

Inkscape-Jabber program code interface:

Last updated: February 9, 2005

Jonas Collaros

Inkscape side functionality:

`void connected()`

This notification function is called by the Jabber side when this user has successfully been logged into Jabber via `connect_and_auth()`. After being connected, Inkscape side should enable options for document sharing with users and chatrooms.

`void error_connection(char* error_message)`

This notification function is called when the Jabber side has encountered an error that has resulted in the termination or failure of the TCP connection (TCP fatal error). Prompt the user to reconnect via connect dialog (`connect_and_auth` dialog) and display the given error information.

Parameters:

`char* error_message`

Error information to be displayed to the user (e.g. "Authentication failed. Check server name, user name and password.").

`void error_timeout(char* error_message)`

This notification function is called when the Jabber side has encountered an error that has resulted in a possible loss of document synchronization. This user is no longer "connected" to the document, although the user is still connected to the Jabber server. Prompt the user for confirmation to reconnect to the document as a new user (`send_connect_request_to_user/chatroom` dialog). If the user opts to reconnect, conflict resolution handshaking will commence.

NOTE: It is not necessary to re-`connect_and_auth` after this error.

Parameters:

`char* error_message`

Error information to be displayed to the user (e.g. "Document synchronization error - Do you want to reconnect?").

`void receive_buddy_list(char** user_root_JIDs)`

This listener function is called after the Jabber side has retrieved the user's buddy list (per request from `trigger_recieve_buddy_list()`). The action to be taken by the Inkscape side is to display this list to the user. Note that this list is of all of the user's online buddies, whether or not they are currently connected to any of the user's whiteboard sessions.

Parameters:

`char** user_root_JIDs`

An list of JIDs buddies that are currently online.

`void receive_change(char* change_message)`

This listener function for incoming change messages is notified when another user connected to this document does a `send_change()`. Changes sent to the root JID (without an inkboard resource) will be ignored.

Parameters:

char* change_message

Incoming change message sent by another Inkscape user connected to the current document. Parse and implement this change.

```
void receive_connect_request (char* sender_root_JID, char*
document_description)
```

This listener function is called by the Jabber side when it receives a one-to-one document sharing invitation from another user. The action needed to be taken by the Inkscape side will be to display a dialog for the user to either accept or decline the invitation. If accepted, the user should either be prompted to save the workspace (the workspace will be cleared when the document arrives) or a new instance of inkscape should open, and the user should wait for a document transmission via `receive_doc()`. Inkscape side must call `send_connect_request()` with the user's chosen response.

NOTE: Inkscape side needs not be notified when a connect request comes in a chatroom setting. Jabber will handle this internally.

Parameters:

char* sender_root_JID

JID of the user sending the whiteboard invitation (resource is hidden)

char* document_description

The description of the document that is being shared (provided by user)

```
void receive_connect_request_response(char* response_message, bool
accepted_request)
```

This listener function is called by the Jabber side when a user answers a one-to-one document sharing invitation. The response will either indicate that the user has accepted the invitation and is ready to receive the document, or that the user has declined the invitation (or that there is no response). The action needed to be taken by the Inkscape side will be to display the response message to the user. If accepted Inkscape-side will also add listeners to the screen objects to broadcast changes and call `send_doc()` (don't wait for the trigger) and continue to wait for a `user_connected_to_whiteboard()` signal. If declined, reactivate 1-1 whiteboard sharing request options.

Parameters:

char* response_message

Response information to be displayed to the user (e.g. "Whiteboard session refused.").

bool accepted_request

True if the invitation was accepted, false otherwise.

```
void receive_doc (char* document_message)
```

This listener function for incoming document messages is notified after another user provides the user a document with `send_doc()`. After this function is called, this user can be considered as "connected" to the document. When the user chooses to receive a document, Inkscape will make all changes to a clean document (either a new instance of Inkscape or through the user-authorized deletion of the objects on the current instance). This function forces the user to wait (making no changes) while the document is loading.

(NOTE to Jabber side: when calling this function, also send a `user_connected` message to the other user)

Parameters:

`char*` `document_message`
Full XML document representation message.

(future function?)

`void receive_document_list(char** document_names)`

This listener function is called after the Jabber side has retrieved the room document list (per request from `trigger_recieve_document_list()`). The action to be taken by the Inkscape side is to display this list to the user.

Parameters:

`char**` `document_names`
A list of names of documents available in this chatroom

`void receive_user_list(char** user_root_JIDs)`

This listener function is called after the Jabber side has retrieved the room user list (per request from `trigger_recieve_user_list()`). The action to be taken by the Inkscape side is to display this list to the user.

Parameters:

`char**` `user_root_JIDs`
An list of JIDs of users connected to the whiteboard session (resource is hidden)

`void trigger_send_doc()`

This request function is called by the Jabber side when another user requests the whole document (due to initial connect or timeout error). The action to be taken by the Inkscape side is to call the Jabber side function `send_doc()`, providing it the current XML representation of the document to share.

`void user_connected_to_whiteboard(char* joined_root_JID)`

This notification function is called by the Jabber side when another user has received the initial send of a document in a one-to-one or chatroom sharing session. The other user is now “connected” to the document, and the Inkscape side can notify other users that a user has joined. This provides independent notice of an addition of a user. The full buddy list will only be shown on specific request.

Parameters:

`char*` `joined_root_JID`
JID of the user joining the whiteboard session (resource is hidden)

`void user_disconnected_from_whiteboard(char* dropped_root_JID)`

This notification function is called by the Jabber side when a user leaves a one-to-one or chatroom sharing session, voluntarily or via error. The other user has now “disconnected” from the document (though not necessarily logged out from the chatroom or server), and the Inkscape side can notify other users that a user has dropped. This provides independent notice of the drop of a user. The full buddy list will only be shown on specific request.

Parameters:

`char* new_root_JID`
JID of the user leaving the whiteboard session (resource is hidden)

Jabber side functionality:

`void connect_and_auth(char* server, char* username, char* pw)`

This request function logs this user in to the specified server as the specified username with the specified password at resource Inkboard. If there are problems connecting, calling this method may result in a call to `error_connect()`. While connecting, the ability to make a new connection is locked (to prevent multiple connections). The user only needs to connect to the server once, each new document connected here out will have a unique resource associated with this JID. NOTE: The resulting full JID connected will be `<username>@<server>/Inkboard`

Parameters:

`char* server`
Jabber server (e.g. “jabber.anywise.com”)
`char* username`
Jabber username, JID node (e.g. “inkSender”)
`char* pw`
Jabber password

`void disconnect_from_document()`

This request function is called by the Inkscape side when the user wants to disconnect from the currently shared document. Jabber will disconnect the user and Inkscape should prompt the user to save.

`void disconnect_from_server()`

This request function is called by the Inkscape side when the user wants to log off of the Jabber server, either voluntarily or via quitting the program. If there are whiteboards open at the time the user indicated their intent to log out, Inkscape side should first warn the user that logging out before closing all open whiteboards will cause them to timeout and disconnect.

`void send_change(char* change_message, bool is_repeatable)`

This request function is called by the Inkscape side when it has a change message to send other users connected to this document. Changes should be sent to the JID with the appropriate inkboard resource: `(username@server/resource/inkboard#)`.

Parameters:

`char* change_message`
Outgoing change message.
`char* is_repeatable`
This value designates if this change may or may not be executed multiple times (in the case of out-of-order message handling). Examples of changes that shouldn’t be repeated are create and delete changes.

`void send_connect_request_response(bool accepted_request)`

This listener function is called by the Inkscape side to send a response to a one-to-one document sharing invitation. The response will either indicate that the user

has accepted the invitation and is ready to receive the document, or that the user has declined the invitation.

Parameters:

bool accepted_request
True if the invitation was accepted, false otherwise

void send_connect_request_to_chatroom (const char* chatroom_root_JID)
This request function will connect the user to the specified chatroom JID and wait for document reception. After calling this function, this user should wait for the document to arrive via `receive_doc()` before making any changes to the workspace (the workspace will be cleared when the document arrives).

Parameters:

const char* chatroom_root_JID
JID of the chatroom to connect to (resources are handled internally and any provided resource will be ignored)

void send_connect_request_to_user (const char* recipient_root_JID,
char* document_name)

This request function connects the user with the specified single user JID. After calling this function, this user should wait for a `connect_request_response()`. In addition, Jabber side will call `trigger_send_doc()` if and when the target user is ready to receive the document. When the other user has been connected successfully, `user_connected()` will be called, and then the session will be established.

Parameters:

const char* recipient_root_JID
Root JID of the single user to share to document with (e.g. node@domain – resources are handled internally and any provided resource will be ignored)
char* document_description
The description of the document that is being shared (provided by user)

void send_doc(char* document_message)

This request function is called by the Inkscape side to send the entire current XML representation of the document to another user connected to this document, as well as that user's queue of any changes not yet implemented. Inkscape side should use always use this function if requested to do so by `trigger_send_doc()`, and also in `receive_connect_request_response()`.

Parameters:

char* document_message
Full XML document representation message.

(future function?)

void share_doc_in_chat()

This request function is called by the Inkscape side, after a user connects to a chatroom, when the user wants to make the current document loaded in the user's workspace available to other users connected to this chatroom (to "upload" it to the chat). This should only be allowed by users with documents not currently shared in the current chatroom.

(future function?)

```
void trigger_receive_document_list()
```

This request function is called by the Inkscape side, after a user connects to a chatroom, when the user requests to see the list of documents currently hosted in this chatroom. The Jabber side will retrieve the list and then send it to

```
recieve_document_list()
```

```
void trigger_receive_buddy_list()
```

This request function is called by the Inkscape side when the user requests to see the user's buddy list. The Jabber side will retrieve the list and then send it to

```
recieve_buddy_list().
```

```
void trigger_receive_user_list()
```

This request function is called by the Inkscape side when the user requests to see the list of users connected to this document. The Jabber side will retrieve the list and then send it to `recieve_user_list()`.

Examples:

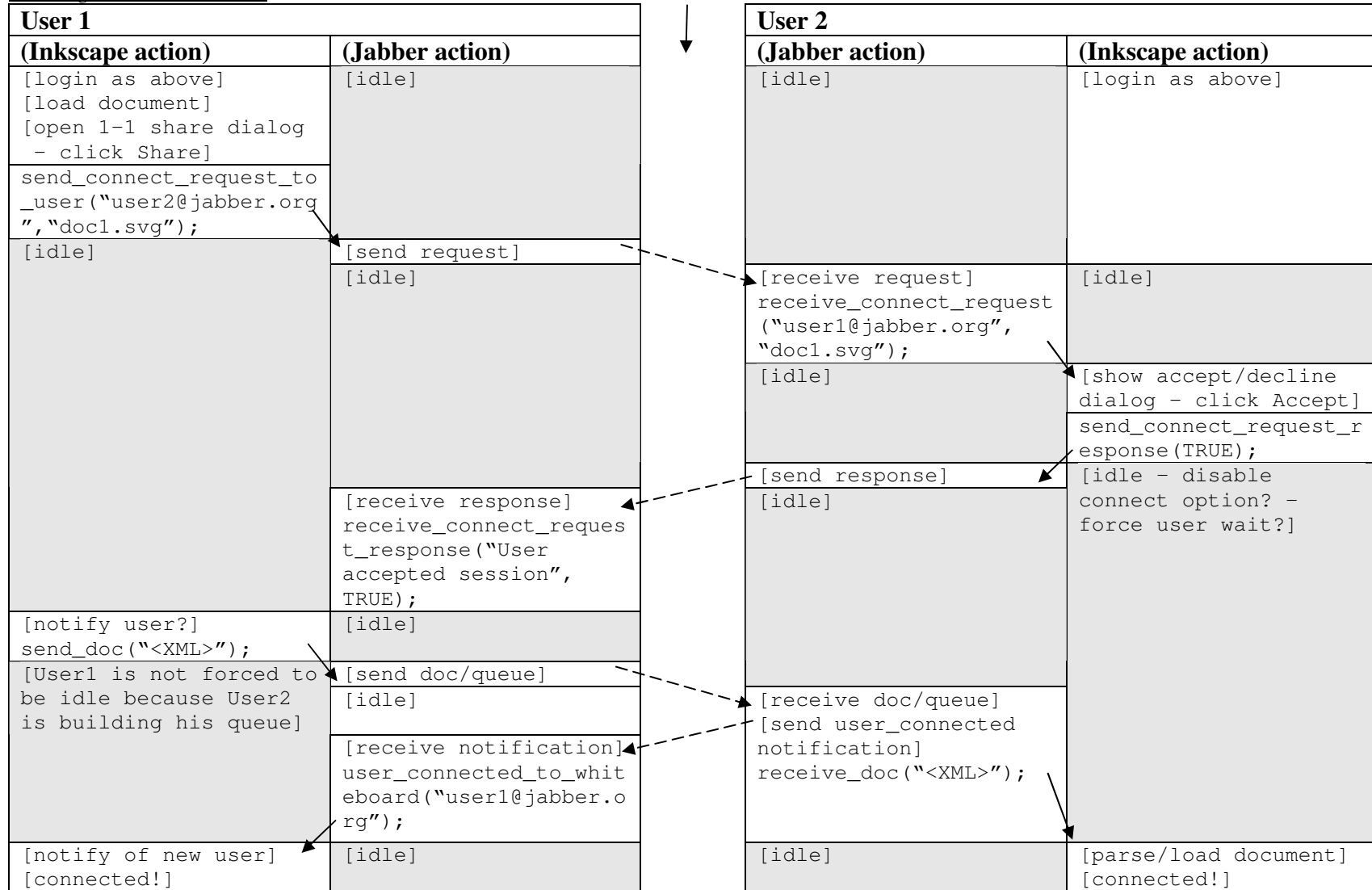
Note that “Inkscape side” and “Jabber side” columns do not indicate ownership of the functions called, but instead the use of those functions. In fact, Jabber side will only call Inkscape side owned functions, and vice-versa.

Logging on to a jabber server:

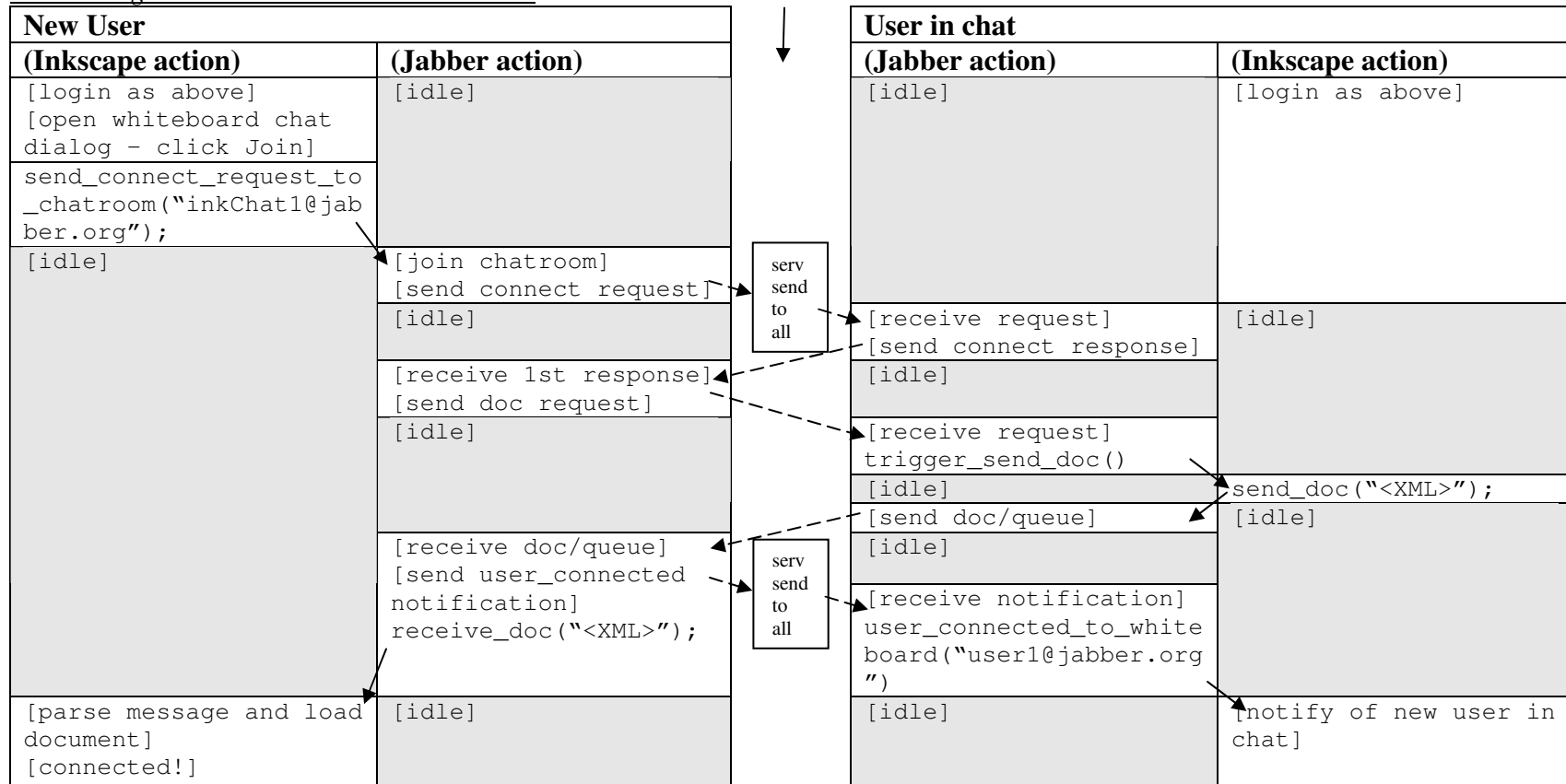
User 1	
(Inkscape side action)	(Jabber side action)
[launch]	[idle]
[login dialog - click Connect]	
connect_and_auth (jabber.org,user1,pass);	
[idle]	[connects to jabber server and authenticates]
	connected();
[enable doc sharing]	[idle]

Sharing a document 1-1:

NETWORK

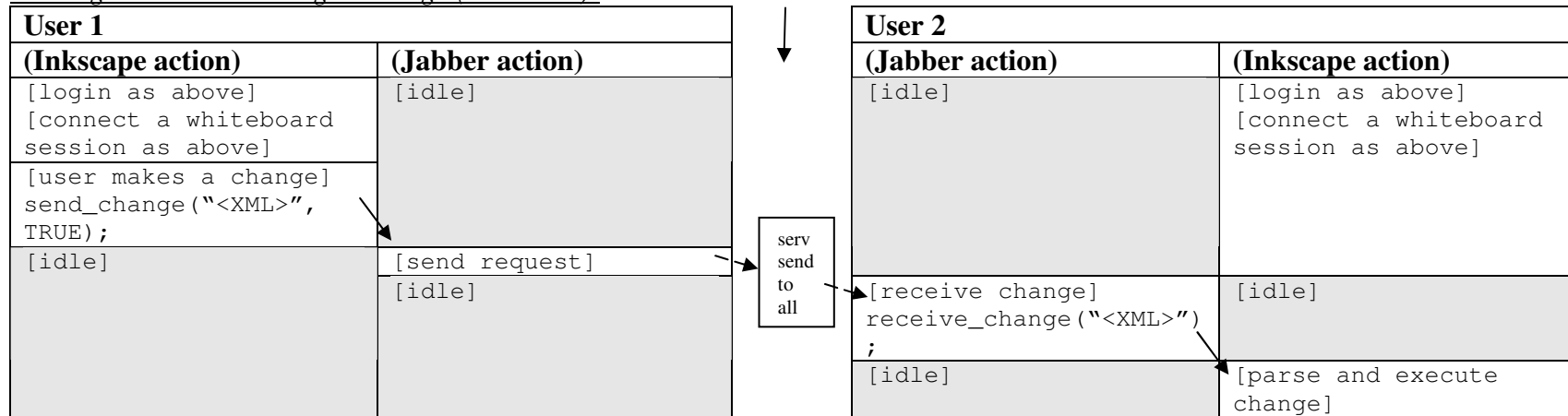


Connecting to a chatroom hosted document:



Note that a new user will receive a blank document if no document is currently available.

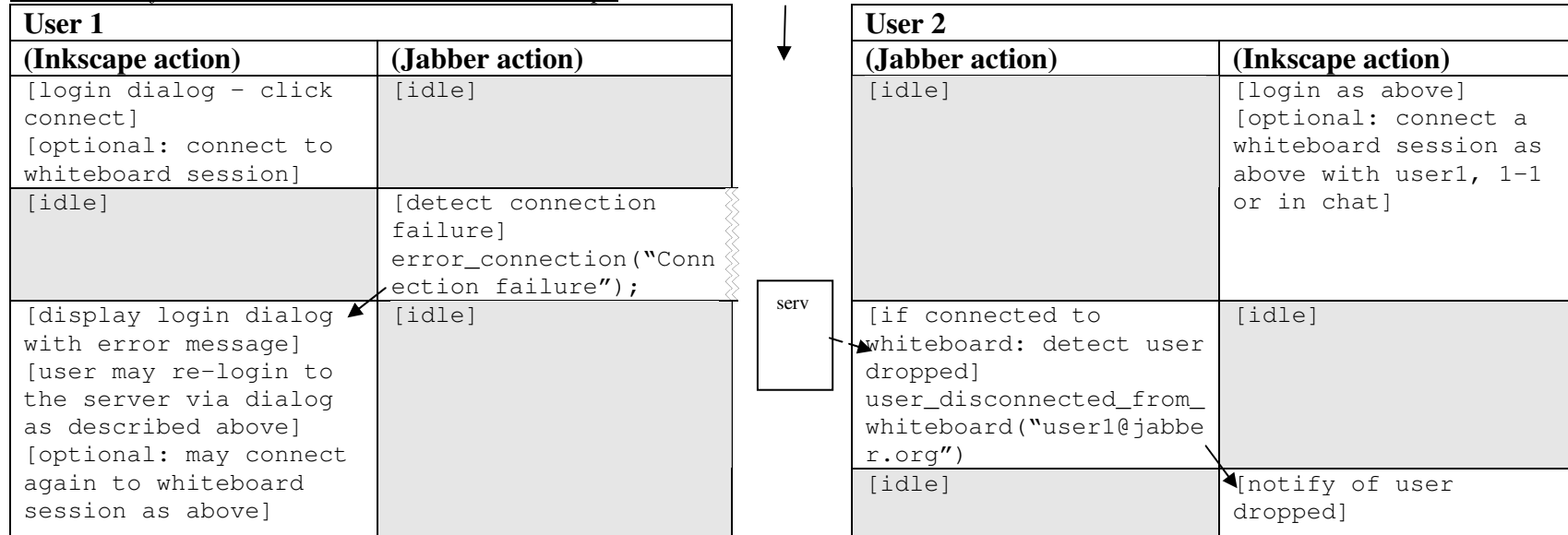
Sending a document change message (chatroom):



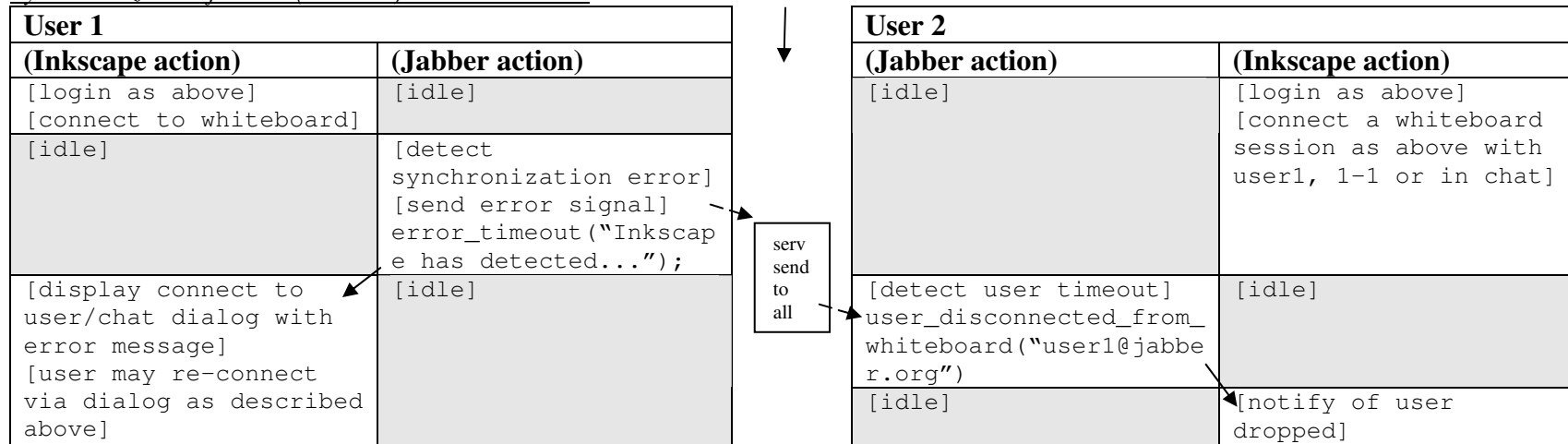
Note that the process of sending change messages in a 1-1 whiteboard session is functionally equivalent within the interface. The only difference in the code flow diagram would be a direct connection to user2 without the intermediary server broadcasting change messages.

Connection failure and reconnect/detect user drop:

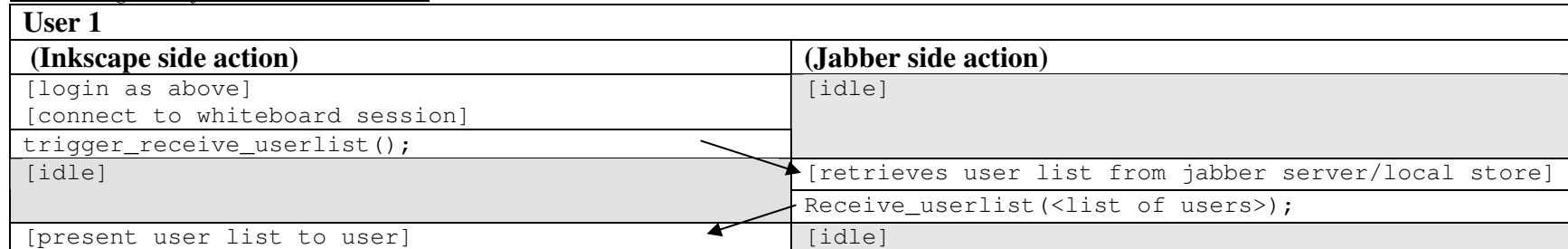
NETWORK



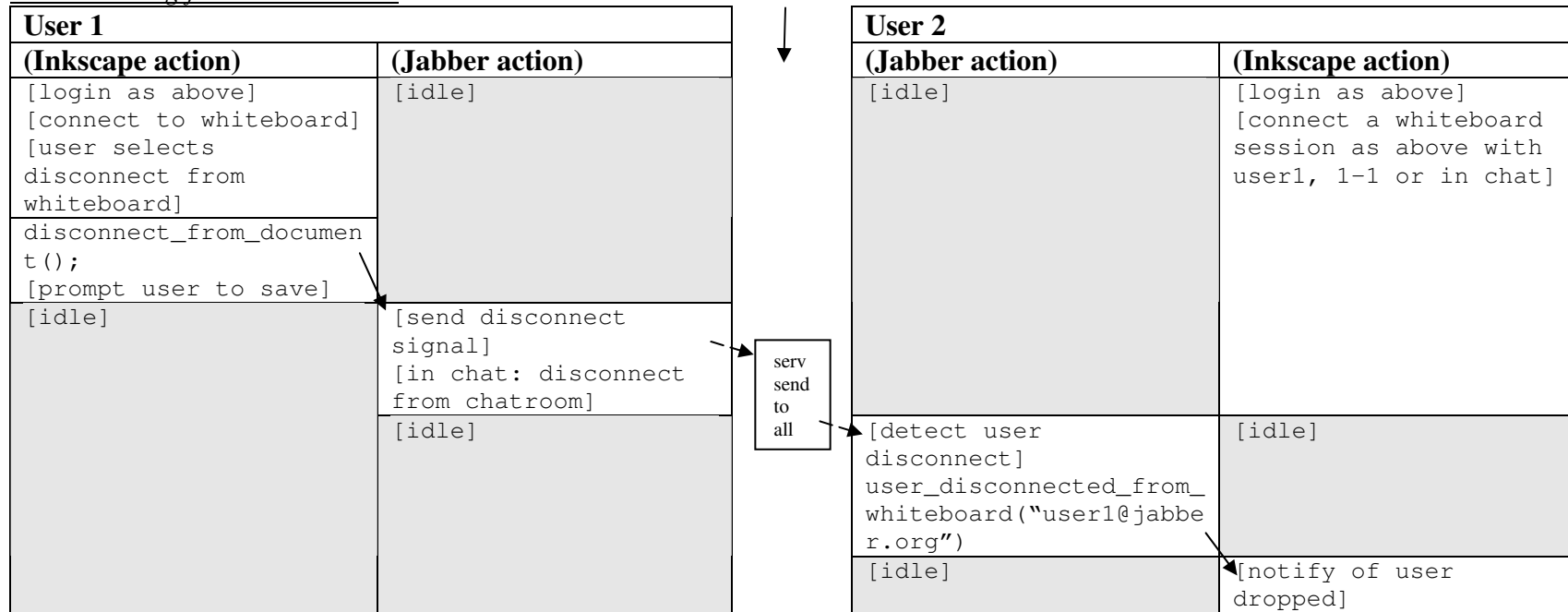
Synchronization failure (timeout) and reconnect:



Retrieving list of users in chatroom:



Disconnecting from a document:



Disconnecting from server:

