

# ECLIPSE – IoT WG MEETING

2017-03-01

## Titan

Dr. György Réthy, Ericsson Hungary (ETH)  
Test Competence Center

[gyorgy.rethy@ericsson.com](mailto:gyorgy.rethy@ericsson.com)

# What is Titan?



- › Titan is a full-featured TTCN-3 test toolset to develop, execute and evaluate tests
- › One of most widely used test automation toolsets in Ericsson
  - 4200+ active users, 15+ years of development
- › Has been open sourced in 2014-2015
  - An [Eclipse Tools](http://www.eclipse.org) project, EPL 1.0 license  
[www.eclipse.org/titan](http://www.eclipse.org/titan)
- › Types of testing
  - Functional (conformance, function, integration verification)
  - Performance (performance, high load, stability)
- › Interest from various areas
  - Avionics, Automotive (C-ITS), Webservices, Mobile user equipment (3G/4G) testing, Smart metering, **IoT testing**



# What is TTCN-3?



## › Standard test language

- Continuous language development and maintenance financed by ETSI

## › Built-in features for automated testing

- Parallelism

- Communication handling

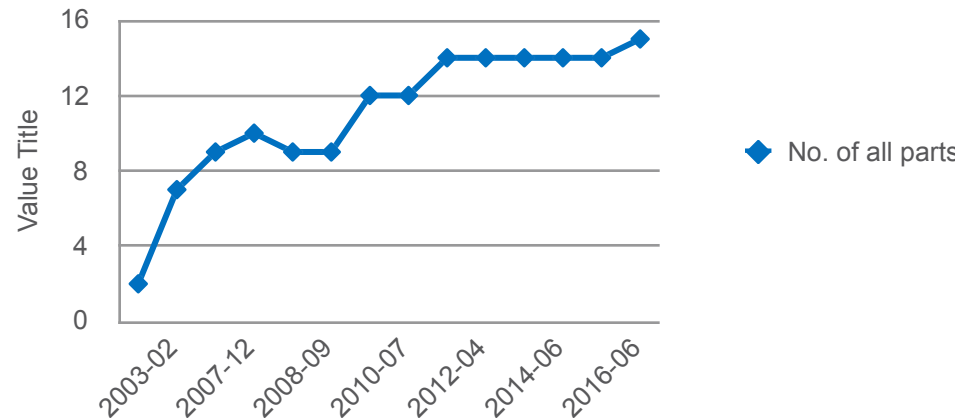
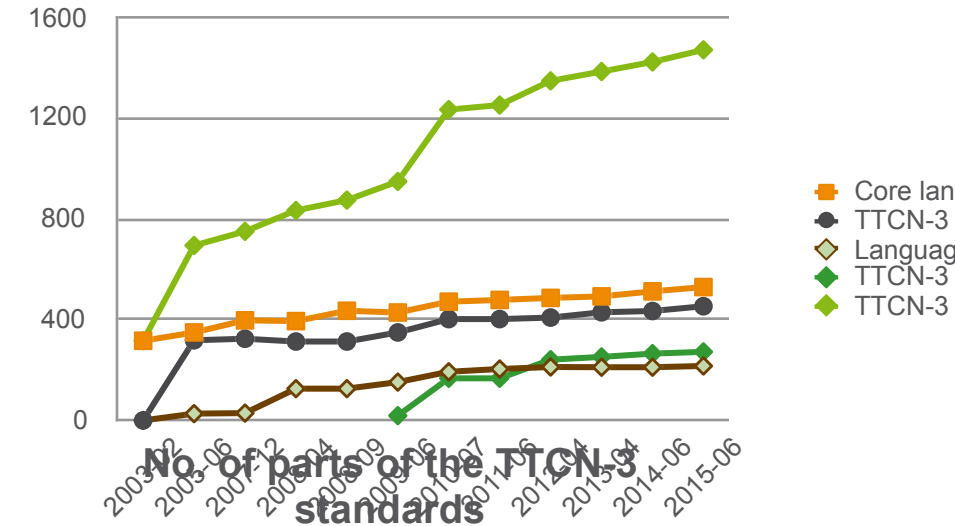
- › Message-based and procedure-based
- › Wide set of wildcards to check received content
- › Handling alternative responses (incl. no response)

- Imports XSD and ASN.1 specifications

- Supports binary (e.g. CoAP), textual (e.g. HTTP), ASN.1 (e.g. mobile, C-ITS), XML and JSON message formats

- Many more...

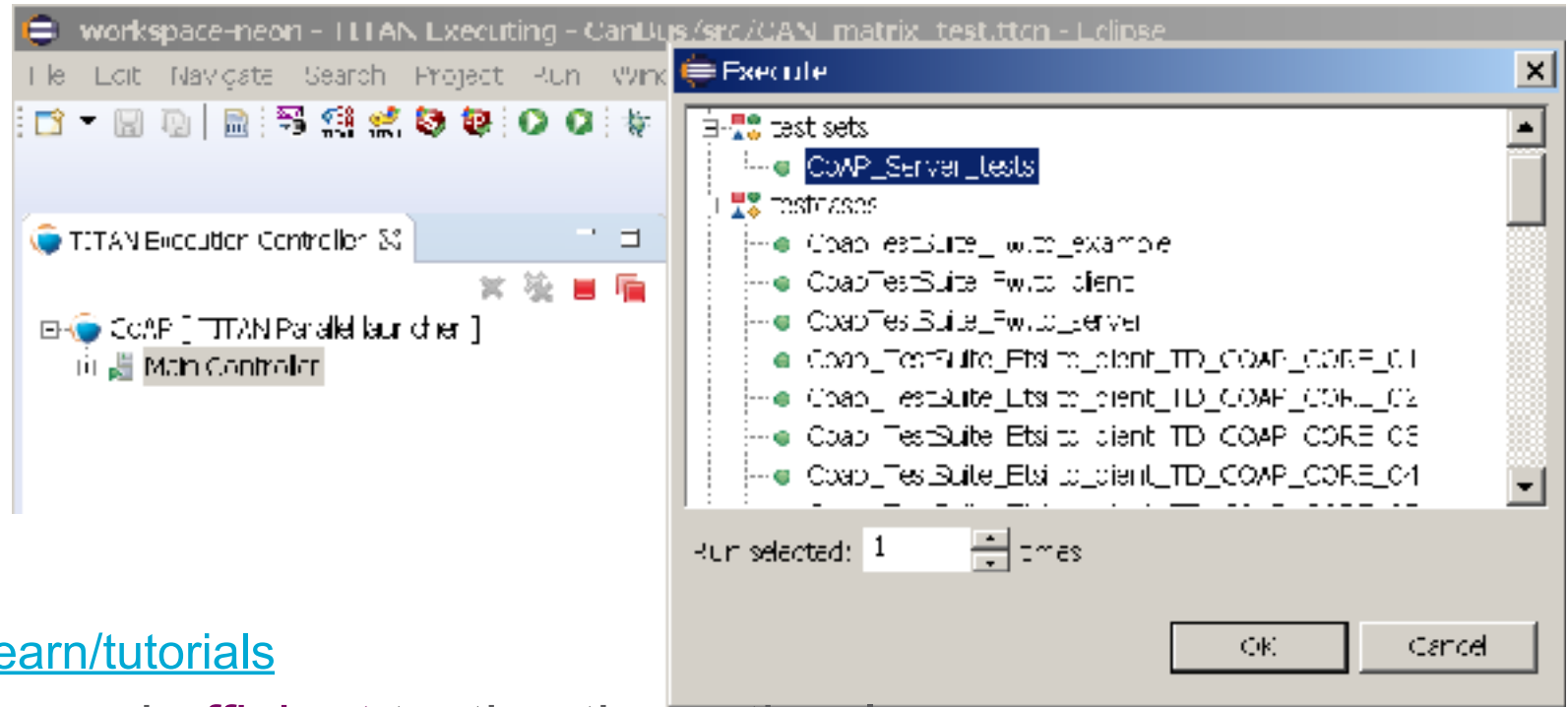
Number of pages of the TTCN-3 standards (without conformance test suite)



# Why should I learn (yet) another language?



- › To **execute** already created test suites
- › In Model-based Testing  
-> **You need not!**



- › To **change** existing tests and **develop own test cases** -> **Yes, you need**

<http://www.ttcn-3.org/index.php/learn/tutorials>

- › TTCN-3 allows **more precise** and **efficient** testing than other languages

- › Comparison with Python

<http://www.site.uottawa.ca/~bernard/A%20comparison%20between%20ttcn-3%20and%20python%20v%202012.pdf>

<https://mail.python.org/pipermail/python-list/2008-August/491315.html>



# IoT projects

# oneM2M IoT conformance testing

 Welcome to oneM2M Conformance Testing Project!

## Licenses

- Eclipse Public License v1.0
- BSD 3-Clause License (OSI Approved open source license)



## Members Logo



## Latest News

[2017 January]

- **oneM2MTester version 1.0 released!!**
- Huawei (China), shows high interest in oneM2MTester project and joined oneM2MTester project in January 2017!! Welcome Huawei to join oneM2MTester project!!

[2017 February]

- oneM2MTester version 1.2 is available for download!!

<http://support.iotocean.org:7591/onem2m/conformancetesting/oneM2MTester>



- › Develops security feature tests using TTCN-3
- › Model-based security testing with TTCN-3 code generation  
<http://www.armour-project.eu/wp-content/uploads/2016/08/D21-Generic-test-patterns-and-test-models-for-IoT-security-testing.pdf>  
<http://www.armour-project.eu/wp-content/uploads/2016/08/D22-Test-generation-strategies-for-large-scale-IoT-security-testing-v1.pdf>



Police dog tested, benchmarked and certified security & trust solutions for large-scale IoT by a European FP7 project in 2014



**TOOLBOX**



**EXPERIMENTS**



**CERTIFICATION**

### Latest Posts



Understanding the limits of LoRaWAN  
July 2016  
Exploring the limits of LoRaWAN: the gatekeeper of LPWAN... TBC



Minimal Security Framework for OTS-II  
July 2016



- › German project supported by Industrial and academic partners and the Federal Ministry of Economics and Technology  
<http://www.iot-t.de/en/>
- › IoT Testware is being developed by Fraunhofer FOKUS SQC
  - It is considering free online services, dedicated protocol tester and test features from IoT development kits



OVERVIEW ▾

CASE STUDIES ▾

INNOVATIONS AND

- functionality
- interoperability
- robustness
- safety
- trustworthiness

of the technologies and to improve the quality of IoT solutions.

## IOT TESTWARE

Fraunhofer FOKUS – System Quality Center

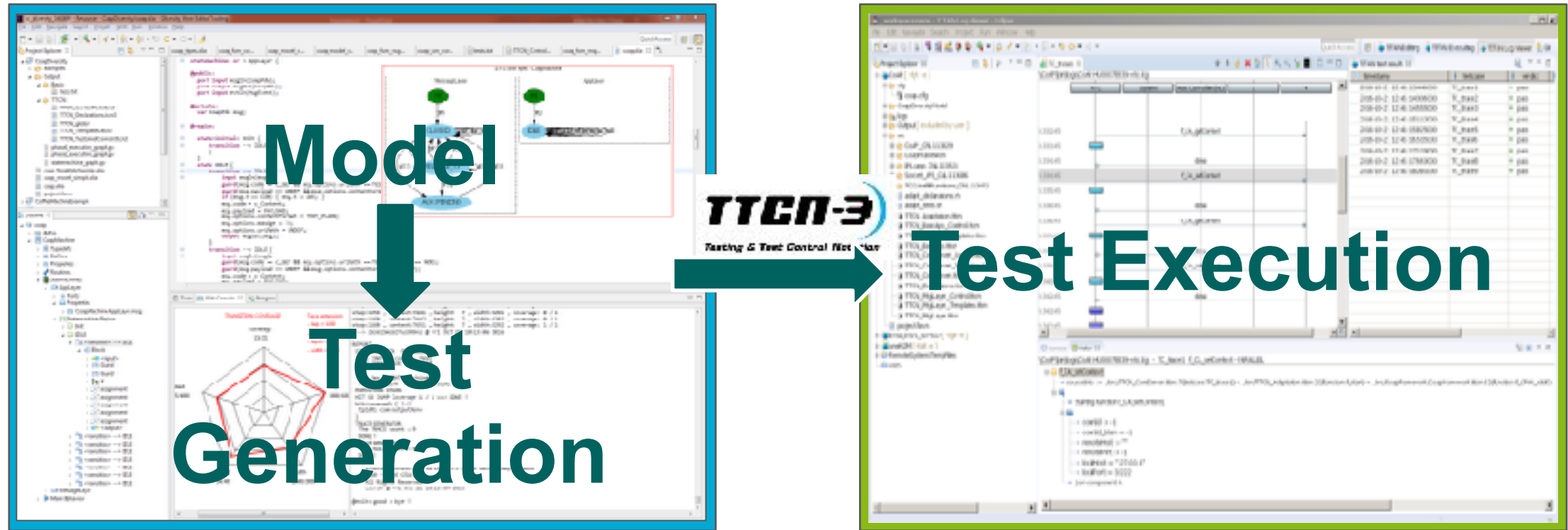
Special attention is given to e.g. security and scalability testing. Here we do include also advanced test methods like fuzzing or risk-based testing.



# Open source Model-driven test toolchain



- › We have carried out a successful pilot with CoAP



**Diversity**



<http://projects.eclipse.org/projects/modeling.efm>



**ERICSSON**