# Concepts for remote control of VLBI-telescopes and for the Integration of new VLBI2010 Devices into the Field System

FESG

**Alexander Neidhardt (FESG)**
**neidhardt@fs.wettzell.de**

Max-Planck-Institut für Radioastronomie

Bundesamt für Kartographie und Geodäsie

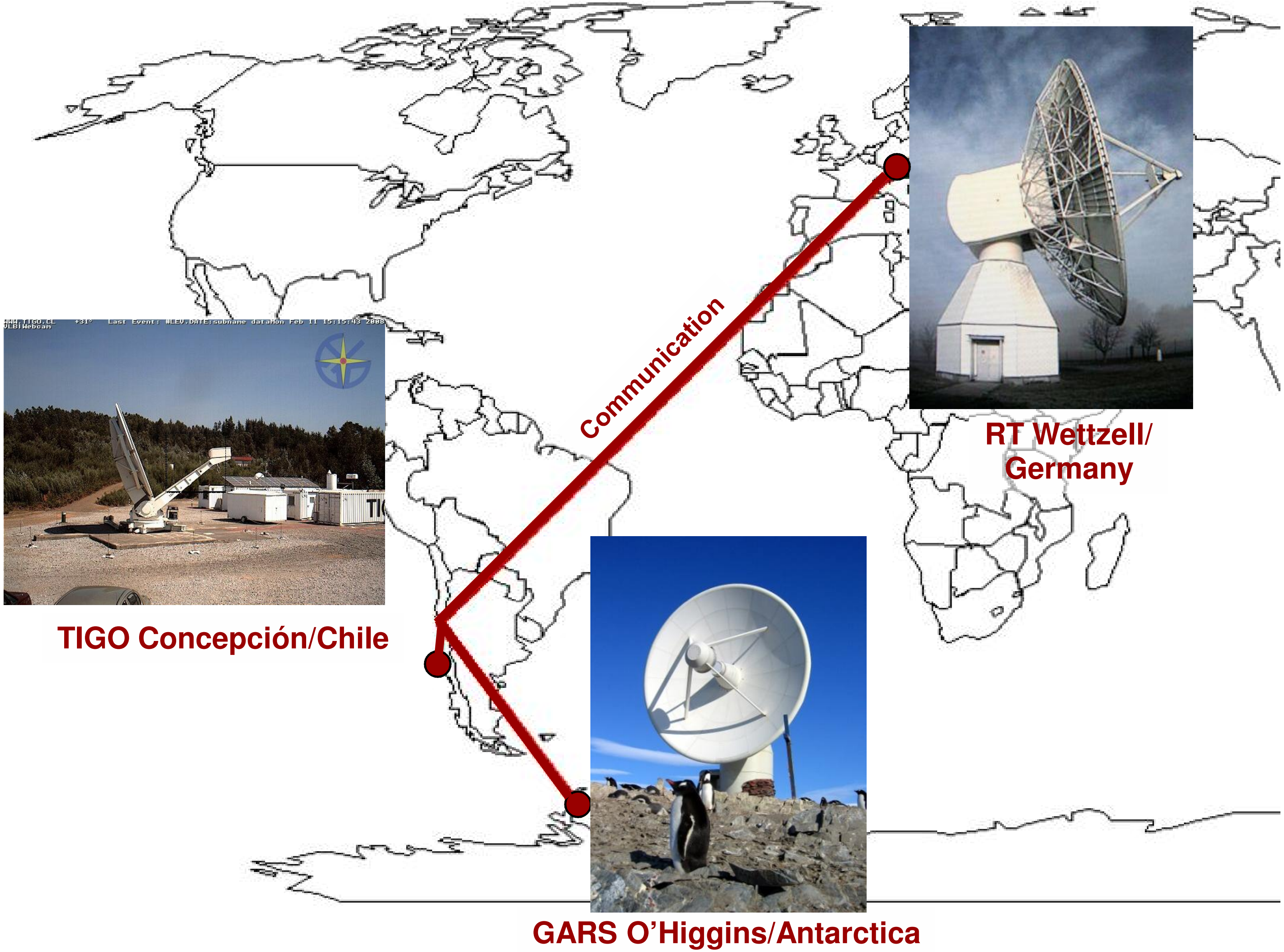
GODDARD SPACE FLIGHT CENTER

Universidad de Concepción
CHILE

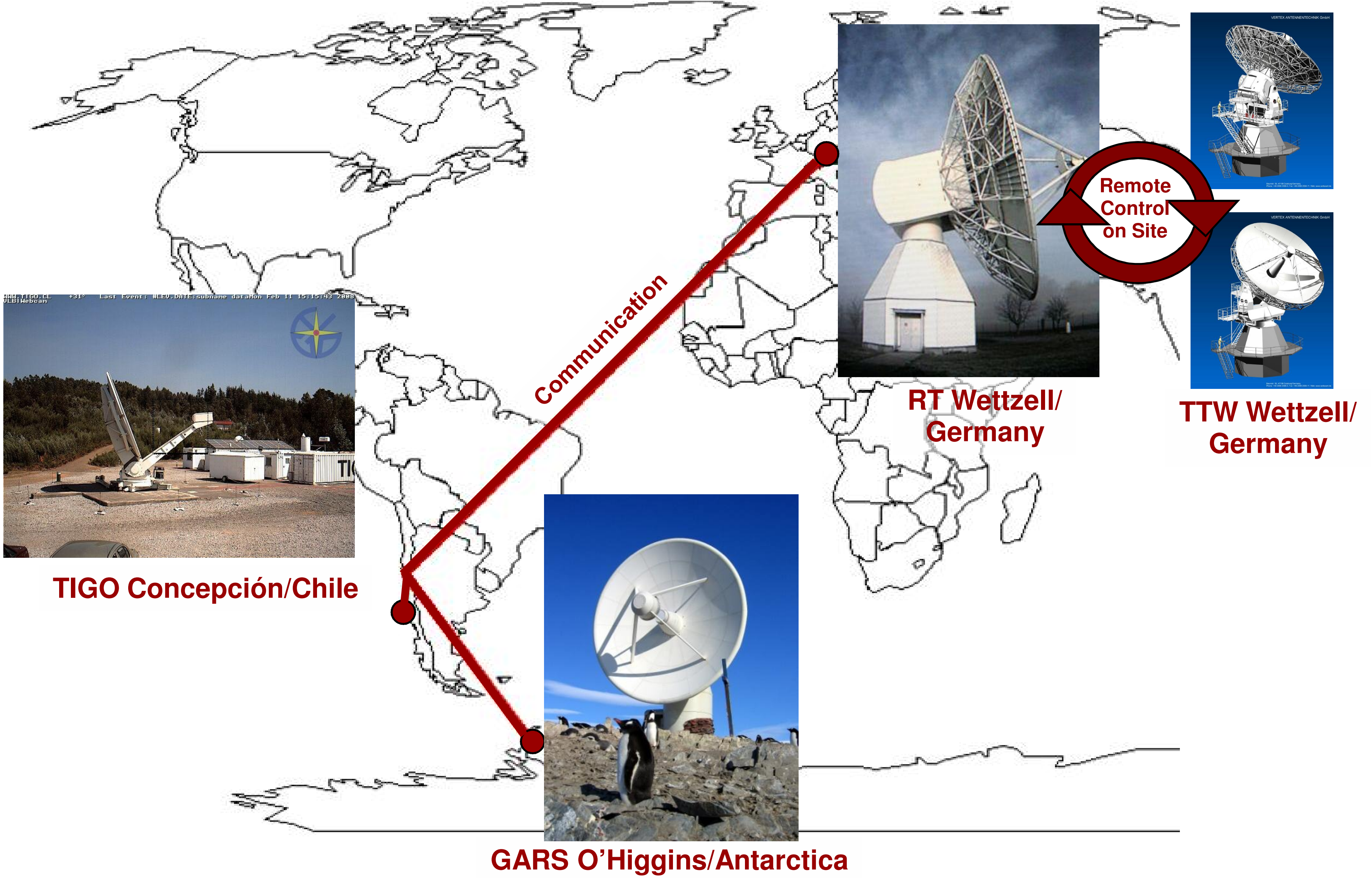
**Martin Ettl (FESG),**
**Reiner Dassing (BKG), Hayo Hase (BKG), Matthias Mühlbauer (BKG), Christian Plötz (BKG),**
**Sergio Sobarzo (UdeC), Cristian Herrera (UdeC),**
**Walter Alef (MPIfR), Helge Rottmann (MPIfR),**
**Ed Himwich (NASA/GSFC/NVI)**

# Wettzell and the idea of controlling VLBI telescopes by remote

# The idea: remote attendance and control of VLBI telescopes Wettzell, O'Higgins/Antarctica and TIGO/Concepción


WWW.TIGO.CL +31° Last Event: #LEV.DATE:subname dataMon Feb 11 15:15:43 2008
VLBIWebcam
Communication
RT Wettzell/
Germany
TIGO Concepción/Chile
GARS O'Higgins/Antarctica

# The idea: remote attendance and control of VLBI telescopes Wettzell, O'Higgins/Antarctica and TIGO/Concepción


WWW.TIGO.CL +31° Last Event: #LEV.DATE:subname dataMon Feb 11 15:15:43 2008
VLBIWebcam

Communication

Remote Control on Site

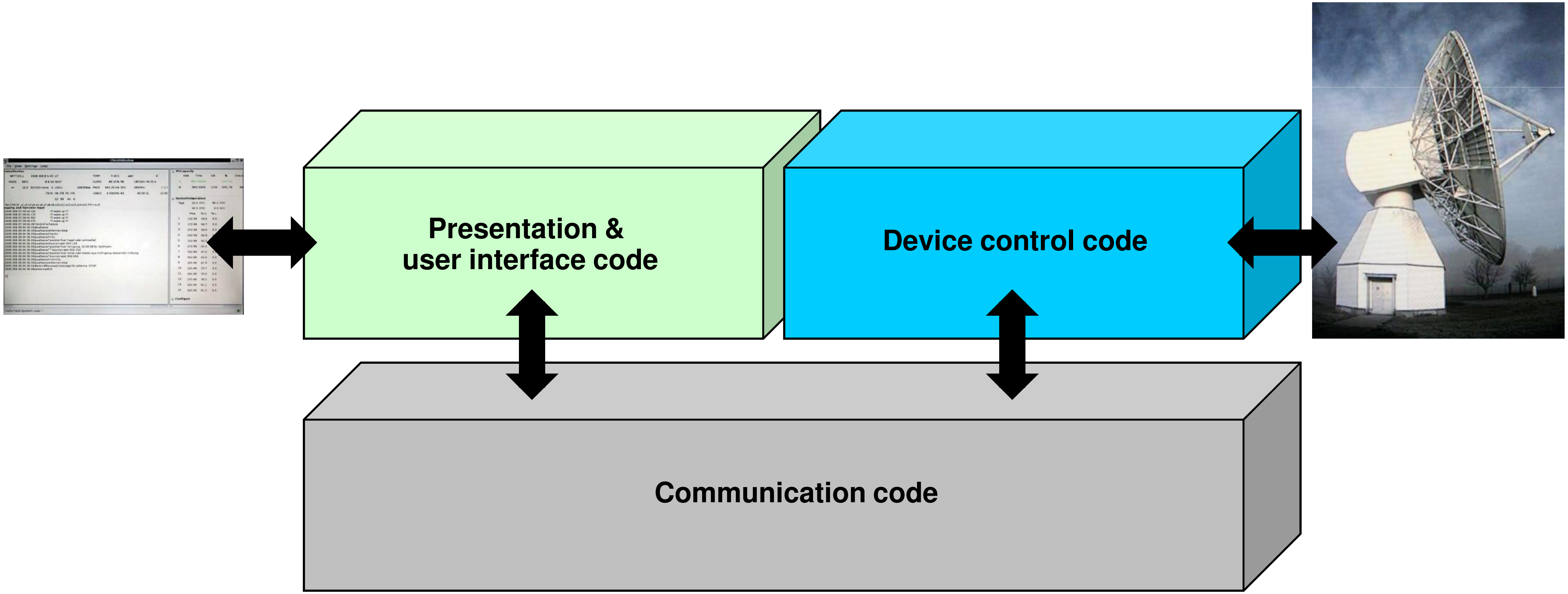
RT Wettzell/ Germany

TTW Wettzell/ Germany

TIGO Concepción/Chile

GARS O'Higgins/Antarctica

# The idea: remote attendance and control of VLBI telescopes Wettzell, O’Higgins/Antarctica and TIGO/Concepción



Presentation & user interface code

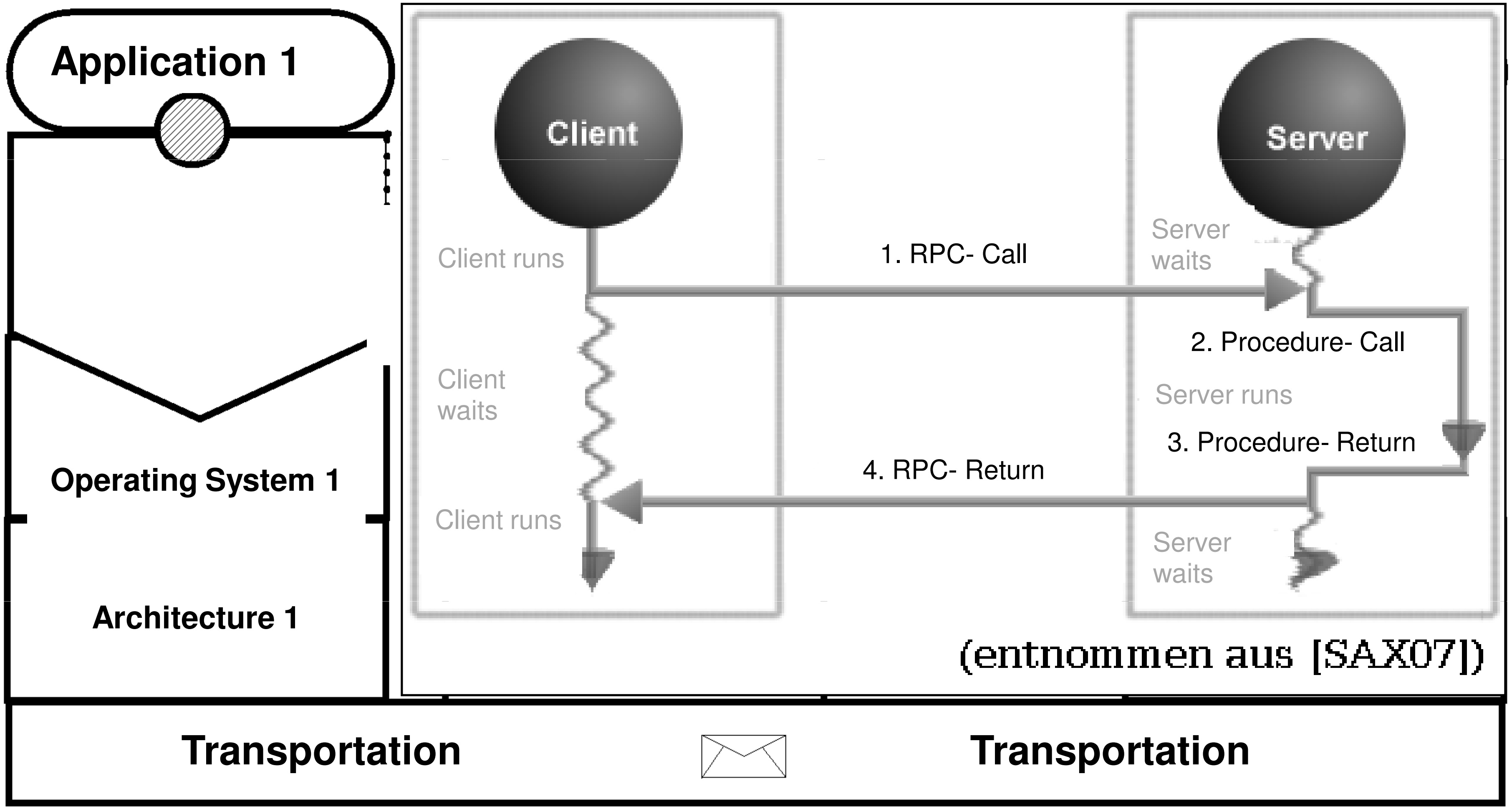
Device control code

Communication code

**Idea of a strict design-separation of these parts**

# The communication – with a remote procedure call middleware and ssh

# The communication – with a remote procedure call middleware

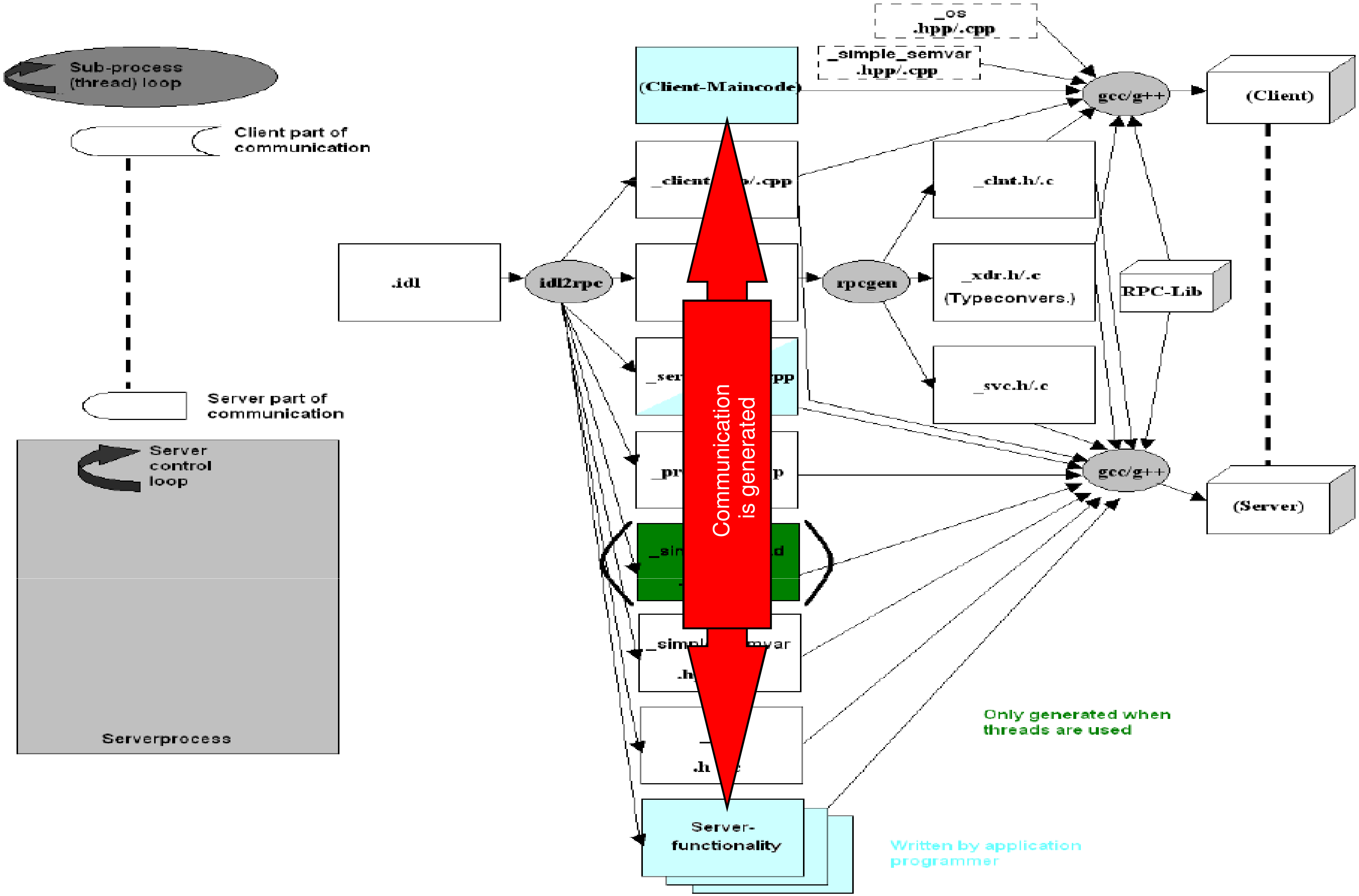


(entnommen aus [SAX07])

(nach [PUD01] a.a.O. S. 25)

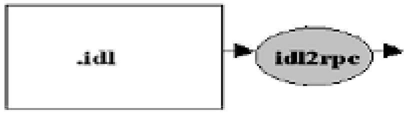
[SAX07]: Saxonia Systems: Remote Procedure Call, http://www.linuxfibel.de/rpc.htm, Download 23.04.2007
[PUD01]: Puder, Arno; Römer, Kay: Middleware für vereteilte Systeme, 1.Auflage, dpunkt.verlag GmbH Heidelberg 2001

# The communication – using a middleware generator



# The communication – using a middleware generator

## Step 1: a simple interface definition for fieldsystem monitoring



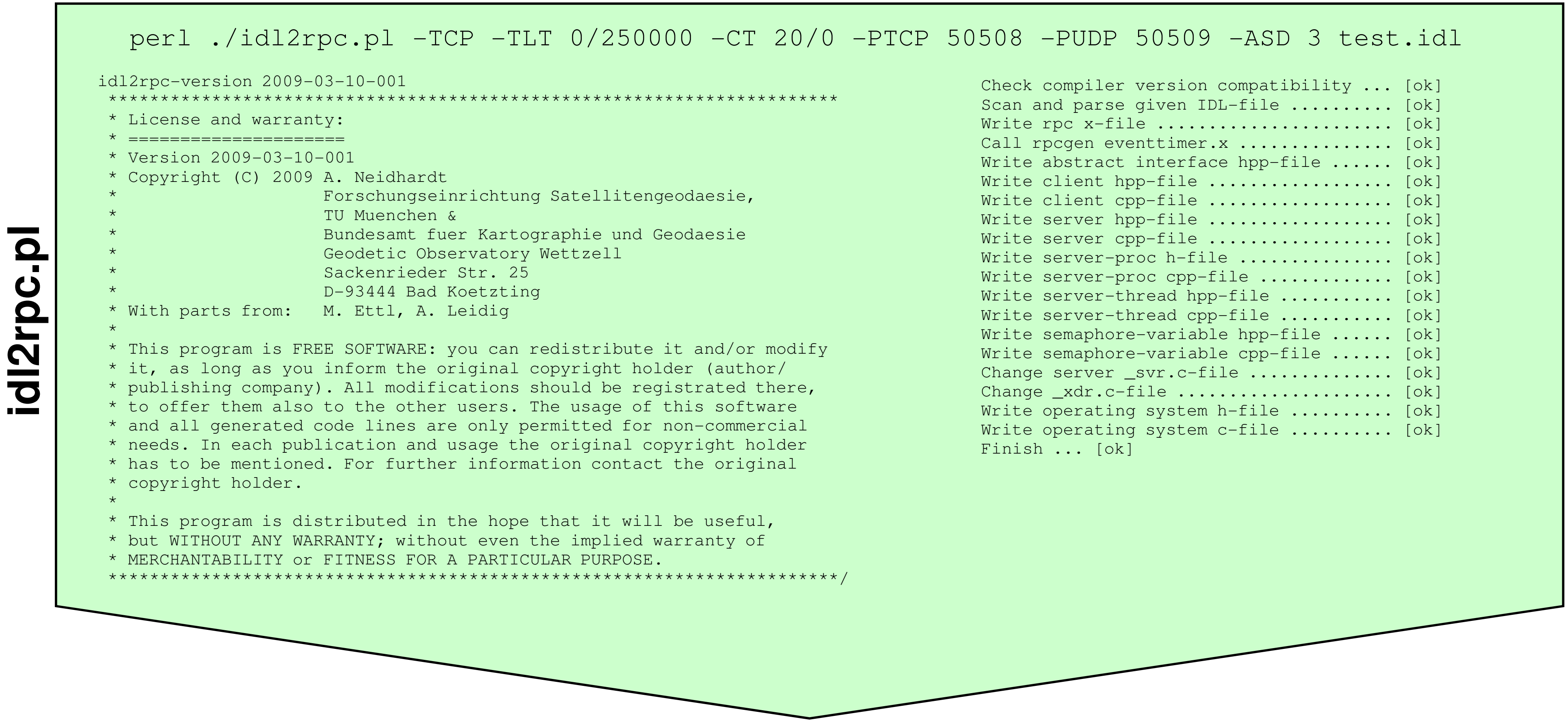
**fsmc.idl**

```
...
interface fsmc
{
    void vReset();
    // Monitoring methods
    unsigned int uiGetSystemStatusMonitorText (out string strStatusTags <>);
    unsigned int uiGetSystemTemperatureText (out string strTempTags <>);
    unsigned int uiGetSystemMark5Text (out string strMark5Tags <>);
    unsigned int uiSetFSCommand (in string strCommandTags);
    unsigned int uiGetFSLogFile (in unsigned long ulLogDescriptor,
                                 out string strLogText,
                                 out string strAdditionalLogText);
    unsigned int uiGetSystemOverallStateText (out string strStatusTags <>,
                                              out string strTempTags <>,
                                              out string strMark5Tags <> ,
                                              in unsigned long ulLogDescriptor,
                                              out string strLogText,
                                              out string strAdditionalLogText);
};
```

# The communication – using a middleware generator

## Step 2: create the communication moduls in C++ for fieldsystem monitoring

**fsmc.idl**



```
perl ./idl2rpc.pl -TCP -TLT 0/250000 -CT 20/0 -PTCP 50508 -PUDP 50509 -ASD 3 test.idl
```

```
idl2rpc-version 2009-03-10-001
 ***********************************************************************
 * License and warranty:
 * =====================
 * Version 2009-03-10-001
 * Copyright (C) 2009 A. Neidhardt
 *                    Forschungseinrichtung Satellitengeodaesie,
 *                    TU Muenchen &
 *                    Bundesamt fuer Kartographie und Geodaesie
 *                    Geodetic Observatory Wettzell
 *                    Sackenrieder Str. 25
 *                    D-93444 Bad Koetzting
 * With parts from:   M. Ettl, A. Leidig
 *
 * This program is FREE SOFTWARE: you can redistribute it and/or modify
 * it, as long as you inform the original copyright holder (author/
 * publishing company). All modifications should be registrated there,
 * to offer them also to the other users. The usage of this software
 * and all generated code lines are only permitted for non-commercial
 * needs. In each publication and usage the original copyright holder
 * has to be mentioned. For further information contact the original
 * copyright holder.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
 ***********************************************************************/
```

```
Check compiler version compatibility ... [ok]
Scan and parse given IDL-file .......... [ok]
Write rpc x-file ....................... [ok]
Call rpcgen eventtimer.x ............... [ok]
Write abstract interface hpp-file ...... [ok]
Write client hpp-file .................. [ok]
Write client cpp-file .................. [ok]
Write server hpp-file .................. [ok]
Write server cpp-file .................. [ok]
Write server-proc h-file ............... [ok]
Write server-proc cpp-file ............. [ok]
Write server-thread hpp-file ........... [ok]
Write server-thread cpp-file ........... [ok]
Write semaphore-variable hpp-file ...... [ok]
Write semaphore-variable cpp-file ...... [ok]
Change server _svr.c-file .............. [ok]
Change _xdr.c-file ..................... [ok]
Write operating system h-file .......... [ok]
Write operating system c-file .......... [ok]
Finish ... [ok]
```

**fsmc.h fsmc_client.cpp fsmc_interface.hpp fsmc_server.cpp fsmc_simple_semvar.hpp fsmc_svc.c fsmc.idl fsmc_client.hpp fsmc_proc.cpp fsmc_server.hpp fsmc_simple_thread.cpp fsmc_xdr.c fsmc.x fsmc_clnt.c fsmc_proc.hpp fsmc_simple_semvar.cpp fsmc_simple_thread.hpp**

# The communication – using a middleware generator

## Step 3: write the remote activity (fsmc_server.cpp/.hpp)

**fsmc_server.hpp**

```
...
#ifndef __fsmc_server__
#define __fsmc_server__

#include <rpc/rpc.h>
#include "fsmc_interface.hpp"
#include "fsmc_simple_semvar.hpp"
// USERDEFINCLUDEBEG: Userdefined includes

// USERDEFINCLUDEEND

class fsmc_server : fsmc
{
...
    // USERDEFATTRIBBEG: Userdefined attributes
    unsigned int uiFatalError;
    std::string strLogFilepath;
    fsmc_semvar<std::string *> pstrSystemStatusMonitorTextValues;
    fsmc_semvar<std::string *> pstrSystemTemperaturesTextValues;
    fsmc_semvar<std::string *> pstrSystemMark5TextValues;
    fsmc_semvar<unsigned int> uiError;
    // USERDEFATTRIBEND
    // USERDEFATTMETHODBEG: Userdefined attribute methodes

    // USERDEFATTMETHODEND
...
```

**fsmc_server.cpp**

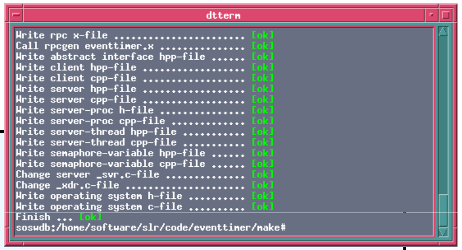
```
...
/*****************************************************
 *   class      fsmc_server
 *   function   uiSetFSCommand
 *****************************************************
/*!             Generated interface methode. See interfa
                (defined by user)
 *****************************************************
/*   author     Alexander Neidhardt
 *   date       14.05.2007
 *   revision   -
 *   info       Part of the idl2rpc.pl - generator!
 *****************************************************
unsigned int fsmc_server::uiSetFSCommand (const std::s
throw (_interface_throw)
{
// USERDEFMETHODBEG: Userdefined methode body
    std::string strCommandPath("/usr2/fs/bin/inject_sn
    std::string FSCommand;
    FSCommand = strCommandPath + "'"+strCommandTags+"'
    return(system(FSCommand.c_str()));

// USERDEFMETHODEND
}
...
```

# The communication – using a middleware generator

## Step 4: write a client (main)

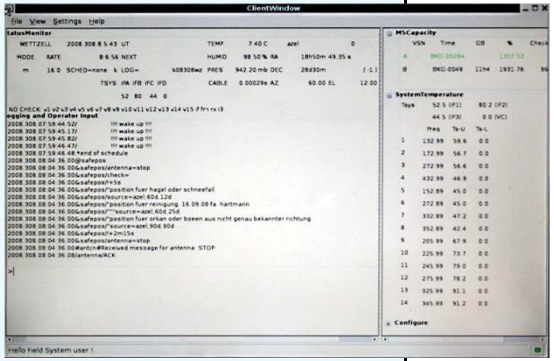
### client.cpp

```
#include <iostream>
#include "fsmc_client.hpp"

int main ()
{
    fsmc_client CClient;

    std::cout << "Open" << std::endl;
    try
    {
         if (CClient._usOpenInterface ("127.0.0.1"))
         {
             std::cout << "ERROR open" << std::endl;
             return 1;
         }

    }
    catch (...)
    {
         std::cout << "ERROR catch open" << std::endl;
         return 1;
    }
    std::cout << "Reset" << std::endl;
    try
    {
        CClient.vReset ();
    }
    catch (...)
    {
         std::cout << "ERROR catch reset" << std::endl;
    }
...
```



```
Write rpc x-file ........................ [ok]
Call rpcgen eventtimer.x ................ [ok]
Write abstract interface hpp-file ...... [ok]
Write client hpp-file .................. [ok]
Write client cpp-file .................. [ok]
Write server hpp-file .................. [ok]
Write server cpp-file .................. [ok]
Write server-proc h-file ............... [ok]
Write server-proc cpp-file ............. [ok]
Write server-thread hpp-file ........... [ok]
Write server-thread cpp-file ........... [ok]
Write semaphore-variable hpp-file ...... [ok]
Write semaphore-variable cpp-file ...... [ok]
Change server _svr.c-file .............. [ok]
Change _xdr.c-file ..................... [ok]
Write operating system h-file .......... [ok]
Write operating system c-file .......... [ok]
Finish ... [ok]
soswdb:/home/software/slr/code/eventtimer/make#
```

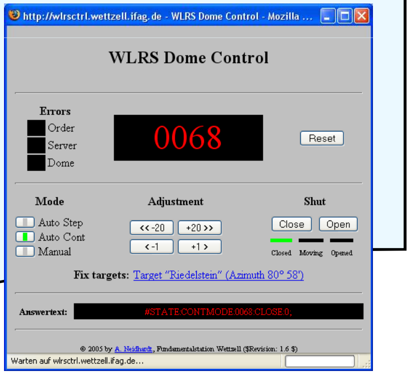
Command line shell



```
WETTZELL   2008 308 8 5 43  UT              TEMP    7.40 C      azel          0
MODE   RATE        8 6 56 NEXT              HUMID   98.50 % RA  18h50m 49.35 s
 m     16.0  SCHED=none  k LOG=    k08308wz PRES    942.20 mb DEC  28d30m   (-1.)
                 TSYS  IFA IFB IFC IFD      CABLE   0.00029s AZ   60.00 EL  12.00
                        52  80  44  0
NO CHECK: v1 v2 v3 v4 v5 v6 v7 v8 v9 v10 v11 v12 v13 v14 v15 if fr1 rx i3
ogging and Operator Input
2008.308.07:59:44.52/          !!!! wake up !!!!
2008.308.07:59:45.17/          !!!! wake up !!!!
2008.308.07:59:45.82/          !!!! wake up !!!!
2008.308.07:59:46.47/          !!!! wake up !!!!
2008.308.07:59:46.48 *end of schedule
2008.308.08:04:36.00@safepos
2008.308.08:04:36.00&safepos/antenna=stop
2008.308.08:04:36.00&safepos/check=
2008.308.08:04:36.00&safepos/!+5s
2008.308.08:04:36.00&safepos/"position fuer hagel oder schneefall
2008.308.08:04:36.00&safepos/source=azel,60d,12d
2008.308.08:04:36.00&safepos/"position fuer reinigung, 16.09.08 fa. hartmann
2008.308.08:04:36.00&safepos/"""source=azel,60d,25d
2008.308.08:04:36.00&safepos/"position fuer orkan oder boeen aus nicht genau bekannter richtung
2008.308.08:04:36.00&safepos/"source=azel,90d,90d
2008.308.08:04:36.00&safepos/!+2m15s
2008.308.08:04:36.00&safepos/antenna=stop
2008.308.08:04:36.00#antcn#Received message for antenna: STOP
2008.308.08:04:36.08/antenna/ACK

>|

Hello Field System user !
```

```
VSN          Time    GB        %     Check
A  BKG-0028+         1357.52
B  BKG-0049  11h4    1931.76         96

Tsys   52.5 (IF1)   80.2 (IF2)
       44.5 (IF3)    0.0 (VC)
     Freq     Ts-U   Ts-L
 1   132.99   59.6   0.0
 2   172.99   56.7   0.0
 3   272.99   56.6   0.0
 4   432.99   46.9   0.0
 5   152.89   45.0   0.0
 6   272.89   45.0   0.0
 7   332.89   47.2   0.0
 8   352.89   42.4   0.0
 9   205.99   67.9   0.0
10   225.99   73.7   0.0
11   245.99   79.0   0.0
12   275.99   78.2   0.0
13   325.99   91.1   0.0
14   345.99   91.2   0.0
```

Graphical User Interface (GUI)



http://wlrsctrl.wettzell.ifag.de - WLRS Dome Control - Mozilla ...

WLRS Dome Control

Fix targets: Target "Riedelstein" (Azimuth 80° 58')

Answertext: #STATE:CONTMODE:0068:CLOSE:0;

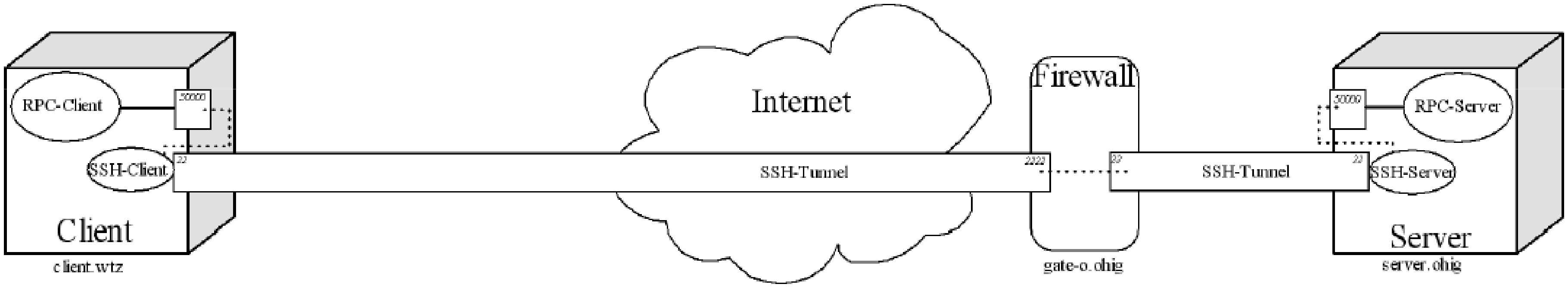
© 2005 by A. Neidhardt, Fundamentalstation Wettzell ($Revision: 1.6 $)

Warten auf wlrsctrl.wettzell.ifag.de...

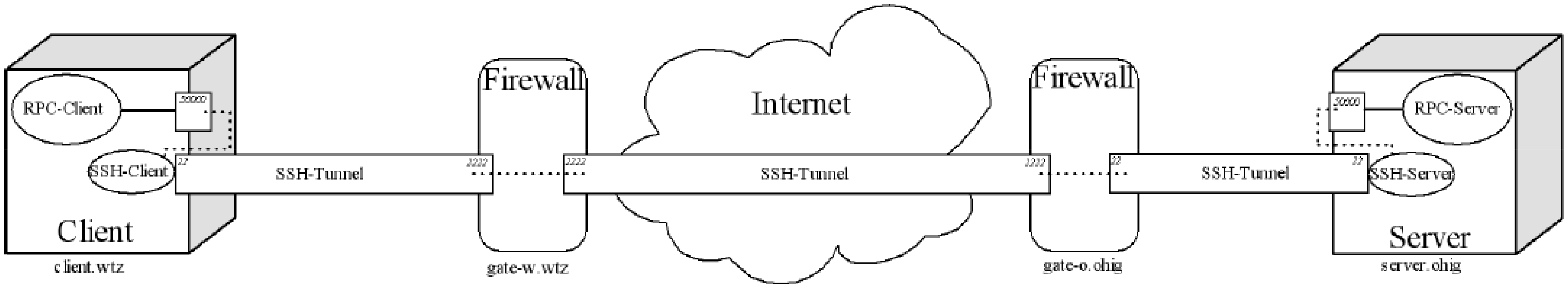
Web Interface

**& Compiling**
**=> Finish!**

# The communication – ssh - tunneling



```
ssh -l <user>
    -p 2222
    -L 50000:127.0.0.1:50000
    gate-o.ohig
```

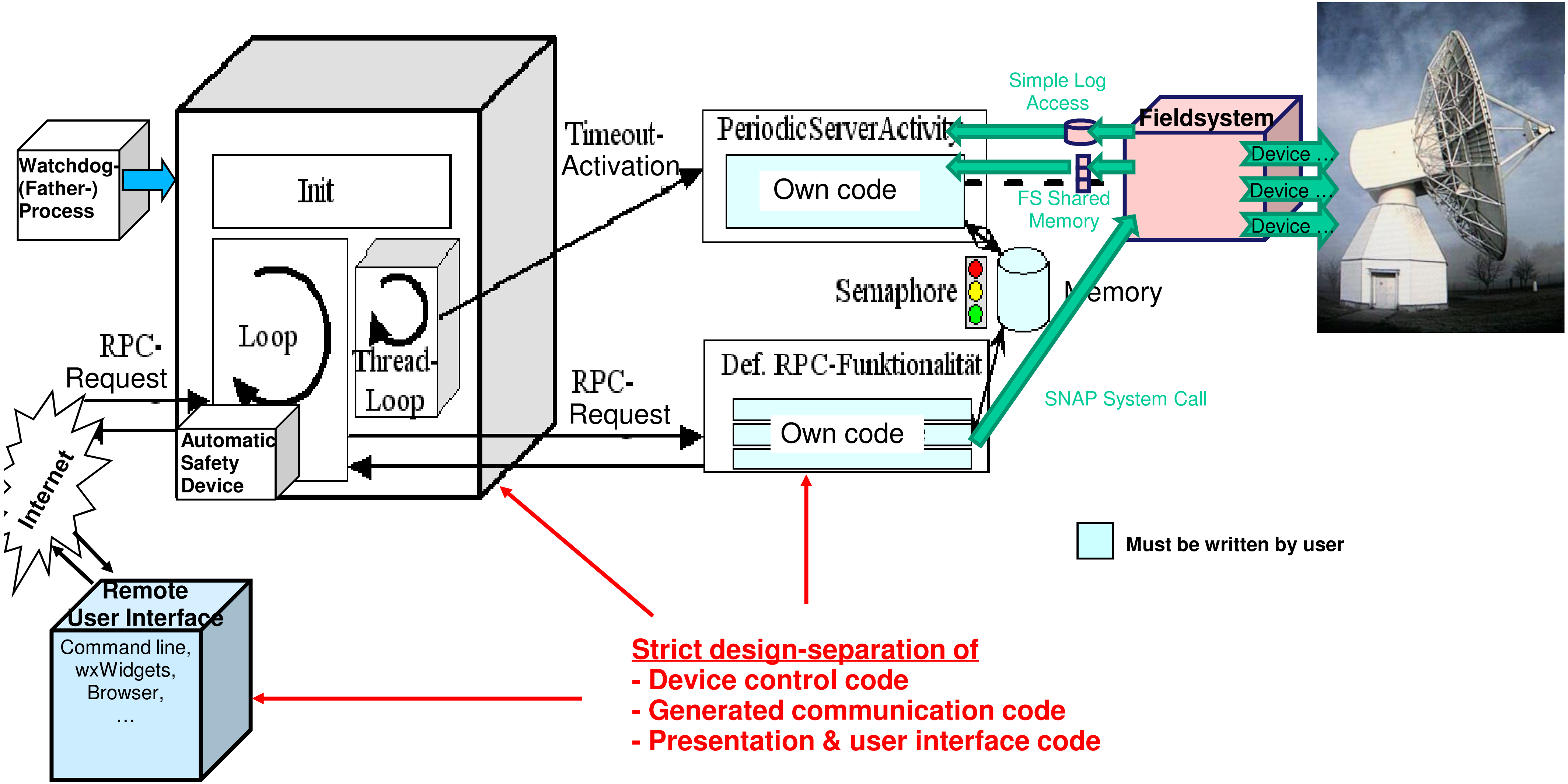


```
ssh -l <user>
    -p 2222
    -L 50000:127.0.0.1:50000
    gate-w.ohig
```

# A fieldsystem extension – remote accessible, autonomous process cells

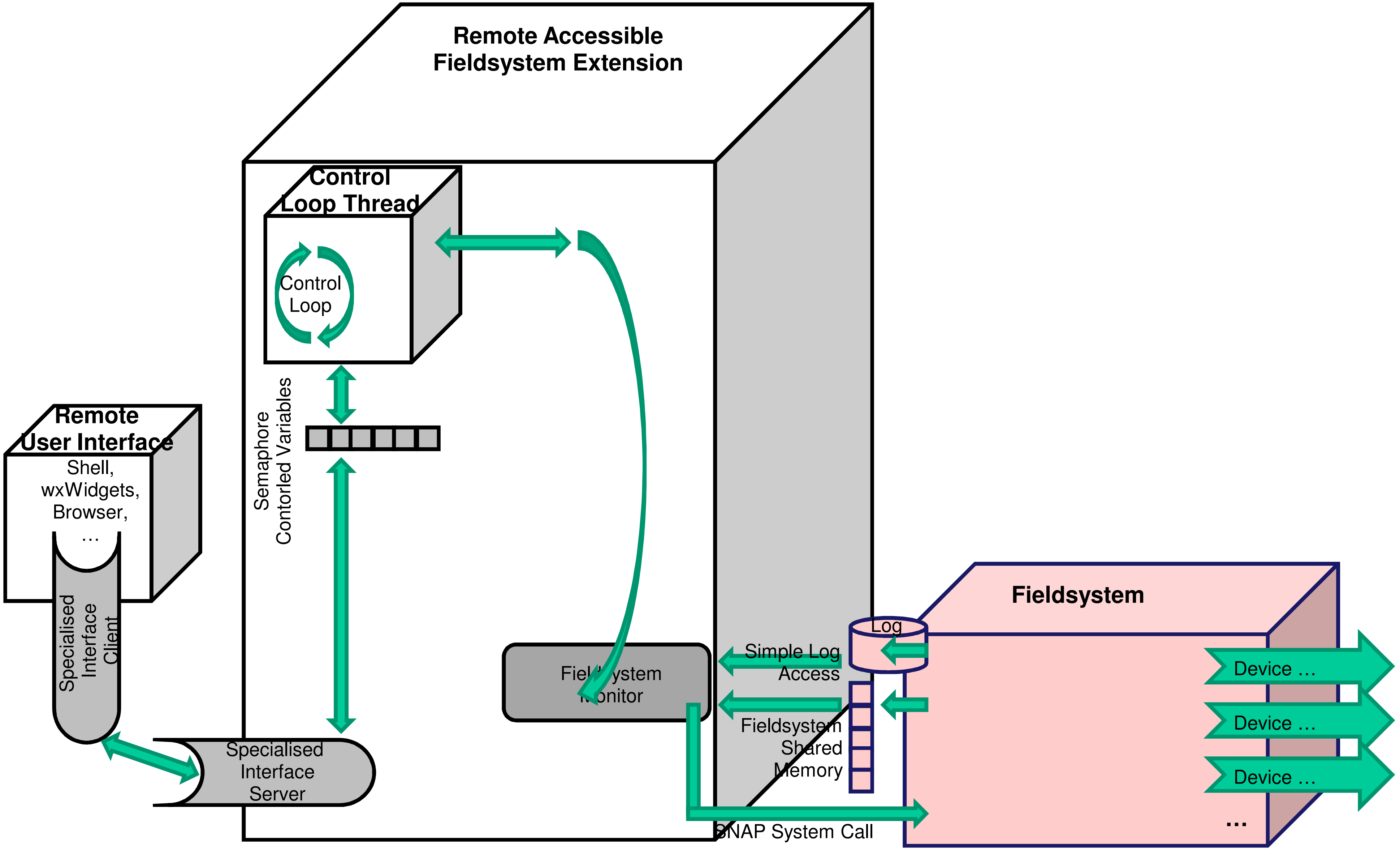
# A fieldsystem extension – autonomous process cells

**Autonomous process cell offers remote fieldsystem monitoring**



**Strict design-separation of**
**- Device control code**
**- Generated communication code**
**- Presentation & user interface code**

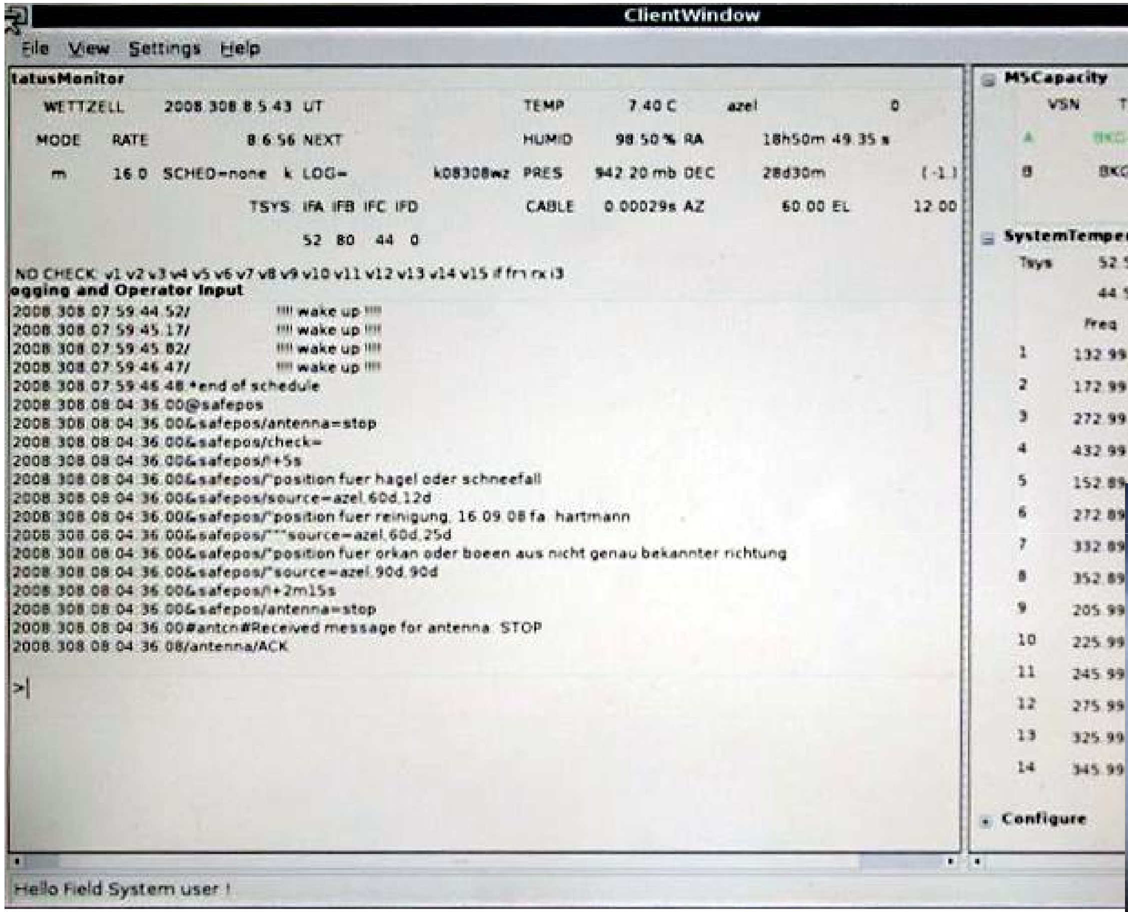
# A fieldsystem extension – autonomous process cells



# A fieldsystem client – remote (graphical) user interface

# A fieldsystem client – graphical, textual or browser based

- Separation of control and presentation logic
- Interchangeability of presentation layer (console shell (ncurses), graphical user interface (wxWidgets), web access via Browser, web service, …)
- Remote controllable via client-server-architecture on idl2rpc-middle-ware
- Modularity in window units and additionaly possible, separately created administration user interfaces for each device
- Basis for graphical user interface: wxWidgets (C++ based Open-Source-Framework for plattform indefendent developement of graphical user interfaces)



```
tatusMonitor
  WETTZELL     2008.308.8.5.43 UT                          TEMP      7.40 C      azel              0
 MODE    RATE              8.6.56 NEXT                     HUMID    98.50 % RA     18h50m 49.35 s
   m      16.0  SCHED=none   k LOG=              k08308wz  PRES    942.20 mb DEC   28d30m          (-1)
                       TSYS  IFA  IFB  IFC  IFD            CABLE   0.00029s AZ       60.00 EL     12.00
                              52   80   44   0
 NO CHECK  v1 v2 v3 v4 v5 v6 v7 v8 v9 v10 v11 v12 v13 v14 v15 if fm rx i3
ogging and Operator Input
2008.308.07:59:44.52/              !!!! wake up !!!!
2008.308.07:59:45.17/              !!!! wake up !!!!
2008.308.07:59:45.82/              !!!! wake up !!!!
2008.308.07:59:46.47/              !!!! wake up !!!!
2008.308.07:59:46.48 *end of schedule
2008.308.08:04:36.00@safepos
2008.308.08:04:36.00&safepos/antenna=stop
2008.308.08:04:36.00&safepos/check=
2008.308.08:04:36.00&safepos/!+5s
2008.308.08:04:36.00&safepos/"position fuer hagel oder schneefall
2008.308.08:04:36.00&safepos/source=azel,60d,12d
2008.308.08:04:36.00&safepos/"position fuer reinigung, 16.09.08 fa. hartmann
2008.308.08:04:36.00&safepos/"""source=azel,60d,25d
2008.308.08:04:36.00&safepos/"position fuer orkan oder boeen aus nicht genau bekannter richtung
2008.308.08:04:36.00&safepos/"source=azel,90d,90d
2008.308.08:04:36.00&safepos/!+2m15s
2008.308.08:04:36.00&safepos/antenna=stop
2008.308.08:04:36.00#antcn#Received message for antenna: STOP
2008.308.08:04:36.08/antenna/ACK

>|
```

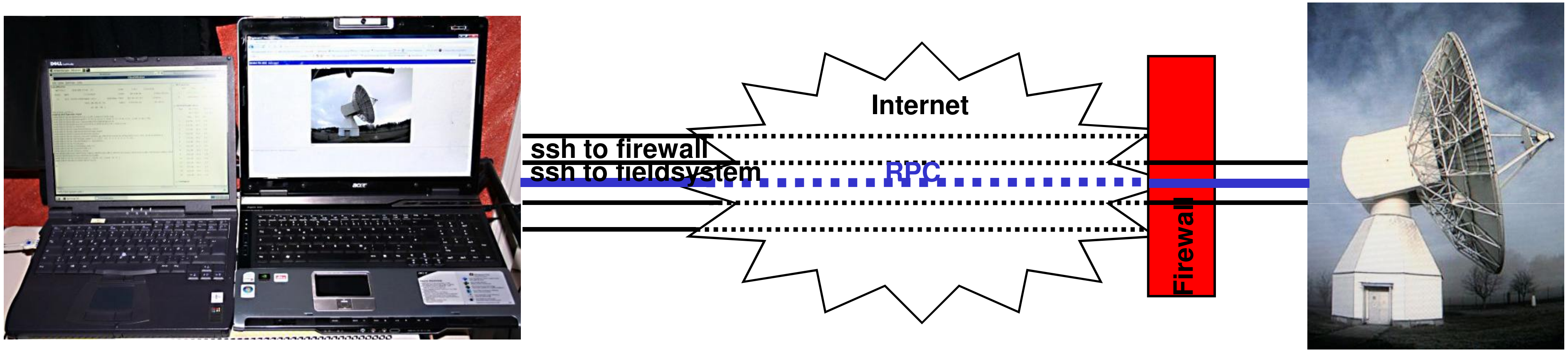
```
MSCapacity
      VSN       Time       GB       %        Check
  A       BKG-0028=              1357.52
  B       BKG-0049     11h4     1931.76       96

SystemTemperature
  Tsys      52.5 (IF1)      80.2 (IF2)
            44.5 (IF3)       0.0 (VC)
          Freq     Ts-U    Ts-L
  1      132.99    59.6     0.0
  2      172.99    56.7     0.0
  3      272.99    56.6     0.0
  4      432.99    46.9     0.0
  5      152.89
  6      272.89
  7      332.89
  8      352.89
  9      205.99
  10     225.99
  11     245.99
  12     275.99
  13     325.99
  14     345.99

Configure
```

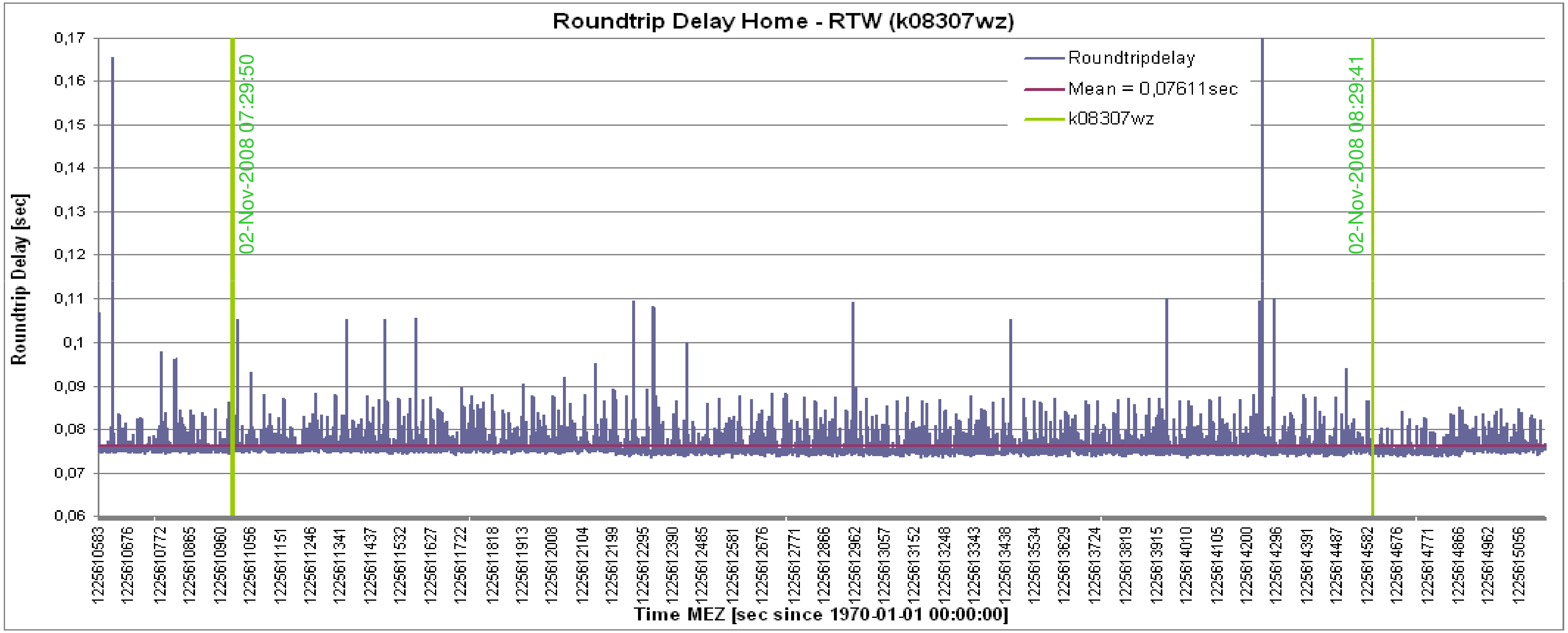
```
Hello Field System user !
```

Webcam

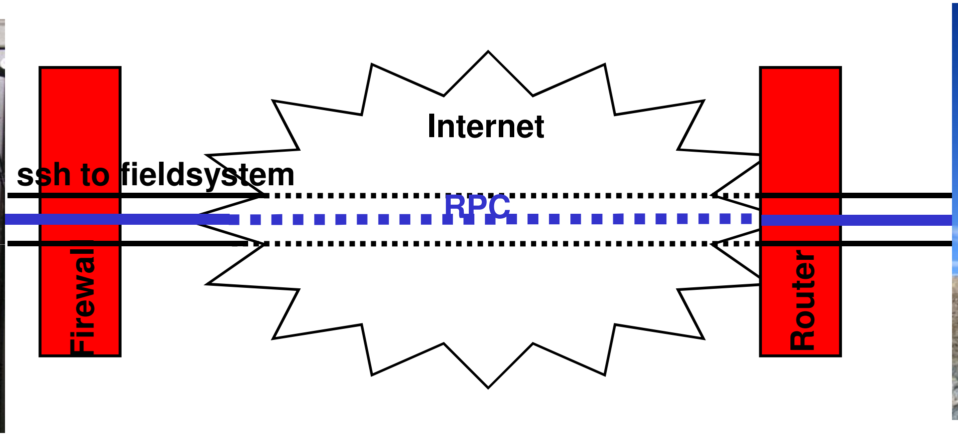
# The first tests – Wettzell, O’Higgins and TIGO go remote

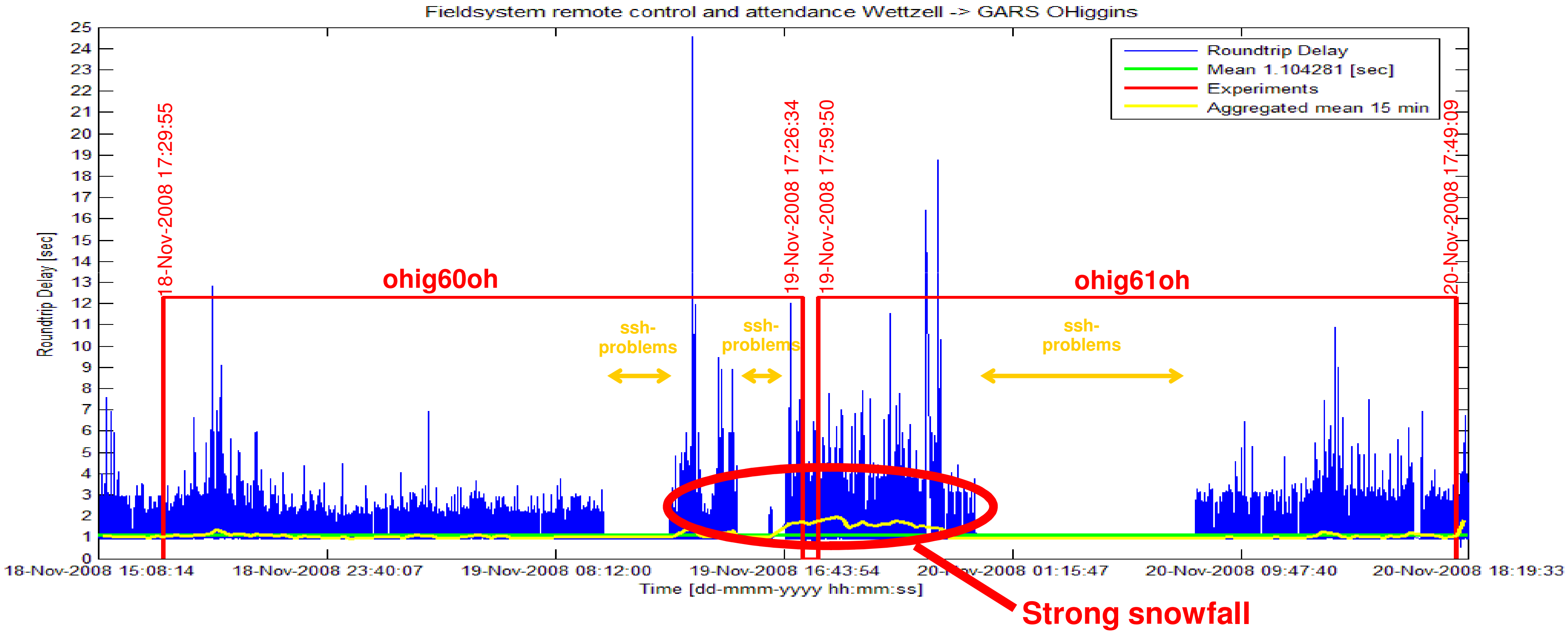
# The first tests – Radiotelescope Wettzell (RTW)/Germany



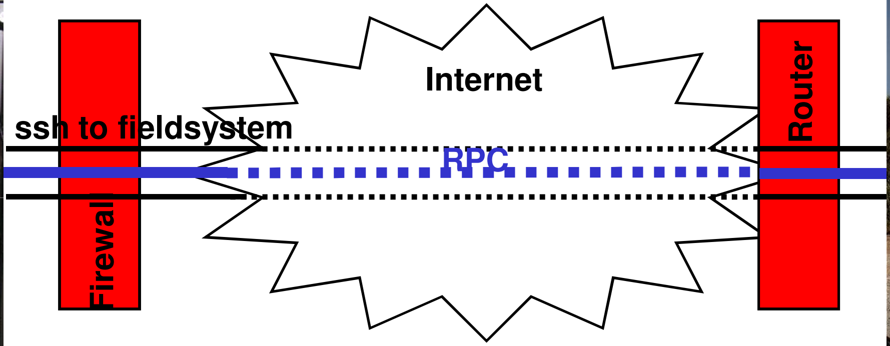


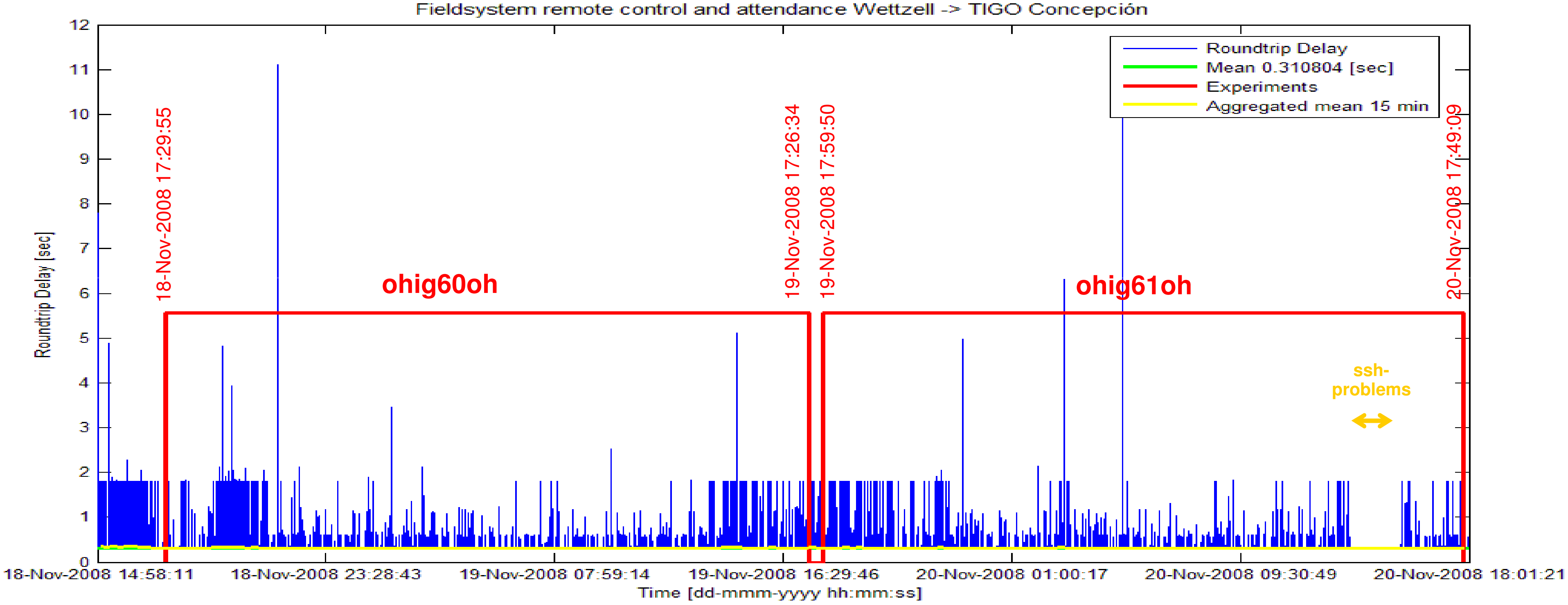
# The first tests – GARS O'Higgins/Antarctica





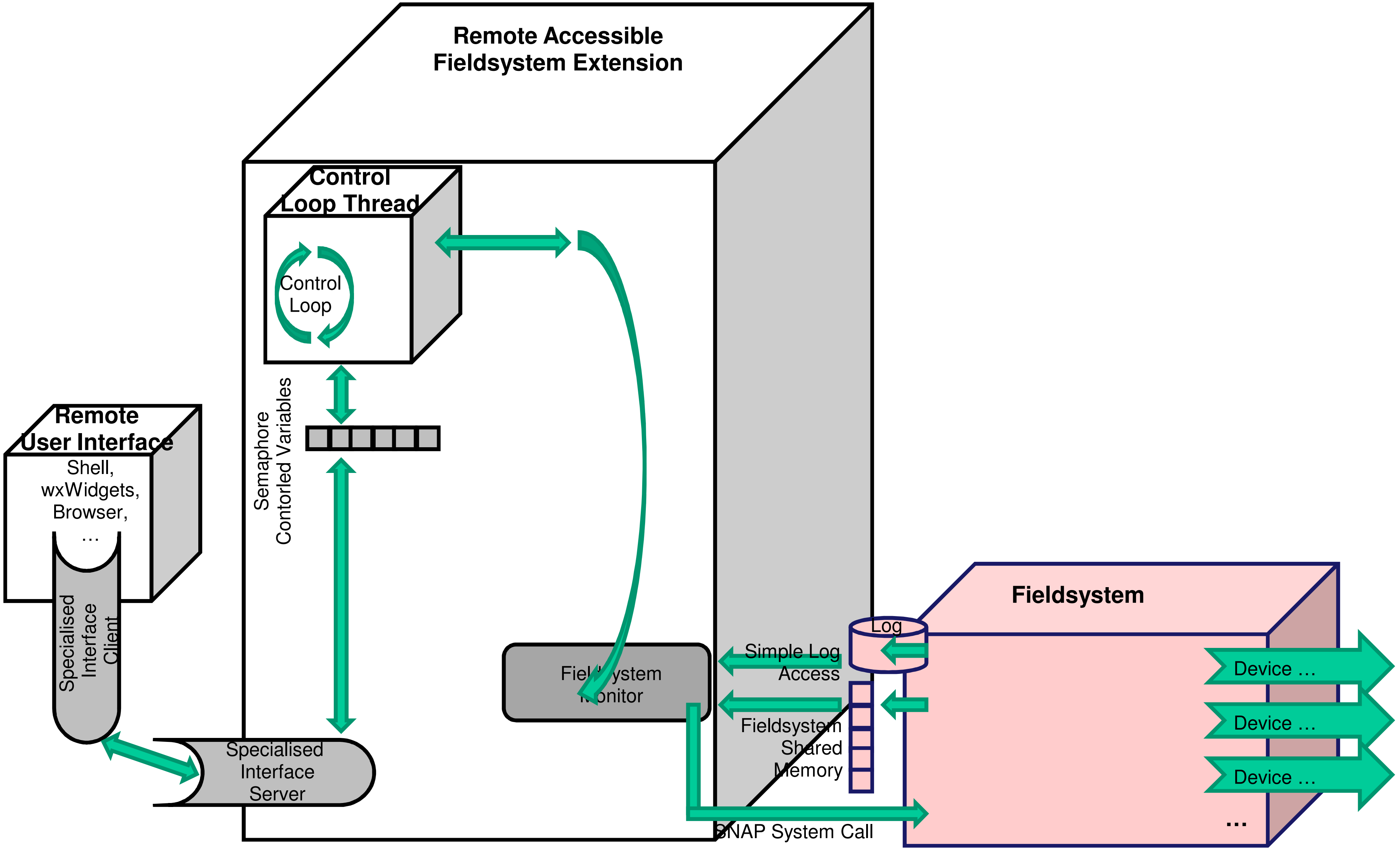
# The first tests – TIGO Concepción/Chile



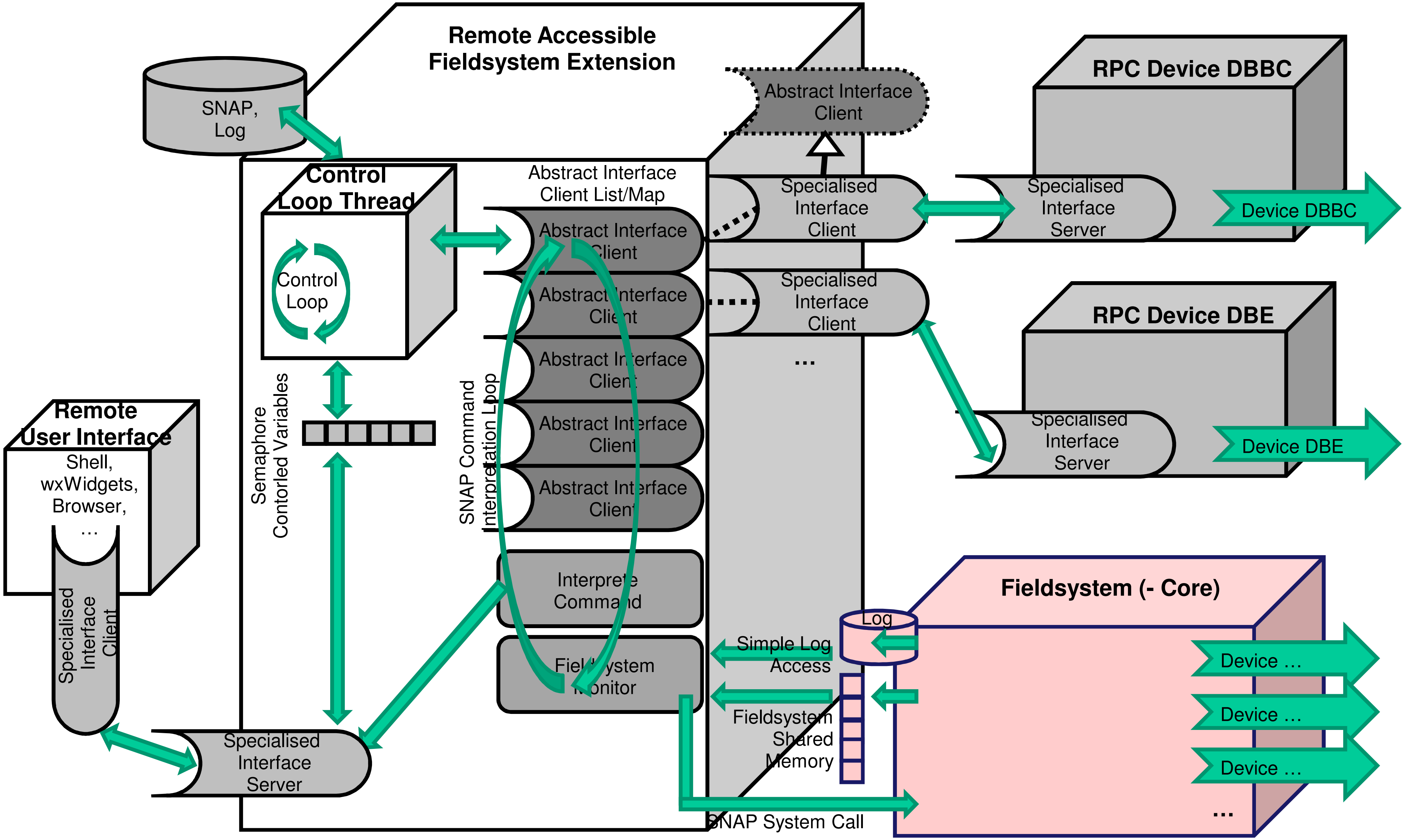


# A possible future – New ideas come with the possibilities

# A fieldsystem extension – autonomous process cells

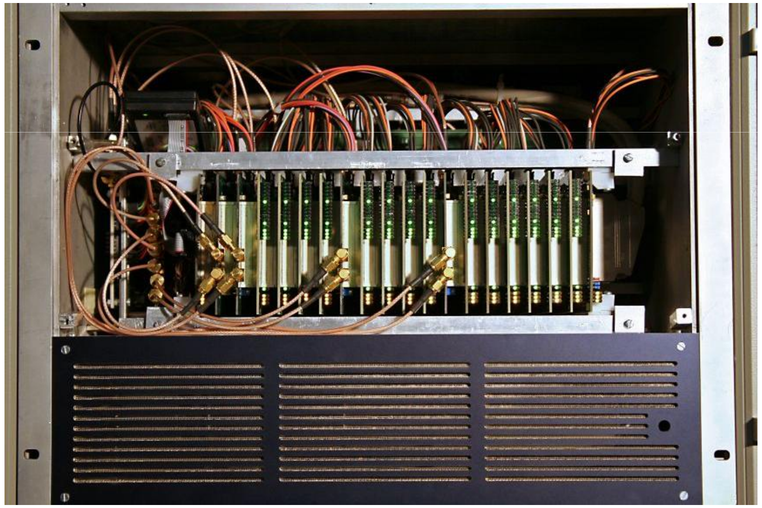


# A fieldsystem extension – remote controlled, autonomous system



# A fieldsystem extension – remote controlled, autonomous devices

## e.g. possible integration of new devices



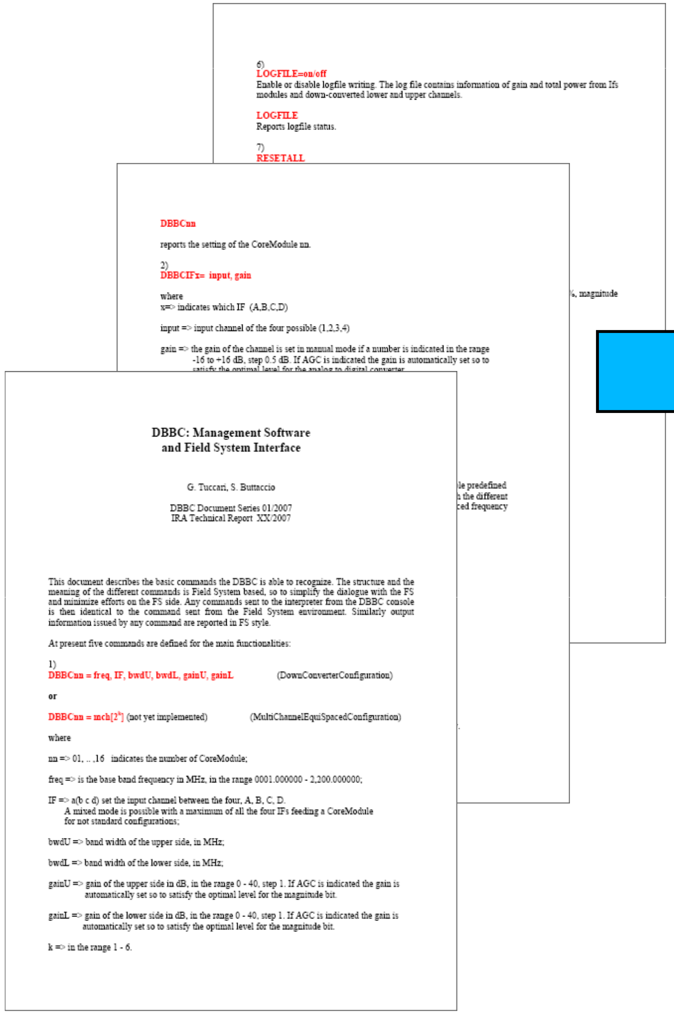
**DBBC (INAF)**

**eVLBI Raid Systems**

# A fieldsystem extension – remote controlled, autonomous devices

## e.g. DBBC

(but at the moment only Linux and on field system side C++ is supported)



6)
**LOGFILE=on/off**
Enable or disable logfile writing. The log file contains information of gain and total power from Ifs modules and down-converted lower and upper channels.

**LOGFILE**
Reports logfile status.

7)
**RESETALL**

%, magnitude

**DBBCnn**

reports the setting of the CoreModule nn.

2)
**DBBCIFx= input, gain**

where
x=> indicates which IF (A,B,C,D)

input => input channel of the four possible (1,2,3,4)

gain => the gain of the channel is set in manual mode if a number is indicated in the range -16 to +16 dB, step 0.5 dB. If AGC is indicated the gain is automatically set so to satisfy the optimal level for the analog to digital converter

le predefined
the different
ced frequency

**DBBC: Management Software and Field System Interface**

G. Tuccari, S. Buttaccio

DBBC Document Series 01/2007
IRA Technical Report XX/2007

This document describes the basic commands the DBBC is able to recognize. The structure and the meaning of the different commands is Field System based, so to simplify the dialogue with the FS and minimize efforts on the FS side. Any commands sent to the interpreter from the DBBC console is then identical to the command sent from the Field System environment. Similarly output information issued by any command are reported in FS style.

At present five commands are defined for the main functionalities:

1)
**DBBCnn = freq, IF, bwdU, bwdL, gainU, gainL** (DownConverterConfiguration)

**or**

**DBBCnn = mch[$2^k$]** (not yet implemented) (MultiChannelEquiSpacedConfiguration)

where

nn => 01, .. ,16 indicates the number of CoreModule;

freq => is the base band frequency in MHz, in the range 0001.000000 - 2,200.000000;

IF => a(b c d) set the input channel between the four, A, B, C, D.
A mixed mode is possible with a maximum of all the four IFs feeding a CoreModule for not standard configurations;

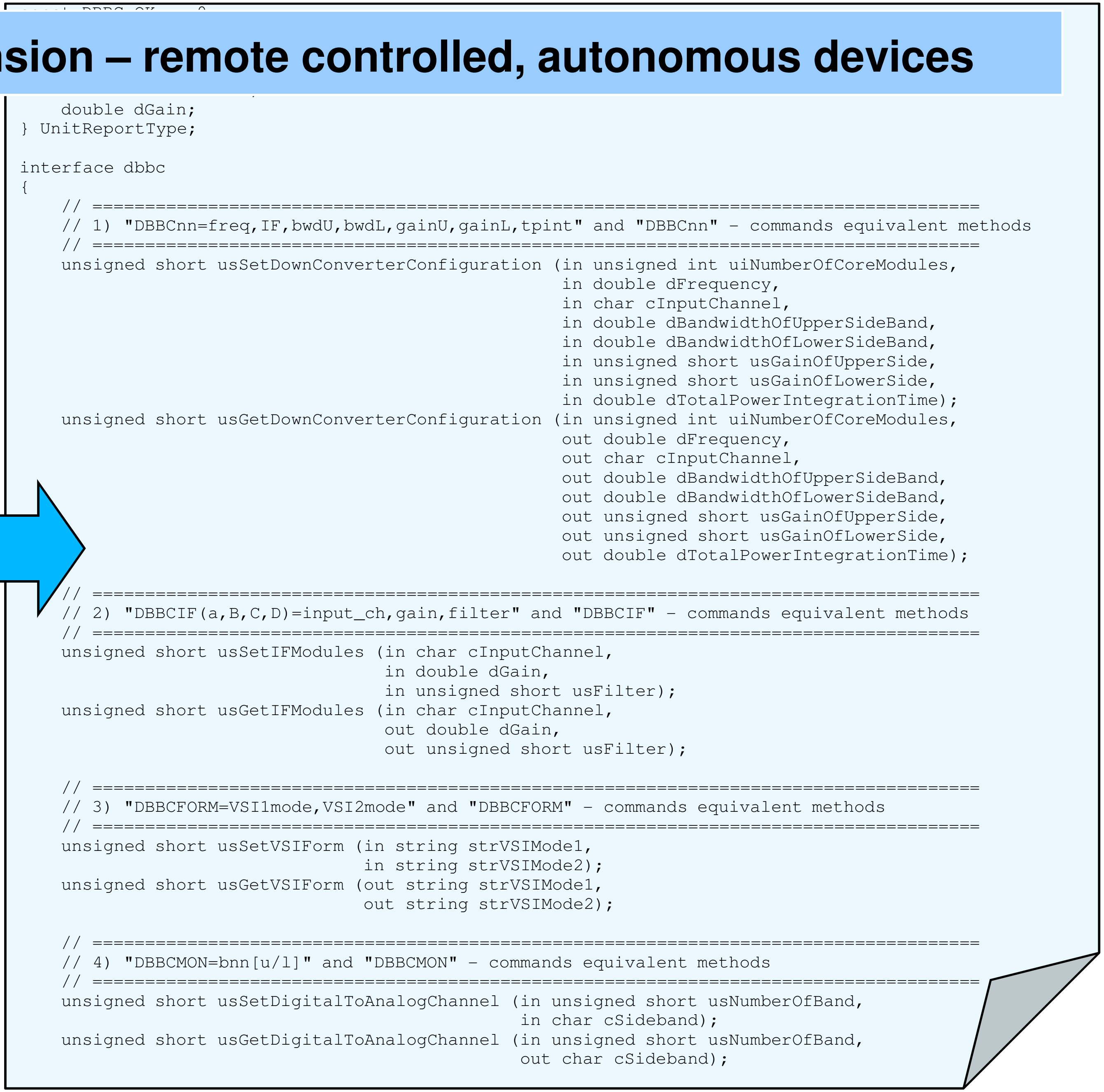
bwdU => band width of the upper side, in MHz;

bwdL => band width of the lower side, in MHz;

gainU => gain of the upper side in dB, in the range 0 - 40, step 1. If AGC is indicated the gain is automatically set so to satisfy the optimal level for the magnitude bit.

gainL => gain of the lower side in dB, in the range 0 - 40, step 1. If AGC is indicated the gain is automatically set so to satisfy the optimal level for the magnitude bit.

k => in the range 1 - 6.



```
    double dGain;
} UnitReportType;

interface dbbc
{
    // ==========================================================================================
    // 1) "DBBCnn=freq,IF,bwdU,bwdL,gainU,gainL,tpint" and "DBBCnn" - commands equivalent methods
    // ==========================================================================================
    unsigned short usSetDownConverterConfiguration (in unsigned int uiNumberOfCoreModules,
                                                    in double dFrequency,
                                                    in char cInputChannel,
                                                    in double dBandwidthOfUpperSideBand,
                                                    in double dBandwidthOfLowerSideBand,
                                                    in unsigned short usGainOfUpperSide,
                                                    in unsigned short usGainOfLowerSide,
                                                    in double dTotalPowerIntegrationTime);
    unsigned short usGetDownConverterConfiguration (in unsigned int uiNumberOfCoreModules,
                                                    out double dFrequency,
                                                    out char cInputChannel,
                                                    out double dBandwidthOfUpperSideBand,
                                                    out double dBandwidthOfLowerSideBand,
                                                    out unsigned short usGainOfUpperSide,
                                                    out unsigned short usGainOfLowerSide,
                                                    out double dTotalPowerIntegrationTime);

    // ==========================================================================================
    // 2) "DBBCIF(a,B,C,D)=input_ch,gain,filter" and "DBBCIF" - commands equivalent methods
    // ==========================================================================================
    unsigned short usSetIFModules (in char cInputChannel,
                                   in double dGain,
                                   in unsigned short usFilter);
    unsigned short usGetIFModules (in char cInputChannel,
                                   out double dGain,
                                   out unsigned short usFilter);

    // ==========================================================================================
    // 3) "DBBCFORM=VSI1mode,VSI2mode" and "DBBCFORM" - commands equivalent methods
    // ==========================================================================================
    unsigned short usSetVSIForm (in string strVSIMode1,
                                 in string strVSIMode2);
    unsigned short usGetVSIForm (out string strVSIMode1,
                                 out string strVSIMode2);

    // ==========================================================================================
    // 4) "DBBCMON=bnn[u/l]" and "DBBCMON" - commands equivalent methods
    // ==========================================================================================
    unsigned short usSetDigitalToAnalogChannel (in unsigned short usNumberOfBand,
                                                in char cSideband);
    unsigned short usGetDigitalToAnalogChannel (in unsigned short usNumberOfBand,
                                                out char cSideband);
```

# A fieldsystem extension – remote controlled, autonomous devices

## e.g. DBBC

(but at the moment only Linux and on field system side C++ is supported)

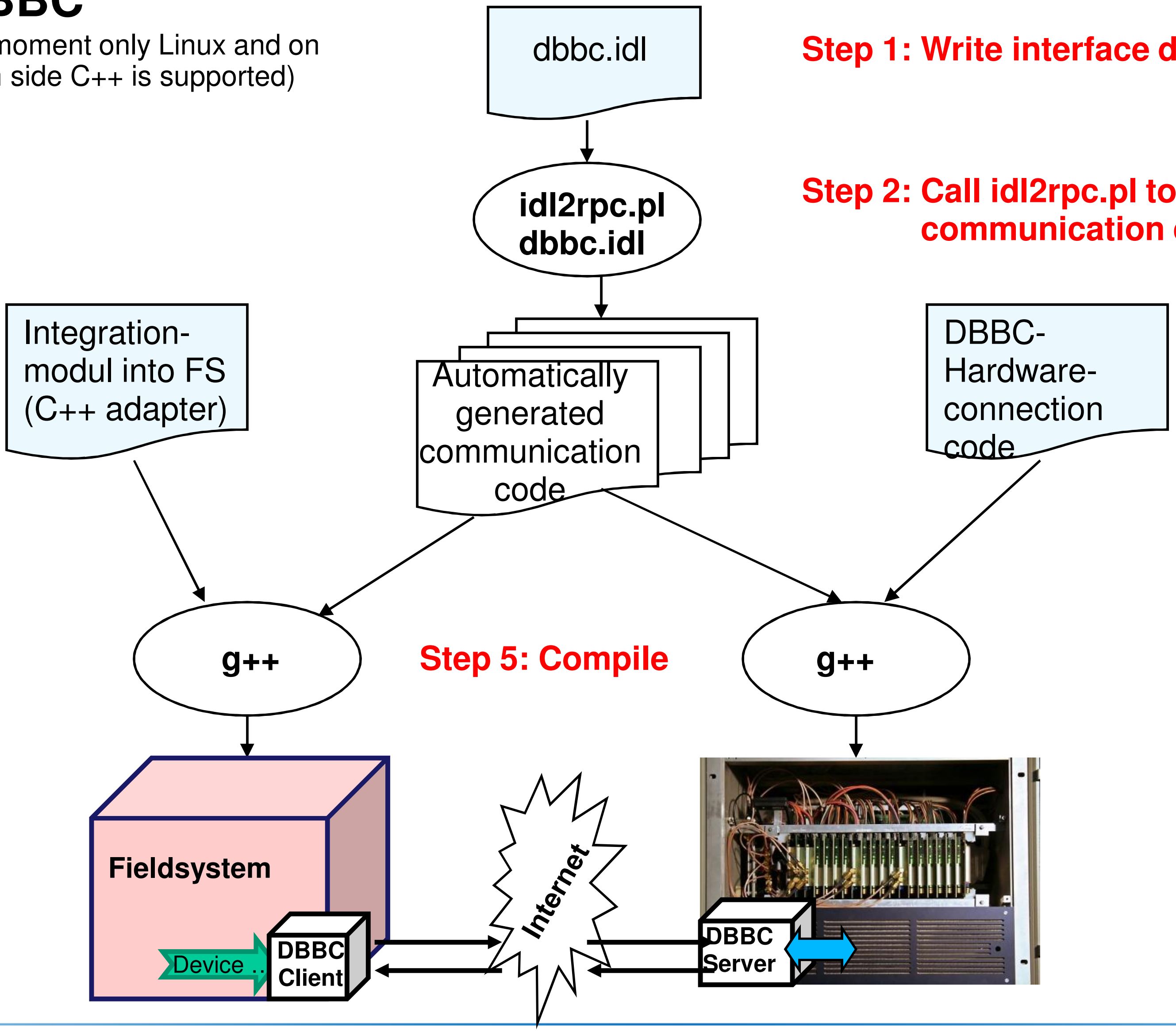
Step 1: Write interface definition for DBBC

Step 2: Call idl2rpc.pl to generate communication code

Step 3: Write code to connect hardware

Step 4: Write code to connect to field system

Step 5: Compile

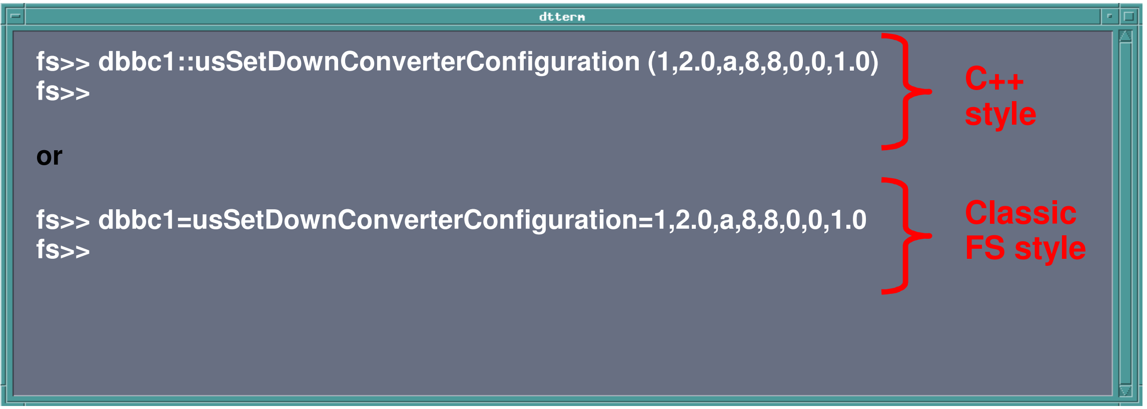

dbbc.idl
idl2rpc.pl dbbc.idl
Integration-modul into FS (C++ adapter)
Automatically generated communication code
DBBC-Hardware-connection code
g++
g++
Fieldsystem
Device ...
DBBC Client
Internet
DBBC Server

# A fieldsystem extension – remote controlled, autonomous devices

## e.g. DBBC

(a concept for the future: automatic command interpreter generation out of the IDL-description)

```
interface dbbc
{
    // =====================================================================================
    // 1) "DBBCnn=freq,IF,bwdU,bwdL,gainU,gainL,tpint" and "DBBCnn" - commands equivalent methods
    // =====================================================================================
    unsigned short usSetDownConverterConfiguration (in unsigned int uiNumberOfCoreModules,
                                                    in double dFrequency,
                                                    in char cInputChannel,
                                                    in double dBandwidthOfUpperSideBand,
                                                    in double dBandwidthOfLowerSideBand,
                                                    in unsigned short usGainOfUpperSide,
                                                    in unsigned short usGainOfLowerSide,
                                                    in double dTotalPowerIntegrationTime);
    unsigned short usGetDownConverterConfiguration (in unsigned int uiNumberOfCoreModules,
                                                    out double dFrequency,
                                                    out char cInputChannel,
                                                    out double dBandwidthOfUpperSideBand,
                                                    out double dBandwidthOfLowerSideBand,
                                                    out unsigned short usGainOfUpperSide,
                                                    out unsigned short usGainOfLowerSide,
                                                    out double dTotalPowerIntegrationTime);
```

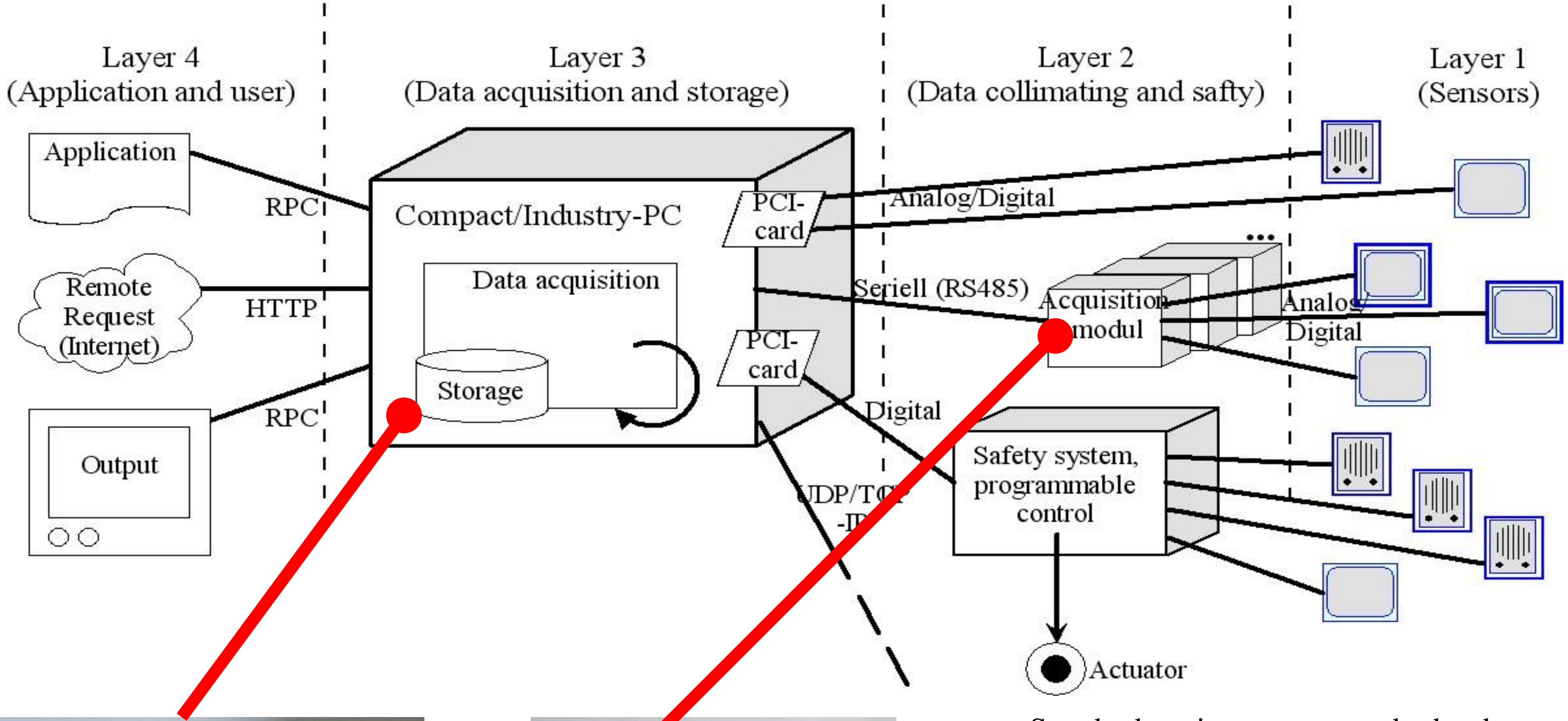


```
fs>> dbbc1::usSetDownConverterConfiguration (1,2.0,a,8,8,0,0,1.0)
fs>>

or

fs>> dbbc1=usSetDownConverterConfiguration=1,2.0,a,8,8,0,0,1.0
fs>>
```

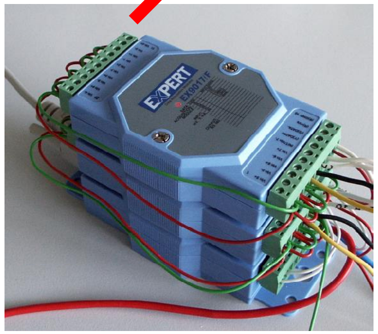
(but not yet implemented)

# A fieldsystem extension – second (safety) monitoring system

Additional control of the system with system monitoring is under construction



Receiver
ACU
Power
Switches
Temp.
Racks
Security
...



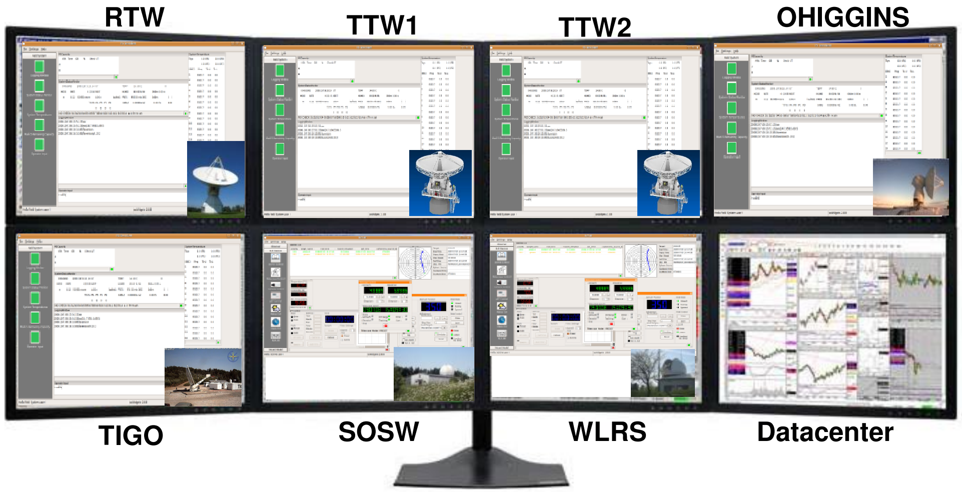
- Standard equipment on standard, robust architectures
- Modular, multi-layer system
- Open for several devices and sonsors
- Passive system for monitoring without actuators
- Linux-operating system (maybe minimal installation)
- Open Source
- C/C++
- Communication internal with idl2rpc-generator
- Vendor independence

# A future concept–
# Combined control of different systems
# in a geodetic observatory

# Combining ideas

## e.g. combined control of different systems in a geodetic observatory

- Think about optimizing work flows
- Increasing the number of observations with automation and remote attendance/control
- Time sharing of measuring equipment
- Just-on-time scheduling and updating to adapt flexible observation programs
- Second integrated security system
- Standardization of system software
- <u>BUT:</u> There will be allways situations where highly educated personnel must be at the observatories



➡ Think about the technical realisations of GGOS ?

'icture similar to: Hase, Hayo; et. al.: Twin Telescope Wettzell (TTW) - a VLBI2010 Radio Telescope Project. IVS General Meeting 2008

# Thank you!

And this is a lucky remote observer in his private “home observatory” controlling the radiotelescope Wettzell immediatly after waking up!