eclipseCON™ 2008

Using ECF for Lightweight Distributed Team Collaboration

http://www.eclipse.org/ecf

Scott Lewis
Committer and ECF Project Lead

3/5/2008

Team Communication for Open Source Projects

- How are OS Projects Different?
  - Membership
    - Distributed
    - Each Person in Multiple Groups
      - Within and cross-organizational
    - Frequent Resource Changes
    - Diverse Skills and Backgrounds
    - Diverse Cultures
  - Project Organization
    - Flat
    - Self-driven/Voluntary
    - Project Leadership Has Less Control Over Team
    - Communication/Collaboration Within and Across Teams
  - Community Expectations
    - Open Planning Required
    - Responsiveness to Community Contributions
      - Bug reports, patches, documentation, etc.

ECF: Communication/Collaboration for Open Projects

- Multi-Protocol/Interoperability
  - People use Different Communication Tools
    - IRC, Yahoo, MSM, Google Talk, etc.
  - Each member in multiple groups
- Integrate/Simplify
  - With UI - Don't want 10 IM User Interfaces for 5 Teams
  - With Other Tools – Workbench, Mylyn, Editors, Debuggers, Reporting, etc.
- Extensible
  - Integrate with Team-specific and/or Proprietary Tools
  - Add support for specific processes/approaches/conventions

Project Themes: Interoperable, Integrated, Extensible

Goal: Lower Barriers to Team and Community Communication

to

Increase Productivity

Increase Trust

Increase Community Participation

Increasing Productivity

- *The Difference by Scott Page*
  - **Diversity Trumps Ability**
    - Pre-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'?

Leverage Diversity with Communication

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

Demos

- IM/Chat
  - Multi-Protocol, public services, sharing info as well as direct communication

Increasing Integration

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

Demos

- Real-Time Collaboration
  - Integrate communication into tooling
  - Provide common UI
- Shared Editing
  - Use Case: Code Review and others

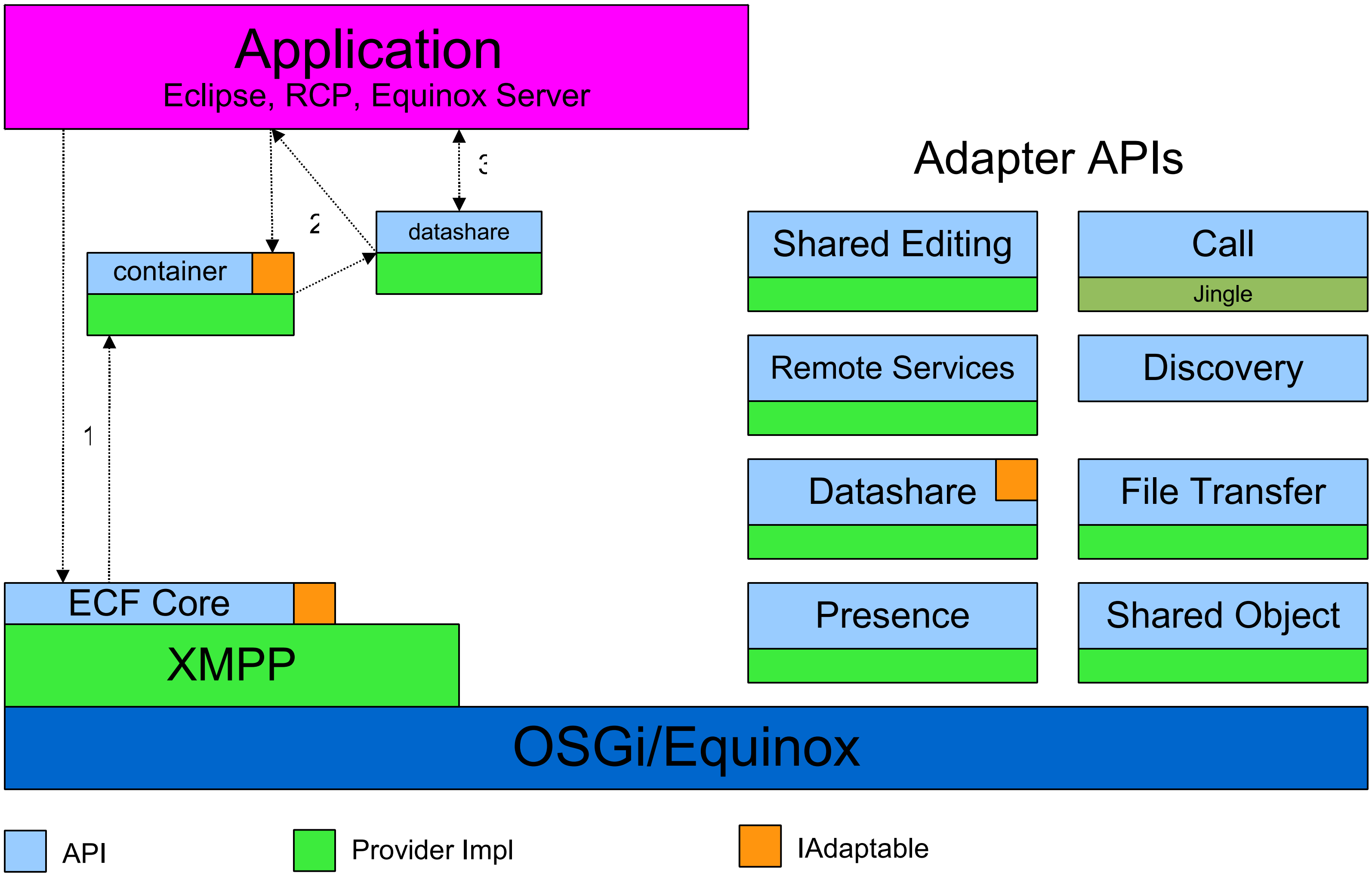
Can Absence Make a Team Grow Stronger? *Majchrzak, et. al.*

- 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 or paper

Framework-Provided Extensibility

- APIs through Adapters
  - Core: Very small
    - Identity, Distributed container - IContainer
  - API plugins/bundles – aka Container Adapters
    - Discovery, File Transfer, Remote Services, Datashare/Channels, Shared Objects, Call Setup, Etc. (by you OR me)
- Providers Implement: core API + 0 or more Adapters
- Extensibility: Also Allows Exploiting Diversity In Community
  - Deep additions by others
  - Sense of community ownership

ECF Provider Architecture



Application

Eclipse, RCP, Equinox Server

container

datashare

ECF Core

XMPP

OSGi/Equinox

Adapter APIs

Shared Editing

Call

Jingle

Remote Services

Discovery

Datashare

File Transfer

Presence

Shared Object

🟦 API

🟩 Provider Impl

🟧 IAdaptable

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
    - Get properties for existing service types: ITunes, http, etc.
    - Register new types for equinox-based services: updatesite, remotesvcs, etc.
  - Two Providers So Far...
    - Zeroconf/Rendevous (Apple iTunes)
    - 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