126089