

Факультатив ФКН

# РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ НА БАЗЕ ПЛАТФОРМЫ



Преподаватель

Папулин Сергей Юрьевич ([papulin\\_hse@mail.ru](mailto:papulin_hse@mail.ru))

Ассистент

Цырлин Никита ([nstsyrlin@gmail.com](mailto:nstsyrlin@gmail.com))



# Нативный сервис

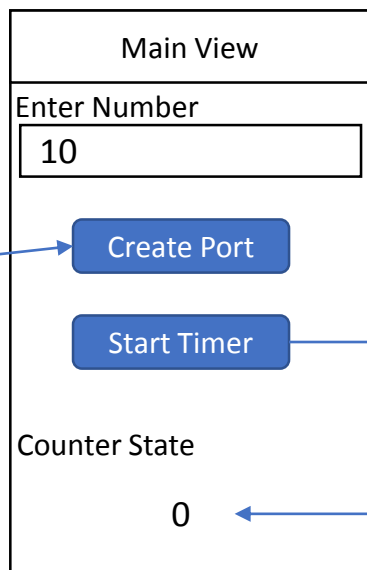


# Описание работы приложений

## ClientServiceMAppNative

`appid="org.example.clientservicemappnative"`

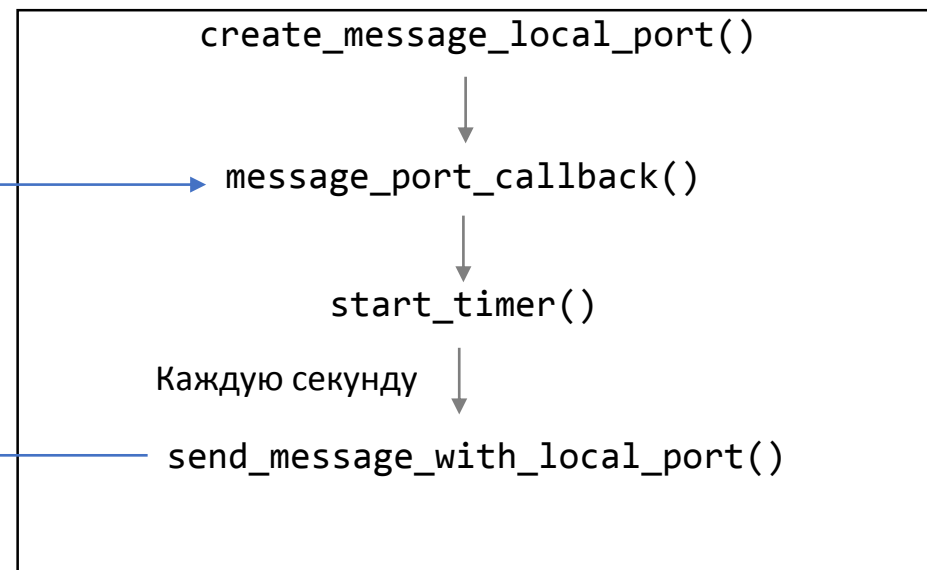
Создание  
локального порта



*Нативное  
приложение*

## FirstServiceMNative

`appid="org.example.firstservicemnative"`

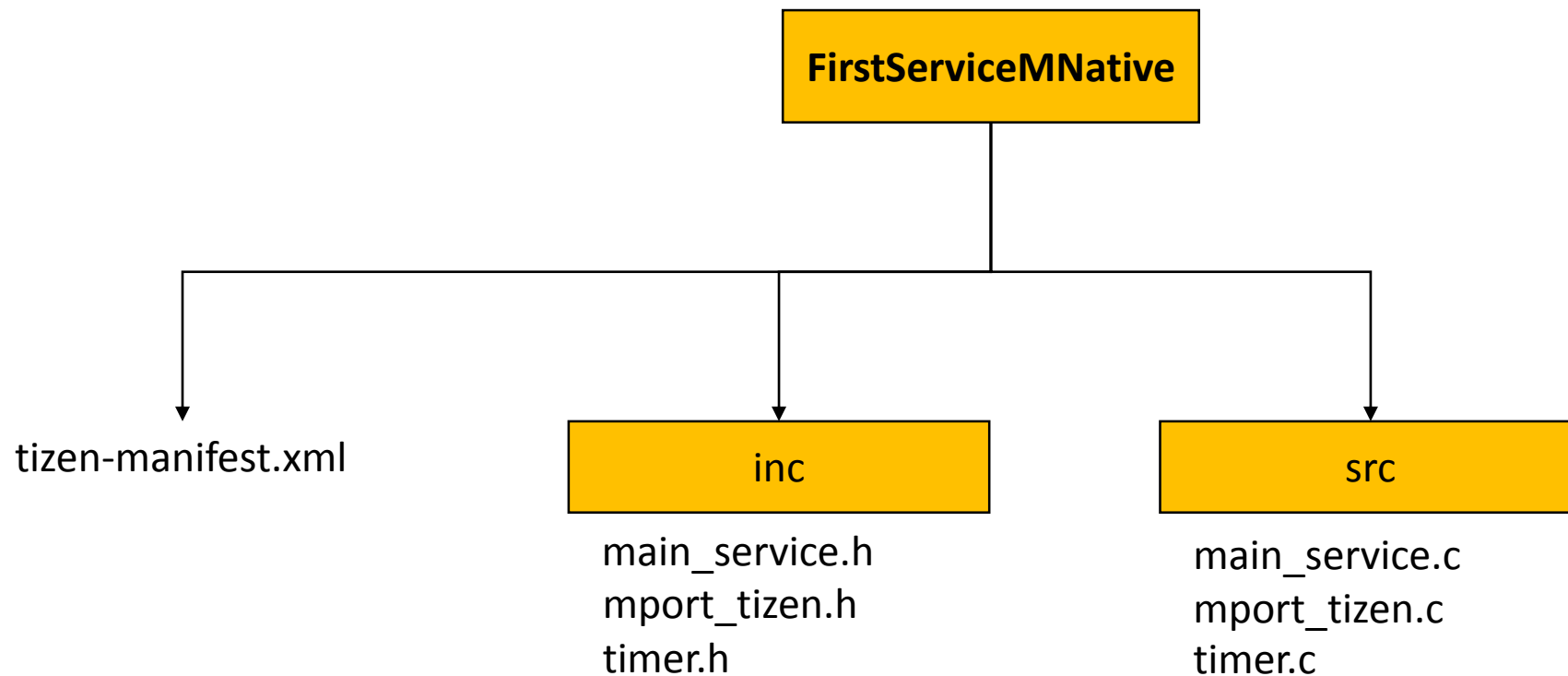


*Нативный  
сервис*

Коммуникация: **MessagePort API**



# Структура приложения FirstServiceMNative





# FirstServiceMNative – Заголовки

---

## mport\_tizen.h

```
void create_message_local_port();

void message_port_callback(int local_port_id,
                           const char *remote_app_id,
                           const char *remote_port,
                           bool trusted_remote_port,
                           bundle *message,
                           void *user_data);

void send_message_with_local_port(int local_port_id,
                                  char *remote_app_id,
                                  char *remote_port_name,
                                  int counter,
                                  int isFinished);

void start_timer(int counter, int local_port_id, char *remote_app_id, char *remote_port_name);
```

## timer.h

```
void one_second_timer();
```



# FirstServiceMNative – Создание локального порта MessagePort

mport\_tizen.c

```
void create_message_local_port()
```

```
    char *local_port_name = "ServiceLocalPort";
```

```
    int local_port_id = message_port_register_local_port(local_port_name, message_port_callback, NULL);
```

```
void message_port_callback(int local_port_id, const char *remote_app_id, const char *remote_port,  
                           bool trusted_remote_port, bundle *message, void *user_data)
```

```
    char *command = NULL;
```

```
    char *str_counter = NULL;
```

```
    bundle_get_str(message, "command", &command);
```

```
    bundle_get_str(message, "maxcounter", &str_counter);
```



# FirstServiceMNative – Создание локального порта MessagePort

mport\_tizen.c

```
void message_port_callback(int local_port_id, const char *remote_app_id, const char *remote_port,
                           bool trusted_remote_port, bundle *message, void *user_data)

char *command = NULL;
char *str_counter = NULL;

bundle_get_str(message, "command", &command);
bundle_get_str(message, "maxcounter", &str_counter);

if (strcmp(command, "start") == 0) {
    bundle *reply = bundle_create();

    bundle_add_str(reply, "status", "starting");
    bundle_add_str(reply, "counter", "0");

    ret = message_port_send_message(remote_app_id, remote_port, reply);

    bundle_free(reply);

    int counter;
    sscanf(str_counter, "%i", &counter);

    start_timer(counter, local_port_id, remote_app_id, remote_port);
}
```



# FirstServiceMNative – Отправка сообщений

mport\_tizen.c

```
void start_timer(int counter, int local_port_id, char *remote_app_id, char *remote_port_name)
{
    int i = 0;

    for(i; i < counter; i++)
    {
        one_second_timer();
        send_message_with_local_port(local_port_id, remote_app_id, remote_port_name, i+1, 0);
    }

    send_message_with_local_port(local_port_id, remote_app_id, remote_port_name, i+1, 1);
}
```





# FirstServiceMNative – Отправка сообщений

mport\_tizen.c

```
void send_message_with_local_port(int local_port_id, char *remote_app_id, char *remote_port_name,
                                   int counter, int isFinished)

    bundle *b = bundle_create();

    if (isFinished == 1) {
        bundle_add_str(b, "status", "finished");
        bundle_add_str(b, "counter", "-1");
    }
    else
    {
        char str_counter[64];

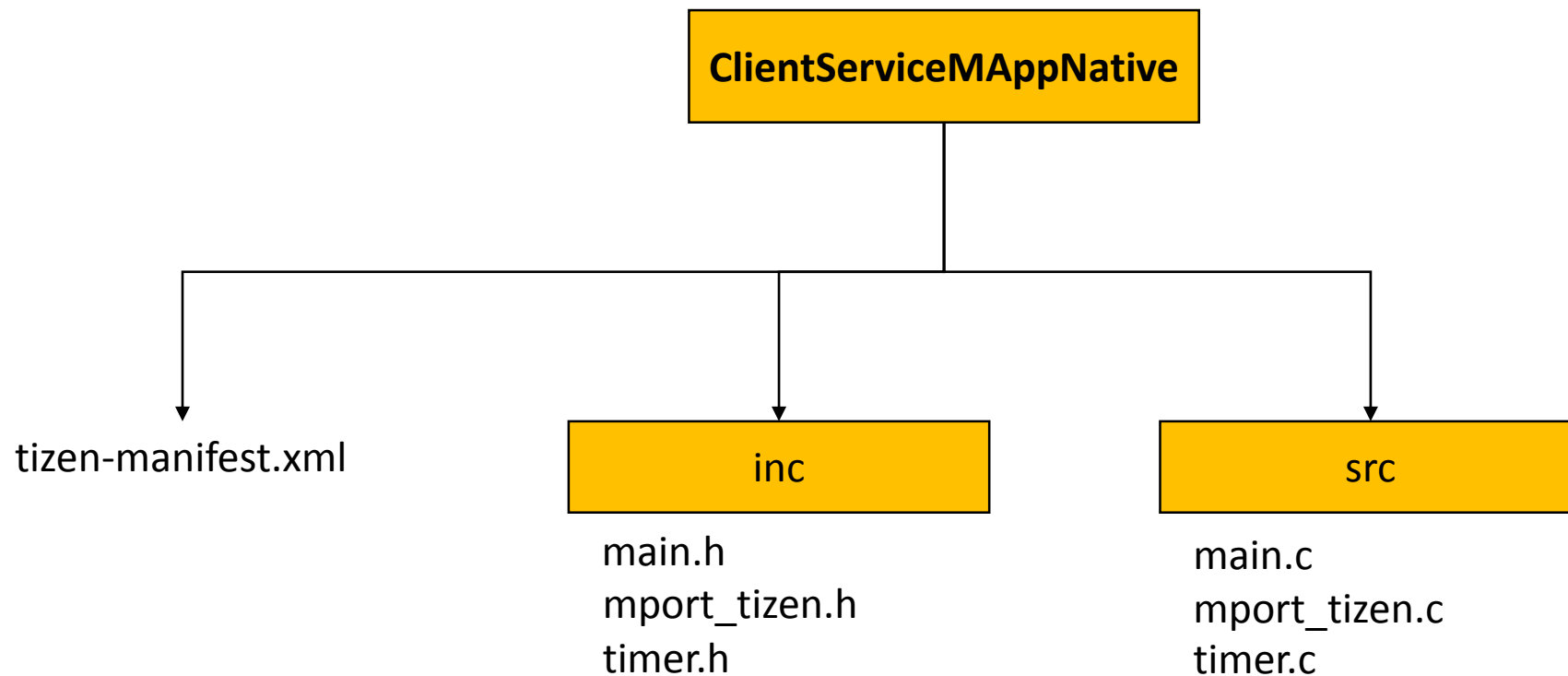
        sprintf(str_counter, "%d", counter);

        bundle_add_str(b, "status", "active");
        bundle_add_str(b, "counter", str_counter);
    }

    ret = message_port_send_message_with_local_port(remote_app_id, remote_port_name, b, local_port_id);
```



# Структура приложения ClientServiceMAppNative





# ClientServiceMAppNative – Заголовки

---

main.h

```
typedef struct MessagePort
{
    int local_port_id;
    char *local_port_name;
    char *remote_port_name;
    char *remote_service;

} messageport_s;

typedef struct appdata {

    Evas_Object *win;
    Evas_Object *conform;
    Evas_Object *nf;
    Evas_Object *txt;
    Evas_Object *lbl;
    messageport_s *mp;

} appdata_s;
```



# ClientServiceMAppNative – Заголовки

---

## main\_view.h

```
void create_main_view(appdata_s *ad);

void start_timer_btn_clicked_cb(void *data, Evas_Object *obj, void *event_info);
void create_port_btn_clicked_cb(void *data, Evas_Object *obj, void *event_info);
```

## mport\_tizen.h

```
void send_message_with_local_port(messageport_s *mp, char *maxcounter);
void create_message_local_port(messageport_s *mp, Evas_Object *lbl);
void message_port_callback(int local_port_id, const char *remote_app_id, const char *remote_port,
bool trusted_remote_port, bundle *message, void *user_data);
```



# ClientServiceMAppNative – Создание локального порта MessagePort

mport\_tizen.c

```
void create_message_local_port(messageport_s *mp, Evas_Object *lbl)
```

```
    char *local_port_name = "AppLocalPort";
```

```
    int local_port_id = message_port_register_local_port(local_port_name, message_port_callback, lbl);
```

```
    mp->local_port_id = local_port_id;
```

```
    mp->remote_service = REMOTE_SERVICE_ID;
```

```
    mp->local_port_name = local_port_name;
```

```
    mp->remote_port_name = "ServiceLocalPort";
```

Инициализация удаленного порта



```
void message_port_callback(int local_port_id, const char *remote_app_id, const char *remote_port,  
                           bool trusted_remote_port, bundle *message, void *user_data)
```

```
    char *counter = NULL;
```

```
    char *status = NULL;
```

```
    bundle_get_str(message, "status", &status);
```

```
    bundle_get_str(message, "counter", &counter);
```



```
void send_message_with_local_port(messageport_s *mp, char *maxcounter)
```

```
bundle_add_str(b, "command", "start");
```

[illegible]



# Ссылки

---

\* ETL API:

<https://developer.tizen.org/development/guides/native-application/user-interface/efl>

<https://developer.tizen.org/development/ui-practices/native-application/efl/ui-components>

<https://developer.tizen.org/development/ui-practices/native-application/efl/scaling/scalability-support>

<https://developer.tizen.org/development/ui-practices/native-application/efl/ui-containers/box>