



Instrumentation
Technologies

www.i-tech.si

Aleš Bardorfer

CSPI: Integrating Libera into Control System(s)

ales@i-tech.si, tomaz.beltram@i-tech.si

Libera **WORKSHOP**
2007

24-25 September 2007

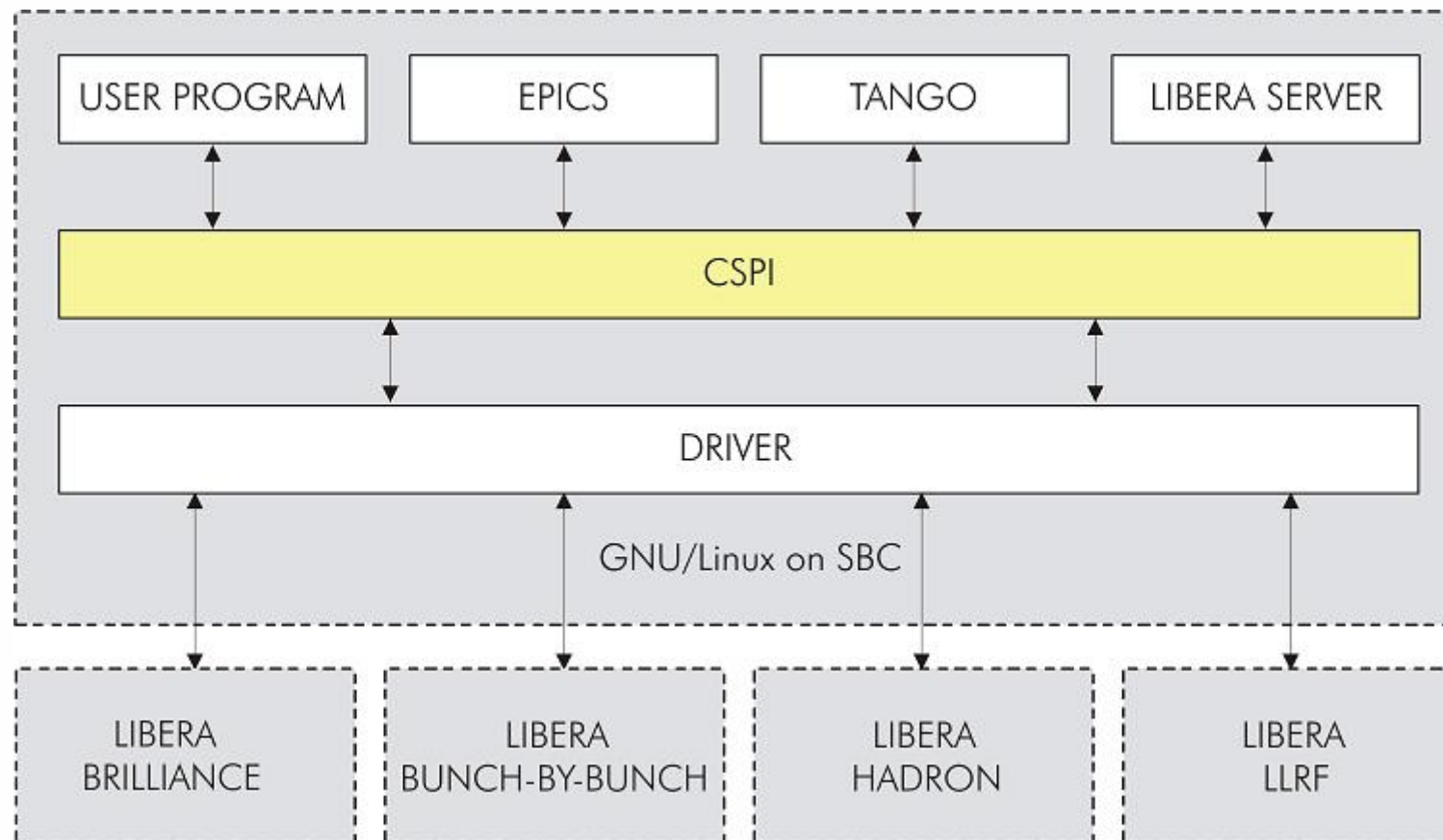
Introduction

- **What CSPI is and what it isn't**
- **The role of CSPI**
- **The structure of CSPI**
- **CSPI as an API for integrating into control system(s)**
 - **EPICS**
 - **Tango**
 - **Generic server**

What is CSPI?

- **CSPI – Control System Programming Interface**
- **High-level C interface**
- **Library to be linked with user application**
- **Standard Libera API**
 - Long term API compatibility
 - Hides HW details and extracts data paths
- **Not a Control System by itself**

CSPI Structure



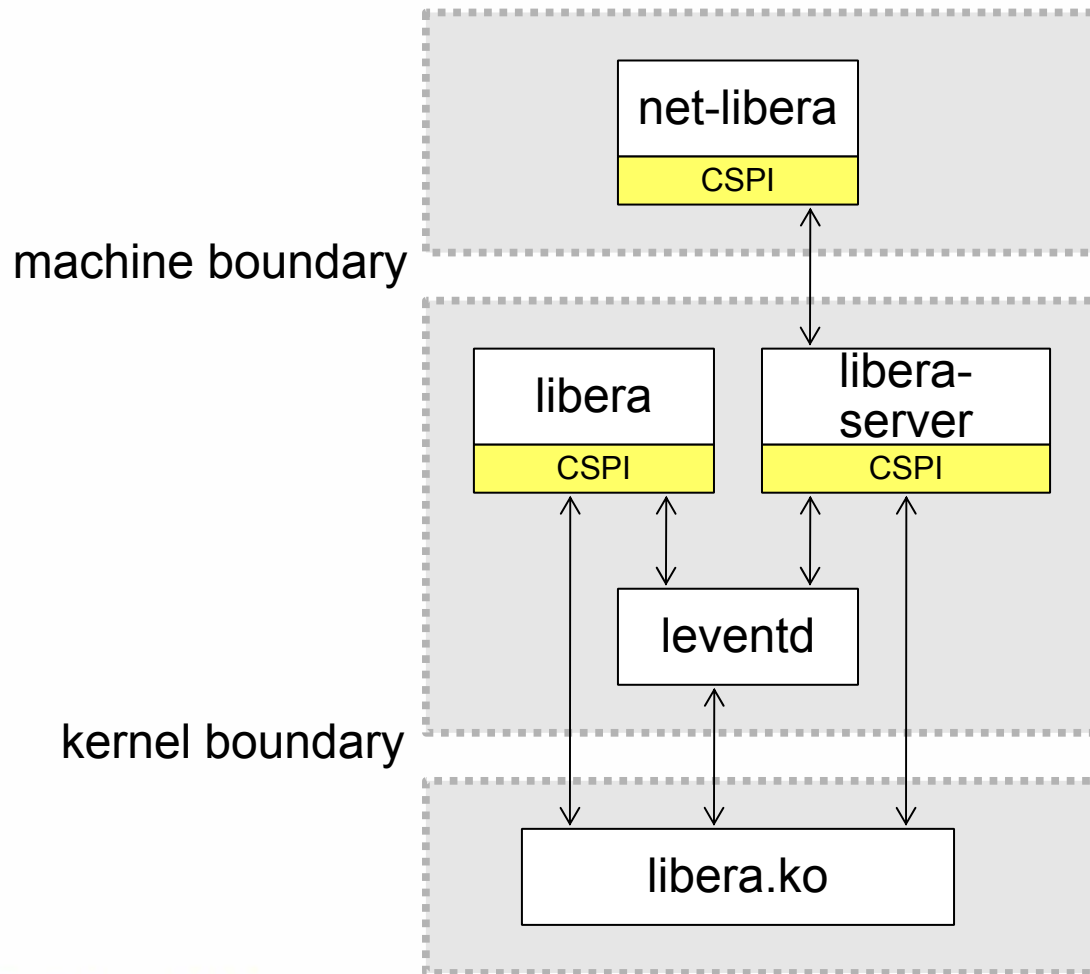
CSPI and Libera Family

- **Single API interface for all members of the Libera family**
 - Libera Electron & Libera Brilliance
 - Libera Bunch-by-Bunch
 - Libera Hadron
 - Libera Low Level RF
- **Differences only in handle and connection parameters & structures:**
 - Maximum interoperability
 - Familiar same API for all Libera members

CSPI Roles

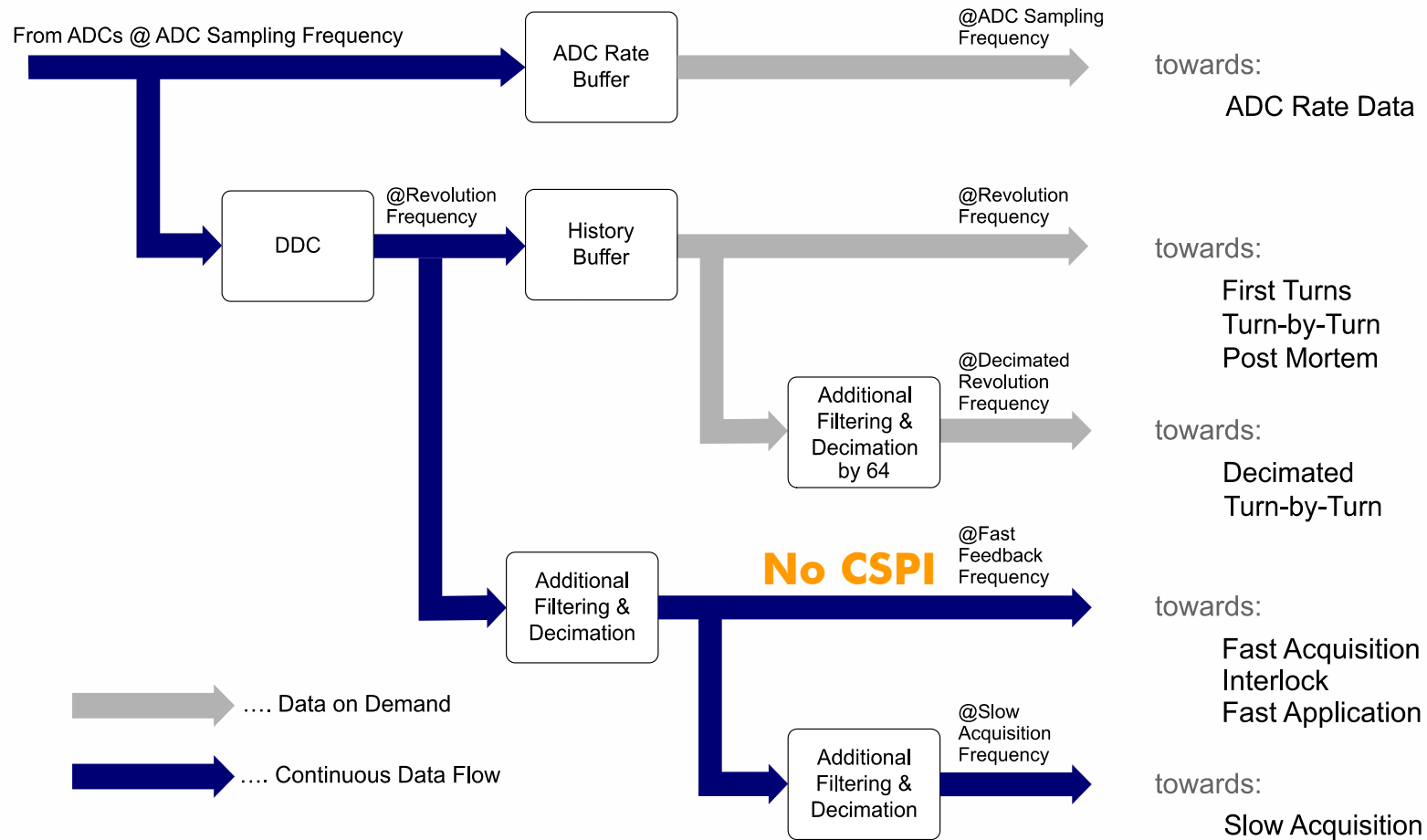
- **Libera as a standard API**
 - **Relatively constant API**
 - **Covers a whole range of OS platforms**
- **A Shield for the user:**
 - **“Proper” use of Libera**
 - **Prevents from Libera use scenarios that don’t make any sense**
 - **Multi-user safe**
 - **Multi-application safe**

High Level Architecture ^{7/18}



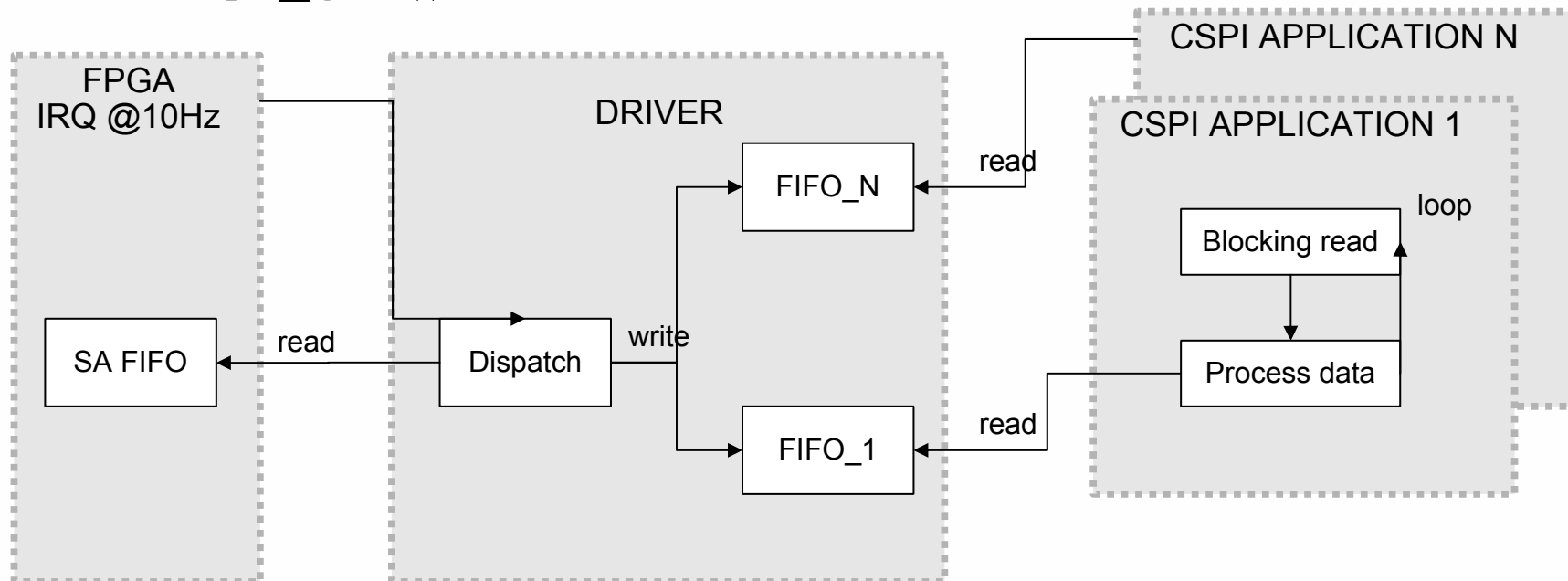
- **libera utility**
- **CSPI library**
- **event daemon**
- **kernel module**

Libera Data Sources

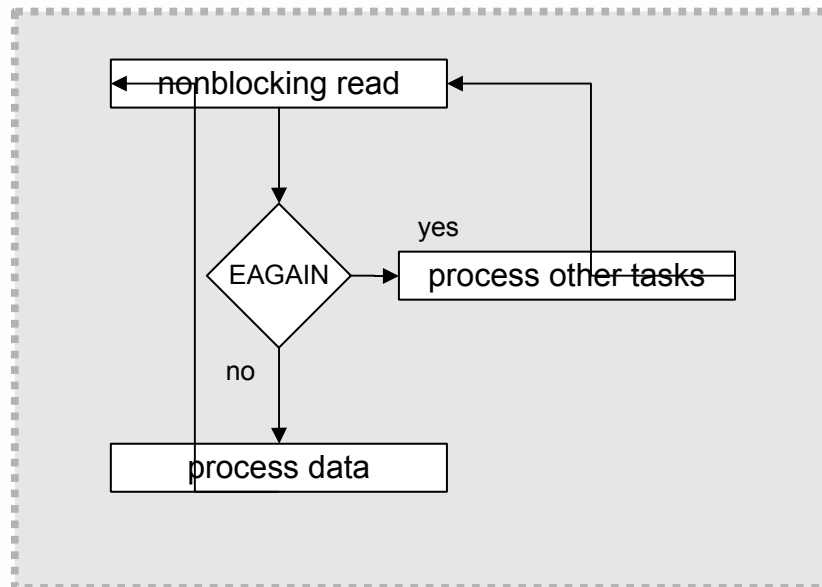


Streaming Data Paths

- **Slow acquisition (SA)**
- **Data delivery frequency dictated by FPGA (synchronous to TbT)**
- **`cspi_get()`**



Nonblocking read

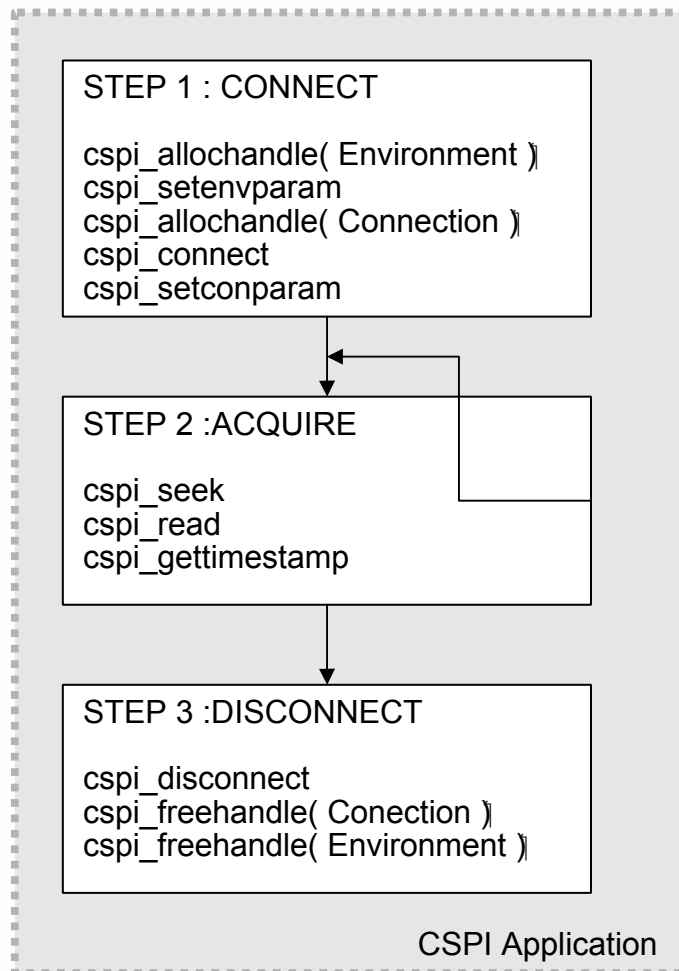


- **read() returns EAGAIN**
- **Use poll() on a file descriptor or select() on file descriptor set**

Non-Streaming Data Paths

- **Data on demand (DD)**
 - **ADC rate buffer (ADC)**
 - **Post mortem (PM)**
-
- **Data delivery frequency dictated by CSPI user application**

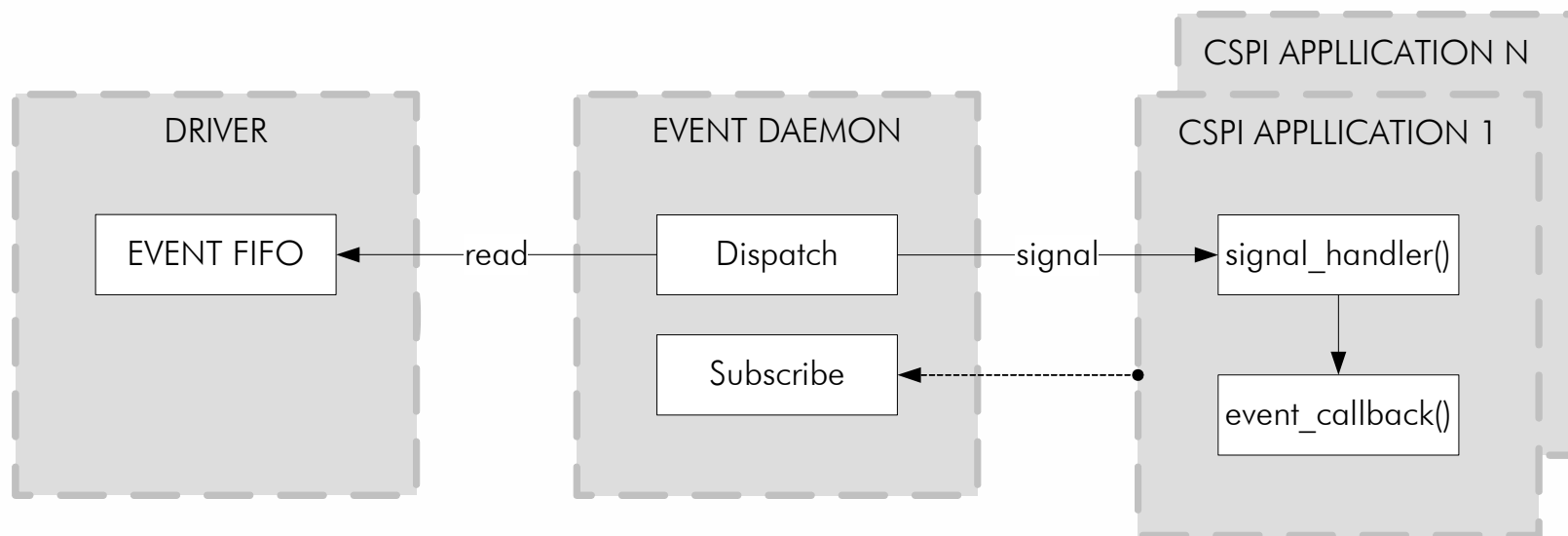
General CSPI Application Flow



- **Handles:**
 - **Environment**
 - **Connection**
- **Parameters**
- **Data retrieval functions:**
 - **`cspi_seek()`**
 - **`cspi_read()`**
 - **`cspi_get()`**

Async Notification Mechanism

CFG, OVERFLOW(s), SA, TRIGGERS, PM...



Discussion

- **Event notification:**
 - Linux signals
 - Message pumps through FIFOs
- **Default run-time parameter values**
 - /etc/default/libera
- **Scaling factors**
 - $K_x, K_y \rightarrow K_a, K_b, K_c, K_d$
 - API change

Integration into Control Systems

15/18

- **The challenge:**
 - **Many different Control Systems available**
 - **Inherent differences in CS structures, paradigms, mechanisms.**
- **The solution:**
 - **CSPI aims to fit in and cover all the CS specifics**
 - **CSPI is generic in it's fundamental design**

Existing Control System Integrations

16/18

- **EPICS**
 - **DLS driver (lot of additions to libera functionality)**
 - **Instrumentation Technologies driver (basic, lightweight)**
- **TANGO**
 - **First generation based on Generic Server**
 - **Second generation runs on Libera, ported to Libera by Soleil, Elettra, ...**
- **Generic Server**
 - **Used with Tango, Matlab, LabView (Solaris @ infn-Inf)**

Integration into Other Control Systems

- **On Libera (ARM):**
 - **Development toolchain available**
 - **Linux 2.4.21 & 2.6.20.14 on Debian**
 - **Moderate resources available on Libera:**
 - **400 MHz**
 - **64 MB RAM**
 - **32 MB flash**
- **OS independent, (Generic server & client lib):**
 - **A UNIX based OS, possible Windows support**
 - **Development toolchain (gcc, ...)**
 - **POSIX thread dependency**

Conclusion

- **Libera as a network attached instrument**
- **CSPI: Standard access to Libera**
- **Integration:**
 - **Local (ARM)**
 - **Generic server & client lib**
- **Simple deployment and upgrades with deb packages**
- **GPL – Full open source code eases integration**