



Санкт-Петербургский государственный университет  
Кафедра системного программирования

# Разработка СПО прикладного уровня системы оплаты проезда российского производства на платных автодорогах

Пантелеймонов Андрей Радиевич, группа 23.M04-мм

**Научный руководитель:** д.ф.-м.н. А.Н. Терехов, профессор кафедры системного программирования

**Консультант:** А.Г. Шадрин, ктн, Инженер-исследователь ООО «ЛИС»

Санкт-Петербург  
2024

- Ежедневно платными дорогами пользуются сотни тысяч водителей
- Ключ к большой пропускной способности - бесперебойное и быстрое функционирование системы оплаты проезда
- Уход иностранных организаций, осуществлявших обслуживание на платных автодорогах породил острую необходимость создания российской системы

- Компании ООО «ЛИС» и ООО «Мобил-групп» в тесном сотрудничестве создают комплекс ПО и АО (аппаратное обеспечение), которое позволит в полном объёме заменить ушедшие аналоги
- Продуктом будет АО с установленным ПО, позволяющим проводить весь цикл обработки и передачи информации

# Постановка задачи

**Цель:** Разработать ПО для аппаратного обеспечения для работы на системах оплаты проезда

## Задачи:

- Изучить имеющуюся документацию предыдущих производителей на российском рынке
- Изучить соответствующие стандарты необходимые для разработки системы
- Реализовать обмен между канальным уровнем и прикладным
- Реализовать выдачу информации в режиме реального времени для информирования оператора о текущем статусе в терминах описанных в документах протоколов
- Реализовать логирование работы системы для анализа ошибок и сбора статистики
- Реализовать демонизацию ПО
- Провести апробацию продукта

# Обзор предметной области

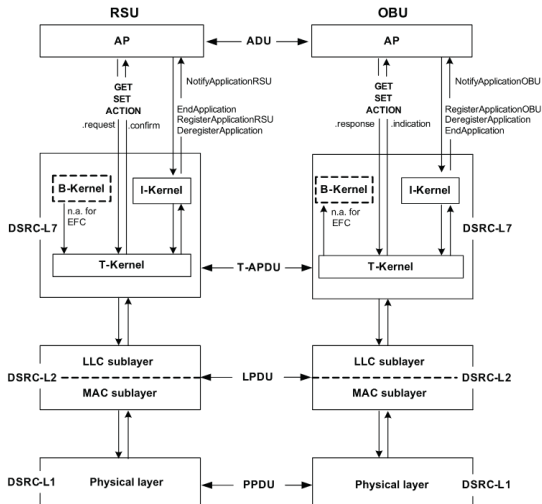


Рис.: Архитектура стека DSRC

# Обзор предметной области





|                      |                                                                                   |                                                                                                                                                                             |
|----------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Initialisation       |  | <b>'Hello, welcome, where do you come from, how do you want to pay'</b><br>Negotiation of the EFC contract to use                                                           |
| Presentation         |  | <b>'Please give me your payment details and your entry ticket'</b><br>The RSE reads OBE data (details on contract, account, vehicle classification, last transaction, etc.) |
| Receipt              |  | <b>'Here is your receipt'</b><br>The RSE writes an electronic receipt (which may also serve as an entry ticket)                                                             |
| Tracking and Closing |  | <b>'Thank you and good bye'</b><br>The RSE tracks the vehicle through the communication zone and eventually closes the transaction.                                         |

Рис.: Процесс транзакции кратко

# Обзор предметной области

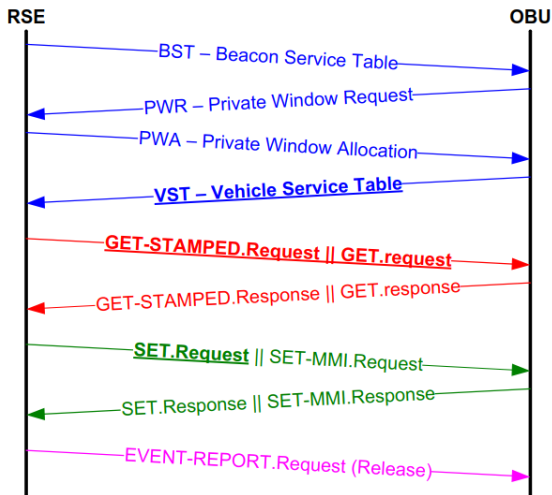


Рис.: Процесс транзакции чуть более подробно

- Norbit(Норвегия) — комплекс ПО и АО для осуществления полного цикла обмена информацией между RSU (Road-Side Unit, или устройство, принимающее сигнал с транспондера) и OBU (On-Board Unit, транспондер, который находится непосредственно в автомобиле)

Решение Norbit являлось фактически монополистом в системах, которые отвечают за систему оплаты проезда на платных автодорогах в России, в связи с чем было решено отталкиваться именно от документации, некогда предоставленной этим разработчиком.



# Выбор редактора диаграмм

Было принято решение использовать уже существующий редактор диаграмм, на основе элементов которого строить диаграммы сети, после чего отправлять данные в другие компоненты приложения

## Требования

- Возможность добавлять и хранить метаданные для элементов диаграмм
- Возможность экспорта диаграммы не только в графическом формате, но и в виде кода на одном из языков разметки
- Лицензия, позволяющая использовать данный редактор в разработке коммерческого ПО

- В качестве языка программирования используется язык C++ 11 стандарта
- В качестве базовой ОС, на которой будет работать ПО, была выбрана Ubuntu

## Формат входных данных

В качестве входящих данных прикладному уровню с канального уровня через LLC подуровень приходит строка в hex формате, например:

```
100a002c 00600000 24760e71 c0039190 0001c101 02105700 01ff0070 02021dd1  
0204118e 0d7bf301 00320100 00320100
```

Здесь первые 2 октета - заголовок, а последний - незначащий. Однако для дальнейшей работы необходимо перевести информацию в двоичный формат.

# Формат входных данных

На скриншоте видно, что код страны - 10 бит, причём в кодировке ITA2

|    |                                       |           |                                                                                                                      |
|----|---------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------|
|    | AID DSRCApplicationEntityID           | 00 0001   | No extension, AID = 1, EFC                                                                                           |
| 23 | EID                                   | 0000 0101 | Associated with a context mark. Example : 5 <sub>10</sub>                                                            |
| 24 | Parameter CONTAINER {                 | 0000 0010 | No extension, Container Choice = 2 <sub>10</sub> , Octet string                                                      |
| 25 |                                       | 0001 0000 | No extension, octet string length = 16 <sub>10</sub>                                                                 |
| 26 | EFC-ContextMark SEQUENCE {            |           |                                                                                                                      |
|    | ContractProvider SEQUENCE {           |           |                                                                                                                      |
|    | CountryCode BIT STRING (SIZE(10))     | 1010 0100 | 10 bit country code according to ISO 3166 with ITA2 binary                                                           |
|    |                                       | 00        | Encoding based on ISO 14816. Example : SE                                                                            |
| 27 | IssuerIdentifier INTEGER (0..16383) } | 00 0000   | 14 bits issuer identifier. Example : 1 <sub>10</sub> (Öresundskonsortiet)                                            |
| 28 |                                       | 0000 0001 |                                                                                                                      |
| 29 | TypeOfContract OCTET STRING (SIZE(2)) | 0000 0000 | Type of contract. Example : 2 <sub>10</sub>                                                                          |
| 30 |                                       | 0000 0010 |                                                                                                                      |
| 31 | ContextVersion INTEGER (0..127,...) } | 0000 0001 | No extension, context version. Example : 1 <sub>10</sub>                                                             |
| 32 | CONTAINER {                           | 0000 0010 | No extension, Container Choice = 2 <sub>10</sub> , Octet string                                                      |
| 33 |                                       | 0000 0010 | No extension, octet string length = 2 <sub>10</sub>                                                                  |
| 34 | AC_CR-Reference SEQUENCE {            | 0000 0001 | AC_CR-Reference to, consisting of AC_CR-MasterKeyRef and AC_CR-Diversifier, used for the computation of AC_CRKey and |
|    | AC-MasterKeyRef Int1,                 |           |                                                                                                                      |
| 35 | AC_CR-Diversifier Int1 ) }            | 0000 0001 | AC_CR.                                                                                                               |
| 36 | CONTAINER {                           | 0000 0010 | No extension, Container Choice = 2 <sub>10</sub> , Octet string                                                      |

Рис.: Подробная структура полей, пример из стандарта ISO14906

# Протокол EARP

Компанией Norbit был разработан протокол EARP (EFC Attribute Read Protocol), который позволяет получить информацию о совершенных транзакциях. Примером вывода информации согласно этому протоколу является:

```
20100611T095008.513 578,00008,00001,001:-----  
0000/002a/0000  
[104/00645 +015.7/05.1/00.0]  
AutoPASS <0EC7AC9DB0EA114C> |  
099:021830C008CC3CDE140000C28422C6DF9C187A21EDF42A770F2B |
```

На скриншоте каждый элемент имеет конкретное описание. Так, например, первые 19 символов - текущее время, первые 3 цифры после пробела - код страны данной транзакции согласно ISO 3166-1. На скриншоте это число 578 - Норвегия (согласно 3166-1)

В качестве выходных данных протокола EARP (для наглядности) для примера со входными данными, получается следующая строка:

**20240104T004349.744**

**643,00001,65280,112:02021dd10204118e0d7b f301/0032/0100**

Для теста (входных данных) был взят транспондер одного из автовладельцев, ежедневно пользующегося услугами ЗСД, код 643 соответствует России.

## Формат выходных данных

Страна приходит в бинарном формате длиной 10 бит в алфавите ITA2, поэтому необходимо в двоичном потоке найти этот кусок и выделить 10 бит:

```
Есть EID  
Есть Parameter  
00000001 - EID в сообщении  
0000001000010000 - Parameter в сообщении  
0101011100 - Country Code в сообщении
```

## Формат выходных данных

Следующим шагом является уже установка соответствия согласно алфавиту ITA-2

Table 1 — ITA-2 alphabet

|   |       |   |       |
|---|-------|---|-------|
| A | 11000 | N | 00110 |
| B | 10011 | O | 00011 |
| C | 01110 | P | 01101 |
| D | 10010 | Q | 11101 |
| E | 10000 | R | 01010 |
| F | 10110 | S | 10100 |
| G | 01011 | T | 00001 |
| H | 00101 | U | 11100 |
| I | 01100 | V | 01111 |
| J | 11010 | W | 11001 |
| K | 11110 | X | 10111 |
| L | 01001 | Y | 10101 |
| M | 00111 | Z | 10001 |

Так, на предыдущем слайде поле country code имело значение 01010'11100, тогда согласно этой таблице получим, что страна Россия (RU)



# Текущие результаты

- Изучена необходимая документация для реализации прикладного уровня
- Изучены стандарты необходимые для реализации
- Частично реализован цикл обмена информацией между RSU и OBU
- Реализован перевод данных в протокол EARP
- Реализовано логирование поступающей информации

## Дальнейшие планы

- Реализовать полный цикл обмена информацией
- Реализовать отправку информации в биллинг
- Провести апробацию продукта

Исходный код находится на локальном Gitlab