



Case de uso:

Desenvolvimento de um Infotainment System com Qt WebEngine e Yocto

Luis Gustavo S. Barreto
<gustavosbarreto@gmail.com>



Sobre mim



Sumário

1. Introdução
2. Yocto
3. Qt WebEngine
4. Qt WebChannel
5. O Infotainment System
6. A Plataforma OWSTPlayer

Introdução

O case de uso consiste no desenvolvimento de um infotainment system utilizando Qt WebEngine e Yocto para ser embarcado em cabines de um navio cruzeiro da *Viking Cruises*.

Yocto

- Framework para compilação cruzada
- Código aberto (mas pode ser usado para compilação de código proprietário)
- Utilizado por fabricantes da indústria de dispositivo embarcados
- O que ele faz?
 - Download do código fonte
 - Aplicação de patches
 - Compilação cruzada
 - Gerenciamento de pacotes
- O que ele gera?
 - Pacotes binários
 - Imagens de sistema Linux
 - Toolchains
 - SDKs

Qt WebEngine

O Qt WebEngine é um módulo do Qt que provêm ferramentas para renderizar conteúdo HTML.

- Suporte a áudio e vídeo
- Drag & Drag
- HTML5 Geolocation
- HTTP/2
- Plugins
- WebRTC

Principais Classes

- QWebView
 - Renderiza documentos HTML e controla ações de navegação (back, forward, reload, stop)
- QWebEnginePage
 - Utilizada para controlar e obter informações da página do QWebView
 - Principais métodos:
 - load()
 - save()
 - setHtml()
 - runJavaScript()
 - title()
 - Provêm uma maneira de interceptar alerts, mensagens de console, requisições de novas janelas, requests de navegação, etc...

Utilizando o Qt WebEngine

Exemplo em C++

```
#include <QApplication>
#include <QWebEngineView>
#include <QUrl>

int main(int argc, char *argv[]) {
    QApplication app(argc, argv);

    QWebEngineView *view = new QWebEngineView();
    view->load(QUrl("https://br.qtcon.org"));
    view->show();

    return app.exec();
}
```

Exemplo em Qt Quick/QML

```
import QtQuick 2.0
import QtWebEngine 1.4

WebView {
    url: "https://br.qtcon.org"
}
```

QWebChannel

Permite a comunicação entre a aplicação C++/QML e a aplicação HTML por meio de uma biblioteca JavaScript.

- Possibilita chamar métodos do C++ no JavaScript
- Possibilita chamar métodos do JavaScript no C++
- Conecta em sinais de C++ no JavaScript

Exemplo QWebChannel - Aplicação C++

```
class Sensors: public QObject {
public slots:
    QString read(const QString &sensorName) {
        QFile file("/proc/sensors/" + sensorName);
        return file.readAll();
    }
}

QWebChannel *channel = new QWebChannel();
channel->registerObject("Sensors", new Sensors());

QWebEngineView *webView = new QWebEngineView();
webView->page()->setWebChannel(channel);
webView->show();
```

Exemplo QWebChannel - Página HTML

```
<script type="text/javascript" src="qrc:///qtwebchannel/qwebchannel.js"></script>
<script>
$( 'document' ).ready( function() {
    new QWebChannel( qt.webChannelTransport, function( channel ) {
        window.Sensors = channel.objects.Sensors;
    });
    Sensors.read( "temperatura", function( value ) {
        $( 'temperatura' ).val( value );
    });
})
</script>
```

Temperatura: <input id="temperatura" type="text"/>

○ Infotainment System

Visão Geral do Infotainment System



Interface gráfica do Infotainment system

Tecnologias

HTML



CSS



JS





54.5°F

7:54 PM

TUES, AUGUST 15 2017



MY CRUISE



ENTERTAINMENT



SHORE EXCURSIONS



THE SPA



DINING



VIKING DAILY



SHIP TRACKER



MY CALENDAR



CRUISE ITINERARY



ONBOARD ACCOUNT

MOVIES

COMEDY



DRAMA



ADVENTURE



SHIP TRACKER

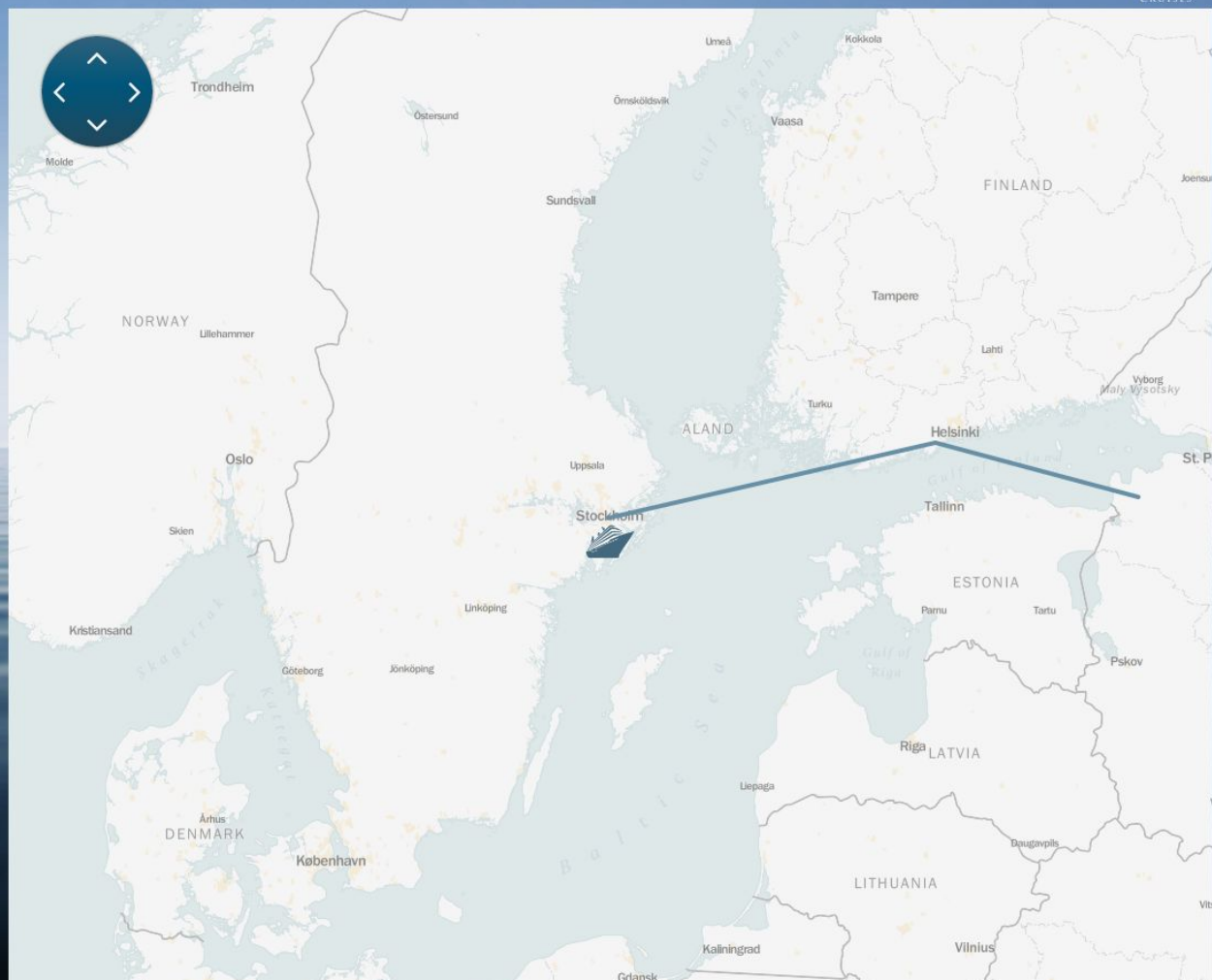
THE VIKING STAR

COORDINATES

59° 06' 03.6" N, 18° 06' 03.6" E

HEADING

99.32 degrees





00:16:02

00:16:02

02:08:15



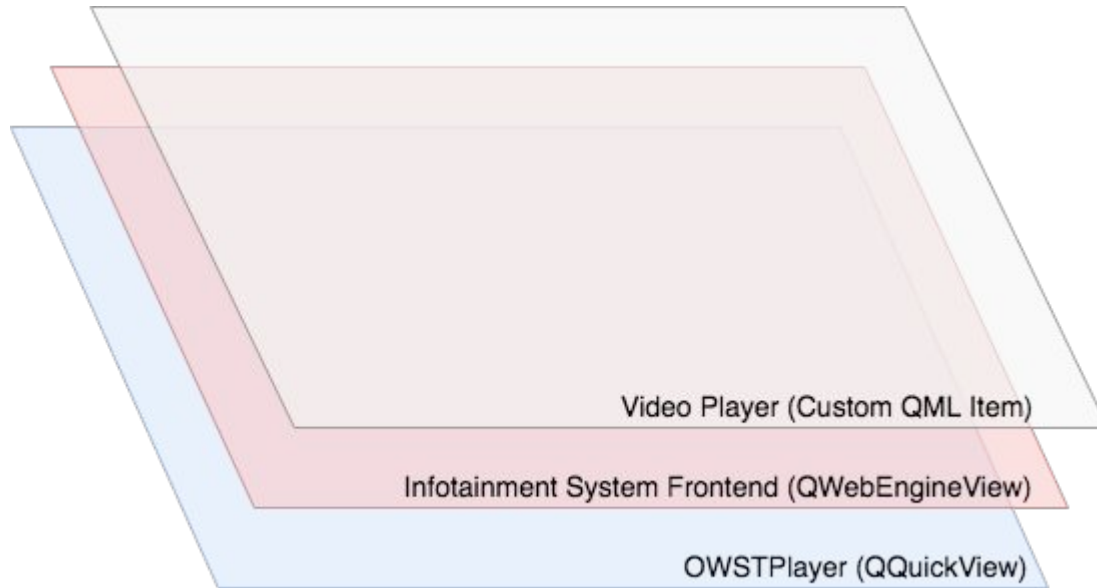
PAUSED

O.S. Systems WebEngine Stream Player

O.S. Systems WebEngine Stream Player ou simplesmente **OWSTPlayer**, é uma plataforma desenvolvida pela O.S. Systems que provêem um web browser com capacidade de reproduzir stream de vídeo.

- Core escrito em C++/Qt
- Interface gráfica do player de vídeo construída com QtQuick/Qml
- Disponibiliza uma API JavaScript para interação com o aplicativo de infotainment
- Disponibiliza um canal de comunicação via WebSocket para gerenciamento remoto

Camadas da interface gráfica



API JavaScript

API exposta através do [QtWebChannel](#)

Métodos:

- `Player.play()`
- `Player.pause()`
- `Player.hideDialog()`
- `Player.showDialog()`
- `Player.hideDialog()`
- `Player.volumeUp()`
- `Player.volumeDown()`
- `Player.mute()`
- `Player.unmute()`

Sinais:

- `Player.onReady()`
- `Player.onPlay()`
- `Player.onPause()`
- `Player.onVolumeChange()`
- `Player.onMute()`
- `Player.onPlaybackFinished()`
- `Player.onMessageReceived()`
- `Player.onStreamStatusChange()`

Mais informações

- Documentação oficial:

<http://doc.qt.io/qt-5/qtwebengine-index.html>

Obrigado!