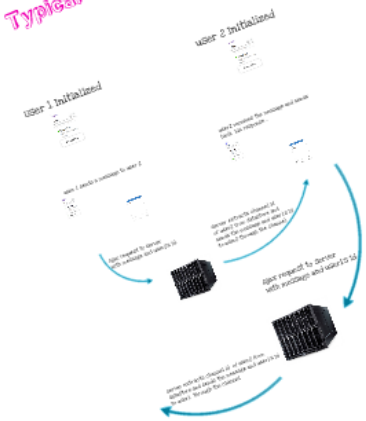
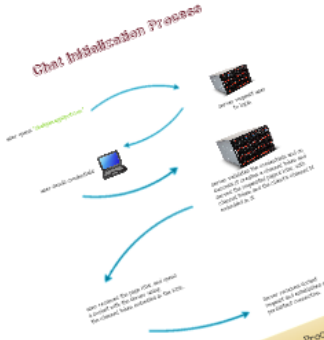
Typical Chat Process



Chat Implementation :

Chat Initialization Process



This completes the Initialization Process

Thank You

Chat App on GAE

Key features:

one - one text chating
group chating

text with text replacement
text with image macro's

video and audio chat
Chat history
setting Status

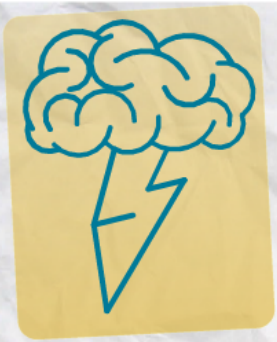
Team Members

Tushar Chandra

Dontula Kapil

Konduri Praneeth

Surya Prakash Reddy

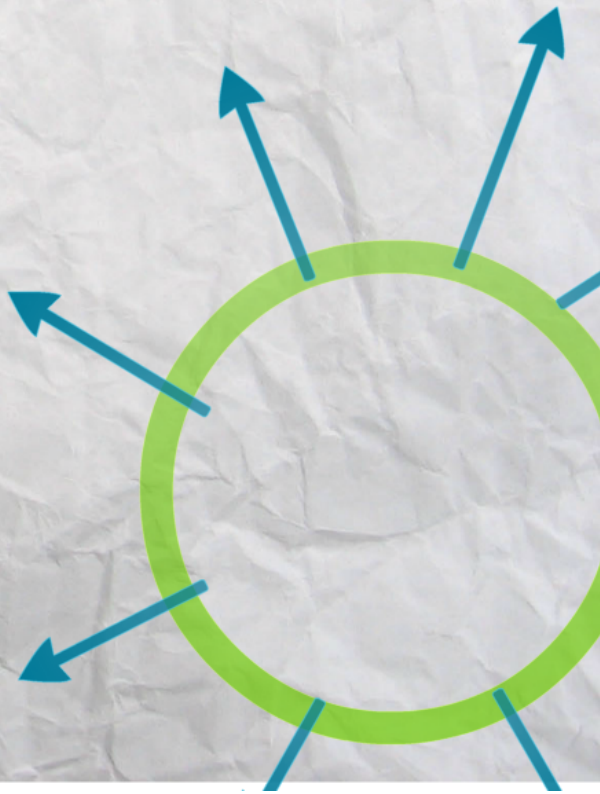


Key features:

one - one text chating
group chating

text with text replacement
text with image macro's

video and audio chat
Chat history
setting Status



one - one text chating
group chating

text with text replacement
text with image macro's

video and audio chat
Chat history
setting Status

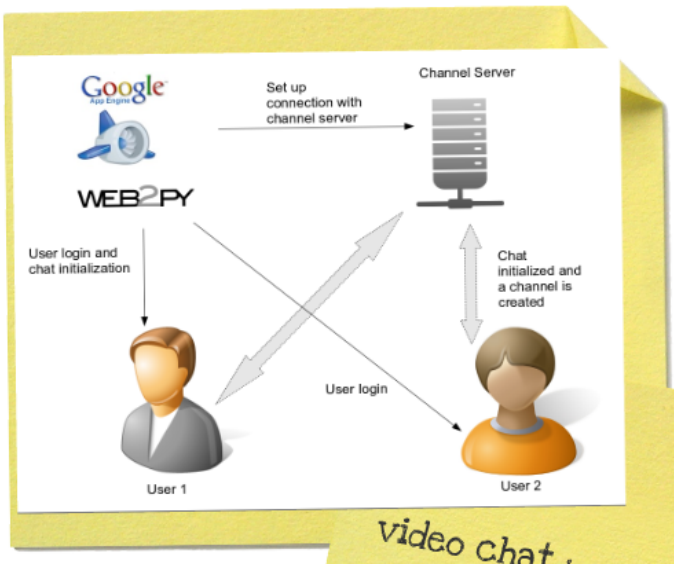
Important Details

video chat is based on rtmfp protocol which is a peer - peer connection using python based media server.

Chat overview

chat:

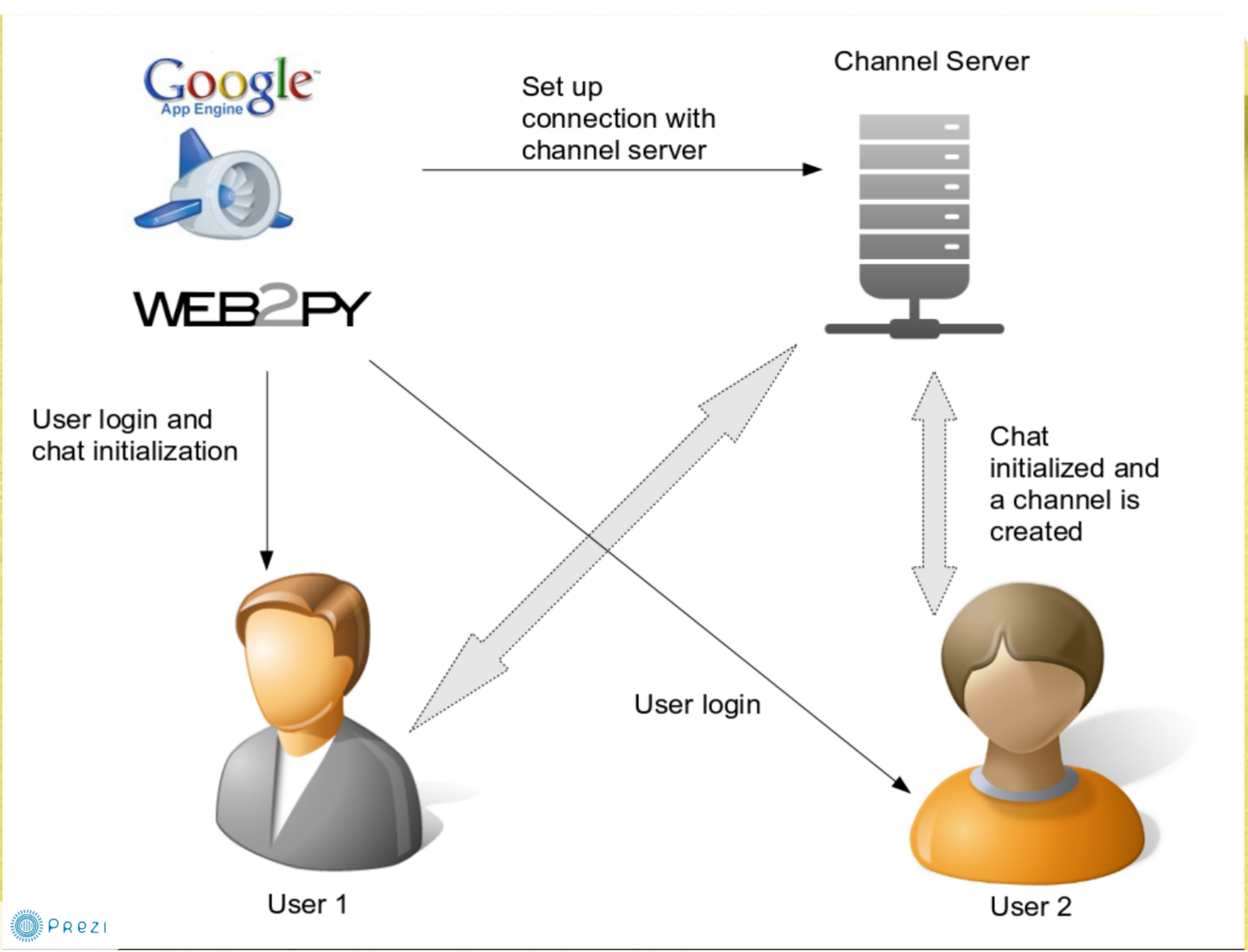
chat with his friends
rsistent channel.



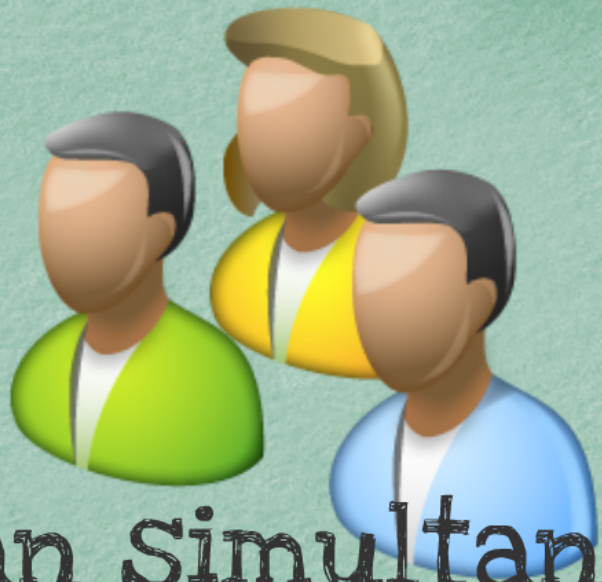
Prezi

one - one chat:

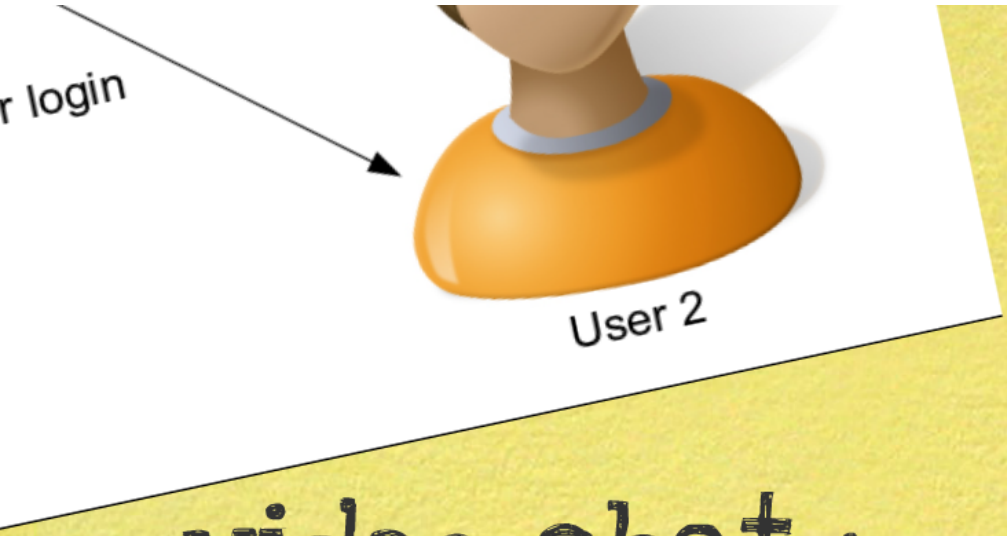
user can chat with his friends using a persistent channel.


Google
App Engine
WEB2PY
Channel Server
Set up
connection with
channel server
User login and
chat initialization
Chat
initialized and
a channel is
created
User login
User 1
User 2

group chat :



single user can simultaneously chat with a number of users.



video chat :

A user can video chat with a single person at a time.

Chat Implementation :

Chat Initialization Process

"chatgae.appspot.com"

user opens "chatgae.appspot.com"

Server request user to login

user sends credentials

Server validates the credentials and on success it creates a channel token and serves the requested page's HTML with channel token and the client's channel Id embedded in it.

user recieves the page HTML and opens a socket with the server using the channel token embedded in the HTML.

Server receives socket request and establishes a persistent connection.

This completes the Initialization Process

user opens "chatgae.appspot.com"

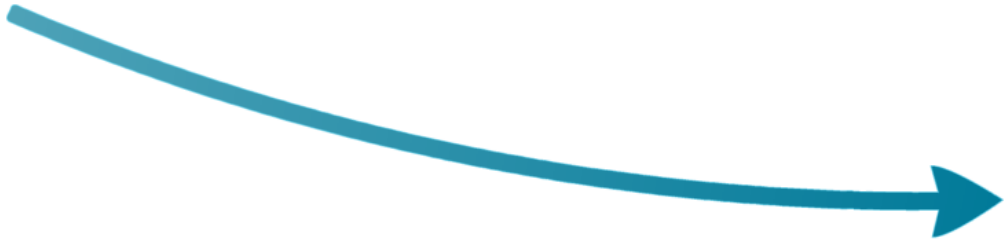
Server request user to login

user sends credentials

Server validates the credentials and on success it creates a channel token and serves the requested page's HTML with channel token and the client's channel Id embedded in it.

user recieves the page HTML and opens
a socket with the server using
the channel token embedded in the HTML.

Server receives socket request and establishes a persistent connection.



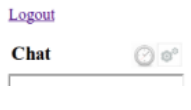
Server receives socket request and establishes a persistent connection.

This completes the Initialization Process

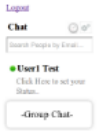
Typical Chat Process

User 1 Initialized

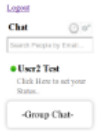
User 2 Initialized



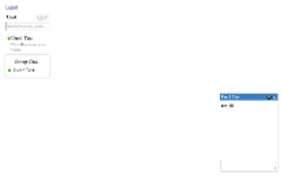
User 1 Initialized



uSer 2 Initialized



uSer 1 Sends a meSSage to uSer 2



uSer2 receiveS the meSSage and SendS back hiS reSponSe ..

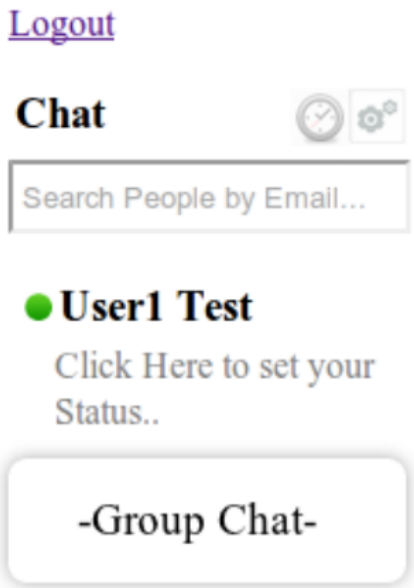
Ajax requeSt to Server with meSSage and uSer2'S Id.

Server extractS channel id of uSer2 from dataStore and SendS the meSSage and uSer1'S Id to uSer2 through the channel

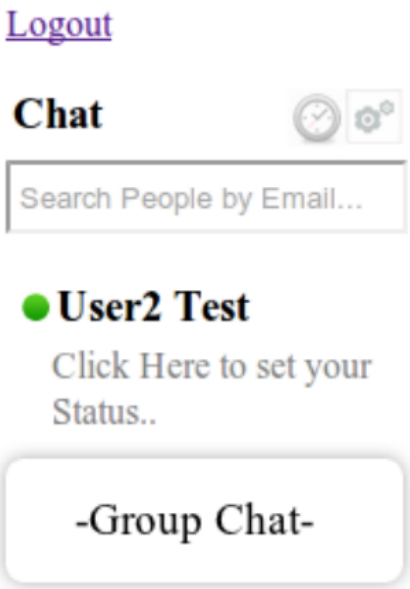
Ajax requeSt to Server with meSSage and uSer1'S Id.

Server extractS channel id of uSer2 from dataStore and SendS the meSSage and uSer1'S Id to uSer1 through the channel

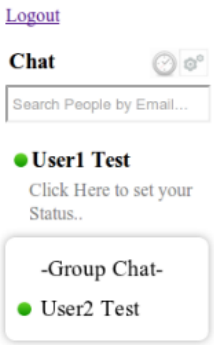
User 1 Initialized



user 2 Initialized



user 1 sends a message to user 2



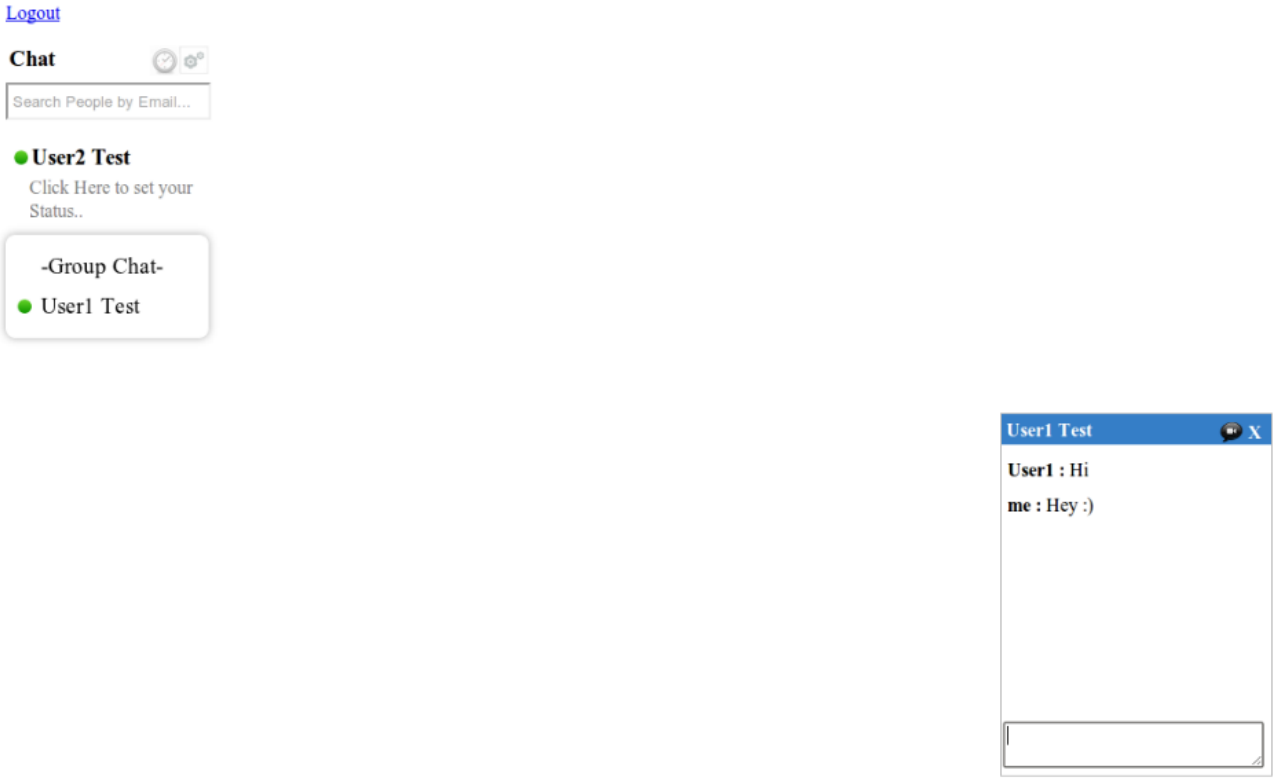
Ajax request to server
with message and user2's Id.

Server
nd user2's Id.

Server extracts channel id
of user2 from dataStore and
sends the message and user1's Id
to user2 through the channel

Ajax request to server
with message and user1's Id.

user2 receives the message and sends back his response ..



extracts channel id
of user2 from dataStore and
sends the message and user1's Id
to user2 through the channel

Ajax request to server
with message and user1's Id.

Server extracts channel id of user2 from
dataStore and sends the message and user1's Id
to user1 through the channel

Video Chat Implementation

User 1

Video Chat Implementation

User 1

Initiation of video chat

Connects to the Adobe Stratus Server through "rtmfp protocol" for registering a nearId for the given publish id, which in our case is the channel id.

VideoIO loads and starts self video instance!!

Stratus Server responds by generating a nearId for the given publish id and sends it back to the user.

User 2

User 1 sends publish id with nearId to User 2 through the channel.

User2 recieve
User 1 and I

Us

Initiation of video chat

VideoIO loads and starts self video instance!!

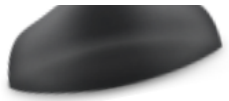
ser 1

Connects to the Adobe stratus server through "rtmfp protocol" for registering a nearId for the given publish id, which in our case is the channel id.

s and starts
instance!!

Stratus Server reSponds by generating a nearId for the given publish id and Sends it back to the uSer.

Us



stratus server
the given publish id and send

User 1 sends publish id with nearId to User 2 through the channel.

User 2

User 2

blish id with nearId
hrough the channel.

User2 recieves the publish id and nearId of User 1 and Initiates his self videoIO instance

User 2 publishes his video to generate his nearId and sends it through channel to User 1

Starts User 1's video instance using the publish id and nearId of User 1.

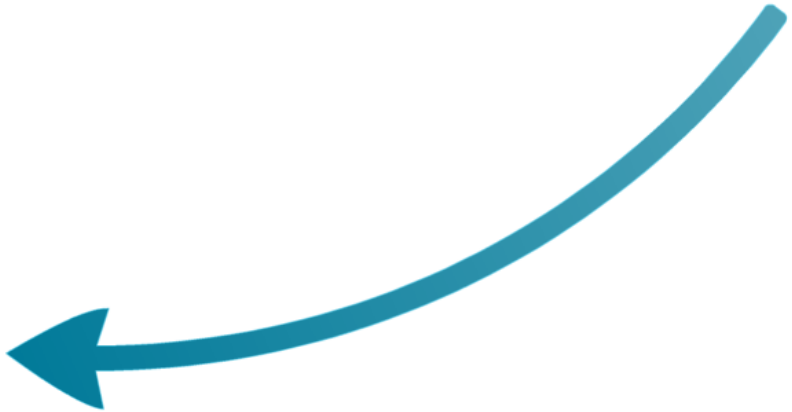
User 1 recevies the url of the User 2 and starts the corresponding videoIO and plays his stream

User 2

User2 recieves the publish id and nearId of User 1 and Initiates his self videoIO instance

User 2 publishes

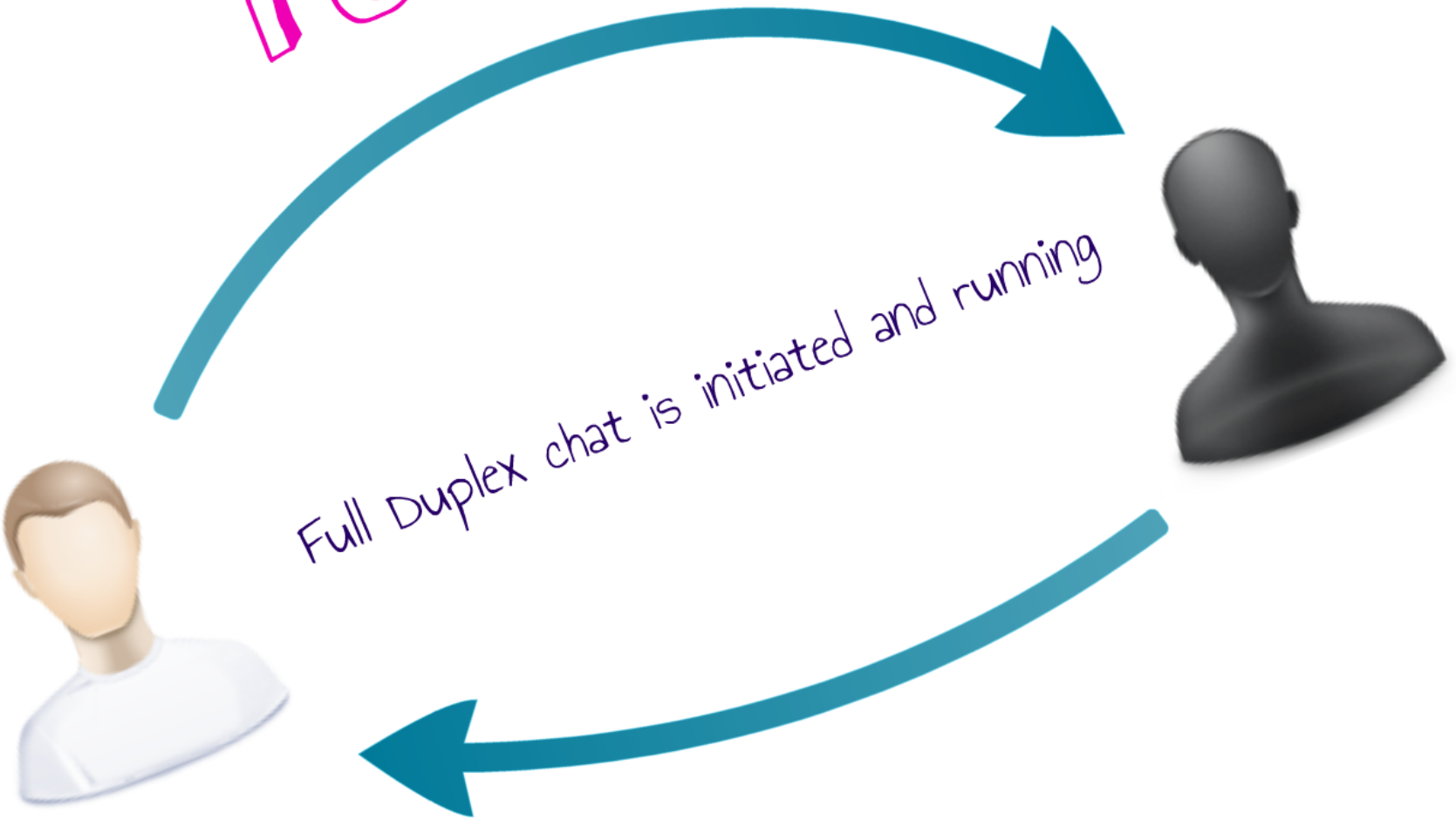
User 2 publishes his video to generate his nearId and sends it through channel to User 1



Starts User 1's video instance using the publish id and nearId of User 1.

User 1 recevies the url of the User 2 and starts the corresponding videoIO and plays his stream

result



Group Chat

User 1 initiates Group Chat

user 1 selects users from his friends list and initiates a group chat.

User 1 sends a message to the group thus Group Chat is initialized at the server side.

UserIds of other group members and the message are sent to server.

Server extracts the channel Ids of the group members and sends the message and Userids of the other group members to each user

User 1 initiates Group Chat

User 1 initiates Group Chat

user 1 selects users from his friends list and initiates a group chat.

User 1 sends a message to the group thus Group Chat is initialized at the server side.

UserIds of members

User 1 sends a message to the group thus Group Chat is initialized at the server side.

UserIds of other group members and the message are sent to server.

Server extracts the channel Ids of the group members and sends the message and Userids of the other group members to each user

sage are

user settings:

- set his status message/flag

Can set his status message/flag.
Can see his chat history
Add/Remove block/unblock a friend

Set his personal text-text and
text-image replacement macro's

Tools Used :

- Web2py Framework
- Flash VideoIO Plugin
- Google Data Store
- Channel API
- AJAX & jQuery Library

- Web2py Framework
- FlaSh VideoIO Plugin
- Google Data Store
- Channel API
- JavaScript, AJAX & jQuery Library
- Stratus Server for peer-peer connection initialization for Video/Audio Chat.
- Chrome Extension API.

web2py

Free open source full-stack framework for rapid development of fast, scalable, secure and portable database-driven web-base applications. Written and programmable in Python

It follows MVC architecture

Channel API

The Channel API creates a persistent connection between your application and Google servers, allowing your application to send messages to JavaScript clients in real time without the use of polling

Adobe Stratus Server

It acts as an intermediate server for initializing a peer to peer connection between users trying to stream video/audio data.

Using custom Javascript

Ch

chatgae.appspot.com

Chrome Extension

Cross Platform, Cross Browser,

Cross Domain, highly scalable

Text/Audio/Video chat to users.

Results

Cross Platform, Cross Browser,
Cross Domain, highly scalable
Text/Audio/Video chat to users.

Using custom Javascript

chatgae.appspot.com

Chrome Extension