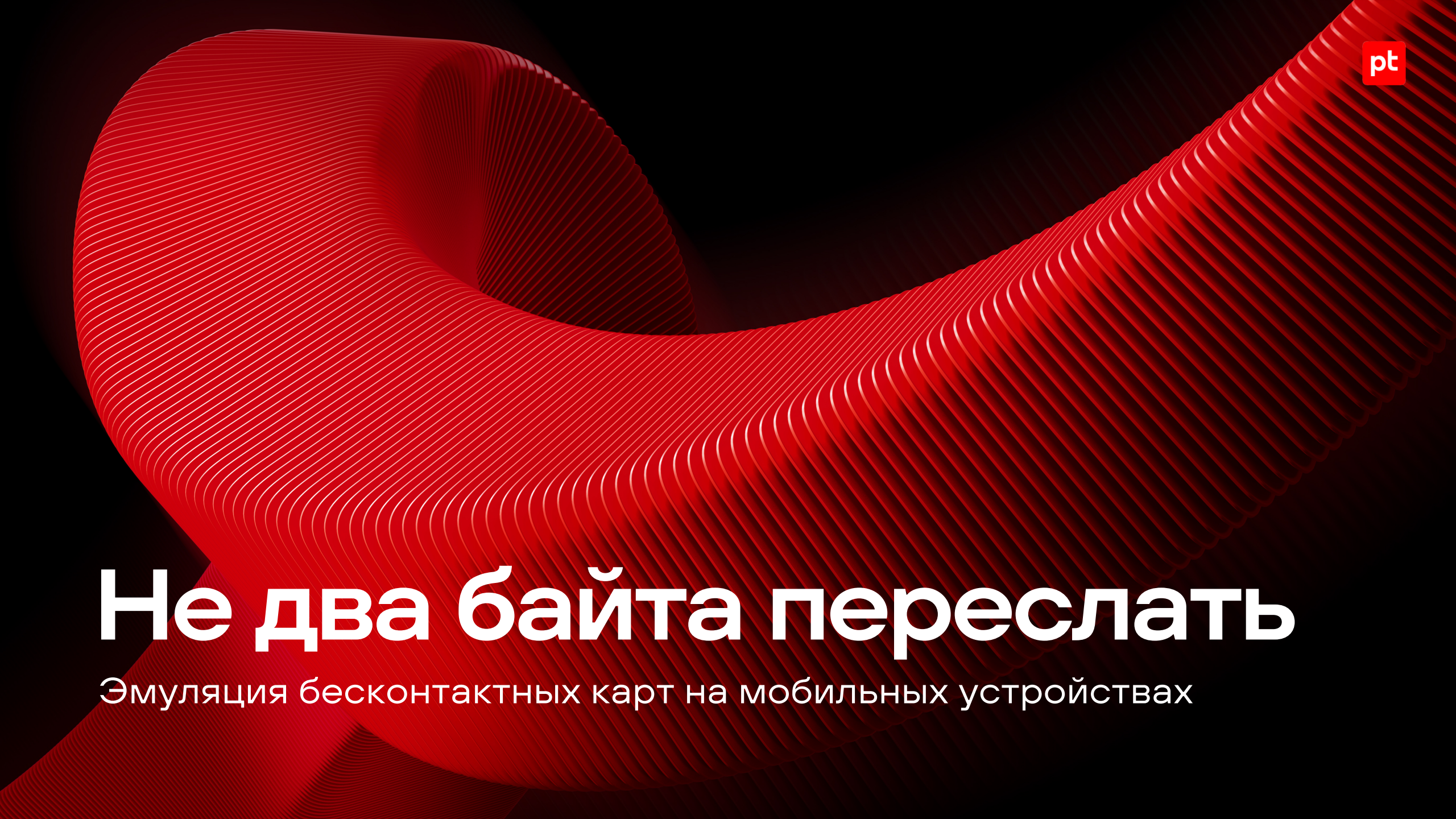


Не два байта переслать

Эмуляция бесконтактных карт на мобильных устройствах

О себе

- Специалист по информационной безопасности мобильных приложений в Positive Technologies
- Работаю с Android с 2016 года
- Работаю с необычными возможностями мобильных ОС, погружаюсь в детали

О себе и NFC:

- Участвовал в разработке ПО для транспортной системы использующей NFC карты и смартфоны, в том числе клиентских и служебных приложений
- Делал интеграции с разными картами разных типов от разных вендоров, поддержкой их кастомных протоколов, секурити итд
- Проектировал распределение ключей для карт разных типов и SAM модулей, персонализацию и прошивание, интеграцию, безопасную аутентификацию SAM модулей, и т.д.
- Писал интеграции с платёжными терминалами на базе андроида, логику взаимодействия через разные ридеры, PC/SC драйвер, нормальное апи андроида, ненормальное апи андроида и т.д.
- Написал для себя небольшой инструментарий по взаимодействию, сборке APDU трафика, итд

Почему:

- Интересный интерфейс, физический, P2P, может быть полезен при передаче некоторых видов сообщений, аутентификации, итд
- Редко используется, недостаточно популярен, мало информации
- Хорошо описано чтение тэгов, но не их эмуляция

Содержание:

Теория

общая информация про NFC
и необходимая матчасть

Практика

пример передачи реального
сообщения между смартфонами

Общее

Общее – NFC

NFC

группа стандартов определяющая общение в БП

- Связь на расстоянии 10-20 см
- Карты оплаты, транспортные карты, карты лояльности, СКУД, бирки, другое (светодиоды, термометры, биоимпланты)

Общее – NFC

Физические типы:

LF

125 KHz, 134 KHz, ...

HF

13.56 MHz

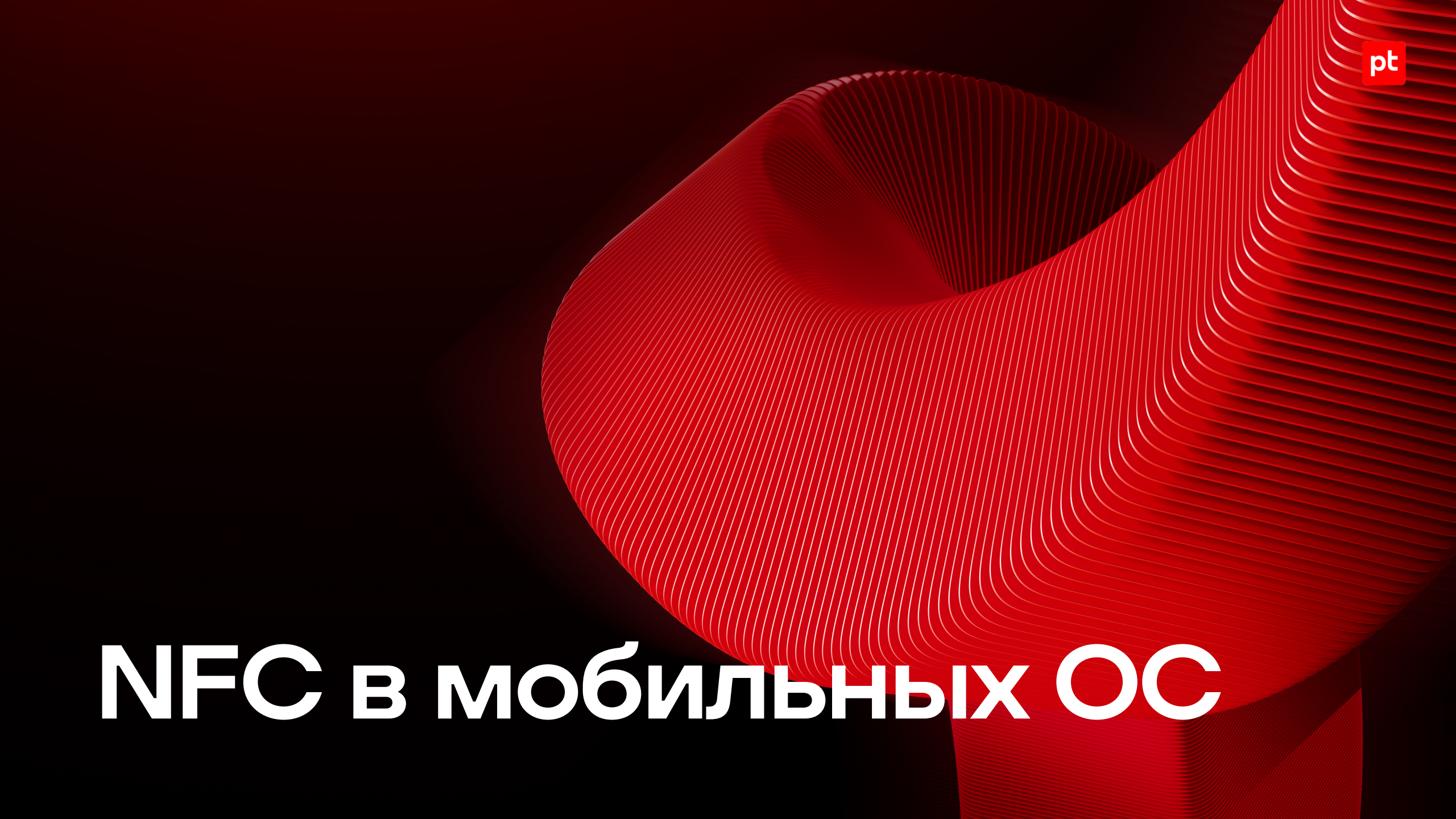
В смартфонах поддерживаются **только HF**

Общее – NFC

Режимы работы:

- Ридер
- Тэг
- P2P

Для мобильных ОС важны:
ридер/тэг и порядок коммуникации


pt
NFC в мобильных ОС

NFC в мобильных ОС – Обзор

Можем читать

Можем эмулировать

Удобно и эргономично

NFC в мобильных ОС – Обзор – iOS

Из системы работает:

- Чтение тэгов в фоне
- Эмуляция через Apple Pay
- Эмуляция через Apple Wallet

Из приложений работает:

- Чтение тэгов по запросу
- Чтение тегов в режиме ридера

NFC в мобильных ОС – Обзор – iOS

TL/DR

можем только читать,
но НЕ эмулировать

NFC в мобильных ОС – Обзор – Android

Из системы работает:

- Чтение тэгов в фоне
- Эмуляция через Google Pay
- Эмуляция через Google wallet

Из приложений работает:

- Чтение тэгов по запросу
- Чтение тэгов в фоне с передачей управления в приложение
- Чтение тэгов в режиме ридера
- Эмуляция в режиме тэга

NFC в мобильных ОС – Обзор – Android

TL/DR

можем читать
и эмулировать

Эмуляция NFC в Android

Эмуляция NFC в Android – API

```
Pavel Vasilev
internal class NfcType4TagNdefEmulationService : HostApduService() {

    private var ndefEmulation: NdefEmulation? = null
    private var ndefEmulator: ApduExecutor? = null

    Pavel Vasilev
    override fun onDeactivated(reason: Int) {
        ndefEmulation = null
        ndefEmulator = null
    }

    Pavel Vasilev
    override fun processCommandApdu(commandApdu: ByteArray?, extras: Bundle?): ByteArray? {
        try {
            if (commandApdu == null) {
                return ApduConstants.SW_ERROR_INPUT_DATA_ABSENT
            }
```

Эмуляция NFC в Android – API

```xml
<?xml version="1.0" encoding="utf-8"?>
<host-apdu-service xmlns:android="http://schemas.android.com/apk/res/android"
    android:description="@string/nfc_type4_tag_emulation_service_name"
    android:requireDeviceUnlock="false">
    <aid-group android:description="@string/nfc_type4_tag_emulation_description" android:category="other">
        <aid-filter android:name="D2760000850101"/>
    </aid-group>
</host-apdu-service>
```

Эмуляция NFC в Android – API

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android">
    <uses-feature android:name="android.hardware.nfc.hce" android:required="false"/>
    <uses-permission android:name="android.permission.NFC"/>
    <application>
        <service android:name=".hce.NfcType4TagNdefEmulationService"
            android:permission="android.permission.BIND_NFC_SERVICE"
            android:exported="true">
            <intent-filter>
                <action android:name="android.nfc.cardemulation.action.HOST_APDU_SERVICE"/>
                <category android:name="android.intent.category.DEFAULT"/>
            </intent-filter>
            <meta-data
                android:name="android.nfc.cardemulation.host_apdu_service"
                android:resource="@xml/nfc_type4_tag_emulation"/>
        </service>
    </application>
</manifest>
```

Теория — стандарты

Теория – Стандарты

ISO
14443

ISO
7816

Теория – Стандарты – ISO-14443

Низкоуровневый протокол:

- Физический уровень
- Протокол инициализации
- Транспортный протокол обмена данными

Теория – Стандарты – APDU

Application program data unit

TL/DR – APDU == пакет

Теория – Стандарты – ISO-7816

Прикладной протокол:

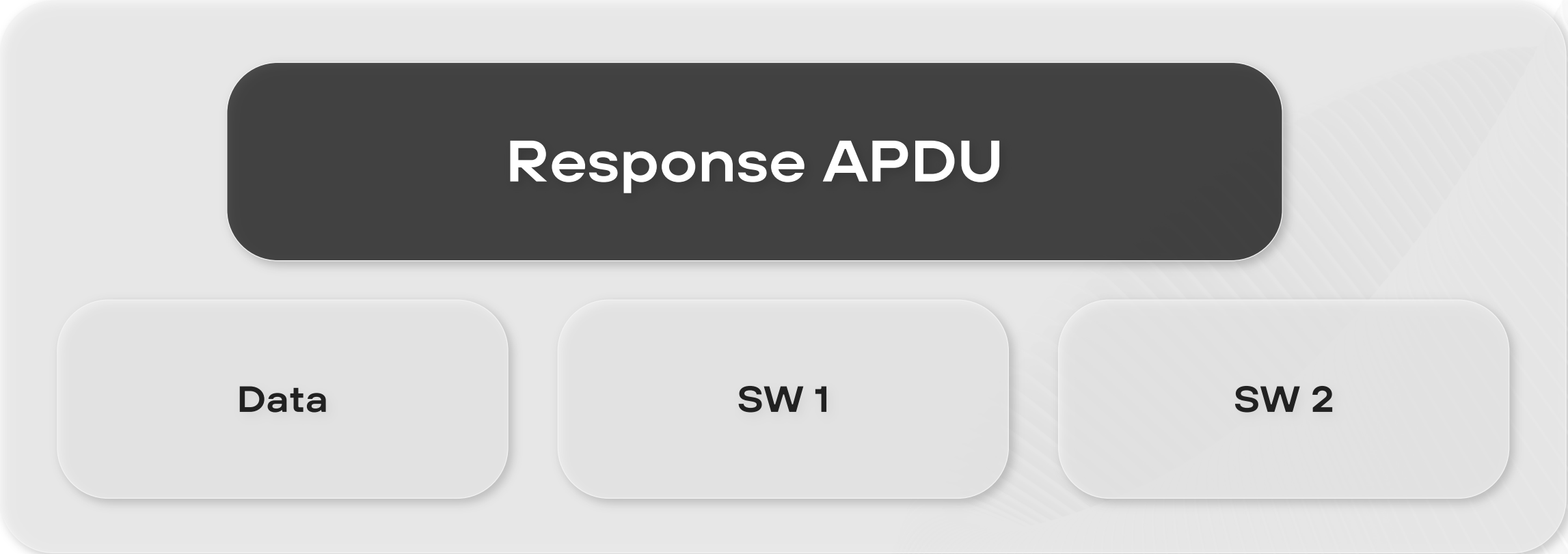
- APDU, SW_CODES
- Команды: SELECT, READ, WRITE итд
- Апплеты и их структура (AF/DF/EF, AID/FID)
- Примерный процесс

Теория – Стандарты – ISO-7816 – APDU

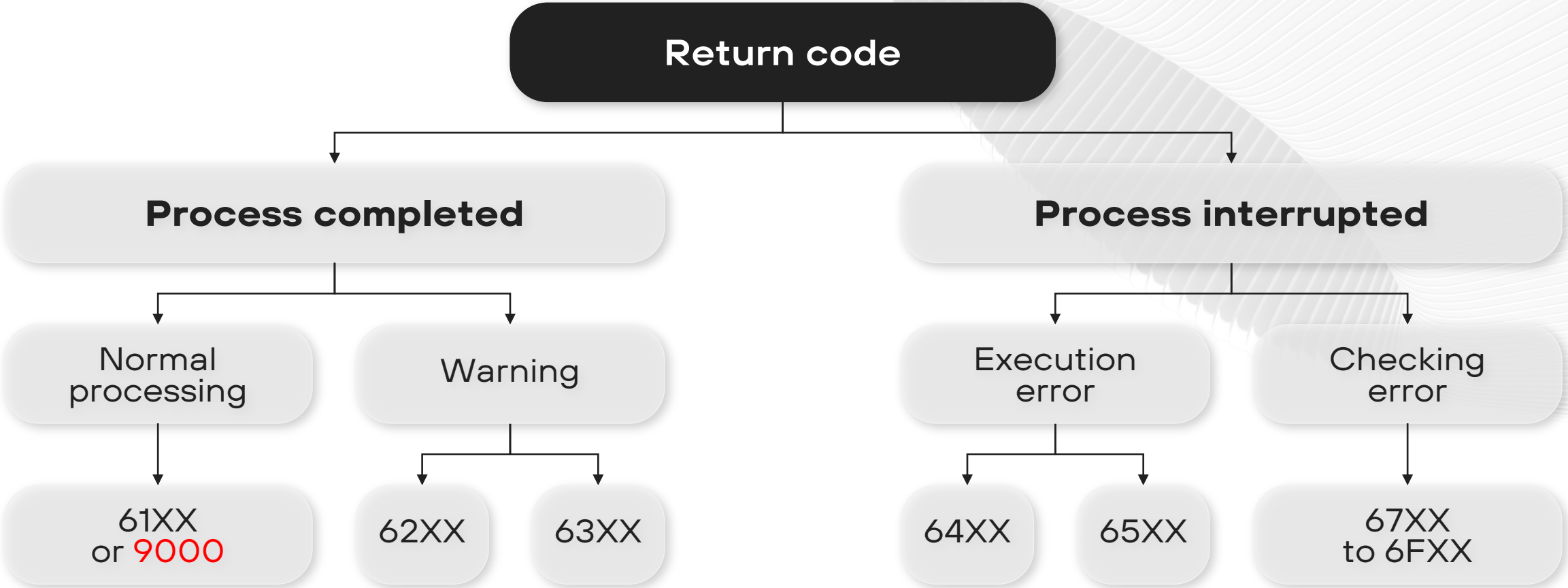
- Формат команды и запроса **(C-APDU, R-APDU)**
- Вся коммуникация происходит только полными циклами **1 запрос – 1 ответ**
- Ридер передаёт команду в формате **command-apdu**
- Ридер всегда **только инициирует коммуникацию**
- Тег отвечает респонс в формате **response-apdu**
- Тег всегда **только отвечает**

Теория – Стандарты – ISO-7816 – APDU

Теория – Стандарты – ISO-7816 – APDU



Теория – Стандарты – ISO-7816 – APDU



Теория – Стандарты – ISO-7816 – APDU

C-APDU : 00 A4 04 00 0E FF 4C 56 39 32 41 4C 47 2D 4B 45 2D 30 31
R-APDU : 90 00
C-APDU : 00 EE 00 00 B6 51 28 52 47 45 65 61 5F 54 74 4E 77 49 71 60 29 21 4E 47 41 76 69 6B 24 48 28 2E 59 3F 28 79 3F 71 54 6C 62 62 61 2A 41 78 24 63 67 79 60 21 61 76 70 45 6B 43 41 6A 2A 4F 29 42 3F 62 67 56 69 6D 24 6D 71 74 59 52 78 77 44 4A 51 75 64 73 42 55 2C 46 60 4E 67 77 79 24 64 3F 43 52 2C 6F 4E 6B 6F 42 62 62 7A 44 60 4C 2C 58 78 4D 6D 50 63 63 57 4B 2A 72 4C 6B 71 27 43 57 5A 6A 7A 43 59 44 65 41 64 2E 5F 55 57 53 7A 45 5F 3B 75 57 60 6D 43 61 76 58 66 6A 45 2E 44 50 49 2E 45 61 62 54 3B 3A 2A 68 21 69 5A 62 6F 4E 71 55 66 50 2A 76 6E 6B 62 24 55 00
R-APDU : 44 62 67 48 65 61 53 59 6B 73 5F 61 3A 65 68 42 67 76 4D 59 65 65 6B 4F 4D 62 4F 24 66 60 7A 62 79 6C 44 29 45 67 77 4B 54 43 29 6E 6C 21 66 21 63 6A 7A 58 4D 7A 2A 2A 66 4E 6B 2A 4F 52 2C 21 6E 49 2C 46 53 63 27 29 2E 66 72 6C 78 63 21 60 61 4D 63 50 45 71 6E 3F 51 47 4E 4B 63 5F 24 2C 51 48 6A 6F 60 57 6F 52 6B 6E 7A 50 6F 5A 55 63 42 75 6D 68 70 54 60 6B 77 43 70 71 28 53 45 6B 48 44 46 43 65 53 42 7A 3F 52 6F 59 29 63 4E 4A 50 66 3B 72 4D 71 21 44 71 4A 78 6A 5A 60 5F 7A 43 75 4C 2A 76 52 71 6A 54 69 7A 77 77 67 45 6A 57 6E 47 4A 4F 5A 90 00

Теория – Стандарты – ISO-7816 – APDU

C-APDU : 00 A4 04 00 0E FF 4C 56 39 32 41 4C 47 2D 4B 45 2D 30 31
R-APDU : 90 00
C-APDU : 00 EE 00 00 B6 51 28 52 47 45 65 61 5F 54 74 4E 77 49 71 60 29 21 4E 47 41 76 69 6B 24 48 28 2E 59 3F 28 79 3F 71 54 6C 62 62 61 2A 41 78 24 63 67 79 60 21 61 76 70 45 6B 43 41 6A 2A 4F 29 42 3F 62 67 56 69 6D 24 6D 71 74 59 52 78 77 44 4A 51 75 64 73 42 55 2C 46 60 4E 67 77 79 24 64 3F 43 52 2C 6F 4E 6B 6F 42 62 62 7A 44 60 4C 2C 58 78 4D 6D 50 63 63 57 4B 2A 72 4C 6B 71 27 43 57 5A 6A 7A 43 59 44 65 41 64 2E 5F 55 57 53 7A 45 5F 3B 75 57 60 6D 43 61 76 58 66 6A 45 2E 44 50 49 2E 45 61 62 54 3B 3A 2A 68 21 69 5A 62 6F 4E 71 55 66 50 2A 76 6E 6B 62 24 55 00
R-APDU : 44 62 67 48 65 61 53 59 6B 73 5F 61 3A 65 68 42 67 76 4D 59 65 65 6B 4F 4D 62 4F 24 66 60 7A 62 79 6C 44 29 45 67 77 4B 54 43 29 6E 6C 21 66 21 63 6A 7A 58 4D 7A 2A 2A 66 4E 6B 2A 4F 52 2C 21 6E 49 2C 46 53 63 27 29 2E 66 72 6C 78 63 21 60 61 4D 63 50 45 71 6E 3F 51 47 4E 4B 63 5F 24 2C 51 48 6A 6F 60 57 6F 52 6B 6E 7A 50 6F 5A 55 63 42 75 6D 68 70 54 60 6B 77 43 70 71 28 53 45 6B 48 44 46 43 65 53 42 7A 3F 52 6F 59 29 63 4E 4A 50 66 3B 72 4D 71 21 44 71 4A 78 6A 5A 60 5F 7A 43 75 4C 2A 76 52 71 6A 54 69 7A 77 77 67 45 6A 57 6E 47 4A 4F 5A 90 00

C-APDU: CLA – INS – P1 – P2 – LC – DATA – LE

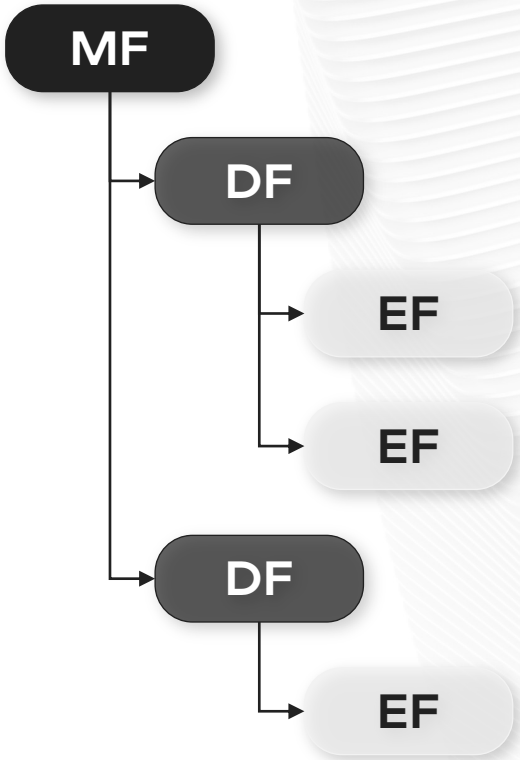
R-APDU: DATA-SW

Теория – Стандарты – ISO-7816-4

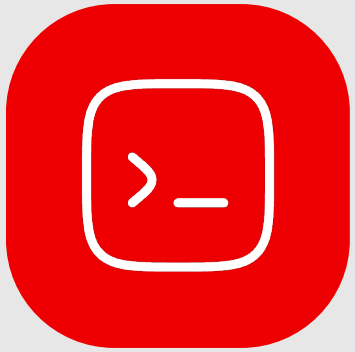
- Организация данных в иерархические структуры
- Всё – файл
- Master file (MF) / dedicated file (DF) / elementray file (EF)
- Основой всего является ADF
- Идентификаторы: AID (<16) / FID (2)

Теория – Стандарты – ISO-7816 – Апплеты

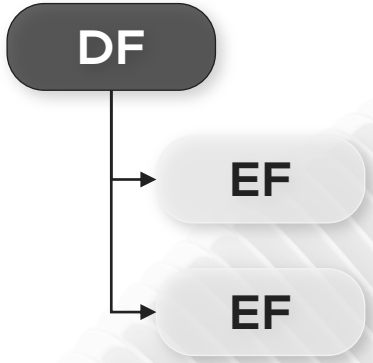
Стандарт



Теория – Стандарты – ISO-7816 – Апплеты



Практика



Теория – Стандарты – ISO-7816 – SELECT

- Файлы – **выбираются**
- Одновременно может быть выбран **только один файл одного уровня**
- С выбранными файлами производятся **дальнейшие действия**
  (аутентификация, чтение, запись, транзакции, итд)

Теория – Стандарты – ISO-7816 - Процесс

- > SELECT ADF BY AID 1122334455
- < OK
- > SELECT EF BY FID 6677
- < OK
- > READ, X BYTES
- < DATA + OK
- > SELECT EF BY FID 8899
- < OK
- > WRITE, Y BYTES + DATA
- < OK

Эмуляция NFC в Android

Эмуляция NFC в Android

- ОС получает от драйвера информацию о том что **устройство попало в поле**
- Производится стандартная инициализация сообщения по NFC где устройство **выбирает роль тега,** управление пока у ОС
- **Ридер отправляет тегу** APDU ISO SELECT ADF BY AID

Эмуляция NFC в Android

```xml
<?xml version="1.0" encoding="utf-8"?>
<host-apdu-service xmlns:android="http://schemas.android.com/apk/res/android"
    android:description="@string/nfc_type4_tag_emulation_service_name"
    android:requireDeviceUnlock="false">
    <aid-group android:description="@string/nfc_type4_tag_emulation_description" android:category="other">
        <aid-filter android:name="D2760000850101"/>
    </aid-group>
</host-apdu-service>
```

Ридер отправляет тегу APDU ISO SELECT ADF BY AID

C-APDU: 00 A4 04 00 07 D2 76 00 00 85 01 01 00

Эмуляция NFC в Android

ОС принимает APDU, парсит, извлекает DATA элемент и смотрит по объявленным в системе приложениями в манифестах компонентам есть ли кто объявлял этот AID

Если такое приложение есть, система стартует его процесс, запускает его компонент HostApduService и передаёт ему управление в processCommandApdu()

Pavel Vasilev

```
override fun processCommandApdu(commandApdu: ByteArray?, extras: Bundle?): ByteArray? {

}
```

Эмуляция NFC в Android – API

После получения управления в промежутке между ISO SELECT и деактивацией наши действия почти не ограничены, но!

- Должен соблюдаться установленный порядок C-APDU/R-APDU
- Респонсы должны быть в корректном формате ISO-7816-4 R-APDU
- Команды должны быть в корректном формате ISO-7816-4 C-APDU
- Ответ должен укладываться в определённый таймаут

Эмуляция NFC в Android – API

ISO определяет
только правила и формат передачи данных,
но не их содержание

Эмуляция NFC в Android – API

Понятно **«как»** эмулировать

Непонятно **«что»** эмулировать

Практика

Практика – что эмулируем?

В теории –
что угодно

Практика – что эмулируем?

- ПО на читающей стороне ожидает карту какого-то конкретного типа
- Эта система команд позволяет передать данные в определённом формате
- У карты есть своя система команд
- Необходимо повторить такую карту программно, на базе HostApduService

Практика – что эмулируем?

Варианты:

- Из всех наборов технологий беспроводных карт это должны быть именно **ВЧ карты (13.56 МГц)**
- Из всех ВЧ карт это должны быть **карты с транспортным протоколом ISO-14443-4**
- Из всех ISO-14443-4 карт это должны быть **карты с полноценным протоколом команд ISO-7816-4 (ISO SELECT)**

Практика – ищем технологию

Внезапно - найти что именно эмулировать – очень сложно !

Мало тегов которые подходят под критерии

Большинство карт имеет закрытую документацию

Мало ПО и ридеров знают о нужных тегах

Практика – для кого эмулируем?

В теории – для кого угодно

Самое ходовое устройство
в котором есть возможность читать
NFC – **современный смартфон**

Практика – для кого эмулируем?

iOS:

NFCNDEFReaderSession()

Android:

<action android:name="android.nfc.action.NDEF_DISCOVERED" />

NDEF

Теория – Стандарты – NDEF

NFC data exchange format

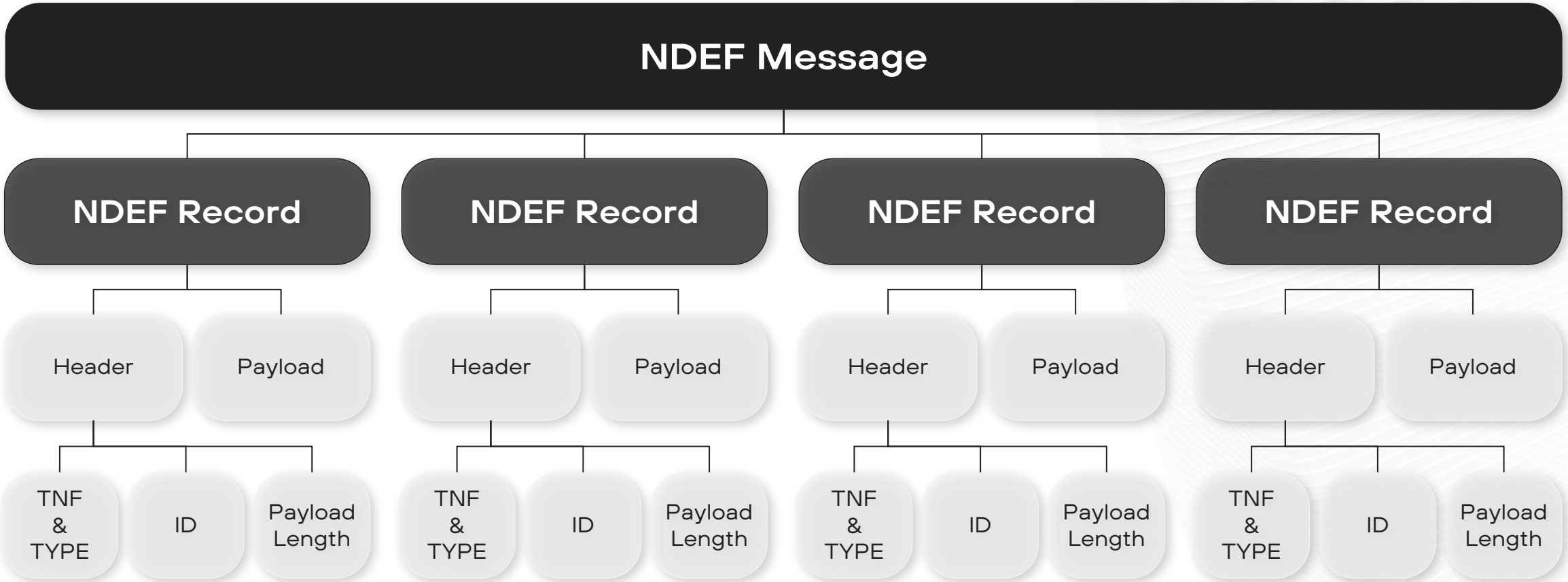
Стандартизированный несколькими компаниями и общепринятый всеми остальными способ передачи коротких сообщений с данными в заранее установленных форматах между устройствами

Теория – Стандарты – NDEF

TL/DR:

NDEF == QR

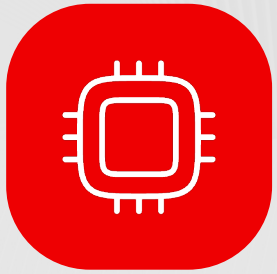
Теория – Стандарты – NDEF



Практика – ищем технологию

Со стороны ридера

NDEF форматированное сообщение о котором все знают



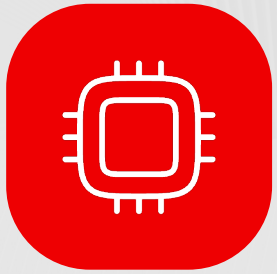
Со стороны карты

ISO-14443-4/ISO-7816-4 тэги, предусматривающие запись NDEF форматированных сообщений

Практика – ищем технологию

Со стороны ридера

NDEF форматированное сообщение о котором все знают



Со стороны карты

ISO-14443-4/ISO-7816-4 тэги, предусматривающие запись NDEF форматированных сообщений

NFC forum type 4 tag

Практика – NFC forum type 4 tag

- **Выработанный NFC forum стандарт** для NDEF форматированного тега
  учитывающий все современные реалии (ISO-14443-4, ISO-7816-4)
- **Известен большинству ПО,** включая мобильные ОС
- **Внезапно – не реализован** на реальном железе

Практика – NFC forum type 4 tag

```
<service android:name=".hce.NfcType4TagNdefEmulationService"
    android:permission="android.permission.BIND_NFC_SERVICE"
    android:exported="true">
    <intent-filter>
        <action android:name="android.nfc.cardemulation.action.HOST_APDU_SERVICE"/>
        <category android:name="android.intent.category.DEFAULT"/>
    </intent-filter>
    <meta-data
        android:name="android.nfc.cardemulation.host_apdu_service"
        android:resource="@xml/nfc_type4_tag_emulation"/>
</service>
```

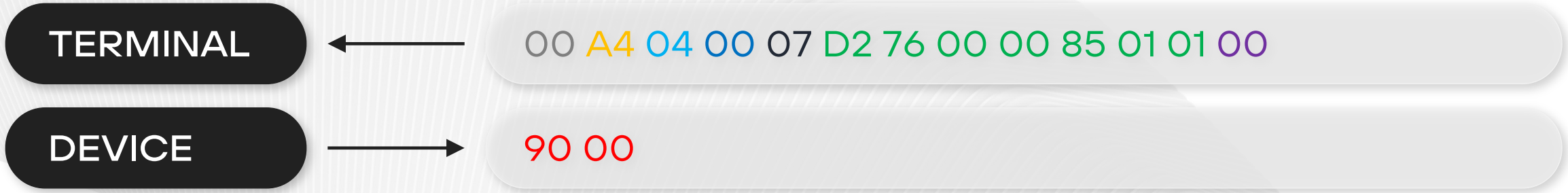
```
<?xml version="1.0" encoding="utf-8"?>
<host-apdu-service xmlns:android="http://schemas.android.com/apk/res/android"
    android:description="@string/nfc_type4_tag_emulation_service_name"
    android:requireDeviceUnlock="false">
    <aid-group android:description="@string/nfc_type4_tag_emulation_description" android:category="other">
        <aid-filter android:name="D2760000850101"/>
    </aid-group>
</host-apdu-service>
```

```
override fun processCommandApdu(commandApdu: ByteArray?, extras: Bundle?): ByteArray? {
    return byteArrayOf(0x90.toByte(), 0x00.toByte())
}
```

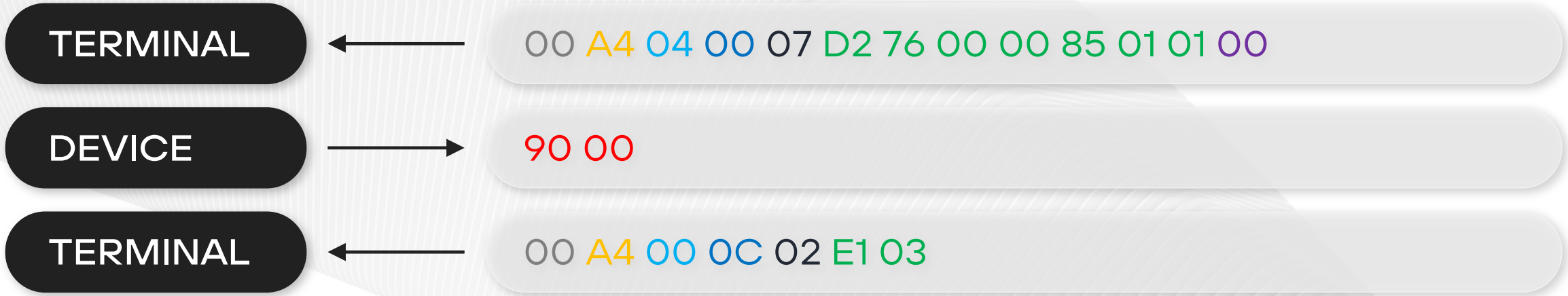
Практика – NFC forum type 4 tag – Реализация

```
// SELECT ADF NFC forum type 4 tag
if (cApdu.cla == 0x00.toByte()
    && cApdu.ins == 0xA4.toByte()
    && cApdu.p1 == 0x04.toByte()
    && cApdu.p2 == 0x00.toByte()
    && cApdu.lc != null && cApdu.lc.size == 1 && cApdu.lc[0] == 0x07.toByte()
    && cApdu.data != null && cApdu.data.size == cApdu.lc[0].toPositiveInt()
) {
    return if (NDEF_AID.contentEquals(cApdu.data)) {
        adfSelected = true
        ApduConstants.SW_OK
    } else {
        ApduConstants.SW_ERROR_NO_SUCH_DF
    }
}
```

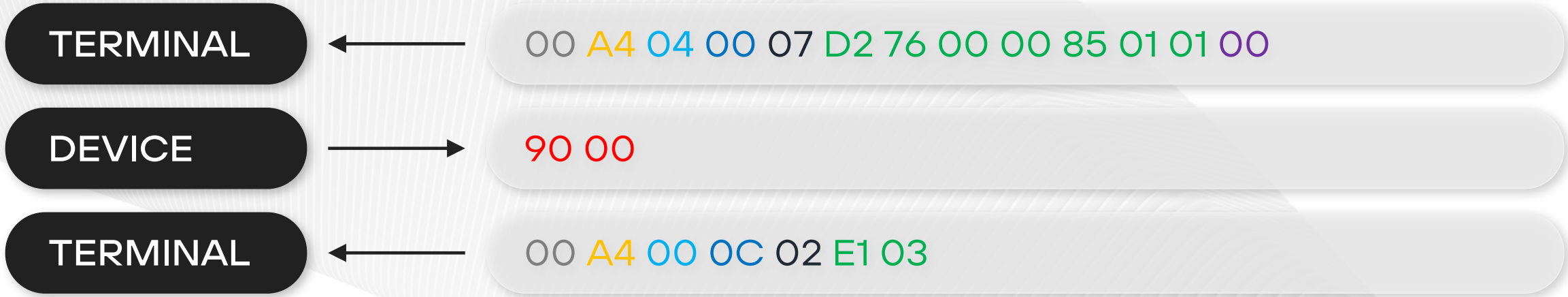
Практика – NFC forum type 4 tag



Практика – NFC forum type 4 tag



Практика – NFC forum type 4 tag



00 A4 00 0C == SELECT EF by FID

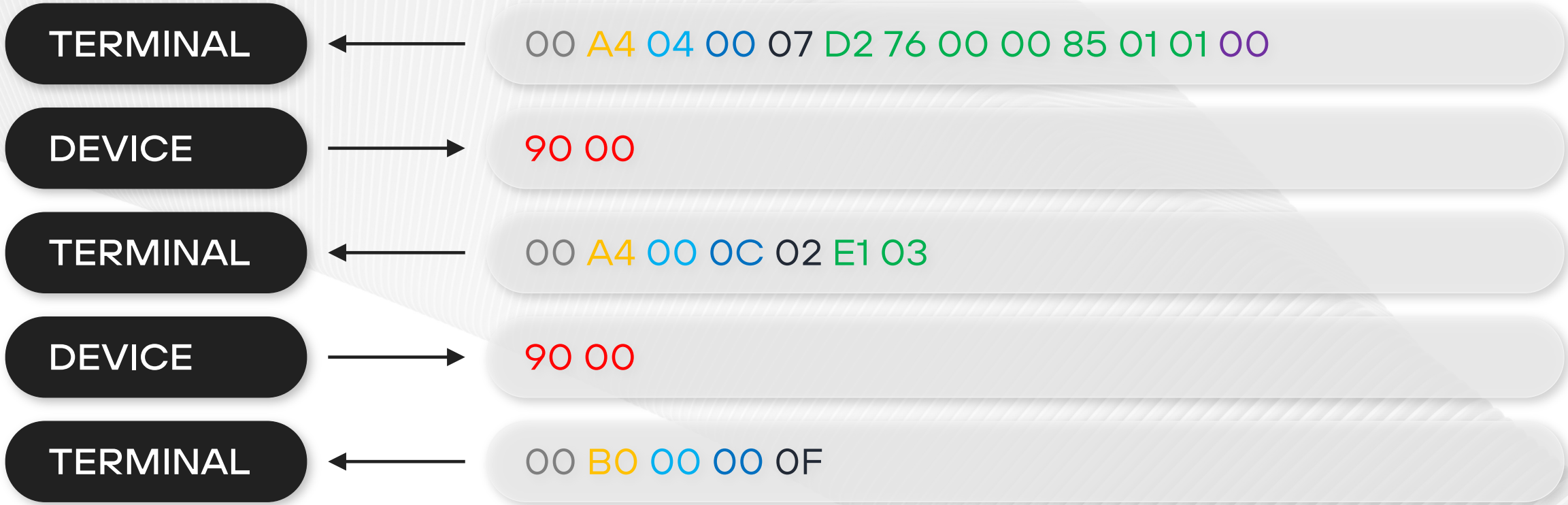
Практика – NFC forum type 4 tag – Реализация

```
// SELECT EF NFC forum type 4 tag CC
if (cApdu.cla == 0x00.toByte()
    && cApdu.ins == 0xA4.toByte()
    && cApdu.p1 == 0x00.toByte()
    && cApdu.p2 == 0x0C.toByte()
    && cApdu.lc != null && cApdu.lc.size == 1 && cApdu.lc[0] == 0x02.toByte()
    && cApdu.le == null
    && cApdu.data != null && cApdu.data.size == cApdu.lc[0].toPositiveInt()
    && adfSelected && !efCcSelected && !efNdefSelected
) {
    return if (NDEF_FID_CC.contentEquals(cApdu.data)) {
        efCcSelected = true
        ApduConstants.SW_OK
    } else {
        ApduConstants.SW_ERROR_NO_SUCH_DF
    }
}
```

Практика – NFC forum type 4 tag

TERMINAL	←	00 A4 04 00 07 D2 76 00 00 85 01 01 00
DEVICE	→	90 00
TERMINAL	←	00 A4 00 0C 02 E1 03
DEVICE	→	90 00

Практика – NFC forum type 4 tag



Практика – NFC forum type 4 tag

TERMINAL	←	00 A4 04 00 07 D2 76 00 00 85 01 01 00
DEVICE	→	90 00
TERMINAL	←	00 A4 00 0C 02 E1 03
DEVICE	→	90 00
TERMINAL	←	00 B0 00 00 0F

00 B0 00 00 0F == READ selected EF (15 bytes)

Практика – NFC forum type 4 tag

Структура данных

T4T ADF (AID == D2760000850101)

CC EF (FID == E103)

DATA EF (FID == E104)

Memory configuration of the Type 4 Tag

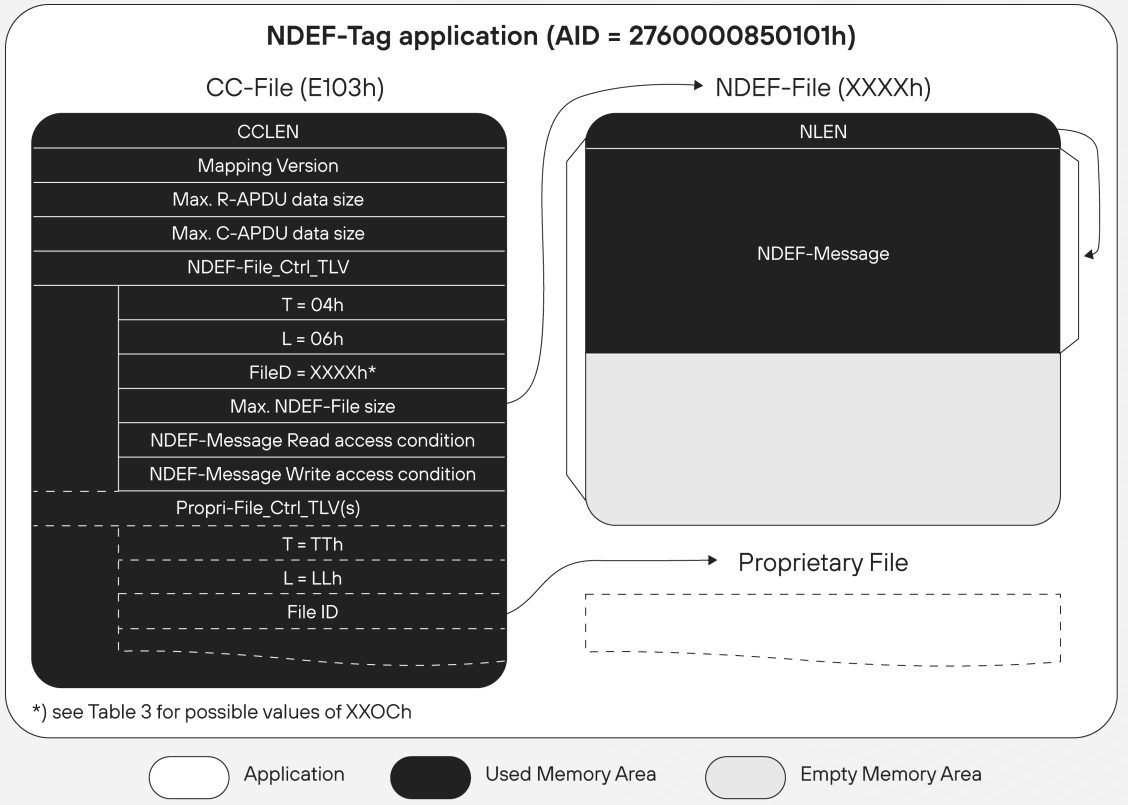


Figure 1: Example Application and File System Structure

Практика – NFC forum type 4 tag

Допустимые DATA EFID

Memory configuration of the Type 4 Tag

4.2 File Identifiers and Access Conditions

Table 3 lists the file identifiers that are available in an [ISO/IEC_7816-4] compliant file system.

Table 3: File Identifiers

Value	Description
0000h	Reserved ([ISO/IEC_7816-4])
0000h – 3EFFh	Valid range
3F00h	Reserved ([ISO/IEC_7816-4])
3F01h – 3FFEh	Valid range
3FFFh	Reserved ([ISO/IEC_7816-4])
4000h – E101h	Valid range
E102h	Reserved
E103h	FID_CC-File
E104h – FFFEh	Valid range
FFFFh	RFU ([ISO/IEC_7816-4])

Table 4 lists the READ access conditions for files within the NFC Forum application with AID_NDEF. This includes the READ access conditions for the NDEF Message.

Практика – NFC forum type 4 tag

Права на чтение запись

Table 4 lists the READ access conditions for files within the NFC Forum application with AID_NDEF. This includes the READ access conditions for the NDEF Message.

Table 4: File READ Access Conditions

Value	Description
00h	READ access granted without any security
01h – 7Fh	RFU
80h – FEh	Limited READ access, granted based on proprietary methods
FFh	RFU

Table 5 lists the WRITE access conditions for files within the NFC Forum application with AID_NDEF. This includes the WRITE access conditions for the NDEF Message.

Table 5: File WRITE Access Conditions

Value	Description
00h	WRITE access granted without any security
01h – 7Fh	RFU
80h – FEh	Limited WRITE access, granted based on proprietary methods
FFh	No WRITE access granted at all (i.e., Read only)

Практика – NFC forum type 4 tag – CC EF

Что получилось:

```
00 0F 20 FF FF FF FF 04 06 E1 04 00 35 00 FF
```

Практика – NFC forum type 4 tag – CC EF

> Что получилось:

00 0F 20 FF FF FF FF 04 06 E1 04 00 35 00 FF

[cc ef len - >= 0x0f]
[mapping version – 2.0]
[max c-apdu size – max]
[max r-apdu size – max]
[TLV – tag NDEF file, len 6 bytes]
[NDEF FID]
[NDEF data max size]
[read access – fully allowed]
[write access – fully forbidden]

Практика – NFC forum type 4 tag – CC EF

> Что получилось:

```
00 0F 20 FF FF FF FF 04 06 E1 04 00 35 00 FF
```

[cc ef len - >= 0x0f]
[mapping version – 2.0]
[max c-apdu size – max]
[max r-apdu size – max]
[TLV – tag NDEF file, len 6 bytes]
[NDEF FID]
[NDEF data max size]
[read access – fully allowed]
[write access – fully forbidden]

Практика – NFC forum type 4 tag – CC EF

Что получилось:

```
00 0F 20 FF FF FF FF 04 06 E1 04 00 35 00 FF
```

[cc ef len - >= 0x0f]
[mapping version – 2.0]
[max c-apdu size – max]
[max r-apdu size – max]
[TLV – tag NDEF file, len 6 bytes]
[NDEF FID]
[NDEF data max size]
[read access – fully allowed]
[write access – fully forbidden]

Практика – NFC forum type 4 tag – CC EF

Что получилось:

```
00 0F 20 FF FF FF FF 04 06 E1 04 00 35 00 FF
```

[cc ef len - >= 0x0f]
[mapping version – 2.0]
[max c-apdu size – max]
[max r-apdu size – max]
[TLV – tag NDEF file, len 6 bytes]
[NDEF FID]
[NDEF data max size]
[read access – fully allowed]
[write access – fully forbidden]

Практика – NFC forum type 4 tag – CC EF

Что получилось:

```
00 0F 20 FF FF FF FF 04 06 E1 04 00 35 00 FF
```

[cc ef len - >= 0x0f]
[mapping version – 2.0]
[max c-apdu size – max]
[max r-apdu size – max]
[TLV – tag NDEF file, len 6 bytes]
[NDEF FID]
[NDEF data max size]
[read access – fully allowed]
[write access – fully forbidden]

Практика – NFC forum type 4 tag – CC EF

> Что получилось:

```
00 0F 20 FF FF FF FF 04 06 E1 04 00 35 00 FF
```

[cc ef len - >= 0x0f]
[mapping version – 2.0]
[max c-apdu size – max]
[max r-apdu size – max]
[TLV – tag NDEF file, len 6 bytes]
[NDEF FID]
[NDEF data max size]
[read access – fully allowed]
[write access – fully forbidden]

Практика – NFC forum type 4 tag – CC EF

Что получилось:

```
00 0F 20 FF FF FF FF 04 06 E1 04 00 35 00 FF
```

[cc ef len - >= 0x0f]
[mapping version – 2.0]
[max c-apdu size – max]
[max r-apdu size – max]
[TLV – tag NDEF file, len 6 bytes]
[NDEF FID]
[NDEF data max size]
[read access – fully allowed]
[write access – fully forbidden]

Практика – NFC forum type 4 tag – CC EF

Что получилось:

```
00 0F 20 FF FF FF FF 04 06 E1 04 00 35 00 FF
```

[cc ef len - >= 0x0f]
[mapping version – 2.0]
[max c-apdu size – max]
[max r-apdu size – max]
[TLV – tag NDEF file, len 6 bytes]
[NDEF FID]
[NDEF data max size]
[read access – fully allowed]
[write access – fully forbidden]

Практика – NFC forum type 4 tag – CC EF

Что получилось:

```
00 0F 20 FF FF FF FF 04 06 E1 04 00 35 00 FF
```

[cc ef len - >= 0x0f]
[mapping version – 2.0]
[max c-apdu size – max]
[max r-apdu size – max]
[TLV – tag NDEF file, len 6 bytes]
[NDEF FID]
[NDEF data max size]
[read access – fully allowed]
[write access – fully forbidden]

Практика – NFC forum type 4 tag – CC EF

> Что получилось:

00 0F 20 FF FF FF FF 04 06 E1 04 00 35 00 FF

[cc ef len - >= 0x0f]
[mapping version – 2.0]
[max c-apdu size – max]
[max r-apdu size – max]
[TLV – tag NDEF file, len 6 bytes]
[NDEF FID]
[NDEF data max size]
[read access – fully allowed]
[write access – fully forbidden]

Практика – NFC forum type 4 tag – Реализация

```
// READ EF NFC forum type 4 tag CC
if (cApdu.cla == 0x00.toByte()
    && cApdu.ins == 0xB0.toByte()
    && cApdu.p1 == 0x00.toByte()
    && cApdu.p2 == 0x00.toByte()
    && cApdu.lc == null
    && cApdu.data == null
    && cApdu.le != null && cApdu.le.size == 1 && cApdu.le[0] == 0x0F.toByte()
    && adfSelected && efCcSelected && !efNdefSelected
) {
    val ccPrefix = defineByteArrayOf(0x00, 0x0F, 0x20, 0xFF, 0xFF, 0xFF, 0xFF, 0x04, 0x06)
    return DataUtil.concatByteArrays(ccPrefix, NdefConstants.NDEF_FID_DATA, ndefMessageMaxSize, NDEF_CC_READ_MODE_GRANTED, NDEF_CC_WRITE_MODE_FORBIDDEN, ApduConstants.SW_OK)
}
```

Практика – NFC forum type 4 tag

TERMINAL	←	00 A4 04 00 07 D2 76 00 00 85 01 01 00
DEVICE	→	90 00
TERMINAL	←	00 A4 00 0C 02 E1 03
DEVICE	→	90 00
TERMINAL	←	00 B0 00 00 0F
DEVICE	→	00 0F 20 FF FF FF FF 04 06 E1 04 00 35 00 FF 90 00

Практика – NFC forum type 4 tag

TERMINAL	←	00 A4 04 00 07 D2 76 00 00 85 01 01 00
DEVICE	→	90 00
TERMINAL	←	00 A4 00 0C 02 E1 03
DEVICE	→	90 00
TERMINAL	←	00 B0 00 00 0F
DEVICE	→	00 0F 20 FF FF FF FF 04 06 E1 04 00 35 00 FF 90 00
TERMINAL	←	00 A4 00 0C 02 E1 04

Практика – NFC forum type 4 tag – Реализация

```
// SELECT EF NDEF data
// * checking only for adf selection for simplicity and not CC or NDEF selection because ios tend to duplicate select fid commands multiple times
if (cApdu.cla == 0x00.toByte()
    && cApdu.ins == 0xA4.toByte()
    && cApdu.p1 == 0x00.toByte()
    && cApdu.p2 == 0x0C.toByte()
    && cApdu.lc != null && cApdu.lc.size == 1 && cApdu.lc[0] == 0x02.toByte()
    && cApdu.le == null
    && cApdu.data != null && cApdu.data.size == cApdu.lc[0].toPositiveInt()
    && adfSelected
) {
    return if (NdefConstants.NDEF_FID_DATA.contentEquals(cApdu.data)) {
        efCcSelected = false
        efNdefSelected = true
        ApduConstants.SW_OK
    } else {
        ApduConstants.SW_ERROR_NO_SUCH_DF
    }
}
```

Практика – NFC forum type 4 tag

TERMINAL	←	00 A4 04 00 07 D2 76 00 00 85 01 01 00
DEVICE	→	90 00
TERMINAL	←	00 A4 00 0C 02 E1 03
DEVICE	→	90 00
TERMINAL	←	00 B0 00 00 0F
DEVICE	→	00 0F 20 FF FF FF FF 04 06 E1 04 00 35 00 FF 90 00
TERMINAL	←	00 A4 00 0C 02 E1 04
DEVICE	→	90 00

Практика – NFC forum type 4 tag

TERMINAL	←	00 A4 04 00 07 D2 76 00 00 85 01 01 00
DEVICE	→	90 00
TERMINAL	←	00 A4 00 0C 02 E1 03
DEVICE	→	90 00
TERMINAL	←	00 B0 00 00 0F
DEVICE	→	00 0F 20 FF FF FF FF 04 06 E1 04 00 35 00 FF 90 00
TERMINAL	←	00 A4 00 0C 02 E1 04
DEVICE	→	90 00
TERMINAL	←	00 B0 00 00 35

Практика – NFC forum type 4 tag – NDEF EF

Формат пейлоада – NDEF size + NDEF message

00 33 D9 01 2C 02 55 E1 04 00 68 74 74 70 73 3A 2F 2F 77 77 77 2E 79 6F 75 74 75 62 65 2E 63 6F 6D 2F 77 61 74 63 68 3F 76 3D 64 51 77 34 77 39 57 67 58 63 51

NDEF size – 0x0033 bytes

Flags + TNF well known (0x01) + etc

RTD URI (0x55)

NDEF FID (0xE104)

Data — ссылка

Практика – NFC forum type 4 tag – NDEF EF

Формат пейлоада –
NDEF size + NDEF message

00 33 D9 01 2C 02 55 E1 04 00 68 74 74 70 73 3A 2F 2F 77 77 77 2E 79 6F 75 74 75 62 65 2E 63 6F 6D 2F 77 61 74 63 68 3F 76 3D 64 51 77 34 77 39 57 67 58 63 51

NDEF size – 0x0033 bytes

Flags + TNF well known (0x01) + etc

RTD URI (0x55)

NDEF FID (0xE104)

Data — ссылка

Практика – NFC forum type 4 tag – NDEF EF

Формат пейлоада – NDEF size + NDEF message

00 33 D9 01 2C 02 55 E1 04 00 68 74 74 70 73 3A 2F 2F 77 77 77 2E 79 6F 75 74 75 62 65 2E 63 6F 6D 2F 77 61 74 63 68 3F 76 3D 64 51 77 34 77 39 57 67 58 63 51

NDEF size – 0x0033 bytes

Flags + TNF well known (0x01) + etc

RTD URI (0x55)

NDEF FID (0xE104)

Data — ссылка

Практика – NFC forum type 4 tag – NDEF EF

Формат пейлоада –
NDEF size + NDEF message

00 33 D9 01 2C 02 55 E1 04 00 68 74 74 70 73 3A 2F 2F 77 77 77 2E 79 6F 75 74 75 62
65 2E 63 6F 6D 2F 77 61 74 63 68 3F 76 3D 64 51 77 34 77 39 57 67 58 63 51

NDEF size – 0x0033 bytes

Flags + TNF well known (0x01) + etc

RTD URI (0x55)

NDEF FID (0xE104)

Data — ссылка

Практика – NFC forum type 4 tag – NDEF EF

Формат пейлоада – NDEF size + NDEF message

00 33 D9 01 2C 02 55 E1 04 00 68 74 74 70 73 3A 2F 2F 77 77 77 2E 79 6F 75 74 75 62
65 2E 63 6F 6D 2F 77 61 74 63 68 3F 76 3D 64 51 77 34 77 39 57 67 58 63 51

NDEF size – 0x0033 bytes

Flags + TNF well known (0x01) + etc

RTD URI (0x55)

NDEF FID (0xE104)

Data — ссылка

Практика – NFC forum type 4 tag – NDEF EF

Формат пейлоада –
NDEF size + NDEF message

00 33 D9 01 2C 02 55 E1 04 00 68 74 74 70 73 3A 2F 2F 77 77 77 2E 79 6F 75 74 75 62
65 2E 63 6F 6D 2F 77 61 74 63 68 3F 76 3D 64 51 77 34 77 39 57 67 58 63 51

NDEF size – 0x0033 bytes

Flags + TNF well known (0x01) + etc

RTD URI (0x55)

NDEF FID (0xE104)

Data — ссылка

Практика – NFC forum type 4 tag – NDEF EF

Формат пейлоада –
NDEF size + NDEF message

https://www.youtube.com/watch?v=dQw4w9WgXcQ

00 33 D9 01 2C 02 55 E1 04 00 68 74 74 70 73 3A 2F 2F 77 77 77 2E 79 6F 75 74 75 62
65 2E 63 6F 6D 2F 77 61 74 63 68 3F 76 3D 64 51 77 34 77 39 57 67 58 63 51

NDEF size – 0x0033 bytes

Flags + TNF well known (0x01) + etc

RTD URI (0x55)

NDEF FID (0xE104)

Data — ссылка

Практика – NFC forum type 4 tag – NDEF EF

Table 1. TNF Field Values

Type Name Format	Value
Empty	0x00
NFC Forum well-known type [NFC RTD]	0x01
Media-type as defined in RFC 2046 [RFC 2046]	0x02
Absolute URI as defined in RFC 3986 [RFC 3986]	0x03
NFC Forum external type [NFC RTD]	0x04
Unknown	0x05
Unchanged (see section 2.3.3)	0x06
Reserved	0x07

Практика – NFC forum type 4 tag – NDEF EF

Для TNF well known записей
RTD – Record type definition

```
RTD Text type. For use with TNF_WELL_KNOWN.
See Also: TNF_WELL_KNOWN
public static final byte[] RTD_TEXT = {0x54};  // "T"

RTD URI type. For use with TNF_WELL_KNOWN.
See Also: TNF_WELL_KNOWN
public static final byte[] RTD_URI = {0x55};   // "U"
```

Для media-type записей
указываем просто mime-тип

Практика – NFC forum type 4 tag – NDEF EF

NDEF record с данными нам может помочь подготовить сам андроид

(конструкторы NdefMessage(NdefRecord())), но нужно заложить или указание языка или 0x00 перед сообщением

```
return NdefRecord(NdefRecord.TNF_WELL_KNOWN, NdefRecord.RTD_URI, NdefConstants.NDEF_FID_DATA, recordPayload)
```

Практика – NFC forum type 4 tag – Реализация

```
// READ EF NDEF data
// * taking multiple last bytes as Le here because android can sometimes send 3-bytes Le in extended apdu for
// * returning content in slices here because one response may not be enough to transfer the entire data, als
if (cApdu.cla == 0x00.toByte()
    && cApdu.ins == 0xB0.toByte()
    && cApdu.le != null && cApdu.le.size > 0 && cApdu.le.size <= 2
    && adfSelected && !efCcSelected && efNdefSelected
) {
    val offset = DataUtil.bytesToShort(byteArrayOf(cApdu.p1, cApdu.p2))
    val length = DataUtil.bytesToShort(cApdu.le)
    val responseFull = DataUtil.concatByteArrays(ndefMessageBinarySize, ndefMessageBinary)
    val responseChunk = responseFull.sliceArray( indices: offset ≤ until < offset + length)
    val response = ByteArray( size: responseChunk.size + ApduConstants.SW_OK.size)
    System.arraycopy(responseChunk, srcPos: 0, response, destPos: 0, responseChunk.size)
    System.arraycopy(ApduConstants.SW_OK, srcPos: 0, response, responseChunk.size, ApduConstants.SW_OK.size)
    return response
}
```

Практика – NFC forum type 4 tag – Работа

TERMINAL	←	00 A4 04 00 07 D2 76 00 00 85 01 01 00
DEVICE	→	90 00
TERMINAL	←	00 A4 00 0C 02 E1 03
DEVICE	→	90 00
TERMINAL	←	00 B0 00 00 0F
DEVICE	→	00 0F 20 FF FF FF FF 04 06 E1 04 00 35 00 FF 90 00
TERMINAL	←	00 A4 00 0C 02 E1 04
DEVICE	→	90 00
TERMINAL	←	00 B0 00 00 35
DEVICE	→	00 33 D9 01 2C 02 55 E1 04 00 68 74 74 70 73 3A 2F 2F 77 77 77 2E 79 6F 75 74 75 62 65 2E 63 6F 6D 2F 77 61 74 63 68 3F 76 3D 64 51 77 34 77 39 57 67 58 63 51 90 00

Практика – NFC forum type 4 tag - Работа

- > SELECT AID
- < SW_OK
- > SELECT CC FID
- < SW_OK
- > READ
- < CC DATA + SW_OK
- > SELECT NDEF FID
- < OK
- > READ LEN BYTES
- < NDEF DATA + SW_OK

Практика – NFC forum type 4 tag – Проблемы

- **Если сделать совсем наивную реализацию –** работать не будет ни при чтении с Android, ни при чтении с iOS

- К сожалению, необходима **пара небольших костылей для них**

Практика – NFC forum type 4 tag – Проблемы – iOS

- **Читает** CC EF,
  получает NDEF EF len
- **Игнорирует** TLV в NDEF EF,
  читает L-2 байт
- В первом прочитанном сообщении
  получает общую длину сообщения,
  понимает нужно ли читать дальше
- **Читает** последние два байта

RX: 00 A4 00 0C 02 E1 04

TX: 90 00

RX: 00 B0 00 00 02

TX: 00 33 90 00

RX: 00 B0 00 00 33

TX: 00 33 D9 01 2C 02 55 E1 04 00 68 74 74 70 73 3A 2F 2F 77 77 77 2E 79 6F 75 74 75 62 65 2E 63 6F 6D 2F 77 61 74 63 68 3F 76 3D 64 51 77 34 77 39 57 67 58 90 00

RX: 00 B0 00 33 02

TX: 63 51 90 00

Практика – NFC forum type 4 tag – Проблемы – iOS

- Может неожиданно **оборвать** чтение тега **и начать заново**
- Посылает команды **SELECT DF** и **SELECT EF** по многу раз

RX: 00 A4 04 00 07 D2 76 00 00 85 01 01 00
TX: 90 00
RX: 00 A4 04 00 07 D2 76 00 00 85 01 01 00
TX: 90 00
RX: 00 A4 00 0C 02 E1 03
TX: 90 00
RX: 00 B0 00 00 0F
TX: 00 0F 20 FF FF FF FF 04 06 E1 04 00 35 00 FF 90 00
...
RX: 00 A4 04 00 07 D2 76 00 00 85 01 01 00
TX: 90 00
RX: 00 A4 00 0C 02 E1 03
TX: 90 00
RX: 00 A4 00 0C 02 E1 03
TX: 90 00

Практика – NFC forum type 4 tag – Проблемы – Android

Читает CC EF, **получает** NDEF EF len

Присылает команду чтения NDEF EF **в extended-length-APDU формате**

RX: 00 A4 00 0C 02 E1 03
TX: 90 00
RX: 00 B0 00 00 0F
TX: 00 0F 20 FF FF FF FF 04 06 E1 04 00 35 00 FF 90 00
...
RX: 00 A4 00 0C 02 E1 04
TX: 90 00
RX: 00 B0 00 00 02
TX: 00 33 90 00
RX: 00 B0 00 02 00 00 33
TX: D9 01 2C 02 55 E1 04 00 68 74 74 70 73 3A 2F 2F 77 77 77 2E 79 6F 75 74 75 62 65 2E 63 6F 6D 2F 77 61 74 63 68 3F 76 3D 64 51 77 34 77 39 57 67 58 63 51 90 00

Практика – NFC forum type 4 tag – Проблемы – Android

TABLE 5-1 Extended APDU Format

Field	Description	Number of Bytes
Command Header	Class byte CLA	1
Command Header	Instruction byte INS	1
Command Header	Parameter bytes P1- P2	2
LC Field	Absent for Nc = 0. Present for Nc > 0	0, 1 or 3
Data Field	Absent if Ne = 0, present if Ne >0	Nc
LE Field	Absent for Ne = 0, present for Ne > 0	0, 1, 2 or 3
Response Data	Absent if Nr = 0, present if Nr >0	Nr (max. Ne)
Response Status	Status bytes SW1 SW2	2
	NOTATION Nc = command data length Ne = expected response data length Nr = actual response data length	

Практика – NFC forum type 4 tag – Код

https://github.com/LuigiVampa92/ndef-emulator

Однострочное апи:

```
NdefEmulation(this).currentEmulatedNdefData =

UriNdefData("https://www.youtube.com/watch?v=dQw4w9WgXcQ")
```

Практика – NFC forum type 4 tag – Типы сообщений

URI (TNF well known, RTD URI)

URL - https://www.youtube.com/watch?v=dQw4w9WgXcQ

URI/Deeplink:

- vnd.youtube://www.youtube.com/watch?v=dQw4w9WgXcQ
- tg://msg?to=+79123456789
- https://wa.me/79123456789

- Самый частый случай
- То для чего и был сделан NDEF, тот самый аналог QR
- Подходит для IPC, запуска приложений итд
- Работает из коробки на Android и на iOS

Практика – NFC forum type 4 tag – Типы сообщений

Text (TNF well known, RTD text)

Обычный текст

- Редкий случай
- Не работает из коробки ни на Android, ни на iOS, нужно читать приложением

Практика – NFC forum type 4 tag – Типы сообщений

GEO (TNF well known, RTD URI)

URI geo:
https://geouri.org , https://ru.wikipedia.org/wiki/Geo-URI

- geo:25.245470718844146,51.45400942457904
- geo:37.786971,-122.399677;u=35
- geo:323482,4306480;CRS=epsg:32718;U=20;mapcolors=for_daltonic

Хороший случай

Работает из коробки на Android, на iOS требует приложения поддерживающего схему, например MapsMe

Практика – NFC forum type 4 tag – Типы сообщений

WiFi (TNF MIME media, mime-type "application/vnd.wfa.wsc")

Описание подключения к WiFi точке – SSID, PSK, ...
https://cs.android.com/android/platform/superproject/main/+/main:packages/apps/Nfc/src/com/android/nfc/NfcWifiProtectedSetup.java

- Удобный случай, но с ограничениями
- Работает из коробки на Android но не работает на iOS

Практика – NFC forum type 4 tag – Типы сообщений

Contact (TNF MIME media, mime-type "text/vcard")

Визитка:
- Имя
- Фамилия
- Номер телефона
- Почта

- Удобный случай, но с ограничениями
- Работает из коробки на Android но не работает на iOS

Практика – NFC forum type 4 tag – Типы сообщений

NDEF

- Прочитали средствами Android NDEF сообщение
- Сразу же кинули его эмулироваться и раздали дальше


@luigivampa92
luigivampa92@gmail.com
Спасибо!