→ Force Last chunk

Force Game data [Meta] Last one

→ ~~Game~~ Freq = 50

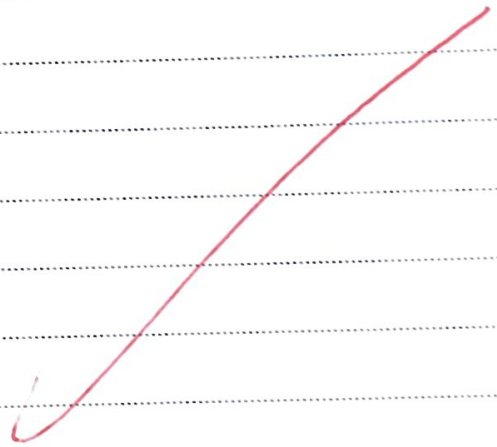
@ Pending X X

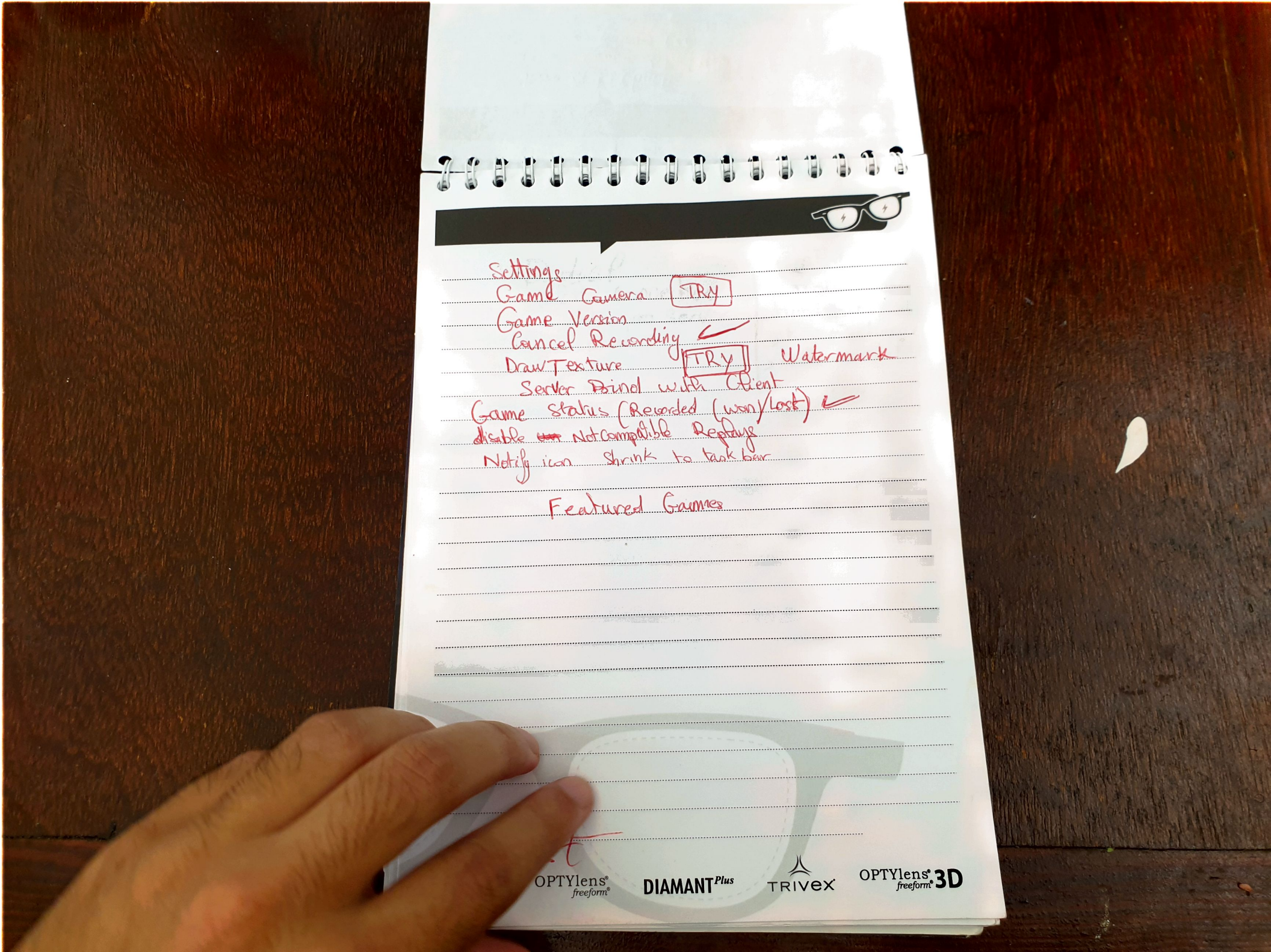
→ if

→ get Last chunk info (duration)

→ Start chunk

→ end chunk

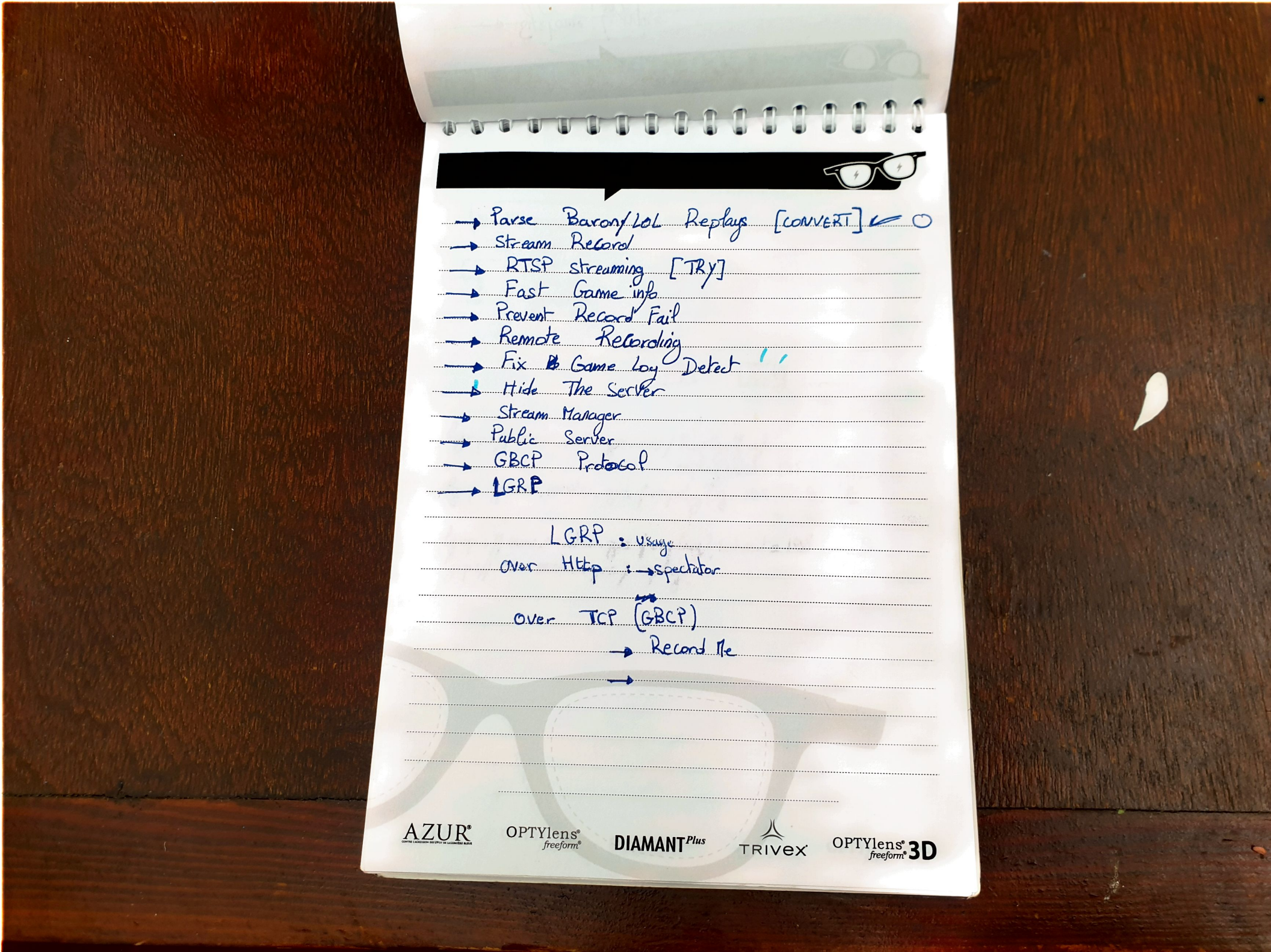


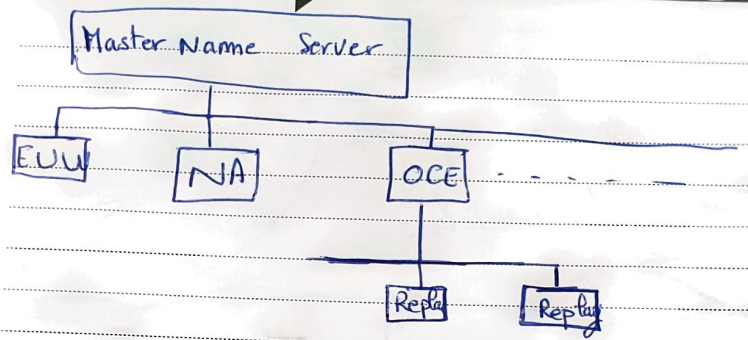

Settings
Game Camera
TRY
Game Version
Cancel Recording
Draw Texture
TRY
Watermark
Server Bind with Client
Game status (Recorded (won/lost)
disable ~~not~~ Not compatible Replays
Notify icon shrink to task bar
Featured Games
OPTYlens freeform
DIAMANT Plus
TRIVEX
OPTYlens freeform 3D

Protocol
P2P connection
Stream server
chat

GOSTBLADE
Protcol
→ Handshake
→ Key Exchange
→ Communicate
→

- Système Logique
- Eq diff
- Thermodynamique
- Electromagnétisme
- Electronique


Parse Baron/LoL Replays [CONVERT]
Stream Record
RTSP streaming [TRY]
Fast Game info
Prevent Record Fail
Remote Recording
Fix Game Log Detect
Hide The Server
Stream Manager
Public Server
GBCP Protocol
LGRP
LGRP : usage
over Http : → spectator
over TCP (GBCP)
→ Record Me
→
AZUR
OPTYlens freeform
DIAMANT Plus
TRIVEX
OPTYlens freeform 3D



lgrp://euw.master~~s~~.grs/~~Replay?info~~=
replay?stat=198...9...

lgrp://euw.master.grs/replay?g=198...9...

lgrp://euw.master.grs/replay?submit=198...9...
lgrp://euw.master.grs/replay?

→ Logging

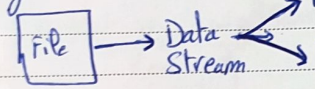
→ get Ld properties / get league of Legends.exe

→

Read

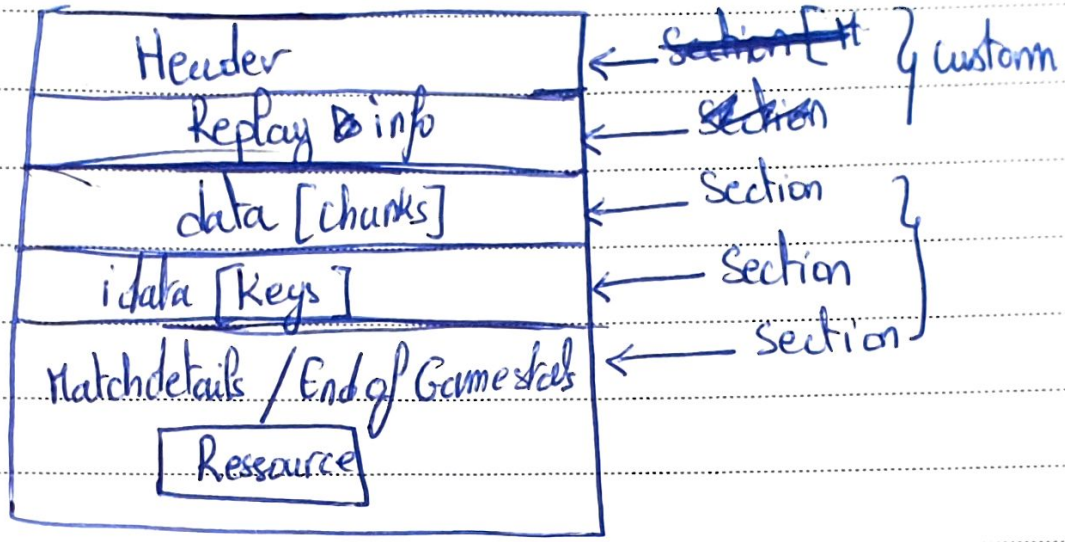
→ Read Header : Read Head
→ Read Limited : info + Head
→ Read Full : Read all

Reading :



→ Replay Header
Type : Signed / Unsigned (1)
~~Encrypted : Algorithm ?~~
Signature : bytes [RawEntry]
Certificate : bytes [RawEntry]
Version : Recorder_Version (1)
Application : Recorder [1]
MatchHistory : boolean
Spectator : boolean
EndofGamestats : boolean
ClientVersion : int / long
ObserverKey : Raw Req data

File structure



Header

Header : Replay Header
- Version : byte, Application : byte
- Signature : Raw Entry
- Certificate : Raw Entry / Server Version

Replay Info :
- ClientVersion : Long
- Spectate
- Match History } 1 byte
- EOGS
- dokey : Raw Entry

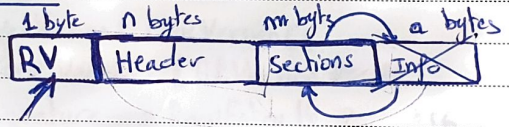
Replay Sections

Replay Structure

Necessary : ob-key
Keys
Chunks
Last Chunk infos
Metadata
} server / client

→ end of Game stats : client

Definition



RV : ReplayVersion
Standard

Replay :
RV : Replay Version
Header : Replay Header
~~Info : ReplayInfo~~
Section : ReplaySection[]

ReplayVersion : [1 byte]
Standard = 0, GZIP = 1, DEFLATE = 2
LolReplay = 3, BaronReplay = 4, Unknown = 5
Encrypted, Signed = 7, Unsigned = 8,

Replay Header
ServerVersion : LGRVersion64
ClientVersion : LGRVersion ~~32~~ 24
Signature : RawEntry16 } 256
Certificate : RawEntry16 } 1500
~~Replay Information~~
obKey : RawEntry8 + 3

Sections
→ ~~Bad~~ChunkInfo : Section
→ Metadata : Section
→ EndOGS : Section
→ Match Detail : Section ? [optional]
→ Chunks : Section
Keys : Section

LGRVersion64 :
- ~~Length~~ VersionBase : long
  - Major : Usort
  - Minor : Usort
  - Build : Ushort
  - Revision : Ushort

LGRVersion24
- VersionBase : byte[3]
  - Major : byte
  - Minor : byte
  - Build : byte

RawEntry8
- Length : byte
- Data : byte[]

RawEntry16
- Length : ushort
- Data : byte[]

RawEntry32
- Length : Uint
- Data : byte[]

Section :
- ~~Section~~ Type : SectionType
- CountBytesLength : Uint
  - Data : byte[]
  - or { Entries : RawEntry[]

SectionType

→ Int 24

{ HEADER } Uncompressed

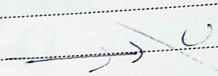
{ SECTIONS } Compressed

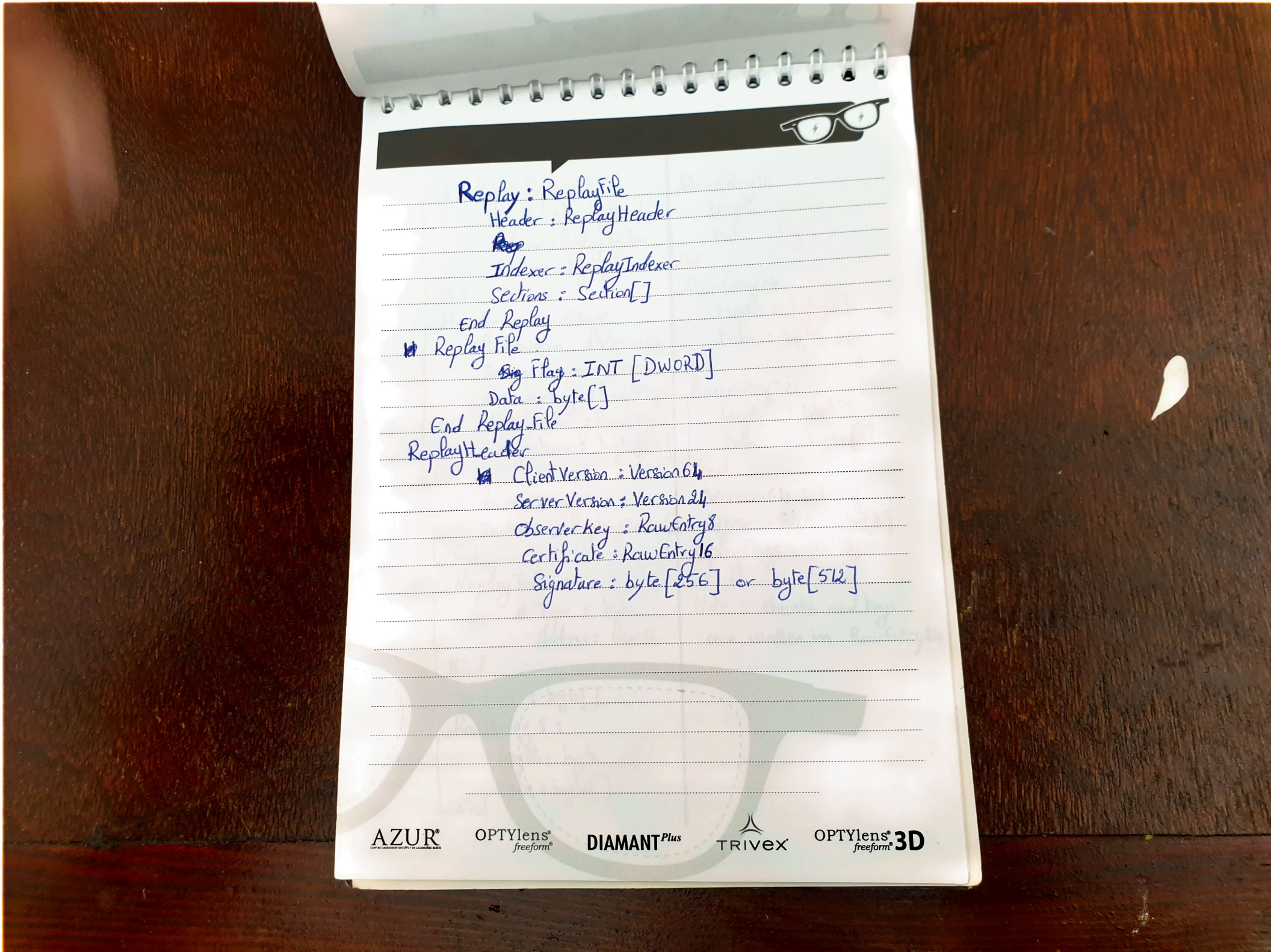
→ Update via SVN [GITHUB]
→ Ghostreplays server
→ Svn Update Checker
→ Installer
→ Updater [Reinstall / install shell extension]
→

32

Big	Header	Sections

Signature : int
Header




Replay : ReplayFile
Header : ReplayHeader
Indexer : ReplayIndexer
Sections : Section[]
End Replay
Replay File
Flag : INT [DWORD]
Data : byte[]
End Replay_File
ReplayHeader
ClientVersion : Version64
ServerVersion : Version24
Observerkey : RawEntry8
Certificate : RawEntry16
Signature : byte[256] or byte[512]
AZUR
OPTYlens freeform
DIAMANT Plus
TRIVEX
OPTYlens freeform 3D

Replay : ReplayFile
  Header : ReplayHeader
  Indexer : ReplayIndex
  Sections : byte[]
End

Header : RawEntry16
  ClientVersion : Version64
  ServerVersion : Version64
  observerKey : RawEntry8
  Certificate : RawEntry16
  Signature : byte[256]
End

Indexer :
  Indexes : ReplayIndex[]
End

ReplayIndex :
  Type : byte
  Address : Uint32
End

Section : byte[]

RawEntry8 :
  Length : byte
  Data : byte[]
End

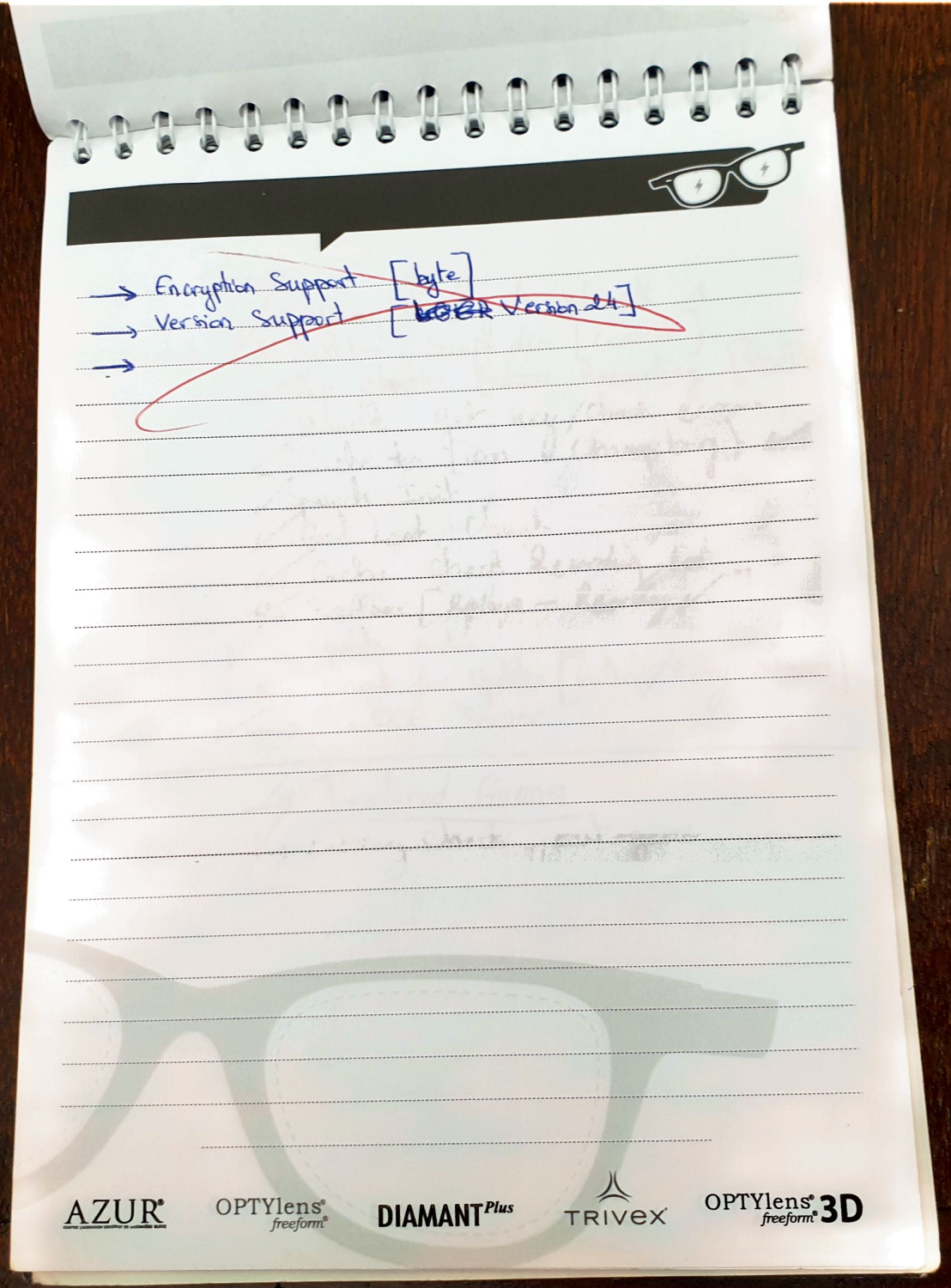
RawEntry16
  Length : Uint16
  Data : byte[]
End

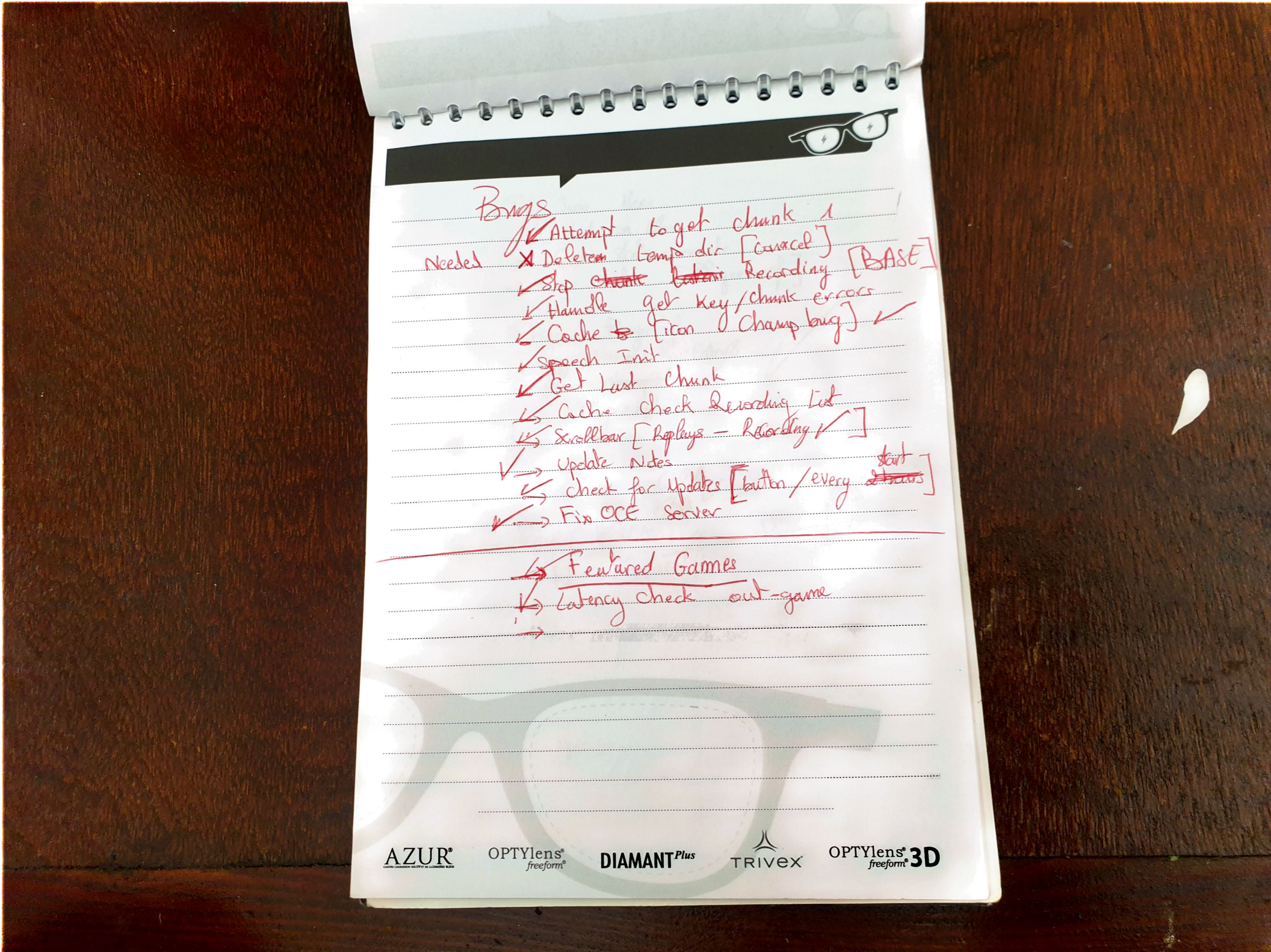
RawEntry24
  Length : byte[3]
  Data : byte[]
End

RawEntry32
  Length : Uint32
  Data : byte[]
End

Version64 : Long
Version24 : byte[3]
ReplayFlags : Uint32

Note : chunks and keys are written in RawEntry24


→ Encryption Support [byte]
→ Version Support [Version 24]
→
AZUR
OPTYlens freeform
DIAMANT Plus
TRIVEX
OPTYlens freeform 3D


Bugs
Attempt to get chunk 1
Needed
Delete temp dir [cancel]
Stop Recording [BASE]
Handle get key/chunk errors
Cache [icon champ bug]
Speech Init
Get Last Chunk
Cache check Recording List
Scrollbar [Replays – Recording]
Update Notes
check for Updates [button / every start]
Fix OCE server
Featured Games
Latency check out-game
AZUR
OPTYlens freeform
DIAMANT Plus
TRIVEX
OPTYlens freeform 3D

→ Tray Icon Menu
→ Hide on Close
→ Game Stats override
→ New Section. [Player info]

Section: Player info
Player Name: Raw Entry8
~~Skin~~ ~~Champid~~
Map: byte
}

OFF
~~ON~~

Buy Premives Random
Controll

Test

Replay [/]
Record [/]
Server Support

Server	Record	Replay
TR / RU / KR	/ /	/ /
OCE / NA / EUW	/ /	/ /
EUNE / ~~BBE~~	/ /	/ /
BR / LAN / LAS	/ /	/ /

→ Log :

→ Crash Report :

→ Shell Ext :

→ Generate Cert :

TRANSLATION

FR / DE / ESP / IT /

TR / RU / ~~DE~~ KR / AR / CN

~~Soon~~ Beta Tester Report

→ Cancel Recording seems to be broken

→ New Update Server

→ Default API key

→ Disable [Normal Updates Shell Ext]

→ Disable

→

Beta Release

INTERR_NULLABLE
GENERIC_DIMENSION
LITERAL
GENERATECOMPLETION
OPEN_BRACKET_EXPR
INTERR_OT

replace (open parens any)
remove anybut
replace <this accent> <that accent>
replace <thing or text> ?

R#

Goal: Mix of C#, D, C++, Python, Java...

Architecture: Multiple

~~Struct~~ Type: Native

Syntax: primary (C#), secondary (D), T (Java)

R#

Platform : ~~.NET~~

Influenced By : C#, Java

Features : Combines C#, Java, C++, D Python, Ruva

Target Architecture : ~~Assembly~~ 8086 / x86 / x64

=> Target Assembly : NASM

Problems & Challenges

Supporting

→ Generics : different type size.
→ Variadic functions
→ Exception Handling
→ Cross-platform implementation
→ ~~Ha~~ Type Checking

Objectives

Anonymous functions.
Nested functions
Generics (functions, Types)
Exception Handling
Anonymous Types
Lambda expression
Inline assembly
Object oriented
Procedural
Statements (: with, safe, sync) exit, loop, next, restrict
Operator Definition, overloading
Arrays
Tuple
Properties (getter, setter)
Extension methods
Reference Types
Initializations : Structs, arrays, blocks
Static methods, variables
Struct/Union methods
Alias (typedef /methodde/)
Slope import

Literal ~~prefix~~

Literals :

~~ref~~ Nullable type is a reference

int? => size = ~~4~~ sizeof addr

long? => size = " " "

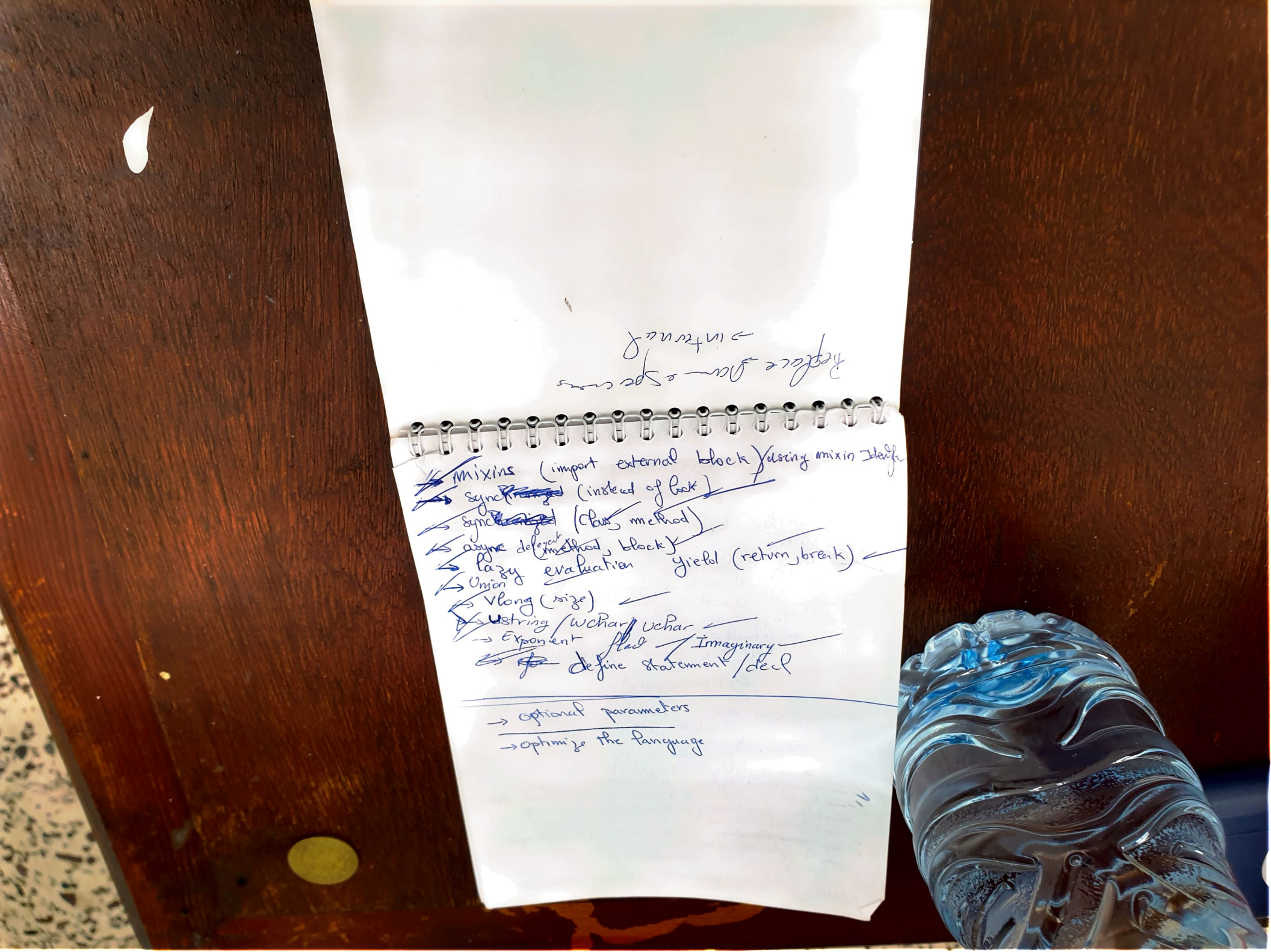
int? a;

a = 5; ⇔ [a] = 5;

~~x~~ = a; ⇔ x = [a];

x = a ?? 1 ⇔ if (a == null)
x = 1
else x = [a];

→ Complex type
→ wstring
→ Lazy Evaluation
    1) Create an anonymous subroutine
    2) add lazy code
a lazy code is a delegate
exemple:
log(lazy string a) { write(a()); }
log(tostring(i));
<=> ano__(int i) {
    return tostring(i);
}
log(anon__);
a is delegate string(int);
→ anonymous types declaration
→ const functions (doesnot modify external state)
→ inline functions (code always imported)
→ reference return
→ ~~auto evaluation~~ safe context (no pointer used)
→ Compiler Attribute
→ template constraints


Mixins (import external block) / using mixin Hello
sync (instead of lock)
sync (class, method)
async delegate (method, block)
lazy evaluation yield (return, break)
Union
Vlong (size)
Ustring / Wchar / Uchar
→ Exponent
for define statement / decl
→ optional parameters
→ optimize the language

Anss Lensoft
2012 - 2013

- Projects Definition
- Research

Writer:
Anssen Ibadi

Developer Studio 2013
Final Release V2.0

→ types / variables, func and procs to Intellissence ✓
→ Unit Decompiler or type extractor ✓
→ Unit Compile ✓
→ ~~Com extractor~~
→ office 2013 style + (Change the hole GUI)
→ Documentation library (ASDN)
→ ~~Halt = language~~
→ ~~Reference Manager~~
→ Program Manager
→ Signing of ~~Check security~~
→ ~~Program verification~~
→ ~~News downloader~~
→ ~~Syntax coloration~~ modification ✓
→ add description to main references
~~Ext Debugger~~ → ≃ Debugger plugin
→ Plugin interface.
→ Fix error ✓
→ Secret Manager
→ Resource editor ✓
→ code version form ✓
→ Compile Resource. ✓

- Compilation + Release options ✓
- ~~Form designer~~ (if it can) ✓
- Recently opened

Development

→ Create the GUI ✓ ~~[illegible]~~
→ Create the code [illegible]
→ Verify [illegible]
→ add modification 1.0.0
→ ~~[illegible]~~ Build

Release

Signing ✓
~~[illegible]~~
Documentation

Release :

Media fire
Home page
Face book page
Inject Ad

Developer Studio 2013
Feature list

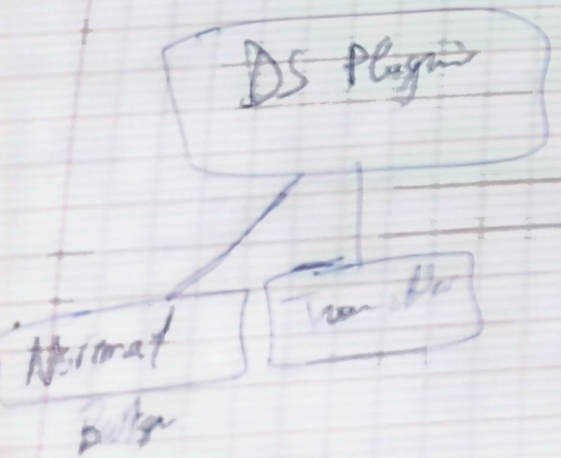
→ New user interface
→ Speech ~~[illegible]~~ Service
→ Intellisense
→ [illegible] units
→ Unit ~~Symbol~~ extender
→ Syntax colorisation
→ Publish
→ Samples
→ Documentation
→ Ressource Implementation

Translate : [illegible] + Product
Feature

DS 13 x86 x64

3.0 F.Rel

- → Windows 64 compiler ✓
- → Remove MinGW ✓
- → Nospace intellisense ✓
- → ARM-wince compiler ✓
- → Setting environment ✓
- → Plugin Manager ✓



Developer Studio 2013

V3.0

Final Release

- ~~Developer Network~~ ✓
- MSDN Documentation library ✓
- Cross Compilation (wince arm, Win32, Win64) ✓
- Integration
- Plugin Manager ✓
- ~~Debugger~~ (Win32 or Win64)
- Publish options ✓
- Error manager ✓
- First Run Manager (play audio, playIntro)
- Fast Initialization => ASync
- ~~Intelligent keyword Recognizer~~ (Plugin)
- ~~Secure Code extention~~
- Password Manager (Random password generator) ✓
- Program Trust Checker (Virus Total)
- Updates Manager (Plugin seeker, Update seeker,
- Fix Bugs and link catch → Error report
- ~~Set output directory for executable~~
- Create Rc. + Compile Rc
- ~~KeyBoard Shortcuts~~ ✓

- ~~Digital Clock at Startup Screen~~
- ~~Chat Toolbar (right)~~
- ~~Admin Panel~~ Page view (edited Chat Rules)
- Multi-lang (fr - en - ca)
- ~~Voice Call~~
- Ressource editor
- ~~Multi-extention editor~~
- Document Namespaces
- Default Included ~~Interface~~ Namespaces.
- ~~Adding~~ New Namespaces. (optim)
- New about form
- Developer Home form
- Developer
  - Touch Keyboard + OnScreen
  - Basic Definition editor
  - Scheduler
- Code Template
- ~~Compiler Template plugin~~
- Pascal Translator plugin
- ~~Reference page~~

Developer Network

Textual-chat

DSTP

DS Text Transmission Protocol

4	8	
CRC32	CMD	Number
data		

2048 bytes

Audio-chat

DSATP

DS Audio Transmission protocol

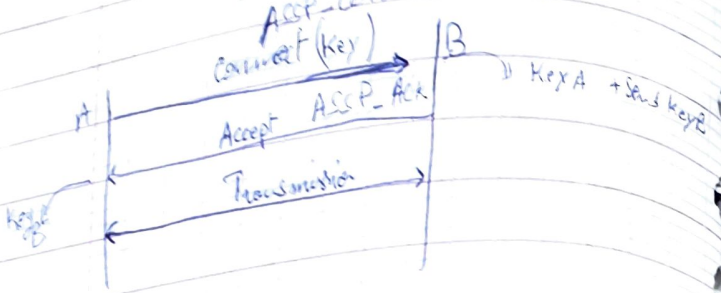
4	
CRC32	
data 6396	

16388

Security

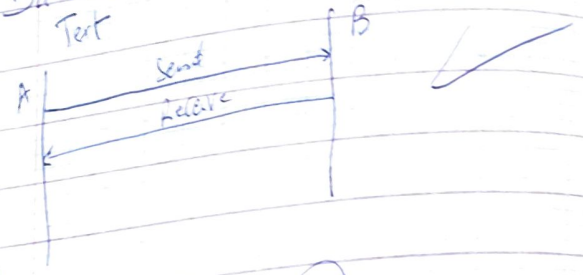
ASSP

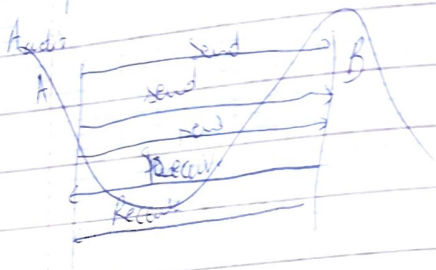
Assistant Secure Stream Protocol



Data Transmission

Test





Developer studio 2013

Final Devsteps

Dev:

- ~~Developer Home~~
- ~~Update Manager (every day / once)~~
- ~~Font Init~~
- ~~Font Buffer~~
- ~~Font Run Handle~~
- Reflection (Reflect Parser)
- Relation Based Graph
- ~~Task Scheduler~~
- ~~Multi Execution~~ Splash Screen Test

VI API

Test:

- Windows 32 : Compile
  - XP
  - 7
- Windows 64
  - ~~XP~~ 7
- All functions

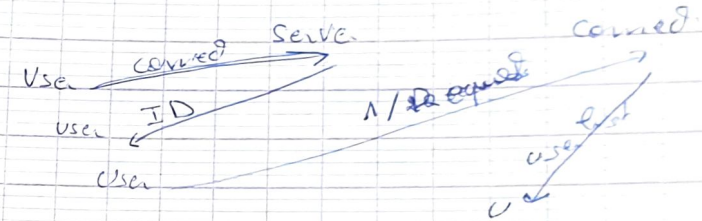
Publish
- Installer
- Links
- Upload Mediafire
- ~~Youtube~~ Intro ~~Trailer~~
- Forehand page
- Copy right

Features:
- → Intellisense
- Fast
- ~~Multi-Threaded Compilation~~ Cross Compile
- 8087 unit
- Documentation library
- Plugin Integration
- Debugger Console
- Task Scheduler
- Amazing GUI
- Secure
- Program Trust Checker
- Touch KeyBoard

Developer Network Specification
Plugin

→ Chat-

~~Rejected~~



Transfer-
Pascal Code
Text

Axsslensoft
Website

- Andromeda ✓
- chat Module ✓
- Forum, Blog, article ✓
- Mobile Switcher ✓
- CCK ✓
- Services ✓
- Translate ✓
- other Module 2 ✓

DS Rey

- .Net 3.0
- Dotnet Bar 10.8
- SAPI 5.1
- ~~MSAgent~~

Website Management

- Change Prod page
- Change certificate (SSL, Code signing, ~~Timestamp~~)
- originsign page
- Developer Studio 2013 page
- Kavprot page
- IPDTP page
- developer network
- ~~Firewall page~~
- LHPCC page (Invariom)
  - Change about
  - Change Asearch theme
  - add license to registration

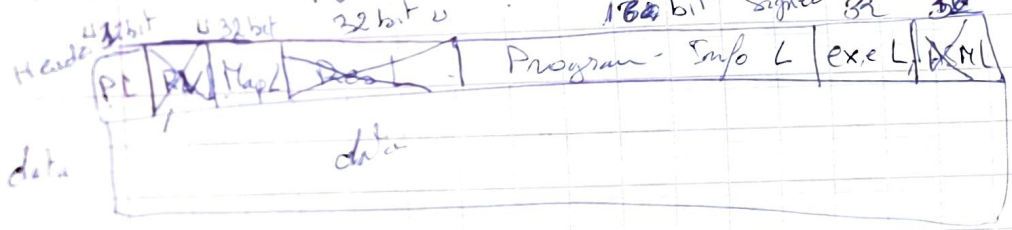
DS Package file

- Header
- PL
- REV
- Map
- Res
- PGuid
- exe
- asm

DSP
- Pascal source
- ~~version~~ + revision
- ~~dependencies~~ Map
- Resource
- assembly information

assembly info

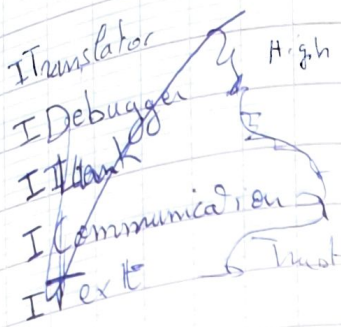
```
<pg version="1.0" >
  <permission require admin="val" />
  <version Major="" Minor="" Build="" />
  <output name="" path="" type="console" />
  <arch proc="Win32" debug="true" />
</pg>
```



- dsp

if (PIL is neg)
  decrypt(dsp)

DS Plugin Sys: Rev 2



Assistsoft SDK 1.0 for DS

Crypt/Authenticode (signtool, certmaker, encryption/~~decryption~~)
dc (documentation generator)
Dev (gacutil, certmgr, ~~C# compiler, NASH~~)
Appu Discover, Reflector, Intellisense generator
+ other ~~Plugin~~ / TaskScheduler /

other Tips

New (desktop context menu edit
open with shell
→ plugin Tab

Final Steps

→ optimize
→ Translate FR, EN, DE, AR
→ Video → code Simple pascal
→ Reflection fix
→ font error fix
→ remove removed types
→ ~~Splash~~ Clear pascal Template
→ all Begin, end to lower
→ parser error
→ ~~No~~ speech engine
→ Audio messages X
→ Local library
→ Refresh intellisense
→ Fix Syntax Highlight
→ Remove arm wince

Forms
~~Main About Settings, splash screen, Main, Plugin~~
~~Project Splash~~
CLI
Fix doc style path
Intellisense cert
Shell Menu

D$ Local

→ Translate buttons, Items, ok message

- Messages Local
- Settings Local
- Main Locale
- Global Local

Test: open file, help, create file, compile, run, config, Type declare, remove, intellisense

try

→ Type
→ Var
→ instance space
→ fonction

Erreur fatale

Init

Init compil.

Init speech

Init Intellisense

Init form

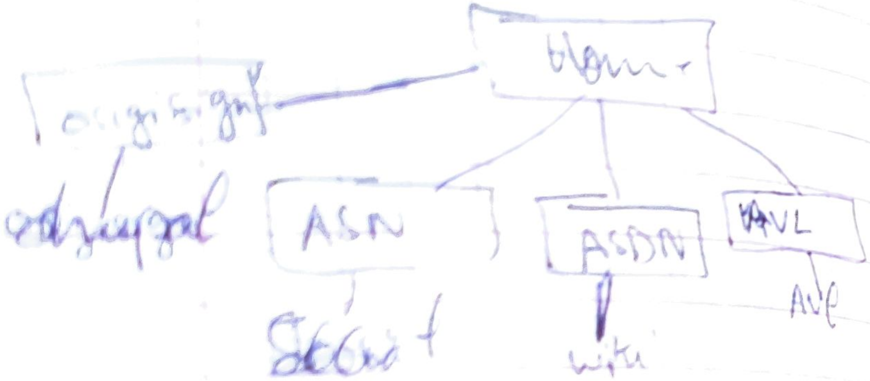
CA, Aide, ID, Key, Roles, OID, EV, SSL, ...

[time stamp], Serial generation

~~[illegible]~~ Netscape,

Amslersoft website

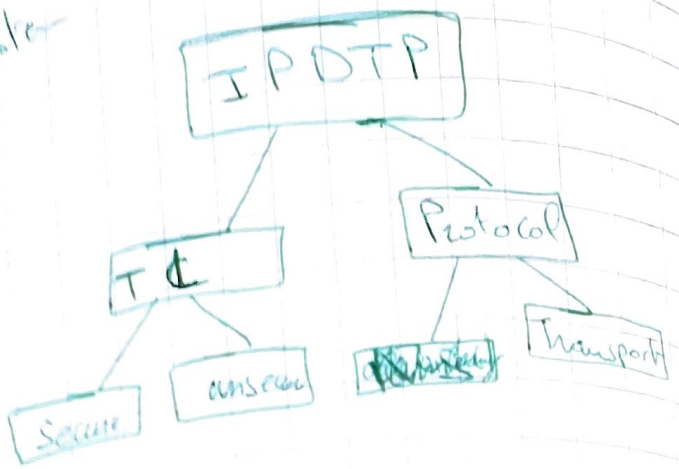
~~[illegible]~~ Schem.



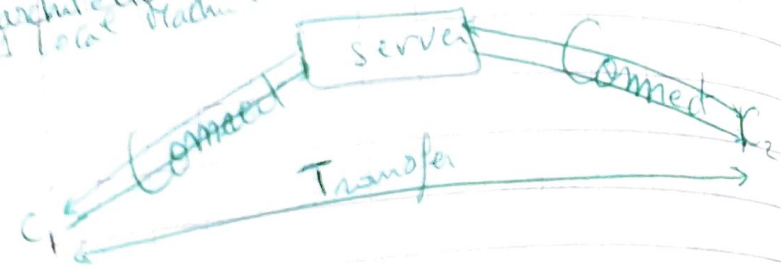
~~Ione~~

IPDTP

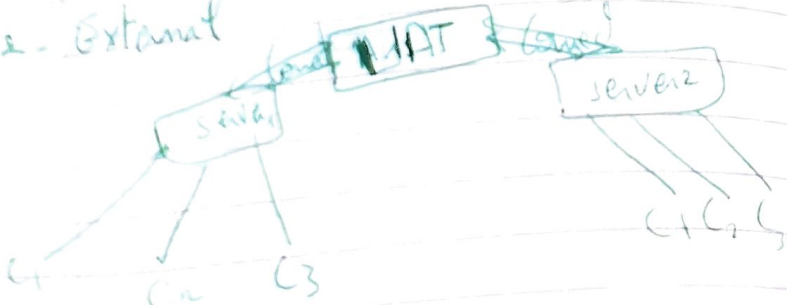
I Algèbre



II Architecture :
1 Local Machine



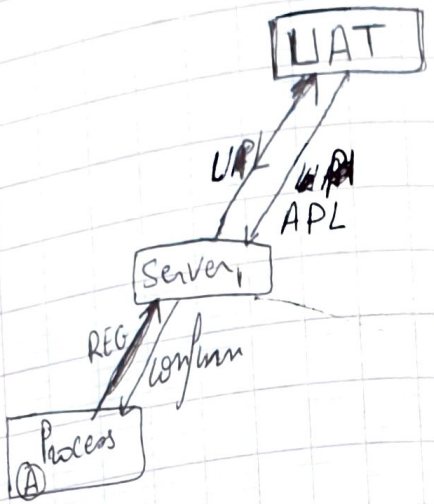
2. External



→ The IPDTP server connects to the UAT servers to ~~translate~~ get a new ~~uid~~ ~~uniform Add locator~~
→ the clients also connects to the local server to get new UpL and LAA

- UpL : Uniform process Locator
  upl : // ~~process~~ server . pid
- UPCSL : uniform Procès communication server locale
  upcsl : // server_id , counter
- PCSA : process communication server address
  2 bytes , 2 bytes , 2 bytes
- LAA
  byte . byte . byte

IPDTS

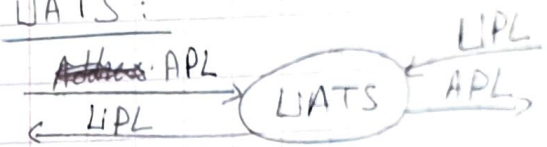


IPDTS : Interprocess Data Transmission System is an organised network composed from interconnected processes to make communication safe and secure also to Ensure the Transfer Integrity.

IPDTS Composed from

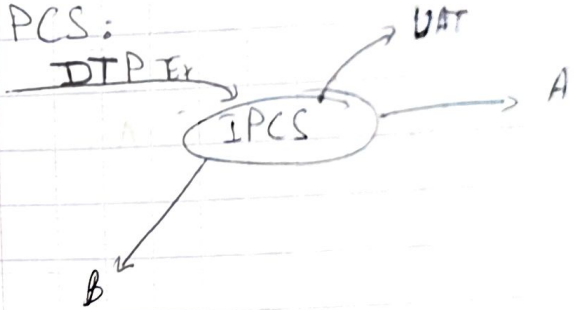
- UATS : Uniform Addressing Translation Server
- IPCS : InterProcess Communication Server
- SDTS : Secondary data Transmission Server
- ...

UATS :



it : REGISTER, GET (GET, SET)

IPCS :



it : accepts external requests, send requests to external fichiers

LIAT Table

LIPL

APL

IP.Port:PID

eg:

41.224.192.55:55

69.253.1.252:55

eg:

IDADI-PC . AE.Karpat.TN

ARSSLEN-PC . AE.Karpat.US

Subdomain

domain

LIAT Table

LIPL

APL

IP:Port:AID

app . Machine . Activity country

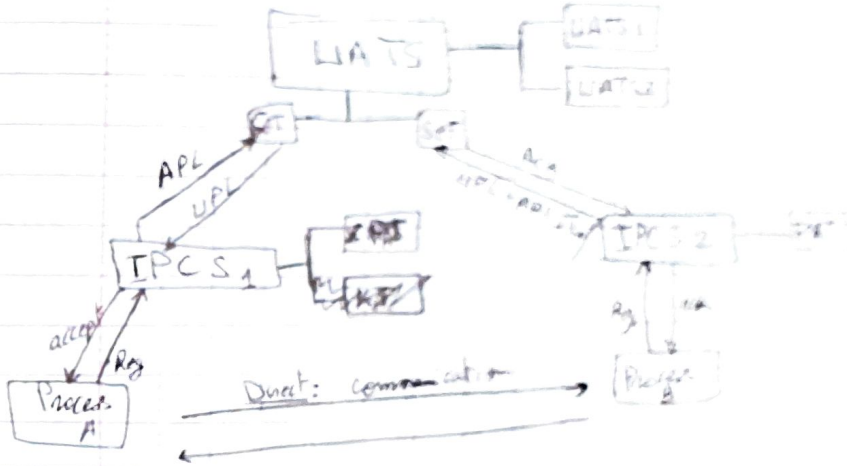
e.g

Karpat.Arslen-PC.AD1.TN

e.g 41.224.192.55:350:1

Protocol System Description

I Schematic

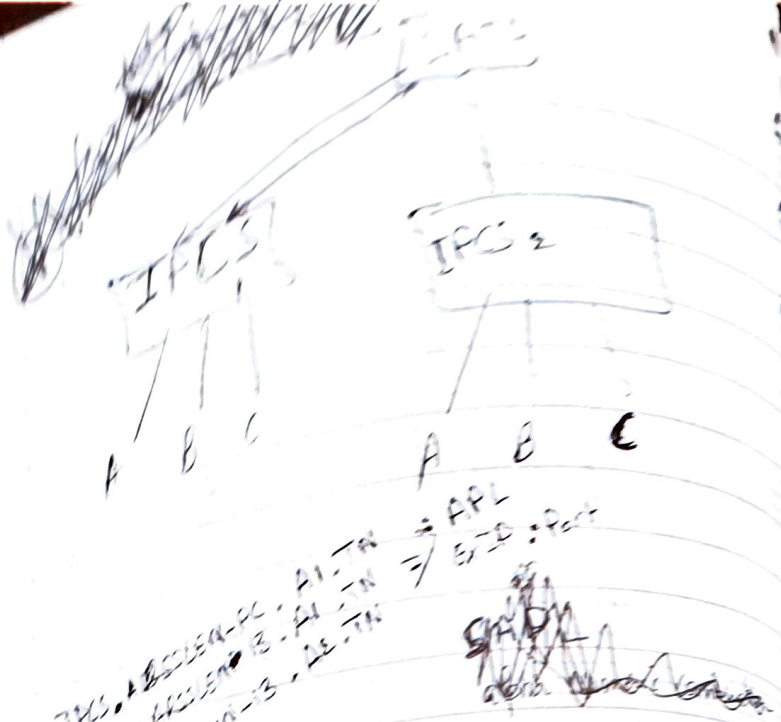


II Design

Level 1 IPCS → IPCP

Level 2 LIATS → LIAP

Level 3 Protocol



Analysis

→ each server have it's own address registered in the UATS, this address can be static or dynamic, dynamic addresses are changed @ every connect operation. static addresses are fix and defined by Administrator.
each address is composed from

S: for static

Application . Machine . Network . Country

Interface for static

USL:

ASL: External IP: Port

→ the Translation occurs in the UATS a database engine Looks for a USL or a ASL and return a response

UATS commands and UATP (Post request over http)

REGISTER , IDENTIFY, TRANSLAT

↳

Command	ASL	App Name	Machine Name

UATP

IDENTIFY or TRANSLAT

Command	USL or ASL

→ In the Translation Registration the Local Router must be Setup.

→ a LAN can contain from 1 to Machine Number of IPCS, each IPCS is a Local router or Address Translator : it Translate LIPL to APL, the IPCS identified by (LIPL : IPCS1. Machine.PC , APL : localadr (32bit) , Localport (16bit)

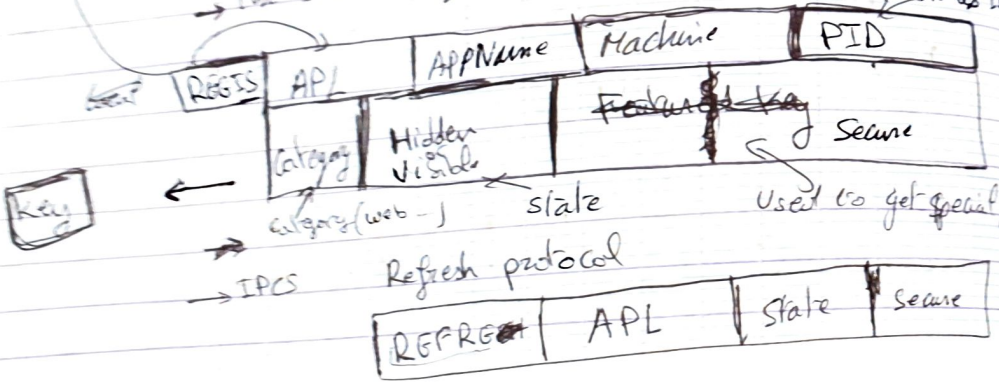
the processe are identified by (LIPL : Name. Machine. category

APL : local adr (32 bits) , local port (16 bits)

the IPCS contains a PAT (Process Address Table)

this Table contains all APL, LIPL of connected proceses and is refreshed every 5 mins (after connection or last refresh)

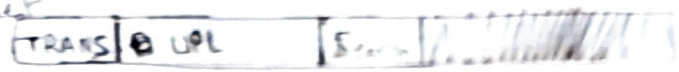
→ the IPCS Registration and Handshake protocol.



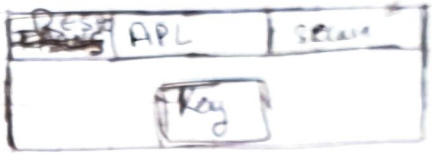
→ IPCS Refresh protocol

→ IPCS Response of translation (Name / Ident)

Request



Response

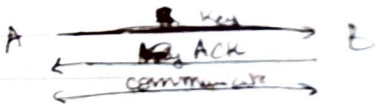


IPCS - Protocol

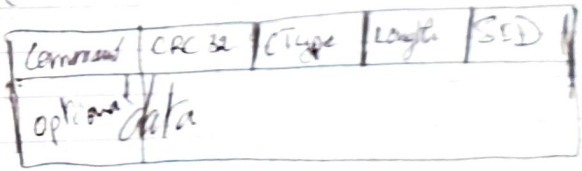
commands : REGIS, REFRE, TRANS, RESPONDENT

finally : IPDTP-C

Hand shake



Protocol



HSK

→ if command is ~~Handshk~~ Handshake then the data will be the key content and the communication information.

The data are Transmitted as below

A —— send data ——→ B

A ←—— Receive data —— B

→ if command is TAK the data will be variable from STRING, IMAGE, SOUND, or Data

→ if command is ~~SEC~~ SEC the data will be encrypted by the Receiver by the Handshake key.

→ if command is DEC, the data will be decrypted by the receiver the Handshake key

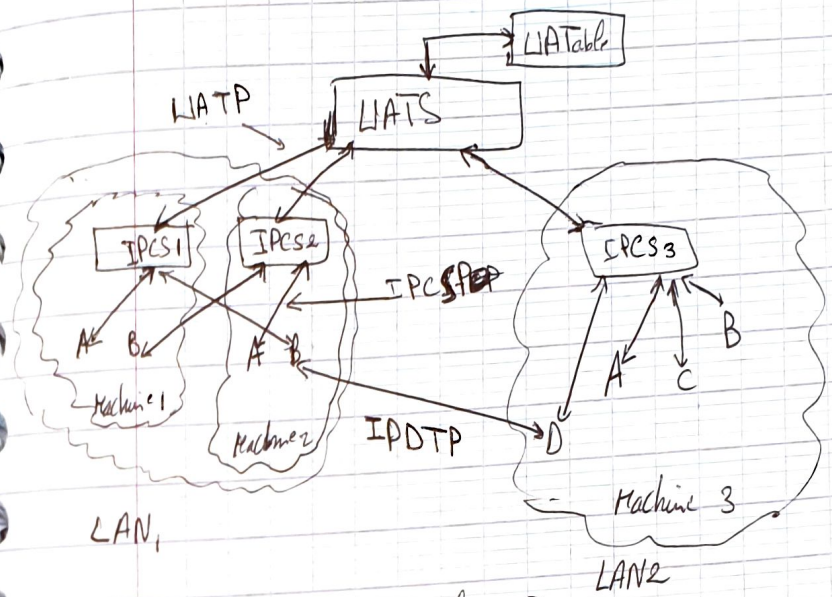
⇒ Note - the optional field is variable with - the command - each new generated send packet is identified by new session ID

e.g A —— switch to SIP ——→ B

A ←—— step with SIP ——— B

- the protocol admit 4 encryption algorithms AES, ARC6 as TOP SECRET and 3DES for simple
- the protocol admits 2 compression algorithms GZIP, Deflate.

⇒ Global Schematic



UATP { 1 single protocol + HSP

IPCSP { 1 Single protocol + HSP

IPDTP { IPDTP
~~IPSEC~~ IPTLS
Unknown

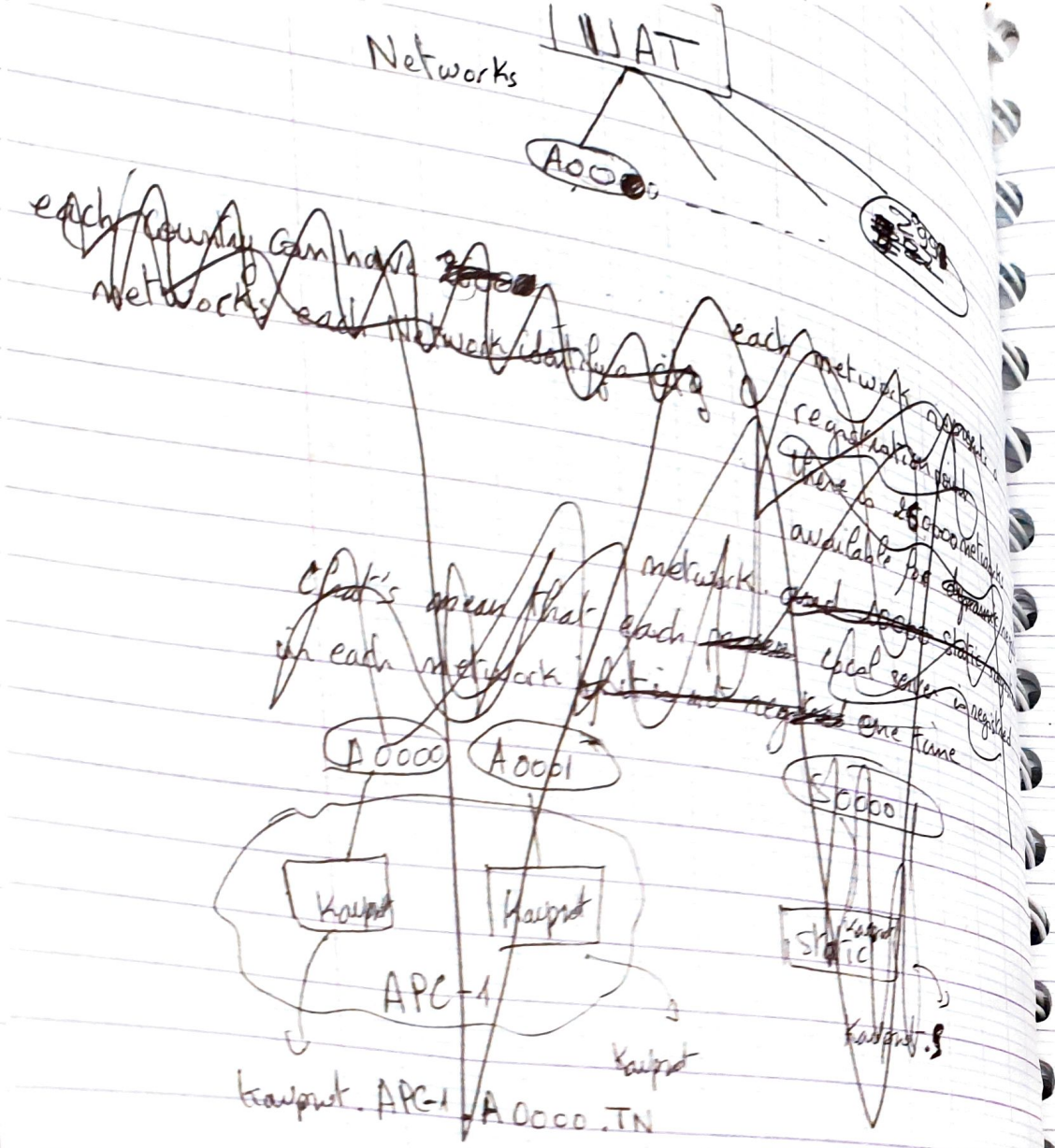
This is a prototype researches are continuing

Work :

→ Web Server Management
→ Servers Definition or partitionning

Developer Studio 2013 3.1

→ Icons in [illegible] 64
→ Breakpoint Manager
→ Debuggers
→ Setup Creator
→ Resource editor (integrated)
→ Unit [illegible] declaration
→ First time Assistation
→ IPDTS
→ Designer (Visual Form Designer)
→ Plugin System Upgrade
→ Credits form + Product announcement
→ GLS Upgrade
→ online Backup + (Automatic)
→ First time Message (Plugin and TPack)
→ Code Network (publish + find)



each country can have 46656 network composed from 3 alphanumeric caracters (A..Z)(0..9)
each network is created automatically if no space found inside a network.

→ for security reason only TRUSTED PARTIES who COMMUNICATE with UATS.
→ each application name stay constant until the network is Fully Blocked the application name should be changed
→ each hours the address should be refreshed (max
→ each IP can Have only 1 server Registred in the UAT
→ each IPCS choose a UPNP port via TCP (port number N° ed by IANA)

IPDTS

1. Definition

IPDT S (Inter-Process Data Transmission System) is a Transmission system designed to transfer data over the internet and networks between processes. This system is composed from 3 Different layers (Local layer, Server layer, External layer)

1/ Local layer (required)

the Local layer is composed from local processes in the Hole LAN who supports IPDTP

2/ Server layer (required)

this layer is composed from only 1 server in a Single LAN. This server is responsible on Registering Processes Addresses and generates equivalent UPL

3/ External layer or Translation layer (optionnal)

this layer is responsible on registring server into UAT with a corresponding ASL and USL.

II Local layer

1/ Rules

- each process has it's own APL and UPL
  APL: Addressing Process Locator (local IP, Port)
  UPL: Uniform Process Locator (Process. Machine. ID)
- each process utilise the IPDTP for Handshake, Transmission, Security, INformation demand, Register, unregister
- each process connect it self to IPDTS to have a new identity
- each process disconnect itself from IPDTS to take a new identity next time and to free the PLT
  PLT: Process Location Table.
- each process can't use (undefined) encryption
- each process should give the correct information when
- each process can connect it self to any IPDTS on a LAN

III Server layer.

1/ Rules

- each LAN have a right to connect only one IPDTServer to the UATS
- each server integrates a PLT
  PLT: Process Location Table
- each uses IPDTP for Communication and HTTP for Translation

- each server shares it's information with any IPDTS compatible process
* the informations includes
  - the compression Algorithm
  - the encryption Algorithm
  - Uniform Server Locator (USL)
  - Addressing Server Locator (ASL)
- each server uses Handshake, Information, Translate connected, Register, Unregister to communicate with clients
- each server connect it self to the router and register a new UPNP Port (the port is pre defined) and the used protocol is TCP.
- each server can accept more than 1000 process

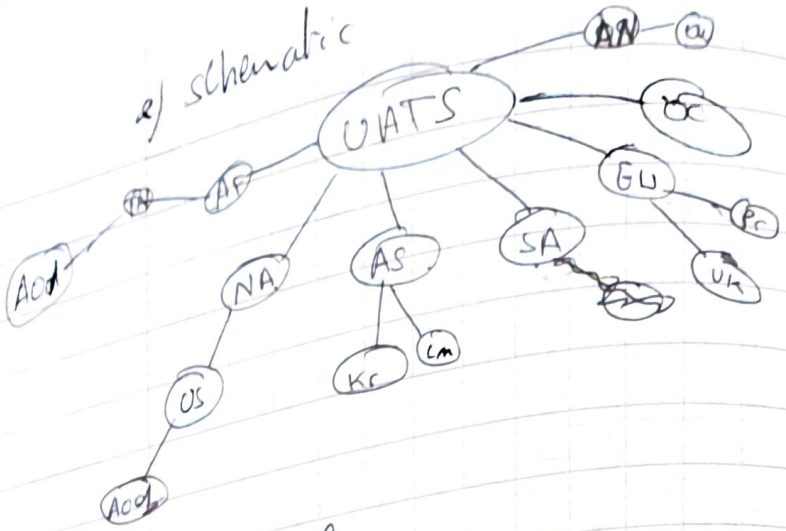
III External layer

1) Rules
- each UATS is composed from many Translator
- each Translator is responsible for Translating USL to ASL for world wide users.

- There is 5 Translator and registration point in the world
  - Africa, Europe, America, Asia, Australia

- each country point register and manage the countries in a continent
- each country have a rigth to oblain 46656 network.
- each network is composed from one or more IPDTS locations
- each network is created after a last network is filled
- each registration is removed if it wasn't refreshed every hour. 60 min
- each UATS have it's own Who is that defines the locations of Translation servers
- UATS database is devided on 7 parts : 7 databases
  AF, AN, SA, NA, EU, OC, AS
- each database is devided on a number of table identified by a country Suffix
- each table contains the registred USL in this country
- a UATS have a Table of static ASL - USL that can't be changed (e.g) : (Jane. static. A01. sta)
- A static address is given 1 time for each process and it can be use once (1 per process)

a/ schematic



I Protocol

A-EXP : 1/ HBB

a/ Header

b/ Body

2/ ~~Header~~ HWB

B- Def. the IPDTP ~~Header~~ is a ~~32~~bit size 9 bytes

CMD : 3 bits

0: TRAN
1: HSK
2: ~~ERR~~ ACC
3: DEC
4: REG
5: URG
6: INF
7: TAK

CType : 2 bits

0: string
1: DATA
2: IMAGE
3: SOUND

GZIP 1bit
~~ENC~~ ENC 1bit
ERR 1bit
CRC 32bit
data Min 1 Max 2^{32} bytes
length : 32bit

eg

CMD = TAK, GZIP, NO ENC, NO ERR, ~~Status~~

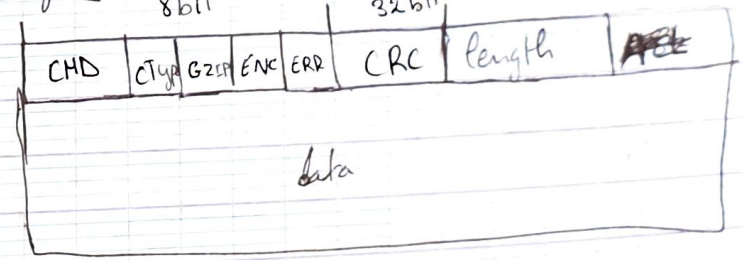
~~110~~00100 = 2^2+2^5 = 128+4 = 132

CMD = URG, GZIP, NO ENC, ERR, ~~DATA~~

10101101 = $2^7+2^5+2^3+2^2+2^0$ =

128 + 32 + 8 + 4 + 1 = 173

C-Design :



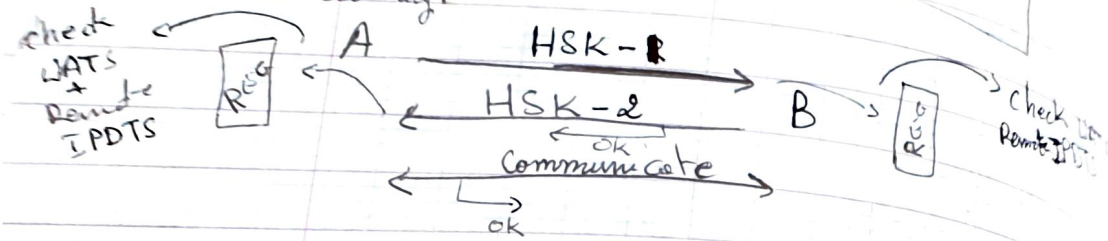
required

APL, USL, UPL

Key, SecretID...

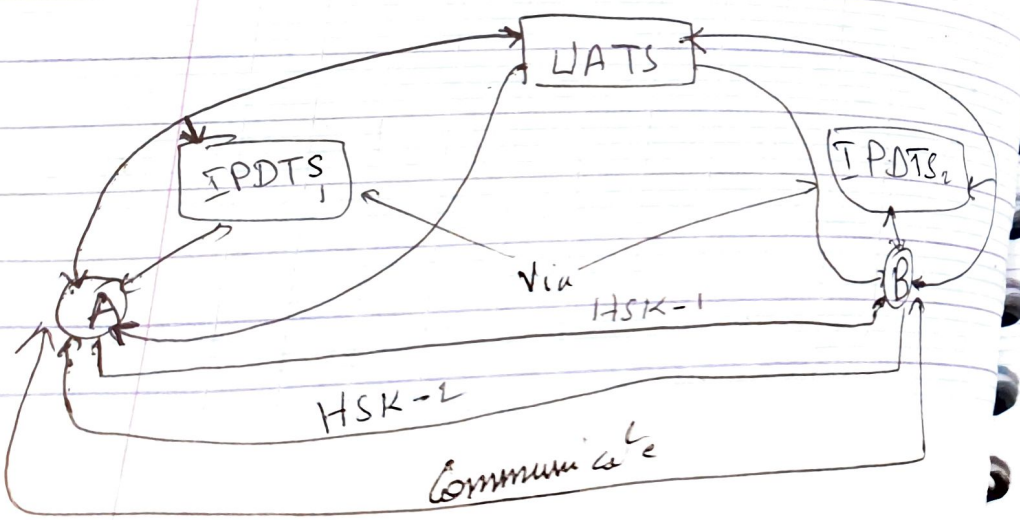
optional, secondary.

HSK	DATA	GZIP	ENC	ERR	CRC	Length
USL	data					



A send a Handshake Request to B.
B verify the identity of A and if everything is ok
B send a Handshake to A, A checks identity of B
and validate

E- CONNECTION



① A looks up for IPDTS to contact the UATS of the APL
② A find IPDTS 2 address ASL and send a SYN request of B
③ the IPDTS 2 return informations about B process
④ A send Packets to IPDTS 2 and wait B to respond
⑤ B validate A and send Handshake
⑥ A and B are connected via IPDTS 1, IPDTS 2 and can communicate.

objectifs
- Transmit data Securely
- communication between procesn
- remote communication
- Handshake
- Uniform Cotor

I System

Clients

TRASMIT
- Secure
- communicate
- Lookup

IPDTS	UATS
Port Opener	Translate UPL → APL
Registrator	TRANSLATE APL → UPL
ORGANISER	organize locations
REFRESHER	define UPL
½ UPL definer	Network definition
Network Manager	Suffix definition
verification service	

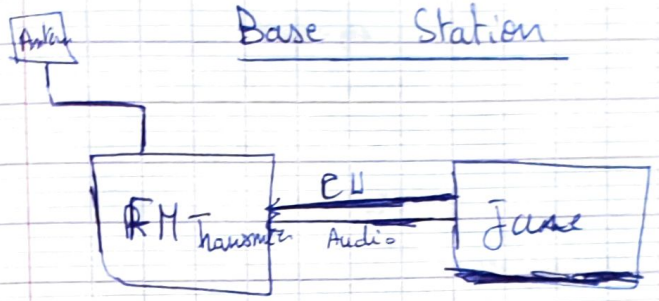
II Protocol

- CMD 3 bits
  TRN, REG, UPG, CHK, HSK, AHK, INF, UPL
- Ctype : 2 bits
  DATA, STRING, IMAGE, SOUND
- GZIP , Encrypt, 2bits
- CRC 32bits
- length : 32bits
- SRC : 48bits

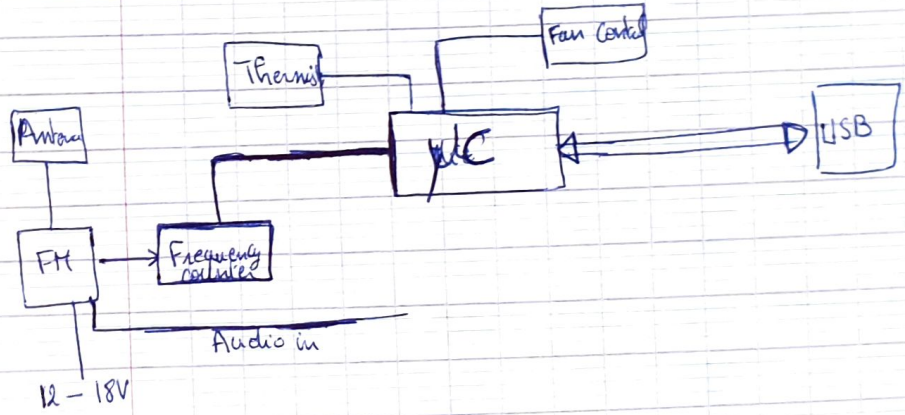
The Return of JANE

→ The total control of her own base station
→ the maximum intelligence algorithms included
→ the best performance wanted
⇛ the perfect AI System.

Base Station



1) Audio : Mono
2) CU : Data Transmission (2) / RF Transmission (2)



Estimated cost of the system : 35 TND

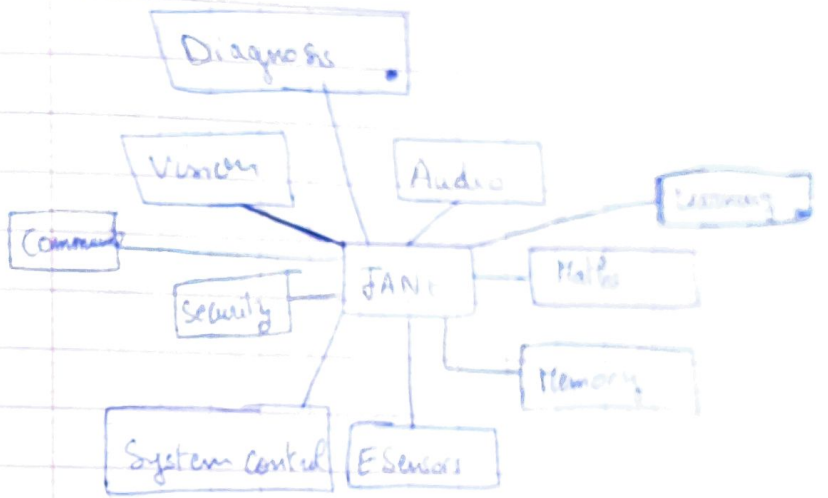
Choosing the name

Context: Artificial / Intelligence / Autonomous / Integration / System / Analyst / Virtual / Hyper / Extreme

the name is gone ~~the~~ EVISA

FANE Extreme Virtual Intelligent System Analyst

Analysis



~~Th~~ Note,

→ each component will include a learning Module and a memory storage space to access learned data and experience.

→ each component includes diagnostic System to detect problems and to make corrections them.

→

① Building the Speech platform
A- Speech Synthesis
- an organized Speech Synthesis Module
- Integrates Diagnosis
- Languages : en/fr/de/it/cn/ar
- Speak in wav
- Wav to mp3
B- Speech Recognizer
- Establish Custom Grammar
- Send Received data to the appropriate callb
- Put Recognition choices in Waiting list for execution
- Next option. Dictation option

② Building the AIML
- Each person have his own profile
- Include learning
- Increase Seeking
- Collect Information profile
- Data storage and Sync with other compu

③ Building Vision
- Facial/object Recognition
- object Detection → read text on that object
- OCR
- Motion Detection.
- object tracking

③ Emotional Network
Conditions :
- Change emotions gradually if results are 60 → 70 %
- Change emotions totally if results are 91 → 100 %
-