



# **OPC UA**™ **Device Support**

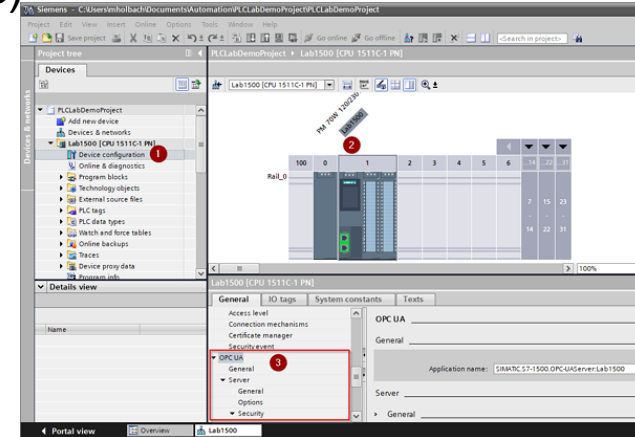
## *Overview and Status*

Ralph Lange, ITER Organization

*Disclaimer: The views and opinions expressed herein do not necessarily reflect those of the ITER Organization*

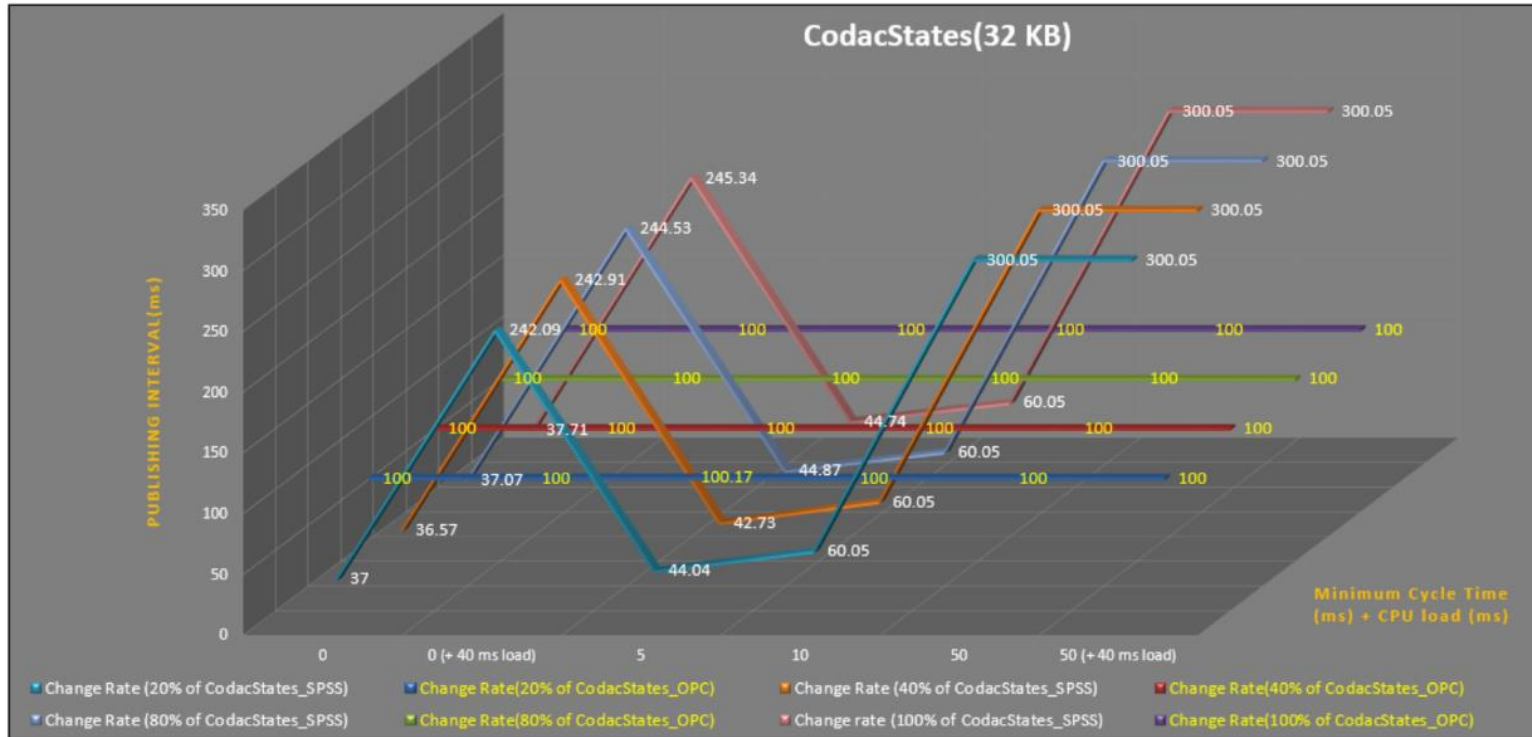
# Why OPC UA?

- Industrial standard (2006) to interface SCADA to PLCs
  - Covers data, alarms, events, historical data, remote methods
- Based on OPC Classic (Microsoft; 1996), plus
  - Portability
  - Safety/security (authentication, encryption)
  - Information modeling (user defined structures)
- Gaining momentum in industrial context as universal integration standard
- Siemens S7-1500 series PLCs include an embedded OPC UA server



# Performance

- It all depends – mostly on the server tests by TCS: S7-1518, 32kB data (integers), “empty” / 40ms artificial load



# EPICS Device Support

- Based on commercial C++ Client SDK
  - Vendor: Unified Automation (<4k€ for sources and 1yr support)
  - Binaries can be deployed/distributed royalty-free
  - Platforms: Windows and Linux
- Prototype done by Bernhard Kuner (HZB / BESSY II)
- ITER use cases tested by F4E (Spain) and TCS (India)
  - Against S7-1516/1518 embedded OPC UA server
  - Against WinCC-OA embedded OPC UA server

- Requirements Specification v1.1: <https://bit.ly/opcua-srs-11>
- Design done (no formal doc yet)
- Implementation in progress, “usable pre-release” state
  - All basic data types and arrays thereof (*read/write/subscribe*)
  - User-defined structures (*read/write/subscribe*)
  - Server-side queues (sample fast, publish slow)
  - Throttling of read and write requests
  - Support for all applicable record types in EPICS Base
- Used in production at ITER, BESSY II, ASIPP
- Under evaluation at ESS, Fermilab, AS, LNLS, Varian

- Soon:
  - Configurable behavior on connection (similar to PINI)
  - Read EPICS time stamp from within data structure
  - Safety/security features (authentication, encryption)
- Later:
  - Support for OPC UA methods (remote execution of PLC FBs)
  - Support for (free) open62541 client library
- ESS has recently pledged to contribute to S7 end-to-end tests
- Under EPICS license, upstream repository on GitHub:  
<https://github.com/ralphlange/opcua>