Testing with

xPanel & PCO2

(an introduction)

xPanel & PCO2 … *outline*

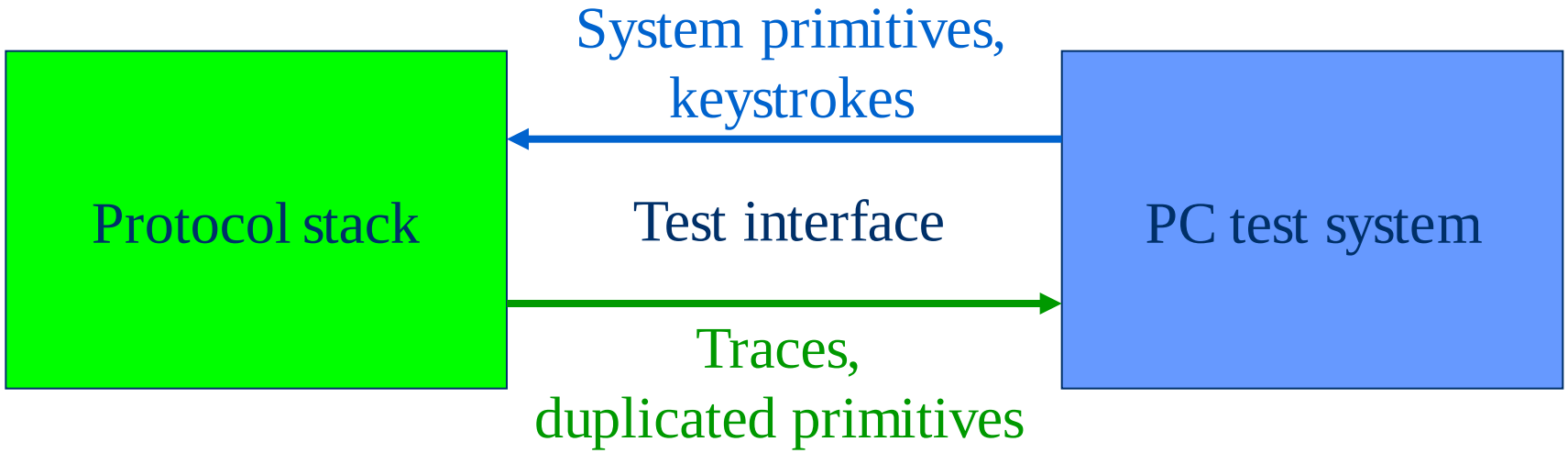
General Concept

Usage of the Tools

Background

xPanel & PCO2 … *general concept*

- Test interface approach:
  - ⇨data interface between G23 protocol stack and a PC test system
  - ⇨usually a standard serial cable, COM-ports on both ends



xPanel & PCO2 … *general concept*

- On stack side:
  - ⇨ test interface entity included in the GPF-FRAME
  - ⇨ uses corresponding hardware driver for communication
- On PC test system side:
  - ⇨ test interface executable using the GPF-FRAME
    - ◆ connects via standard OS drivers
  - ⇨ used by tools like the former PANEL
  - ⇨ .. and of course by xPanel & PCO
  - ⇨ these tools finally provide GUI-stack-access for testers

xPanel & PCO2 … *general concept*

- xPanel - eXtended Panel:
  - ⇨capable to display text & graphics output of mobile MMI
  - ⇨mutable layout, easy to change
- PCO2 - Point of Control and Observation:
  - ⇨filtered watching of traces and duplicated primitives
  - ⇨intuitive configuration (traceclasses, routing) of protocol stack
  - ⇨server, controller, extensible set of viewers

xPanel & PCO2 … *outline*

General Concept

Usage of the Tools

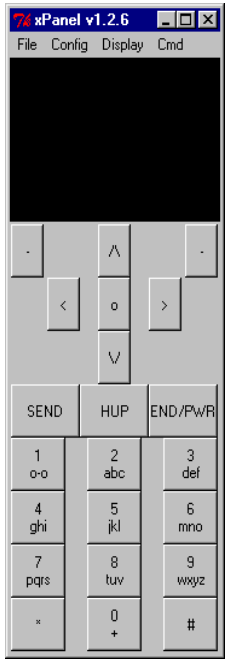
Overview
Logging and Replay
Filter Setup
Communication Setup
Parameter Observation
Important Files

Background

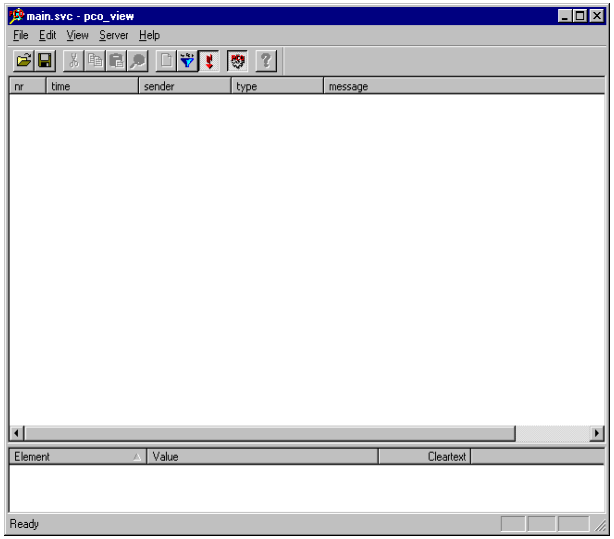
Dataflow
Interpreting/Decoding
Software Layers
Configuration Files

xPanel & PCO2 … *usage overview*

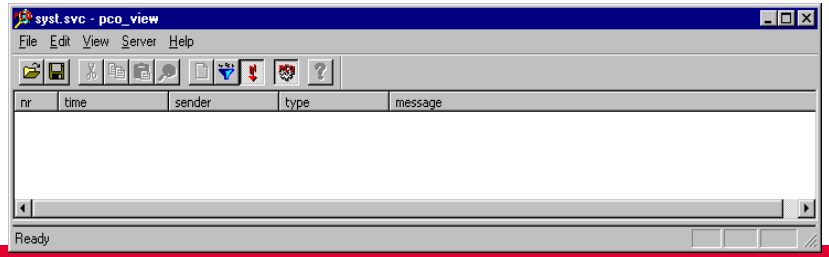
- "Take off":
  - ⇨ starting of pco2.bat in the bin-directory (ClearCase: \gpf\BIN) …
  - ⇨ will per default result in such a scenario:

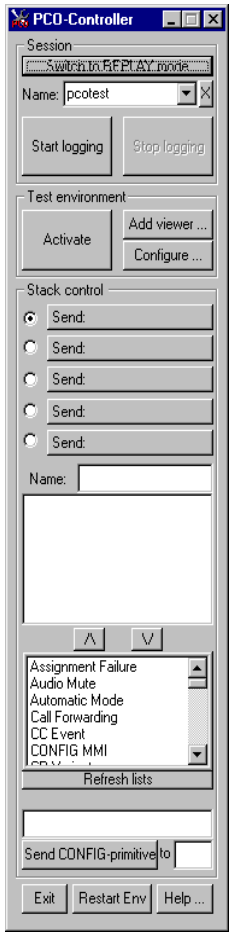


xPanel



PCO-Viewers





PCO-Controller

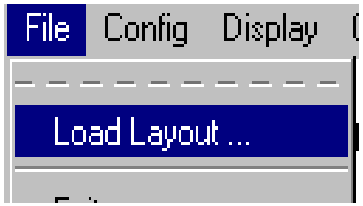
PCO-Server

xPanel & PCO2 … *usage overview*

- The xPanel:

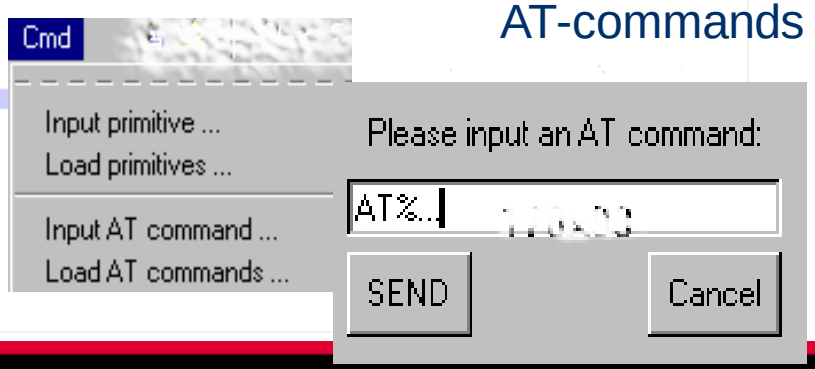
⇨ can be used like a real mobile
- to press keys (e.g. for switching on)
- to see display output

⇨ is easily adaptable to your layout



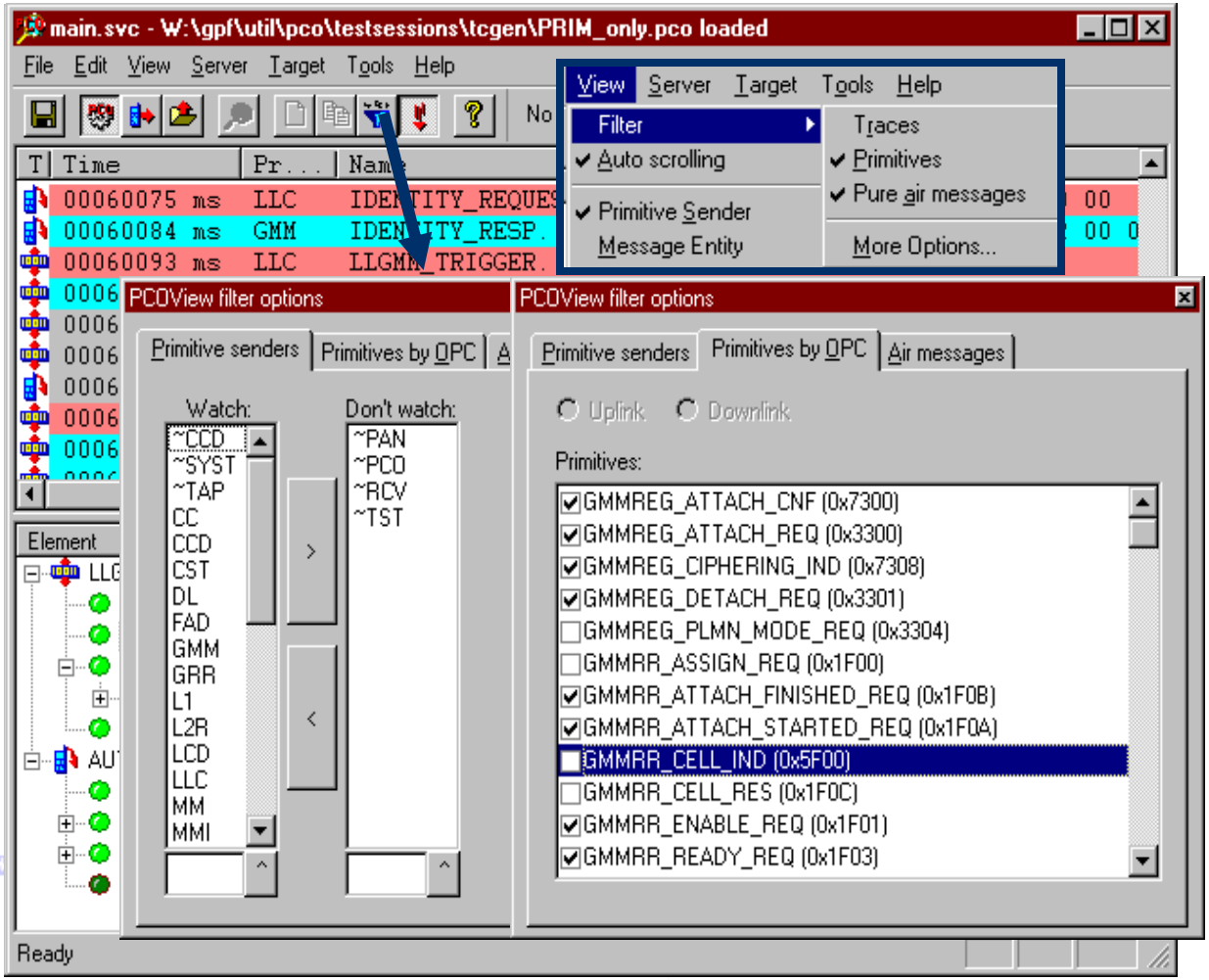
⇨ provides GUI to select communication type

⇨ provides interface to send FRAME system primitives and AT-commands



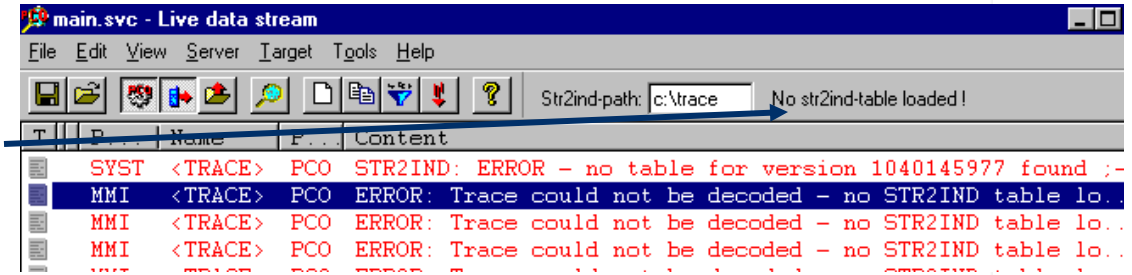
xPanel & PCO2 … *usage overview*

- The PCO-Viewer(s):
  - ⇨ watch traces of selected entities
    - ordered by time
    - distinguished by colors
  - ⇨ watch redirected primitives/messages
    - as hexdump
    - as structure
  - ⇨ filter by sender or OPC
  - ⇨ configuration can be stored as a ".svc"-file



xPanel & PCO2 … *usage overview*

- Compressed Tracing with Str2Ind-Tables:
  - ⇨ for performance and memory reasons traces are compressed at compile time
  - ⇨ each PS build creates a str2ind-table containing [ID]->[Trace string] combinations
  - ⇨ initially no table for interpretation of Trace-IDs is loaded

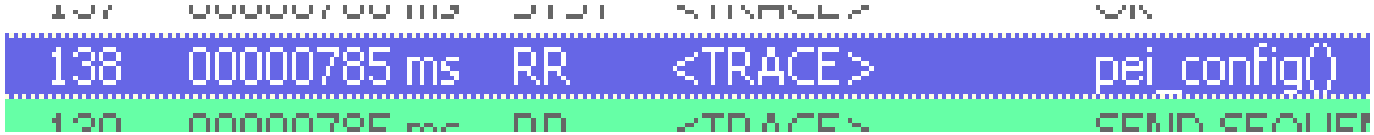


  - ⇨ after reset or by explicit request
    - PS sends a version number
    - Viewer searches for matching .tab-file in specified directory structure

  - ⇨ after loading of the table all traces will be shown as expected

xPanel & PCO2 … *usage overview*

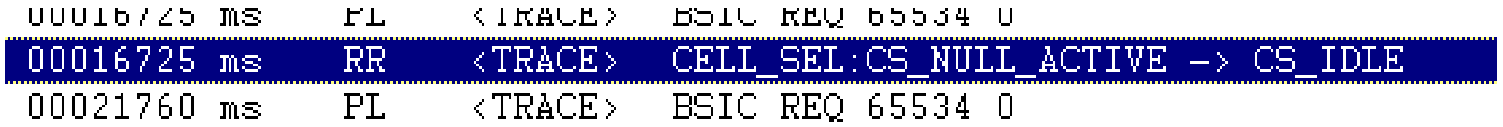
- Types of traces shown by PCO-Viewer:
  - ⇨ Function traces …



  - ⇨ Event traces …

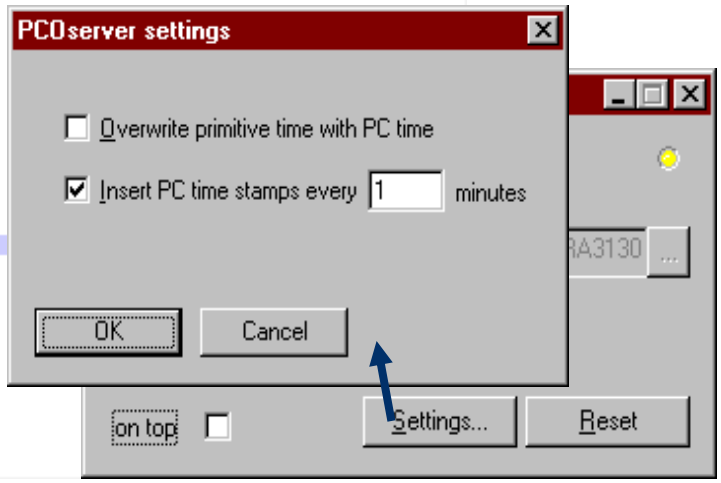
  - ⇨ Primitive traces …

  - ⇨ State traces …



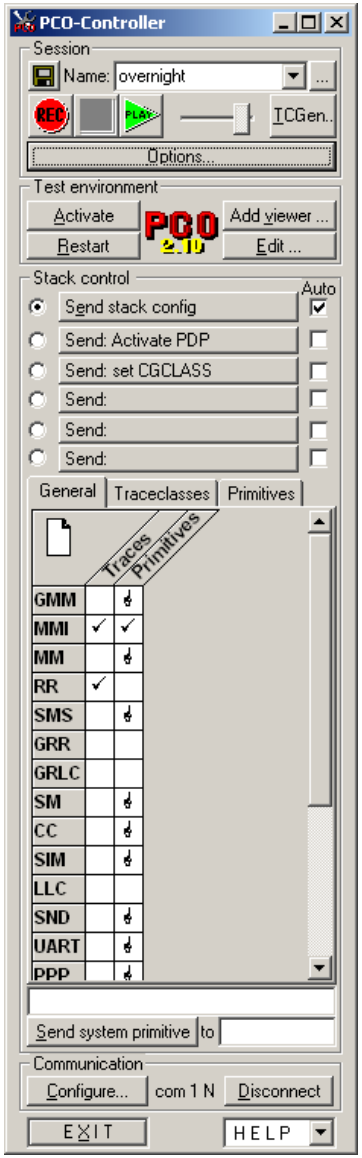
xPanel & PCO2 … *usage overview*

- The PCO-Server:
  - ⇨ receives all traces and redirected primitives from the target
    - forwards them to viewers
    - may store them into logfiles
  - ⇨ for replay it can forward logged data to viewers, too
  - ⇨ may insert extra timestamp traces
  - ⇨ is controlled by the PCO-Controller



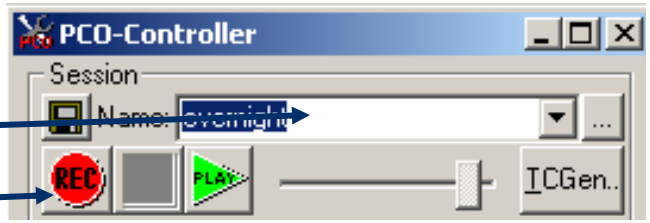
xPanel & PCO2 … *usage overview*

- The PCO-Controller:
  - ⇨ is actually executed by pco2.bat and starts a configurable set of other applications, like xPanel
  - ⇨ provides access to PCO-Server
    - to start logging of data
    - to replay logged data
  - ⇨ allows sending of FRAME system primitives to the protocol stack (via PCO-Server)
    - from direct user input
    - out of a pool of predefined system primitives loaded from a dedicated file (usually View.txt)
    - by selections in a matrix (stored per default in pco_stack.xml) which may be ex-/imported
  - ⇨ enables setup of the communication drivers to be used
  - ⇨ on exit: shuts down all initially started applications
    - called the “test environment”



xPanel & PCO2 … *logging and replay*

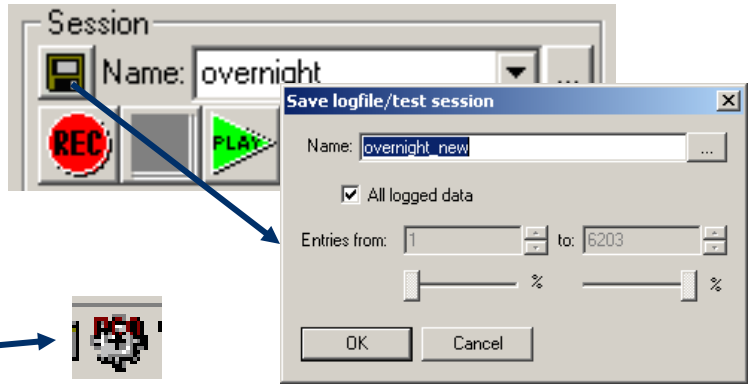
- Logging / Recording:
  - ⇨ specify name of test session
  - ⇨ start logging process



  - ⇨ now every trace/primitive received via the test interface will be logged
    - Independent of any filter setting in a PCO-Viewer

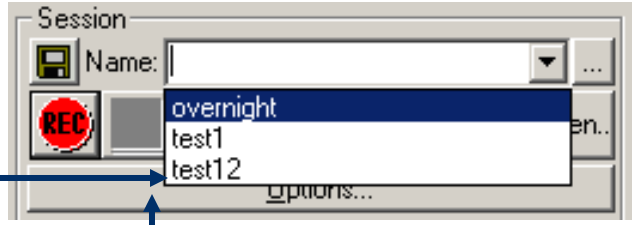
  - ⇨ PCO-Server appears green

  - ⇨ after pressing the “STOP” button …
    - a <session name>.pco file can be found in the current session dir of PCO-Server
    - a copy of (selected parts of) the logged session can be stored somewhere else (and, e.g., be sent to developers)
  - ⇨ PCO-Server appears red again



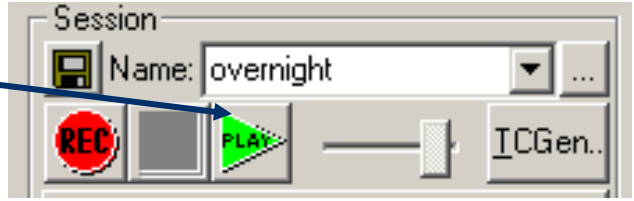
xPanel & PCO2 … *logging and replay*

- Replay:
  - ⇨ select test session name
  - ⇨ or drag/drop .pco-file to PCO-Controller

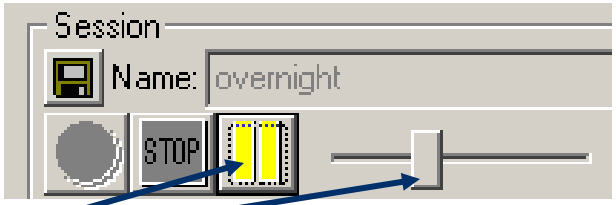


overnight.pco

  - ⇨ press the „PLAY“ button



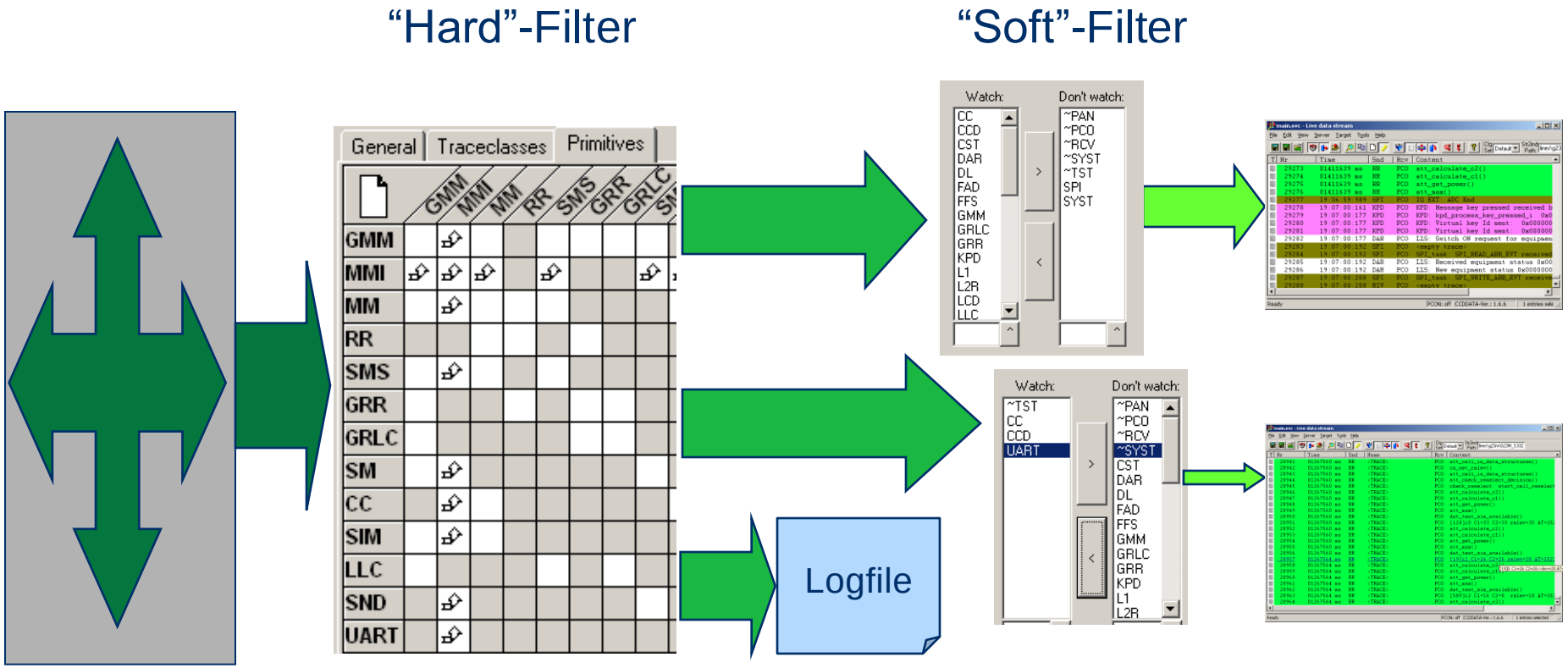
  - ⇨ now logged traces/primitives will be replayed in connected PCO-Viewers
    - depending on the individual filter settings
  - ⇨ pausing and repositioning are possible



  - ⇨ PCO-Server appears yellow

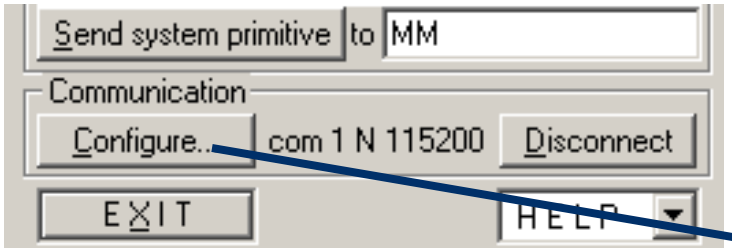
xPanel & PCO2 … *filter setup*

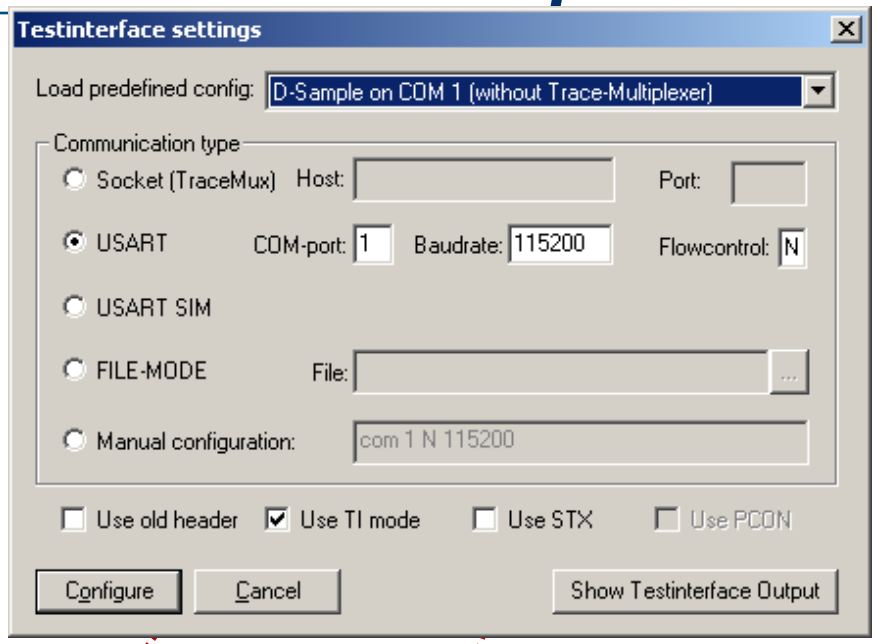
- Trace/primitive filtering is done in two stages:



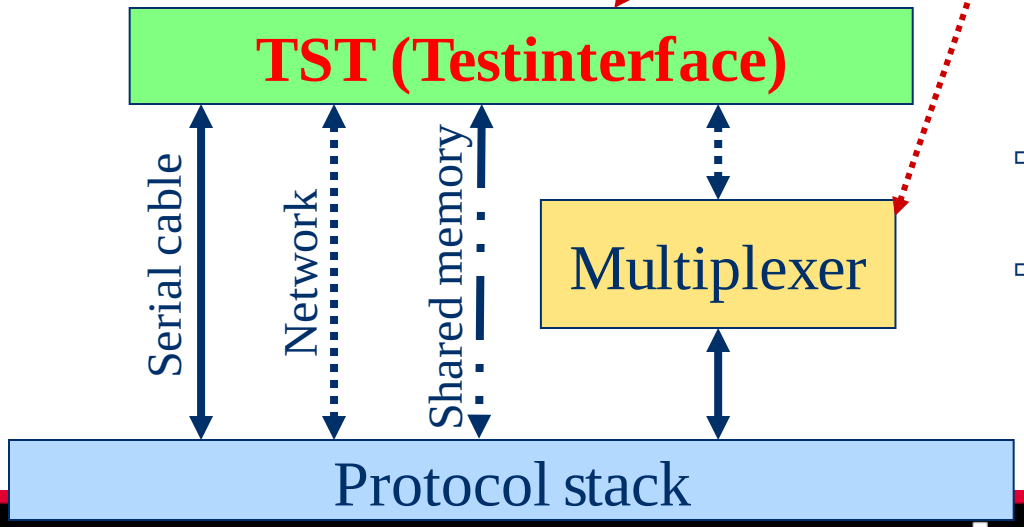
xPanel & PCO2 … *communication setup*

PCO-Controller:





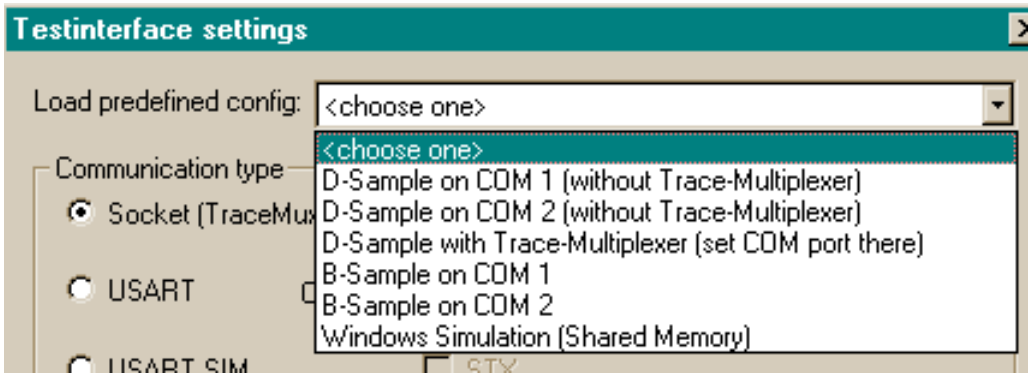
- ⇨ selection of mode
- ⇨ secification of individual parameters



- ⇨ configuration of test interface
- ⇨ evtl. start of extra tools

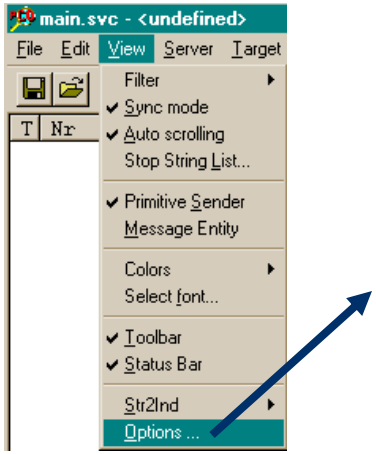
xPanel & PCO2 … *communication setup*

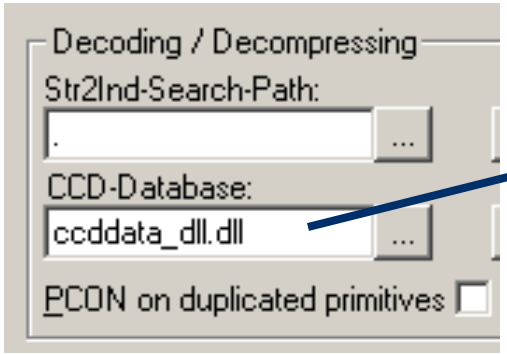
⇨ for convenience several default configurations exist



⇨ it has to be ensured that a matching ccddata-DLL is selected

PCO-Viewer:



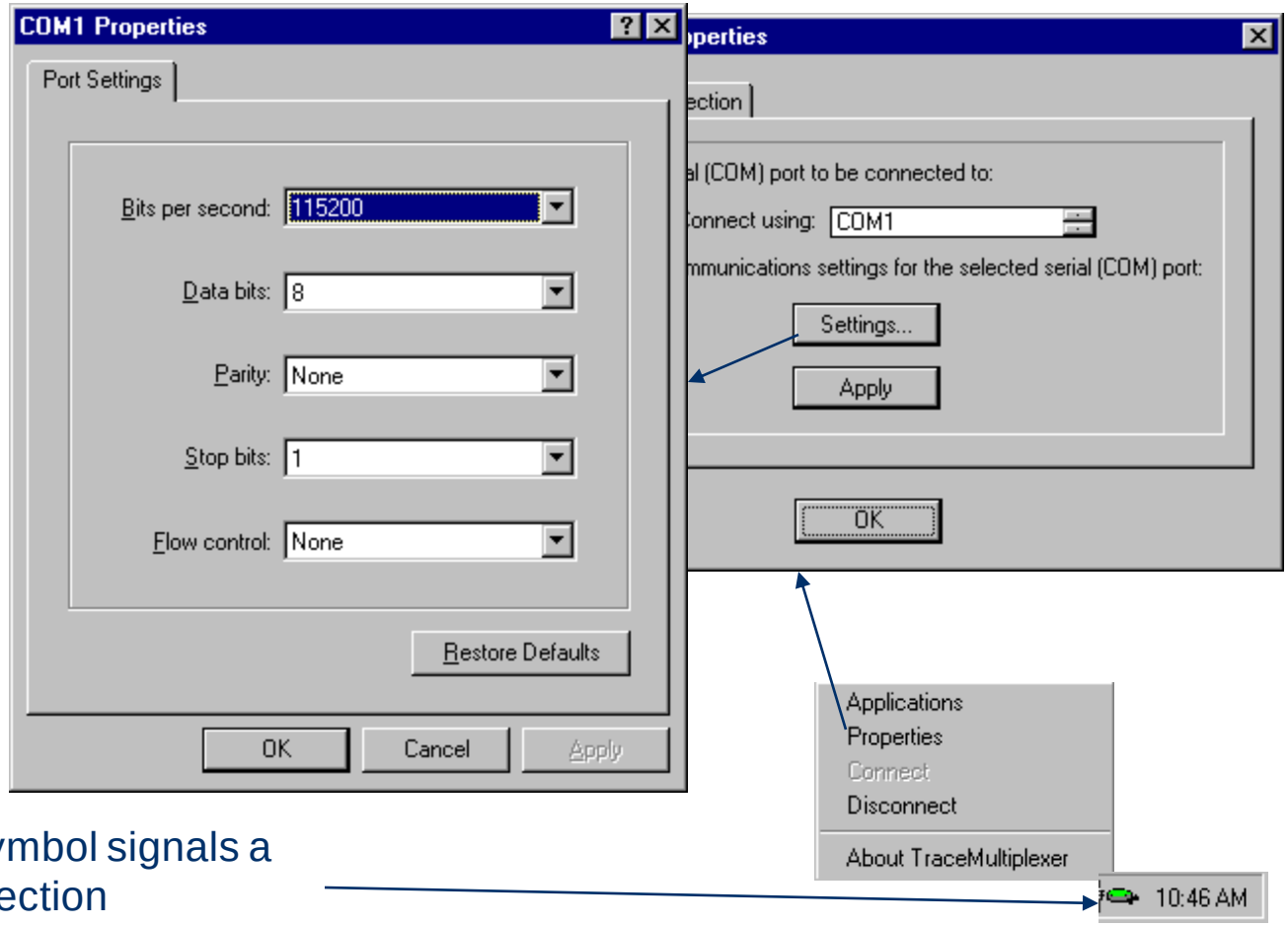


- either a DLL which has been delivered together with the PS-image
- or a prebuild DLL from …\ccddata\ (e.g. ccddata_G23M_333_S64.dll for older B-Sample-Releases)

xPanel & PCO2 … *communication setup*

- Communication via TraceMultiplexer:
  - ⇨ if using the TraceMultiplexer for the first time it has to be configured:

1. choose COM port
2. select baudrate of 115200
3. disable flow control



- ⇨ the green tray symbol signals a successful connection

xPanel & PCO2 … *parameter observation*

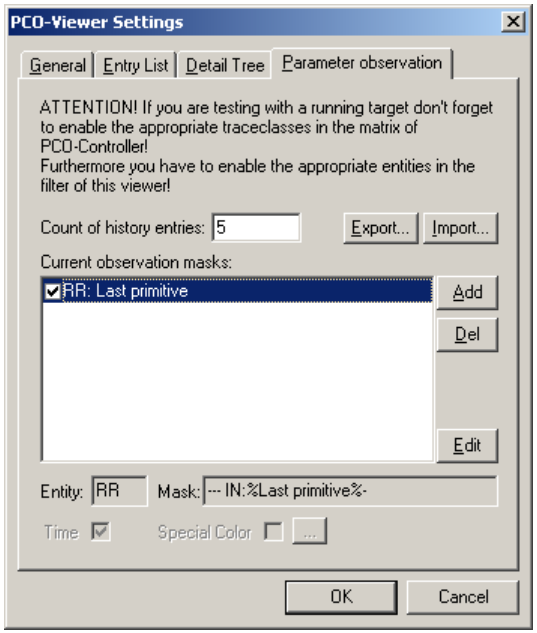
- Parameter observation is a special feature of PCO-Viewer ...
  - ⇨ allowing to filter specific strings out of traces
  - ⇨ ... and displaying them in an extra sub-window

- First you have to …

```
RR   <TRACE>  PCO   --- IN:MPH_MEASUREMENT_IND-(0x1701)
```

  - ⇨ find out how traces containing the wanted parameter look like
  - ⇨ … or request specific traces from the developer

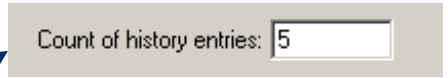
- Afterwards you must …
  - ⇨ translate an example trace into a mask
  - ⇨ … where the actual parameter values are exchanged by a placeholder starting and ending with ‘%’-characters,
    for example: “--- IN:%Last primitive%-”

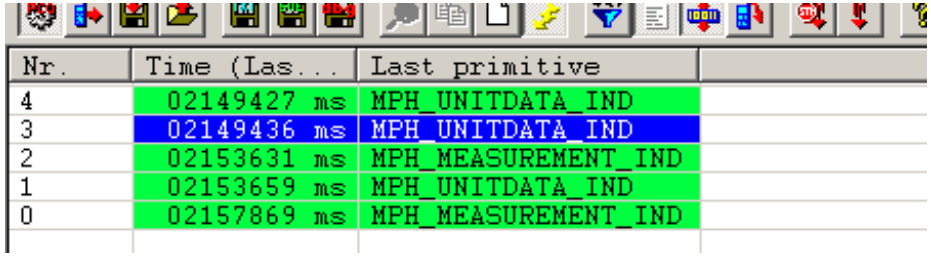


- This mask can now be added to the list …
  - ⇨ by opening the options dialog, e.g., via the menu entry “view/options…”
  - ⇨ ... and together with more settings, like the entity

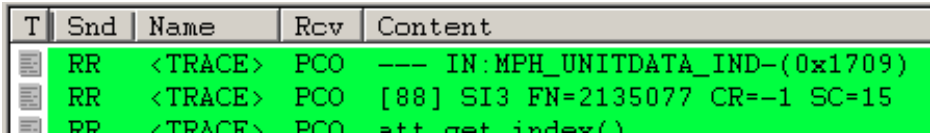
xPanel & PCO2 … *parameter observation*

- Now the latest parameter values ...
  - ⇨ can be watched in the special sub-window
  - ⇨ ... together with their historical values depending on the setting in the options dialog





Nr.	Time (Las...	Last primitive
4	02149427 ms	MPH_UNITDATA_IND
3	02149436 ms	MPH_UNITDATA_IND
2	02153631 ms	MPH_MEASUREMENT_IND
1	02153659 ms	MPH_UNITDATA_IND
0	02157869 ms	MPH_MEASUREMENT_IND



T	Snd	Name	Rcv	Content
	RR	<TRACE>	PCO	--- IN:MPH_UNITDATA_IND-(0x1709)
	RR	<TRACE>	PCO	[88] SI3 FN=2135077 CR=-1 SC=15

- Values in this grid ...
  - ⇨ can be copied to clipboard
  - ⇨ ... and sorted
- A click on a value ...
  - ⇨ moves the main entry list to the corresponding trace

- Attention! The sender entities of the expected traces containing parameters have to be in the watch list of the PCO-Viewer filter!

xPanel & PCO2 … *important files*

- Volatile files:
  - ⇨ have to be build together with the used protocol stack
    - **ccddata_dll.dll** (database with primitive symbols)
    - **str2ind.tab** (table with “ID <-> trace text” associations)
- Detailed documentation:
  - ⇨ **pco_userguide.doc**
    (ClearCase: \gpf\DOC\pco\pco_userguide.doc)
  - ⇨ **xpan_userguide.doc**
    (ClearCase: \gpf\DOC\xpanel\xpan_userguide.doc)
  - ⇨ **xPanel_plus_design_spec.doc**
    (ClearCase: \gpf\DOC\xpanel\xPanel_plus_design_spec.doc)
    - contains requirements for a graphical display-driver
  - ⇨ **frame_users_guide.doc**
    (ClearCase: \gpf\DOC\frame_users_guide.doc)
    - contains description of FRAME system primitives
  - ⇨ **pco_intro.pps** (this document)
    (ClearCase: \gpf\DOC\pco\pco_intro.pps)

xPanel & PCO2 … *outline*

General Concept

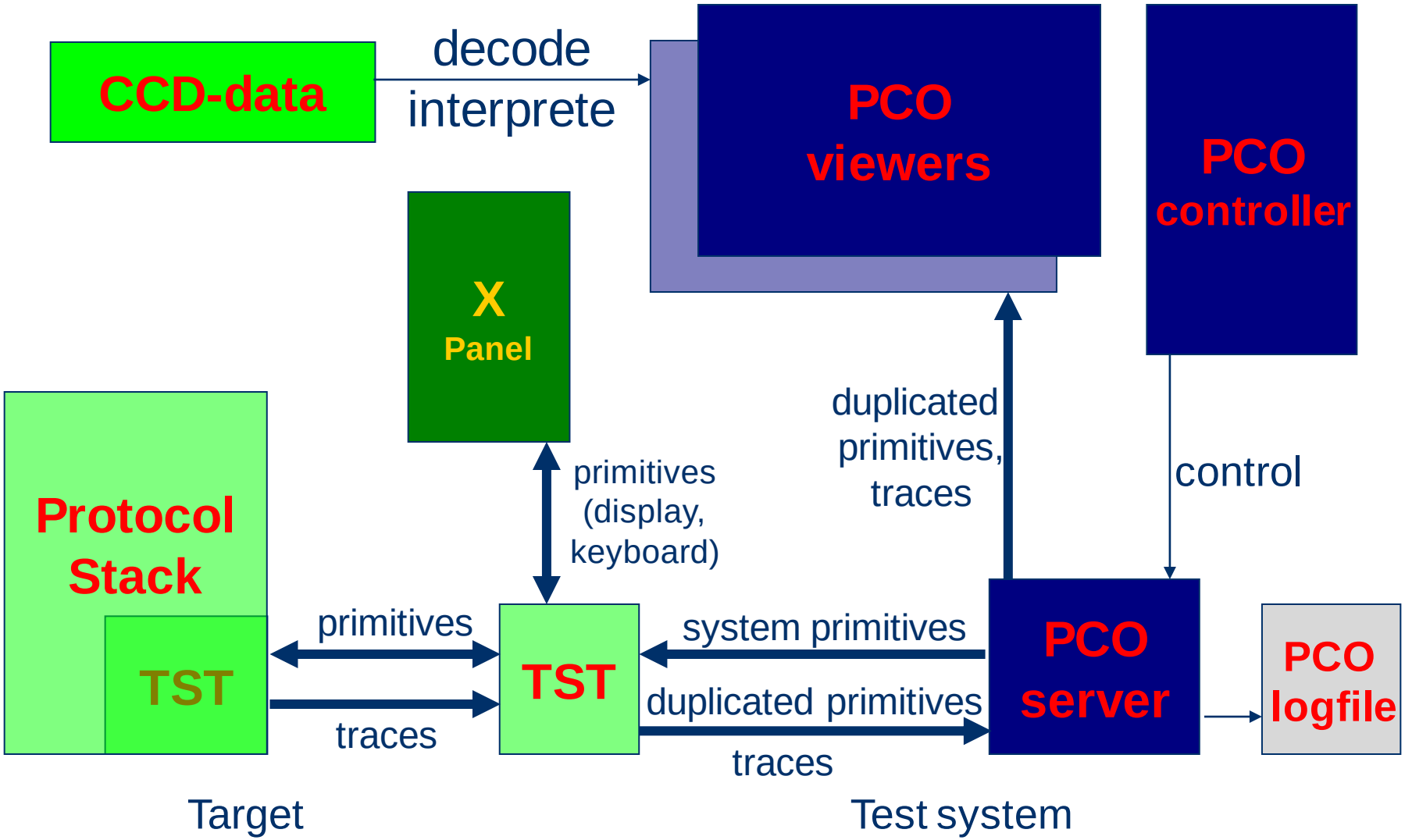
Usage of the Tools

Overview
Logging and Replay
Filter Setup
Communication Setup
Parameter Observation
Important Files

Background

Dataflow
Interpreting/Decoding
Software Layers
Configuration Files

xPanel & PCO2 … *data flow*



xPanel & PCO2 … *interpreting/decoding*

- What for do I need a ccddata-DLL (e.g. ccddata_dll.dll) ?
  - ⇨ contains information about all primitive and air message structures used in the corresponding protocol stack

Without:

```
MMI     <TRACE>  PCO  ---OUT:##OPC:0x0E0A##
MMI     <TRACE>  PCO  ---OUT:##OPC:0x0E0A##
```

```
GMM     <PRIMITIVE:0x2E01>        MM   01 01 FF CA
MM      <PRIMITIVE:0x80004004>    DL   FF 00 00 00
```

Element	Value
⊟ <no ccddata-DLL loaded>	
0x0000	01 01 FF CA

With a matching DLL:

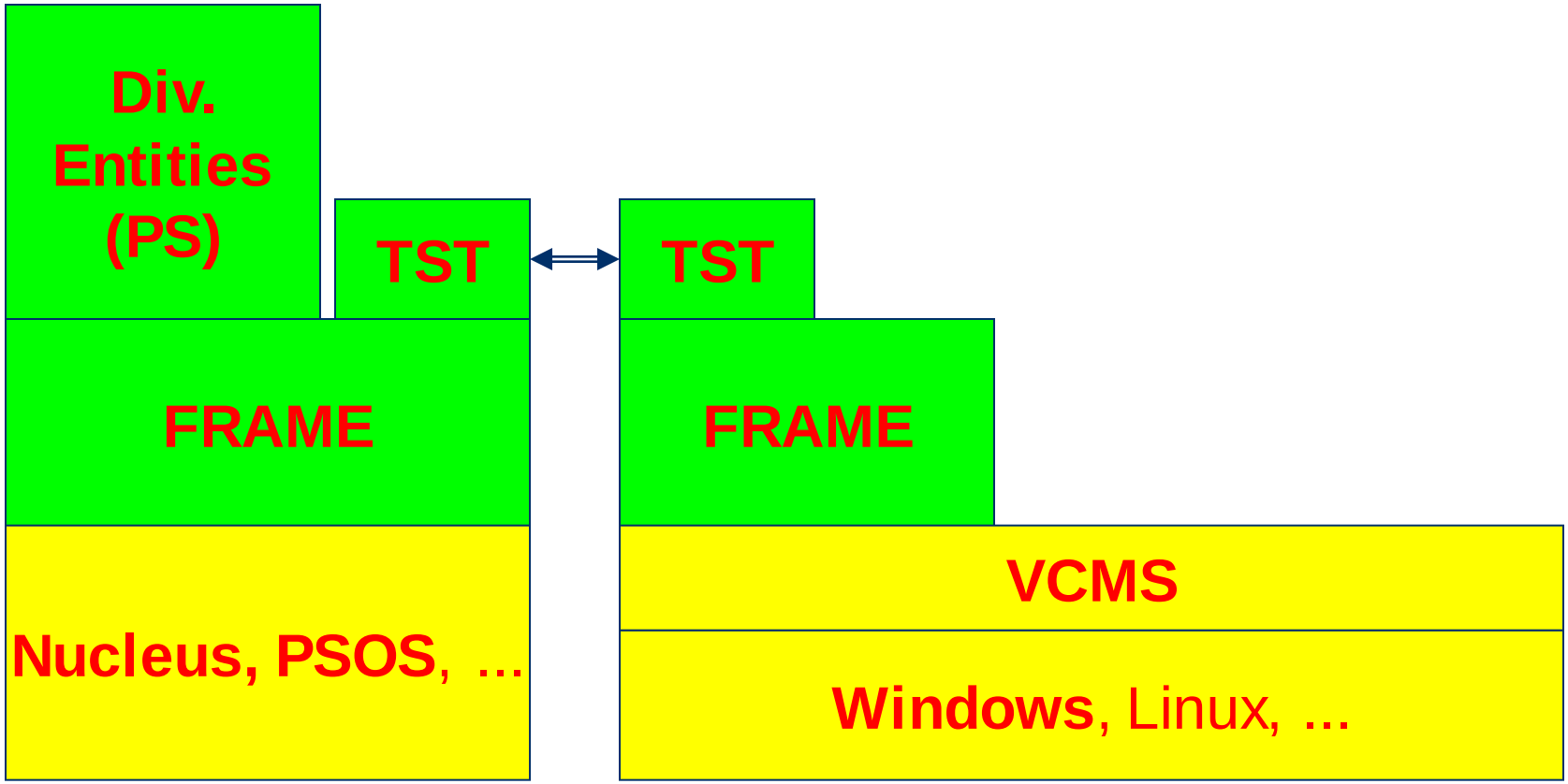
```
MMI     <TRACE>  PCO  ---OUT:MMI_DISPLAY_REQ-(0x0E0A)
MMI     <TRACE>  PCO  ---OUT:MMI_DISPLAY_REQ-(0x0E0A)
```

```
GMM     MMGMM_NREG_REQ          MM   01 01 FF CA
MM      MDL_RELEASE_REQ         DL   FF 00 00 00
```

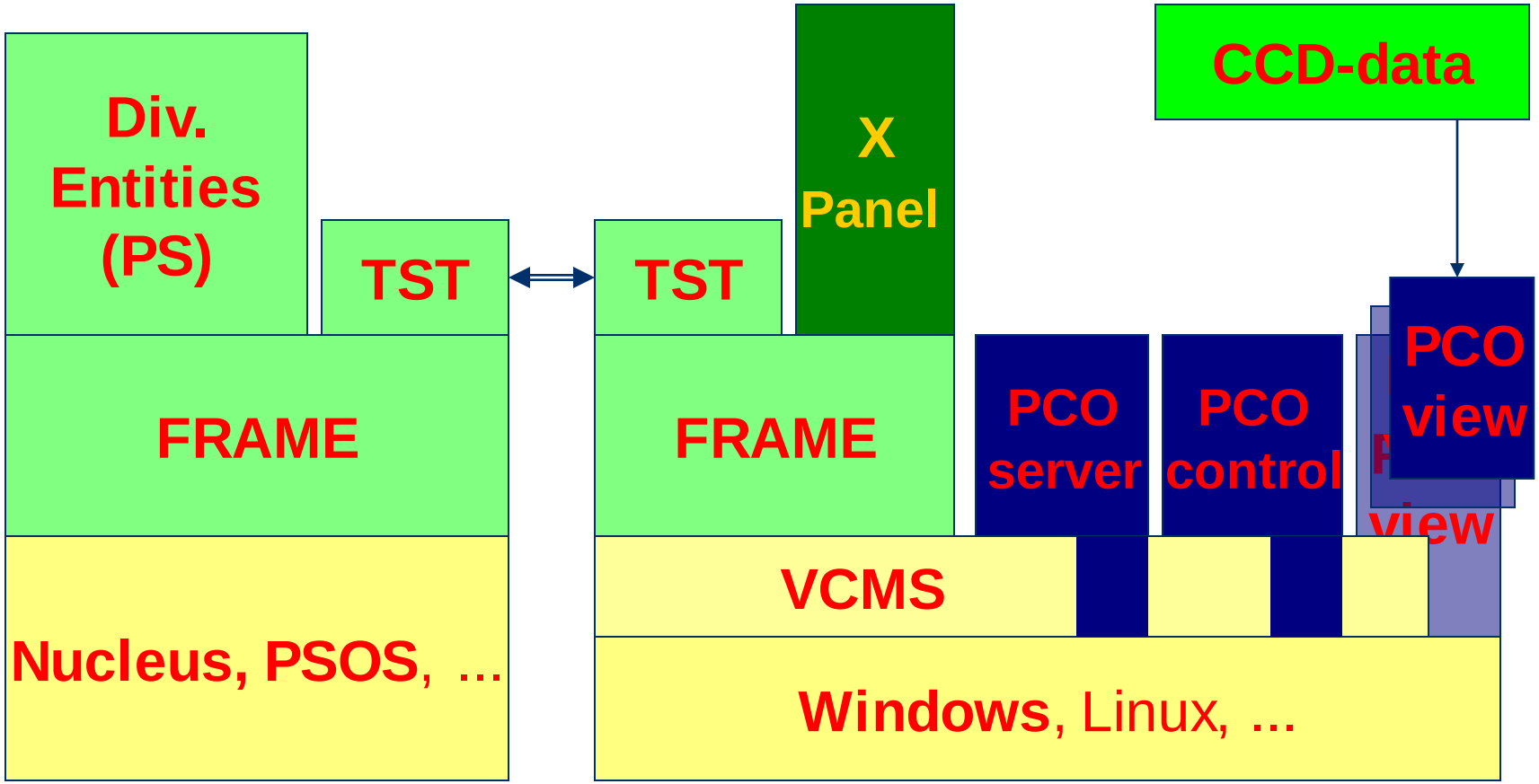
Element	Value	Cleartext/Info
⊟ MMGMM_NREG_REQ	OPC: 0x2E01	
detach_cause ..	01	Power off and d
detach_done (..	01	detach done
cause (MM or ..	FF CA	No error cause

Element	Value	Cleartext/Info
⊟ RR_ESTABLISH_REQ	OPC: 0x80040..	
estcs (establ..	00 04	service reque
⊟ U_LOC_UPD_REQ	08 00 00 00 ..	<AIR MESSAGE>
msg_type (M..	08	
⊞ loc_upd_typ..	00 02 00 00	<Sub structur
⊞ ciph key nu..	06 00 00 00	<Sub structur

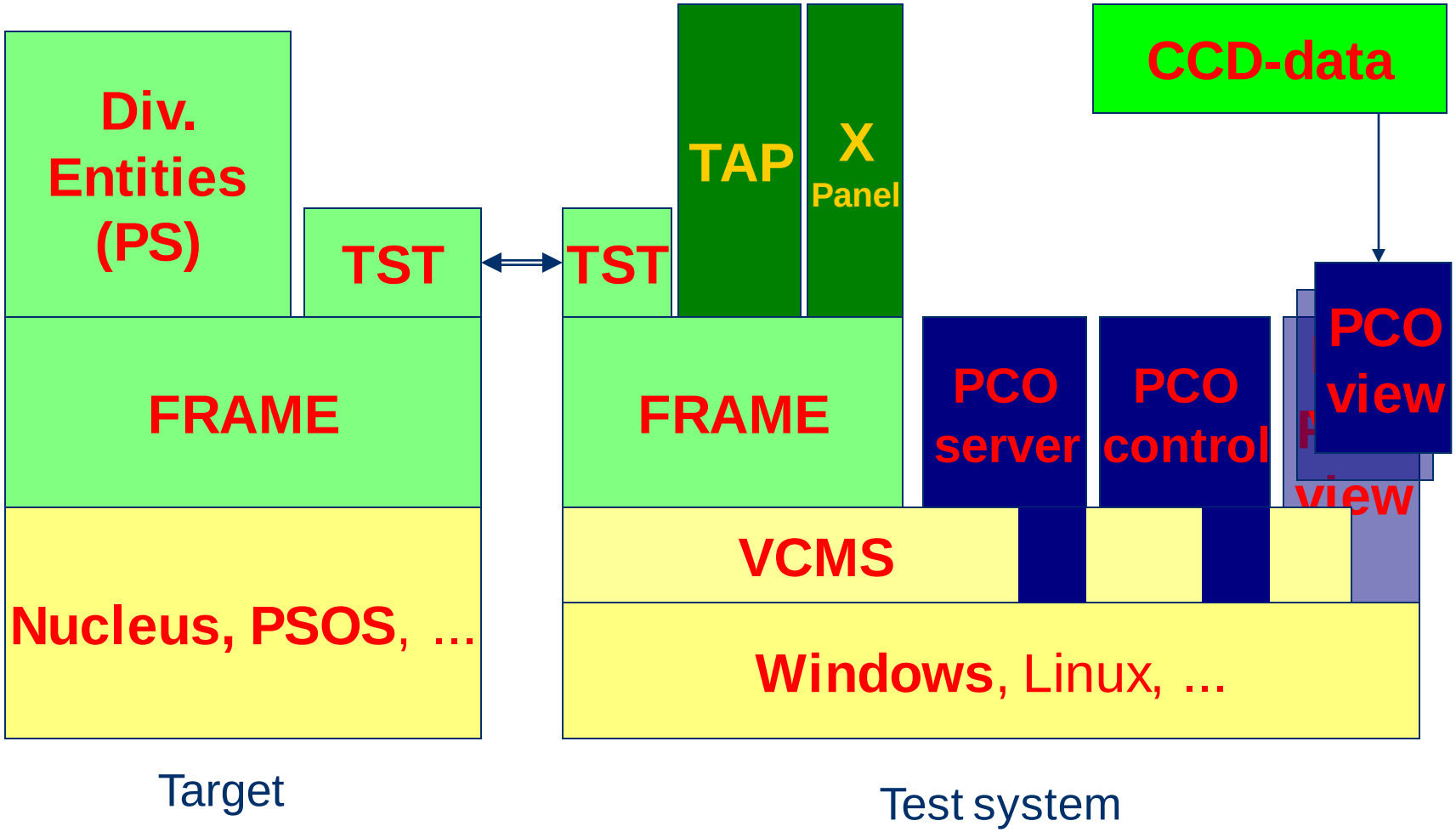
xPanel & PCO2 … *software layers*



xPanel & PCO2 … *software layers*

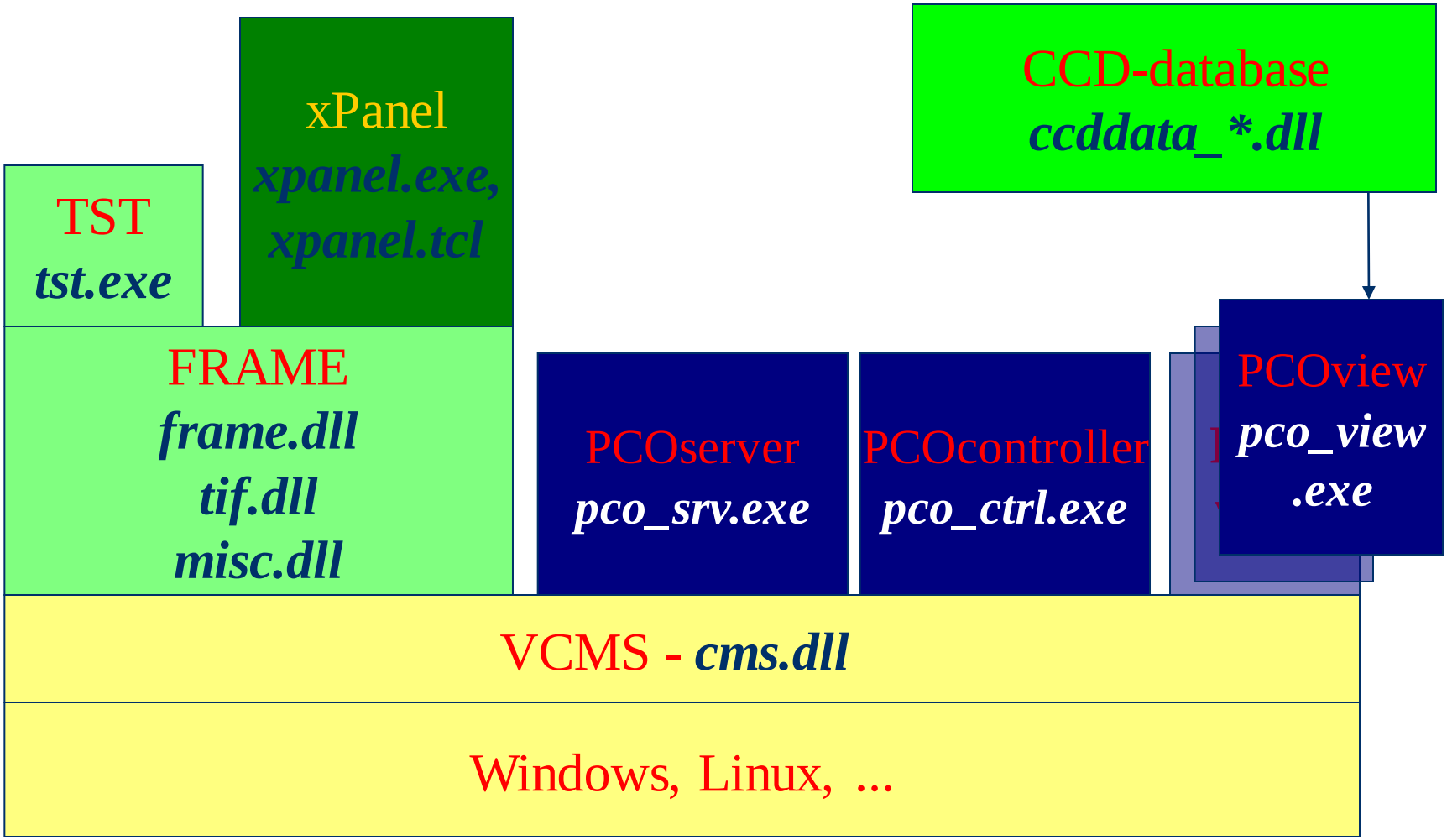


xPanel & PCO2 … *software layers*



xPanel & PCO2 … *software layers*

- Mapping to executables and DLLs:



xPanel & PCO2 … *configuration files*

- xPanel configuration files:

  ⇨ **xpan.ini**
  (ClearCase: \gpf\cfg\xpan.ini)
  - contains all settings for xPanel
  - like communication port and baudrate
    (if xpanel is used alone)

```
# directory to search layouts in
layoutdir .

# layout to use
layout    TARGET

# initial primitives to execute
#initpri  /GPF/util/xpanel/init.pri

# initial zoom level
zoom      3

# communication mode to use (REAL,SIM (USART); SOCKET)
comm-mode SOCKET

# COM-port to use
#comport  2

# flow control type (N,D,R,V)
flowctrl  N

# baudrate to use (38400, 19200, ...)
baudrate  38400
```

  ⇨ **TARGET_lo.tcl**
  (ClearCase: \gpf\cfg\TARGET_lo.tcl)
  - default layout for xPanel
  - here e.g. new keys may be added

xPanel & PCO2 … *configuration files*

- PCO configuration files (1):

  ⇨ **pco.ini**
  (ClearCase: \gpf\cfg\pco.ini)
  - contains all settings for the PCO components

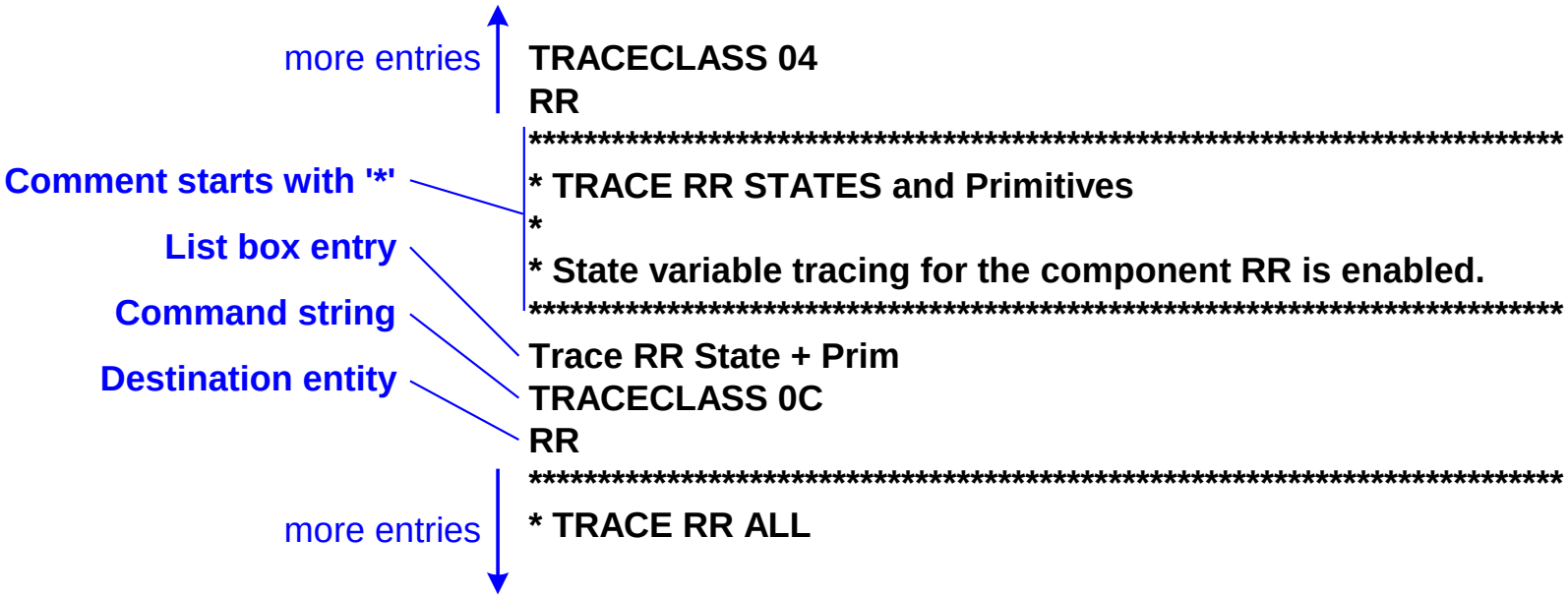
```
[General]
ServerName=PCOS
DataSize=1600

[Server]
TopMost=0
Tray=1
TestSessionPath=W:\gpf\util\pco\testsessions
DataQueueSize=23

[Controller]
Primlist=0100010000
PrimFile=View.txt
StartList=pco_start.lst
```

  ⇨ **pco_start.lst**
  (ClearCase \gpf\cfg\pco_start.lst)
  - list with applications to be started
  - like xPanel, PCO-viewers
  - editable by GUI (PCO-Controller)

```
# start GUI pco-server
pco_srv
# start standard viewer with default config
pco_view main.svc
# start TI-Multiplexer minimized
min TraceMultiplexer
# start xpanel (xpan.ini should contain SOC
ide xpanel
```

xPanel & PCO2 … *configuration files*

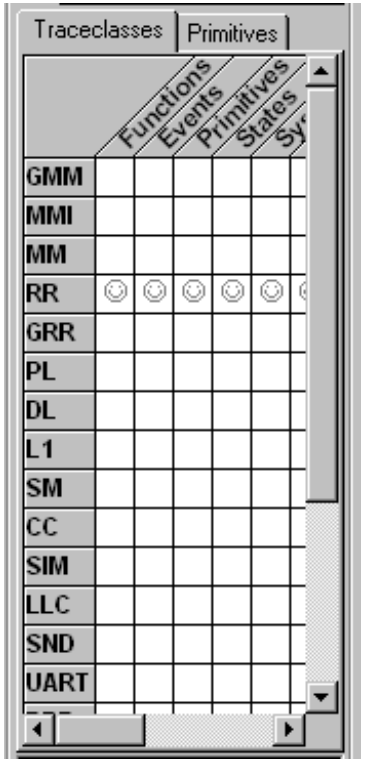
- PCO configuration files (2):
  - ⇨ **View.txt** (ClearCase: \gpf\cfg\View.txt)
    - contains predefined FRAME system primitives
      e.g. “DUPLICATE MM PCO” -> RR
      “CONFIG AT+CFUN=1” -> MMI (to send At-Commands)
    - these may be selected and sent using the PCO-Controller



xPanel & PCO2 … *configuration files*

- PCO configuration files (3):
  - ⇨ **pco_stack.xml** (ClearCase: \gpf\cfg\pco_stack.xml)
    - contains all “Matrix”-entries of the PCO-Controller
    - may be edited to e.g. change the entry-order

```
<stack>
  <entity name="GMM" traceclass="00000040"/ duplicates=""/>
  <entity name="MMI" traceclass="00000040"/ duplicates=""/>
  <entity name="MM" traceclass="00000040"/ duplicates=""/>
  <entity name="RR" traceclass="00000040"/ duplicates=""/>
  <entity name="GRR" traceclass="00000040"/ duplicates=""/>
  <entity name="SM" traceclass="00000040"/ duplicates=""/>
  <entity name="CC" traceclass="00000040"/ duplicates=""/>
  <entity name="SIM" traceclass="00000040"/ duplicates=""/>
  <entity name="LLC" traceclass="00000040"/ duplicates=""/>
  <entity name="SND" traceclass="00000040"/ duplicates=""/>
  <entity name="UART" traceclass="00000040"/ duplicates=""/>
  <entity name="PPP" traceclass="00000040"/ duplicates=""/>
  <entity name="SMS" traceclass="00000040"/ duplicates=""/>
  <entity name="SS" traceclass="00000040"/ duplicates=""/>
  <entity name="FAD" traceclass="00000040"/ duplicates=""/>
  <entity name="RLP" traceclass="00000040"/ duplicates=""/>
  <entity name="L2R" traceclass="00000040"/ duplicates=""/>
  <entity name="T30" traceclass="00000040"/ duplicates=""/>
  <entity name="L1" traceclass="00000040"/ duplicates=""/>
  <entity name="PL" traceclass="00000040"/ duplicates=""/>
  <entity name="DL" traceclass="00000040"/ duplicates=""/>
</stack>
```



Thanx for your patience !

xPanel & PCO2 … *TI internal issues*

- Maintenance and versions:
  - ⇨ xPanel and PCO are maintained by the GPF-team
  - ⇨ delivery is usually done inside the gpf_x.csi file included in the ConfigSpec of each developer
  - ⇨ local setup-packages can be obtained at \\dbgs2\deveng\cc\gpf\projects\pco\delivery\index.html
- Build of ccddata_dll.dll:
  - ⇨ automatically done while running makcdg-script
  - ⇨ default output dir: \gpf\BIN
- Build of str2ind.tab:
  - ⇨ automatically done during protocol stack build
  - ⇨ default output dir: \g23m__out__\...\trace\...
- Complaining:
  - ⇨ can be done directly by using the “moan”-button in the system tray 
  - ⇨ see \gpf\DOC\moanbtn\mbtn_userguide.doc