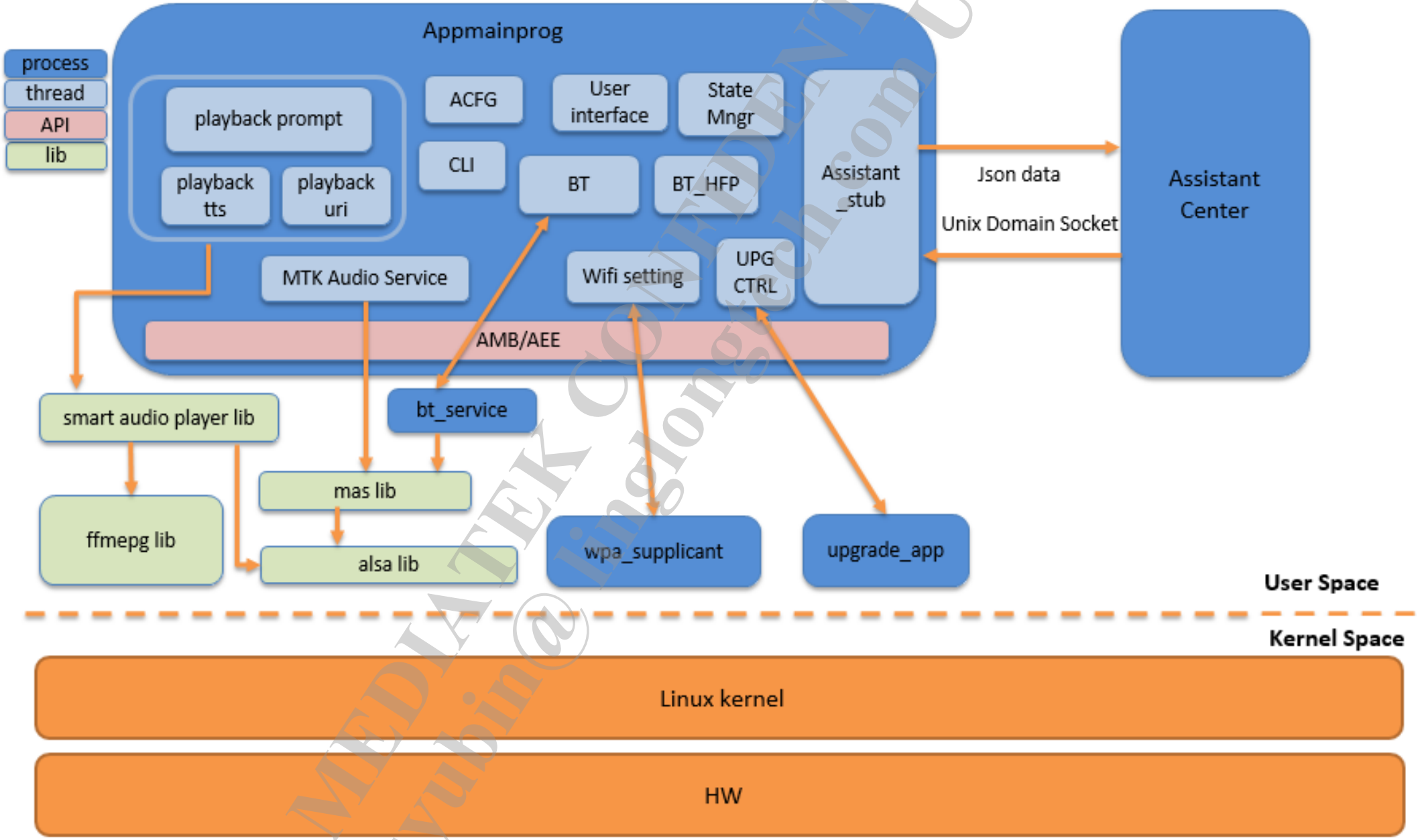
Assistant Stub Intro

Presenter： Pingan.Liu
Department:： MSZ_HSD_SW9_SS3

Agenda

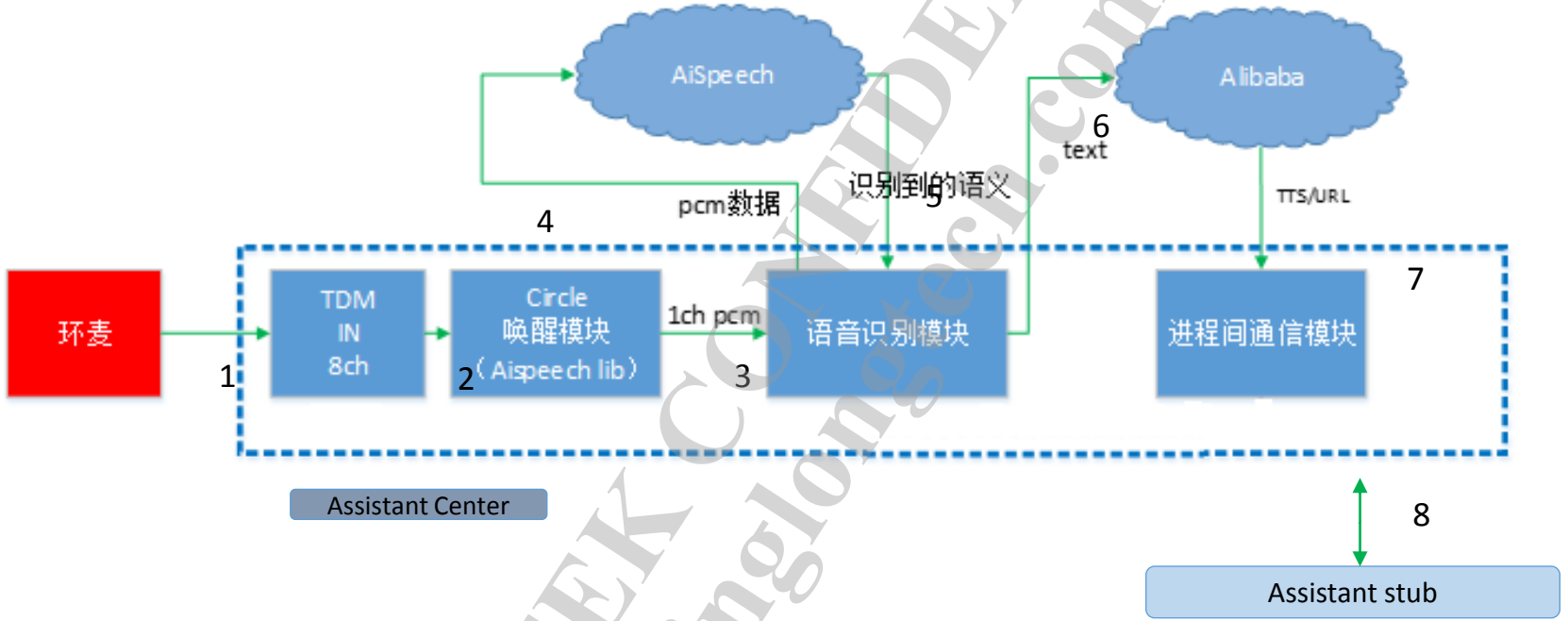
- MT8516 Software Architecture
- App Feature Introduction
- Code Introduction
- Q&A

MT8516 Software Architecture



App Feature Introduce

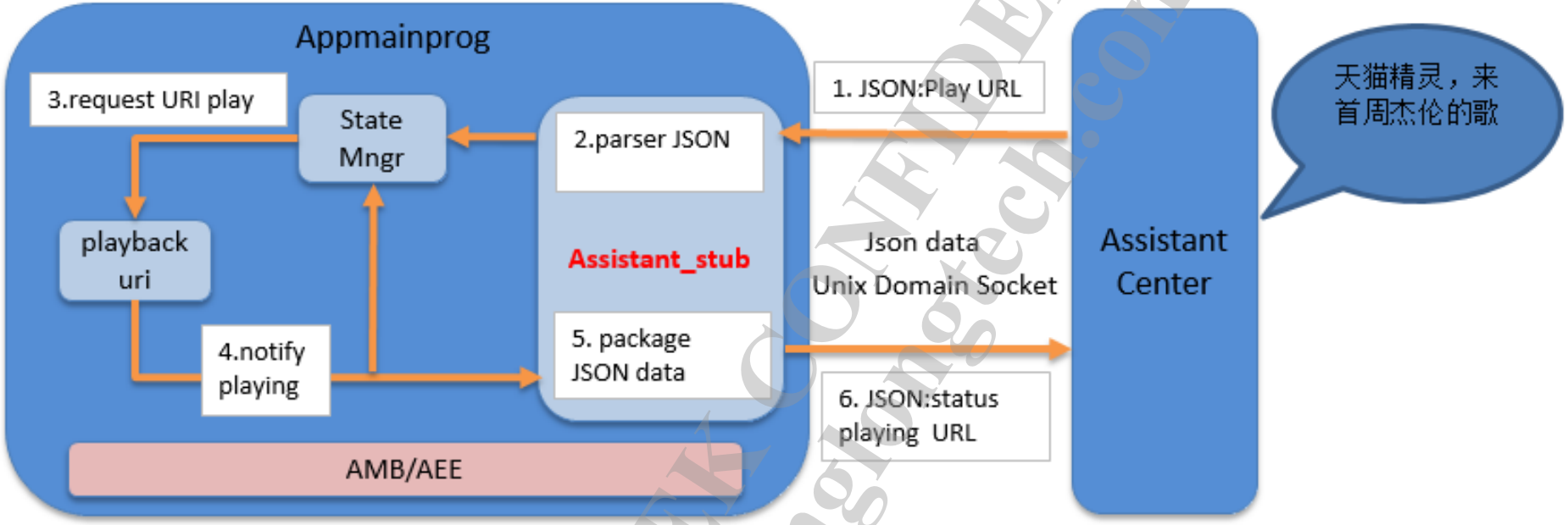
- **Assistant Center Process**



- Step 1: 通过alsa lib read 8路TDM in音频
- Step 2: 唤醒模块识别主要是将8路PCM数据合成1路
- Step 4/5: 语义识别模块将合成的1路PCM数据送往Aispeech语端，作用是将音频识别成字符串并返回
- Step 6/7:将识别后的字串以文本方式上传给客户服务器，并由客户服务器理解并转换成最终要平台端播放tts或者url或者是本地提示音prompt
- Step 8：将识别后的语义命令通过json格式打包并发往appmainprog的assistant stub 接收线程

App Feature Introduce

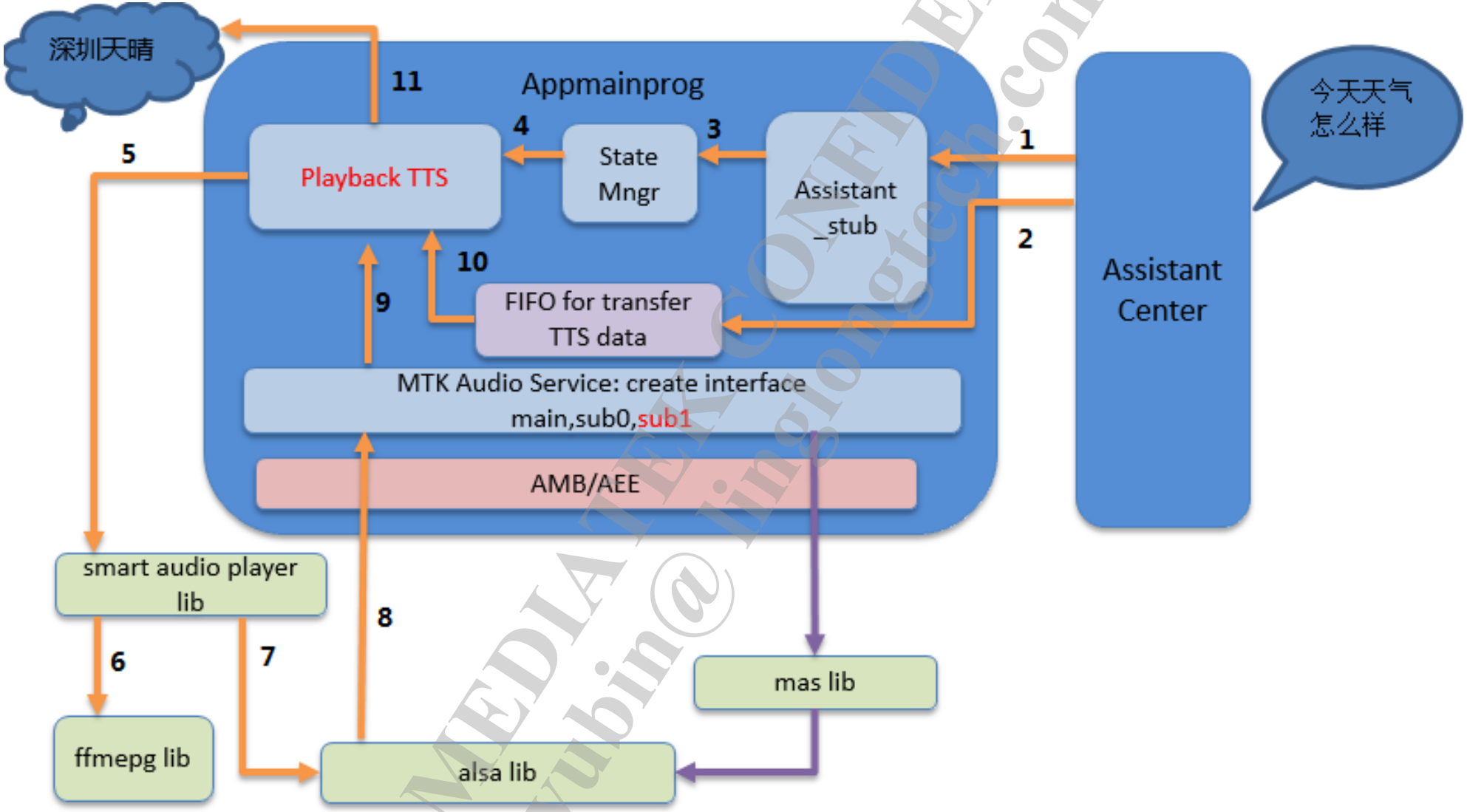
- **Assistant Stub**



- Step 1: Assistant center模块收到语音并转化为播放URL的请求，接着将请求按一定格式封装成JSON数据，并通过socket方式发送数据
- Step 2: Assistant_stub线程会通过socket方式接收数据并解析JSON，然后将播放请求发给State_Mngr模块
- Step 3: State_Mngr模块会判断当前是否有source在播放，如果有则暂停当前播放，接着发送request请求至playback_url app
- Step 4: playback url app收到播放请求后，会根据url去播放，并将起播状态同时通知给到state_mngr以及assistant_stub模块
- Step 5: 模块收到播放状态后，会按一定格式打包成JSON数据格式
- Step 6: 打包好后，通过socket发送至 assistant center

App Feature Introduce

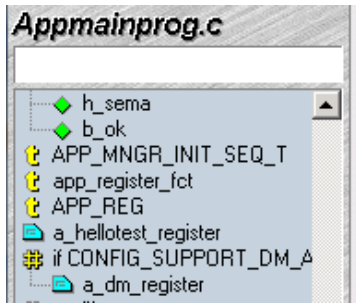
- **Playback -- TTS for example（Text to speak）**



code Introduce

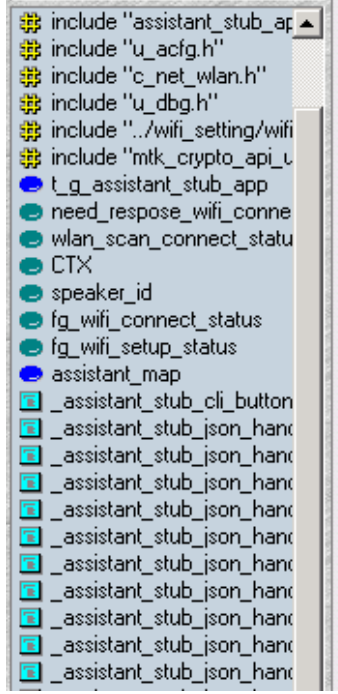
- **assistant stub—register**

Appmainprog.c




```
00090: static APP_REG app_to_reg[] = {
00091:         {a_timerd_register,{0},TIMERD_THREAD_NAME},
00092:         {a_acfg_register,{0},ACFG_THREAD_NAME},
00093:         {a_sm_register,{0},SM_THREAD_NAME},
00094:         {a_assistant_stub_register,{0},ASSISTANT_STUB_THREAD_NAME},
00095:         {a_wifi_setting_register,{0},WIFI_SETTING_THREAD_NAME},
00096:         {a_misc_register,{0},MISC_THREAD_NAME},
00097:         {a_upg_register,{0},UPG_THREAD_NAME},
00098:         {a_playback_uri_register,{0},PB_URI_THREAD_NAME},
```

Assistant_stub_app.c

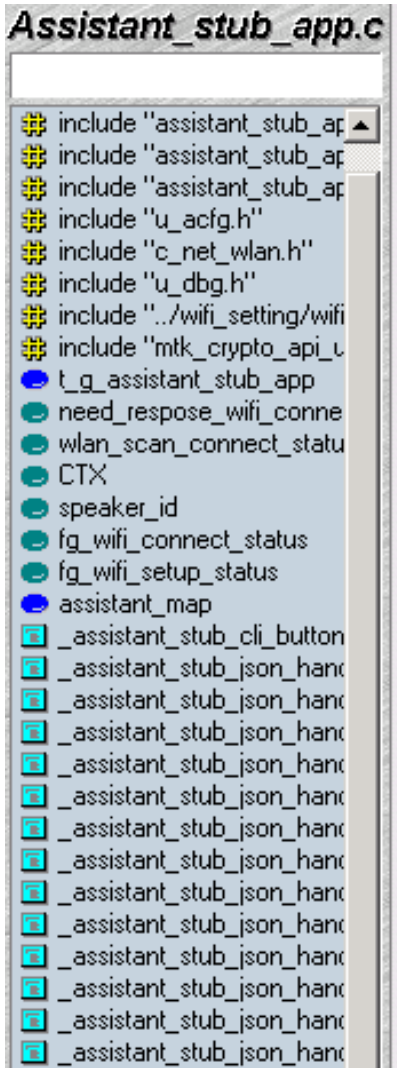



```
01394: VOID a_assistant_stub_register(AMB_REGISTER_INFO_T* pt_reg)
01395: {
01396:     FUNCTION_BEGIN
01397:
01398:     if (t_g_assistant_stub_app.b_app_init_ok)
01399:     {
01400:         printf("<ASSISTANT_STUB_APP> a_assistant_stub_register done,just return\n");
01401:         return;
01402:     }
01403:
01404:     strncpy(pt_reg->s_name, ASSISTANT_STUB_THREAD_NAME,sizeof(ASSISTANT_STUB_THREAD_NAME));
01405:     pt_reg->t_fct_tbl.pf_init                    = _assistant_stub_app_init;
01406:     pt_reg->t_fct_tbl.pf_exit                    = _assistant_stub_app_exit;
01407:     pt_reg->t_fct_tbl.pf_process_msg             = _assistant_stub_app_process_msg;
01408:     pt_reg->t_desc.ui8_flags = AEE_FLAG_WRITE_CONFIG|AEE_FLAG_WRITE_FLM|AEE_FLAG_WRITE_TSL|.
01409:     pt_reg->t_desc.t_thread_desc.z_stack_size = ASSISTANT_STUB_STACK_SZ;
01410:     pt_reg->t_desc.t_thread_desc.ui1_priority = ASSISTANT_STUB_THREAD_PRIORITY;
01411:     pt_reg->t_desc.t_thread_desc.ui2_num_msgs = ASSISTANT_STUB_THREAD_MSG_NUM;
01412:     pt_reg->t_desc.ui4_app_group_id = 0;
01413:     pt_reg->t_desc.ui4_app_id = 0;
01414:     pt_reg->t_desc.ui2_msg_count = ASSISTANT_STUB_MSGS_COUNT;
01415:     pt_reg->t_desc.ui2_max_msg_size = ASSISTANT_STUB_MAX_MSGS_SIZE;
```

code Introduce

- **assistant stub—app initial**

Assistant_stub_app.c




```
01199:
01200: static INT32 _assistant_stub_app_init(const CHAR* ps_name, HANDLE_T h_app)
01201: {
01202:     FUNCTION_BEGIN
01203:
01204:     INT32    i4_ret;
01205:     HANDLE_T h_thread = NULL_HANDLE;
01206:     UINT8    volume;
01207:
01208:     memset(&t_g_assistant_stub_app, 0, sizeof(ASSISTANT_STUB_APP_OBJ_T));
01209:
01210:     t_g_assistant_stub_app.h_app=h_app;
01211:
01212:     if (t_g_assistant_stub_app.b_app_init_ok)
01213:     {
01214:         return AEER_OK;
01215:     }
01216:
01217:     /*create assistance json data recv thread*/
01218:     i4_ret = u_thread_create(&h_thread,
01219:                              ASSISTANT_STUB_JSON_RECV_THREAD_NAME,
01220:                              ASSISTANT_STUB_JSON_RECV_THREAD_STACK_SIZE,
01221:                              ASSISTANT_STUB_JSON_RECV_THREAD_PRIORATY,
01222:                              _assistant_stub_json_recv_thread,
01223:                              0,
01224:                              NULL);
```

code Introduce

- **assistant stub—recv json thread**

Assistant_stub_app_json_recv.c

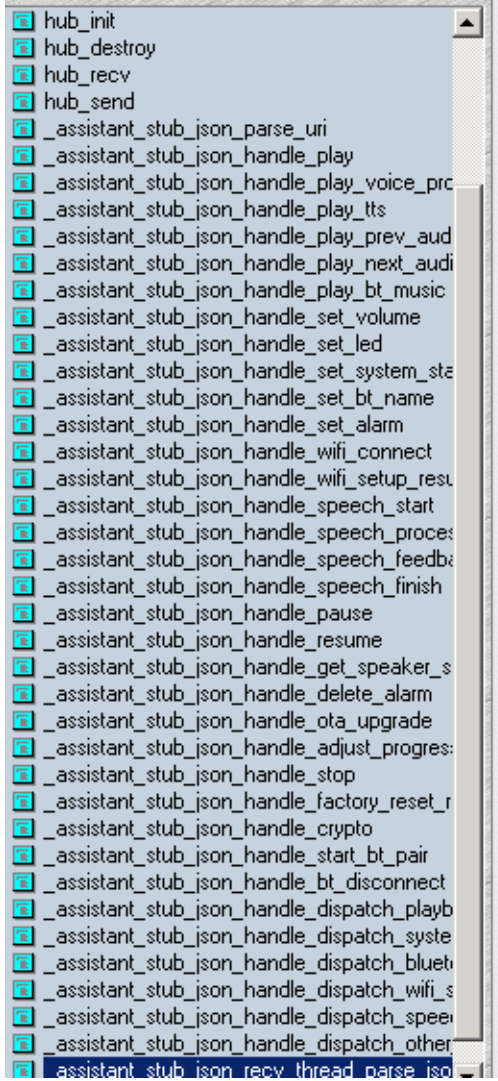
- hub_init
- hub_destroy
- hub_recv
- hub_send
- _assistant_stub_json_parse_uri
- _assistant_stub_json_handle_play
- _assistant_stub_json_handle_play_voice_prc
- _assistant_stub_json_handle_play_tts
- _assistant_stub_json_handle_play_prev_aud
- _assistant_stub_json_handle_play_next_audi
- _assistant_stub_json_handle_play_bt_music
- _assistant_stub_json_handle_set_volume
- _assistant_stub_json_handle_set_led
- _assistant_stub_json_handle_set_system_sta
- _assistant_stub_json_handle_set_bt_name
- _assistant_stub_json_handle_set_alarm
- _assistant_stub_json_handle_wifi_connect
- _assistant_stub_json_handle_wifi_setup_resu
- _assistant_stub_json_handle_speech_start
- _assistant_stub_json_handle_speech_proces
- _assistant_stub_json_handle_speech_feedba
- _assistant_stub_json_handle_speech_finish
- _assistant_stub_json_handle_pause
- _assistant_stub_json_handle_resume
- _assistant_stub_json_handle_get_speaker_s
- _assistant_stub_json_handle_delete_alarm
- _assistant_stub_json_handle_ota_upgrade
- _assistant_stub_json_handle_adjust_progres
- _assistant_stub_json_handle_stop
- _assistant_stub_json_handle_factory_reset_r
- _assistant_stub_json_handle_crypto
- _assistant_stub_json_handle_start_bt_pair
- _assistant_stub_json_handle_bt_disconnect
- _assistant_stub_json_handle_dispatch_playb
- _assistant_stub_json_handle_dispatch_syste
- _assistant_stub_json_handle_dispatch_bluet

```
01568: VOID _assistant_stub_json_recv_thread(VOID * arg)
01569: {
01570:     FUNCTION_BEGIN
01571:     char cmd[HUB_CMD_LENGTH_MAX+1] = {0};
01572:
01573:     hub_init(&CTX, HUB_APP);
01574:
01575:     while(1)
01576:     {
01577:         memset(cmd, 0, sizeof(cmd));
01578:         hub_recv(&CTX, cmd, sizeof(cmd));
01579:         printf("<ASSISTANT_STUB_APP> json recv:%s\n",cmd);
01580:
01581:         cJSON * assistant_cmd = cJSON_Parse(cmd);
01582:         if (NULL == assistant_cmd)
01583:         {
01584:             cJSON_Delete(assistant_cmd);
01585:             printf("<ASSISTANT_STUB_APP> error , assistant_cmd is NULL\n");
01586:             return ASSISTANT_STUB_APPR_FAIL;
01587:         }
01588:
01589:         _assistant_stub_json_recv_thread_parse_json(assistant_cmd);
01590:
01591:         cJSON_Delete(assistant_cmd);
01592:     }
01593:
01594:     hub_destroy(&CTX);
01595:     FUNCTION_END
01596:
01597:     pthread_exit(NULL);
01598: } ? end _assistant_stub_json_recv_thread ?
```

code Introduce

- **assistant stub—json parse**

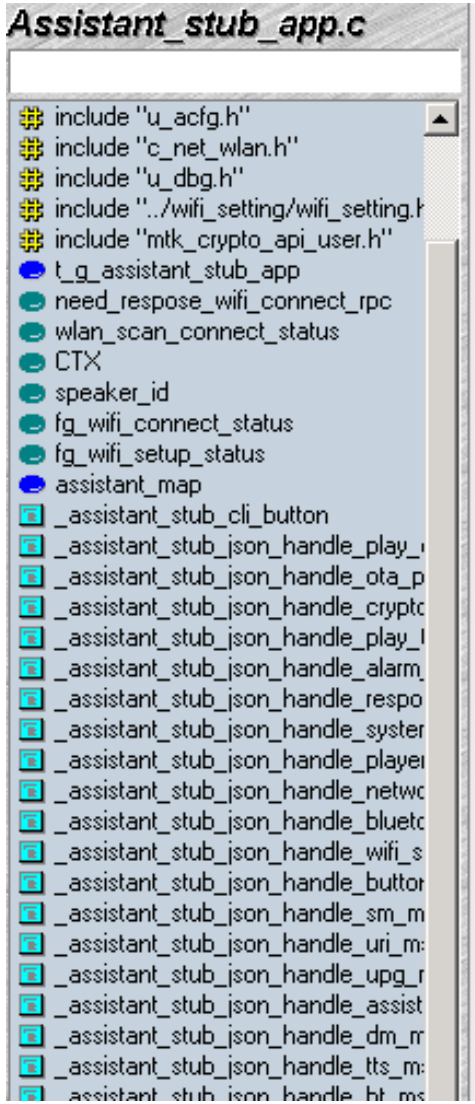
Assistant_stub_app_json_recv.c




```
01531: static VOID _assistant_stub_json_recv_thread_parse_json(cJSON * assistant_cr
01532: {
01533:     FUNCTION_BEGIN
01534:     char cmd[ASSISTANT_STUB_COMMAND_MAX_LENGTH+1] = {0};
01535:
01536:     strncpy(cmd,
01537:             cJSON_GetObjectItem(assistant_cmd,ASSISTANT_STUB_COMMAND)->valuestring,
01538:             ASSISTANT_STUB_COMMAND_MAX_LENGTH);
01539:
01540:     if(NULL != strstr(cmd,"/playback/"))
01541:     {
01542:         _assistant_stub_json_handle_dispatch_playback_cmd(assistant_cmd);
01543:     }
01544:     else if (NULL != strstr(cmd,"/system/"))
01545:     {
01546:         _assistant_stub_json_handle_dispatch_system_cmd(assistant_cmd);
01547:     }
01548:     else if (NULL != strstr(cmd,"/bluetooch/"))
01549:     {
01550:         _assistant_stub_json_handle_dispatch_bluetooth_cmd(assistant_cmd);
01551:     }
01552:     else if (NULL != strstr(cmd,"/wifi_setting/"))
01553:     {
01554:         _assistant_stub_json_handle_dispatch_wifi_setting_cmd(assistant_cmd);
01555:     }
01556:     else if (NULL != strstr(cmd,"/speech/"))
01557:     {
01558:         _assistant_stub_json_handle_dispatch_speech_cmd(assistant_cmd);
01559:     }
01560:     else
01561:     {
01562:         _assistant_stub_json_handle_dispatch_other_cmd(assistant_cmd);
01563:     }
01564:
```

code Introduce

- **assistant stub—assistant stub process message**

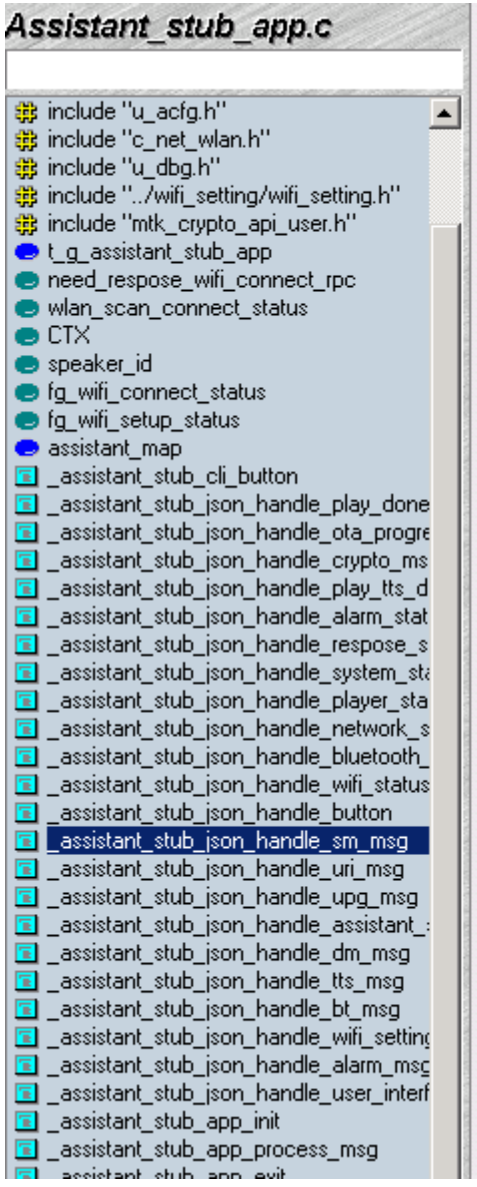



```
01248: static INT32 _assistant_stub_app_process_msg(HANDLE_T      h_app,
01249:                                              UINT32        ui4_type,
01250:                                              const VOID*   pv_msg,
01251:                                              SIZE_T        z_msg_len,
01252:                                              BOOL          b_paused)
01253: {
01254:     FUNCTION_BEGIN
01255:
01256:     UCHAR*        puc_name;
01257:     INT32         i4_ret = ASSISTANT_STUB_APPR_OK;
01258:     APPMSG_T *    app_msg = (APPMSG_T * )pv_msg;
01259:
01260:     if (!t_g_assistant_stub_app.b_app_init_ok)
01261:     {
01262:         return AEER_FAIL;
01263:     }
01264:
01265:     if (ui4_type < AMB_BROADCAST_OFFSET)
01266:     {
01267:         /* private message */
01268:         switch(ui4_type)
01269:         {
01270:             case E_APP_MSG_TYPE_ASSISTANT_STUB:
01271:             {
01272:                 printf("<ASSISTANT_STUB_APP> E_APP_MSG_TYPE_ASSISTANT_STUB\n");
01273:                 switch(app_msg->ui4_sender_id)
01274:                 {
01275:                     case MSG_FROM_SM:
```

code Introduce

- **assistant stub—assistant stub process sm message**

Assistant_stub_app.c




```
00966: static VOID _assistant_stub_json_handle_sm_msg(APPMSG_T * t_app_msg)
00967: {
00968:     FUNCTION_BEGIN
00969:     switch(t_app_msg->ui4_msg_type)
00970:     {
00971:         case ASSISTANT_STUB_CMD_SYSTEM_STATUS_CHANGE:
00972:         {
00973:             _assistant_stub_json_handle_system_status_change((VOID *)(t_app_msg->p_us
00974:         }
00975:         break;
00976:
00977:         case ASSISTANT_STUB_CMD_RESPOSE_SPEAKER_STATUS:
00978:         {
00979:             _assistant_stub_json_handle_respose_speaker_status((VOID *)(t_app_msg->p_
00980:         }
00981:         break;
00982:
00983:         case  ASSISTANT_STUB_CMD_NETWORK_STATUS_CHANGE:
00984:         {
00985:             _assistant_stub_json_handle_network_status_change((VOID *)(t_app_msg->p_us
00986:         }
00987:         break;
00988:
00989:         default:
00990:         {
00991:             printf("<ASSISTANT_STUB_APP> ERROR SM MSG TYPE!\n");
00992:         }
00993:         break;
00994:
00995:     }
00996:     FUNCTION_END
00997: } ? end _assistant_stub_json_handle_sm_msg ?
00998:
```

code Introduce

- **assistant stub—assistant stub package json data response to assistant center program**

Assistant_stub_app.c


```
00606: static VOID _assistant_stub_json_handle_system_status_change(VOID * p_usr_msg
00607: {
00608:     FUNCTION_BEGIN
00609:     char * out = NULL;
00610:     char send_buffer[HUB_CMD_LENGTH_MAX+1] = {0};
00611:     ASSISTANT_STUB_SYSTEM_STATUS_CHANGE_T system_status_change = {0};
00612:     memcpy(&system_status_change,
00613:           (ASSISTANT_STUB_SYSTEM_STATUS_CHANGE_T *)p_usr_msg,
00614:            sizeof(ASSISTANT_STUB_SYSTEM_STATUS_CHANGE_T));
00615:
00616:     printf("<ASSISTANT_STUB_APP> command is %s \n",system_status_change.command);
00617:     printf("<ASSISTANT_STUB_APP> status is %s \n",system_status_change.status);
00618:
00619:     /*create cjson object*/
00620:     cJSON * params = cJSON_CreateObject();
00621:     cJSON * notificatioon = cJSON_CreateObject();
00622:     /*add  jsonrpc && command*/
00623:     cJSON_AddItemToObject(notificatioon,
00624:                           ASSISTANT_STUB_COMMAND,
00625:                           cJSON_CreateString(system_status_change.command));
00626:     /*add params && status*/
00627:     cJSON_AddItemToObject(notificatioon,ASSISTANT_STUB_PARAMS,params);
00628:     cJSON_AddItemToObject(params,
00629:                           ASSISTANT_STUB_STATUS,
00630:                           cJSON_CreateString(system_status_change.status));
00631:     /*transfer cjson to string and send to assistance server*/
00632:     out = cJSON_Print(notificatioon);
00633:     strncpy(send_buffer,out,HUB_CMD_LENGTH_MAX);
00634:     printf("<ASSISTANT_STUB_APP> send msg is %s\n",send_buffer);
00635:     hub_send(&CTX,send_buffer,strlen(send_buffer));
00636:     /*delete cjson object*/
00637:     cJSON_Delete(notificatioon);
00638:     free(out);
00639:     out = NULL;
```

Q&A

Thank you!

everyday genius