

A Light for Science



European Synchrotron Radiation Facility

# TANGO kernel status and evolution

- Brief introduction
- What's new since Icalepcs 2007
- New projects
- Re-thinking the Tango event system

# What is Tango ?

- An object oriented control system based on CORBA
- Device supports commands (actions) and attributes (data)
- 3 types of communication: Synchronous, Asynchronous and Event driven
- 3 languages: C++, Python and Java
- Collaboration between several institutes



## What's new since 2007 ?

- 3 kernel libraries releases
  - Tango 6.1 with special emphasis on configuration database load
  - Major release Tango 7
    - IDL union instead of CORBA Any
    - Optimized JPEG encoder/decoder
    - Event queues to improve decoupling between event publishers and subscribers
    - Thread pool instead of a unique thread to do internal device polling
    - Device locking
    - Automatic discovery of device hierarchy
  - Tango 7.1
    - Unix sockets for communication between process on the same host (Unix only)
    - Bug fixes

## What's new since 2007 ?

- Languages

- Python binding has been optimized
- Posters THP 016 and THP 079 (Thursday)



- Graphical User Interfaces

- A new Python GUI for Tango: TAU (TAnGo User interface)
  - Based on PyQt 4
- The C++ GUI (QTango) is now in its release 3 (also based on Qt 4)
  - Poster THP 096 (Thursday)
- New widgets added to the Java GUI (ATK)



## What's new since 2007 ?

- JDDD (Java Doocs Data Display)
  - Interactive panel builder developed at Desy
  - Now support the Tango ATK widget set
  - Data source for ATK widget selected via a Tango device/attribute tree
  - No coding application
  - Poster TUP 035 (Tuesday)
- Tango WEB protocol
  - Based on JBoss
  - Uses Java Web Start to download the application on user host
  - ATK based application directly usable on the WEB



## On-Going projects

- Tango Debian packaging
  - Install Tango core in a few clicks
- Tango device server in Java
  - Re-use the C++ layer using a JNI layer
- Re-write of our code generator
  - Use Xtext/Xpand and OpenArchitectureWare to generate Tango classes
  - Poster THP 080 (thursday)

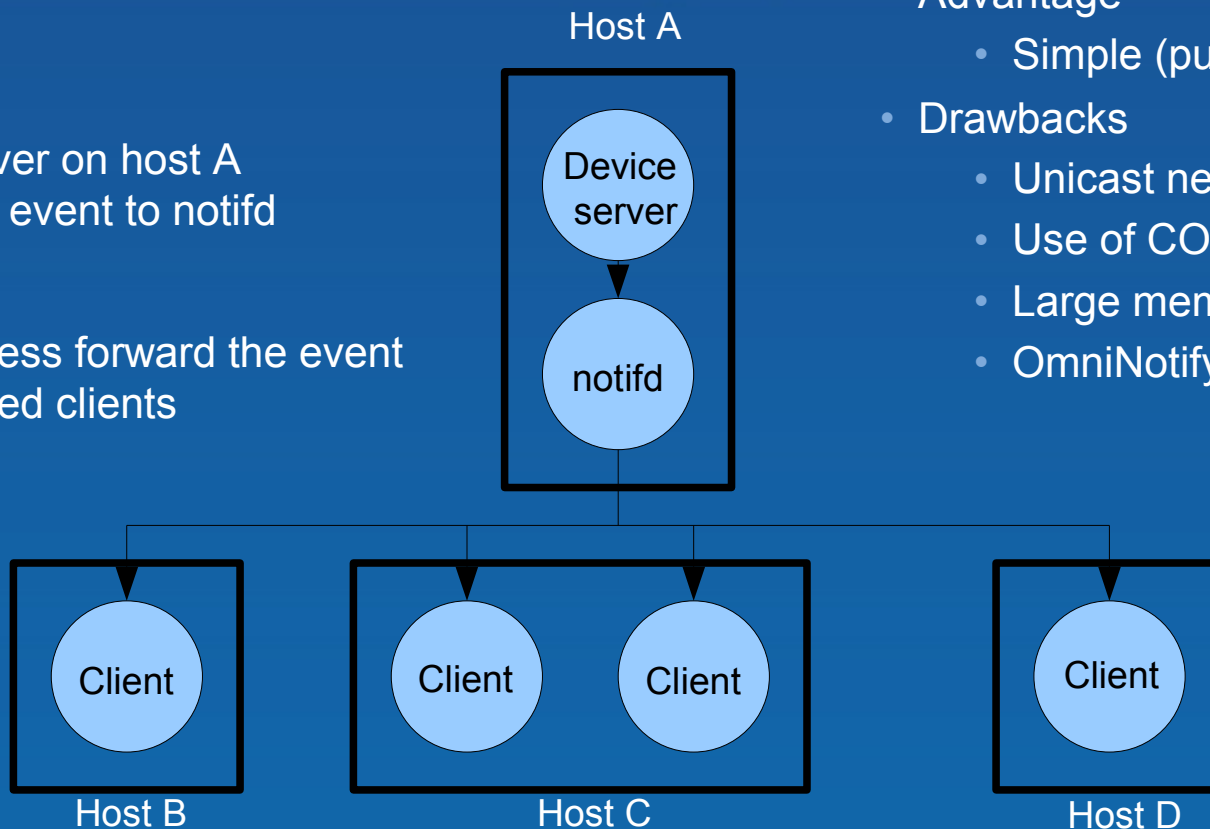


# Re-thinking the event system

- Today based on the CORBA Notification service (omniNotify implementation)

Device server on host A pushes the event to notifd

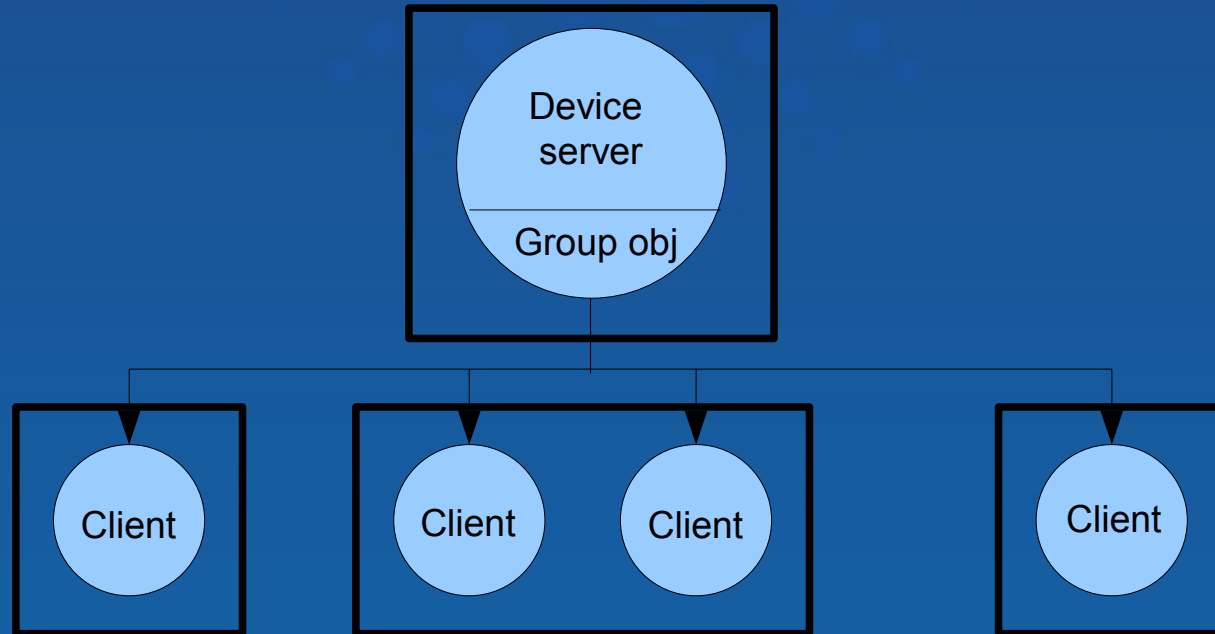
Notifd process forward the event to subscribed clients



- Advantage
  - Simple (publisher side)
- Drawbacks
  - Unicast network transfer
  - Use of CORBA Any's
  - Large memory consumption
  - OmniNotify is a dead project



# Re-thinking the event system (Group)



- Advantages

- Simple
- No dependency

- Drawbacks

- Still unicast
- Code to be written
  - (group members mgnt)

## Re-thinking the event system (DDS)



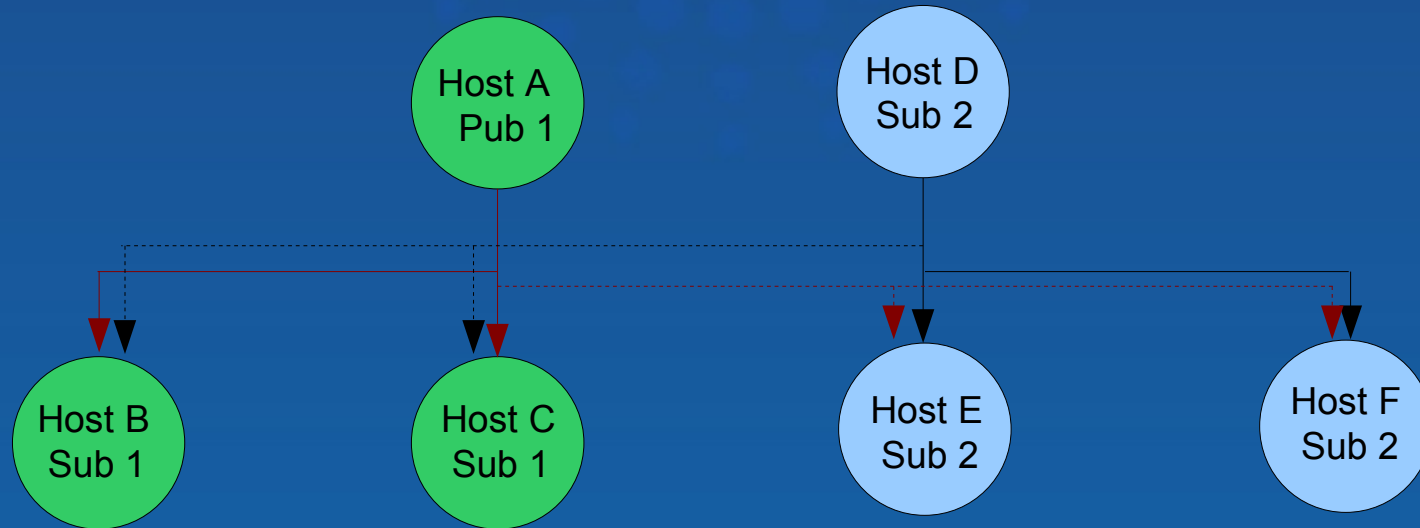
- Data Distribution Service is an OMG specification
- Publisher/Subscriber communication system
- Large number of Quality of Services (QoS) to tune the communication between pub/sub
- Standardized protocol RTPS (Real Time Publish Subscribe) to allow DDS Interoperability (DDSI)
  - Able to run over **multicast** and connection transport like UDP/IP

## Re-thinking the event system (DDS)

- OpenSplice implementation from PrismTech company
  - LGPL licensed software with
    - RTPS, their own network protocol (multicast enable), DCPS
  
- Preliminary test
  - Too many quality of services ?
    - Seven QoS had to be tuned
      - Reliability, Deadline, Liveliness, History, ResourceLimit, Destination order, Partition
  - Spreading the events into multicast group (address)
    - Find a way to automatically distribute the event on the available multicast group



# Re-thinking the event system (DDS)



Host B and C are "polluted" by event 2

Host E and F are "polluted" by event 1

OpenSplice DDS is able to manage several multicast address using Partitioning  
The problem is reduced but not solved

# Re-thinking the event system (DDS)

- DDS advantages
  - Performance (mainly when the numbers of subscribers increase)
  - QoS (highly tunable)
- Drawbacks
  - Multicast address selection
  - QoS (too many)
  - RTPS not usable (PrismTech implementation)
  - Extra processes (2) and shared memory segment on each host (PrismTech implementation)

## Conclusion

- Tango Event system
  - No conclusion yet
  - Decision in next Tango meeting (Winter/Spring 2010)
  - First implementation for Summer 2010
  
- Tango
  - It's still an evolving project
  - Problem is not the lack of ideas but rather a lack of resources