Distributed and Fun !

by

Andy Götz (ESRF)

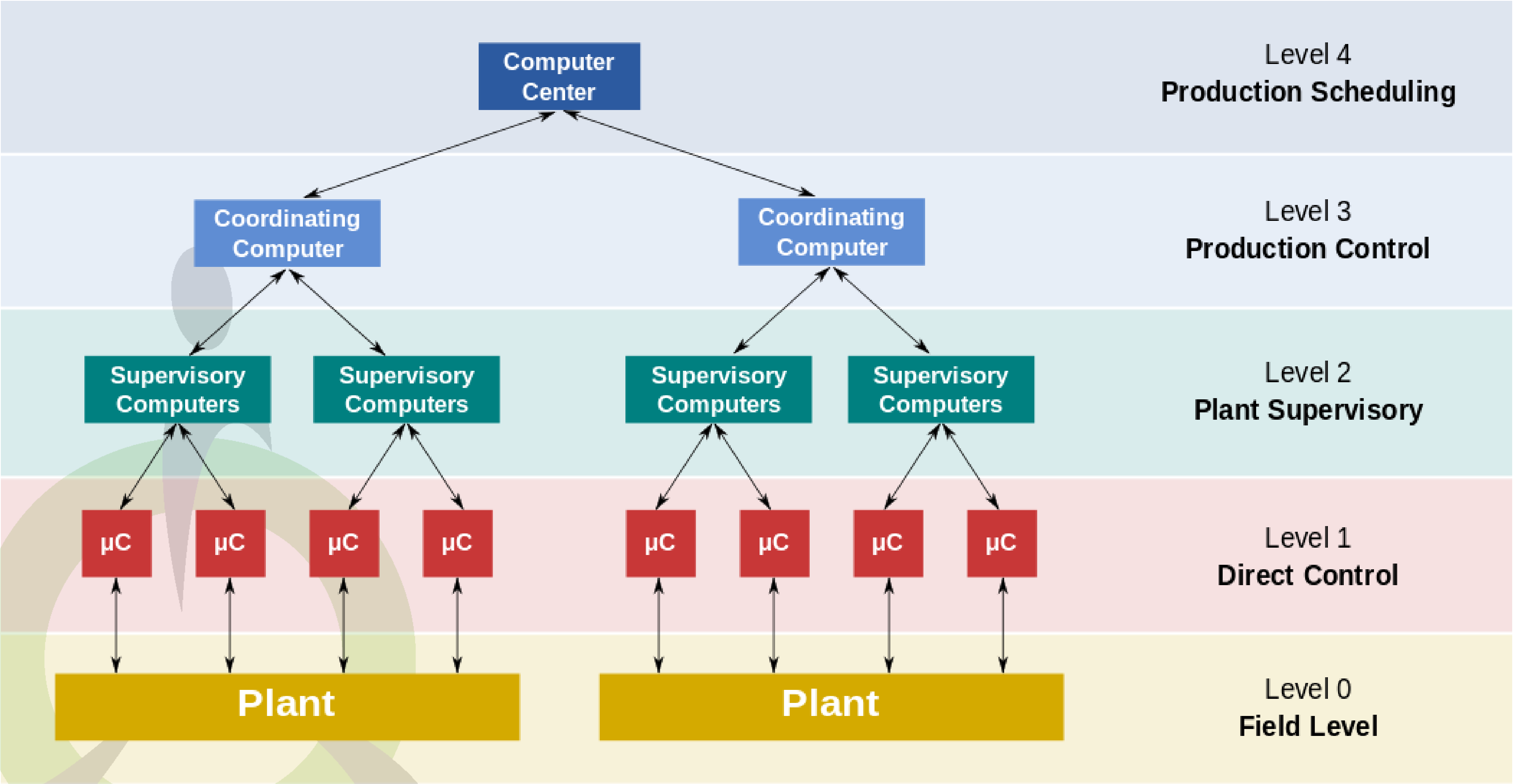
+

Mihael Koep (Softwareschneiderei)

HOW EASY IS TANGO?

- **Basic TANGO is not difficult –** students write their first device server within hours
- **TANGO is a very rich toolkit** – 15 years of development have seen it adapted to many use cases
- **Distributed programming** - learning how to design, program and debug a distributed system is the challenging part

A DISTRIBUTED CONTROL SYSTEM



SOLUTIONS OUT THERE

- **TANGO and EPICS** – the two most mature open source distributed control systems

- **Industrial systems** - other alternatives are industrial systems which are more restrictive in terms of hardware + features

- **Build your own** – the most difficult option !

WHAT IS TANGO?

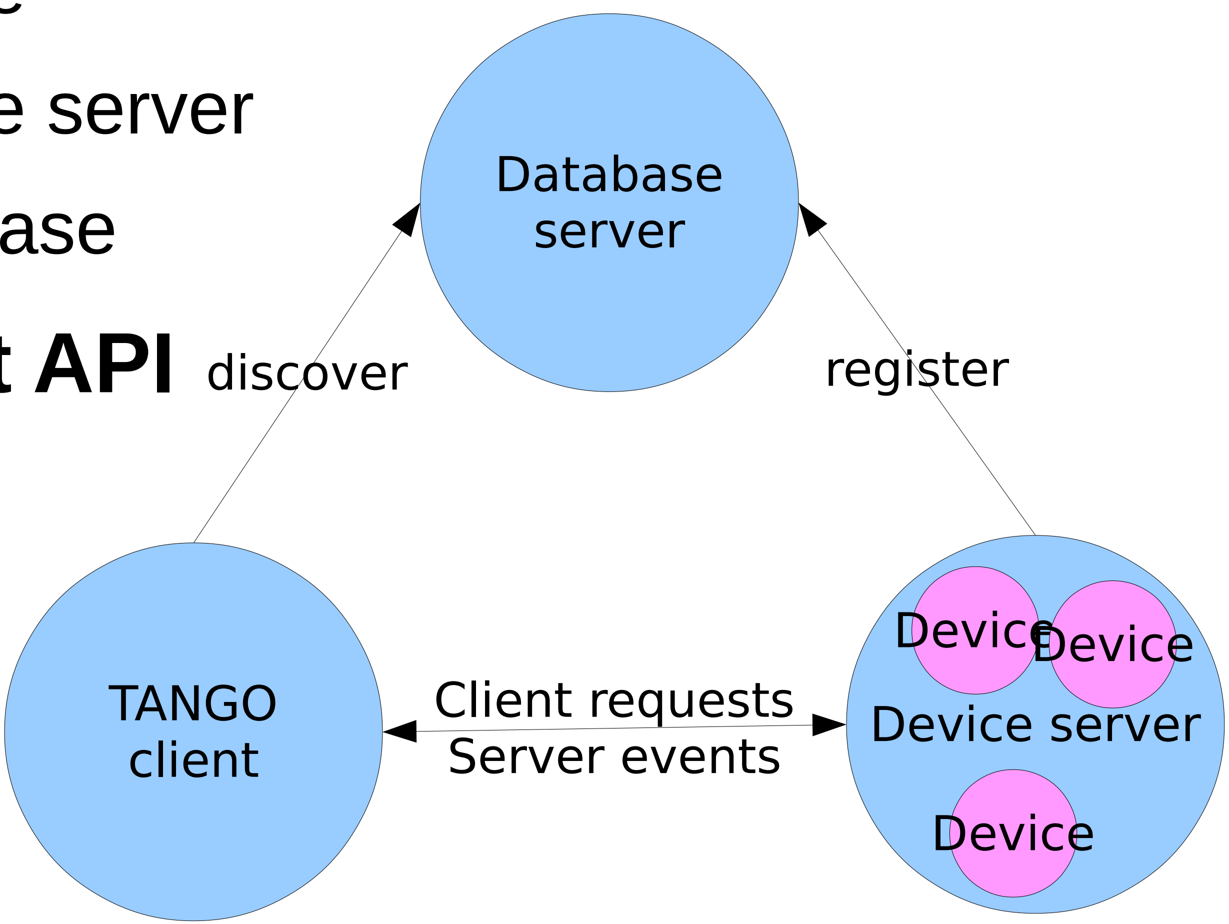
- High performance framework for distributed control systems
- Multi-Language - C++, Java, Python, JavaScript
- Multi-Platform - Windows, Linux, Macintosh etc.
- Integration into many 3rd-party systems (Matlab, LabVIEW, IGOR Pro etc.)
- Unified interface to hardware devices and equipment

HOW TANGO COLLABORATION WORKS

- **One collaboration meeting per year**
- **One TANGO coordinator per site**
- **A mailing list (tango@esrf.fr)**
- **Project Web Site http://www.tango-controls.org**
  - Documentation, Forum, Howto, Events, News
- **Open Source Software (OSS) hosted on SourceForge**
  - Change requests
  - Patches
  - Bug reports
  - Virtual machine
  - Packages (Debian, Windows)

TANGO CONCEPTS

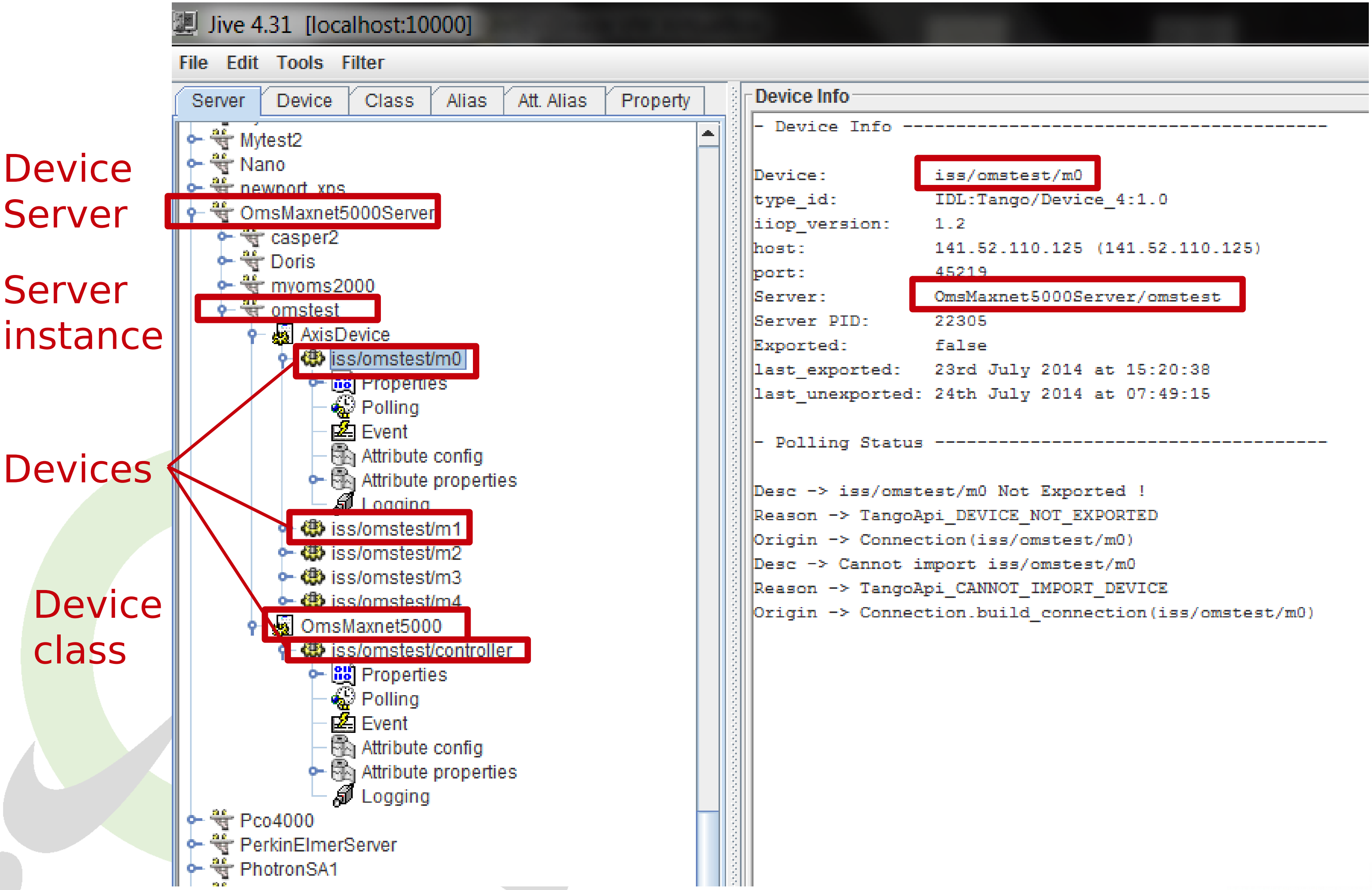
- **Three major building blocks**
  - TANGO device
  - TANGO device server
  - TANGO database
- **TANGO client API**



TANGO DATABASE

- **Database server is a TANGO server with a device itself**
- **MySQL-backend for storing configuration**
  - Register device servers and devices
  - Remember device properties
  - Memorize device attributes (optional)
- **Communication device end points (IOR) for p2p-communication**

DATABASE BROWSER - JIVE



Device Server

Server instance

Devices

Device class

```
- Device Info ----------------------------------------

Device:          iss/omstest/m0
type_id:         IDL:Tango/Device_4:1.0
iiop_version:    1.2
host:            141.52.110.125 (141.52.110.125)
port:            45219
Server:          OmsMaxnet5000Server/omstest
Server PID:      22305
Exported:        false
last_exported:   23rd July 2014 at 15:20:38
last_unexported: 24th July 2014 at 07:49:15

- Polling Status -------------------------------------

Desc -> iss/omstest/m0 Not Exported !
Reason -> TangoApi_DEVICE_NOT_EXPORTED
Origin -> Connection(iss/omstest/m0)
Desc -> Cannot import iss/omstest/m0
Reason -> TangoApi_CANNOT_IMPORT_DEVICE
Origin -> Connection.build_connection(iss/omstest/m0)
```

TANGO DEVICE SERVER

- **Runnable piece of software containing TANGO devices**
  - Device classes are defined in the code
  - Device instances are defined in the TANGO database
- **Server instances are registered at the TANGO database**
  - Identified by executable name + instance name
- **Creates devices specified in database on startup**
- **Can be written in C++, Java or Python**

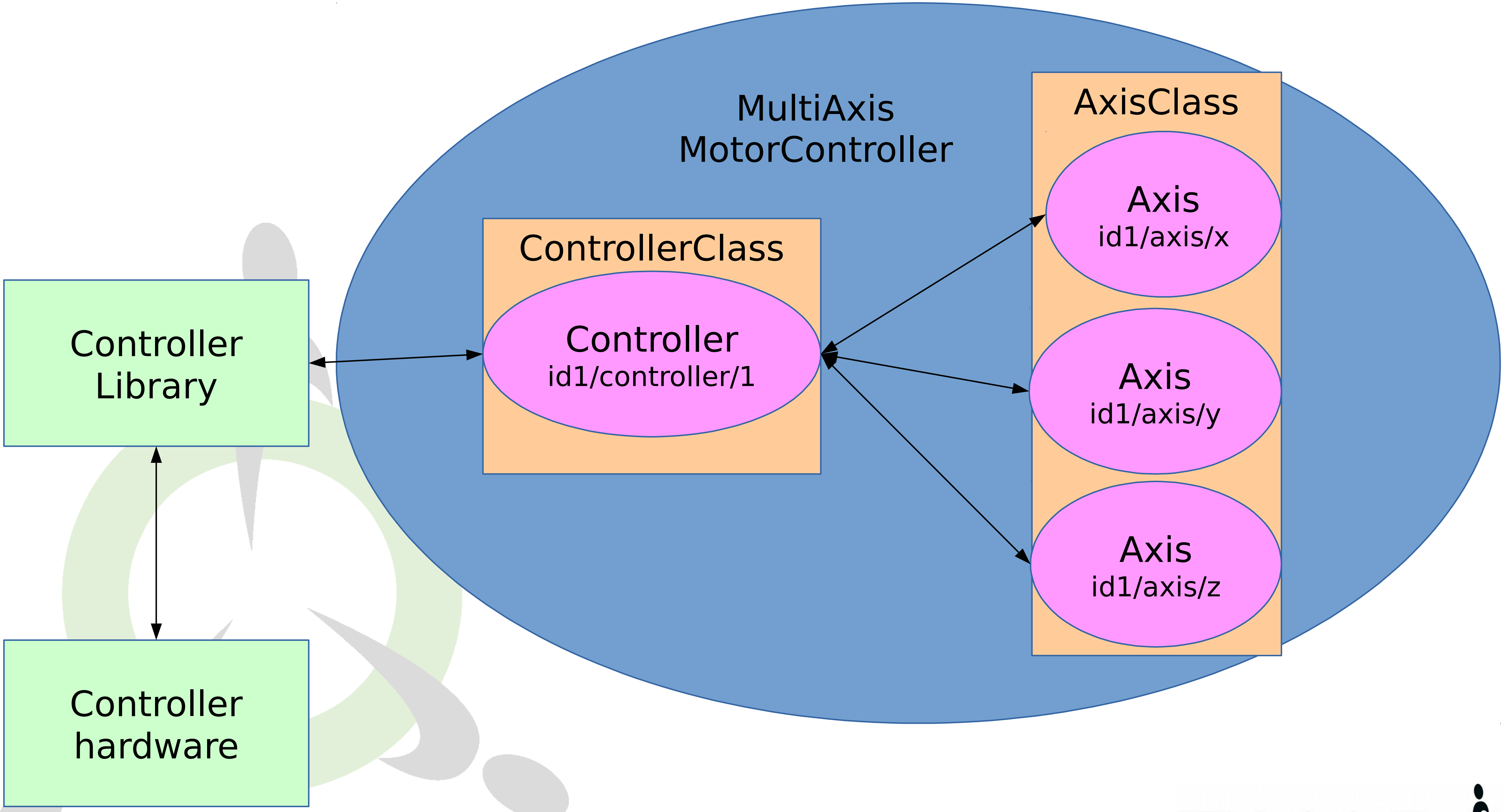
TANGO DEVICE

- **Fundamental element of interaction**
- **Interface to existing hardware or logical devices**
- **Identified by a three field name**

  “domain/family/member”

- **Every device belongs to a TANGO class**
- **Configured by device properties**
- **Exposes attributes and commands**

TYPICAL DEVICE SERVER

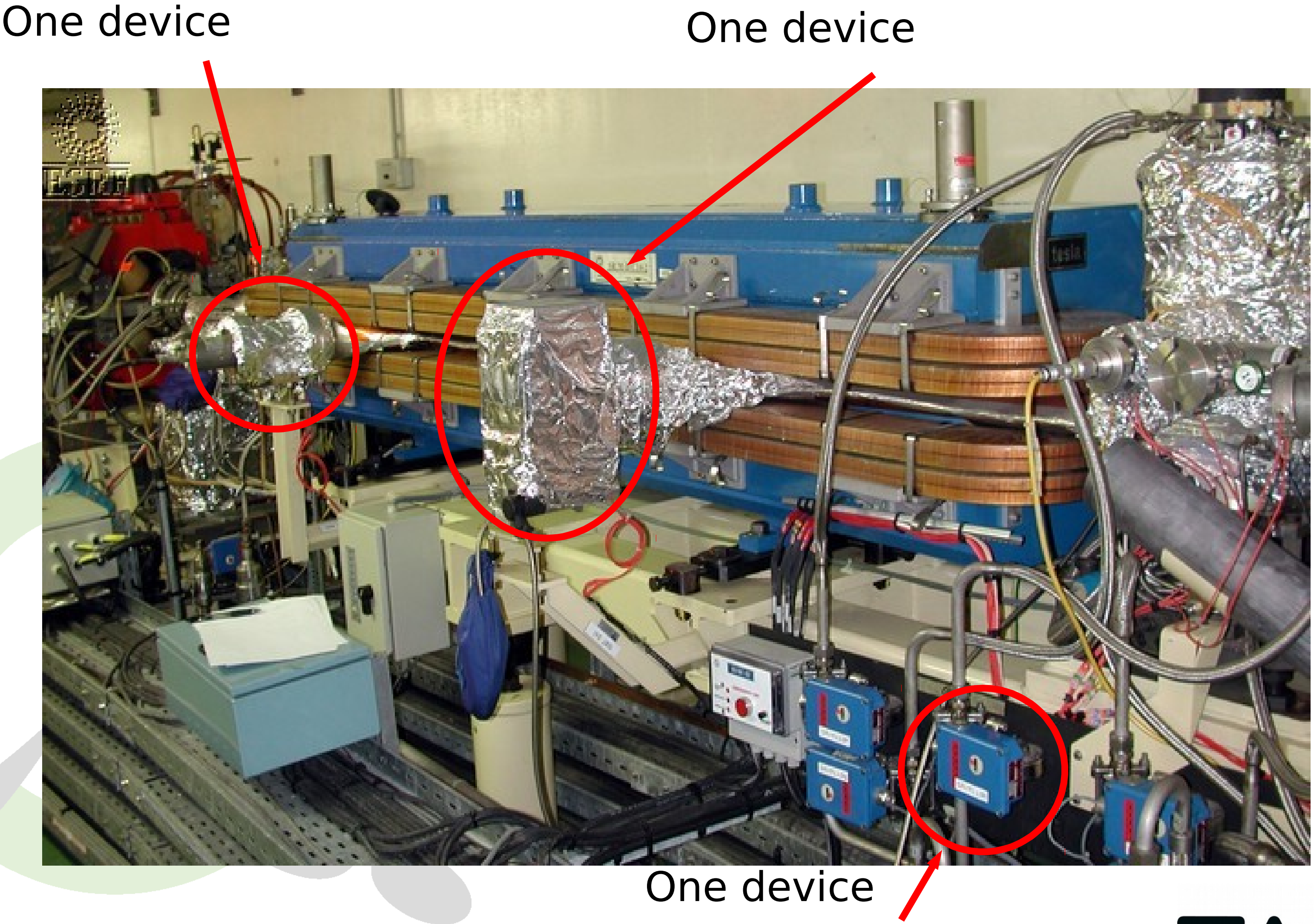


A CLOSER LOOK AT TANGO DEVICES

- **Commands:** perform an action on a device
- **Attributes:** represent physical values
- **Properties:** configuration used at initialisation

  e.g. IP adress, default shutter time
- **State and Status:** indicators for device state

REAL WORLD DEVICES



COMMANDS

- **Have one input parameter and a return value**
- Only limited set of data types (approximately 15)
- But also arrays
- e.g. PowerOn(), Stop(axisNumber), StopAll()

ATTRIBUTES

- **Self-describing data via attribute properties**

  e.g Description, Unit, data_type, min/max, alarm values

- **May be read-only, write-only or read-write**
- **All typical primitive data types like boolean, integer, double, string etc.**
- **Three data formats :**
  - Scalar (one value)
  - Spectrum (one-dimensional array)
  - Image (two-dimensional array)

PROPERTIES

- Properties are stored in the TANGO database

- Manage using the tool Jive

- Can be defined at class, device and attribute level

- Basic data types as scalar or array values

STATE + STATE MACHINE

- State management is essential for control systems

- 14 defined states are available

  e.g ON, OPEN, MOVING, FAULT, ALARM etc.

- Explanatory message available as Status attribute/command

- Support through „state machine“ and „allowed states“

TANGO CLIENTS

- **Can be written in C++, Java, Python**
- **Implementations for many tools exist**

  e.g. Matlab, LabView, IgorPro, concert

- **Different communication mechanisms**

  Synchronous calls

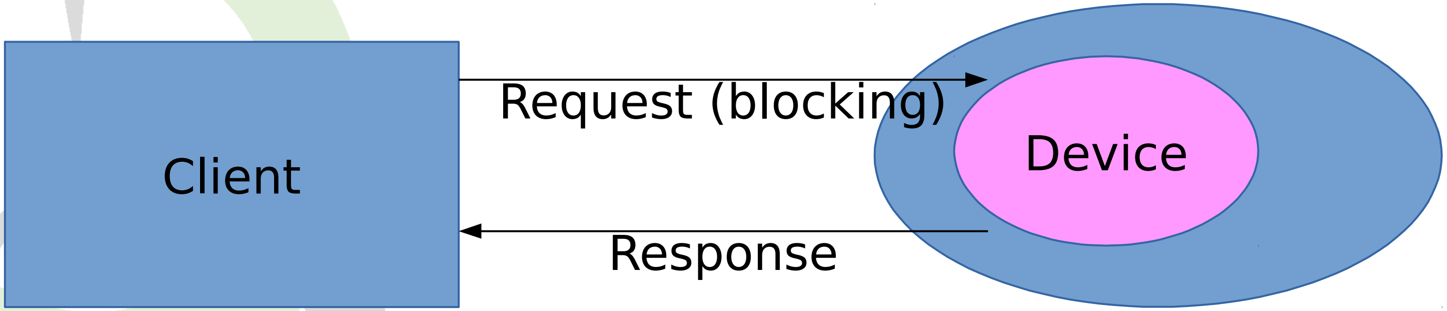
  Asynchronous calls

  Events

  Group Calls

SYNCHRONOUS CALLS

- Network transparency etc. using DeviceProxy
- Easy to use calls like command_inout(), read_attribute(), write_attribute()
- Result objects can contain data and metadata
- Exceptions are of type DevFailed



ASYNCHRONOUS CALLS

- **Non-blocking request to a device**
- **Device notifies clients via callback**
- **No changes on the server side required**
- **Supported for**

  command_inout

  read_attribute(s)

  write_attribute(s)

EVENTS

- **Different communication paradigm**

  No polling from the clients

  Devices notify clients about „interesting“ changes

  Only available for attributes

- **Clients need to subscribe to events and are notified using callbacks**
- **Different types like Periodic, Change, Data ready, Archive etc.**

- **Jive**

  Database management

- **POGO**

  Device server code generator

- **Astor**

  Device server control

- **AtkPanel**

  Generic device gui

- **Jdraw**

  Synoptic editor

APIS AND FRAMEWORKS

- **JTango for Java**
- **PyTango for Python**
- **GUI-Toolkits**
  - ATK for Java/Swing
  - Taurus for Python/Qt4
  - Qtango for C++/Qt4
- **mTango for RESTful http**
- **Sardana for beamlines**
- **Jddd for supervision**

QUESTIONS