

# TANGO Device Servers in Libera instruments

*Elvis Janežič, May, 2015*



# Content of the talk

- **Libera instruments by platform**
- **Libera BASE: connectivity, interfaces, benefits**
- **TANGO Device Server by I-Tech: Configuration, Attributes**
- **Conclusion**

Basic Application  
Support Environment



# Libera instruments by platform



Instrument	Brilliance, Brilliance Single Pass	Brilliance+, Single Pass E	Spark
Op.system	Embedded Linux	Ubuntu	Xilinx Linux
Device server	Remote server*	Instrument or remote server	Instrument or remote server
Upper layer	CSPI	MCI	SCPI, MCI
Developed by	Elettra, ESRF, Soleil	I-Tech / MAX-IV	I-Tech / ESRF

\* Also in the instrument at FERMI@Elettra

# Libera BASE

- Many instruments, many people → control systems matrix → complex support
- Leverage development between different solutions
- Common building blocks (HW monitoring, data streams, FPGA artifacts)

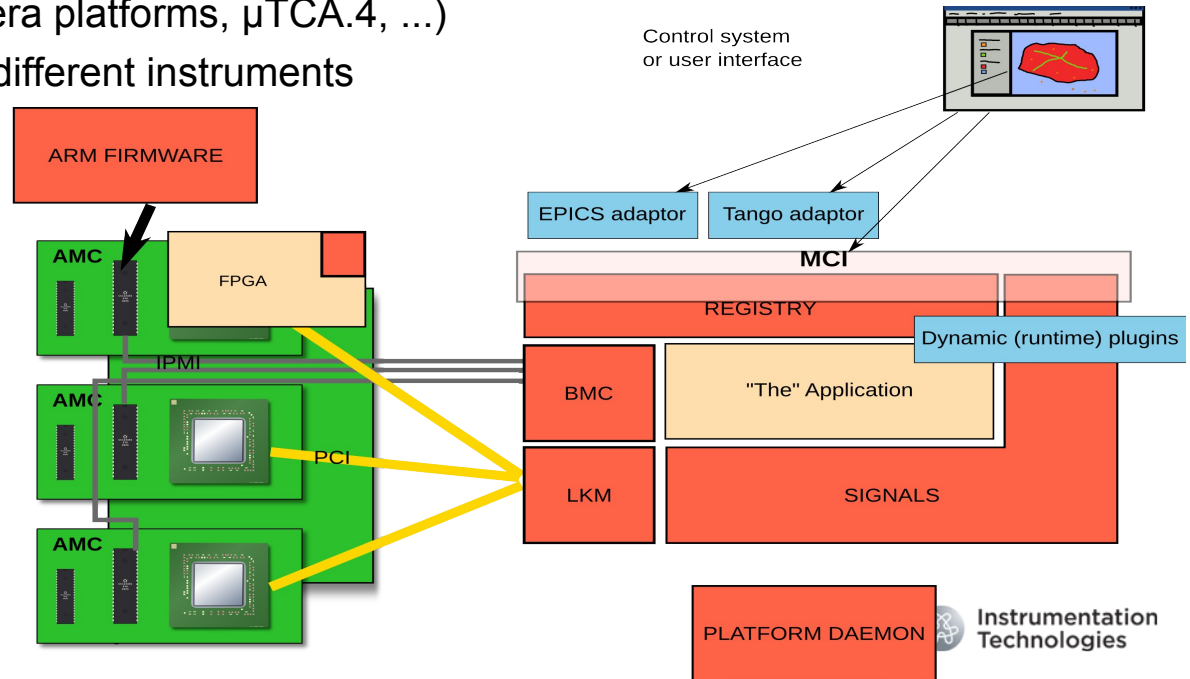
Platform / Controls	TANGO	EPICS	DOOCS/TINE	MATLAB	MADOCA	CUSTOM1
Brilliance+, Single Pass E	1	2	3	4	5	6
Spark	7	8	9	10	11	12

# Libera BASE benefits for I-Tech

- Easy start-up and control
- Rapid (software) prototyping
- Rapid application development
- Connecting the instrument with several Control Systems
- Overall system reliability assurance
- Cross-platform compatibility (x86, x86\_64, Ubuntu, Scientific Linux, Xilinx Linux)
- Available in platforms B and C

# Libera BASE connectivity

- Middle layer
- Hardware abstraction (Libera platforms,  $\mu$ TCA.4, ...)
- Common denominator for different instruments
- Development framework



# Libera BASE interfaces

- I-reg – registry structure (XML)
- Platform management based on IPMI
- MCI – external API
  - List / dump all registry nodes
  - Info – attributes
  - Get / Set – value access
  - Listen – notification
  - Acquire – stream and data on demand

```
// Get reference to the SA signal from the registry
ireg::Node root = GetRoot();
ireg::Node sa_signal = root.GetNode(("boards", "raf", "signals", "sa"));

// Acquire the signal (1024 samples)
isig::SignalSharedPtr signal =
ireg::SignalNode::CreateRemoteSignal(sa_signal);
typedef isig::RemoteStream<isig::SignalTraitsVarInt16> SaStream; SaStream *strm = dynamic_cast<SaStream*>
(signal.get());

SaStream::Client cl(strm, "my-client");
SaStream::Buffer a(cl.NewBuffer(1024));

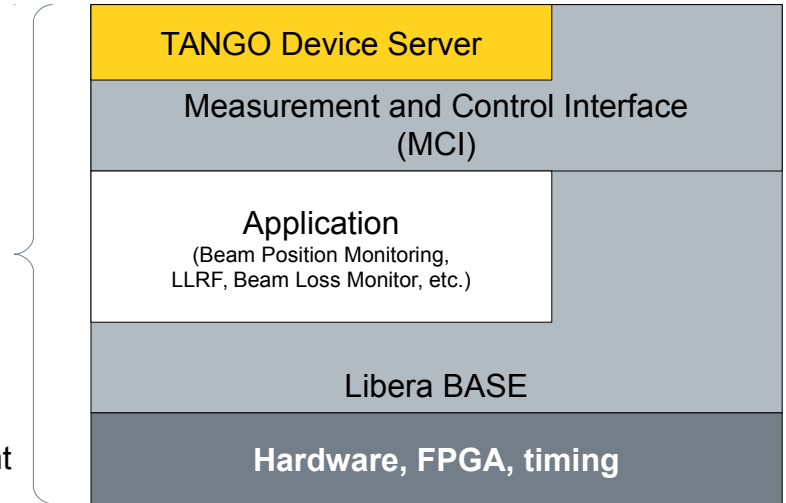
isig::SuccessCode ret = cl.Read(a);

// and dump the acquired signal
for(auto i = 0; i < a.GetLength(); i++) {
    for (size_t j = 0; j < a[i].GetComponents(); ++j)
        std::cout << setw(11) << a[i][j] << " ";
    std::cout << std::endl;
}
```

# Libera BASE benefits for users

- Same upper layer software interface (MCI)
- Generic TANGO DS code for application-specific instruments
- Network transparent

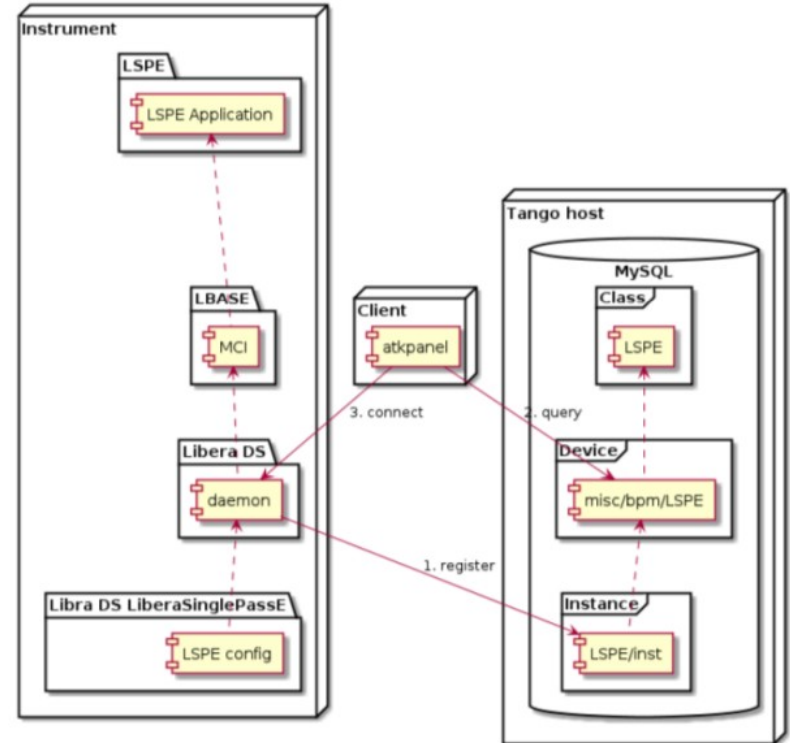
Libera instrument





# TANGO Device Server architecture (I-Tech)

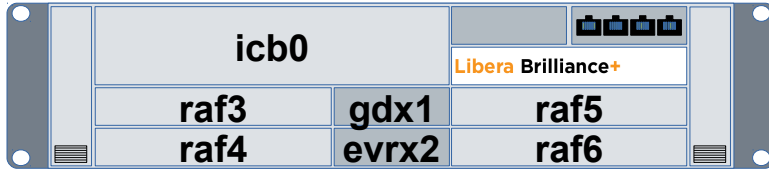
- 1) Device Server connects to Libera application via MCI layer (inside the Instrument)
- 2) Device Server registers to Database (TANGO host, right side) and accepts client connections
- 3) Configuration (attributes) is loaded from text files



# Configuration

Specified in text files, two levels: **top** / **sub-tree**

**top**: includes the path to modules (bpm, timing, gdx)



**sub-tree**: includes the relative path to parameters/signals

```
root@libera:~# cat /var/opt/libera/cfg/tango/ds-top
app-ebpm application. App
board-gdx boards.gdx1. Gdx
board-evrx boards.evr2. Evrx
board-raf $INSTANCE Bpm
```

```
root@libera:~# cat /var/opt/libera/cfg/tango/board-raf
info.revision InfoRevision 34 3 1
info.customer InfoCustomer 34 8 1
info.health_status InfoHealth_status 34 3 1
.
.
.
signals.sa SignalsSaEnable -1 6 16
signals.sa SignalsSaVa 1026 6 16
signals.sa SignalsSaVb 1026 6 16
```

# Attributes

## APPLICATION SPECIFIC ATTRIBUTES

- app-ebpm
- board-evrx
- board-raf
- board-gdx

## PLATFORM SPECIFIC ATTRIBUTES

- pm-evrx
- pm-fans
- pm-icb
- pm-os
- pm-raf

## MCI flavors (to TANGO)

- single or array value nodes (parameters)
- streamed data or data on demand signal nodes (data)
- executable nodes

# Conclusion

- TANGO Device Server is available for all Platform B Libera instruments (e.g. Libera Brilliance+, Single Pass E, etc.) from early 2015
- Same code will be reused for platform C instruments (Spark, Beam Loss Monitor, Digitizer, Photon, etc.)
- Tested with ATK, PyTango
- User-friendly Graphical User Interface still has to be developed (Qt?)