Testing best practice: from debug to full conformance

Dimitri Doutriax, Trialog

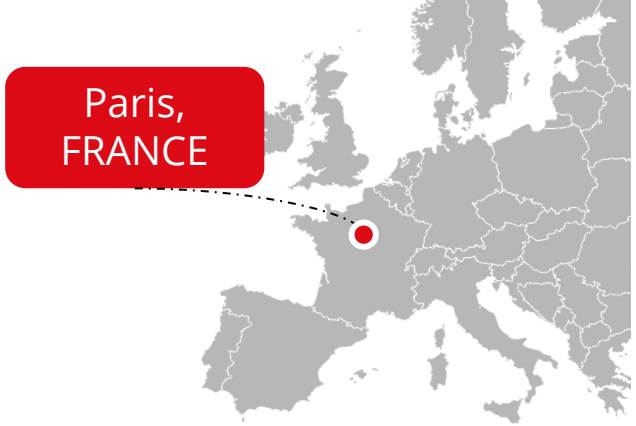


Testing best practices in eMobility

From debug to full conformance

Dimitri Doutriaux, Trialog | OCPP Plugfest, Canberra

Trialog in short



Created in 1987, based in Paris

38 engineers

Value proposition

Support integration and security of technologies and interoperability in innovative systems.

Mains domains

- **Energy** - Smart metering, Smart grid, REN integration
- **Cybersecurity** and **privacy**
- **Electro-mobility** - Smart Charging, Plug&Charge, Interoperability

Agenda

01

Why testing

Understanding the necessity and underlying risks

02

What to test

Key functional areas, features, and edge cases

03

Tools

Leveraging the right automation & testing software

04

Best practices

Proven strategies for reliable interoperability

05

Conclusion

Summary & key takeaways

Why testing

Implementing OCPP does not guarantee interoperability

A bug costs far more in the field than in CI

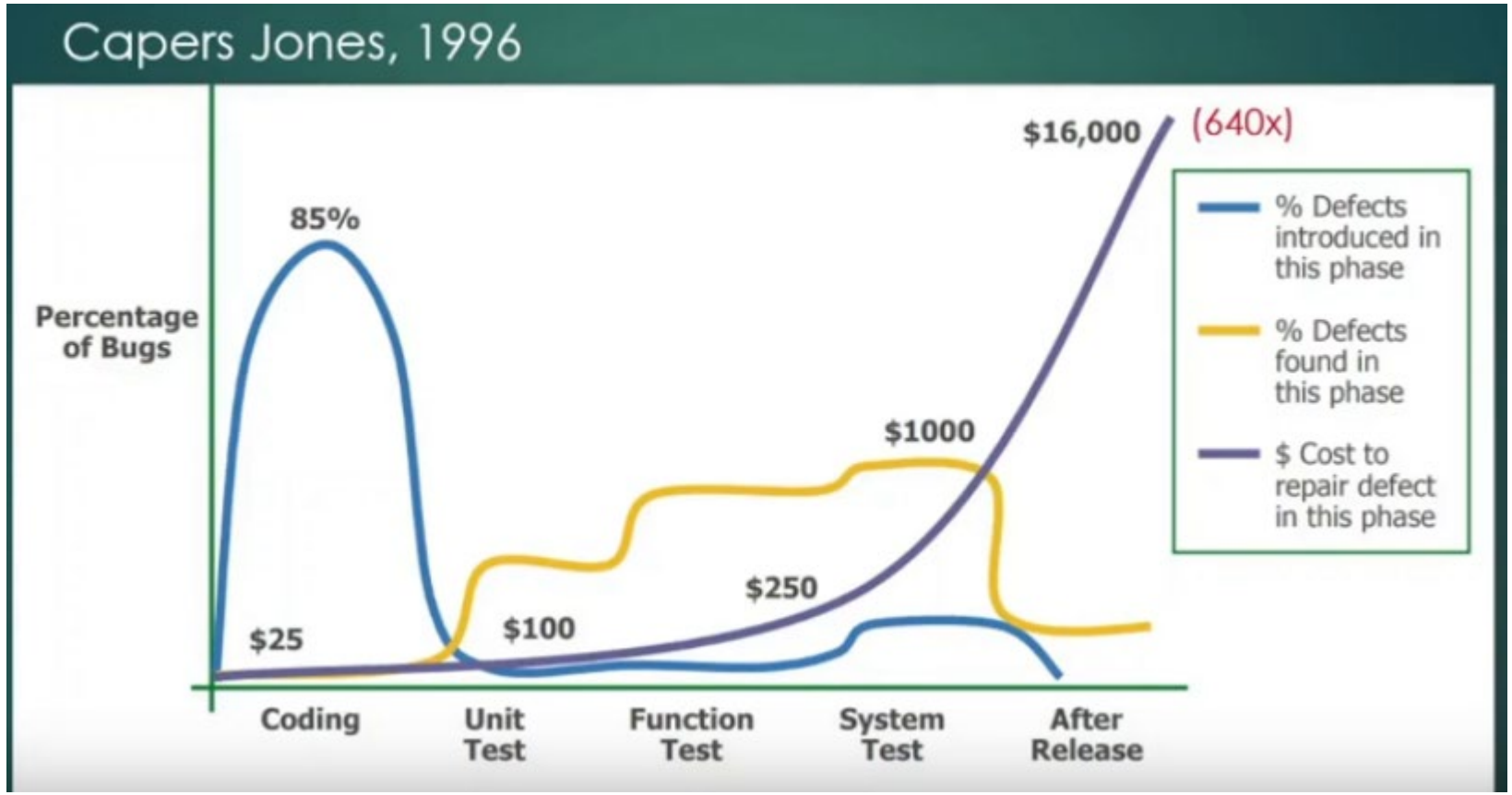
In the field

- Stuck sessions and billing disputes
- Site visits
- Firmware rollout across a whole fleet
- Lost trust with the CPO or CSMS partner

In CI

- A red pipeline
- A fix before the release
- No customer ever sees it

Not testing costs more than spending time on test infrastructure



OCPP gives a common language, not guaranteed interoperability

Optional features: each side picks a different subset

Configuration variables: defaults and supported keys differ

Error handling and timing: retries, timeouts, CALLERROR

Open clauses: two valid readings of the same text

Bugs seen at Plugfests

Wrong timestamp format

Offline queue replayed out of order

seqNo gaps in TransactionEvent

TLS Certificate chain rejected

What to test

Five layers, from cheap and frequent to costly and rare

Five test layers, from every commit to every Plugfest

Layer	What it checks	When it runs
Unit tests	Message handling, state machines, parsing	Every commit
Schema validation	Every message against the official OCPP JSON schemas	Every commit
Functional tests	Full use cases against a simulated peer	Every merge, nightly
Compliance (OCTT)	Conformance to the spec, certification	Before each release
Interoperability	A real partner and its real behavior	Plugfests, 1:1 sessions

Stations and CSMS face different test challenges

	Charging station vendor	CSMS vendor
Test partner	You need a CSMS to drive your station, and every CSMS behaves differently	You need many stations, across models and firmware versions
Physical world	RFID, plug-in, EV behavior, real power flow	No hardware of your own, but stations reboot, go offline and respond late
Where bugs hide	Offline queue and replay, firmware update, certificates	Scale, concurrency, long-running connections

Tools

One tool per test layer, and simulators for the physical world

Unit Tests

Standardizing test automation across environments and language stacks

1. CI/CD Integration Platform

Easily run tests within your preferred CI/CD environment

- GitLab
- GitHub
- Bitbucket
- Jenkins

Fully compatible with custom pipelines, cloud-native runners, and existing DevOps infrastructure.

2. Test Framework Flexibility

Use your preferred test framework native to the programming language of your application.

Functional validation - CSMS

As a CSMS:

- Challenges
  - Need for a set of charging stations available
  - But also electric vehicles
  - Physical actions need to be performed on charging stations
- Solutions
  - Private physical lab
  - Own full software simulation tool from custom/open-source software
  - Fully automated simulation lab

Functional validation - CSMS

Trialog Solution: OCPPvs, Your Virtual Testing Lab

Virtual Infrastructure

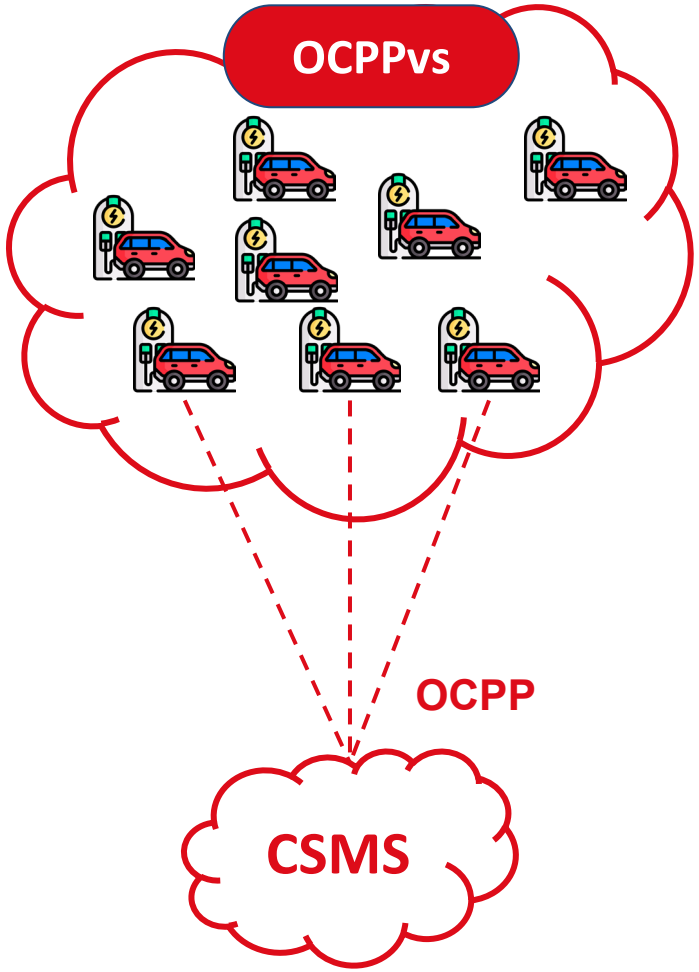
Fully virtual charging stations and EV fleets for complete test coverage.

Cloud Availability

SaaS solution available 24/7 for continuous integration and automated testing.

Industry Standards

OCPP 2.0.1 certified to guarantee full protocol compliance.



Functional validation - Charging Station

As a CS:

- Challenges
  - Need for electric vehicles to test with
  - Physical actions need to be performed on charging stations
  - How to prove conformity to the standard to clients ?
- Solutions
  - Private physical lab with employees EVs
  - Simulated EV
  - Full conformance test tool

Functional validation - CS

Trialog Solution: ComboCS, your simulator and compliance testing tool





Protocol compliance

Best option : compliance testing tools

OCPP Compliance

Gold standard: OCTT

Official tool provided by the Open Charge Alliance.

✓ Full OCPP conformance tests

✓ Automated testing is possible

✓ Extensive test report generated

✓ Same tool used during the OCPP certification

ISO 15118

Private vendor solutions

Check specialized solutions from key market vendors:

Trialog

Other industry vendors



Trialog

Interoperability

Once functional validation and conformance is tested, it is still necessary to test interoperability.

OCA Plugfest

Participate in official Open Charge Alliance testing events to validate compatibility with diverse industry implementations.

1:1 Integration

Direct testing with partner/client systems on the OCPP side to ensure seamless end-to-end communication.

Fleet Validation

Rent the most used cars in the market to validate real-world interoperability across various vehicle models.

Best practices

Make testing cheap enough that it always runs

Automate the pipeline and feed it with real bugs

Automate in CI/CD: fast suite on every merge request, full suite and soak tests nightly

Turn every interop bug into a regression test: capture the OCPP log, replay it as a test, then fix

Trace tests to OCPP use cases: use case IDs (E01, C01…) show coverage and gaps

Test unhappy paths under real network conditions

Unhappy paths

Reconnection & offline replay

Timeouts

Concurrent transactions

Malformed or duplicate messages

Real conditions

Network impairment

Latency, packet loss, TLS with netem or fault-injection proxy

Telemetry & error tracking

Monitor CALLERROR rates per peer and firmware version

Field bugs test plans often skip

Offline & resilience

Network loss
Message queuing
Reconnection backoff
Reboot during a transaction

Security

Security profiles 1 to 3
Certificate install and renewal
Plug & Charge certificates

Firmware updates

Breaks often, painful to fix remotely

Scale and soak

Thousands of stations, long runs to catch leaks and connection drift

Negative tests

Malformed messages,
Unexpected CALLERROR,
Timeouts,
Duplicates

Come to Plugfests with the home testing done

Test every version and profile you claim: 1.6J, 2.0.1 and 2.1 multiply the test matrix

Run OCTT first: Plugfest time is for interop, not for bugs you could find at home

Bring your feature sheet: partners know what to expect

Have log capture ready: every bug found here becomes a regression test

Four things to have in place before your next Plugfest

Unit tests and schema validation: running in CI

Automated core use cases: boot, authorize, transaction, offline, firmware update, against a simulator

A clean OCTT run: for your target version and certification profiles

A feature sheet: OCPP version, profiles, optional features, config keys

The feature sheet links both sides

Stations publish what they support

CSMS vendors check against it before testing

Cheap to write, saves hours at every Plugfest

Testing costs time upfront and pays back at every release

Bugs get more expensive: the later they are found

Regression tests keep paying: each Plugfest bug turned into a test protects every future release

Interoperability is shared: bugs found together and fixed on both sides improve the whole ecosystem



Thank you