
TTCN-3 User Conference 2005

# A Domain-embracing System Test Tool

**Winfried Bock, Siemens AG, Paderborn, Germany**
**Alexander Kraas, Fraunhofer ESK, Munich, Germany**

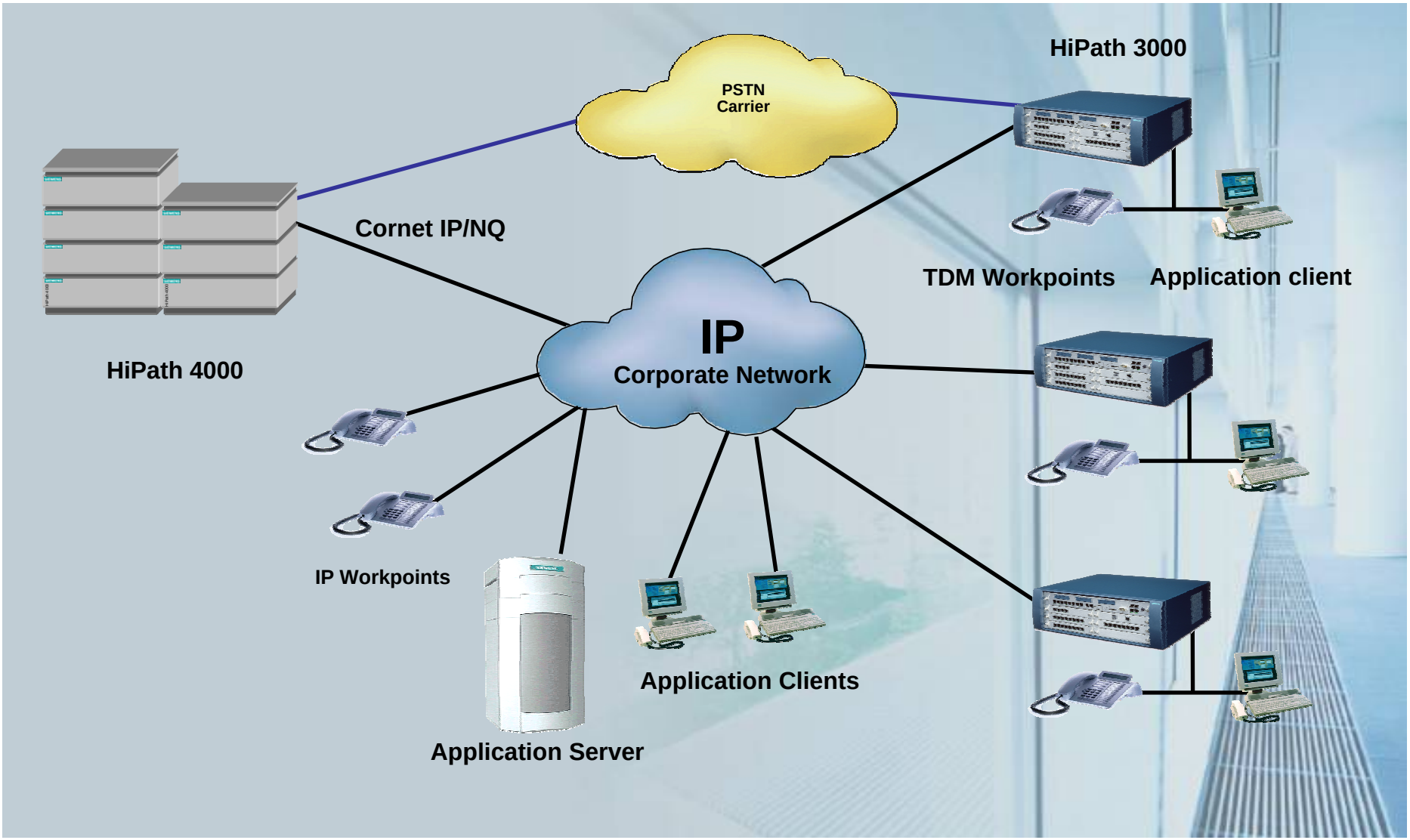
# Motivation

- **Need of more test automation during system test because of**
  - rapidly increasing complexity of SW systems
  - the need of regression tests due to incremental SW development
  - the requirements to increase product quality
  - and to decrease time to customer and costs, even of testing
- **Under this background I like to describe the realized test suite, that combines the testing of different system interfaces.**
- **The test suite acts like an umbrella tool for functional tests at user and protocol interfaces and even for non functional related tests.**

# Outline

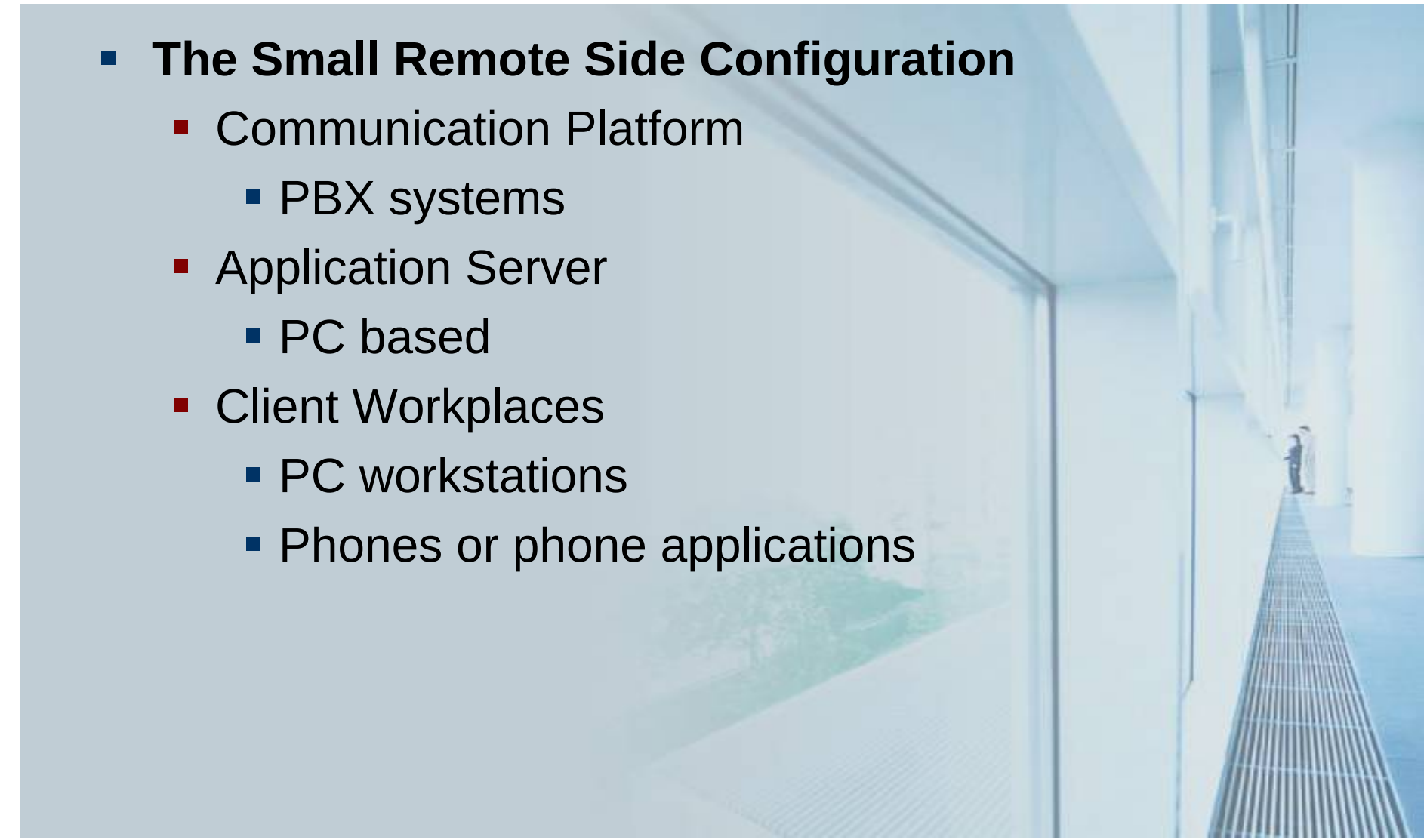
- **System Test Task:**
  - Landscape HiPath Small Remote Side Configuration
  - Multi-domain Application Overview
- **Types of Testing at System Test Level**
- **Approach to Automated System Testing**
- **Requirements to a Multi-Domain Test Tool**
- **One Solution: A TTCN-3 Test Suite for CSTA XML**
  - Realization of the TTCN-3 Test Suite
  - Overview of the TTCN-3 Test Architecture
  - Example: Snippet of the MakeCall Test Case
- **Advantages of the TTCN-3 Test Suite**
- **Summary**

# Landscape HiPath Small Remote Side Configuration



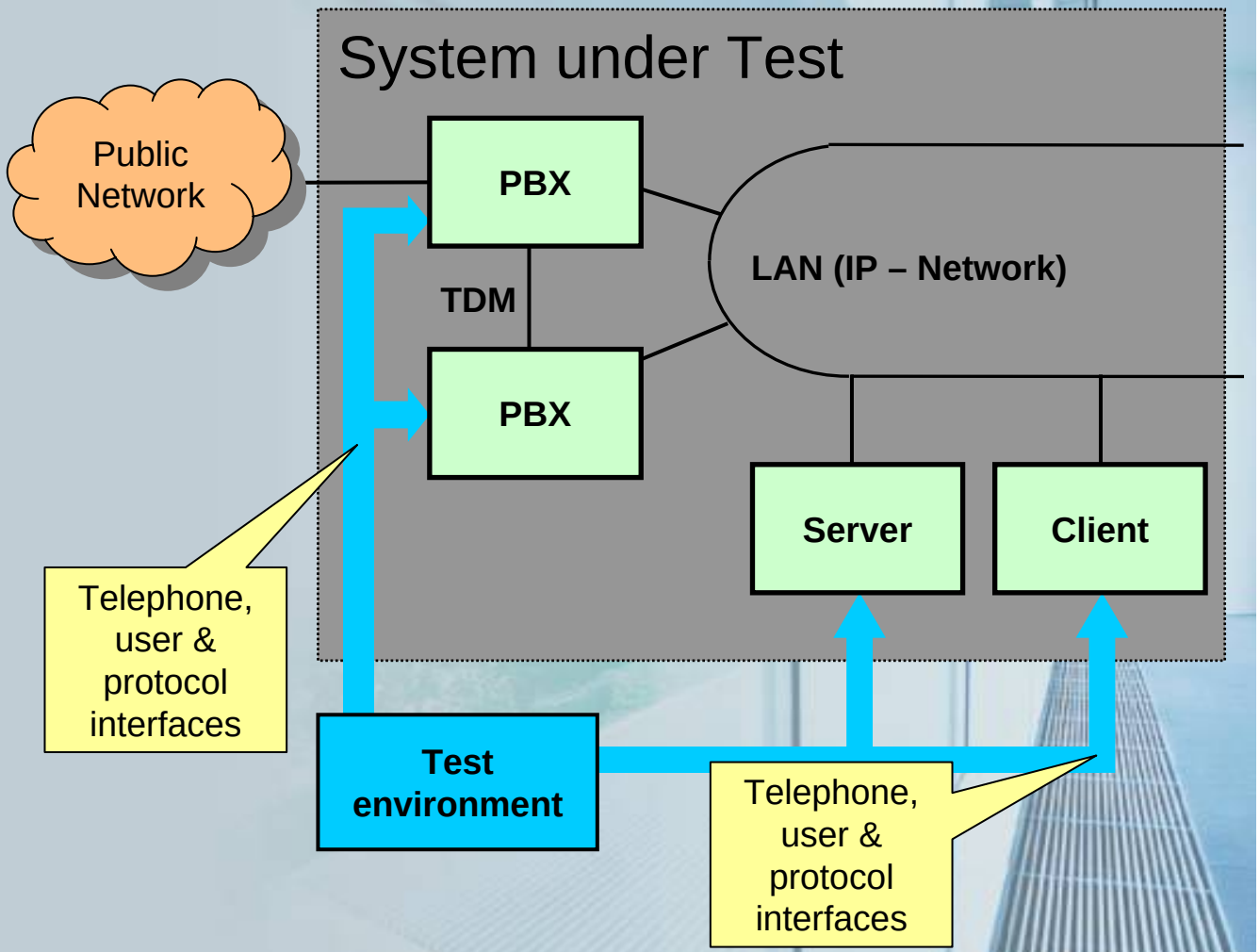
# Multi-Domain Application Overview

- **The Small Remote Side Configuration**
  - Communication Platform
    - PBX systems
  - Application Server
    - PC based
  - Client Workplaces
    - PC workstations
    - Phones or phone applications



# Types of Testing at System Test Level

- Functional tests at user interfaces
- Functional tests at protocol interfaces
- Non functional-related tests (e.g. load and stress tests)



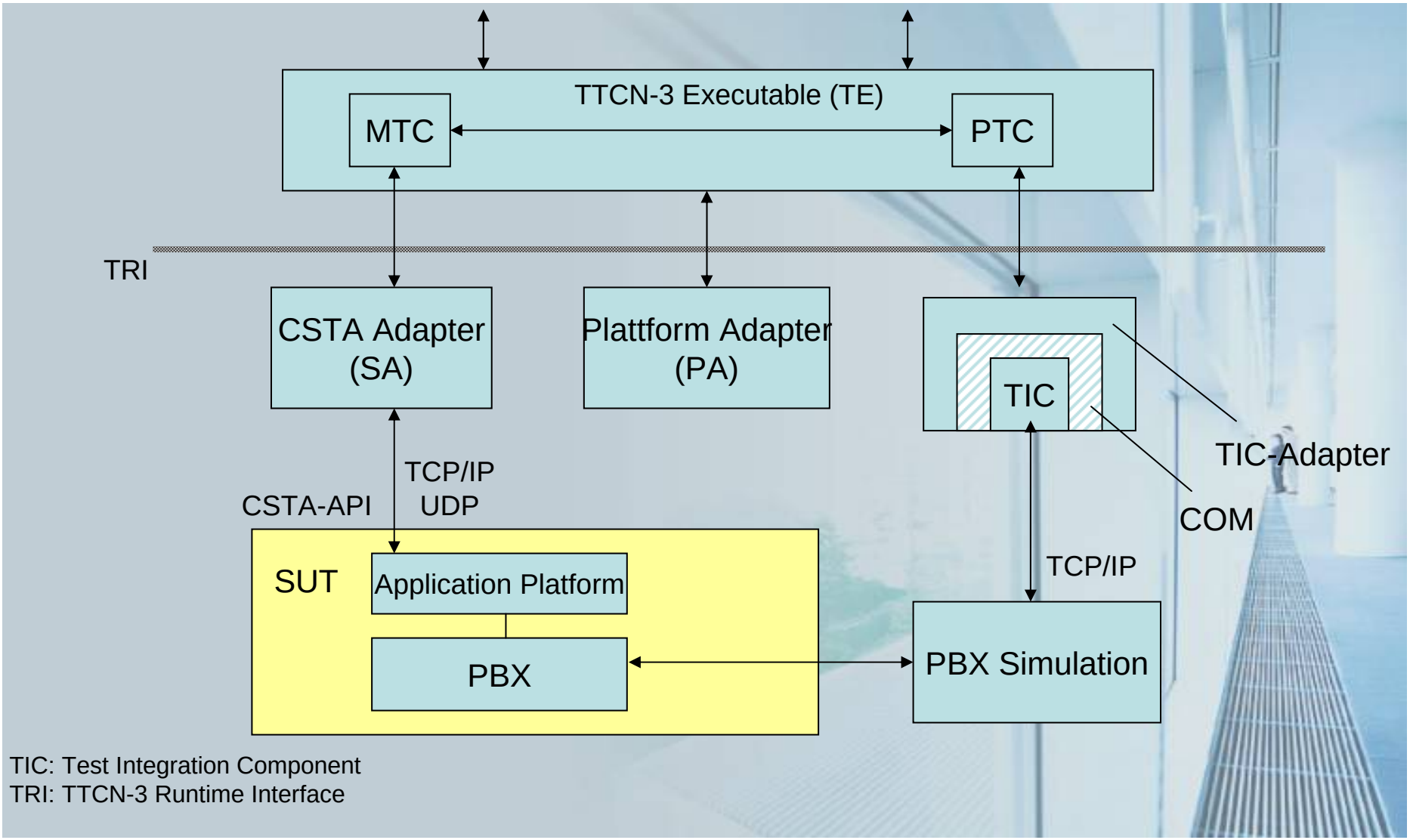
# Approach to Automated System Testing

- **Automate protocol interfaces with a protocol test equipment**
  - Generate and send protocol messages
  - Receive and evaluate protocol messages
- **Automate phones with a phone simulator**
  - Generate outgoing calls
  - Answer incoming calls
- **Create test suite in a test management system**
  - Control clients & server at the protocol interface level
  - Control phone simulator

# Requirements to a Multi-domain Test Tool

- **Integration of different test tools and interfaces into a single test suite**
  - Test evaluation based on a single test log with a defined test verdict
- **Common test platform for all test engineers**
  - Common basic test tool GUI
  - Common basic script language
- **Coordinated and synchronized test execution at different interfaces of the complete SUT**
  - Single test scripts that contain test commands for the effected test tools
  - Test commands are executed in a coordinated, defined order
- **Flexible test environment**
  - Test monitor tools are “plugged” into the test execution environment as required

# One Solution: A TTCN-3 Test Suite for CSTA XML




TIC: Test Integration Component
TRI: TTCN-3 Runtime Interface

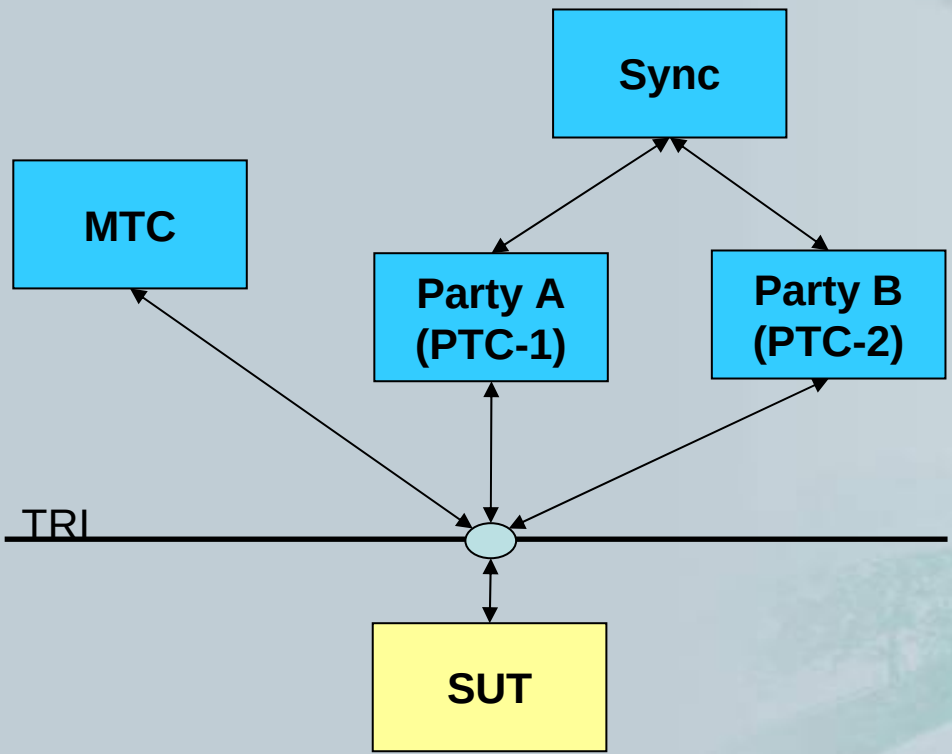
# Realization of the TTCN-3 Test Suite

- **Test suite covers multi-domain scenario**
  - Controls the CSTA XML interface of an application server and
  - Controls the PBX phone simulation via TIC
- **To realize this, two adapters were needed**
  - CSTA Adapter to link the CSTA protocol interface
  - TIC Adapter to link the PBX phone simulator
- **Furthermore  a codec to support message data in XML format is needed as well**
- **IBM Rational Real Time Test is the test management tool with the DANET tool box plug-in, used as the TTCN-3 runtime environment**

# Overview of the TTCN-3 Test Architecture

**Example:**

**Test Case for the “MakeCall” service**




**Test setup:**

- MTC establishes a connection to the SUT by using ACSE.
- Parties A and B are created by the MTC for test execution.
- PTC Sync is responsible for synchronization purposes.

**Test purpose:**
Execute the “MakeCall” service and verify the associated CSTA events.

**Test behavior:**
Party A uses the “MakeCall” service to initiates a call to Party B.

# Example: MakeCall Service Test

- **Used TTCN-3 Features**
  - **Parameterized templates for CSTA messages and events**
    - Parameterization of CSTA ‘mandatory’ fields
    - All other CSTA fields are set to ‘optional’
  - **Matching rules (*, ?) for some fields in a CSTA Service Response message and CSTA Events**
  - **Specification of TTCN-3 functions for common operations on CSTA Services and Events**
    - E.g. calculation of the CSTA ConnectionID from a message
- **Benefits**
  - **High degree of reuse of test data through the usage of parameterized templates**
  - **80 functions covers 50 CSTA Services and 30 CSTA Events needed**

# Sample Template for “MakeCall” Request

```
template MakeCall makeCallDefault
    (in DeviceID sender_, in DeviceID receiver_) :=
    {
     callingDevice := sender_,
      calledDirectoryNumber := receiver_,
      accountCode := omit,
      authCode := omit,
      autoOriginate := omit,
      correlatorData := omit,
      userData := omit,
      callCharacteristics := omit,
      mediaCallCharacteristics := omit,
      callingConnectionInfo := omit,
      subjectOfCall := omit,
      languagePreferences := omit,
      extensions := omit
    }
```

The parameters of the MakeCall template. These can be set dynamically during the test runtime.

All other values of the template are omitted, because these are ‘optional’ CSTA fields

# Sample Snippet of the “MakeCall” Test Case

```
xmlDeviceMonitorStart (PartyNameA, monitorId);
sync(); // 1.Sync()

xmlMakeCall (makeCall_doNotPrompt)
(
   PartyNameA,
   PartyNumberB
),connectionId_A);

xmlServiceInitiatedEvent (serviceInitiatedEventDefault
(
   monitorId,
   connectionId_A,
   PartyNumberA,
   (makeCall,newCall)
),connectionId_A);
...
ticCheckState_ConnlParams_Handfree(PBXSwitchA,PartyNumberB);
```

Initial Synchronization of all parties (PTCs) before the test starts.

Invocation of the make-Call function (hands-free mode) with a parameterized template. As result we get the relevant **ConnectionID**

Check of an CSTA Event with all ‘mandatory’ parameters.

Allowing two possibilities for the CSTA Event **cause.**

Check of phone state with correct party B number in the display

# Advantages of the TTCN-3 Test Suite

- Support of testing distributed and complex real-time solutions
- Single use of TTCN-3 as the only test script language for all test interfaces
- Reduced complexity in comparison to an earlier command line approach that used its own scripting language
- Development of test cases is faster and less error-prone
- Fast setup of new test projects due to reuse of test adaptors and codec and large parts of the TTCN-3 test suite
- Reduced training efforts, only one test script language to be learned

# Summary

- Integration of communication protocol interfaces and application programming interfaces (APIs) into a single test tool possible
- TTCN-3 lets the test engineer focus on black-box tests at the system's interfaces, i.e. no distraction from system internals
- Even load and stress testing is doable with little additional efforts
- TTCN-3 offers the right abstraction level to test distributed complex scenarios
- Implementation of codecs, e.g. for XML or TAPI messages, is laborious and error-prone

# Credit

**The TTCN-3 CSTA Test Suite project was realized by the Siemens COM Enterprise System Test Department HD 31 together with the Fraunhofer ESK coached by Siemens Corporate Technology**