



PRECISION AND VACUUM TECHNOLOGY

Aleksander Stanik
Chief Software Architect

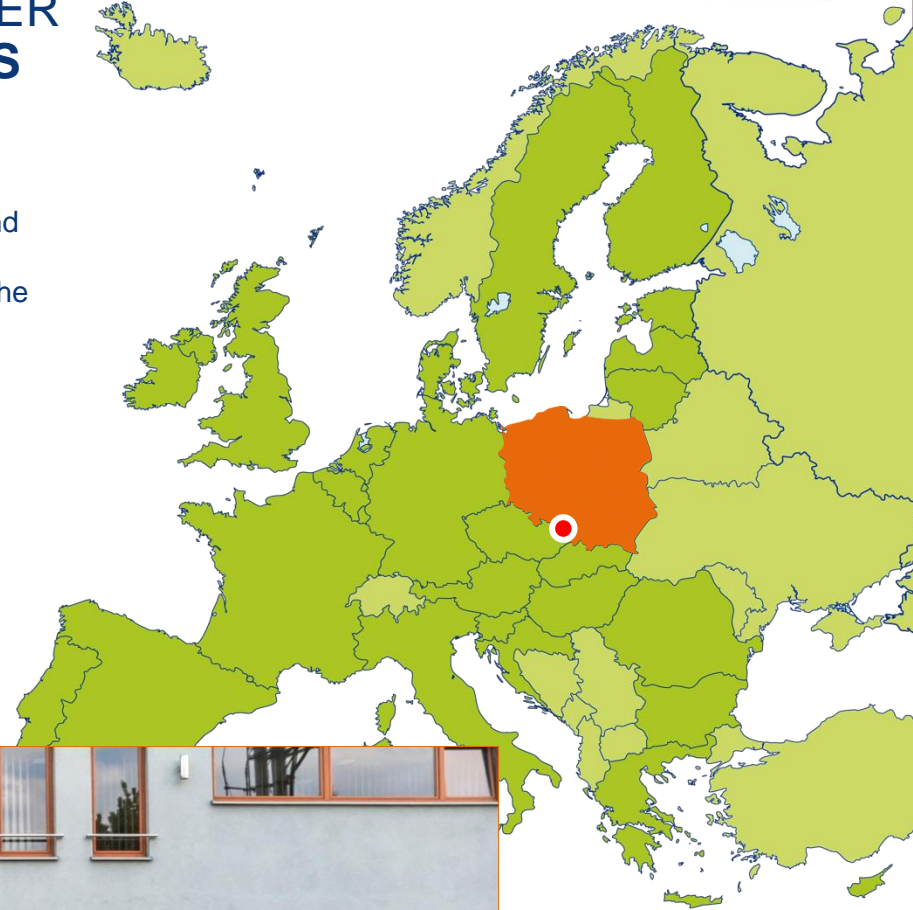
HV & UHV SYSTEMS ■ MANIPULATORS ■ CHAMBERS & MECHANISMS ◉ SAMPLE HOLDERS ■ INSTRUMENTS ■ ACCESSORIES ■ ELECTRONICS ■ SOFTWARE ■ SERVICES

PREVAC THE MAIN PRODUCER OF UHV SYSTEMS

PREVAC was founded in 1996 in Rogów, Upper Silesia, Poland.

PREVAC can be distinguished by its highly skilled, young, dynamic and ambitious personnel consisting of the best specialists who along with the long time experience in the field of vacuum technology constitute the greatest potential of the company.

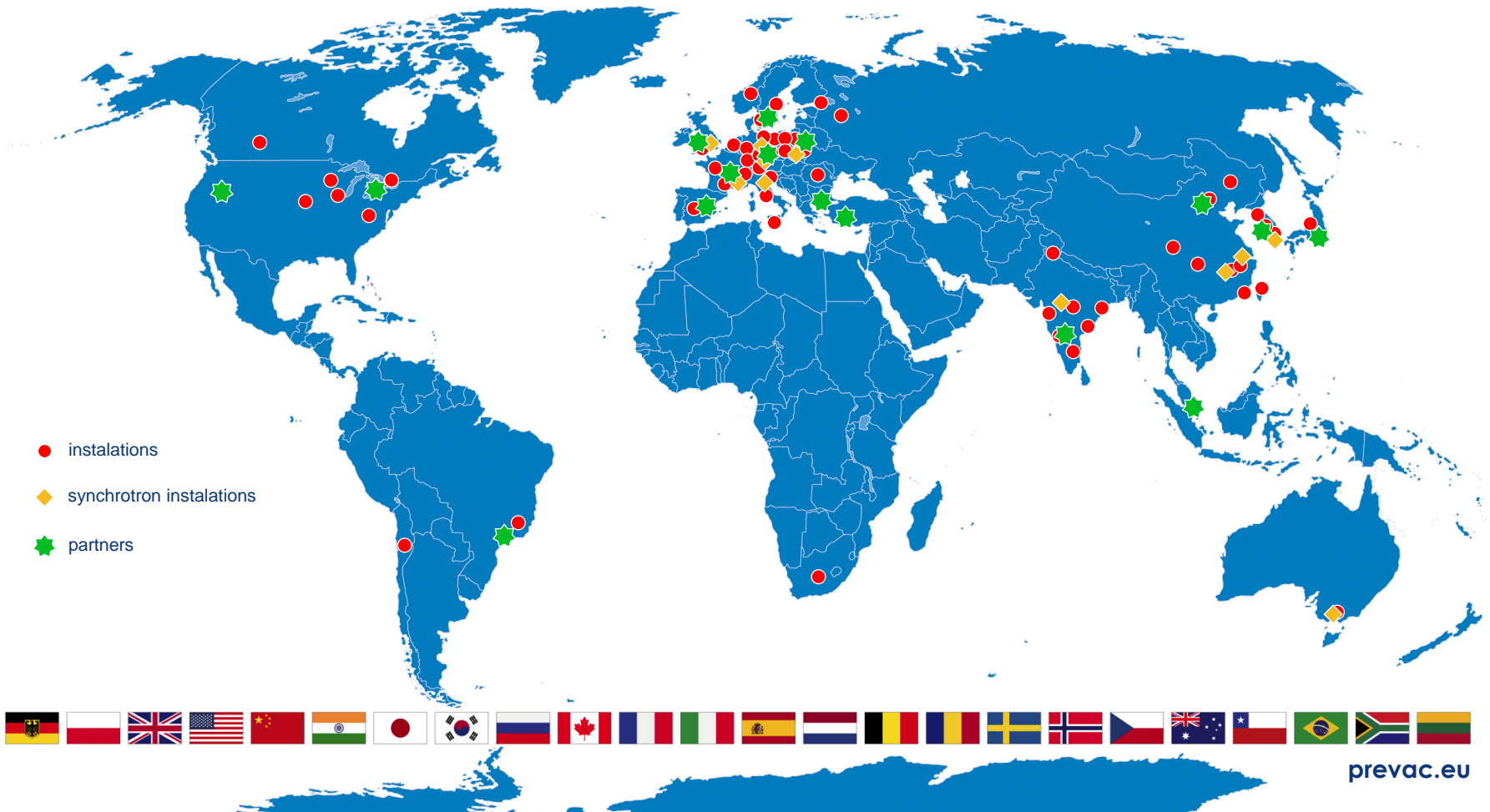
PREVAC employs a highly qualified staff of over 200 professionals – R&D constructors, manufacturing engineers, software developers etc., who works in various departments.



SALES

Since its foundation PREVAC has been an international known leading manufacturer of scientific-research equipment used for studies under conditions of high and ultra-high vacuum.

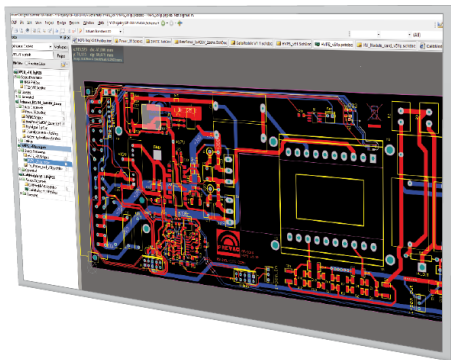
Products made by the company operate presently in such countries as Germany, France, Great Britain, Sweden, Norway, Italy, Spain, the Czech Republic, the USA, Canada, Japan, China, Russia, Australia, the Republic of South Africa, Poland etc.



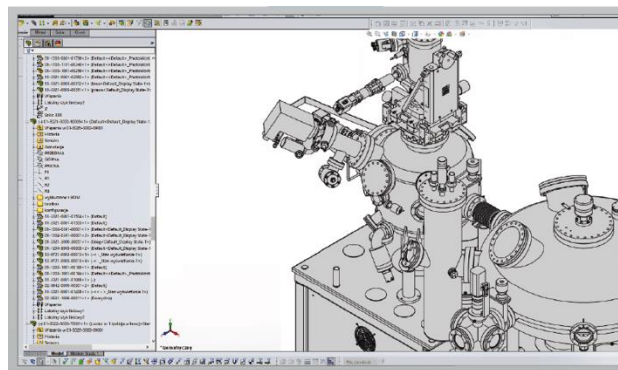
YOUR PROJECT



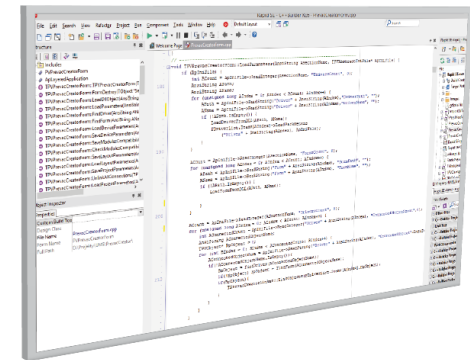
ELECTRONIC DEVICES



MECHANICAL DESIGNING

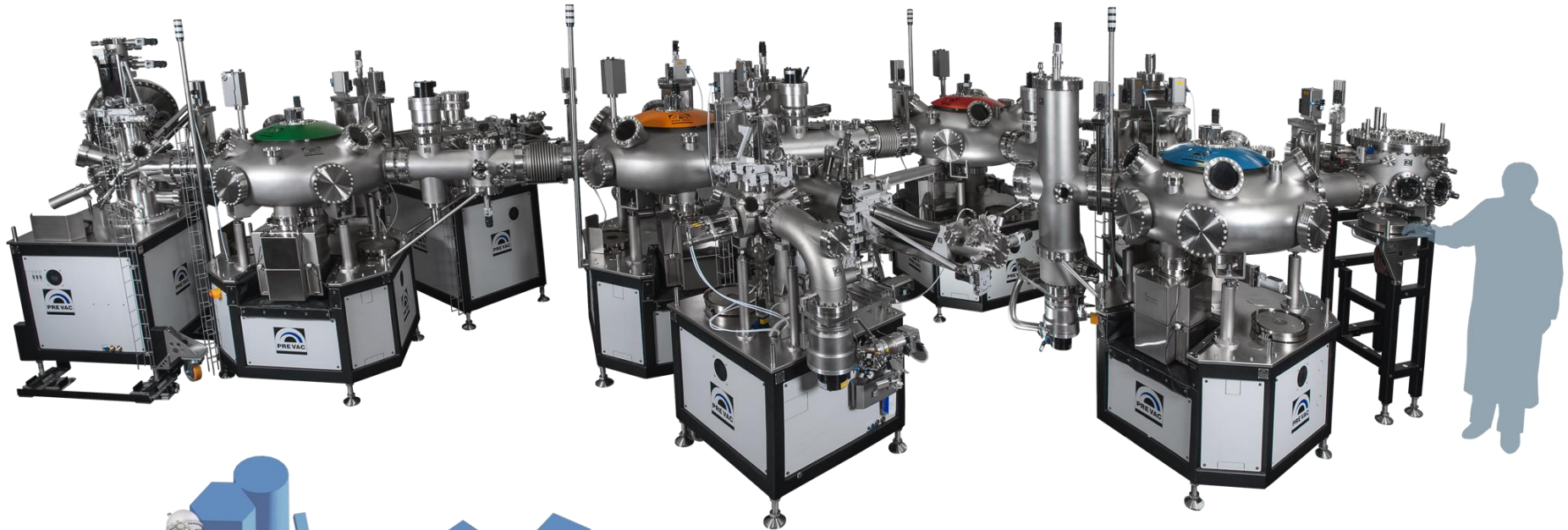


SOFTWARE APPLICATIONS



OUR PROJECT TEAM WORKS **FOR YOU**

MULTITECHNIQUE UHV



SISSY@EMIL
Helmholtz Zentrum Berlin



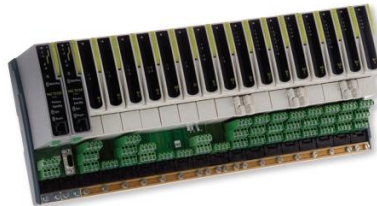
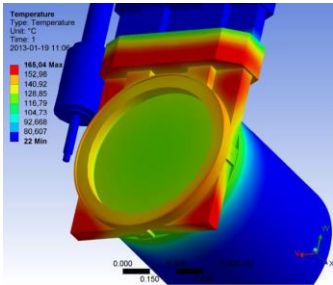
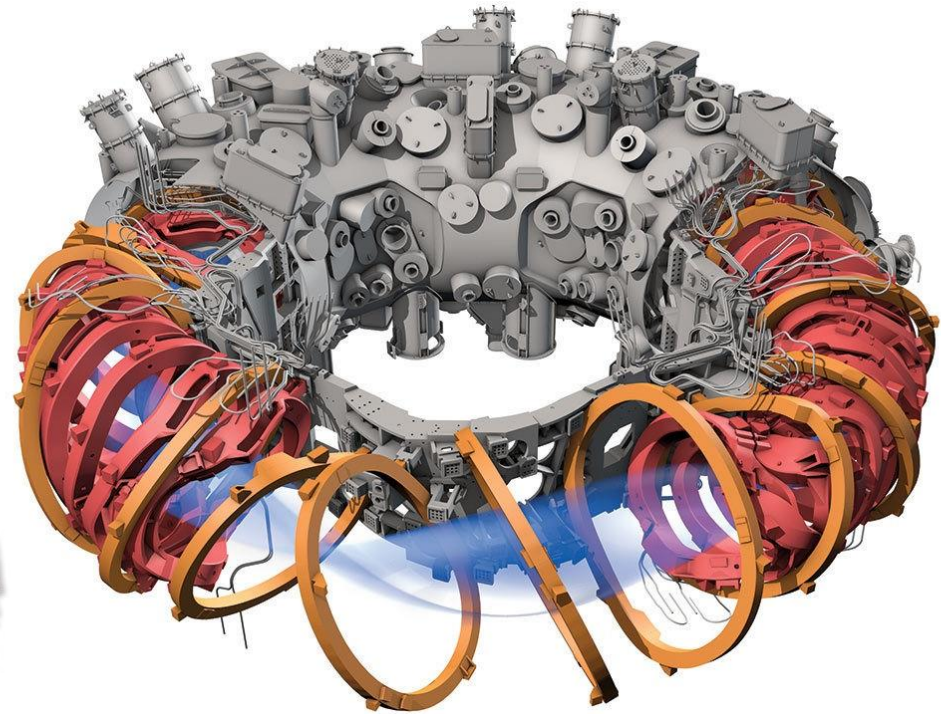
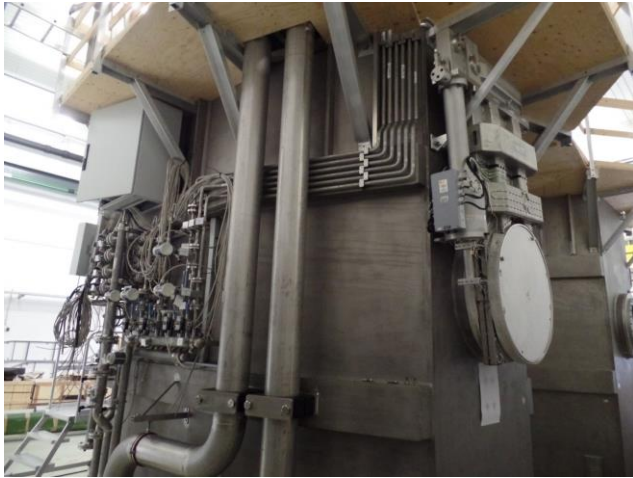
END STATIONS



BEAMLINE ENDSTATION

?? Somewhere ☺

LARGE SCALE PROJECTS



WENDELSTEIN 7-X

Helmholtz Zentrum Berlin

NEW COMPONENTS



EA15
R&D PREVAC

LAB CUSTOMERS NEEDS



- Focused on experiment
- Extend the experiment with smallest possible cost
- Access to experiments data

LAB CUSTOMERS REALITY



- Equipment from one or many suppliers
- Every supplier/integrator has own custom solutions
- Manage experiment is not easy
- Extend the experiment cost a lot

WHATS WRONG ?



THERE IS NO STANDARD !

STANDARD LAB CS

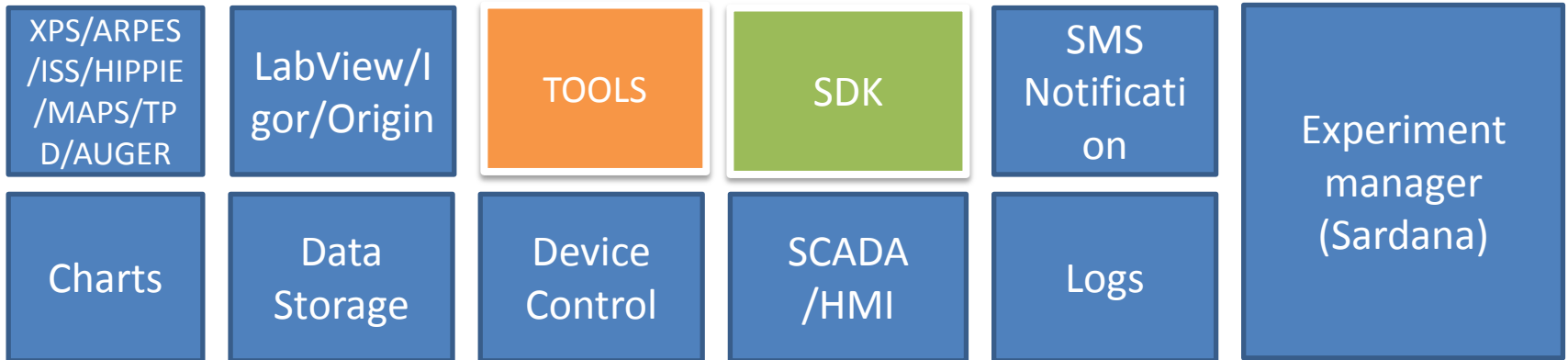


BRING TANGO TO LAB !

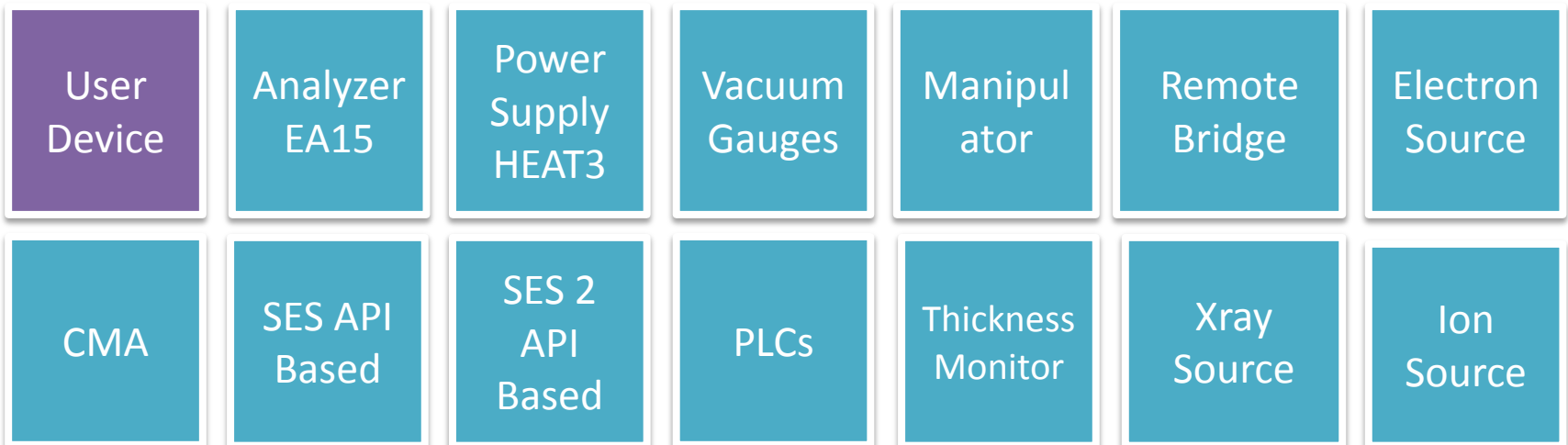
TANGO IN LAB = RAPID FX

- 3 Tango base installations in 2016
- >20 installations in 2017
- and more...

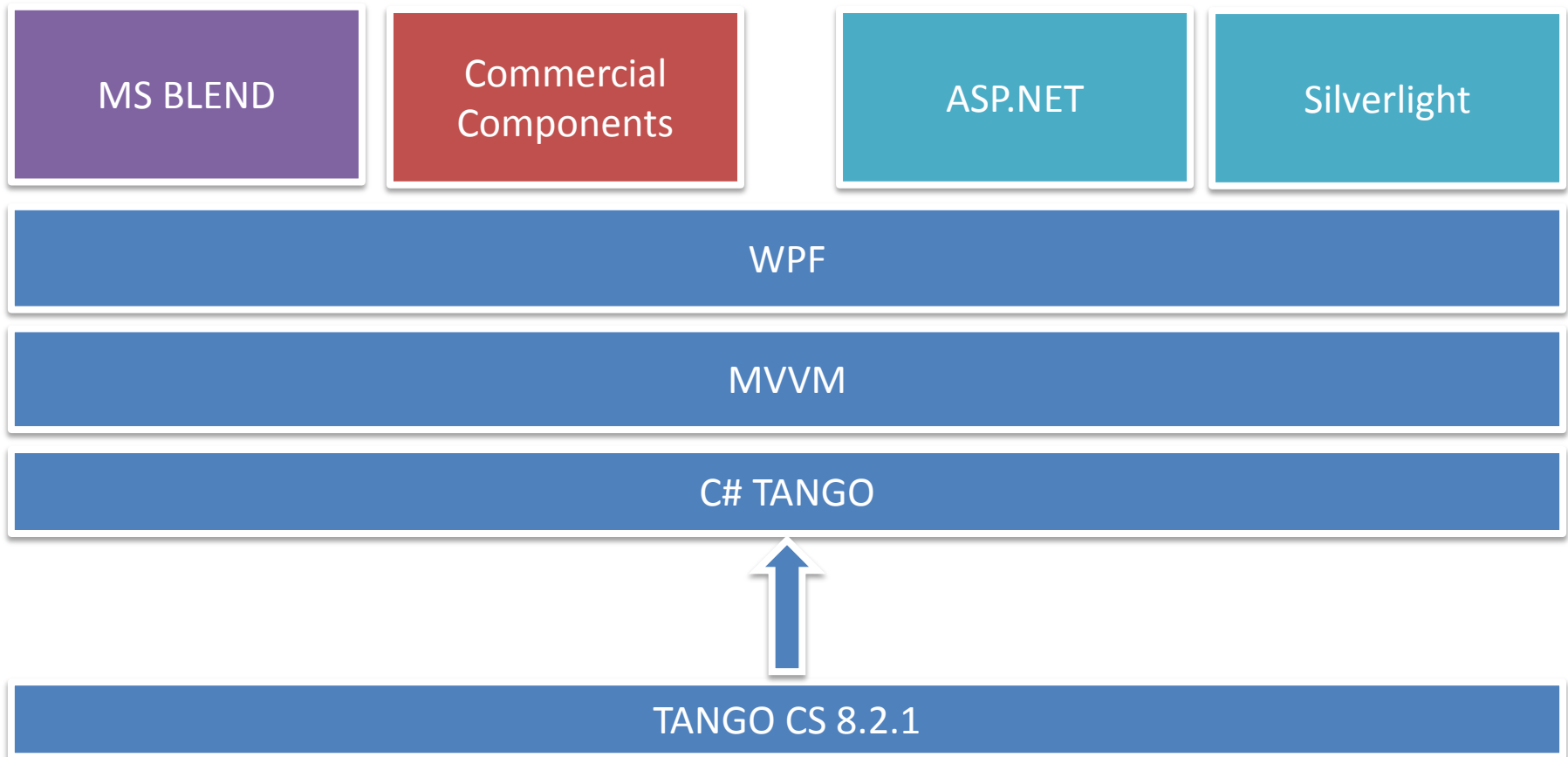
RAPID FX PLATFORM



TANGO Framework – **Python** – C++/boost



RAPID FX PLATFORM



RAPID FX SYSTEM VIEW

temperature

237.1 [°C]

manipulator

X=0.0 [mm]
Y=0.0 [mm]
Z=10.0 [mm]

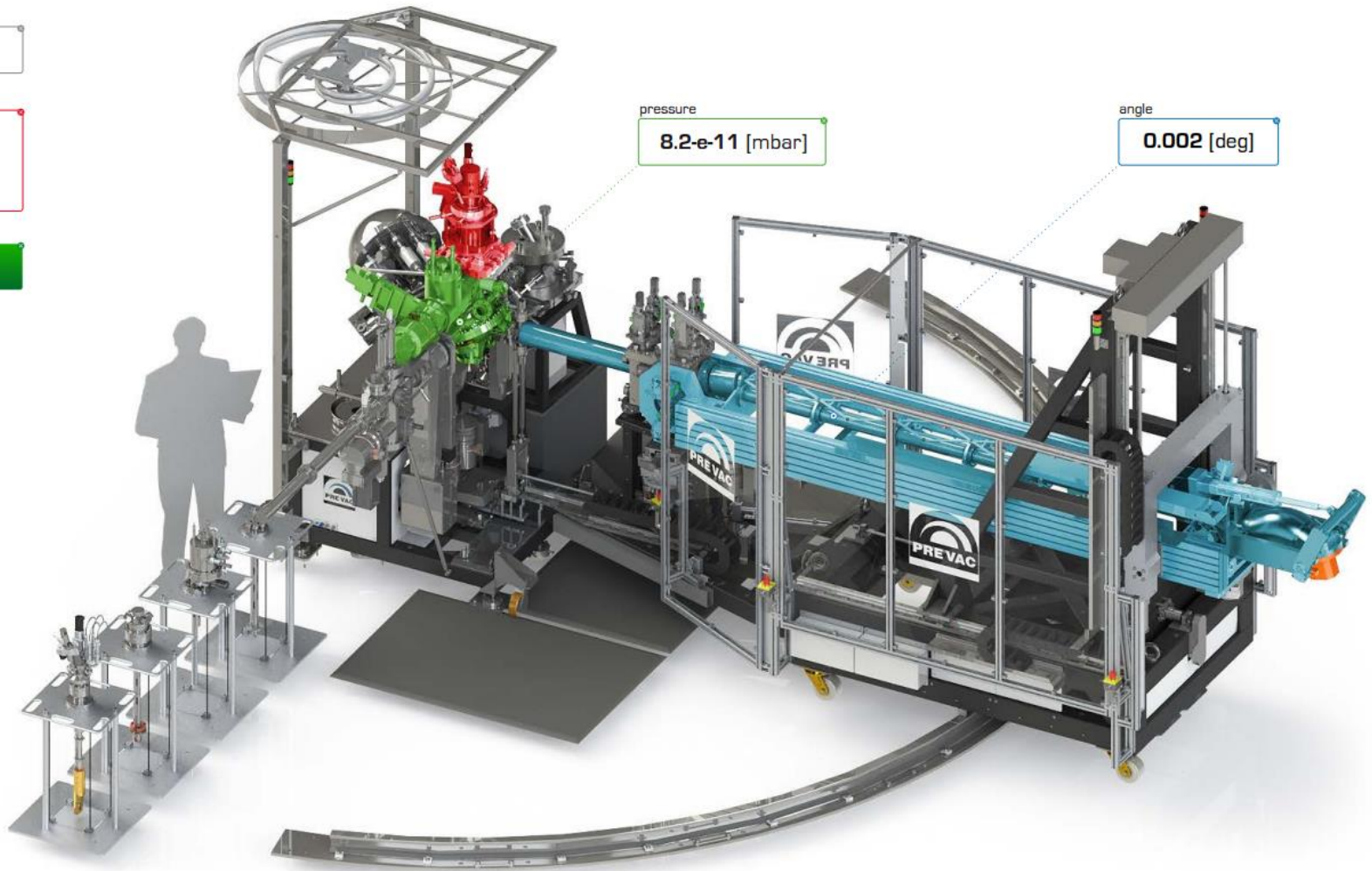
System ready

pressure

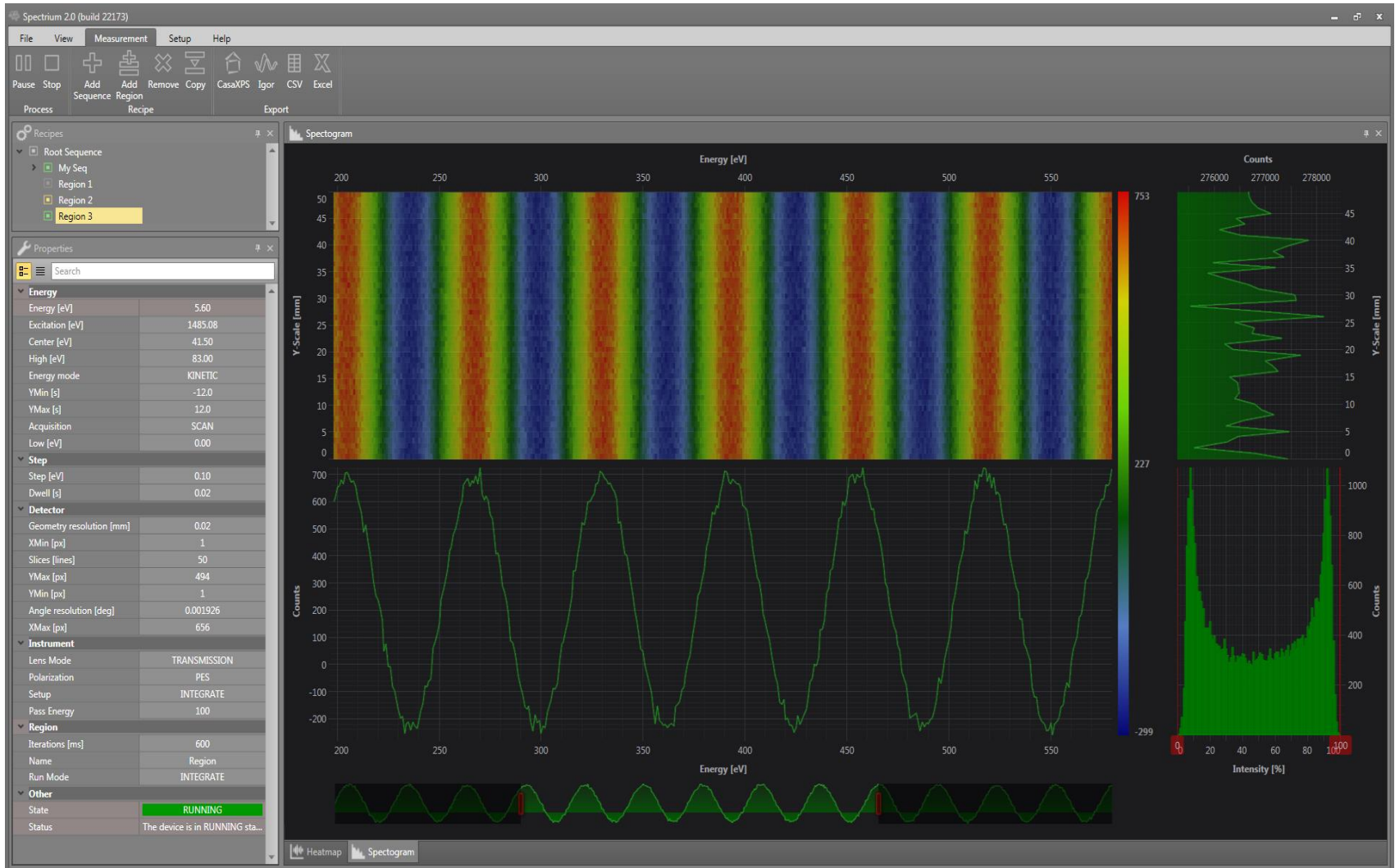
8.2-e-11 [mbar]

angle

0.002 [deg]



RAPID FX SPECTRIUM





RAPID FX XMANAGER

Spectrum Experiment Manager

Home Macro

Run Stop Abort Pause Resume Add macro Remove macro Save Project Project Tools

Experiment Manager

Tools

- BASIC MACROS
 - Sequence
 - Region
 - While
 - IF
 - IF ELSE condition
 - Manipulator
- USER MACROS
 - AnalyzerRegion
 - AnalyzerTestMacro
- SARDANA MACROS

Main

Program

- Sequence
 - Region
 - Region
- Sequence
 - Region
 - While
 - Region
- Sequence
 - Manipulator
- AnalyzerRegion

Script Editor

```
from sardana.macroserver.macro import Macro, Type
import PyTango
import time
from PyTango import DevState

class AnalyzerRegion(Macro):
    param_def = [ [ "dwell", Type.Float, 0.1, "region dwell" ],
                  [ "step", Type.Float, 0.1, "region step" ],
                  [ "high", Type.Float, 10.0, "region high" ],
                  [ "low", Type.Float, 0.0, "region low" ],
                  [ "center", Type.Float, 5.0, "region center" ],
                  [ "iteration", Type.Integer, 2, "region iteration" ] ]
    result_def = [ [ "Result", Type.Boolean, True, "macro status" ] ]
    """analyzer/test/1 region block"""
    def run(self, dwell, step, high, low, center, iteration):
        device=PyTango.DeviceProxy("analyzer/test/1")
        status = device.state()
        if status != DevState.RUNNING:
            device.write_attribute("region_dwell",dwell)
            device.write_attribute("region_step",step)
            device.write_attribute("region_high",high)
            device.write_attribute("region_low",low)
            device.write_attribute("region_center",center)
            device.write_attribute("region_iterations", iteration)
            device.command_inout("Start")
            status = device.state()
            while status == DevState.RUNNING:
                time.sleep(1)
                status = device.state()
            self.output(status)
            if status != DevState.STANDBY:
                return False
            return True
```

Settings

Search

| | |
|-----------|------|
| dwell | 0.1 |
| step | 0.1 |
| high | 10.0 |
| low | 0.0 |
| center | 5.0 |
| iteration | 2 |

Log Viewer



THANK YOU FOR YOUR ATTENTION