Server-Based High-Level API

A Modern Solution for Testing and Integration

The server-based API ensures simple and standardized integration via TCP/IP connections, supporting a variety of programming languages such as Python, QT, NI LabVIEW/ TestStand, and C/C++. It can also be easily incorporated into existing test sequencer software, enabling broad compatibility with different development environments.

Key Features:

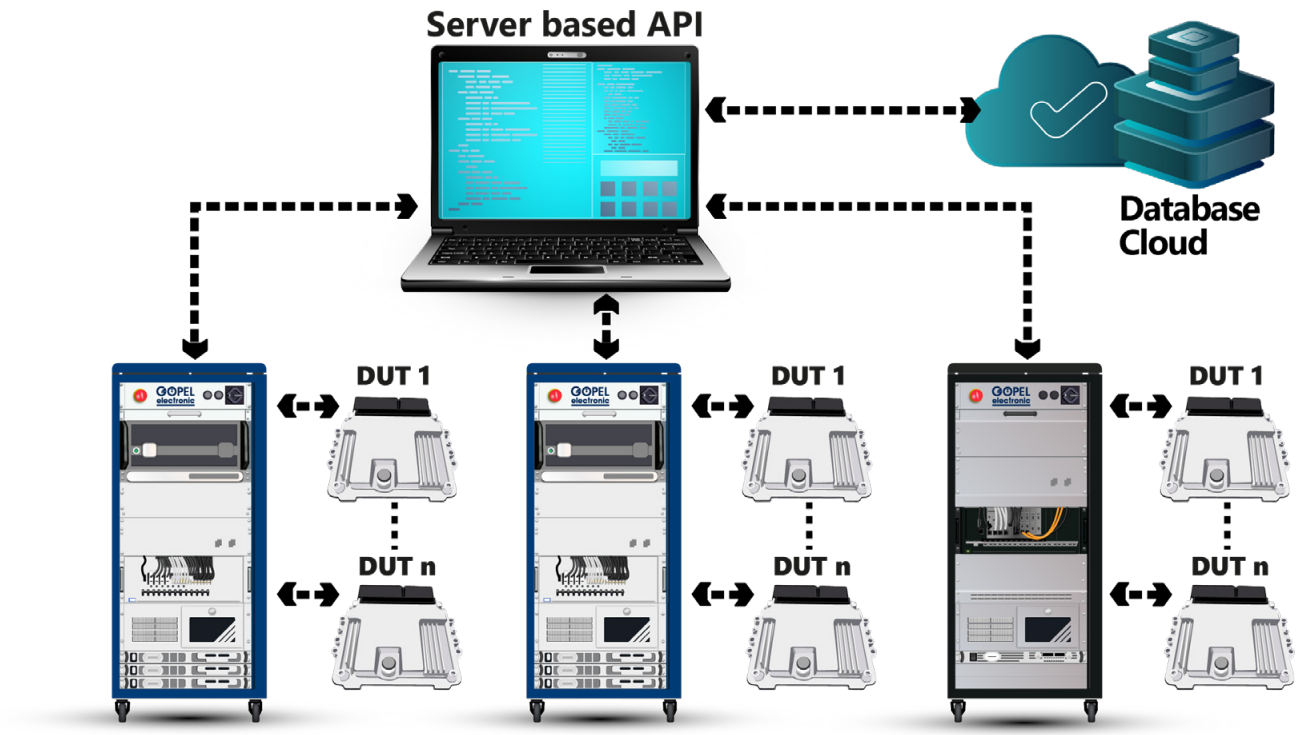
· simple Integration via TCP/IP Connection

· hardware Abstraction Layer (HAL)

· modular Design

· parallelism

· scalability



GOPEL electronic

Technical specification & application

Server-Based API

- compatible with existing GOEPEL test systems and standalone cards (e.g. MagicCar compact, Network Tester NG+, EMV Test System, Serie62 cards)
- capable of customized software development for customer projects
- easily integrable into existing customer test architectures and test sequencers
- adaptable to different programming languages via TCP/IP connection (e.g. NI TestStand, Python, QT)
- open for integration of third-party hardware

Simple Integration via TCP/IP Connection

- supports various software interfaces, operating systems, and platforms
- compatible with established sequencers and scripting interfaces
- facilitates logging and database adaptations

C# LabVIEW python Qt TestStand

Overview

A server-based API (Application Programming Interface) represents an innovative and flexible solution for efficiently integrating and managing complex systems. This technology offers numerous benefits for both developers and end users. Below, we outline the key features and advantages of a server-based API.

Advantages

A core function of the server-based API is the provision of an abstraction layer. This reduces the complexity of underlying hardware and software by offering standardized interfaces for communication. Developers can focus on implementing functionalities without delving deeply into the specifics of the underlying technology.

The API integrates seamlessly with both proprietary and third-party hardware through a robust Hardware Abstraction Layer (HAL). This flexibility ensures that a wide range of hardware configurations can be utilized without compromising system functionality or performance.

Parameters

host interface	PXIe
DUT interfaces	four independent 10*/100/1000 Mbit/s interfaces
extension boards	10Base-T1S*, 100Base-T1, 1000Base-T1
packet generator	· FPGA-based · up to 100 % bus load · jumbo frame support
software support	G API for C/C++ and LabVIEW
features	MACsec*, PTP*, TC10 Sleep/Wake-up, DoIP, DoSoAd, TLS, XCP, iPerf

Advantages

Scalability

The server-based API is designed for maximum scalability. It can be deployed in both small and large systems. As demands grow or user numbers increase, the API can be seamlessly expanded to enhance performance and efficiently utilize additional resources.

Parallelism

Modern server-based APIs support the simultaneous processing of multiple requests. This parallelism ensures that even under high request volumes, there are no bottlenecks, and fast response times are maintained. This is especially advantageous in scenarios involving numerous concurrent users.

Hardware Abstraction Layer (HAL)

- supports hardware expansion and integration
- enables control of current/voltage supplies
- facilitates switching and measuring paths
- provides access to data sinks/sources and other measuring and testing equipment

Modular Design

The modularity of the API ensures high flexibility. Functions can be easily added, removed, or updated without impacting the overall system. This allows for gradual enhancement of functionality to meet evolving requirements.

Summary

The server-based high-level API provides a comprehensive solution for system integration and testing. Its support for multiple programming environments, modular architecture, scalability, and compatibility with diverse hardware and operating systems make it an indispensable tool for modern development and testing frameworks.

Architecture

The architecture of the server-based API enables flexible and scalable integration, ensuring seamless communication between different system components.

The table shows various possible implementations:

Architecture		
Single Test Tower Setup	**Multiple Test Towers Setup**	**Distributed Control and Management**
· Server-Based API · 1 Test Tower · Multiple Devices under Test · Database/Cloud integration	· 1 Server-Based API · Multiple Test Towers · Multiple Devices under Test · Server-based communication · Client connections · Database/Cloud integration	· Multiple Control and Management Instances · 1 Server-Based API · Multiple Test Towers · Multiple Devices under Test · Scalable data handling with Database/Cloud support

GÖPEL electronic GmbH
Goeschwitzer Str. 58/60
07745 Jena · Germany
ats.sales@goepel.com
www.goepel.com
+49 3641 · 6896 0
+49 3641 · 6896 944
ISO 9001, ISO 14001 certified