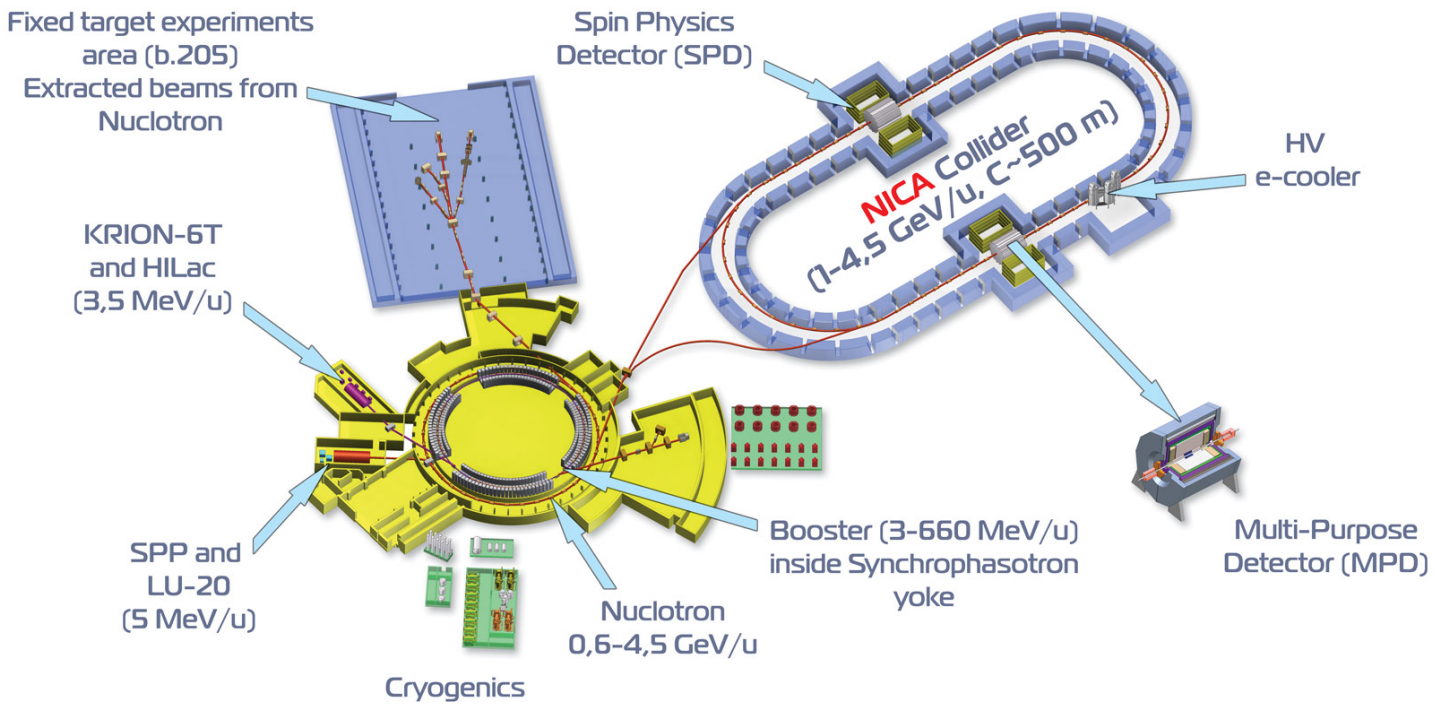
Status of the Nuclotron and NICA control system development

V.Andreev, E.Gorbachev*, A.Kirichenko, D. Monakhov, S. Romanov, G.Sedykh, T. Rukoyatkina, V.Volkov

LHEP, JINR, Dubna

* gorbe@sunse.jinr.ru

NICA accelerator complex



- Equipment distributed on large area.
- Strict synchronization
- Beam diagnostics

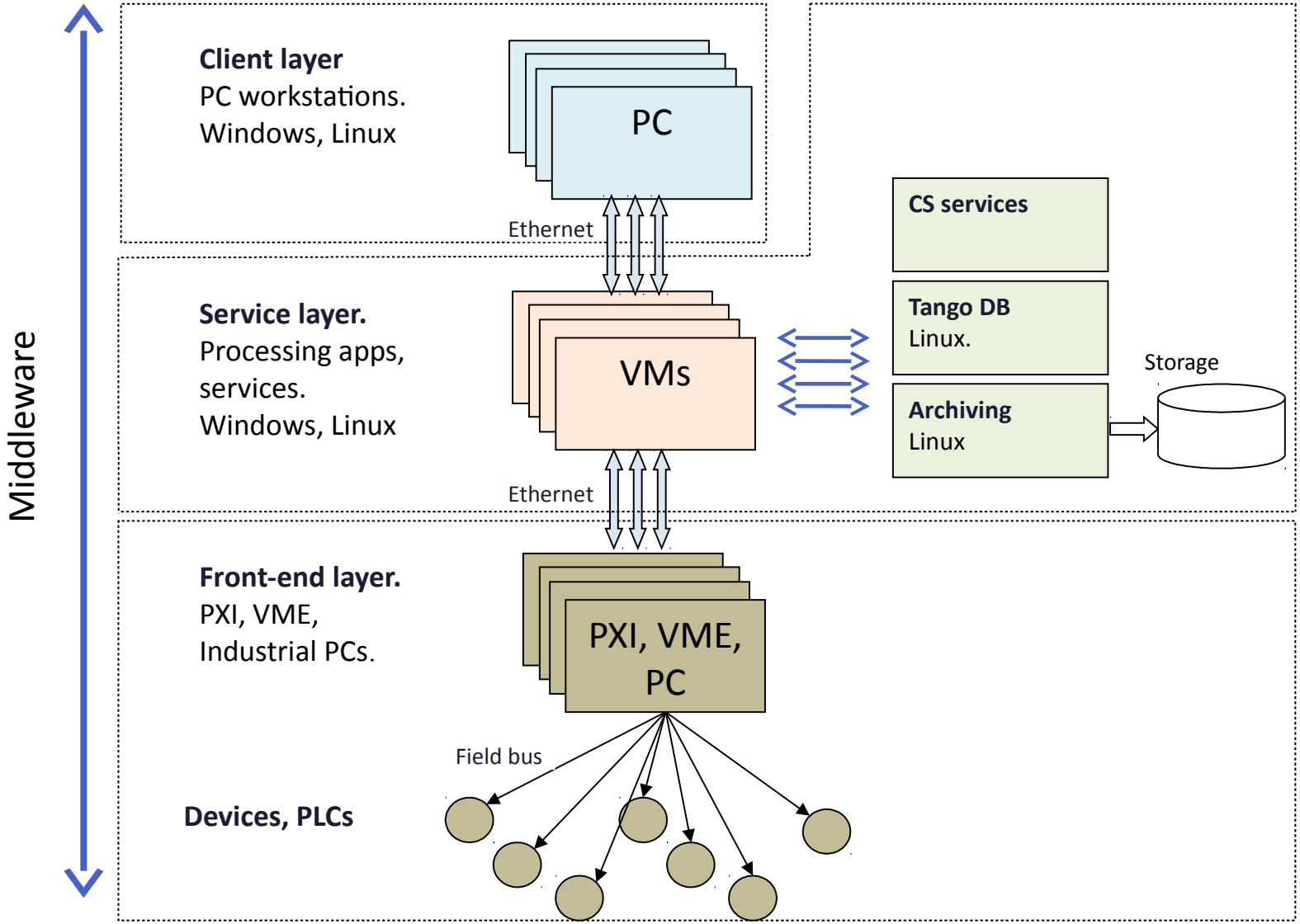
Control System development goals

- Centralized control of distributed equipment
- Reliable operation and fast recovery
- Provide safe operation, access control
- Easy support, modification and scalability
- Easy and fast development and deployment
- Integration of third-party and existing systems

Plan

- Control system layout
  - Middleware
  - Front-end layer
  - Client layer
  - Service layer
- Control System services
- Nuclotron and NICA development
- Conclusion and future plans.

Control System layout



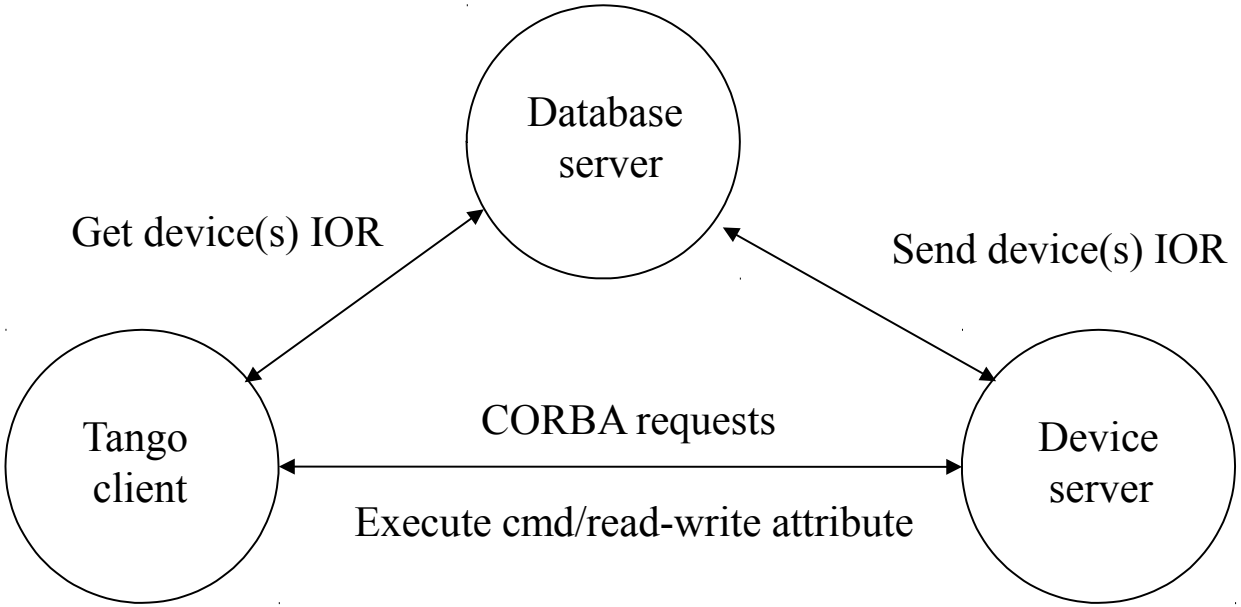
Middleware

Based on TANGO:

- CORBA based distributed object-oriented control system framework
- Multiplatform
- Hides network location and technical details
- Provides unified interface for hardware equipment or service:
  - Commands
  - Attributes
  - Properties
- Has services and tools for control implementation.

TANGO architecture

- Fundamental unit: device
- Unique name for each device: domain/family/member (ex: extraction/daqmxdi/1)
- Uses database for storing configuration



TANGO programming

- Tools to generate code in C++, Java, python.
- High-level client API to implement synchronous/asynchronous interface, events

More details about TANGO concept and its usage in other reports

Frontend layer

Few possible variants to implement DAQ or control:

- Custom hardware
- Commercial off-the-shelf (COTS)

Primary requirements:

- Ease of development
- Quality, reliability and performance
- Maintenance of equipment during the accelerator complex life-time

Frontend layer




National Instruments PXI– high performance Eurocard packaging modular platform for measurements and automation systems:

- PCI express and cPCI busses with additional synchronization buses.
- Excellent performance: PXIe-1085 – throughput up to 4GB/s per slot and up to 12GB/s per system.
- Wide range of available modules (1500+) – controllers, acquisition boards, analog and digital I/O, signal generators, digital multimeters, counters and timers, high speed digitizers, industrial interfaces, RF and others.
- Windows and Linux programming – LabVIEW, C++ and C# libraries.
- Excellent support.

Frontend layer

- Digital signal processing (DSP) modules FlexRIO –PXI and PXIe modules with Xilinx FPGA and custom I/O modules. Supports peer-to-peer streaming up to 3GB/s.
- Compact RIO - High performance embedded systems and intellectual controllers based on Xilinx FPGA + ARM with standard or custom I/O module.

Frontend layer

Tango drivers with runtime configurable properties for NI equipment were developed:

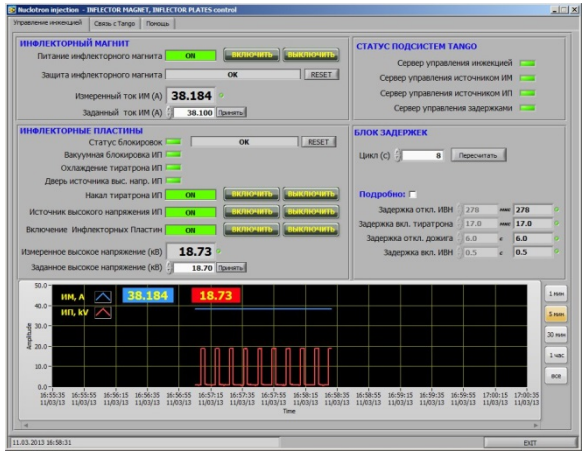
- Digitizers and scopes
- Analog and digital I/O
- Timers and counters
- Digital multimeters
- RTD input modules
- TANGO interface for FlexRIO and CompactRIO.

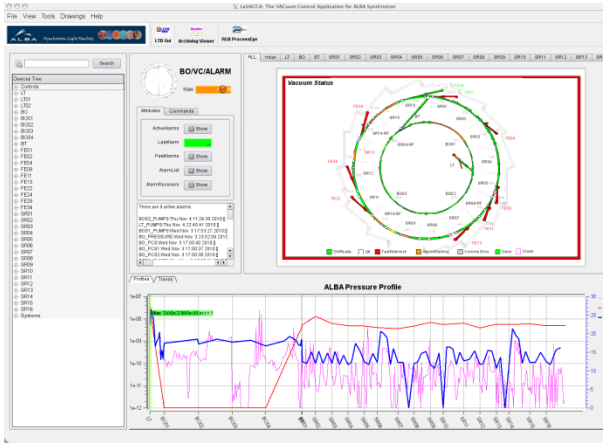
Combination of tango devices can be used to quickly deploy a solutuion based on almost any NI acquisition equipment.

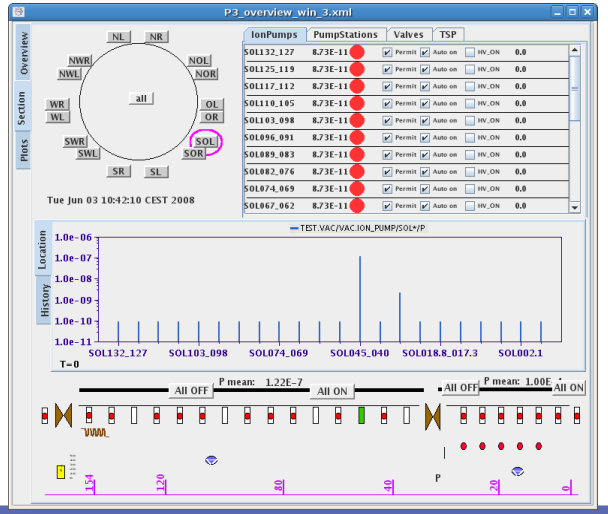
Client layer: Desktop apps

The **client layer** is responsible for presentation of the accelerator complex state to the operator, visualize measurements and provide user interface to fulfill some control tasks. Operator should be able to access the control system of the complex in whole and down to the lower level subsystems with access control.
There are standard clients, custom client software can be built with:

- Java and TangoATK;
- C++ and Qtango;
- Python and Taurus;
- LabView and LV bindings

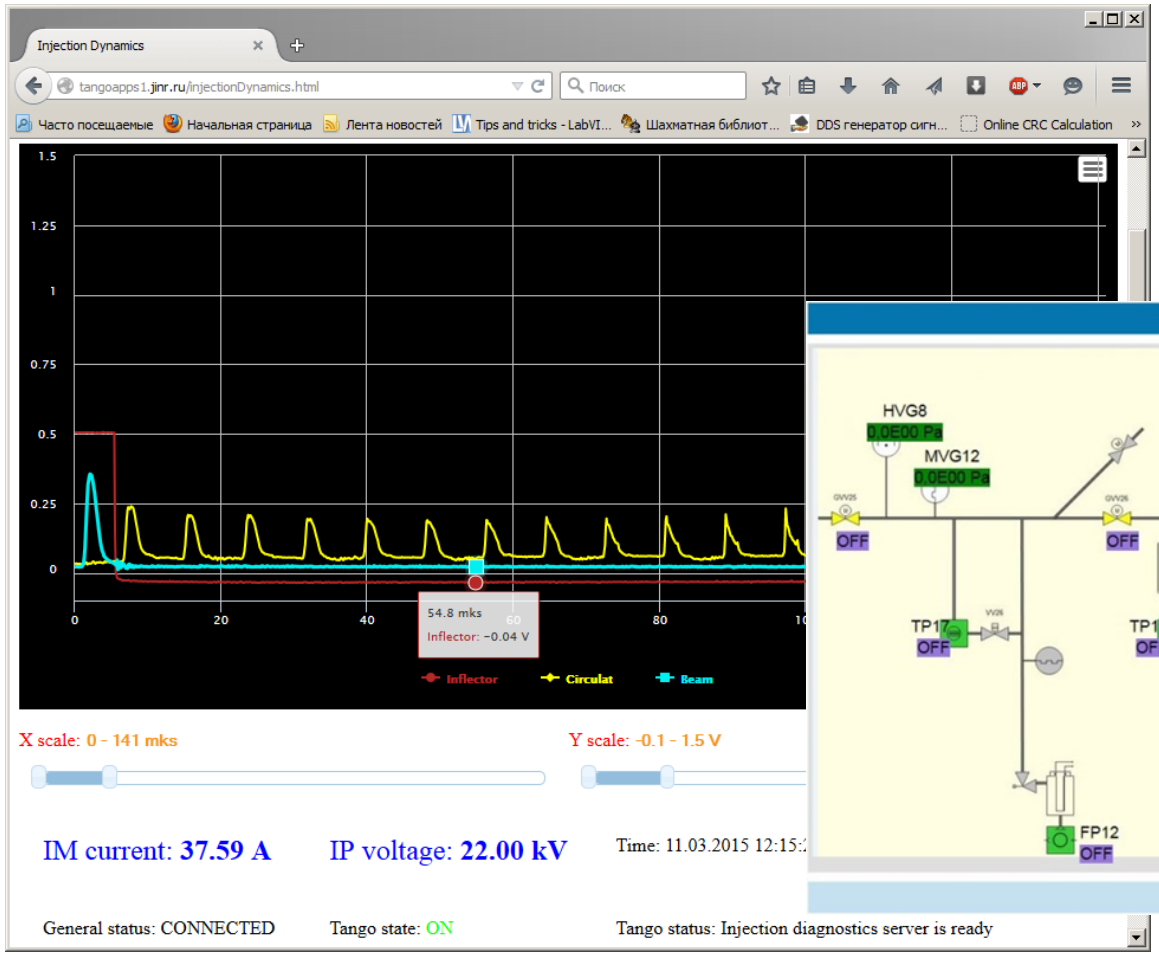


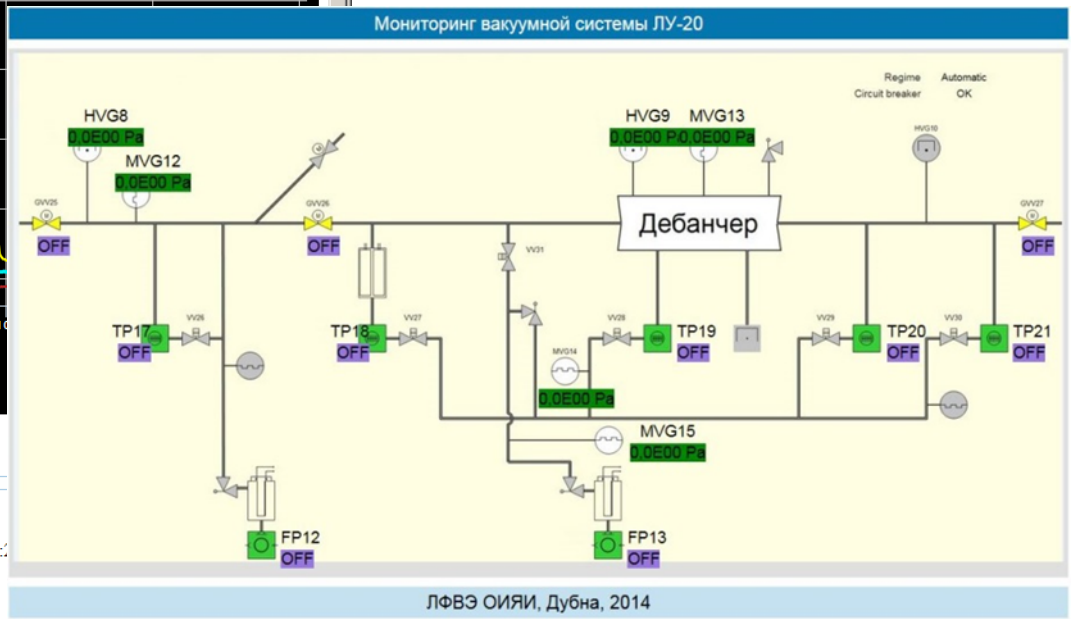




Client layer: Web apps

- Java and JSF, PrimeFaces, TangORB
- JavaScript and our custom REST and Websockets servers



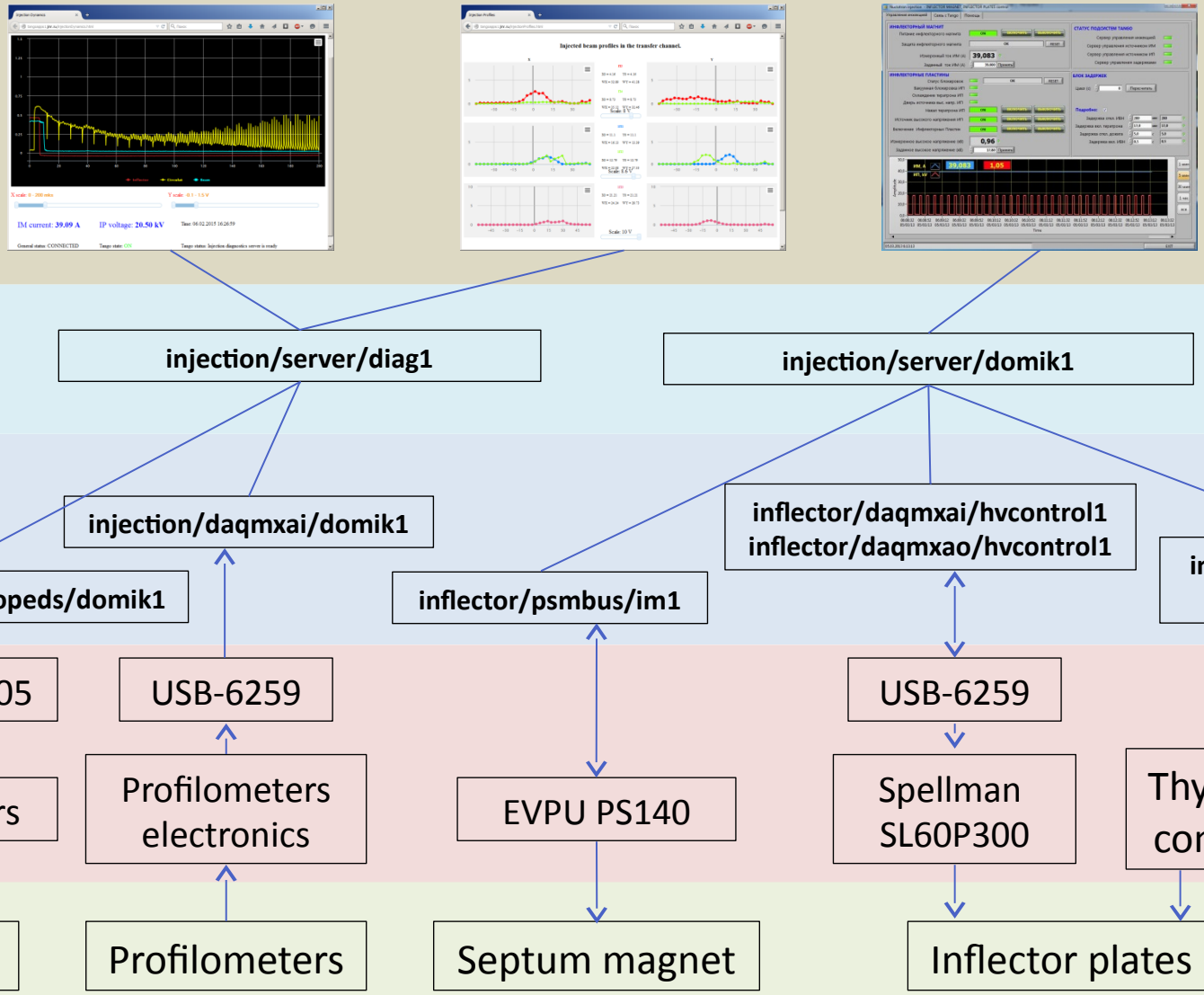


Service layer

Contain high-level TANGO device servers, representing whole subsystem:

- Collects and process data from low-level devices
- Performs control loop
- Provides standard interface to client applications: commands, attributes

Example: Nuclotron Injection



Service layer: Infrastructure

Applications running on service layer are hardware independent and can be virtualized (run on VM):

- Easier management – fast VM deployment from templates, cloning, backups.
- Better tasks isolation – assign VM per task or group of tasks.
- Resources tuning – can assign appropriate CPU cores, disk space, RAM size, disk I/O.
- Resource utilization – efficient usage of host's RAM and CPU.
- High Availability - restore VM from failed server.

Service layer: Virtualization

Proxmox VE - complete open source virtualization management solution for servers. It manages virtual machines, storage, virtualized networks, HA clustering and supports both:

- **Kernel-based Virtual Machine (KVM)**

Open source hypervisor KVM is a full virtualization solution for Linux on x86 hardware containing virtualization extensions (Intel VT or AMD-V). Can run unmodified Linux or Windows images.

- **Container-based virtualization**

OpenVZ is container-based virtualization for Linux. OpenVZ creates multiple secure, isolated Linux containers (otherwise known as VPSs) with no impact on performance.

Proxmox VE features:

- Web GUI
- Role-based administration
- Integrated backup tool
- Flexible Storage

Service layer: Shared storage

- Crucial component of virtualization to provide high availability of VMs
- Need good performance , both transfer rate and IOPS to manage read/write operations of number of VM images
- Have to be redundant to provide data consistency in case of hardware or network failure
- Need to be scalable – should be able to add more storage space without loosing performance

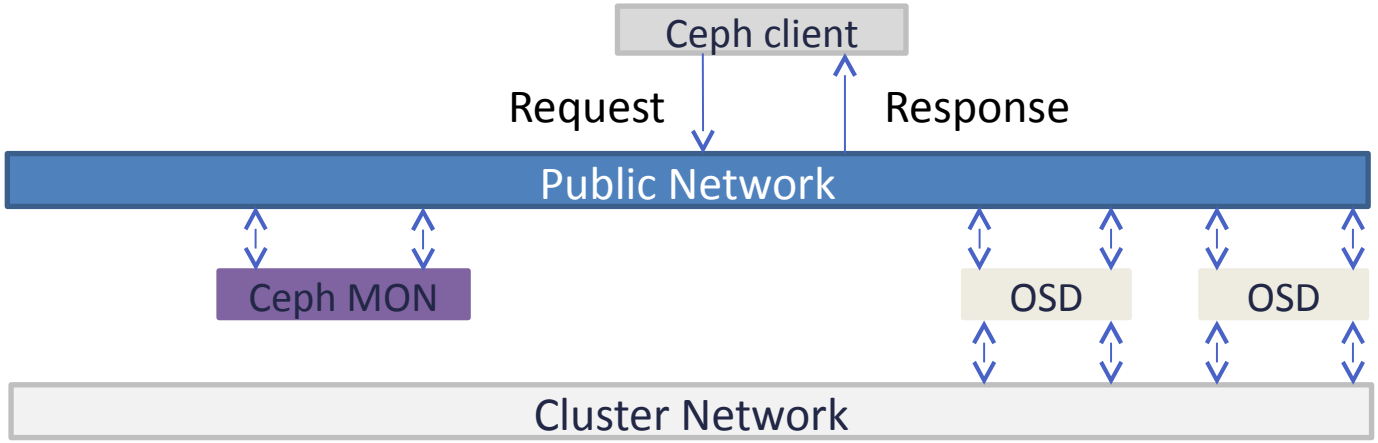
Service layer: CEPH storage

is a distributed object store and file system. Ceph is highly reliable, easy to manage and open-source. It provides redundancy, excellent performance and scalability.

- Redundancy choices: replication (use space, save CPU) or erasure coding (use CPU, save space).
- Ceph's RADOS Block Device (RBD) provides access to block device images that are striped and replicated across the entire storage cluster. Linux kernel client and QEMU/KVM drivers.
- Thin provisioning – unused space reclaimed.

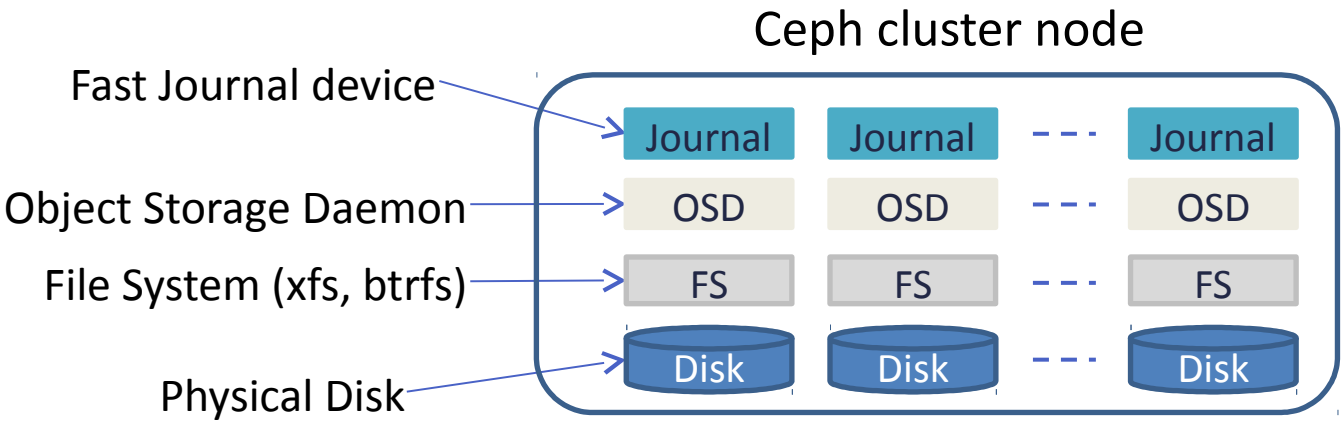
Service layer: CEPH storage cluster

Ceph storage cluster consists of large number of nodes which communicate with each other to distribute and replicate data dynamically.
Self healing, self managed, distributed.
Using CRUSH algorithm for object placement.



OSD heartbeat, data replication, cluster rebalancing and recovery traffic

Service layer: CEPH node



- SSD journals accelerate bursts and random write IO.
- For sustained writes that overflow the journal, performance degrades to HDD levels.
- Caching controller can increase random write IO significantly.

Service layer: CEPH architecture

3 Ceph nodes at the moment, each with:

- Dual socket E5-2600 Intel Xeon
- 32Gb RAM
- 5 x 600G SAS2 15k RPM drives
- 1 enterprise class SSD Intel S3710
- LSI 2128 RAID controller with 512Mb cache
- 2x1Gb Ethernet card for public network
- 2x10Gb Ethernet card for CEPH networks

Networking equipment:

- 1Gb network switch for public network
- 1Gb -> 10Gb Cisco Nexus switch for CEPH network

Service Layer: CEPH performance

Test setup	Block size	Sequential Read	Sequential Write	Random Read	Random Write
Linux KVM on CEPH 1GbE, 3 replicas	4k	166.4 MB/s 41600 IOPS	47 MB/s 11765 IOPS	79.1 MB/s 19764 IOPS	11.0 MB/s 2750 IOPS
	128k	165.9 MB/s 1296 IOPS	52.6MB/s 411 IOPS	258.2 MB/s 2017 IOPS	23.1 MB/s 180 IOPS
	4M	166.3 MB/s 40 IOPS	53.1 MB/s 12 IOPS	164.4 Mb/s 40 IOPS	48MB/s 11 IOPS
Single 15k RPM SAS2 disk (with RAID controller write-back cache)	4k	22.9 MB/s 5713 IOPS	143.6 MB/s 35897 IOPS	2.8 MB/s 702 IOPS	3.9MB/s 974
	128k	111.9 MB/s 873 IOPS	143.6 MB/s 1121	31.6 MB/s 246 IOPS	29.2 MB/s 225
	4M	125.1 MB/s 30 IOPS	145.4 MB/s 35	110.9 MB/s 27 IOPS	114.5 MB/s 27
4x15k RPM SAS2 disks in RAID5 (with RAID controller write-back cache)	4k	213.8 MB/s 53447 IOPS	262.1 MB/s 65529 IOPS	11.4MB/s 2800 IOPS	27.9 MB/s 6980 IOPS
	128k	235.1 MB/s 1837 IOPS	241.82 MB/s 1889 IOPS	104 MB/s 814 IOPS	130.4 MB/s 1018 IOPS
	4M	264.3 MB/s 64 IOPS	237.5 MB/s 57 IOPS	236.4 MB/s 57 IOPS	188.9 MB/s 46 IOPS

Service layer: High availability

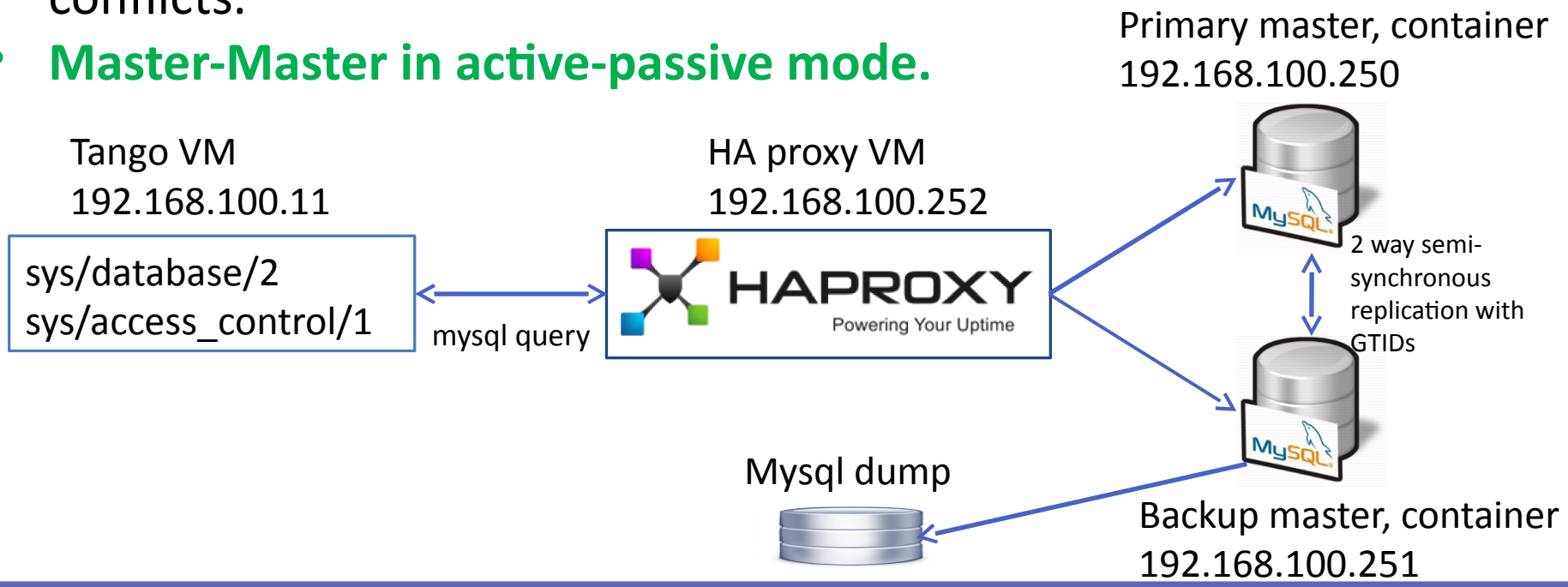
We need to provide high availability of Tango services and fast failover in case of hardware problems.
Hardware measures (UPS devices controlled by Network UPS tools, reliable enterprise class hard drives) + data redundancy (RAID, replication) + backups (VMs and databases):

- Virtual machines are running on 3-node Proxmox VE cluster with VM images stored on CEPH storage (replication=3). Any VM can be started on any node.
- Tango database: vital for all other Control System Tango devices, running in Linux container on local storage for performance reasons.

Service layer: TANGO DB

High Availability for Tango database can be achieved:

- Mysql/Galera cluster – reliable but complicated to manage.
- Master-slave mysql replication. Complicated to sync changes back to master.
- Master-master replication in active-active mode. Can cause conflicts.
- **Master-Master in active-passive mode.**



Service layer: Scalability

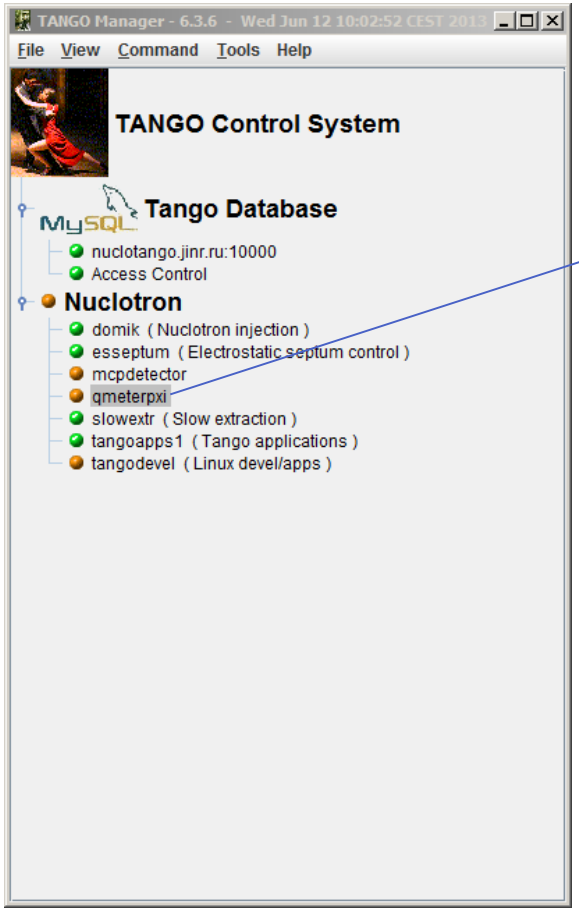
The scalability achieved by adding nodes to cluster. CEPH retains most characteristics at scale or even improves. Adding more nodes to run more VMs:

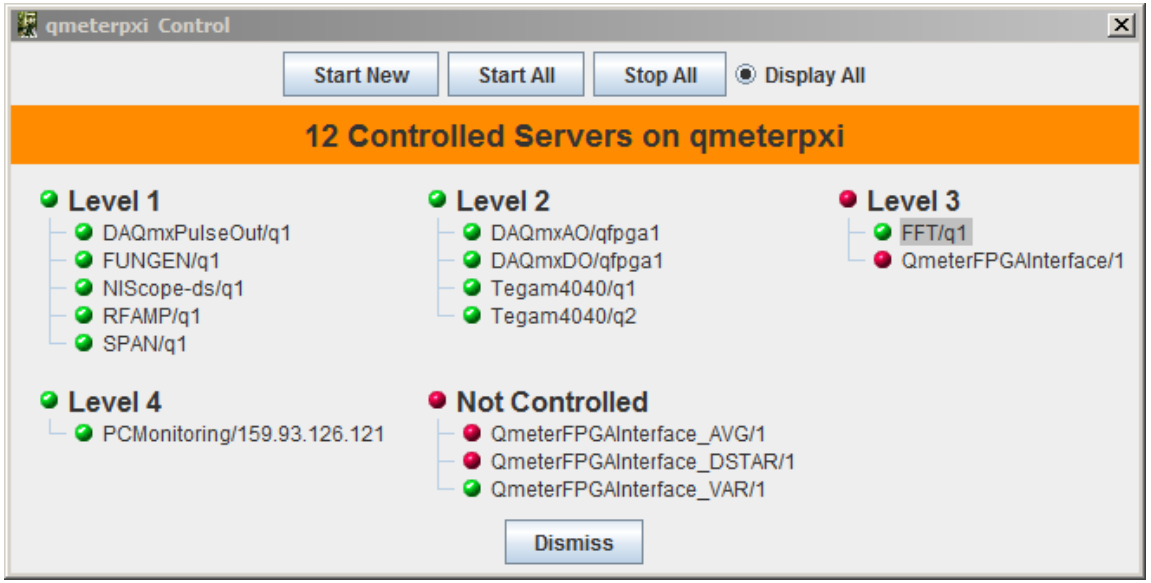
- Capacity increases
- Throughput increases
- IOPS increase
- CPU cores and RAM increase

Temporary impact during re-balancing.

Service layer: Administration

Astor/Starter –tools for remote control and monitoring of TANGO applications

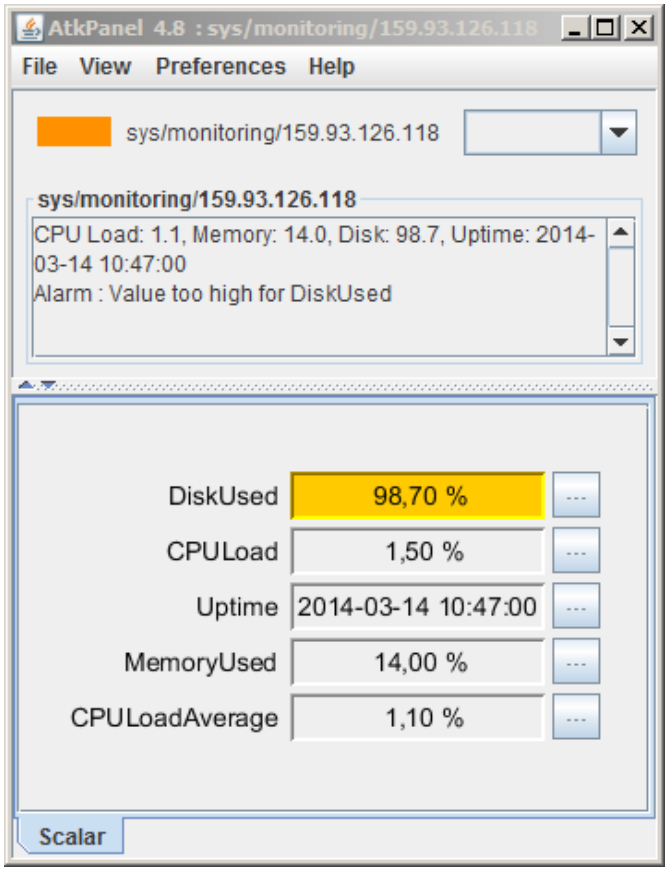


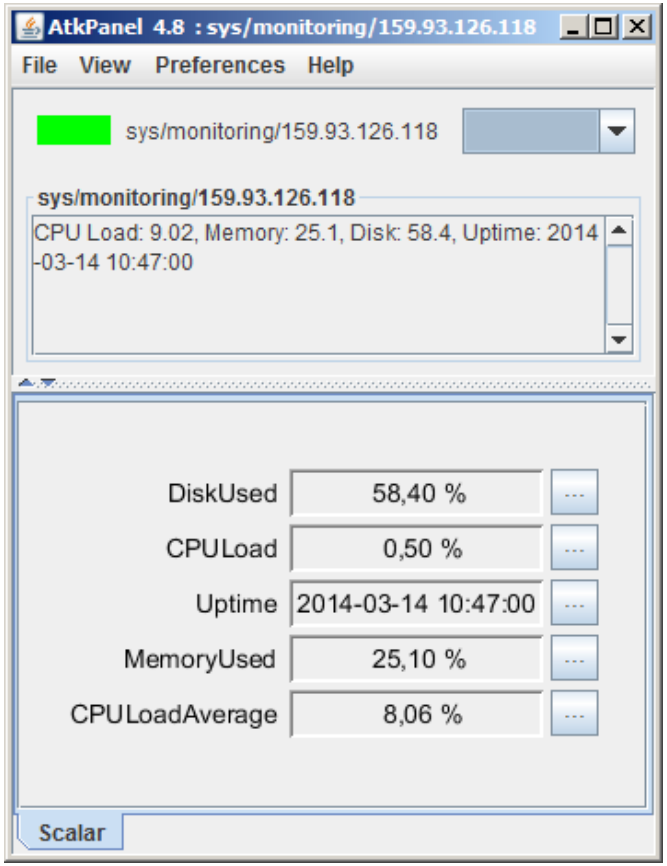


Hosts	Servers
• All controlled servers are running. • Starter is starting server(s). • At least, one controlled server is stopped. • Starter is not running on host.	• Server is running • Server is running but not alive (Starting ?) • Server is not running.

Service layer: Monitoring TANGO

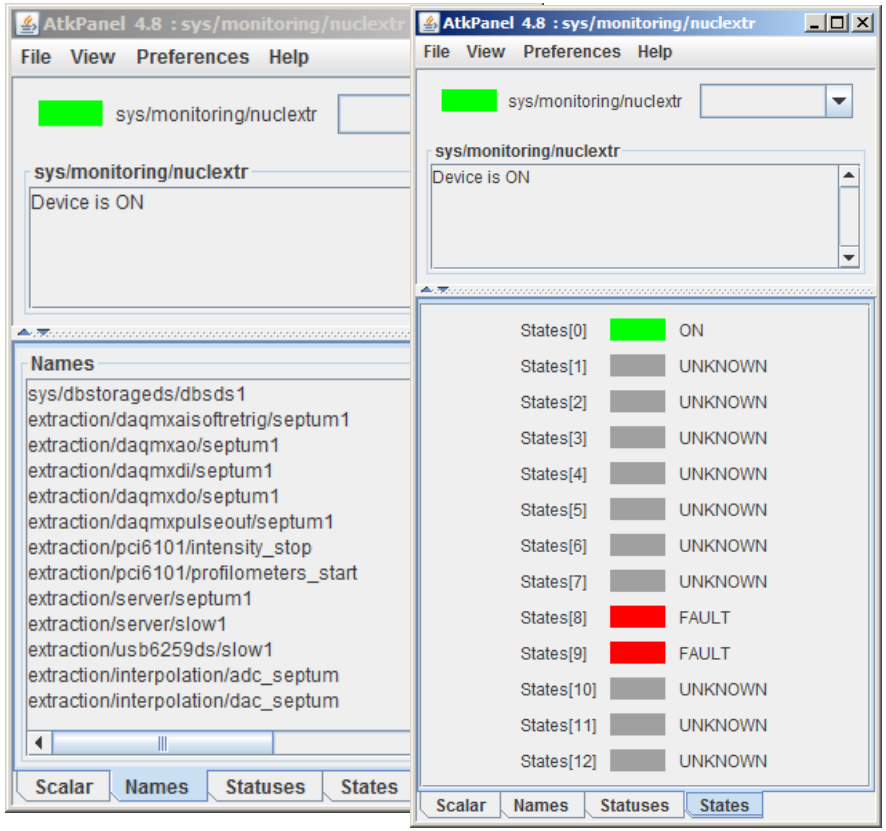
Special TANGO device is running on every front-end computer. It presents information about computer metrics via TANGO attributes. Linux+Windows





Service layer: monitoring TANGO

Another TANGO device monitoring states of every TANGO device of certain subsystem. The information about members retrieved from NicaControls db. Alerts and notifications are sent via e-mail to tango admins.



От tango@tangodevel.jinr.ru
Тема **Monitoring alert - Nuclotron extraction, device: extraction/usb625**
Кому "E. Gorbachev"

extraction/usb6259ds/slow1 state was changed to UNKNOWN

От tango@tangodevel.jinr.ru
Тема **Monitoring alert - Nuclotron extraction**
Кому "E. Gorbachev"

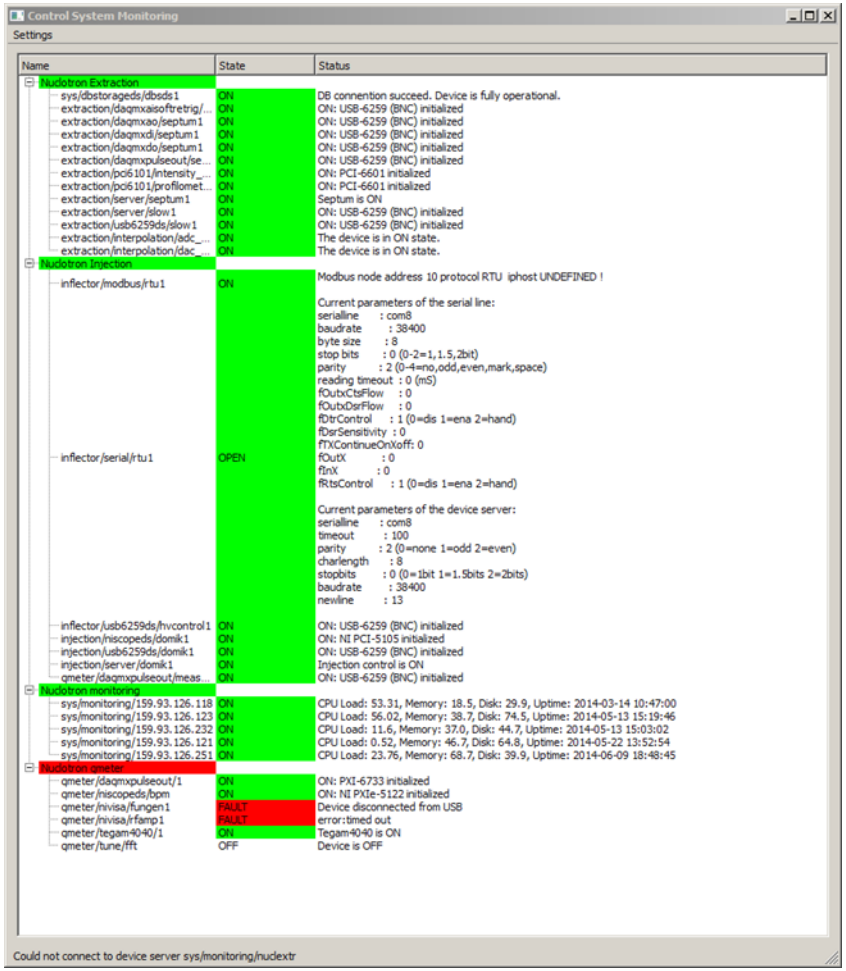
Subsystem: Nuclotron extraction
Name: sys/dbstorageds/dbsds1, State: FAULT, Status: MySQL server has gone away

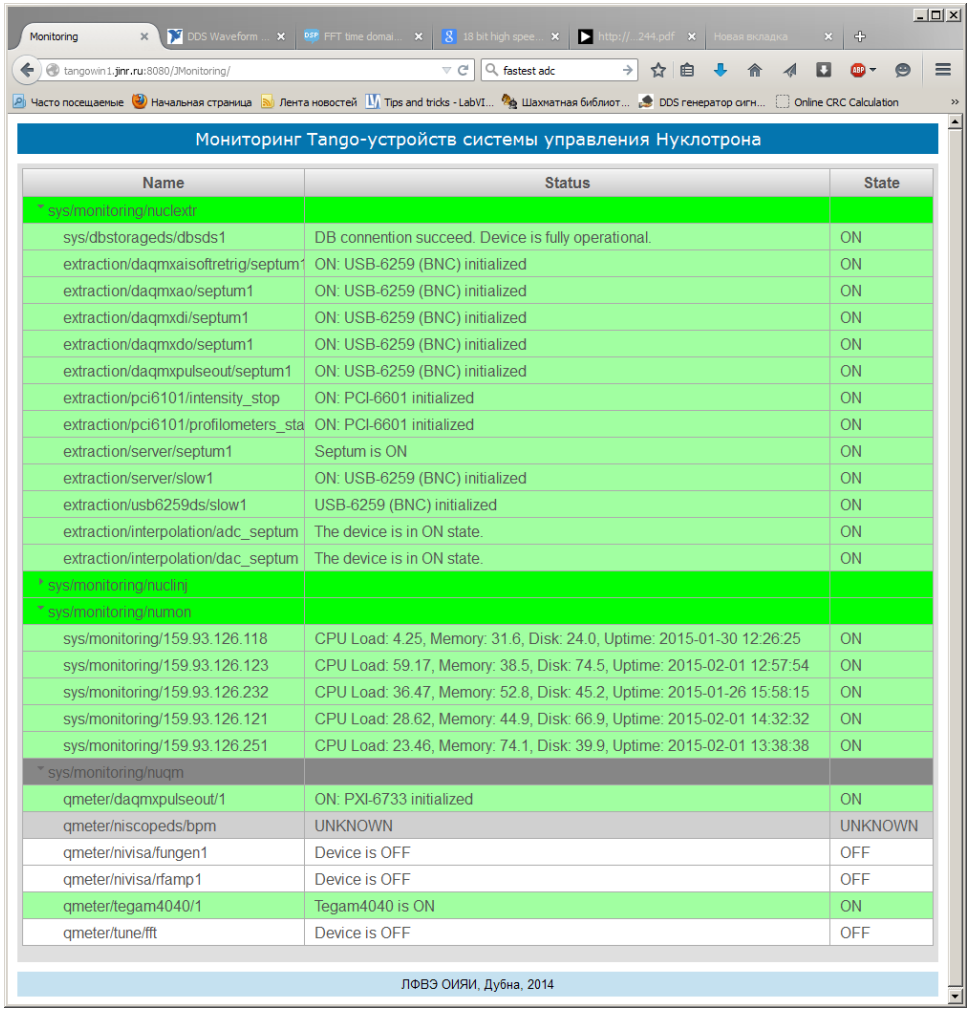
От tango@tangodevel.jinr.ru
Тема **Monitoring alert - Nuclotron qmeter**
Кому "E. Gorbachev"

Subsystem: Nuclotron qmeter
Name: qmeter/nivisa/fungen1, State: FAULT, Status: Device disconnected from USB
Name: qmeter/nivisa/rfamp1, State: FAULT, Status: error:timed out

Service layer: monitoring

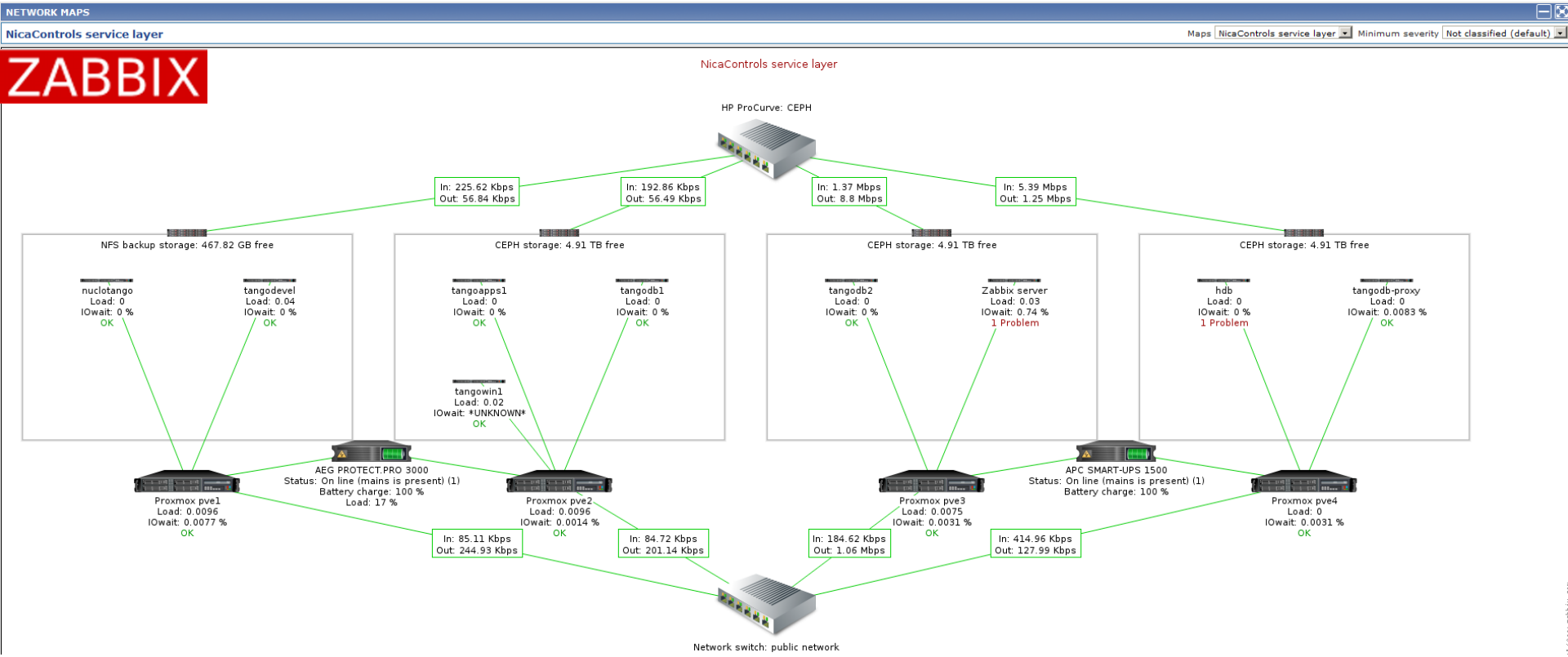
Desktop and web clients to represent states and statuses of TANGO



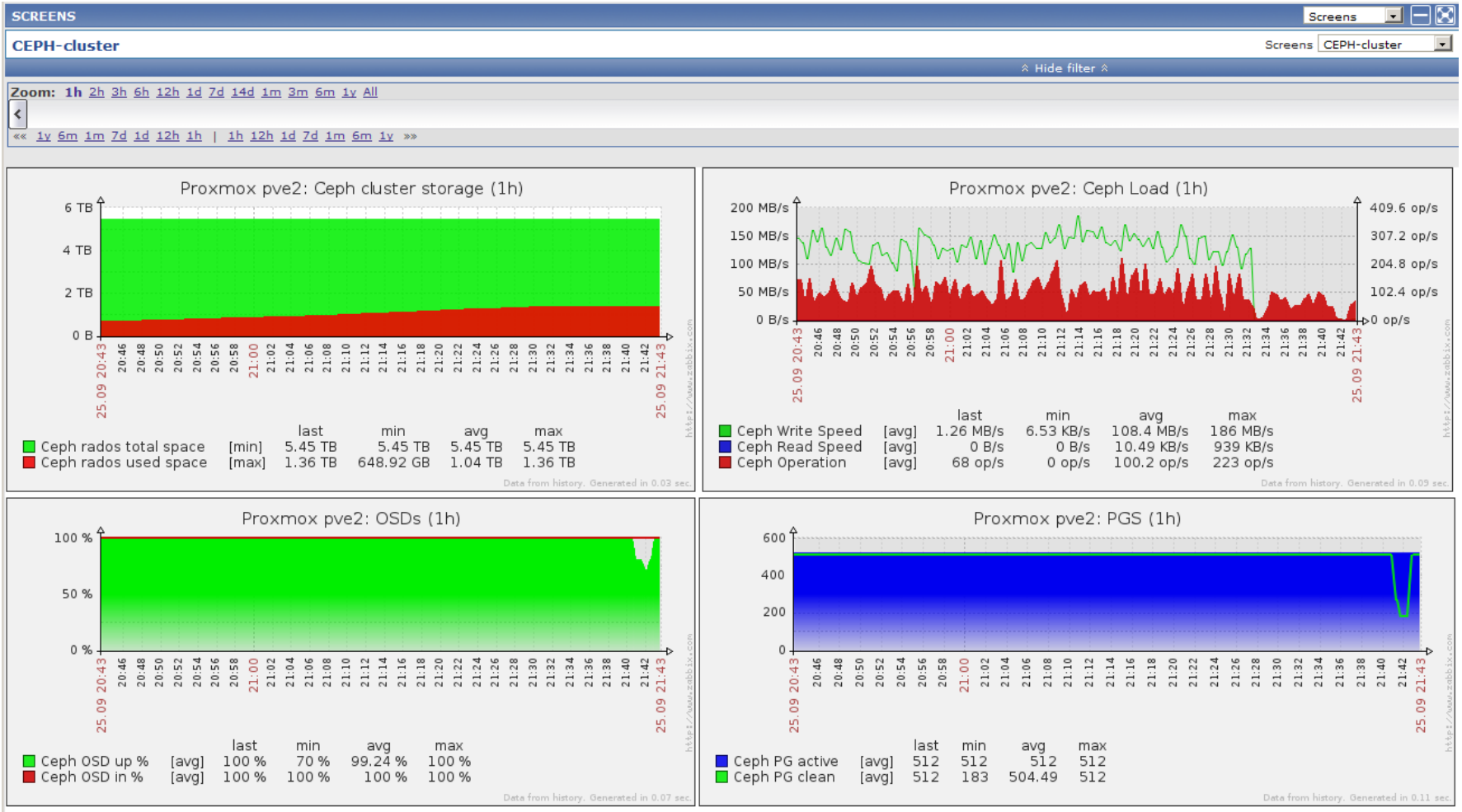


Service layer: Monitoring Proxmox

Zabbix – open-source distributed monitoring software. CPU, RAM, disk space, network traffic, disk I/O and many other items. Alerts and notifications are sent via e-mail to cluster admins.



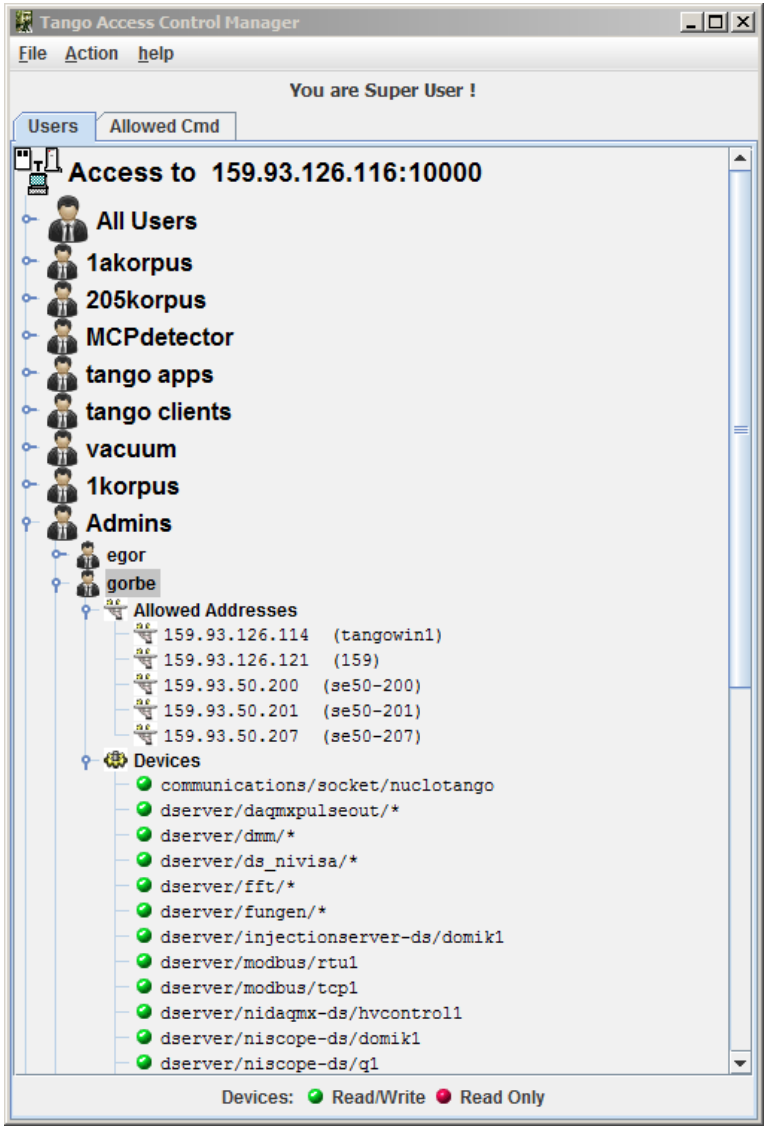
Service layer: Monitoring CEPH



Service layer: Access control

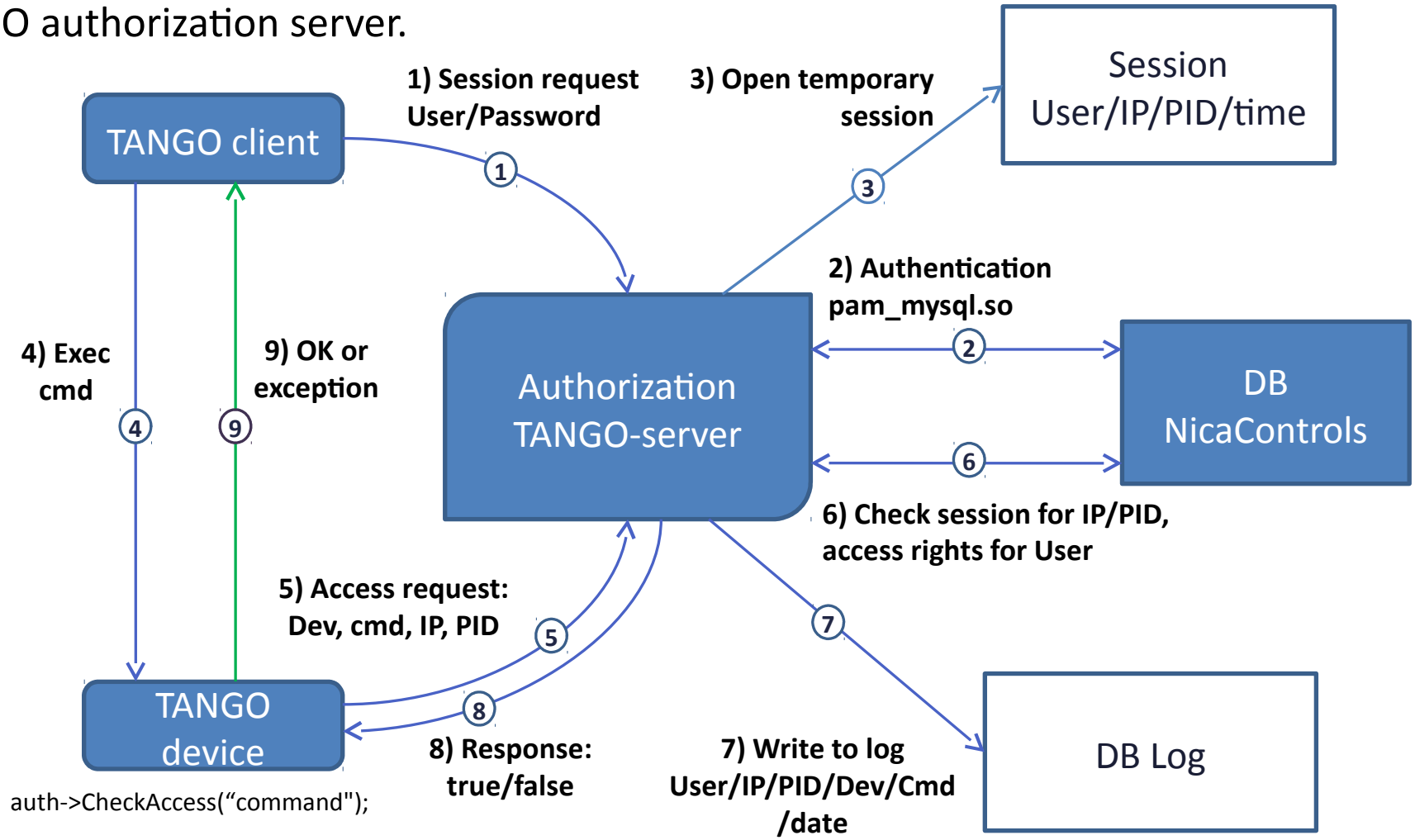
Access control and security are vital to safely run a distributed control system.

- **Network** configuration.
  - Firewall configuration to provide access for certain IP range.
  - Private sub networks.
- **Client-side** access control
  - Tango provides AccessControl service allowing access to commands and write access to attributes based on user+IP checks. Access to other devices is read-only.
  - Security checks on client side
  - Not flexible, EASY TO PASS



Service layer: Access control

We developed additional server-side authorization system using custom TANGO authorization server.



Service layer: Access control

- Flexible access tuning using mysql regexps:

device_name	username	ip	priority
training/authtest/1/On	tangotest	159\\.93\\.126\\.12	10
training/authtest/1/On	tangotest	159\\.93\\.126\\..*	10
training/authtest/1/.*	tangotest	159\\.93\\.126\\..*	10
training/authtest/.*/.*	tangotest	159\\.93\\.126\\..*	10
training/authtest/1/.*		159\\.93\\.126\\.12	10
training/authtest/1/.*		159\\.93\\.126\\..*	10

- Operator-expert restrictions:

device_name	username	ip	priority
training/authtest/1/.*		159\\.93\\.126\\.12	10
training/authtest/1/On	tangotest	159\\.93\\.126\\.12	0

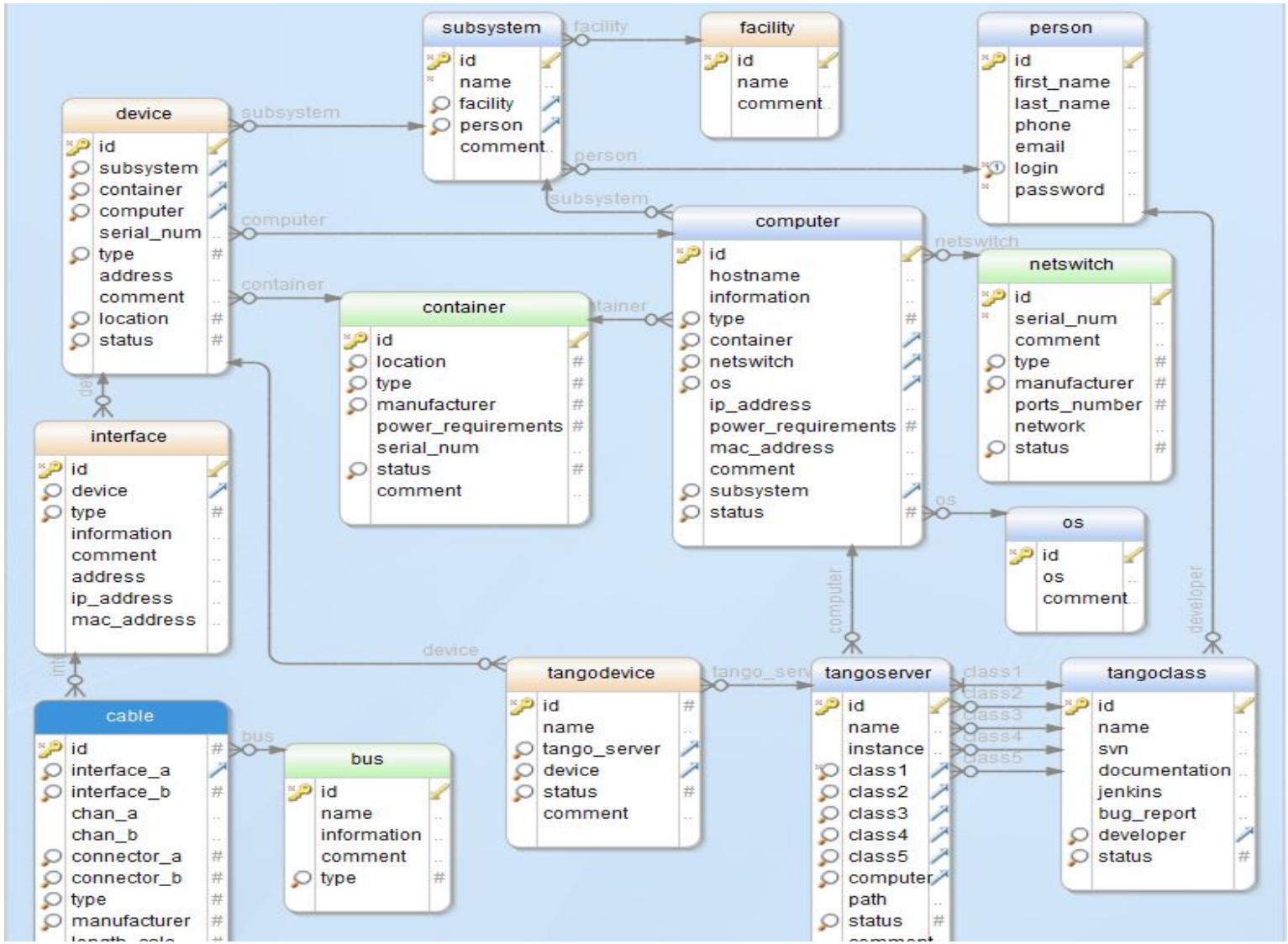
NicaControls database

NICA complex:
- LU-20
- HILAC
- Booster
- Nuclotron
- Collider ring 1,2
- Ion sources
- Beam transfer lines

Subsystems:
- Injection
- Magnetic field cycle
- Magnetic field correction
- RF
- Vacuum
- Thermometry
- Beam diagnostics
- Beam loss monitoring
- Quench protection
- Electron cooling
- Stochastic cooling
- Extraction

Global systems:
- Cryogenics
- Synchronization and timing
- Safety

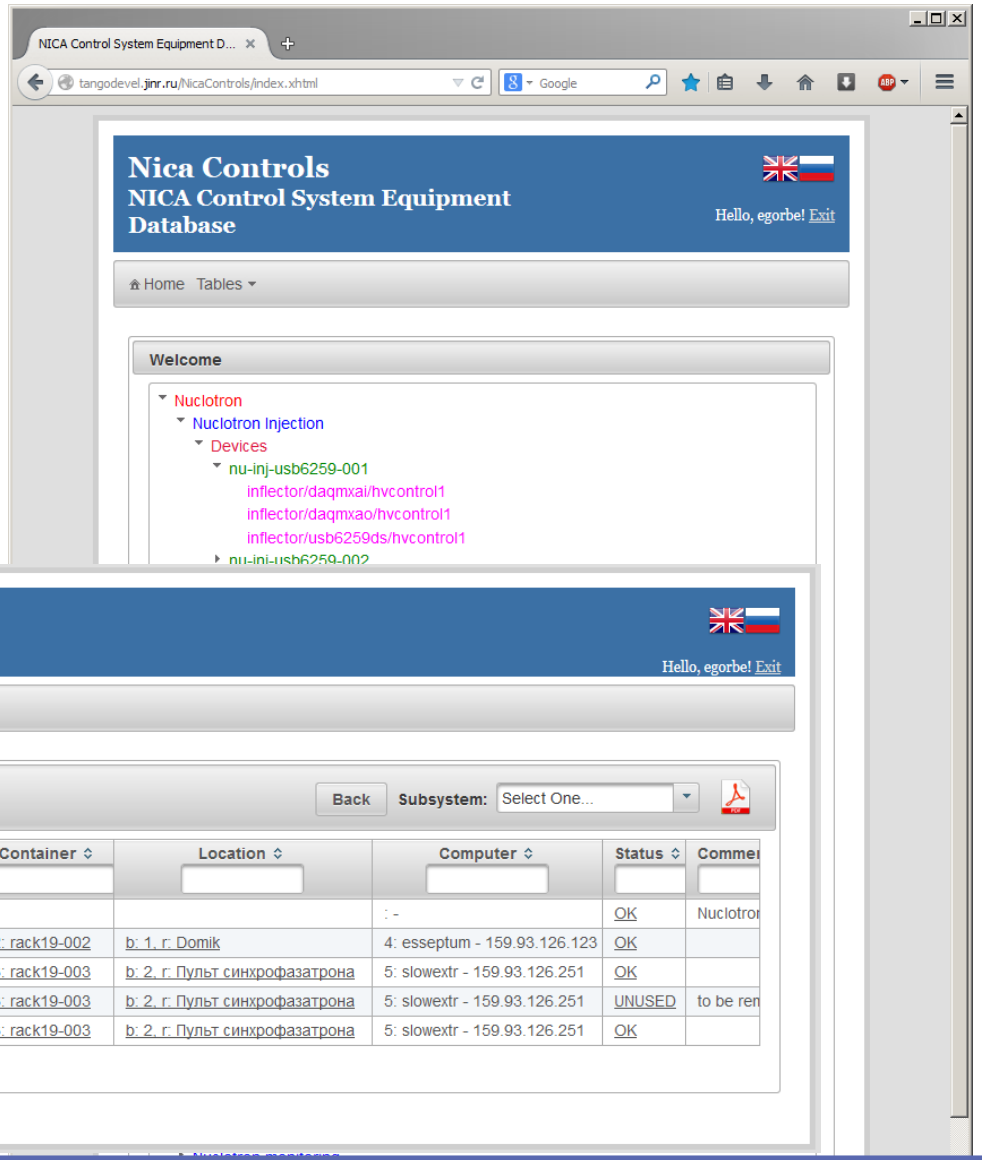
NicaControls database



NicaControls database

Database WEB-manager written on Java (JSF, JPA, PrimeFaces):

- Tree view to display subsystems hierarchy.
- Plain tables output. Sorting, filtration, navigation between tables are provided.



Tango development services

Linux and Windows development virtual machines with:

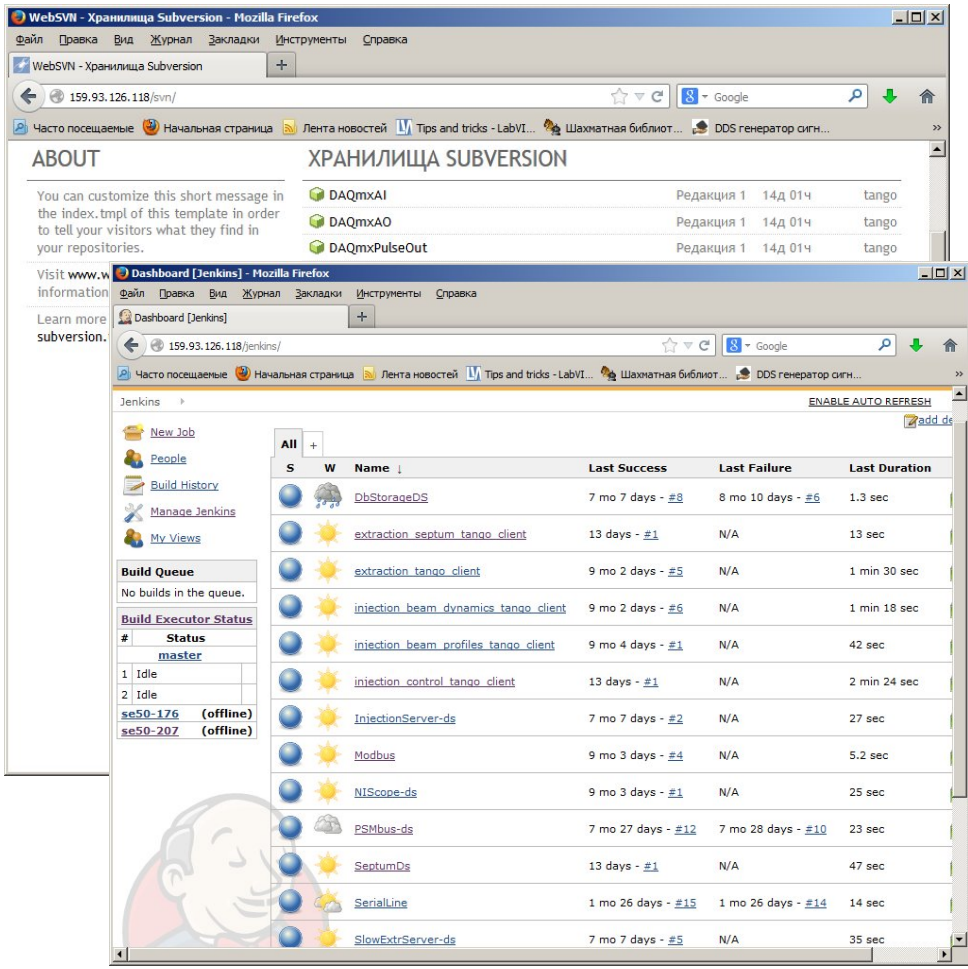
- All necessary build tools, TANGO libraries
- LabVIEW 2013-2015 with TANGO bindings
- LabVIEW FPGA module
- LabVIEW Real Time module
- MS VisualStudio 2005-2012 with TANGO libraries

Tango development services

Version control: Subversion - open source version control system.
Allows to store and access the source of all control system applications, including servers and clients.

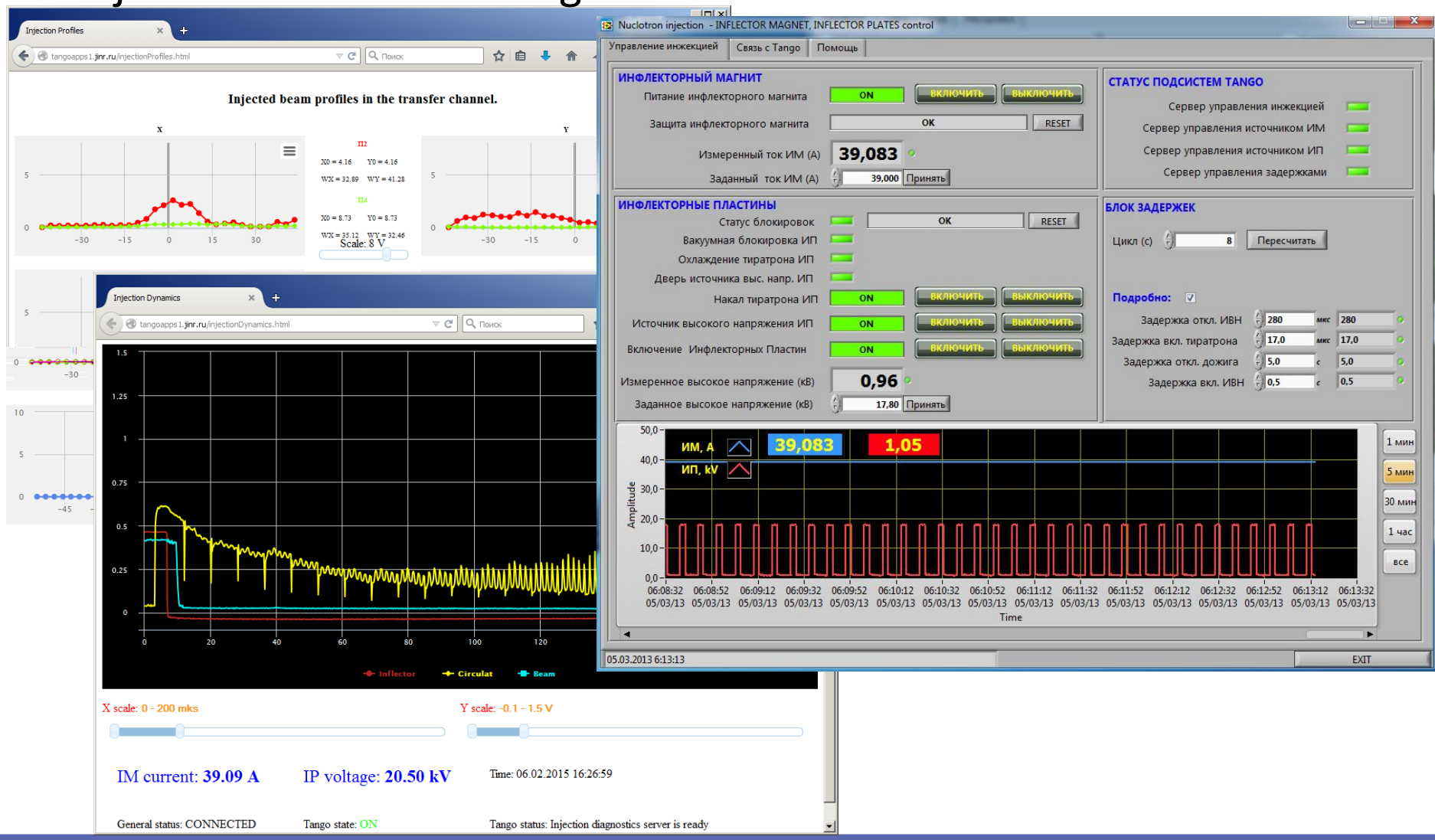
Continuous Integration: Jenkins to automatically build Windows and Linux C++ and LabVIEW applications. The build initiated by project source code change in SVN.

Documentation: wiki to make software documentation available for all users. It provides service description, individual devices docs links to source and executable.



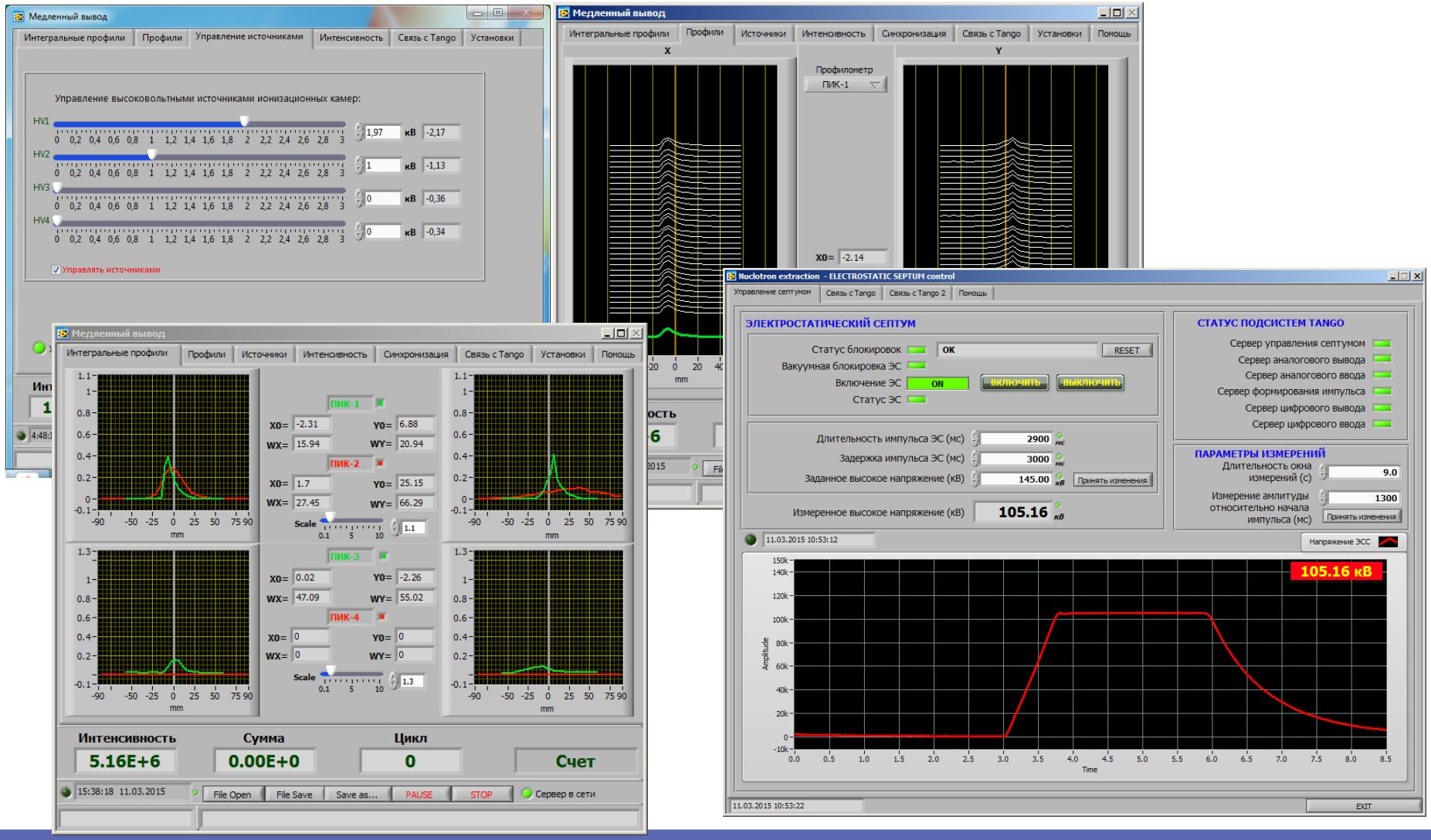
Nuclotron CS development

- Injection control and diagnostics



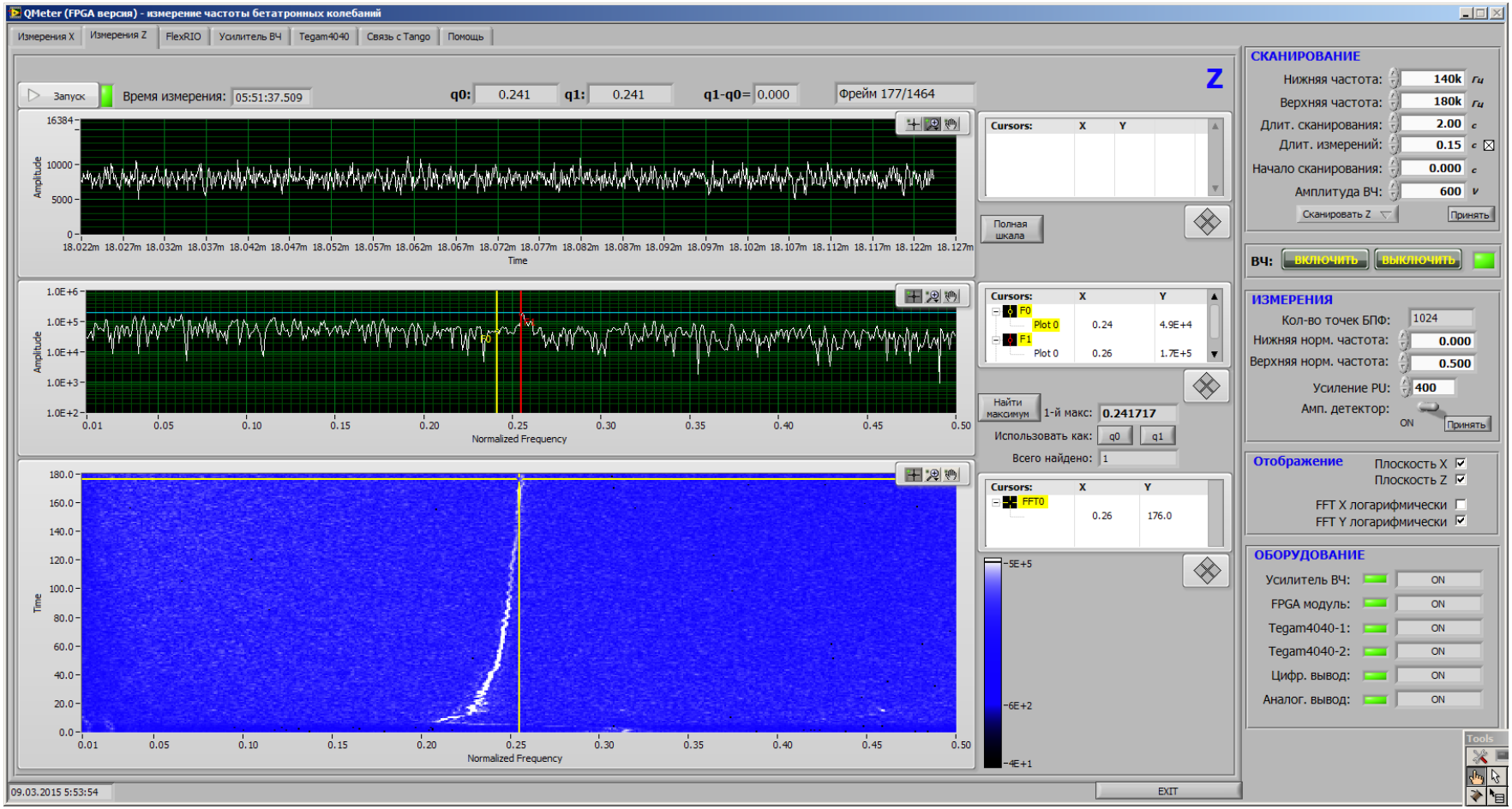
Nuclotron CS development

- Slow extraction control and diagnostics



Nuclotron CS development

- Betatron tune measurement



NICA CS components prototypes

- Booster thermo diagnostics – 80 channels PXIe based RTD measurements.
- Booster magnetic cycle control – FlexRIO + custom IO modules.
- Booster RF – integration of third-party CS.
- Booster vacuum – integration of third-party Zenon SCADA control.
- NICA injection/extraction control – CompactRIO + custom IO modules.

Conclusions

Distributed, scalable control system infrastructure based on Tango has been developed to provide fast development, deployment and safe execution: code generation, TANGO drivers for NI equipment, deployment on dedicated VM in HA cluster, various developers tools and libraries, server-based access control, hardware and software monitoring, data archiving, equipment and software database.

Future plans

- Migration to 10GbE network switch for CEPH
- Tango ver. 9:
  - Attributes forwarding
  - Complex structures attributes
- HDB++:
  - Event based archiving system
- Proxmox VE ver. 4:
  - LXC containers on local storage and NFS, CEPH RBD storage
- Continue development of Nuclotron, HILAC, Booster and collider diagnostics and control.

Thank you for your attention!