PJSIP PROGRAMING INTERFACE

Programming interface PJSIP

Day #1	
Waktu	Agenda
09:00 - 10:00	SIP Overview
10:00 - 12:00	PJSIP Architecture
12:00 - 13:00	Lunch & Break
13:00 - 14:00	PJSIP Installation & Configuration
14:00 - 15:00	PJSIP Client "Hello Word"
15:00 - 15:30	Break
15:30 - 17:00	Creating PJSIP SimpleEcho Program

Day #2	
Waktu	Agenda
09:00 - 10:00	PJSUA Overview
10:00 - 12:00	PJSUA Endpoint
12:00 - 13:00	Lunch & Break
13:00 - 14:00	PJSUA Accounts (Creating Userless Accounts)
14:00 - 15:00	- Creating Account
15:00 - 15:30	Break
15:30 - 17:00	- Account Configuration & Operation

Day #3	
Waktu	Agenda
09:00 - 10:00	PJSUA Media
10:00 - 12:00	Audio Device Management
12:00 - 13:00	Lunch & Break
13:00 - 14:00	PJSUA Calls
14:00 - 15:00	Making Outgoing Calls
15:00 - 15:30	Break
15:30 - 17:00	Receiving Incoming Calls

Day #4	
Waktu	Agenda
09:00 - 10:00	Working with Call’s Audio Media
10:00 - 12:00	Call Settings
12:00 - 13:00	Lunch & Break
13:00 - 15:00	Call Operations #1
15:00 - 15:30	Break

Day #5	
Waktu	Agenda
09:00 - 10:00	Python with PJSUA
10:00 - 12:00	
12:00 - 13:00	Lunch & Break
13:00 - 15:00	Building All SIP Client
15:00 - 15:30	Break

SIP OVERVIEW

- SIP (Session Initiation Protocol) is a signaling protocol used to create, manage and terminate sessions in an IP based network. A session could be a simple two-way telephone call or it could be a collaborative multi-media conference session. This makes possible to implement services like voice-enriched e-commerce, web page click-to-dial or Instant Messaging with buddy lists in an IP based environment. Don't worry if you don't know about these services. You don't need to know them before you learn about SIP.

Components of SIP

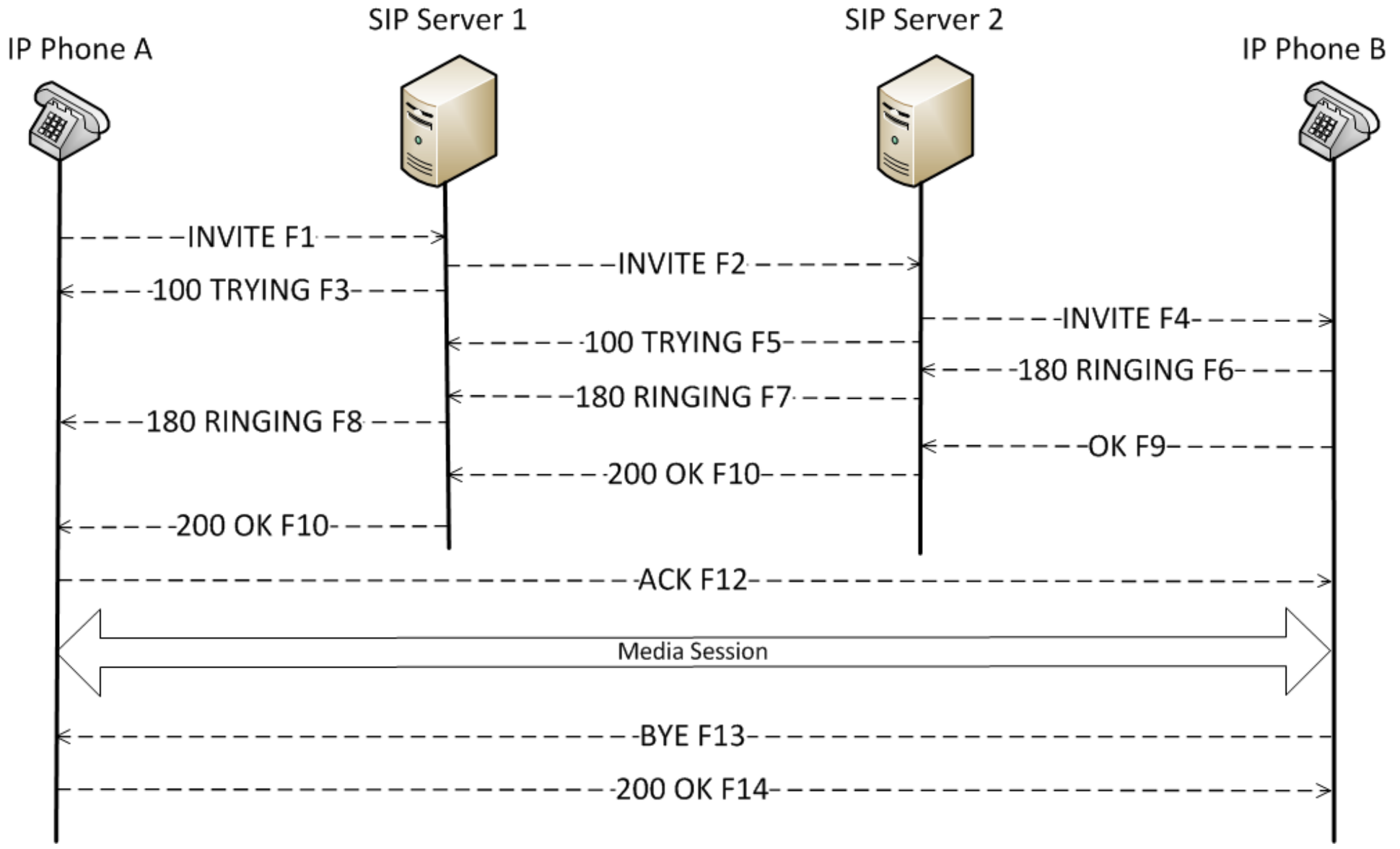
- User Agent Client (UAC) : It generates requests and send those to servers.

- User Agent Server (UAS) : It gets requests, processes those requests and generate responses.

Commands of SIP

- **INVITE** :Invites a user to a call
- **ACK** : Acknowledgement is used to facilitate reliable message exchange for INVITEs.
- **BYE** :Terminates a connection between users
- **CANCEL** :Terminates a request, or search, for a user. It is used if a client sends an INVITE and then changes its decision to call the recipient.
- **OPTIONS** :Solicits information about a server's capabilities.
- **REGISTER** :Registers a user's current location
- **INFO** :Used for mid-session signaling

Commands of SIP



Request Message Format of SIP

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 36.89.173.222:5060;received=36.89.173.222;branch=z9hG4bKf19c.e69d64f6.0
Via: SIP/2.0/UDP 192.168.1.100:5060;rport=1033;received=36.71.24.132;branch=z9hG4bKPjGYZTLnJTrvV2APN5TGLWhNqNQOOlvIRb
Record-Route: <sip:36.89.173.222;lr;ftag=FezYAbjRCLjIGsToghr7rzioEqvbSRlR;nat=yes>
Call-ID: CeEKOptEorFvF2j6N4RH7V2rHgw8FGZ9
From: "16005" <sip:16005@36.89.173.222>;tag=FezYAbjRCLjIGsToghr7rzioEqvbSRlR
To: <sip:16004@36.89.173.222>
CSeq: 12476 INVITE
Content-Length:  0
```

Response Types of SIP

- **1xx: Provisional** -- request received, continuing to process the request;
- **2xx: Success** -- the action was successfully received, understood, and accepted;
- **3xx: Redirection** -- further action needs to be taken in order to complete the request;
- **4xx: Client Error** -- the request contains bad syntax or cannot be fulfilled at this server;
- **5xx: Server Error** -- the server failed to fulfill an apparently valid request;
- **6xx: Global Failure** -- the request cannot be fulfilled at any server.

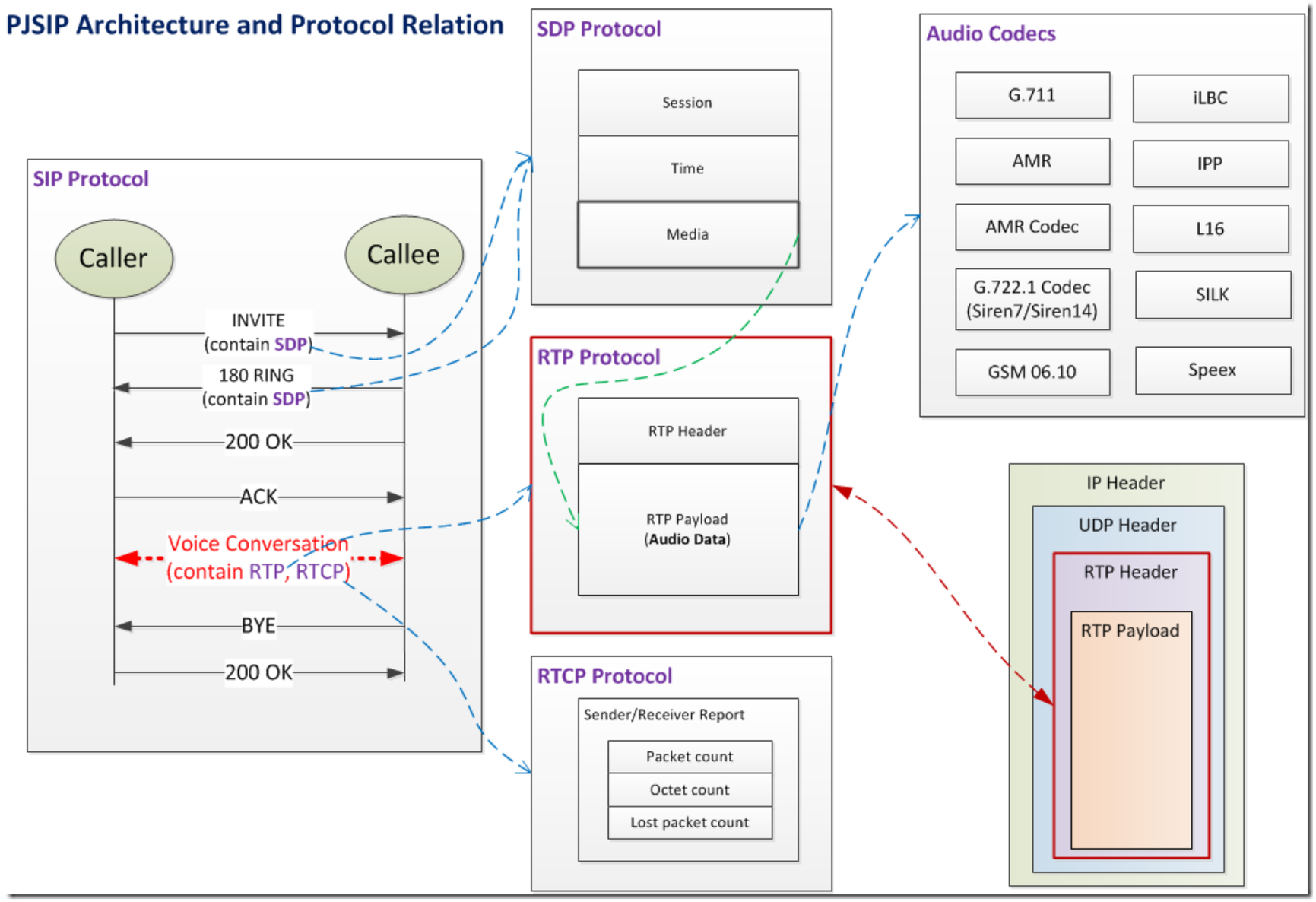
PJSIP

- PJSIP is a free and open source multimedia communication library written in C language implementing standard based protocols such as SIP, SDP, RTP, STUN, TURN, and ICE. It combines signaling protocol (SIP) with rich multimedia framework and NAT traversal functionality into high level API that is portable and suitable for almost any type of systems ranging from desktops, embedded systems, to mobile handsets.

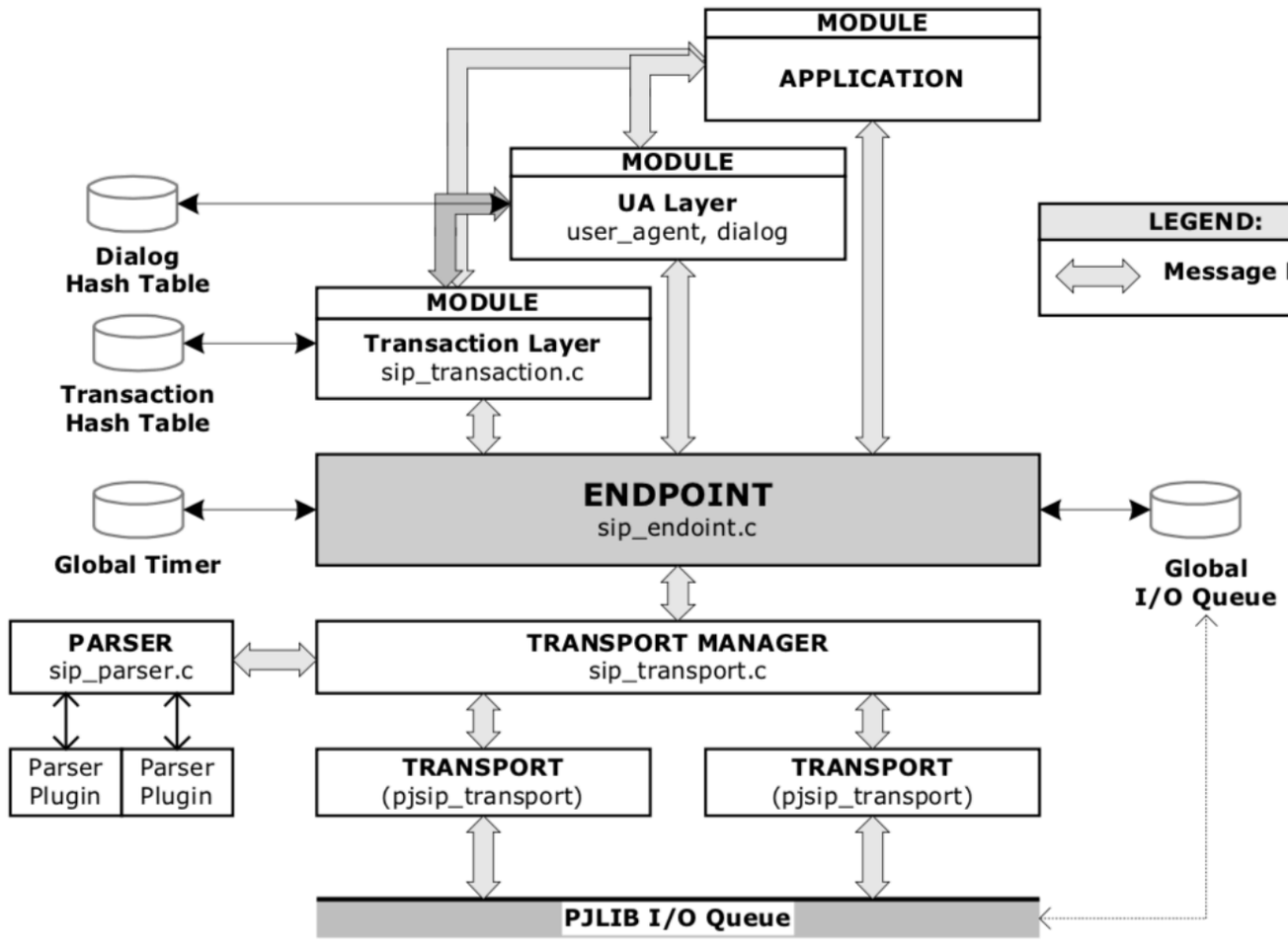
PJSIP


PJSIP
PJMEDIA
PJSIP
Transport, Parsing, Transaction Layer,
Dialog Layer, Authentication, etc.
PJNATH
ICE, STUN, TURN
PJLIB-UTIL

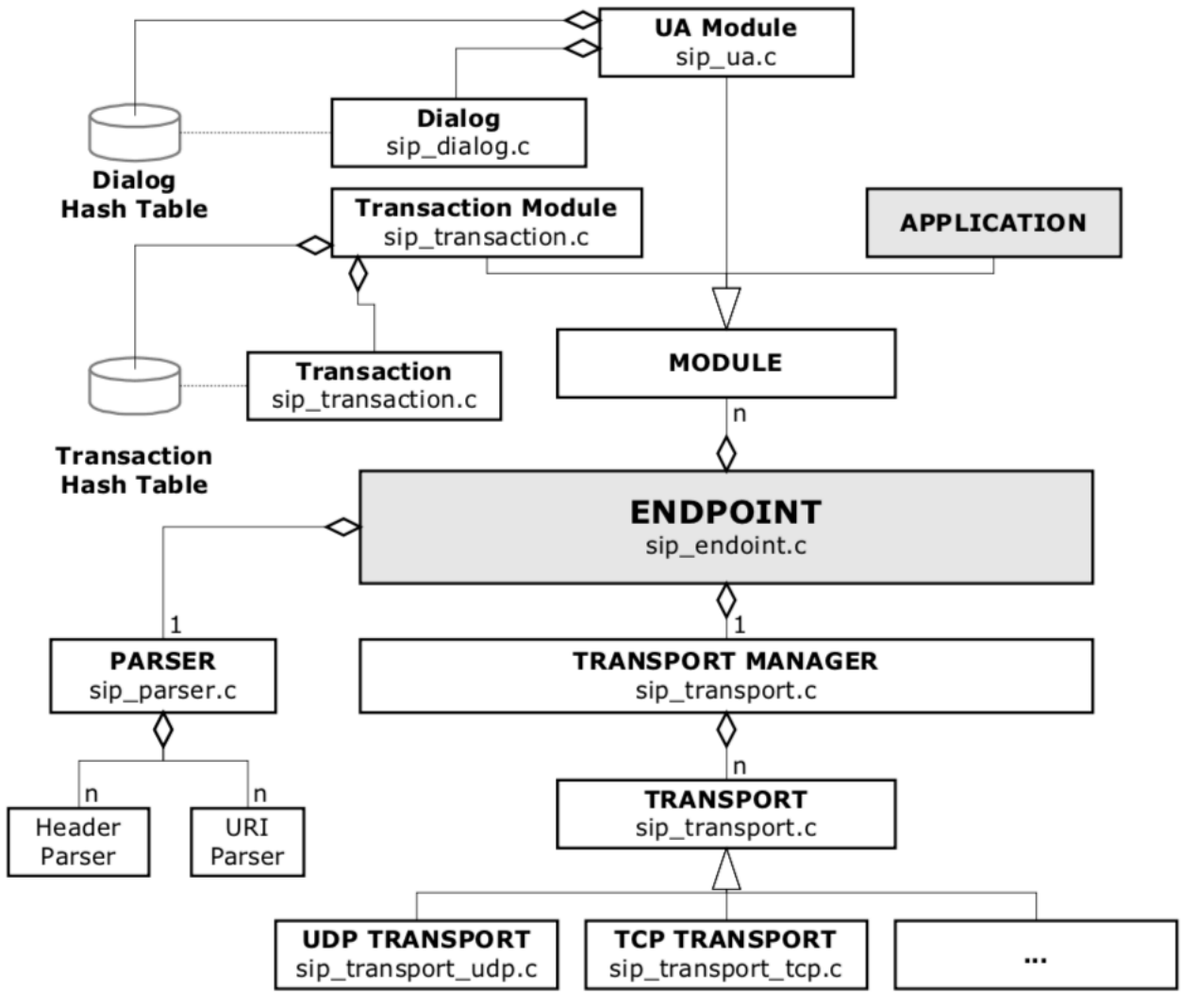
PJSIP Architecture



PJSIP Communication Diagram



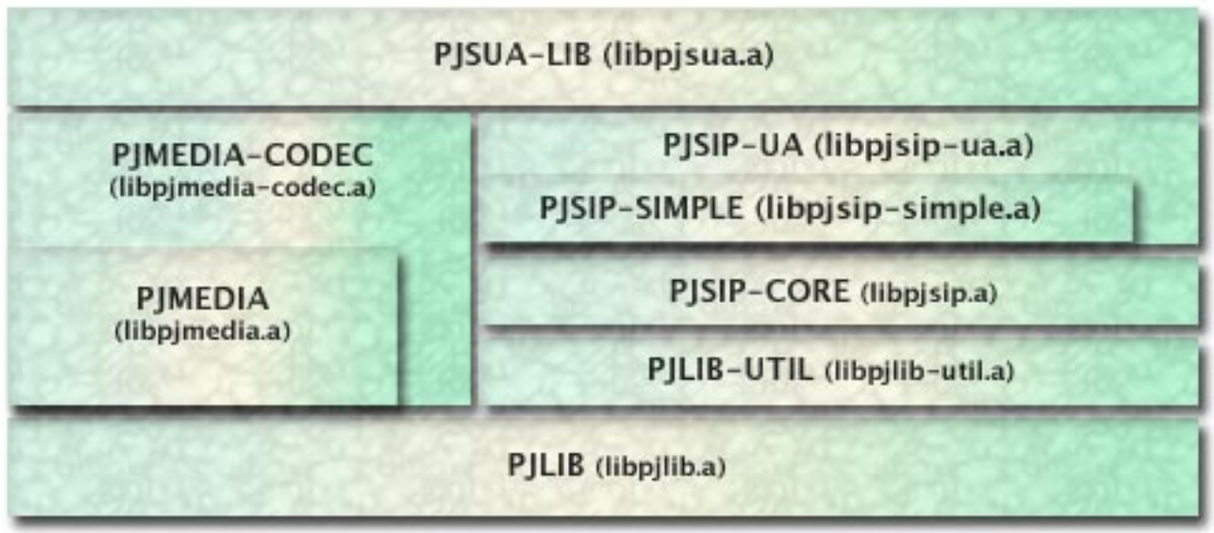
PJSIP Class Diagram



SIP Capabilities

- Base specs:
  - Core methods: RFC 3261: INVITE, CANCEL, BYE, REGISTER, OPTIONS, INFO
  - Digest authentication (RFC 2617)
- Transports:
  - UDP, TCP, TLS (server or mutual)
  - DNS SRV resolution (RFC 3263)
  - IPv6
  - QoS (DSCP, WMM)
- Routing/NAT:
  - rport (RFC 3581)
  - Service-Route header (RFC 3608)
  - SIP outbound for TCP/TLS (RFC 5626)
  - Path header (with SIP outbound) (RFC 3327)
- Call:
  - Offer/answer (RFC 3264)
  - hold, unhold
  - redirection
  - transfer/REFER (attended and unattended):
    - Base (RFC 3515)
    - replaces (RFC 3891)
    - Referred-by (RFC 3892)
  - sipfrag support (RFC 3420)
  - norefersub (RFC 4488)
  - UPDATE (RFC 3311)
  - 100rel/PRACK (RFC 3262)
  - tel: URI (RFC 3966)
  - Session Timers (RFC 4028)
  - Reason header (RFC 3326, partially supported)
  - P-Header (RFC 3325, partially supported)
- SDP:
  - RFC 2327 (obsoleted by RFC 4566)
  - RTCP attribute (RFC 3605)
  - IPv6 support (RFC 3266)
- Multipart (RFC 2046, RFC 5621)
- Presence and IM:
  - Event framework (SUBSCRIBE, NOTIFY) (RFC 3265)
  - Event publication (PUBLISH) (RFC 3903)
- Presence and IM:
  - Event framework (SUBSCRIBE, NOTIFY) (RFC 3265)
  - Event publication (PUBLISH) (RFC 3903)
  - MESSAGE (RFC 3428)
  - typing indication (RFC 3994)
  - pidf+xml (RFC 3856, RFC 3863)
  - xpidf+xml
  - RPID (subset) (RFC 4480)
- Other extensions:
  - INFO (RFC 2976)
  - AKA, AKA-v2 authentication (RFC 3310, RFC 4169)
  - ICE option tag (RFC 5768)
  - Message summary/message waiting indication (MWI, RFC 3842)
- Compliance, best current practices:
  - Issues with Non-INVITE transaction (RFC 4320)
  - Issues with INVITE transaction (RFC 4321)
  - Multiple dialog usages (RFC 5057)
  - SIP torture messages (RFC 4475, tested when applicable)
  - SIP torture for IPv6 (RFC 5118)
  - Message Body Handling (RFC 5621. Partial compliance: multipart is supported, but *Content-Disposition* header is not handled)
  - The use of SIPS (RFC 5630. Partial compliance: SIPS is supported, but still make use of *transport=tls* parameter)

PJSIP Modules



PJSIP Static Library

- Static Library Layout
  Enumerating the static libraries from the bottom:
    - PJLIB, is the platform abstraction and framework library, on which all other libraries depend,
    - PJLIB-UTIL, provides auxiliary functions such as text scanning, XML, and STUN,
    - PJMEDIA is the multimedia framework,
    - PJMEDIA-CODEC is the placeholder for media codecs,
    - Core SIP Library (PJSIP-CORE) is the very core of the PJSIP library, and contains the SIP Endpoint, which is the owner/manager for all SIP objects in the application, messaging elements, parsing, transport management, module management, and stateless operations, and also contains:
        - The Transaction Layer module inside PJSIP-CORE provides stateful operation, and is the base for higher layer features such as dialogs,
        - The Base User Agent Layer/Common Dialog Layer module inside PJSIP-CORE manages dialogs, and supports dialog usages,
    - Event and Presence Framework (PJSIP-SIMPLE) provides the base SIP event framework (which uses the common/base dialog framework) and implements presence on top of it, and is also used by call transfer functions,
    - User Agent Library (PJSIP-UA) is the high level abstraction of INVITE sessions (using the common/base dialog framework). This library also provides SIP client registration and call transfer functionality,
    - and finally, PJSUA API - High Level Softphone API (PJSUA-LIB) is the highest level of abstraction, which wraps together all above functionalities into high level, easy to use API.

PJSIP Installation

- PJSIP Version : 2.6
- Working Directory : **/home/bssn/**
- Download file pjproject-2.6.tar.bz2

```
wget http://www.pjsip.org/release/2.6/pjproject-2.6.tar.bz2
```

- Extract file

```
tar -jxvf pjproject-2.6.tar.bz2
```

- Rename folder to **pjproject**

```
mv pjproject-2.6 pjproject
```

PJSIP Installation

- Building PJSIP

```
cd pjproject
```

```
./configure --enable-shared
```

```
make && make install
```

Using PJPROJECT with Applications

Regardless of the build system being used, the following tasks are normally needed to be done in order to build application to use PJSIP and PJMEDIA:

1. Put these include directories in the include search path:

    - pjlib/include
    - pjlib-util/include
    - pjnath/include
    - pjmedia/include
    - pjsip/include

2. Put these library directories in the library search path:

    - pjlib/lib
    - pjlib-util/lib
    - pjnath/lib
    - pjmedia/lib
    - pjsip/lib

Using PJPROJECT with Applications

Include the relevant PJ header files in the application source file. For example, using these would include ALL APIs exported by PJ:

```
#include <pjlib.h>
#include <pjlib-util.h>
#include <pjnath.h>
#include <pjsip.h>
#include <pjsip_ua.h>
#include <pjsip_simple.h>
#include <pjsua-lib/pjsua.h>
#include <pjmedia.h>
#include <pjmedia-codec.h>
```

PJPROJECT Environment

PJDIR	/home/bssn/pjproject/
PROJECT APP DIR	/home/bssn/pjproject/latihan-apps
	./bin
	./build
	Makefile
	Latihan.mak
	./docs
	./lib
	./src/apps

```
GIT Repository
https://github.com/deje212/pjsip_training
```

```
cd build
make latihan
```

PJSUA

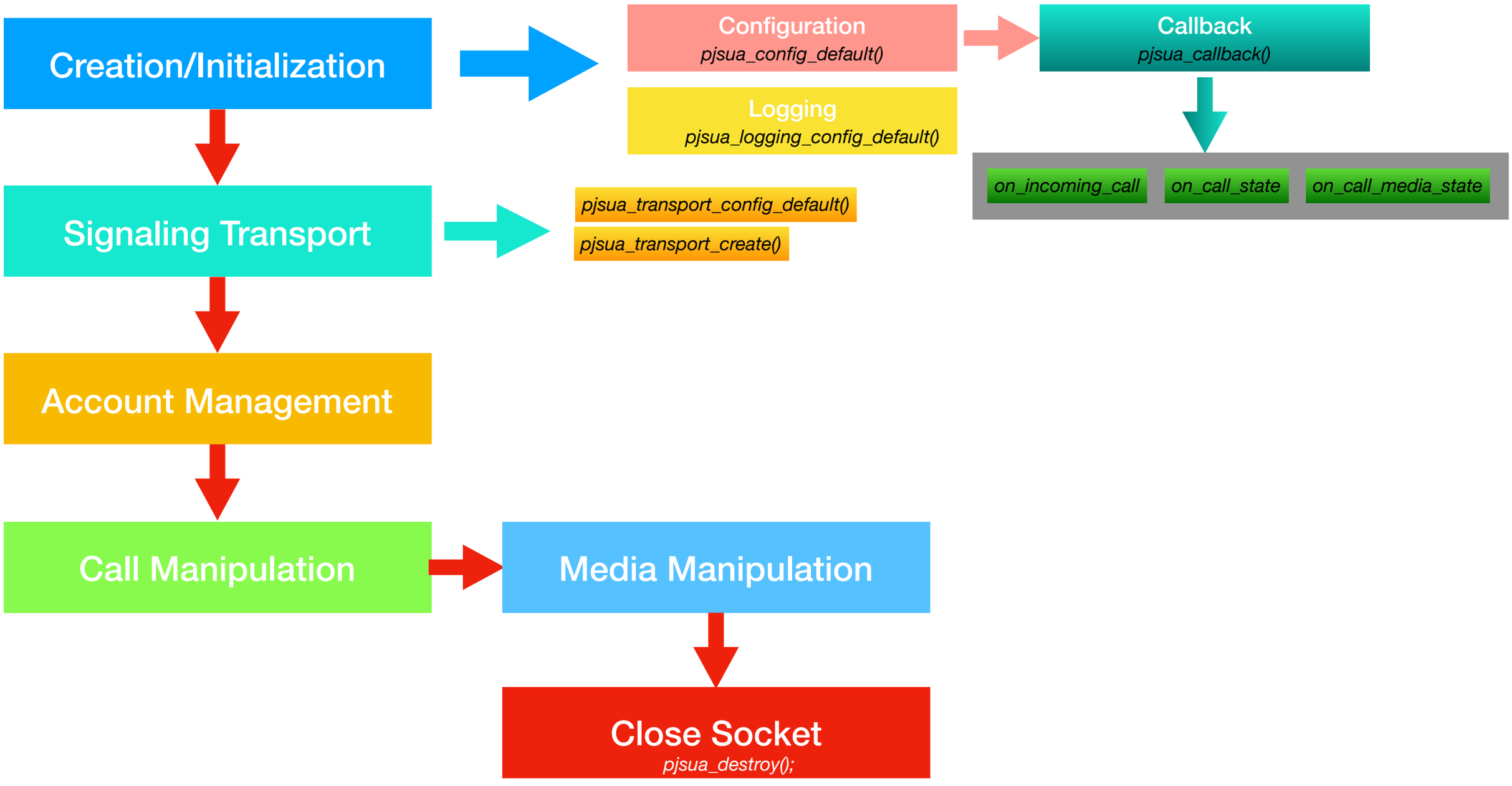
- Very high level API for constructing SIP UA applications.

Modules

	PJSUA-API Basic API
	Basic application creation/initialization, logging configuration, etc.
	PJSUA-API Signaling Transport
	API for managing SIP transports.
	PJSUA-API Accounts Management
	PJSUA Accounts management.
	PJSUA-API Calls Management
	Call manipulation.
	PJSUA-API Buddy, Presence, and Instant Messaging
	Buddy management, buddy's presence, and instant messaging.
	PJSUA-API Media Manipulation
	Media manipulation.
	PJSUA-API Video
	Video support.

https://www.pjsip.org/docs/latest/pjsip/docs/html/group__PJSUA__LIB.htm

PJSUA


Creation/Initialization
Configuration
pjsua_config_default()
Logging
pjsua_logging_config_default()
Callback
pjsua_callback()
on_incoming_call
on_call_state
on_call_media_state
Signaling Transport
pjsua_transport_config_default()
pjsua_transport_create()
Account Management
Call Manipulation
Media Manipulation
Close Socket
pjsua_destroy();

PJSUA-API Basic API

- The base PJSUA API controls PJSUA creation, initialization, and startup, and also provides various auxiliary functions.

- Before anything else, application must create PJSUA by calling ***pjsua_create()***. This, among other things, will initialize PJLIB, which is crucial before any PJLIB functions can be called, PJLIB-UTIL, and create a SIP endpoint.

- After this function is called, application can create a memory pool (with ***pjsua_pool_create()***) and read configurations from command line or file to build the settings to initialize PJSUA below.

PJSUA-API Basic API

After PJSUA is initialized with **pjsua_init()**, application will normally need/want to perform the following tasks:

1. create SIP transport with ***pjsua_transport_create()***. Application would to call ***pjsua_transport_create()*** for each transport types that it wants to support (for example, UDP, TCP, and TLS).

2. create one or more SIP accounts with ***pjsua_acc_add()*** or ***pjsua_acc_add_local()***. The SIP account is used for registering with the SIP server, if any.

3. add one or more buddies with ***pjsua_buddy_add()***

PJSUA2

- PJSUA2 is an object-oriented abstraction above PJSUA API. It provides high level API for constructing Session Initiation Protocol (SIP) multimedia user agent applications (a.k.a Voice over IP/VoIP softphones). It wraps together the signaling, media, and NAT traversal functionality into easy to use call control API, account management, buddy list management, presence, and instant messaging, along with multimedia features such as local conferencing, file streaming, local playback, and voice recording, and powerful NAT traversal techniques utilizing STUN, TURN, and ICE.