KEYSIGHT

NAF AUTOCON2: WS:D1
Automated & Scalable Network Testing with OTG

A Brief History of Keysight

KEYSIGHT
+
ixia

1939–1998:
Hewlett-Packard years

1999–2013:
Agilent Technologies years

2014+:
Keysight years

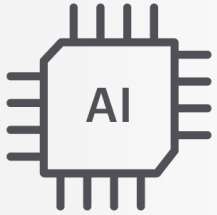
2017:
Keysight acquires Ixia

Network Test Automation

Why is it getting more importance today?

Open networking ecosystems

Industry wide co-design



Networks are bottlenecks for AI

Data center refresh cycles are shrinking

Testing as code

Key challenges

Why is not fully automated yet?

- Different tools at different stages
- CI/CD needs CT
- Proprietary non-standard APIs
- Cost prohibitive
- Lack of community

Open Traffic Generator (OTG)

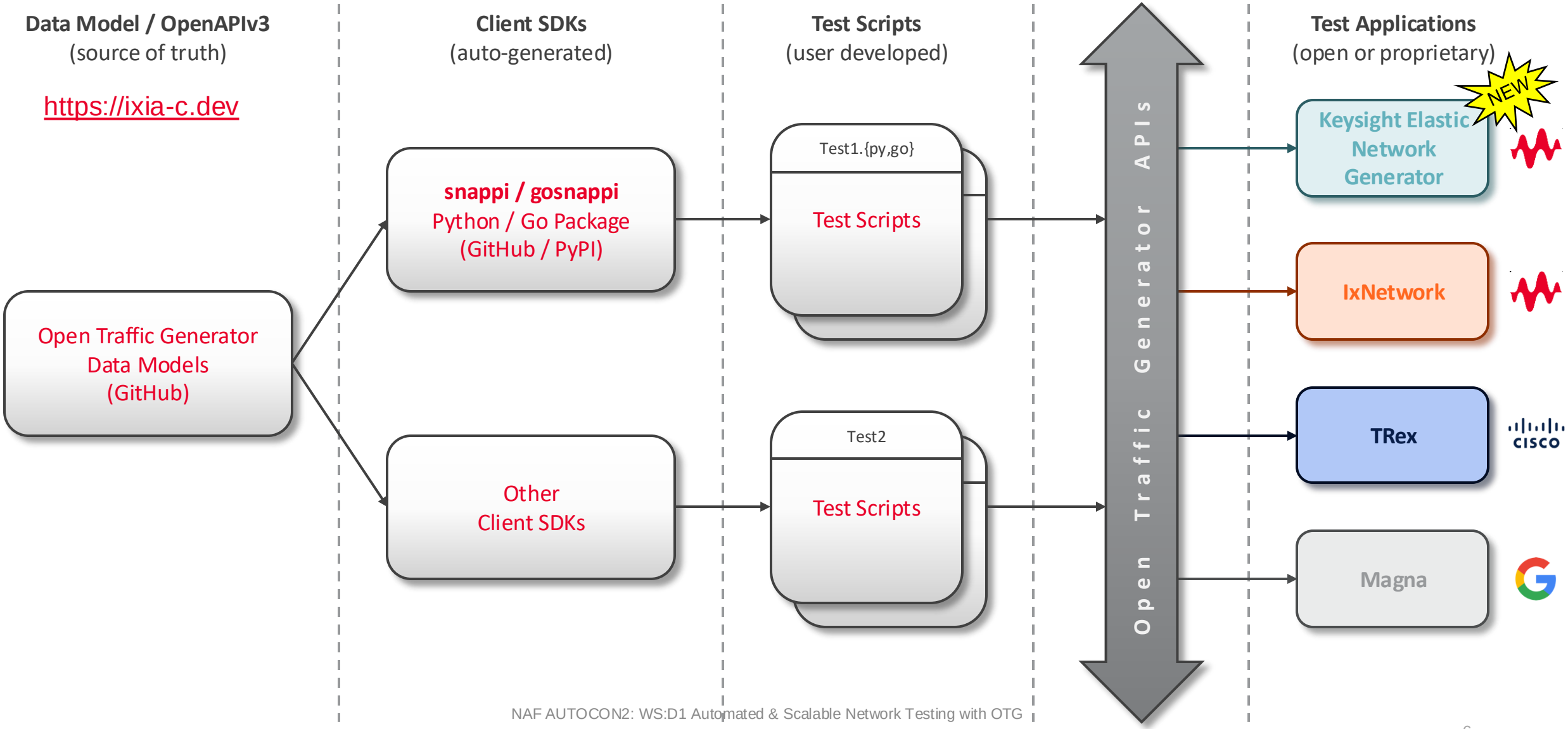




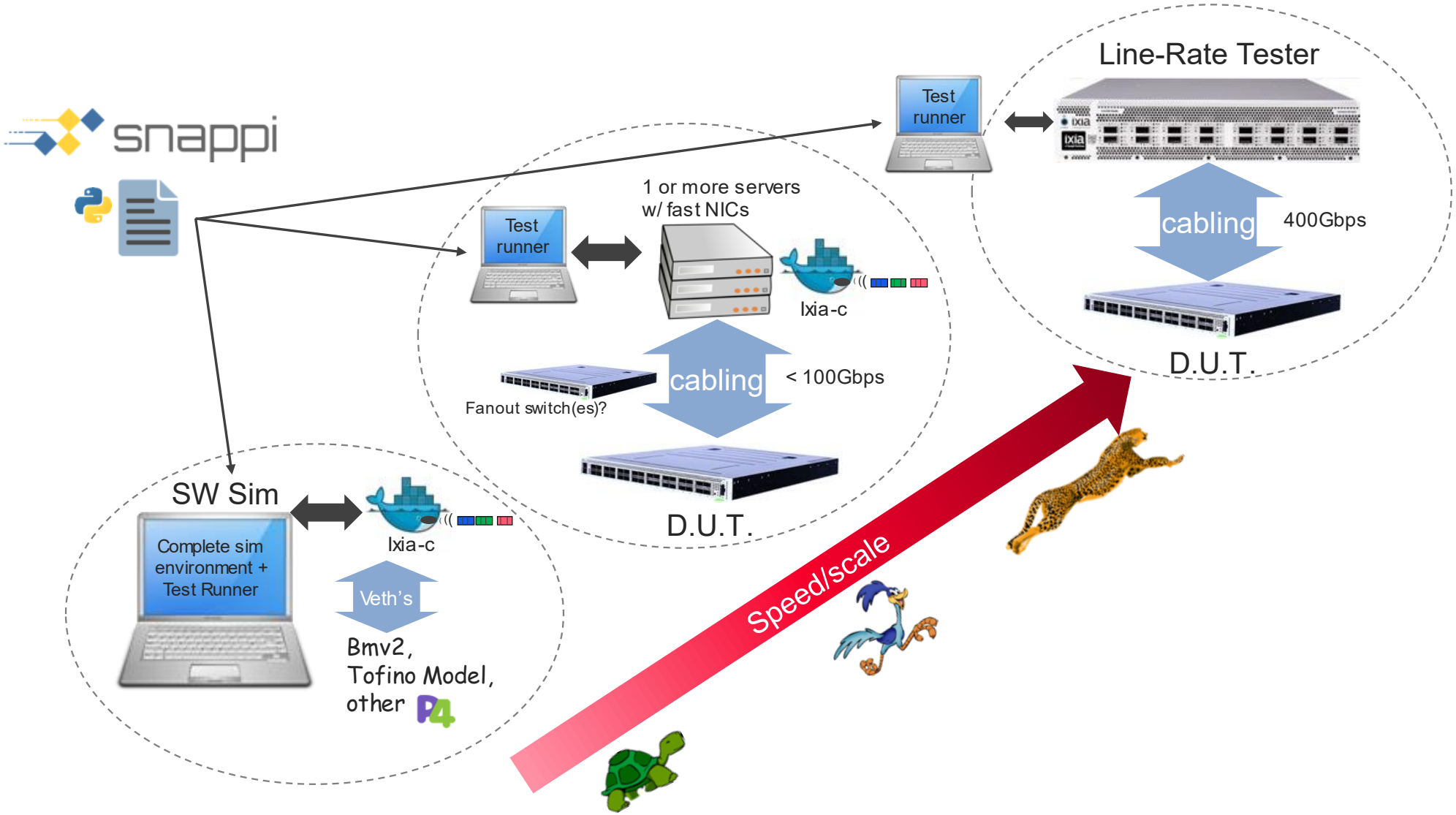
Model-based | Vendor Neutral | Open-API | Use-case driven | Community

Visit https://otg.dev and get involved

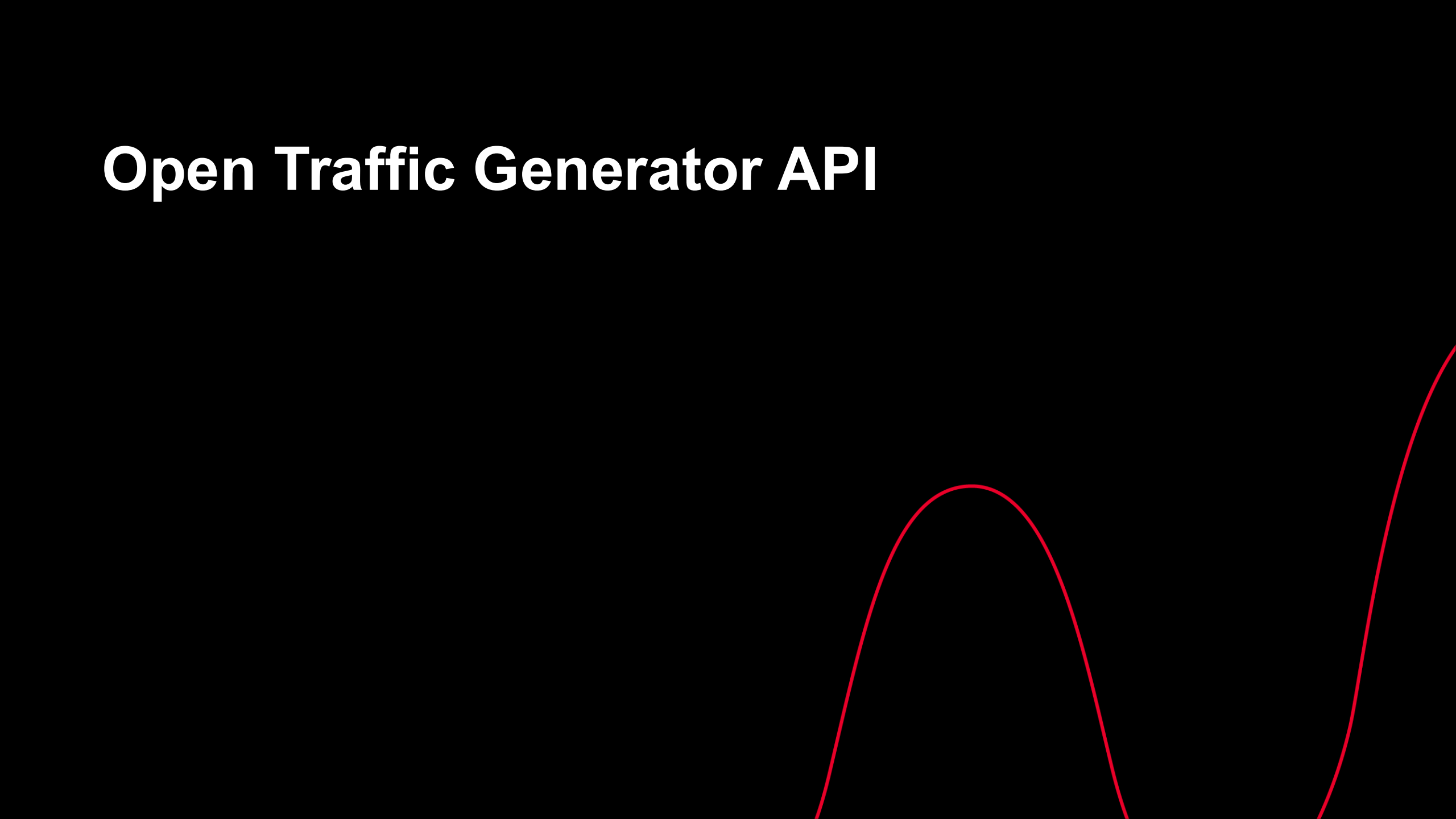
Open Traffic Generator API



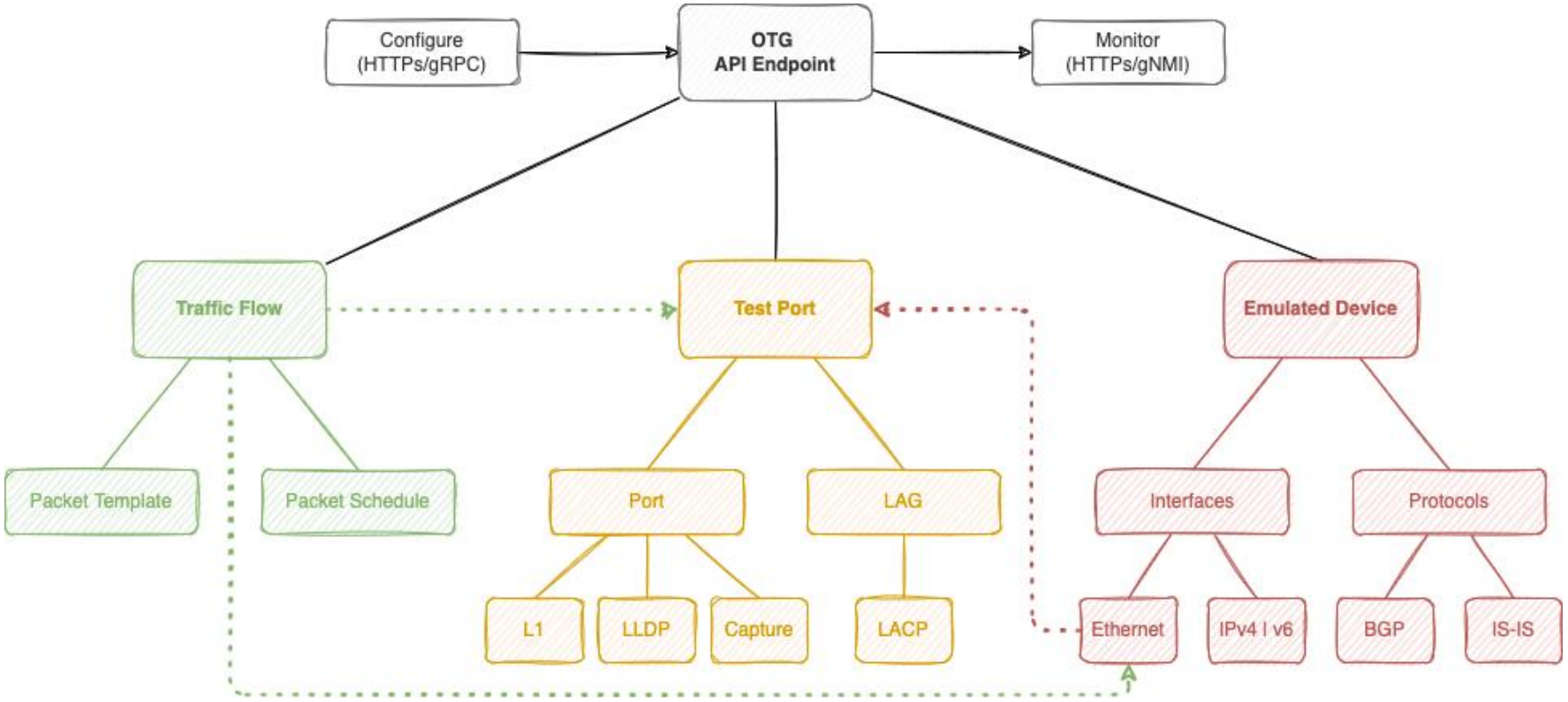
Open traffic generator -- portability



Open Traffic Generator API



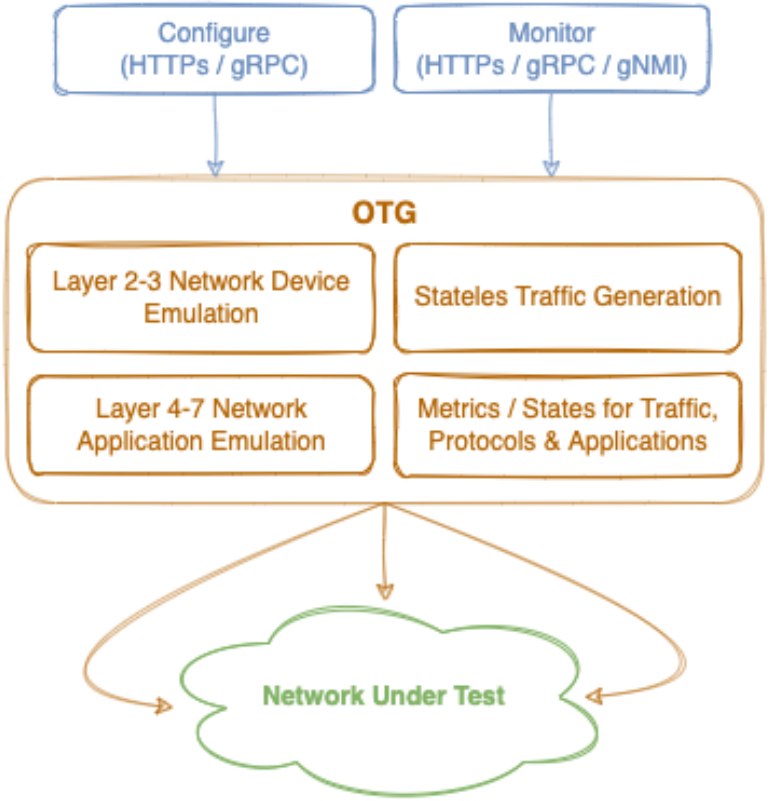
What – OTG Model



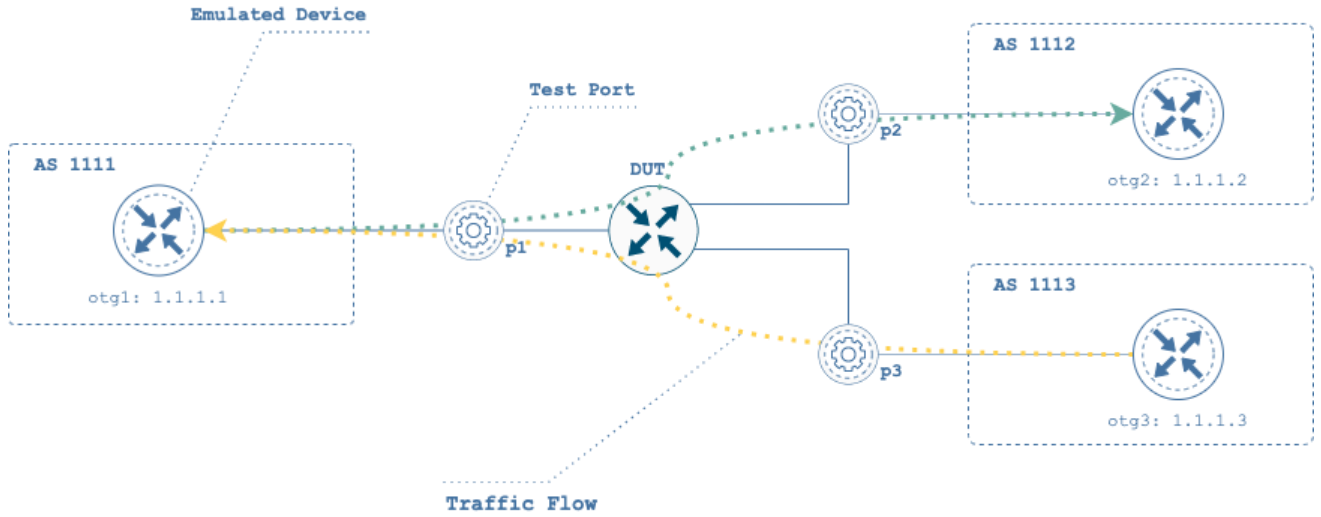
OTG L2-3 Model Hierarchy

What – OTG API

https://otg.dev



OTG API Surface

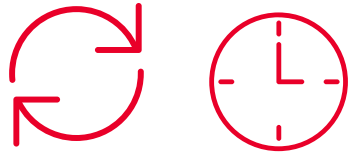


OTG L2-3 Components

- Test Ports
- Emulated Devices
- Traffic Flows

Open Traffic Generator API

Declarative vs Imperative



Declarative – choice for OTG

- One OTG config – one API call to apply
- Speed to apply & speed to fail!
- Model enforcement – client and server

Imperative

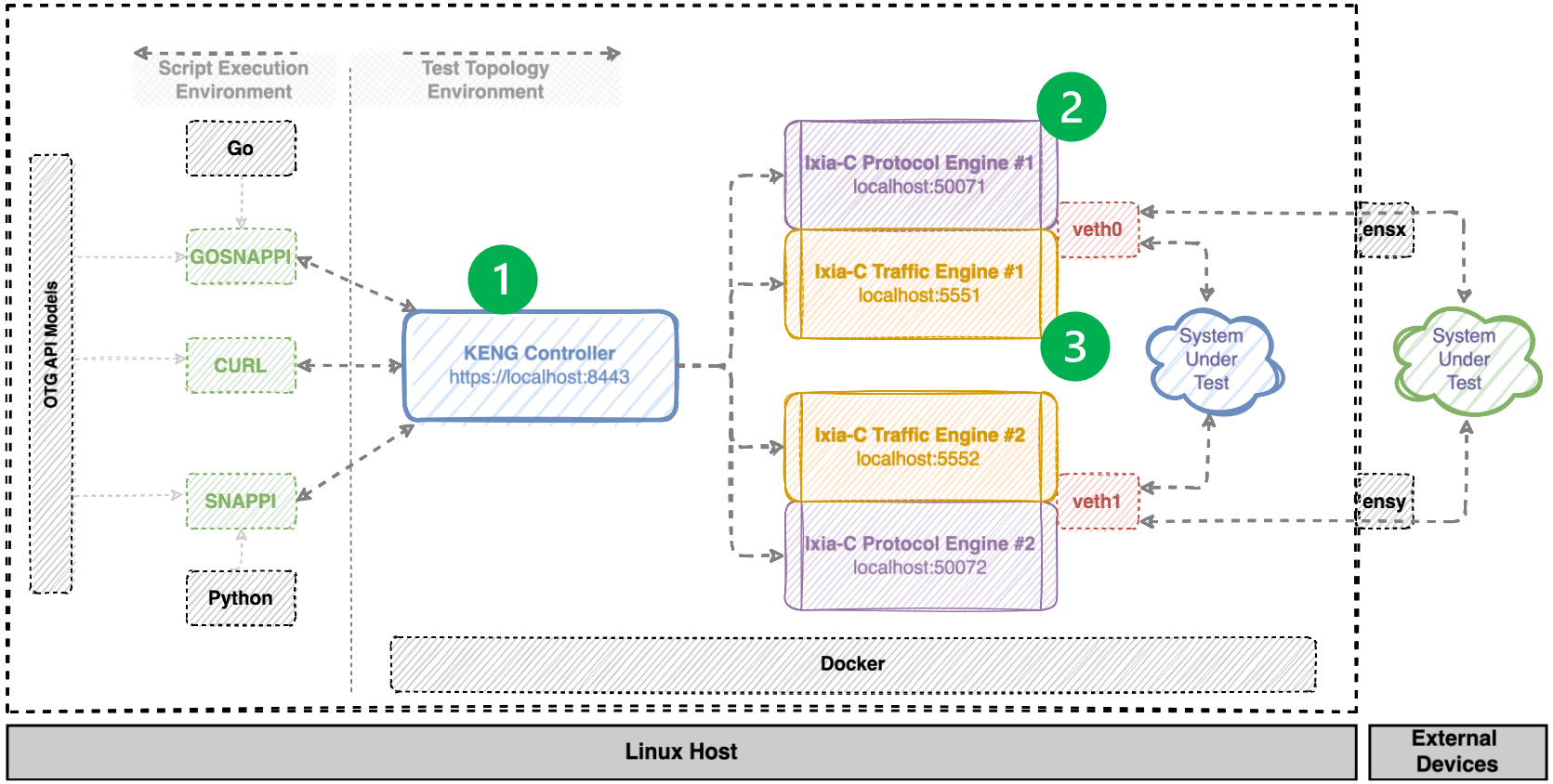
- Sequence of API calls
- Latency of each call compounds
- Slow in getting to a failing API call

- Client-Side library for OTG APIs
- OO, defaults, client-side validations, factory methods, single import, serialize/deserialize whole config or objects…
- Pythonic
- Available on PyPI
  - `pip install snappi`

```python
# test TCP ACL on ASIC
# increment TCP source port

import snappi

api = snappi.api(location = "https://10.3.2.3")
cfg = api.config()

f1 = cfg.flows.flow(name = 'flow1')[-1]
f1.size.fixed = 1518
f1.rate.percentage = 10
f1.metrics.enable = True

eth, ip, tcp = f1.packet.ethernet().ipv4().tcp()

eth.src.value = "00:CD:DC:CD:DC:CD"
eth.dst.value = "00:AB:BC:AB:BC:AB"

ip.src.value = "1.1.1.2"
ip.dst.value = "1.1.1.1"

tcp.src_port.increment.start = 5000
tcp.src_port.increment.step = 2
tcp.src_port.increment.count = 10
```

OTG Components

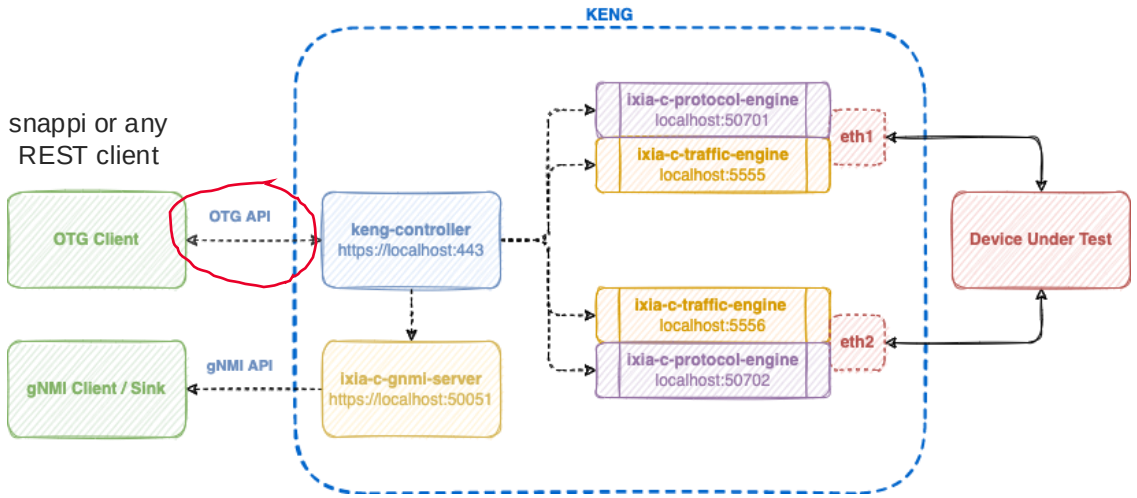
Building blueprint



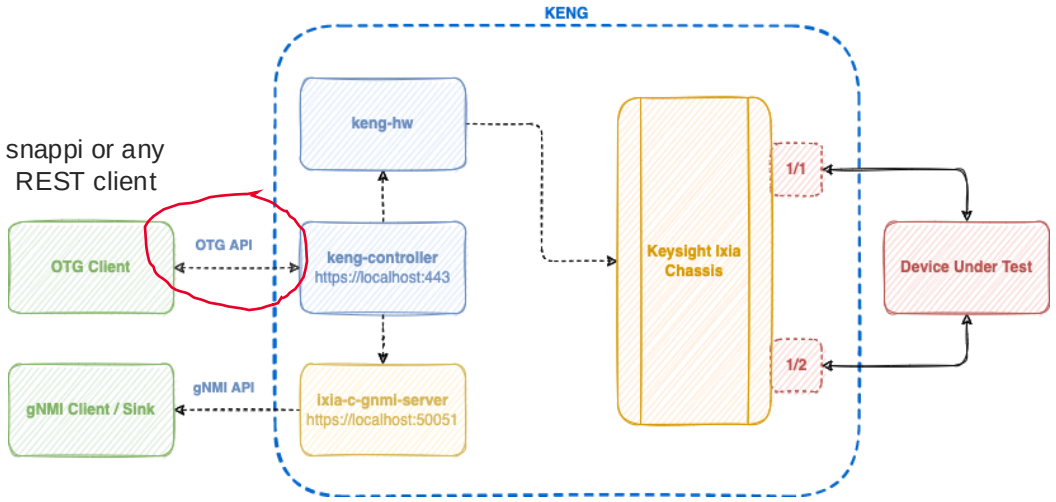
1. KENG Controller: The brain which manages all the components
2. Ixia-c protocol engine: Control plane emulation (BGP, ISIS etc.)
3. Ixia-c traffic engine: Data plane, traffic flows etc.

OTG Components continued

Different implementations of OTG

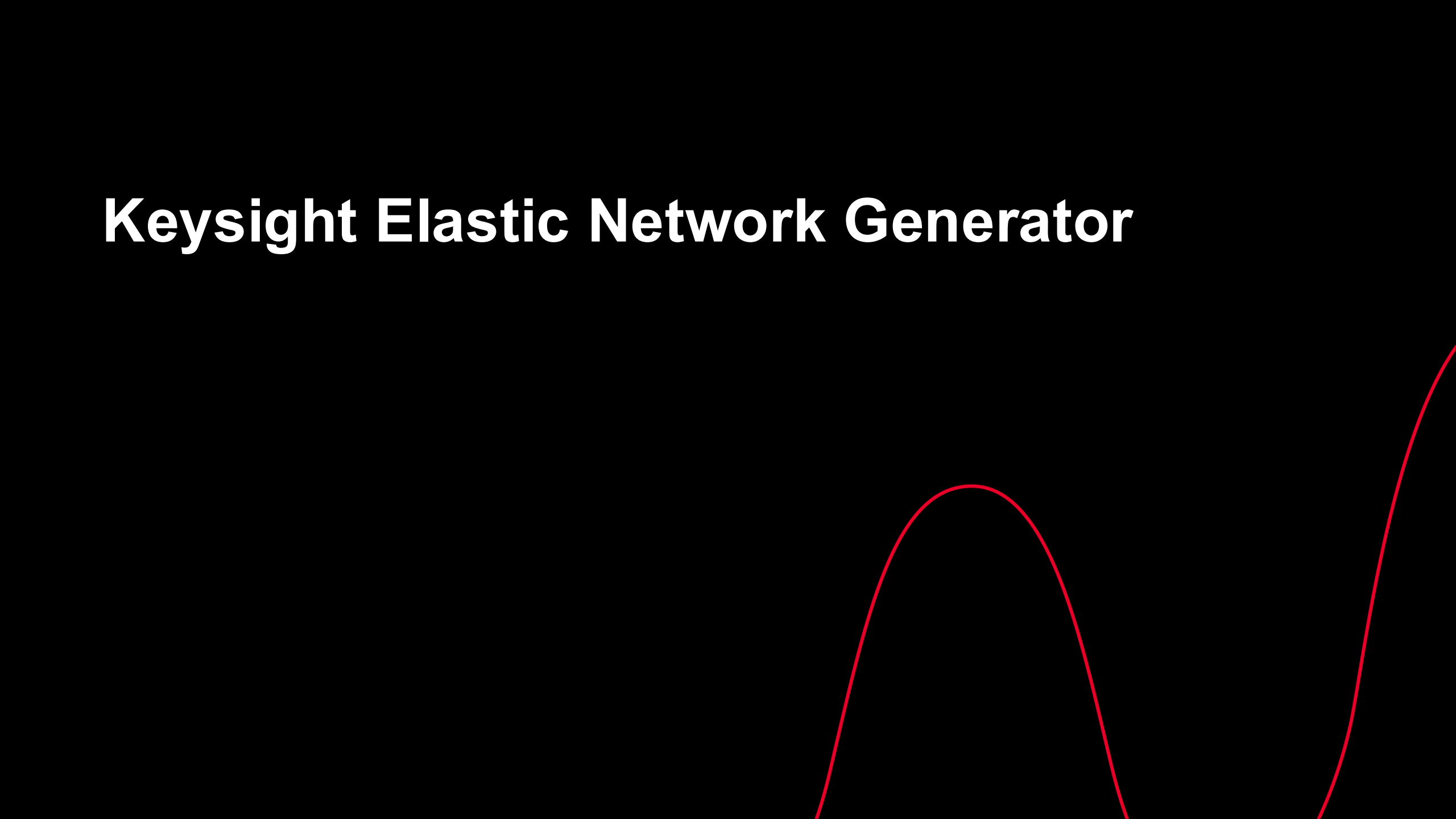


OTG implementation in Ixia-c SW

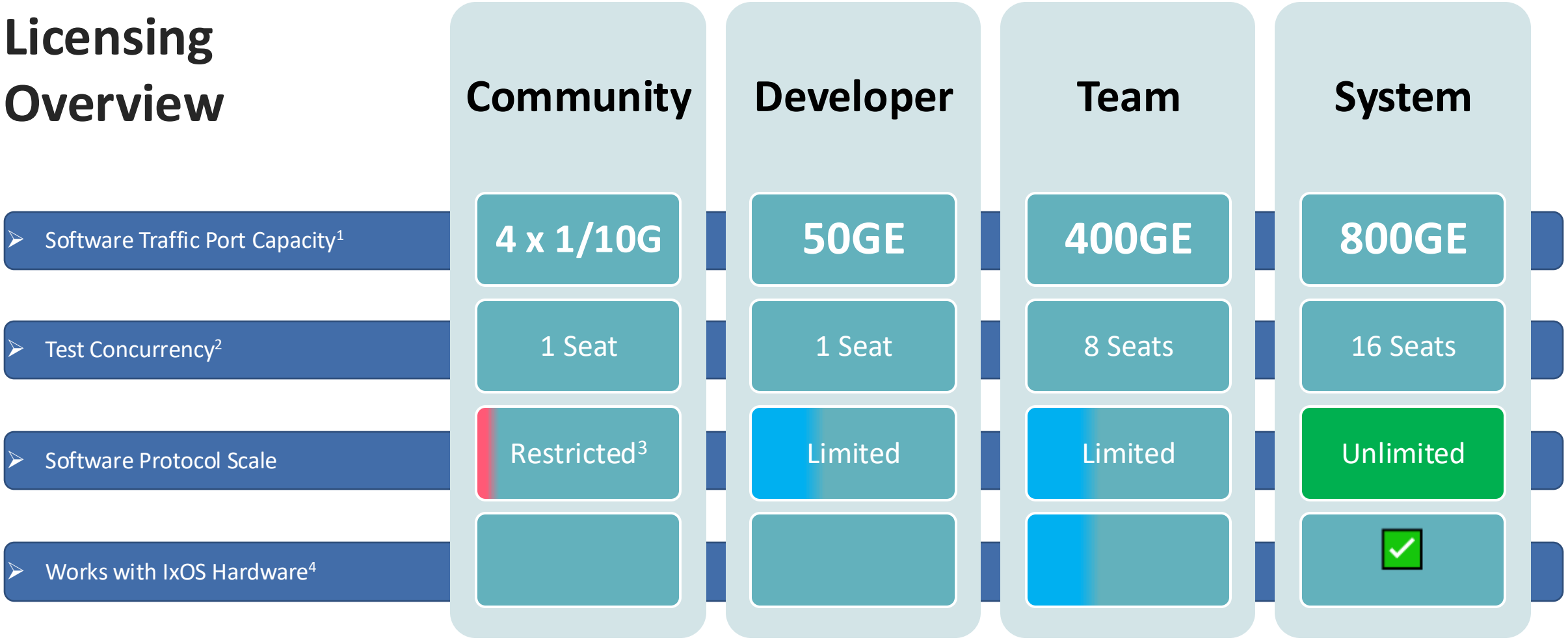


OTG implementation to Ixia HW

Keysight Elastic Network Generator



Licensing Overview



(1) Port capacity applies to Ixia-c software ports and is determined as a sum of configured test port speeds: 100G, 50G, 40G, 25G, 10G, 1G
(2) Concurrently running controller instances with non-empty configuration
(3) Restricted protocol scale: 4 x IP/BGP sessions
(4) Novus and AresONE load modules and appliances

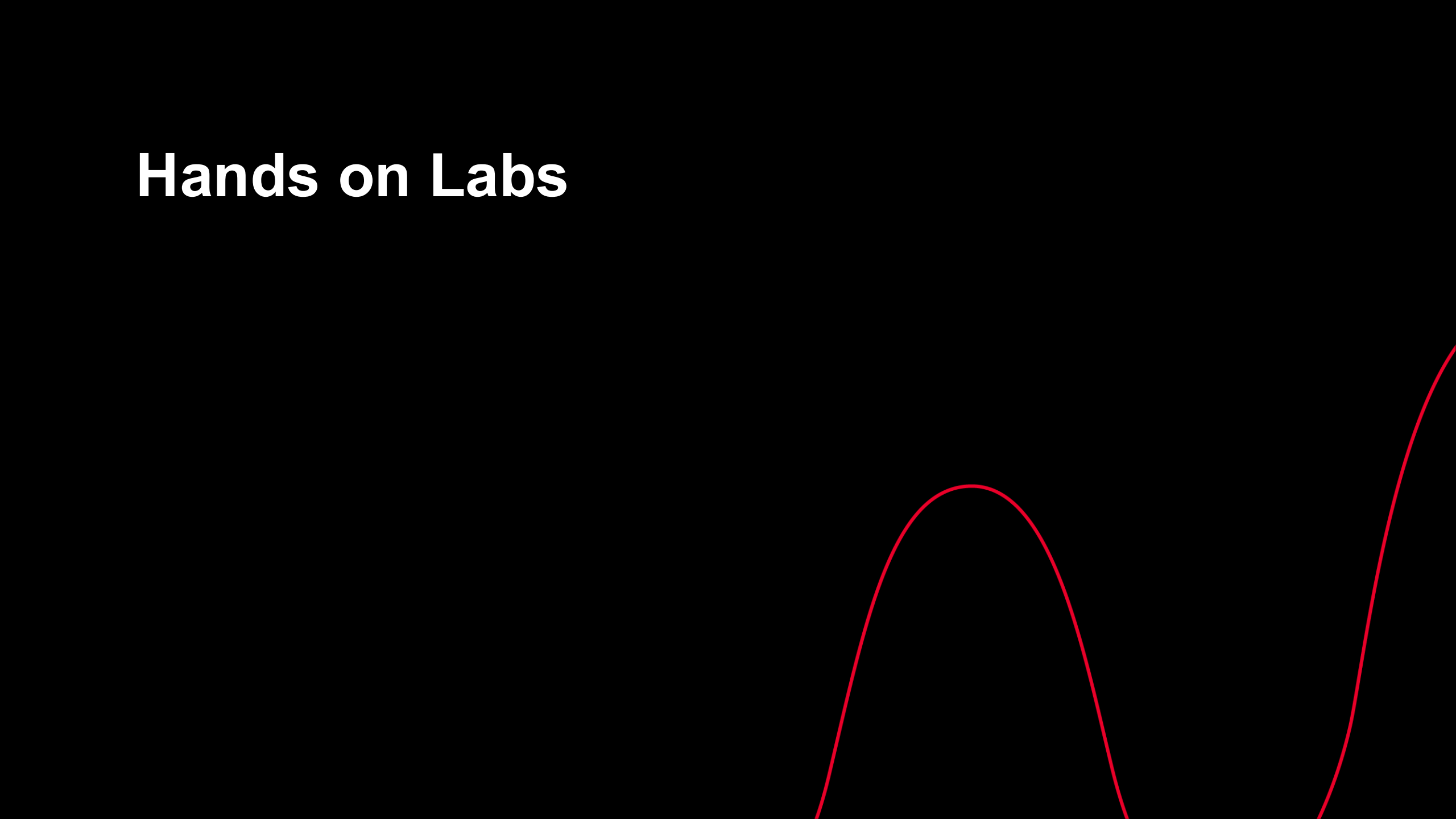
KENG/OTG learning path

- Docs: https://ixia-c.dev/
- OTG GitHub Repository: https://github.com/open-traffic-generator
- OTG Examples: https://github.com/open-traffic-generator/otg-examples
- Quick start with Ixia-c: https://github.com/open-traffic-generator/conformance
- Labs (with deployments on different environments):
  - B2B Ixia-c Traffic
  - Static B2B LAG
  - B2B IxOS Hardware
  - More labs: https://github.com/open-traffic-generator/otg-examples#reference
- Ixia-c Slack channel: https://join.slack.com/t/ixia-c/shared_invite/zt-2p11e5yua-u3o1aWzlJcjJquSAqoDk2Q

Keysight Elastic Network Generator – References

- **Open Traffic Generator**
  - otg.dev
  - OTG Data Models
  - OTG Examples
  - OTG Snappi SDK
  - OTG Snappi Test Collection
- **OSS OTG CI Pipelines**
  - Open Config – Feature Profiles
  - Azure DASH CI Pipeline
- **Other Information**
  - OTG / OTGEN CLI / OTG Examples
- **Keysight Elastic Network Generator**
  - KENG Product Page
  - KENG Data Sheet
  - KENG Automation Video
- **Keysight Ixia-c**
  - Ixia-c Product Page
  - Ixia-c Packages
  - Ixia-c Releases
- **Related OSS Projects**
  - Open Config – Kubernetes Network Emulation (KNE)
  - Open Config – ONDATRA
  - Containerlab / Ixia-c-one

Hands on Labs



Schedule

Task	Time	Owner
Opening presentation	25	Mano
Q&A	5	Team
Lab 00	10	Team
Lab 01	20	Team
Lab 02	20	Team
Break	15	Team
Customer usecase presentation	25	Jasdeep
Q&A	5	Jasdeep
Lab 03	25	Team
Lab 04	25	Team
Break	15	Team
Lab 05	30	Team
Lab 06 Demo	10	Octav
Conclusion	10	Mano

Lab	OTG Test Tool	OTG Test Tool Components	OTG API Client	Infrastructure	DUT	Learning Objective	Duration
00	n / a	n / a	n / a	n / a	n / a	DOCKER --> PYTHON	~ 10 min
01	Ixia-c	Keng Controller Ixia-C Traffic Engine	OTGEN + SNAPPI	DOCKER CLIENT	B2B	SNAPPI IXIA-C OTGEN	~ 20 min
02	Ixia-c	Keng Controller Ixia-C Traffic Engine Ixia-C Protocol Engine	SNAPPI	DOCKER COMPOSE	B2B	DOCKER COMPOSE SNAPPI PROTOCOLS SNAPPI CAPTURES REST STATES REST STATS	~ 20 min
03	Ixia-c	Keng Controller Ixia-C Traffic Engine Ixia-C Protocol Engine	SNAPPI	CONTAINERLAB	Nokia SRL	CONTAINERLAB IXIA-C-ONE DEPLOYMENT EGRESS TRACKING	~ 30 min
04	Ixia-c	Ixia-C-One	SNAPPI	CONTAINERLAB	Nokia SRL	SNAPPI PROTOCOLS SNAPPI TRAFFIC CONTROL ACTIONS	~ 30 min
05	Ixia-c	Keng Operator Keng Controller Ixia-C GNMI Server Ixia-C Traffic Engine Ixia-C Protocol Engine	SNAPPI GOSNAPPI	KIND / KNE	B2B	IXIA-C IN K8S KENG-OPERATOR GOSNAPPI GNMI / GRPC	~ 30 min
Demo	HW Ports	Keng Controller Ixia-C GNMI Server Keng-Layer23-HW Server	SNAPPI	DOCKER COMPOSE	HW DUT	KENG for HW	~ DEMO 10 min