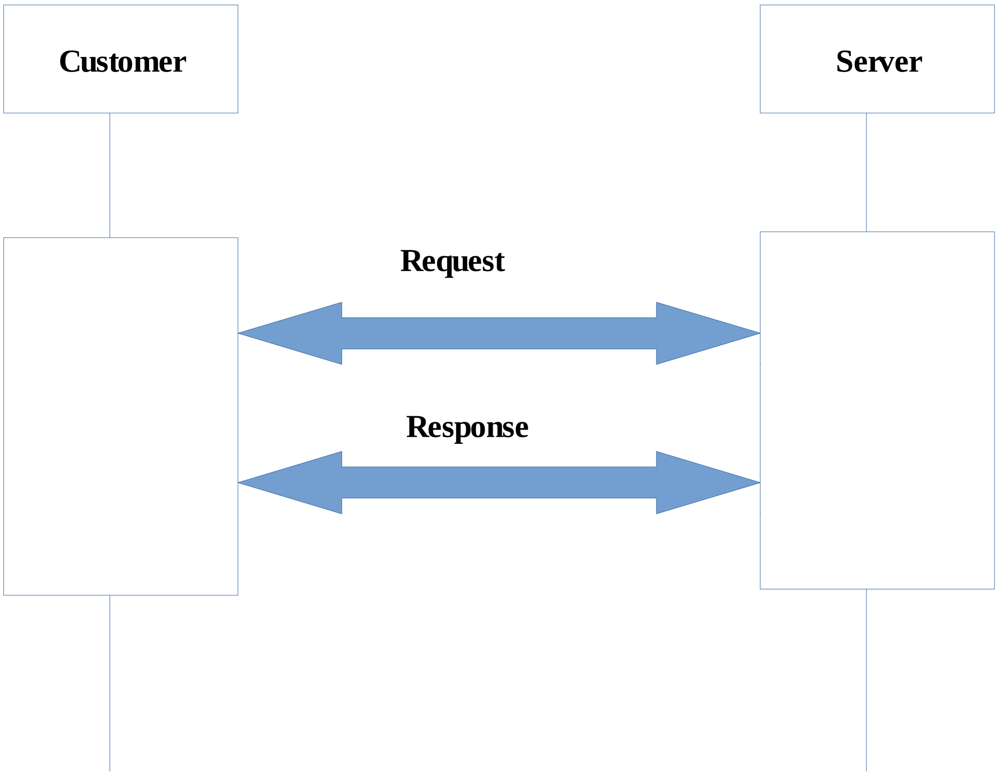
Babel nfc

Protocol:



Request:

Request informations	Request		
Header size, route size, data size	Header	Route	Data

Response:

Request informations	Reponse	
Status size, data size	Status 200, 201, 400, 401, 403, 404	Data

Request/Response informations:
Is int32 informations. Binary data.

Status code correspond to http code. For more information:
https://fr.wikipedia.org/wiki/Liste_des_codes_HTTP

Send by/ receive by	Request			Response	
	Route	Header	Data	Status	Data
Customer / server	/login		user_mail: “”, pass: “”	200	Id: “”, name: “”, token: “”
				401	
Customer / server	/logout		user_mail: “”, token: “”	200	
				401	
Customer / server	/create		user_mail: “”, pass: “”	201	Id: “”, name: “”, token: “”
				403	Error: “”
Customer / server	/delete	id: “”, token: “”		200	
				401	
Customer / server	/contacts	id: “”, token: “”		200	[id:””, name: “”, status: “”]
				401	
Customer / server	/add_contact	id: “”, Token: “”	contact_email: “”	200	
				401	
				403	Error: “”
Customer	/remove_contact	id: “”, token:	contact_id: “”	200	

/ server		“”			
				401	
				403	Error: “”
Customer / server	/call	id: “”, token: “”	contact_id: “”	200	ip: “”
				401	
				403	Error: “”
Customer / Customer	/send	id: “”, token: “”	type: “”, nb_packets: “”, size_packet: “”, packets: []	200	
				401	
				403	Error: “”

Fields:
All name fields should be treat in lowercase.

Format of string information:
You could use all you include inhibitor like \”.

Header:
‘Id’ is the user’s id.

Routes description:
/login
Login user. Return his id, his name and his unique token (random). Token is create on each login action.
Return information on user.

/logout
Logout user.

/create
Create a user with email and password.
Return information on user.

/delete
Delete a user.

/contacts
Return list of contact with id, name and status.

/add_contact

Add contact with email given.
Contact will be add when the other user will accept to add contacts.

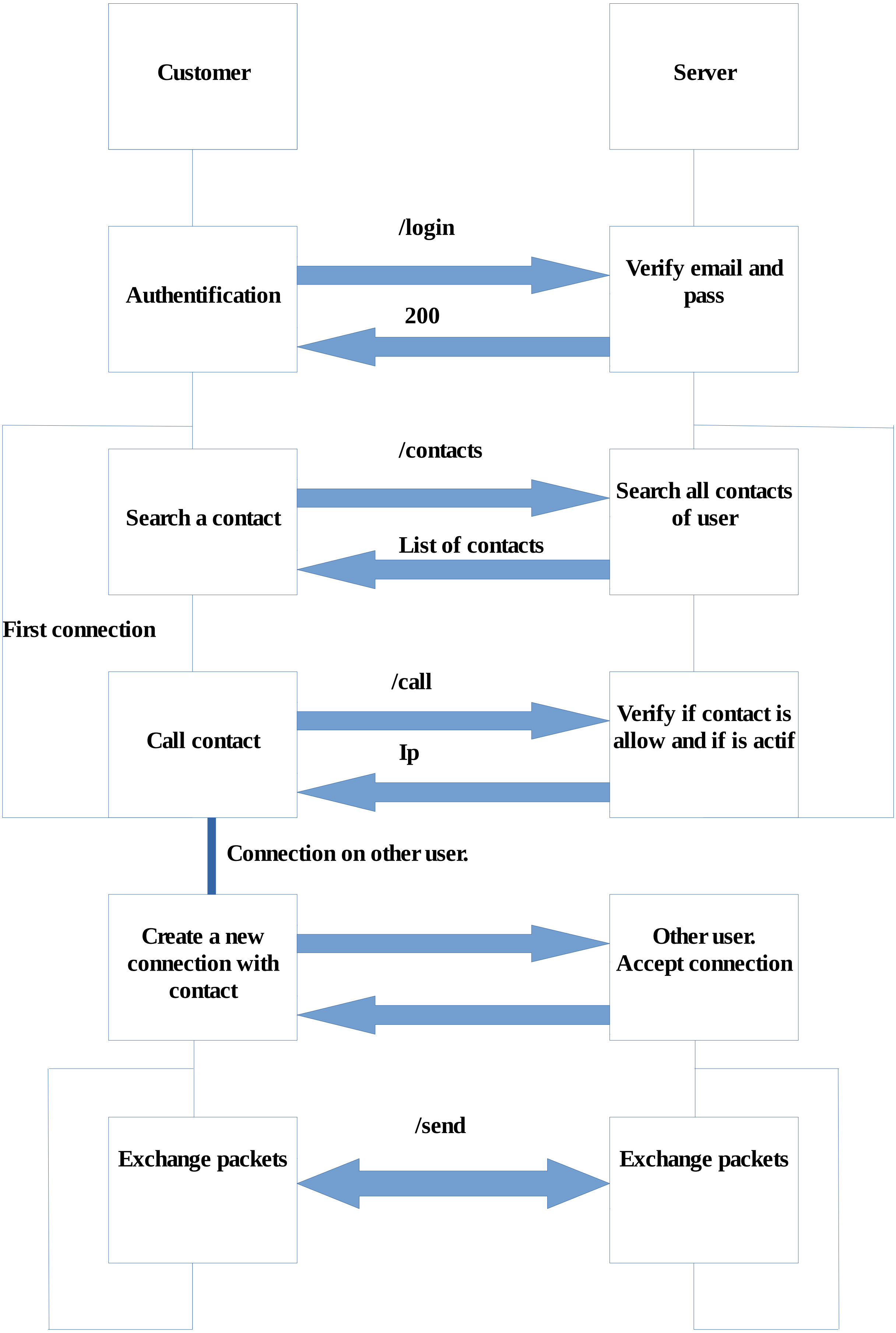
/remove_contact
Remove contact with email.

/call
Return the ip of contact for call.

/send
Send on a call packets.
A packets must be string/binary… It's treat by customer, depend of type. Each packet have same size.

Call protocol:

For create a call use the route "call". This route return a "call_id". This id is unique and have to be pass on route "connect_call" on a new connection on to begin call.
Thanks to this new connection, server and customer can again communicate during a call.


Customer
Server
/login
Authentification
Verify email and pass
200
/contacts
Search a contact
Search all contacts of user
List of contacts
First connection
/call
Call contact
Verify if contact is allow and if is actif
Ip
Connection on other user.
Create a new connection with contact
Other user. Accept connection
/send
Exchange packets
Exchange packets