# Why TANGO ?

- That is a very good question !
- We have very good answers

# First ask yourself this question :

- If you were building a new control system today and you had to choose between RPC and CORBA which one would you choose ?

# In the beginning …

- There was M.Baudot who invented the serial line (1894 !)
- Then there was M.Klotz who invented the home made parallel network (mid 1970)
- Then there was Ethernet (late 1970)
- Then there were Berkeley sockets (early 1980)
- Then there were remote procedure calls (late 1980)
- Then there was **CORBA** (mid 1990)

# Everytime we adapt … and improve …

- EPICS is based on sockets
- TACO is based on RPC
- TANGO is based on CORBA

- Unfortunately M.Baudot is still with us !

# TACO has reached maturity

- It offers the minimum number of features an object oriented control system should have, it is fast, simple and easy to use
- BUT TACO is not perfect …
- Mistakes were made, certain things are not as simple as we would like them, some features are missing, we have new better ideas

# The motivation for TANGO

- The motivation for TANGO is to use the experience of TACO to build a better control system using a modern protocol
- TANGO will be fully compatible with TACO in order to profit from the big code base of TACO
- We have a running system therefore we have time to develop (in theory)

# Why CORBA

- CORBA is the obvious choice today
  - It is object oriented
  - It is widely implemented
  - It is efficient + free
  - It has been standardised
  - It is language independent
  - It has bindings for C++, Java, C, Python,
  - Control systems using it are ACS, ICCS, GNOME Windowing system …

# The Choices for TANGO

- The WEB
  - Access from web browser
  - SOAP to CORBA gateway
- C++
  - Use patterns
  - Provide an object oriented api
- Java support

# The Improvements to TACO

- Automatic data caching
- Better thread support
- System generated events
- Self-describing data types
- Easy system administration
- Automated startup procedure
- Full support for Windows

# The Big Debate … Generic vs. Specific

- Generic
  - is a minimum requirement
  - allows writing simple generic clients
  - supports hooking and checking
- Specific
  - general case of generic
  - allows compile time checking
  - is more object oriented
  - is a source of long discussions

- TANGO proposes to implement specific interfaces on the client side