



Семинар
«Технологии QNX и ЗОСРВ «Нейтрино»
в России»

Москва, 19 апреля 2016

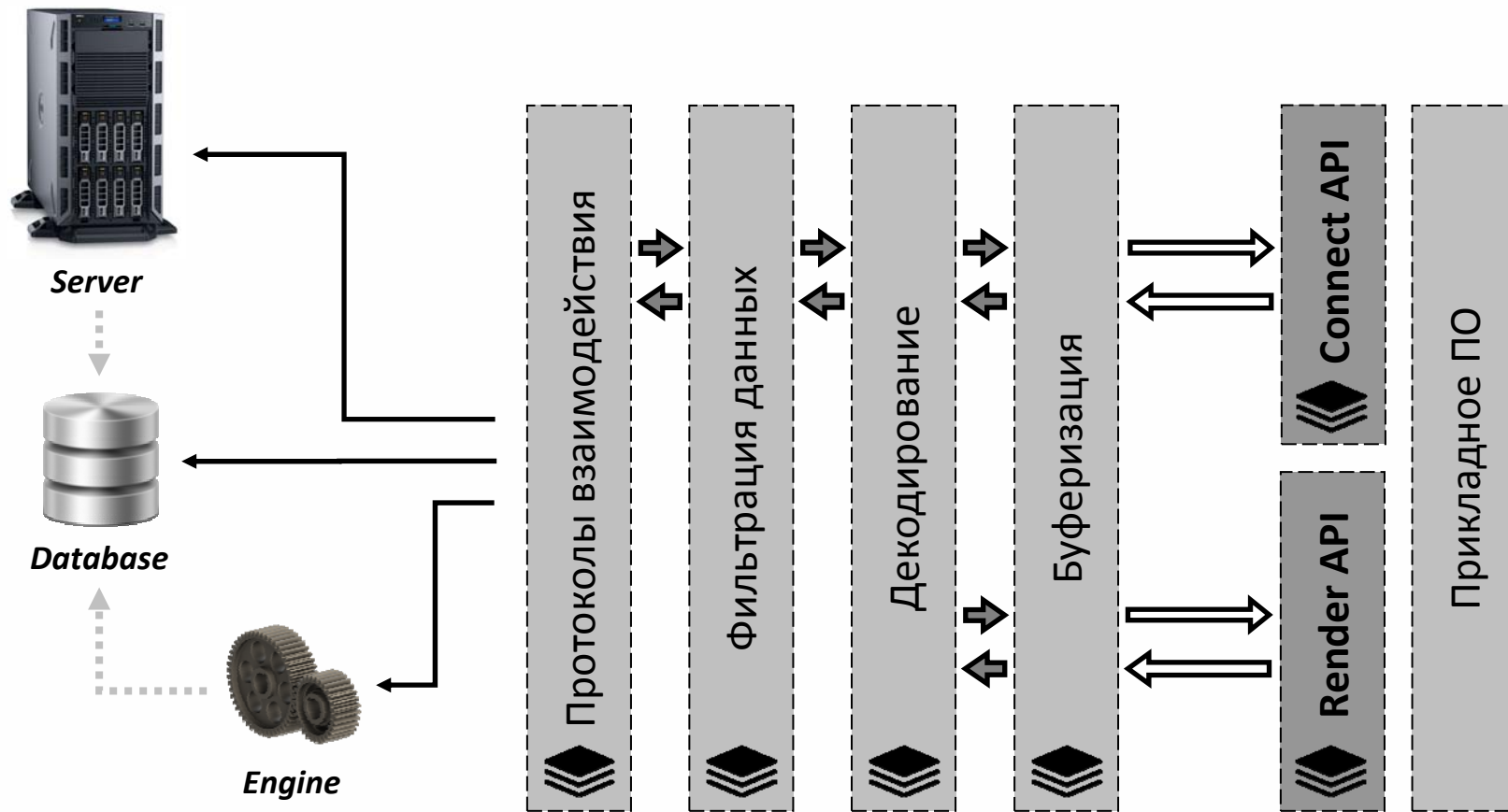
**«Картографический пакет для QNX и ЗОСРВ «Нейтрино»:
функциональность и практические примеры»**

Девяткин Алексей, СВД Встраиваемые Системы

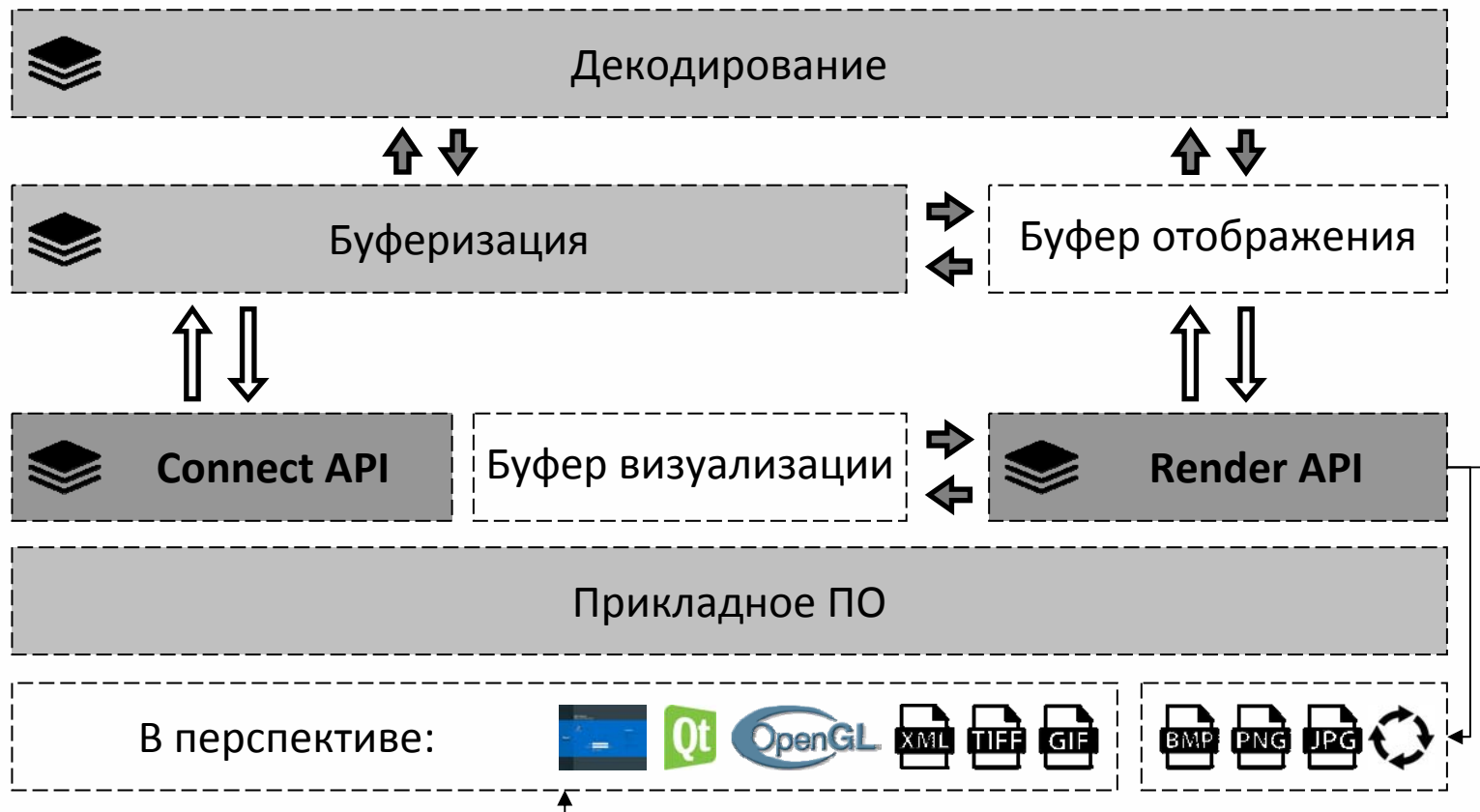


- **Картографический пакет для ЗОСРВ «Нейтрино» / QNX**
 - Структура и функциональность
 - Пример использования API
 - Дальнейшее развитие проекта

Архитектура картографического сервиса













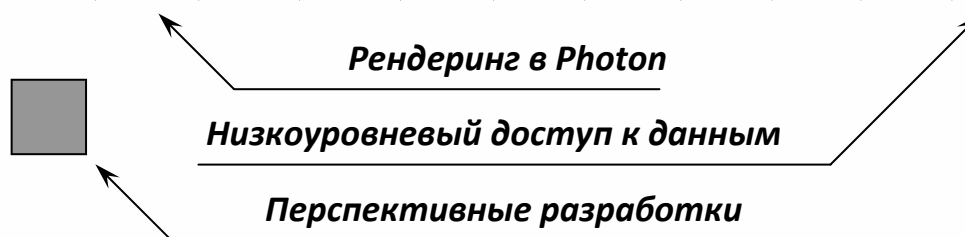
Средства визуализации



Структура и функциональность

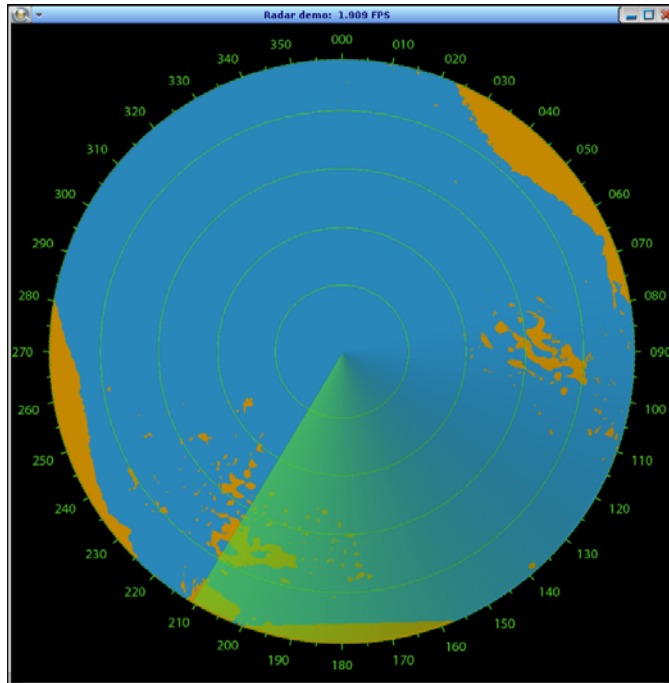
Функциональная совместимость

ОСРВ	S-57	S-63	SXF										
ЗОСРВ «Нейтрино»	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ЗОСРВ «Нейтрино-Э»	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ЗОСРВ КПДА.00002-01	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓
QNX 6.6	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓
QNX 6.5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
QNX 4.25	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓



- **Картографический пакет для ЗОСРВ «Нейтрино» / QNX**
 - Структура и функциональность
 - **Пример использования API**
 - Дальнейшее развитие проекта

Обзор примера



S-57    

Особенности приложения:

- запрос картографической информации в формате S-57;
- низкоуровневый доступ к географическим данным;
- экспорт визуализированной информации в форматах графических файлов BMP, JPEG и PNG.

Установление соединения с сервисом

```
#include <gis/gis.h>
#include <gis/gisrender.h>
```

S-57

```
gis_core_connection_t      connection;
gis_core_map_descriptor_t  map;

gis_core_link_init( &connection );
if ( gis_core_link_connect( &connection ) != EOK )
    printf( "Connection failed\n" );
else
    printf( "Connected\n" );
```

Инициализация переменной

Установка соединения

Коды возврата int gis_core_link_connect():

- EOK – соединение установлено;
- ENOMEM – параметр не проинициализирован (см. gis_core_link_init());
- EINVAL – предыдущее соединение не разорвано;
- EPERM – соединение запрещено;
- ESRCH – ошибка соединения.

Формирование географического дескриптора

S-57

```
if ( gis_core_request_map_descriptor( &map, grid, x, y, w, h, s ) != EOK )  
    printf( "Descriptor initialization failed\n" );
```

Параметры `gis_core_request_map_descriptor()`:

- `map` – географический дескриптор;
- `grid` – координатная сетка (проекция/геоид: `GIS_CORE_MAP_GRID_FLAT/GIS_CORE_MAP_GRID_GEO`);
- `x/y` – долгота/широта центра окна обзора в градусах;
- `w/h` – ширина/высота окна обзора в пикселях;
- `s` – масштаб в метрах на пиксель.

Коды возврата `int gis_core_request_map_descriptor()`:

- `EOK` – дескриптор проинициализирован;
- `ENOMEM` – недостаточно памяти для хранения дополнительных структур;
- `EINVAL` – некорректные параметры.

Запрос картографической информации

S-57

```
if ( gis_core_request_map( &connection, &map ) != EOK )  
    printf( "Map receive request failed\n" );
```

*Запрос на получение карт
(не обязателен)*

```
if ( gis_core_request_map_draw( &connection, &map ) != EOK )  
    printf( "Map draw request failed\n" );
```

*Запрос на получение и декодирование карт в
буфер отображения*

Коды возврата `int gis_core_request_map()` и `int gis_core_request_map_draw()`:

- EOK – запрос выполнен успешно;
- EPIPE – соединение с сервисом не установлено;
- EINVAL – географический дескриптор не корректен (см. `gis_core_request_map_descriptor()`);
- ESRCH – ошибка соединения.

Присоединение буфера отображения



```
if ( gis_core_drawbuffer_attach( &connection ) != EOK )  
    printf( "Map draw buffer attach failed\n" );
```

*Присоединение буфера
отображения*

```
if ( gis_core_drawbuffer_check( &connection ) != EOK )  
    printf( "Invalid buffer: Call gis_core_drawbuffer_invalidate()\n" );
```

Проверка буфера на корректность

Коды возврата int gis_core_drawbuffer_attach():

- EOK – доступ к буферу предоставлен;
- EPIPE – соединение с сервисом не установлено;
- ESRCH – ошибка определения параметров буфера;
- EBADFD – буфер не существует (см. gis_core_request_map_draw());
- EFAULT – ошибка присоединения буфера отображения.

Создание локального буфера визуализации

```
gis_core_render_buffer_t      rbuffer;  
  
if ( gis_render_buffer_alloc( &map, mode, bpp, &rbuffer ) != EOK )  
    printf( "Rendering buffer allocation failed\n" );
```



Параметры `gis_render_buffer_alloc()`:

- `map` – географический дескриптор;
- `mode` – режим рендеринга (`GIS_RENDER_BUFFER_MODE_RAW` – низкоровневый доступ к данным);
- `bpp` – глубина цвета (сейчас доступно только 32 бита на пиксель - `GIS_RENDER_BUFFER_BPP_32`);
- `rbuffer` – указатель на дескриптор буфера визуализации.

Коды возврата `int gis_render_buffer_alloc()`:

- `EOK` – буфер создан;
- `ENOTSUP` – не поддерживаемый режим;
- `ENOMEM` – недостаточно памяти;
- `EINVAL` – некорректный географический дескриптор;
- `ESRCH` – ошибка при создании буфера.

Создание контекста визуализации для низкоуровневого доступа



```
gis_core_render_raw_context_t  ctx;
```

Создание контекста

```
gis_render_raw_context_alloc( &ctx, rbuffer );
```

```
gis_render_raw_context_bgcolor( ctx, 0xffC48900 );
```

```
gis_render_raw_clean_buffer( ctx );
```

Очищение буфера

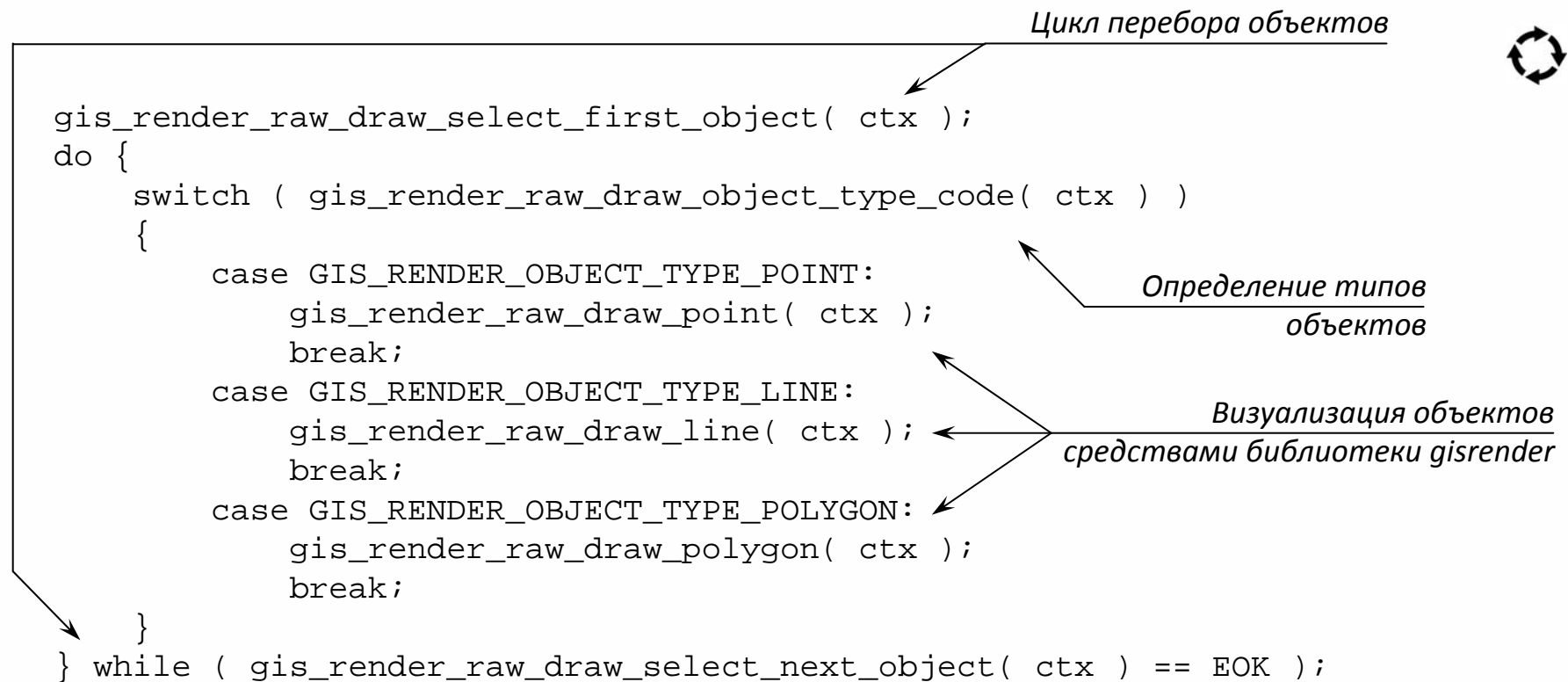
```
gis_render_raw_context_drawbegin( ctx );
```

...

```
gis_render_raw_context_drawend( ctx );
```

Блок визуализации

Визуализация картографической информации



Экспорт



```
gis_render_buffer_export( rbuffer, fname, variant );  
gis_render_buffer_export_ext( rbuffer_, fname, variant, clevel );
```

Параметры `gis_render_buffer_export()` и `gis_render_buffer_export_ext()`:

- `rbuffer` – буфер визуализации;
- `fname` – имя файла для экспорта;
- `variant` – режим экспорта (`GIS_RENDER_BUFFER_EXPORT_PNG`, `GIS_RENDER_BUFFER_EXPORT_JPEG`, `GIS_RENDER_BUFFER_EXPORT_BMP`);
- `clevel` – степень сжатия для режима `GIS_RENDER_BUFFER_EXPORT_JPEG` [0; 100].

Коды возврата `int` `gis_render_buffer_export()` и `int`

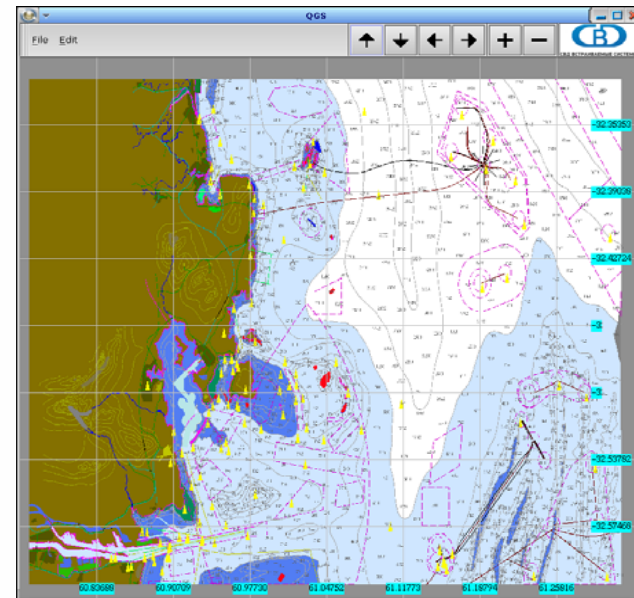
`gis_render_buffer_export_ext()` :

- `EOK` – экспорт успешно завершен;
- `EIO` – некорректный режим экспорта;
- `ENOTSUP` – неподдерживаемый вид буфера визуализации.

- **Картографический пакет для ЗОСРВ «Нейтрино» / QNX**
 - Структура и функциональность
 - Пример использования API
 - **Дальнейшее развитие проекта**

Дальнейшее развитие проекта

- **Расширение перечня встроенных механизмов визуализации:**
 - проекционная визуализация средствами OpenGL;
 - рельефная визуализация средствами OpenGL;
 - визуализация средствами библиотеки Qt;
 - визуализация в Photon;
- Развитие встроенной библиотеки шаблонов;
- Поддержка топографической информации;
- Добавление пользовательских данных;
- Экспорт в форматы XML, GIF, TIFF;
- Расширение числа поддерживаемых картографических серверов/баз данных/движков.



Спасибо за внимание

Алексей Девяткин
Инженер-программист

(812) 346-89-56 доп. 104
support@kpda.ru

www.kpda.ru

