Ipo4ta specification

Version 0,6

Slava Stefanov (@ipo4ta.com/admin)

Overview

Ipo4ta (eepochta) = Internet Post.
Ipo4ta is a new Internet post system that combines email and Web site technologies. It overcomes limitations in the SMTP protocol by removing the need to transfer the message itself. Instead, messages stay on the sender's server until the recipient picks it up. This means that the recipient only needs to be sent a notification that contains the address of the sender and message identification data. This helps to eliminate spam and enables new and interesting features. Additional features : homepage, forum, etc. as native functions of the system. Ipo4ta protocol is extensible. The anti-spam solution "pay-to-recipient" makes spam inefficient in the system Ipo4ta.

Document history

Date	***Version***	***Description***
01/05/05	0.1	First draft.
29/05/05	0.2	Basic specification, functional version
28/09/05	0.3	Added definitions, method description, sequences description. Aim is to give basic explanation of messaging functions
14/10/05	0.4	Added : Use case diagram
10/08/06	0.5	Anti-spam functions. Sender pays to recipient for sending a message. I-Bank network as central payment kernel of the system. HTML message part.
22/11/06	0.52	Question form message part added. User can answer directly.
29/12/06	0.6	Protocol optimisation, redesign.

Terms definitions

PADDR. Ipo4ta address. It starts always with "@".
PADDR := "@domain.name[/path/to/folder]".

USER each user has a global identification. It is a PADDR of the user (USERADDR). E.g. @slava.stefanov.org or @company.domain/john.johnson. Each user has his own PFOLDER on the server and USERADDR must be assigned to user's PFOLDER or to a subfolder of it.

PFOLDER is a normal folder in file system. It can contain some MESSAGEs and files. To

each folder one or more PADDR (FOLDARRD) can be assigned. E.g. a folder "MyRootFolder/WebProjects" of the user John Johnson can have FOLDARRD "@company.domain/john.johnson/projects" and another one "@company.domain/john.johnson/web".

MESSAGE. Ipo4ta message is an XML document with some technical information and content. Content consist of message parts (MSGPART) and attached files(ATTFILE). ATTFILE are just links to files on the sender's Ipo4ta-server.

MSGID. Message ID is unique for each message in boundaries of one Ipo4ta server. MSGID := [a-zA-Z0-9]{32}

MSGPART. Message part can be one of : TEXT, HTML, FORM.

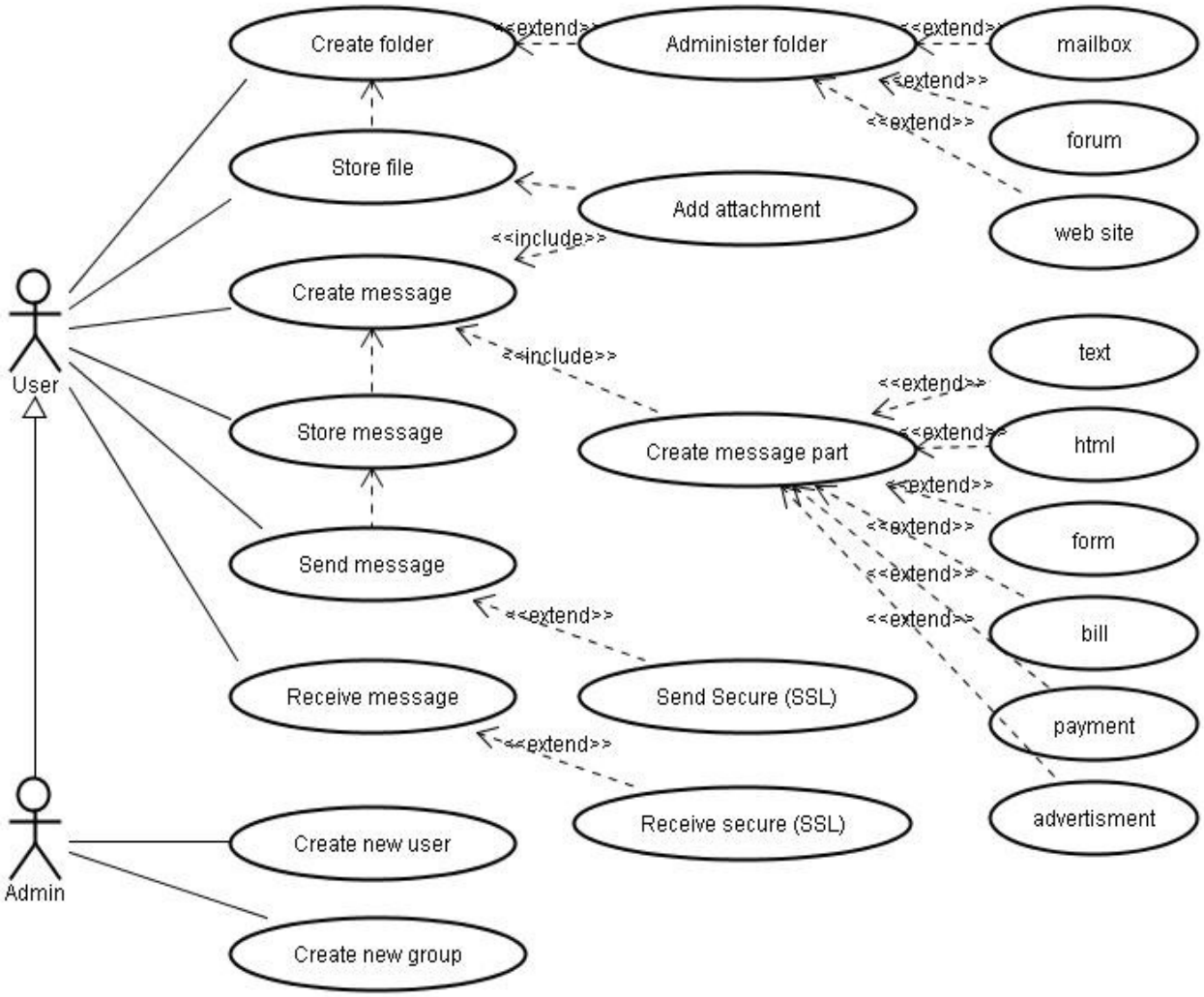
TEXT part contains simple text.

HTML part contains text in HTML format.

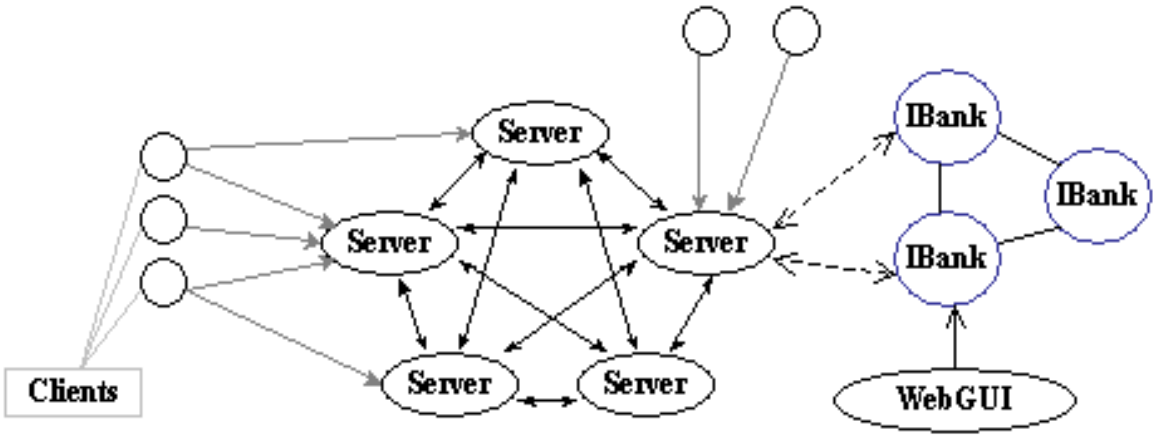
FORM part contains XML object "pflds" which defines a question form. User can directly answer by filling up the form. Answers are sent to sender server in form of XML-object "panswers".

NOTIFICATION. A notification message is an XML-message object without MSGPARTs and ATTFILEs. It is transferred to recipient server when sending a message.

Use case diagram



Network architecture.

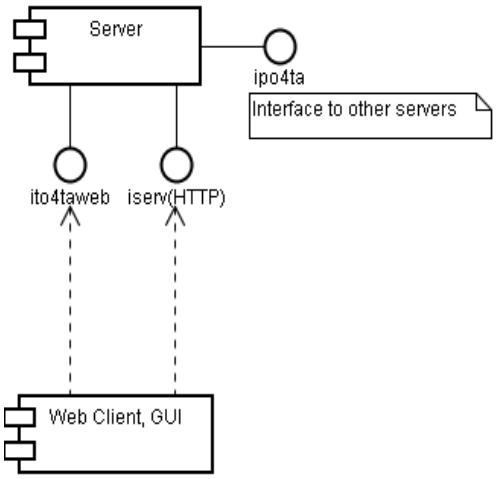


Ipo4ta servers communicate directly to each other. Communication protocol is SOAP. Interface name is "ipo4ta" (urn:soap:ipo4ta:org).

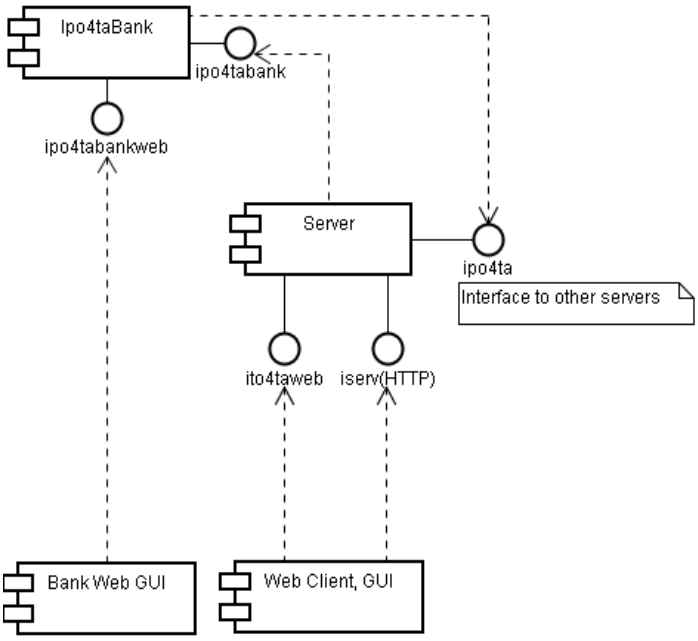
Ipo4ta client software can connect to server. A server provide a client with 2 interfaces : SOAP interface "ipo4taweb" (urn:soap:ipo4ta:org) and separate HTTP interface "iserv" for uploading/downloading files.

To obtain information about a server an application should request file ipo4ta.xml (http://domain.name:80/ipo4ta.xml or http://domain.name:8080/ipo4ta.xml) , containing the URL's for interfaces (e.g. see http://ipo4ta.com/ipo4ta.xml).

Server interfaces :



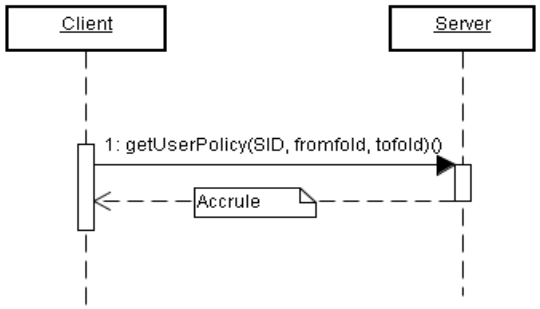
To provide users with spam protection technology "pay-to-recipient" the system must be able to communicate with IBanks. IBank provide the Ipo4ta system with simple payment means. Servers and IBanks can communicate to each other. Communication protocol is SOAP. IBank has 2 interfaces : "ipo4tabank" and "ipo4tabankweb":



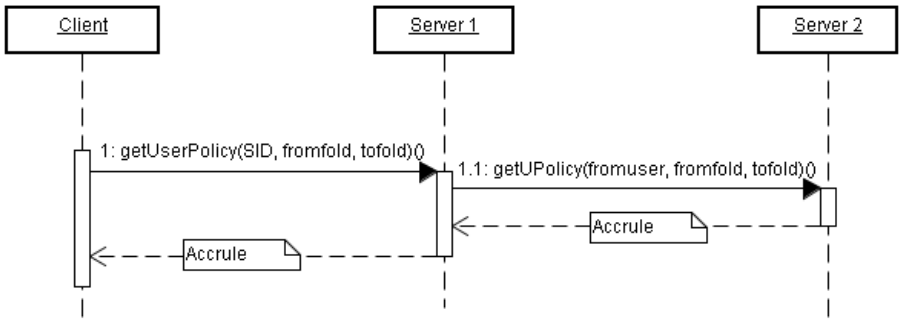
Interface definitions can be found in source distribution (ipo4ta/src/ipo4ta.xsd , ipo4ta/src/ipo4ta.wsdl)
Here are interface methods descriptions. For details see source code.

Interface IPO4TAWEB (SOAP)

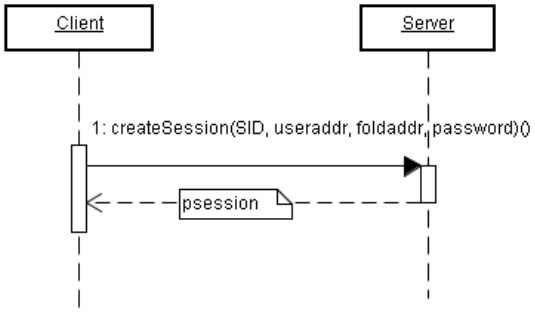
1. getServerInfo(). Get Ipo4ta server information.
2. getUserPolicy(SID, fromfold, tofold) returns access policy for current user (SID), who wants to send a message from folder "fromfold"(FOLDADDR) to folder "tofold"(FOLDADDR). Local case, "tofold" is a local folder :



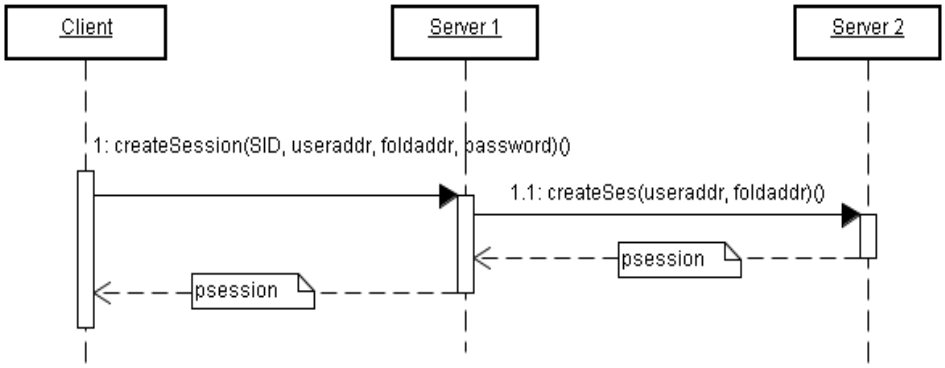
   If "tofold" is not a local PADDR the method ipo4ta.getUPolicy(..) will be called on the server of "tofold" (ipo4taweb. getUserPolicy -> ipo4ta. getUPolicy) :



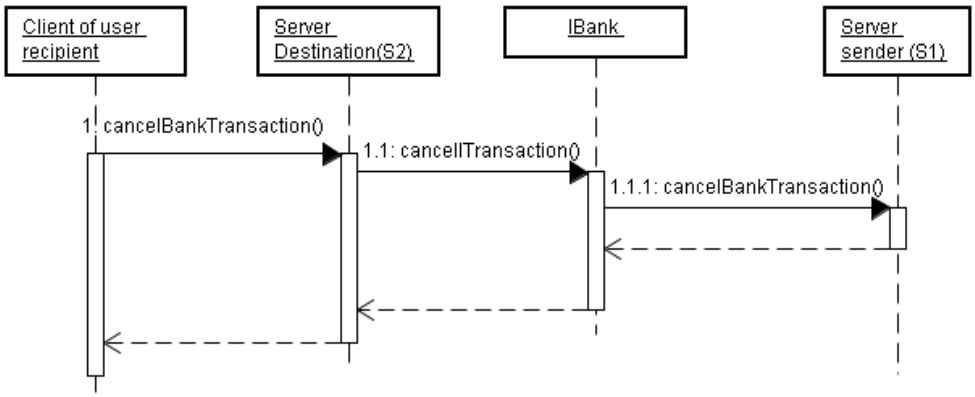
3. createSession(SID, useraddr, foldaddr, password). Initially a user opens a session with "useraddr" (his USERADDR) and "password". Optionally "foldaddr" (FOLDADDR) can de specified if the user need to access a folder(on the same server), anther than his own one. If the session is already opened and a user need to access another folder (on the same or on the remote server) SID and "foldaddr" must be not null. Local session case:



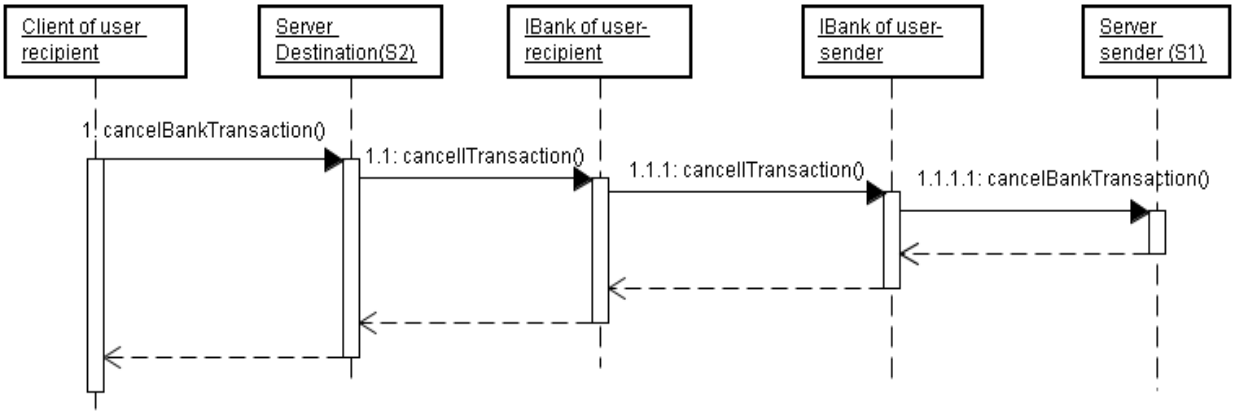
Remote session case (ipo4taweb.createSession -> ipo4ta.createSes) :



4. getSession(SID). Get session by SID, local and remote (ipo4taweb.getSession -> ipo4ta.getSes) cases.
5. putMessage(SID,msginf). Create/update a message on the server.
6. getMessage(SID, msgnum). msgnum := [0-9]+ , numeric ID of the message.
7. getMessages(SID, condition). Get an aray of messages by the condition "condition".
8. deleteMessages(SID, foldnum, msgnums). Delete messages from folder with numeric ID "foldnum". "msgnums" is the list of message numbers.
9. cancelBankTransaction(SID, msgnum, itid). Cancel payment transaction for message number "msgnum" and transaction ID = "itid".
   Case when recipient and sender have the same IBank :

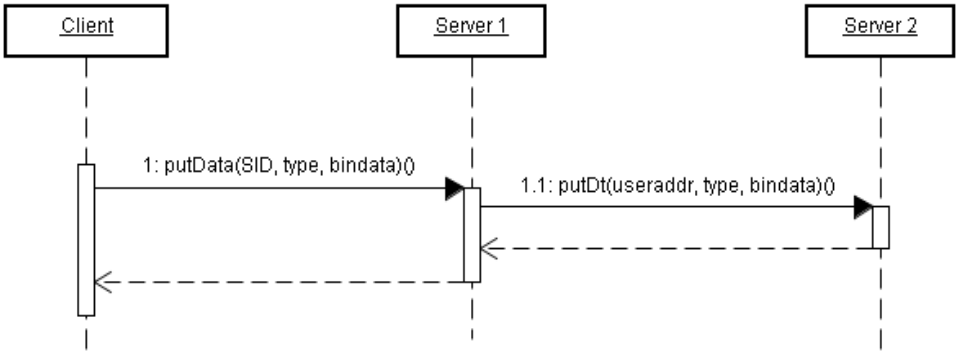


Case when recipient and sender have different IBanks :

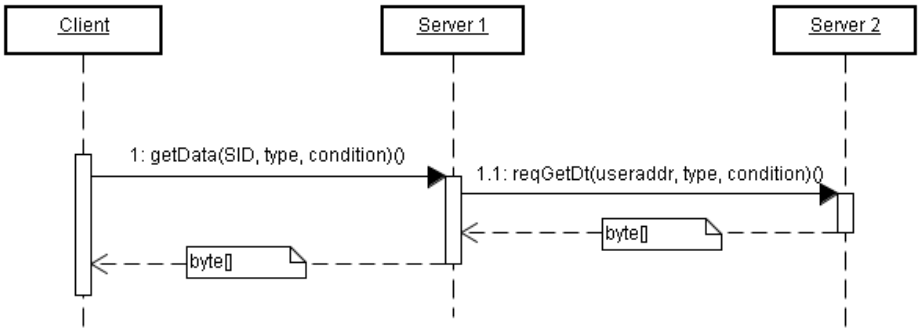


10. closeBankTransaction(SID, msgnum, itid). Commits a transaction. Similar to cancelBankTransaction(..).

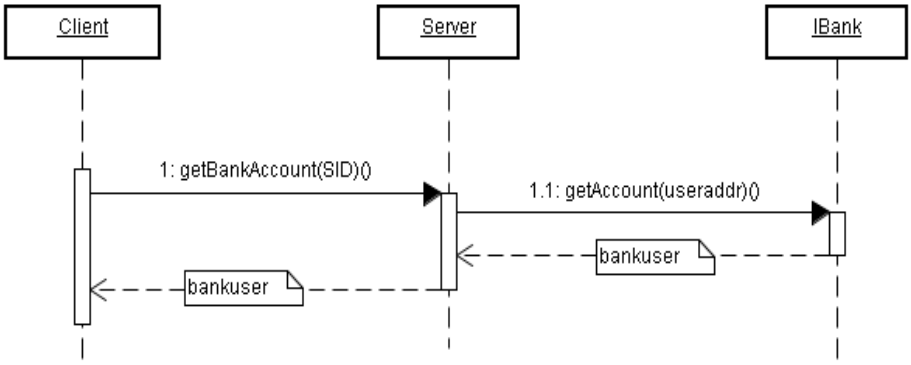
11. putData(SID, type, bindata). Extensibility method. Permits to work with different types of message related data. Currently is used to store user answers to FORM.



12. getData(SID, type, condition). Extensibility method. Currently is used to get user answers to FORM.



13. delData(SID, type, id[]). Extensibility method. Currently is not used.
14. getBankAccount(SID). Gets user bank account data.



15. getFolders(). Client GIU method (C-GUI). Gets list of folders.
16. putFolder(). C-GUI. Creates/updates a folder.
17. deleteFolder(). C-GUI. Deletes a folder.

18. addPaddress(). C-GUI. Adds a new PADDR (FOLDADDR) for a folder.
19. deletePaddress(). C-GUI. Deletes a FOLDADDR.

20. getFiles(). C-GUI. Gets list of files in folder.
21. storeFile(). Currently is not used.
22. deleteFile(). C-GUI. Deletes a file.

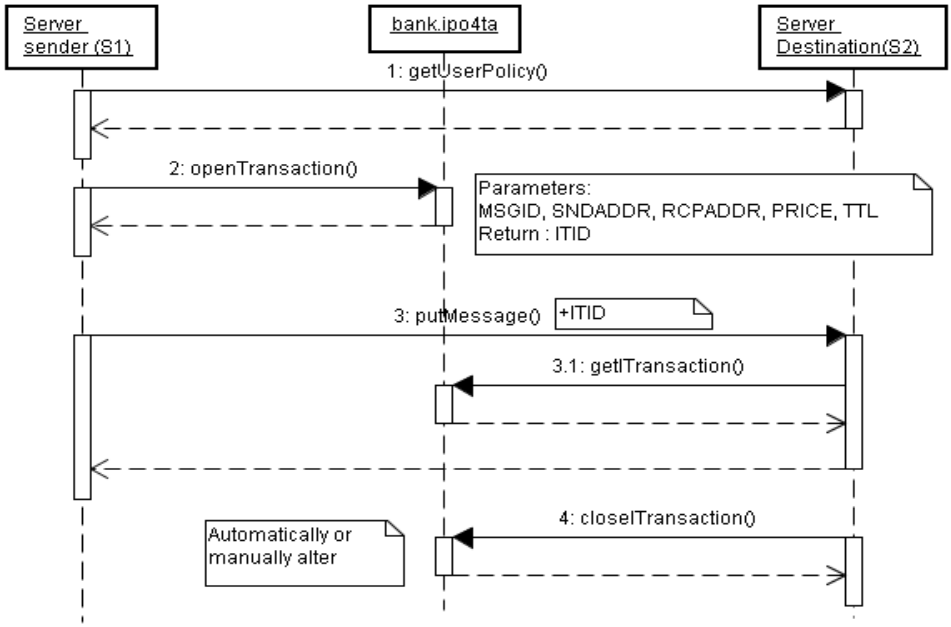
23. getUsers(). C-GUI. Gets list of users profile.
24. putUser(). C-GUI. Creates/updates a user profile.
25. deleteUser(). C-GUI. Deletes a user profile.
26. putUserPrefs(). C-GUI. Stores a user preferences.

27. getAddrBookUsers(). C-GUI. Gets list of address book profile.
28. putAddrBookUser(). C-GUI. Creates/updates an address book profile.
29. deleteAddrBookUser(). C-GUI. Deletes an address book profile.

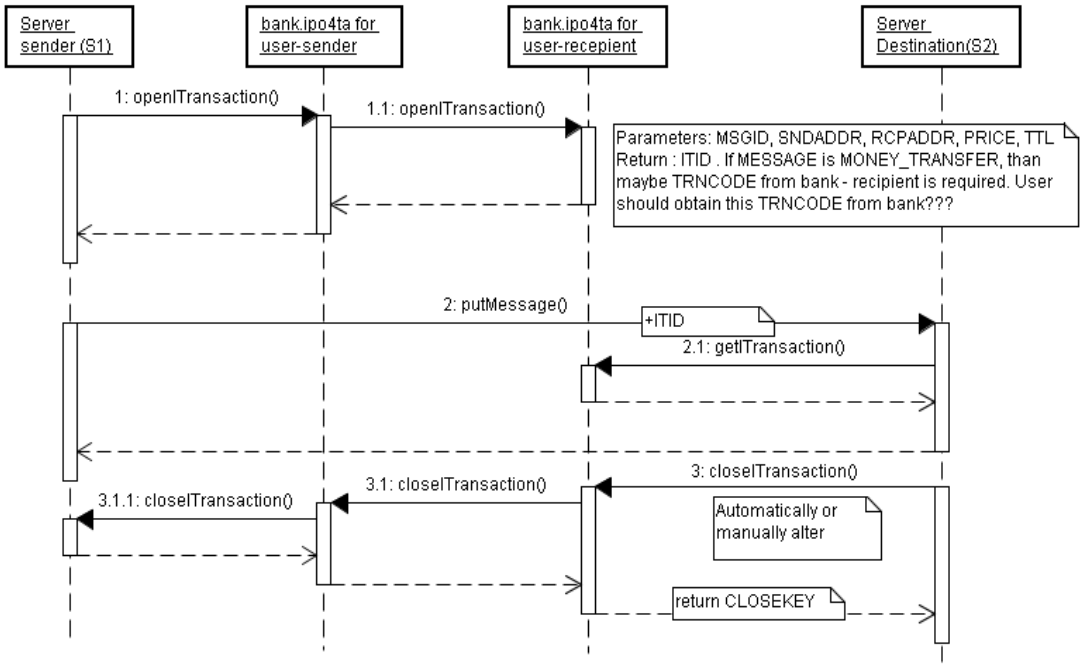
30. putAccrule(). C-GUI. Adds a new access rule for a folder.
31. deleteAccrule(). C-GUI. Deletes an access rule of a folder.

Interface IPO4TA (SOAP)

1. GetSInfo(). Get Ipo4ta server information.
2. GetUInfo(SID,USERADDR). Get local or remote user profile. Authorised user (SID!=null) can get his own profile (USERADDR==null) or a remote profile (USERADDR!=null). Non-authorised user (SID==null) can get only local user profile.
3. getUPolicy(). see IPO4TAWEB.getUserPolicy .
4. createSes(). see IPO4TAWEB.createSession .
5. getSes(). Gets a session by SID. Refresh session.
6. putMes(). If a message was saved by the user with status READY, the server can transfer a notification to a destination one with putMes() method. If sending a notification require payment from user-sender the sequence is following (case user-sender and user-recipient have the same IBank):



if banks are different :

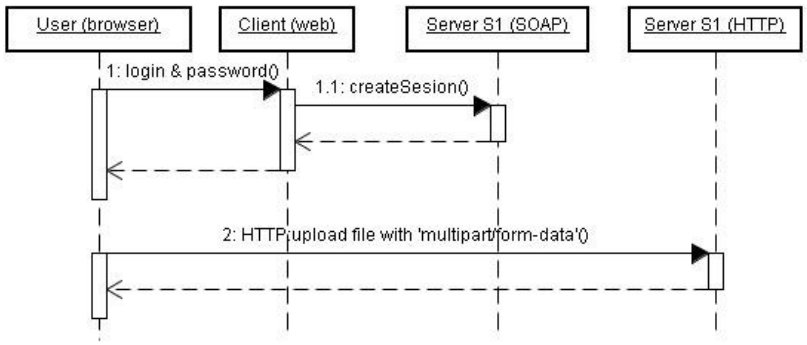


7. getMes(). Get one message from sender-server if notification was received by recipient-server.
8. openMes(). Recipient-server can inform sender-server if the user-recipient open a cached message. Message aim is to provide a sender with open-message statistic.
9. cancelBTransaction(). see IPO4TAWEB.cancelBankTransaction() .
10. closeBTransaction(). Commits a transaction. similar to cancelBTransaction().
11. putDt(). see IPO4TAWEB.putData() .
12. getDt(). see IPO4TAWEB.getData() .
13. delDt(). Currently is not used.

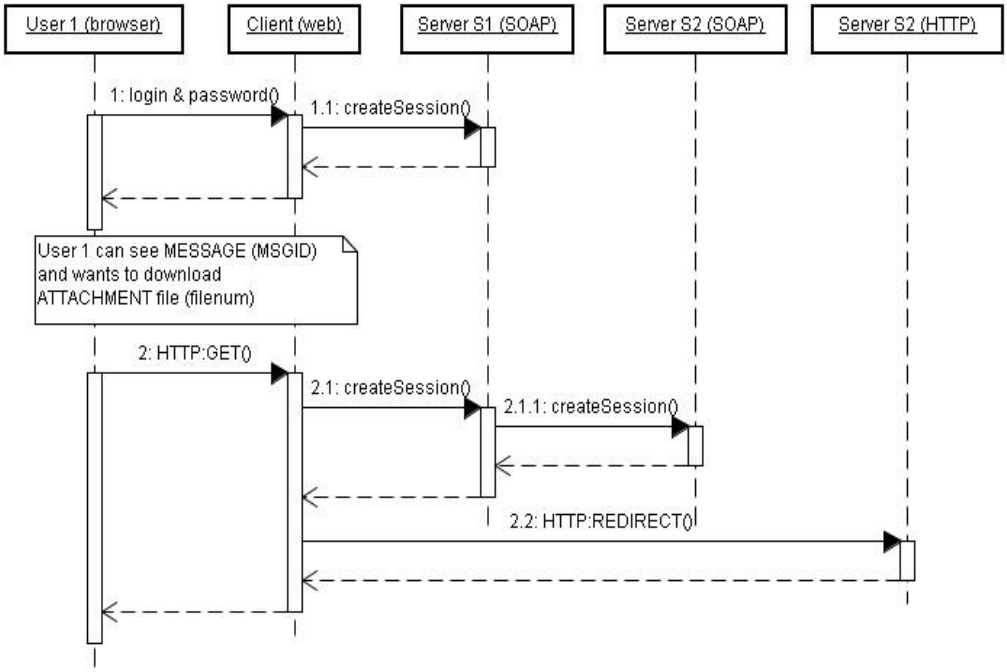
Interface IDISP (HTTP).

for details see org.ipo4ta.server.fileDisp.java

1. Upload a file. Parameters of HTTP-multypart/form (sid=SID, foldnum=FOLDER_NUMBER, gotourl=URL_TO_RETURN, filename=FILENAME)



2. Download a file.
   http://S2_HTTP_INTERFACE_URL?sid=SID_S2&msgid=MSGID&filenum=FILENUM



TODO

folder management
message part types (add invoice, payment)
user profile management
security improvements (e.g. a message available only by SSL, encryption)
group mails
scheduling
active rules (e.g. if sender==”aaa” than reply to “bbb”)
payments
administrator tools
P2P kernel