



imotion

MediServer

Configuration

Technical Description

Version: 1.0
State: Released
Classification: Internal use only
Author: WES
Creation date: 2012-03-02
Repository: \$/Main/Common/Documents/TD_MediServerConfig.docx

Gorba AG
Sandackerstrasse
9245 Oberbüren
Switzerland

Table of contents

1	Introduction.....	4
2	Protran >1.0.15 and Infomedia 1.0.x.....	5
2.1	Protran and Infomedia on a single TFT.....	6
2.2	Master and Slave TFT using 2x Infomedia	7
2.3	Master and Multiple Slaves TFT all running Infomedia.....	8
2.4	Two-head operation	9

Modification management

Version	Date	Name	Dept.	Modifications	State
0.1	2012-03-02	WES	SW	Initial version	draft
0.2	2012-03-05	COS	SW	First review	draft

Review

Version	Date	Name	Dept.	Remarks
0.1	2012-03-05	COS	SW	First review.

Release

Version	Date	Name	Dept.	Remarks
1.0	2012-03-12	WES	SW	First release for version 1.0.1211

1 Introduction

MediServer is the main component for the Gorba Message Dispatching (Medi) protocol. It allows connecting different units (e.g. TFTs) with each other.

Currently the Medi protocol is only used to communicate between Protran and Infomedia, but its field of application is not limited to this.

MediServer can be configured manually via the command line or by simply placing a file called "medi.config" next to the MediServer.exe application.

This document shows possible usage scenarios and provides the necessary MediServer configurations for each unit.

2 Protran >1.0.15 and Infomedia 1.0.x

Protran 1.0.15 and up uses the Medi protocol to send Ximple messages to Infomedia. Since Infomedia 1.0.x doesn't understand Medi but only EventHandler, MediServer has to be configured to translate between the two protocols. To achieve this, on every unit (TFT) that is running Infomedia, you need a MediServer running which contains the local EventHandlerServer configuration (see below).

Infomedia should always be configured to connect to the local MediServer (127.0.0.1) on the EventHandler port 1598. Extract from infomedia.xml:

```
<DataPreferencesListItem>
  <Name>SocketAddress</Name>
  <Value>127.0.0.1</Value>
</DataPreferencesListItem>
<DataPreferencesListItem>
  <Name>SocketPort</Name>
  <Value>1598</Value>
</DataPreferencesListItem>
```

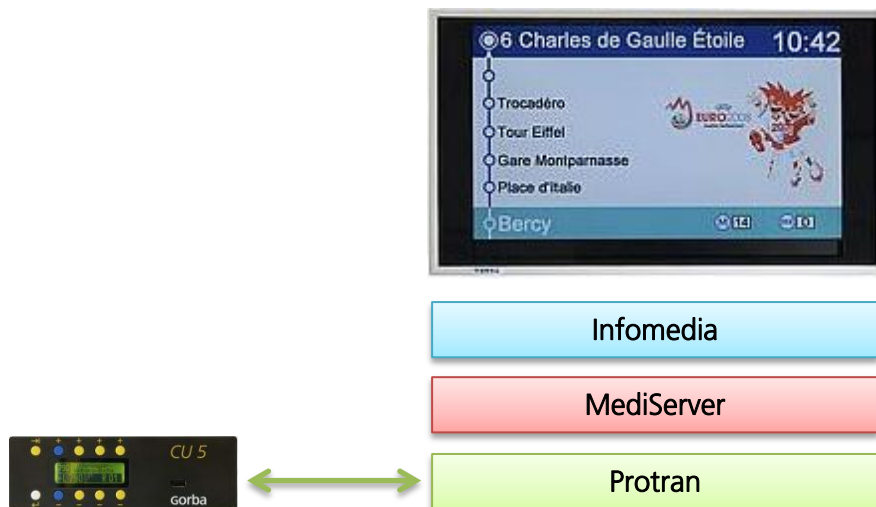
Protran should always be configured to connect to the local MediServer (127.0.0.1) on the Medi port 1596. Extract from cfg.xml:

```
<EventHandlers>
  <EventHandler description="...">
    <IsLocal description="Flag to localize an EventHandler server.">true</IsLocal>
    <IP description="The IP of the EventHandler's server.">127.0.0.1</IP>
    <Port description="The port of the EventHandler's server.">1596</Port>
  </EventHandler>
</EventHandlers>
```

It should be noted that the configuration of Protran is called "EventHandler", but this is actually the address and port of the MediServer, not the EventHandler.

2.1 Protran and Infomedia on a single TFT

This is the simplest case; you only have a single Topbox containing both Protran and Infomedia:

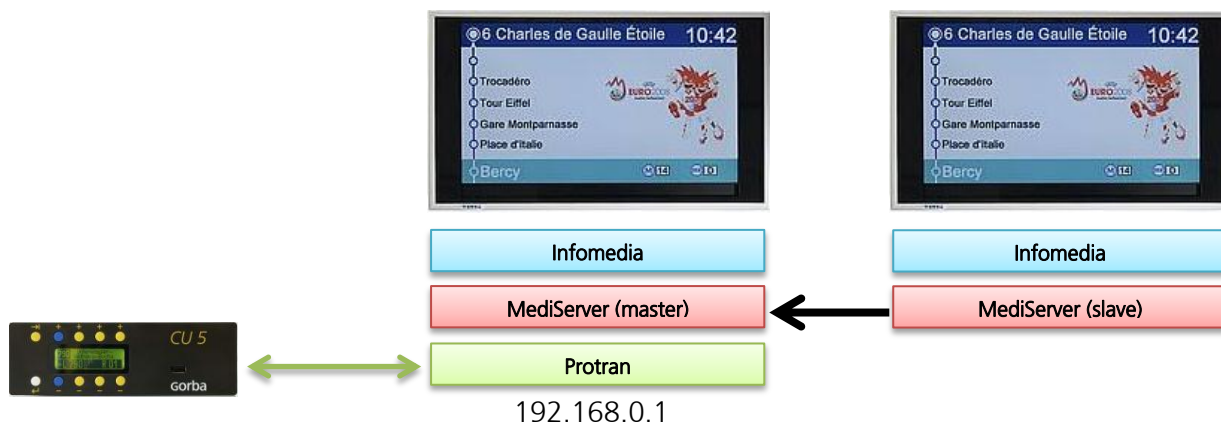


Medi.config:

```
<?xml version="1.0" encoding="utf-8"?>
<MediConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Peers>
    <PeerConfig xsi:type="ServerPeerConfig">
      <Codec xsi:type="XmlCodecConfig">
        <Encoding>utf-8</Encoding>
      </Codec>
      <Transport xsi:type="TcpTransportServerConfig">
        <LocalPort>1596</LocalPort>
      </Transport>
      <BlockBroadcast>false</BlockBroadcast>
    </PeerConfig>
    <PeerConfig xsi:type="EventHandlerPeerConfig">
      <LocalPort>1598</LocalPort>
      <SupportedMessages>
        <string>Gorba.Common.Protocols.Ximple.Ximple</string>
      </SupportedMessages>
    </PeerConfig>
  </Peers>
</MediConfig>
```

2.2 Master and Slave TFT using 2x Infomedia

In this case, you have a master TFT that runs Protran and Infomedia and a slave TFT that runs only Infomedia (both of course have a MediServer as mentioned above):



Master medi.config:

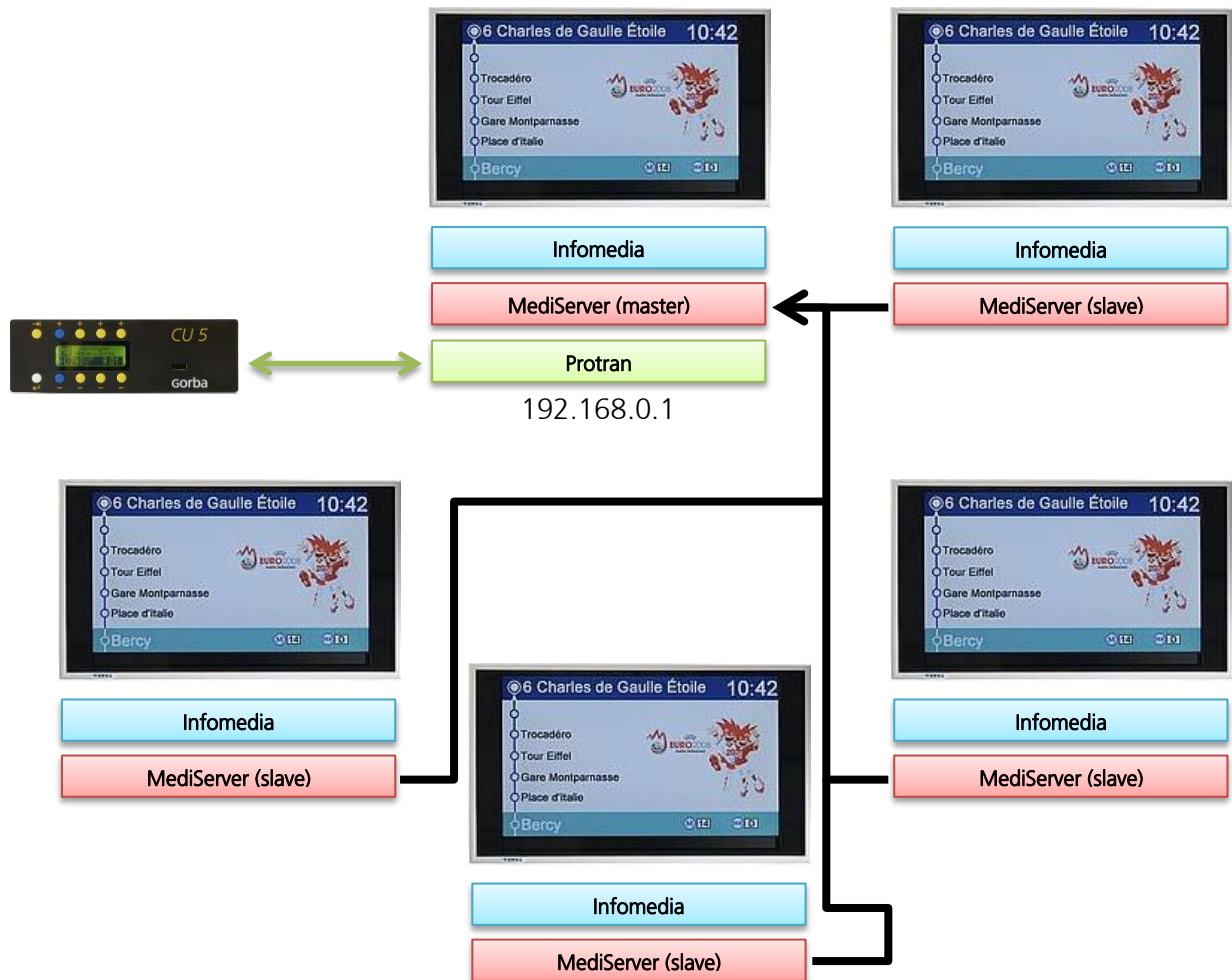
```
<?xml version="1.0" encoding="utf-8"?>
<MediConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Peers>
    <PeerConfig xsi:type="ServerPeerConfig">
      <Codec xsi:type="XmlCodecConfig">
        <Encoding>utf-8</Encoding>
      </Codec>
      <Transport xsi:type="TcpTransportServerConfig">
        <LocalPort>1596</LocalPort>
      </Transport>
      <BlockBroadcast>>false</BlockBroadcast>
    </PeerConfig>
    <PeerConfig xsi:type="EventHandlerPeerConfig">
      <LocalPort>1598</LocalPort>
      <SupportedMessages>
        <string>Gorba.Common.Protocols.Ximple.Ximple</string>
      </SupportedMessages>
    </PeerConfig>
  </Peers>
</MediConfig>
```

Slave medi.config (assuming the master TFT has the IP address 192.168.0.1):

```
<?xml version="1.0" encoding="utf-8"?>
<MediConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Peers>
    <PeerConfig xsi:type="ClientPeerConfig">
      <Codec xsi:type="XmlCodecConfig">
        <Encoding>utf-8</Encoding>
      </Codec>
      <Transport xsi:type="TcpTransportClientConfig">
        <RemoteIp>192.168.0.1</RemoteIp>
        <RemotePort>1596</RemotePort>
      </Transport>
      <BlockBroadcast>>false</BlockBroadcast>
    </PeerConfig>
    <PeerConfig xsi:type="EventHandlerPeerConfig">
      <LocalPort>1598</LocalPort>
      <SupportedMessages>
        <string>Gorba.Common.Protocols.Ximple.Ximple</string>
      </SupportedMessages>
    </PeerConfig>
  </Peers>
</MediConfig>
```

2.3 Master and Multiple Slaves TFT all running Infomedia

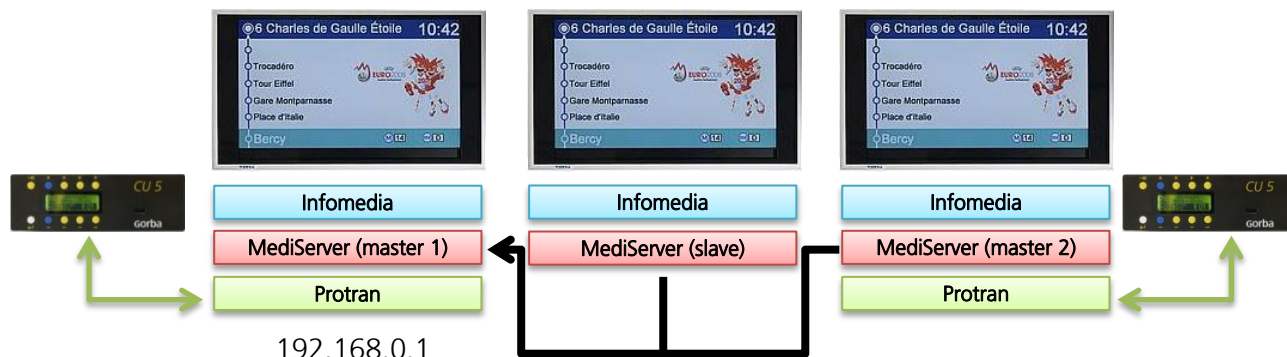
This is exactly the same case as in chapter 2.2, but with multiple slaves:



In this scenario, the **master** is configured exactly like the master in chapter 2.2, and all **slaves** are configured exactly the same as the slave in chapter 2.2.

2.4 Two-head operation

This scenario is used in trains or trams that can drive in two directions where both heads have a board computer and/or CU. Of course it has to be ensured that only one of the two board computers is sending information to Protran, but this is outside the scope of this document.



In this scenario, we still connect each Protran to its local MediServer, but there are now two masters to be configured. The **master 1** is configured exactly like the master in chapter 2.2, and all **slaves** are configured exactly the same as the slave in chapter 2.2. It is suggested that the motor car (Triebwagen) is used as master 1 and the control car (Steuernwagen) as master 2.

At least the IP address of the master 1 has to be statically assigned, otherwise this scenario doesn't work. Currently there is no solution to setups where all IP addresses are assigned dynamically (DHCP).

Only **master 2** is different since it needs a port for Protran as well as a connection to master 1.

Master 2 medi.config:

```
<?xml version="1.0" encoding="utf-8"?>
<MediConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Peers>
    <PeerConfig xsi:type="ServerPeerConfig">
      <Codec xsi:type="XmlCodecConfig">
        <Encoding>utf-8</Encoding>
      </Codec>
      <Transport xsi:type="TcpTransportServerConfig">
        <LocalPort>1596</LocalPort>
      </Transport>
      <BlockBroadcast>>false</BlockBroadcast>
    </PeerConfig>
    <PeerConfig xsi:type="ClientPeerConfig">
      <Codec xsi:type="XmlCodecConfig">
        <Encoding>utf-8</Encoding>
      </Codec>
      <Transport xsi:type="TcpTransportClientConfig">
        <RemoteIp>192.168.0.1</RemoteIp>
        <RemotePort>1596</RemotePort>
      </Transport>
      <BlockBroadcast>>false</BlockBroadcast>
    </PeerConfig>
    <PeerConfig xsi:type="EventHandlerPeerConfig">
      <LocalPort>1598</LocalPort>
      <SupportedMessages>
        <string>Gorba.Common.Protocols.Ximple.Ximple</string>
      </SupportedMessages>
    </PeerConfig>
  </Peers>
</MediConfig>
```