Doubango Framework

The quickest route to develop IMS/RCS/VoLTE solutions

Inspiring The Future

Mamadou Diop

Agenda

- Doubango Telecom
- Architecture
- Skipped technical subjects
- Debugging
- OOP in ANSI-C
- Ragel
- FSM
- Networking
- Plugins
- SIP
- Media
- Services
- iOS proxy video producer example
- iOS4+ multitasking
- Reported issues
- QR

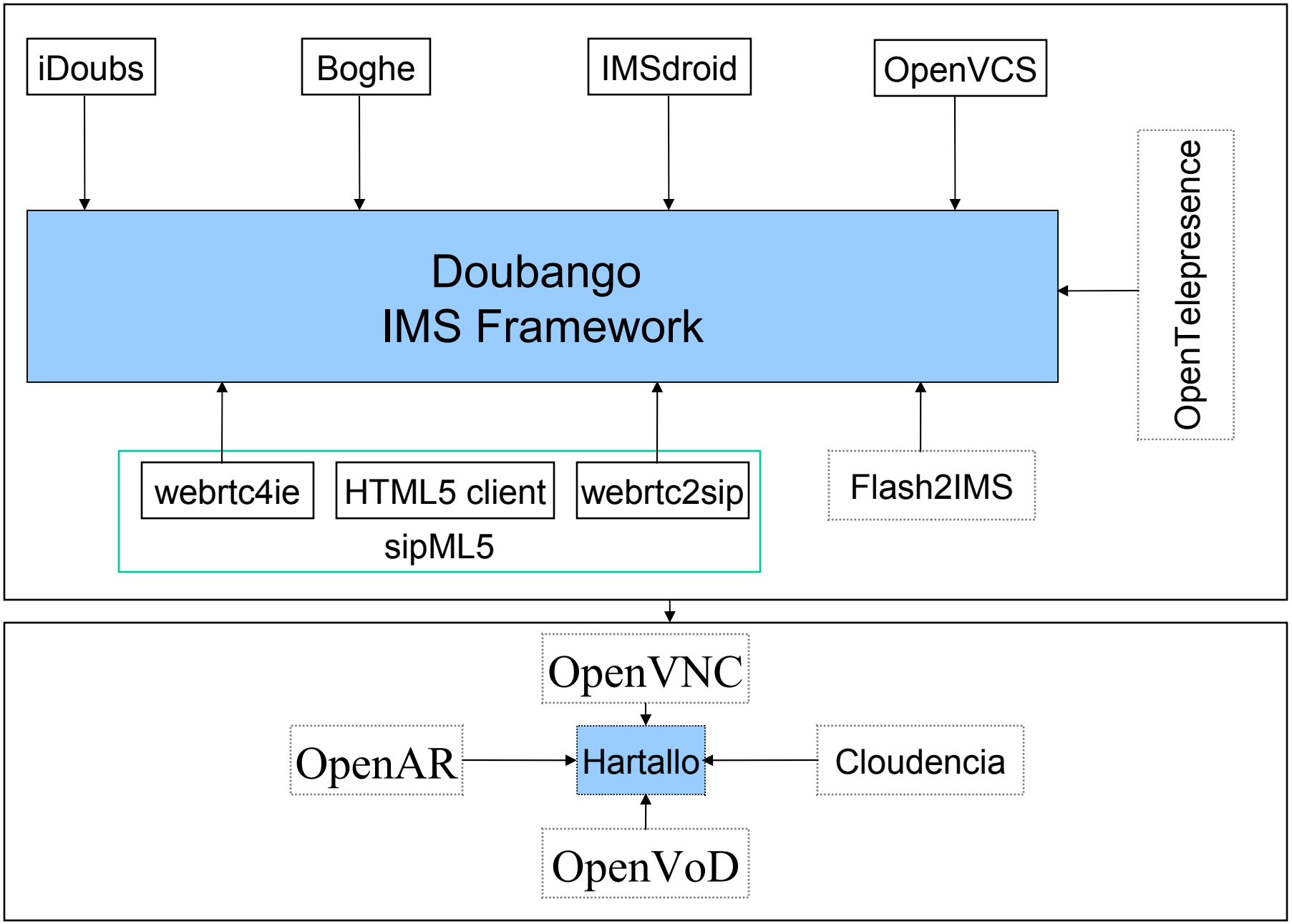
Key Numbers

- *5 supported Operating Systems*
  - *iOS, Android, Windows, MAC OS X and Linux*
- *More Than 4.5 M line of code*
- *100% Open Source*
- *27 active contributors (CA; NDA)*
- *23 technical partners*
- *48 commercial deployments*
- *World's 1st open source IMS/RCS/VoLTE framework*
- *World's 1st open source videophone for Android*
- *World's 1st open source videophone for iOS*
- *World's 1st HTML5 SIP client (WebRTC)*

Our Company

They trust us

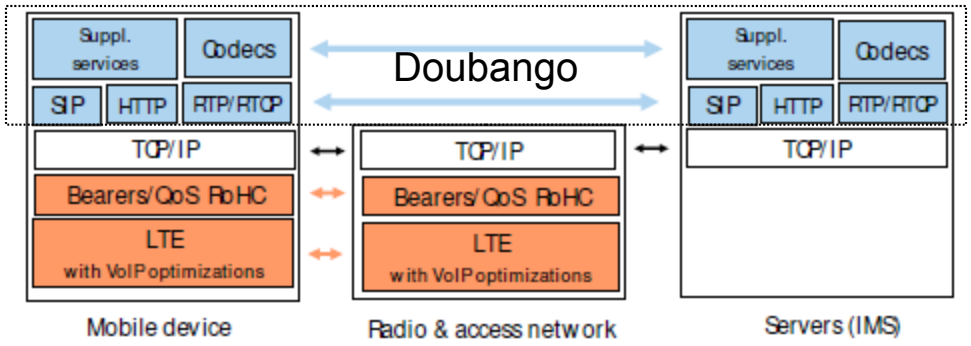
Ecosystem

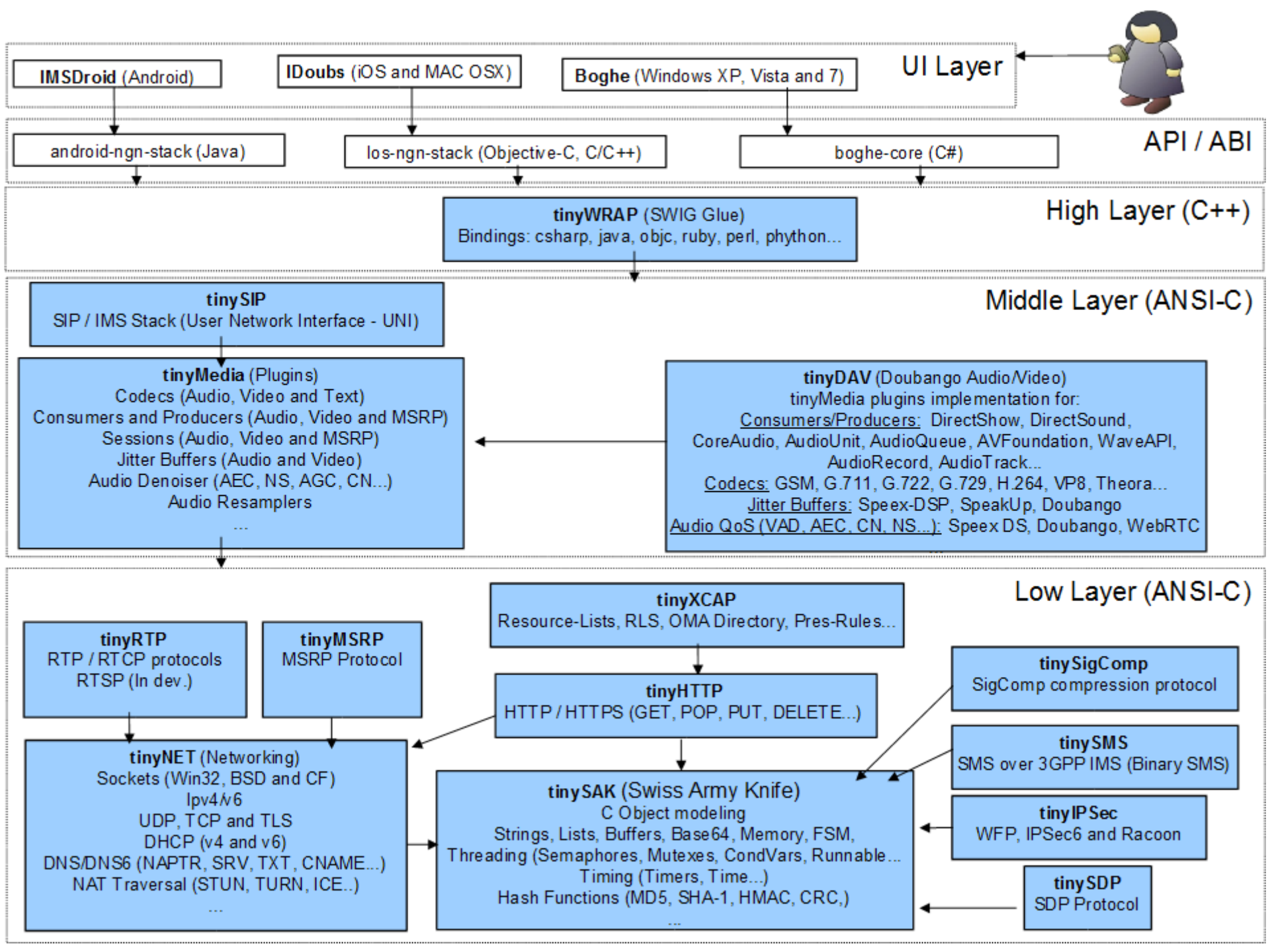

iDoubs
Boghe
IMSdroid
OpenVCS
OpenTelepresence
Doubango
IMS Framework
webrtc4ie
HTML5 client
webrtc2sip
sipML5
Flash2IMS
OpenVNC
OpenAR
Hartallo
Cloudencia
OpenVoD

Agenda

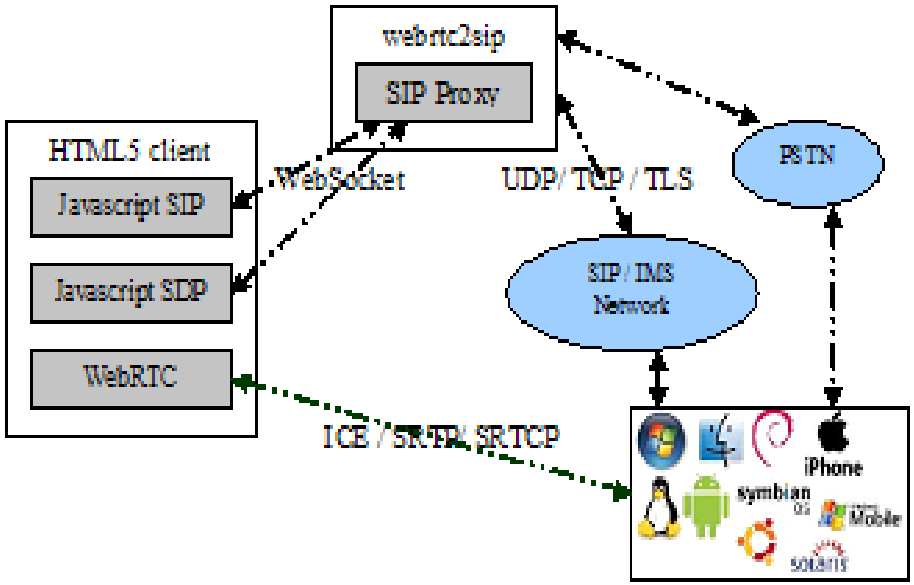
- Doubango Telecom
- Architecture
- Skipped technical subjects
- Debugging
- OOP in ANSI-C
- Ragel
- FSM
- Networking
- Plugins
- SIP
- Media
- Services
- iOS proxy video producer example
- iOS4+ multitasking
- Reported issues
- QR

Architecture (1/3)




IMSDroid (Android)
IDoubs (iOS and MAC OSX)
Boghe (Windows XP, Vista and 7)
UI Layer
android-ngn-stack (Java)
Ios-ngn-stack (Objective-C, C/C++)
boghe-core (C#)
API / ABI
tinyWRAP (SWIG Glue)
Bindings: csharp, java, objc, ruby, perl, phython...
High Layer (C++)
tinySIP
SIP / IMS Stack (User Network Interface - UNI)
tinyMedia (Plugins)
Codecs (Audio, Video and Text)
Consumers and Producers (Audio, Video and MSRP)
Sessions (Audio, Video and MSRP)
Jitter Buffers (Audio and Video)
Audio Denoiser (AEC, NS, AGC, CN...)
Audio Resamplers
...
tinyDAV (Doubango Audio/Video)
tinyMedia plugins implementation for:
Consumers/Producers: DirectShow, DirectSound, CoreAudio, AudioUnit, AudioQueue, AVFoundation, WaveAPI, AudioRecord, AudioTrack...
Codecs: GSM, G.711, G.722, G.729, H.264, VP8, Theora...
Jitter Buffers: Speex-DSP, SpeakUp, Doubango
Audio QoS (VAD, AEC, CN, NS...): Speex DS, Doubango, WebRTC
...
Middle Layer (ANSI-C)
tinyXCAP
Resource-Lists, RLS, OMA Directory, Pres-Rules...
tinyRTP
RTP / RTCP protocols
RTSP (In dev.)
tinyMSRP
MSRP Protocol
tinyHTTP
HTTP / HTTPS (GET, POP, PUT, DELETE...)
tinySigComp
SigComp compression protocol
tinySMS
SMS over 3GPP IMS (Binary SMS)
tinyNET (Networking)
Sockets (Win32, BSD and CF)
Ipv4/v6
UDP, TCP and TLS
DHCP (v4 and v6)
DNS/DNS6 (NAPTR, SRV, TXT, CNAME...)
NAT Traversal (STUN, TURN, ICE..)
...
tinySAK (Swiss Army Knife)
C Object modeling
Strings, Lists, Buffers, Base64, Memory, FSM,
Threading (Semaphores, Mutexes, CondVars, Runnable...
Timing (Timers, Time...)
Hash Functions (MD5, SHA-1, HMAC, CRC,)
...
tinyIPSec
WFP, IPSec6 and Racoon
tinySDP
SDP Protocol
Low Layer (ANSI-C)

Architecture (3/3)



3rd parties

FFmpeg	(L)GPL
x264	GPL
Theora, libogg, libvorbis	BSD-style
VPX	New BSD
libgsm	3-clause BSD
Opencore-amr	Apache License 2.0
Speex	revised-BSD
iLBC	GIPS Public License
Simple XML	Apache License 2.0
libsrtp	BSD-based
OpenSSL	OpenSSL and SSLeay licenses

Agenda

- Doubango Telecom
- Architecture
- Skipped technical subjects
- Debugging
- OOP in ANSI-C
- Ragel
- FSM
- Networking
- Plugins
- SIP
- Media
- Services
- iOS proxy video producer example
- iOS4+ multitasking
- Reported issues
- QR

Skipped technical subjects

- SigComp (tinySigComp)
- XCAP (tinyXCAP)
- Binary SMS (tinySMS)
- HTTP /HTTPS (tinyHTTP)
- IPSec (tinyIPSec)
- MSRP (tinyMSRP)
- DSSL (tinyDEMO)
- Low level functions
- Windows, Android, Linux and S60

Agenda

- Doubango Telecom
- Architecture
- Skipped technical subjects
- Debugging
- OOP in ANSI-C
- Ragel
- FSM
- Networking
- Plugins
- SIP
- Media
- Services
- iOS proxy video producer example
- iOS4+ multitasking
- Reported issues
- QR

Debuging

- tinySAK project
- Levels: INFO, WARN, ERROR, FATAL
- Macros: TSK_DEBUG_XXX(“code=%hi”, 1)
- Output: **stderr**
  - **log4net** using callbacks
  - **file** or **sdtout** using **freopen**

Agenda

- Doubango Telecom
- Architecture
- Skipped technical subjects
- Debugging
- OOP in ANSI-C
- Ragel
- FSM
- Networking
- Plugins
- SIP
- Media
- Services
- iOS proxy video producer example
- iOS4+ multitasking
- Reported issues
- QR

OOP in ANSI-C

- tinySAK project
- ANSI-C structures
- Constructor, destructor and comparator
- Reference counting
- Memory leak tracking
- Inheritance

OOP in ANSI-C

Definition

```
typedef struct tsk_object_def_s{
   size_t size;
   tsk_object_t* (* constructor) (tsk_object_t *, va_list *);
   tsk_object_t* (* destructor) (tsk_object_t *);
   int (*comparator) (const tsk_object_t *, const tsk_object_t *);

}tsk_object_def_t;
```

Implementation

```
typedef struct person_s{
   TSK_DECLARE_OBJECT;
   char* name;
   struct person_s* girlfriend;
}
person_t;
```

Declarartion

```
static const tsk_object_def_t person_def_t = {
   sizeof(person_t),
   person_ctor,
   person_dtor,
   person_cmp
 };
```

OOP in ANSI-C

Implementation

```
static tsk_object_t* person_ctor(tsk_object_t * self, va_list * app){
    person_t *person = self;
    if(person) person->name = tsk_strdup(va_arg(*app, const char *));
    return self;
}
static tsk_object_t * person_dtor(tsk_object_t * self){
    person_t *person = self;
    if(person){
     TSK_FREE(person->name);
     tsk_object_unref(person->girlfriend);
    }
    return self;
}
static int person_cmp(const tsk_object_t *_p1, const tsk_object_t *_p2){
    const person_t *p1 = _p1;
    const person_t *p1 = _p2;
    int ret;

    // do they have the same name?
    if((ret = tsk_stricmp(p1->name, p2->name))){
     return ret;
    }
    // do they have the same girlfriend?
    if((ret = tsk_object_cmp(p1->girlfriend, p2->girlfriend))){
     return ret;
    }

    // they are equal
    return 0;
 }
```

OOP in ANSI-C

Instantiation

```
tsk_object_t* tsk_object_new(const tsk_object_def_t *objdef, ...);
tsk_object_t* tsk_object_new_2(const tsk_object_def_t *objdef, va_list* ap);
```

Reference counting

```
tsk_object_t* tsk_object_ref(tsk_object_t *self);
tsk_object_t* tsk_object_unref(tsk_object_t *self);
void tsk_object_delete(tsk_object_t *self);
```

Inheritance

```
// (a student is a person)
typedef struct student_s{
    struct person_s person;
    char* school;
}student_t;
// (as a student is a person you can do)
student_t* s;
((person_t*)s)->name = tsk_strdup("bob");
```

Usage

```
person_t* bob = tsk_object_new(&person_def_t, "bob");
bob->girlfriend = tsk_object_new(&person_def_t, "alice");
tsk_object_unref(bob);
```

Agenda

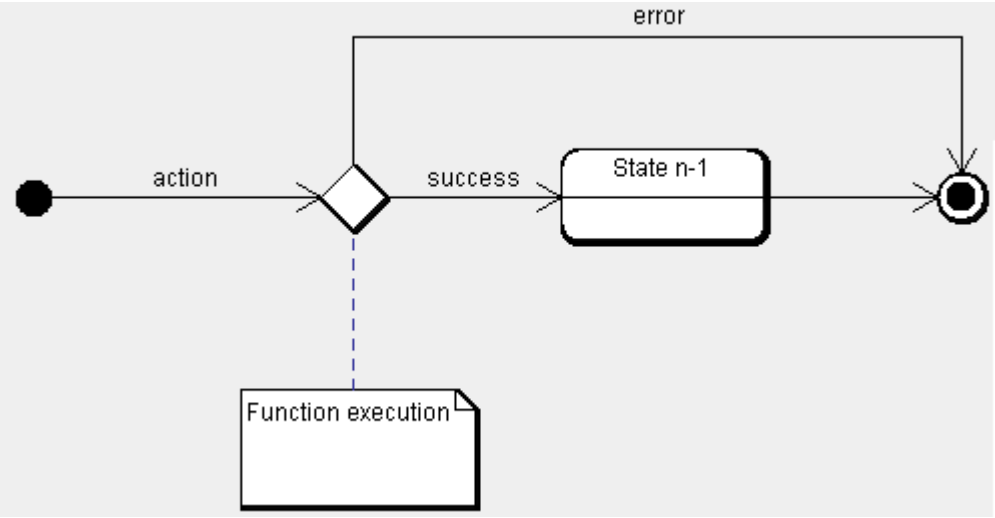
- Doubango Telecom
- Architecture
- Skipped technical subjects
- Debugging
- OOP in ANSI-C
- Ragel
- FSM
- Networking
- Plugins
- SIP
- Media
- Services
- iOS proxy video producer example
- iOS4+ multitasking
- Reported issues
- QR

Agenda

- Doubango Telecom
- Architecture
- Skipped technical subjects
- Debugging
- OOP in ANSI-C
- Ragel
- FSM
- Networking
- Plugins
- SIP
- Media
- Services
- iOS proxy video producer example
- iOS4+ multitasking
- Reported issues
- QR

FSM

- SIP Transactions
- SIP dialogs
- ICE handshake and conncheck
- RFB handshake



FSM

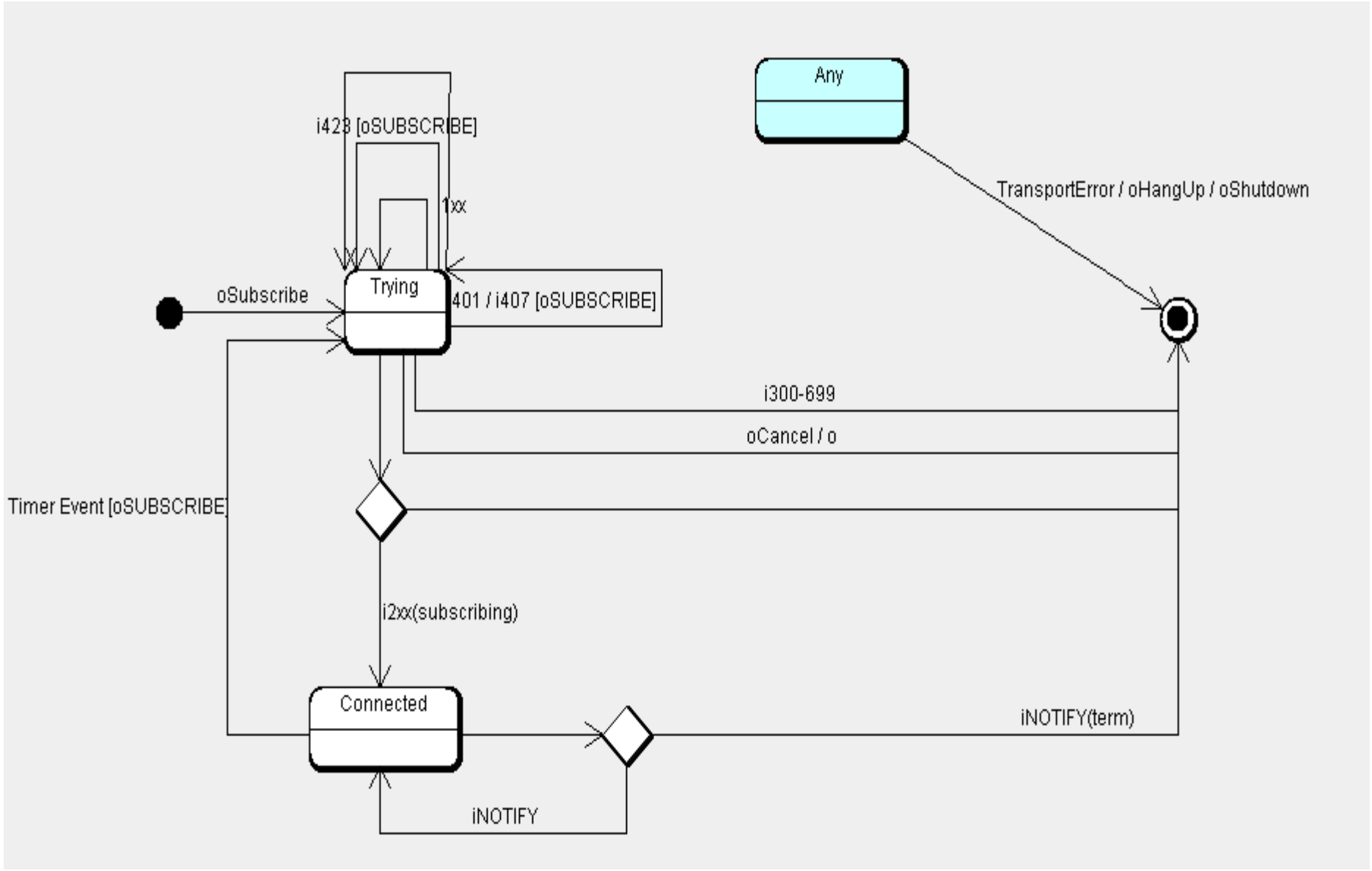
Useful functions:

- `tsk_fsm_t* tsk_fsm_create(tsk_fsm_state_id state_curr, tsk_fsm_state_id state_term);`
- `int tsk_fsm_set(tsk_fsm_t* self, ...);`
- `int tsk_fsm_set_callback_terminated(tsk_fsm_t* self, tsk_fsm_onterminated_f callback, const void* callbackdata);`

Transitions building:

- TSK_FSM_ADD(from, action, cond, to, exec, desc)
- TSK_FSM_ADD_ALWAYS(from, action, to, exec, desc)
- TSK_FSM_ADD_NOTHING(from, action, cond, desc)
- TSK_FSM_ADD_ALWAYS_NOTHING(from, desc)
- TSK_FSM_ADD_NULL()

FSM: SIP subscription



FSM: SIP subscription

```
tsk_fsm_set(self,
      TSK_FSM_ADD_ALWAYS(S__Started, A__oSUBSCRIBE, S__Trying, F__0, "F__0"),
      TSK_FSM_ADD_ALWAYS_NOTHING(S__Started, "F__1"),

      TSK_FSM_ADD_ALWAYS(S__Trying, A__i1xx, S__Trying, F__2, "F__2"),
      TSK_FSM_ADD(S__Trying, A__2xx, C__unsubscribing, S__Terminated, F__3, "F__3"),
      TSK_FSM_ADD(S__Trying, A__2xx, C__subscribing, S_Connected, F__4, "F__4"),
      TSK_FSM_ADD_ALWAYS(S__Trying, A__i401_i407_i421_i494, S__Trying, F__5, "F__5"),
      TSK_FSM_ADD_ALWAYS(S__Trying, A__423, S__Trying, F__6, "F__6"),
      TSK_FSM_ADD_ALWAYS(S__Trying, A__i300_to_i699, S__Terminated, F__7, "F__7"),
      TSK_FSM_ADD_ALWAYS(S__Trying, A__oCancel, S__Terminated, F__8, "F__8"),
      TSK_FSM_ADD_ALWAYS(S__Trying, A__iNOTIFY, S__Trying, F__9, "F__9"),
      TSK_FSM_ADD_ALWAYS(S__Trying, A__oHangup, S__Terminated, F__NULL, "F__NULL"),
      TSK_FSM_ADD_ALWAYS(S__Trying, A__oShutdown, S__Terminated, F__NULL, "F__NULL"),

      TSK_FSM_ADD_ALWAYS(S__Connected, A__oSUBSCRIBE, S__Trying, F__10, "F__10"),
      TSK_FSM_ADD(S__Connected, A__ iNOTIFY, C_NotifyNotTerm, S__Connected, F__11, "F__11"),
      TSK_FSM_ADD(S__Connected, A__ iNOTIFY, C_NotifyTerm, S__Terminated, F__11, "F__12"),

      TSK_FSM_ADD(S__any, A__oHangup, C_NotSilentHangup, S__Trying, F__13, "F__13"),
      TSK_FSM_ADD(S__any, A__oHangup, C_SilentHangup, S__Terminated, F__NULL, "F__NULL"),
      TSK_FSM_ADD(S__any, A__oShutdown, C_NotSilentShutdown, S__Trying, F__14, "F__14"),
      TSK_FSM_ADD(S__any, A__oShutdown, C_SilentShutdown, S__Terminated, F__NULL, "F__NULL"),
      TSK_FSM_ADD_ALWAYS(S__any, A__oShutdownTimedout, S__Terminated, F__NULL, "F__NULL"),
      TSK_FSM_ADD_ALWAYS(S__any, A__TransportError, S__Terminated, F__15, "F__15"),
      TSK_FSM_ADD_ALWAYS(S__any, A__Error, S__Terminated, F__16, "F__16"),

      TSK_FSM_ADD_NULL()
);
```

Agenda

- Doubango Telecom
- Architecture
- Skipped technical subjects
- Debugging
- OOP in ANSI-C
- Ragel
- FSM
- Networking
- Plugins
- SIP
- Media
- Services
- iOS proxy video producer example
- iOS4+ multitasking
- Reported issues
- QR

Networking

- tinyNET project
- Thread-safe
- Protocols: TCP, TLS, UDP, WS and WSS
- NATT: STUN, TURN and ICE
- 3 Transports:
  - WinSock2: Windows XP, Vista and 7
  - Poll: iOS3-, OS X, Android and Linux
  - CFSocket: iOS4+
- DHCPv4 / DHCPv6:
  - Parameter Request List (RFC 2132)
  - Domain Name Server Option (RFC 2132)
  - DHCP option for SIP Servers (RFC 3361)
- ENUM (E.164 Number Mapping) protocol
- DNS: CNAME, MX, NAPTR, PTR, SRV...

Networking: Sockets

- Non-blocking
- IPv4 / IPv6
- Agnostic API:
  - WinSock2 sockets on Windows XP, Vista and 7
  - BSD sockets on iOS, OS X, Linux and Android

```
int ret;
tsk_size_t ret_size;

// create IPv6/tcp socket
tnet_socket_t* socket = tnet_socket_create(TNET_SOCKET_HOST_ANY, TNET_SOCKET_PORT_ANY,
    tnet_socket_type_tcp_ipv6);

// create destination address
struct sockaddr_storage to;
ret = tnet_sockaddr_init("ipv6.google.com", 80, socket->type, &to);

// connect the socket
ret = tnet_sockfd_connectto(socket->fd, (const struct sockaddr_storage *)&to);

// send data
ret_size = tnet_sockfd_send(socket->fd, "test", tsk_strlen("test"), 0);

// destroy the socket (close + free)
TSK_OBJECT_SAFE_FREE(socket);
```

Networking: DNS

```
tnet_dns_response_t* tnet_dns_resolve(tnet_dns_ctx_t* ctx,  const char* qname,
            tnet_dns_qclass_t qclass,  tnet_dns_qtype_t qtype);
```

Example:

```
tnet_dns_response_t * response = tnet_dns_resolve(ctx, "sip2sip.info", qclass_in,
    qtype_naptr);

// ((tnet_dns_naptr_t*)response)->services == "_sip._udp.sip2sip.info"

response = tnet_dns_resolve(ctx, "_sip._udp.sip2sip.info", qclass_in, qtype_srv);

// ((tnet_dns_srv_t*)response)->target == "proxy.sipthor.net"
// ((tnet_dns_srv_t*)response)->port == 5060
```

```
tnet_dns_response_t* tnet_dns_enum(tnet_dns_ctx_t* ctx, const char* e164num,
    const char* domain);

char* tnet_dns_enum_2(tnet_dns_ctx_t* ctx, const char* service, const char*
    e164num, const char* domain);
```

Example:

```
char* uri = tnet_dns_enum_2(ctx, "E2U+SIP", "+1-800-555-5555", "e164.org");

uri == sip:16416418000-555-5555@sip.tollfreegateway.com
```

Networking: DHCP

```
tnet_dhcp_reply_t* tnet_dhcp_query(tnet_dhcp_ctx_t* ctx,
   tnet_dhcp_message_type_t type, tnet_dhcp_params_t* params);
```

Example:

```
tnet_dhcp_ctx_t *ctx = tnet_dhcp_ctx_create();
tnet_dhcp_params_t *params = tsk_null;
tnet_dhcp_reply_t *reply = tsk_null;

params = tnet_dhcp_params_create();
tnet_dhcp_params_add_code(params, dhcp_code_SIP_Servers_DHCP_Option); /* SIP Servers */
tnet_dhcp_params_add_code(params, dhcp_code_Domain_Server); /* DNS Server */

reply = tnet_dhcp_query_inform(ctx, params);
switch(reply->type){
  case dhcp_type_ack:{
     // loop through reply->options
     break;
   }
}
TSK_OBJECT_SAFE_FREE(reply);
TSK_OBJECT_SAFE_FREE(params);
TSK_OBJECT_SAFE_FREE(ctx);
```

Networking: Transport

- Agnostic API (WinSock2, BSD, CFSocket)
- Must be stated before any action
- Multi-threaded and thread-safe
- Incoming data received in worker thread
- Use callbacks to forward data on wk thread
- Outgoing data sent using caller thread
- Manage array of non-blocking sockets
- Notion of Socket ownership
- Monitor worker thread :
  - Master socket on windows
  - R/W pipes on *NIX

Networking: Transport

```
// forward declaration for the callback function
static int __udp_cb(const tnet_transport_event_t* e);


// create the transport
tnet_transport_handle_t *udp_transport = tnet_transport_create(TNET_SOCKET_HOST_ANY, TNET_SOCKET_PORT_ANY,
     tnet_socket_type_udp_ipv4, "UDP/IPV4 TRANSPORT");
// set callback data
tnet_transport_set_callback(udp_transport, __udp_cb, udp_transport);
// start the transport
int ret = tnet_transport_start(udp_transport);
// get master FD
tnet_fd_t fd = tnet_transport_get_master_fd(udp_transport);
// send data
struct sockaddr_storage to;
ret = tnet_sockaddr_init("sipml5.org", 5060, tnet_transport_get_type(udp_transport), &to);
tsk_size_t ret_size = tnet_transport_sendto(udp_transport, fd, (const struct sockaddr *)&to, "test", tsk_strlen("test"));


static int __udp_cb(const tnet_transport_event_t* e){
     switch(e->type){
       case event_data:{
                    // e->data (const void*): pointer to the data
                    // e->size (tsk_size_t): data size (UInt8 unit)
                    // e->callback_data (const void*): user-defined callback data
                    // e->local_fd (tnet_fd_t): local file descriptor used to receive the data
                    // e->remote_addr struct sockaddr_storage): remote address (sender's address). Only for Dgram transport.
                    break;
               }
       case event_closed: case event_connected: default: break;
     }
     return 0;
}
```

Agenda

- Doubango Telecom
- Architecture
- Skipped technical subjects
- Debugging
- OOP in ANSI-C
- Ragel
- FSM
- Networking
- Plugins
- SIP
- Media
- Services
- iOS proxy video producer example
- iOS4+ multitasking
- Reported issues
- QR

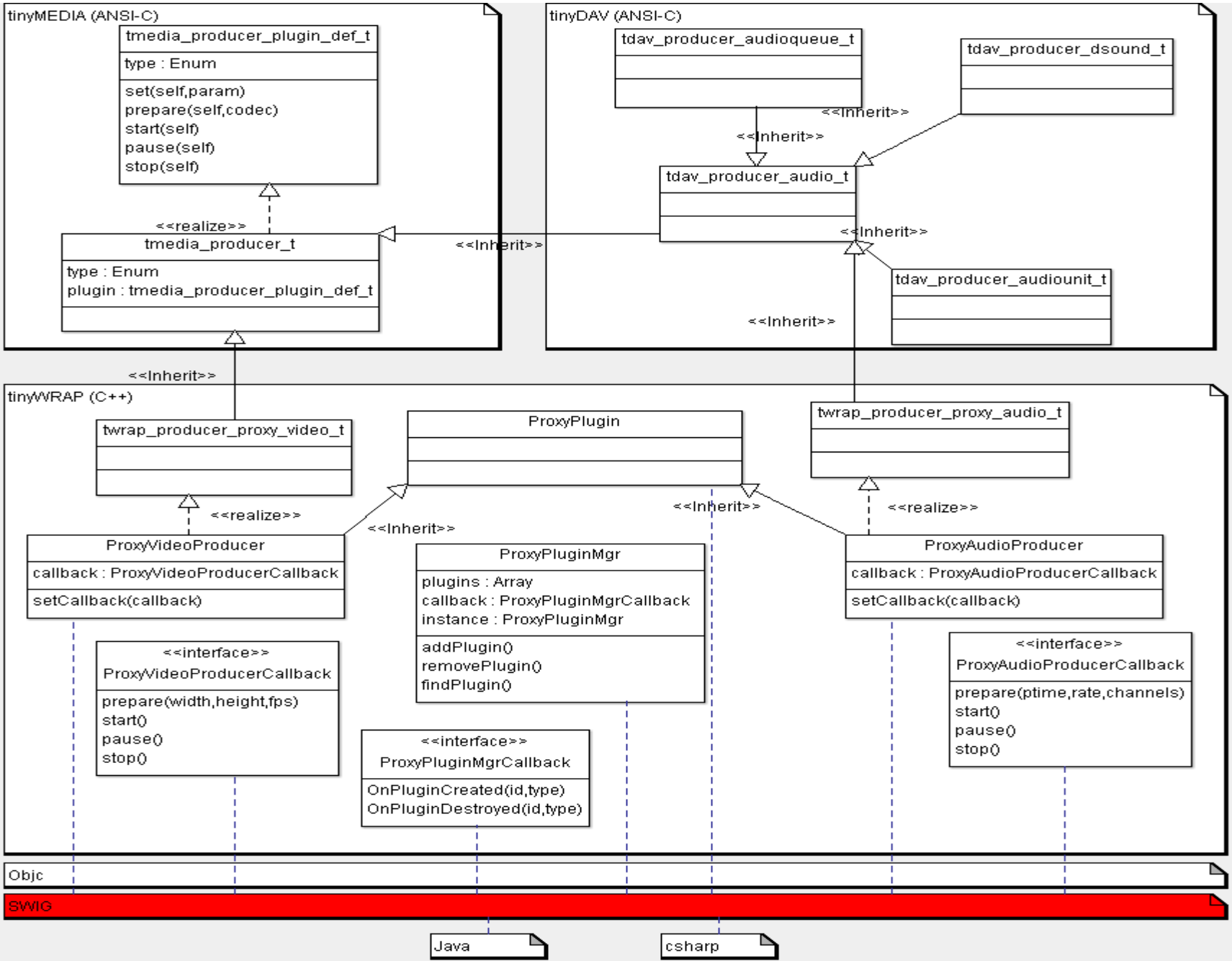
Overview

	Windows XP/ Vista / 7	Android	iOS	OS X
Consumer / Producer	WaveAPI DirectSound DirectShow	AudioTrack AudioRecord	AudioUnit AVFoundation	AudioUnit AudioQueue QT
Session	Audio, Video, MSRP, Ghost			
Denoiser	Speex-DSP, WebRTC			
JitterBuffer	SpeakUp, Speex-DSP			
Resampler	Speex-DSP			
Codecs	AMR, Speex, G.729, G.711, GSM, iLBC, VP8, H.264, H.263, Theora, MP4			

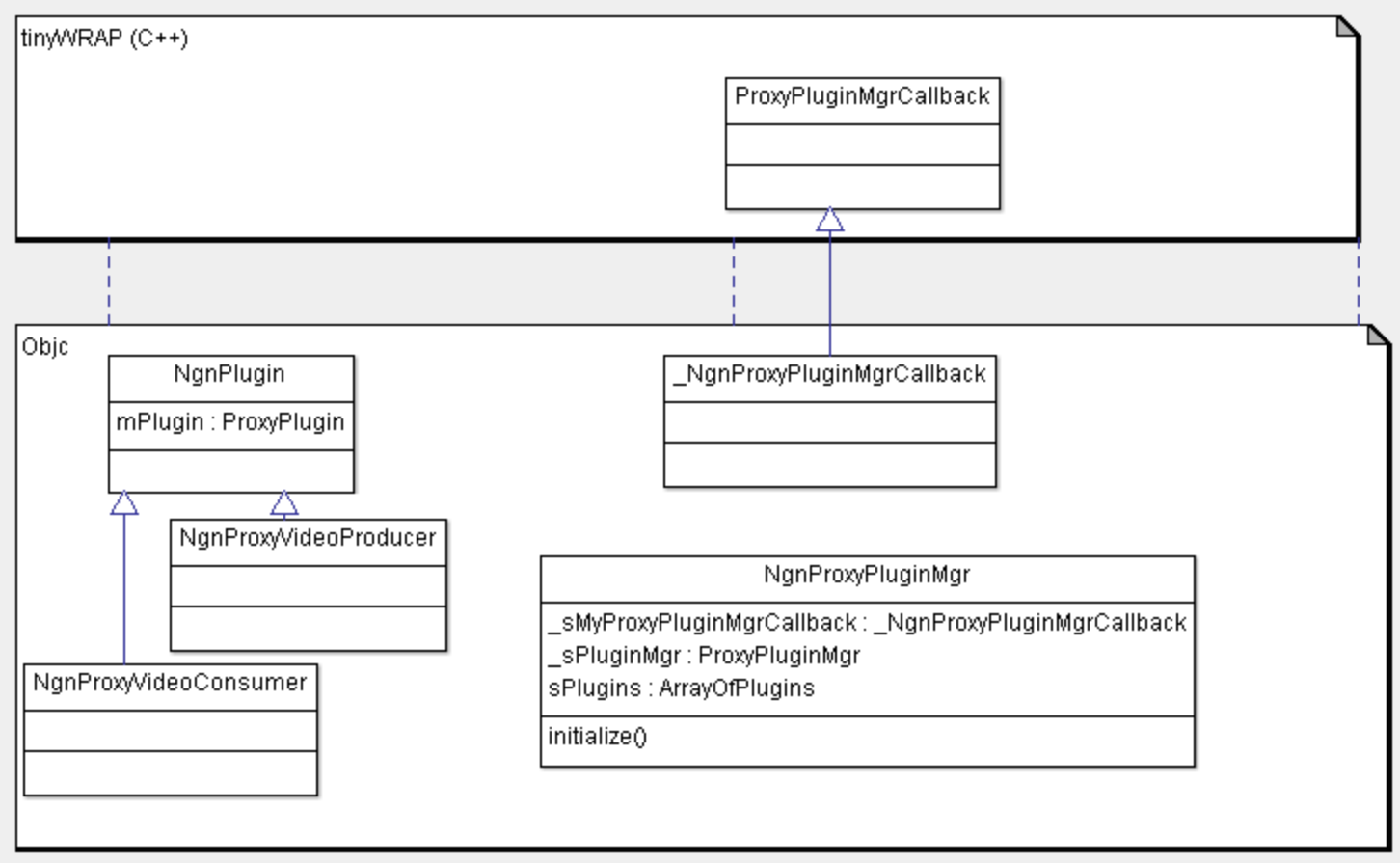
Proxy

- tinyWRAP project
- ANSI-C structs
  - wrapped in C++ classes
  - extended in managed code
- Audio/Video consumers and producers
- Proxying
- Native -> Managed (SWIG directories)
- Managed -> Native (JNI or P/InvokeSpecial) memory tracking

Producer (1 / 2)


tinyMEDIA (ANSI-C)
tmedia_producer_plugin_def_t
type : Enum
set(self,param)
prepare(self,codec)
start(self)
pause(self)
stop(self)
<<realize>>
tmedia_producer_t
type : Enum
plugin : tmedia_producer_plugin_def_t
<<Inherit>>
tinyDAV (ANSI-C)
tdav_producer_audioqueue_t
tdav_producer_dsound_t
<<Inherit>>
<<Inherit>>
tdav_producer_audio_t
<<Inherit>>
tdav_producer_audiounit_t
<<Inherit>>
<<Inherit>>
tinyWRAP (C++)
twrap_producer_proxy_video_t
ProxyPlugin
twrap_producer_proxy_audio_t
<<realize>>
<<Inherit>>
<<Inherit>>
<<realize>>
ProxyVideoProducer
callback : ProxyVideoProducerCallback
setCallback(callback)
ProxyPluginMgr
plugins : Array
callback : ProxyPluginMgrCallback
instance : ProxyPluginMgr
addPlugin()
removePlugin()
findPlugin()
ProxyAudioProducer
callback : ProxyAudioProducerCallback
setCallback(callback)
<<interface>>
ProxyVideoProducerCallback
prepare(width,height,fps)
start()
pause()
stop()
<<interface>>
ProxyPluginMgrCallback
OnPluginCreated(id,type)
OnPluginDestroyed(id,type)
<<interface>>
ProxyAudioProducerCallback
prepare(ptime,rate,channels)
start()
pause()
stop()
Objc
SWIG
Java
csharp

Producer (2 / 2)


tinyWRAP (C++)
ProxyPluginMgrCallback
Objc
NgnPlugin
mPlugin : ProxyPlugin
_NgnProxyPluginMgrCallback
NgnProxyVideoProducer
NgnProxyPluginMgr
_sMyProxyPluginMgrCallback : _NgnProxyPluginMgrCallback
_sPluginMgr : ProxyPluginMgr
sPlugins : ArrayOfPlugins
initialize()
NgnProxyVideoConsumer

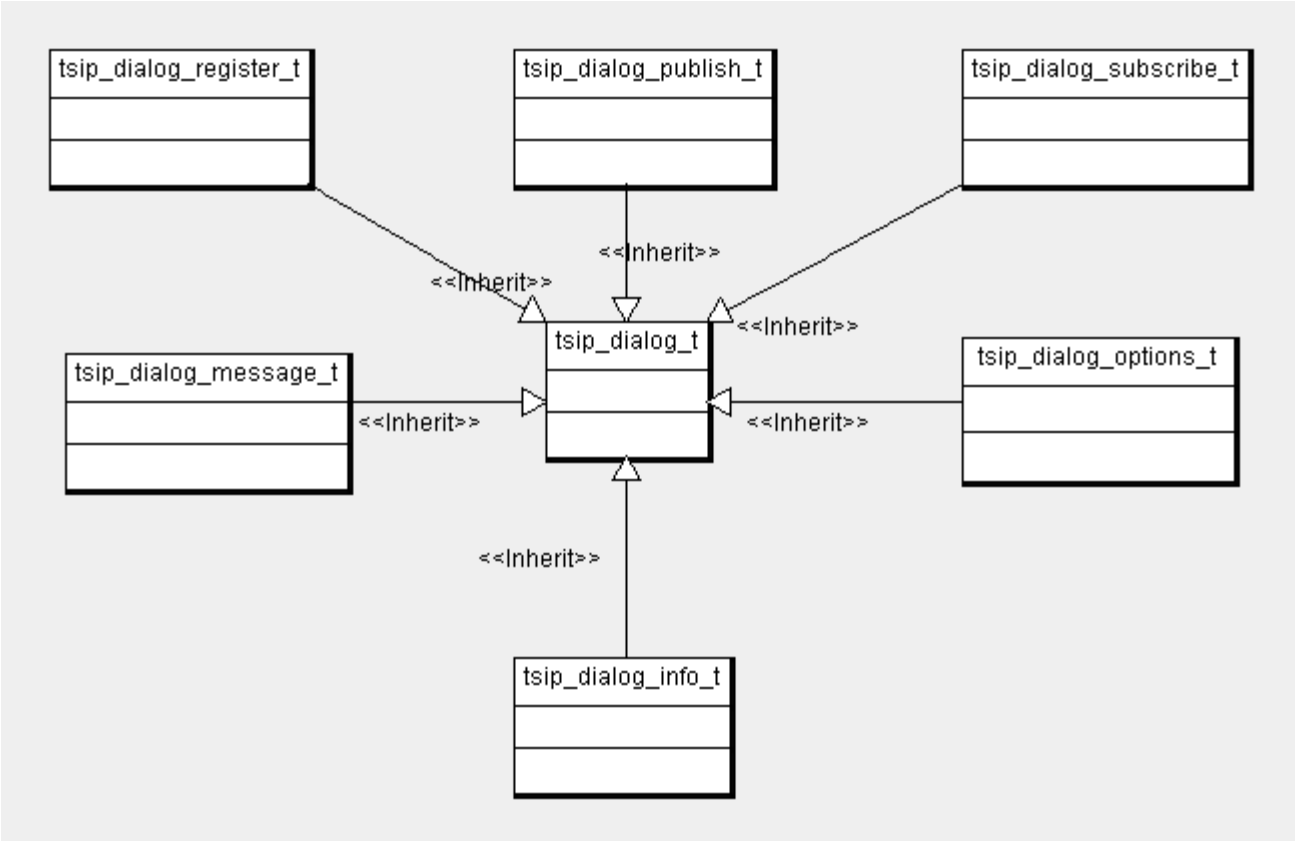
Agenda

- Doubango Telecom
- Architecture
- Skipped technical subjects
- Debugging
- OOP in ANSI-C
- Ragel
- FSM
- Networking
- Plugins
- SIP
- Media
- Services
- iOS proxy video producer example
- iOS4+ multitasking
- Reported issues
- QR

Dialog (1 / 2)

- Dialog
  - State machine transitions
  - Authentication
  - Refresh (e.g. reREGISTER, rePUBLISH…)
  - Dispatch and execute user action
  - Request / response creation and decoration
  - Dispatch outgoing messages to the transaction layer
- Dialog layer
  - Manage SIP dialogs
  - Dispatch incoming SIP message to the right dialog

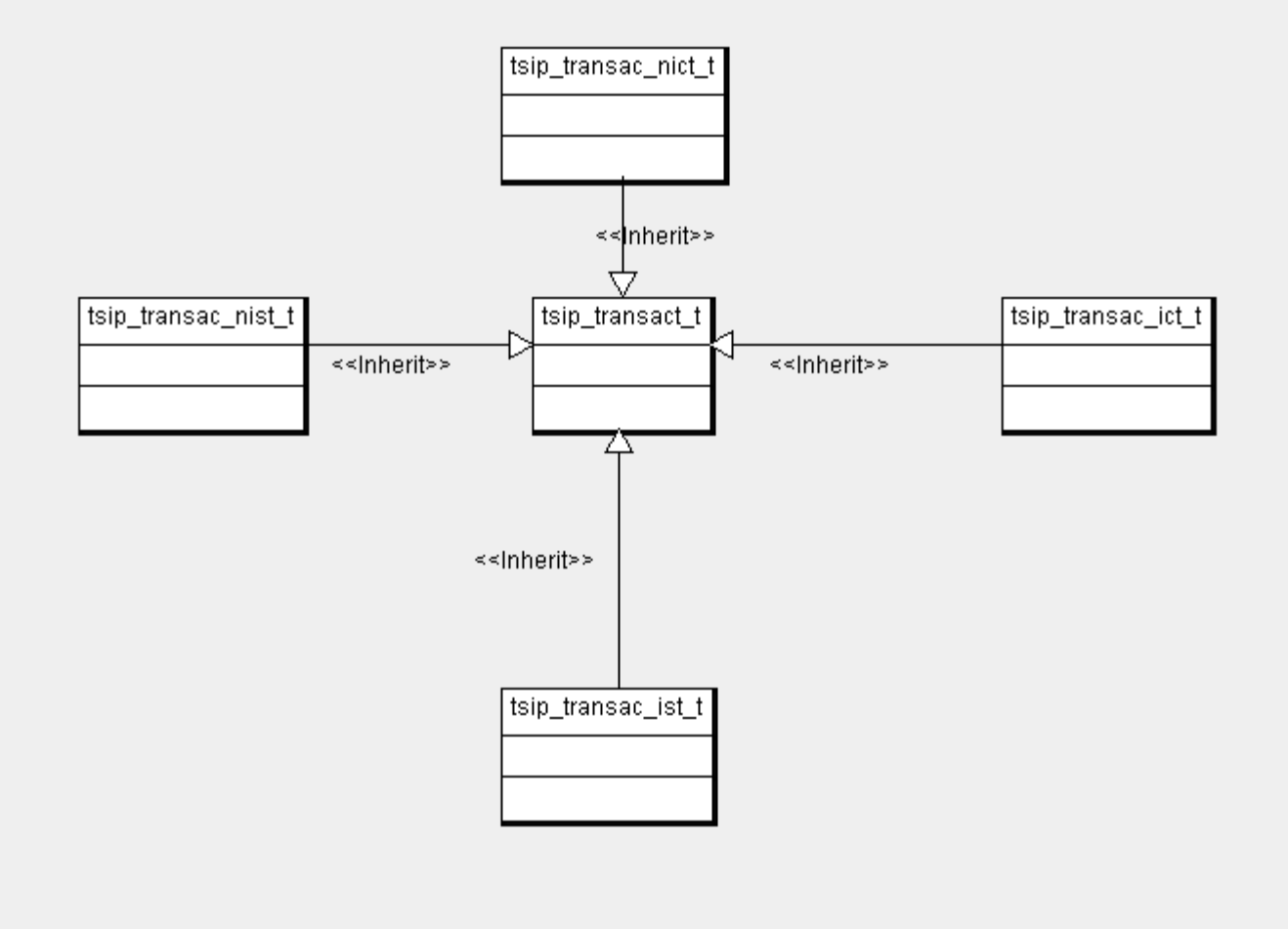
Dialog (2 / 2)


tsip_dialog_register_t
tsip_dialog_publish_t
tsip_dialog_subscribe_t
<<Inherit>>
<<Inherit>>
<<Inherit>>
tsip_dialog_t
tsip_dialog_message_t
<<Inherit>>
<<Inherit>>
tsip_dialog_options_t
<<Inherit>>
tsip_dialog_info_t

Transaction (1 / 2)

- Transaction
  - State machine transitions
  - Messages retransmission
  - Forward outgoing messages (branch)
  - Forward outgoing messages to the transport layer
  - Forward incoming messages to the right dialog
- Transaction layer
  - Manage SIP transactions
  - Forward incoming messages to the right transaction

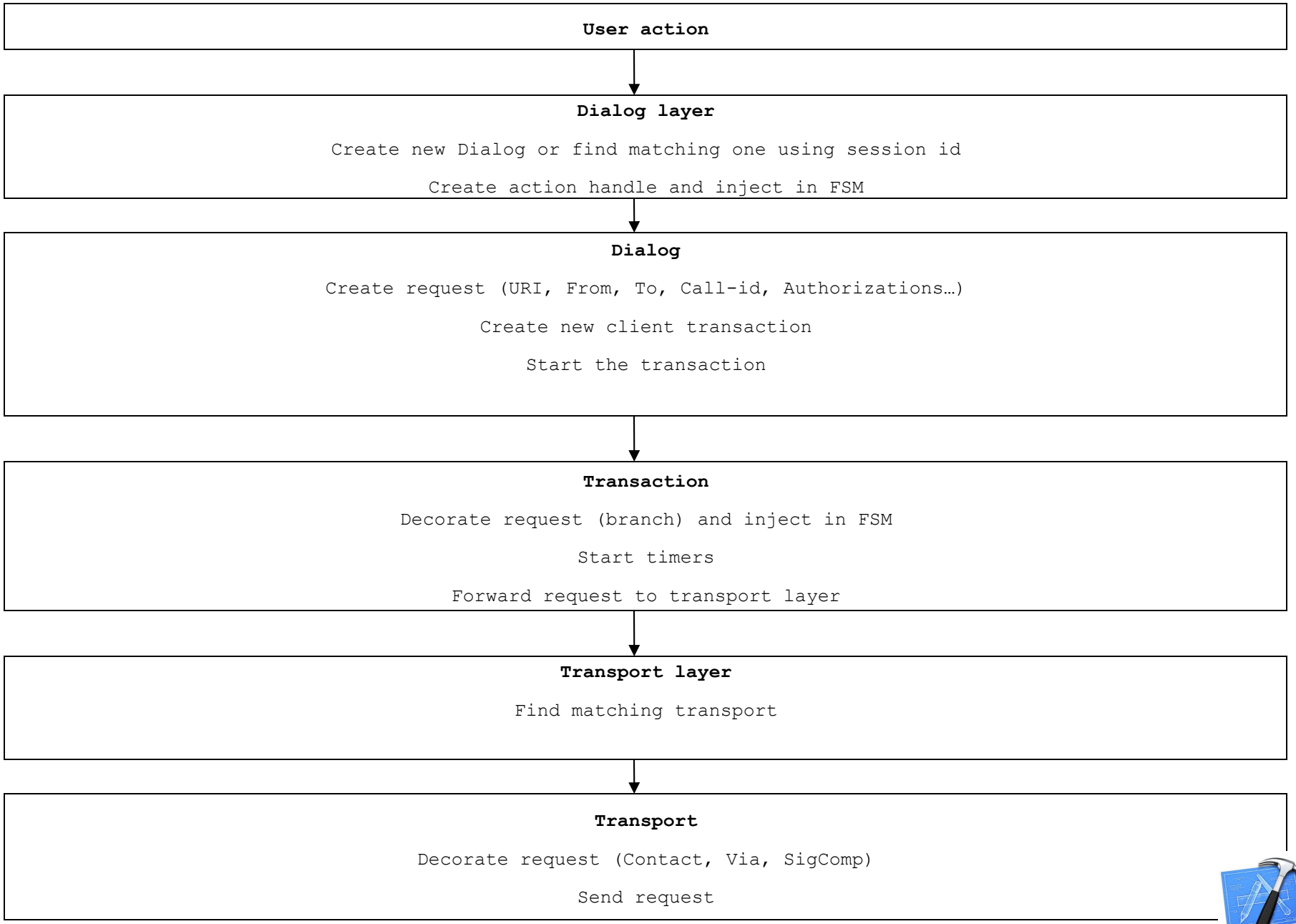
Transaction (2 / 2)



Stack

- Thread-safe
- Manage layers
  - Transport
  - Transaction
  - Dialogs
- SIP events notifications using worker thread
- Entry point for user actions
- Stack-level headers
- Stores user preferences
  - Credentials
  - TLS files
  - DNS servers
  - Connection information
  - ...

Sending request



Agenda

- Doubango Telecom
- Architecture
- Skipped technical subjects
- Debugging
- OOP in ANSI-C
- Ragel
- FSM
- Networking
- Plugins
- SIP
- Media
- Services
- iOS proxy video producer example
- iOS4+ multitasking
- Reported issues
- QR

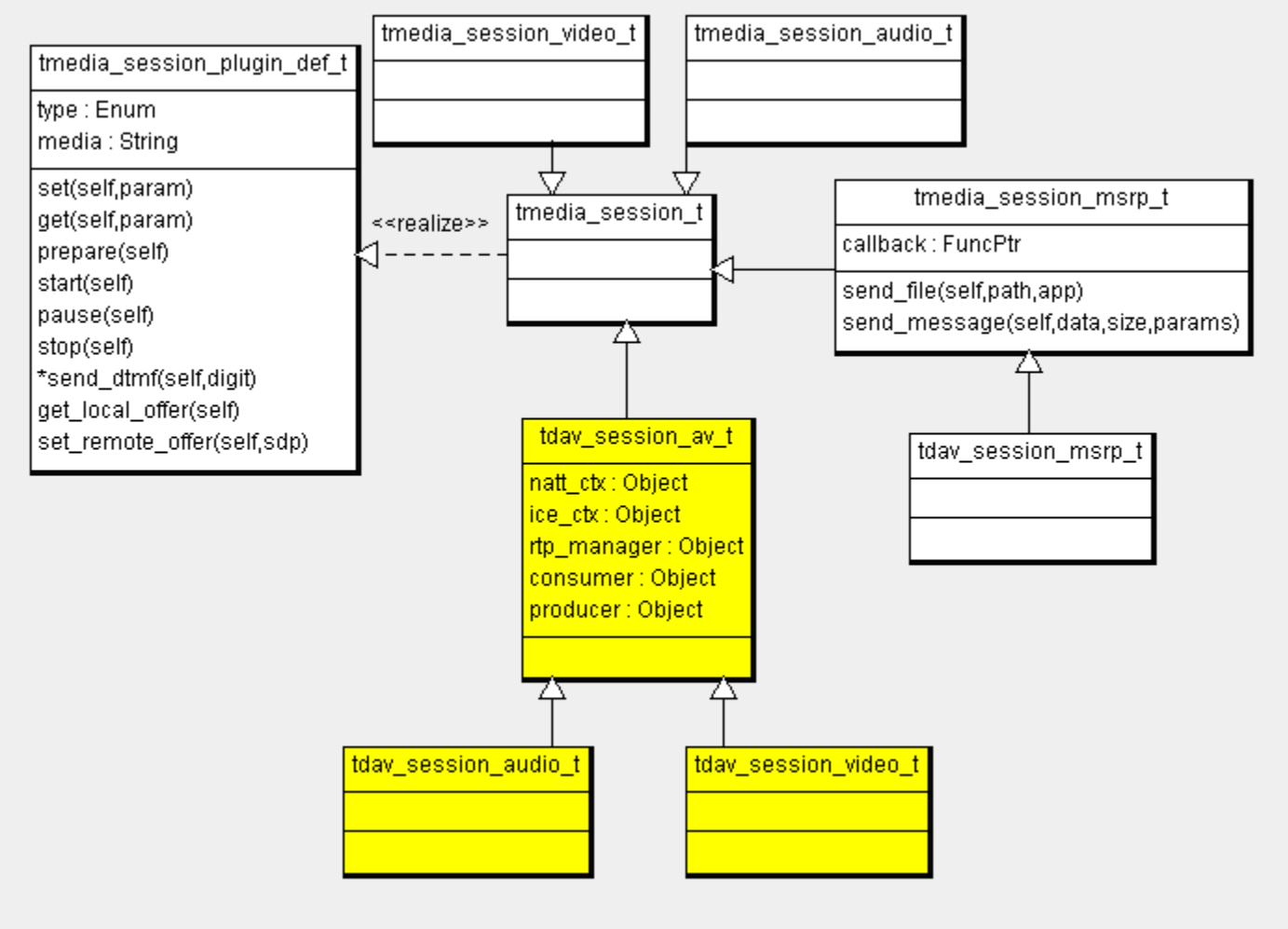
Overview

- Plugins definition: tinyMEDIA
- Plugins implementation: tinyDAV
- Media types: Audio, Video and MSRP (blob)
- Protocols: (S)RTP, (S)RTCP, MSRP and SDP
- Entry point: Manager
  - tmedia_session_mgr_t
  - Entry point for sessions creation and user actions :
    - hold/resume
    - start/stop
    - add/remove media
    - ...
  - Configured using SDP messages
  - No dependencies with SIP (see webrtc4ie)
  - Can be used with any protocol (e.g. XMPP)

Session (1 / 2)

- Types: Audio, video and MSRP as plugins
- Holds local and remote offer (m=line)
- Holds negotiated codecs
- Create / destroy consumers and producers
- Audio/Video
  - Receive outgoing data from the producer using callbacks
  - Encode the outgoing data using negotiated codes
  - Forward the encoded data to the RTP manager
  - Encapsulate data and send over the network
  - Receive incoming data from RTP manager
  - Decode RTP packets and store payload + header in the JB

Session (2 / 2)


tmedia_session_plugin_def_t
type : Enum
media : String
set(self,param)
get(self,param)
prepare(self)
start(self)
pause(self)
stop(self)
*send_dtmf(self,digit)
get_local_offer(self)
set_remote_offer(self,sdp)
tmedia_session_video_t
tmedia_session_audio_t
<<realize>>
tmedia_session_t
tmedia_session_msrp_t
callback : FuncPtr
send_file(self,path,app)
send_message(self,data,size,params)
tdav_session_av_t
natt_ctx : Object
ice_ctx : Object
rtp_manager : Object
consumer : Object
producer : Object
tdav_session_msrp_t
tdav_session_audio_t
tdav_session_video_t

(S)RTP / (S)RTCP (1 / 2)

- tinyRTP project
- Entry point: RTP manager
  - Send / receive (S)RTP packets
  - RTCP-MUX (RTC 5761)
  - Serialize / deserialize (S)RTP pkts
  - Dispatch (S)RTP / (S)RTCP pkts to the session
  - Decode and render the data
  - (S)RTCP
    - SR, RR, SDES, BYE, APP, PSFB and RTPFB
      - PSFB: PLI, FIR, AFB
      - RTPFB: NACK

(S)RTP / (S)RTCP (2 / 2)

Packet loss handling: Pseudo-code

```
if(packet_lost(n)){
  if(send_nack(n) == ok){ // pkt retransmitted
    // do nothing
  }
  else{ // the remote cannot retransmit the pkt
    if(pkt_type(n) == I || lost_chunck_is_config()){ // Intra frame or lost chunck contains config (e.g.
                                                     quant tables for VP8, SPS/PPS for H.264 etc...)
      send_fir();
    }
    else{ // predicted (P or B for H.264 or Gold for VP8 ...)
      if(decode(n) == -1){ // fails to decode/subjective quality is below a threshold
         send_fir();
      }
      else{
        if(subjective < threshold){
          send_pli(); // let the remote party send FIR or whatever based on the bandwidth mgt
        }
        pkt_display(n);
      }
    }
  }
}
```

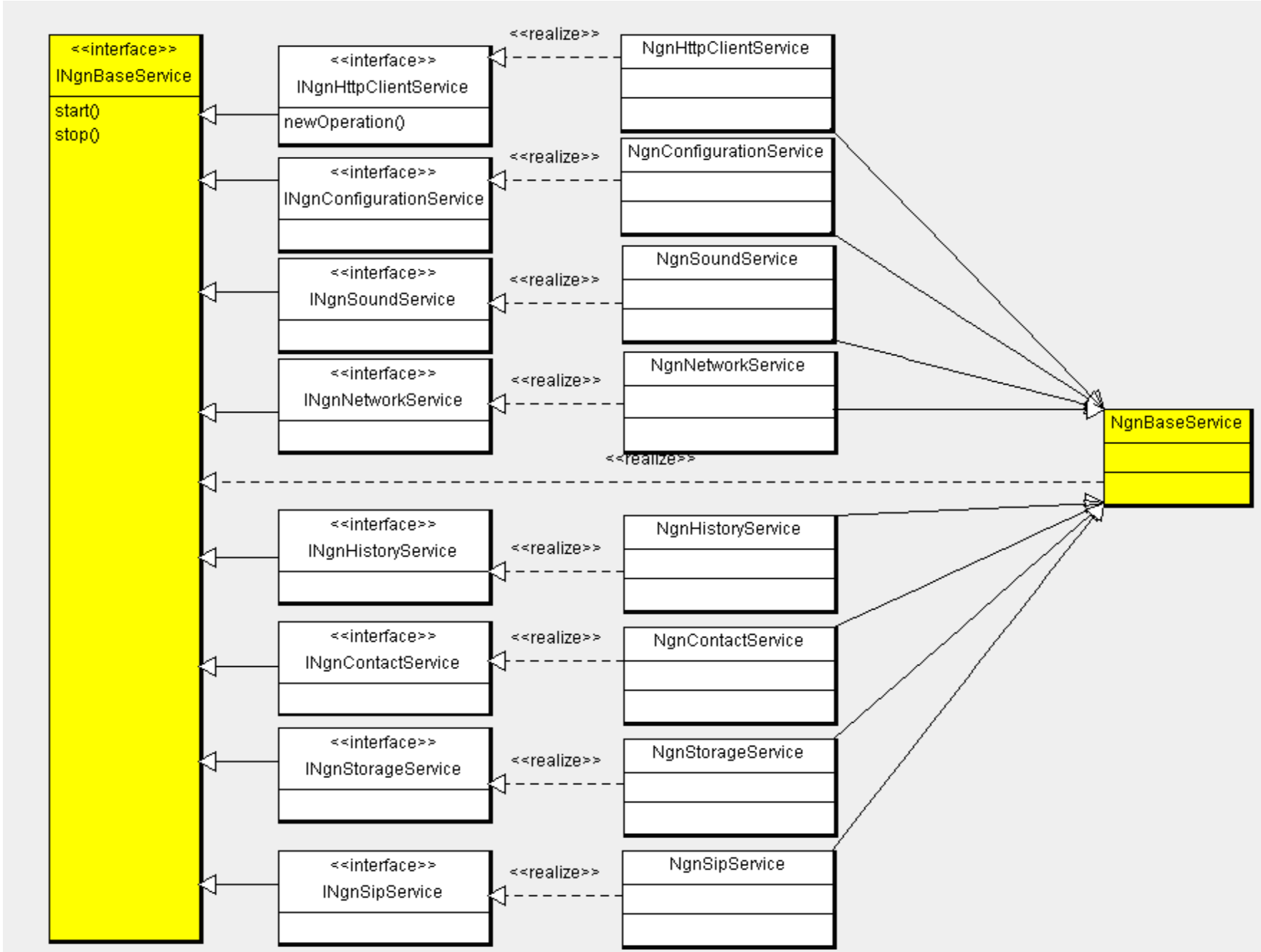
Sample code

```
tmedia_session_mgr_t* mgr;
const tsdp_message_t* sdp_lo;
tsdp_message_t* sdp_ro;

/* - create manager
 - create audio, video and MSRP sessions using plugins registry
- it's up to each session to create a consumer and producer using plugins registry
- audio/video sessions will create RTP manager
*/
mgr = tmedia_session_mgr_create((tmedia_audio | tmedia_video | tmedia_msrp),
      "0.0.0.0", tsk_false/*IPv4*/, tsk_true/*offerer*/);

/* - create empty SDP message
- for each session, request the corresponding media line ("m=") and append it to the empty SDP message
*/
sdp_lo = tmedia_session_mgr_get_lo(mgr);

/* - split remote SDP message per media ("m=" line) and call "session_set_ro(m)" for each session
*/
if((sdp_ro = tsdp_message_parse("sdp…", tsk_strlen("sdp…")))){
     tmedia_session_mgr_set_ro(mgr, sdp_ro);
     TSK_OBJECT_SAFE_FREE(sdp_ro);
}

tmedia_session_mgr_start(mgr);

TSK_OBJECT_SAFE_FREE(mgr);
```

Agenda

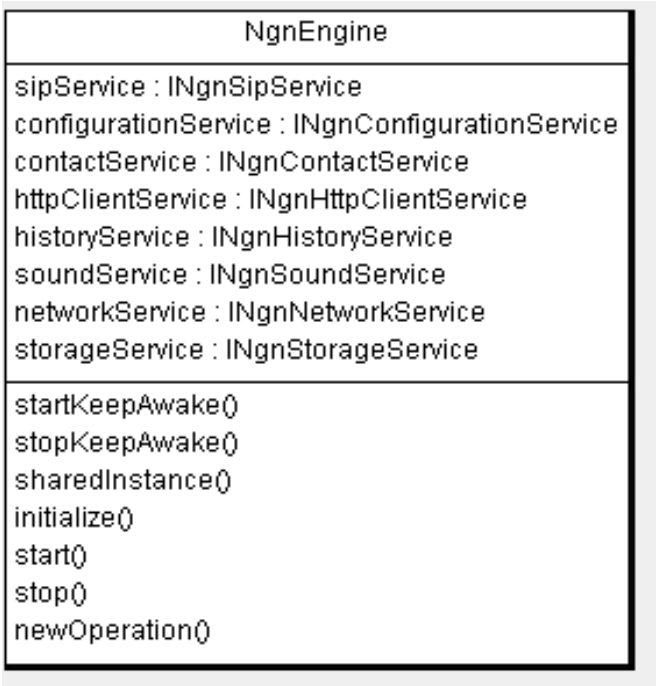
- Doubango Telecom
- Architecture
- Skipped technical subjects
- Debugging
- OOP in ANSI-C
- Ragel
- FSM
- Networking
- Plugins
- SIP
- Media
- Services
- iOS proxy video producer example
- iOS4+ multitasking
- Reported issues
- QR

Overview


<<interface>>
INgnBaseService
start()
stop()
<<interface>>
INgnHttpClientService
newOperation()
<<realize>>
NgnHttpClientService
<<interface>>
INgnConfigurationService
<<realize>>
NgnConfigurationService
<<interface>>
INgnSoundService
<<realize>>
NgnSoundService
<<interface>>
INgnNetworkService
<<realize>>
NgnNetworkService
<<realize>>
NgnBaseService
<<interface>>
INgnHistoryService
<<realize>>
NgnHistoryService
<<interface>>
INgnContactService
<<realize>>
NgnContactService
<<interface>>
INgnStorageService
<<realize>>
NgnStorageService
<<interface>>
INgnSipService
<<realize>>
NgnSipService

NgnEngine

- Root entry
- Singleton
- Access to all services
- Must be started before use
- Thread safe
- Fake multitasking managment
- NSAutoreleasePool for POSIX threads



NgnConfigurationService

- Stored in user preferences
- Not propagated to the native code
- Provide notifications when configuration is changed from settings

```
@protocol INgnConfigurationService <INgnBaseService>
    -(NSDictionary*) getDefaults;
    -(void)synchronize;
    -(NSString*)getStringWithKey: (NSString*)key;
    -(int)getIntWithKey: (NSString*)key;
    -(float)getFloatWithKey: (NSString*)key;
    -(BOOL)getBoolWithKey: (NSString*)key;
    -(void)setStringWithKey: (NSString*)key andValue:
      (NSString*)value;
    -(void)setIntWithKey: (NSString*)key andValue:(int)value;
    -(void)setFloatWithKey: (NSString*)key andValue:(float)value;
    -(void)setBoolWithKey: (NSString*)key andValue:(BOOL)value;
@end
```

NgnStorageService

- Stores data in sqlite3 database
- Database created when application starts for the first time using sql file
- Versioned using SQL #PRAGMA
- Stores application info, call history, favorites and IMs
- Used by NgnHistoryService to store history events

```
@protocol INgnStorageService <INgnBaseService>
            -(int) databaseVersion;
            -(sqlite3 *) database;
            -(BOOL) execSQL: (NSString*)sqlQuery;
            -(NSDictionary*) favorites;
            -(NgnFavorite*) favoriteWithNumber:(NSString*)number andMediaType:(NgnMediaType_t)mediaType;
            -(NgnFavorite*) favoriteWithNumber:(NSString*)number;
            -(BOOL) addFavorite: (NgnFavorite*) favorite;
            -(BOOL) deleteFavorite: (NgnFavorite*) favorite;
            -(BOOL) deleteFavoriteWithId: (long long) id;
            -(BOOL) clearFavorites;
@end
```

NgnHistoryService

- Stores history events (audio/video calls and messages) in sqlite3 database using storage service
- Unload entries if memory warnings

```
@protocol INgnHistoryService <INgnBaseService>
     -(BOOL) load;
     -(BOOL) isLoading;
     -(BOOL) addEvent: (NgnHistoryEvent*) event;
     -(BOOL) updateEvent: (NgnHistoryEvent*) event;
     -(BOOL) deleteEvent: (NgnHistoryEvent*) event;
     -(BOOL) deleteEventAtIndex: (int) location;
     -(BOOL) deleteEventWithId: (long long) eventId;
     -(BOOL) deleteEvents: (NgnMediaType_t) mediaType;
     -(BOOL) deleteEvents: (NgnMediaType_t) mediaType withRemoteParty:
        (NSString*)remoteParty;
     -(BOOL) deleteEventsArray: (NSArray*) events;
     -(BOOL) clear;
     -(NgnHistoryEventDictionary*) events;
@end
```

NgnNetworkService

- Network state monitoring and notifications
- Thread-safe

```
@protocol INgnNetworkService <INgnBaseService>
     -(NSString*)getReachabilityHostName;
     -(void)setReachabilityHostName:(NSString*)hostName;
     -(NgnNetworkType_t) getNetworkType;
     -(NgnNetworkReachability_t) getReachability;
     -(BOOL) isReachable;

     @property(readwrite, retain, getter=getReachabilityHostName,
        setter=setReachabilityHostName) NSString* reachabilityHostName;
     @property(readonly, getter=getNetworkType) NgnNetworkType_t networkType;
     @property(readonly, getter=getReachability) NgnNetworkReachability_t reachability;
     @property(readonly, getter=isReachable) BOOL reachable;
@end
```

NgnHttpClientService

- Sending receiving HTTP / HTTPS messages
- Thread-unsafe

```
@protocol INgnHttpClientService <INgnBaseService>

      -(NSData*) getSynchronously:(NSString*)uri;

      -(NSData*) postSynchronously:(NSString*) uri withContentData:(NSData*)contentData withContentType:
          (NSString*)contentType;

@end
```

NgnContactService

- Managing native contacts (load, add, remove and update)
- Indexed search
- Unload entries if memory warnings
- Thread-safe

```
@protocol INgnContactService <INgnBaseService>
     -(void) load:(BOOL) asyn;
     -(void) unload;
     -(BOOL) isLoading;
     -(NSArray*) contacts;
     -(NSDictionary*) numbers2ContactsMapper;
     -(NSArray*) contactsWithPredicate:(NSPredicate*)predicate;
     -(NgnContact*) getContactByUri:(NSString*)uri;
     -(NgnContact*) getContactByPhoneNumber:(NSString*)phoneNumber;
@end
```

NgnSIPService

- Stores SIP credentials
- SIP registration / deregistration
- SIP notifications using NSNotificationCenter
- Thread-safe

```
@protocol INgnSipService <INgnBaseService>
      -(NSString*)getDefaultIdentity;
      -(void)setDefaultIdentity: (NSString*)identity;
      -(NgnSipStack*)getSipStack;
      -(BOOL)isRegistered;
      -(ConnectionState_t)getRegistrationState;
      -(int)getCodecs;
      -(void)setCodecs: (int)codecs;
      -(BOOL)stopStackAsynchronously;
      -(BOOL)stopStackSynchronously;
      -(BOOL)registerIdentity;
      -(BOOL)unRegisterIdentity;


      @property(readwrite, retain, getter=getDefaultIdentity, setter=setDefaultIdentity) NSString*
          defaultIdentity;
      @property(readonly, getter=getSipStack) NgnSipStack* stack;
      @property(readonly, getter=isRegistered) BOOL registered;
      @property(readonly, getter=getRegistrationState) ConnectionState_t registrationState;
      @property(readwrite, getter=getCodecs, setter=setCodecs) int codecs;
@end
```

Agenda

- Doubango Telecom
- Architecture
- Skipped technical subjects
- Debugging
- OOP in ANSI-C
- Ragel
- FSM
- Networking
- Plugins
- SIP
- Media
- Services
- iOS proxy video producer example
- iOS4+ multitasking
- Reported issues
- QR

iOS proxy video producer

Initialization

[NgnProxyPluginMgr.mm]

```
[c++]class [_NgnProxyPluginMgrCallback: ProxyPluginMgrCallback] =
{
    OnPluginCreated(id, type){
     store(new NgnPlugin(id));
    };
    OnPluginDestroyed(id, type){
     remove(id)
    };
};
_NgnProxyPluginMgrCallback _sMyProxyPluginMgrCallback;

ProxyVideoProducer::registerPlugin();
ProxyVideoConsumer::registerPlugin();

ProxyVideoConsumer::setDefaultChroma(tmedia_chroma_rgb32);
ProxyVideoConsumer::setDefaultAutoResizeDisplay(YES);
ProxyVideoProducer::setDefaultChroma(tmedia_chroma_nv12);

_sMyProxyPluginMgrCallback = new _NgnProxyPluginMgrCallback();
_sPluginMgr = ProxyPluginMgr::createInstance(_sMyProxyPluginMgrCallback);
```

iOS proxy video producer

Obj-c implementation

```
[iOSProxyVideoProducer.mm or OSXProxyVideoProducer.mm]
[objc]class [NgnProxyVideoProducer:NgnProxyPlugin] (_producer:ProxyVideoProducer)=
{
     _mProducer = [native]_producer;
     Int prepare(int width, int heigth, int fps) = { <<AVFoundation or QT>> };
     Int start() = {<<AVFoundation or QT>> }
     Int pause() = { <<AVFoundation or QT>> }
     Int stop() = {<<AVFoundation or QT>> }
}
```

iOS proxy video producer

Proxying

[iOSProxyVideoProducer.mm or OSXProxyVideoProducer.mm]

```
[c++]class [_NgnProxyVideoProducerCallback:ProxyVideoProducerCallback]
    (producer:NgnProxyVideoProducer)=

{

     mProducer = [retain] producer;

     [Pool] Int prepare(int width, int heigth, int fps) = { return
        mProducer.prepare(width, height, fps)};

     [Pool] Int start() = { return mProducer.start(); }

     [Pool] Int pause() = { return mProducer.pause(); }

     [Pool] Int stop() = { mProducer.stop(); }

}
```

Agenda

- Doubango Telecom
- Architecture
- Skipped technical subjects
- Debugging
- OOP in ANSI-C
- Ragel
- FSM
- Networking
- Plugins
- SIP
- Media
- Services
- iOS proxy video producer example
- iOS4+ multitasking
- Reported issues
- QR

iOS4+ multitasking

Unofficial

- Silent sound when application in background

Official

- Requires TCP transport
- TCP sockets tagged as VoIP (idoubs-Info.plist)
- Application tagged as Audio (idoubs-Info.plist)
- All threads and timers are suspended when application transits from foreground to background unless ‘UIBackgroundTask’ token is registered
- 10 minutes to finish operations when ‘UIBackgroundTask’ token is registered before idle state
- Only ONE ‘UIBackgroundTask’ token can be requested per transition
- Application is woken up every 10 minutes (time to refresh SIP registration) or if there are incoming TCP packets

Agenda

- Doubango Telecom
- Architecture
- Skipped technical subjects
- Debugging
- OOP in ANSI-C
- Ragel
- FSM
- Networking
- Plugins
- SIP
- Media
- Services
- iOS proxy video producer example
- iOS4+ multitasking
- Reported issues
- QR

Agenda

- Doubango Telecom
- Architecture
- Skipped technical subjects
- Debugging
- OOP in ANSI-C
- Ragel
- FSM
- Networking
- Plugins
- SIP
- Media
- Services
- iOS proxy video producer example
- iOS4+ multitasking
- Reported issues
- QR

Q/R

Inspiring The Future

Mamadou Diop

Thank You!

Inspiring The Future

Mamadou Diop