Live-Jamming

Technical Documentation

15/05/2010

Project Name	**LIVE-JAMMING**
Project Leader	NESPO Pierre
Members	DUPUY Mathieu O'CONNEL Gregory PETIT Aylic SARDA Charles
Filename	2011_TD_FR_LIVEJAMMING
Date	15/05/2010
Author	NESPO Pierre

Summary

I.Global Modeling

1.Client

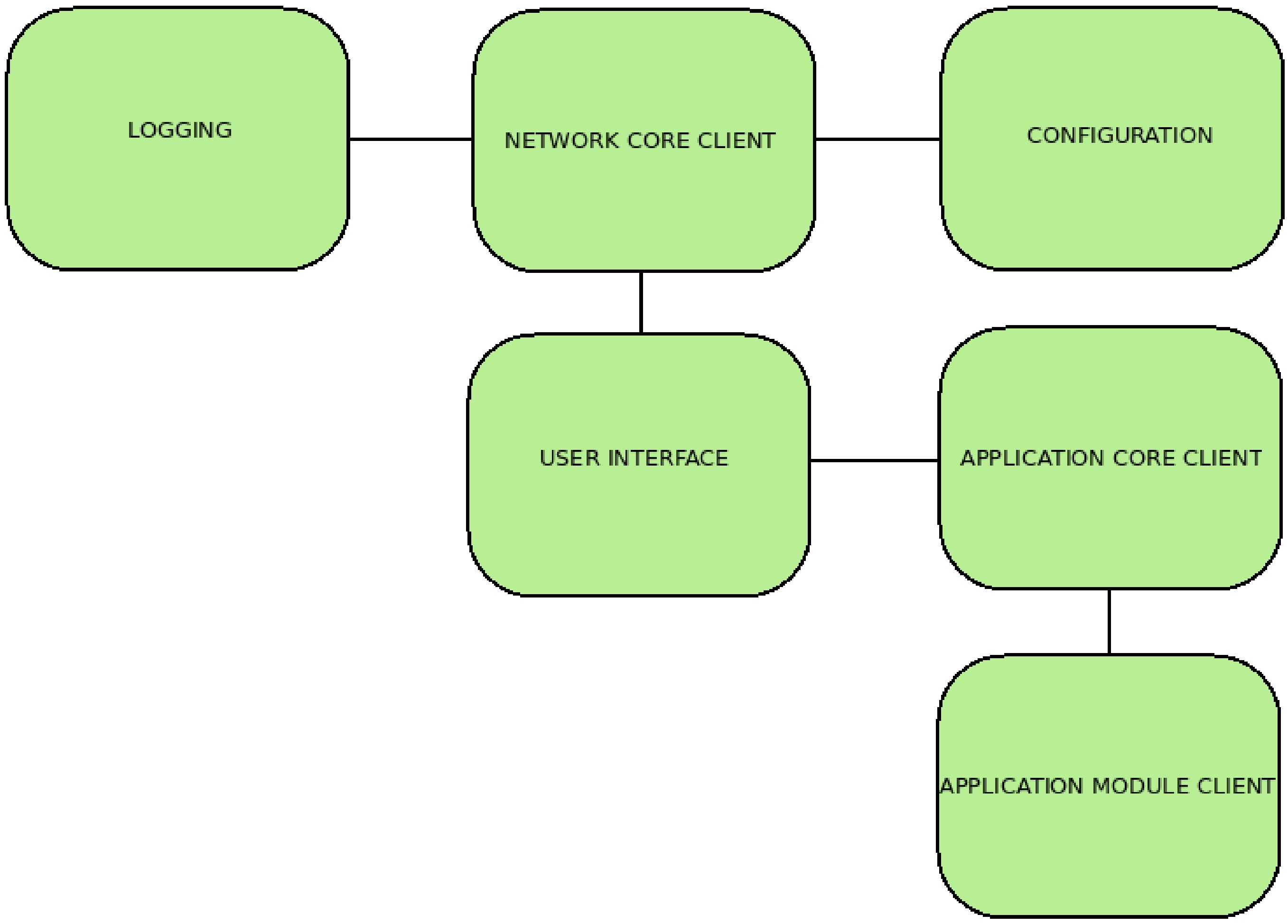
The main components of Live-Jamming client are :

- A User Interface.
- A Network Core.
- An Application Core.
- An interface IapplicationModule which lets the possibility to easily add functionalities without modifying Application core.

Some minors components :

- A Logging module which trace every actions done on the client.
- Configuration module which loads client parameters from a configuration file.

Detailed descriptions of each interface and module in the scheme above.


LOGGING
NETWORK CORE CLIENT
CONFIGURATION
USER INTERFACE
APPLICATION CORE CLIENT
APPLICATION MODULE CLIENT

1.Server

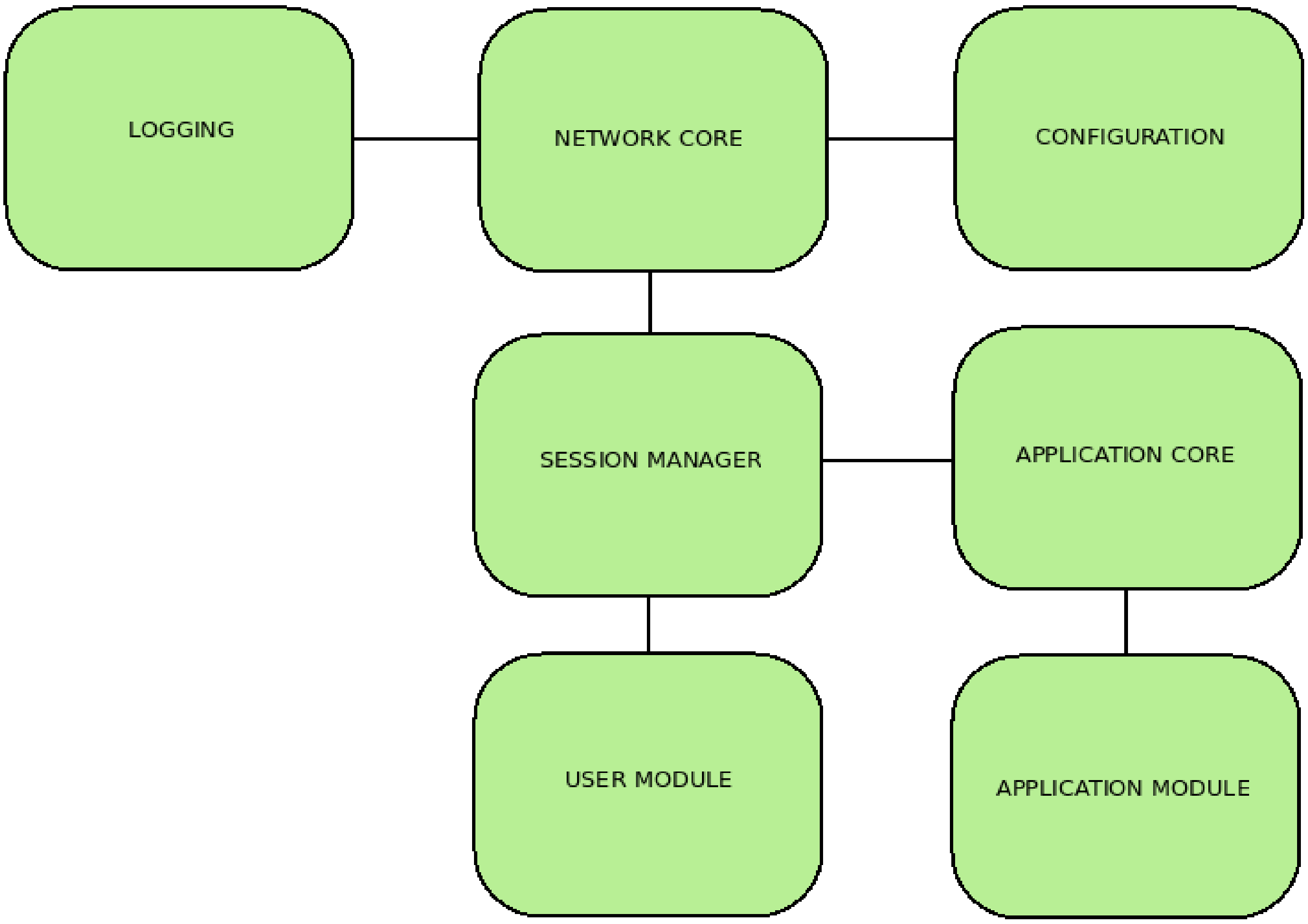
The main components of Live-Jamming server are :

- A Network Core.
- A Session management module to handle user sessions.
- An Application Core.
- An interface IUserModule which lets easily add user functionnalities whitout modifying application / network core.
- An interface IapplicationModule which lets easily add server functionnalities whitout modifying application / network core.

Some minors components :

- A Logging module which trace every actions done on the client.
- Configuration module which loads client parameters from a configuration file.

Detailed descriptions of each interface and module in the scheme above.

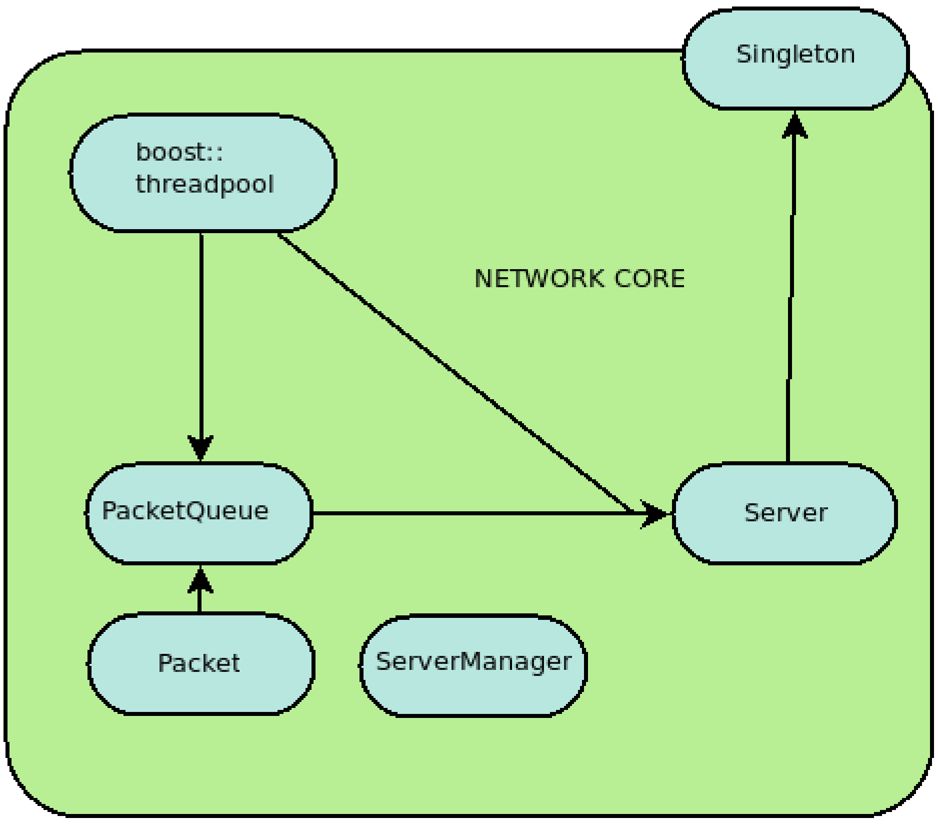

LOGGING
NETWORK CORE
CONFIGURATION
SESSION MANAGER
APPLICATION CORE
USER MODULE
APPLICATION MODULE

I.Detailed Modeling

1.Components

a.Server

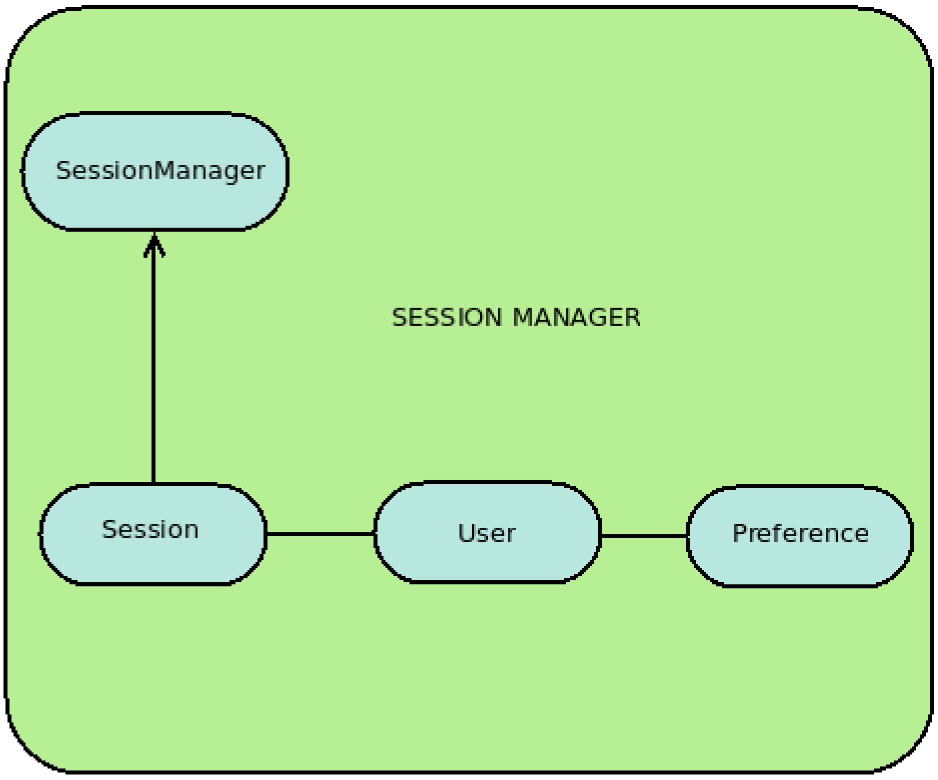
- Network core :



The patter used is the ThreadPool (one priority datas reception thread which handle a queue and *n*Thread putting packets in in.

The different components of application don't need the same prioritization, a prioritization mechanism is possible (audio > userinfo)

- Session Management :

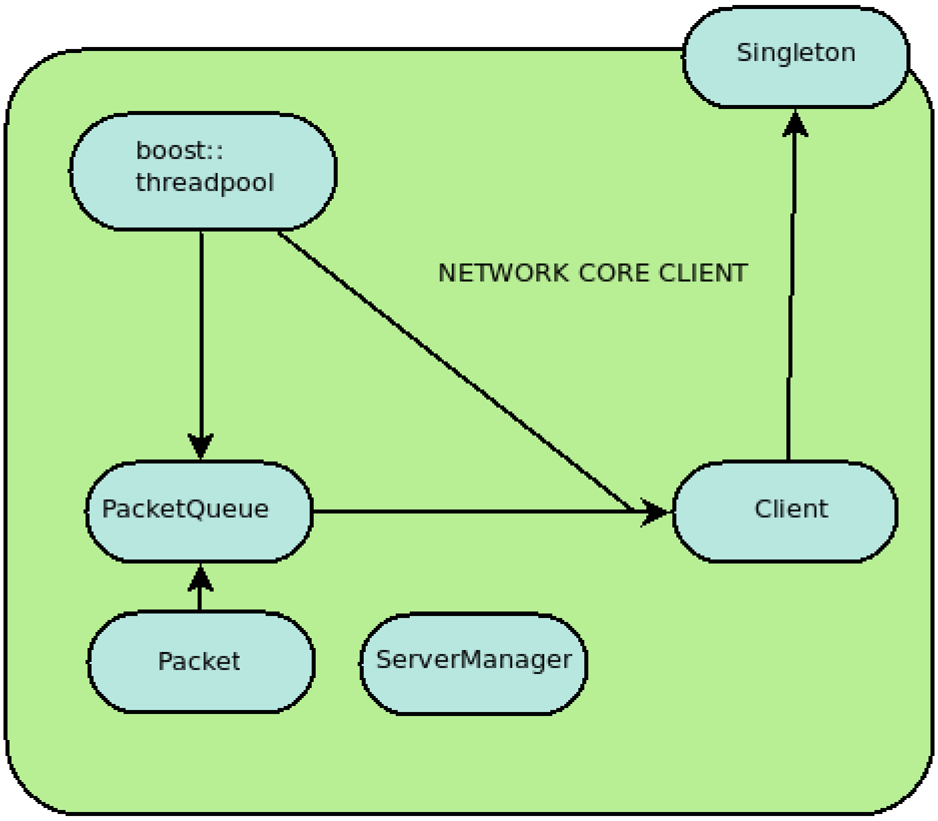


This component will allow to handle user authentication datas and informations.

It manages automatically the authentication part and validate authenticated packet.

a.Client

- Network core



The patter used is the ThreadPool (one priority datas reception thread which handle a queue and *n*Thread putting packets in in.

The different components of application don't need the same prioritization, a prioritization mechanism is possible (audio > userinfo)

a.Common

- Logging :

  Module which trace every events done.

- Configuration :

  Component which handle load configuration from configuration file with YAML and the external library yaml-cpp.

1.Modules

a.Server

- UserModule:

  Backend retrieving user datas and authentication.

- ApplicationModule :

  Additional functionalities outside the application core.

a.Client

Functionalities outside the application core.

2.Les interfaces

The interfaces IapplicationModule and IuserModule force to follow constraint that modules have to respect in order to be used by the application.

3.External libraries

a.Boost

Powerful portable C++ libraries.
License : Boost Licence (~GPL).
More informations : http://www.boost.org/doc/

b.Boost ::threadpoool

Powerful portable C++ with use of thread pool
License : Boost Licence (~GPL).
More informations: http://threadpool.sourceforge.net/reference/annotated.html

c.Fmod

Powerful portable C++ for audio real time low latency input/output audio.
More informations : http://www.fmod.org

d.LibVorbis

Powerful portable C++ for manipulation, encoding, decoding of Ogg/Vorbis format.
More informations : http://xiph.org/vorbis/doc/

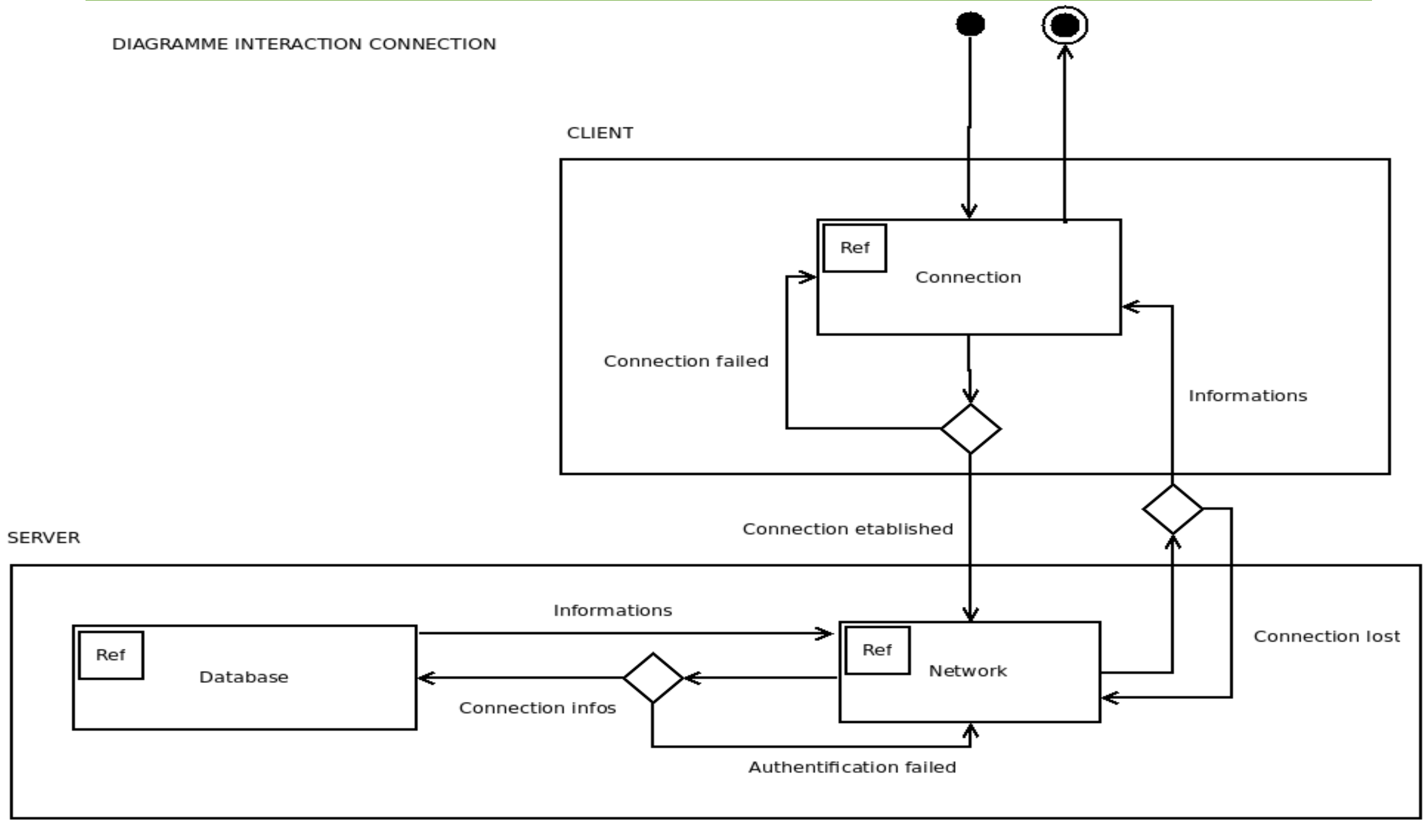
e.Yaml-cpp

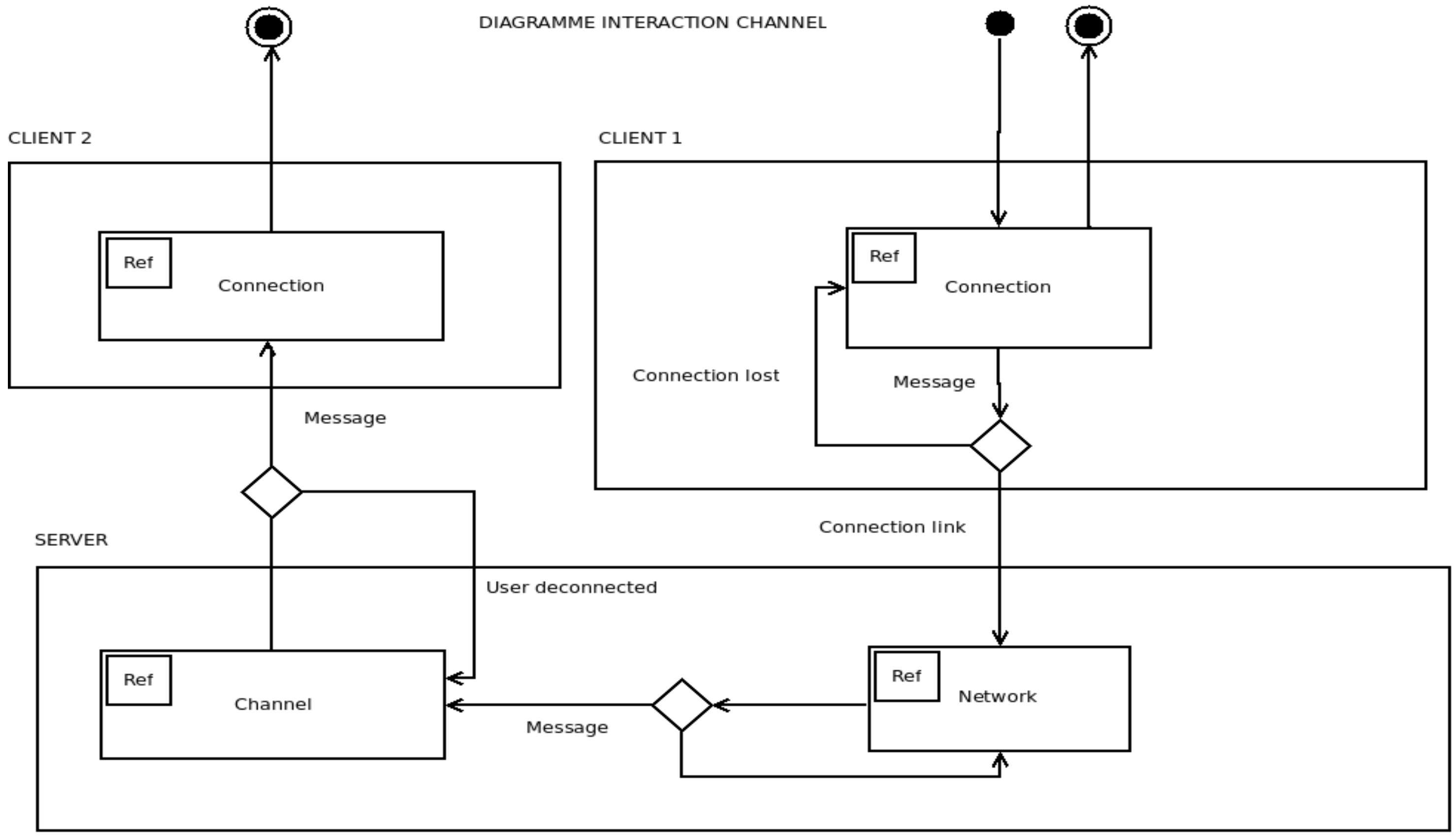
Parser and manipulator of YAML ressources.
More informations : http://code.google.com/p/yaml-cpp/

f.Mysql++

MySql++ is a C++ library for interacting with MysSql.
More informations : http://tangentsoft.net/mysql++/doc/

II.Interactions





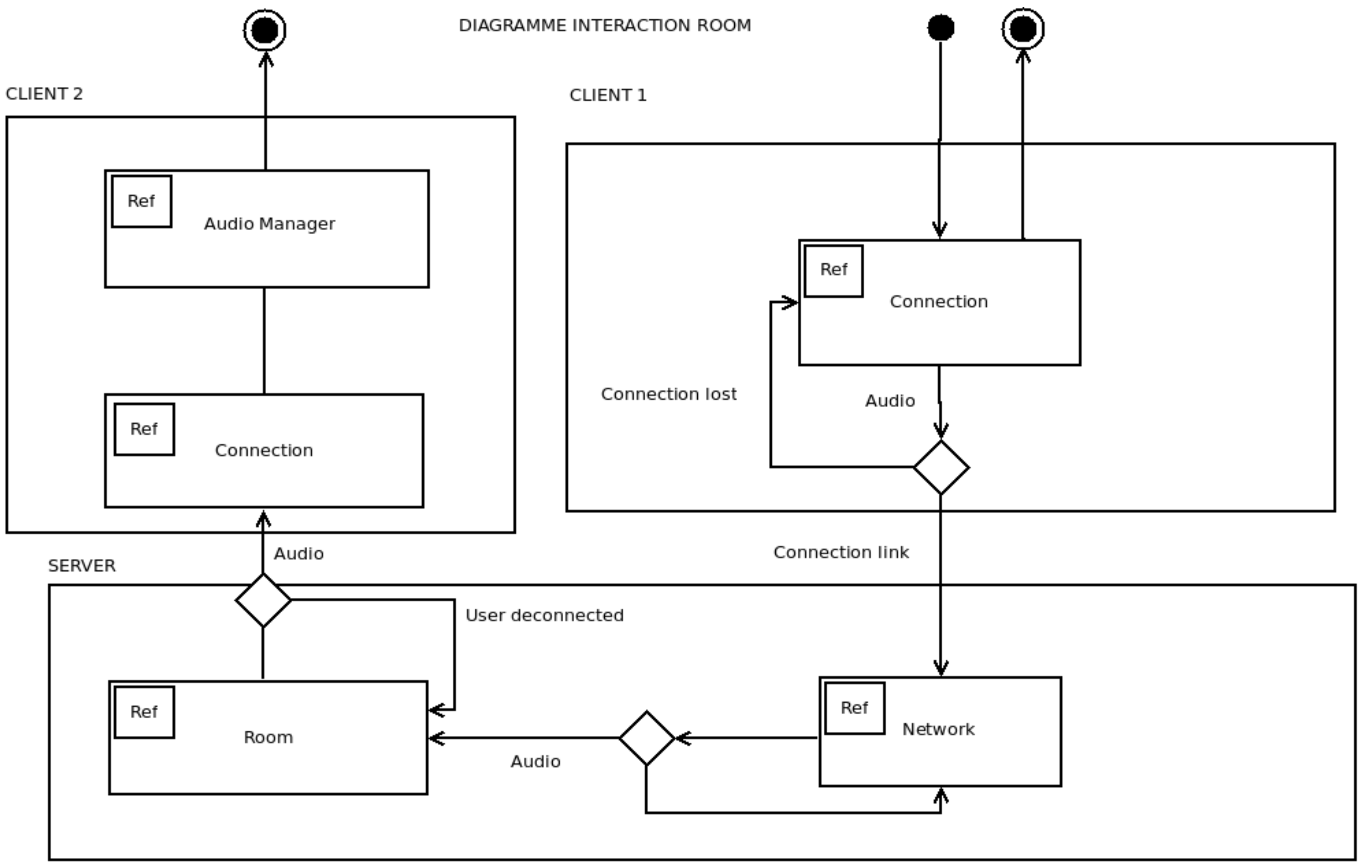


DIAGRAMME INTERACTION ROOM
CLIENT 2
Ref
Audio Manager
Ref
Connection
CLIENT 1
Ref
Connection
Connection lost
Audio
Connection link
Audio
SERVER
User deconnected
Ref
Room
Audio
Ref
Network

III.Live-Jamming protocol

UDP packet description :

```
-------------------------------------------- --------------------------------------------------------------------
| proto[4bit] | componentId[10bit] | requestId[6bit] |sessionId[32bit]  | datalen[12] |  |\\|  datas[...] |
-------------------------------------------- --------------------------------------------------------------------
```

Proto id :	version 1	1
Components id :	- session	1
	- channel	2
	- room	3
	- jam	4

Connection Steps :

Step 1: Client requests session with server:

```
{
 componentId :          SESSION_COMPONENTID          1

 requestId :            SESSION_AUTHREQUEST          1

 sessionId :            SESSION_ID                   0

 datas :          login
                  password
}
```

packet format example:

```
------------------------------------------------------------------------------------------------------------------------ -----------------------------------------------------------------
PROTO | SESSION_COMPONENTID | SESSION_AUTHREQUEST | SESSION_ID | DATALEN |  DATAS =  null_terminated_login,null_terminated_pass |
------------------------------------------------------------------------------------------------------------------------ -----------------------------------------------------------------
```

Step 2: Server informs client that session has been created:

```
{
 componentId :          SESSION_COMPONENTID          1

 requestId :            SESSION_AUTHREQUEST_OK       2

 sessionId :            SESSION_ID                   X

 datas :          user informations
}
```

packet format example:

```
------------------------------------------------------------------------------------------------------------------------ --------------------------------------------------
PROTO | SESSION_COMPONENTID | SESSION_AUTHREQUEST_OK | SESSION_ID | DATALEN |  DATAS =  null_terminated_userinfos |
------------------------------------------------------------------------------------------------------------------------ --------------------------------------------------
```

Step 2 (alt): Server informs client that session has NOT beed created, bad authentification:

```
{
 componentId :          SESSION_COMPONENTID                  1

 requestId :            SESSION_AUTHREQUEST_NOK_BADAUTH      3
 sessionId :                SESSION_ID                                   0
}
```

packet format example:

```
---------------------------------------------------------------------------------------------------------------------------- --------------------------------------------
PROTO | SESSION_COMPONENTID | SESSION_AUTHREQUEST_NOK_BADAUTH | SESSION_ID | DATALEN |  DATAS =  nul l |
---------------------------------------------------------------------------------------------------------------------------- --------------------------------------------
```

<u>Step 2 (alt): Server informs client that session has NOT beed created, authentification duplicate:</u>

```
{
 componentId :              SESSION_COMPONENTID                          1

 requestId :                SESSION_AUTHREQUEST_NOK_DUPLICATE            3

 sessionId :                SESSION_ID                                   0
}
```

packet format example:

```
---------------------------------------------------------------------------------------------------------------------------- ----------------------------------------------
PROTO | SESSION_COMPONENTID | SESSION_AUTHREQUEST_NOK_DUPLICATE | SESSION_ID | DATALEN |  DATAS =  nul l |
---------------------------------------------------------------------------------------------------------------------------- ----------------------------------------------
```

<u>Step 3 : Client disconnects from server :</u>

```
{
 componentId :              SESSION_COMPONENTID                          1

 requestId :                SESSION_DISCONNECT                           5

 sessionId :                SESSION_ID                                   0
}
```

packet format example:

```
---------------------------------------------------------------------------------------------------------------------------- -----------------
PROTO | SESSION_COMPONENTID | SESSION_DISCONNECT | SESSION_ID | DATALEN |  DATAS =  nul l |
---------------------------------------------------------------------------------------------------------------------------- -----------------
```

<u>Step 3 : Server disconnects from client:</u>

```
{
 componentId :              SESSION_COMPONENTID                  1

 requestId :                SESSION_DISCONNECTED                 6

 sessionId :                SESSION_ID                           0
}
```

packet format example:

```
---------------------------------------------------------------------------------------------------------------------------- ---------------------
PROTO | SESSION_COMPONENTID | SESSION_DISCONNECTED | SESSION_ID | DATALEN |  DATAS =  nul l |
---------------------------------------------------------------------------------------------------------------------------- ---------------------
```

Channel / Message Steps :

CHANNEL_ID =	16bit
CLIENT_SESSION_ID =	32bit

User Join Channel :

```
{
 componentId :          CHANNEL_COMPONENTID          2

 requestId :            CHANNEL_JOIN                 1

 sessionId :            SESSION_ID                   X

 datas :                CHANNEL_ID                   X
}
```

packet format example:

```
---------------------------------------------------------------------------------------------------------------------- ------------------------
PROTO | CHANNEL_COMPONENTID | CHANNEL_JOIN | SESSION_ID | DATALEN |  DATAS =  CHANNEL_ID  |
---------------------------------------------------------------------------------------------------------------------- ------------------------
```

User receive joined notification of a user in the Channel :

```
{
 componentId :          CHANNEL_COMPONENTID          2

 requestId :            CHANNEL_JOINED               4

 sessionId :            SESSION_ID                   X

 datas :                CHANNEL_ID                   X
                        CLIENT_SESSION_ID            X
}
```

packet format example:

```
---------------------------------------------------------------------------------------------------------------------- -----------------------------
PROTO | CHANNEL_COMPONENTID | CHANNEL_LEAVED | SESSION_ID | DATALEN |  DATAS =  CHANNEL_ID  |
---------------------------------------------------------------------------------------------------------------------- -----------------------------
```

User leave Channel :

```
{
 componentId :          CHANNEL_COMPONENTID          2

 requestId :            CHANNEL_LEAVE                10

 sessionId :            SESSION_ID                   X

 datas :                CHANNEL_ID                   X
}
```

packet format example:

```
---------------------------------------------------------------------------------------------------------------------- ------------------------
PROTO | CHANNEL_COMPONENTID | CHANNEL_JOIN | SESSION_ID | DATALEN |  DATAS =  CHANNEL_ID  |
--------------------------------------------------------------------------------------------------------------------------- -----------------------
```

<u>User receive leaved notification of a user in the Channel :</u>

```
{
 componentId :        CHANNEL_COMPONENTID          2

 requestId :          CHANNEL_LEAVED               13

 sessionId :          SESSION_ID                   X

 datas :              CHANNEL_ID                   X
                      CLIENT_SESSION_ID            X
}
```

packet format example:

```
--------------------------------------------------------------------------------------------------------------------------- ----------------------------
PROTO | CHANNEL_COMPONENTID | CHANNEL_LEAVED | SESSION_ID | DATALEN |  DATAS =  CHANNEL_ID  |
--------------------------------------------------------------------------------------------------------------------------- ----------------------------
```

<u>Send channel message :</u>

```
{
 componentId :        CHANNEL_COMPONENTID          2

 requestId :          CHANNEL_MESSAGE              6

 sessionId :          SESSION_ID                   X

 datas :              CHANNEL_ID                   X
                      MESSAGE                      X
}
```

packet format example:

```
---------------------------------------------------------------------------------------------------------------------------------------------------------
PROTO | CHANNEL_COMPONENTID | CHANNEL_MESSAGE | SESSION_ID | DATALEN |  DATAS =  CHANNEL_ID  / MESSAGE |
--------------------------------------------------------------------------------------------------------------------------- ---------------------------------------------
```

<u>Receive channel message :</u>

```
{
 componentId :        CHANNEL_COMPONENTID          2

 requestId :          CHANNEL_MESSAGE_RECV         8

 sessionId :          SESSION_ID                   X

 datas :         CHANNEL_ID                   X
                 CLIENT_SESSION_ID            X
                 MESSAGE                      X
}
```

packet format example:

```
------------------------------------------------------------------------------------------------------------------------------------ ------------------------------------------------------------------------------------------
PROTO | CHANNEL_COMPONENTID | CHANNEL_MESSAGE_RECV  | SESSION_ID | DATALEN |  DATAS =  CHANNEL_ID  / CLIENT_SESSION_ID / MESSAGE |
------------------------------------------------------------------------------------------------------------------------------------ ------------------------------------------------------------------------------------------
```

<<<<<STEPS TO IMPLEMENT NOT DECIDED YET>>>>>

User informations Steps :

Change status :

```
{
 sessionid :          sessionid

 type :               informations : status_changed

 datas :              status

 version :             proto_version
}
```

packet format example:

```
----------------------------------------------------------------------------------------------------------------------------- --------------------
sessionid | type = auth_request | datalen = ... | proto_version = ...| [datas] = null_terminated_login,null_terminated_pass |
----------------------------------------------------------------------------------------------------------------------------- --------------------
```

Get user profil :

```
{
 sessionid :                    sessionid

 type :                         informations : get_profil

 datas :                        user

 version :                      proto_version
}
```

packet format example:

```
----------------------------------------------------------------------------------------------------------------------------- --------------------
sessionid | type = auth_request | datalen = ... | proto_version = ...| [datas] = null_terminated_login,null_terminated_pass |
----------------------------------------------------------------------------------------------------------------------------- --------------------
```

Evaluate latency :

```
{
 sessionid :                    sessionid

 type :                         informations : evaluate_latency

 datas :                        latency

 version :                      proto_version
}
```

packet format example:

```
------------------------------------------------------------------------------------------------------------------------- -------------------
sessionid | type = auth_request | datalen = ... | proto_version = ...| [datas] = null_terminated_login,null_terminated_pass |
------------------------------------------------------------------------------------------------------------------------- -------------------
```

Room Steps :

Create room :

```
{
 sessionid :                    sessionid

 type :                         jam : create

 datas :                        room_name
                                room_settings
                                room_participants

 version :                      proto_version
}
```

packet format example:

```
------------------------------------------------------------------------------------------------------------------------- -------------------
sessionid | type = auth_request | datalen = ... | proto_version = ...| [datas] = null_terminated_login,null_terminated_pass |
------------------------------------------------------------------------------------------------------------------------- -------------------
```

Leave room :

```
{
 sessionid :                    sessionid

 type :                         jam : end

 datas :                        room_name
                                room_participants

 version :                       proto_version
}
```

packet format example:

```
------------------------------------------------------------------------------------------------------------------------- -------------------
sessionid | type = auth_request | datalen = ... | proto_version = ...| [datas] = null_terminated_login,null_terminated_pass |
------------------------------------------------------------------------------------------------------------------------- -------------------
```

Send invitation to room :

```
{
 sessionid :                    sessionid

 type :                         jam : invite

 datas :                        room_name
                                room_invited
                                room_invitation_message

 version :                      proto_version
```

}

packet format example:

```
----------------------------------------------------------------------------------------------------------------------------- -------------------
sessionid | type = auth_request | datalen = ... | proto_version = ...| [datas] = null_terminated_login,null_terminated_pass |
----------------------------------------------------------------------------------------------------------------------------- -------------------
```

<u>Receive invitation to a room :</u>

```
{
 sessionid :                    sessionid

 type :                         jam : invite

 datas :                        room_name
                                room_host
                                room_invitation_message

 version :                      proto_version
}
```

packet format example:

```
----------------------------------------------------------------------------------------------------------------------------- -------------------
sessionid | type = auth_request | datalen = ... | proto_version = ...| [datas] = null_terminated_login,null_terminated_pass |
----------------------------------------------------------------------------------------------------------------------------- -------------------
```

<u>Send kick from room :</u>

```
{
 sessionid :                    sessionid

 type :                         jam : kick

 datas :                        room_name
                                room_kicked
                                room_kick_reason

 version :                      proto_version
}
```

packet format example:

```
----------------------------------------------------------------------------------------------------------------------------- -------------------
sessionid | type = auth_request | datalen = ... | proto_version = ...| [datas] = null_terminated_login,null_terminated_pass |
----------------------------------------------------------------------------------------------------------------------------- -------------------
```

<u>Receive kick from a room :</u>

```
{
 sessionid :                    sessionid

 type :                         jam : kick

 datas :                        room_name
                                room_host
                                room_kick_reason
 version : proto_version
}
```

packet format example:

```
---------------------------------------------------------------------------------------------------------------------------- -------------------
sessionid | type = auth_request | datalen = ... | proto_version = ...| [datas] = null_terminated_login,null_terminated_pass |
---------------------------------------------------------------------------------------------------------------------------- -------------------
```

<u>Room settings :</u>

```
{
 sessionid :                sessionid

 type :             jam : settings

 datas :            room_name
                    room_settings

 version :          proto_version
}
```

packet format example:

```
---------------------------------------------------------------------------------------------------------------------------- -------------------
sessionid | type = auth_request | datalen = ... | proto_version = ...| [datas] = null_terminated_login,null_terminated_pass |
---------------------------------------------------------------------------------------------------------------------------- -------------------
```

<u>Jam Steps :</u>

<u>Start jam :</u>

```
{
 sessionid :        sessionid

 type :             jam : start

 datas :            room_name

 version :          proto_version
}
```

packet format example:

```
---------------------------------------------------------------------------------------------------------------------------- -------------------
sessionid | type = auth_request | datalen = ... | proto_version = ...| [datas] = null_terminated_login,null_terminated_pass |
---------------------------------------------------------------------------------------------------------------------------- -------------------
```

<u>Record jam :</u>

```
{
 sessionid :        sessionid

 type :             jam : record

 datas :            room_name
```

version : proto_version
}

packket format example:

```
-------------------------------------------------------------------------------------------------------------------------------- -------------------
sessionid | type = auth_request | datalen = ... | proto_version = ...| [datas] = null_terminated_login,null_terminated_pass |
-------------------------------------------------------------------------------------------------------------------------------- -------------------
```

<u>Stop jam :</u>
{
sessionid : sessionid

type : jam : stop

datas : room_name

version : proto_version
}

packet format example:

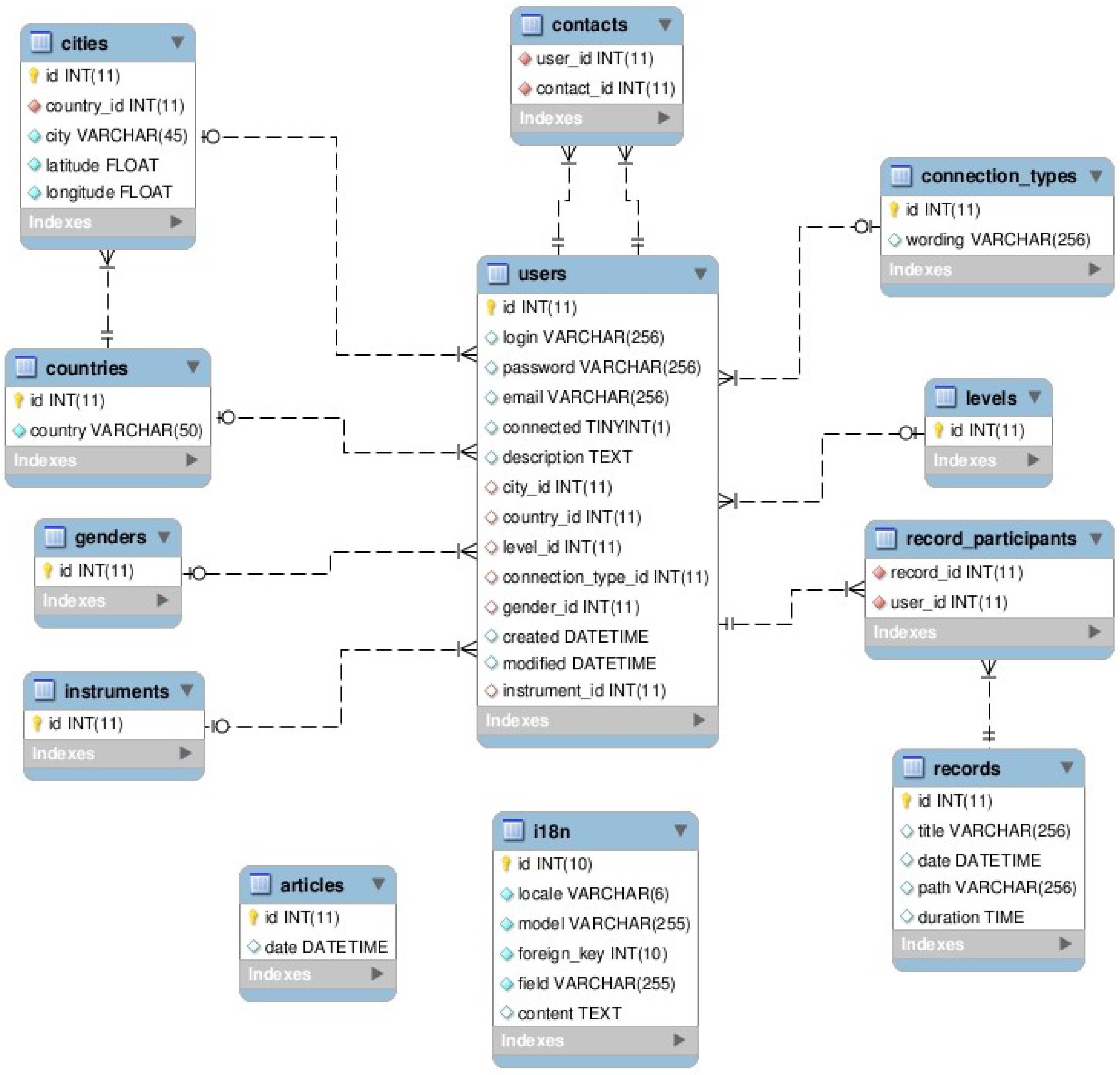
```
-------------------------------------------------------------------------------------------------------------------------------- -------------------
sessionid | type = auth_request | datalen = ... | proto_version = ...| [datas] = null_terminated_login,null_terminated_pass |
-------------------------------------------------------------------------------------------------------------------------------- -------------------
```

IV.Website / Database

1.Website

Realized with CakePhp Framework (http://cakephp.org/) available at http://www.live-jamming.com .

2.Database



cities
id INT(11)
country_id INT(11)
city VARCHAR(45)
latitude FLOAT
longitude FLOAT
Indexes
contacts
user_id INT(11)
contact_id INT(11)
Indexes
connection_types
id INT(11)
wording VARCHAR(256)
Indexes
users
id INT(11)
login VARCHAR(256)
password VARCHAR(256)
email VARCHAR(256)
connected TINYINT(1)
description TEXT
city_id INT(11)
country_id INT(11)
level_id INT(11)
connection_type_id INT(11)
gender_id INT(11)
created DATETIME
modified DATETIME
instrument_id INT(11)
Indexes
countries
id INT(11)
country VARCHAR(50)
Indexes
levels
id INT(11)
Indexes
genders
id INT(11)
Indexes
record_participants
record_id INT(11)
user_id INT(11)
Indexes
instruments
id INT(11)
Indexes
records
id INT(11)
title VARCHAR(256)
date DATETIME
path VARCHAR(256)
duration TIME
Indexes
i18n
id INT(10)
locale VARCHAR(6)
model VARCHAR(255)
foreign_key INT(10)
field VARCHAR(255)
content TEXT
Indexes
articles
id INT(11)
date DATETIME
Indexes