OpenTAP: Introduction to Test Composition and Automation

Lars Nielsen, Keysight Technologies
5G-VINNI WP4, 03/03/20

This project has received funding from the EU’s Horizon 2020 research and innovation programme under grant agreement No 815279.
Agenda

• OpenTAP overview
  – Concept
  – Architecture
• Main concepts for use
  – UIs
  – Workflow
• Practical example from test composition to results
  – Demo
Introduction to OpenTAP

OPENTAP OVERVIEW
What is OpenTAP

- OpenTAP: Open(source) Test Automation Platform
- OpenTAP is a lightweight sequencing engine
- OpenTAP has a highly modular structure
  - Everything (including GUIs) is seen as a plugin
- Configure DUTs/SUTs, tools
  - via instruments and test steps contained in plugins
- OpenTAP provides a low entry barrier for development and usage
  - Quick development of plugins and test steps tailored to individual needs
  - Simple drag’n’drop of test steps in test plan composition
- It is possible to “rule” all the tools (not only Keysight’s) and DUTs
Why OpenTAP for Test Automation

• **Simplicity**
  – Low technical barrier of entry. TAP is not “yet another programming language”
  – Does just what you need, not trying to be “everything for everyone”
  – Simple for everyone: programmers, non-programmers, and operators

• **Scalability**
  – Modular software architecture centered around a core sequencing engine
  – IP encapsulated as plugins. Build solutions through re-use, not re-invention
  – Plugins can be shared and reused

• **Speed**
  – Optimized for manufacturing (speed / time = $), yet robust for R&D usage
  – Powerful analytics such as Timing Analyzer; continuously & efficiently improve
  – Get to market faster...accelerate your deployments, outpace the competition
How does the architecture look like?

- **DUT Plugins**
  - Custom DUT
  - VNF DUT

- **Test Step Plugins**
  - Traffic load step
  - Custom Step
  - Configure Tool

- **Instrument Plugins**
  - Hawkeye
  - Yardstick
  - IxLoad
  - Custom Inst

- **User Interfaces**
  - CLI
  - TUI
  - Web Editor
  - Desktop Editor
  - REST API

- **Result Listeners**
  - Text Log
  - RabbitMQ
  - DB Storage
  - Custom UI

**PathWave**

- Automation Engine
- DB Storage
- CLIs
- Desktop Editor
- Web Editor
- REST API

Components:
- VNF DUT
- Configure Tool
- Custom UI
- Custom Inst
- IxLoad
- RabbitMQ
- Web Editor
From a User’s Perspective
Community Edition Developer GUI

- Free to use for non-commercial organizations & open source projects
- Download via OpenTAP.io or Keysight.com
- Ready to use in under 85 seconds!
- Quickly create & edit test plans without any programming experience
- Experiment with demo plugins requiring no hardware
Join – The Community

https://gitlab.com/OpenTAP
Introduction to OpenTAP

MAIN CONCEPTS FOR USE OF OPENTAP
OpenTAP GUI Components - Desktop

Test Step Panel

Test plan Panel

Settings Panel

Log Panel

Resource Bar: Instruments, DUTs, result listeners
OpenTAP GUI Components - TUI

- Textual based UI
- Create and edit OpenTAP plans
- Modify bench settings
- Run in almost every terminal including Docker containers
Example of OpenTAP Workflow

• Prerequisites
  – Plan what to test and how
  – OpenTAP plugins to control DUT and test tools
  – DB to store results
  – Result listener to post results to DB

• Flow
  – Install needed plugins using OpenTAP package manager
  – Configure instruments in OpenTAP editor
  – Compose OpenTAP plan using editor
  – Configure result listener
  – Run OpenTAP test plan
  – View results, e.g. in Grafana
Generic Example of Vertical Application

Use case
- Emergency vehicle transporting patient to hospital
- Communicate status to personal at hospital to prepare/initiate treatment
- Communication equipment in vehicle and hospital
- Equipment support functions to connect and transmit status, live video, vital signs, etc.
- Equipment can be controlled via offered APIs
- Test engineer to write scripts to test system capabilities in terms of functionality and performance
Generic Example of Vertical Application

Use case – OpenTAP approach
• Create plugin to control each entity
• Plugin offers steps to use when composing test plan (drag and drop)
• OpenTAP engine executes test steps as sequence
• Expose external parameters of selected variables
• Easy automation of tests
Introduction to OpenTAP

PRACTICAL OPENTAP USAGE EXAMPLE
Test Overview - Hawkeye

- Configure and run Hawkeye test between 2 endpoints
- Endpoints acting as both client and server
- Store results in DB of choice
- Create result visualizations

Hawkeye
- Application-layer end2end network performance test tool
- Central server and distributed endpoints
- Server acting as central registration/command and control server
- Endpoints acting as worker nodes
  - Register with server
  - Pull test specifications from server
  - Execute tests of network performance between endpoints
  - Push results to server after tests
- Supporting wide range of measurement types and application layer traffic emulation
Overview of Planned Test – OpenTAP Managed

- Create OpenTAP plan using test steps from Hawkeye plugin
- Configure tests and run
- Pull results from server
- Post results in DB via Result Listener
- Visualize results in Grafana dashboard
Use OpenTAP to Run Hawkeye Tests

DEMO
Thank you for your attention

Q&A

Resources:
OpenTAP: https://www.opentap.io/
OpenTAP on Gitlab: https://gitlab.com/OpenTAP/opentap
5G-VINNI: https://www.5g-vinni.eu/

This project has received funding from the EU’s Horizon 2020 research and innovation programme under grant agreement No 815279.