ECF Inside and Out

Marcelo Mayworm <mayworm@ieee.org>
Scott Lewis <slewis@eclipsesource.com>

November 19th, 2008

Docshare and Real Time Shared Editing

Demo

http://live.eclipse.org

Docshare and Real Time Shared Editing

- Implements real-time shared editing
- Connect via ECF Datashare API
- Supported provider: XMPP/Google Talk, Skype, ECF generic, JMS, Javagroups
- Generalization / Sync API

Sync API

- Synchronization service for document model
- Synchronization factory can provide different types of synchronization
- Document synchronization implemented with Cola – default implementation
- IDocumentChange is applied to local document change and to a replicated document
- ECF DocShare for share editing
  - Uses Sync API
  - Over Datashare API

Sync API

- Extensible for other Eclipse Editors
- Accessible through OSGI service
- ECF 2.1
- Beyond document model
- More information about Cola algorithm
  - http://wiki.eclipse.org/index.php/RT_Shared_Editing

Sync API

- Start State: Two or more peers have replica of single document (String)

- Client 1 Change (keyboard input)
  - synchronizationStrategy.registerLocalChange(IModelChange)
  - sendMessage A (via some transport) about operation

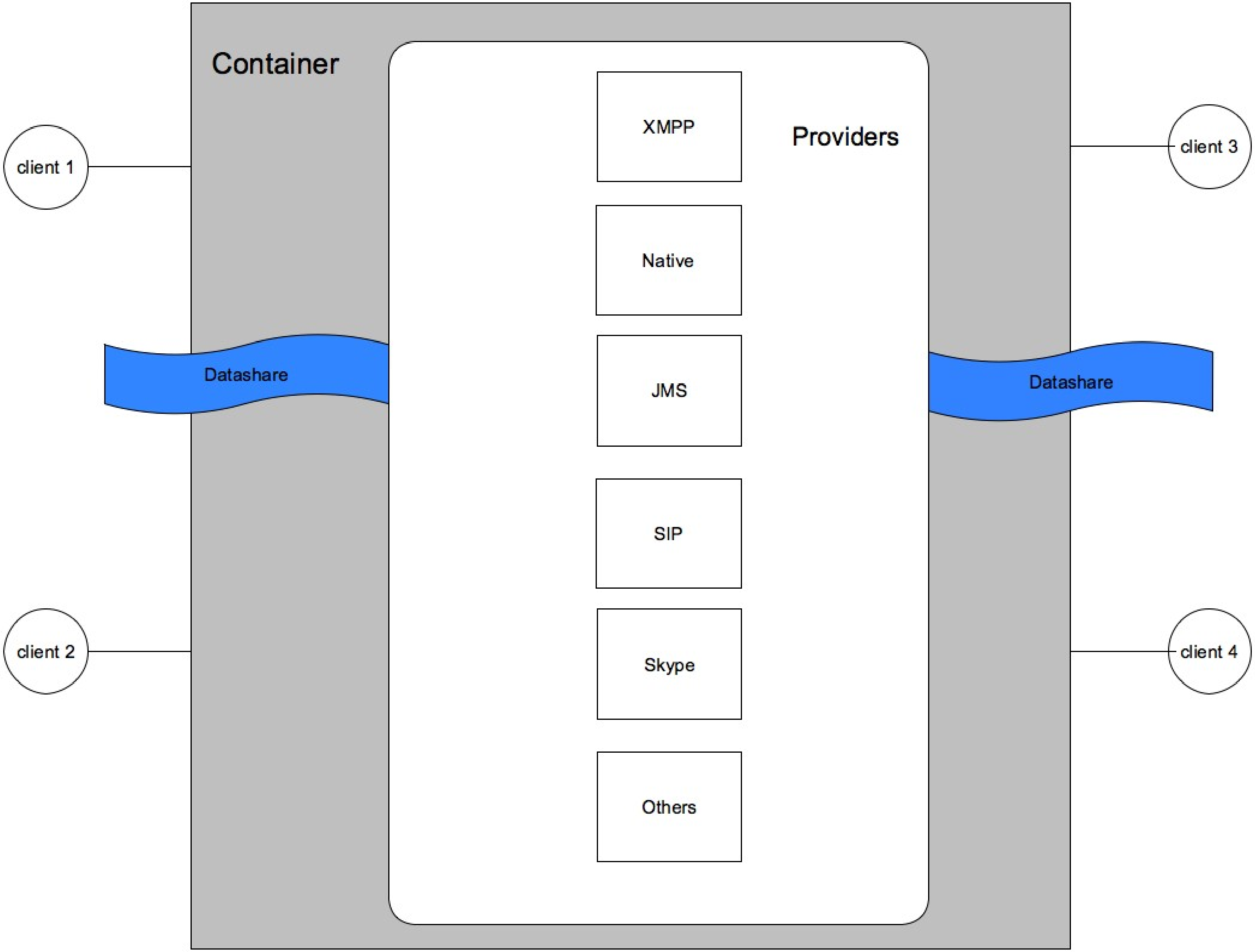
- Client 2
  - Receive Message A
  - synchronizationStrategy.tranformRemoteChange(A)

Sync API

- Underneath registerLocalChange and transformRemoteChange is Cola/Operational Transformation (Sun and Ellis)
  - Operations are transformed on both/all clients to get back to consistent state
- Synchronization strategy is operation specific (e.g. ins/del character)
- Approach allows other types of operations

Datashare

- Provides messaging channels for asynchronously sending messages to remote processes
  - XMPP
    - Clients
    - Groups
- This API also runs on XMPP as well as other transports
- Sender Ordering (FIFO). Receivers guaranteed to get messages in same order they were sent
  - No stronger ordering guarantees


Container
XMPP
Providers
Native
JMS
SIP
Skype
Others
Datashare
Datashare
client 1
client 2
client 3
client 4

DataShare: Example Code

```
// Create container instance via named provider
IChannelContainerAdapter channelContainer = getChannelContainer.....

//Create a channel linked to a listener
channelContainer.createChannel(<ChannelStringID>, <ChannelListener>, null);

//Get the channel
IChannel sender = channelContainer.getChannel(<ChannelStringID>);

//Channel used to asynchronously send arbitrary data (byte arrays)
sender.sendMessage(new String("hello").getBytes());
```

Demo

- Collab
  - Chat
  - URL Sharing
  - Remote view opening
- IPhone discovery services – Take two? :-)

Ongoing Efforts

- Model Synchronization
- Real-Time Collaboration for IDE
- ECF for Runtimes
- Remote OSGi Services – RFC 119
- IM/Chat Features
- Application Integration
- Packaging and Deployment

Conclusion

- Support from the community **welcomed** and **appreciated**!
    - Ways to help
        - Improve / Extend existing applications
        - Build provider implementations
        - Contribute to API Design (we need feedback!)
        - Join ECF Community! (commercial or not!)

Q&A

Resources

- Website
  - http://www.eclipse.org/ecf
- Mailing list
  - http://dev.eclipse.org/newslists/news.eclipse.technology.ecf/maillist.
- Newsgroup
  - news://news.eclipse.org/eclipse.technology.ecf
- Project Plan For ECF - version 3.0
  - http://www.eclipse.org/projects/project-plan.php?projectid=rt.ecf
- Wiki
  - http://wiki.eclipse.org/Eclipse_Communication_Framework_Project
- Other ECF repository
  - http://ecf1.osuosl.org/

Thank You!

http://www.eclipse.org/ecf