ETICS supporting compliance and interoperability

Gabriele Giammatto
Engineering Group

14 November 2013
The OCEAN project is building a directory of open-source cloud-oriented projects focused on quality and interoperability of software components.

ETICS tool is used to in OCEAN to:

- extract **quality** metrics from software
- verify **interoperability**
- check **compliance** to open cloud standards
ETICS

ETICS is a **build and test automation** tool

- **Web-based** portal for managing configuration, builds, tests, packages, reports
- **independent** from programming language, project structure and target platform:
  - C, C++, Java, Scala, Python, Ruby, ...
  - Make, Ant, Maven, SCons, Rake, ...
  - RedHat, Debian, Windows, ...
ETICS is a **build and test automation** tool

- **Dependency management**: build/run-time, static/dynamic, ranges
- Built-in **packing system** to create distribution packages (tar.gz, deb, rpm, sources, ...)
- Central **repository** for packages, reports, logs compatible with yum/apt formats
- Testing capabilities extendible through **plug-ins**
Testing with ETICS - Overview

- **static analysis**
  - integration of specialised tools for code analysis like *Checkstyle*, *Findbugs*, *PMD*, *CppCheck*, *CCCC*, *PyLint*, ...
Testing with ETICS - Overview

• **static analysis**
  • integration of specialised tools for code analysis like *Checkstyle*, *Findbugs*, *PMD*, *CppCheck*, *CCCC*, *PyLint*, ...

• **unit testing**
  • integration of *JUnit*, *CppUnit*, *PyUnit*
Testing with ETICS - Overview

- **static analysis**
  - integration of specialised tools for code analysis like Checkstyle, Findbugs, PMD, CppCheck, CCCC, PyLint, ...

- **unit testing**
  - integration of Junit, CppUnit, PyUnit

- **functional testing**
  - execution and reports of tests written by developers
Testing with ETICS - Overview

- **static analysis**
  - integration of specialised tools for code analysis like Checkstyle, Findbugs, PMD, CppCheck, CCCC, PyLint, ...

- **unit testing**
  - integration of Junit, CppUnit, PyUnit

- **functional testing**
  - execution and reports of tests written by developers

- **multi-node testing**
  - realisation of complex testing scenarios involving multiple nodes
ETICS Multi-node testing

ETICS provides a tool-kit for the definition and automation of complex test scenarios involving multiple nodes that interact each other in a synchronised way.

Applicability:

- **Deployment** testing of services
- Functional testing of **client-server** applications
- **Interoperability** testing between services
Multi-node characteristics

- tests definition through the web portal
  - number and characteristics of nodes
  - software to be installed on each node
  - testing scripts

- define once run multiple times:
  - on regular base, on every commit to source repository
  - on different versions of the software and platforms
  - with different values for parameters

- nodes dynamically created and destroyed for each test
ETICS provides a **synchronisation primitives** to be used in testing scripts:

- **etics-set**: publish an information
- **etics-get**: retrieve an information (blocks until published)

**e.g.**:
- on server node: `etics-set SERVER_IP 192.168.100.12`
- on client node: `etics-get SERVER_IP`
Multi-node testing – A simple example

Test the deployment of a Java web application

**Node A script:**
1. deploy *Tomcat*
2. deploy *MyWebApp in Tomcat*
3. start *Tomcat*
4. publish node address

**Node B script:**
1. retrieve node A address
2. contact the web application and verify returned values are the expected ones
Multi-node testing – A complex example

Test deployment and set-up of a **Cassandra** cluster using four nodes:

1. **Cassandra 0** (seeder) node
2. **Cassandra 1** node
3. DB initialisation node
4. Test Client

**Workspace**

- My Workspace
  - cassandra-cluster-test.HEAD
  - cassandra-cluster-client.HEAD
  - Default platform
    - Test Commands
    - Properties
    - Environment
  - Deployed Software
    - apache-cassandra-1.2.1
  - cassandra-data-filler.HEAD
  - cassandra-node1.HEAD
  - cassandra-seeder-node.HEAD

**Test Commands**

description:

- clean: etics-get -b NODE0_READY
- init: etics-get -b SEED
- $(apache-cassandra.location)/bin/node工具 --host etics-get -b SEED
- ring
- test:
Interoperability Testing

Multi-node testing will be used to support interoperability tests in OCEAN project

- modelling software to test in ETICS
- writing tests to make services interact one each other

Benefit are multiple:

- **automate** execution, run tests remotely
- once the test is defined, it can be executed **multiple times using different combination** of software components available
- early **discovery** bugs that could **break the interoperability**
Compliance Verification

Focusing on OCCI for which some test-suites are already available:

- **OCCI Compliance Tool**
- **DoYouSpeakOcci (DYSO)**

Our goal is to:

- extend/complete these test-suites
- integrate them in ETICS:
  - execution simplified and automated
  - make them available to all OCEAN projects
Conclusion

- OCEAN aims at creating a directory for open-source cloud-oriented projects

- Projects information will be enriched with data extracted by ETICS

- Made possible by ETICS testing capabilities:
  - static analysis plug-ins
  - unit and functional testing plug-ins
  - multi-node testing mechanism to support
    - compliance and interoperability testing
Thank You!

http://etics.res.eng.it/

http://ocean-project.eu/