Running Test Cases with

TAPCaller

(an introduction)

General concept

Preparations

Usage with old TAP

Usage with TAP2 and PCO2

Important files

TAPCaller ... *general concept*

● test interface approach:

⇨ data interface between G23 protocol stack and a PC test system

⇨ for running test cases: usually shared memory on one PC

TAPCaller ... *general concept*
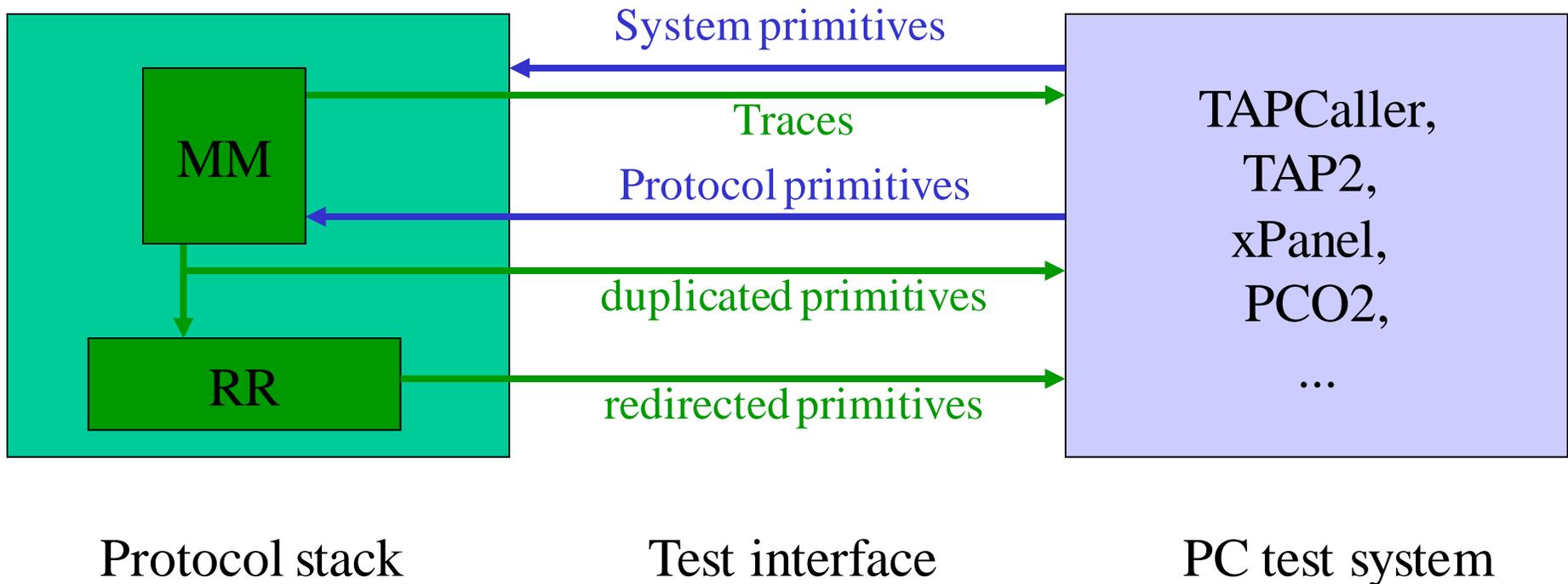
- On stack side:
  - ⇨ test interface entity included in FRAME
  - ⇨ uses corresponding hardware driver for communication

- On PC test system side:
  - ⇨ test interface executable using the FRAME
    - ◆ connects using standard OS drivers
  - ⇨ used by tools like xPanel
  - ⇨ .. and of course by TAP2 & PCO2

TAPCaller ... *general concept*

- **TAP (Test Application Process):**

  ⇨ program which executes test cases by sending, receiving and comparing primitives to/from the protocol stack

- **PCO (Point of Control and Observation):**

  ⇨ tools cabable to stimulate the protocol stack and display traces and duplicated primitives

  ⇨ used here for logging and replaying test sessions

- **TAPCaller:**

 ⇨ GUI-tool for easy handling TAP and PCO

TAPCaller ... *general concept*

- The software layers:

Div. Entities (PS)	**TST**
FRAME	
Nucwin, Nucleus, …	

↔

TST	
FRAME	
VCMS	
Windows, Linux, …	

Protocol stack   Test system
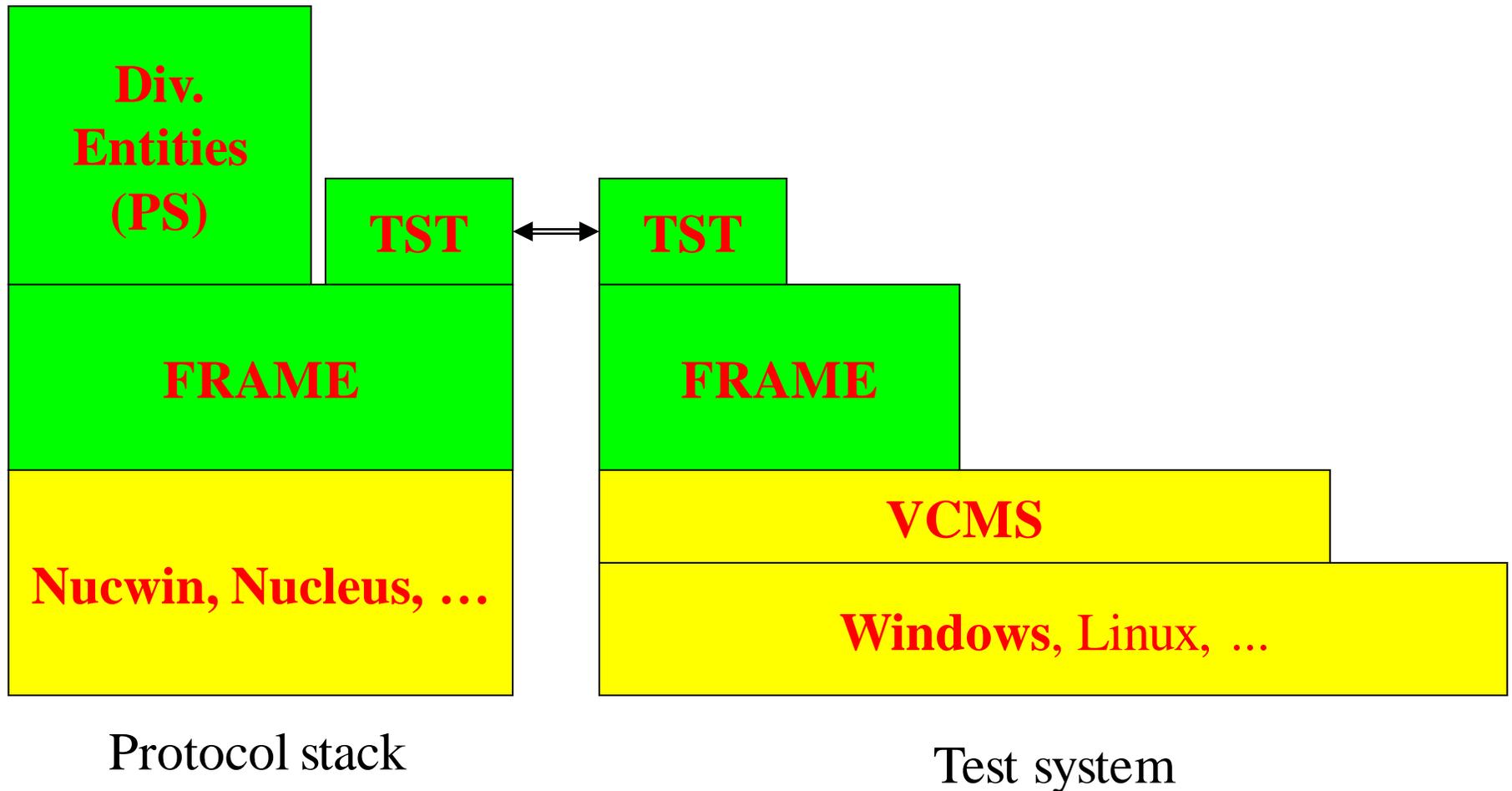
TAPCaller ... *general concept*

- The software layers:

Protocol stack

Test system

TAPCaller ... *preparations*
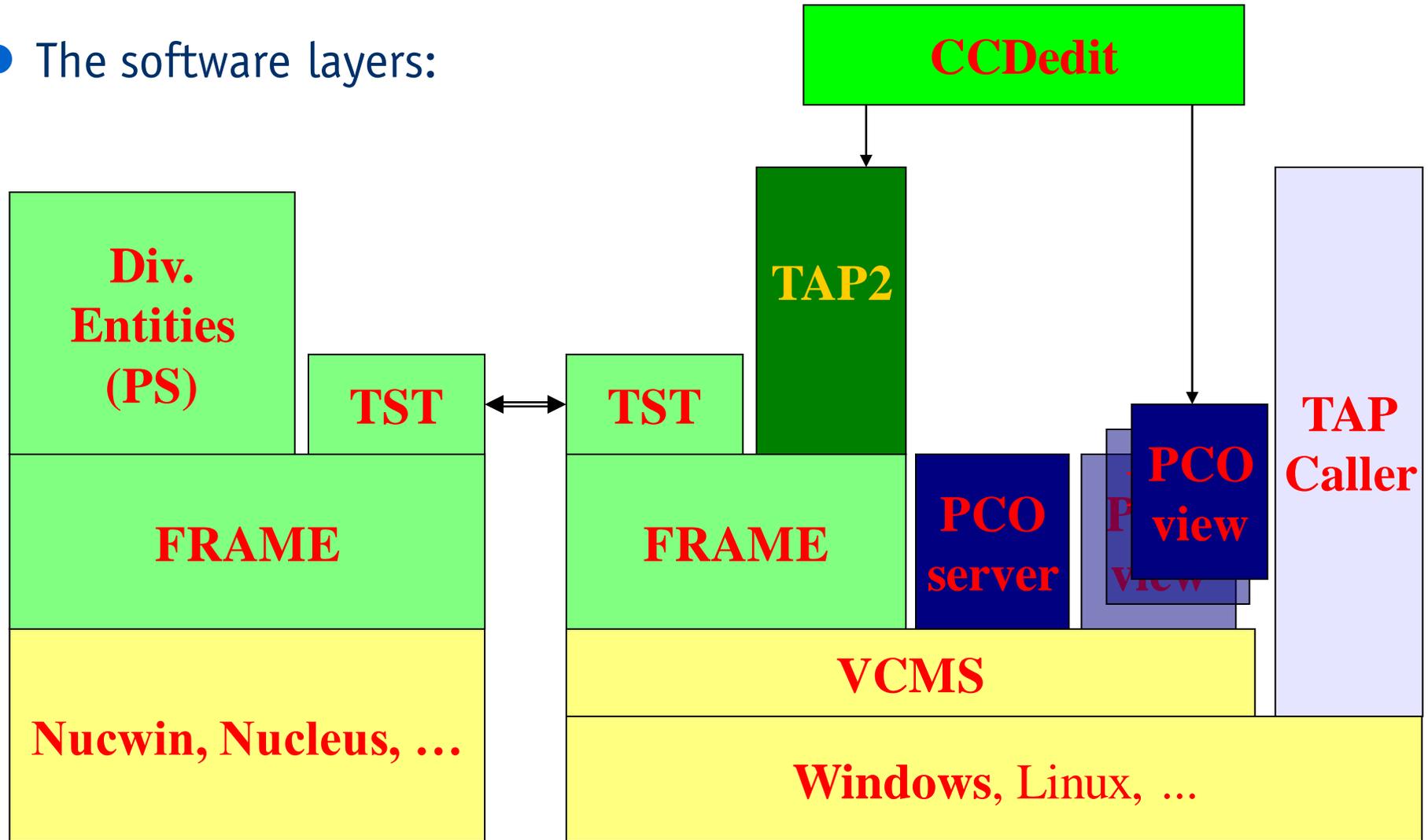
● **Build of necessary executables:**

⇨ before testing some executables have to be (re)build after each run of makcdg:

♦ the protocol stack – in any case ;-)

♦ tap2_{gprs|gsm}.exe (ClearCase: \gpf\BIN\tap2_xxx.exe)
  – old TAP for usage without PCO2
  – call gnumake in \gpf\TAP

♦ ccddata_dll.dll (ClearCase: \gpf\BIN\ccddata_dll.dll)
  – used by new TAP2 (tap2.exe) and PCO2
  – automatically build by makcdg-script

© Texas Instruments, Berlin 2003

TAPCaller ... *preparations*

- Optional modification of test cases:

  ⇨ To receive not only traces but also duplicated primitives while testing ...

  ⇨ ... extra system primitives have to be inserted into the test document (e.g. l2r_steps.cpp)

```
                    ...
    COMMAND ("L2R REDIRECT MMI TAP")
          |        |         |
    COMMAND ("TAP REDIRECT TAP L2R")
                    ...
```

→

```
                    ...
    COMMAND ("L2R DUPLICATE MMI PCO")
    COMMAND ("L2R REDIRECT MMI TAP")
          |        |          |
    COMMAND ("TAP DUPLICATE TAP PCO")
    COMMAND ("TAP REDIRECT TAP L2R")

                    ...
```

TAPCaller ... *preparations*

● Build of test cases:

⇨ Generators
  ◆ doc2txt
  ◆ xgen
  ◆ ccdgen

⇨ Compilers
  ◆ msdev / cl
  ◆ (mktc)

GRR.DOC	"makcdg tdclib"	p_grr_val.h
SAP Document		Primitive / Message Constants and Structs
GRR.dsp	Compilers Linker	**Testcase DLL**
TDC Testcases		
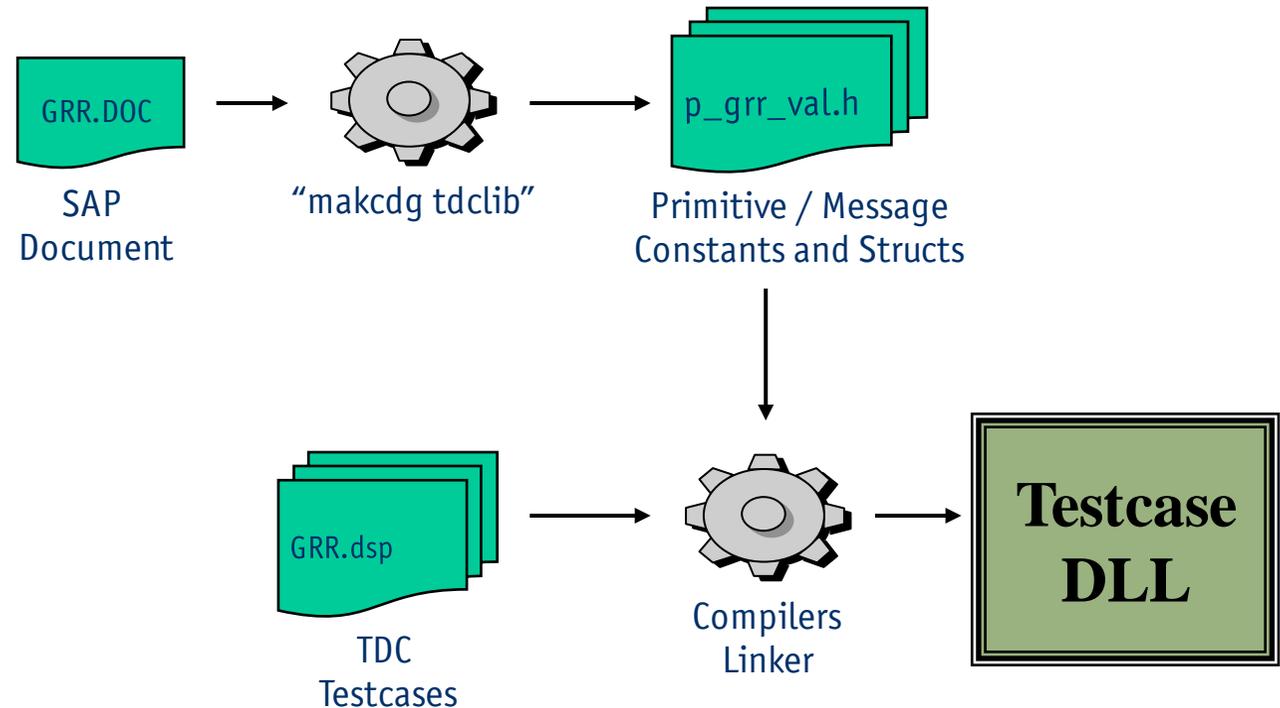
⇨ For more informations see:
  ◆ protocol_stack_testing.ppt
  ◆ 8415_028.doc
   (ClearCase: \GPF\DOC\8415_028.doc)

TAPCaller ... *preparations*

● Selection of test cases:

⇨ Start tapcaller.bat

(ClearCase:
\gpf\BIN\tapcaller.bat)

⇨ choose directory

⇨ enable testcases
   from DLL(s)
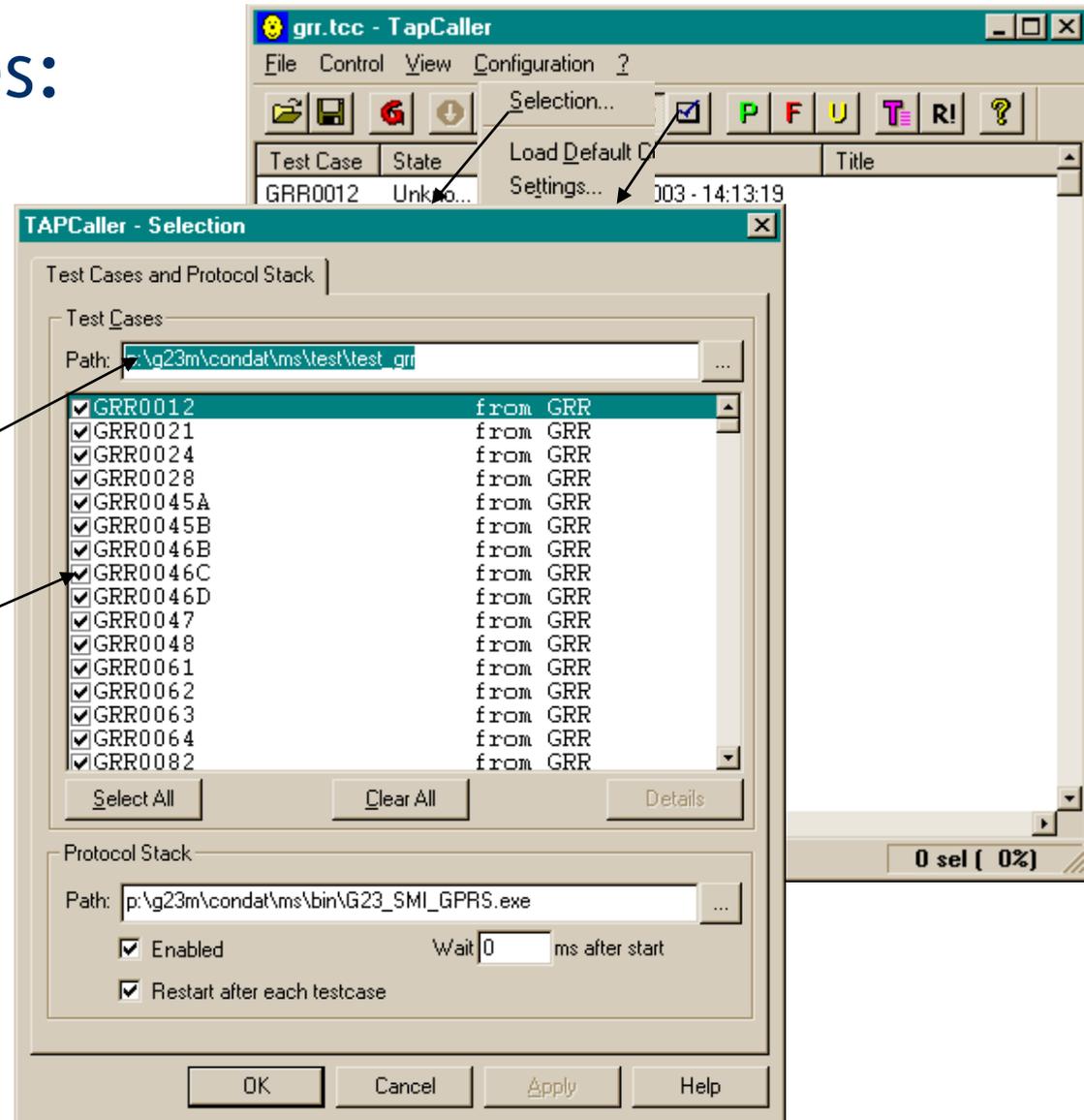
# TAPCaller ... *preparations*

● Protocol stack (PS) selection:

⇨ Start a protocol stack by your own, e.g. inside VisualStudio ...

⇨ ... or specify an executable ...

⇨ ... and enable automatic start of PS (before each test case execution)

TEXAS INSTRUMENTS

- Data flow:

RR
MM

Trace windows

Protocol Stack

TST

primitives

traces

Testcase DLL

TAP2

TST

Start
Stop

TAP Caller

.dbg files

Protocol stack

Test system
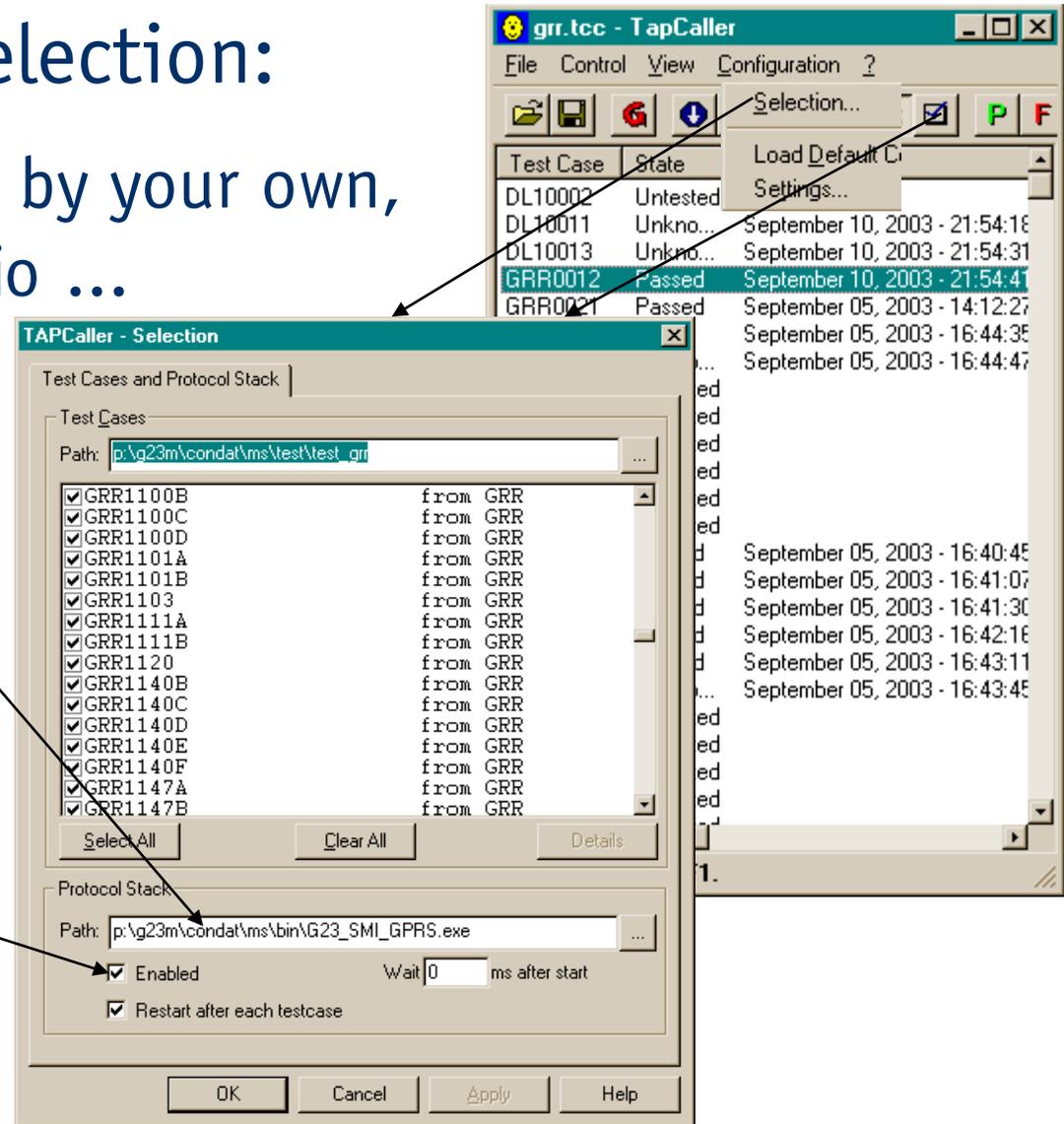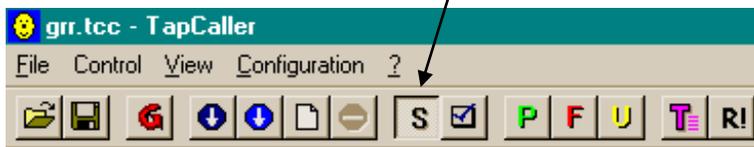
TAPCaller ... *usage with old TAP*

- Snapshot:

primitives

traces

TAP2
tap2_gprs.exe

Start
Stop

Protocol stack

Test system

TAPCaller ... *usage with old TAP*

- Configuration:

  ⇨ select TAP executable which has to be rebuild after each makcdg call ...

  ⇨ ... and choose „old TAP"-mode

  ⇨ evtl. modify communication type, if testing with a real target
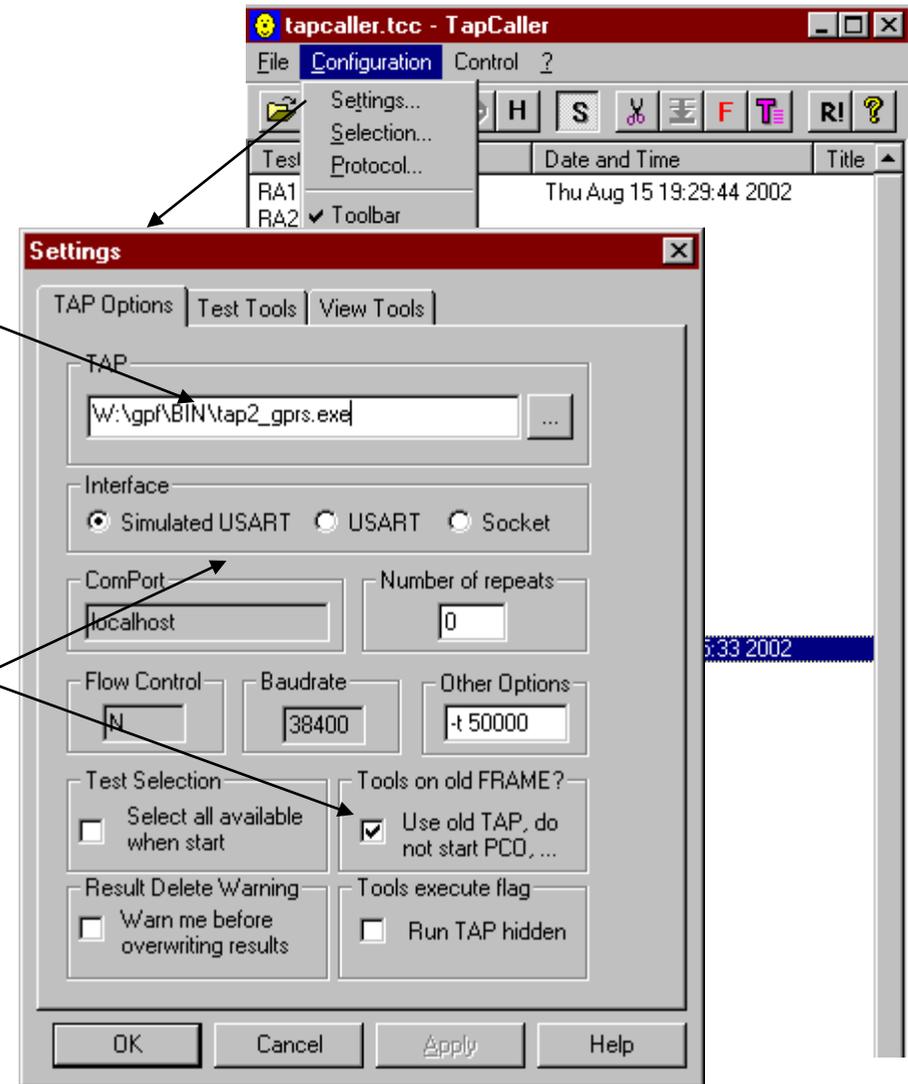
TAPCaller ... *usage with TAP2 & PCO2*

- Data flow:

CCDedit

PCO viewers

TAP Caller

decode interprete

Testcase DLL

TAP2

start stop

Protocol Stack

TST

duplicated primitives, traces

.dbg files

control

primitives

primitives

TST

duplicated primitives

PCO server

PCO logfile

traces

traces

Protocol stack

Test system

TAPCaller ... *usage with TAP2 & PCO2*

- Snapshot:

PCO-Viewer

⇨ testcase running

TAP2
tap2.exe

TST
tst.exe
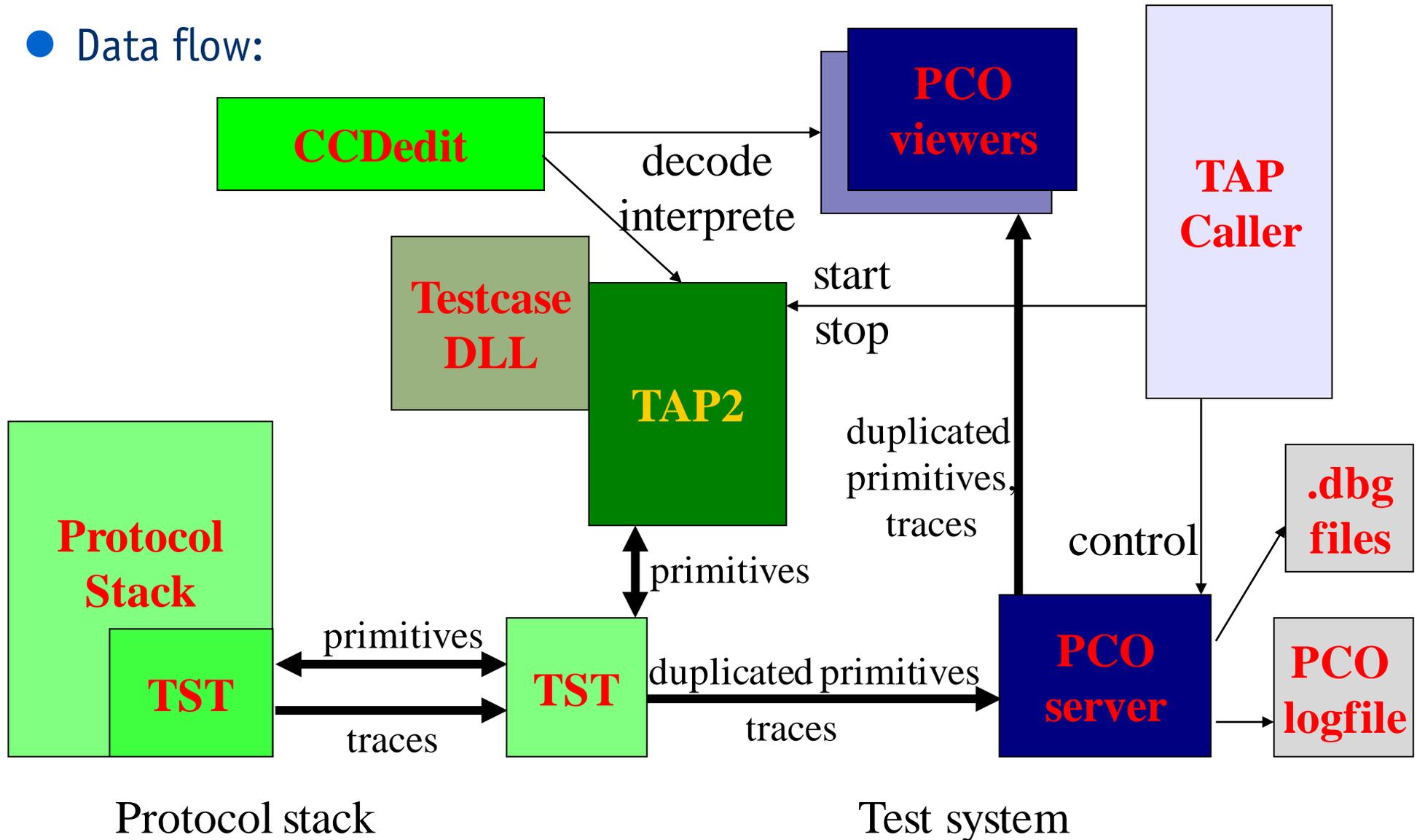
primitives

traces
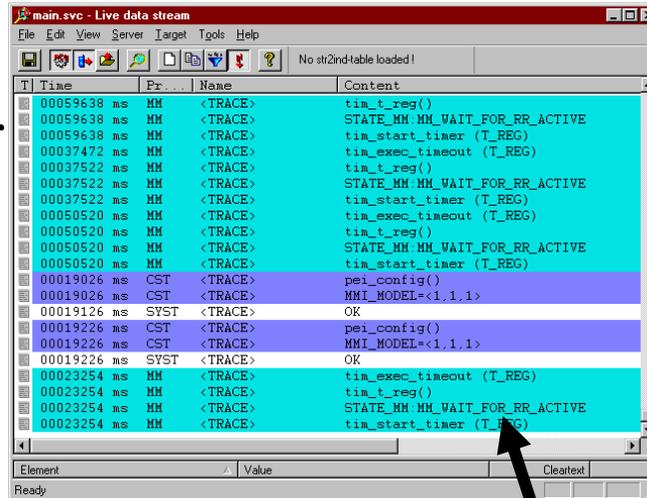
Protocol stack

start
stop
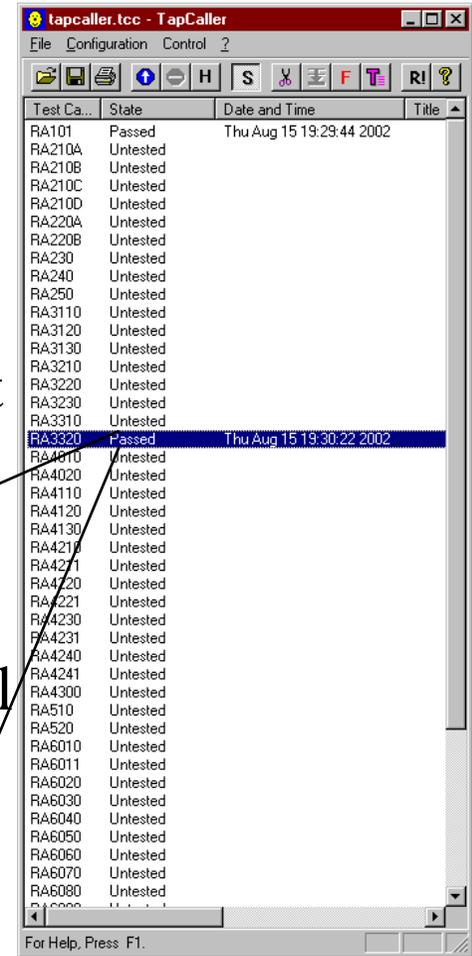
control
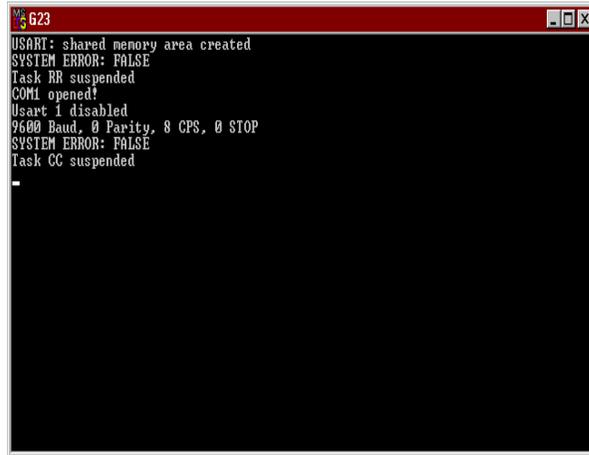
traces,
duplicated
primitives
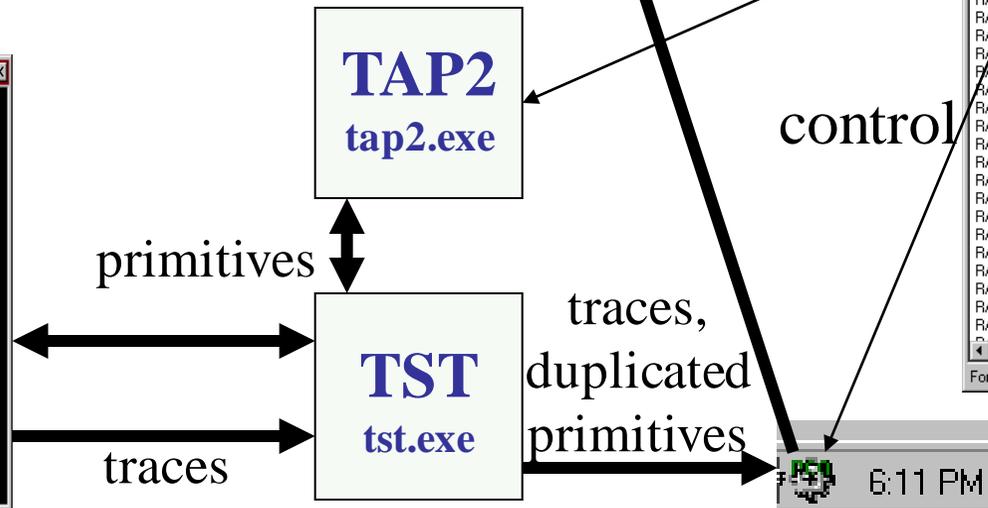
PCO-Server

Test system

TAPCaller ... *usage with TAP2 & PCO2*

● Snapshot:

⇨ testcase replay
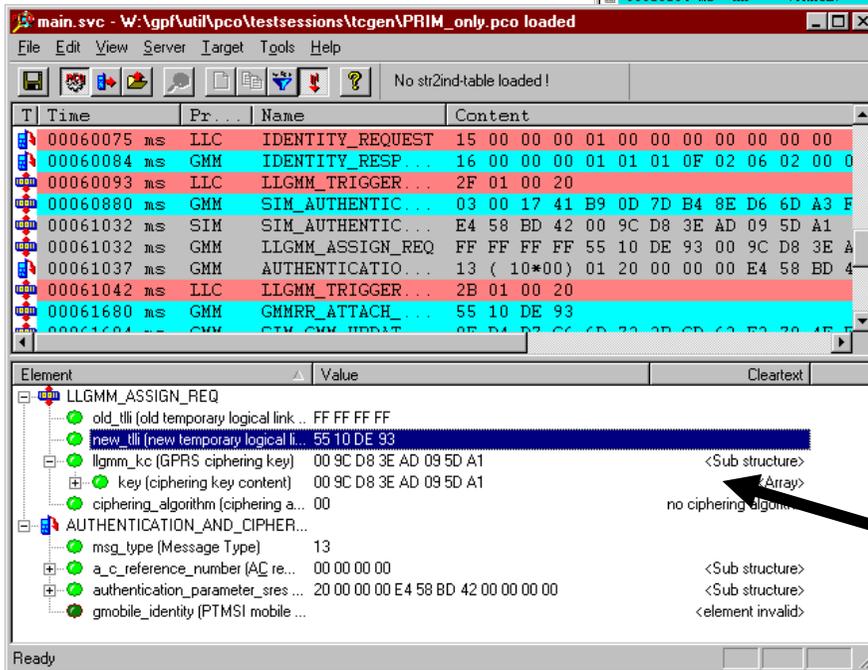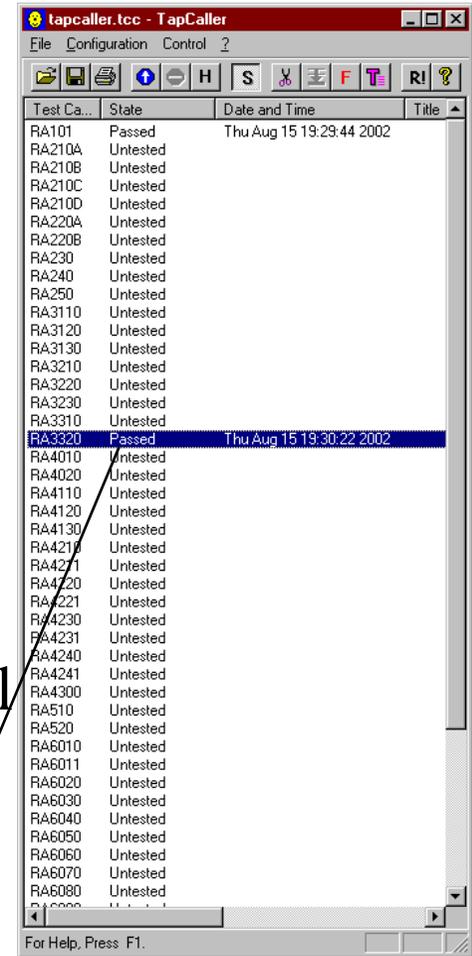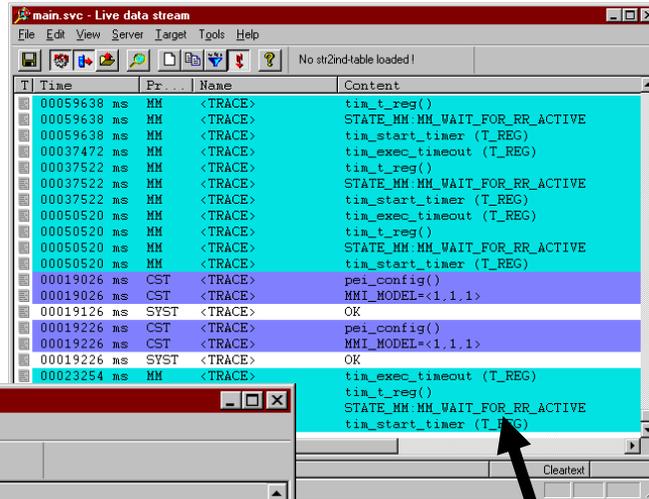
PCO-
Viewers

control

traces,
duplicated
primitives

PCO-
Server

Test system

TAPCaller ... *usage with TAP2 & PCO2*

● Configuration (1):

⇨ select path to tap2.exe ...

⇨ ... and deselect „old TAP"-mode

⇨ evtl. modify communication type, if testing with a real target
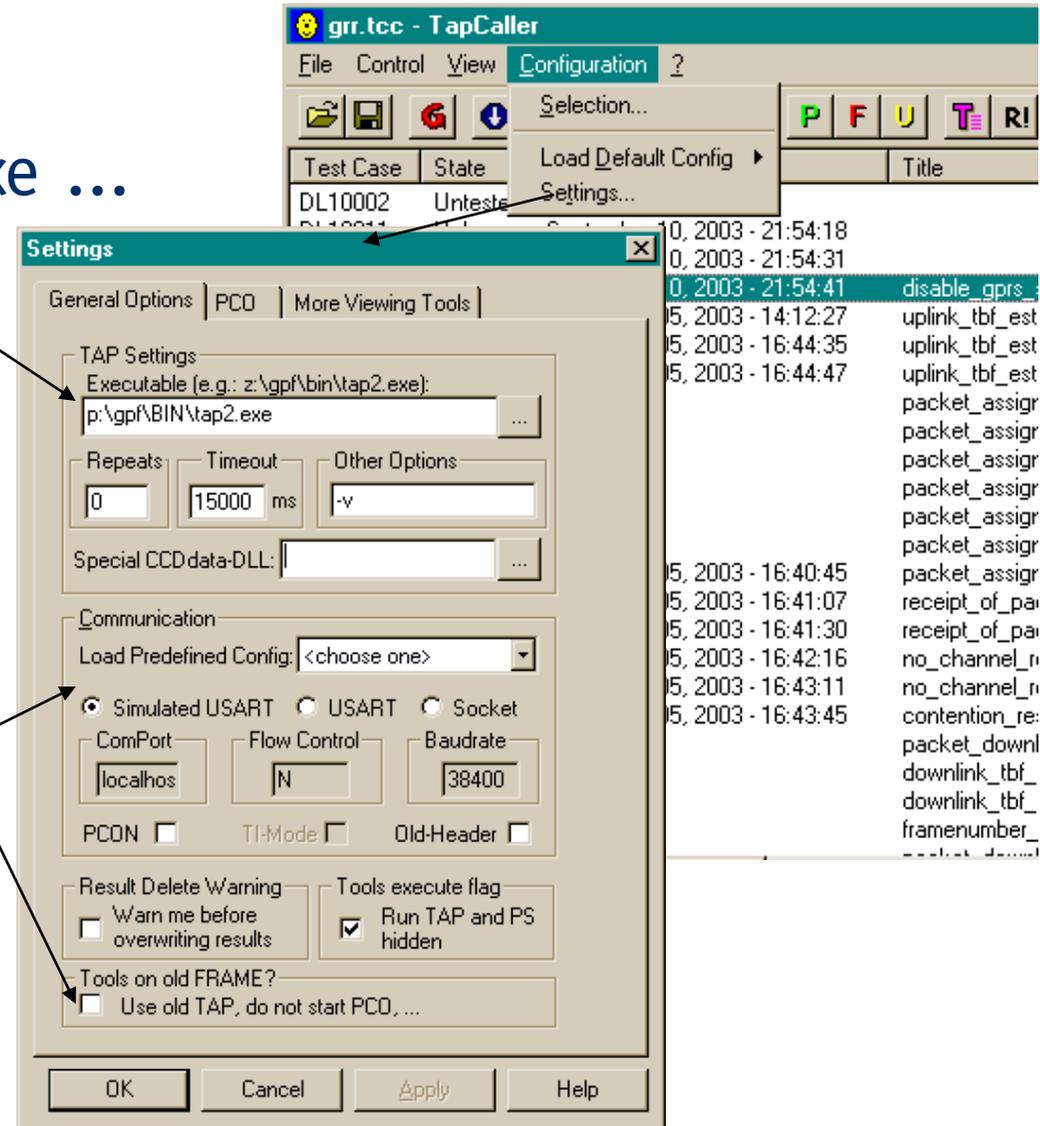
TAPCaller ... *usage with TAP2 & PCO2*

- Configuration (2):

  ⇨ enable logging with PCO ...

  ⇨ ... and choose path to PCO executables (should be the same as for tap2.exe)

  ⇨ evtl. select a viewer configuration, with which a PCO-Viewer should be started while running testcases
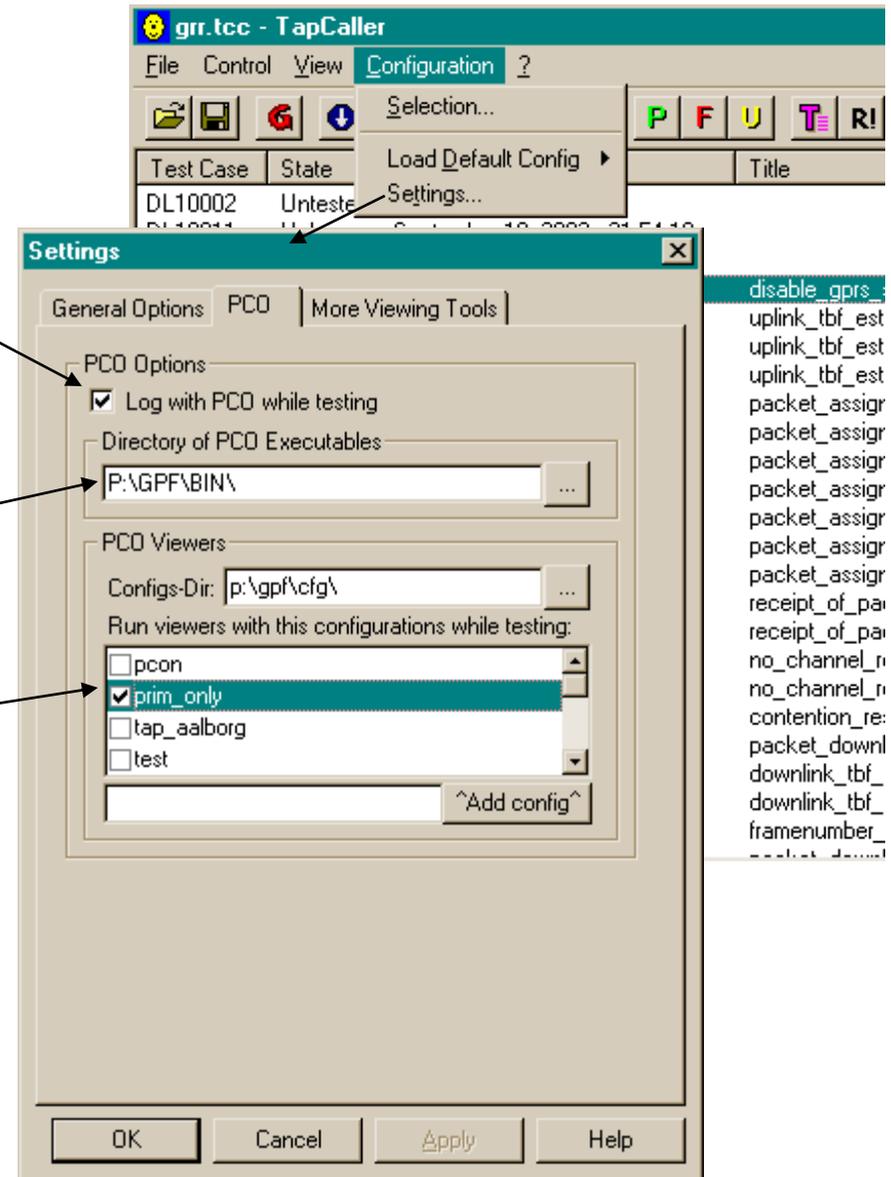
TAPCaller ... *important files*

- Executables and DLLs:

CCDedit
ccddata_dll.dll, *ccddata_load.dll*

TAP2
tap2.exe
(tap2_gprs.exe)

TST
tst.exe

FRAME
frame_wn.dll
tif_wn.dll
misc_wn.dll

PCOserver
pco_srv.exe

PCOview
pco_view
.exe

TAP Caller
tapcaller.exe

VCMS - *cms.dll*

Windows, Linux, ...
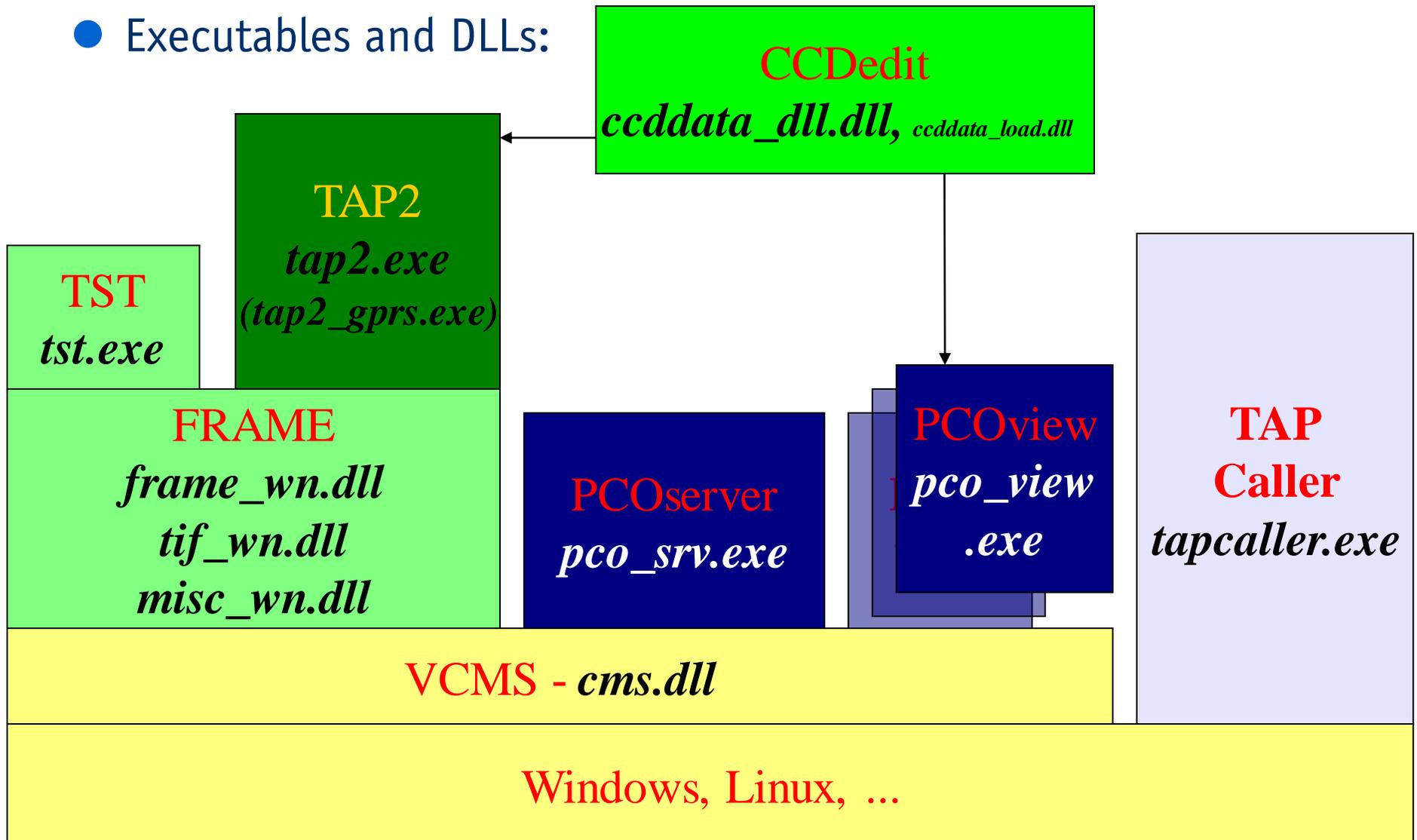
TAPCaller ... *important files*

- ● PCO configuration files:

  - ⇨ **pco.ini**
    (ClearCase: \gpf\BIN\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
```

  - ⇨ ***.svc-files**
    - ◆ **Standard-Viewer-Configuration files**
    - ◆ **tap.svc.default**
      -> default viewer config, will be copied to tap.svc by tapcaller.bat

- ● TAPCaller configuration files:

  - ⇨ ***.tcc-files**
    - ◆ **TAP-Caller-Configuration files**

Confidential

TAPCaller ... *important files*

- Volatile files:
  - ⇨ have to be build together with the used protocol stack
    - ◆ **ccddata_dll.dll** (table with primitive symbols)
    - ◆ **str2ind.tab** (table with trace texts - introduced with GPRS_1.3.2)
- Detailed documentations:
  - ⇨ **tapcaller_userguide.doc**
    (ClearCase: \gpf\DOC\tapcaller\tapcaller_userguide.doc)
  - ⇨ **pco_userguide.doc**
    (ClearCase: \gpf\DOC\pco\pco_userguide.doc)
  - ⇨ **frame_users_guide.doc**
    (ClearCase: \gpf\DOC\frame_users_guide.doc)
  - ⇨ **testplatform_userguide.doc**
    (ClearCase: \gpf\DOC\testplatform_usersguide.doc)
  - ⇨ **tapcaller_intro.ppt** (this document)
    (ClearCase: \gpf\DOC\tapcaller\tapcaller_intro.ppt)

TAPCaller ... *TI internal issues*

- Maintenance and versions:
  - ⇨ TAPCaller, TAP and PCO are maintained by the GPF-team
  - ⇨ delivery is usually done inside the gpf.csi file included in the ConfigSpec of each developer
  - ⇨ the very newest version can be obtained by including gpf_testtools.csi right after "mkbranch xxx" (at least before gpf.csi) in the ConfigSpec

- Complaining:
  - ⇨ can be done directly by using the "moan"-button in the system tray
  - ⇨ see \gpf\DOC\moanbtn\mbtn_userguide.doc

5:38 PM