basysKom

OPC UA APPLICATIONS WITH OPEN62541

Embedded World Conference
02/2019

AGENDA

- About us
- What is open62541?
- The ecosystem
- Applications
- Conclusion

ABOUT US

About basysKom

Software development service company located in Darmstadt and Nürnberg

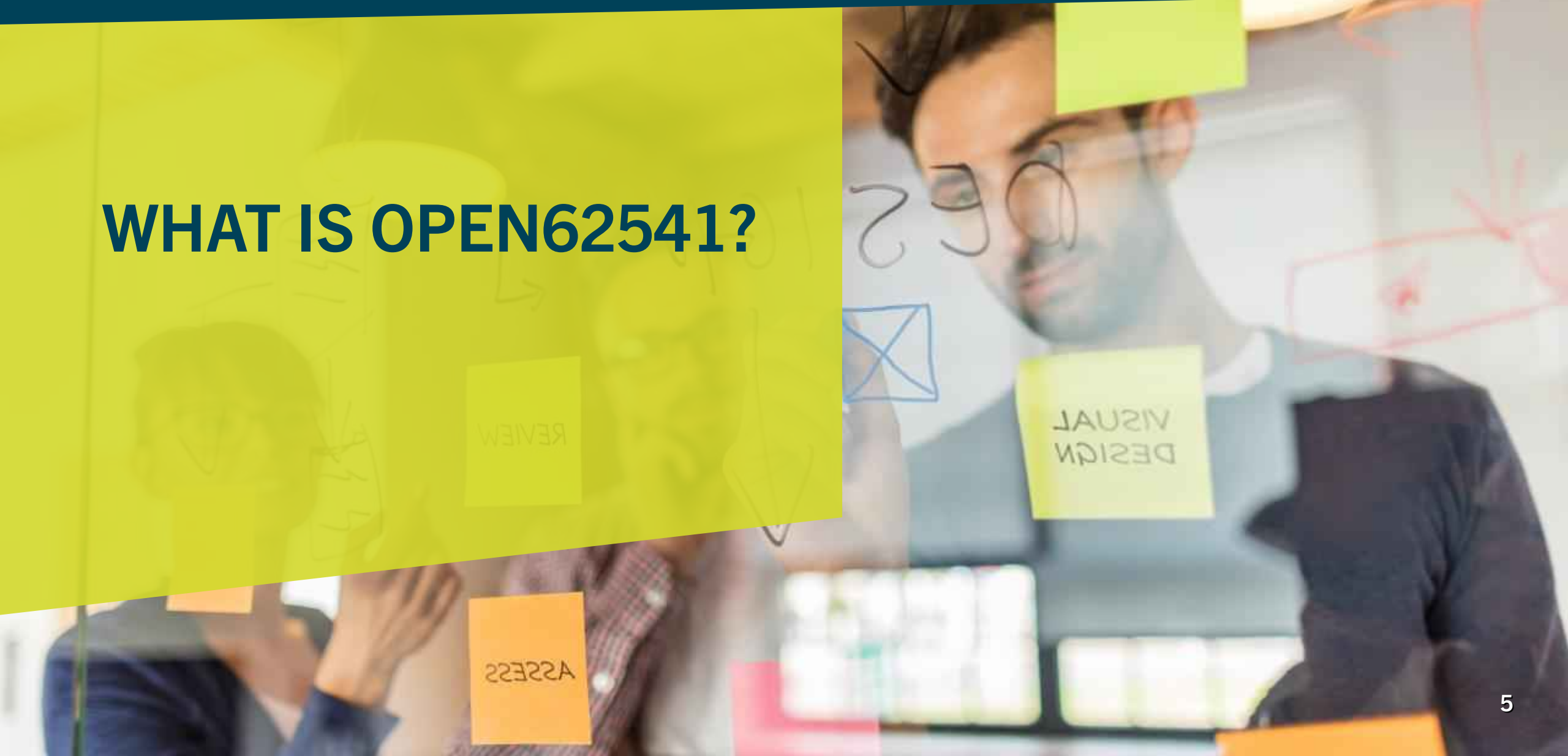
HMI and application development

Typical customer is a from DACH (machine manufacturing, measurement applications, automation)

30 people

Open Source => open62541 & Qt OPC UA

WHAT IS OPEN62541?



What is open62541?

What is OPC UA?

- Protocol and framework for industrial applications
- Developed as open standard by the OPC-Foundation

An independent open source implementation of IEC62541 / OPC UA

Stack + SDK (Server/Client) + Tooling for code generation

Implemented in C99

Web: open62541.org

What is open62541?

Lean, Configurable

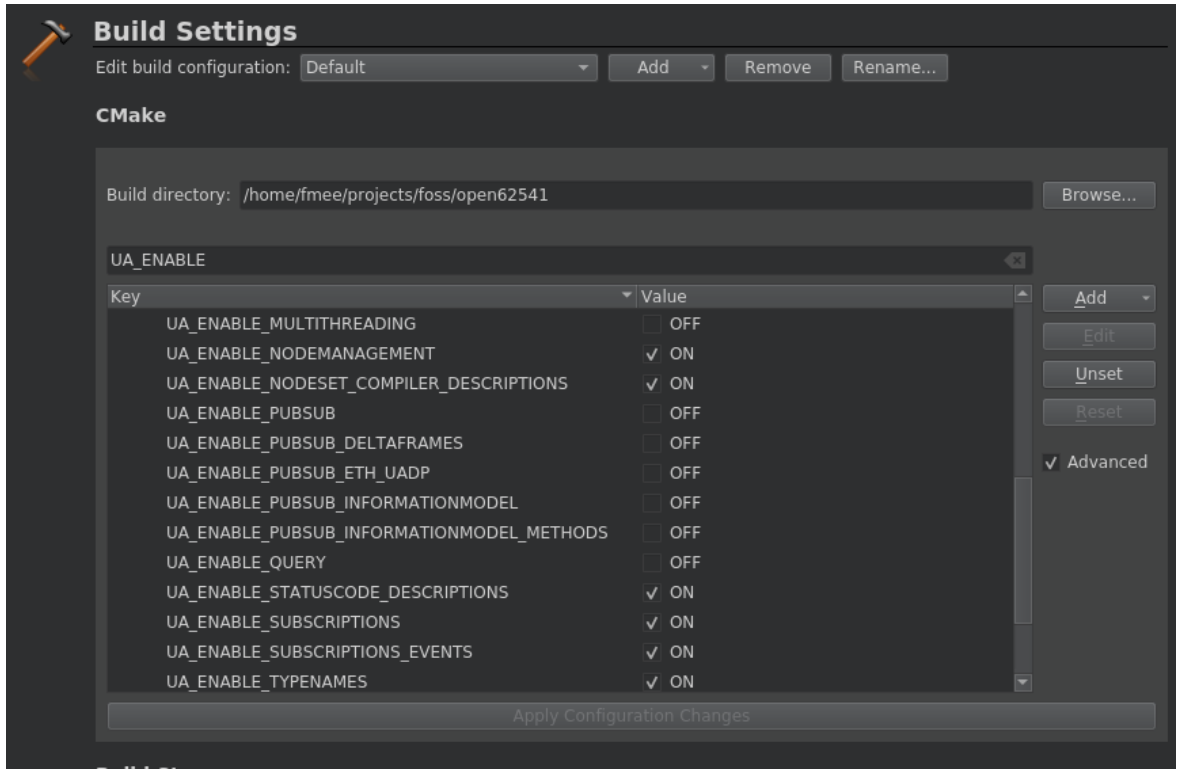
Quite portable

- Platform specific functionality is implemented via plugins
- Windows (Visual Studio, MinGW), Linux, Android, IOS, Microcontroller, …

A small set of dependencies

Focused on embedded use-cases

- Implements the Micro Embedded Device Server Profile (plus more and more additional features)



Development model

Developed on master

Releases are branched

- API stability on the released versions, bugfixes only

Last release is 0.3 (final release in december 2018 after ~1 year of RCs)

master will become 0.4

- Quite a few new features currently on master

Supported features

See: https://github.com/open62541/open62541/blob/master/FEATURES.md

TL;DR

- Encoding OPC UA Binary
- Transport UA-TCP UA-SC Binary
- Encryption (client-side only on master)
- Authentification (Anonymous, User Name)
- Server side: almost all services (TransferSubscription, Query Service are missing)
- Client side: all services

Supported features (0.3)

Read/Write of attributes

Monitoring for datachanges

Monitoring for events (client-side)

Method calls

Browsing

Resolving of browse paths

Adding/removing nodes and references

Supported features (master, 0.4)

Publish/Subscribe

Events (Server side)

- Event filters are missing still

Local Discovery

Historical data access (partial)

- Event history as well as the information model from part8 are missing

ECOSYSTEM

Open Source

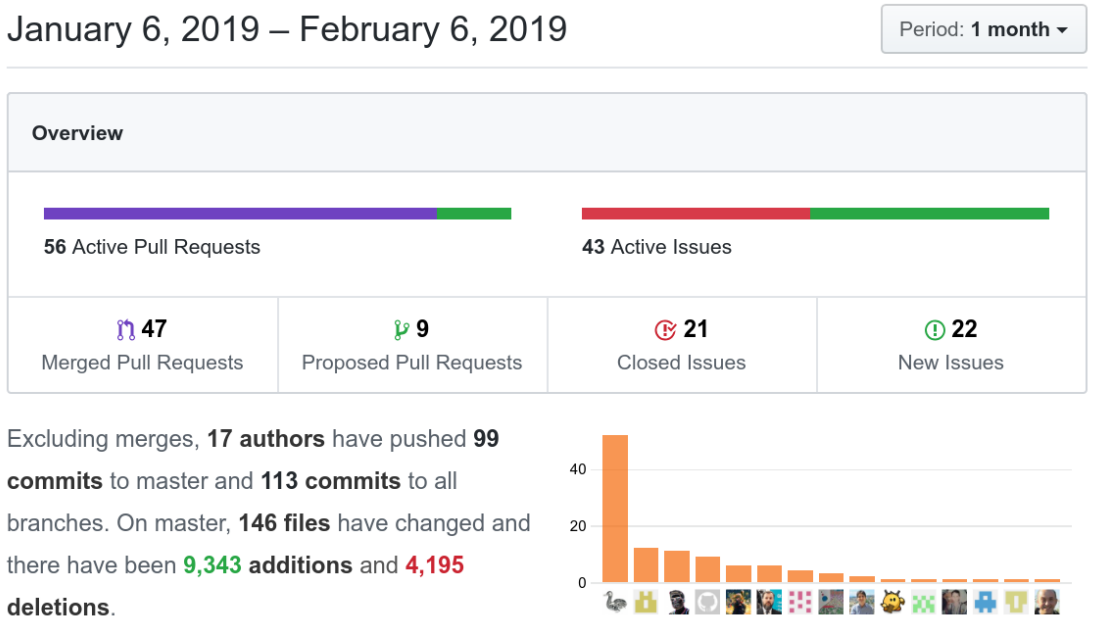
Active Open-Source project

- Hosted at github
- github.com/open62541/open62541/
- First commit end of 2013

Licensed as MPL2

- Easy to use in commercial products

50k+ downloads of the 0.3 release



Maintainers

The founders

Steering of the project

Most of the contributions

Fraunhofer IOSB

ACPLT Chair of Process Control Engineering | RWTH AACHEN UNIVERSITY

TECHNISCHE UNIVERSITÄT DRESDEN

Chair of PROCESS CONTROL SYSTEMS ENGINEERING

fortiss

Ecosystem

Contributing Companies

- Active contributions within the last two years

Commercial Support

- Need to be contributors

APPLICATIONS

Lets create a server

Show a very simple server

- Still supports browsing
- Still supports data changes / subscriptions

Make use of modeling and code generation

Show how to implement a „method node“

Task

„have a server hosting a variable node that can be incremented via a remote method call“

Creating an address space

Either

- manually in C
- Using code generation

```c
#include "open62541.h"

int main()
{
    UA_ServerConfig *config = UA_ServerConfig_new_default();
    UA_Server *server = UA_Server_new(config);

    bool running = true;
    return UA_Server_run(server, &running) == UA_STATUSCODE_GOOD ? 0 : 1;
}
```

A simple model

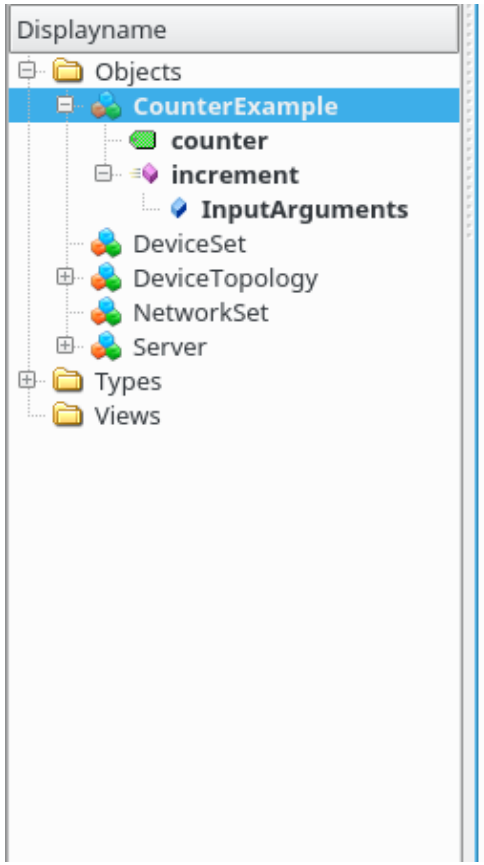
CounterExample → object node

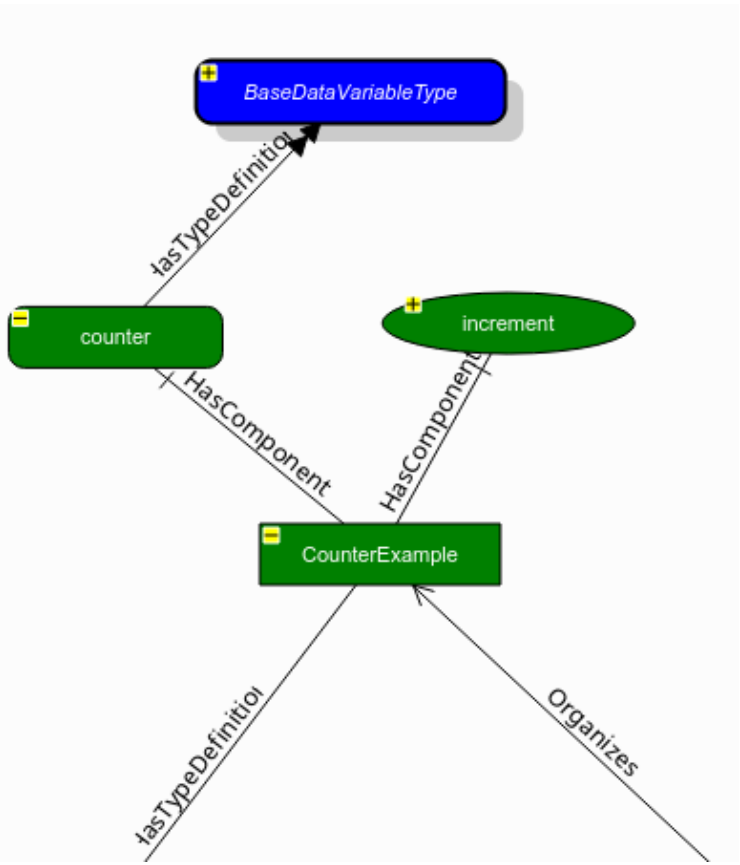
- counter → Variable node
- increment → Method node

Created via the UaModeler

XML export

- → counterexample.xml





Code generation

nodeset_compiler.py

- Part of the tooling shipped with open62541

Commandline for our example

- `./nodeset_compiler.py --types-array=UA_TYPES --types-array=UA_TYPES --existing ../../deps/ua-nodeset/Schema/Opc.Ua.NodeSet2.xml --xml ~/counterexample.xml ua_namespace_counterexample`
- → ua_namespace_counterexample.c/.h

C-Compiler

- `gcc -DUA_ENABLE_AMALGAMATION main.c ua_namespace_counterexample.c open62541.c -o counter_example`

```c
#include "open62541.h"
#include "ua_namespace_counterexample.h"

static UA_NodeId counterVariableId;

static UA_StatusCode handleIncrement(UA_Server *server, const UA_NodeId *sessionId,
    void *sessionContext, const UA_NodeId *methodId,void *methodContext, const UA_NodeId *objectId,
    void *objectContext, size_t inputSize, const UA_Variant *input, size_t outputSize,
    UA_Variant *output)
{
    UA_Variant currentValue;
    UA_Server_readValue(server, counterVariableId, &currentValue);
    UA_UInt32 sum = *(UA_UInt32 *)input[0].data + *(UA_UInt32 *)currentValue.data;
    UA_Variant result;
    UA_Variant_init(&result);
    UA_Variant_setScalar(&result, &sum, &UA_TYPES[UA_TYPES_UINT32]);
    return UA_Server_writeValue(server, counterVariableId, result);
}

int main()
{
    UA_ServerConfig *config = UA_ServerConfig_new_default();
    UA_Server *server = UA_Server_new(config);

    ua_namespace_counterexample(server);
    UA_UInt16 nsIndex = UA_Server_addNamespace(server, "http://basysKom.com/CounterExample/");
    counterVariableId = UA_NODEID_NUMERIC(nsIndex, 6004);
    UA_Server_setMethodNode_callback(server, UA_NODEID_NUMERIC(nsIndex, 7002), handleIncrement);
    bool running = true;
    return UA_Server_run(server, &running) == UA_STATUSCODE_GOOD ? 0 : 1;
}
```

Compilation + Demo

Commandline

- `gcc -DUA_ENABLE_AMALGAMATION main.c ua_namespace_counterexample.c open62541.c -o counter_example`

Based on open62541: Qt OPC UA

C++/Qt module with the goal of making it easy to integrate OPC UA in Qt applications

- Open Source

Standard-API

Several backends

- Unified Automation
- open62541

https://doc-snapshots.qt.io/qtopcua/index.html

https://blog.basyskom.com/building-qt-opc-ua-with-open62541/

CONCLUSION

Conclusion

open62541 is

- currently the most active open source community around a C/C++ stack
- good choice for embedded uses of OPC UA (as opposed to IT-focused)
- is gaining new features fast

THANK YOU!

QUESTIONS?