# All-in-one functional, integration, performance testing and QoE monitoring with CLIF

Bruno Dillenseger
Orange Labs

CLIF in a nutshell

➔ **A generic and adaptable Java framework for distributed performance testing and monitoring**

- load injectors generating traffic and measuring response times
  - supported protocols: TCP, UDP, DNS, HTTP, LDAP, MQTT, RTP, SIP...
  - bring your own protocols and data sets
- probes measuring resources usage
  - CPU, disk, RAM, network, JVM, RTP
  - bring your own probes

System Under Test

➔ **High power**

- up to 1000 load injectors in parallel X millions of virtual users...
- ... but also OK with 1 virtual user on a single load injector

➔ **Toolbox**

- Scenario edition and execution, test monitoring and reporting
- Integration to Eclipse and Jenkins, command line interface
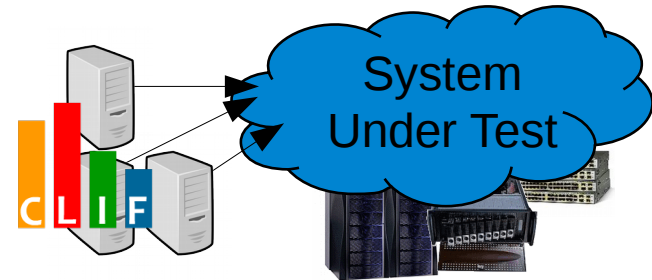
CLIF scenarios' fundamentals

➔ CLIF scenarios define virtual users behaviors...
- basic statements for execution control:
  - conditional statements (while loop, if-then-else, preemption)
  - probabilistic branching
- actions implemented by plug-ins:
  - perform a request and measure its response time (*sample*)
  - make a pause, aka think time (*timer*)
  - evaluate some *condition* for a conditional statement (while, if, preemption)
  - set some plug-in parameters (*control*)
  - get some data set from an external source (*data provisioning*)

➔ ... and the number of active virtual users
- load profile: $n = f(time)$

Performance testing


Tester

System Under Test

Load injectors
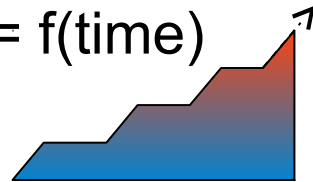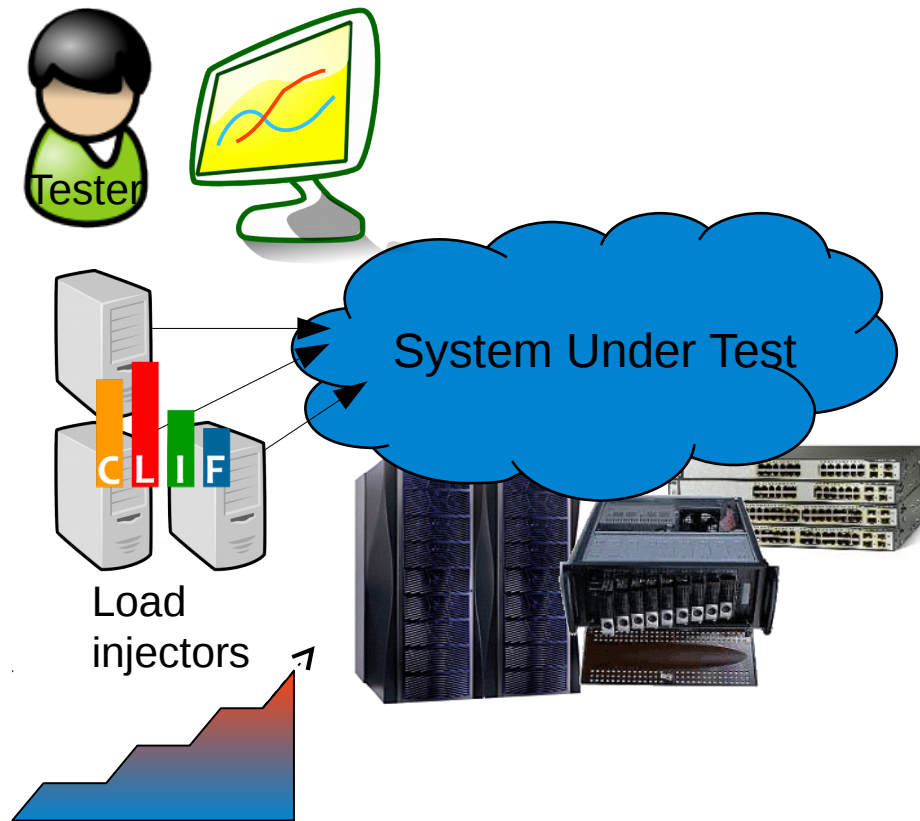
➜ Performance testing is CLIF's primary target

➜ The benchmarking plan relies on:

- typical users behavior(s)
- requests and expected results
- data sets
- metrics of interest
- a variety of load profiles
  - ≠ numbers of virtual users
  - ≠ durations
  - ≠ aggressiveness

Quality of Experience monitoring

Ops

QoE alarms

On-line system

Real users

CLIF
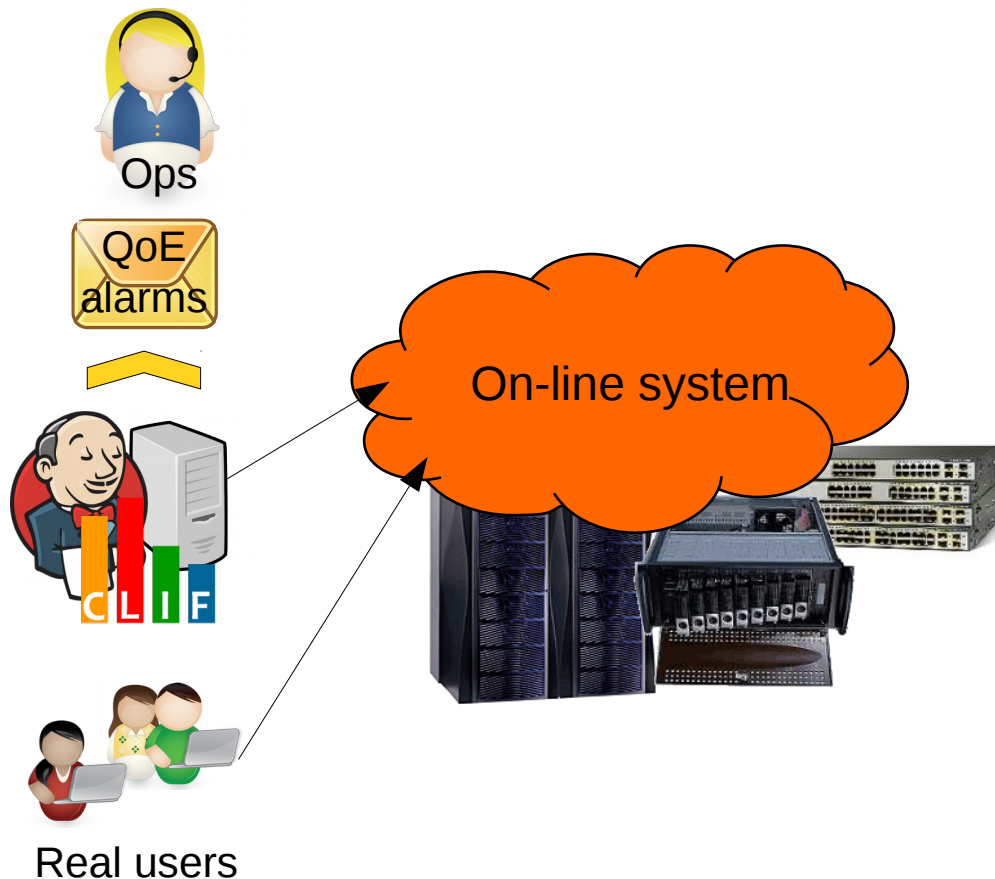
→ Typical needs:
- periodically check *health*, *correctness* and *responsiveness* of the on-line system
- notify a supervision team

→ Reuse the benchmarking scenarios:
- with thin/unit load profile
- scheduling, reporting and notifications done by any automation tool (e.g. Jenkins + CLIF plug-in)

Functional, Integration testing

Developers
Testers
Integrators

test automation

System under Development
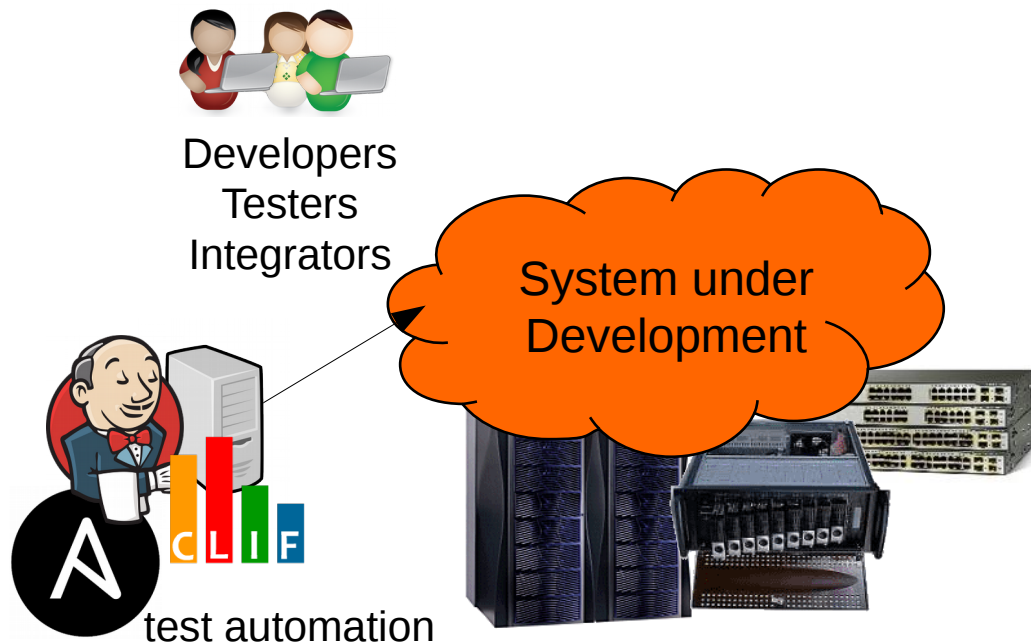
- ➔ Typical needs:
  - ▪ check correctness based on realistic use cases
  - ▪ why not getting some unit performance metrics for free?

- ➔ Still the same scenarios
  - ▪ with thin/unit load profile
  - ▪ test scheduling, deployment and reporting supported by automation tools (Jenkins + CLIF plug-in, Gitlab CI, Ansible...)

Conclusion

➔ With CLIF, write your test scenarios once for all:

- Functional, integration testing
- Performance, load testing
- Quality of Experience monitoring
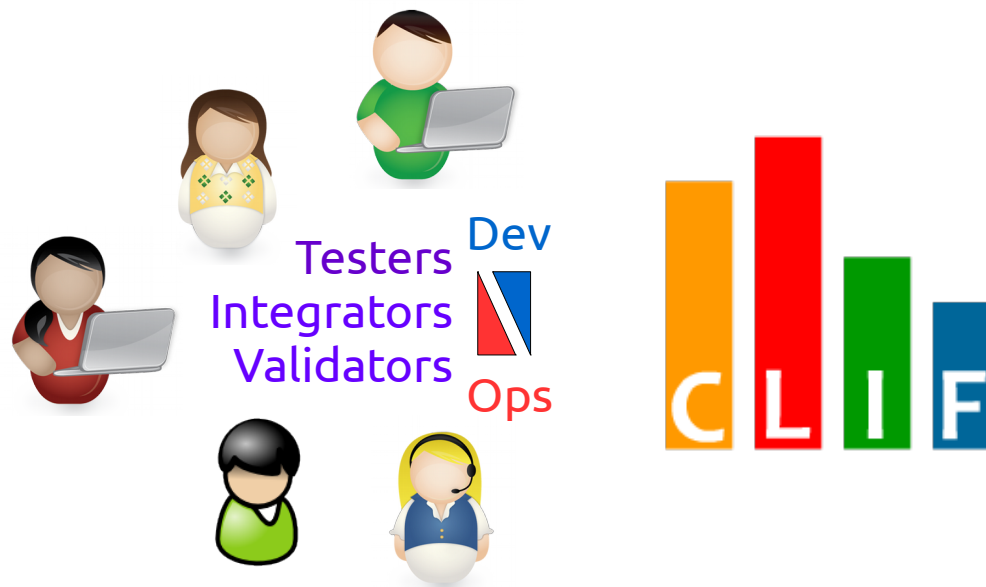
And now, what's up?

➔ CLIF is moving to OW2 Gitlab
- https://gitlab.ow2.org/clif/clif-legacy
- Fresh CLIF distributions generated by Gitlab CI
- clif@ow2.org

➔ Container image on Docker Hub: dillense/clif

➔ Thank you for your attention

➔ Any Question?